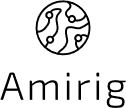
**Руководство по установке**

**системы DCDM**



|  |  |  |  |
| --- | --- | --- | --- |
| **Разработано** | **Редакция** | **Согласовано** | Версия – v.1.2 |
| Стариков П. | Сирачков Ф.А. | Маслянинов В.П. | Дата выпуска  09.09.2020 |

Оглавление

[**1.** **Подготовка рабочего места Системы DCDM** 3](#_Toc47957331)

[1.1. Конфигурация персонального компьютера 3](#_Toc47957332)

[1.2. Стандартные требования к системе (оборудованию) 3](#_Toc47957333)

[1.3. Конфигурация программного обеспечения 3](#_Toc47957334)

[**2.** **Порядок установки программного обеспечения** 4](#_Toc47957335)

[2.1. Подготовка рабочего места 4](#_Toc47957336)

[2.1.1. Установка Node.js 4](#_Toc47957337)

[2.1.2. Установка PostgreSQL 5](#_Toc47957338)

[2.2. Создание пользователя и базы данных 10](#_Toc47957339)

[2.3. Наполнение базы данных 15](#_Toc47957340)

[2.4. Настройка логической репликации Системы DCDM 90](#_Toc47957341)

[2.5. Установка Системы DCDM 93](#_Toc47957342)

[2.6. Развертывание системы поиска DCDM 95](#_Toc47957343)

[**3.** **Вход в систему и начало работы** 97](#_Toc47957344)

[3.1. Авторизация в Системе DCDM 97](#_Toc47957345)

1. **Подготовка рабочего места Системы DCDM**
   1. **Конфигурация персонального компьютера**

Система DCDM устанавливается на персональный компьютер под управлением ОС Windows версии 10Pro.

В настоящем документе рассматривается «стандартная» конфигурация персонального компьютера, обеспечивающая следующие условия:

| **Наименование** | **Кол-во параметров** |
| --- | --- |
| Запись (логирование) параметров | 500 параметров |
| Получение информации от сервера (виджеты) | 300 параметров |
| Запуск сторонних (не относящихся системе DCDM) программ и сервисов | Не допускается |

Стандартные условия подразумевают, что система обслуживает определенное количество логируемых переменных, определенное количество подключённых клиентов для «чтения» данных, наличие определенного количества репликаций, а также отсутствие исполнения других (не относящихся системе DCDM) программ и сервисов.

* 1. **Стандартные требования к системе (оборудованию)**

| **Наименование** | **Кол-во параметров** |
| --- | --- |
| CPU | CPU Intel(R) Core(TM) i7, 8 логич. процессоров |
| OZU | 16Gb • DDR4 |
| Дисковый накопитель OS | SSD 120Gb |
| Дисковый накопитель BD | 1 ТБ, 7200 RPM, SATA-III |
| Операционная система | Windows 10 Pro |
| База данных | PostgreSQL от 11 FREE и выше |

* 1. **Конфигурация программного обеспечения**

В настоящем документе рассматривается список программного обеспечения необходимого для работы системы.

Список программного обеспечения:

* Cреда исполнения Node.js
* Система DCDM
* PostgreSQL
* Система поиска DCDM

1. **Порядок установки программного обеспечения**

Ниже представлен порядок установки программного обеспечения необходимого для работы Системы DCDM.

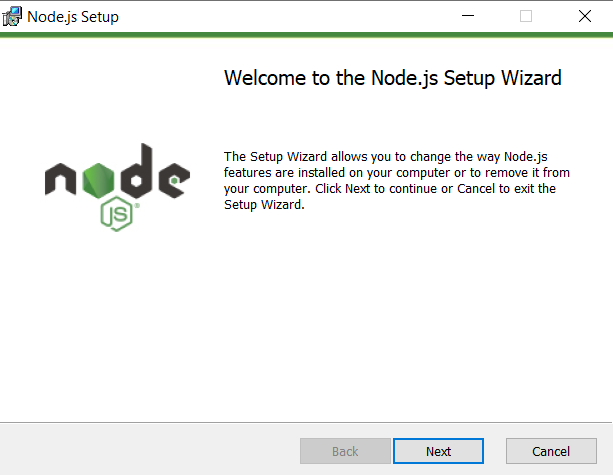
* 1. **Подготовка рабочего места**
     1. **Установка Node.js**

Загрузите Node.js для Windows по ссылке: <https://nodejs.org/en/>

На момент написания инструкции актуальная версия: **node-v12.16.3**

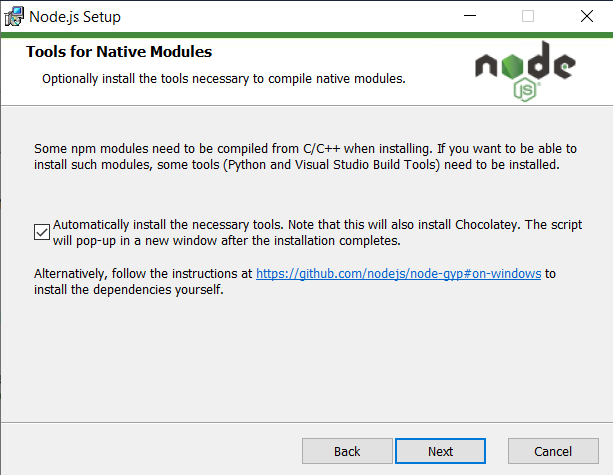
Запустите установочный файл среды исполнения Node.js с расширением **.msi**

Откроется окно установки следующего вида:

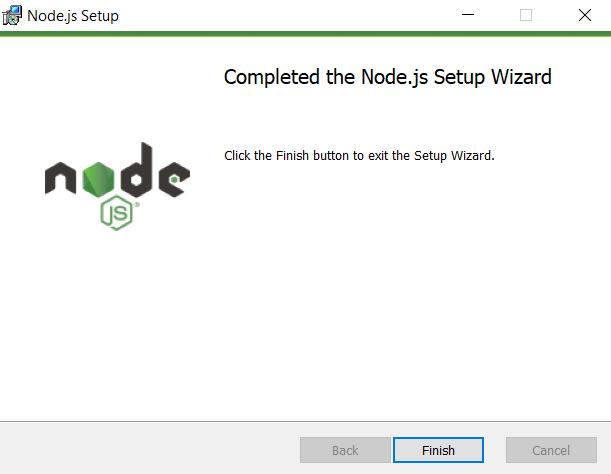


Нажмите **Next** (Далее) и согласитесь со всеми следующими этапами установки.

На последнем этапе установки, будет предложено установить дополнительные инструменты.



Активируйте флаг установки дополнительных инструментов и нажмите **Next** (Далее)



Дождитесь завершения установки Node.js и нажмите **Finish** (Завершить)

Готово! Среда исполнения Node.js установлена.

* + 1. **Установка PostgreSQL**

Загрузите PostgreSQL для Windows по ссылке:

<https://www.postgresql.org/download/windows/>

На момент написания инструкции актуальная версия: **postgresql-12.2-4**

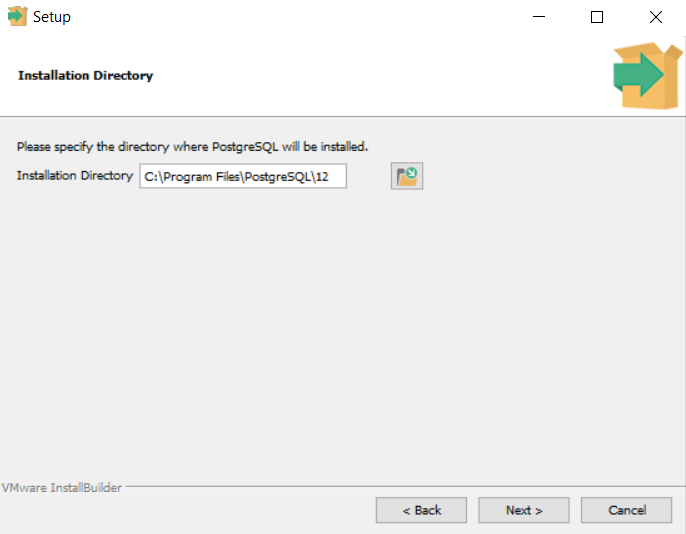
Запустите установочный файл системы управления базами данных PostgreSQL с расширением **.exe**

Откроется окно установки следующего вида:



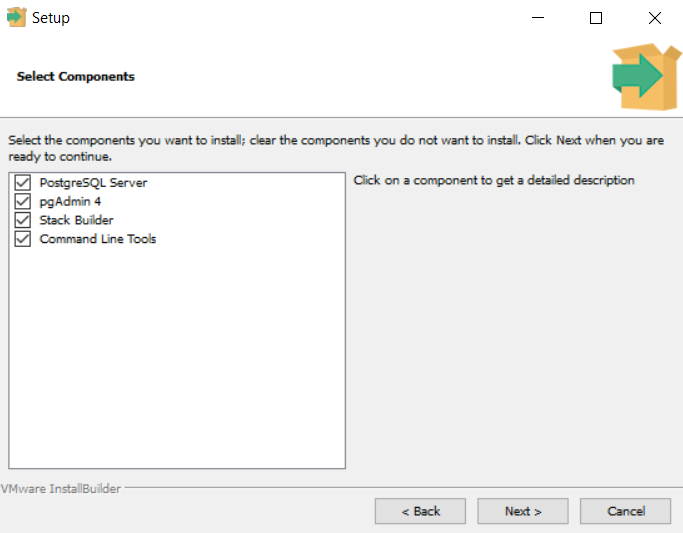
Нажмите **Next** (Далее).

На следующем этапе установки необходимо указать путь установки системы управления базами данных PostgreSQL.



Оставьте путь установки без изменений и нажмите **Next** (Далее).

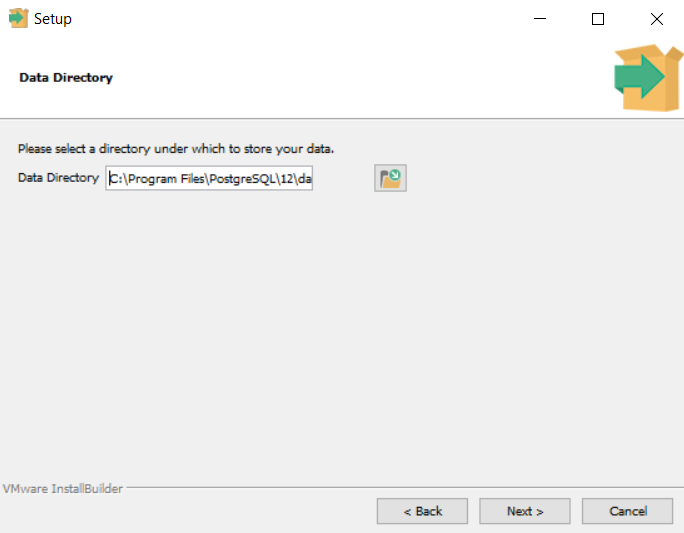
Далее необходимо выбрать набор компонентов.



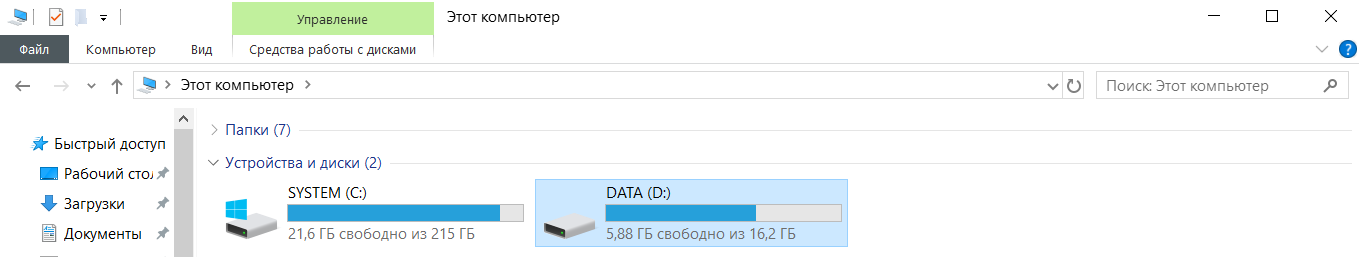
Оставьте набор компонентов без изменений и нажмите **Next** (Далее).

На следующем этапе установки необходимо указать путь хранения Data (Базы данных).

По умолчанию путь будет следующим: **C:\Program Files\PostgreSQL\12\data**

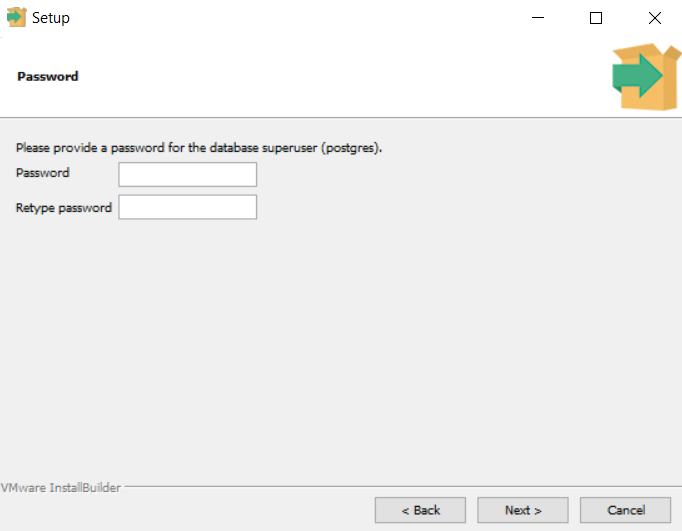


Если в вашей системе имеется несколько дисков, рекомендуется хранить данные Data на другом диске.



После выбора пути установки, нажмите **Next** (Далее).

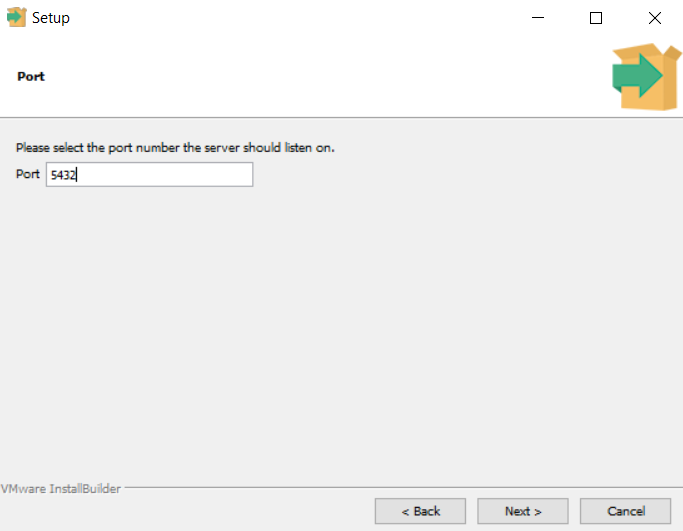
На следующем этапе установки придумайте, укажите и повторите пароль суперпользователя:



Нажмите **Next** (Далее).

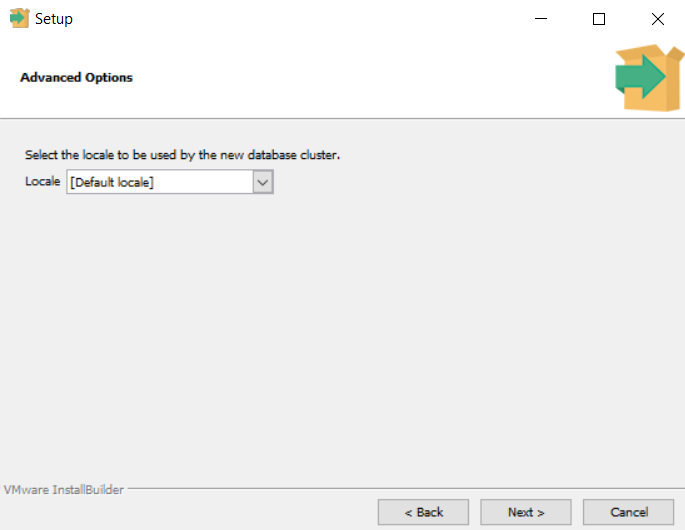
**Рекомендуется записать и сохранить созданный пароль суперпользователя в надежном месте.**

На следующем этапе установки необходимо указать порт работы PostgreSQL.



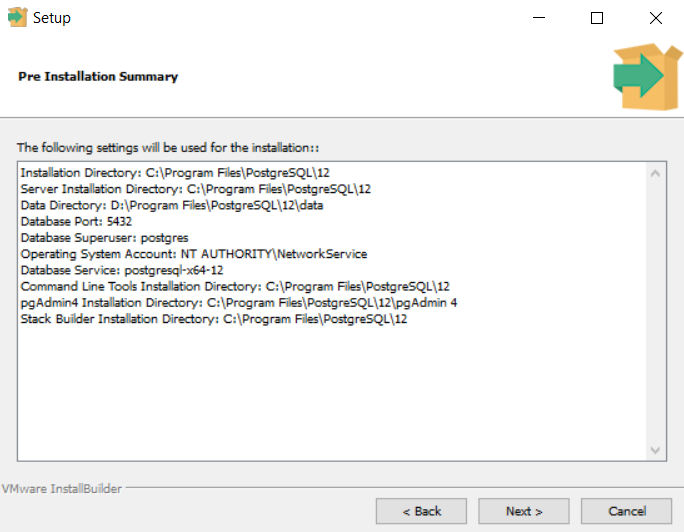
Оставьте порт без изменений **(5432)** и нажмите **Next** (Далее).

На следующем этапе установки необходимо выбрать локацию кластера базы данных.



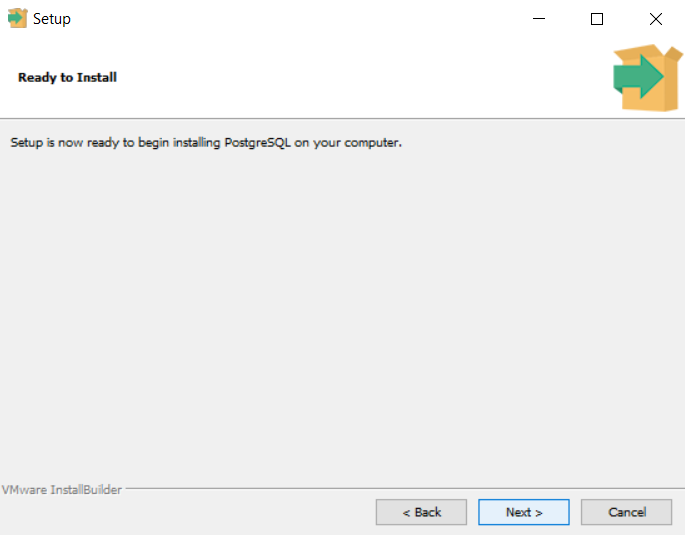
Оставьте локацию без изменений и нажмите **Next** (Далее).

На следующем этапе установки вы увидите предварительные параметры установки.



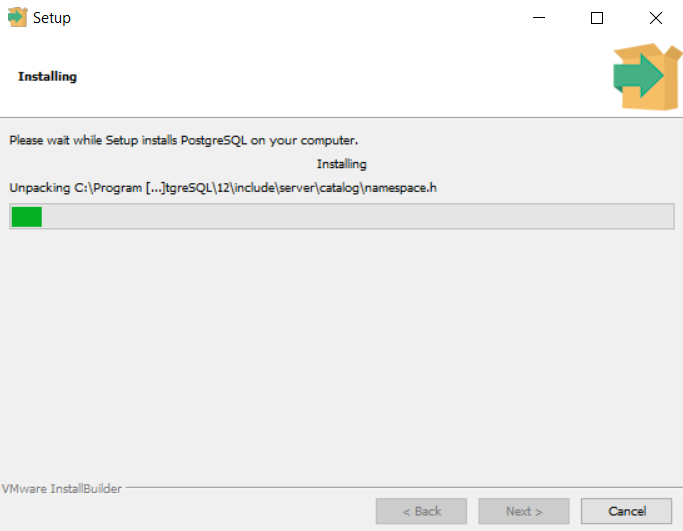
Нажмите **Next** (Далее).

На следующем этапе установки вы увидите сообщение о том, что все готово для установки PostgreSQL на ваш компьютер.



Нажмите **Next** (Далее).

Процесс установки PostgreSQL запущен, пожалуйста дождитесь завершения (2-5 минут).



Установка автоматически завершится.

На следующем этапе установки вы увидите сообщение о том, что установка PostgreSQL завершена и вам будет предложено загрузить и установить дополнительные компоненты.



Снимите данный флаг и нажмите **Finish** (Завершить)

Готово! Система управления базами данных PostgreSQL установлена.

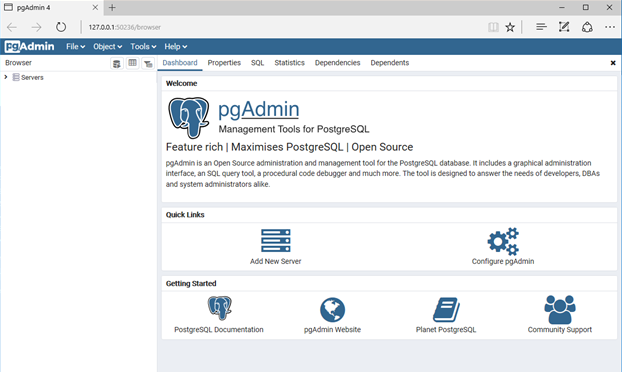
**2.2. Создание пользователя и базы данных**

Запустите панель администрирования pgAdmin 4 системы баз данных из меню пуск Windows.

По умолчанию данный файл также доступен по следующему пути: C:\Program Files\PostgreSQL\12\pgAdmin 4\bin\pgAdmin4.exe

После первого запуска файла, операционная система Windows предложит выбрать браузер по умолчанию, для запуска pgAdmin 4 (выберите Microsoft Edge или Google Chrome).

Откроется страница по адресу <http://127.0.0.1:50236/browser/>



Далее на панели слева, раскройте выпадающий список серверов и выберите вашу версию сервера PostgreSQL.



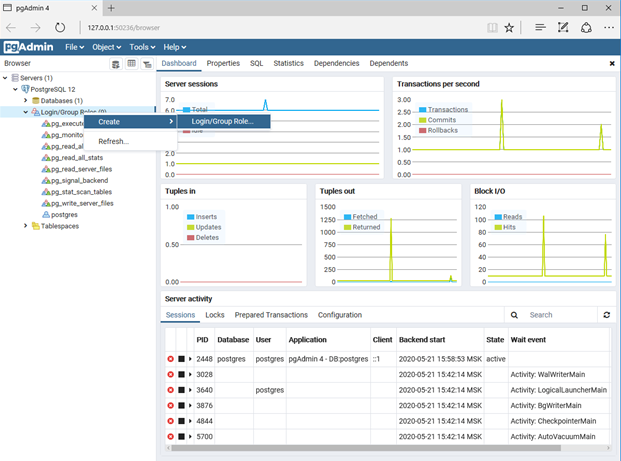
На данном этапе может потребоваться ввод пароля суперпользователя.

Введите созданный ранее пароль суперпользователя PostgreSQL.

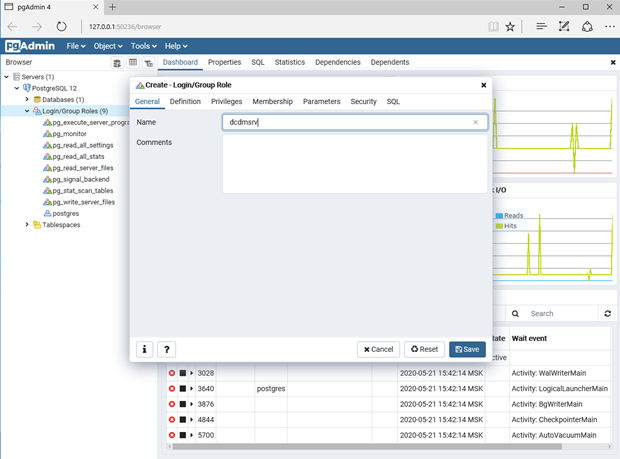
Подробнее о пароле суперпользователя смотрите в Разделе 2.1.2. «Установка PostgreSQL».

Далее необходимо создать пользователя базы данных для Системы DCDM.

Далее перейдите во вкладку панели слева Login/Group Roles, кликните правой кнопкой мыши на пункт меню, и выберите Create>Login/Group Roles.



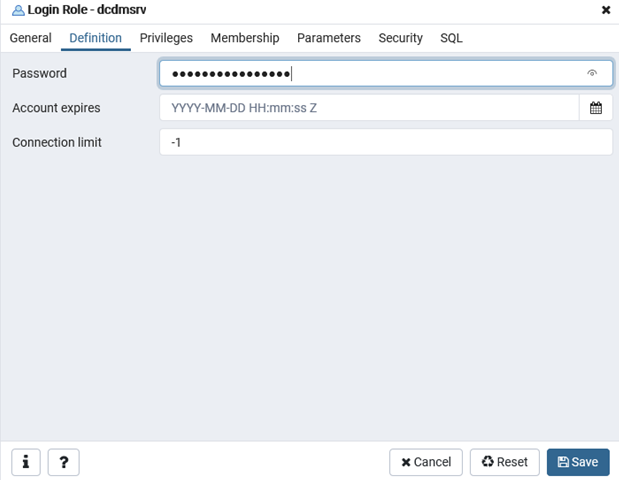
Откроется окно создания пользователя, пароля, назначения привилегий.



Далее необходимо создать пользователя базы данных с именем **dcdmsrv.**

Рекомендуются не создавать пользователя с другим именем, т.к. это может привести к ошибке при выполнении скрипта наполнения базы. Внутри скрипта наполнения базы присутствует информация о пользователе **dcdmsrv.**

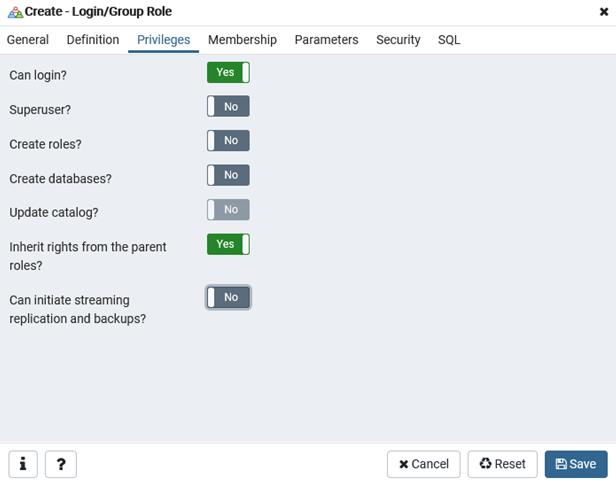
После создания пользователя, переместитесь во вкладку **Definition.**



Далее необходимо создать пароль пользователя.

Рекомендуем записать и сохранить пароль в надежном месте.

Далее созданному пользователю необходимо назначить привилегии.

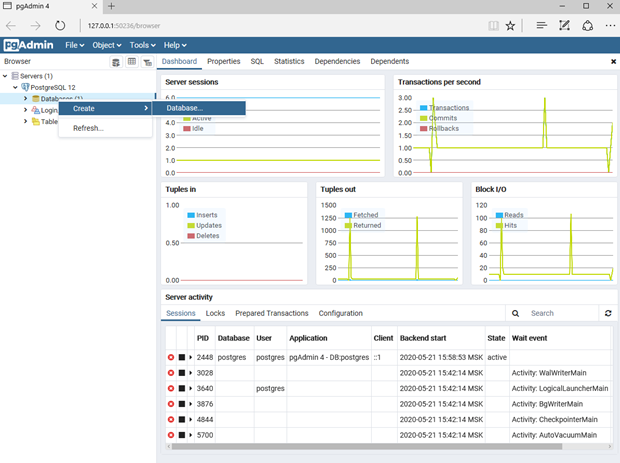


Установите пользователю **dcdmsrv** привилегии аналогично скриншоту выше и нажмите **Save** (Сохранить).

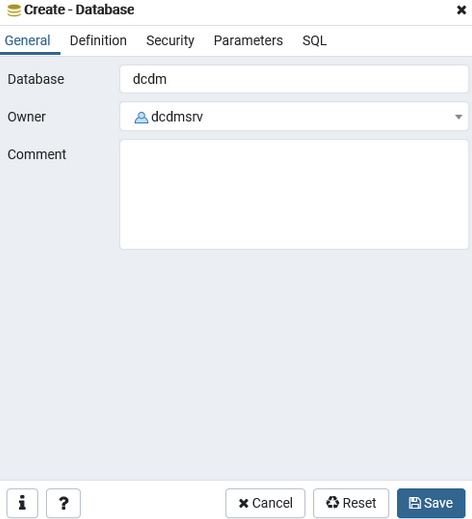
Готово! Создание пользователя завершено.

Далее необходимо создать базу данных **dcdm** для Системы DCDM.

Перейдите во вкладку Databases панели слева, кликните правой кнопкой мыши на пункт меню, и выберите Create>Database.

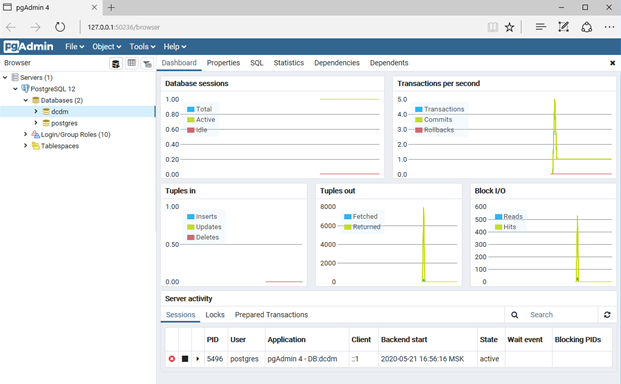


Откроется окно создания базы и назначения владельца:



В поле Database введите название базы **dcdm**, в поле Owner назначьте на базу владельца **dcdmsrv** созданного ране пользователя. По завершению нажмите **Save** (Сохранить).

После завершения данной операции, в списке баз данных вы увидите новую базу **dcdm**, база будет перечеркнута и не активна, нужно подождать запуска 1-2 мин, после чего логотип базы данных станет желтым цветом, что означает готовность базы к работе.

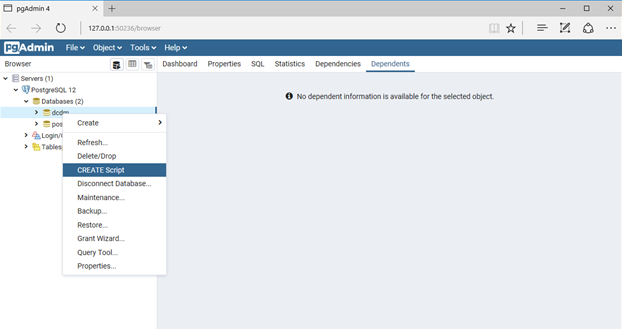


Готово! Пользователь и база данных созданы.

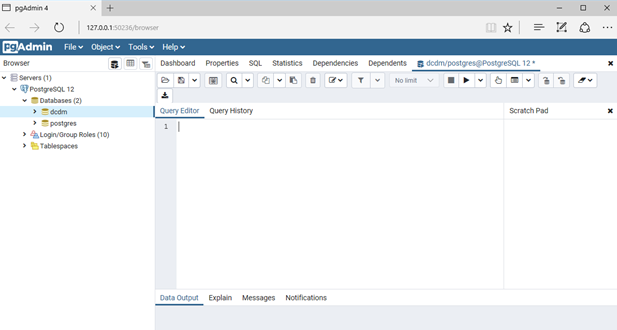
* 1. **Наполнение базы данных**

Запустите панель администрирования pgAdmin 4.

В левой панели панели выберите базу данных **dcdm**, кликните правой кнопкой мыши на пункт меню, и выберите CREATE Script:

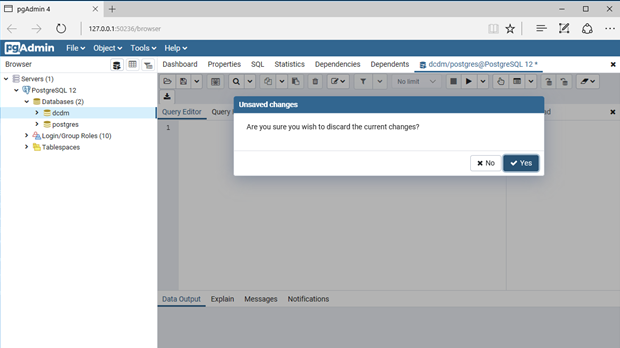


Далее откроется окно ввода запроса, необходимо выбрать и очистить содержимое вкладки Query Editor.



Далее нажмите на кнопку **Open File** (со значком открытой папки)

Откроется окно следующего вида:



В следующем окне с помощью файл менеджера выберите файл **play.sql** из комплекта установочных файлов программы.

Содержимое файла:

**CREATE** EXTENSION **IF** NOT **EXISTS** "uuid-ossp";

**CREATE** EXTENSION **IF** NOT **EXISTS** tablefunc;

*--#############################################################################################*

**Create** **table** DCDM\_Language(

Lng  **varchar**(3)            NOT NULL **PRIMARY** **KEY**,                                                *-- Код языка жалательно по ISO 639-3 но необязятельно*

Name **varchar**(50)           NOT NULL **UNIQUE**                                                      *-- Название языка на этом самом языке*

);

**GRANT** **ALL** **PRIVILEGES** **ON** **TABLE** DCDM\_Language **TO** dcdmsrv;

*-- Исходные языки*

**INSERT** **INTO** dcdm\_language(lng, name) **VALUES** ('ru','Русский');

**INSERT** **INTO** dcdm\_language(lng, name) **VALUES** ('en','English');

*--#############################################################################################*

**Create** **table** DCDM\_VarType(

Id  **smallint**            NOT NULL **PRIMARY** **KEY**,                     *-- Идентификатор*

Name **varchar**(20)        NOT NULL **UNIQUE**                           *-- Название*

);

**GRANT** **ALL** **PRIVILEGES** **ON** **TABLE** DCDM\_VarType **TO** dcdmsrv;

*-- Стандартные типы*

**INSERT** **INTO** DCDM\_VarType(Id,Name) **VALUES** (1,'Int');

**INSERT** **INTO** DCDM\_VarType(Id,Name) **VALUES** (2,'Float');

**INSERT** **INTO** DCDM\_VarType(Id,Name) **VALUES** (3,'Text');

**INSERT** **INTO** DCDM\_VarType(Id,Name) **VALUES** (4,'JSON');

*--##############################################################################################*

**Create** **table** DCDM\_CommentsAndFilesType(

Id  **smallint**            NOT NULL **PRIMARY** **KEY**,                     *-- Идентификатор*

Name **varchar**            NOT NULL **UNIQUE**                           *-- Название (мультиязычная строка)*

);

**GRANT** **ALL** **PRIVILEGES** **ON** **TABLE** DCDM\_CommentsAndFilesType **TO** dcdmsrv;

**INSERT** **INTO** DCDM\_CommentsAndFilesType(Id,Name) **VALUES** ( 1,':en Comment 1 :ru Комментарий 1 :es Comentario 1');

**INSERT** **INTO** DCDM\_CommentsAndFilesType(Id,Name) **VALUES** ( 2,':en Comment 2 :ru Комментарий 2 :es Comentario 2');

**INSERT** **INTO** DCDM\_CommentsAndFilesType(Id,Name) **VALUES** ( 3,':en Comment 3 :ru Комментарий 3 :es Comentario 3');

**INSERT** **INTO** DCDM\_CommentsAndFilesType(Id,Name) **VALUES** ( 4,':en Comment 4 :ru Комментарий 4 :es Comentario 4');

**INSERT** **INTO** DCDM\_CommentsAndFilesType(Id,Name) **VALUES** ( 5,':en Comment 5 :ru Комментарий 5 :es Comentario 5');

**INSERT** **INTO** DCDM\_CommentsAndFilesType(Id,Name) **VALUES** ( 6,':en Lithology :ru Литология :es Litología');

**INSERT** **INTO** DCDM\_CommentsAndFilesType(Id,Name) **VALUES** ( 7,':en Fluorescence :ru ЛБА :es Fluorescencia');

**INSERT** **INTO** DCDM\_CommentsAndFilesType(Id,Name) **VALUES** ( 8,':en Technological :ru Технологические :es Tecnológicos');

**INSERT** **INTO** DCDM\_CommentsAndFilesType(Id,Name) **VALUES** ( 9,':en Timing :ru Хронометраж :es Tiempo');

**INSERT** **INTO** DCDM\_CommentsAndFilesType(Id,Name) **VALUES** (10,':en Complications :ru Осложнения :es Complicaciones');

*--##############################################################################################*

**Create** **table** DCDM\_AccessGroups(

  Id           **serial**       NOT NULL **PRIMARY** **KEY**,                                             *-- Идентификатор*

  Name         **varchar**      NOT NULL **UNIQUE**                                                  *-- Название*

);

**GRANT** **ALL** **PRIVILEGES** **ON** **TABLE** DCDM\_AccessGroups **TO** dcdmsrv;

*--##############################################################################################*

**Create** **table** DCDM\_AcGroupsFld(

  Id           **serial**       NOT NULL **PRIMARY** **KEY**,                                             *-- Идентификатор*

  AcGroupId    **int**          NOT NULL **REFERENCES** DCDM\_AccessGroups(Id),                        *-- Группа*

  Name         **varchar**(20)  NOT NULL,                                                         *-- Название поля журнада DCDM\_WellLog*

**UNIQUE**(AcGroupId,Name)

);

**GRANT** **ALL** **PRIVILEGES** **ON** **TABLE** DCDM\_AcGroupsFld **TO** dcdmsrv;

*--##############################################################################################*

**Create** **table** DCDM\_Users(

  Id           **serial**       NOT NULL **PRIMARY** **KEY**,                                             *-- Идентификатор*

  Login        **varchar**(50)  NOT NULL **UNIQUE**,                                                  *-- Логин*

  Name         **varchar**(100) NULL,                                                             *-- Имя*

  PassHash     **varchar**(130) NOT NULL,                                                         *-- Соленый хеш пароля*

  Salt         **varchar**(20) NOT NULL,                                                         *-- Соль*

  Permission   **varchar**(50)  NOT NULL,                                                         *-- Права пользователя*

  Config       json         NULL,                                                             *-- Конфигурация интерфейса пользователя*

  Props        json         NULL,                                                             *-- Другие пользовательские настройки*

  AcGroup      **int**          NULL **REFERENCES** DCDM\_AccessGroups(Id),                            *-- Группа доступа назначенная пользователю*

  Lang         **varchar**(3)   NULL **REFERENCES** DCDM\_Language(Lng),                               *-- Язык интерфейса для пользователя*

  IsDisabled   **boolean**      NOT NULL **default** false,                                           *-- Флаг отключения аккаунта*

  LatestUse    **timestamp**    NULL                                                              *-- Время последнего входа*

);

**GRANT** **ALL** **PRIVILEGES** **ON** **TABLE** DCDM\_Users **TO** dcdmsrv;

*/\**

*ALTER TABLE DCDM\_Users ADD COLUMN AcGroup int NULL;*

*ALTER TABLE DCDM\_Users ADD CONSTRAINT fk\_AcGroup FOREIGN KEY (AcGroup) REFERENCES DCDM\_AccessGroups(Id);*

*\*/*

*-- Встроенная запись администратора с паролем admin*

**INSERT** **INTO** dcdm\_users(login, name, PassHash, Salt, permission,lang)        **VALUES** ('Administrator','Built-in administrator account','2cbd97469403df5d34144359ddb6aaa3a3dbc06a676c1a974718e3f8f64e8e6e7627ee08f093f6f6fa5f3c96830bf927e9da27206c306eb0a9161fc11f9bde4e','a2d121cc6d76f44d','A','en');

*--##############################################################################################*

**Create** **table** DCDM\_Messages(

  Id          bigserial     NOT NULL **PRIMARY** **KEY**,                                             *-- Идентификатор*

  UserId      **int**           NULL **REFERENCES** DCDM\_Users(Id) **ON** **DELETE** **SET** NULL,                *-- Идентификатор получателя*

  IsReaded    **boolean**       NOT NULL **DEFAULT** false,                                           *-- Признак прочитанного сообщения*

  TypeCode    **smallint**      NOT NULL **DEFAULT** 0,                                               *-- Код типа сообщения (для внутренних нужд 0-обчные сообщения 1-сообщения службы отчетов)*

**Time**        **timestamp**     NOT NULL **DEFAULT** **CURRENT\_TIMESTAMP**,                               *-- Дата и время создания сообщения*

  SenderName  **varchar**(50)   NOT NULL,                                                         *-- Отправитель*

  Subj        **varchar**       NOT NULL,                                                         *-- Тема сообщенич (HTML)*

  Message     **varchar**       NULL                                                              *-- Сообщение (HTML)*

);

**GRANT** **ALL** **PRIVILEGES** **ON** **TABLE** DCDM\_Messages **TO** dcdmsrv;

*--##############################################################################################*

**Create** **table** DCDM\_VarGroups(

  Id          **serial**        NOT NULL **PRIMARY** **KEY**,                                             *-- Идентификатор*

  Name        **varchar**       NULL,                                                             *-- Название*

  IsWidget    **boolean**       NOT NULL **DEFAULT** false,                                           *-- Доступна в виджетах*

