

Программный модуль «Работа с СУБД Oracle»

НПО «Криста» 2015

Содержание

1. Общая информация о работе АС «Бюджет» с базой данных под управлением СУБД Oracle	3
1.1. Назначение	3
1.2. Цели внедрения	3
1.3. Функциональные возможности	3
1.4. Отличительные особенности	4
2. Общие сведения и администрирование базы данных Oracle.....	4
2.1. Основные термины и понятия, связанные с БД Oracle	4
2.2. Основные файлы БД Oracle	4
2.3. Пояснения по используемой терминологии: База данных, Пользователь, Схема	6
2.4. Экземпляр Oracle и этапы запуска БД	7
2.5. Этапы восстановления файлов данных.....	8
2.6. Обслуживание БД Oracle	8
3. Установке и настройке СУБД Oracle.....	10
3.1. Рекомендуемый порядок действий, выполняемых при установке и настройке СУБД Oracle.....	10
3.2. Установка сервера Oracle	11
3.3. Создание Listener	17
3.4. Создание БД	19
3.5. Порядок действий после создания БД	33
4. Регистрация пользователей	35
4.1. Пользователи, права и роли	35
4.2. Регистрация пользователей в СУБД Oracle.....	36
4.3. Регистрация пользователей СУБД Oracle в АС «Бюджет».....	39
5. Миграция с Firebird	40
5.1. Перенос данных из БД Firebird в БД Oracle	40
5.2. Перенос логинов пользователей с Interbase (Firebird) на Oracle.....	45
6. Решение типовых задач по администрированию БД Oracle	45
6.1. Использование консоли для задания режима автоматического расширения.....	46
6.2. Кэширование таблиц Oracle.....	46
6.3. Резервное копирование базы данных.....	47
6.4. Получение данных на момент времени в прошлом, аудит изменений	52
6.5. Перенос пользователей из одной БД Oracle в другую.....	53
6.6. Запрет/разрешение подключений к базе данных, ограничение сетевого доступа	54
7. Утилита OraTools.....	57
7.1. Запуск программы и подключение к БД.....	57
7.2. Структура окна программы и принципы работы	59
7.3. Закладка «Импорт схемы»	64
7.4. Закладка «Экспорт схемы»	68
7.5. Закладка «Работа с пользователями».....	70
7.6. Закладка «DataPump»	71
7.7. Настройка программы OraTools.exe.....	73
8. Перечень возможных ошибок, возникающих при работе с Oracle	73

1. Общая информация о работе АС «Бюджет» с базой данных под управлением СУБД Oracle

1.1. Назначение

ПМ «Работа с СУБД Oracle» реализует возможность использования в качестве хранилища данных АС «Бюджет» СУБД Oracle, что позволяет в дальнейшем наращивать функциональность системы без потери ее производительности и снижения надежности.

1.2. Цели внедрения

- Возможность одновременной работы большого числа пользователей с большими объемами данных без существенного снижения производительности.
- Более широкое применение автоматизации в процессах планирования и исполнения бюджета и расширение объема хранимой и обрабатываемой информации (учет новых блоков информации, хранение и обработка изображений документов) без падения мощности системы.
- Повышение масштабируемости системы.
- Увеличение надежности функционирования системы.

1.3. Функциональные возможности

Программный модуль обеспечивает возможность работы АС «Бюджет» с базой данных под управлением СУБД Oracle с использованием всех ее преимуществ:

- Повышение производительности системы за счет возможности использования больших распределенных дисковых массивов, лучшего использования ресурсов сервера и более качественного планировщика запросов.
- Повышение надежности системы за счет большей устойчивости сервера, нескольких режимов сохранения резервных копий БД, возможности ведения журналов транзакций, возможности использования резервных серверов для быстрого восстановления работы в случае выхода из строя основного.
- Широкие возможности по оптимизации и кэшированию, в том числе материализованные виды и параллельные запросы.
- Возможность практического использования журналирования операций для анализа изменений данных БД за счет Oracle Flashback Query и LogMiner.
- Повышение масштабируемости системы за счет больших функциональных возможностей СУБД (кластерные решения, лучшее использование многопроцессорных систем и больших объемов оперативной памяти).
- Использование различных платформ для развертывания сервера БД (различные варианты Windows, Unix, Linux), в том числе и для 64-бит архитектуры.
- Широкие возможности по администрированию, за счет системы ролей, более развитой внутренней инфраструктуры, доступной для PL\SQL, а так же встроенной возможности выполнять Java-код.
- Полнотекстовый поиск по БД с учетом русского языка.

1.4. Отличительные особенности

- Реализован автоматический перенос данных из БД Interbase (Firebird) при переходе на СУБД Oracle.
- Выполнение наиболее частых задач администрирования облегчается наличием специализированной утилиты, позволяющей осуществлять следующие операции: резервное копирование, пересчет статистики, поднятие схемы, добавление нового пользователя и т.д.

2. Общие сведения и администрирование базы данных Oracle

Данное описание является лишь обзором некоторых моментов работы с СУБД Oracle, которые представляются разработчикам АС «Бюджет» наиболее важными при первом знакомстве с этой СУБД, поэтому для полноты восприятия информации рекомендуем читать официальную документацию по СУБД Oracle.

2.1. Основные термины и понятия, связанные с БД Oracle

ЧЕКПОИНТ (контрольная точка) – это SCN, записываемый в файлы данных. Для ускорения работы изменения вносятся в файлы данных не сразу, а в определенные моменты времени. Контрольная точка отражает момент времени (номер операции), в который все блоки, измененные до него, были записаны в файлы данных. Таким образом, в файле данных могут быть блоки, измененные после контрольной точки, но не может быть незаписанных изменений, сделанных до контрольной точки.

SCN (system change number) – порядковый номер операции, вносящей изменения в БД (в данные или в метаданные), т.н. системный номер контрольной точки.

Блок данных – минимальная единица хранения данных на физическом уровне. Каждая таблица состоит из набора блоков. Меняя одну запись на логическом уровне, мы меняем весь блок данных на физическом уровне (так как меньше чем блок мы записать не можем). Запись также может храниться в нескольких блоках.

