ООО «БФТ»
129085, г. Москва, ул. Годовикова, д. 9, стр. 17
+7 (495) 784-70-00
ineed@bftcom.com
bftcom.com

Утвержден
БАРМ.00004-55 32 01-6-ЛУ

Система автоматизации финансово-экономических органов – Автоматизированный Центр Контроля процесса планирования и анализа бюджета АЦК-Планирование

Блок администрирования
Подсистема администрирования
Установка и настройка сервера БД

Автоматизированное рабочее место финансового органа

Руководство администратора

БАРМ.00004-55 32 01-6

Листов 67

АННОТАЦИЯ

Приводится руководство администратора системы «АЦК-Планирование» по установке и настроке сервера БД.

Содержание документа соответствует ГОСТ 19.503-79 «Единая система программной документации. РУКОВОДСТВО СИСТЕМНОГО ПРОГРАММИСТА. Требования к содержанию и оформлению».

«Система автоматизации финансово-экономических органов – Автоматизированный Центр Контроля процесса планирования и анализа бюджета» («АЦК-Планирование») зарегистрирована в Федеральной службе по интеллектуальной собственности, патентам и товарным знакам, Свидетельство № 2008610923 от 21 февраля 2008 г.

ООО «БФТ» оставляет за собой право вносить изменения в программное обеспечение без внесения изменений в эксплуатационную документацию.

Оперативное внесение изменений в программное обеспечение отражается в сопроводительной документации к выпускаемой версии.

Документ соответствует версии системы «АЦК-Планирование» – 2.55.0.1. Последние изменения внесены 29.06.2022 г.

СОДЕРЖАНИЕ

1

Установка и настройка сервера БД

Для сохранения, модификации, обработки и удаления данных сервер приложений «АЦК-Планирование» использует СУБД. В настоящий момент поддерживается совместимость и оптимальная работа со следующими типами СУБД: Oracle, Firebird, PostgreSQL. Установка, настройка и администрирование сервера есть задача тривиальная и описана в большом количестве популярной технической литературы. Рассмотрим вопросы установки и администрирования на уровне достаточном для установки и запуска стенда «АЦК-Планирование».

1.1 Отличия СУБД Oracle, СУБД Firebird и СУБД PostgreSQL

Несмотря на то, что СУБД по своей идеологии должна выполнять стандартизированный набор функций, производители этих систем предпочитают наделять свои программные продукты особенностями, в той и иной степени улучшающими некоторые характеристики или упрощающие процесс ее эксплуатации.

К наиболее наглядным отличиям СУБД Oracle и СУБД PostgreSQL от СУБД Firebird следует отнести то, что в Firebird база данных представлена в виде отдельного файла, а в Oracle и PostgreSQL база данных состоит из нескольких файлов. СУБД Oracle и СУБД PostgreSQL, в отличие от Firebird, поддерживают схемы, которые, по сути, являются отдельными базами данных внутри данной базы.

СУБД PostgreSQL использует процедурный язык PL/pgSQL, но также предоставляет возможность использовать PL/Perl, PL/Python, PL/Tcl, как СУБД Oracle использует только PL/SQL.

1.2 Установка и настройка СУБД PostgreSQL на версии 9.5

СУБД PostgreSQL – это свободное и полностью открытое программное обеспечение, которое имеет традиционные возможности коммерческих СУБД с расширениями СУБД нового поколения. СУБД PostgreSQL может быть использована в среде ОС Windows и Linux. В процессе эксплуатации СУБД под разными платформами есть отличия только в процессе ее инсталляции.

Инсталляции СУБД PostgreSQL в среде ОС Windows

Установка СУБД PostgreSQL в среде ОС Windows состоит из нескольких этапов.

Шаг 1. Запустить файл **setup.exe**. Выводится окно установки: Нажать кнопку **Next**

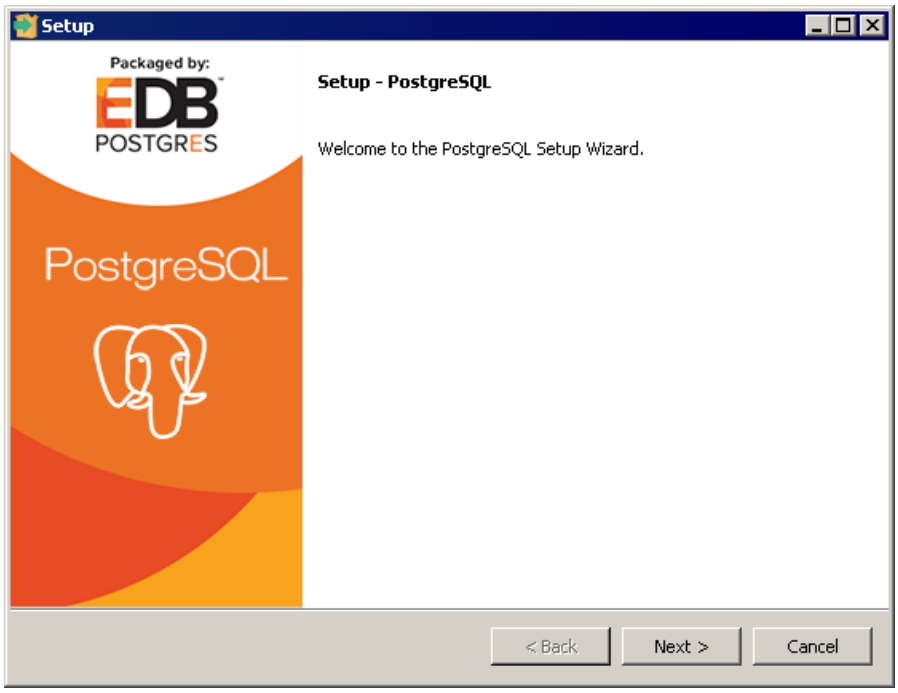

Рисунок 1 – Установка СУБД PostgreSQL. Шаг 1

Шаг 2. Необходимо выбрать папку установки. По умолчанию СУБД PostgreSQL устанавливается в каталог **c:\Program Files\PostgreSQL\9.5**.

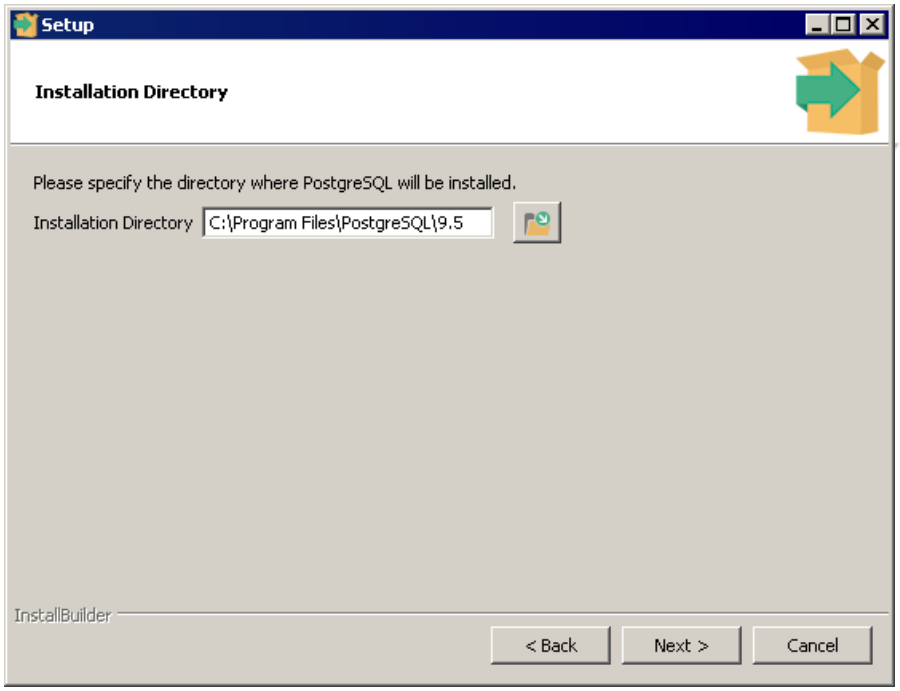

Рисунок 2 – Установка СУБД PostgreSQL. Шаг 2

Шаг 3. Выбрать расположение каталога баз данных. Здесь будет находиться хранимая в СУБД информация. По умолчанию устанавливается каталог **c:\Program Files\PostgreSQL\9.5\data\.**

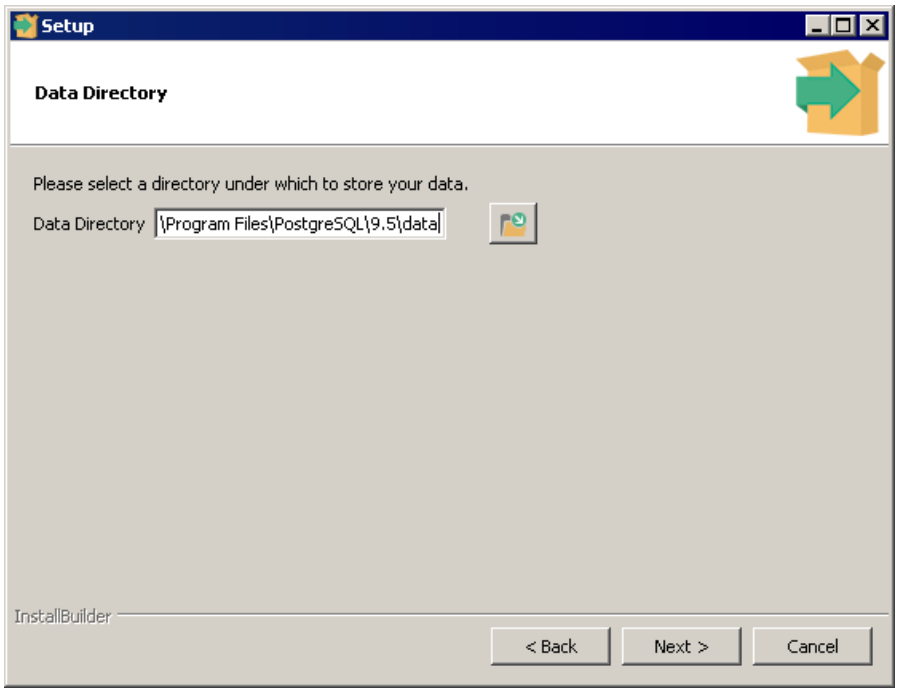

Рисунок 3 – Установка СУБД PostgreSQL. Шаг 3

Шаг 4. Необходимо ввести и подтвердить повторным вводом пароль пользователя для СУБД PostgreSQL.

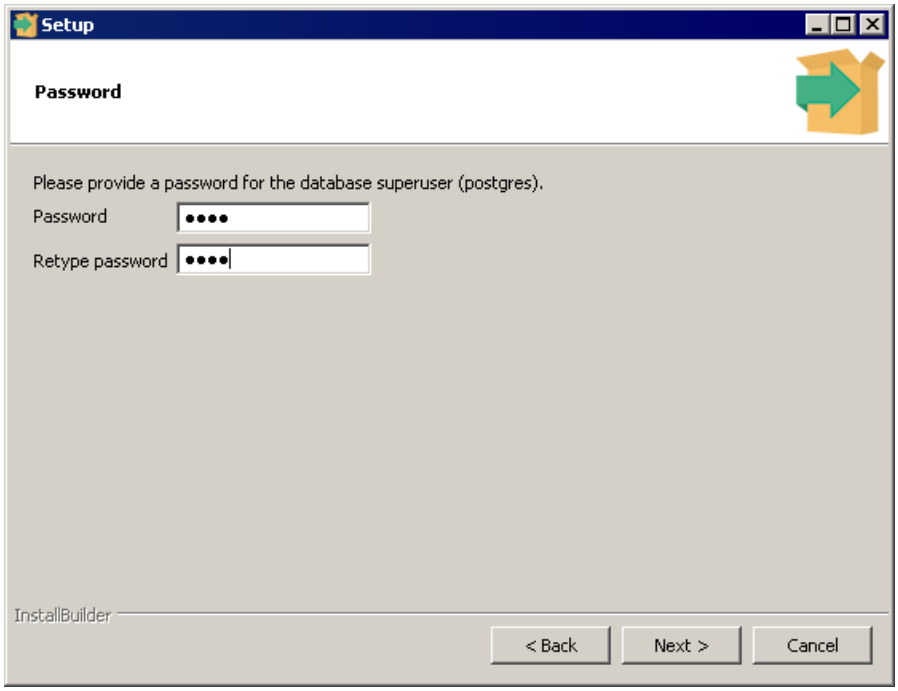

Рисунок 4 – Установка СУБД PostgreSQL. Шаг 4

Шаг 5. Порт оставить по умолчанию.

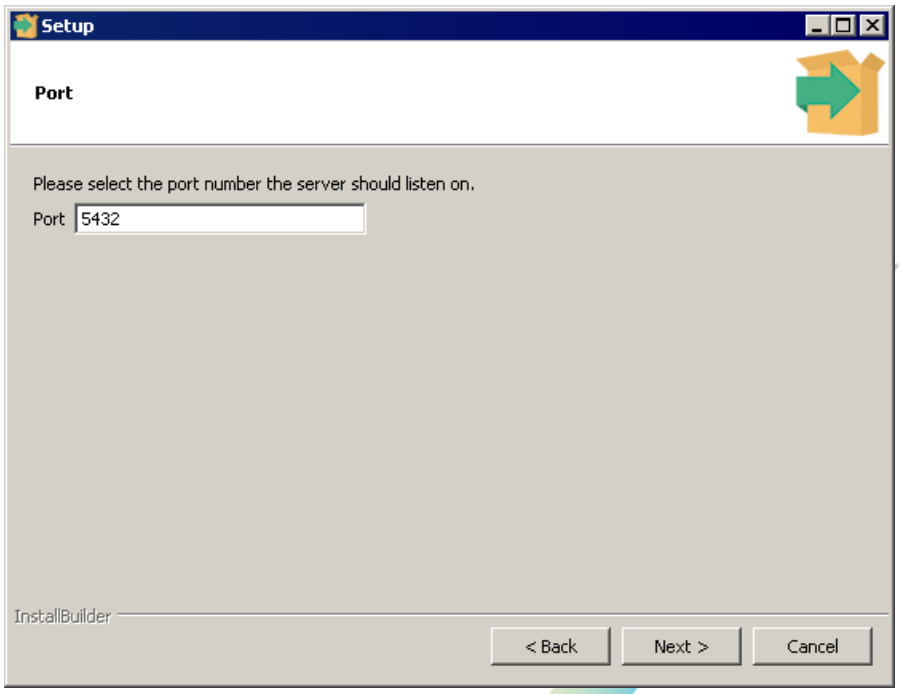

Рисунок 5 – Установка СУБД PostgreSQL. Шаг 5

Шаг 6. Необходимо выбрать локализацию «Russian, Russia», чтобы данные хранились на русском языке.

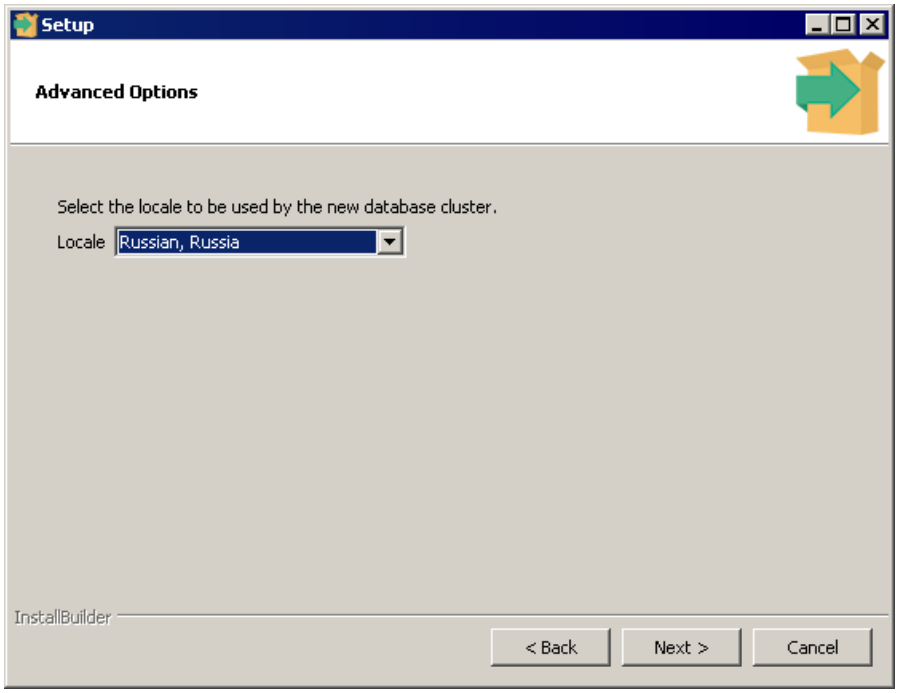

Рисунок 6 – Установка СУБД PostgreSQL. Шаг 6

Шаг 7. Выводится окно предупреждения об установке СУБД PostgreSQL на компьютер. После нажатия кнопки **Next** идет ход установки СУБД PostgreSQL.

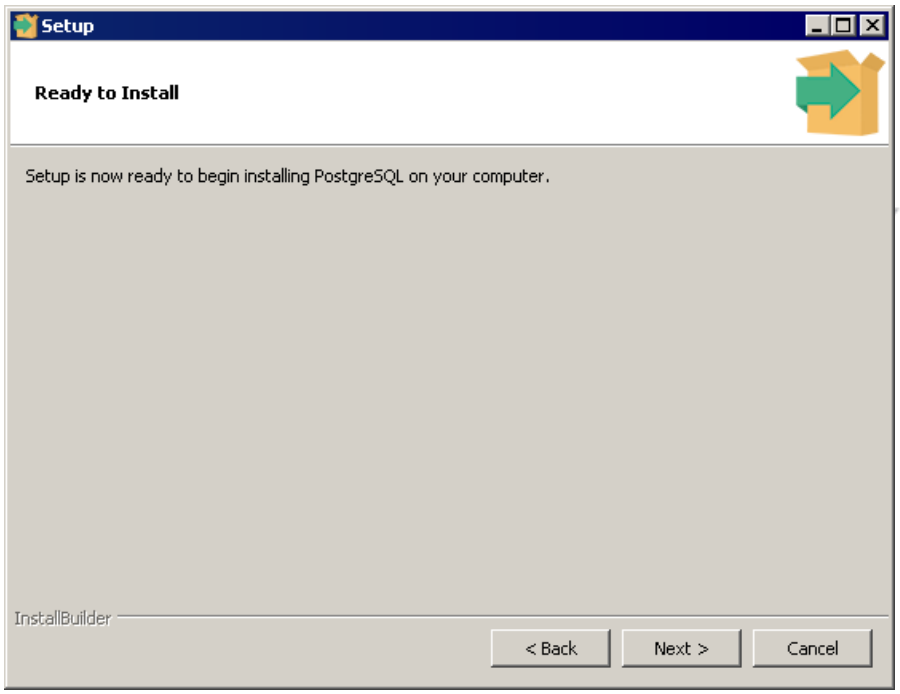

Рисунок 7 – Установка СУБД PostgreSQL. Шаг 7

Шаг 8. При завершении установки предлагается загрузить дополнительные инструменты, драйвера и приложения для СУБД PostgreSQL. Для завершения установки нажать кнопку **Finish**.