  IsImport    **boolean**       NOT NULL **DEFAULT** false,                                           *-- Доступна в импорте*

  IsIPoint    **boolean**       NOT NULL **DEFAULT** false                                            *-- Доступна в точках интеграции*

);

**GRANT** **ALL** **PRIVILEGES** **ON** **TABLE** DCDM\_VarGroups **TO** dcdmsrv;

*--##############################################################################################*

**Create** **table** DCDM\_Vars(

  SysName     **varchar**(10)   NOT NULL **PRIMARY** **KEY** **CHECK**(SysName~\* '^[A-Z].[A-Z0-9\_]\*$'),       *-- Мнемоническое имя*

**Type**        **smallint**      NOT NULL **REFERENCES** DCDM\_VarType(Id),                             *-- Тип*

  InitExpr    **varchar**       NULL,                                                             *-- Строка инициализации (стартовое значение при запуске сервера)*

  CalcExpr    **varchar**       NULL,                                                             *-- Строка вычисления параметра или обработки таймаута (если это внешняя переменная)*

  TlrDvg      float8        NULL,                                                             *-- Терпимое расхождение (Переменная изменяется только если превышен этот порог. Актуально только для Int и Double типов)*

  AvgMethod   **int**           NULL,                                                             *-- Режим усреднения (0-отключен, >0-по указанной глубине, <0-по указанному времени в минутах)*

  LoggedPar   **varchar**(20)   NULL,                                                             *-- Название логируемого параметра или NULL если данные не логируются*

  IsExternal  **bool**          NOT NULL,                                                         *-- Флаг внешнего источника данных если он true то пересчет не используется*

  Timeout     **int**           NULL,                                                             *-- Время таймаута для внешней переменной*

  Description **varchar**       NULL,                                                             *-- Описание переменной (мультиязычное)*

  SortPos     **smallint**      NOT NULL,                                                         *-- Позиция сортировки*

  Lag         **varchar**(10)   NULL,                                                             *-- Запаздывание*

  GroupId     **int**           NULL **REFERENCES** DCDM\_VarGroups(Id),                               *-- Идентификатор группы*

  filter      **varchar**       NULL                                                              *-- Фильтр*

);

**CREATE** **UNIQUE** **INDEX** namecasecontrol **ON** DCDM\_Vars (lower(SysName));

**GRANT** **ALL** **PRIVILEGES** **ON** **TABLE** DCDM\_Vars **TO** dcdmsrv;

*--##############################################################################################*

**Create** **table** DCDM\_Lags(

  SysName     **varchar**(10)   NOT NULL **PRIMARY** **KEY** **CHECK**(SysName~\* '^[A-Z].[A-Z0-9\_]\*$'),       *-- Мнемоническое имя*

  InitVar     **varchar**(10)   NOT NULL **REFERENCES** DCDM\_Vars(SysName),                           *-- Переменная инициализации*

  TruncVar    **varchar**(10)   NOT NULL **REFERENCES** DCDM\_Vars(SysName)                            *-- Переменная усечения*

);

**CREATE** **UNIQUE** **INDEX** namelagcasecontrol **ON** DCDM\_Lags (lower(SysName));

**GRANT** **ALL** **PRIVILEGES** **ON** **TABLE** DCDM\_Lags **TO** dcdmsrv;

**ALTER** **TABLE** dcdm\_vars

**ADD** **CONSTRAINT** dcdm\_vars\_lag\_fkey **FOREIGN** **KEY** (lag)

**REFERENCES** dcdm\_lags(sysname) **MATCH** **SIMPLE**

**ON** **UPDATE** **NO** **ACTION**

**ON** **DELETE** **NO** **ACTION**;

*--##############################################################################################*

**Create** **table** DCDM\_Profiles(

  Id           **serial**       NOT NULL **PRIMARY** **KEY**,                                             *-- Идентификатор*

  Name         **varchar**(50)  NOT NULL **UNIQUE**,                                                  *-- Название*

  Description  **varchar**      NULL,                                                             *-- Описание*

  IsActive     **boolean**      NOT NULL **DEFAULT** false                                            *-- Флаг активности профиля (активным должен быть только один)*

);

**GRANT** **ALL** **PRIVILEGES** **ON** **TABLE** DCDM\_Profiles **TO** dcdmsrv;

*--##############################################################################################*

**Create** **table** DCDM\_IPoints(

  Id           **serial**       NOT NULL **PRIMARY** **KEY**,                                             *-- Идентификатор*

  ProfileId    **int**          NOT NULL **REFERENCES** DCDM\_Profiles(Id) **ON** **DELETE** **CASCADE**,          *-- Профиль*

  PointType    **varchar**(20)  NOT NULL,                                                         *-- Тип точки интеграции*

  Name         **varchar**(50)  NOT NULL **UNIQUE**,                                                  *-- Название*

  Config       json         NOT NULL                                                          *-- Конфигурация*

);

**GRANT** **ALL** **PRIVILEGES** **ON** **TABLE** DCDM\_IPoints **TO** dcdmsrv;

*--##############################################################################################*

**Create** **table** DCDM\_ReferencesList(

  Id          **serial**        NOT NULL **PRIMARY** **KEY**,                                             *-- Идентификатор*

  SysName     **varchar**(10)   NOT NULL **UNIQUE** **CHECK**(SysName~\* '^[A-Z].[A-Z0-9\_]\*$'),            *-- Мнемоническое имя справочника. Например: Ref1*

  Name        **varchar**       NOT NULL,                                                         *-- Название справочника (мультиязычная строка)*

  IsWellDep   **boolean**       NOT NULL,                                                         *-- Скважинозависимый справочник (пустой для новой скважины)*

  BrowseData  jsonb         NULL                                                              *-- Информация о том откуда могут браться данные для данной таблице и как мапируются поля источника в данную таблицу*

);

**CREATE** **UNIQUE** **INDEX** **ON** DCDM\_ReferencesList (lower(SysName));

**GRANT** **ALL** **PRIVILEGES** **ON** **TABLE** DCDM\_ReferencesList **TO** dcdmsrv;

*--##############################################################################################*

*-- Описание колонки отдельно взятого справочника*

**Create** **table** DCDM\_ReferenceCol(

  Id          **serial**        NOT NULL **PRIMARY** **KEY**,                                             *-- Идентификатор*

  RefListId   **int**           NOT NULL **REFERENCES** DCDM\_ReferencesList(Id) **ON** **DELETE** **CASCADE**,    *-- Идентификатор справочника в реестре*

**Type**        **smallint**      NOT NULL **REFERENCES** DCDM\_VarType(Id),                             *-- Тип*

  SysName     **varchar**(10)   NOT NULL **CHECK**(SysName~\* '^[A-Z].[A-Z0-9\_]\*$'),                   *-- Мнемоническое имя поля. Например: Fld1*

  ColName     **varchar**       NOT NULL,                                                         *-- Название параметра (мультиязычная строка)*

  InitExpr    **varchar**       NULL,                                                             *-- Строка инициализации. Актуально только для скважинозависимых справочников*

**UNIQUE**(RefListId,SysName)

);

**CREATE** **UNIQUE** **INDEX** **ON** DCDM\_ReferenceCol (lower(SysName));

**GRANT** **ALL** **PRIVILEGES** **ON** **TABLE** DCDM\_ReferenceCol **TO** dcdmsrv;

*--##############################################################################################*

**Create** **table** DCDM\_Well(

  Id          **serial**        NOT NULL **PRIMARY** **KEY**,                                             *-- Идентификатор*

  Name        **varchar**       NOT NULL,                                                         *-- Название скважины (мультиязычное)*

  Latitude    **real**          NULL,                                                             *-- Широта*

  Longitude   **real**          NULL,                                                             *-- Долгота*

  CrDate      **timestamp**     NOT NULL **default** **current\_timestamp**,                               *-- Дата создания*

  IsActive    **boolean**       NOT NULL **default** false                                           *-- признак активной скважины*

);

**GRANT** **ALL** **PRIVILEGES** **ON** **TABLE** DCDM\_Well **TO** dcdmsrv;

*--##############################################################################################*

*-- Данные справочников*

**Create** **table** DCDM\_ReferenceData(

  Id          bigserial     NOT NULL **PRIMARY** **KEY**,                                             *-- Идентификатор*

  WellId      **int**           NULL **REFERENCES** DCDM\_Well(Id) **ON** **DELETE** **CASCADE**,                 *-- Идентификатор скважины. (Актуально для скважинозависибых справочников)*

  RefListId   **int**           NOT NULL **REFERENCES** DCDM\_ReferencesList(Id) **ON** **DELETE** **CASCADE**,   *-- Идентификатор справочника в реестре*

  SubId       **int**           NOT NULL,                                                         *-- Субидентификатор сортировки спрвочников*

  GrpName     **varchar**(10)   NULL **CHECK**(GrpName~\* '^[A-Z].[A-Z0-9\_]\*$'),                       *-- Префикс для образования массива данных. Все поля с одним префиксом входят в один массив*

  Val         json          NOT NULL                                                          *-- json значение*

);

**CREATE** **UNIQUE** **INDEX** **ON** DCDM\_ReferenceData(WellId,RefListId,SubId) **WHERE** WellId IS NOT NULL;

**CREATE** **UNIQUE** **INDEX** **ON** DCDM\_ReferenceData(RefListId,SubId) **WHERE** WellId IS NULL;

**GRANT** **ALL** **PRIVILEGES** **ON** **TABLE** DCDM\_ReferenceData **TO** dcdmsrv;

*--##############################################################################################*

**Create** **table** DCDM\_WellLogFieldDesc(

ParamName  **varchar**(20)  NOT NULL **PRIMARY** **KEY**,                                                *-- Название поля*

Descr      **varchar**      NULL                                                                 *-- Мультиязычное описание*

);