Бэкап (англ. Backup) – резервная копия БД, предназначенная для восстановления данных в оригинальном месте их расположения в случае их повреждения или разрушения.

Дамп – файл, содержащий информацию о состоянии системы или ее части.



Понятия дампа и бэкапа следует четко разделять, т.к. в СУБД Oracle этими терминами именуется принципиально различные способы резервирования БД.

Патч – программное средство, используемое для устранения проблем в программном обеспечении или изменения его функционала (исправление ошибок, изменение внешнего вида, улучшение производительности программ, другие изменения, сделанные разработчиком). Исправление может применяться к уже установленной программе, либо к ее исходным кодам.

2.2. Основные файлы БД Oracle

База данных под управлением Oracle состоит из файлов различного назначения, основными

из которых являются следующие:

- Инициализационный файл (pfile или spfile);
- Файл паролей;
- Управляющий файл (control-файл);
- Файлы табличных пространств: системное табличное пространство (SYSTEM), табличные пространства с пользовательскими данными (USERS, INDX и т.п.), табличное пространство UNDO, табличное пространство TEMP;
- Файлы логов (redo-логи);
- Архивные файлы логов.

Инициализационный файл представляет собой файл, как правило, с расширением *.ini. Он бывает двух видов: **pfile** – доступен для редактирования вручную и не может меняться командами sql; **spfile** – можно править командами sql, но не рекомендуется его править вручную, т.к. он имеет бинарный формат. При запуске БД (точнее, при запуске экземпляра БД) первым считывается инициализационный файл, обычно он размещен в папке <ORACLE_HOME>/database (в Windows), или в <ORACLE_HOME>/dbs (в Linux) и называется **init<SID>.ora** или **spfile<SID>.ora** (если инициализационный файл является бинарным). Инициализационный файл содержит:

- имя базы данных и имя экземпляра БД;
- пути к каталогам с логами процесса (параметры <*>_dump_dest) – текстовые файлы, в которые экземпляр делает записи о наиболее важных действиях, а также пишет сообщения об ошибках;
- пути к control-файлам (параметр control_files);
- путь к каталогу (каталогам) для архивирования логов (параметр log_archive_dest, или log_archive_dest_<n>, или db_recovery_file_dest);
- имя табличного пространства UNDO (undo_tablespace);
- прочие настройки БД и экземпляра (память, язык и форматы, оптимизация производительности, **shared server** и т.п.).

Если используется **pfile**, то он может содержать ссылку (параметр IFILE или SPFILE) на другой файл с описанием параметров.

Файл паролей. При подключении к БД проверка логинов осуществляется за счет данных, хранимых в системной таблице (обратиться к ней можно через view dba_users). Однако в ряде случаев требуется работать с экземпляром, который еще не подключен к БД (например, при ее восстановлении из резервной копии (бэкапа)). В этом случае используется файл паролей. В нем хранятся пароли пользователей SYS и INTERNAL (также в файл паролей могут быть добавлены и другие логины). Указанием на то, что логин нужно проверять по файлу паролей, является тип подключения SYSDBA или SYSOPER (например, **connect** «SYS/password@database as SYSDBA»).

Управляющий файл содержит следующую информацию:

- DBID – уникальный идентификатор БД (число);
- местоположение всех файлов БД (табличные пространства, логи);
- местоположение архивных логов;
- может использоваться в качестве «каталога восстановления» и, следовательно, содержать информацию о всех бэкапах, выполненных при помощи RMAN;
- информацию о контрольных точках и значение последнего SCN.

Системное табличное пространство (SYSTEM) содержит данные о структуре БД (метаданные). На логическом уровне данные о структуре БД хранятся в схемах SYS, SYSTEM (эти схемы физически размещены в табличном пространстве SYSTEM).

Табличные пространства с пользовательскими данными (USERS, INDX и т.п.) содержат пользовательские данные.

Табличное пространство UNDO хранит измененные блоки данных, для отката транзакции и получения данных на момент времени в прошлом (например, при открытии обычной транзакции, мы получаем данные на момент ее начала, и не видим изменений, сделанных другими пользователями). Таким образом, если мы поменяли некоторые данные, то измененные блоки данных будут записаны в файлы данных, а прежние версии этих блоков будут записаны в UNDO. Данные в UNDO хранятся определенное время, указанное в инициализационном файле, после чего могут быть затерты. Если места для хранения данных в UNDO не хватит, БД выдаст ошибку и приостановит свою работу.

Табличное пространство TEMP используется для сортировок и аналогичных операций, требующих значительных объемов места на непродолжительный период времени. Если Oracle не хватает места для сортировки данных в памяти, то он выполняет ее на диске в табличном пространстве TEMP. Табличное пространство TEMP может отсутствовать, тогда место для сортировок выделяется в табличном пространстве SYSTEM (но данный режим работы не рекомендуется).

Файлы логов (redo-логи) – это файлы, в которые записывается информация обо всех операциях, выполняемых над БД, причем запись об операции производится до внесения реальных изменений в данные (так называемая журнализация). Как правило, используется три группы redo-логов. Одна из групп является текущей (в нее ведется запись). Когда место в текущей группе заканчивается, экземпляр переключается на следующую группу, и так по кругу (старые данные при этом перезаписываются). В нормальном режиме работы требуются только логи с момента последней контрольной точки (для восстановления после внезапной остановки (например, свет выключили)), все остальные данные уже записаны в файлы табличных пространств. Для целей восстановления БД из бэкапа могут потребоваться данные и о более старых операциях.

Архивные файлы логов – если БД работает в режиме **ARCHIVELOG** (задается в инициализационном файле), то redo-логи перед перезаписью копируются в указанное место. Данный процесс называется архивированием. Архивные файлы логов используются для восстановления БД из резервной копии (бэкапа). Существует средство **LogMiner**, позволяющее просматривать лог-файлы (оперативные и архивные), так что администратор может отследить операции, которые выполнялись над БД.