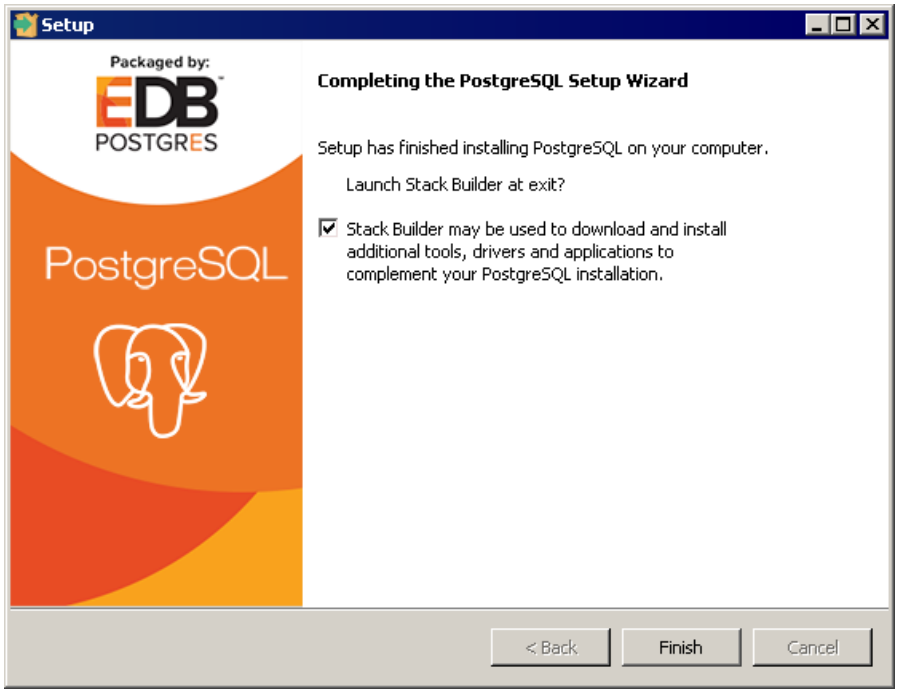

Рисунок 8 – Установка СУБД PostgreSQL. Шаг 8

Инсталляции СУБД PostgreSQL в среде ОС Linux

Для установки необходимо подключить пакетный репозиторий.

Для ОС Debian выполняем в терминале следующие команды:

```
$ sudo apt-get install lsb-release
$ sudo sh -c 'echo "deb \
http://repo.postgrespro.ru/pgpro-9.5/debian \
$(lsb_release -cs) main" > \
/etc/apt/sources.list.d/posgrespro.list'
```

Для ОС Ubuntu команды выглядят так:

```
$ sudo sh -c 'echo "deb \
http://repo.postgrespro.ru/pgpro-9.5/ubuntu \
$(lsb_release -cs) main" > \
/etc/apt/sources.list.d/posgrespro.list'
```

Дальше все одинаково для обеих систем:

```
$ wget --quiet -O - http://repo.postgrespro.ru/pgpro\
-9.5/keys/GPG-KEY-POSTGRESPRO-95 | sudo apt-key add -
$ sudo apt-get update
```

Перед установкой необходимо проверить настройки локализации:

```
$ locale
```

Для хранения данных на русском языке нужно изменить значение переменных *LC_CTYPE* и *LC_COLLATE* на *ru_RU.UTF8* . При необходимости можно установить эти переменные.

В операционной системе должна быть установлена соответствующая локаль:

```
$ locale -a | grep ru_RU
ru_RU.utf8
```

Если в операционной системе не установлена соответствующая локаль, то сгенерируем ее:

```
$ sudo locale-gen ru_RU.utf8
```

Теперь необходимо установить СУБД PostgreSQL:

```
$ sudo apt-get install postgrespro-9.5
```

СУБД PostgreSQL установлена, запущена и готова к работе.

При установке PostgreSQL автоматически создается специальный пользователь *postgres*, под именем которого работают процессы, обслуживающие сервер, и которому принадлежат все файлы, относящиеся к СУБД. PostgreSQL будет автоматически запускаться при перезагрузке операционной системы.

Для подключения к серверу СУБД используются программа **pgAdmin III** или терминальный клиент **SQL Shell (psql)**.

В Windows программа **pgAdmin III** запускается из папки меню **Пуск**.

***Примечание**: В Linux нужно установить пакет **pgadmin3**:*

```
$ sudo apt-get install pgadmin3
```

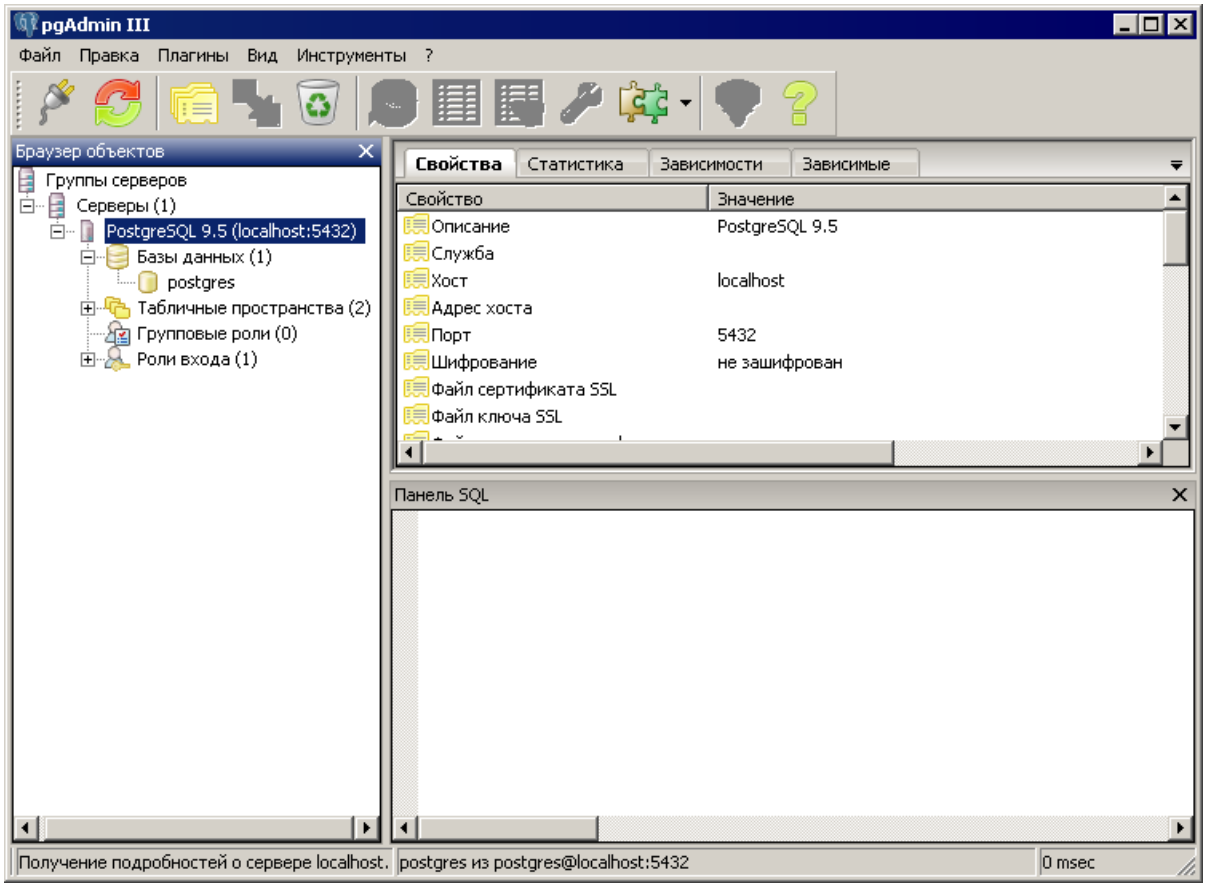

Рисунок 9 – Интерфейс pgAdmin III

Настройка подключения к серверу осуществляется через меню **Файл→Добавить сервер.**

В левой части основного окна находится навигатор объектов. В правой части окна выводится справочная информация. Внизу правой части отображается команда SQL, с помощью которой можно создать объект.

В Windows программа **SQL Shell (psql)** запускается из папки меню **Пуск**.

***Примечание**: В системе Linux запуск **psql** выполняется через команду:*

$ sudo -u postgres psql.

В терминале часто происходит ошибка неправильного отображения русских букв. Для устранения ошибки нужно сделать два пункта:

1. Закрыть окно терминала и отредактировать файл **SQL Shell (psql)** – добавить в него первой строкой команду:

```
chcp 1251
```

2. Снова запустить psql и в свойствах окна изменить шрифт с растрового на TrueType (обычно «Lucida Console»).

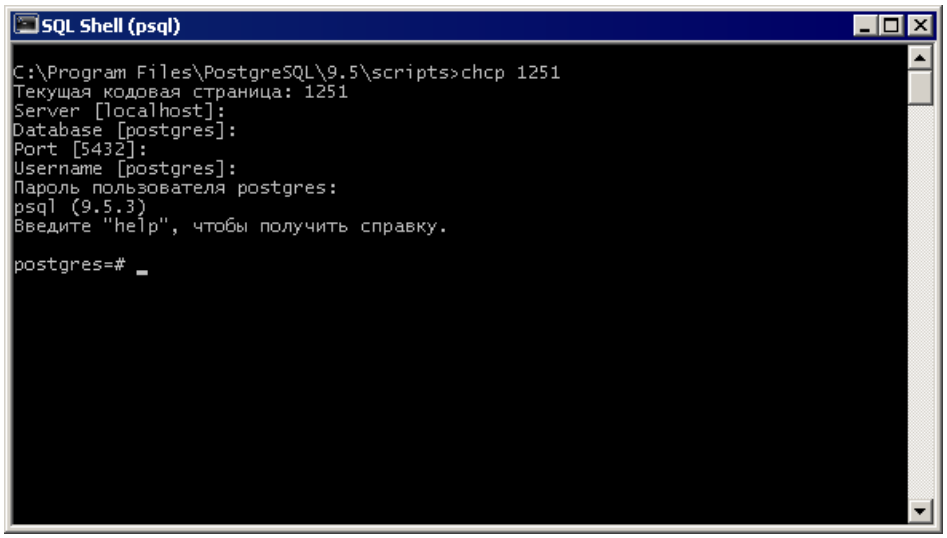

Рисунок 10 – Терминал SQL Shell (psql)

В Windows и Linux выводится в терминале одинаковое приглашение:

```
postgres=#
```

где *postgres* – имя базы данных, которая подключена в данный момент.

Для создания новой базы данных в программе **SQL Shell (psql)** нужно выполнить командную строку:

```
postgres=# CREATE DATABASE {имя_базы};
CREATE DATABASE
```

1.3 Установка и настройка СУБД Oracle версии 11.2

В этом разделе рассмотрим установку СУБД Oracle версии 11.2.0.3 на серверный компьютер, запущенный под управлением Oracle Enterprise Linux 4/5, RH Linux AS 4/5, SUSe Linux 9/10/11.

Перед началом установки нужно убедиться в наличии библиотек. Для Oracle Linux 5/Red Hat Enterprise Linux 5 должны быть установлены следующие библиотеки: (все действия выполняются пользователем root):

- binutils-2.17.50.0.6
- compat-libstdc++-33-3.2.3
- compat-libstdc++-33-3.2.3 (32 bit)
- elfutils-libelf-0.125
- elfutils-libelf-devel-0.125
- gcc-4.1.2
- gcc-c++-4.1.2
- glibc-2.5-24
- glibc-2.5-24 (32 bit)
- glibc-common-2.5
- glibc-devel-2.5
- glibc-devel-2.5 (32 bit)
- glibc-headers-2.5
- ksh-20060214
- libaio-0.3.106
- libaio-0.3.106 (32 bit)
- libaio-devel-0.3.106
- libaio-devel-0.3.106 (32 bit)
- libgcc-4.1.2
- libgcc-4.1.2 (32 bit)
- libstdc++-4.1.2
- libstdc++-4.1.2 (32 bit)
- libstdc++-devel 4.1.2
- make-3.81
- sysstat-7.0.2

Проверить их наличие можно с помощью команды, при этом выводится полный список всех установленных в системе пакетов:

```
#rpm -q -a
```

Список, выводимый на экран, можно ограничить, применив фильтрацию. Для этого следует использовать следующую комбинацию команд:

```
#rpm -q -a | grep {имя библиотеки}
```

Если необходимые библиотеки отсутствуют, то их можно установить с помощью команды:

```
#rpm -ivh {имя библиотеки}
```

Важно! *Настройку базы данных можно осуществить автоматически с помощью следующих пакетов:*

- ***oracle-validated.rpm*** *для версии Oracle Linux ниже 5x;*
- ***oracle-rdbms-server-11gR2-preinstall*** *для версии Oracle Linux ниже 6x.*

Перед установкой СУБД Oracle должна быть установлена виртуальная машина Java (JRE) и создана символическая ссылка на каталог /usr/local/java. Например, следующей командой:

```
#ln -s /usr/local/jdk1.8.0 /usr/local/java
```

Создаем группу «dba», которая будет давать привилегии для пользователя SYSDBA для СУБД Oracle.

```
#groupadd dba
```

Создаем группу «oinstall», которая будет владельцем установочных файлов.

```
#groupadd oinstall
```

Создаем пользователя oracle, под которым будет выполняться установка СУБД Oracle, и изменим ему пароль:

```
#useradd -c "Oracle software owner" -g oinstall -G dba oracle
#passwd oracle
```

Создаем директории для установки Oracle:

```
#mkdir /opt/oracle
#mkdir /opt/oracle/product
#mkdir /opt/oracle/product/9.2.0
#mkdir /var/opt/oracle
```

Меняем владельца директории для установки Oracle:

```
#chown -R oracle.oinstall /opt/oracle
#chown oracle.dba /var/opt/oracle
```

Изменяем права доступа на директорию:

```
#chmod 755 /var/opt/oracle
```

Установим переменные окружения для пользователя oracle, для чего добавим в файл **/home/oracle/.bash_profile** следующие строки:

```
# Get the aliases and functions
if [ -f ~/.bashrc ]; then
source . ~/.bashrc
fi

# User specific environment and startup programs

PATH=$PATH:$HOME/bin

export PATH
export ORACLE_BASE=/opt/oracle
export ORACLE_HOME=$ORACLE_BASE/product/11.2.0
export ORACLE_SID=azk11
export NLS_LANG=AMERICAN_AMERICA.CL8MSWIN1251
export
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$ORACLE_HOME/lib:/home/oracle/lib:/usr/lib:/usr/local/
lib
export PATH=$PATH:$ORACLE_HOME/bin:/usr/bin:/sbin:/bin:/usr/local/bin
export LIBPATH=$ORACLE_HOME/lib
export THREADS_FLAG=native
umask 022
```

Необходимо разрешить доступ к графическому ядру с помощью утилиты *xhost*:

```
xhost +
```

Далее установка СУБД Oracle, выполняется под пользователем oracle в графической оболочке. Для чего запускаем файл **./runInstaller**.

Рассмотрим этапы работы визуальной программы инсталляции. В ходе инсталляции будет задано некоторое количество стандартных вопросов:

Шаг 1. Configure Security Updates: нажимается кнопка **Следующий**, при появлении предупреждения нажимается кнопка **Yes**.

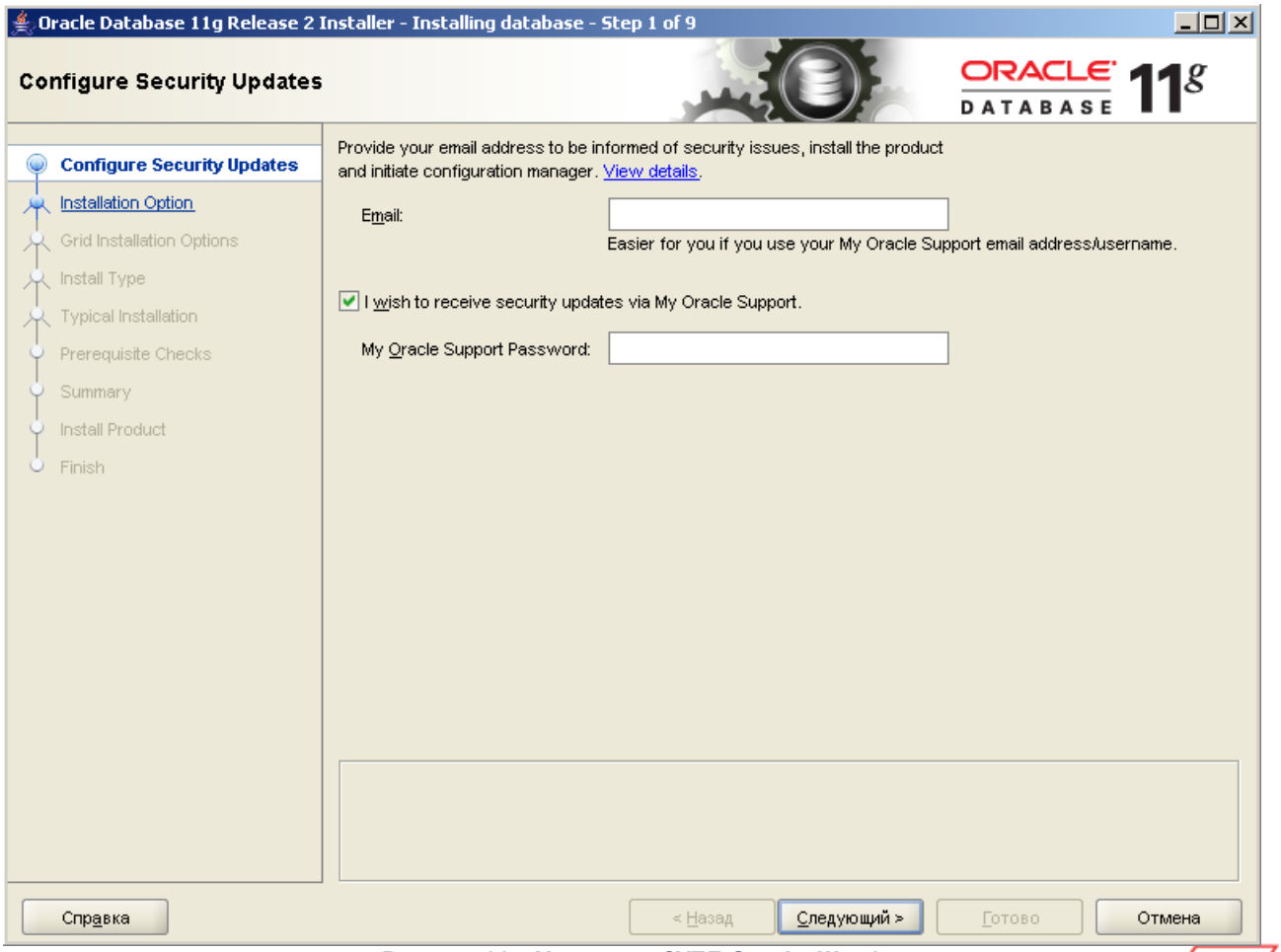

Рисунок 11 – Установка СУБД Oracle. Шаг 1

Шаг 2. Installation Option: выбирается *Install database software only*.