**GRANT** **ALL** **PRIVILEGES** **ON** **TABLE** DCDM\_WellLogFieldDesc **TO** dcdmsrv;

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('AAFV',':en Acc. Actual Displacement :es Desplazamiento Actual Acumulado :ru Накопленное фактическое наполнение :cn 累积实际加入 :pt Acc. Deslocamento Atual');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('AFV',':en Actual Displacement :es Desplazamiento Actual :ru Фактическое наполнение :cn 实际加入 :pt Deslocamento Atual');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('AH2S1',':en Ambient H2S 1 :es H2S 1 Ambiente :ru Окружающий сероводород 1 :cn 大气中硫化񬈡 :pt H2S 1 Ambiente');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('AH2S2',':en Ambient H2S 2 :es H2S 2 Ambiente :ru Окружающий сероводород 2 :cn 大气中硫化񬈢 :pt H2S 2 Ambiente');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('AH2S3',':en Ambient H2S 3 :es H2S 3 Ambiente :ru Окружающий сероводород 3 :cn 大气中硫化񬈣 :pt H2S 3 Ambiente');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('AH2S4',':en Ambient H2S 4 :es H2S 4 Ambiente :ru Окружающий сероводород 4 :cn 大气中硫化񬈤 :pt H2S 4 Ambiente');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('AMD',':en Actual Mud Density');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('AP1',':en Air Pressure GCT OUT');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('AP2',':en Air Pressure GCT IN');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('AROP',':en ROP Rolling :es ROP :ru Скорость проходки вращением :cn 机械钻速滚动 :pt ROP Rotação');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('BBT1',':en Enclosure Temp GCT OUT');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('BBT2',':en Enclosure Temp GCT IN');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('BCB1',':en Benzene Out :es Benzeno de Salida :pt Benzeno de Sa');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('BCB2',':en Benzene In :es Benzeno de Entrada :pt Benzeno de Entrada');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('BCBD',':en Delta Benzene :es Delta Benzeno :cn Benzene Delta :pt Delta Benzeno');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('BCON',':en Bit Condition :es Condición de Mecha :ru Состояние бура :cn 钻头条件 :pt Condição da Broca');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('BCR1',':en Benzene Out(Raw) :es Benzeno de Salida (Raw) :pt Benzeno de Sa (Raw)');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('BCR2',':en Benzene In(Raw) :es Benzeno de Entrada (Raw) :pt Benzeno de Entrada (Raw)');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('BCRD',':en Delta Benzene(Raw) :es Delta Benzeno (Raw) :pt Delta Benzeno (Raw)');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('BCW1',':en QC Benzene Out :es QC Benzeno de Salida :pt QC Benzeno de Sa');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('BCW2',':en QC Benzene In :es QC Benzeno de Entrada :pt QC Benzeno de Entrada');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('BCWD',':en QC Delta Benzene :es QC Delta Benzeno :pt QC Delta Benzeno');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('BDEN',':en Bulk Density :es Densidad Volumétrica :ru Объемная плотность :cn 体积密度 :pt Volumen de Densidade');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('BDIR',':en Block Direction :es Dirección del Bloque :ru Направление блока :cn 游车方向 :pt Direção do Bloco');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('BDTI',':en Bit Drilled Time :es Tiempo de Perforación de la Mecha :ru Время бурения :cn 钻头使用时间 :pt Tempo de Perfuração da Broca');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('BHALEN',':en BHA Length :es Longitud del BHA :ru Длина скважинного узла :cn 钻具组合长度 :pt Comprimento do BHA');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('BHP',':en Hydraulic HP @ Bit :es Hidráulica HP @ Mecha :ru Гидравлический бур высокого давления :cn 钻头水马力 :pt Hidráulica HP @ Broca');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('BHPA',':en Bit Hydraulic Power/Area :es Área/Poder de la Mecha :ru Мощность/площадь гидравлического бура :cn 钻头水功率 :pt Àrea / Broca Poder de Hidráulica');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('BLAF',':en BL Air Flow');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('BLCS',':en BL Calibration Slope');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('BLDC',':en BL Detector Current');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('BLDT',':en BL Detector Temperature');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('BLFF',':en BL Fuel Flow');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('BLSF',':en BL Sample Flow');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('BLTG',':en BL Total Gas Raw');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('BLTGM',':en BL Total Gas Measured');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('BLZO',':en BL Zero Offset');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('BMF\_LAGGED',':en Lagged Channels :es Canales de Retraso :pt Canais de Atrasos (Lagged)');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('BMF\_RECALCULATED',':en Recalculated Channels :es Canales Recalculados :pt Canais Recalculados');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('BMOS',':en Block Movement Out-Of-Slips :es Movimiento del Bloque (Fuera de Cu༚s) :ru Отсутствуют смещения дивжения блока :cn 无卡瓦时游车运动 :pt Movimento do Bloco (Out-Of-Slips)');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('BMT',':en Block Movement :es Movimiento del Bloque :ru Движение блока :cn 游车运动 :pt Movimento do Bloco');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('BOPT',':en Bit Off Bottom Time :es Tiempo de Mecha Fuera de Fondo :ru Время низа бура :cn 钻头非钻进时间 :pt Tempo da Broca Fora de Fundo');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('BP1',':en Enclosure Pressure GCT OUT');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('BP2',':en Enclosure Pressure GCT IN');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('BPFI',':en Booster Pump Flow :es Flujo de Bomba Booster :ru Поток бустерного насоса :cn 泵启动流量 :pt Fluxo da Bomba do (Booster)');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('BPOS',':en Block Position :es Posición del Bloque :ru Позиция блока :cn 游车位置 :pt Posição do Bloco');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('BPVM',':en Booster Pump Volume Moved :es Volumen Movido por la Bomba Booster :pt Volumen Movido da Bomba do (Booster)');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('BRAT',':en Bal R MLS');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('BRAT2',':en Bal R GCT');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('BRI',':en Block Rate Instantaneous :es Rata Instantánea de Movimiento del Bloque :ru Мнгновенная частота блоков :cn 瞬时游车速度 :pt Taxa Instantânea do Bloco');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('BRMC',':en Total Bit Distance :es Metros Totales de la Mecha :pt Metros Total da Broca');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('BRTI',':en Bit Reamed Time :es Tiempo de Repaso de la Mecha :ru Время развёртывания бура :cn 钻头划眼时间 :pt Tempo Repassado da Broca');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('BRVC',':en Total Bit Revolutions :es Revoluciones Totales de la Mecha :ru Общее число оборотов бура :cn 钻头总转数 :pt Revoluçཞs Total da Broca');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('BTMP',':en Block Temperature :es Temperatura del Bloque :ru Температура блока :cn 游车温度 :pt Temperatura do Bloco');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('BTPOSBM',':en Bit Position due to Block Movement :es Posición de la Mecha debida al Movimiento del Bloque :ru Позиция бура к движению блока :cn 因游车运动的钻头位置 :pt Posição Adequada da Broca para Movimento do Bloco');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('BTTI',':en Bit Total Time :es Tiempo Total de la Mecha :ru Общее время бура :cn 总钻头使用时间 :pt Tempo Total da Broca');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('BWEAR',':en Bit Wear :es Desgaste de la Mecha :ru Износ бура :cn 钻头磨损 :pt Desgaste da Broca');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('C1C2',':en C1/C2 (Chromat)');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('C1C3',':en C1/C3 (Chromat)');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('C1C4',':en C1/C4 (Chromat)');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('C1C5',':en C1/C5 (Chromat)');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('C1CB1',':en Methane Out :es Metano de Salida :pt Metano de Sa');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('C1CB2',':en Methane In :es Metano de Entrada :pt Metano de Entrada');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('C1CBD',':en Delta Methane :es Delta Metano :pt Delta Metano');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('C1CFG',':en C1 Corrected For Free Gas');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('C1CR1',':en Methane Out(Raw) :es Metano de Salida (Raw) :pt Metano de Sa (Raw)');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('C1CR2',':en Methane In(Raw) :es Metano de Entrada (Raw) :pt Metano de Entrada (Raw)');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('C1CRD',':en Delta Methane(Raw) :es Delta Metano (Raw) :pt Delta Metano (Raw)');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('C1CW1',':en QC Methane Out :es QC Metano de Salida :pt QC Metano de Sa');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('C1CW2',':en QC Methane In :es QC Metano de Entrada :pt QC Metano de Entrada');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('C1CWD',':en QC Delta Methane :es QC Delta Metano :pt QC Delta Metano');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('C1D',':en C1 Density');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('C1FC',':en C1 Flow Corrected');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('C1PC',':en C1 Pressure Compensated');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('C1SF',':en C1 Solubility Factor');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('C1THC',':en C1/THC (Chromat)');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('C2C3',':en C2/C3 (Chromat)');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('C2CB1',':en Ethane Out :es Etano de Salida :pt Etano de Sa');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('C2CB2',':en Ethane In :es Etano de Entrada :pt Etano de Entrada');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('C2CBD',':en Ethane Delta :es Delta Ethano :ru Delta Ethane :pt Delta Etano');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('C2CR1',':en Ethane Out(Raw) :es Etano de Salida (Raw) :pt Etano de Sa (Raw)');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('C2CR2',':en Ethane In(Raw) :es Etano de Entrada (Raw) :pt Etano de Entrada (Raw)');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('C2CRD',':en Delta Ethane(Raw) :es Delta Etano (Raw) :pt Delta Etano (Raw)');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('C2CW1',':en QC Ethane Out :es QC Etano de Salida :pt QC Etano de Sa');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('C2CW2',':en QC Ethane In :es QC Etano de Entrada :pt QC Etano de Entrada');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('C2CWD',':en QC Delta Ethane :es QC Delta Etano :pt QC Delta Etano');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('C2FC',':en C2 Flow Corrected');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('C2PC',':en C2 Pressure Compensated');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('C3CB1',':en Propane Out :es Propano de Salida :pt Propano de Sa');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('C3CB2',':en Propane In :es Propano de Entrada :pt Propano de Entrada');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('C3CBD',':en Delta Propane :es Delta Propano :pt Delta Propano');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('C3CR1',':en Propane Out(Raw) :es Propano de Salida (Raw) :pt Propano de Sa (Raw)');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('C3CR2',':en Propane In(Raw) :es Propano de Entrada (Raw) :pt Propano de Entrada (Raw)');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('C3CRD',':en Delta Propane(Raw) :es Delta Propano (Raw) :pt Delta Propano (Raw)');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('C3CW1',':en QC Propane Out :es QC Propano de Salida :pt QC Propano de Sa');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('C3CW2',':en QC Propane In :es QC Propano de Entrada :pt QC Propano de Entrada');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('C3CWD',':en QC Delta Propane :es QC Delta Propano :pt QC Delta Propano');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('C3FC',':en C3 Flow Corrected');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('C3PC',':en C3 Pressure Compensated');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('C6CB1',':en N-Hexane Out :es N-Hexano de Salida :pt N-Hexano de Sa');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('C6CB2',':en N-Hexane In :es N-Hexano de Entrada :pt N-Hexano de Entrada');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('C6CBD',':en Delta N-Hexane :es Delta N-Hexano :pt Delta N-Hexano');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('C6CR1',':en N-Hexane Out(Raw) :es N-Hexano de Salida (Raw) :pt N-Hexano de Sa (Raw)');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('C6CR2',':en N-Hexane In(Raw) :es N-Hexano de Entrada (Raw) :pt N-Hexano de Entrada (Raw)');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('C6CRD',':en Delta N-Hexane(Raw) :es Delta N-Hexano (Raw) :pt Delta N-Hexano (Raw)');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('C6CW1',':en QC N-Hexane Out :es QC N-Hexano de Salida :pt QC N-Hexano de Sa');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('C6CW2',':en QC N-Hexane In :es QC N-Hexano de Entrada :pt QC N-Hexano de Entrada');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('C6CWD',':en QC Delta N-Hexane :es QC Delta N-Hexano :pt QC Delta N-Hexano');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('C7CB1',':en N-Heptane Out :es N-Heptano de Salida :pt N-Heptano de Sa');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('C7CB2',':en N-Heptane In :es N-Heptano de Entrada :pt N-Heptano de Entrada');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('C7CBD',':en Delta N-Heptane :es Delta N-Heptano :pt Delta N-Heptano');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('C7CR1',':en N-Heptane Out(Raw) :es N-Heptano de Salida (Raw) :pt N-Heptano de Sa (Raw)');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('C7CR2',':en N-Heptane In(Raw) :es N-Heptano de Entrada (Raw) :pt N-Heptano de Entrada (Raw)');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('C7CRD',':en Delta N-Heptane(Raw) :es Delta N-Heptano (Raw) :pt Delta N-Heptano (Raw)');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('C7CW1',':en QC N-Heptane Out :es QC N-Heptano de Salida :pt QC N-Heptano de Sa');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('C7CW2',':en QC N-Heptane In :es QC N-Heptano de Entrada :pt QC N-Heptano de Entrada');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('C7CWD',':en QC Delta N-Heptane :es QC Delta N-Heptano :pt QC N-Heptano Delta');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('C7HCB1',':en Methylcyclohexane Out :es Methylcyclohexano de Salida :pt Methylcyclohexano de Sa');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('C7HCB2',':en Methylcyclohexane In :es Methylcyclohexano de Entrada :pt Methylcyclohexano de Entrada');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('C7HCBD',':en Delta Methylcyclohexane :es Delta Methylcyclohexano :pt Delta Methylcyclohexano');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('C7HCR1',':en Methylcyclohexane Out(Raw) :es Methylcyclohexano de Salida (Raw) :pt Methylcyclohexano de Sa (Raw)');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('C7HCR2',':en Methylcyclohexane In(Raw) :es Methylcyclohexano de Entrada (Raw) :pt Methylcyclohexano de Entrada (Raw)');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('C7HCRD',':en Delta Methylcyclohexane(Raw) :es Delta Methylcyclohexano (Raw) :pt Delta Methylcyclohexano (Raw)');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('C7HCW1',':en QC Methylcyclohexane Out :es QC Methylcyclohexano de Salida :pt QC Methylcyclohexano de Sa');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('C7HCW2',':en QC Methylcyclohexane In :es QC Methylcyclohexano de Entrada :pt QC Methylcyclohexano de Entrada');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('C7HCWD',':en QC Delta Methylcyclohexane :es QC Delta Methylcyclohexano :pt QC Delta Methylcyclohexano');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('C8CB1',':en N-Octane Out :es N-Octano de Salida :pt N-Octano de Sa');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('C8CB2',':en N-Octane In :es N-Octano de Entrada :pt N-Octano de Entrada');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('C8CBD',':en Delta N-Octane :es Delta N-Octano :pt Delta N-Octano');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('C8CR1',':en N-Octane Out(Raw) :es N-Octano de Salida (Raw) :pt N-Octano de Sa (Raw)');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('C8CR2',':en N-Octane In(Raw) :es N-Octano de Entrada (Raw) :pt N-Octano de Entrada (Raw)');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('C8CRD',':en Delta N-Octane(Raw) :es Delta N-Octano (Raw) :pt Delta N-Octano (Raw)');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('C8CW1',':en QC N-Octane Out :es QC N-Octano de Salida :pt QC N-Octano de Sa');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('C8CW2',':en QC N-Octane In :es QC N-Octano de Entrada :pt QC N-Octano de Entrada');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('C8CWD',':en QC Delta N-Octane :es QC Delta N-Octano :pt QC Delta N-Octano');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('CALP1',':en CalGas Pressure GCT OUT');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('CALP2',':en CalGas Pressure GCT IN');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('CAT:AN',':en Anadrill Record 63 :es Registro 63 Anadrill :ru Запись Anadrill 63 :cn Anadrill 记� :pt Registro 63 Anadrill');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('CAT:BI',':en Bit Parameters :es Parámetros de Mecha :ru Параметры бура :cn 钻头参数 :pt Parâmetros da Broca');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('CAT:CM',':en Comments :es Comentarios :ru Комментарии :cn 备注 :pt Comentários');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('CAT:DB',':en Debug Parameters :es Parámetros de Debug :ru Отладочные параметры :cn 调试参数 :pt Debug Parâmetros');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('CAT:DE',':en Depth Parameters :es Parámetros de Profundidad :ru Параметры глубины :cn 井深参数 :pt Parâmetros de Profundidade');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('CAT:DP',':en Drilling Parameters :es Parámetros de Perforación :ru Параметры бурения :cn 钻井参数 :pt Parâmetros de Perfuração');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('CAT:EHYD',':en eHydraulics Parameters :es Parámetros de eHydraulics :pt Parâmetros eHidráulica');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('CAT:GA',':en GCT Gas');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('CAT:GCTH',':en GCT Hardware');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('CAT:MPD',':en GCT MPD');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('CAT:GCTR',':en GCT Ratios');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('CAT:LI',':en Lithology :es Litolog໚ :ru Литология :cn 岩性 :pt Litologia');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('CAT:LP',':en Lag Parameters');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('CAT:MI',':en Miscellaneous Parameters :es Parámetros Diversos :ru Смешанные параметры :cn 其它参数 :pt Parâmetros Diversos');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('CAT:MLSG',':en MLS Gas');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('CAT:MU',':en Mud Parameters :es Parámetros de Lodo :ru Параметры бурильного раствора :cn 泥浆参数 :pt Parâmetros de Lama');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('CAT:MWD',':en MWD Parameters :es Parámetros de MWD :ru Параметры скважинных исследований :cn MWD参数 :pt Parâmetros de MWD');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('CAT:NORTHLAND',':en Northland Energy :ru Северная энергия :pt Energia Northland');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('CAT:PI',':en Pit Parameters :es Parámetros de Tanques :ru Параметры шахты :cn 泥浆池参数 :pt Parâmetros de Volumens');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('CAT:PIPE',':en Pipe Parameters :es Parámetros de Tuber໚ :ru Параметры трубы :cn 管线参数 :pt Parâmetros de Tubos (Pipe)');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('CAT:PR',':en Pressure Parameters :es Parámetros de Presión :ru Параметры давления :cn 压力参数 :pt Parâmetros de Pressão');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('CAT:PU',':en Pump Parameters :es Parámetros de Bombas :ru Параметры насоса :cn 泵参数 :pt Parâmetros de Bomba');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('CAT:TP',':en Trip Parameters :es Parámetros de Viaje :pt Parâmetros de Viagem');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('CAT:WL',':en Wireline Parameters :es Parámetros de Wireline :ru Линейные параметры :cn 电缆参数 :pt Parâmetros Wireline');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('CAT:WWSRA',':en Source Rock Analyzer');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('CBG1',':en Combustible 1 :ru Горючее 1 :cn 可燃񠉱 :pt Combustível 1');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('CBG2',':en Combustible 2 :ru Горючее 2 :cn 可燃񠉲 :pt Combustível 2');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('CBG3',':en Combustible 3 :ru Горючее 3 :cn 可燃񠉳 :pt Combustível 3');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('CBG4',':en Combustible 4 :ru Горючее 4 :cn 可燃񠉴 :pt Combustível 4');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('CBWEAR',':en Cumulative Bit Wear :es Desgaste Acumulado de la Mecha :ru Совокупный износ бура :cn 累积的钻头磨损 :pt Cumulativo Visto da Broca');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('CCGS',':en CC Gas Sensor :es Sensor de Gas CC :ru Сенсор газовой критической трещины :cn 催化燃烧气体分析仪 :pt CC Sensor de Gàs');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('CCQC',':en CC QC Ratio');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('CDO',':en Calcimetry Dolomite :es Calcimetr໚ de Dolomitas :ru Доломитовая кальциметрия :cn 含碳酸盐的白云岩 :pt Calcimetria de Dolomite');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('CHKP',':en Casing Pressure :es Presión de Revestidor :ru Давление в кольцевом пространстве :cn 套压 :pt Pressão de Casing');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('CHOKEP',':en Choke Pressure :es Presión de Choque :pt Pressão de Choke');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('CLD',':en Chromat Lag Depth');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('CLI',':en Calcimetry Limestone :es Calcimetr໚ de Carbonatos :ru Известняковая кальциметрия :cn 含碳酸盐的石灰岩 :pt Calcimetria Limestone');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('CLT',':en Chromat Lag Time');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('CO',':en CO :ru Моноокись углерода :cn 一氧化碳');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('CO2',':en CO2 :ru Двуокись углерода :cn 二氧化碳');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('CO2CB1',':en Carbon Dioxide Out :es Dióxido de Carbono de Salida');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('CO2CB2',':en Carbon Dioxide In :es Dióxido de Carbono de Entrada');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('CO2CBD',':en Delta Carbon Dioxide :es Delta Dióxido de Carbono');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('CO2CR1',':en Carbon Dioxide Out(Raw) :es Dióxido de Carbono de Salida (Raw) :pt Gás Carbônico de Saida (Raw)');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('CO2CR2',':en Carbon Dioxide In(Raw) :es Dióxido de Carbono de Entrada (Raw) :pt Gás Carbônico de Entrada (Raw)');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('CO2CRD',':en Delta Carbon Dioxide(Raw) :es Delta Dióxido de Carbono (Raw) :pt Gás Carbônico Delta (Raw)');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('CO2CW1',':en QC Carbon Dioxide Out :es QC Dióxido de Carbono de Salida :pt Gás Carbônico de Saida');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('CO2CW2',':en QC Carbon Dioxide In :es QC Dióxido de Carbono de Entrada :pt Gás Carbônico de Entrada');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('CO2CWD',':en QC Delta Carbon Dioxide :es QC Delta Dióxido de Carbono :pt Gás Carbônico Delta');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('CO2\_1',':en CO2 Out :es CO2 de Salida :ru Двуокись углерода исходящая :cn 排出二氧化碳 :pt CO2 de Saida');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('CO2\_2',':en CO2 In :es CO2 de Entrada :ru Двуокись углерода входящая :cn 输入二氧化碳 :pt CO2 de Entrada');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('COM1',':en Comment1 :es Comentario1 :ru Комментарий 1 :cn 备񬺁 :pt Comentário 1');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('COM2',':en Comment2 :es Comentario 2 :ru Комментарий 2 :cn 备񬺂 :pt Comentário 2');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('COM3',':en Comment3 :es Comentario 3 :ru Комментарий 3 :cn 备񬺃 :pt Commentário 3');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('COM4',':en Comment4 :es Comentario 4 :ru Комментарий 4 :cn 备񬺄 :pt Commentário 4');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('COM5',':en Comment5 :es Comentario 5 :ru Комментарий 5 :cn 备񬺅 :pt Commentário 5');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('COM6',':en Comment6 :es Comentario 6 :ru Комментарий 6 :cn 备񬺆 :pt Commentário 6');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('COM7',':en Comment7 :es Comentario 7 :ru Комментарий 7 :cn 备񬺇 :pt Commentário 7');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('COM8',':en Comment8 :es Comentario 8 :ru Комментарий 8 :cn 备񬺈 :pt Commentário 8');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('COM9',':en SRA Comment :es Comentario 9 :ru Комментарий 9 :cn 备񬺉 :pt Commentário 9');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('COM10',':en Comment10 :es Comentario 10 :ru Комментарий 10 :cn 备� :pt Commentário 10');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('COMD',':en Depth Comment :es Comentario de Profundidad :ru Комментарий глубины :cn 井深备注 :pt Comentário de Profundidade');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('COMI',':en Interpreted Lithology Comment :es Comentario de Litolog໚ Interpretada :ru Интерпретированный комментарий литологии :cn 岩性解释备注 :pt Comentàrios de Litologia Interpretada');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('COMM',':en Memos :ru Заметки :cn 备忘录');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('COMP',':en Percentage Lithology Comment :es Comentario de Litolog໚ Porcentual :ru Комментарий процентной литологии :cn 岩性百分比注释 :pt Comentários de Litologia Porcentual');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('COMT',':en Time Comment :es Comentario de Tiempo :ru Временной комментарий :cn 时间备注 :pt Comentário Tempo');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('CO\_1',':en CO Out :es CO de Salida :ru Моноокись углерода исходящая :cn 排出一氧化碳 :pt CO Saida');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('CO\_2',':en CO In :es CO de Entrada :ru Моноокись углерода входящая :cn 输入一氧化碳 :pt CO Entrada');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('CP1',':en Carrier Pressure GCT OUT');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('CP2',':en Carrier Pressure GCT IN');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('CPDC',':en Cost/Distance :es Costo/Distancia :ru Цена/Расстояние :cn 花费/距离 :pt Custo/Distância');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('CPOS',':en Comp Position :es Posición del Compensador :ru Расчётная позиция :cn 位置补偿 :pt Comp. Posição');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('CRAT',':en Char R MLS');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('CRAT2',':en Char R GCT');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('CT1',':en Chromat Temp GCT OUT');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('CT2',':en Chromat Temp GCT IN');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('CTTL',':en Chromat Time to Lag');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('CWTD',':en Tracer Delta Mud Temp :es Tracer - Temperatura de Lodo Delta :pt Tracer Temp Lama Delta');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('CWTI',':en Tracer In Mud Temp :es Tracer - Temperatura de Lodo de Entra :pt Tracer Temp de Entrada da Lama');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('CWTO',':en Tracer Out Mud Temp :es Tracer - Temperatura de Lodo de Salida :pt Tracer Temp de Saida da Lama');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('DAYNO',':en Day Number :es Número de D໚ :ru Номер дня :cn 天数 :pt Número de Dia');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('DBSD',':en Bit Start Depth :es Inicio de Profundidad de la Mecha :ru Стартовая глубина бура :cn 钻头起始位置 :pt Inicio de Profundidade da Broca');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('DBST',':en Bit Start Time :es Tiempo de Inicio de la Mecha :ru Стартовое время бура :cn 钻头起始时间 :pt Inicio do Tempo da Broca');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('DBTM',':en Bit Depth :es Profundidad de la Mecha :ru Глубина бура :cn 钻头井深 :pt Profundidade da Broca');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('DBTMKB',':en Bit Depth TRKB :es Profundidad de la Mecha TRKB :ru Глубина бура истинный относительный ротор :cn 钻头井深(实际转盘补芯) :pt Profundidade da Broca TRKB');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('DCV1',':en System DC Voltage GCT OUT');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('DCV2',':en System DC Voltage GCT IN');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('DGM',':en Depth G.Ray MWD :es Profundidad de MWD Gamma Ray :ru Глубина гамма-лучей измеренная во время бурения :cn 随钻测量伽马井深 :pt Profundidade G.Ray MWD');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('DGM2',':en Depth G. Ray 2 MWD');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('DGM2O',':en Gamma 2 Offset');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('DGMO',':en Gamma Offset');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('DIPA',':en Dip Angle :es Angulo DIP :cn 倾角 :pt Dip Ângulo');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('DMEA',':en Hole Depth :es Profundidad de Hoyo :ru Глубина скважины :cn 井深 :pt Profundidade do Poço');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('DMEAKB',':en Hole Depth TRKB :es Profundidad de Hoyo TRKB :ru Глубина скважины истинный относительный ротор :cn 井深(实际转盘补芯) :pt Profundidade do Poço TRKB');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('DP1M',':en Porosity Depth 1 :es Profundidad 1 de Porosidad :pt Profundidade 1 Porosidade');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('DP1MO',':en Porosity 1 Offset');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('DP2M',':en Porosity Depth 2 :es Profundidad 2 de Porosidad :pt Profundidade 2 Porosidade');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('DP2MO',':en Porosity 2 Offset');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('DPD',':en Delta Pipe Displacement');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('DPLEN',':en Drill Pipe Length :es Longitud de Tuber໚ de Perforación :ru Длина бурильной трубы :cn 钻柱长度 :pt Comprimento da Tubulação da Broca');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('DPLENB',':en Drill Pipe Length @ Bottom');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('DRME',':en Ream Depth :es Profundidad de Repaso :ru Глубина райберования :cn 划眼深度 :pt Profundidade Ream');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('DRTM',':en Lag Depth :es Profundad de Atraso :ru Запаздывание глубины :cn 迟到井深 :pt Última Profundidade de Atraso (Lag Depth)');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('DRTMG',':en Lag Depth Gas :es Profundidad de Gas de Atraso :ru Запаздывание глубины газа :cn 气体迟到井深 :pt Última Profundidade de Gás (Lag Depth Gás)');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('DRTT',':en Lag Time :es Tiempo de Atraso :ru Время запаздывания :cn 迟到时间 :pt Último Tempo (Lag Time)');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('DRTTB',':en Lag Time @ Bottom');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('DRTTT',':en Lag Time Theoretical :es Tiempo de Atraso Teorético :ru Теоретическое время запаздывания :cn 理论迟到时间 :pt Último Tempo Teórico (Lag Time Teórico)');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('DS',':en Down Strokes :es Emboladas de Bajada :ru Ход поршня вниз :cn 井口到钻头的泵冲数 :pt Movimentos para Baixo');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('DSAM',':en Sample Depth :es Profundidad de Muestreo :ru Глубина образца :cn 取样井深 :pt Profundidade da Amostra');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('DSAT',':en Time to Sample :es Tiempo para Muestreo :ru Время к образцу :cn 取样等待时间 :pt Tempo até a Amostra');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('DSB',':en Down Strokes @ Bottom :es Down Strokes @ Botto');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('DSNI',':en Stands In Hole :es Parejas en el Hoyo :ru Свечи в шахте :cn 井内立柱数 :pt Seçཞs de Tubo no Poço');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('DSNO',':en Total Stands :es Parejas Totales :ru Все свечи :cn 总立柱数 :pt Número Total de Seçཞs de Tubos');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('DSNP',':en Stands Pulled :es Parejas Sacadas :ru Вытянутые свечи :cn 已起立柱数 :pt Seçཞs de Tubo Puxadas');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('DSR2KB',':en TRKB->SRKB Diff :es Diferencial TRKB->SRKB :ru Истинный относительный ротор->Статический относительный ротор Разн. :cn 实际补芯->静止补芯距离');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('DSTK',':en Delta Strokes :es Emboladas Deltas :ru Дельта перемещений :cn 泵冲数差 :pt Movimentos Delta');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('DSVM',':en S.Depth :es Profundidad de Muestreo :ru Глубина обследования :cn 测量井深 :pt Profundidade da Amostra');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('DT',':en Down Time :es Tiempo de Bajada :ru Время простоя :cn 井口到钻头循环时间 :pt Tempo até em Baixo');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('DTB',':en Down Time @ Bottom');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('DX',':en D Exponent :es Exponente D :ru D Экспонента :cn D 指数 :pt Expoente D');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('DXC',':en Dc Exponent :es Exponente DC :ru Уточнённая D Экспонента :cn Dc指数 :pt Expoente Dc');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('ECD',':en ECD :ru Эквивалентная плотность циркулирующего бурового раствора :cn 当量泥浆密度');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('EHYD\_ECD',':en eHydraulics ECD :es eHydraulics - ECD :pt ECD do eHydraulics');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('EHYD\_SWSU',':en eHydraulics Surge/Swab Pressure :es eHydraulics - Swap/Surge :pt Pressão do Limpador do Poço/Surgência do eHydraulics');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('EMD',':en Expected Mud Density');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('ETH',':en C2 :ru Этан :cn 乙烷');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('ETH1',':en C2 Out :es C2 de Salida :ru Этан исходящий :cn 排出乙烷 :pt C2 de Sa');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('ETH2',':en C2 In :es C2 de Entrada :ru Этан входящий :cn 输入乙烷 :pt C2 de Entrada');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('ETVTA',':en Expected Active PVT');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('ETVTT',':en Expected Trip Tank PVT');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('FCI1',':en Flow Controler IN GCT OUT');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('FCI2',':en Flow Controler In GCT IN');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('FCO1',':en Flow Controler Out GCT OUT');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('FCO2',':en Flow Controler Out GCT IN');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('FFAC',':en Formation Factor :es Factor de Formación :ru Фактор формации :cn 地层系数 :pt Fator de Formação');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('FFPG',':en Formation Gradient :es Gradiente de Formación :ru Градиент формации :cn 地层梯度 :pt Gradiente de Formação');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('FIDQC',':en FID QC Ratio');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('FIDQC2',':en FID 2 QC Ratio');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('FILES',':en Attached Files :es Archivos Adjuntos :ru Присоединённые файлы :cn 附加文件 :pt Arquivos Anexados');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('FLAIRP',':en Flare Line Pressure :es Presión de Línea de Quemado :pt Linha de Pressão do Fluido');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('FLC',':en Flow Line Compensation');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('FLCSA',':en Flow Line Cross Section Area');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('FLD',':en Flow Line Diameter');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('FOBG',':en Overburden Gradient :es Gradiente de Sobrecarga :ru Градиент горного давления :cn 上覆压力梯度 :pt Gradiente de Sobrecarga');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('FPPG',':en Calc Est Porosity :es Porosidad Estimada Calculada :ru Расчётная предполагаемая пористость :cn 理论空隙度 :pt Porosidade Calculada Estimado');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('FSO1',':en FS Out');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('FSO2',':en FS Out (IN)');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('FUELF',':en Fuel Flow :es Flujo de Combustible :ru Поток топлива :cn 燃料流 :pt Vazão de Combustível');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('FUELL',':en Fuel Level :es Nivel de Combustible :ru Уровень топлива :cn 燃料量 :pt Nível de Combustível');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('FVE',':en Theo. Displacement :es Desplazamiento Teórico :pt Deslocamento Teórico');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('FVEC',':en Acc. Theo. Displacement :es Desplazamiento Teórico Acumulado :pt Deslocamento Teórico Acumulado');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('GASA',':en Total Gas Sensor(avg) :es Sensor de Gas Total (Promedio) :cn 传感器气体总量(平均) :pt Sensor de Gás Total (Mຝia)');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('GASC',':en Total Chromat Gas GCT :es Gas Total de Cromatograf໚ :ru Общий хромат газа :cn 色谱气体总量 :pt Cromatografia Gasosa Total');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('GASC1',':en Total Chromat Gas Out GCT :es Gas Total de Cromatograf໚ de Salida :ru Общий исходящий хромат газа :cn 总排出色谱气 :pt Cromatografia Gasosa Total de Sa');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('GASC2',':en Total Chromat Gas In GCT :es Gas Total de Cromatograf໚ de Entrada :ru Общий входящий хромат газа :cn 总流入色谱气 :pt Cromatografia Gasosa Total de Entrada');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('GASC3',':en Total Chromat Gas');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('GASN',':en Normalized Gas :es Gas Normalizado :ru Нормализованный газ :cn 校正后气体量 :pt Gás Normalizado');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('GASS',':en Total Gas Sensor :es Sensor de Gas Total :ru Общий датчик газа :cn 总气体传感器 :pt Sensor de Gás Total');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('GASS1',':en Total Gas Out :es Gas Total de Salida :cn 排出端气体总量 :pt Gás Total de Sa');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('GASS2',':en Total Gas In :es Gas Total de Entrada :cn 流入端的气体总量 :pt Gás Total de Entrada');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('GCC',':en Calc CC Gas Sensor');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('GCF4',':en Arom/Alk');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('GCF8',':en C1 Content');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('GCF19',':en C1/C6');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('GCF23',':en C1/ROP');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('GCF24',':en C1/CO2');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('GCF25',':en C1/Benzene');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('GCF26',':en Benzene+Toluene');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('GCF30',':en PROETH');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('GCF31',':en BUTISO');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('GCF32',':en PENTISO');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('GCTCT',':en GCT Cycle Time');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('GCTD',':en GCT Density');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('GCTG',':en GCT Grade');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('GCUT',':en Total Cuttings Gas :es Gas Total de Cortes :ru Общий бурильный газ :cn 全部岩屑内气体 :pt Gás Total dos Cuttings');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('GFID',':en Calc FID Total Gas :es Gas Total FID Calculado :ru Расчётный общий газ детектора пламенной ионизации :cn 计算的火焰电离总气体量 :pt Gás Total Calc FID');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('GFID2',':en CALC FID 2 TOTAL GAS');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('GFO',':en Gas Flow Out :es Flujo de Salida de Gas :ru Исходящий газовый поток :cn 排出气体量 :pt Vazão de Sa do Gás');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('GHW',':en Calc HW Total Gas :es Gas Total HW Calculado :ru Расчётный тепловой общий газ :cn 计算的催化燃烧总气体量 :pt Gás Total Calc. HW');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('GRAV',':en Total Gravity :es Gravedad Total :cn 总重力 :pt Gravidade Total');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('GSG',':en Gas SG :ru Обследованный газ :cn 气体比重 :pt Gás SG');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('GTCD',':en Calc TCD Gas Sensor');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('H',':en H :ru Водород :cn 氢气');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('H1',':en H Out :es H de Salida :ru Водород исходящий :cn 排出氢气 :pt H de Sa');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('H2',':en H In :es H de Entrada :ru Водород входящий :cn 输入氢气 :pt H de Entrada');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('H2S',':en H2S :ru Сероводород :cn 硫化氢');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('H2S1',':en H2S Out :es H2S de Salida :ru Сероводород исходящий :cn 排出硫化氢 :pt H2S de Sa');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('H2S2',':en H2S In :es H2S de Entrada :ru Сероводород входящий :cn 输入硫化氢 :pt H2S de Entrada');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('HD',':en Hole Drag :es Arrastre del Hoyo :ru Сопротивление скважины :cn 井眼阻力 :pt Arrasto do Poço');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('HDIA',':en Hole Diameter :es Diámetro del Hoyo :ru Диаметр шахты :cn 井眼直径 :pt Diâmetro do Poço');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('HE',':en He :ru Гелий :cn 氦');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('HE1',':en He Out :es He de Salida :ru Гелий исходящий :cn 排出氦气 :pt He de Sa');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('HE2',':en He In :es He de Entrada :ru Гелий входящий :cn 输入氦气 :pt He de Entrada');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('HEAV',':en Heave :es Movimiento de Marea :ru Вертикальная качка :cn 升降');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('HKLA',':en Avg Hookload :es Carga del Gancho Promedio :cn 平均钩载 :pt Carga no Gancho Mຝia');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('HKLI',':en Hookload :es Carga del Gancho :ru Грузоподъёмность :cn 大钩载荷 :pt Carga no Gancho');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('HKLN',':en Min Hookload :es Cargar del Gancho Mínima :pt Carga no Gancho Mínima');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('HKLX',':en Max Hookload :es Cargar del Gancho Máxima :cn 最大钩载 :pt Carga no Gancho Máxima');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('HMOS',':en Hook Movement Out-of-Slips :es Movimiento del Gancho Fuera de Cu༚s :ru Отсутствуют смещения движения крюка :cn 大钩移动脱离卡瓦 :pt Movimento do Gancho Fora das Cunhas');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('HMT',':en Hook Movement :es Movimiento del Gancho :ru Движение крюка :cn 大钩移动 :pt Movimento do Gancho');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('HPOS',':en Hook Position :es Posición del Gancho :ru Позиция крюка :cn 大钩位置 :pt Posição do Gancho');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('HSPD',':en Hook Speed :es Velocidad del Gancho :ru Скорость крюка :cn 大钩移动速度 :pt Velocidade do Gancho');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('HUM1',':en Humidity GCT OUT');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('HUM2',':en Humidity GCT IN');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('HVMP',':en Heave Rate :es Rata de Movimiento de Marea :ru Интенсивность вертикальной качки :cn 升降速率 :pt Taxa de Heave');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('HVSG',':en Heave Amplitude :es Amplitud de Marea :ru Амплитуда вертикальной качки :cn 升降幅度 :pt Amplitude de Heave');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('HYDC',':en Total Hydrocarbons GCT :es Hidrocarburos :ru Углеводороды :cn 烃 :pt Hidrocarbonos');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('HYDC1',':en Total Hydrocarbons Out GCT :es Hidrocarburos de Salida :ru Углеводороды исходящие :cn 排出烃 :pt Hidrocarbonos de Sa');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('HYDC2',':en Total Hydrocarbons In GCT :es Hidrocarburos de Entrada :ru Углеводороды входящие :cn 输入烃 :pt Hidrocarbonos de Entrada');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('HYDC3',':en Total Hydrocarbons MLS :es Total Hydrocarbon Chromat Out :ru Total Hydrocarbon Chromat Out :cn Total Hydrocarbon Chromat Out :pt Total Hydrocarbon Chromat Out');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('HYDG',':en Hydrostatic Gradient :es Gradiente Hidrostático :ru Гидростатический градиент :cn 静水压力梯度 :pt Gradiente Hidrostático');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('HYDP',':en Hydrostatic Pressure :es Presión Hidrostática :ru Гидростатическое давление :cn 静水压力 :pt Pressão Hidrostática');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('IBUT',':en iC4 :ru и-Бутан :cn 异丁烷');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('IBUT1',':en iC4 Out :es iC4 de Salida :ru и-Бутан исходящий :cn 排出异丁烷 :pt iC4 de Sa');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('IBUT2',':en iC4 In :es iC4 de Entrada :ru и-Бутан входящий :cn 输入异丁烷 :pt iC4 de Entrada');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('IC4CB1',':en Iso-Butane Out :es Iso-Butano de Salida :pt Iso-Butano de Sa');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('IC4CB2',':en Iso-Butane In :es Iso-Butano de Entrada :pt Iso-Butano de Entrada');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('IC4CBD',':en Delta Iso-Butane :es Delta Iso-Butano :pt Delta Iso-Butano');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('IC4CR1',':en Iso-Butane Out(Raw) :es Iso-Butano de Salida (Raw) :pt Iso-Butano de Sa (Raw)');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('IC4CR2',':en Iso-Butane In(Raw) :es Iso-Butano de Entrada (Raw) :pt Iso-Butano de Entrada (Raw)');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('IC4CRD',':en Delta Iso-Butane(Raw) :es Delta Iso-Butano (Raw) :pt Delta Iso-Butano (Raw)');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('IC4CW1',':en QC Iso-Butane Out :es Iso-Butano de Salida :cn Iso-Butane Out :pt QC Iso-Butano de Sa');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('IC4CW2',':en QC Iso-Butane In :es Iso-Butano de Entrada :cn Iso-Butane In :pt QC Iso-Butano de Entrada');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('IC4CWD',':en QC Delta Iso-Butane :es Delta Iso-Butano :cn Delta Iso-Butane :pt QC Delta Iso-Butano');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('IC4FC',':en iC4 Flow corrected');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('IC5CB1',':en Iso-Pentane Out :pt Iso-Pentano de Sa');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('IC5CB2',':en Iso-Pentane In :pt Iso-Pentano de Entrada');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('IC5CBD',':en Delta Iso-Pentane :es Delta Iso-Pentano :pt Delta Iso-Pentano');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('IC5CR1',':en Iso-Pentane Out(Raw) :es Iso-Pentano de Salida (Raw) :pt Iso-Pentano de Sa (Raw)');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('IC5CR2',':en Iso-Pentane In(Raw) :es Iso-Pentano de Entrada (Raw) :pt Iso-Pentano de Entrada (Raw)');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('IC5CRD',':en Delta Iso-Pentane(Raw) :es Delta Iso-Pentano (Raw) :pt Delta Iso-Pentano (Raw)');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('IC5CW1',':en QC Iso-Pentane Out :es QC Iso-Pentano de Salida :cn Iso-Pentane Out :pt QC Iso-Pentano de Sa');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('IC5CW2',':en QC Iso-Pentane In :es QC Iso-Pentano de Entrada :cn Iso-Pentane In :pt QC Iso-Pentano de Entrada');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('IC5CWD',':en QC Delta Iso-Pentane :es QC Delta Iso-Pentano :cn Delta Iso-Pentane :pt QC Delta Iso-Pentano');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('IC5FC',':en iC5 Flow Corrected');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('IFA',':en Impact Force Area :es Área de Fuerza de Impacto :ru Зона ударной силы :cn 冲击压力区域 :pt Área de For๺ de Impacto');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('IPEN',':en iC5 :ru и-Пентан :cn 异戊烷');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('IPEN1',':en iC5 Out :es iC5 de Salida :ru и-Пентан исходящий :cn 排出异戊烷 :pt iC5 de Sa');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('IPEN2',':en iC5 In :es iC5 de Entrada :ru и-Пентан входящий :cn 输入异戊烷 :pt iC5 de Entrada');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('JIF',':en Bit Impact Force :es Fuerza de Impacto de la Mecha :ru Ударная сила бура :cn 钻头冲击力 :pt For๺ de Impacto da Broca');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('JV',':en Jet Velocity :es Velocidad de Jet :ru Скорость струи :cn 射流速度 :pt Velocidade do Jato');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('KT',':en Kick Tolerance :es Tolerancia de Kick :ru Допустимая доза выброса :cn 井涌余量 :pt Tolerância do Kick');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('LAC1',':en Fossils 1 :es Fósiles 1 :ru Ископаемые 1 :cn 化񷼱 :pt Fosséis 1');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('LAC2',':en Fossils 2 :es Fósiles 2 :ru Ископаемые 2 :cn 化񷼲 :pt Fosséis 2');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('LAC3',':en Minerals 1 :es Minerales 1 :ru Минералы 1 :cn 矿񲚑 :pt Minerais 1');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('LAC4',':en Minerals 2 :es Minerales 2 :ru Минералы 2 :cn 矿񲚒 :pt Minerais 2');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('LAC5',':en Oil Shows 1 :es Presencia de Aceites 1 :ru Нефтепроявления 1 :cn 油气显񹎡 :pt Óleos 1');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('LAC6',':en Oil Shows 2 :es Presencia de Aceites 2 :ru Нефтепроявления 2 :cn 油气显񹎢 :pt Óleos 2');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('LAC7',':en Geological 1 :es Geológico 1 :ru Геологический 1 :cn 地򍊁 :pt Geológico 1');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('LAC8',':en Geological 2 :es Geológico 2 :ru Геологический 2 :cn 地򍊂 :pt Geológico 2');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('LAC9',':en Engineering 1 :es Ingenier໚ 1 :ru Инженерный 1 :cn 工񺂱 :pt Engenharia 1');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('LAC10',':en Engineering 2 :es Ingenier໚ 2 :ru Инженерный 2 :cn 工񺂲 :pt Engenharia 2');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('LFL1',':en Floresence 1 :es Fluorescencia 1 :ru Расцвет 1 :cn 荧񑒑 :pt Florescência 1');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('LFL2',':en Floresence 2 :es Fluorescencia 2 :ru Расцвет 2 :cn 荧񑒒 :pt Florescência 2');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('LMG',':en WL Gamma :es WL Gama :ru Гамма керна :cn 测井伽马 :pt WL Gama');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('LMGA',':en WL Gamma Avg :es WL Gama Promedio :ru Гамма керна средняя :cn 测井伽马平均值 :pt WL Gama Mຝia');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('LMGX',':en WL Gamma Max :es WL Gama Máxima :ru Гамма керна максимальная :cn 测井伽马最大值 :pt WL Gama Máxima');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('LMR',':en WL Resistivity :es Resistividad WL :ru Сопротивление керна :cn 电阻率测井 :pt Resistividade WL');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('LMS',':en WL Sonic :es WL Sónico :ru Звук керна :cn 声波测井 :pt WL Sônico');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('LPOR1',':en Porosity 1 :es Porosidad 1 :ru Пористость 1 :cn 孔򖦑 :pt Porosidade 1');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('LPOR2',':en Porosity 2 :es Porosidad 2 :ru Пористость 2 :cn 孔򖦒 :pt Porosidade 2');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('LPRT',':en Porosity Type :es Tipo de Porosidad :ru Тип пористости :cn 孔隙类型 :pt Tipo de Porosidade');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('LPT',':en Percentage Lithology :es Litolog໚ Porcentual :ru Процентная литология :cn 岩性百分数 :pt Percentual de Litologia');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('LRD',':en Roundness :es Redondamiento :ru Закругленность :cn 圆度 :pt Arredondamento');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('LSO',':en Sorting :es Segregación :ru Классификация :cn 分选 :pt Segregação');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('LSTK',':en Lag Strokes :es Emboladas de Atraso :ru Задержки хода :cn 迟到泵冲数 :pt Movimentos de Lagging');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('LSTKB',':en Lag Strokes @ Bottom');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('LSZ',':en Grain Size :es Tamaño de Grano :ru Размер зерна :cn 粒度 :pt Tamanho do Grão');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('LTY',':en Interpreted Lithology :es Litolog໚ Interpretada :ru Интерпретированная литология :cn 岩性解释 :pt Litologia Interpretada');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('LVA',':en Lag Volume Adjust :es Ajuste de Volumen de Retraso :ru Настройка величины запаздывания :cn 调整延迟容积 :pt Ajuste do Volume de Lagging');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('LVL',':en Lag Volume Left :es Volumen de Retraso Restante :ru Оставшаяся величина запаздывания :cn 剩余迟到体积 :pt Volume de Lagging Remanescente');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('LWEAR',':en Line Wear :es Desgaste de Línea de Perforación :ru Износ струны :cn 大绳磨损 :pt Desgaste da Linha');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('MBPA',':en BH Annulus Pressure :es Presión Anular de Fondo del Pozo :pt Pressão Anular do Fundo do Poço');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('MBPI',':en BH Internal Pressure :es Presión Interna de Fondo del Pozo :pt Pressão Interna do Fundo do Poço');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('MCDA',':en Delta Mud Cond. :es Conductividad de Lodo Delta :ru Дельта состояния бурового раствора :cn 泥浆电导率差 :pt Cond. Lama Delta');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('MCIA',':en Mud Cond. In :es Conductividad de Lodo de Entrada :ru Состояние входящнго бурового раствора :cn 入口泥浆电导率 :pt Cond. Lama de Entrada');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('MCOA',':en Mud Cond. Out :es Conductividad de Lodo de Salida :ru Состояние исходящнго бурового раствора :cn 出口泥浆电导率 :pt Cond. Lama de Sa');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('MD',':en Meters Drilled :es Metros Perforados :ru Пробурено метров :cn 进尺 :pt Metros Perfurados');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('MDDA',':en Delta Mud Density :es Densidad de Lodo Delta :ru Дельта плотности бурового раствора :cn 泥浆密度差 :pt Densidade da Lama Delta');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('MDH2S',':en MDH2S :cn 泥浆罐硫化氢');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('MDIA',':en Mud Density In :es Densidad de Lodo de Entrada :ru Входящая плотность бурового раствора :cn 入口泥浆密度 :pt Densidade da Lama de Entrada');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('MDOA',':en Mud Density Out :es Densidad de Lodo de Salida :ru Исходящая плотность бурового раствора :cn 出口泥浆密度 :pt Densidade da Lama de Sa');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('METH',':en C1 :ru Метан :cn 甲烷');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('METH1',':en C1 Out :es C1 de Salida :ru Метан исходящий :cn 排出甲烷 :pt C1 de Sa');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('METH2',':en C1 In :es C1 de Entrada :ru Метан входящий :cn 输入甲烷 :pt C1 de Entrada');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('MFDF',':en EKDS Flow Diff :es EKDS Flujo Diferencial :pt Diff. Fluxo EKDS');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('MFFP',':en Fracture Gradient :es Gradiente de Fractura :ru Градиент давления гидравлического разрыва :cn 压裂梯度 :pt Gradiente de Fratura');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('MFGL',':en Flow G/L :es Ganancia/Pérdida de Flujo :ru Прирост/потеря потока :cn 流量增/减 :pt Ganho/Perda de Vazão');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('MFGLR',':en Flow G/L Rate :es Tasa de Ganancia/Pérdida de Flujo :ru Скорость прироста/потери потока :cn 流量增/减 :pt Taxa de Ganho/Perda de Vazão');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('MFII',':en Flow In :es Flujo de Entrada :ru Втекание :cn 流入流量 :pt Vazão de Entrada');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('MFOI',':en Flow Out :es Flujo de Salida :ru Вытекание :cn 流出流量 :pt Vazão de Sa');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('MFOP',':en Flow Out Percent :es Flujo de Salida Porcentual :ru Процент вытекания :cn 流出百分数 :pt Vazão de Sa Percentual');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('MG',':en Gamma :ru Гамма :cn 伽马 :pt Gama');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('MG2',':en Gamma 2');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('MGA',':en Gamma Avg :es Gamma Promedio :ru Гамма средняя :cn 平均伽马 :pt Gama Mຝio');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('MGX',':en Gamma Max :es Gama Máximo :ru Гамма максимальная :cn 最大伽马 :pt Gama Máximo');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('MIH',':en Mud In Hole :es Lodo en el Hoyo :ru Буровой раствор в скважине :cn 井中泥浆 :pt Lama no Poço');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('MM',':en Molar Mass :es Masa Molar :ru Молярная масса :cn 摩尔质量 :pt Massa Molar');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('MPHD',':en Delta Mud pH :es pH de Lodo Delta :ru Дельта водородного показателя бурового раствора :cn 泥浆pH差 :pt pH da Lama Delta');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('MPHI',':en Mud pH In :es pH de Lodo de Entrada :ru Входящий водородный показателя бурового раствора :cn 流入泥浆pH :pt pH da Lama de Entrada');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('MPHO',':en Mud pH Out :es pH de Lodo de Salida :ru Исходящий водородный показателя бурового раствора :cn 流出泥浆pH :pt pH da Lama de Sa');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('MR',':en Resistivity Deep');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('MR2',':en Resistivity Shallow');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('MRD',':en Resistivity Depth Deep');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('MRD2',':en Resistivity Depth Shallow');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('MRD2O',':en Resistivity Shallow Offset');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('MRDO',':en Resistivity Deep Offset');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('MS',':en Sonic :es Sónico :ru Звуковой :cn 声波 :pt Sônico');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('MSD',':en Sonic Depth');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('MSDO',':en Sonic Offset');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('MSE',':en MSE');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('MSEADJ',':en MSE Adjusted :es MSE Ajustado :pt MSE Ajustado');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('MSEER',':en MSE Efficiency Ratio :es Razón de Eficiencia de MSE :pt Razão de Eficiência MSE');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('MTDA',':en Delta Mud Temp. :es Temperatura de Lodo Delta :ru Дельта температуры бурового раствора :cn 泥浆温度差 :pt Temperatura da Lama Delta');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('MTFA',':en DH Fluid Temp Ann :es Temperatura de Fluido en Fondo del Hoyo (Anular) :ru Температура скважинных флюидов в кольце :cn 井下环空泥浆温度 :pt Temper. Fluido no Fundo do Poço (Anular)');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('MTFP',':en DH Fluid Temp Pipe :es Temperatura de Fluido en Fondo del Hoyo (Tuber໚) :ru Температура скважинных флюидов в трубе :cn 井下钻柱内泥浆温度 :pt Temper. Fluido no Fundo do Poço (no Tubo)');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('MTIA',':en Mud Temp. In :es Temperatura de Lodo de Entrada :ru Температура входящего бурового раствора :cn 入口泥浆温度 :pt Temper. Lama de Entrada');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('MTOA',':en Mud Temp. Out :es Temperatura de Lodo de Salida :ru Температура исходящего бурового раствора :cn 出口泥浆温度 :pt Temper. Lama de Sa');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('MVM',':en Mud Volume Moved :es Volumen de Lodo Movido :ru Объём перемещённого бурового раствора :cn 流动的泥浆体积 :pt Volume de Lama Movido');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('MVO',':en Mud Velocity');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('N2',':en N2 :ru Азот :cn 氮气');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('N2CB1',':en Nitrogen Out :es Nitrógeno de Salida :pt Nitrogênio de Sa');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('N2CB2',':en Nitrogen In :es Nitrógeno de Entrada :pt Nitrogênio de Entrada');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('N2CBD',':en Delta Nitrogen :es Delta Nitrógeno :pt Delta Nitrogênio');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('N2CR1',':en Nitrogen Out(Raw) :es Nitrógeno de Salida (Raw) :pt Nitrogênio de Sa (Raw)');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('N2CR2',':en Nitrogen In(Raw) :es Nitrógeno de Entrada (Raw) :pt Nitrogênio de Entrada (Raw)');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('N2CRD',':en Delta Nitrogen(Raw) :es Delta Nitrógeno (Raw) :pt Delta Nitrogênio (Raw)');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('N2CW1',':en QC Nitrogen Out :es QC Nitrógeno de Salida :pt QC Nitrogênio de Sa');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('N2CW2',':en QC Nitrogen In :es QC Nitrógeno de Entrada :pt QC Nitrogênio de Entrada');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('N2CWD',':en QC Delta Nitrogen :es QC Delta Nitrógeno :pt QC Delta Nitrogênio');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('N2\_1',':en N2 Out :es N2 de Salida :ru Азот исходящий :cn 排出氮气 :pt N2 de Sa');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('N2\_2',':en N2 In :es N2 de Entrada :ru Азот входящий :cn 输入氮气 :pt N2 de Entrada');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('NBUT',':en nC4 :ru Нео-Бутан :cn 正丁烷');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('NBUT1',':en nC4 Out :es NC4 de Salida :ru н-Бутан исходящий :cn 排出正丁烷 :pt nC4 de Sa');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('NBUT2',':en nC4 In :es NC4 de Entrada :ru н-Бутан входящий :cn 输入正丁烷 :pt nC4 de Entrada');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('NC4CB1',':en N-Butane Out :es N-Butano de Salida :pt N-Butano de Sa');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('NC4CB2',':en N-Butane In :es N-Butano de Entrada :pt N-Butano de Entrada');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('NC4CBD',':en Delta N-Butane :es Delta N-Butano :pt Delta N-Butano');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('NC4CR1',':en N-Butane Out(Raw) :es N-Butano de Salida (Raw) :pt N-Butano de Sa (Raw)');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('NC4CR2',':en N-Butane In(Raw) :es N-Butano de Entrada (Raw) :pt N-Butano de Entrada (Raw)');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('NC4CRD',':en Delta N-Butane(Raw) :es Delta N-Butano (Raw) :pt Delta N-Butano (Raw)');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('NC4CW1',':en N-Butane Out :es QC N-Butano de Salida :ru QC N-Butane Out :cn QC N-Butane Out :pt QC N-Butano de Sa');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('NC4CW2',':en N-Butane In :es QC N-Butano de Entrada :ru QC N-Butane In :cn QC N-Butane In :pt QC N-Butano de Entrada');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('NC4CWD',':en Delta N-Butane :es QC Delta N-Butano :ru QC Delta N-Butane :cn QC Delta N-Butane :pt QC Delta N-Butano');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('NC4FC',':en nC4 Flow corrected');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('NC5CB1',':en N-Pentane Out :pt N-Pentano de Sa');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('NC5CB2',':en N-Pentane In :pt N-Pentano de Entrada');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('NC5CBD',':en Delta N-Pentane :es N-Pentane Delta :pt Delta N-Pentano');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('NC5CR1',':en N-Pentane Out(Raw) :es N-Pentano de Salida (Raw) :pt N-Pentano de Sa (Raw)');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('NC5CR2',':en N-Pentane In(Raw) :es N-Pentano de Entrada (Raw) :pt N-Pentano de Entrada (Raw)');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('NC5CRD',':en Delta N-Pentane(Raw) :es Delta N-Pentano (Raw) :pt Delta N-Pentano (Raw)');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('NC5CW1',':en QC N-Pentane Out :es QC N-Pentano de Salida :pt QC N-Pentano de Sa');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('NC5CW2',':en QC N-Pentane In :es QC N-Pentano de Entrada :pt QC N-Pentano de Entrada');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('NC5CWD',':en QC Delta N-Pentane :es QC Delta N-Pentano :pt QC Delta N-Pentano');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('NC5FC',':en nC5 Flow Corrected');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('NDDMEA',':en Next Down Depth :es Próxima Profundidad de Fondo :ru Следующая глубина спуска :cn 方入 :pt Profundidade Próxima do Fundo');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('NPEN',':en nC5 :ru Нео-Пентан :cn 正戊烷');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('NPEN1',':en nC5 Out :es NC5 de Salida :ru н-Пентан исходящий :cn 排出正戊烷 :pt nC5 de Sa');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('NPEN2',':en nC5 In :es NC5 de Entrada :ru н-Пентан входящий :cn 输入正戊烷 :pt nC5 de Entrada');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('O2',':en O2 :ru Кислород :cn 氧气');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('O2\_1',':en O2 Out :es O2 de Salida :ru Кислород исходящий :cn 排出氧气 :pt O2 de Sa');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('O2\_2',':en O2 In :es O2 de Entrada :ru Кислород входящий :cn 输入氧气 :pt O2 de Entrada');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('OSLIPSTAT',':en Slips Override :es Superposición de Cu༚s :cn 取消卡瓦 :pt Sobreposição da Cunha');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('OVERPULL',':en Overpull :es Sobretensión (Overpull) :ru Сверхтяга :cn 过拉 :pt Overpull (Tração Excessiva)');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('PFG',':en % Free Gas');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('PFO',':en Pason Flow Out :es Pason - Flujo de Salida :ru Вытекающий пасон :cn Pason 流出量 :pt Fluxo Sa Pason');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('PLA',':en Pressure Loss Annulus :es Pérdida de Presión Anular :ru Кольцо потери давления :cn 环空压力损失 :pt Perda de Pressão Anular');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('PLB',':en Pressure Loss @ Bit :es Pérdida de Presión en la Mecha :ru Потеря давления @ бур :cn 钻头压力损失 :pt Perda de Pressão @ Broca');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('PLDS',':en Pressure Loss Pipe :es Pérdida de Presión en la Tuber໚ :ru Труба потери давления :cn 钻柱内压力损失 :pt Perda de Pressão Tubo');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('PLPB',':en % Pressure Loss @ Bit :es Pérdida % de Presión en la Mecha :ru % Потеря давления @ бур :cn %钻头压力损失 :pt Percentual de Perda de Pressão @ Broca');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('PLSU',':en Pressure Loss @ Surface :es Pérdida de Presión en Superficie :ru Потеря давления @ поверхность :cn 地面压力损失 :pt Perda de Pressão @ Superfໜie');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('PP1',':en Probe Pressure GCT OUT');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('PP2',':en Probe Pressure GCT IN');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('PPM',':en % Mud');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('PPRES',':en Pipe Pressure');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('PROFILE',':en Profile Data :es Datos del Perfil :ru Данные профиля :cn 结构数据 :pt Dados de Perfil');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('PRP',':en C3 :ru Пропан :cn 丙烷');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('PRP1',':en C3 Out :es C3 de Salida :ru Пропан исходящий :cn 排出丙烷 :pt C3 de Sa');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('PRP2',':en C3 In :es C3 de Entrada :ru Пропан входящий :cn 输入丙烷 :pt C3 de Entrada');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('PVO1',':en Pump Output 1 :es Salida de Bomba 1 :ru Мощность насоса 1 :cn 泵排򑳱 :pt Sa da Bomba 1');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('PVO2',':en Pump Output 2 :es Salida de Bomba 2 :ru Мощность насоса 2 :cn 泵排򑳲 :pt Sa da Bomba 2');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('PVO3',':en Pump Output 3 :es Salida de Bomba 3 :ru Мощность насоса 3 :cn 泵排򑳳 :pt Sa da Bomba 3');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('PVO4',':en Pump Output 4 :es Salida de Bomba 4 :ru Мощность насоса 4 :cn 泵排򑳴 :pt Sa da Bomba 4');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('RAT1',':en C1/(C3+C4+C5)');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('RAT2',':en (C3+C4+C5)/C1');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('RAT3',':en C1/C2');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('RAT4',':en C1/C3');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('RAT5',':en C1/C4');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('RAT6',':en C1/C5');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('RAT7',':en C2/C3');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('RAT8',':en C6/Benz');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('RAT9',':en C7/Tol');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('RAT10',':en iC4/nC4');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('RAT11',':en iC5/nC5');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('RAT12',':en (C2+C3)/C1');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('RAT13',':en (C1+C3)/C2');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('RAT14',':en (C1+C2)/C3');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('RAT15',':en (C4+C5)/(C1+C2) :pt (C4+C5)/(C1 C2)');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('RAT16',':en C1/㭌');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('RAT17',':en (C1+C4+C5)/(C1+C2+C4)');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('RAT18',':en (C1+C2)/(C3+C4+C5)');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('RAT19',':en (C4+C5)/C3');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('RAT20',':en (C2+C3+C4+C5)/δ\*100.0');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('RAT21',':en Gas/Liquid Ratio');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('RAT22',':en Aromatics/Alkanes');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('RAT23',':en HC');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('RAT24',':en C1/THC');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('RAT25',':en Fluid Saturation');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('RAT26',':en C1/C7');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('RAT27',':en C1/C8');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('RAT28',':en GIO3');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('RAT29',':en Fluid Saturation 2');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('RAT30',':en Heavies/Lights');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('RAT31',':en Heavies/C3');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('RAT32',':en GIO1');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('RAT33',':en GIO2');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('RATA',':en Ratio Analysis MLS');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('RATA2',':en Ratio Analysis GCT');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('RCO',':en Riser Correction Offset :es Desvío de Corrección del Riser :pt Correção de Desvio do Riser');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('RIGSTAT',':en Rig Status :es Estado del Taladro :ru Состояние бура :cn 钻机状态 :pt Situação da Plataforma');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('ROPA',':en ROP Average :es ROP Promedio :ru Средняя скорость проходки :cn 平均钻速 :pt ROP Mຝio');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('ROPI',':en ROP :ru Скорость проходки :cn 钻速');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('ROPU',':en User ROP :es ROP del Usuario :ru Пользовательская скорость проходки :cn 用户钻速 :pt ROP do Usuário');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('RPMI',':en RPM :ru Скорость вращения :cn 转速');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('RPMMI',':en Mud Motor RPM :es RPM de Motor de Fondo :ru Скорость вращения забойного двигателя :cn 泥浆马达转速 :pt RPM do Motor de Lama');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('RPMSI',':en Table RPM :es RPM de Mesa Rotaria :ru Скорость вращения стола :cn 转盘转速 :pt RPM da Mesa');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('RPMTI',':en Top Drive RPM :es RPM de Top Drive :ru Скорость верхнего привода :cn 顶驱转速 :pt RPM do Top Drive');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('RPOS',':en Riser Position :es Posición del Riser :ru Позиция шахтного ствола :cn 隔水管位置 :pt Posição do Riser');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('RSA',':en Running Speed :es Velocidad de Corrida :ru Рабочая скорость :cn 运行速度 :pt Velocidade de Funcionamento');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('RSDN',':en Running Speed Down :es Velocidad de Corrida Bajando :ru Снижение рабочей скорости :cn 下钻速度 :pt Velocidade de Funcionamento Baixa');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('RSUP',':en Running Speed Up :es Velocidad de Corrida Subiendo :ru Повышение рабочей скорости :cn 起钻速度 :pt Velocidade de Funcionamento Alta');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('SAZC',':en S.Azimuth :es S. Azimut :ru Азимут обследования :cn 测量方位角 :pt S. Azimutal');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('SDEN',':en Shale Density :es Densidad de Lutita :ru Плотность сланца :cn 页岩密度 :pt Densidade do Folhelho');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('SDS',':en Surface Down Strokes');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('SDT',':en Surface Down Time');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('SFAC',':en Shale Factor :es Factor de Lutita :ru Показатель сланца :cn 页岩因数 :pt Fator do Folhelho');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('SGTF',':en Gravity TF :es ToolFace Gravedad :ru Гравитационный циферблат :cn 重力工具面 :pt Gravidade TF');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('SIGMA',':en Sigmalog :ru Сигма :cn 西格马');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('SINC',':en S.Inclination :es Inclinación de Survey :ru Наклон обследования :cn 测量井斜 :pt Inclinação S');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('SLIPSTAT',':en Slip Status :es Estado de Cu༚s :ru Состояние клиньев :cn 卡瓦状态 :pt Situação da Cunha');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('SMTF',':en Magnetic TF :es ToolFace Magnético :ru Магнитный циферблат :cn 磁工具面 :pt TF Magnético');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('SO2',':en SO2 :ru Диоксид серы :cn 二氧化硫');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('SO2\_1',':en SO2 Out :es SO2 de Salida :ru Диоксид серы исходящий :cn 排出二氧化硫 :pt SO2 de Sa');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('SO2\_2',':en SO2 In :es SO2 de Entrada :ru Диоксид серы входящий :cn 输入二氧化硫 :pt SO2 de Entrada');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('SPHD',':en Delta Sulphides :es Delta Sulfatos :ru Дельта сульфиды :cn 硫化物差 :pt Delta Sulfetos');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('SPHI',':en Sulphides In :es Sulfuros de Entrada :ru Входящие сульфиды :cn 流入硫化物 :pt Sulfetos de Entrada');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('SPHO',':en Sulphides Out :es Sulfuros de Salida :ru Исходящие сульфиды :cn 流出硫化物 :pt Sulfetos de Sa');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('SPHT',':en Total Sulphides :es Sulfuros Total :ru Общие сульфиды :cn 总硫化物量 :pt Sulfetos Totais');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('SPM1',':en SPM 1 :ru Число двойных ходов в минуту 1 :cn 泵򐇱');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('SPM2',':en SPM 2 :ru Число двойных ходов в минуту 2 :cn 泵򐇲');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('SPM3',':en SPM 3 :ru Число двойных ходов в минуту 3 :cn 泵򐇳');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('SPM4',':en SPM 4 :ru Число двойных ходов в минуту 4 :cn 泵򐇴');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('SPPD',':en SPP Differential :es Presión de Bomba Diferencial :pt Diferencial SPP');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('SPPI',':en Standpipe Pressure :es Presión de Bomba :ru Давление на стояке :cn 立压 :pt Pressão do Stand Pipe');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('SRAFH',':en SRA Free Hydrocarbons');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('SRAHI',':en SRA Hydrogen Index (HI)');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('SRAHP',':en SRA Hydrocarbon Potential');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('SRAM',':en SRA Maturity');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('SRANO',':en SRA Normalized Oil Content');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('SRAOI',':en SRA OI');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('SRAOR',':en SRA Organic Richness');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('SRAPI',':en SRA Production Index (PI)');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('SRAS3',':en SRA S3');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('SRATT',':en SRA T TEMP');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('SSF',':en Stick-Slip Factor :es Factor de Stick-Slip :pt Fator de Aderência da Cunha');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('SSFX',':en Stick-Slip Factor(Max) :es Factor de Stick-Slip (Máximo) :pt Fator de Aderência da Cunha (Máximo)');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('STDGI',':en TGD In Sample Line Delay :es Tiempo de Atraso de Gas de Muestra de Entrada :ru Время задержки газа в образце :cn 气体进入分析仪的延迟时间 :pt Retardo de Entrada da Amostra de Gás');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('STDGO',':en TGD Out Sample Line Delay :es Tiempo de Atraso de Gas de Muestra de Salida :ru Время задержки газа из образца :cn 气体流出分析仪的延迟时间 :pt Retardo de Sa da Amostra de Gás');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('STK1',':en Strokes 1 :es Emboladas 1 :ru Перемещения 1 :cn 泵冲񥜁 :pt Movimento 1');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('STK2',':en Strokes 2 :es Emboladas 2 :ru Перемещения 2 :cn 泵冲񥜂 :pt Movimento 2');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('STK3',':en Strokes 3 :es Emboladas 3 :ru Перемещения 3 :cn 泵冲񥜃 :pt Movimento 3');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('STK4',':en Strokes 4 :es Emboladas 4 :ru Перемещения 4 :cn 泵冲񥜄 :pt Movimento 4');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('STKC',':en Total Strokes :es Emboladas Totales :ru Всего перемещений :cn 总泵冲数 :pt Movimentos Totais');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('STLEN',':en String Length :es Longitud de Tuber໚ de Perforación :ru Длина строки :cn 管串长度 :pt Comprimento da Coluna');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('STLENB',':en String Length @ Bottom');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('STWT',':en String Weight :es Peso de Tuber໚ de Perforación :ru Масса строки :cn 管串长度 :pt Peso da Coluna');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('SWSU',':en Surge/Swab Pressure :es Presión de Swap/Surge :ru Давление Волны/Сваба :cn 激动/抽吸压力 :pt Pressão do Limpador do Poço/Surgência');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('SWSUP',':en Surge Pressure :es Presión de Surge :ru Давление волны :cn 激动压力 :pt Pressão de Surgência');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('SWSWP',':en Swab Pressure :es Presión de Swap :ru Давление сваба :cn 抽吸压力 :pt Pressão do Limpador do Poço');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('TBSPM',':en Total Boost SPMs :es Emboladas Totales Booster :pt Boost SPMs Totais');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('TCB1',':en Toluene Out :es Tolueno de Salida :pt Tolueno de Sa');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('TCB2',':en Toluene In :es Tolueno de Entrada :pt Tolueno de Entrada');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('TCBD',':en Toluene Delta :es Delta Tolueno :ru Delta Toluene :cn Delta Toluene :pt Delta Tolueno');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('TCDGS',':en TCD Gas Sensor :es Sensor de Gas TCD :ru Газовый сенсор :cn 热导率气体分析仪 :pt Sensor de Gás TCD');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('TCDQC',':en TCD QC Ratio');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('TCR1',':en Toluene Out(Raw) :es Tolueno de Salida (Raw) :pt Tolueno de Sa (Raw)');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('TCR2',':en Toluene In(Raw) :es Tolueno de Entrada (Raw) :pt Tolueno de Entrada (Raw)');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('TCRD',':en Delta Toluene(Raw) :es Delta Tolueno (Raw) :pt Delta Tolueno (Raw)');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('TCS',':en Total Circulating Strokes :es Emboladas de Circulación Total :ru Всего циркулирующих перемещений :cn 总循环泵冲数 :pt Movimentos Totais de Circulação');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('TCT',':en Total Circulating Time :es Tiempo Total Circulando :ru Общее время циркуляции :cn 总循环时间 :pt Tempo Total de Circulação');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('TCV',':en Total Circulating Volume :es Volumen Total Circulando :ru Общая величина циркуляции :cn 总循环体积量 :pt Volume Total de Circulação');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('TCW1',':en QC Toluene Out :es QC Tolueno de Salida :pt QC Tolueno de Sa');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('TCW2',':en QC Toluene In :es QC Tolueno de Entrada :pt QC Tolueno de Entrada');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('TCWD',':en QC Delta Toluene :es QC Delta Tolueno :pt QC Delta Tolueno');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('TDRTM',':en Time to Lag :es Tiempo para Retraso :ru Время запаздывания :cn 迟到等待时间 :pt Tempo até o Retorno');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('TDRTMG',':en Time to GCT Out Lag :es Tiempo para Retraso Gas de Salida :ru Время задержки исходящего газа :cn 气体迟到等候时间 :pt Tempo até o Retorno de Sa do Gás');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('TGLD',':en Total Gas Lag Depth');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('TGLT',':en Total Gas Lag Time');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('TGTTL',':en Total Gas Time to Lag');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('THHP',':en Total Hydraulic HP :es Poder Hidráulico Total :ru Общее гидравлическое давление :cn 总水马力 :pt HP Hidráulico Total');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('THKLI',':en Theoretical Hookload :es Carga del Gancho Teorética :ru Теоретическая грузоподъёмность :cn 理论大钩载荷 :pt Carga no Gancho Teórica');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('TIME',':en Time :es Tiempo :ru Время :cn 时间 :pt Tempo');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('TPD',':en Pipe Displacement :es Desplazamiento de Tuber໚ :ru Замещение трубы :cn 钻柱顶替量 :pt Deslocamento do Tubo');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('TPL',':en Total Pressure Loss :es Pérdida de Presión Total :ru Общая потеря давления :cn 总压力损失 :pt Perda de Pressão Total');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('TPS',':en Time Per Stand :es Tiempo por Pareja :cn 时间/柱 :pt Tempo por Seção de Tubo');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('TPSA',':en Avg Time Per Stand :es Tiempo Promedio por Pareja :cn 平均时间/柱 :pt Tempo Mຝio por Seção de Tubo');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('TQEI',':en Torque Electric :es Torque Elຜtrico :ru Электрический крутящий момент :cn 扭矩电量 :pt Torque Elétrico');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('TQHI',':en Torque Hydraulic :es Torque Hidráulico :pt Torque Hidráulico');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('TQI',':en Torque :ru Крутящий момент :cn 扭矩');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('TRD',':en Trip Direction :es Dirección del Viaje :ru Направление рейса :cn 运行方向 :pt Direção da Manobra');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('TRF',':en Theoretical Return Flow');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('TRV',':en Theoretical Return Volume');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('TSIGMA',':en T.Sigmalog');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('TSPM',':en Total SPMs :es Emboladas Totales :ru Общее число двойных ходов в минуту :cn 总泵冲数');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('TV1',':en Pit 1 :es Tanque 1 :ru Шахта 1 :cn 泥浆񬘁 :pt Tanque 1');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('TV2',':en Pit 2 :es Tanque 2 :ru Шахта 2 :cn 泥浆񬘂 :pt Tanque 2');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('TV3',':en Pit 3 :es Tanque 3 :ru Шахта 3 :cn 泥浆񬘃 :pt Tanque 3');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('TV4',':en Pit 4 :es Tanque 4 :ru Шахта 4 :cn 泥浆񬘄 :pt Tanque 4');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('TV5',':en Pit 5 :es Tanque 5 :ru Шахта 5 :cn 泥浆񬘅 :pt Tanque 5');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('TV6',':en Pit 6 :es Tanque 6 :ru Шахта 6 :cn 泥浆񬘆 :pt Tanque 6');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('TV7',':en Pit 7 :es Tanque 7 :ru Шахта 7 :cn 泥浆񬘇 :pt Tanque 7');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('TV8',':en Pit 8 :es Tanque 8 :ru Шахта 8 :cn 泥浆񬘈 :pt Tanque 8');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('TV9',':en Pit 9 :es Tanque 9 :ru Шахта 9 :cn 泥浆񬘉 :pt Tanque 9');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('TV10',':en Pit 10 :es Tanque 10 :ru Шахта 10 :cn 泥浆� :pt Tanque 10');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('TV11',':en Pit 11 :es Tanque 11 :ru Шахта 11 :cn 泥浆� :pt Tanque 11');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('TV12',':en Pit 12 :es Tanque 12 :ru Шахта 12 :cn 泥浆� :pt Tanque 12');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('TV13',':en Pit 13 :es Tanque 13 :ru Шахта 13 :cn 泥浆� :pt Tanque 13');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('TV14',':en Pit 14 :es Tanque 15 :ru Шахта 14 :cn 泥浆� :pt Tanque 14');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('TV15',':en Pit 15 :es Tanque 15 :ru Шахта 15 :cn 泥浆� :pt Tanque 15');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('TV16',':en Pit 16 :es Tanque 16 :ru Шахта 16 :cn 泥浆� :pt Tanque 16');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('TV17',':en Trip Tank 1 :es Tanque de Viaje 1 :ru Резервуар для долива :cn 起下钻泥浆罐 :pt Tanque de Manobra 1');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('TV18',':en Trip Tank 2 :es Tanque de Viaje 2 :pt Tanque de Manobra 2');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('TVA',':en Annular Volume :es Volumen Anular :ru Кольцевой объём :cn 环空容积 :pt Volume Anular');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('TVAB',':en Dead Volume');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('TVD',':en TVD :ru Фактическая вертикальная глубина :cn 实际垂直井深');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('TVL1',':en Pit 1 Level :pt Nível do Tanque 1');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('TVL2',':en Pit 2 Level :pt Nível do Tanque 2');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('TVL3',':en Pit 3 Level :pt Nível do Tanque 3');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('TVL4',':en Pit 4 Level :pt Nível do Tanque 4');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('TVL5',':en Pit 5 Level :pt Nível do Tanque 5');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('TVL6',':en Pit 6 Level :pt Nível do Tanque 6');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('TVL7',':en Pit 7 Level :pt Nível do Tanque 7');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('TVL8',':en Pit 8 Level :pt Nível do Tanque 8');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('TVL9',':en Pit 9 Level :pt Nível do Tanque 9');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('TVL10',':en Pit 10 Level :pt Nível do Tanque 10');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('TVL11',':en Pit 11 Level :pt Nível do Tanque 11');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('TVL12',':en Pit 12 Level :pt Nível do Tanque 12');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('TVL13',':en Pit 13 Level :pt Nível do Tanque 13');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('TVL14',':en Pit 14 Level :pt Nível do Tanque 14');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('TVL15',':en Pit 15 Level :pt Nível do Tanque 15');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('TVL16',':en Pit 16 Level :pt Nível do Tanque 16');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('TVL17',':en Trip Tank 1 Level :es Nivel de Tanque 1 :pt Nível do Tanque de Manobra 1');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('TVL18',':en Trip Tank 2 Level :es Nivel de Tanque de Viaje 1 :pt Nível do Tanque de Manobra 2');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('TVP',':en Pipe Volume :es Volumen de Tuber໚ :ru Объём трубы :cn 钻杆容积 :pt Volume do Tubo');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('TVPB',':en Pipe Volume @ Bottom');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('TVR',':en Riser Volume :es Volumen del Riser :ru Объём стояка :cn 隔水管容积 :pt Volume do Riser');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('TVT1',':en Active PVT :es PVT Activo :ru Активный общий объём :cn 在用泥浆池总量 :pt PVT Ativo');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('TVT2',':en PVT 2 :ru Общий объём 2 :cn 泥浆池总򑳲');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('TVT3',':en PVT 3 :ru Общий объём 3 :cn 泥浆池总򑳳');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('TVT4',':en PVT 4 :ru Общий объём 4 :cn 泥浆池总򑳴');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('TVT5',':en Trip Tank Total :es Total de Tanque de Viaje :ru Общий резервуар для долива :cn 起下罐总量 :pt Total do Tanque de Manobra');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('TVTGL1',':en Active PVT G/L :es Ganancia/Pérdida de PVT Activo :ru Прирост/потеря активного общего объёма :cn 在用泥浆池总量增/减 :pt Ganho/Perda do PVT Ativo');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('TVTGL2',':en PVT 2 G/L :es Ganancia/Pérdida de PVT 2 :ru Прирост/потеря общего объёма 2 :cn 泥浆池总򑳲 增/减 :pt Ganho/Perda do PVT 2');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('TVTGL3',':en PVT 3 G/L :es Ganancia/Pérdida de PVT 3 :ru Прирост/потеря общего объёма 3 :cn 泥浆池总򑳳 增/减 :pt Ganho/Perda do PVT 3');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('TVTGL4',':en PVT 4 G/L :es Ganancia/Pérdida de PVT 4 :ru Прирост/потеря общего объёма 4 :cn 泥浆池总򑳴增/减 :pt Ganho/Perda do PVT 4');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('TVTGL5',':en Triptank G/L :es Ganancia/Pérdida de Tanque de Viaje :ru Прирост/потеря в резервуаре для долива :cn 起下罐增减 :pt Ganho/Perda do Tanque de Manobra');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('TVTGLR1',':en Active PVT G/L Rate :es Rata de Ganancia/Pérdida de PVT Activo :ru Скорость прироста/потери активного общего объёма :cn 在用泥浆池总量增/减速率 :pt Taxa de Ganho/Perda do PVT Ativo');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('TVTGLR2',':en PVT2 G/L Rate :es Rata de Ganancia/Pérdida de PVT 2 :ru Скороть прироста/потери общего объёма 2 :cn 泥浆池总򑳲 增/减速率 :pt Taxa de Ganho/Perda do PVT 2');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('TVTGLR3',':en PVT3 G/L Rate :es Rata de Ganancia/Pérdida de PVT 3 :ru Скороть прироста/потери общего объёма 3 :cn 泥浆池总򑳳 增/减速率 :pt Taxa de Ganho/Perda do PVT 3');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('TVTGLR4',':en PVT4 G/L Rate :es Rata de Ganancia/Pérdida de PVT 4 :ru Скороть прироста/потери общего объёма 4 :cn 泥浆池总򑳴增/减速率 :pt Taxa de Ganho/Perda do PVT 4');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('TVTGLR5',':en PVT5 G/L Rate :es Rata de Ganancia/Pérdida de PVT 5 :ru Скороть прироста/потери общего объёма 5 :cn 泥浆池总򑳵 增/减速率 :pt Taxa de Ganho/Perda do PVT 5');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('TZ',':en Timezone :es Uso Horario :ru Часовой пояс :cn 时区 :pt Fuso Horário');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('UD1',':en User Defined 1 :es Definido por el Usuario 1 :ru Определено пользователем 1 :cn 已有用񢍱 :pt Definido pelo Usuário 1');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('UD2',':en User Defined 2 :es Definido por el Usuario 2 :ru Определено пользователем 2 :cn 已有用񢍲 :pt Definido pelo Usuário 2');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('UD3',':en User Defined 3 :es Definido por el Usuario 3 :ru Определено пользователем 3 :cn 已有用񢍳 :pt Definido pelo Usuário 3');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('UD4',':en User Defined 4 :es Definido por el Usuario 4 :ru Определено пользователем 4 :cn 已有用񢍴 :pt Definido pelo Usuário 4');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('UD5',':en User Defined 5 :es Definido por el Usuario 5 :ru Определено пользователем 5 :cn 已有用񢍵 :pt Definido pelo Usuário 5');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('UD6',':en User Defined 6 :es Definido por el Usuario 6 :ru Определено пользователем 6 :cn 已有用񢍶 :pt Definido pelo Usuário 6');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('UD7',':en User Defined 7 :es Definido por el Usuario 7 :ru Определено пользователем 7 :cn 已有用񢍷 :pt Definido pelo Usuário 7');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('UD8',':en User Defined 8 :es Definido por el Usuario 8 :ru Определено пользователем 8 :cn 已有用񢍸 :pt Definido pelo Usuário 8');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('UD9',':en User Defined 9 :es Definido por el Usuario 9 :ru Определено пользователем 9 :cn 已有用񢍹 :pt Definido pelo Usuário 9');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('UD10',':en User Defined 10 :es Definido por el Usuario 10 :ru Определено пользователем 10 :cn 已有用� :pt Definido pelo Usuário 10');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('UD11',':en User Defined 11 :es Definido por el Usuario 11 :ru Определено пользователем 11 :cn 已有用� :pt Definido pelo Usuário 11');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('UD12',':en User Defined 12 :es Definido por el Usuario 12 :ru Определено пользователем 12 :cn 已有用� :pt Definido pelo Usuário 12');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('UD13',':en User Defined 13 :es Definido por el Usuario 13 :ru Определено пользователем 13 :cn 已有用� :pt Definido pelo Usuário 13');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('UD14',':en User Defined 14 :es Definido por el Usuario 14 :ru Определено пользователем 14 :cn 已有用� :pt Definido pelo Usuário 14');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('UD15',':en User Defined 15 :es Definido por el Usuario 15 :ru Определено пользователем 15 :cn 已有用� :pt Definido pelo Usuário 15');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('UD16',':en User Defined 16 :es Definido por el Usuario 16 :ru Определено пользователем 16 :cn 已有用� :pt Definido pelo Usuário 16');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('UR1',':en User Ratio 1 :es Razón del Usuario 1 :ru Соотношение пользователя 1 :cn 用户比񳡱 :pt Razão do Usuário 1');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('UR2',':en User Ratio 2 :es Razón del Usuario 2 :ru Соотношение пользователя 2 :cn 用户比񳡲 :pt Razão do Usuário 2');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('UR3',':en User Ratio 3 :es Razón del Usuario 3 :ru Соотношение пользователя 3 :cn 用户比񳡳 :pt Razão do Usuário 3');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('UR4',':en User Ratio 4 :es Razón del Usuario 4 :ru Соотношение пользователя 4 :cn 用户比񳡴 :pt Razão do Usuário 4');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('VFR',':en Volumetric Flow Rate');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('WDIR',':en Wind Direction :es Dirección del Viento :ru Направление ветра :cn 风向 :pt Direção do Vento');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('WOBI',':en WOB :ru Масса на буре :cn 钻压');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('WRAT',':en Wet R MLS');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('WRAT2',':en Wet R GCT');