2.3. Пояснения по используемой терминологии: База данных, Пользователь, Схема

В первом приближении **БД** – это просто набор файлов, а **СУБД** – это программа, которая позволяет с ней взаимодействовать (например, с помощью SQL-запросов).

ANSI-стандарт предусматривает, что в БД будут храниться различные объекты (таблицы, триггера, индексы, и прочее), с которыми будут работать **пользователи** (user). Каждый пользователь имеет имя (уникальное в пределах данной БД) и пароль. Кроме просмотра и изменения данных, пользователь может создавать новые объекты (если у него есть соответствующие права). По умолчанию **владельцем** (owner) создаваемых объектов является пользователь, который их создал. А все множество объектов, принадлежащих некоторому пользователю, называется **схемой** (schema) данного пользователя.

Пользователь может не иметь схемы (если он не владеет ни одним объектом), но схема без пользователя существовать не может. При работе с БД пользователь и схема – это практически одно и то же. А вот с точки зрения обычного человека это абсолютные разные вещи.

Разработчики ANSI стандарта учли это и добавили возможность создавать объекты в схеме другого пользователя. Т.е. пользователь, подключившийся как VASIA, по умолчанию создает объекты в схеме VASIA, но он также может создать объект в схеме PETIA (для этого ему надо будет указать на данный факт в скрипте создания).

В СУБД Interbase (Firebird) база данных всегда содержит только одну схему, которая принадлежит SYSDBA (он же администратор БД). Когда пользователь создает новый объект, ему не надо указывать, в какой схеме он хочет его создать, т.к. схема в СУБД Interbase (Firebird) только одна. Это приводит к тому, что люди, привыкшие к Interbase (Firebird), зачастую не видят разницы между БД и схемой.

ORACLE – более прогрессивная СУБД, поддерживающая несколько схем в одной БД и более сложную организацию самой БД: в Interbase (Firebird) БД часто состоит из одного gdb файла, а в Oracle даже простая БД состоит из десятка (а то и более) различных файлов. Поэтому **несколько** схем Interbase (Firebird), размещенных в нескольких БД, могут располагаться в разных схемах **одной** БД Oracle. Создание БД Oracle также является более сложной операцией и занимает гораздо больше времени.

2.4. Экземпляр Oracle и этапы запуска БД

Работа с БД Oracle выполняется при помощи специальной программы, обслуживающей файлы БД. В терминах Oracle такая программа называется экземпляром (instance).

В операционной системе Windows экземпляр выглядит как служба, и им можно управлять так же, как и обычными службами. Например, службу можно остановить — это приведет к тому, что БД станет недоступной (следует отметить, что остановка экземпляра вручную приводит к остановке экземпляра в аварийном режиме, и потому не следует злоупотреблять этой возможностью). Имя службы чаще всего бывает OracleService<SID> (где <SID> – это имя БД).

Кроме экземпляра для нормальной работы с Oracle в Windows требуется служба **Listener** (часто она называется OracleOraHome92TNSListener или OracleOraDb10g_home1TNSListener, хотя у нее могут быть и другие названия). Данная служба используется для подключения к экземпляру Oracle (один Listener может подключать пользователей к нескольким экземплярам Oracle). Для работы с экземпляром Listener не нужен. Например, если остановить службу Listener-а, то это приведет к тому, что подключиться к БД будет нельзя, но все подключенные пользователи продолжат работу. По умолчанию Listener использует порт 1521 (но при его настройке можно указать и другое значение). Порт Listener должен быть открыт в межсетевом экране (firewall), иначе Вам не удастся подключиться к Oracle. В 10-ой версии Oracle вполне достаточно открыть только порт Listener-а. В 9-ой версии Oracle каждое подключение (после того, как Listener связал пользователя и экземпляр), по умолчанию, использует свой порт (такой режим работы может быть изменен, см. параметр USE_SHARED_SOCKET).

Нужно уточнить, что локально подключиться к экземпляру можно без Listener-а. Такой способ подключения используется в служебных целях (например, для перезапуска экземпляра). Следует помнить, что в этом случае аутентификация выполняется либо с использованием файла паролей, либо средствами операционной системы (пользователь должен быть членом группы ORA_DBA в Windows, или dba в Linux).

Если сервер Oracle работает в ОС Linux (или *nix), то в его работе (по сравнению с ОС Windows) есть ряд отличий. В Linux также есть служба Oracle, однако под этим именем скрывается скрипт, который запускает и экземпляр, и Listener. Вторым отличием является то, что в Windows экземпляр – это один процесс, в рамках которого работает множество нитей, обслуживающих клиентские запросы, а в Linux экземпляр — это множество процессов. Чаще всего они называются oracle. Кроме них, присутствует также процесс Listener-а (называется чаще всего tnslnsr). Третьим отличием является то, что в Linux запуск экземпляра осуществляется при помощи утилиты **sqlplus**, а в Windows при остановленной службе утилита **sqlplus** не может подключиться к экземпляру (и не может его запустить).

Итак, экземпляр Oracle может быть остановлен или запущен. При этом важно помнить, что экземпляр имеет несколько состояний. Поэтому, если у Вас запущен экземпляр (и он нормально работает), то это еще не значит, что есть доступ к БД (поскольку наличие доступа определяется тем, в каком состоянии сейчас находится экземпляр). Всего состояний у запущенного экземпляра четыре. Первое состояние — это простаивающий экземпляр (так называется экземпляр, который запущен, но пока не обслуживает БД). В Windows к простаивающему экземпляру можно подключиться с помощью стандартных утилит **sqlplus** и **sqlplusw**. В Linux с помощью утилиты **sqlplus** (утилиты **sqlplusw** в Linux нет) можно не только подключиться к простаивающему экземпляру, но и запустить сам экземпляр.