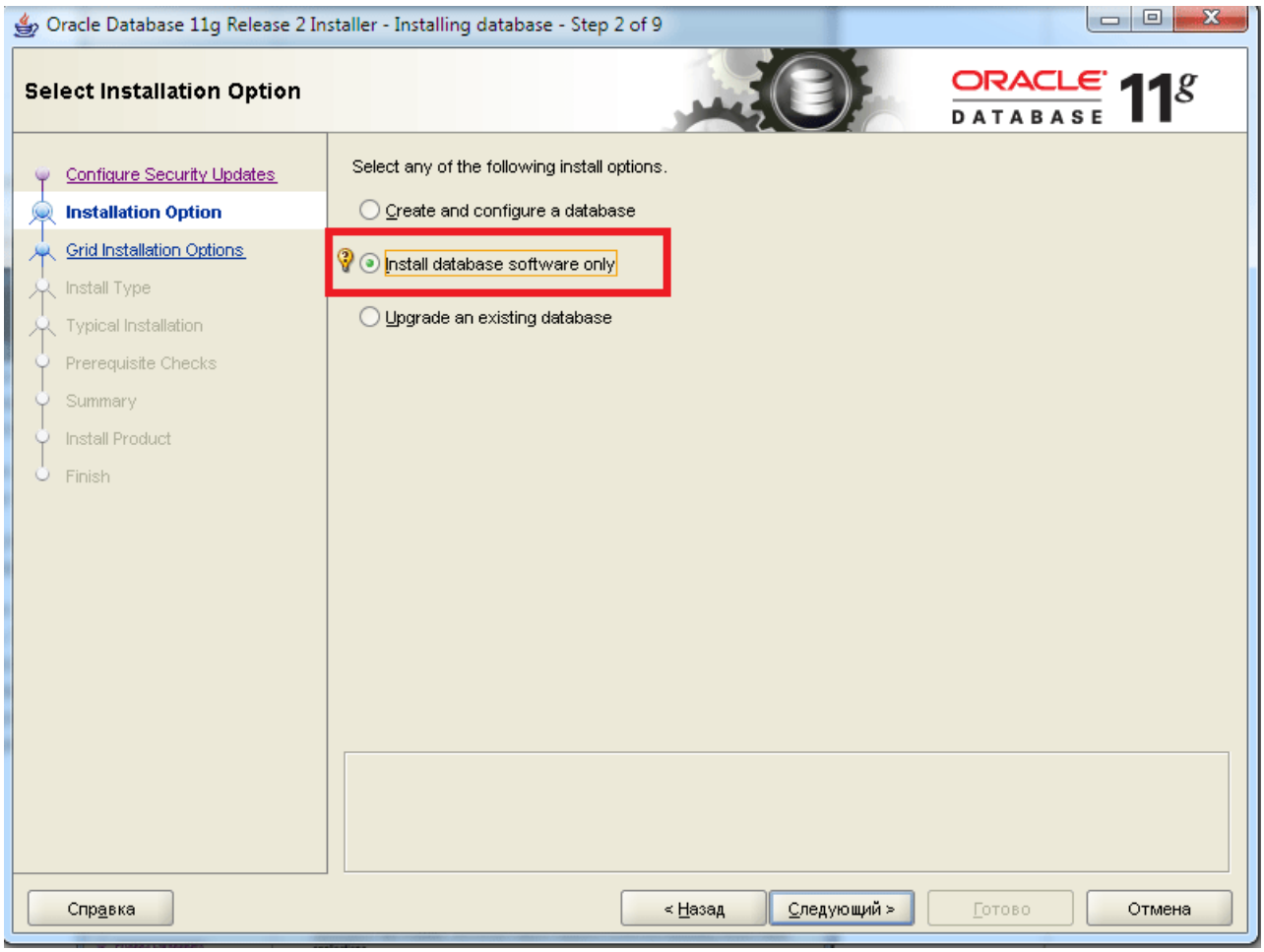

Рисунок 12 – Установка СУБД Oracle. Шаг 2

Шаг 3.Grid Options: выбирается *Single instance*.

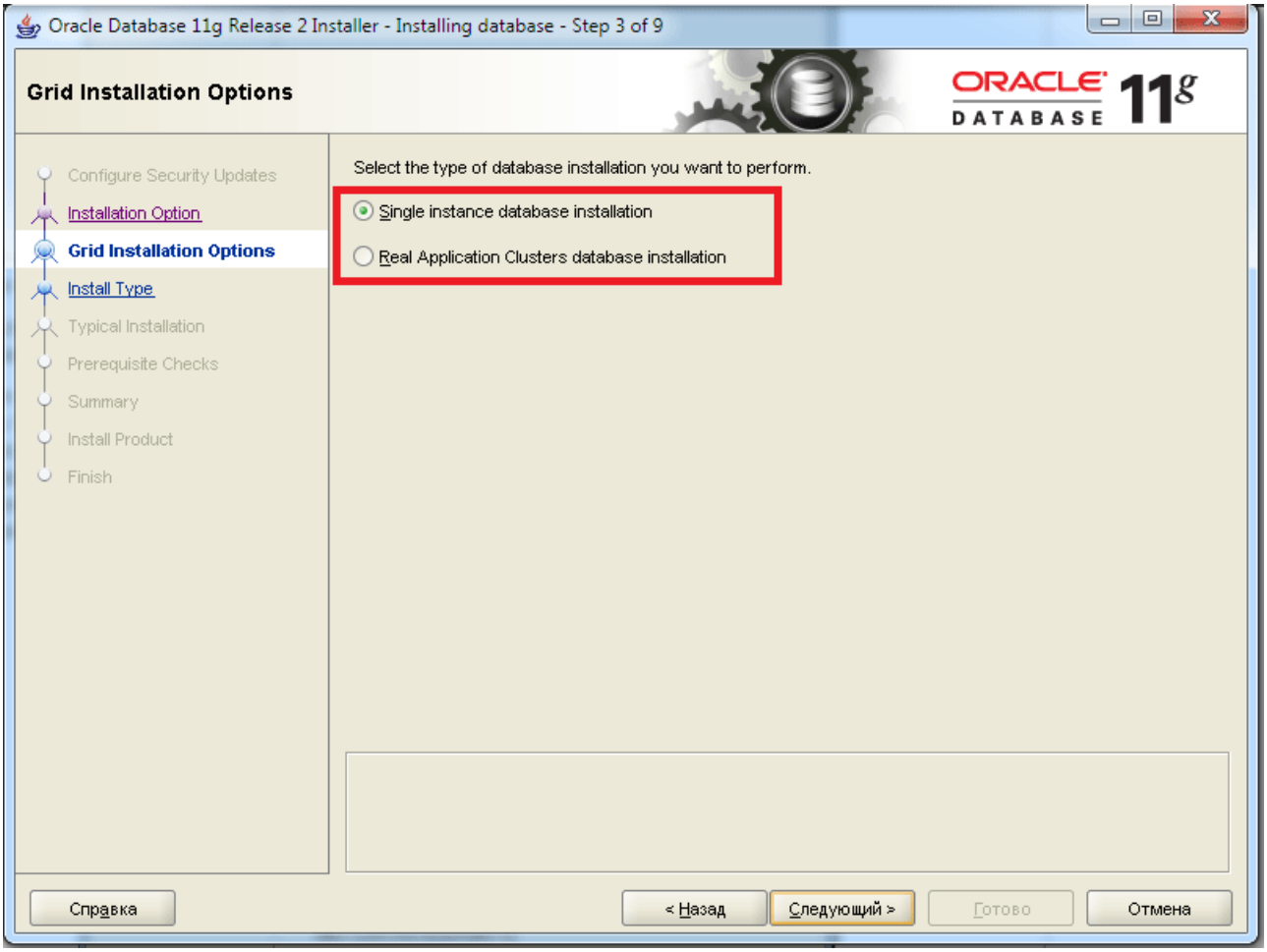

Рисунок 13 – Установка СУБД Oracle. Шаг 3

Шаг 5. Database Edition: выбирается тип редакции, согласно лицензии.

Шаг 6. Installation Location: указывается папка, которая создана для установки Oracle.

Шаг 7. Prerequisite Checks: выполняются проверки, которые должны быть пройдены. При возникновении ошибок необходимо устранить их. Для некоторых ошибок, Oracle может автоматически подготовить скрипт, который необходимо выполнить.

Шаг 8. Summary: нажимается кнопка **Готово**. Начинается процесс установки.

Шаг 9. Install Product: ход установки.

Шаг 10. Finish: окончание установки.

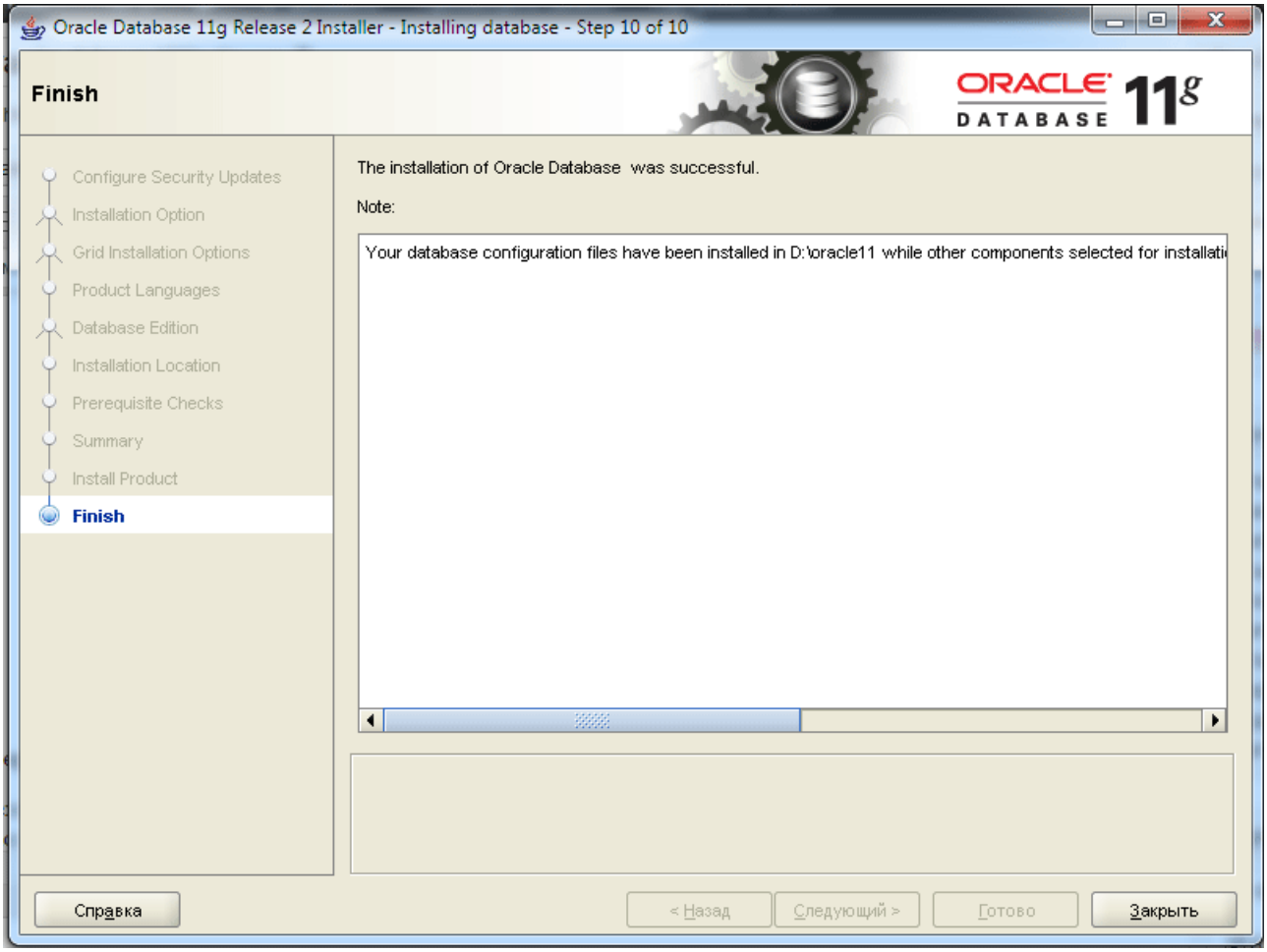

Рисунок 14 – Установка СУБД Oracle. Шаг 10

Для соединения пользователей с СУБД Oracle используется прослушиватель (listener). Для его настройки используется утилита *netca* (Network Configuration Assistant). Под пользователем oracle выполняем команду netca и в графической оболочке следуем указаниям мастера. Если листенер настраивается первый раз, то можно все оставить по умолчанию, либо поменять имя или прослушиваемый порт (по умолчанию 1521).

Далее осуществляется создание экземпляра базы данных. Для создания базы данных наберите в консоли команду **dbca &** (**&** – означает запустить как отдельный процесс). Результатом команды является запуск графического ассистента:

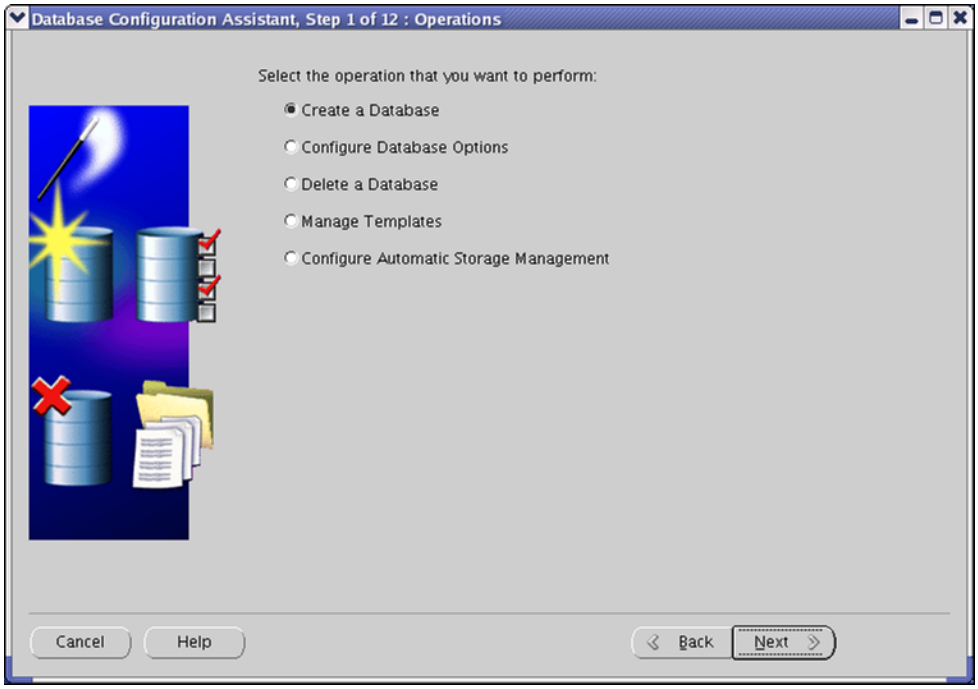

Рисунок 15 – Создание экземпляра базы данных. Шаг 1

Выбирается Create a database (Создать базу). Нажимается кнопка **Next**.

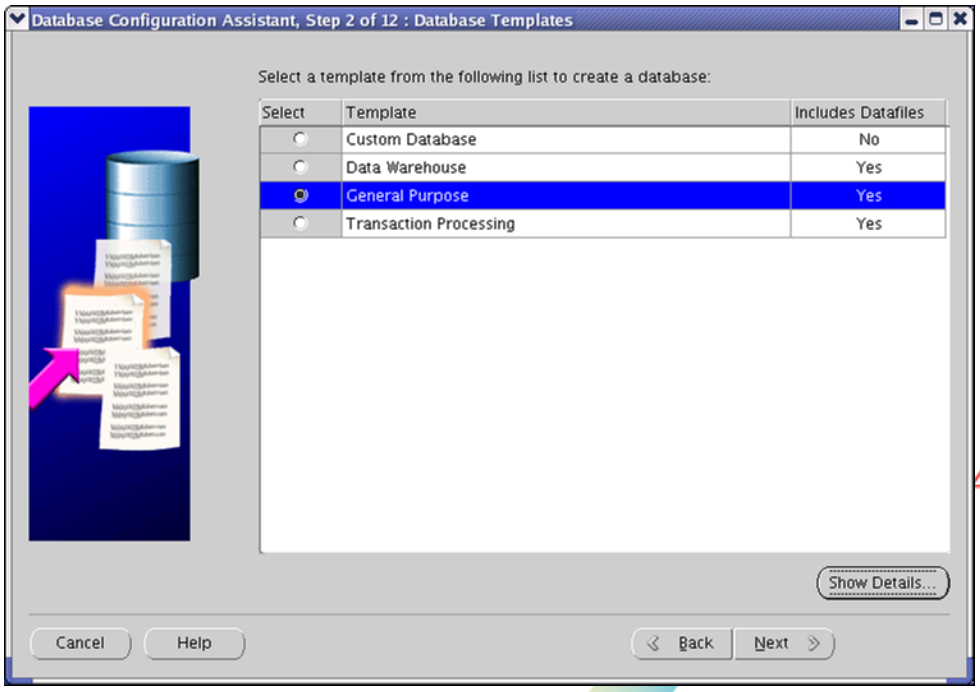

Рисунок 16 – Создание экземпляра базы данных. Шаг 2

Выбирается General Purpose (Универсальный шаблон). Нажимается кнопка **Next**.

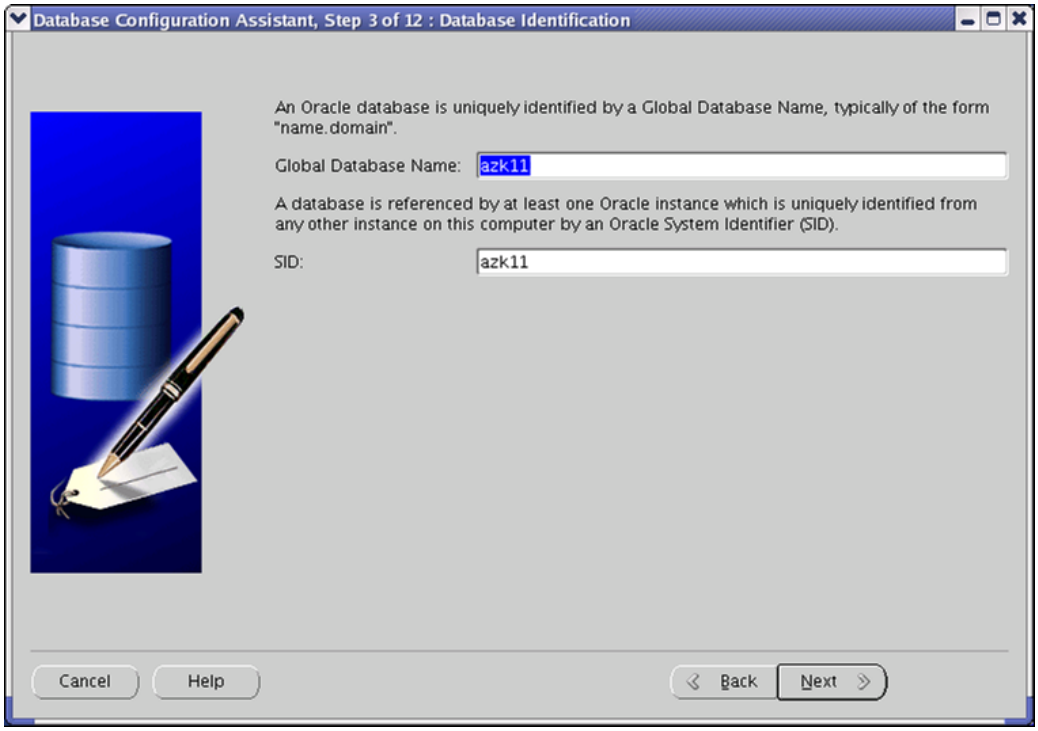

Рисунок 17 – Создание экземпляра базы данных. Шаг 3

Указывается название базы согласно значению переменной ORACLE_SID (ограничение 8 символов и не должно начинаться с цифры). Нажимается кнопка **Next**.