**INSERT** **INTO** DCDM\_WellLogFieldDesc(ParamName,Descr) **VALUES** ('WSPD',':en Wind Speed :es Velocidad del Viento :ru Скорость ветра :cn 风速 :pt Velocidade do Vento');

*--##############################################################################################*

**Create** **table** DCDM\_WellLog(

  Id          bigserial     NOT NULL **PRIMARY** **KEY**,                                             *-- Идентификатор*

  WellId      **int**           NOT NULL **REFERENCES** DCDM\_Well(Id) **ON** **DELETE** **CASCADE**,              *-- Идентификатор скважины.*

  UserId      **int**           NULL **REFERENCES** DCDM\_Users(Id) **ON** **DELETE** **SET** NULL,                *-- Ответственный за данные пользователь (если это пользователь добавляет данные)*

**Mode**        **smallint**      NOT NULL,                                                         *-- Код режима работы установки (0-неизвесно 1-бурение)*

**Time**        **timestamp**     NULL **DEFAULT** **CURRENT\_TIMESTAMP**,                                   *-- Дата и время события*

  DelayMSec   **int**           NULL,                                                             *-- Задержка с момента предыдущего измерения этого же параметра в миллисекундах (NULL если нет такий информации)*

  Depth       float8        NULL,                                                             *-- Глубина*

  VDepth      float8        NULL,                                                             *-- Вертикальная(абсолютная) глубина*

  ParamName   **varchar**(20)   NOT NULL,                                                         *-- Название параметра*

  Val         jsonb         NOT NULL,                                                         *-- Значение*

  isnew       **boolean**       NULL,                                                             *-- Флаг новых данных*

  isold       **boolean**       NULL                                                              *-- флаг усторевших данных*

);

**GRANT** **ALL** **PRIVILEGES** **ON** **TABLE** DCDM\_WellLog **TO** dcdmsrv;

**CREATE** **INDEX** dcdm\_welllog\_dpidx **ON** dcdm\_welllog (wellid,depth);                               *-- Индекс для ускорения выборки по максимальной глубине (без параметров)*

**CREATE** **INDEX** dcdm\_welllog\_baseidx **ON** dcdm\_welllog (wellid,paramname,depth);                   *-- Индекс для ускорения выборки по парметрам и глубине*

**CREATE** **INDEX** dcdm\_welllog\_basetidx **ON** dcdm\_welllog (wellid,paramname,**time**);                   *-- Индекс для ускорения выборки по параметрам и времени*

**CREATE** **INDEX** dcdm\_welllog\_prmwellidx **ON** dcdm\_welllog (paramname,wellid);                      *-- Индекс для ускорения выборки по параметрам*

**CREATE** **INDEX** dcdm\_welllog\_isnew **ON** dcdm\_welllog (isnew);                                      *-- Оба      индекса          поиск                    операций с БД*

**CREATE** **INDEX** dcdm\_welllog\_isold **ON** dcdm\_welllog (isold);                                      *--     этих         ускоряют       назафиксированнных*

*--##############################################################################################*

**Create** **table** DCDM\_CommentsAndFiles(

  Id          bigserial     NOT NULL **PRIMARY** **KEY**,                                             *-- Идентификатор*

  WellId      **int**           NOT NULL **REFERENCES** DCDM\_Well(Id) **ON** **DELETE** **CASCADE**,              *-- Идентификатор скважины.*

  UserId      **int**           NULL **REFERENCES** DCDM\_Users(Id) **ON** **DELETE** **SET** NULL,                *-- Пользователь добавивший комментарий*

  CreateTime  **timestamp**     NOT NULL **DEFAULT** **CURRENT\_TIMESTAMP**,                               *-- Время создания/последнего изменения*

  ChangeTime  **timestamp**     NULL,                                                             *-- Время последнего изменения*

**Type**        **smallint**      NOT NULL **REFERENCES** DCDM\_CommentsAndFilesType(Id),                *-- Тип комментария*

  IsIndexed   **boolean**       NOT NULL **DEFAULT** false,                                           *-- Индексирована ли запись поисковой системой*

  SnapDepth   float8        NULL,                                                             *-- Привязка к глубине*

  SnapVDepth  float8        NULL,                                                             *-- Привязка к вертикальной(абсолютная) глубине*

  SnapTime    **timestamp**     NULL,                                                             *-- Привязка к дате и времи*

**Comment**     **varchar**       NULL,                                                             *-- Комментарий*

  OrgFileName **varchar**       NULL,                                                             *-- Оригинальное имя файла*

  FileData    bytea         NULL                                                              *-- Данные файла*

);

**GRANT** **ALL** **PRIVILEGES** **ON** **TABLE** DCDM\_CommentsAndFiles **TO** dcdmsrv;

*--##############################################################################################*

*--Различные параметры конфигурации*

**Create** **table** DCDM\_Config(

**Key**        **varchar**(20)    NOT NULL **PRIMARY** **KEY**,                                             *-- Идентификатор*

**Data**       **varchar**        NULL                                                              *-- Значение параметра*

);

**GRANT** **ALL** **PRIVILEGES** **ON** **TABLE** DCDM\_Config **TO** dcdmsrv;

**INSERT** **INTO** dcdm\_config(**key**, **data**) **VALUES** ('LYTHCODEMAP','{

"13":{"f":"anhydrite.png","n":":en Anhydrite :ru Ангидрит :es Anhidrita"},

"20":{"f":"chert.png","n":":en Chert :ru Кремнистый известняк :es Esquisto"},

"18":{"f":"claystone.png","n":":en Claystone :ru Аргиллит :es Arcilla"},

"15":{"f":"coal.png","n":":en Coal :ru Уголь :es Carbón"},

"14":{"f":"conglomerate.png","n":":en Conglomerate :ru Конгломерат :es Conglomerado"},

"1":{"f":"dolomite.png","n":":en Dolomite :ru Доломит :es Dolomita"},

"19":{"f":"gypsum.png","n":":en Gypsum :ru Гипс :es Yeso"},

"255":{"f":"halite.png","n":":en Halite :ru Галит(Каменная соль) :es Hálito"},

"16":{"f":"limestone.png","n":":en Limestone :ru Известняк :es Caliza"},

"22":{"f":"metamorphic.png","n":":en Metamorphic :ru Метаморфик :es Metamórfica"},

"12":{"f":"plutonic.png","n":":en Plutonic :ru Плутоническая :es Plutónica"},

"10":{"f":"sand.png","n":":en Sand :ru Песок :es Arena"},

"17":{"f":"shale.png","n":":en Shale :ru Сланец :es Esquisto"},

"11":{"f":"siltstone.png","n":":en Siltstone :ru Алевролиты :es Piedra de limo"},

"7":{"f":"tuff.png","n":":en Tuff :ru Туф :es Toba"},

"21":{"f":"volcanic.png","n":":en Volcanic :ru Вулканический :es Volcánica"},

"9999":{"f":"unknown.png","n":":en Unknown :ru Неизвестная :es Desconocida"}}');

**INSERT** **INTO** dcdm\_config(**key**, **data**) **VALUES** ('LYTHNAMEMAP','{

"LTY\_Anhydrite":{"f":"anhydrite.png","n":":en Anhydrite :ru Ангидрит :es Anhidrita"},

"LTY\_Chert":{"f":"chert.png","n":":en Chert :ru Кремнистый известняк :es Esquisto"},

"LTY\_Claystone":{"f":"claystone.png","n":":en Claystone :ru Аргиллит :es Arcilla"},

"LTY\_Coal":{"f":"coal.png","n":":en Coal :ru Уголь :es Carbón"},

"LTY\_Conglomerate":{"f":"conglomerate.png","n":":en Conglomerate :ru Конгломерат :es Conglomerado"},

"LTY\_Dolomite":{"f":"dolomite.png","n":":en Dolomite :ru Доломит :es Dolomita"},

"LTY\_Gypsum":{"f":"gypsum.png","n":":en Gypsum :ru Гипс :es Yeso"},

"LTY\_Halite":{"f":"halite.png","n":":en Halite :ru Галит(Каменная соль) :es Hálito"},

"LTY\_Limestone":{"f":"limestone.png","n":":en Limestone :ru Известняк :es Caliza"},

"LTY\_Metamorphic":{"f":"metamorphic.png","n":":en Metamorphic :ru Метаморфик :es Metamórfica"},

"LTY\_Plutonic":{"f":"plutonic.png","n":":en Plutonic :ru Плутоническая :es Plutónica"},

"LTY\_Sand":{"f":"sand.png","n":":en Sand :ru Песок :es Arena"},

"LTY\_Shale":{"f":"shale.png","n":":en Shale :ru Сланец :es Esquisto"},

"LTY\_Siltstone":{"f":"siltstone.png","n":":en Siltstone :ru Алевролиты :es Piedra de limo"},

"LTY\_Tuff":{"f":"tuff.png","n":":en Tuff :ru Туф :es Toba"},

"LTY\_Volcanic":{"f":"volcanic.png","n":":en Volcanic :ru Вулканический :es Volcánica"},

"LTY\_Unknown":{"f":"unknown.png","n":":en Unknown :ru Неизвестная :es Desconocida"}}');

*--##############################################################################################*

**Create** **table** DCDM\_DBMUserScripts(

  id           uuid         **PRIMARY** **KEY** **DEFAULT** uuid\_generate\_v4(),                            *-- Идентификатор*

  UserId       **int**          NULL **REFERENCES** DCDM\_Users(Id) **ON** **DELETE** **CASCADE**,                 *-- Пользователь добавивший комментарий*

  ChangeTime   **timestamp**    NOT NULL **DEFAULT** **CURRENT\_TIMESTAMP**,                               *-- Время последнего изменения*

  Name         **varchar**      NULL,                                                             *-- Имя*

  Script       json         NULL,                                                             *-- Скрипт*

  IsPublic     **boolean**      NOT NULL **default** false                                            *-- Публичный*

);

**GRANT** **ALL** **PRIVILEGES** **ON** **TABLE** DCDM\_DBMUserScripts **TO** dcdmsrv;

*--##############################################################################################*

**Create** **table** DCDM\_MLib(

  FileName    **varchar**(50)   NOT NULL **CHECK**(FileName ~ '^[a-z0-9\_-]+(\.[a-z0-9\_-]{1,})?$'),*-- Имя файла*

  Dir         **varchar**(50)   NULL **CHECK**(Dir ~ '^[a-z0-9\_-]+$'),                            *-- Каталог*

  ChangeTime  **timestamp**     NULL **DEFAULT** **CURRENT\_TIMESTAMP**,                               *-- Время последнего изменения*

  FileData    bytea         NULL,                                                              *-- Данные файла*

**PRIMARY** **KEY**(FileName,Dir)

);

**GRANT** **ALL** **PRIVILEGES** **ON** **TABLE** DCDM\_MLib **TO** dcdmsrv;

**INSERT** **INTO** DCDM\_MLib(Dir,ChangeTime,FileName) **VALUES** ('rocks',NULL,'dolomite.png');

**INSERT** **INTO** DCDM\_MLib(Dir,ChangeTime,FileName) **VALUES** ('rocks',NULL,'tuff.png');

**INSERT** **INTO** DCDM\_MLib(Dir,ChangeTime,FileName) **VALUES** ('rocks',NULL,'sand.png');

**INSERT** **INTO** DCDM\_MLib(Dir,ChangeTime,FileName) **VALUES** ('rocks',NULL,'siltstone.png');

**INSERT** **INTO** DCDM\_MLib(Dir,ChangeTime,FileName) **VALUES** ('rocks',NULL,'plutonic.png');

**INSERT** **INTO** DCDM\_MLib(Dir,ChangeTime,FileName) **VALUES** ('rocks',NULL,'anhydrite.png');

**INSERT** **INTO** DCDM\_MLib(Dir,ChangeTime,FileName) **VALUES** ('rocks',NULL,'conglomerate.png');

**INSERT** **INTO** DCDM\_MLib(Dir,ChangeTime,FileName) **VALUES** ('rocks',NULL,'coal.png');

**INSERT** **INTO** DCDM\_MLib(Dir,ChangeTime,FileName) **VALUES** ('rocks',NULL,'limestone.png');

**INSERT** **INTO** DCDM\_MLib(Dir,ChangeTime,FileName) **VALUES** ('rocks',NULL,'shale.png');

**INSERT** **INTO** DCDM\_MLib(Dir,ChangeTime,FileName) **VALUES** ('rocks',NULL,'claystone.png');

**INSERT** **INTO** DCDM\_MLib(Dir,ChangeTime,FileName) **VALUES** ('rocks',NULL,'gypsum.png');

**INSERT** **INTO** DCDM\_MLib(Dir,ChangeTime,FileName) **VALUES** ('rocks',NULL,'chert.png');

**INSERT** **INTO** DCDM\_MLib(Dir,ChangeTime,FileName) **VALUES** ('rocks',NULL,'volcanic.png');

**INSERT** **INTO** DCDM\_MLib(Dir,ChangeTime,FileName) **VALUES** ('rocks',NULL,'metamorphic.png');

**INSERT** **INTO** DCDM\_MLib(Dir,ChangeTime,FileName) **VALUES** ('rocks',NULL,'halite.png');

**INSERT** **INTO** DCDM\_MLib(Dir,ChangeTime,FileName) **VALUES** ('rocks',NULL,'unknown.png');

**INSERT** **INTO** DCDM\_MLib(Dir,ChangeTime,FileName) **VALUES** ('rocks',NULL,'undefined.png');

**INSERT** **INTO** DCDM\_MLib(Dir,ChangeTime,FileName) **VALUES** ('snd',NULL,'alert\_caution.mp3');

**INSERT** **INTO** DCDM\_MLib(Dir,ChangeTime,FileName) **VALUES** ('snd',NULL,'alert\_danger.mp3');

**INSERT** **INTO** DCDM\_MLib(Dir,ChangeTime,FileName) **VALUES** ('snd',NULL,'alert\_norma.mp3');

*--##############################################################################################*

**Create** **table** DCDM\_MarkersType(

Id  **smallint**            NOT NULL **PRIMARY** **KEY**,                     *-- Идентификатор*

Name **varchar**            NOT NULL **UNIQUE**                           *-- Название (мультиязычная строка)*

);

**GRANT** **ALL** **PRIVILEGES** **ON** **TABLE** DCDM\_MarkersType **TO** dcdmsrv;

**INSERT** **INTO** DCDM\_MarkersType(Id,Name) **VALUES** ( 1,':en Marker 1 :ru Маркер 1 :es Marcador 1');

**INSERT** **INTO** DCDM\_MarkersType(Id,Name) **VALUES** ( 2,':en Marker 2 :ru Маркер 2 :es Marcador 2');

**INSERT** **INTO** DCDM\_MarkersType(Id,Name) **VALUES** ( 3,':en Marker 3 :ru Маркер 3 :es Marcador 3');

**INSERT** **INTO** DCDM\_MarkersType(Id,Name) **VALUES** ( 4,':en Marker 4 :ru Маркер 4 :es Marcador 4');

**INSERT** **INTO** DCDM\_MarkersType(Id,Name) **VALUES** ( 5,':en Marker 5 :ru Маркер 5 :es Marcador 5');

*--##############################################################################################*

**CREATE** **TYPE** floatrange **AS** RANGE (subtype=float8,subtype\_diff=float8mi);

**Create** **table** DCDM\_Markers(

  Id          bigserial     NOT NULL **PRIMARY** **KEY**,                                             *-- Идентификатор*

  WellId      **int**           NOT NULL **REFERENCES** DCDM\_Well(Id) **ON** **DELETE** **CASCADE**,              *-- Идентификатор скважины.*

  UserId      **int**           NULL **REFERENCES** DCDM\_Users(Id) **ON** **DELETE** **SET** NULL,                *-- Пользователь добавивший маркер*

  CreateTime  **timestamp**     NOT NULL **DEFAULT** **CURRENT\_TIMESTAMP**,                               *-- Время создания*

  ChangeTime  **timestamp**     NULL,                                                             *-- Время последнего изменения*

**Type**        **smallint**      NOT NULL **REFERENCES** DCDM\_MarkersType(Id),                         *-- Тип маркера*

  Depth       floatrange    NULL,                                                             *-- Диапазон привязки к глубине*

  VDepth      floatrange    NULL,                                                             *-- Диапазон привязки к вертикальной(абсолютная) глубине*

**Time**        tsrange       NULL,                                                             *-- Диапазон привязки к дате и времи*

  Code        **int**           NOT NULL                                                          *-- Код маркера*

);

**GRANT** **ALL** **PRIVILEGES** **ON** **TABLE** DCDM\_Markers **TO** dcdmsrv;

*--##############################################################################################*

*--##############################################################################################*

*--##############################################################################################*

*--##############################################################################################*

*--##############################################################################################*

*--##############################################################################################*

*--##############################################################################################*

*--##############################################################################################*

*--##############################################################################################*

*--##############################################################################################*

*--##############################################################################################*

*--##############################################################################################*

*--##############################################################################################*

*--GRANT USAGE, SELECT ON ALL SEQUENCES IN SCHEMA public TO dcdmsrv;*

*--##############################################################################################*

**CREATE** OR **REPLACE** **FUNCTION** GetUserList(i\_isadmin **bool**)

**RETURNS** **TABLE**(id dcdm\_users.id%**TYPE**,login dcdm\_users.login%**TYPE**,name dcdm\_users.name%**TYPE**,

                         permission dcdm\_users.permission%**TYPE**,isdisabled dcdm\_users.isdisabled%**TYPE**,

           latestuse dcdm\_users.latestuse%**TYPE**,acgroupname dcdm\_accessgroups.name%**TYPE**)

IMMUTABLE **LANGUAGE** 'plpgsql' **AS** $$

**BEGIN**

**RETURN** QUERY **select** DCDM\_Users.Id,

                     DCDM\_Users.Login,

                                      DCDM\_Users.Name,

                                      DCDM\_Users.Permission,

                                      DCDM\_Users.IsDisabled,

                                      DCDM\_Users.LatestUse,

                                      dcdm\_accessgroups.name **as** acgroupname

**from** DCDM\_Users

**left** **outer** **join** dcdm\_accessgroups **on** dcdm\_accessgroups.id=dcdm\_users.acgroup

**where** i\_isadmin=true OR (i\_isadmin=false AND POSITION('A' IN DCDM\_Users.Permission)=0 )

**order** **by** DCDM\_Users.Id;

**END**

$$;

**GRANT** **EXECUTE** **ON** **FUNCTION**  GetUserList(**bool**) **TO** dcdmsrv;

*--##############################################################################################*

**CREATE** OR **REPLACE** **FUNCTION** GetUserItem(i\_isadmin **bool**,i\_id dcdm\_users.id%**TYPE**)

**RETURNS** **TABLE**(id dcdm\_users.id%**TYPE**,login dcdm\_users.login%**TYPE**,name dcdm\_users.name%**TYPE**,

                         permission dcdm\_users.permission%**TYPE**,lang dcdm\_users.lang%**TYPE**,isdisabled dcdm\_users.isdisabled%**TYPE**,

                         acgroup dcdm\_users.acgroup%**TYPE**,latestuse dcdm\_users.latestuse%**TYPE**)

IMMUTABLE **LANGUAGE** 'plpgsql' **AS** $$

**BEGIN**

**RETURN** QUERY **select** DCDM\_Users.Id,

                     DCDM\_Users.Login,

                                      DCDM\_Users.Name,

                                      DCDM\_Users.Permission,

                                      DCDM\_Users.Lang,

                                      DCDM\_Users.IsDisabled,

                                      DCDM\_Users.AcGroup,

                                      DCDM\_Users.LatestUse

**from** DCDM\_Users

**where** DCDM\_Users.Id=i\_id AND (i\_isadmin=true OR (i\_isadmin=false AND POSITION('A' IN DCDM\_Users.Permission)=0));

**END**

$$;

**GRANT** **EXECUTE** **ON** **FUNCTION**  GetUserItem(**bool**,dcdm\_users.id%**TYPE**) **TO** dcdmsrv;

*--##############################################################################################*

**CREATE** OR **REPLACE** **FUNCTION** AddChangeUserData(i\_isadmin **bool**,i\_id dcdm\_users.id%**TYPE**,i\_login dcdm\_users.login%**TYPE**,i\_name dcdm\_users.name%**TYPE**,

                                                     i\_perm dcdm\_users.permission%**TYPE**,i\_lang dcdm\_users.lang%**TYPE**,i\_hash dcdm\_users.passhash%**TYPE**,

                                                     i\_salt dcdm\_users.salt%**TYPE**,i\_disflg dcdm\_users.isdisabled%**TYPE**,i\_acgroup dcdm\_users.acgroup%**TYPE**)

**RETURNS** **TABLE**(id dcdm\_users.id%**TYPE**,login dcdm\_users.login%**TYPE**,name dcdm\_users.name%**TYPE**,

                         permission dcdm\_users.permission%**TYPE**,isdisabled dcdm\_users.isdisabled%**TYPE**,

           latestuse dcdm\_users.latestuse%**TYPE**,acgroupname dcdm\_accessgroups.name%**TYPE**)

VOLATILE **LANGUAGE** 'plpgsql' **AS** $$

**BEGIN**

*-- Проверим корректны ли права редактируемого/добавляемого пользователя*

**if** POSITION('A' IN i\_perm)>0 AND i\_isadmin=false **then** RAISE EXCEPTION 'You have not enough permission to set this privileges' **USING** HINT='DCDM', DETAIL='PermError';**end** **if**;

**if** i\_id=0 **then**

*-- Добавление*

**INSERT** **INTO** dcdm\_users(login,name,passhash,salt,permission,lang,isdisabled,acgroup)

**VALUES** (i\_login,i\_name,i\_hash,i\_salt,i\_perm,i\_lang,i\_disflg,i\_acgroup);

**select** lastval() **into** i\_id;

**else**

*-- Редактирование*

**if** length(**coalesce**(i\_hash,''))>0 AND length(**coalesce**(i\_salt,''))>0

**then** **UPDATE** dcdm\_users **SET** passhash=i\_hash, salt=i\_salt **WHERE** dcdm\_users.id=i\_id;

**end** **if**;

**UPDATE** dcdm\_users **SET** login=i\_login, name=i\_name, permission=i\_perm,

                                              lang=i\_lang, isdisabled=i\_disflg, acgroup=**case** **when** i\_acgroup=0   **then** null **else** i\_acgroup **end**

**WHERE** dcdm\_users.id=i\_id;

**end** **if**;

*-- Вернём отредактированный результат*

**RETURN** QUERY **select** dcdm\_users.id,dcdm\_users.login,dcdm\_users.name,dcdm\_users.permission,dcdm\_users.isdisabled,dcdm\_users.latestuse,dcdm\_accessgroups.name **as** acgroupname

**from** dcdm\_users **left** **outer** **join** dcdm\_accessgroups **on** dcdm\_accessgroups.id=dcdm\_users.acgroup

**where** dcdm\_users.id=i\_id;

EXCEPTION **WHEN** unique\_violation **then** RAISE EXCEPTION 'Login must be unique' **USING** HINT='DCDM', DETAIL='LoginError';

**END**

$$;

**GRANT** **EXECUTE** **ON** **FUNCTION**  AddChangeUserData(**bool**,dcdm\_users.id%**TYPE**,dcdm\_users.login%**TYPE**,dcdm\_users.name%**TYPE**,dcdm\_users.permission%**TYPE**,

                                                                                    dcdm\_users.lang%**TYPE**,dcdm\_users.passhash%**TYPE**,dcdm\_users.salt%**TYPE**,dcdm\_users.isdisabled%**TYPE**,

                                                                                    dcdm\_users.acgroup%**TYPE**) **TO** dcdmsrv;

*--##############################################################################################*

**CREATE** OR **REPLACE** **FUNCTION** RemoveUserItem(i\_isadmin **bool**,i\_id dcdm\_users.id%**TYPE**)

**RETURNS** **integer** VOLATILE **LANGUAGE** 'plpgsql' **AS** $$

**DECLARE**

ResulrCount **int**;

**BEGIN**

**DELETE** **FROM** dcdm\_users **WHERE** dcdm\_users.id=i\_id AND ((POSITION('A' IN dcdm\_users.permission)>0 AND i\_isadmin=true) OR POSITION('A' IN dcdm\_users.permission)=0);

GET DIAGNOSTICS ResulrCount:=ROW\_COUNT;

**RETURN** ResulrCount;

**END**

$$;

**GRANT** **EXECUTE** **ON** **FUNCTION**  RemoveUserItem(**bool**,dcdm\_users.id%**TYPE**) **TO** dcdmsrv;

*--##############################################################################################*

**SET** plpgsql.extra\_warnings **TO** 'shadowed\_variables';

**CREATE** OR **REPLACE** **FUNCTION** AddChangeVarData(

i\_orgname dcdm\_vars.sysname%**TYPE**,

i\_name dcdm\_vars.sysname%**TYPE**,

i\_type dcdm\_vars.**type**%**TYPE**,

i\_init dcdm\_vars.initexpr%**TYPE**,

i\_calc dcdm\_vars.calcexpr%**TYPE**,

i\_tlrdvg dcdm\_vars.tlrdvg%**TYPE**,

i\_loggedpar dcdm\_vars.loggedpar%**TYPE**,

i\_isext dcdm\_vars.isexternal%**TYPE**,

i\_timeout dcdm\_vars.timeout%**TYPE**,

i\_avgmethod dcdm\_vars.avgmethod%**TYPE**,

i\_desc dcdm\_vars.description%**TYPE**,

i\_lag dcdm\_vars.lag%**TYPE**,

i\_group dcdm\_vars.groupid%**TYPE**,

i\_filter dcdm\_vars.filter%**TYPE**

)

**RETURNS** **TABLE**(

sysname dcdm\_vars.sysname%**TYPE**,

**type** dcdm\_vartype.name%**TYPE**,

groupid dcdm\_vars.groupid%**TYPE**,

groupname dcdm\_vargroups.name%**TYPE**,

description dcdm\_vars.description%**TYPE**,

loggedpar dcdm\_vars.loggedpar%**TYPE**,

isexternal dcdm\_vars.isexternal%**TYPE**,

avgmethod dcdm\_vars.avgmethod%**TYPE**,

sortpos dcdm\_vars.sortpos%**TYPE**)

VOLATILE **LANGUAGE** 'plpgsql' **AS** $$

**DECLARE**

 sp dcdm\_vars.sortpos%**TYPE**;

**BEGIN**

**IF** i\_orgname IS NULL **then**

*-- Добавление новой переменной*

*-- Получим максимальное значение позиции сортировки*

**select** **COALESCE**(MAX(dcdm\_vars.sortpos),0)+1 **from** dcdm\_vars **into** sp;

*-- Добавим переменную*

**INSERT** **INTO** dcdm\_vars(sysname, groupid, **type**, initexpr, calcexpr, tlrdvg, loggedpar, isexternal, timeout, avgmethod, description, sortpos,lag,filter)