Процесс запуска БД разделяется на несколько этапов, которые выполняются последовательно. По завершении каждого этапа экземпляр переходит в новое состояние. Далее (в порядке запуска) перечислены эти состояния:

1. **nomount** – для перехода в это состояние нужен инициализационный файл. Следует отметить, что для успешного запуска требуется наличие на диске каталогов для логов. Если эти каталоги не существуют, то запуск завершится сообщением об ошибке. Возможен запуск без инициализационного файла (при помощи утилиты **RMAN**).
2. **mount** – для перехода в это состояние требуется наличие control-файла. После его открытия становится доступной информация о расположении файлов с данными. На этапах **nomount** и **mount** возможна выборка данных только из фиксированных таблиц. В этих таблицах можно увидеть служебную информацию, которая физически хранится в инициализационном файле, control-файле, а также в виртуальной памяти экземпляра.
3. **open** – база данных полностью открыта. Возможно выполнение всех стандартных SQL-запросов.

2.5. Этапы восстановления файлов данных

1. Восстановление файлов данных из бэкапа. После восстановления файлов в БД обязательно должны быть табличные пространства SYSTEM и UNDO (восстановление возможно и без UNDO, но в таком случае данные будут не согласованы).
2. Накат логов, которые имели место после записанной в файлы контрольной точки. Также производится накат логов и в пространстве UNDO. Логи накатываются не все, а только до определенного момента времени (это может использоваться для восстановления БД в результате ее некорректного редактирования).
3. Открывается БД, при этом производится откат незафиксированных транзакций, старые блоки данных подтянутся из пространства UNDO.

2.6. Обслуживание БД Oracle

Обслуживание базы данных состоит в систематическом выполнении рекомендуемых процедур:

1. Регулярный сбор статистики для стоимостного оптимизатора (например, каждую ночь). В 10-ой (и последующих) версии по умолчанию настроен автоматический сбор статистики для всех схем – так что там ничего настраивать не нужно. В 9-ой версии для сбора статистики рекомендуется использовать скрипт **dbms_stats.sql**. Для его запуска следует подключиться к БД под именем владельца схемы. Не рекомендуется собирать статистику для системных пользователей (SYS, SYSTEM и других), т.к. это может привести к замедлению работы СУБД. Для выполнения скрипта можно использовать утилиту **sqlplus**.

Пример запуска скрипта сбора статистики, при помощи утилиты **sqlplus**:

```
sqlplus BUDGET_2006/password@budserv @dbms_stats.sql.
```

Пример скрипта **dbms_stats.sql** приведен ниже:

```
spool dbms_stats.lst
ALTER SESSION SET NLS_NUMERIC_CHARACTERS=',.';
/
BEGIN
  DBMS_STATS.GATHER_SCHEMA_STATS(user, null, false, 'FOR ALL COLUMNS SIZE AUTO',
    cascade=>TRUE);
END;
/
SPOOL OFF
exit
/
```

Вместо одной команды `dbms_stats.gather_schema_stats` можно указать несколько (рекомендуется их писать в разных блоках BEGIN/END). Вместо `user` (первый параметр), в этом случае нужно написать имена схем (см. ниже). Если схема имеет большой размер, то можно собирать не точную, а оценочную статистику: вместо `null` (второй параметр) рекомендуется указать `dbms_stats.auto_sample_size`, или указать процент выборки – от 1 до 100.

```
dbms_stats.gather_schema_stats ('BUDGET2009', dbms_stats.auto_sample_size, false, 'FOR ALL COLUMNS
SIZE AUTO', cascade=>TRUE);
dbms_stats.gather_schema_stats ('CRYPTO2009', null, false, 'FOR ALL COLUMNS SIZE AUTO',
cascade=>TRUE).
```

2. Периодическое перестроение индексов (`ALTER INDEX <имя> REBUILD`). Балансировка индексного дерева ускоряет поиск, однако в большинстве случаев перестроение индексов не дает заметного ускорения, т.к. индексные деревья редко бывают сильно разбалансированы. Для перестроения индексов можно использовать **OraTools**.
3. Для повышения производительности в схемах большого объема рекомендуется изменять политику кэширования отдельных таблиц схемы АС «Бюджет» (смотрите п. 6.2 «Кэширование таблиц Oracle» раздела 6 «Решение типовых задач по администрированию БД Oracle»).
4. Рекомендуется регулярно проверять отсутствие в БД отключенных и инвалидных объектов (такие объекты могут появиться после выполнения DDL скриптов). Для этих целей также можно использовать **OraTools**.
5. Крайне рекомендуется регулярное выполнение резервирования БД (смотрите п. 6.3 «Резервное копирование базы данных» раздела 6 «Решение типовых задач по администрированию БД Oracle»).

6. При переходе на новый год рекомендуется переносить данные в пустую (стандартную) схему. Например, в СУБД Oracle 10-ой версии есть следующая проблема: если взять дампы схемы прошлого года, удалить оттуда данные и закатать новые через **ExprMove.exe**, то место, выделенное под таблицы, не уменьшится. Поэтому, в ряде случаев при поиске данных (а их в начале года будет немного) СУБД будет искать по всем блокам (в том числе и пустым), следовательно, БД будет работать медленнее, хотя к концу года ситуация выровняется. Сократить размер места, занятого под таблицу, в 10-ой версии можно за счет операции **shrink**, но это дополнительные действия.

Для выполнения ряда операций (в частности, для выполнения скриптов обновления) можно использовать бесплатные утилиты **SqlTools** и **SqlTools++**. Также, для исполнения скриптов можно использовать бесплатную утилиту **SqlDeveloper**, разработанную фирмой Oracle и размещенную на ее сайте.

3. Установке и настройке СУБД Oracle

Следует отметить, что АС «Бюджет» поддерживает работу с Oracle 11.2.0.2 и выше (версии 11.1.x и 11.2.0.1 не поддерживаются).

Для реальной работы (а не тестовой эксплуатации) крайне рекомендуется использовать 64-х разрядную операционную систему и 64-х разрядную версию СУБД. Также при установке Oracle следует помнить, что антивирус может серьезно влиять на скорость работы с дисковой подсистемой.