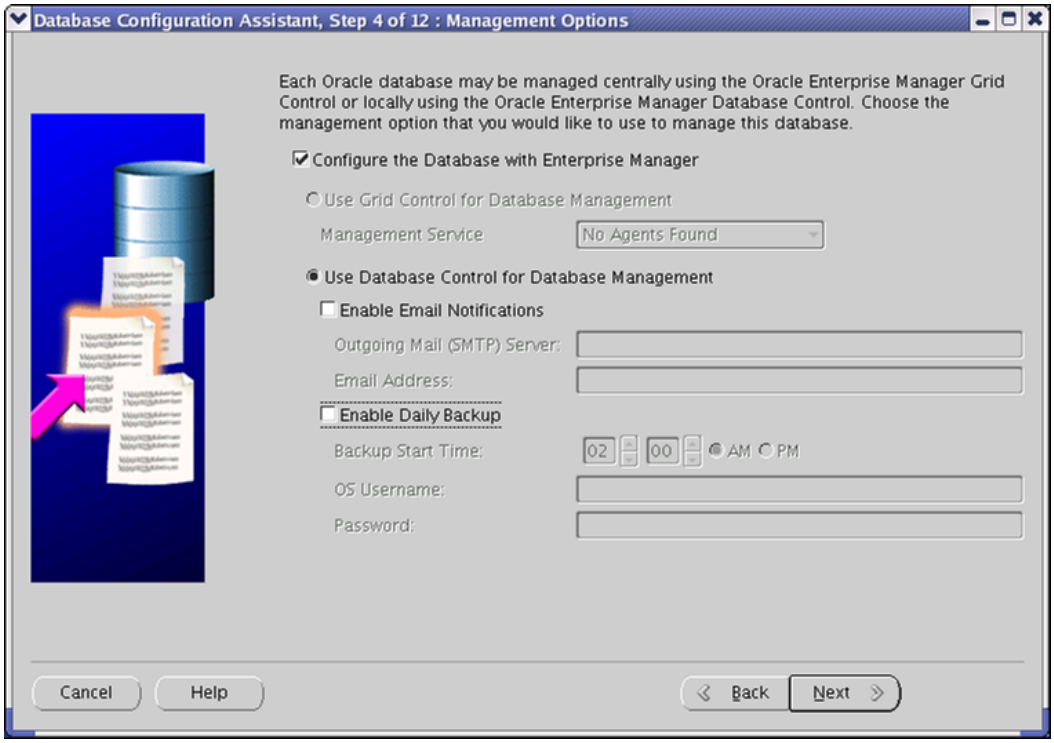

Рисунок 18 – Настройка конфигурации базы. Шаг 4

Соглашаемся на конфигурирование базы с «Enterprise Manager». Нажимается кнопка **Next**.

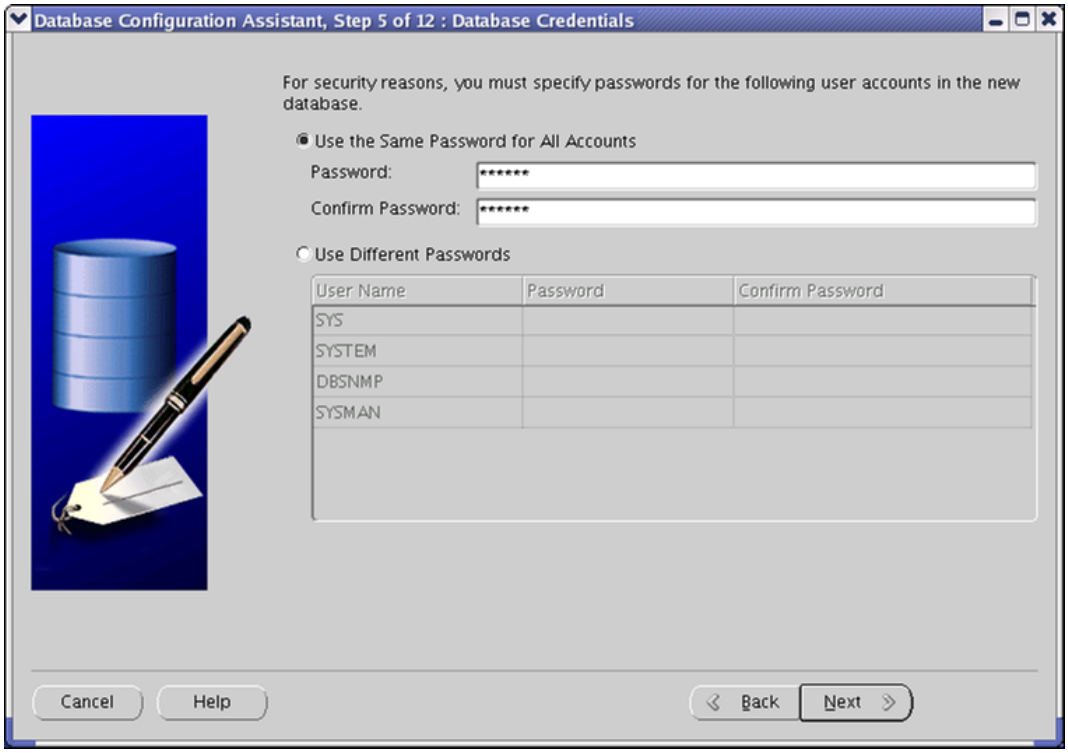

Рисунок 19 – Установка паролей для учетных записей базы данных. Шаг 5

Создается для всех учетных записей пароль «oracle». Нажимается кнопка **Next**.

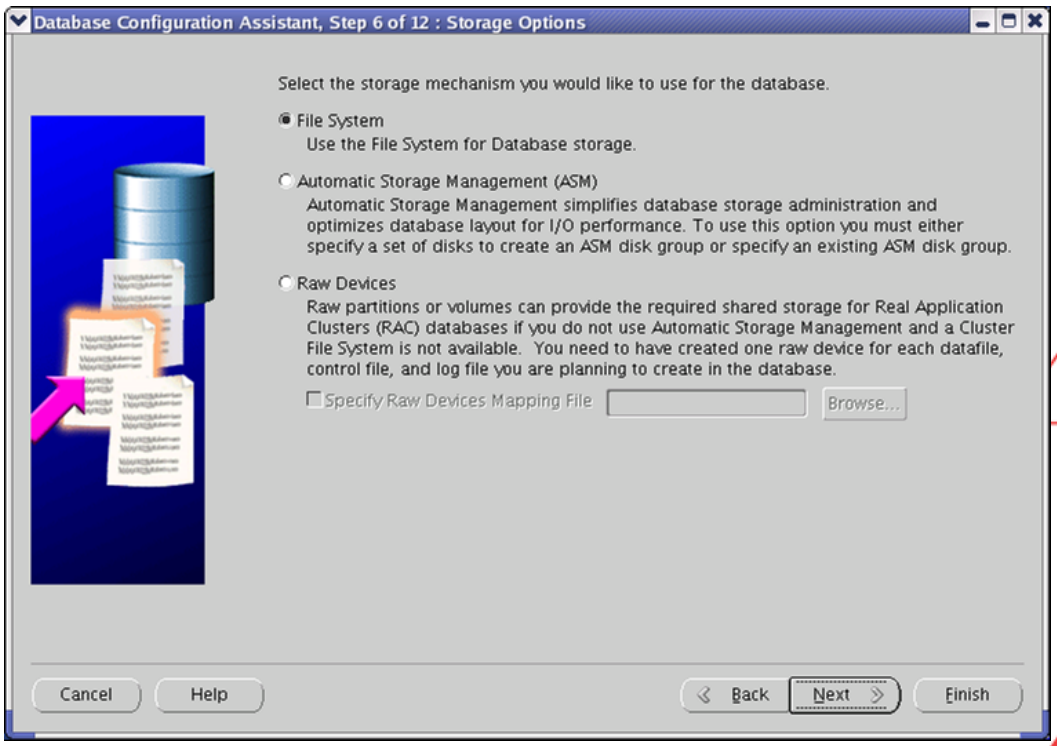

Рисунок 20 – Выбор механизма хранения. Шаг 6

Устанавливается механизм хранения «File system». Нажимается кнопка **Next**.

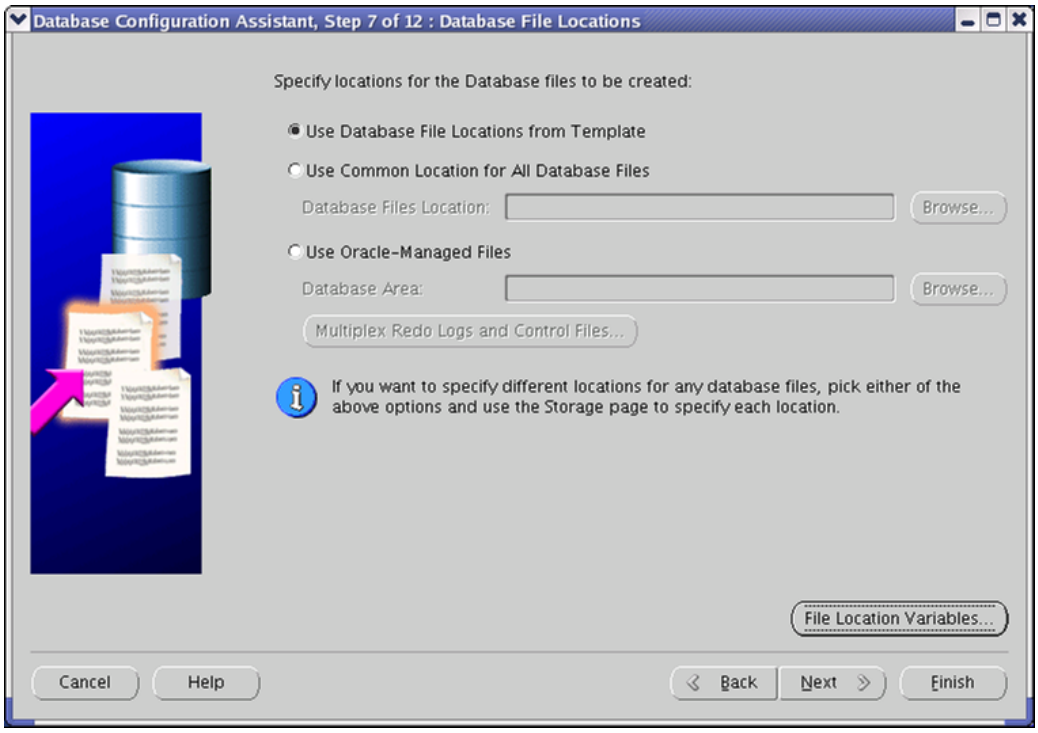

Рисунок 21 – Расположение файлов базы данных. Шаг 7

Настройки пути оставить по умолчанию. Нажимается кнопка **Next**.

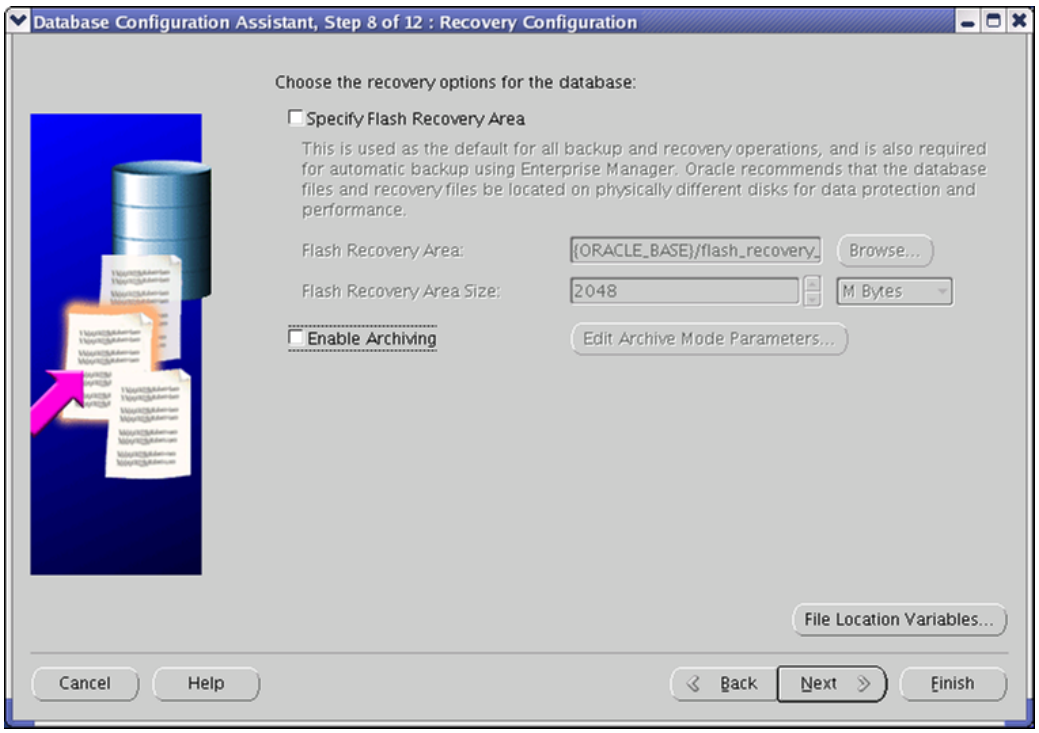

Рисунок 22 – Конфигурация области восстановления. Шаг 8

Настройки пути оставить по умолчанию. При желании Flashback область можно вынести на другие физические диски. Нажимается кнопка **Next**.

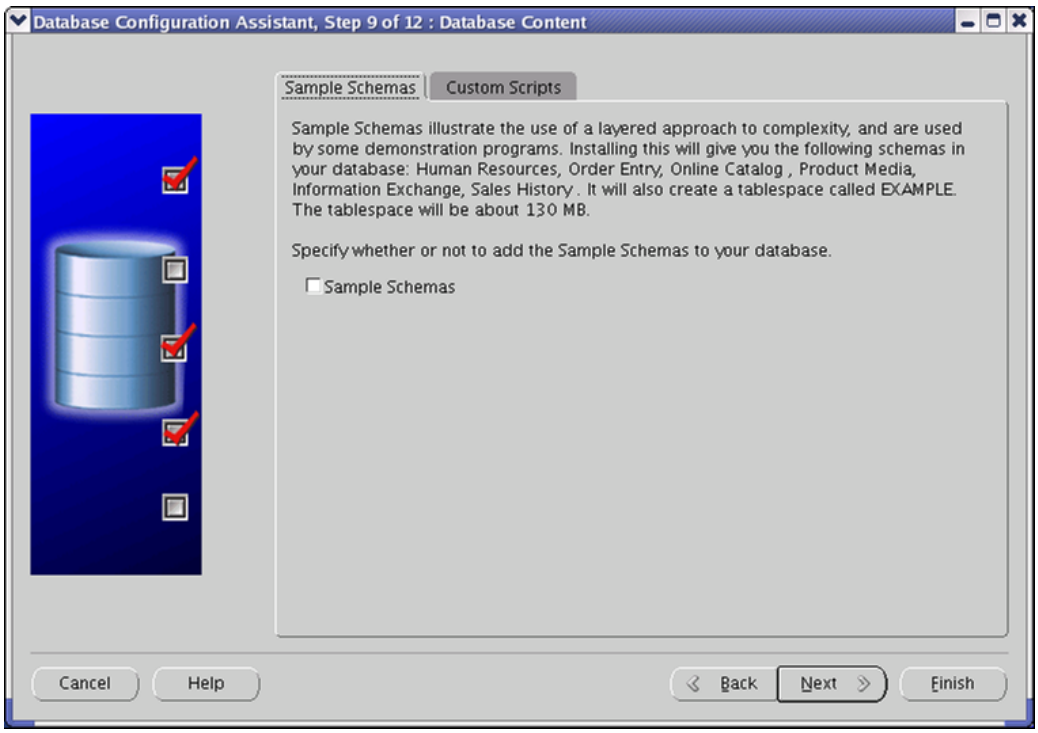

Рисунок 23 – Наполнение БД. Шаг 9

Не устанавливаем демонстрационные схемы. Нажимается кнопка **Next**.

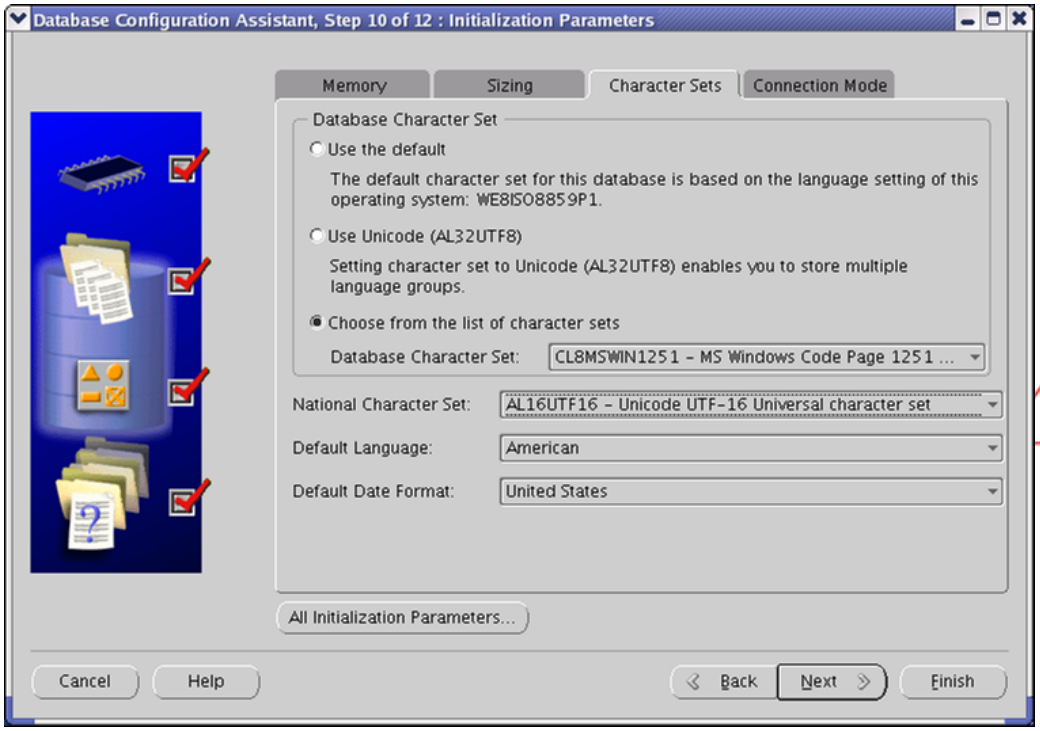

Рисунок 24 – Параметры инициализации. Шаг 10

Выбирается закладка **Character Sets**. Значения остальных закладок настраиваем согласно рекомендуемым.