**VALUES** (i\_name,i\_group,i\_type,i\_init,i\_calc,i\_tlrdvg,i\_loggedpar,i\_isext,i\_timeout,i\_avgmethod,i\_desc,sp,i\_lag,i\_filter);

**else**

*-- редактирование*

**UPDATE** dcdm\_vars **SET** sysname=i\_name, groupid=i\_group, **type**=i\_type, initexpr=i\_init, calcexpr=i\_calc,

   tlrdvg=i\_tlrdvg, loggedpar=i\_loggedpar, isexternal=i\_isext, timeout=i\_timeout, avgmethod=i\_avgmethod, description=i\_desc,

   lag=i\_lag,filter=i\_filter

**WHERE** dcdm\_vars.sysname=i\_orgname;

**end** **if**;

**RETURN** QUERY **select** dcdm\_vars.sysname,dcdm\_vartype.name **as** **type**,dcdm\_vars.groupid,dcdm\_vargroups.name **as** groupname ,dcdm\_vars.description,dcdm\_vars.loggedpar,dcdm\_vars.isexternal,dcdm\_vars.avgmethod,dcdm\_vars.sortpos

**from** dcdm\_vars

**inner** **join** dcdm\_vartype **on** dcdm\_vartype.id=dcdm\_vars.**type**

**left** **outer** **join** dcdm\_vargroups **on** dcdm\_vargroups.id=dcdm\_vars.groupid

**where** dcdm\_vars.sysname=i\_name;

EXCEPTION

**WHEN** unique\_violation **then** RAISE EXCEPTION 'Name must be unique' **USING** HINT='DCDM', DETAIL='VarNameError';

**WHEN** check\_violation **then** RAISE EXCEPTION 'Invalid character' **USING** HINT='DCDM', DETAIL='VarInvalidChar';

**END**

$$;

**GRANT** **EXECUTE** **ON** **FUNCTION**  AddChangeVarData(

dcdm\_vars.sysname%**TYPE**,dcdm\_vars.sysname%**TYPE**,dcdm\_vars.**type**%**TYPE**,dcdm\_vars.initexpr%**TYPE**,dcdm\_vars.calcexpr%**TYPE**,

dcdm\_vars.tlrdvg%**TYPE**,dcdm\_vars.loggedpar%**TYPE**,dcdm\_vars.isexternal%**TYPE**,dcdm\_vars.timeout%**TYPE**,

dcdm\_vars.avgmethod%**TYPE**,dcdm\_vars.description%**TYPE**,dcdm\_vars.lag%**TYPE**,dcdm\_vars.groupid%**TYPE**,dcdm\_vars.filter%**TYPE**)

**TO** dcdmsrv;

*--#############################################################################################*

*/\**

*CREATE OR REPLACE FUNCTION GetVarItem(i\_sysname dcdm\_vars.sysname%TYPE)*

*RETURNS TABLE(sysname dcdm\_vars.sysname%TYPE, type dcdm\_vars.type%TYPE, initexpr dcdm\_vars.initexpr%TYPE,calcexpr dcdm\_vars.calcexpr%TYPE,*

*tlrdvg dcdm\_vars.tlrdvg%TYPE, loggedpar dcdm\_vars.loggedpar%TYPE, isexternal dcdm\_vars.isexternal%TYPE, timeout dcdm\_vars.timeout%TYPE, avgmethod dcdm\_vars.avgmethod%TYPE,description dcdm\_vars.description%TYPE)*

*IMMUTABLE LANGUAGE 'plpgsql' AS $$*

*BEGIN*

*RETURN QUERY SELECT dcdm\_vars.sysname, dcdm\_vars.type, dcdm\_vars.initexpr, dcdm\_vars.calcexpr, dcdm\_vars.tlrdvg, dcdm\_vars.loggedpar, dcdm\_vars.isexternal, dcdm\_vars.timeout, dcdm\_vars.avgmethod, dcdm\_vars.description*

*FROM dcdm\_vars*

*WHERE dcdm\_vars.sysname=i\_sysname;*

*END*

*$$;*

*GRANT EXECUTE ON FUNCTION  GetVarItem(dcdm\_vars.sysname%TYPE) TO dcdmsrv;*

*\*/*

*--#############################################################################################*

**CREATE** OR **REPLACE** **FUNCTION** RemoveVarItem(i\_sysname dcdm\_vars.sysname%**TYPE**)

**RETURNS** **integer** VOLATILE **LANGUAGE** 'plpgsql' **AS** $$

**DECLARE**

ResulrCount **int**;

**BEGIN**

**DELETE** **FROM** dcdm\_vars **WHERE** dcdm\_vars.sysname=i\_sysname;

GET DIAGNOSTICS ResulrCount:=ROW\_COUNT;

**RETURN** ResulrCount;

**END**

$$;

**GRANT** **EXECUTE** **ON** **FUNCTION**  RemoveVarItem(dcdm\_vars.sysname%**TYPE**) **TO** dcdmsrv;

*--#############################################################################################*

**CREATE** OR **REPLACE** **FUNCTION** SortVar(i\_sysname dcdm\_vars.sysname%**TYPE**,i\_direction **char**(1))

**RETURNS** **TABLE**(

sysname dcdm\_vars.sysname%**TYPE**,

**type** dcdm\_vartype.name%**TYPE**,

groupid dcdm\_vars.groupid%**TYPE**,

groupname dcdm\_vargroups.name%**TYPE**,

description dcdm\_vars.description%**TYPE**,

loggedpar dcdm\_vars.loggedpar%**TYPE**,

isexternal dcdm\_vars.isexternal%**TYPE**,

avgmethod dcdm\_vars.avgmethod%**TYPE**,

sortpos dcdm\_vars.sortpos%**TYPE**,

filter dcdm\_vars.filter%**TYPE**)

VOLATILE **LANGUAGE** 'plpgsql' **AS** $$

**DECLARE**

v\_pos dcdm\_vars.sortpos%**TYPE**;

v\_newpos dcdm\_vars.sortpos%**TYPE**;

v\_psysname dcdm\_vars.sysname%**TYPE**;

**BEGIN**

**select** dcdm\_vars.sortpos **from** dcdm\_vars **where** dcdm\_vars.sysname=i\_sysname **into** v\_pos;

**if** i\_direction='D' **then**

**select** dcdm\_vars.sortpos,dcdm\_vars.sysname **into** v\_newpos,v\_psysname **from** dcdm\_vars **where** dcdm\_vars.sortpos>v\_pos **order** **by** dcdm\_vars.sortpos **limit** 1;

elsif i\_direction='U' **then**

**select** dcdm\_vars.sortpos,dcdm\_vars.sysname **into** v\_newpos,v\_psysname **from** dcdm\_vars **where** dcdm\_vars.sortpos<v\_pos **order** **by** dcdm\_vars.sortpos **desc** **limit** 1;

**else**

  v\_newpos:=null;

  v\_psysname:=null;

**end** **if**;

**if** v\_psysname IS NOT NULL AND v\_newpos IS NOT NULL **then**

**UPDATE** dcdm\_vars **SET** sortpos=v\_newpos **WHERE** dcdm\_vars.sysname=i\_sysname;

**UPDATE** dcdm\_vars **SET** sortpos=v\_pos **WHERE** dcdm\_vars.sysname=v\_psysname;

**else**

  RAISE EXCEPTION 'Unknown sort direction' **USING** HINT='DCDM', DETAIL='VarSortError';

**end** **if**;

**RETURN** QUERY **select** dcdm\_vars.sysname,dcdm\_vartype.name **as** **type**,dcdm\_vars.groupid,dcdm\_vargroups.name **as** groupname,

      dcdm\_vars.description,dcdm\_vars.loggedpar,dcdm\_vars.isexternal,dcdm\_vars.avgmethod,dcdm\_vars.sortpos,dcdm\_vars.filter

**from** dcdm\_vars

**inner** **join** dcdm\_vartype **on** dcdm\_vartype.id=dcdm\_vars.**type**

**left** **outer** **join** dcdm\_vargroups **on** dcdm\_vargroups.id=dcdm\_vars.groupid

**where** dcdm\_vars.sysname in(i\_sysname,v\_psysname);

**END**

$$;

**GRANT** **EXECUTE** **ON** **FUNCTION**  SortVar(dcdm\_vars.sysname%**TYPE**,**char**) **TO** dcdmsrv;

*--#############################################################################################*

**CREATE** OR **REPLACE** **FUNCTION** AddChangeProfileData(i\_id dcdm\_profiles.id%**TYPE**,i\_name dcdm\_profiles.name%**TYPE**,i\_desc dcdm\_profiles.description%**TYPE**)

**RETURNS** **TABLE**(id dcdm\_profiles.id%**TYPE**,name dcdm\_profiles.name%**TYPE**,description dcdm\_profiles.description%**TYPE**,isactive dcdm\_profiles.isactive%**TYPE**)

VOLATILE **LANGUAGE** 'plpgsql' **AS** $$

**BEGIN**

**if** i\_id=0 **then**

*-- Добавление*

**INSERT** **INTO** dcdm\_profiles(name, description) **VALUES** (i\_name,i\_desc);

**select** lastval() **into** i\_id;

**else**

*-- Редактирование*

**UPDATE** dcdm\_profiles **SET** name=i\_name, description=i\_desc **WHERE** dcdm\_profiles.id=i\_id;

**end** **if**;

*-- Вернём отредактированный результат*

**RETURN** QUERY **SELECT** dcdm\_profiles.id, dcdm\_profiles.name, dcdm\_profiles.description, dcdm\_profiles.isactive **FROM** dcdm\_profiles **where** dcdm\_profiles.id=i\_id;

 EXCEPTION **WHEN** unique\_violation **then** RAISE EXCEPTION 'Name must be unique' **USING** HINT='DCDM', DETAIL='PrfNameError';

**END**

$$;

**GRANT** **EXECUTE** **ON** **FUNCTION**  AddChangeProfileData(dcdm\_profiles.id%**TYPE**,dcdm\_profiles.name%**TYPE**,dcdm\_profiles.description%**TYPE**) **TO** dcdmsrv;

*--#############################################################################################*

**CREATE** OR **REPLACE** **FUNCTION** RemoveProfileItem(i\_id dcdm\_profiles.id%**TYPE**)

**RETURNS** **integer** VOLATILE **LANGUAGE** 'plpgsql' **AS** $$

**DECLARE**

ResulrCount **int**;

**BEGIN**

**DELETE** **FROM** dcdm\_profiles **WHERE** dcdm\_profiles.id=i\_id;

GET DIAGNOSTICS ResulrCount:=ROW\_COUNT;

**RETURN** ResulrCount;

**END**

$$;

**GRANT** **EXECUTE** **ON** **FUNCTION**  RemoveProfileItem(dcdm\_profiles.id%**TYPE**) **TO** dcdmsrv;

*--##############################################################################################*

**CREATE** OR **REPLACE** **FUNCTION** ProfileActivate(i\_id dcdm\_profiles.id%**TYPE**)

**RETURNS** dcdm\_profiles.id%**TYPE** VOLATILE **LANGUAGE** 'plpgsql' **AS** $$

**BEGIN**

*-- Сбросим флаг активации со всех профилей*

**UPDATE** dcdm\_profiles **SET** isactive=false **WHERE** isactive=true;

*-- Установим флаг на нужный профиль*

**UPDATE** dcdm\_profiles **SET** isactive=true       **WHERE** id=i\_id;

**RETURN** i\_id;

**END**

$$;

**GRANT** **EXECUTE** **ON** **FUNCTION**  ProfileActivate(dcdm\_profiles.id%**TYPE**) **TO** dcdmsrv;

*--#############################################################################################*

**CREATE** OR **REPLACE** **FUNCTION** AddChangeIPointData(i\_id dcdm\_ipoints.id%**TYPE**,i\_profileid dcdm\_ipoints.profileid%**TYPE**,i\_pointtype dcdm\_ipoints.pointtype%**TYPE**,

                                                     i\_name dcdm\_ipoints.name%**TYPE**,i\_config dcdm\_ipoints.config%**TYPE**)

**RETURNS** **TABLE**(id dcdm\_ipoints.id%**TYPE**,profileid dcdm\_ipoints.profileid%**TYPE**,pointtype dcdm\_ipoints.pointtype%**TYPE**,name dcdm\_ipoints.name%**TYPE**)

VOLATILE **LANGUAGE** 'plpgsql' **AS** $$

**BEGIN**

**if** i\_id=0 **then**

*-- Добавление*

**INSERT** **INTO** dcdm\_ipoints(profileid, pointtype, name, config) **VALUES** (i\_profileid,i\_pointtype,i\_name,i\_config);

**select** lastval() **into** i\_id;

**else**

*-- Редактирование*

**UPDATE** dcdm\_ipoints **SET** pointtype=i\_pointtype, name=i\_name, config=i\_config

**WHERE** dcdm\_ipoints.id=i\_id AND dcdm\_ipoints.profileid=i\_profileid;

**end** **if**;

*-- Вернём отредактированный результат*

**RETURN** QUERY **SELECT** dcdm\_ipoints.id, dcdm\_ipoints.profileid, dcdm\_ipoints.pointtype, dcdm\_ipoints.name

**FROM** dcdm\_ipoints

**WHERE** dcdm\_ipoints.id=i\_id;

 EXCEPTION **WHEN** unique\_violation **then** RAISE EXCEPTION 'Name must be unique' **USING** HINT='DCDM', DETAIL='IPointNameError';

**END**

$$;

**GRANT** **EXECUTE** **ON** **FUNCTION**  AddChangeIPointData(dcdm\_ipoints.id%**TYPE**,dcdm\_ipoints.profileid%**TYPE**,dcdm\_ipoints.pointtype%**TYPE**,dcdm\_ipoints.name%**TYPE**,dcdm\_ipoints.config%**TYPE**) **TO** dcdmsrv;

*--#############################################################################################*

**CREATE** OR **REPLACE** **FUNCTION** RemoveIPointItem(i\_id dcdm\_ipoints.id%**TYPE**)

**RETURNS** **integer** VOLATILE **LANGUAGE** 'plpgsql' **AS** $$

**DECLARE**

ResulrCount **int**;

**BEGIN**

**DELETE** **FROM** dcdm\_ipoints **WHERE** dcdm\_ipoints.id=i\_id;

GET DIAGNOSTICS ResulrCount:=ROW\_COUNT;

**RETURN** ResulrCount;

**END**

$$;

**GRANT** **EXECUTE** **ON** **FUNCTION**  RemoveIPointItem(dcdm\_ipoints.id%**TYPE**) **TO** dcdmsrv;

*--#############################################################################################*

**CREATE** OR **REPLACE** **FUNCTION** AddChangeRefData(i\_id dcdm\_referenceslist.id%**TYPE**,i\_name dcdm\_referenceslist.name%**TYPE**,i\_sysname dcdm\_referenceslist.sysname%**TYPE**,

                                                     i\_iswelldep dcdm\_referenceslist.iswelldep%**TYPE**)

**RETURNS** **TABLE**(id dcdm\_referenceslist.id%**TYPE**,name dcdm\_referenceslist.name%**TYPE**,sysname dcdm\_referenceslist.sysname%**TYPE**,iswelldep dcdm\_referenceslist.iswelldep%**TYPE**)

VOLATILE **LANGUAGE** 'plpgsql' **AS** $$

**BEGIN**

**if** i\_id=0 **then**

*-- Добавление*

**INSERT** **INTO** dcdm\_referenceslist(name, sysname, iswelldep) **VALUES** (i\_name,i\_sysname,i\_iswelldep);

**select** lastval() **into** i\_id;

**else**

*-- Редактирование*

**UPDATE** dcdm\_referenceslist **SET** name=i\_name, sysname=i\_sysname, iswelldep=i\_iswelldep  **WHERE** dcdm\_referenceslist.id=i\_id;

**end** **if**;

*-- Вернём отредактированный результат*

**RETURN** QUERY **SELECT** dcdm\_referenceslist.id, dcdm\_referenceslist.name, dcdm\_referenceslist.sysname, dcdm\_referenceslist.iswelldep

**FROM** dcdm\_referenceslist

**WHERE** dcdm\_referenceslist.id=i\_id;

 EXCEPTION **WHEN** unique\_violation **then** RAISE EXCEPTION 'SysName must be unique' **USING** HINT='DCDM', DETAIL='SysnameError';

**END**

$$;

**GRANT** **EXECUTE** **ON** **FUNCTION**  AddChangeRefData(dcdm\_referenceslist.id%**TYPE**,dcdm\_referenceslist.name%**TYPE**,dcdm\_referenceslist.sysname%**TYPE**,dcdm\_referenceslist.iswelldep%**TYPE**) **TO** dcdmsrv;

*--#############################################################################################*

**CREATE** OR **REPLACE** **FUNCTION** RemoveRefItem(i\_id dcdm\_referenceslist.id%**TYPE**)

**RETURNS** **integer** VOLATILE **LANGUAGE** 'plpgsql' **AS** $$

**DECLARE**

ResulrCount **int**;

**BEGIN**

**DELETE** **FROM** dcdm\_referenceslist **WHERE** dcdm\_referenceslist.id=i\_id;

GET DIAGNOSTICS ResulrCount:=ROW\_COUNT;

**RETURN** ResulrCount;

**END**

$$;

**GRANT** **EXECUTE** **ON** **FUNCTION**  RemoveRefItem(dcdm\_referenceslist.id%**TYPE**) **TO** dcdmsrv;

*--#############################################################################################*

**CREATE** OR **REPLACE** **FUNCTION** AddChangeRefColData(i\_id dcdm\_referencecol.id%**TYPE**,i\_reflistid dcdm\_referencecol.reflistid%**TYPE**,i\_colname dcdm\_referencecol.colname%**TYPE**,i\_sysname dcdm\_referencecol.sysname%**TYPE**,

                                                     i\_type dcdm\_referencecol.**type**%**TYPE**,i\_initexpr dcdm\_referencecol.initexpr%**TYPE**)

**RETURNS** **TABLE**(id dcdm\_referencecol.id%**TYPE**,reflistid dcdm\_referencecol.reflistid%**TYPE**,colname dcdm\_referencecol.colname%**TYPE**,sysname dcdm\_referencecol.sysname%**TYPE**,**type** dcdm\_vartype.name%**TYPE**)

VOLATILE **LANGUAGE** 'plpgsql' **AS** $$

**BEGIN**

**if** i\_id=0 **then**

*-- Добавление*

**INSERT** **INTO** dcdm\_referencecol(reflistid, colname, sysname, **type**, initexpr) **VALUES** (i\_reflistid,i\_colname,i\_sysname,i\_type,i\_initexpr);

**select** lastval() **into** i\_id;

**else**

*-- Редактирование*

**UPDATE** dcdm\_referencecol **SET** reflistid=i\_reflistid, colname=i\_colname, sysname=i\_sysname, **type**=i\_type, initexpr=i\_initexpr  **WHERE** dcdm\_referencecol.id=i\_id;

**end** **if**;

*-- Вернём отредактированный результат*

**RETURN** QUERY **SELECT** dcdm\_referencecol.id, dcdm\_referencecol.reflistid, dcdm\_referencecol.colname, dcdm\_referencecol.sysname, dcdm\_vartype.name

**FROM** dcdm\_referencecol

**inner** **join** dcdm\_vartype **on** dcdm\_vartype.id=dcdm\_referencecol.**type**

**WHERE** dcdm\_referencecol.id=i\_id;

 EXCEPTION **WHEN** unique\_violation **then** RAISE EXCEPTION 'SysName must be unique' **USING** HINT='DCDM', DETAIL='SysnameError';

**END**

$$;

**GRANT** **EXECUTE** **ON** **FUNCTION**  AddChangeRefColData(dcdm\_referencecol.id%**TYPE**,dcdm\_referencecol.reflistid%**TYPE**,dcdm\_referencecol.colname%**TYPE**,dcdm\_referencecol.sysname%**TYPE**,dcdm\_referencecol.**type**%**TYPE**,dcdm\_referencecol.initexpr%**TYPE**) **TO** dcdmsrv;

*--#############################################################################################*

**CREATE** OR **REPLACE** **FUNCTION** RemoveRefColItem(i\_id dcdm\_referencecol.id%**TYPE**)

**RETURNS** **integer** VOLATILE **LANGUAGE** 'plpgsql' **AS** $$

**DECLARE**

ResulrCount **int**;

**BEGIN**

**DELETE** **FROM** dcdm\_referencecol **WHERE** dcdm\_referencecol.id=i\_id;

GET DIAGNOSTICS ResulrCount:=ROW\_COUNT;

**RETURN** ResulrCount;

**END**

$$;

**GRANT** **EXECUTE** **ON** **FUNCTION**  RemoveRefColItem(dcdm\_referencecol.id%**TYPE**) **TO** dcdmsrv;

*--#############################################################################################*

**CREATE** OR **REPLACE** **FUNCTION** AddChangeWellData(i\_id dcdm\_well.id%**TYPE**,i\_name dcdm\_well.name%**TYPE**,i\_lat dcdm\_well.latitude%**TYPE**,i\_lon dcdm\_well.longitude%**TYPE**)

**RETURNS** **TABLE**(id dcdm\_well.id%**TYPE**,name dcdm\_well.name%**TYPE**,latitude dcdm\_well.latitude%**TYPE**,longitude dcdm\_well.longitude%**TYPE**,crdate dcdm\_well.crdate%**TYPE**,isactive dcdm\_well.isactive%**TYPE**)

VOLATILE **LANGUAGE** 'plpgsql' **AS** $$

**BEGIN**

**if** i\_id=0 **then**

*-- Добавление*

**INSERT** **INTO** dcdm\_well(name, latitude, longitude) **VALUES** (i\_name,i\_lat,i\_lon);

**select** lastval() **into** i\_id;

**else**

*-- Редактирование*

**UPDATE** dcdm\_well **SET** name=i\_name, latitude=i\_lat, longitude=i\_lon  **WHERE** dcdm\_well.id=i\_id;

**end** **if**;

*-- Вернём отредактированный результат*

**RETURN** QUERY **SELECT** dcdm\_well.id, dcdm\_well.name, dcdm\_well.latitude, dcdm\_well.longitude,dcdm\_well.crdate, dcdm\_well.isactive

**FROM** dcdm\_well

**WHERE** dcdm\_well.id=i\_id;

**END**

$$;

**GRANT** **EXECUTE** **ON** **FUNCTION**  AddChangeWellData(dcdm\_well.id%**TYPE**,dcdm\_well.name%**TYPE**,dcdm\_well.latitude%**TYPE**,dcdm\_well.longitude%**TYPE**) **TO** dcdmsrv;

*--#############################################################################################*

**CREATE** OR **REPLACE** **FUNCTION** RemoveWellItem(i\_id dcdm\_well.id%**TYPE**)

**RETURNS** **integer** VOLATILE **LANGUAGE** 'plpgsql' **AS** $$

**DECLARE**

ResulrCount **int**;

**BEGIN**

**DELETE** **FROM** dcdm\_well **WHERE** dcdm\_well.id=i\_id;

GET DIAGNOSTICS ResulrCount:=ROW\_COUNT;

**RETURN** ResulrCount;

**END**

$$;

**GRANT** **EXECUTE** **ON** **FUNCTION**  RemoveWellItem(dcdm\_well.id%**TYPE**) **TO** dcdmsrv;

*--##############################################################################################*

**CREATE** OR **REPLACE** **FUNCTION** ActivateWellItem(i\_id dcdm\_well.id%**TYPE**)

**RETURNS** **integer** VOLATILE **LANGUAGE** 'plpgsql' **AS** $$

**DECLARE**

ResulrCount **int**;

**BEGIN**

**UPDATE** dcdm\_well **SET** isactive=false;

**UPDATE** dcdm\_well **SET** isactive=true **WHERE** dcdm\_well.id=i\_id;

GET DIAGNOSTICS ResulrCount:=ROW\_COUNT;

**RETURN** ResulrCount;

**END**

$$;

**GRANT** **EXECUTE** **ON** **FUNCTION**  ActivateWellItem(dcdm\_well.id%**TYPE**) **TO** dcdmsrv;

*--##############################################################################################*

*/\**

*Сохранение информации в справочнике*

*$1 - Идентификатор записи или 0 в случае создания новой*

*$2 - Идентификатор справочника*

*$3 - Идентификатор скважины для оперативных справочников или NULL для общих*

*$4 - Префикс группы данных (мнемоническое имя) или NULL если не требуется*

*$5 - Данные справочника (json объект)*

*\*/*

**CREATE** OR **REPLACE** **FUNCTION** AddChangeRefDataItem(i\_id dcdm\_referencedata.id%**TYPE**,i\_reflistid dcdm\_referencedata.reflistid%**TYPE**,

                                                                                            i\_wellid dcdm\_referencedata.wellid%**TYPE**,i\_grpname dcdm\_referencedata.grpname%**TYPE**,

                                                                                         i\_val dcdm\_referencedata.val%**TYPE**)

**RETURNS** **TABLE**(id dcdm\_referencedata.id%**TYPE**,wellid dcdm\_referencedata.wellid%**TYPE**,reflistid dcdm\_referencedata.reflistid%**TYPE**,

                         subid dcdm\_referencedata.subid%**TYPE**,grpname dcdm\_referencedata.grpname%**TYPE**,val dcdm\_referencedata.val%**TYPE**)

VOLATILE **LANGUAGE** 'plpgsql' **AS** $$

**DECLARE**

v\_subid **int**;

**BEGIN**

*-- Сперва проверим соответствует ли тип справочника заявленным параметрам*

**IF** **EXISTS** (**SELECT** \* **FROM** dcdm\_referenceslist **where** dcdm\_referenceslist.id=i\_reflistid AND ((i\_wellid IS NOT NULL AND dcdm\_referenceslist.iswelldep=true) OR (i\_wellid IS NULL AND dcdm\_referenceslist.iswelldep=false))) **THEN**

*-- TODO: Сделать проверку корректности json объектов*

**if** i\_id=0 **then**

*-- Добавление*

*-- Найдем максимальное значение для SubId указанного справочника*

**SELECT** **coalesce**(MAX(dcdm\_referencedata.subid),0)+1 **FROM** dcdm\_referencedata **where** dcdm\_referencedata.reflistid=i\_reflistid AND ((i\_wellid IS NOT NULL AND dcdm\_referencedata.wellid=i\_wellid) OR (i\_wellid IS NULL AND dcdm\_referencedata.wellid IS NULL)) **into** v\_subid;

*-- Добавим запись*

**INSERT** **INTO** dcdm\_referencedata(wellid, reflistid, subid, grpname, val) **VALUES** (i\_wellid,i\_reflistid,v\_subid,i\_grpname,i\_val);

*-- Узнаем идентификатор добавленной записи*

**select** lastval() **into** i\_id;

**else**

*-- Обновление*

**UPDATE** dcdm\_referencedata **SET** wellid=i\_wellid, reflistid=i\_reflistid, grpname=i\_grpname, val=i\_val **WHERE** dcdm\_referencedata.id=i\_id;

**end** **if**;

**RETURN** QUERY **SELECT** dcdm\_referencedata.id, dcdm\_referencedata.wellid, dcdm\_referencedata.reflistid, dcdm\_referencedata.subid, dcdm\_referencedata.grpname, dcdm\_referencedata.val

**FROM** dcdm\_referencedata **where** dcdm\_referencedata.id=i\_id;

**ELSE**

  RAISE EXCEPTION 'Invalid reference or well id' **USING** HINT='DCDM', DETAIL='RefIdOrWellIdErr';

**END** **IF**;

**END**

$$;

**GRANT** **EXECUTE** **ON** **FUNCTION**  AddChangeRefDataItem(dcdm\_referencedata.id%**TYPE**,dcdm\_referencedata.reflistid%**TYPE**,dcdm\_referencedata.wellid%**TYPE**,dcdm\_referencedata.grpname%**TYPE**,dcdm\_referencedata.val%**TYPE**) **TO** dcdmsrv;

*--#############################################################################################*

*-- 2 параметра*

**CREATE** OR **REPLACE** **FUNCTION** GetRefDataItem(i\_reflistid dcdm\_referencedata.reflistid%**TYPE**,i\_wellid dcdm\_referencedata.wellid%**TYPE**)

**RETURNS** **TABLE**(id dcdm\_referencedata.id%**TYPE**,wellid dcdm\_referencedata.wellid%**TYPE**,reflistid dcdm\_referencedata.reflistid%**TYPE**,

                         subid dcdm\_referencedata.subid%**TYPE**,grpname dcdm\_referencedata.grpname%**TYPE**,val dcdm\_referencedata.val%**TYPE**)

IMMUTABLE **LANGUAGE** 'plpgsql' **AS** $$

**BEGIN**

**IF** **EXISTS** (**SELECT** \* **FROM** dcdm\_referenceslist **where** dcdm\_referenceslist.id=i\_reflistid AND ((i\_wellid IS NOT NULL AND dcdm\_referenceslist.iswelldep=true) OR (i\_wellid IS NULL AND dcdm\_referenceslist.iswelldep=false))) **THEN**

**RETURN** QUERY **select** dcdm\_referencedata.id, dcdm\_referencedata.wellid, dcdm\_referencedata.reflistid, dcdm\_referencedata.subid, dcdm\_referencedata.grpname, dcdm\_referencedata.val

**FROM** dcdm\_referencedata

**where** dcdm\_referencedata.reflistid=i\_reflistid AND ((i\_wellid IS NULL AND dcdm\_referencedata.wellid IS NULL) OR (i\_wellid IS NOT NULL AND dcdm\_referencedata.wellid=i\_wellid))

**ORDER** **BY** dcdm\_referencedata.subid,dcdm\_referencedata.grpname;

**ELSE**

  RAISE EXCEPTION 'Invalid reference or well id' **USING** HINT='DCDM', DETAIL='RefIdOrWellIdErr';

**END** **IF**;

**END**

$$;

**GRANT** **EXECUTE** **ON** **FUNCTION**  GetRefDataItem(i\_reflistid dcdm\_referencedata.reflistid%**TYPE**,i\_wellid dcdm\_referencedata.wellid%**TYPE**) **TO** dcdmsrv;

*--#############################################################################################*

*-- 3 параметра*

**CREATE** OR **REPLACE** **FUNCTION** GetRefDataItem(i\_id dcdm\_referencedata.id%**TYPE**,i\_reflistid dcdm\_referencedata.reflistid%**TYPE**,i\_wellid dcdm\_referencedata.wellid%**TYPE**)

**RETURNS** **TABLE**(id dcdm\_referencedata.id%**TYPE**,wellid dcdm\_referencedata.wellid%**TYPE**,reflistid dcdm\_referencedata.reflistid%**TYPE**,

                        subid dcdm\_referencedata.subid%**TYPE**,grpname dcdm\_referencedata.grpname%**TYPE**,val dcdm\_referencedata.val%**TYPE**)

IMMUTABLE **LANGUAGE** 'plpgsql' **AS** $$

**BEGIN**

**IF** **EXISTS** (**SELECT** \* **FROM** dcdm\_referenceslist **where** dcdm\_referenceslist.id=i\_reflistid AND ((i\_wellid IS NOT NULL AND dcdm\_referenceslist.iswelldep=true) OR (i\_wellid IS NULL AND dcdm\_referenceslist.iswelldep=false))) **THEN**

**RETURN** QUERY **select** dcdm\_referencedata.id, dcdm\_referencedata.wellid, dcdm\_referencedata.reflistid, dcdm\_referencedata.subid, dcdm\_referencedata.grpname, dcdm\_referencedata.val

**FROM** dcdm\_referencedata

**where** dcdm\_referencedata.id=i\_id AND dcdm\_referencedata.reflistid=i\_reflistid AND ((i\_wellid IS NULL AND dcdm\_referencedata.wellid IS NULL) OR (i\_wellid IS NOT NULL AND dcdm\_referencedata.wellid=i\_wellid));

**ELSE**

  RAISE EXCEPTION 'Invalid reference or well id' **USING** HINT='DCDM', DETAIL='RefIdOrWellIdErr';

**END** **IF**;

**END**

$$;

**GRANT** **EXECUTE** **ON** **FUNCTION**  GetRefDataItem(dcdm\_referencedata.id%**TYPE**,dcdm\_referencedata.reflistid%**TYPE**,dcdm\_referencedata.wellid%**TYPE**) **TO** dcdmsrv;

*--#############################################################################################*

**CREATE** OR **REPLACE** **FUNCTION** RemoveRefDataItem(i\_id dcdm\_referencedata.id%**TYPE**,i\_reflistid dcdm\_referencedata.reflistid%**TYPE**,i\_wellid dcdm\_referencedata.wellid%**TYPE**)

**RETURNS** **integer** VOLATILE **LANGUAGE** 'plpgsql' **AS** $$

**DECLARE**

ResulrCount **int**;

**BEGIN**

*-- Сперва проверим соответствует ли тип справочника заявленным параметрам*

**IF** **EXISTS** (**SELECT** \* **FROM** dcdm\_referenceslist **where** dcdm\_referenceslist.id=i\_reflistid AND ((i\_wellid IS NOT NULL AND dcdm\_referenceslist.iswelldep=true) OR (i\_wellid IS NULL AND dcdm\_referenceslist.iswelldep=false))) **THEN**

**DELETE** **FROM** dcdm\_referencedata

**where** dcdm\_referencedata.id=i\_id AND dcdm\_referencedata.reflistid=i\_reflistid AND ((i\_wellid IS NULL AND dcdm\_referencedata.wellid IS NULL) OR (i\_wellid IS NOT NULL AND dcdm\_referencedata.wellid=i\_wellid));

  GET DIAGNOSTICS ResulrCount:=ROW\_COUNT;

**RETURN** ResulrCount;

**ELSE** RAISE EXCEPTION 'Invalid reference or well id' **USING** HINT='DCDM', DETAIL='RefIdOrWellIdErr';

**END** **IF**;

**END**

$$;

**GRANT** **EXECUTE** **ON** **FUNCTION**  RemoveRefDataItem(i\_id dcdm\_referencedata.id%**TYPE**,i\_reflistid dcdm\_referencedata.reflistid%**TYPE**,i\_wellid dcdm\_referencedata.wellid%**TYPE**) **TO** dcdmsrv;

*--#############################################################################################*

**CREATE** OR **REPLACE** **FUNCTION** ReposRefDataItem(i\_id dcdm\_referencedata.id%**TYPE**,i\_reflistid dcdm\_referencedata.reflistid%**TYPE**,i\_wellid dcdm\_referencedata.wellid%**TYPE**,i\_newsubid dcdm\_referencedata.subid%**TYPE**)

**RETURNS** **integer** VOLATILE **LANGUAGE** 'plpgsql' **AS** $$

**DECLARE**

v\_oldsubid **int**;

ResulrCount **int**;

curs **cursor** **for** **select** dcdm\_referencedata.id*--,dcdm\_referencedata.subid*

**from** dcdm\_referencedata

**where** dcdm\_referencedata.reflistid=i\_reflistid

                               AND ((i\_wellid IS NULL AND dcdm\_referencedata.wellid IS NULL) OR (i\_wellid IS NOT NULL AND dcdm\_referencedata.wellid=i\_wellid))

                            AND dcdm\_referencedata.subid>=i\_newsubid

**ORDER** **BY** dcdm\_referencedata.subid **desc**;

v\_id **int**;

**BEGIN**

*-- Сперва проверим соответствует ли тип справочкика заявленым параметрам*

**IF** **EXISTS** (**SELECT** \* **FROM** dcdm\_referenceslist **where** dcdm\_referenceslist.id=i\_reflistid AND ((i\_wellid IS NOT NULL AND dcdm\_referenceslist.iswelldep=true) OR (i\_wellid IS NULL AND dcdm\_referenceslist.iswelldep=false))) **THEN**

*-- Узнаем текущую позицию элемента*

**select** dcdm\_referencedata.subid **into** v\_oldsubid **from** dcdm\_referencedata

**where** dcdm\_referencedata.id=i\_id AND dcdm\_referencedata.reflistid=i\_reflistid AND ((i\_wellid IS NULL AND dcdm\_referencedata.wellid IS NULL) OR (i\_wellid IS NOT NULL AND dcdm\_referencedata.wellid=i\_wellid));

**IF** v\_oldsubid<>i\_newsubid **THEN**

*-- проверим свободен ли указанный идентификатор*

**IF** **EXISTS**(**select** \* **FROM** dcdm\_referencedata **where** dcdm\_referencedata.subid=i\_newsubid AND dcdm\_referencedata.reflistid=i\_reflistid AND ((i\_wellid IS NULL AND dcdm\_referencedata.wellid IS NULL) OR (i\_wellid IS NOT NULL AND dcdm\_referencedata.wellid=i\_wellid))) **THEN**

*-- Место занято и надо сдвигать subid*

**OPEN** curs;

**loop**

**FETCH** **FROM** curs **INTO** v\_id;

**EXIT** **WHEN** NOT FOUND;

**UPDATE** dcdm\_referencedata **SET** subid=dcdm\_referencedata.subid+1 **WHERE** dcdm\_referencedata.id=v\_id;

**end** **loop**;

**close** curs;

**UPDATE** dcdm\_referencedata **SET** subid=i\_newsubid **WHERE** dcdm\_referencedata.id=i\_id AND dcdm\_referencedata.reflistid=i\_reflistid AND ((i\_wellid IS NULL AND dcdm\_referencedata.wellid IS NULL) OR (i\_wellid IS NOT NULL AND dcdm\_referencedata.wellid=i\_wellid));

    GET DIAGNOSTICS ResulrCount:=ROW\_COUNT;

**ELSE**

*-- место свободно*

**UPDATE** dcdm\_referencedata **SET** subid=i\_newsubid **WHERE** dcdm\_referencedata.id=i\_id AND dcdm\_referencedata.reflistid=i\_reflistid AND ((i\_wellid IS NULL AND dcdm\_referencedata.wellid IS NULL) OR (i\_wellid IS NOT NULL AND dcdm\_referencedata.wellid=i\_wellid));

    GET DIAGNOSTICS ResulrCount:=ROW\_COUNT;

**END** **IF**;

**ELSE**

   ResulrCount:=0;

**END** **IF**;

**RETURN** ResulrCount;

**ELSE**

  RAISE EXCEPTION 'Invalid reference or well id' **USING** HINT='DCDM', DETAIL='RefIdOrWellIdErr';

**END** **IF**;

**END**;

$$;

**GRANT** **EXECUTE** **ON** **FUNCTION**  ReposRefDataItem(dcdm\_referencedata.id%**TYPE**,dcdm\_referencedata.reflistid%**TYPE**,dcdm\_referencedata.wellid%**TYPE**,dcdm\_referencedata.subid%**TYPE**) **TO** dcdmsrv;

*--#############################################################################################*

*/\**

*Очистка справочника*

*$1 - Идентификатор справочника*

*$2 - Идентификатор скважины для оперативных справочников или NULL для общих*

*\*/*

**CREATE** OR **REPLACE** **FUNCTION** ClearRefData(i\_reflistid dcdm\_referencedata.reflistid%**TYPE**,i\_wellid dcdm\_referencedata.wellid%**TYPE**)

**RETURNS** void

VOLATILE **LANGUAGE** 'plpgsql' **AS** $$

*#variable\_conflict use\_variable*

**BEGIN**

*-- Сперва проверим соответствует ли тип справочкика заявленым параметрам*

**IF** **EXISTS** (**SELECT** \* **FROM** dcdm\_referenceslist **where** dcdm\_referenceslist.id=i\_reflistid AND ((i\_wellid IS NOT NULL AND dcdm\_referenceslist.iswelldep=true) OR (i\_wellid IS NULL AND dcdm\_referenceslist.iswelldep=false))) **THEN**

**DELETE** **FROM** dcdm\_referencedata **where** dcdm\_referencedata.reflistid=i\_reflistid AND ((i\_wellid IS NULL AND dcdm\_referencedata.wellid IS NULL) OR (i\_wellid IS NOT NULL AND dcdm\_referencedata.wellid=i\_wellid));

**END** **IF**;

**END**;

$$;

**GRANT** **EXECUTE** **ON** **FUNCTION**  ClearRefData(dcdm\_referencedata.reflistid%**TYPE**,dcdm\_referencedata.wellid%**TYPE**) **TO** dcdmsrv;

*--#############################################################################################*

*/\**

*Сохранение параметра конфигурации*

*$1 - Ключ*

*$2 - Значение*

*\*/*

**CREATE** OR **REPLACE** **FUNCTION** SaveConfigParam(i\_key dcdm\_config.**key**%**TYPE**,i\_data dcdm\_config.**data**%**TYPE**)

**RETURNS** **integer** VOLATILE **LANGUAGE** 'plpgsql' **AS** $$

**DECLARE**

ResultCount **int**;

**BEGIN**

**INSERT** **INTO** dcdm\_config (**key**, **data**) **VALUES** (i\_key,i\_data) **ON** CONFLICT(**key**) **DO** **UPDATE** **SET** **data**=i\_data;

GET DIAGNOSTICS ResultCount:=ROW\_COUNT;

**RETURN** ResultCount;

**END**

$$;

**GRANT** **EXECUTE** **ON** **FUNCTION**  SaveConfigParam(dcdm\_config.**key**%**TYPE**,dcdm\_config.**data**%**TYPE**) **TO** dcdmsrv;

*--#############################################################################################*