В данном руководстве рассмотрен только один из вариантов установки сервера СУБД (один вариант настроек). Данный вариант является наиболее общим и рекомендуется использовать в тех случаях, когда у Вас нет особого опыта в работе с СУБД Oracle. Существует множество способов установки сервера Oracle, их подробное описание выходит за рамки данного руководства, поэтому если Вас интересуют какие-то варианты установки или опции, не рассмотренные в данном руководстве, то Вам следует обратиться к документации по серверу Oracle.

3.1. Рекомендуемый порядок действий, выполняемых при установке и настройке СУБД Oracle

При установке следует помнить, что инсталлятор Oracle не воспринимает **пути с русскими буквами, пробелами** и прочими нестандартными символами.

Также обратите внимание на особенности работы клиентского ПО: приложения, работающие с Oracle (в частности АС «Бюджет»), не должны располагаться в папках, в имени которых есть **круглые скобки** (иначе они не смогут подключиться к Oracle).

Еще один важный момент, который многие упускают из вида при **установке патча**: если патч ставится **до** создания БД (как это прописано в инструкции), то БД обновлять не надо, поскольку ее еще нет. Если же установка патча выполняется **после** создания БД, то ее тоже следует обновлять, выполняя скрипты и производя все действия, которые прописаны в инструкции по установке патча.



Если Oracle уже установлен, и нужно пересоздать базу, то сначала воспользуйтесь утилитой **Database Configuration Assistant** для удаления базы, а затем создайте новую. Проверьте, чтобы файлы с данными также были удалены, при необходимости удалите их вручную. Перед удалением

просмотрите, какие файлы принадлежат данной БД.

Ниже приводится рекомендуемый порядок действий, которые выполняются при установке СУБД Oracle:

1. Установите сервер Oracle.
2. Установите клиент Oracle (таким образом, на сервере будут доступны дополнительные утилиты для работы с СУБД). Лучше устанавливать клиента в тот же HOME (при инсталляции, как правило, будет возможность выбора – главное не пропустить вопрос), т.к. в каждом HOME свои настройки, поэтому, если их несколько, то будет легко запутаться.
3. Поставьте патч, если это необходимо.
4. Создайте БД (при помощи утилиты **Database Configuration Assistant**).
5. Создайте службу Listener (например, с помощью **Net Configuration Assistant**). После создания листнера настройте статическую привязку листнера к БД (если он сам ее динамически не нашел). Привязку настраивайте в **Net Manager** (в свойствах службы Listener).
6. Настройте файлы **tnsnames.ora** (описание подключений) и **sqlnet.ora** (общие настройки клиента). Используйте для этого **Net Manager**.



В ряде случаев возникает проблема подключения к серверу Oracle, связанная с тем, что клиент и сервер Oracle установлены на компьютерах, не являющихся членами одного домена, или в случаях, когда запуск клиента Oracle производится от имени локального пользователя (так часто запускают сервер АС «УРМ»). Для устранения проблемы в файле **sqlnet.ora**, расположенном на клиенте Oracle, должна быть строка `SQLNET.AUTHENTICATION_SERVICES=(NONE)`. Если на сервере установлена ОС Windows, то там лучше оставить `SQLNET.AUTHENTICATION_SERVICES= (NTS)`, т.к. это позволит подключаться /as SYSDBA.

7. Поставьте клиентов и распределите по ним файлы **tnsnames.ora** и **sqlnet.ora** (скопировать эти файлы будет быстрее, чем настраивать их на каждом клиентском компьютере).
8. Установите схемы АС «Бюджет».

3.2. Установка сервера Oracle

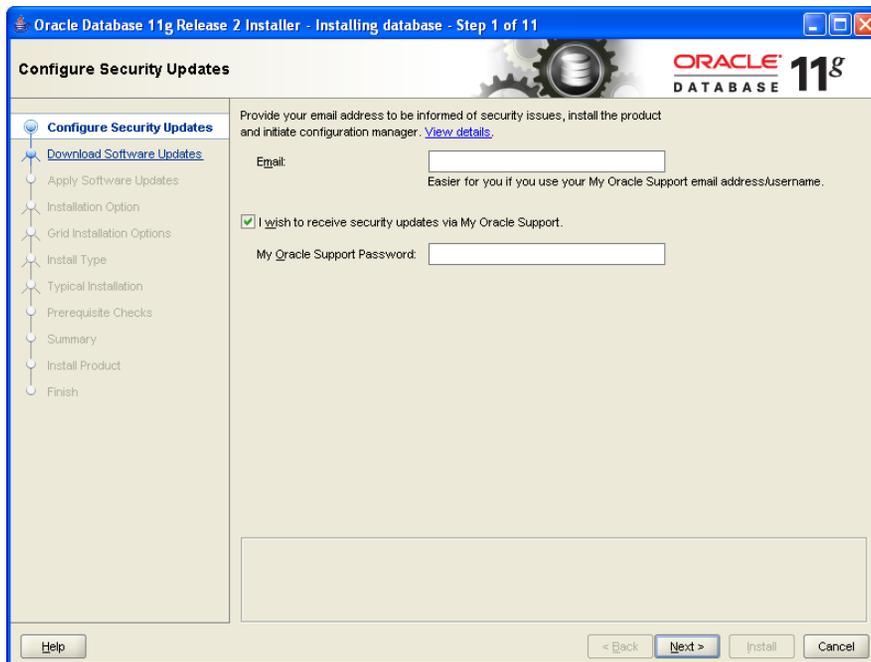
Процесс установки и дальнейшей настройки Oracle рассмотрен на примере версии 11.2.0.2.

Дистрибутив Oracle 11.2.0.2 разбит на несколько CD-дисков – первые два это дистрибутив сервера. Перед установкой нужно взять папки database и слить их вместе (например, из второго диска берем папку database и копируем поверх папки database из первого диска). Для установки Oracle нужно запустить инсталлятор (как правило, это файл setup.exe в папке database).

Запускать установку Oracle лучше из-под пользователя с английским именем, т.к. распаковка временных файлов выполняется в каталог с именем пользователя, а Oracle не всегда адекватно реагирует на папки с русскими именами (этот момент также следует помнить при выборе пути установки).