На закладке **Character Sets** устанавливаем переключатель на Choose from the list of character sets и выбираем из списка CL8MSWIN1251 (указывает кодовую страницу, в которой будут храниться объекты в базе). При нажатии кнопки **All initialization parameters** откроется окно со списком параметров сервера, в котором нажимаем кнопку **Show advanced parameters**. Устанавливаем следующие параметры:

Open_cursors=3000

optimizer_index_cost_adj=1

undo_retention=10800

recyclebin=off

db_securefile=always

audit_trail=none

db_ultra_safe=data_only

deferred_segment_creation = false

control_management_pack_access = "DIAGNOSTIC+TUNING" (если при установке был выбран тип Standard Edition).

Нажимаем кнопку **Next**.

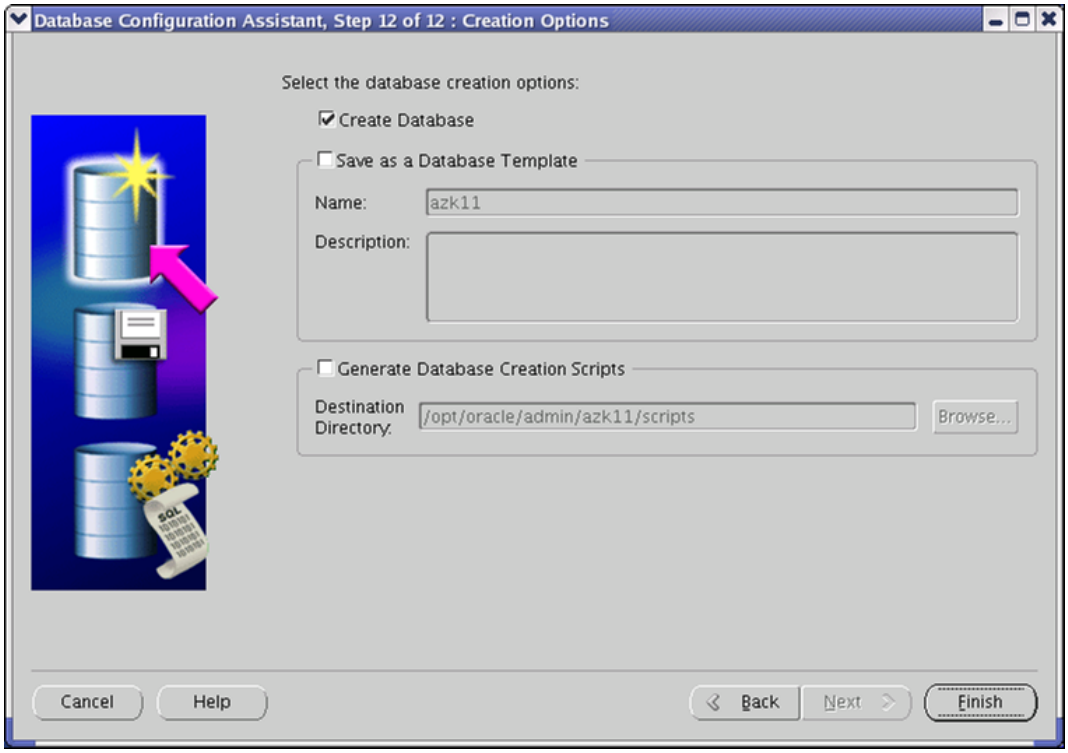

Рисунок 25 – Создание базы. Шаг 12

Устанавливается параметр **Create Database**. Нажимается кнопка **Finish**. Запускается процесс создания базы.

По окончании создания базы СУБД Oracle установлена и запущена.

1.4 Обновление СУБД Oracle до версии 11.2

Прямое обновление до Oracle 11g возможно с существующих баз данных версий 9.2.0.4 и выше, 10.1.0.2 и выше, или 10.2.0.1 и выше. Обновление с остальных версий возможно только через промежуточное обновление до поддерживаемой обновление версии.

Если установка Oracle 11g производится на тот же сервер, где стоит Oracle 9i/10g необходимо:

Создать пользователя oracle11, под которым будет выполняться установка Oracle и необходимые директории:

```
useradd -c "Oracle software owner" -g oinstall -G dba oracle11
passwd oracle11
```

Примечание. *Подразумевается, что группы oinstall и dba созданы во время установки Oracle 9i/10g*

```
mkdir /opt/oracle/product/11.2
chown oracle10.oinstall /opt/oracle/product/10.2
```

Установить переменные окружения. Редактируем файл **.bash_profile** в директории **/home/oracle11**. Прописываем в нем все необходимые переменные для установки Oracle 11g.

```
export ORACLE_BASE=/opt/oracle
export ORACLE_HOME=$ORACLE_BASE/product/11.2
export ORACLE_SID=azk11
export NLS_LANG=AMERICAN_AMERICA.CL8MSWIN1251
export
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$ORACLE_HOME/lib:/home/oracle/lib:/usr/lib:/usr/local/
lib
export PATH=$PATH:$ORACLE_HOME/bin:/usr/bin:/sbin:/bin:/usr/local/bin
export LIBPATH=$ORACLE_HOME/lib
export THREADS_FLAG=native
umask 022
```

Если установка Oracle 11g производится на отдельный сервер, необходимо:

Создать пользователя oracle, под которым будет выполняться установка Oracle и необходимые директории.

Важно! *Настройку базы данных можно осуществить автоматически с помощью следующих пакетов:*

- ***oracle-validated.rpm*** *для версии Oracle Linux не ниже 5x;*
- ***oracle-rdbms-server-11gR2-preinstall*** *для версии Oracle Linux не ниже 6x.*

Для ручной настройки выполняются следующие действия.

- Установить переменные окружения. Редактируем файл **.bash_profile** в директории **/home/oracle**. Прописываем в нем все необходимые переменные для

установки Oracle 11g (аналогично установке Oracle 11g на сервер, где стоит Oracle 9i/10g).

Добавляем следующие строчки в файл **/etc/sysctl.conf**:

```
kernel.shmmax = 536870912
```

Примечание. *Определяется как 50% от физической оперативной памяти. Максимальный размер разделяемого сегмента памяти.*

```
kernel.sem = 250 32000 100 1024
```

Примечание. *Устанавливает количество семафоров в системе.*

```
net.ipv4.ip_local_port_range = 1024 65000
```

Примечание. *Настройка socket*

```
net.core.wmem_max = 262144
```

Примечание. *Настройка буферов TCP.*

```
net.core.rmem_max = 1048576
net.core.wmem_default = 262144
net.core.rmem_default = 1048576
```

После сохранения данного файла необходимо выполнить команду *sysctl -p*

- Далее осуществляется проверка наличия необходимых библиотек командой **rpm -qa | grep имя_библиотеки**. Минимально необходимые требования для различных платформ находятся в папке Req. При отсутствии каких либо библиотек, их необходимо установить из дистрибутива ОС.

Установка Oracle

Установка Oracle, выполняется пользователем oracle11 в графической оболочке. Запускаем **/home/oracle/soft/Database/runInstaller** (см. раздел Установка и настройка СУБД Oracle версии 11.2.0.3 [17]). В завершении установщик снова попросит выполнить скрипт **/opt/oracle/product/11.2/root.sh** от пользователя root.

Если Oracle устанавливается первый раз, то установщик попросит выполнить так же скрипт **/opt/oracle/oraInventory/orainstRoot.sh** от пользователя root.

Далее осуществляется создание базы данных (см. раздел **Установка и настройка СУБД Oracle версии 11.2.0.3**[20])

По окончании создания базы СУБД Oracle установлена и запущена.

После выполнения данного скрипта, необходимо выполнить экспорт переносимых схем с помощью командного файла **exportdp.sh(exportdp.cmd)**.

После получения файлов дампов их необходимо импортировать в созданную базу 11g. Необходимо выполнить импорт полученных дампов с помощью командного файла **importdp.sh (importdp.cmd)**.

Раздача привилегий для пользователя Транспорта

Так как в Oracle 11g ограничены привилегии роли CONNECT, используемой пользователем в работе Транспортной подсистеме, то следующим скриптом необходимо раздать ему привилегии:

```
GRANT "CONNECT" TO WT15;
GRANT "RESOURCE" TO WT15;
```

Кроме этого необходимо проверить, чтобы на клиенте использовались ODBC драйвера от Oracle 11g.

1.5 Установка и настройка СУБД Firebird

СУБД Firebird – это свободно распространяемый программный продукт с открытым исходным кодом, который показал достаточно устойчивую работу на базах малого и среднего размера. СУБД может быть использована в среде ОС Windows и Linux. В процессе эксплуатации СУБД под разными платформами есть отличия только в процессе ее инсталляции.

Инсталляции СУБД Firebird в среде ОС Windows

Для инсталляции СУБД Firebird в среде ОС Windows необходимо распаковать файлы дистрибутива в отдельный каталог (например, **C:\Program Files\Firebird**); как правило, дистрибутив поставляется в виде архивного файла. Для регистрации и запуска службы необходимо запустить файл **.\Firebird\bin\install_super.bat**. После исполнения этого файла сервер запущен и может отвечать на запросы. После первого запуска сервера

приложений для администратора базы данных (SYSDBA) установлен пароль по умолчанию (masterkey). Это общеизвестный факт. Поэтому в целях безопасности необходимо изменить этот пароль на другой. Для более простого вызова утилит Firebird командной строки рекомендуется добавить путь к каталогу **.\Firebird\bin** в переменную окружения PATH. Для устойчивой работы Firebird удалите с компьютера все его предыдущие версии. Так же на устойчивость в работе СУБД может повлиять совместная установка на одной машине Firebird и Interbase. Для остановки службы Firebird и отмены регистрации ее в реестре необходимо использовать командный файл **.\Firebird\bin\uninstall.bat**; после чего каталог с Firebird можно удалять.

Инсталляции СУБД Firebird в среде ОС Linux

Дистрибутивы Firebird для использования под управлением ОС Linux обычно поставляются в виде инсталляционных пакетов (**.rpm*).

Для просмотра наличия СУБД на сервере необходимо выполнить команду *rpm -qa | grep Fi*. Для установки пакета необходимо выполнить команду *rpm – i* путь к пакету.

Пример:
rpm -i /root/Firebird-2.5.1.26351_1_Win32.rpm

Для установки Firebird войдите в систему с правами администратора системы (root) и используйте следующую команду:

#rpm -Uvh <firebird_install.rpm>

где **<firebird_install.rpm>** – имя инсталляционного пакета Firebird. Обычно каталоги и файлы Firebird по умолчанию распаковываются в каталог **opt/firebird/**.

После инсталляции сервер запущен и готов отвечать на запросы. В ходе инсталляции, в отличие от инсталляции под ОС Windows, для системного администратора (SYSDBA) генерируется уникальный пароль, значение которого хранится в файле **./firebird/SYSDBA.password**. Это значение, при необходимости, можно изменить, используя командный файл **./firebird/bin/changeDBAPassword.sh**, указав старый и новый пароль. Перед тем как устанавливать Firebird на вашу систему, в ней будет создан пользователь firebird, принадлежащий одноименной группе. Все файлы Firebird будут принадлежать ему. Из этого следует то, что все файлы баз данных должны иметь соответствующие атрибуты или параметры владельца, чтобы была возможна работа с ними для СУБД. Можно изменить

владельца файлов и группу для файлов Firebird на root с помощью командного файла **./firebird/bin/CSrestoreRootRunUser.sh**. Из соображения безопасности делать это не рекомендуется.

Чтобы удалить Firebird из системы, необходимо использовать команду

```
#rpm -e <packet_name>
```

где **<packet_name>** – наименование установленного пакета Firebird.

После удаления Firebird с машины, файлы базы данных остаются на прежних местах. Их можно использовать только с той версией Firebird, под управлением которой они были созданы (разархивированы). Во всех остальных случаях строго необходимо мигрировать данные через создание резервной копии БД.

Для работы СУБД Firebird в составе «АЦК-Планирование» функциональность первой была расширена за счет внешних подключаемых модулей. Поэтому по окончанию установки Firebird на Ваш компьютер, скопируйте файл **rpl_version.dll** (для ОС Windows) или **rpl_version.so** (для ОС Linux) в каталог **./firebird/udf/**.

1.6 Резервное копирование БД

1.6.1 Резервное копирование СУБД Firebird

1.6.1.1 Резервное копирование с помощью командной строки

Наиболее универсальным инструментом, позволяющим осуществить резервное копирование базы данных на любой платформе (OS Windows, OS Linux), является ***gbak*** – утилита командной строки, входящая в поставку Firebird.

Для того чтобы создать резервную копию базы данных, необходимо воспользоваться следующим образцом запуска ***gbak***:

```
gbak [-b] [options] <база_данных-источник> <файл резервной копии>
```

Где:

- ***-b*** *– означает, что необходимо выполнить резервное копирование базы данных, путь к которой указан как <база_данных-источник>, а результаты резервного копирования упаковать в файл, указанный как <файл резервной копии>:*

*• <**база_данных-источник**> – путь к базе данных, для которой будет создана резервная копия (например, base.fdb);*

*• <**файл резервной копии**> – результат резервного копирования в виде файла (например, base.fbk);*

*• **options** – дополнительные опции из следующего списка:*

*○ **-e**(xpand) – не производить сжатие резервной копии;*

*○ **-g**(arbage_collect) – не собирать «мусор» во время резервного копирования;*

*○ **-ig**(nore) – инорировать контрольные суммы;*

*○ **-m**(etadata) – произвести резервное копирование только метаданных;*

*○ **-u**(ser) name – имя пользователя, который подключается к базе данных для резервного копирования;*

*○ **-pas**(sword) text – пароль пользователя, подключающегося к базе данных для резервного копирования;*

*○ **-v**(erbose) – включить показ подробного протокола действий gbak во время backup;*

*○ **-y** <path> – направлять сообщения в файл (файла с таким именем не должно существовать) или подавить вывод сообщений;*

*○ **-z** – показать версию gbak и версию ядра Firebird-сервера.*

Далее указаны примеры запуска ***gbak*** на платформах OS Linux и OS Windows.

1.6.1.1.1 Рекомендации для OS Windows

В OS Windows утилита ***gbak*** обычно расположена по пути ***C:\Program Files\Firebird\Firebird_2_5\bin\gbak.exe***. Пример ввода команд для запуска утилиты ***gbak*** с параметрами для резервного копирования (командой ***cd*** осуществляется переход в другой каталог):

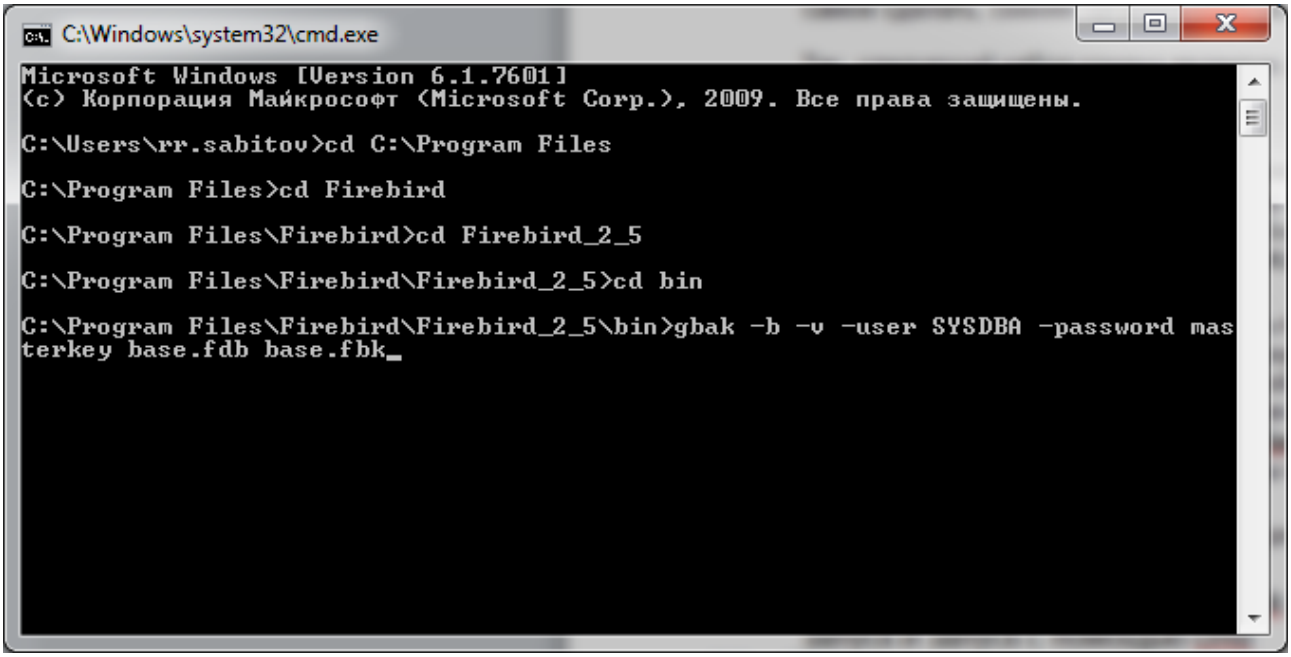

Рисунок 26 – Пример ввода команд

Если добавить ***–y log.txt*** , то процесс резервного копирования будет сохраняться в журнал изменений в указанный файл ***log.txt***.

Пример. *gbak –b –v –y log.txt –user SYSDBA –password masterkey base.fdb base.fbk*

Где:

- ***-b*** *– ключ, отвечающий за необходимость делать резервное копирование;*
- ***-user SYSDBA*** *– имя пользователя, который подключается к базе данных для резервного копирования;*
- ***-pass masterkey*** *– пароль пользователя, подключающегося к базе данных для резервного копирования;*
- ***-v*** *– включить показ подробного протокола действий* ***gbak*** *во время* ***backup****;*
- ***-y log.txt*** *– направлять сообщения в файл* ***log.txt****;*
- ***base.fdb*** *– база данных-источник;*
- ***backup.fbk*** *– файл резервной копии.*

Далее запустится процесс формирования резервной копии базы.

1.6.1.1.2 Рекомендации для OS Linux

Для запуска ***gbak*** на OS Linux, необходимо:

1. Открыть терминал: **Приложения→Терминал** или **Alt+F2→gnome-terminal**.
2. Сменить пользователя на администратора (для упрощения используется учетная запись *root*, но в реальных условиях так делать не следует). Для смены учетной записи ввести команду ***su*** и указать пароль (при вводе пароль не отображается звездочками, точками и т.д.).

```
$ su root
Пароль:
#
```

3. Утилиты Firebird располагаются в ***/opt/firebird/bin***.

```
# cd /opt/firebird/bin
```

4. Перед первым запуском необходимо поменять пароль на пользователя ***SYSDBA***. Для смены пароля пользователя необходимо запустить ***changeDBAPassword.sh***.