*/\**

*Сохранение конфигурации интерфейса пользователя*

*$1 - идентификатор пользователя*

*$2 - json объект интерфейса*

*\*/*

**CREATE** OR **REPLACE** **FUNCTION** SaveuserInterfaceCfg(i\_id dcdm\_users.id%**TYPE**,i\_cfg dcdm\_users.config%**TYPE**)

**RETURNS** **integer** VOLATILE **LANGUAGE** 'plpgsql' **AS** $$

**DECLARE**

ResultCount **int**;

**BEGIN**

**UPDATE** dcdm\_users **SET** config=i\_cfg **WHERE** id=i\_id;

 GET DIAGNOSTICS ResultCount:=ROW\_COUNT;

**RETURN** ResultCount;

**END**

$$;

**GRANT** **EXECUTE** **ON** **FUNCTION**  SaveuserInterfaceCfg(dcdm\_users.id%**TYPE**,dcdm\_users.config%**TYPE**) **TO** dcdmsrv;

*--#############################################################################################*

*/\**

*Сохранение настроек пользователя*

*$1 - идентификатор пользователя*

*$2 - json объект настроек*

*\*/*

**CREATE** OR **REPLACE** **FUNCTION** SaveuserProps(i\_id dcdm\_users.id%**TYPE**,i\_prop dcdm\_users.props%**TYPE**)

**RETURNS** **integer** VOLATILE **LANGUAGE** 'plpgsql' **AS** $$

**DECLARE**

ResultCount **int**;

**BEGIN**

**UPDATE** dcdm\_users **SET** props=i\_prop **WHERE** id=i\_id;

 GET DIAGNOSTICS ResultCount:=ROW\_COUNT;

**RETURN** ResultCount;

**END**

$$;

**GRANT** **EXECUTE** **ON** **FUNCTION**  SaveuserProps(dcdm\_users.id%**TYPE**,dcdm\_users.props%**TYPE**) **TO** dcdmsrv;

*--#############################################################################################*

**CREATE** OR **REPLACE** **FUNCTION** AddChangeAcGroupData(i\_id dcdm\_accessgroups.id%**TYPE**,i\_name dcdm\_accessgroups.name%**TYPE**)

**RETURNS** **TABLE**(id dcdm\_accessgroups.id%**TYPE**,name dcdm\_accessgroups.name%**TYPE**)

VOLATILE **LANGUAGE** 'plpgsql' **AS** $$

**BEGIN**

**if** i\_id=0 **then**

*-- Добавление*

**INSERT** **INTO** dcdm\_accessgroups(name) **VALUES** (i\_name);

**select** lastval() **into** i\_id;

**else**

*-- Редактирование*

**UPDATE** dcdm\_accessgroups **SET** name=i\_name **WHERE** dcdm\_accessgroups.id=i\_id;

**end** **if**;

*-- Вернём отредактированный результат*

**RETURN** QUERY **SELECT** dcdm\_accessgroups.id, dcdm\_accessgroups.name

**FROM** dcdm\_accessgroups

**WHERE** dcdm\_accessgroups.id=i\_id;

**END**

$$;

**GRANT** **EXECUTE** **ON** **FUNCTION**  AddChangeAcGroupData(dcdm\_accessgroups.id%**TYPE**,dcdm\_accessgroups.name%**TYPE**) **TO** dcdmsrv;

*--#############################################################################################*

**CREATE** OR **REPLACE** **FUNCTION** RemoveAcGroupItem(i\_id dcdm\_accessgroups.id%**TYPE**)

**RETURNS** **integer** VOLATILE **LANGUAGE** 'plpgsql' **AS** $$

**DECLARE**

ResulrCount **int**;

**BEGIN**

**DELETE** **FROM** dcdm\_accessgroups **WHERE** dcdm\_accessgroups.id=i\_id;

GET DIAGNOSTICS ResulrCount:=ROW\_COUNT;

**RETURN** ResulrCount;

**END**

$$;

**GRANT** **EXECUTE** **ON** **FUNCTION**  RemoveAcGroupItem(dcdm\_accessgroups.id%**TYPE**) **TO** dcdmsrv;

*--#############################################################################################*

**CREATE** OR **REPLACE** **FUNCTION** SaveAcGrpFldList(i\_grpid dcdm\_acgroupsfld.acgroupid%**TYPE**,i\_fields **varchar**)

**RETURNS** **integer** VOLATILE **LANGUAGE** 'plpgsql' **AS** $$

**DECLARE**

ResulrCount **int**;

**BEGIN**

*-- сперва удалим старые поля*

**DELETE** **FROM** dcdm\_acgroupsfld **WHERE** dcdm\_acgroupsfld.acgroupid=i\_grpid;

*-- добавим новые*

**INSERT** **INTO** dcdm\_acgroupsfld(acgroupid,name) **SELECT** i\_grpid,json\_array\_elements\_text(i\_fields::json);

GET DIAGNOSTICS ResulrCount:=ROW\_COUNT;

**RETURN** ResulrCount;

**END**

$$;

**GRANT** **EXECUTE** **ON** **FUNCTION**  SaveAcGrpFldList(dcdm\_acgroupsfld.acgroupid%**TYPE**,**varchar**) **TO** dcdmsrv;

*--#############################################################################################*

**CREATE** OR **REPLACE** **FUNCTION** AddChangeCommentAndFile(

i\_id      dcdm\_commentsandfiles.id%**TYPE**,

i\_wellid  dcdm\_commentsandfiles.wellid%**TYPE**,

i\_userid  dcdm\_commentsandfiles.userid%**TYPE**,

i\_type    dcdm\_commentsandfiles.**type**%**TYPE**,

i\_sdepth  dcdm\_commentsandfiles.snapdepth%**TYPE**,

i\_svdepth dcdm\_commentsandfiles.snapvdepth%**TYPE**,

i\_stime   dcdm\_commentsandfiles.snaptime%**TYPE**,

i\_comment dcdm\_commentsandfiles.**comment**%**TYPE**,

i\_fname   dcdm\_commentsandfiles.orgfilename%**TYPE**,

i\_fdata   dcdm\_commentsandfiles.filedata%**TYPE**

)

**RETURNS** **TABLE**(

id dcdm\_commentsandfiles.id%**TYPE**,

wid dcdm\_commentsandfiles.wellid%**TYPE**,

uid     dcdm\_commentsandfiles.userid%**TYPE**,

un  dcdm\_users.name%**TYPE**,

crd float8,

chd float8,

t   dcdm\_commentsandfiles.**type**%**TYPE**,

tn  dcdm\_commentsandfilestype.name%**TYPE**,

isi dcdm\_commentsandfiles.isindexed%**TYPE**,

sd  dcdm\_commentsandfiles.snapdepth%**TYPE**,

svd dcdm\_commentsandfiles.snapvdepth%**TYPE**,

st  float8,

msg dcdm\_commentsandfiles.**comment**%**TYPE**,

fn  dcdm\_commentsandfiles.orgfilename%**TYPE**,

fs  **int**

)

VOLATILE **LANGUAGE** 'plpgsql' **AS** $$

**BEGIN**

**if** i\_id=0 **then**

*-- Добавление*

**INSERT** **INTO** dcdm\_commentsandfiles(wellid,userid,**type**,snapdepth,snapvdepth,snaptime,**comment**,orgfilename,filedata)

**VALUES** (i\_wellid,i\_userid,i\_type,i\_sdepth,i\_svdepth,i\_stime,i\_comment,i\_fname,i\_fdata);

**select** lastval() **into** i\_id;

**else**

*-- Редактирование*

**UPDATE** dcdm\_commentsandfiles **SET** changetime=**CURRENT\_TIMESTAMP**, **type**=i\_type, isindexed=false,

                        snapdepth=i\_sdepth,snapvdepth=i\_svdepth,snaptime=i\_stime,

**comment**=i\_comment*--,orgfilename=i\_fname,filedata=i\_fdata*

**WHERE** dcdm\_commentsandfiles.id=i\_id AND dcdm\_commentsandfiles.wellid=i\_wellid AND dcdm\_commentsandfiles.userid=i\_userid;

**end** **if**;

*-- Вернём отредактированный результат*

**RETURN** QUERY **select** dcdm\_commentsandfiles.id,dcdm\_commentsandfiles.wellid **as** wid,dcdm\_commentsandfiles.userid **as** uid,dcdm\_users.name **as** un,

                     date\_part('epoch',dcdm\_commentsandfiles.createtime) **as** crd,date\_part('epoch',dcdm\_commentsandfiles.changetime) **as** chd,dcdm\_commentsandfiles.**type** **as** t,dcdm\_commentsandfilestype.name **as** tn,

                                      dcdm\_commentsandfiles.isindexed **as** isi,dcdm\_commentsandfiles.snapdepth **as** sd,dcdm\_commentsandfiles.snapvdepth **as** svd,date\_part('epoch',dcdm\_commentsandfiles.snaptime) **as** st,

                                      dcdm\_commentsandfiles.**comment** **as** msg,dcdm\_commentsandfiles.orgfilename **as** fn,octet\_length(dcdm\_commentsandfiles.filedata)

**from** dcdm\_commentsandfiles

**inner** **join** dcdm\_users **on** dcdm\_users.id=dcdm\_commentsandfiles.userid

**inner** **join** dcdm\_commentsandfilestype **on** dcdm\_commentsandfilestype.id=dcdm\_commentsandfiles.**type**

**where** dcdm\_commentsandfiles.id=i\_id;

**END**

$$;

**GRANT** **EXECUTE** **ON** **FUNCTION**  AddChangeCommentAndFile(dcdm\_commentsandfiles.id%**TYPE**,dcdm\_commentsandfiles.wellid%**TYPE**,dcdm\_commentsandfiles.userid%**TYPE**,

                                                                             dcdm\_commentsandfiles.**type**%**TYPE**,dcdm\_commentsandfiles.snapdepth%**TYPE**,

                                                                                                dcdm\_commentsandfiles.snapvdepth%**TYPE**,dcdm\_commentsandfiles.snaptime%**TYPE**,dcdm\_commentsandfiles.**comment**%**TYPE**,

                                                                                                dcdm\_commentsandfiles.orgfilename%**TYPE**,dcdm\_commentsandfiles.filedata%**TYPE**) **TO** dcdmsrv;

*--#############################################################################################*

**CREATE** OR **REPLACE** **FUNCTION** GetFilteredCommentList(

i\_start **bigint**,

i\_wellid dcdm\_commentsandfiles.wellid%**TYPE**,

i\_orderby **varchar**(5),

i\_isc **bool**,

i\_isf **bool**,

i\_tytes **int**[],

i\_rt **varchar**(3),

i\_dfrom float8,

i\_dto float8,

i\_tfrom **timestamp**,

i\_tto **timestamp**

)

**RETURNS** **TABLE**(

id dcdm\_commentsandfiles.id%**TYPE**,

wid dcdm\_commentsandfiles.wellid%**TYPE**,

uid     dcdm\_commentsandfiles.userid%**TYPE**,

un  dcdm\_users.name%**TYPE**,

crd float8,

chd float8,

t   dcdm\_commentsandfiles.**type**%**TYPE**,

tn  dcdm\_commentsandfilestype.name%**TYPE**,

isi dcdm\_commentsandfiles.isindexed%**TYPE**,

sd  dcdm\_commentsandfiles.snapdepth%**TYPE**,

svd dcdm\_commentsandfiles.snapvdepth%**TYPE**,

st  float8,

msg dcdm\_commentsandfiles.**comment**%**TYPE**,

fn  dcdm\_commentsandfiles.orgfilename%**TYPE**,

fs  **int**

)

IMMUTABLE **LANGUAGE** 'plpgsql' **AS** $$

**DECLARE**

*-- v\_ordercolnum int;*

v\_recperquery **int**:=100;

**BEGIN**

*-- Первый запрос вернет число записей*

**IF** i\_start is NOT NULL **then**

**RETURN** QUERY **select** i\_start,count(\*)::**int**,v\_recperquery,null::**varchar**,null::float8,null::float8,

         null::**smallint**,null::**varchar**,null::**bool**,null::float8,null::float8,null::float8,null::**varchar**,

              null::**varchar**,null::**int**

**from** dcdm\_commentsandfiles

**where** dcdm\_commentsandfiles.wellid=i\_wellid

         AND dcdm\_commentsandfiles.**type**=**ANY**(i\_tytes)

                AND ((i\_isc=true AND dcdm\_commentsandfiles.**comment** IS NOT NULL) OR i\_isc=false)

               AND ((i\_isf=true AND dcdm\_commentsandfiles.filedata IS NOT NULL) OR i\_isf=false)

               AND ((i\_rt='A')

                   OR (i\_rt='D' AND dcdm\_commentsandfiles.snapdepth BETWEEN i\_dfrom AND i\_dto)

                    OR (i\_rt='V' AND dcdm\_commentsandfiles.snapvdepth BETWEEN i\_dfrom AND i\_dto)

                    OR (i\_rt='T' AND dcdm\_commentsandfiles.snaptime BETWEEN i\_tfrom AND i\_tto)

                   );

**end** **if**;

*-- Второй непосредственно результат*

**RETURN** QUERY **select** dcdm\_commentsandfiles.id,dcdm\_commentsandfiles.wellid **as** wid,dcdm\_commentsandfiles.userid **as** uid,dcdm\_users.name **as** un,

                     date\_part('epoch',dcdm\_commentsandfiles.createtime) **as** crd,

                                      date\_part('epoch',dcdm\_commentsandfiles.changetime) **as** chd,

                                      dcdm\_commentsandfiles.**type** **as** t,dcdm\_commentsandfilestype.name **as** tn,

                                      dcdm\_commentsandfiles.isindexed **as** isi,

                                      dcdm\_commentsandfiles.snapdepth **as** sd,

                                      dcdm\_commentsandfiles.snapvdepth **as** svd,

                                      date\_part('epoch',dcdm\_commentsandfiles.snaptime) **as** st,

                                      dcdm\_commentsandfiles.**comment** **as** msg,dcdm\_commentsandfiles.orgfilename **as** fn,octet\_length(dcdm\_commentsandfiles.filedata)

**from** dcdm\_commentsandfiles

**inner** **join** dcdm\_users **on** dcdm\_users.id=dcdm\_commentsandfiles.userid

**inner** **join** dcdm\_commentsandfilestype **on** dcdm\_commentsandfilestype.id=dcdm\_commentsandfiles.**type**

**where** dcdm\_commentsandfiles.wellid=i\_wellid

        AND dcdm\_commentsandfiles.**type**=**ANY**(i\_tytes)

               AND ((i\_isc=true AND dcdm\_commentsandfiles.**comment** IS NOT NULL) OR i\_isc=false)

               AND ((i\_isf=true AND dcdm\_commentsandfiles.filedata IS NOT NULL) OR i\_isf=false)

               AND ((i\_rt='A')

                    OR (i\_rt='D' AND dcdm\_commentsandfiles.snapdepth BETWEEN i\_dfrom AND i\_dto)

                    OR (i\_rt='V' AND dcdm\_commentsandfiles.snapvdepth BETWEEN i\_dfrom AND i\_dto)

                    OR (i\_rt='T' AND dcdm\_commentsandfiles.snaptime BETWEEN i\_tfrom AND i\_tto)

                   )

**order** **by**

**case** **when** i\_orderby='CrT' **then** dcdm\_commentsandfiles.createtime **end**,

**case** **when** i\_orderby='ChT' **then** dcdm\_commentsandfiles.changetime **end**,

**case** **when** i\_orderby='Dp' **then** dcdm\_commentsandfiles.snapdepth **end**,

**case** **when** i\_orderby='VDp' **then** dcdm\_commentsandfiles.snapvdepth **end**,

**case** **when** i\_orderby='EvT' **then**  dcdm\_commentsandfiles.snaptime **end**

**limit** **case** **when** i\_start is not null **then** v\_recperquery **else** null **end** **offset** **case** **when** i\_start is not null **then** i\_start **else** 0 **end**;

**END**

$$;

**GRANT** **EXECUTE** **ON** **FUNCTION**  GetFilteredCommentList(**bigint**,dcdm\_commentsandfiles.wellid%**TYPE**,

**varchar**(5),**bool**,**bool**,**int**[],**varchar**(3),float8,float8,**timestamp**,**timestamp**) **TO** dcdmsrv;

*--#############################################################################################*

**CREATE** OR **REPLACE** **FUNCTION** RemoveCommentItem(i\_WellId dcdm\_commentsandfiles.wellid%**TYPE**,i\_CommentId dcdm\_commentsandfiles.id%**TYPE**)

**RETURNS** **integer** VOLATILE **LANGUAGE** 'plpgsql' **AS** $$

**DECLARE**

ResulrCount **int**;

**BEGIN**

**DELETE** **FROM** dcdm\_commentsandfiles **WHERE** dcdm\_commentsandfiles.id=i\_CommentId AND dcdm\_commentsandfiles.wellid=i\_WellId;

GET DIAGNOSTICS ResulrCount:=ROW\_COUNT;

**RETURN** ResulrCount;

**END**

$$;

**GRANT** **EXECUTE** **ON** **FUNCTION**  RemoveCommentItem(dcdm\_commentsandfiles.wellid%**TYPE**,dcdm\_commentsandfiles.id%**TYPE**) **TO** dcdmsrv;

*--##############################################################################################*

**CREATE** OR **REPLACE** **FUNCTION** BuildGetRawDataQuery(

i\_start **bigint**,

i\_limit **int**,

i\_wellid dcdm\_welllog.wellid%**TYPE**,

i\_dfrom dcdm\_welllog.depth%**TYPE**,

i\_dto dcdm\_welllog.depth%**TYPE**,

i\_vdfrom dcdm\_welllog.vdepth%**TYPE**,

i\_vdto dcdm\_welllog.vdepth%**TYPE**,

i\_tfrom dcdm\_welllog.**time**%**TYPE**,

i\_tto dcdm\_welllog.**time**%**TYPE**,

i\_params **text**

)

**RETURNS** **varchar**

IMMUTABLE **LANGUAGE** 'plpgsql' **AS** $body$

**DECLARE**

        v\_firstquery **varchar**:='SELECT '||

**case** **when** i\_vdfrom IS NOT NULL AND i\_vdto IS NOT NULL **then** 'vdepth' **when** i\_tfrom IS NOT NULL AND i\_tto IS NOT NULL **then** 'time' **else** 'depth'    **end**

               ||', paramname, val as val FROM dcdm\_welllog where wellid='||i\_wellid||' and (isnew is null or isnew=false) AND paramname=any(string\_to\_array('||quote\_literal(i\_params)||','',''))'||

**case** **when** i\_dfrom IS NOT NULL AND i\_dto IS NOT NULL **then** ' and depth between '||i\_dfrom||' and '||i\_dto

**when** i\_vdfrom IS NOT NULL AND i\_vdto IS NOT NULL **then** ' and vdepth between '||i\_vdfrom||' and '||i\_vdto

**when** i\_tfrom IS NOT NULL AND i\_tto IS NOT NULL **then** ' and time between '||quote\_literal(i\_tfrom)||' and '||quote\_literal(i\_tto) **else** ''

**end**||' order by 1';

        v\_secondquery **varchar**:='SELECT unnest(string\_to\_array('||quote\_literal(i\_params)||','',''))';

        v\_t **varchar**;

        v\_ctparam **varchar**:='';

        v\_basequery **varchar**;

**BEGIN**

FOREACH  v\_t in array string\_to\_array(i\_params,',') **LOOP** v\_ctparam:=v\_ctparam||', "'||v\_t||'" json'; **END** **LOOP**;

v\_basequery:='select \* from crosstab('||quote\_literal(v\_firstquery)||','||quote\_literal(v\_secondquery)||') as ct('||

**case** **when** i\_vdfrom IS NOT NULL AND i\_vdto IS NOT NULL **then** 'vdepth float8' **when** i\_tfrom IS NOT NULL AND i\_tto IS NOT NULL **then** 'time timestamp' **else** 'depth float8'  **end**||

        v\_ctparam||')'|| **case** **when** i\_limit is not null **then** ' limit '||i\_limit||' offset '||i\_start||';' **else** ';' **end**;

**return** v\_basequery;

**END**

$body$;

**GRANT** **EXECUTE** **ON** **FUNCTION**  BuildGetRawDataQuery(**bigint**,**int**,dcdm\_commentsandfiles.wellid%**TYPE**,

dcdm\_welllog.depth%**TYPE**,dcdm\_welllog.depth%**TYPE**,

dcdm\_welllog.vdepth%**TYPE**,dcdm\_welllog.vdepth%**TYPE**,

dcdm\_welllog.**time**%**TYPE**,dcdm\_welllog.**time**%**TYPE**,**text**) **TO** dcdmsrv;

*--#############################################################################################*

**CREATE** OR **REPLACE** **FUNCTION** GetCommentsListByIds(

i\_start **bigint**,

i\_idslist **varchar**

)

**RETURNS** **TABLE**(

id dcdm\_commentsandfiles.id%**TYPE**,

wid dcdm\_commentsandfiles.wellid%**TYPE**,

uid     dcdm\_commentsandfiles.userid%**TYPE**,

un  dcdm\_users.name%**TYPE**,

crd float8,

chd float8,

t   dcdm\_commentsandfiles.**type**%**TYPE**,

tn  dcdm\_commentsandfilestype.name%**TYPE**,

isi dcdm\_commentsandfiles.isindexed%**TYPE**,

sd  dcdm\_commentsandfiles.snapdepth%**TYPE**,

svd dcdm\_commentsandfiles.snapvdepth%**TYPE**,

st  float8,

msg dcdm\_commentsandfiles.**comment**%**TYPE**,

fn  dcdm\_commentsandfiles.orgfilename%**TYPE**,

fs  **int**,

pr      **int**

)

IMMUTABLE **LANGUAGE** 'plpgsql' **AS** $$

**DECLARE**

 v\_recperquery **int**:=100;

**BEGIN**

*-- Первый запрос вернет число записей*

**RETURN** QUERY **select** i\_start,count(\*)::**int**,v\_recperquery,null::**varchar**,null::float8,null::float8,

        null::**smallint**,null::**varchar**,null::**bool**,null::float8,null::float8,null::float8,null::**varchar**,

               null::**varchar**,null::**int**,null::**int**

**FROM** unnest(i\_idslist::**varchar**[]) list\_id **LEFT** **JOIN** dcdm\_commentsandfiles t **on** t.id=substring(list\_id **from** '^(\d+)')::**bigint**;

*-- Второй непосредственно результат*

**RETURN** QUERY **select** dcdm\_commentsandfiles.id,dcdm\_commentsandfiles.wellid **as** wid,dcdm\_commentsandfiles.userid **as** uid,dcdm\_users.name **as** un,

                     date\_part('epoch',dcdm\_commentsandfiles.createtime) **as** crd,date\_part('epoch',dcdm\_commentsandfiles.changetime) **as** chd,dcdm\_commentsandfiles.**type** **as** t,dcdm\_commentsandfilestype.name **as** tn,

                                      dcdm\_commentsandfiles.isindexed **as** isi,dcdm\_commentsandfiles.snapdepth **as** sd,dcdm\_commentsandfiles.snapvdepth **as** svd,date\_part('epoch',dcdm\_commentsandfiles.snaptime) **as** st,

                                      dcdm\_commentsandfiles.**comment** **as** msg,dcdm\_commentsandfiles.orgfilename **as** fn,octet\_length(dcdm\_commentsandfiles.filedata),

                                substring(list\_id **from** ':(\d+)')::**int** **as** pr

**FROM** unnest(i\_idslist::**varchar**[]) list\_id **LEFT** **JOIN** dcdm\_commentsandfiles **on** dcdm\_commentsandfiles.id=substring(list\_id **from** '^(\d+)')::**bigint**

**inner** **join** dcdm\_users **on** dcdm\_users.id=dcdm\_commentsandfiles.userid

**inner** **join** dcdm\_commentsandfilestype **on** dcdm\_commentsandfilestype.id=dcdm\_commentsandfiles.**type**

**order** **by** pr **desc**,dcdm\_commentsandfiles.id **desc**

**limit** v\_recperquery **offset** i\_start;

**END**

$$;

**GRANT** **EXECUTE** **ON** **FUNCTION**  GetCommentsListByIds(**bigint**,**varchar**) **TO** dcdmsrv;

*--##############################################################################################*

**CREATE** OR **REPLACE** **FUNCTION** GetPreliminaryWellLogData(

i\_wellid dcdm\_welllog.wellid%**TYPE**,

i\_depth dcdm\_welllog.depth%**TYPE**,

i\_vdepth dcdm\_welllog.vdepth%**TYPE**,

i\_time dcdm\_welllog.**time**%**TYPE**,

i\_params **text**

)

**RETURNS** **TABLE**(paramname dcdm\_welllog.paramname%**TYPE**,val dcdm\_welllog.val%**TYPE**,**type** **smallint**)

IMMUTABLE **LANGUAGE** 'plpgsql' **AS** $$

**BEGIN**

**IF** i\_depth IS NOT NULL **then**

**RETURN** QUERY **select** **distinct** **on** (dcdm\_welllog.paramname) dcdm\_welllog.paramname, dcdm\_welllog.val,1::**smallint** **as** **type**

**FROM** dcdm\_welllog

**where** dcdm\_welllog.wellid=i\_wellid AND (dcdm\_welllog.isnew is null or dcdm\_welllog.isnew=false) AND dcdm\_welllog.paramname=**any**(string\_to\_array(i\_params,',')) and dcdm\_welllog.depth <i\_depth

**order** **by** dcdm\_welllog.paramname, dcdm\_welllog.depth **desc**;

**RETURN** QUERY **select** dcdm\_welllog.paramname,('{"val":'||max(dcdm\_welllog.depth)||'}')::jsonb,2::**smallint**

**FROM** dcdm\_welllog

**where** dcdm\_welllog.wellid=i\_wellid AND (dcdm\_welllog.isnew is null or dcdm\_welllog.isnew=false) AND dcdm\_welllog.paramname=**any**(string\_to\_array(i\_params,','))

**group** **by** dcdm\_welllog.paramname;

**ELSEIF** i\_vdepth IS NOT NULL **then**

**RETURN** QUERY **select** **distinct** **on** (dcdm\_welllog.paramname) dcdm\_welllog.paramname, dcdm\_welllog.val,1::**smallint** **as** **type**

**FROM** dcdm\_welllog

**where** dcdm\_welllog.wellid=i\_wellid AND (dcdm\_welllog.isnew is null or dcdm\_welllog.isnew=false) AND dcdm\_welllog.paramname=**any**(string\_to\_array(i\_params,',')) and dcdm\_welllog.vdepth <i\_vdepth

**order** **by** dcdm\_welllog.paramname, dcdm\_welllog.vdepth **desc**;

**RETURN** QUERY **select** dcdm\_welllog.paramname,('{"val":'||max(dcdm\_welllog.vdepth)||'}')::jsonb,2::**smallint**

**FROM** dcdm\_welllog

**where** dcdm\_welllog.wellid=i\_wellid AND (dcdm\_welllog.isnew is null or dcdm\_welllog.isnew=false) AND dcdm\_welllog.paramname=**any**(string\_to\_array(i\_params,','))

**group** **by** dcdm\_welllog.paramname;

**ELSEIF** i\_time IS NOT NULL **then**

**RETURN** QUERY **select** **distinct** **on** (dcdm\_welllog.paramname) dcdm\_welllog.paramname, dcdm\_welllog.val,1::**smallint** **as** **type**

**FROM** dcdm\_welllog

**where** dcdm\_welllog.wellid=i\_wellid AND (dcdm\_welllog.isnew is null or dcdm\_welllog.isnew=false) AND dcdm\_welllog.paramname=**any**(string\_to\_array(i\_params,',')) and dcdm\_welllog.**time** <i\_time

**order** **by** dcdm\_welllog.paramname, dcdm\_welllog.**time** **desc**;

**RETURN** QUERY **select** dcdm\_welllog.paramname,('{"val":'||trunc(date\_part('epoch',max(dcdm\_welllog.**time**))\*1000)||'}')::jsonb,2::**smallint**

**FROM** dcdm\_welllog

**where** dcdm\_welllog.wellid=i\_wellid AND (dcdm\_welllog.isnew is null or dcdm\_welllog.isnew=false) AND dcdm\_welllog.paramname=**any**(string\_to\_array(i\_params,','))

**group** **by** dcdm\_welllog.paramname;

**END** **IF**;

**END**

$$;

**GRANT** **EXECUTE** **ON** **FUNCTION**  GetPreliminaryWellLogData(dcdm\_welllog.wellid%**TYPE**,dcdm\_welllog.depth%**TYPE**,dcdm\_welllog.vdepth%**TYPE**,dcdm\_welllog.**time**%**TYPE**,**text**) **TO** dcdmsrv;

*--##############################################################################################*

**CREATE** OR **REPLACE** **FUNCTION** commitunfixeddata(i\_userid dcdm\_welllog.userid%**TYPE**)

**RETURNS** **integer** VOLATILE **LANGUAGE** 'plpgsql' **AS** $$

**DECLARE**

ResulrCount **int**;

**BEGIN**

*-- сперва удалим старые записи*

**DELETE** **FROM** dcdm\_welllog **WHERE** isold=true and (i\_userid is null or (i\_userid is not null and userid=i\_userid));

*-- Зафиксируем новые*

**UPDATE** dcdm\_welllog **SET** userid=null,isnew=null **WHERE** isnew=true and (i\_userid is null or (i\_userid is not null and userid=i\_userid));

 GET DIAGNOSTICS ResulrCount:=ROW\_COUNT;

**RETURN** ResulrCount;

**END**

$$;

**GRANT** **EXECUTE** **ON** **FUNCTION**  commitunfixeddata(dcdm\_welllog.userid%**TYPE**) **TO** dcdmsrv;

*--##############################################################################################*

**CREATE** OR **REPLACE** **FUNCTION** dropunfixeddata(i\_userid dcdm\_welllog.userid%**TYPE**)

**RETURNS** **integer** VOLATILE **LANGUAGE** 'plpgsql' **AS** $$

**DECLARE**

ResulrCount **int**;

**BEGIN**

*-- сперва удалим новые записи*

**DELETE** **FROM** dcdm\_welllog **WHERE** isnew=true and (i\_userid is null or (i\_userid is not null and userid=i\_userid));

*-- вернем старые*

**UPDATE** dcdm\_welllog **SET** userid=null,isold=null **WHERE** isold=true and (i\_userid is null or (i\_userid is not null and userid=i\_userid));

 GET DIAGNOSTICS ResulrCount:=ROW\_COUNT;

**RETURN** ResulrCount;

**END**

$$;

**GRANT** **EXECUTE** **ON** **FUNCTION**  dropunfixeddata(dcdm\_welllog.userid%**TYPE**) **TO** dcdmsrv;

*--##############################################################################################*

**SET** plpgsql.extra\_warnings **TO** 'shadowed\_variables';

**CREATE** OR **REPLACE** **FUNCTION** AddChangeLagData(

i\_oldname dcdm\_lags.sysname%**TYPE**,

i\_name dcdm\_lags.sysname%**TYPE**,

i\_init dcdm\_lags.initvar%**TYPE**,

i\_trunc dcdm\_lags.truncvar%**TYPE**)

**RETURNS** **TABLE**(

sysname dcdm\_lags.sysname%**TYPE**,

initvar dcdm\_lags.initvar%**TYPE**,

*/\*idesc dcdm\_vars.description%TYPE,\*/*

truncvar dcdm\_lags.truncvar%**TYPE**

*/\*,tdesc dcdm\_vars.description%TYPE\*/*)

VOLATILE **LANGUAGE** 'plpgsql' **AS** $$

**BEGIN**

**IF** i\_oldname IS NULL **then**

*-- Добавление*

**INSERT** **INTO** dcdm\_lags(sysname,initvar, truncvar) **VALUES** (i\_name,i\_init,i\_trunc);

**else**

*-- редактирование*

**UPDATE** dcdm\_lags **SET** sysname=i\_name, initvar=i\_init, truncvar=i\_trunc **WHERE** dcdm\_lags.sysname=i\_oldname;

**end** **if**;

**RETURN** QUERY **SELECT** dcdm\_lags.sysname, dcdm\_lags.initvar,*/\*vinit.description as idesc,\*/* dcdm\_lags.truncvar*/\*,vtrunc.description as tdesc\*/*

**FROM** dcdm\_lags

*/\*inner join dcdm\_vars vinit on vinit.sysname=dcdm\_lags.initvar*

*inner join dcdm\_vars vtrunc on vtrunc.sysname=dcdm\_lags.truncvar\*/*

**where** dcdm\_lags.sysname=i\_name;

EXCEPTION

**WHEN** unique\_violation **then** RAISE EXCEPTION 'Name must be unique' **USING** HINT='DCDM', DETAIL='LagNameError';

**WHEN** check\_violation **then** RAISE EXCEPTION 'Invalid character' **USING** HINT='DCDM', DETAIL='LagInvalidChar';

**END**

$$;

**GRANT** **EXECUTE** **ON** **FUNCTION**  AddChangeLagData(dcdm\_lags.sysname%**TYPE**,dcdm\_lags.sysname%**TYPE**,dcdm\_lags.initvar%**TYPE**,dcdm\_lags.truncvar%**TYPE**)

**TO** dcdmsrv;

*--##############################################################################################*

**CREATE** OR **REPLACE** **FUNCTION** RemoveLagItem(i\_sysname dcdm\_lags.sysname%**TYPE**)

**RETURNS** **integer** VOLATILE **LANGUAGE** 'plpgsql' **AS** $$

**DECLARE**

ResulrCount **int**;

**BEGIN**

**DELETE** **FROM** dcdm\_lags **WHERE** dcdm\_lags.sysname=i\_sysname;

GET DIAGNOSTICS ResulrCount:=ROW\_COUNT;

**RETURN** ResulrCount;

**END**

$$;

**GRANT** **EXECUTE** **ON** **FUNCTION**  RemoveLagItem(dcdm\_lags.sysname%**TYPE**) **TO** dcdmsrv;

*--##############################################################################################*

**CREATE** OR **REPLACE** **FUNCTION** addwelllogrec(

i\_acgrp dcdm\_accessgroups.id%**TYPE**,

i\_wellid dcdm\_well.id%**TYPE**,

i\_paramname dcdm\_welllog.paramname%**TYPE**,

i\_type **varchar**(10),

i\_pos float8,

i\_val jsonb)

**RETURNS** **TABLE**(ids dcdm\_welllog.id%**TYPE**)

VOLATILE **LANGUAGE** 'plpgsql' **AS** $$

**BEGIN**

**IF** **EXISTS** (**select** 1 **from** dcdm\_acgroupsfld **where** acgroupid=i\_acgrp and name=i\_paramname) **THEN**

**if** i\_type='depth' **then**

**RETURN** QUERY **INSERT** **INTO** dcdm\_welllog(wellid,**mode**,"time",depth,vdepth,paramname,val) **VALUES** (i\_wellid,0,null,i\_pos,null,i\_paramname,i\_val) returning id;

  elsif  i\_type='vdepth' **then**

**RETURN** QUERY **INSERT** **INTO** dcdm\_welllog(wellid,**mode**,"time",depth,vdepth,paramname,val) **VALUES** (i\_wellid,0,null,null,i\_pos,i\_paramname,i\_val) returning id;

  elsif i\_type='time' **then**

**RETURN** QUERY **INSERT** **INTO** dcdm\_welllog(wellid,**mode**,"time",depth,vdepth,paramname,val) **VALUES** (i\_wellid,0,to\_timestamp(i\_pos),null,null,i\_paramname,i\_val) returning id;

**else**

   RAISE EXCEPTION 'Unknown scale type' **USING** HINT='DCDM', DETAIL='UnkScaleType';

**end** **if**;

**ELSE**

  RAISE EXCEPTION 'Access denied' **USING** HINT='DCDM', DETAIL='AddWellLogAcDen';

**END** **IF**;

**END**

$$;

**GRANT** **EXECUTE** **ON** **FUNCTION**  addwelllogrec(dcdm\_accessgroups.id%**TYPE**,dcdm\_well.id%**TYPE**,dcdm\_welllog.paramname%**TYPE**,**varchar**(10),float8,jsonb) **TO** dcdmsrv;

*--##############################################################################################*

**CREATE** OR **REPLACE** **FUNCTION** addchangemarker(

i\_userid dcdm\_markers.userid%**TYPE**,

i\_wellid dcdm\_markers.wellid%**TYPE**,

i\_param dcdm\_markers.**type**%**TYPE**,

i\_id dcdm\_markers.id%**TYPE**,

i\_type **varchar**(10),

i\_from float8,

i\_to float8,

i\_code dcdm\_markers.code%**TYPE**)

**RETURNS** **TABLE**(id dcdm\_markers.id%**TYPE**)

VOLATILE **LANGUAGE** 'plpgsql' **AS** $$

**BEGIN**

**if** i\_id=0 **then**

**INSERT** **INTO** dcdm\_markers(wellid, userid, **type**, depth, vdepth, "time", code)

**VALUES** (i\_wellid,i\_userid,i\_param,

**case** **when** 'depth'=i\_type **then** floatrange(i\_from,i\_to,'[]') **else** null **end**,

**case** **when** 'vdepth'=i\_type **then** floatrange(i\_from,i\_to,'[]') **else** null **end**,

**case** **when** 'time'=i\_type **then** tsrange(to\_timestamp(i\_from)::**timestamp**,to\_timestamp(i\_to)::**timestamp**,'[]') **else** null **end**,

    i\_code);

**select** lastval() **into** i\_id;

**else**

**UPDATE** dcdm\_markers **SET** userid=i\_userid, changetime=**CURRENT\_TIMESTAMP**,

    depth=**case** **when** 'depth'=i\_type **then** floatrange(i\_from,i\_to,'[]') **else** null **end**,

    vdepth=**case** **when** 'vdepth'=i\_type **then** floatrange(i\_from,i\_to,'[]') **else** null **end**,

    "time"=**case** **when** 'time'=i\_type **then** tsrange(to\_timestamp(i\_from)::**timestamp**,to\_timestamp(i\_to)::**timestamp**,'[]') **else** null **end**,

    code=i\_code

**WHERE** dcdm\_markers.id=i\_id AND dcdm\_markers.wellid=i\_wellid AND dcdm\_markers.**type**=i\_param;

**end** **if**;

**RETURN** QUERY **select** dcdm\_markers.id **from** dcdm\_markers **where** dcdm\_markers.id=i\_id;

**END**

$$;

**GRANT** **EXECUTE** **ON** **FUNCTION**  addchangemarker(dcdm\_markers.userid%**TYPE**,dcdm\_markers.wellid%**TYPE**,dcdm\_markers.**type**%**TYPE**,dcdm\_markers.id%**TYPE**,**varchar**(10),float8,float8,dcdm\_markers.code%**TYPE**) **TO** dcdmsrv;

*--##############################################################################################*

**CREATE** OR **REPLACE** **FUNCTION** SetSortVarPos(i\_sysname dcdm\_vars.sysname%**TYPE**,i\_pos dcdm\_vars.sortpos%**TYPE**)

**RETURNS** **TABLE**(

sysname dcdm\_vars.sysname%**TYPE**,

sortpos dcdm\_vars.sortpos%**TYPE**)

VOLATILE **LANGUAGE** 'plpgsql' **AS** $$

**BEGIN**

*-- Увеличип номер позиции сортировки всех переменных которые больше чем наш номер*

**UPDATE** dcdm\_vars **SET** sortpos=dcdm\_vars.sortpos+1 **WHERE** dcdm\_vars.sortpos>=i\_pos;

*-- Обновим позицию у целевой переменной*

**UPDATE** dcdm\_vars **SET** sortpos=i\_pos **WHERE** dcdm\_vars.sysname=i\_sysname;

**RETURN** QUERY **select** dcdm\_vars.sysname,dcdm\_vars.sortpos **from** dcdm\_vars **where** dcdm\_vars.sysname=i\_sysname;

**END**

$$;

**GRANT** **EXECUTE** **ON** **FUNCTION**  SetSortVarPos(dcdm\_vars.sysname%**TYPE**,dcdm\_vars.sortpos%**TYPE**) **TO** dcdmsrv;

*--##############################################################################################*

**CREATE** OR **REPLACE** **FUNCTION** addlythology(

i\_acgroup dcdm\_accessgroups.id%**TYPE**,

i\_from dcdm\_welllog.depth%**TYPE**,

i\_to dcdm\_welllog.depth%**TYPE**,

i\_wellid dcdm\_welllog.wellid%**TYPE**,

i\_param dcdm\_welllog.paramname%**TYPE**,

i\_code **int**)

**RETURNS** **integer** VOLATILE **LANGUAGE** 'plpgsql' **AS** $$

**DECLARE**

 curcode jsonb;

RecCount **int**;

**BEGIN**

*-- Проверим группу доступа*

**IF** **EXISTS** (**SELECT** id **FROM** dcdm\_acgroupsfld **where** name=i\_param and acgroupid=i\_acgroup) **THEN**

*-- Определим есть ли еще записи за нижней границей диапазона*

**IF** **EXISTS** (**SELECT** id **FROM** dcdm\_welllog **where** wellid=i\_wellid AND paramname=i\_param AND depth>i\_to AND isnew is null or isnew=false **order** **by** depth **limit** 1) **THEN**

*-- Определим чем заканчивается сейчас наш диапазон*

**SELECT** val **into** curcode **FROM** dcdm\_welllog **where** wellid=i\_wellid AND paramname=i\_param AND depth<=i\_to AND isnew is null or isnew=false **order** **by** depth **desc** **limit** 1;