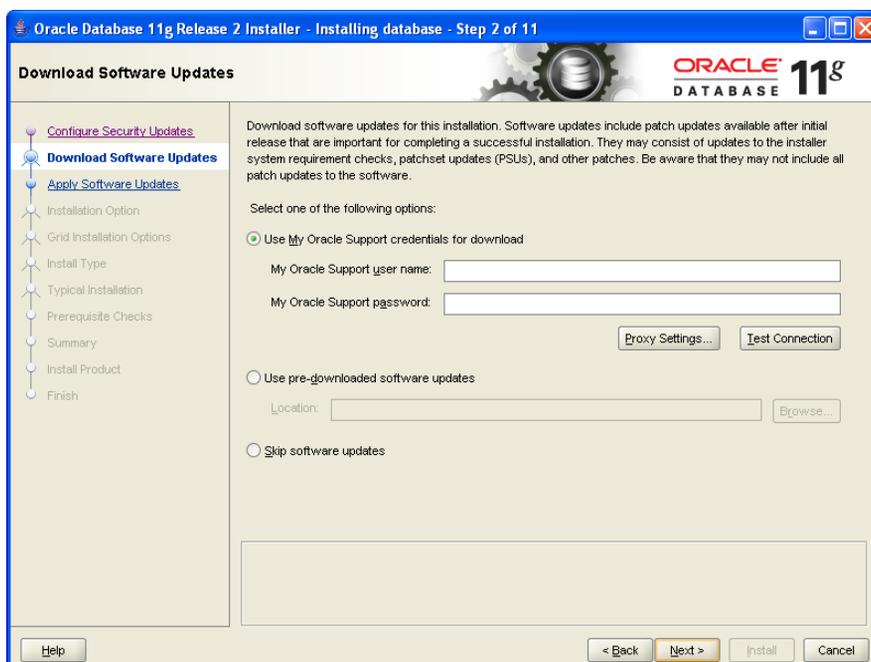
1. На первом шаге нужно указать данные для получения по почте информации технической поддержки Oracle – email и пароль для технической поддержки. Если вы их не укажете, то будет задан вопрос «Уверены ли Вы в том, что не хотите получать уведомления?» – отвечаем утвердительно и идем дальше.

Рисунок 1 – Диалоговое окно с параметрами установки Oracle (шаг 1)



2. На втором шаге Вам предлагают скачать обновления с сайта технической поддержки Oracle или использовать уже закаченные. Но можно просто выбрать Skip software updates и двигаться далее. Как правило, обновления можно не закачивать, поэтому один шаг (их применение) будет пропущен (хотя на следующей картинке и будет написано, что шаг 3, но это будет 3-ий шаг уже из 10, а не из 11 шагов, как на этой). Поэтому если Вы закачивали обновления, то нумерация шагов будет немного отличаться.

Рисунок 2 – Диалоговое окно с параметрами установки Oracle (шаг 2)



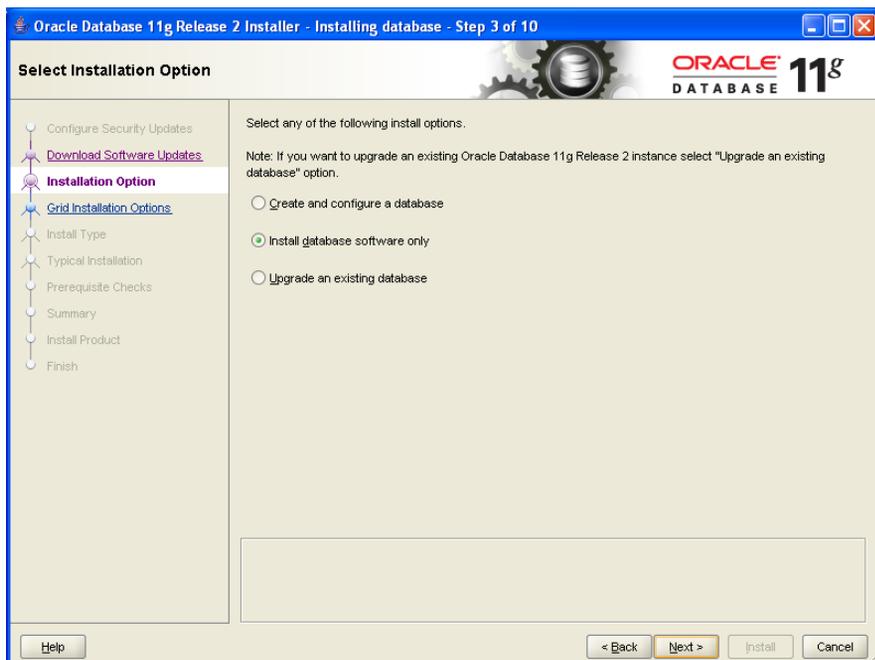
3. Далее нам предлагают выбрать вариант установки:

- ♦ можно установить СУБД и сразу создать базу данных;

- ◆ можно установить СУБД (но не создавать БД);
- ◆ можно обновить существующую БД.

Выбираем вариант установки без создания БД (БД мы создадим позже).