Пароль пользователя ***SYSDBA***, который был сгенерирован при установке, находится в файле ***SYSDBA.password***.

Ниже показан процесс смены пароля после запуска файла ***changeDBAPassword.sh***:

```
# cat SYSDBA.password
Firebird generated password for user SYSDBA is:
ISC_USER=sysdba
ISC_PASSWD=ASddsaQ21
.....
# ./changeDBAPassword.sh
Please enter current password for SYSDBA user: ASddsaQ21
Please enter new password for SYSDBA user: masterkey
# cat SYSDBA.password
Firebird generated password for user SYSDBA is:
ISC_USER=sysdba
ISC_PASSWD=masterkey
```

Ключи команд и параметры запуска утилиты ***gbak*** в OS Linux аналогичны ключам ***gbak*** в OS Windows.

Пример. *Следующая команда выполнит резервное копирование базы* ***base.fdb*** *в файл*

***base.fbk** с подробным выводом информации:*

gbak –b –v –y log.txt –user SYSDBA –password masterkey base.fdb base.fbk

Где:

- ***-b** – ключ, отвечающий за необходимость делать резервное копирование;*
- ***-user SYSDBA** – имя пользователя, который подключается к базе данных для резервного копирования;*
- ***-pass masterkey** – пароль пользователя, подключающегося к базе данных для резервного копирования;*
- ***-v** – включить показ подробного протокола действий **gbak** во время **backup**;*
- ***-y log.txt** – направлять сообщения в файл **–y log.txt**;*
- ***base.fdb** – база данных-источник;*
- ***backup.fbk** – файл резервной копии.*

***Внимание!** Показателем корректного резервного копирования будет отсутствие ошибок в логе. В противном случае, необходимо обратиться к официальной документации (https://www.ibase.ru/gbak/) и исходить из информативности ошибки.*

1.6.1.2 Резервирование с помощью пакетного файла (скрипта)

Для упрощения процедуры резервирования можно использовать запускаемый пакетный файл (скрипт).

1.6.1.2.1 Рекомендации для OS Windows

Чтобы создать исполняемый файл *.bat*, необходимо выполнить следующие действия:

1. Запустить Блокнот или любой другой редактор файлов.

   Блокнот позволяет написать код в виде текста, а затем сохранить его в виде *bat*-файла. Блокнот можно запустить, нажав **Пуск→Программы→Стандартные→Блокнот**. Также можно ввести *notepad* в поле **Выполнить**.

   Наиболее простой набор команд включает в себя следующий код:

```
gbak -b -v -user SYSDBA -password masterkey %1 %2
```

Где:

- *%1 – это путь до базы с расширением db*
- *%2 – это путь, куда нужно сохранить резервную копию базы с расширением fbk.*

2. Указанный код необходимо вписать в блокнот.

Для сохранения файла необходимо выполнить следующие действия:

1. Нажать **Файл→Сохранить как**.
2. Нажать выпадающее меню **Тип файла**.
3. Выбрать *Все файлы*.
4. Ввести имя программы, а затем – расширение *.bat* или *.cmd*.

Для запуска *bat*-файла необходимо выполнить следующие действия:

1. Указать путь до места расположения файла.
2. Написать следующую команду: ***name.bat D:\BASE\base.fdb D:\BACKUP\base.fbk***. Это запустит процесс выполнения резервной копии.

1.6.1.2.2 Рекомендации для OS Linux

Необходимо создать запускаемый файл *gbak.sh* и дать ему права на запуск:

```
# touch gbak.sh
# chmod +x gbak.sh
```

Далее потребуется открыть созданный файл в редакторе. В примере используется командный редактор *nano*:

```
# nano gbak.sh
```

Откроется пустой созданный файл. В него необходимо вписать содержание будущего скрипта:

```
#!/bin/bash
# Скрипт выполняет резервное копирование базы
user=SYSDBA
password=masterkey
```

```
# Полный путь до базы данных
data_base_file=/home/oracle/Downloads/BASE_NAME.FDB
# Полный путь до бэкапа (перезатрется если будет указан существующий)
backup_base_file=/home/oracle/Downloads/BACKUP_NAME.bak
# В лог выведется вся информация о процессе выполнения скрипта
log=/home/oracle/Downloads/backup.log
# Утилита gbak выполняет резервное копирование и направляет вывод в лог
/opt/firebird/bin/gbak -b -g -v -user $user -password $password $data_base_file $backup_base_file
>> $log
echo “Резервное копирование выполнено”
```

Для сохранения файла в редакторе *nano* используется сочетание клавиш **Ctrl**+**X**, после чего необходимо нажать **Y** и **Enter** для подтверждения сохранения.

Скрипт запускается под пользователем *root* командой **./gbak.sh**.

Журнал изменений можно увидеть в файле *$log*, который указан в переменной **$log**.

Для того чтобы автоматически запускать скрипт ***gbak*** в ОС Linux, необходимо использовать планировщик задач – CRON.

CRON – классический планировщик задач в UNIX-подобных операционных системах, использующийся для периодического выполнения заданий в определенное время. Регулярные действия описываются инструкциями, помещенными в файлы *crontab* и в специальные директории. Для этого необходимо ввести команду:

```
# crontab -e
```

Это откроет на редактирование таблицу задач планировщика. Чтобы добавить строку, необходимо нажать клавишу **a** и ввести примерно следующий код:

```
### backup databases ######################################################################
00 3  * * * root /opt/admin/gbak.sh
```

Для сохранения нажать **Esc**, затем ввести **wq** и нажать **Enter**.

После сохранения, планировщик будет выполнять ежедневное резервирование в 3 часа 00 минут.

1.6.1.3 Резервное копирование с помощью специальных программ

IBExpert – графическая программа, предназначенная для разработки и администрирования баз данных InterBase и Firebird, а также для выбора и изменения данных, хранящихся в базах.

Для выполнения резервирования необходимо открыть меню **Службы→Резервирование базы данных**:

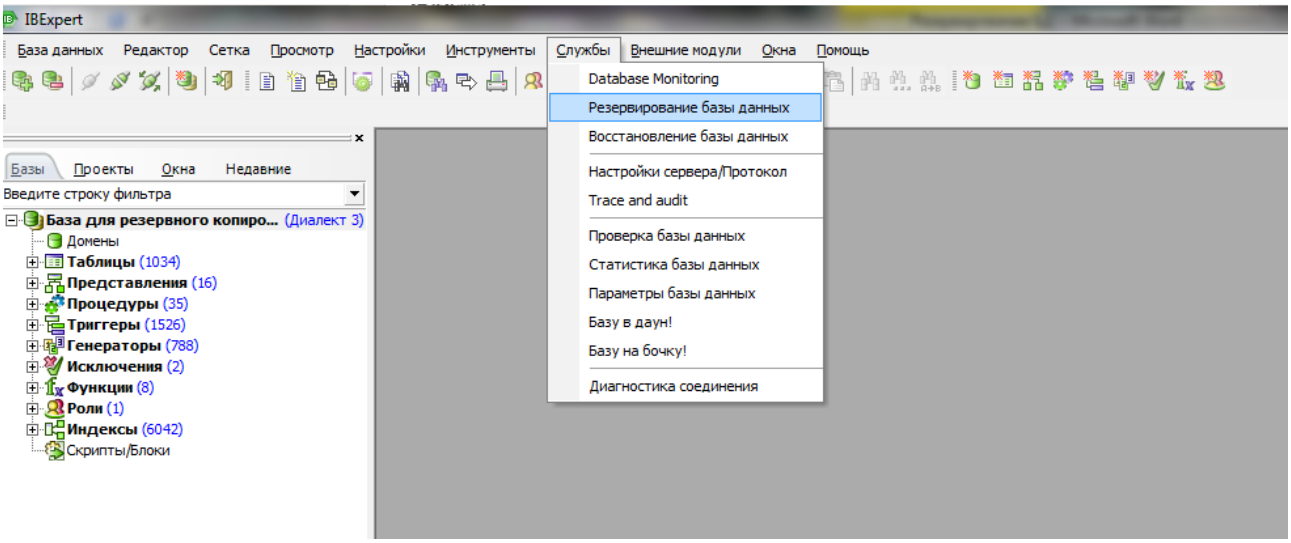

Рисунок 27 – Резервирование базы данных

Откроется окно со следующими настройками:

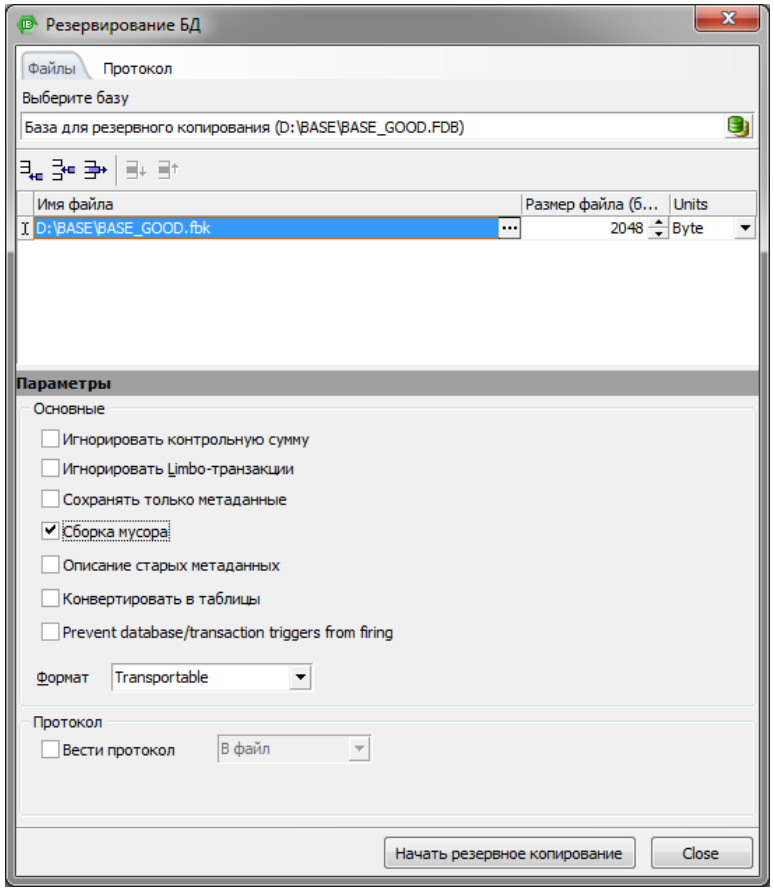

Рисунок 28 – Параметры резервирования базы данных

Для того что бы запустить процесс резервного копирования, необходимо выбрать базу и указать параметры:

- **Игнорировать контрольную сумму** – позволяет игнорировать ошибки контрольных сумм страниц и не останавливать резервное копирование из-за этих ошибок.
- **Игнорировать limbo-транзакции** – не сохраняет в резервной копии БД версии записей, которые созданы транзакциями, находящимися в состоянии *in limbo*. Такое состояние может быть только у не завершившихся транзакций двухфазного коммита (2PC).
- **Сохранять только метаданные** – данные в резервную копию не сохраняются. Используется, когда необходимо сделать копию пустой БД.
- **Сборка мусора (по умолчанию)** – самый важный параметр. Параметр запрещает серверу проверять читаемые записи на наличие мусорных, что ускоряет процесс

бэкапа. Более подробно это описано в документе.

- **Описание старых метаданных** – если параметр включен, то старые описания метаданных включены в базу данных резервного копирования. Это предусмотрено в целях обеспечения совместимости для старых InterBase версий.
- **Конвертировать в таблицы** – параметр преобразует данные базы данных для таблиц в резервной копии. Это касается внешних файлов.
- **Prevent database/transaction triggers from firing** – это равно *isc_dpb_no_db_triggers* опции в ***дополнительных параметрах подключения***.
- **Формат** – выбор формата данных для файла резервной копии базы данных. Рекомендуемое значение по умолчанию – *Переносные*, так как он позволяет восстановить в различные версии Firebird/InterBase, если они пожелают. Он сохраняет данные и метаданные в универсальном формате, в отличие от значения *Non-транспортабельные* .
- **Протокол** – включить признак **Вести Протокол**. Протокол необходим для сохранения операции (в файле), которые производились во время резервного копирования базы.

1.6.1.4 Планировщик АЦК

В некоторых системах линейки продуктов АЦК (АЦК-Финансы, АЦК-Планирование, АЦК-Госзаказ) для резервного копирования баз FireBird существует задание планировщика задач. Для настройки резервного копирования БД FireBird необходимо в файле конфигурации сервера приложения (далее СП) **azk2server.properties** указать путь до файла ***gbak.exe***. Путь к GBAK (только для FireBird):

```
azk.update.backup.gbak=C:/fb/bin/gbak.exe
```

Для автоматизации запуска сервисных задач, таких как внутренние службы, в составе сервера приложений постоянно функционирует фоновый поток планировщика выполнения задач. Его основной задачей является отслеживание расписания запуска заданий и запуск последних в случае, если прошел достаточный интервал времени. Настройка планировщика задач осуществляется из клиентского приложения посредством

пункта меню **Справочники→Планировщик**.

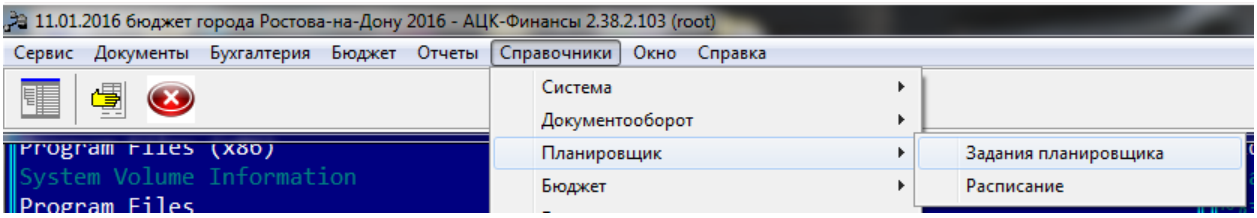

Рисунок 29 – Доступ к заданиям планировщика

В справочнике *Задания планировщика* перечислены процедуры, которые могут быть добавлены в справочник *Расписание планировщика* для выполнения. Существующие способы определения момента запуска процедуры на исполнение позволяют гибко настраивать работу планировщика.

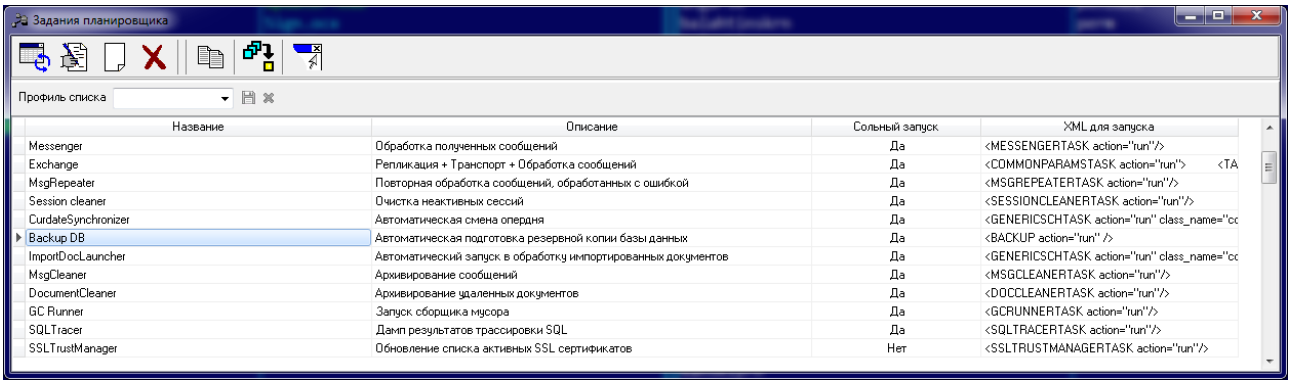

Рисунок 30 – Справочник «Задания планировщика»

При определении задания для планировщика можно ограничить уровень изоляции задания, выключив признак **Запуск на всех серверах приложений**. В этом случае задача будет выполняться только на одном сервере приложений, который входит в кластер.

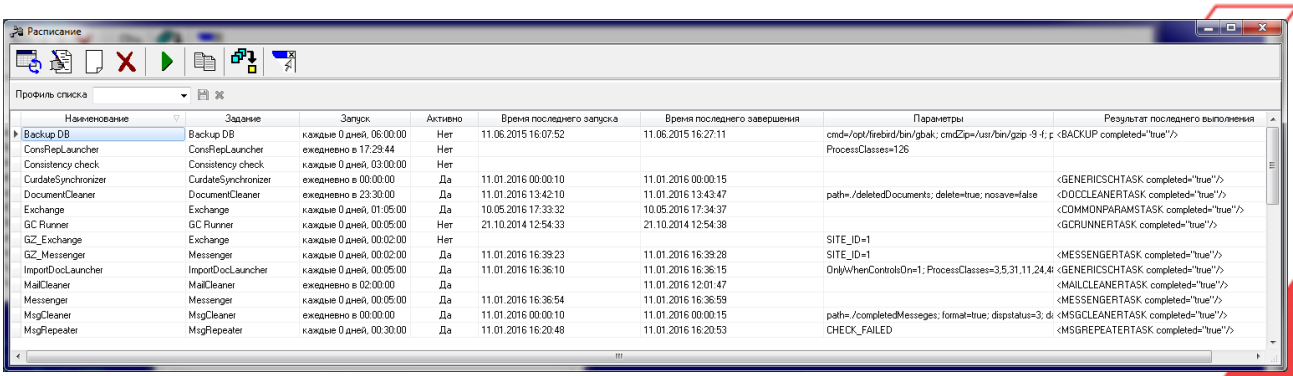

Рисунок 31 – Расписание планировщика

Задания планировщика хранятся в таблице SCHTASK, которая обычно заполняется

из файла ***./xml/schtask.xml.*** В расписании перечислены задания, параметры заданий, время последнего запуска, результаты последнего запуска заданий. Расписание хранится в таблице SCHPLAN.