**ELSE**

    curcode:=jsonb\_build\_object('val',null);

**END** **IF**;

*-- Удалим все что попадает в наш диапазон*

**DELETE** **FROM** dcdm\_welllog     **WHERE**  wellid=i\_wellid AND paramname=i\_param AND depth between i\_from and i\_to AND isnew is null or isnew=false;

*-- Добавим данные начала диапазона*

**INSERT** **INTO** dcdm\_welllog(wellid,**mode**,"time",depth,paramname, val)        **VALUES** (i\_wellid,0,null,i\_from,i\_param,jsonb\_build\_object('val',i\_code));

  GET DIAGNOSTICS RecCount:=ROW\_COUNT;

*-- Добавим данные окончания диапазона только если завершение не null*

**INSERT** **INTO** dcdm\_welllog(wellid,**mode**,"time",depth,paramname, val)        **VALUES** (i\_wellid,0,null,i\_to,i\_param,curcode);

**ELSE**

  RAISE EXCEPTION 'Forbidden' **USING** HINT='DCDM', DETAIL='Forbidden';

**END** **IF**;

**RETURN** RecCount;

**END**

$$;

**GRANT** **EXECUTE** **ON** **FUNCTION**  addlythology(dcdm\_accessgroups.id%**TYPE**,dcdm\_welllog.depth%**TYPE**,dcdm\_welllog.depth%**TYPE**,dcdm\_welllog.wellid%**TYPE**,dcdm\_welllog.paramname%**TYPE**,**int**) **TO** dcdmsrv;

*--#############################################################################################*

**CREATE** OR **REPLACE** **FUNCTION** addsludge(

i\_acgroup dcdm\_accessgroups.id%**TYPE**,

i\_from dcdm\_welllog.depth%**TYPE**,

i\_to dcdm\_welllog.depth%**TYPE**,

i\_wellid dcdm\_welllog.wellid%**TYPE**,

i\_params json)

**RETURNS** **integer** VOLATILE **LANGUAGE** 'plpgsql' **AS** $$

**DECLARE**

 curcodes json;

RecCount **int**;

NewParamCount **int**; *-- число уникальных добавляемых параметров*

GrpParamCount **int**; *-- число уникальных добавляемых параметров в указанной группе доступа*

**BEGIN**

*-- Получим число добавляемых полй и полей в группе доступа*

**select** count(**distinct** obj->>'p') **into** NewParamCount **FROM** json\_array\_elements(i\_params) obj;

**select** count(**distinct** obj->>'p') **into** GrpParamCount **FROM** json\_array\_elements(i\_params) obj **inner** **join** dcdm\_acgroupsfld a **on** a.name=obj->>'p';

*-- Проверим совпадают ли счетчики*

**IF** NewParamCount=GrpParamCount **THEN**

*-- Определим есть ли еще записи за нижней границей диапазона хотя бы для одного из наших параметров*

**IF** **EXISTS** (**SELECT** id **FROM** dcdm\_welllog **where** wellid=i\_wellid AND paramname in (**SELECT** obj->>'p' **FROM**   json\_array\_elements(i\_params) obj) AND depth>i\_to AND isnew is null or isnew=false **order** **by** depth **limit** 1) **THEN**

*-- Запомним чем заканчивается сейчас наш диапазон чтоб потом вставить эти данные после добавляемого блока*

**SELECT** json\_agg(json\_build\_object('p',a.paramname,'v',a.val)) **into** curcodes **FROM**

        (**select** **distinct** **on**(paramname) depth, w.paramname **as** paramname, w.val **as** val

**FROM** dcdm\_welllog w **inner** **join** dcdm\_acgroupsfld a **on** a.name=w.paramname

**where** a.acgroupid=i\_acgroup

               and (w.isnew is null or w.isnew=false)

               and w.wellid=i\_wellid

               and w.paramname in(**SELECT** obj->>'p' **FROM** json\_array\_elements(i\_params) obj)

               and w.depth<=i\_to

**order** **by** paramname,depth **desc**) a;

**ELSE**

*-- Ели данных дальше нет то создадим заглушку с null значениями для всез наших параметров*

**SELECT** json\_agg(json\_build\_object('p',obj->>'p','v',json\_build\_object('val',null))) **into** curcodes **FROM** json\_array\_elements(i\_params) obj;

**END** **IF**;

*-- Удалим все что попадает в наш диапазон*

**DELETE** **FROM** dcdm\_welllog     **WHERE**  wellid=i\_wellid AND paramname in(**SELECT** obj->>'p' **FROM** json\_array\_elements(i\_params) obj) AND depth between i\_from and i\_to AND isnew is null or isnew=false;

*-- Добавим данные начала диапазона*

**INSERT** **INTO** dcdm\_welllog(wellid,**mode**,"time",depth,paramname, val)        **SELECT** i\_wellid,0,null,i\_from,obj->>'p',json\_build\_object('val',(obj->>'v')::**int**) **FROM** json\_array\_elements(i\_params) obj;

  GET DIAGNOSTICS RecCount:=ROW\_COUNT;

*-- Добавим данные окончания диапазона*

**INSERT** **INTO** dcdm\_welllog(wellid,**mode**,"time",depth,paramname, val) **SELECT** i\_wellid,0,null,i\_to,obj->>'p',(obj->>'v')::jsonb **FROM** json\_array\_elements(curcodes) obj;

**ELSE** RAISE EXCEPTION 'Forbidden' **USING** HINT='DCDM', DETAIL='Forbidden'; **END** **IF**;

**RETURN** RecCount;

**END**

$$;

**GRANT** **EXECUTE** **ON** **FUNCTION**  addsludge(dcdm\_accessgroups.id%**TYPE**,dcdm\_welllog.depth%**TYPE**,dcdm\_welllog.depth%**TYPE**,dcdm\_welllog.wellid%**TYPE**,json) **TO** dcdmsrv;

*--##############################################################################################*

**CREATE** OR **REPLACE** **FUNCTION** ApplayParamAcGrpFilter(i\_acgrp DCDM\_Users.AcGroup%**TYPE**,i\_param **varchar**)

**RETURNS** **varchar**(20)[] IMMUTABLE **LANGUAGE** 'plpgsql'

**AS** $$

**BEGIN**

**return** (**select** array( **select** name **from** unnest(string\_to\_array(i\_param,',')) pn

**inner** **join** dcdm\_acgroupsfld a **on** a.name=pn.pn **where** a.acgroupid=i\_acgrp));

**END**

$$;

**GRANT** **EXECUTE** **ON** **FUNCTION**  ApplayParamAcGrpFilter(DCDM\_Users.AcGroup%**TYPE**,**varchar**) **TO** dcdmsrv;

*--##############################################################################################*

*/\**

*Функция подготовки бвзы для открытия проекта.Она удаляет таблицы проекта prj\_dcdm\_%*

*\*/*

**CREATE** OR **REPLACE** **FUNCTION** prj\_clear()

**RETURNS** void VOLATILE **LANGUAGE** 'plpgsql'

**AS** $$

**DECLARE**

**row**      record;

**BEGIN**

*-- Удалим все старые таблицы проекта ориентируясь на префикс имени prj\_dcdm\_*

**FOR** **row** IN

**SELECT** table\_name **FROM** information\_schema.**tables**

**WHERE** table\_type='BASE TABLE' AND table\_schema='public' AND table\_name ILIKE ('prj\_dcdm\_%')

**LOOP**

**EXECUTE** 'DROP TABLE ' || quote\_ident(**row**.table\_name);

**END** **LOOP**;

**END**

$$;

**GRANT** **EXECUTE** **ON** **FUNCTION**  prj\_clear() **TO** dcdmsrv;

*--##############################################################################################*

*/\**

*Функция создает уникальный идентификатор для вызова второго шага восстановления которые записывается*

*в таблицу prj\_dcdm\_cfg*

*\*/*

**CREATE** OR **REPLACE** **FUNCTION** prj\_getrestorecode()

**RETURNS** **varchar** VOLATILE **LANGUAGE** 'plpgsql'

**AS** $$

**DECLARE**

guid     **varchar**=uuid\_generate\_v4()::**text**;

**BEGIN**

**INSERT** **INTO** prj\_dcdm\_cfg(**key**,**value**) **VALUES** ('id',guid);

**return** guid;

**END**

$$;

**GRANT** **EXECUTE** **ON** **FUNCTION**  prj\_getrestorecode() **TO** dcdmsrv;

*--##############################################################################################*

**CREATE** OR **REPLACE** **FUNCTION** prj\_backup(cfg jsonb)

**RETURNS** **varchar** VOLATILE **LANGUAGE** 'plpgsql'

**AS** $$

**DECLARE**

tbl\_list **varchar**='';

**row**      record;

**BEGIN**

*-- Удалим все старые таблицы проекта ориентируясь на префикс имени prj\_dcdm\_*

**FOR** **row** IN

**SELECT** table\_name **FROM** information\_schema.**tables**

**WHERE** table\_type='BASE TABLE' AND table\_schema='public' AND table\_name ILIKE ('prj\_dcdm\_%')

**LOOP**

**EXECUTE** 'DROP TABLE ' || quote\_ident(**row**.table\_name);

**END** **LOOP**;

*-- Создадим таблицу единичных значений которые должны попасть в проект (это не отражание dcdm\_config хотя и пересекается по данным)*

**create** **table** prj\_dcdm\_cfg(**key** **varchar**(20) NOT NULL **PRIMARY** **KEY**,**value** **varchar**);

**INSERT** **INTO** prj\_dcdm\_cfg(**key**,**value**) **VALUES** ('CreateDate',**current\_timestamp**::**text**); *-- Дата создания проекта*

*-- Начнем копирование данных согласно запроса*

*-- Медиатека*

**if**(cfg*#>'{Base,SaveMLib}')::bool=true then*

**create** **table** prj\_dcdm\_mlib(like dcdm\_mlib);

**insert** **into** prj\_dcdm\_mlib **select** \* **from** dcdm\_mlib;

*-- Сохраним базовые карты соответствия*

**INSERT** **INTO** prj\_dcdm\_cfg(**key**,**value**) **select** **key**,**data** **from** dcdm\_config **where** **key** in ('LYTHNAMEMAP','LYTHCODEMAP','MARKCODEMAP');

**end** **if**;

*-- Профили интеграции*

**if**(cfg*#>'{Base,SaveIntg}')::bool=true then*

**create** **table** prj\_dcdm\_profiles(like dcdm\_profiles);

**create** **table** prj\_dcdm\_ipoints(like dcdm\_ipoints);

**insert** **into** prj\_dcdm\_profiles **select** \* **from** dcdm\_profiles **where** id=**any**((**select** array\_agg(**value**::**int**) **from** jsonb\_array\_elements(cfg->'IProfiles'))::**int**[]);

**insert** **into** prj\_dcdm\_ipoints **select** \* **from** dcdm\_ipoints **where** profileid=**any**((**select** array\_agg(**value**::**int**) **from** jsonb\_array\_elements(cfg->'IProfiles'))::**int**[]);

**end** **if**;

*-- Конфигуратор ввода*

**if**(cfg*#>'{Base,SaveCalc}')::bool=true then*

**create** **table** prj\_dcdm\_vargroups(like dcdm\_vargroups);

**create** **table** prj\_dcdm\_vars(like dcdm\_vars);

**create** **table** prj\_dcdm\_lags(like dcdm\_lags);

**insert** **into** prj\_dcdm\_vargroups **select** \* **from** dcdm\_vargroups;

**insert** **into** prj\_dcdm\_vars **select** \* **from** dcdm\_vars;

**insert** **into** prj\_dcdm\_lags **select** \* **from** dcdm\_lags;

*-- Сохраним скрипты перд и пост обработки*

**INSERT** **INTO** prj\_dcdm\_cfg(**key**,**value**) **select** **key**,**data** **from** dcdm\_config **where** **key** in ('VarPreSc','VarPostSc');

**end** **if**;

*-- Публичные справочники*

**if**(cfg*#>'{Base,SavePubRef}')::bool=true then*

**create** **table** prj\_dcdm\_referenceslist(like dcdm\_referenceslist);

**create** **table** prj\_dcdm\_referencecol(like dcdm\_referencecol);

**insert** **into** prj\_dcdm\_referenceslist **select** \* **from** dcdm\_referenceslist

**where** iswelldep=false

         and id=**any**((**select** array\_agg(**value**::**int**) **from** jsonb\_array\_elements(cfg->'PubRefs'))::**int**[]);

**insert** **into** prj\_dcdm\_referencecol **select** dcdm\_referencecol.\*

**from** dcdm\_referencecol

**inner** **join** dcdm\_referenceslist **on** dcdm\_referenceslist.id=dcdm\_referencecol.reflistid

**where** dcdm\_referenceslist.iswelldep=false

               and dcdm\_referenceslist.id=**any**((**select** array\_agg(**value**::**int**) **from** jsonb\_array\_elements(cfg->'PubRefs'))::**int**[]);

*-- Данные публичных справочников (только если сохраняем структуру)*

**if**(cfg*#>'{Base,SavePubRefData}')::bool=true then*

**create** **table** prj\_dcdm\_referencedata(like dcdm\_referencedata);

**insert** **into** prj\_dcdm\_referencedata **select** dcdm\_referencedata.\*

**from** dcdm\_referencedata

**inner** **join** dcdm\_referenceslist **on** dcdm\_referenceslist.id=dcdm\_referencedata.reflistid

**where** dcdm\_referenceslist.iswelldep=false

                 and dcdm\_referenceslist.id=**any**((**select** array\_agg(**value**::**int**) **from** jsonb\_array\_elements(cfg->'PubRefs'))::**int**[]);

**end** **if**;

**end** **if**;

*-- Скважины*

**if**(jsonb\_array\_length(cfg->'Wells')>0) **then**

**create** **table** prj\_dcdm\_well(like dcdm\_well);

**insert** **into** prj\_dcdm\_well **select** \*

**from** dcdm\_well

**where** id=**any**((**select** array\_agg(**value**::**int**) **from** jsonb\_array\_elements(cfg->'Wells'))::**int**[]);

**end** **if**;

*-- Оперативные справочники*

**if**(cfg*#>'{Base,SaveOpRef}')::bool=true then*

**create** **table** **IF** NOT **EXISTS** prj\_dcdm\_referenceslist(like dcdm\_referenceslist);

**create** **table** **IF** NOT **EXISTS** prj\_dcdm\_referencecol(like dcdm\_referencecol);

**insert** **into** prj\_dcdm\_referenceslist **select** \* **from** dcdm\_referenceslist

**where** iswelldep=true

               and dcdm\_referenceslist.id=**any**((**select** array\_agg(**value**::**int**) **from** jsonb\_array\_elements(cfg->'OpRefs'))::**int**[]);

**insert** **into** prj\_dcdm\_referencecol **select** dcdm\_referencecol.\*

**from** dcdm\_referencecol

**inner** **join** dcdm\_referenceslist **on** dcdm\_referenceslist.id=dcdm\_referencecol.reflistid

**where** dcdm\_referenceslist.iswelldep=true

              and dcdm\_referenceslist.id=**any**((**select** array\_agg(**value**::**int**) **from** jsonb\_array\_elements(cfg->'OpRefs'))::**int**[]);

**if**(cfg*#>'{Base,SaveOpRefData}')::bool=true and (jsonb\_array\_length(cfg->'Wells')>0) then*

**create** **table** **IF** NOT **EXISTS** prj\_dcdm\_referencedata(like dcdm\_referencedata);

**insert** **into** prj\_dcdm\_referencedata **select** dcdm\_referencedata.\*

**from** dcdm\_referencedata

**inner** **join** dcdm\_referenceslist **on** dcdm\_referenceslist.id=dcdm\_referencedata.reflistid

**where** dcdm\_referenceslist.iswelldep=true

                 and dcdm\_referenceslist.id=**any**((**select** array\_agg(**value**::**int**) **from** jsonb\_array\_elements(cfg->'OpRefs'))::**int**[])

                 and wellid=**any**((**select** array\_agg(**value**::**int**) **from** jsonb\_array\_elements(cfg->'Wells'))::**int**[]);

**end** **if**;

**end** **if**;

*-- Скрипты базы данных*

**if**(cfg*#>'{Base,SaveScript}')::bool=true then*

**create** **table** prj\_dcdm\_dbmuserscripts(like dcdm\_dbmuserscripts);

**insert** **into** prj\_dcdm\_dbmuserscripts **select** \* **from** dcdm\_dbmuserscripts

**where** ispublic=true and id=**any**((**select** array\_agg(trim('"' **FROM** **value**::**text**)::uuid)

**from** jsonb\_array\_elements(cfg->'Scripts'))::uuid[]);

**end** **if**;

*-- Журнал скважины*

**if**(cfg*#>'{Base,SaveWell}')::bool=true and (jsonb\_array\_length(cfg->'Wells')>0) then*

*-- Выбранные параметры журнала*

**if**(jsonb\_array\_length(cfg->'WellParams')>0) **then**

**create** **table** prj\_dcdm\_welllog(like dcdm\_welllog);

**insert** **into** prj\_dcdm\_welllog **select** \*

**from** dcdm\_welllog

**where** wellid=**any**((**select** array\_agg(**value**::**int**) **from** jsonb\_array\_elements(cfg->'Wells'))::**int**[])

                       and paramname=**any**((**select** array\_agg(**value**::**varchar**) **from** jsonb\_array\_elements\_text(cfg->'WellParams'))::**varchar**[])

                and (isnew is null or isnew=false);

**end** **if**;

*-- Индивидуальные настройки*

**if**(cfg*#>'{Base,SaveWellCfg}')::bool=true then*

**create** **table** **IF** NOT **EXISTS** prj\_dcdm\_welllog(like dcdm\_welllog);

**insert** **into** prj\_dcdm\_welllog **select** \*

**from** dcdm\_welllog

**where** wellid=**any**((**select** array\_agg(**value**::**int**) **from** jsonb\_array\_elements(cfg->'Wells'))::**int**[])

                and depth is null and vdepth is null and **time** is null;

**end** **if**;

*-- Инженерные отметки*

**if**(cfg*#>'{Base,SaveWellMrk}')::bool=true then*

**create** **table** prj\_dcdm\_markers(like dcdm\_markers);

**insert** **into** prj\_dcdm\_markers **select** \*

**from** dcdm\_markers

**where** wellid=**any**((**select** array\_agg(**value**::**int**) **from** jsonb\_array\_elements(cfg->'Wells'))::**int**[]);

*-- обнулим поле userid так как информация о пользователях не сохраняется в проекте*

**UPDATE** prj\_dcdm\_markers **SET** userid=null;

**end** **if**;

*-- Комментарии*

**if**(cfg*#>'{Base,SaveWellCmt}')::bool=true then*

**create** **table** prj\_dcdm\_commentsandfiles(like dcdm\_commentsandfiles);

*-- Копируем все(включая файлы), а потом обнуляем то, что не нужно*

**insert** **into** prj\_dcdm\_commentsandfiles **select** \*

**from** dcdm\_commentsandfiles

**where** wellid=**any**((**select** array\_agg(**value**::**int**) **from** jsonb\_array\_elements(cfg->'Wells'))::**int**[]);

*-- обнулим поле userid так как информация о пользователях не сохраняется в проекте*

**UPDATE** prj\_dcdm\_commentsandfiles **SET** userid=null;

*-- обнулим данные файла если не указано, что мы должны их скопировать*

**if**(cfg*#>'{Base,SaveWellCmtFiles}')::bool=false then*

**UPDATE** prj\_dcdm\_commentsandfiles **SET** filedata=null,orgfilename=null **where** filedata is not null or orgfilename is not null;

**end** **if**;

**end** **if**;

**end** **if**;

*-- Составим список созданных таблиц проекта ориентируясь на префикс имени prj\_dcdm\_*

**FOR** **row** IN

**SELECT** table\_name **FROM** information\_schema.**tables**

**WHERE** table\_type='BASE TABLE' AND table\_schema='public' AND table\_name ILIKE ('prj\_dcdm\_%')

**LOOP**

  tbl\_list:=tbl\_list||**row**.table\_name||',';

**END** **LOOP**;

**return** tbl\_list;

**END**

$$;

**GRANT** **EXECUTE** **ON** **FUNCTION**  prj\_backup(jsonb) **TO** dcdmsrv;

*--#############################################################################################*

*/\**

*Сбор данных о проекте*

*\*/*

**CREATE** OR **REPLACE** **FUNCTION** prj\_get\_config(i\_id **varchar**(40))

**RETURNS** json VOLATILE **LANGUAGE** 'plpgsql'

**AS** $$

**DECLARE**

cfg jsonb='{"Base":{}}';

**BEGIN**

**if** **EXISTS**(**SELECT** 1 **FROM** information\_schema.**tables** **WHERE** table\_schema='public' AND table\_name='prj\_dcdm\_cfg') **then**

**if** **EXISTS**(**SELECT** **key** **FROM** prj\_dcdm\_cfg **WHERE** **key**='id' AND **value**=i\_id) **then**

*-- Медиатека*

**if** **EXISTS**(**SELECT** 1 **FROM** information\_schema.**tables** **WHERE** table\_schema='public' AND table\_name='prj\_dcdm\_mlib') **then** cfg:=jsonb\_set(cfg,'{Base,MLib}','true');**end** **if**;

*-- Профили интеграции*

**if** **EXISTS**(**SELECT** 1 **FROM** information\_schema.**tables** **WHERE** table\_schema='public' AND table\_name='prj\_dcdm\_profiles')

               AND **EXISTS**(**SELECT** 1 **FROM** information\_schema.**tables** **WHERE** table\_schema='public' AND table\_name='prj\_dcdm\_ipoints') **then**

                 cfg:=jsonb\_set(cfg,'{Base,Intg}','true');

                 cfg:=jsonb\_set(cfg,'{Profiles}'::**text**[],(**select** array\_to\_json(array\_agg(row\_to\_json(r))) **from** (**SELECT** id, name **FROM** prj\_dcdm\_profiles) r)::jsonb);

**end** **if**;

*-- Конфигуратор ввода*

**if** **EXISTS**(**SELECT** 1 **FROM** information\_schema.**tables** **WHERE** table\_schema='public' AND table\_name='prj\_dcdm\_vargroups')

               AND **EXISTS**(**SELECT** 1 **FROM** information\_schema.**tables** **WHERE** table\_schema='public' AND table\_name='prj\_dcdm\_vars')

               AND **EXISTS**(**SELECT** 1 **FROM** information\_schema.**tables** **WHERE** table\_schema='public' AND table\_name='prj\_dcdm\_lags') **then** cfg:=jsonb\_set(cfg,'{Base,Calc}','true');**end** **if**;

*-- Структура и данные публичных справочников*

**if** **EXISTS**(**SELECT** 1 **FROM** information\_schema.**tables** **WHERE** table\_schema='public' AND table\_name='prj\_dcdm\_referenceslist')

               AND **EXISTS**(**SELECT** 1 **FROM** information\_schema.**tables** **WHERE** table\_schema='public' AND table\_name='prj\_dcdm\_referencecol') **then**

**if** **EXISTS**(**select** \* **from** prj\_dcdm\_referenceslist **where** iswelldep=false **limit** 1) **then**

                  cfg:=jsonb\_set(cfg,'{Base,PubRef}','true');

                  cfg:=jsonb\_set(cfg,'{PubRefs}'::**text**[],(**select** array\_to\_json(array\_agg(row\_to\_json(r))) **from** (**SELECT** id, name **FROM** prj\_dcdm\_referenceslist **where** iswelldep=false) r)::jsonb);

**if** **EXISTS**(**SELECT** 1 **FROM** information\_schema.**tables** **WHERE** table\_schema='public' AND table\_name='prj\_dcdm\_referencedata') **then**

**if** **EXISTS**(

**select** prj\_dcdm\_referencedata.\*

**from** prj\_dcdm\_referencedata

**inner** **join** prj\_dcdm\_referenceslist **on** prj\_dcdm\_referenceslist.id=prj\_dcdm\_referencedata.reflistid

**where** prj\_dcdm\_referenceslist.iswelldep=false

**limit** 1

                         ) **then**

                          cfg:=jsonb\_set(cfg,'{Base,PubRefData}','true');

**end** **if**;

**end** **if**;

**end** **if**;

**end** **if**;

*-- Скрипты*

**if** **EXISTS**(**SELECT** 1 **FROM** information\_schema.**tables** **WHERE** table\_schema='public' AND table\_name='prj\_dcdm\_dbmuserscripts') **then**

                 cfg:=jsonb\_set(cfg,'{Base,Scripts}','true');

                 cfg:=jsonb\_set(cfg,'{Scripts}'::**text**[],(**select** array\_to\_json(array\_agg(row\_to\_json(r))) **from** (**SELECT** id, name **FROM** prj\_dcdm\_dbmuserscripts) r)::jsonb);

**end** **if**;

*--*

**if** **EXISTS**(**SELECT** 1 **FROM** information\_schema.**tables** **WHERE** table\_schema='public' AND table\_name='prj\_dcdm\_well') **then**

               cfg:=jsonb\_set(cfg,'{Wells}'::**text**[],(**select** array\_to\_json(array\_agg(row\_to\_json(r))) **from** (**select** id,name **from** prj\_dcdm\_well) r)::jsonb);

*-- Структура и данные оперативных справочников*

**if** **EXISTS**(**SELECT** 1 **FROM** information\_schema.**tables** **WHERE** table\_schema='public' AND table\_name='prj\_dcdm\_referenceslist')

                 AND **EXISTS**(**SELECT** 1 **FROM** information\_schema.**tables** **WHERE** table\_schema='public' AND table\_name='prj\_dcdm\_referencecol') **then**

**if** **EXISTS**(**select** \* **from** prj\_dcdm\_referenceslist **where** iswelldep=true **limit** 1) **then**

                        cfg:=jsonb\_set(cfg,'{Base,OpRef}','true');

                       cfg:=jsonb\_set(cfg,'{OpRefs}'::**text**[],(**select** array\_to\_json(array\_agg(row\_to\_json(r))) **from** (**SELECT** id, name **FROM** prj\_dcdm\_referenceslist **where** iswelldep=true) r)::jsonb);

**if** **EXISTS**(**SELECT** 1 **FROM** information\_schema.**tables** **WHERE** table\_schema='public' AND table\_name='prj\_dcdm\_referencedata')  **then**

**if** **EXISTS**(

**select** prj\_dcdm\_referencedata.\*

**from** prj\_dcdm\_referencedata

**inner** **join** prj\_dcdm\_referenceslist **on** prj\_dcdm\_referenceslist.id=prj\_dcdm\_referencedata.reflistid

**where** prj\_dcdm\_referenceslist.iswelldep=true

**limit** 1

                           ) **then**

                            cfg:=jsonb\_set(cfg,'{Base,OpRefData}','true');

*--cfg:=jsonb\_set(cfg,'{Params}'::text[],(select array\_to\_json(array\_agg(row\_to\_json(r))) from (SELECT distinct paramname FROM prj\_dcdm\_welllog where (depth is not null or time is not null or vdepth is not null)) r)::jsonb);*

**end** **if**;

**end** **if**;

**end** **if**;

**end** **if**;

*-- Журнал скважины*

**if** **EXISTS**(**SELECT** 1 **FROM** information\_schema.**tables** **WHERE** table\_schema='public' AND table\_name='prj\_dcdm\_welllog') **then** cfg:=jsonb\_set(cfg,'{Base,Well}','true');**end** **if**;

*-- Инженерные отметки*

**if** **EXISTS**(**SELECT** 1 **FROM** information\_schema.**tables** **WHERE** table\_schema='public' AND table\_name='prj\_dcdm\_markers') **then** cfg:=jsonb\_set(cfg,'{Base,WellMrk}','true');**end** **if**;

*-- Комментарии и файлы*

**if** **EXISTS**(**SELECT** 1 **FROM** information\_schema.**tables** **WHERE** table\_schema='public' AND table\_name='prj\_dcdm\_commentsandfiles') **then** cfg:=jsonb\_set(cfg,'{Base,WellCmt}','true');**end** **if**;

**end** **if**;

**else**

               RAISE EXCEPTION 'Project id not found' **USING** HINT='DCDM', DETAIL='PrjIdNotFound';

**end** **if**;

**else**

   RAISE EXCEPTION 'Project id not found' **USING** HINT='DCDM', DETAIL='PrjIdNotFound';

**end** **if**;

**return** cfg::json;

**END**

$$;

**GRANT** **EXECUTE** **ON** **FUNCTION**  prj\_get\_config(**varchar**) **TO** dcdmsrv;

*--#############################################################################################*

*/\**

*Перенос данных из таблиц проекта в рабочие таблицы (финал открытия проекта)*

*\*/*

**CREATE** OR **REPLACE** **FUNCTION** prj\_open\_final(i\_id **varchar**(40),cfg jsonb,i\_userid dcdm\_users.id%**TYPE**)

**RETURNS** json VOLATILE **LANGUAGE** 'plpgsql'

**AS** $$

**DECLARE**

v\_r record;   *-- Промежуточные данные для переноса*

v\_new\_id **int**; *-- Новый идентификатор перенесенной записи*

**BEGIN**

**if** **EXISTS**(**SELECT** 1 **FROM** information\_schema.**tables** **WHERE** table\_schema='public' AND table\_name='prj\_dcdm\_cfg') AND **EXISTS**(**SELECT** **key** **FROM** prj\_dcdm\_cfg **WHERE** **key**='id' AND **value**=i\_id) **then**

**if**(cfg*#>'{Base,MLib}')::bool=true then*

*-- Медиатека*

**case** (cfg*#>'{Base,MLibType}')::int*

**when** 1 **then** *-- Заменить*

*-- Удаляем старое*

**delete** **from** dcdm\_mlib;

*-- Переносим данные из проекта*

**insert** **into** dcdm\_mlib(filename,dir,changetime,filedata) **select** filename,dir,changetime,filedata **from** prj\_dcdm\_mlib;

*-- При полной замене медиатаке подредактируем также и карты соответствий*

**delete** **from** dcdm\_config **where** **key** in ('LYTHNAMEMAP','LYTHCODEMAP','MARKCODEMAP');

**INSERT** **INTO** dcdm\_config(**key**,**data**) **select** **key**,**value** **from** prj\_dcdm\_cfg **where** **key** in ('LYTHNAMEMAP','LYTHCODEMAP','MARKCODEMAP');

**when** 2 **then** *-- Совместить*

*-- Удалим пересекающиеся данные*

**delete** **from** dcdm\_mlib **using** prj\_dcdm\_mlib **where** dcdm\_mlib.dir=prj\_dcdm\_mlib.dir and dcdm\_mlib.filename=prj\_dcdm\_mlib.filename;

*-- Переносим данные из проекта*

**insert** **into** dcdm\_mlib(filename,dir,changetime,filedata) **select** filename,dir,changetime,filedata **from** prj\_dcdm\_mlib;

**when** 3 **then** *-- Добавить*

**insert** **into** dcdm\_mlib(filename,dir,changetime,filedata)

**SELECT** prj\_dcdm\_mlib.filename,prj\_dcdm\_mlib.dir,prj\_dcdm\_mlib.changetime,prj\_dcdm\_mlib.filedata

**FROM** prj\_dcdm\_mlib **LEFT** **JOIN** dcdm\_mlib **ON** (prj\_dcdm\_mlib.filename=dcdm\_mlib.filename and prj\_dcdm\_mlib.dir=dcdm\_mlib.dir)

**WHERE** dcdm\_mlib.filename IS NULL;

**end** **case**;

**end** **if**;

*-- Точки интеграции*

**if**(cfg*#>'{Base,Intg}')::bool=true then*

**case** (cfg*#>'{Base,IntgType}')::int*

**when** 1 **then** *-- Заменить*

*-- Удаляем старое*

**delete** **from** dcdm\_profiles;

*-- Переносим данные из проекта*

**FOR** v\_r IN **select** \* **from** prj\_dcdm\_profiles **where** id=**any**((**select** array\_agg(**value**::**int**) **from** jsonb\_array\_elements(cfg->'IProfiles'))::**int**[])

**LOOP**

**INSERT** **INTO** dcdm\_profiles(name, description, isactive) **VALUES** (v\_r.name,v\_r.description,v\_r.isactive) returning id **INTO** v\_new\_id;

**INSERT** **INTO** dcdm\_ipoints(profileid, pointtype, name, config) **select** v\_new\_id, pointtype, name, config **from** prj\_dcdm\_ipoints **where** profileid=v\_r.id;

**END** **LOOP**;

**when** 2 **then** *-- Совместить*

*-- Удалим пересекающиеся данные*

**delete** **from** dcdm\_profiles

**using** prj\_dcdm\_profiles

**where** dcdm\_profiles.name=prj\_dcdm\_profiles.name

                       and prj\_dcdm\_profiles.id=**any**((**select** array\_agg(**value**::**int**) **from** jsonb\_array\_elements(cfg->'IProfiles'))::**int**[]);

*-- Переносим данные из проекта*

**FOR** v\_r IN **select** \* **from** prj\_dcdm\_profiles **where** id=**any**((**select** array\_agg(**value**::**int**) **from** jsonb\_array\_elements(cfg->'IProfiles'))::**int**[])

**LOOP**

**INSERT** **INTO** dcdm\_profiles(name, description, isactive) **VALUES** (v\_r.name,v\_r.description,v\_r.isactive) returning id **INTO** v\_new\_id;

**INSERT** **INTO** dcdm\_ipoints(profileid, pointtype, name, config) **select** v\_new\_id, pointtype, name, config **from** prj\_dcdm\_ipoints **where** profileid=v\_r.id;

**END** **LOOP**;

**when** 3 **then** *-- Добавить*

*--    Переносим данные которые отсутствуют в основной таблице*

**FOR** v\_r IN **select** prj\_dcdm\_profiles.\*

**from** prj\_dcdm\_profiles

**LEFT** **JOIN** dcdm\_profiles **ON** prj\_dcdm\_profiles.name=dcdm\_profiles.name

**where** dcdm\_profiles.id is null and prj\_dcdm\_profiles.id=**any**((**select** array\_agg(**value**::**int**) **from** jsonb\_array\_elements(cfg->'IProfiles'))::**int**[])

**LOOP**

**INSERT** **INTO** dcdm\_profiles(name, description, isactive) **VALUES** (v\_r.name,v\_r.description,v\_r.isactive) returning id **INTO** v\_new\_id;

**INSERT** **INTO** dcdm\_ipoints(profileid, pointtype, name, config) **select** v\_new\_id, pointtype, name, config **from** prj\_dcdm\_ipoints **where** profileid=v\_r.id;

**END** **LOOP**;

**end** **case**;

**end** **if**;

*-- Конфигуратор ввода*

**if**(cfg*#>'{Base,Calc}')::bool=true then*

*-- Подготовим дополнительные колонки для переноса данных*

**BEGIN** **ALTER** **TABLE** prj\_dcdm\_vargroups **ADD** **COLUMN** newid **int**; EXCEPTION **WHEN** duplicate\_column **THEN** **END**;*-- Игнорируем ошибку*

**UPDATE** prj\_dcdm\_vargroups **SET** newid=null;

*-- Определимя с режимом*

**case** (cfg*#>'{Base,CalcType}')::int*

**when** 1 **then** *-- Заменить*

*-- Удаляем старое*

**delete** **from** dcdm\_lags;

**delete** **from** dcdm\_vars;

**delete** **from** dcdm\_vargroups;

*-- Переносим данные из проекта*

*-- группы*

**FOR** v\_r IN **select** \* **from** prj\_dcdm\_vargroups

**LOOP**

**INSERT** **INTO** dcdm\_vargroups(name, iswidget, isimport, isipoint) **VALUES** (v\_r.name, v\_r.iswidget, v\_r.isimport, v\_r.isipoint) returning id **INTO** v\_new\_id;

**UPDATE** prj\_dcdm\_vargroups **SET** newid=v\_new\_id **where** id=v\_r.id;

**END** **LOOP**;

*-- переменные (без данных о запаздывании)*

**INSERT** **INTO** dcdm\_vars(sysname, **type**, initexpr, calcexpr, tlrdvg, avgmethod, loggedpar, isexternal, timeout, description, sortpos, groupid, filter)

**SELECT** sysname, **type**, initexpr, calcexpr, tlrdvg, avgmethod, loggedpar, isexternal, timeout, description, sortpos, newid, filter

**FROM** prj\_dcdm\_vars **full** **outer** **join** prj\_dcdm\_vargroups **on** prj\_dcdm\_vars.groupid=prj\_dcdm\_vargroups.id;

*-- запаздывание*

**INSERT** **INTO** dcdm\_lags(sysname, initvar, truncvar) **select** sysname, initvar, truncvar **from** prj\_dcdm\_lags;

*-- настройки запаздывания в переменных*

**UPDATE** dcdm\_vars **SET** (lag)=(**SELECT** lag **FROM** prj\_dcdm\_vars **WHERE** prj\_dcdm\_vars.sysname=dcdm\_vars.sysname);

*-- Скрипты пард и постобработки*

**delete** **from** dcdm\_config **where** **key** in ('VarPreSc','VarPostSc');

**INSERT** **INTO** dcdm\_config(**key**,**data**) **select** **key**,**value** **from** prj\_dcdm\_cfg **where** **key** in ('VarPreSc','VarPostSc');

*--when 2 then -- Совместить*

*--when 3 then -- Добавить*

**else** RAISE EXCEPTION 'Unsupportrd update type' **USING** HINT='DCDM', DETAIL='UnsUpdType';

**end** **case**;

**end** **if**;

*-- Публичные справочники ввода*

**if**(cfg*#>'{Base,PubRef}')::bool=true then*

**case** (cfg*#>'{Base,PubRefType}')::int*

**when** 1 **then** *-- Заменить*

*-- Удаляем старое*

**delete** **from** dcdm\_referenceslist **where** iswelldep=false;

*-- Переносим данные из проекта*

**FOR** v\_r IN **select** \* **from** prj\_dcdm\_referenceslist **where** iswelldep=false and id=**any**((**select** array\_agg(**value**::**int**) **from** jsonb\_array\_elements(cfg->'PubRefs'))::**int**[])

**LOOP**

**INSERT** **INTO** dcdm\_referenceslist(sysname, name, iswelldep, browsedata) **VALUES** (v\_r.sysname, v\_r.name, v\_r.iswelldep, v\_r.browsedata) returning id **INTO** v\_new\_id;

**INSERT** **INTO** dcdm\_referencecol(reflistid, **type**, sysname, colname, initexpr) **select** v\_new\_id, **type**, sysname, colname, initexpr **from** prj\_dcdm\_referencecol **where** reflistid=v\_r.id;

*-- Скопируем также данные если требуется*

**if**(cfg*#>'{Base,PubRefData}')::bool=true then  --and EXISTS(SELECT 1 FROM information\_schema.tables WHERE table\_schema='public' AND table\_name='prj\_dcdm\_referencedata')*

**INSERT** **INTO** dcdm\_referencedata(wellid, reflistid, subid, grpname, val) **select** null,v\_new\_id,subid, grpname, val **from** prj\_dcdm\_referencedata **where** reflistid=v\_r.id;

**end** **if**;

**END** **LOOP**;

**when** 2 **then** *-- Совместить*

*-- Удалим пересекающиеся данные*

**delete** **from** dcdm\_referenceslist

**using** prj\_dcdm\_referenceslist

**where** dcdm\_referenceslist.sysname=prj\_dcdm\_referenceslist.sysname and dcdm\_referenceslist.iswelldep=false and prj\_dcdm\_referenceslist.iswelldep=false

                       and prj\_dcdm\_referenceslist.id=**any**((**select** array\_agg(**value**::**int**) **from** jsonb\_array\_elements(cfg->'PubRefs'))::**int**[]);

*-- Переносим данные из проекта*

**FOR** v\_r IN **select** \* **from** prj\_dcdm\_referenceslist **where** iswelldep=false and id=**any**((**select** array\_agg(**value**::**int**) **from** jsonb\_array\_elements(cfg->'PubRefs'))::**int**[])

**LOOP**

**INSERT** **INTO** dcdm\_referenceslist(sysname, name, iswelldep, browsedata) **VALUES** (v\_r.sysname, v\_r.name, v\_r.iswelldep, v\_r.browsedata) returning id **INTO** v\_new\_id;

**INSERT** **INTO** dcdm\_referencecol(reflistid, **type**, sysname, colname, initexpr) **select** v\_new\_id, **type**, sysname, colname, initexpr **from** prj\_dcdm\_referencecol **where** reflistid=v\_r.id;

*-- Скопируем также данные если требуется*

**if**(cfg*#>'{Base,PubRefData}')::bool=true then  --and EXISTS(SELECT 1 FROM information\_schema.tables WHERE table\_schema='public' AND table\_name='prj\_dcdm\_referencedata')*

**INSERT** **INTO** dcdm\_referencedata(wellid, reflistid, subid, grpname, val) **select** null,v\_new\_id,subid, grpname, val **from** prj\_dcdm\_referencedata **where** reflistid=v\_r.id;

**end** **if**;

**END** **LOOP**;

**when** 3 **then** *-- Добавить*

*--    Переносим данные которые отсутствуют в основной таблице*

**FOR** v\_r IN **select** prj\_dcdm\_referenceslist.\*

**from** prj\_dcdm\_referenceslist

**LEFT** **JOIN** dcdm\_referenceslist **ON** prj\_dcdm\_referenceslist.sysname=dcdm\_referenceslist.sysname and prj\_dcdm\_referenceslist.iswelldep=dcdm\_referenceslist.iswelldep