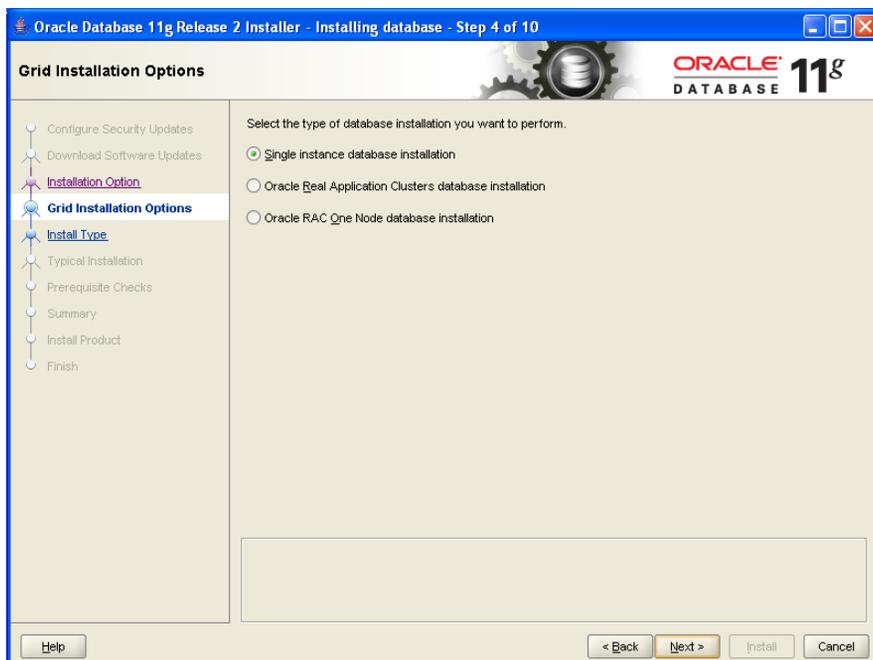
Рисунок 3 – Диалоговое окно с параметрами установки Oracle (шаг 3)



4. Далее нужно выбрать используется RAC (или RAC One). RAC – это специальная опция СУБД Oracle позволяющая создавать кластер из нескольких серверов одновременно работающих с одной БД. Данная опция является платной в Enterprise редакции, и бесплатной (но с учетом ограничений, как правило, бесполезной) в Standard редакции.

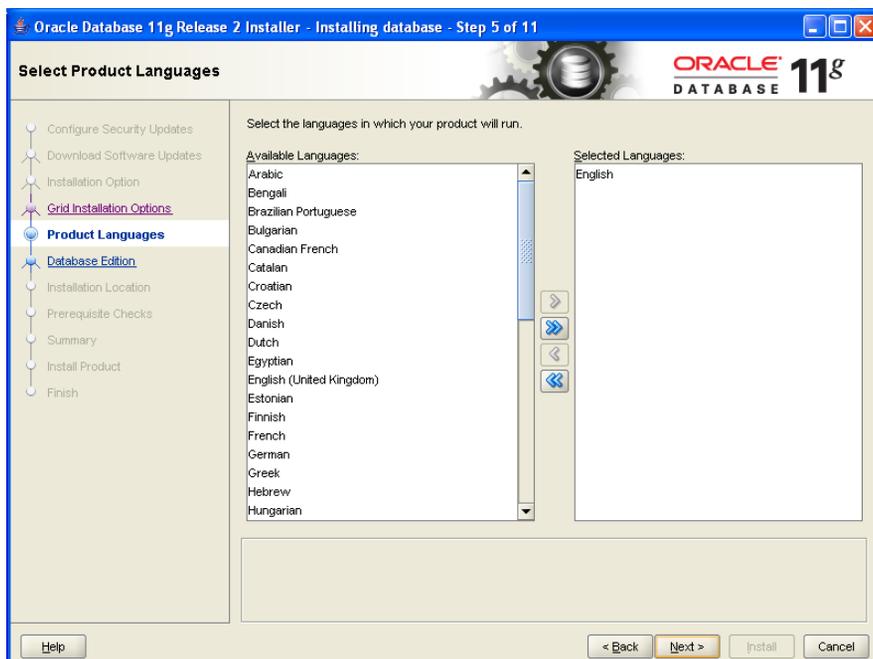
В большинстве случаев RAC с Бюджетом не используется, поэтому выбираем Single instance database installation (не RAC). Если Вы хотите установить именно RAC – то Вам нужно будет выбрать другой вариант.

Рисунок 4 – Диалоговое окно с параметрами установки Oracle (шаг 4)



5. Выбор языка СУБД. Наиболее разумный выбор, по-моему, тот который предлагается по умолчанию, т.е. английский язык. Можно конечно установить русский, но следует помнить, что вся документация написана на английском, поэтому будет довольно сложно сопоставлять английские названия и то, что Вы видите на экране.

Рисунок 5 – Диалоговое окно с параметрами установки Oracle (шаг 5)

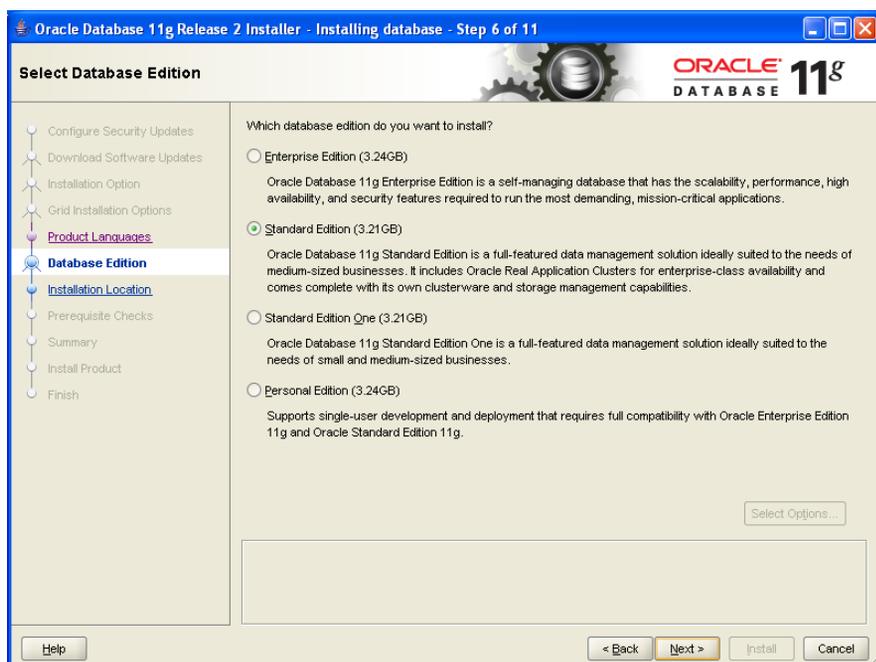


6. Выбор редакции СУБД – укажите ту редакцию, которую приобрели. Следует помнить, что в рамках АС «Бюджет» редакция Personal не поддерживается. Редакция Standard One может устанавливаться на сервера, имеющие 1 или 2 процессорных слота на материнской плате, а редакция Standard на сервера имеющих не более 4-х процессорных слотов.