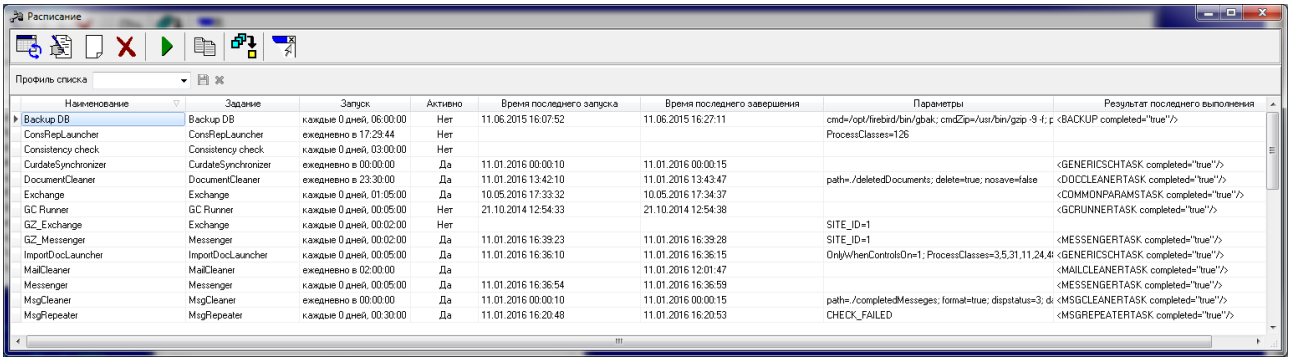

Рисунок 32 – Расписание планировщика

Существует несколько моделей задания времени запуска:

- **Один раз** – если требуется установить дату и время запуска, а также при необходимости повтора задания.
- **Ежегодно** – если необходимо включать контроль каждый год.
- **Периодически** – если требуется указать количество дней, то есть промежуток, через которое задание повториться в назначенное время.
- **Ежедневно** – если задание запускается каждый день в назначенное время.
- **Еженедельно** – если задание запускается в назначенное время в определенные дни недели.

Использование фоновых процессов влечет повышение нагрузки на сервер приложений, поэтому все неиспользуемые службы на сервере приложений рекомендуется удалить из расписания планировщика. Процесс работы планировщика отображается в файле журнализации СП (*azk2.log*).

Теперь рассмотрим ближе содержание задание планировщика BACKUP.

Описание задания в расписании XML для запуска:

<BACKUP action="run" />

***Параметры**: cmd, cmdZip, path, name.*

***Описание**: Автоматическая подготовка резервной копии базы данных.*

***Рекомендуемый период запуска**: 3 часа.*

Процедура предназначена для автоматизированной подготовки резервных копий БД на Firebird для сервера приложений.

Принимает следующие параметры:

*• **cmd**=<**command**> – команда для вызова утилиты командной строки для подготовки резервной копии БД.*

*• **cmdZip**=<**command**> – команда для вызова утилиты сжатия программ; может включать в себя дополнительные параметры командной строки.*

*• **path**=<**path**> – путь, по которому будут размещаться файлы резервных копий.*

*• **fbk**=<**suffix**> – дополнительный суффикс к имени файла резервной копии.*

Пример использования параметров:

cmd=gbak; cmdZip =gzip -9 -f ; path=/backup s; fbk=Tomsk

Значения параметров:

cmd=gbak, можно указать полное имя файла gbak (по умолчанию gbak).

cmdZip =gzip -9 -f , можно указать другой архиватор с валидными ключами (по умолчанию не указан).

path=c:\dir, или host:c:\dir .. и т.п. (по умолчанию .\backups).

fbk=Tomsk – желательно указывать название объекта реализации, обязательно латинскими буквами.

Реализован формат наименования резервной копии БД:

base_X.X.X.X_ГГГГММД Д _ЧЧММСС.f bk,

где:

• base – имя файла, определяется по параметру name;

• X.X.X.X – версия билда;

• ГГГГММДД – текущая дата;

• ЧЧММСС – текущее время

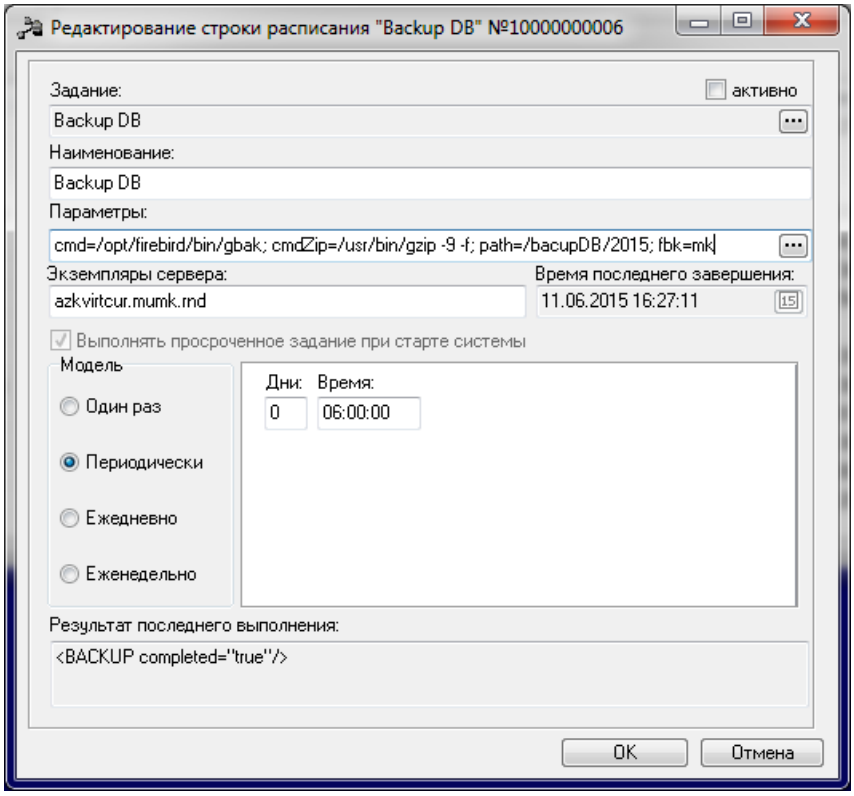

Рисунок 33 – Форма редактирования строки расписания

Можно не задавать ни одного параметра, но архивные копии будут создаваться на том же хосте, что и БД, что крайне нежелательно.

Данный способ рекомендуется для выполнения резервного копирования баз Firebird, т.к. при этом используется утилита ***gbak***, входящая в стандартный состав утилит Firebird и планировщик АЦК упрощает с ней работу.

1.6.2 Резервирование СУБД ORACLE

1.6.2.1 Резервное копирование с помощью командной строки

Резервное копирование БД Oracle осуществляется утилитой ***expd***. Она входит в пакет стандартных утилит.

Для запуска утилиты ***expdp*** (в OS Windows утилита располагается *C:\Oracle\ora92\bin\expdp*, в OS Linux утилита располагается */usr/oracle/bin/expdp*), подготовлен скрипт ***exportdp.cmd*** (для Linux ***exportdp.sh***), который предварительно необходимо создать и настроить (данные строки могут отличаться):

- SYSTEM – имя администратора СУБД.
- PASSWD – пароль администратора СУБД.
- URL – путь для развертывания БД.
- ORACLE_HOME – путь к каталогу Oracle.

Формат вызова утилиты ***exportdp***:

```
exportdp <имя схемы> <название файла>
```

Где:

• <имя схемы> – имя экспортируемой схемы;

• <название файла> – имя получаемого файла дампа (необязательный параметр, по умолчанию создастся файл с именем, включающим в себя имя схемы, дату и время создания дампа).

1.6.2.1.1 Рекомендации для OS Windows

Содержание скрипта ***exportdp.cmd***:

```
@echo off
rem -----------------------------------
set SYSTEM=system
set PASSWD=oracle
set URL=@localhost/xe
set ORACLE_HOME=C:\Oracle\ora92
rem -----------------------------------
if "%1" == "" goto hlp
set USERNAME=%1
set FILENAME=%2
set LOGFILE=backup.log
for /f "tokens=3 delims=." %%D in ("%DATE%") do set YEAR=%%D
for /f "tokens=2 delims=." %%D in ("%DATE%") do set MONTH=%%D
for /f "tokens=1 delims=." %%D in ("%DATE%") do set DAY=%%D
for /f "tokens=1 delims=:" %%T in ("%TIME%") do set HOUR=%%T
for /f "tokens=2 delims=:" %%T in ("%TIME%") do set MINUTE=%%T
for /f "tokens=3 delims=:," %%T in ("%TIME%") do set SECOND=%%T
```

```
if "%FILENAME%" == "" (
   set FILENAME=%1_%YEAR%-%MONTH%-%DAY%_%HOUR%%MINUTE%%SECOND%.dmp
   set LOGFILE=%LOGFILE
)
 if not "%ORACLE_HOME%" == "" (
  set CMD="%ORACLE_HOME%/bin/expdp"
) else (
  set CMD=expdp
)
echo %CMD%
if exist %FILENAME% (
  echo ERROR!
  echo File %FILENAME% is already exist.
  goto end
)
echo ----------------------------------------
echo Backing-up %USERNAME% into %FILENAME% ...
echo ----------------------------------------
set NLS_TIMESTAMP_FORMAT=YYYY-MM-DD HH24:MI:SS
%CMD%  %SYSTEM%/%PASSWD%%URL%  SCHEMAS=%USERNAME%
DIRECTORY=backup_dir  EXCLUDE=TABLE_STATISTICS,INDEX_STATISTICS  DUMPFILE=%
FILENAME% LOGFILE=%LOGFILE% FLASHBACK_TIME="TO_TIMESTAMP(sysdate)"
goto end
```

После сохранения скрипт запускается в командной строке. Например:

```
exportdp.cmd LNXSTEND23 lnx.dmp
```

Где:

```
• LNXSTEND23 – имя схемы.
• lnx.dmp – имя файла, в который сохранится резервная копия.
```

Путь выгрузки файла бэкапа базы данных можно определить, задав перемененную, в нашем случае это переменная ***backup_dir***. Сделать это можно следующим образом:

1. заходим на сервер СУБД в **sqlplus**;
2. определяем данную переменную **backup_dir**:

```
sqlplus /nolog
```

```
conn sys/sys@azk11 as sysdba
SQL> create or replace directory backup_dir as 'd:\backup';
Directory created.
```

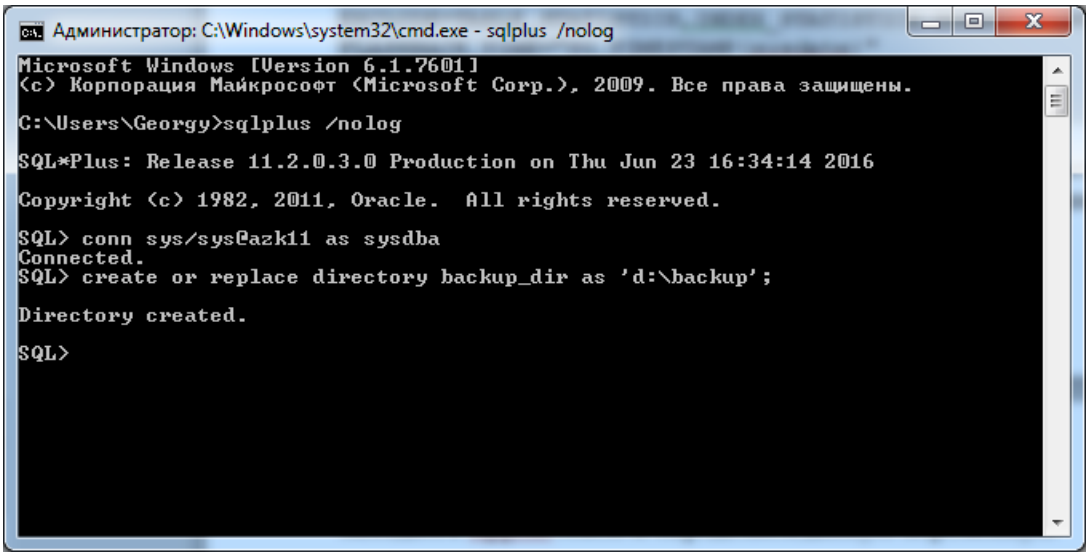

Рисунок 34 – Определение переменной backup_dir

1.6.2.1.2 Рекомендации для OS Linux

Чтобы выполнить резервное копирование на OS Linux, необходимо создать директорию, в которую будут сохраняться резервные копии:

```
# mkdir /expdir
```

Выполнить вход в консоль ***sqlplus*** от пользователя SYS и создать виртуальное пространство для директории:

```
# sqlplus SYS as SYSDBA
```

После входа в консоль ***sqlplus***:

```
1    create directory expdir as '/expdir';
2    select * from dba_directories;
```

Для резервирования рекомендуется использовать отдельного пользователя. Пример создания пользователя приведен ниже:

```
1    CREATE USER EXPORT
2    IDENTIFIED BY password
3    DEFAULT TABLESPACE USERS
```

```
4       TEMPORARY TABLESPACE TEMP
5       PROFILE "UNLIMITED PASSWORD EXPIRATION"
6       ACCOUNT UNLOCK;
7
8       GRANT CREATE SESSION TO EXPORT;
9       GRANT CREATE TABLE TO EXPORT;
10      ALTER USER EXPORT QUOTA UNLIMITED ON USERS;
11      GRANT EXP_FULL_DATABASE TO EXPORT;
12      GRANT READ, WRITE ON DIRECTORY EXPDIR TO EXPORT;
```

Содержание исполняемого файла, который будет запускать процесс резервирования, приведено ниже (ORACLE_SID – экземпляр базы данных, данное значение может отличаться):

```
$ vi ora_expdp_full.sh
#!/bin/sh
STARTTIME=`date`
export ORACLE_SID=xe
export ORACLE_HOME=`cat /etc/oratab|grep ^${ORACLE_SID}:|cut -d':' -f2`
export EXPLOG=expdp_${ORACLE_SID}.log
export EXPDIR=/expdir
export PATH=$PATH:$ORACLE_HOME/bin
DATEFORMAT=`date +%Y%m%d`
STARTTIME=`date`
expdp export/password content=ALL directory=expdir dumpfile=expdp_`echo $ORACLE_SID`_%
U_`echo $DATEFORMAT`.dmp filesize=2G full=Y logfile=$EXPLOG nologfile=N parallel=2
ENDTIME=`date`
SUBJECT=`hostname -s`:$ORACLE_SID:`tail -1 $EXPDIR/$EXPLOG`
echo -e "Start time:" $STARTTIME "\nEnd time:" $ENDTIME | mail -s "$SUBJECT"
dba@mydomain.com
```

Скрипт создаст резервную копию размером 2Гб, которая будет динамически расширяться. Например, первый файл будет называться *expdp_oratst_01_20120503.dmp*, второй *expdp_oratst_02_20120503.dmp* и так далее. После создания вся информация о процессе резервирования будет сохраняться в *log*-файл. Необходимо изменить ORACLE_SID и EXPDIR переменные в скрипте на свои.

Для сохранения файла нажать **ESC**, ввести *wq* и нажать **Enter**. Скрипту дать права на запуск:

```
# chmod 700 ora_expdp_full.sh
```

Для автоматизации процесса резервирования этот скрипт можно добавить в планировщик:

```
# crontab -e
```

Вписать данную строку (путь к файлу указан для примера):

```
00 23 * * * /home/oracle/scripts/export/ora_expdp_full.sh 1>/dev/null 2>&1
```

Внимание! *Показателем корректного резервного копирования будет отсутствие ошибок в логе. В противном случае, необходимо обратиться к официальной документации (https://oracle-base.com/articles/10g/oracle-data-pump-10g) и исходить из информативности ошибки.*

1.6.2.2 С помощью программ

Recovery Manager (RMAN) – это утилита Oracle, которая используется для управления операциями резервирования, копирования из резерва и восстановления базы данных Oracle. RMAN имеет мощный командный язык, который не зависит от ОС.

Enterprise Manager Database Control Console обеспечивают графический пользовательский интерфейс к большинству наиболее используемых возможностей RMAN.

Используя утилиту **Recovery Manager**, можно соединиться со следующими видами баз данных:

- База данных назначения.

  Администратор для соединения с БД назначения должен иметь привилегию SYSDBA. Эта привилегия необходима для успешного соединения. При этом происходит подсоединение к экземпляру целевой БД, в которой необходимо выполнить обычные операции RMAN.

- База данных каталога восстановления.

  Это необязательная база данных, которая конфигурируется для репозитория RMAN. Соединение с базой данных каталога восстановления устанавливается для

получения хранимой в нем информации, например, сведений о резервах или хранимых скриптах.

- Вспомогательная база данных. Такая БД может быть:
  - создана по команде **RMAN DUPLICATE**;
  - временной базой данных, используемой при восстановлении табличного пространства на момент времени в прошлом (*tablespace point-in-time recovery - TSP1TR*);
  - резервной базой данных, представляющей собой копию производственной базы данных и используемой при выходе из строя основной БД.

1.6.2.2.1 Запуск RMAN

Соединение с базой данных назначения без использования каталога восстановления

Локальное соединение

Для локального соединения с RMAN необходимо ввести в командной строке ОС:

- в UNIX:

```
# ORACLE_SID=DB01
# rman target /
```

Где:

```
• ORACLE_SID – экземпляр базы данных, к которой выполнится подключение;
• target – указывает на подключение к целевой базе.
```

- в Windows:

```
C :\> SET ORACLE_SID=DB01
C:\> rman target /
```

Где:

```
• ORACLE_SID – экземпляр базы данных, к которой выполнится подключение;
• target – указывает на подключение к целевой базе.
```

Если в команде старта не указываются имя и пароль, а только косая черта (*slash*), происходит соединение под пользователем SYS с аутентификацией на уровне ОС. Можно

также указать необязательное ключевое слово NOCATALOG:

```
$ rman target / nocatalog
```

Где:

• *NOCATALOG – режим по умолчанию, означающий использование RMAN без каталога восстановления.*

Удаленное соединение

Чтобы соединиться с другого сервера, используется псевдоним имени службы целевой базы данных в Oracle Net:

```
$ rman target sys/target_pwd@DB01
```

Команда **Backup** – это копия данных из вашей БД, используемая для воссоздания данных. В результате операции резервирования, выполняемой с использованием ***RMAN***, могут быть созданы либо копии образов, либо резервные наборы. Для резервирования необходимо выполнить выполнить:

```
RMAN> BACKUP DATABASE;
```

Программа ***RMAN*** имеет свой встроенный язык, огромные возможности и гибкость в настройке – эти особенности делают ***RMAN*** мощным и незаменимым инструментом. Утилиту ***RMAN*** рекомендуется использовать как основное средство резервного копирования.

В официальной документации описано более подробно обо всех возможностях: http://www.oracle.com/technetwork/database/features/availability/rman-overview-096633.html.

1.6.3 Резервное копирование СУБД PostgreSQL

PostgreSQL – свободная кроссплатформенная объектно-реляционная СУБД. Но, так как для PostgreSQL в качестве сервера баз данных чаще всего используют ОС Linux, резервное копирования будет рассматриваться только для этой платформы.

1.6.3.1 Резервное копирование с помощью командной строки

В комплект PostgreSQL входят утилиты ***pg_dump***, ***pg_dumpall*** и ***pg_basebackup***, они позволяют делать резервные копии.

Запускать утилиты следует от пользователя **postgres**, для этого в системе Linux необходимо в терминале ввести:

```
# su postgres
```

1.6.3.1.1 Резервное копирование с помощью утилиты Pg_Dump

pg_dump — это программа для создания резервных копий базы данных PostgreSQL. Она создаёт целостные копии, даже если база параллельно используется. Программа ***pg_dump*** не препятствует доступу других пользователей к базе данных (ни для чтения, ни для записи). Существует несколько режимов работы данной программой, отличающихся форматом полученного дампа, скоростью формирования дампа.

В этом разделе приведены основные возможности по созданию дампа. Более подробно можно почитать тут: https://postgrespro.ru/docs/postgrespro/9.5/app-pgdump.