**where** dcdm\_referenceslist.id is null and prj\_dcdm\_referenceslist.iswelldep=false and prj\_dcdm\_referenceslist.id=**any**((**select** array\_agg(**value**::**int**) **from** jsonb\_array\_elements(cfg->'PubRefs'))::**int**[])

**LOOP**

**INSERT** **INTO** dcdm\_referenceslist(sysname, name, iswelldep, browsedata) **VALUES** (v\_r.sysname, v\_r.name, v\_r.iswelldep, v\_r.browsedata) returning id **INTO** v\_new\_id;

**INSERT** **INTO** dcdm\_referencecol(reflistid, **type**, sysname, colname, initexpr) **select** v\_new\_id, **type**, sysname, colname, initexpr **from** prj\_dcdm\_referencecol **where** reflistid=v\_r.id;

*-- Скопируем также данные если требуется*

**if**(cfg*#>'{Base,PubRefData}')::bool=true then  --and EXISTS(SELECT 1 FROM information\_schema.tables WHERE table\_schema='public' AND table\_name='prj\_dcdm\_referencedata')*

**INSERT** **INTO** dcdm\_referencedata(wellid, reflistid, subid, grpname, val) **select** null,v\_new\_id,subid, grpname, val **from** prj\_dcdm\_referencedata **where** reflistid=v\_r.id;

**end** **if**;

**END** **LOOP**;

**end** **case**;

**end** **if**;

*-- Обработаем скважины так как они нужнв для оперативных справочниов и журнала скважины*

**if**((cfg*#>'{Base,OpRef}')::bool=true and (cfg#>'{Base,OpRefData}')::bool=true) or (cfg#>'{Base,Well}')::bool=true then*

*-- Подготовим дополнительную колонку для сохранения нового идентификатора скважины*

**BEGIN** **ALTER** **TABLE** prj\_dcdm\_well **ADD** **COLUMN** newid **int**; EXCEPTION **WHEN** duplicate\_column **THEN** **END**;*-- Игнорируем ошибку*

**UPDATE** prj\_dcdm\_well **SET** newid=null;

*-- создадим новые скважины на которые будут ссылки*

**FOR** v\_r IN **select** \* **from** prj\_dcdm\_well **where** id=**any**((**select** array\_agg(**value**::**int**) **from** jsonb\_array\_elements(cfg->'Wells'))::**int**[])

**LOOP**

**INSERT** **INTO** dcdm\_well(name, latitude, longitude, crdate)

**VALUES** (v\_r.name, v\_r.latitude, v\_r.longitude,**case** **when** (cfg*#>'{Base,Well}')::bool=true then v\_r.crdate else current\_timestamp end )*

               returning id **INTO** v\_new\_id;

**UPDATE** prj\_dcdm\_well **SET** newid=v\_new\_id **where** id=v\_r.id;

**END** **LOOP**;

**end** **if**;

*-- Оперативные справочники ввода*

**if**(cfg*#>'{Base,OpRef}')::bool=true then*

**case** (cfg*#>'{Base,OpRefType}')::int*

**when** 1 **then** *-- Заменить*

*-- Удаляем старое*

**delete** **from** dcdm\_referenceslist **where** iswelldep=true;

*-- Переносим данные из проекта*

**FOR** v\_r IN **select** \* **from** prj\_dcdm\_referenceslist **where** iswelldep=true and id=**any**((**select** array\_agg(**value**::**int**) **from** jsonb\_array\_elements(cfg->'OpRefs'))::**int**[])

**LOOP**

**INSERT** **INTO** dcdm\_referenceslist(sysname, name, iswelldep, browsedata) **VALUES** (v\_r.sysname, v\_r.name, v\_r.iswelldep, v\_r.browsedata) returning id **INTO** v\_new\_id;

**INSERT** **INTO** dcdm\_referencecol(reflistid, **type**, sysname, colname, initexpr) **select** v\_new\_id, **type**, sysname, colname, initexpr **from** prj\_dcdm\_referencecol **where** reflistid=v\_r.id;

*-- Скопируем также данные если требуется*

**if**(cfg*#>'{Base,OpRefData}')::bool=true then  --and EXISTS(SELECT 1 FROM information\_schema.tables WHERE table\_schema='public' AND table\_name='prj\_dcdm\_referencedata')*

**INSERT** **INTO** dcdm\_referencedata(wellid, reflistid, subid, grpname, val)

**select** prj\_dcdm\_well.newid,v\_new\_id,prj\_dcdm\_referencedata.subid,prj\_dcdm\_referencedata.grpname,prj\_dcdm\_referencedata.val

**from** prj\_dcdm\_referencedata

**inner** **join** prj\_dcdm\_well **on** prj\_dcdm\_referencedata.wellid=prj\_dcdm\_well.id

**where** prj\_dcdm\_referencedata.reflistid=v\_r.id and prj\_dcdm\_referencedata.wellid=**any**((**select** array\_agg(**value**::**int**) **from** jsonb\_array\_elements(cfg->'Wells'))::**int**[]);

**end** **if**;

**END** **LOOP**;

**when** 2 **then** *-- Совместить*

*-- Удалим пересекающиеся данные*

**delete** **from** dcdm\_referenceslist

**using** prj\_dcdm\_referenceslist

**where** dcdm\_referenceslist.sysname=prj\_dcdm\_referenceslist.sysname and dcdm\_referenceslist.iswelldep=true and prj\_dcdm\_referenceslist.iswelldep=true

                       and prj\_dcdm\_referenceslist.id=**any**((**select** array\_agg(**value**::**int**) **from** jsonb\_array\_elements(cfg->'OpRefs'))::**int**[]);

*-- Переносим данные из проекта*

*-- Переносим данные из проекта*

**FOR** v\_r IN **select** \* **from** prj\_dcdm\_referenceslist **where** iswelldep=true and id=**any**((**select** array\_agg(**value**::**int**) **from** jsonb\_array\_elements(cfg->'OpRefs'))::**int**[])

**LOOP**

**INSERT** **INTO** dcdm\_referenceslist(sysname, name, iswelldep, browsedata) **VALUES** (v\_r.sysname, v\_r.name, v\_r.iswelldep, v\_r.browsedata) returning id **INTO** v\_new\_id;

**INSERT** **INTO** dcdm\_referencecol(reflistid, **type**, sysname, colname, initexpr) **select** v\_new\_id, **type**, sysname, colname, initexpr **from** prj\_dcdm\_referencecol **where** reflistid=v\_r.id;

*-- Скопируем также данные если требуется*

**if**(cfg*#>'{Base,OpRefData}')::bool=true then  --and EXISTS(SELECT 1 FROM information\_schema.tables WHERE table\_schema='public' AND table\_name='prj\_dcdm\_referencedata')*

**INSERT** **INTO** dcdm\_referencedata(wellid, reflistid, subid, grpname, val)

**select** prj\_dcdm\_well.newid,v\_new\_id,prj\_dcdm\_referencedata.subid,prj\_dcdm\_referencedata.grpname,prj\_dcdm\_referencedata.val

**from** prj\_dcdm\_referencedata

**inner** **join** prj\_dcdm\_well **on** prj\_dcdm\_referencedata.wellid=prj\_dcdm\_well.id

**where** prj\_dcdm\_referencedata.reflistid=v\_r.id and prj\_dcdm\_referencedata.wellid=**any**((**select** array\_agg(**value**::**int**) **from** jsonb\_array\_elements(cfg->'Wells'))::**int**[]);

**end** **if**;

**END** **LOOP**;

**when** 3 **then** *-- Добавить*

*--    Переносим данные которые отсутствуют в основной таблице*

*-- Переносим данные из проекта*

**FOR** v\_r IN **select** prj\_dcdm\_referenceslist.\*

**from** prj\_dcdm\_referenceslist

**LEFT** **JOIN** dcdm\_referenceslist **ON** prj\_dcdm\_referenceslist.sysname=dcdm\_referenceslist.sysname and prj\_dcdm\_referenceslist.iswelldep=dcdm\_referenceslist.iswelldep

**where** dcdm\_referenceslist.id is null and prj\_dcdm\_referenceslist.iswelldep=true and prj\_dcdm\_referenceslist.id=**any**((**select** array\_agg(**value**::**int**) **from** jsonb\_array\_elements(cfg->'OpRefs'))::**int**[])

**LOOP**

**INSERT** **INTO** dcdm\_referenceslist(sysname, name, iswelldep, browsedata) **VALUES** (v\_r.sysname, v\_r.name, v\_r.iswelldep, v\_r.browsedata) returning id **INTO** v\_new\_id;

**INSERT** **INTO** dcdm\_referencecol(reflistid, **type**, sysname, colname, initexpr) **select** v\_new\_id, **type**, sysname, colname, initexpr **from** prj\_dcdm\_referencecol **where** reflistid=v\_r.id;

*-- Скопируем также данные если требуется*

**if**(cfg*#>'{Base,OpRefData}')::bool=true then  --and EXISTS(SELECT 1 FROM information\_schema.tables WHERE table\_schema='public' AND table\_name='prj\_dcdm\_referencedata')*

**INSERT** **INTO** dcdm\_referencedata(wellid, reflistid, subid, grpname, val)

**select** prj\_dcdm\_well.newid,v\_new\_id,prj\_dcdm\_referencedata.subid,prj\_dcdm\_referencedata.grpname,prj\_dcdm\_referencedata.val

**from** prj\_dcdm\_referencedata

**inner** **join** prj\_dcdm\_well **on** prj\_dcdm\_referencedata.wellid=prj\_dcdm\_well.id

**where** prj\_dcdm\_referencedata.reflistid=v\_r.id and prj\_dcdm\_referencedata.wellid=**any**((**select** array\_agg(**value**::**int**) **from** jsonb\_array\_elements(cfg->'Wells'))::**int**[]);

**end** **if**;

**END** **LOOP**;

**end** **case**;

**end** **if**;

*-- Журнал скважины (только добавление)*

**if**(cfg*#>'{Base,Well}')::bool=true then*

**if** **EXISTS**(**SELECT** 1 **FROM** information\_schema.**tables** **WHERE** table\_schema='public' AND table\_name='prj\_dcdm\_welllog') **then**

**INSERT** **INTO** dcdm\_welllog(wellid, **mode**, "time", delaymsec, depth, vdepth, paramname, val)

**SELECT** prj\_dcdm\_well.newid, prj\_dcdm\_welllog.**mode**, prj\_dcdm\_welllog.**time**, prj\_dcdm\_welllog.delaymsec,

                      prj\_dcdm\_welllog.depth, prj\_dcdm\_welllog.vdepth, prj\_dcdm\_welllog.paramname, prj\_dcdm\_welllog.val

**FROM** prj\_dcdm\_welllog **inner** **join** prj\_dcdm\_well **on** prj\_dcdm\_welllog.wellid=prj\_dcdm\_well.id

**where** (prj\_dcdm\_welllog.isnew is null or prj\_dcdm\_welllog.isnew=false)

                     and prj\_dcdm\_welllog.wellid=**any**((**select** array\_agg(**value**::**int**) **from** jsonb\_array\_elements(cfg->'Wells'))::**int**[]);

**end** **if**;

**if** **EXISTS**(**SELECT** 1 **FROM** information\_schema.**tables** **WHERE** table\_schema='public' AND table\_name='prj\_dcdm\_markers') **then**

**INSERT** **INTO** dcdm\_markers(wellid, createtime, changetime, **type**, depth, vdepth, "time", code)

**select** prj\_dcdm\_well.newid, prj\_dcdm\_markers.createtime, prj\_dcdm\_markers.changetime, prj\_dcdm\_markers.**type**,

                      prj\_dcdm\_markers.depth, prj\_dcdm\_markers.vdepth, prj\_dcdm\_markers.**time**, prj\_dcdm\_markers.code

**FROM** prj\_dcdm\_markers **inner** **join** prj\_dcdm\_well **on** prj\_dcdm\_markers.wellid=prj\_dcdm\_well.id

**where** prj\_dcdm\_markers.wellid=**any**((**select** array\_agg(**value**::**int**) **from** jsonb\_array\_elements(cfg->'Wells'))::**int**[]);

**end** **if**;

**if** **EXISTS**(**SELECT** 1 **FROM** information\_schema.**tables** **WHERE** table\_schema='public' AND table\_name='prj\_dcdm\_commentsandfiles') **then**

**INSERT** **INTO** dcdm\_commentsandfiles(wellid, createtime, changetime, **type**, isindexed, snapdepth, snapvdepth, snaptime, **comment**, orgfilename, filedata)

**select** prj\_dcdm\_well.newid, prj\_dcdm\_commentsandfiles.createtime, prj\_dcdm\_commentsandfiles.changetime, prj\_dcdm\_commentsandfiles.**type**,

                      prj\_dcdm\_commentsandfiles.isindexed, prj\_dcdm\_commentsandfiles.snapdepth, prj\_dcdm\_commentsandfiles.snapvdepth,

                          prj\_dcdm\_commentsandfiles.snaptime, prj\_dcdm\_commentsandfiles.**comment**, prj\_dcdm\_commentsandfiles.orgfilename, prj\_dcdm\_commentsandfiles.filedata

**from** prj\_dcdm\_commentsandfiles **inner** **join** prj\_dcdm\_well **on** prj\_dcdm\_commentsandfiles.wellid=prj\_dcdm\_well.id

**where** prj\_dcdm\_commentsandfiles.wellid=**any**((**select** array\_agg(**value**::**int**) **from** jsonb\_array\_elements(cfg->'Wells'))::**int**[]);

**end** **if**;

**end** **if**;

*-- Скрипты БД*

**if**(cfg*#>'{Base,Scripts}')::bool=true then*

**case** (cfg*#>'{Base,OpScriptsType}')::int*

**when** 1 **then** *-- Заменить*

*-- Удаляем старое*

**delete** **from** dcdm\_dbmuserscripts;

*-- Переносим данные из проекта*

**FOR** v\_r IN **select** \* **from** prj\_dcdm\_dbmuserscripts **where** ispublic=true and id=**any**((**select** array\_agg(trim('"' **FROM** **value**::**text**)::uuid) **from** jsonb\_array\_elements(cfg->'Scripts'))::uuid[])

**LOOP**

**INSERT** **INTO** dcdm\_dbmuserscripts(id, userid, changetime, name, script, ispublic)   **VALUES** (v\_r.id,i\_userid,v\_r.changetime,v\_r.name,v\_r.script,v\_r.ispublic);

**END** **LOOP**;

**when** 2 **then** *-- Совместить*

*-- Удалим пересекающиеся данные*

**delete** **from** dcdm\_dbmuserscripts **where** dcdm\_dbmuserscripts.id=**any**((**select** array\_agg(trim('"' **FROM** **value**::**text**)::uuid) **from** jsonb\_array\_elements(cfg->'Scripts'))::uuid[]);

*-- Переносим данные из проекта*

**FOR** v\_r IN **select** \* **from** prj\_dcdm\_dbmuserscripts **where** ispublic=true and id=**any**((**select** array\_agg(trim('"' **FROM** **value**::**text**)::uuid) **from** jsonb\_array\_elements(cfg->'Scripts'))::uuid[])

**LOOP**

**INSERT** **INTO** dcdm\_dbmuserscripts(id, userid, changetime, name, script, ispublic)   **VALUES** (v\_r.id,i\_userid,v\_r.changetime,v\_r.name,v\_r.script,v\_r.ispublic);

**END** **LOOP**;

**when** 3 **then** *-- Добавить*

*--    Переносим данные которые отсутствуют в основной таблице*

**FOR** v\_r IN **select** prj\_dcdm\_dbmuserscripts.\*

**from** prj\_dcdm\_dbmuserscripts

**LEFT** **JOIN** dcdm\_dbmuserscripts **ON** prj\_dcdm\_dbmuserscripts.id=dcdm\_dbmuserscripts.id

**where** dcdm\_dbmuserscripts.id is null and (prj\_dcdm\_dbmuserscripts.id=**any**((**select** array\_agg(trim('"' **FROM** **value**::**text**)::uuid) **from** jsonb\_array\_elements(cfg->'Scripts'))::uuid[]))

**LOOP**

**INSERT** **INTO** dcdm\_dbmuserscripts(id, userid, changetime, name, script, ispublic)   **VALUES** (v\_r.id,i\_userid,v\_r.changetime,v\_r.name,v\_r.script,v\_r.ispublic);

**END** **LOOP**;

**end** **case**;

**end** **if**;

**else**

   RAISE EXCEPTION 'Project id not found' **USING** HINT='DCDM', DETAIL='PrjIdNotFound';

**end** **if**;

**return** cfg::json;

**END**

$$;

**GRANT** **EXECUTE** **ON** **FUNCTION**  prj\_open\_final(**varchar**,jsonb,dcdm\_users.id%**TYPE**) **TO** dcdmsrv;

*--#############################################################################################*

*--#############################################################################################*

*--##############################################################################################*

*--##############################################################################################*

*--##############################################################################################*

*--##############################################################################################*

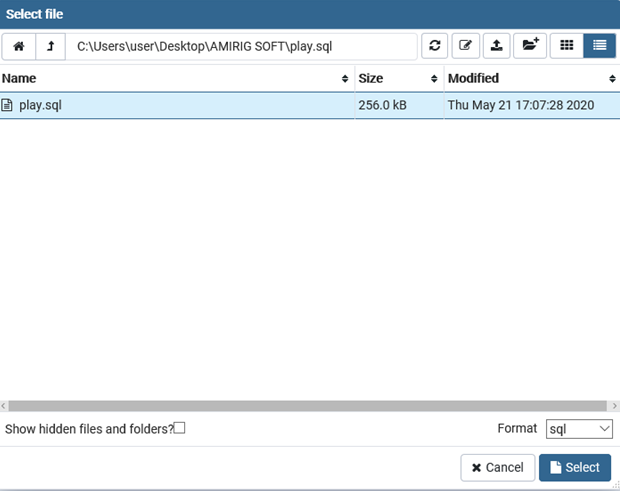
*--##############################################################################################*

*--##############################################################################################*

*--##############################################################################################*

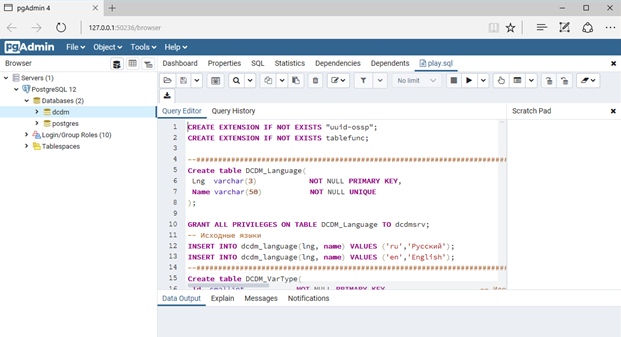
**GRANT** **USAGE**, **SELECT** **ON** **ALL** SEQUENCES IN **SCHEMA** **public** **TO** dcdmsrv;

Далее откроется окно выбора файла:



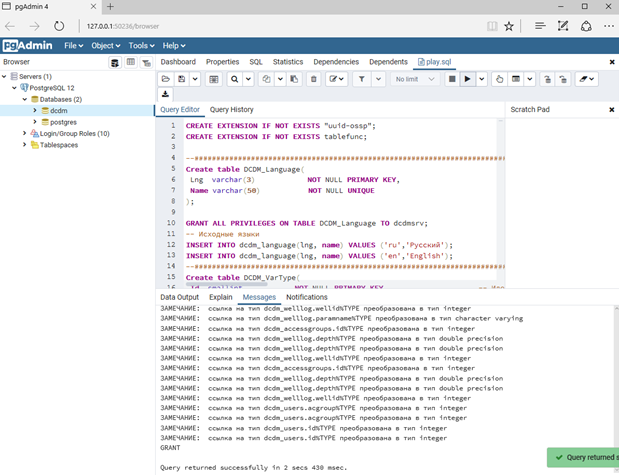
Укажите путь к файлу play.sql, выберите этот файл и нажмите **Select** (Выбрать).

Далее откроется окно, в котором будет видно, что содержимое скрипта наполнение базы заполнилось.



Для выполнения скрипта нажмите Execute/Refresh (F5) (значек Play).

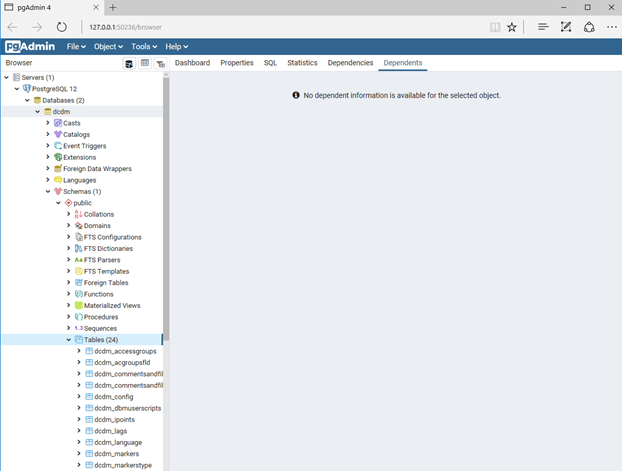
Во время выполнения скрипта вы увидите заполняющийся список системных сообщений:



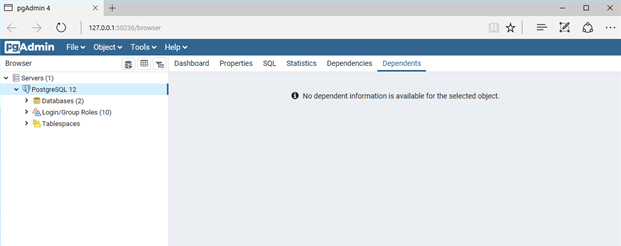
В случае успеха в конце вы увидите сообщение вида: Query returned successfully in 2 secs 430 msec. (пример).

Готово! Скрипт выполнен, база наполнена.

Дополнительно вы можете проверить структуру базы данных на наличие созданных таблиц.



Вернитесь в исходное состояние панели PgAdmin 4



Перейдите к Разделу 2.5. «Установка Системы DCDM».

* 1. **Настройка логической репликации Системы DCDM**

Установите параметры конфигурации в файле **postgresql.conf**

По умолчанию данный файл доступен по следующему пути: C:\Program Files\PostgreSQL\12\data\postgresql.conf

Запустите редактирование файла и найдите в нем параметр: #wal\_level = replica

Раскомментируйте строку, далее параметр replica измените на logical

В результате у вас получится строка, имеющая следующий вид: wal\_level = logical

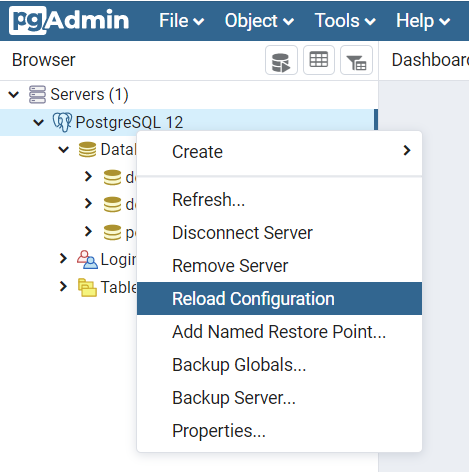
Другие необходимые параметры по умолчанию имеют значения, подходящие для базовой настройки.

Далее в файл **pg\_hba.conf** необходимо внести изменения, чтобы разрешить репликацию (конкретные значения будут зависеть от фактической конфигурации сети и имени пользователя, с которым вы будете подключаться):



По умолчанию данный файл доступен по следующему пути: C:\Program Files\PostgreSQL\12\data\pg\_hba.conf

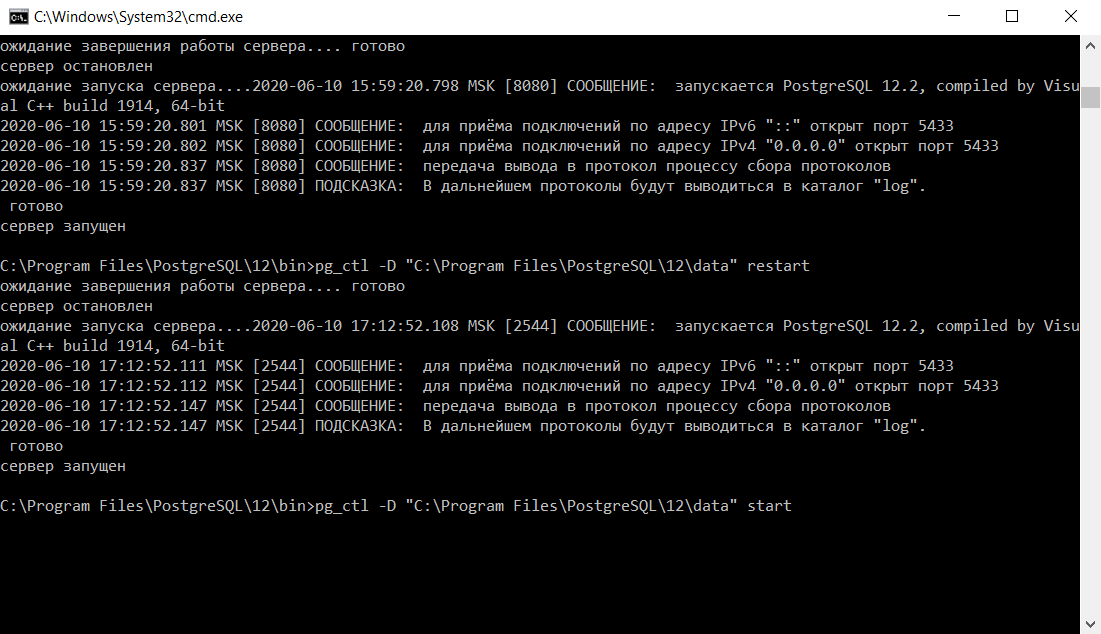
После внесения изменений потребуется выполнить Reload конфигурации PostgreSQL:



Также возможно потребуется перезапуск сервиса PostgreSQL (например с помощью службы Windows **services.msc**)

Также нужно выполнить в командной строке:

pg\_ctl -D "C:\Program Files\PostgreSQL\12\data" restart



Дополнительные команды:

pg\_ctl -D "C:\Program Files\PostgreSQL\12\data" start

pg\_ctl -D "C:\Program Files\PostgreSQL\12\data" stop

Затем в базе данных публикации выполните:

CREATE PUBLICATION mypub FOR TABLE dcdm\_users, dcdm\_vars, dcdm\_profiles, dcdm\_ipoints, dcdm\_well, dcdm\_referencecol, dcdm\_referenceslist,

dcdm\_referencedata, dcdm\_accessgroups, dcdm\_welllog, dcdm\_acgroupsfld, dcdm\_config, dcdm\_commentsandfilestype,

dcdm\_commentsandfiles,dcdm\_messages, dcdm\_dbmuserscripts, dcdm\_lags, dcdm\_vargroups, dcdm\_mlib, dcdm\_markerstype, dcdm\_markers;

И в базе данных подписчика (Внимание! Это пример, используйте ваши реальные данные):

CREATE SUBSCRIPTION mysub CONNECTION 'dbname=dcdm host=127.0.0.1 port=5432 user=repuser password=пароль от базы' PUBLICATION mypub;

Показанная выше команда запустит процесс репликации, который вначале синхронизирует исходное содержимого таблиц dcdm\_users, dcdm\_vars, dcdm\_profiles, dcdm\_ipoints, dcdm\_well, dcdm\_referencecol, dcdm\_referenceslist, dcdm\_referencedata, dcdm\_accessgroups, dcdm\_welllog, dcdm\_acgroupsfld, dcdm\_config, dcdm\_commentsandfilestype,

dcdm\_commentsandfiles,dcdm\_messages, dcdm\_dbmuserscripts, dcdm\_lags, dcdm\_vargroups, dcdm\_mlib, dcdm\_markerstype, dcdm\_markers, а затем начнёт перенос инкрементальных изменений в этих таблицах.

Проверка состояния подписки в базе данных подписчика:

SELECT \* FROM pg\_stat\_subscription

Обозначение полей в результатах выполнения запроса

**received\_lsn** - позиция в журнале, до которой получены изменения;

**latest\_end\_lsn** - позиция в журнале, подтвержденная процессу wal sender.

Получение списка всех процессов в базах данных

SELECT \* FROM pg\_stat\_activity;

Проверка статуса репликации

**В базе данных публикации:**



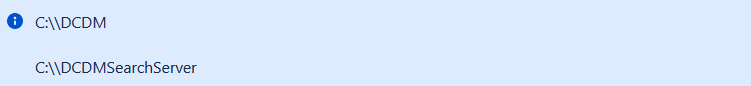
**В базе данных подписчика:**



* 1. **Установка Системы DCDM**

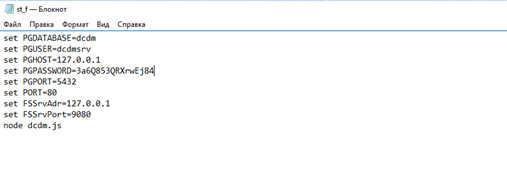
Распакуйте файлы из архива с Системой DCDM на диск C:\\

После распаковки, на диске C:\\ появятся папки:

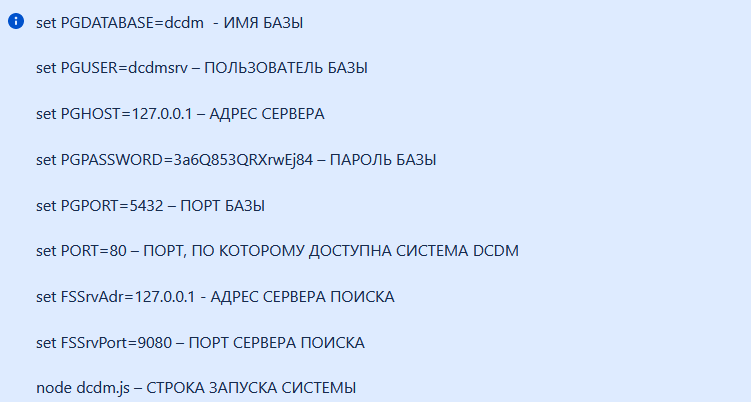


Далее перейдите к настройкам файла C:\DCDM\st\_f.bat

Нажмите на файл **st\_f.bat** правой кнопкой мыши и выберите пункт “Изменить”



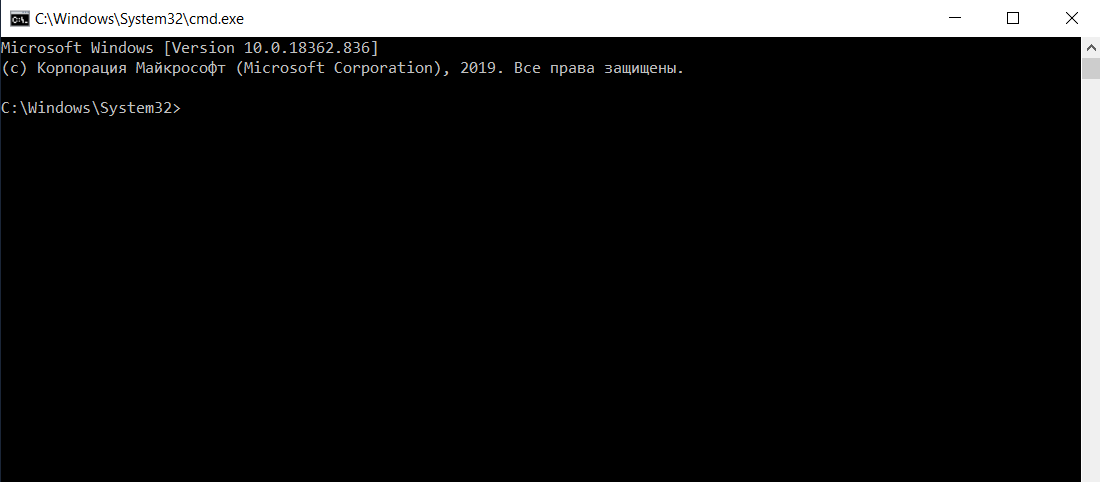
В данном файле необходимо задать следующие параметры вашего сервера:



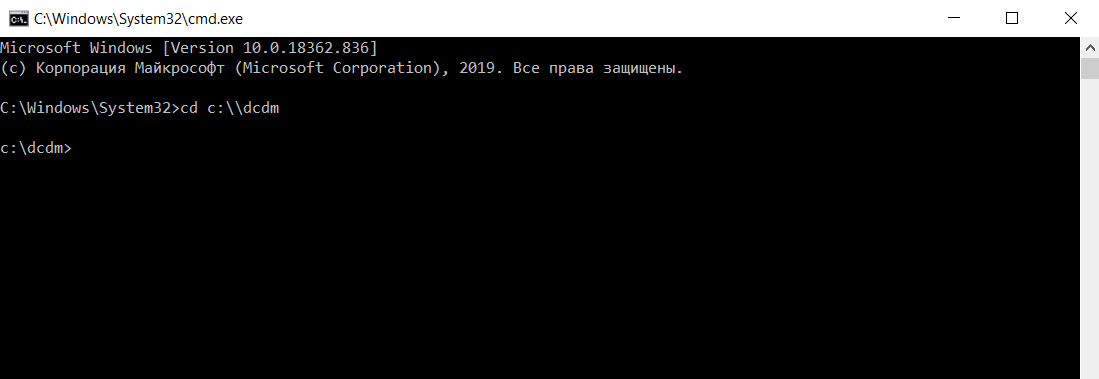
После установки параметров, сохраните изменения и закройте файл.

Далее перейдите к настройкам Node.js

Для выполнения настройки Node.js запустите командную строку Windows:







Далее введите команду:



Начнется установка необходимых модулей Node.js, дождитесь завершения установки.

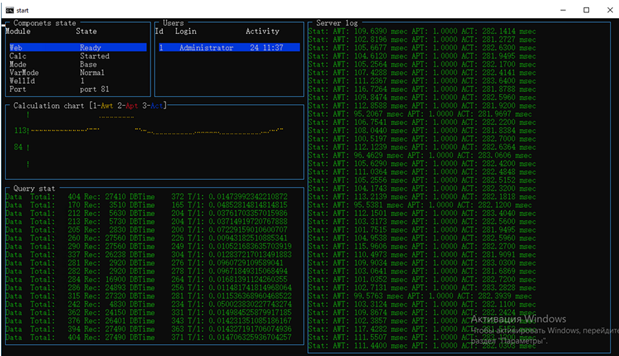
Далее, введите команду:



Начнется установка необходимых сервисов Node.js, дождитесь завершения установки.

Готово! Установка завершена.

Запустите Систему DCDM при помощи выполнения файла **st\_f.bat** из папки C:\\DCDM\



Запуск Системы DCDM выполнен успешно.

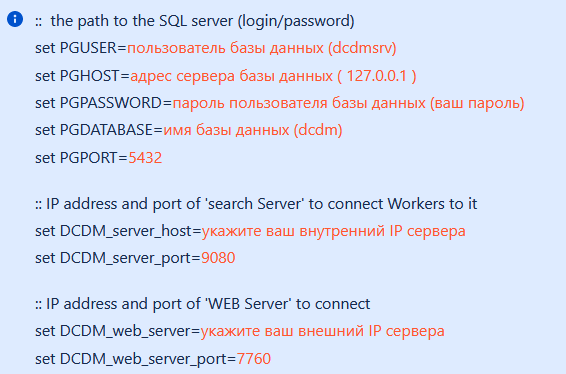
Далее, для настройки и запуска сервера системы поиска DCDM необходимо перейти к Разделу 2.6. «Развертывание системы поиска DCDM».

* 1. **Развертывание системы поиска DCDM**

Нажмите правой кнопкой мыши на файл RUN\_ME.bat и выберите изменить.

После распаковки архива программы, данный файл имеет путь: C:\\DCDMSearchServer\Project\Script\SS\RUN\_ME.bat

Отроется редактирование файла, необходимо найти в файле и настроить следующие параметры:



Запустите файл RUN\_ME.bat двойным кликом.

**Важно!** Запускать Сервер поиска необходимо из директории расположения **RUN\_ME.bat**. т.е файл **RUN\_ME.bat** должен находится в директории **SS** (в нашем примере - С:\DCDMSearchServer\Project\Script\SS) или указать путь, если вы запускаете RUN\_ME.bat из командной строки. В случае организации запуска через ярлык, обязательно указать путь к объекту (файлу RUN\_ME.bat и Рабочую папку (в нашем примере ("C:\DCDMSearchServer\Project\Script\SS").

Откроется окно программы сервера поиска DCDM.



**Важно!** Для полного запуска сервера требуется время, поэтому если сразу смотреть на появившееся окно, то в нем может не отображаться строка с **index worker**.

Готово! Запуск сервера поиска выполнен успешно.

1. **Вход в систему и начало работы**
   1. **Авторизация в Системе DCDM**

Откройте браузер, в адресной строке введите адрес сервера Системы DCDM.

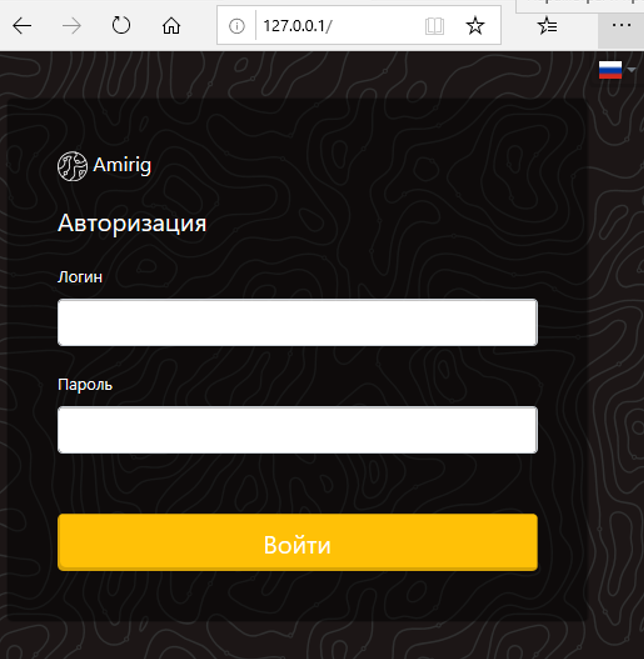
Если браузер запускается с сервера, на котором установлена Система DCDM, то панель администрирования по умолчанию будет доступна по следующим адресам: 127.0.0.1, localhost и по IP адресу сервера. Данный вид входа в панель администрирования рекомендуется только для отладки работы Системы DCDM, во всех остальных случаях вход выполняется исключительно из браузеров, установленных на компьютерах, выступающих в роли клиентов Системы DCDM.

Для доступа к серверу с компьютера, выступающего в роли клиента Системы DCDM, в адресную строку браузера необходимо указывать IP адрес сервера.

Внимание! Не пытайтесь использовать в качестве адреса, значения: 127.0.0.1 и localhost.

Чтобы узнать IP адрес сервера, запустите на сервере Системы DCDM системное приложение “Командная строка“ и выполните команду ipconfig /all

После ввода IP адреса сервера, нажмите “Enter“, откроется окно авторизации для ввода логина и пароля.

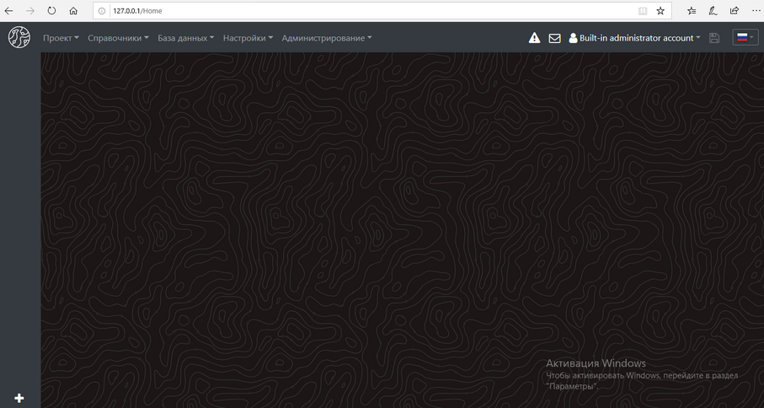


По умолчанию, сразу после установки Системы DCDM, единственным существующим пользователем “Администратором” в системе является пользователь со следующими данными:

Логин: **Administrator**  
Пароль: **admin**  
Для входа в систему введите логин и пароль “Администратора” и нажмите на кнопку “Войти“.

Рекомендуем изменить пароль “Администратора” на пароль, соответствующий политике безопасности вашей организации.

После успешного входа в систему откроется рабочее пространство панели администратора:



Все готово для начала работы.

Что делать если при попытке входа в систему, всплывает сообщение об ошибке авторизации?

1. Проверьте правильность ввода логина и пароля (пример: лишние пробелы, неверная раскладка клавиатуры, включенный CapsLock и прочее.)
2. Проверьте соединение с базой данных PostgreSQL (пример: состояние сервиса PostgreSQL, файлы конфигурации Системы DCDM и их соответствие с данными установленными в PostgreeSQL, пользователь базы данных, роли, пароль пользователя базы.)
3. Проверьте настройки антивирусного программного обеспечения, если оно установлено на сервер.