1.6.3.1.1.1 Резервное копирование с использованием формата «Custom»

Один из наиболее гибких форматов, позволяющим вручную выбирать и сортировать восстанавливаемые объекты. Вывод в этом формате по умолчанию сжимается.

```
pg_dump -h localhost -p 5432 -U postgres -Fc -v –d f_base_160629 –b -f /backup/f_base_160629.dmp
```

1.6.3.1.1.2 Резервное копирование с использованием формата «Directory»

Выгрузить в формате каталога. При этом будет создан каталог, в котором для каждой таблицы и большого объекта будут созданы отдельные файлы, а также файл оглавления в машинно-читаемом формате, понятном для программы восстановления БД из дампа ***pg_restore***. С полученной резервной копией можно работать штатными средствами

ОС Unix/Windows, например, несжатую копию можно сжать посредством архиватора, например: *gzip*, *7z*.

```
pg_dump -h localhost -p 5432 -U postgres -Fd -b -v –d f_base_160629 –j 4 -f /backup/f_base_160629
```

Внимание! *В этой команде используется ключ* ***–j****, позволяющий создавать* ***dump*** *в несколько потоков, количество которых равно количеству ядер процессоров сервера.* ***Самый быстрый способ создания дампа!***

1.6.3.1.1.3 Резервное копирование с использованием формата текстового SQL-скрипта

Это поведение по умолчанию:

```
pg_dump -h localhost -p 5432 -U postgres -C -Fp -b -v –d f_base_160629 -f f_base_160629.txt
```

1.6.3.1.1.4 Резервное копирование с использованием формата tar

Дамп создается в виде архивного файла формата *tar*.

```
pg_dump -h localhost -p 5432 -U postgres -Ft -b -v –d f_base_160629 -f f_base_160629.tar
```

1.6.3.1.1.5 Резервное копирование с использованием конвейеров

Создание резервной копии с последующим сжатием в **gz**.

```
pg_dump -h localhost -O -Fp -c -U postgres f_base_160629| gzip -c > f_base_160629.gz
```

1.6.3.1.2 Резервное копирование с помощью утилиты Pg_DumpAll

Утилита ***pg_dumpall*** позволяет выполнить резервную копию всего кластера базы данных.

```
pg_dumpall > outfile
```

1.6.3.1.3 Резервное копирование с помощью утилиты Pg_BaseBackup

Резервное копирование ***pg_basebackup*** подходит для случаев, когда нужно сделать резервную копию целиком всего кластера БД или настроить *hot standby* реплику (начиная с версии 9.3 входит в PostgreSql). Пример резервирования с ***backup*** сервера в каталог */backup* (каталог должен быть предварительно создан):

```
pg_basebackup -x -h dbserver -U username -D /backup_dir
```

Резервное копирование ***pg_basebackup*** со сжатием:

```
pg_basebackup -D - -Ft | bzip2 > backup.tar.bz2
```

1.6.3.1.4 Список наиболее часто используемых опций

Ниже приведен список наиболее часто используемых опций:

- **-h host** – хост, если не указан то используется *localhost* или значение из переменной окружения *PGHOST*.
- **-p port** – порт, если не указан то используется *5432* или значение из переменной окружения *PGPORT*.
- **-u** – пользователь, если не указан то используется текущий пользователь, также значение можно указать в переменной окружения *PGUSER*.
- **-a, -data-only** – дамп только данных, по-умолчанию сохраняются данные и схема.
- **-b** – включать в дамп большие объекты (*blog'u*).
- **-s**, **-schema-only** – дамп только схемы.
- **-C, -create** – добавляет команду для создания БД.
- **-c** – добавляет команды для удаления (drop) объектов (таблиц, видов и т.д.).
- **-O** – не добавлять команды для установки владельца объекта (таблиц, видов и т.д.).
- **-F, -format {c|d|t|p}** – выходной формат дампа, *custom*, *directory*, *tar*, или *plain text*.
- **-t, -table=TABLE** – указываем определенную таблицу для дампа.

- **-v, -verbose** – вывод подробной информации.
- **-D, -attribute-inserts** – дамп используя команду *INSERT* с списком имен свойств.

1.6.3.2 Резервирование с помощью пакетного файла (скрипта)

Для автоматизации действий, следует использовать планировщик задач и пакетный файл (скрипт). Любую из перечисленных выше команд можно добавить на ежедневное, еженедельное, ежемесячное выполнение с помощью планировщика ***cron***.

CRON – классический планировщик задач в UNIX-подобных операционных системах, использующийся для периодического выполнения заданий в определенное время. Регулярные действия описываются инструкциями, помещенными в файлы *crontab* и в специальные директории. Для этого необходимо ввести команду:

```
# crontab -e
```

Это откроет на редактирование таблицу задач планировщика. Чтобы добавить строку, необходимо нажать клавишу **a** и ввести примерно следующий код:

```
### backup databases ########################################################
00 3  * * * root pg_basebackup -D - -Ft | bzip2 > backup.tar.bz2
```

Для сохранения нажать **Esc**, затем ввести *wq* и нажать **Enter**.

После сохранения, планировщик будет выполнять ежедневное резервирование в 3 часа 00 минут.

1.6.4 Проверка качества созданной резервной копии БД

Вне зависимости от СУБД необходимо раз в месяц (можно и чаще) выполнять проверку качества созданной резервной копии БД, для этого необходимо восстановить дамп. Если во время восстановления возникли ошибки, то ситуация явно вышла из-под контроля и необходимо провести ревизию системы резервного копирования данных.

1.6.4.1 Резервное восстановление СУБД FireBird

Восстановление дампа осуществляется утилитой ***gbak.exe (gbak.sh)***. Пример строки для запуска резервного восстановления дампа приведен ниже:

```
gbak.exe -c -v -user SYSDBA -pas masterkey c:\test\test.bak  c:\test\db\firebird\test_new.fbd
```

Где:

- ***-c*** *– ключ для создания БД.*
- ***-v*** *– вывод на экран лога восстановления БД из дампа.*
- ***-user*** *– пользователь БД.*
- ***-pas*** *– пароль пользователя БД.*
- ***c:\test\test.bak*** *– полный путь дампа.*
- ***c:\test\db\firebird\test_new.fbd*** *– полный путь до восстановленной БД.*

1.6.4.2 Резервное восстановление СУБД Oracle

Для СУБД Oracle можно выделить два варианта восстановления дампа:

- с помощью утилиты ***impdp***;
- с помощью утилиты ***RMAN***.

1.6.4.2.1 Резервное восстановление дампа с помощью утилиты impdp

Восстановление дампа осуществляется утилитой импорта ***impdp***. Пример строки для запуска резервного восстановления дампа приведен ниже:

```
impdp system/Pa$$w0rd SCHEMAS=OracleUser directory=ExportImport
dumpfile=DumpSCHEMAS.dmp logfile=ImportSCHEMAS.log
```

Где:

- ***system/Pa$$w0rd*** *– это логин и пароль пользователя в СУБД;*
- ***SCHEMAS=OracleUser*** *– параметр, который указывает, что мы хотим импортировать конкретную схему (в нашем случае OracleUser);*
- ***directory=ExportImport*** *– параметр указывает директорию, в которой расположен файл дампа данных;*

• ***dumpfile=DumpSCHEMAS.dmp*** *– параметр для указания названия дамп файла;*

• ***logfile=ImportSCHEMAS.log*** *– параметр для указания названия лог файла импорта данных.*

1.6.4.2.2 Резервное восстановление дампа с помощью утилиты RMAN

Соединение с локальной/сетевой БД ранее описано в разделе Запуск RMAN [53].

1. Для восстановления данных целевая БД должна находиться в состоянии NOMOUNT/ MOUNT/ OPEN в зависимости от характера восстановления, например:
   - NOMOUNT – для восстановления контрольных файлов БД (фактически – СУБД).
   - MOUNT – для восстановления БД целиком или табличного пространства SYSTEM.
   - OPEN – для восстановление табличных пространств, помимо SYSTEM (в этом случае перед процедурой восстановления само табличное пространство потребуется перевести в состояние OFFLINE).
2. Восстановление файлов (с данными и служебных) выполняется в ***RMAN*** командой **RESTORE**.

Восстановление данных выполняется либо в ***RMAN***, либо в SQL*Plus командами **RECOVER** при условии наличия восстановленных файлов. Рассмотрим некоторые примеры восстановления:

- Восстановление до момента сбоя («последнего момента»).

Некоторые примеры восстановления:

```
RMAN> RECOVER DATABASE;
RMAN> RECOVER TABLESPACE users;
RMAN> RECOVER DATAFILE 'd:\oracle\oradata\teacher\users01.dbf';
RMAN> RESTORE CONTROLFILE;RMAN> RUN {2> SET ARCHIVELOG DESTINATION TO 'd:
\oracle\oradata\archive';
3> RESTORE ARCHIVELOG ALL; }
```

- Восстановление пространств, закрытых на запись.

```
RMAN> SQL "ALTER TABLESPACE lookup_data OFFLINE";
RMAN> RECOVER TABLESPACE lookup_data;
RMAN> SQL "ALTER TABLESPACE lookup_data ONLINE";
```

- Восстановление до указанного момента в прошлом.

  БД, работающую в режиме архивирования журнала, можно восстанавливать до определенного указанного момента с помощью фраз UNTIL {TIME … | SCN … | SEQUENCE … THREAD…}.

```
Пример. RMAN> RESTORE DATABASE;          # восстановили файлы
RMAN> RECOVER DATABASE UNTIL SCN 375831;    # восстановили БД
RMAN> ALTER DATABASE OPEN RESETLOGS;        # сбросили журнал
```

- Восстановление БД (вторая и третья строчки выше) можно выполнить и в SQL*Plus:

```
SQL > RECOVER DATABASE UNTIL CANCEL;
SQL> ALTER DATABASE OPEN RESETLOGS;
```

  При таком восстановлении необходимо сбросить онлайновый журнал. После этого, как и при традиционном восстановлении со сбросом журналов (RESETLOGS), необходимо снять полную копию БД, так как с этого момента восстановление с более ранних резервных копий станет невозможным из-за того, что история журнальных записей прерывается.

Более подробно с описанием возможностей, функций и команд можно ознакомиться в официальной документации http://www.oracle.com/technetwork/database/features/availability/rman-overview-096633.html.

1.6.4.3 Резервное восстановление СУБД PostgreSQL

Дампы PostgreSQL восстанавливаются с помощью утилиты ***pg_restore***.

В данном разделе приведены основные возможности по созданию дампа. Более подробно можно почитать тут: https://postgrespro.ru/docs/postgrespro/9.5/app-pgrestore

Внимание! *Для форматов* ***Custom***, ***Directory*** *можно и нужно применять восстановление в несколько потоков.*

Перед началом выполнения процедуры восстановления может понадобиться создание БД. Для этого необходимо выполнить команду:

```
CREATE DATABASE newdb
```

1.6.4.3.1 Резервное восстановление дампа с использованием формата «tar»

Пример строки для запуска резервного восстановления дампа приведен ниже:

```
pg_restore -d newdb db.dump
```

Где:

- ***-d*** *– ключ, подключиться к базе данных newdb и восстановить данные непосредственно в неё;*
- ***db.dump*** *– дамп БД.*

1.6.4.3.2 Резервное восстановление дампа с использованием формата «Custom»

Пример строки для запуска резервного восстановления дампа приведен ниже:

```
pg_restore –h localhost -U postgres -Fc -j 4 -e -v -d newdb db.dump
```

Где:

- ***-v*** *– вывод подробной информации;*
- ***-d*** *– ключ, подключиться к базе данных newdb и восстановить данные непосредственно в неё;*
- ***-e*** *– завершать работу в случае возникновения ошибки при выполнении команд SQL в базе данных. По умолчанию процесс восстановления продолжается, а по его окончании выдаётся число ошибок;*
- ***-j*** *– запустить наиболее длительные операции pg_restore (в частности, загрузку данных, создание индексов или ограничений) в нескольких параллельных заданиях. Это позволяет кардинально сократить время восстановления большой базы данных, когда сервер работает на многопроцессорном компьютере. Каждое задание выполняется в отдельном задании или потоке, в зависимости от операционной системы, и использует отдельное подключение к серверу. Оптимальное значение этого параметра зависит от аппаратной конфигурации сервера, клиента и сети. В частности, имеет значение количество*

процессорных ядер и устройство дискового хранилища. В качестве начального значения можно выбрать число ядер на сервере, но и при увеличении этого значения во многих случаях восстановление будет быстрее. Конечно, при слишком больших значениях производительность упадет из-за перегрузки;

*• **db.dump** – дамп БД.*

1.6.4.3.3 Резервное восстановление дампа с использованием формата «Directory»

Пример строки для запуска резервного восстановления дампа приведен ниже:

```
pg_restore –h localhost -U postgres -Fd -j 4 -e -v -d newdb db.dump
```

Где:

• -v – вывод подробной информации.

*• -d – ключ, подключиться к базе данных **newdb** и восстановить данные непосредственно в неё.*

• -e – завершать работу в случае возникновения ошибки при выполнении команд SQL в базе данных. По умолчанию процесс восстановления продолжается, а по его окончании выдаётся число ошибок.

*• -j – запустить наиболее длительные операции **pg_restore** (в частности, загрузку данных, создание индексов или ограничений) в нескольких параллельных заданиях. Это позволяет кардинально сократить время восстановления большой базы данных, когда сервер работает на многопроцессорном компьютере. Каждое задание выполняется в отдельном задании или потоке, в зависимости от операционной системы, и использует отдельное подключение к серверу. Оптимальное значение этого параметра зависит от аппаратной конфигурации сервера, клиента и сети. В частности, имеет значение количество процессорных ядер и устройство дискового хранилища. В качестве начального значения можно выбрать число ядер на сервере, но и при увеличении этого значения во многих случаях восстановление будет быстрее. Конечно, при слишком больших значениях производительность упадёт из-за перегрузки.*

• db.dump – дамп БД.

2

Регламентно-профилактические мероприятия по СУБД

Регламентно-профилактические мероприятия по СУБД включают в себя следующие действия, выполняемые с определенной периодичностью:

Таблица 1 – Действия, выполняемые в рамках регламентно-профилактических мероприятий по СУБД

№	Мероприятия	Периодичность выполнения
1.	Сбор статистики СУБД.	Ежедневно.
2.	Сохранение конфигурационных файлов (Для Oracle – **init.ora**, **listener.ora**, **tnsnames.ora**. Для PG - **postgresql.conf**, **pg_hba.conf**) на физическом отдельном диске.	Периодически и перед выполнением обновления/реконфигурации сборки/серверов/гипервизоров/СХД.
3.	Резервное копирование БД	Полная резервная копия СУБД – 1 раз в неделю. Инкрементальная копия – раз в день. Для Oracle\PG – возможно использование связки master-slave; Для СУБД Firebird – ежедневно; Для СУБД Oracle: • архивирование журнала транзакции для восстановления на момент времени; • полная резервная копия СУБД (full backup) – 1 раз в неделю; • инкрементальная LVL0 – раз в неделю. Инкрементальная копия LVL1 CUMULATIV после создания LVL0 – раз в день.
4.	Проверка резервных копий БД – контрольное восстановление из резервной копии БД.	1 раз в неделю.
5.	Мониторинг логов СУБД на наличие ошибок.	Постоянно.
6.	Проверка наличия невалидных объектов\отключенных триггеров, констрейнтов в БД.	Ежедневно.
7.	Мониторинг количества активных сессии, блокированных сессий.	Постоянно.
8.	Мониторинг дискового пространства, загрузки CPU, RAM.	Постоянно.
9.	Мониторинг наполненности таблицы RPLLOG. Выполнение запроса: *select 'all', count(1) from rpllog* *union all* *select 'nines', count(1) from rpllog where generation = 999999999999999;*	Периодически, несколько раз в день.
10.	Сверка с эталонной БД.	Периодически: 1 раз в квартал.
11.	Проверка полноты применения sql-скриптов после обновления.	после выполнения процедуры обновления сборки системы.

2.1 Рекомендации по параметрам СУБД

2.1.1 СУБД ORACLE

Распределение памяти в пропорции: SGA – 90%, PGA – 10%.

SGA (system global area) – это часть общей памяти, которую разделяют между собой все серверные процесс (включая фоновые).

PGA – специфичная для процессов часть памяти известна как программная глобальная область (program global area).

Необходимо выполнить:

```
SQL>alter system set optimizer_index_cost_adj=1 scope=spfile sid='*';
SQL>alter system set open_cursors=5000 scope=spfile sid='*';
SQL>alter system set undo_retention=10800 scope=spfile sid='*';
SQL>alter system set db_securefile=always scope=spfile sid='*';
SQL>alter system set audit_trail=none scope=spfile sid='*';
SQL>alter system set db_ultra_safe=data_only scope=spfile sid='*';
SQL>alter system set recyclebin=off scope=spfile sid='*';
SQL>alter system set deferred_segment_creation = false scope=spfile sid='*';
```

2.1.2 СУБД PostgreSQL

Начальные параметры конфигурации СУБД PostgreSQL следующие:

1. **max_connections** должно быть менее *1000*.
2. **maintenance_work_mem** – рекомендуется устанавливать значение 50-75% от размера самой большой таблицы или индекса, но чтобы памяти хватило для работы системы и приложений. Рекомендуется устанавливать значения больше чем **work_mem**.
3. **shared_buffers** – рекомендуемое начальное значение – 25% от объёма памяти.
4. **work_mem** – рекомендуемое начальное значение в 2-4% объема оперативной памяти.
5. **max_worker_process** – задается равное количеству процессоров сервера СУБД.
6. **random_page_cost**=*1.2*.
7. **seq_page_cost**=*1.2*.
8. **effective_cache_size** – для начальной установки рекомендуется следующие размеры: не более

1/3 оперативной памяти, далее можно увеличивать пока хватает дискового пространства (800ГБ и более).

9. **default_statistics_target** =*300*.
10. **autovacuum_vacuum_scale_factor** = *0.01*.
11. **autovacuum_analyze_scale_factor** = *0.005*.
12. **vacuum_cost_limit**=*1000*.
13. **autovacuum_vacuum_cost_limit**=*1000*.
14. **log_line_prefix=%m [%p]: [%l-1] [%v] [%x] user=%u,db=%d,client=%h**.
15. **superuser_reserved_connections**=*3*.
16. **max_locks_per_transaction** = *256*.
17. **max_pred_locks_per_transaction** = *2000*.
18. **idle_in_transaction_session_timeout** =*300* .

2.1.3 СУБД Firebird

Начальные параметры конфигурации для данного вида СУБД рекомендуется использовать установленные по умолчанию. В дальнейшем, по результатам мониторинга за состоянием работы СУБД возможно использование оптимизации параметров.

***Примечание**. Оптимизированные файлы конфигурации под определенную версию ОС, СУБД разрядность СУБД, выделенную память и пр. располагаются на сайте https://ib-aid.com/en/optimized-firebird-configuration/.*

НАШИ КОНТАКТЫ

Звоните:

(495) 784-70-00

Пишите:

bft@bftcom.com

Будьте с нами online:

www.bftcom.com

Приезжайте:

129085, г. Москва,
ул. Годовикова, д. 9, стр. 17

Дружите с нами в социальных сетях:

vk.com/bftcom

t.me/ExpertBFT_bot