Если Вы выберете Enterprise Edition, то не забудьте указать дополнительные опции, который были Вами приобретены (кнопка "Select Options..." внизу). Приобретение дополнительных опций, доступно только для редакции Enterprise. Если Вы не приобретали каких-то опций, то нужно снять эти галочки, чтобы не нарушать лицензионного соглашения.

В данном случае выбран вариант Standard Edition.

Рисунок 6 – Диалоговое окно с параметрами установки Oracle (шаг 6)



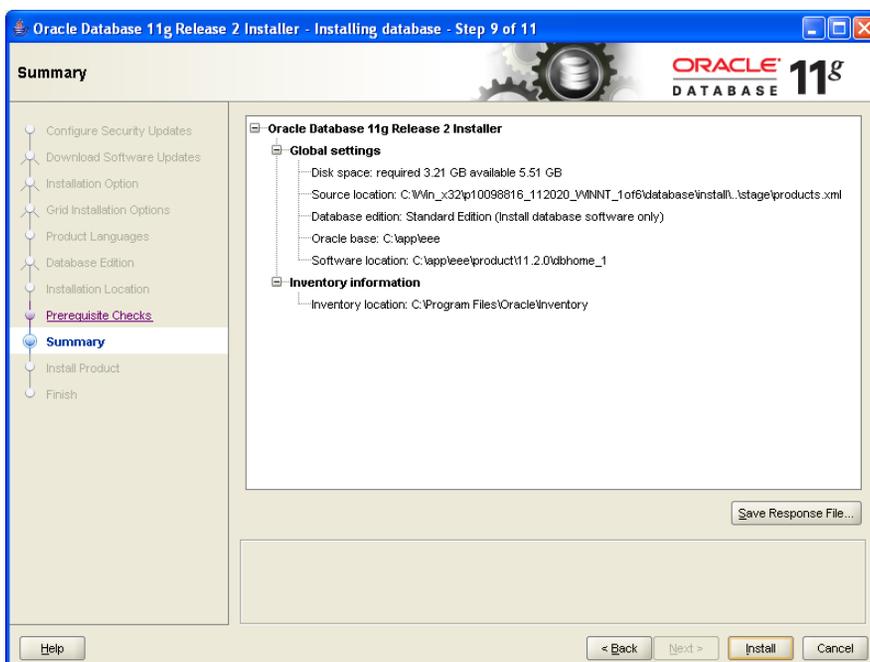
7. Выбор каталога для установки. Как правило, лучше оставить все значения по умолчанию (или почти все). Во-первых, в данном случае – это имя пользователя, под которым мы вошли в систему – его можно из пути исключить (в строке Oracle Base получится c:\app). Во-вторых, если Вы хотите поставить Oracle на диск отличный от предложенного по умолчанию, то в строке Oracle Base измените букву диска (Software Location измениться автоматически). Не нужно пытаться устанавливать Oracle в папку Program Files (а также другие папки в названиях которых есть нестандартные символы).

Рисунок 7 – Диалоговое окно с параметрами установки Oracle (шаг 7)



8. Далее будет выполнена проверка возможности установки выбранной конфигурации на компьютере. Если все прошло успешно, то Вы увидите форму с информацией об установке. Если хотите, то можете сохранить файл с ответами (кнопка Save Response File...), этот файл потом можно передать инсталлятору чтобы он установил СУБД, не задавая вопросов (все ответы будут взяты из файла). В этой форме нужно нажать кнопку Install.

Рисунок 8 – Диалоговое окно с параметрами установки Oracle (шаг 9)



После завершения установки будет выдано соответствующее сообщение в очередном окне – прочитайте его и нажмите кнопку Close.

3.3. Создание Listener

Перед тем как создавать БД нужно создать слушателя (Listener). Для этого запустим утилиту Net Configuration Assistant.

1. В появившемся окне выбираем пункт Listener configuration

Рисунок 9 – Диалоговое окно с параметрами создания слушателя (Listener)



2. Далее выбираем Add. Обратите внимание на пункт Delete – созданный Listener можно удалить также с помощью данного мастера.

Рисунок 10 – Диалоговое окно с параметрами создания слушателя (Listener)



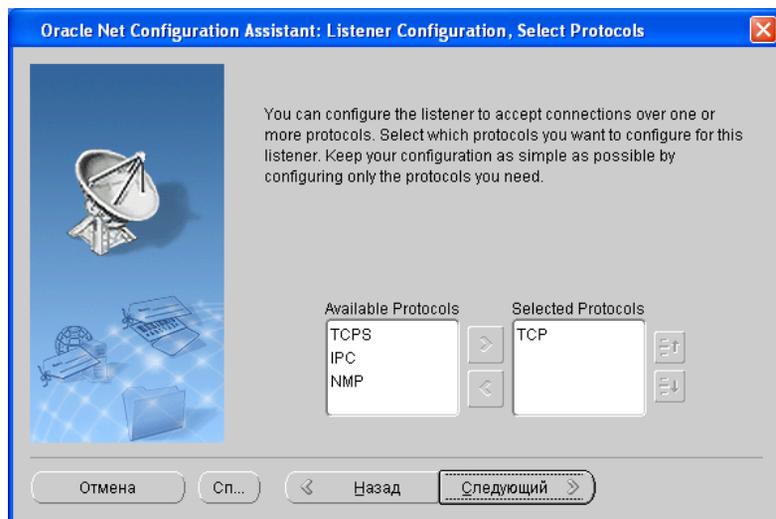
3. Далее вводим имя создаваемого Listener (можно оставить значение по умолчанию).

Рисунок 11 – Диалоговое окно с параметрами создания слушателя (Listener)



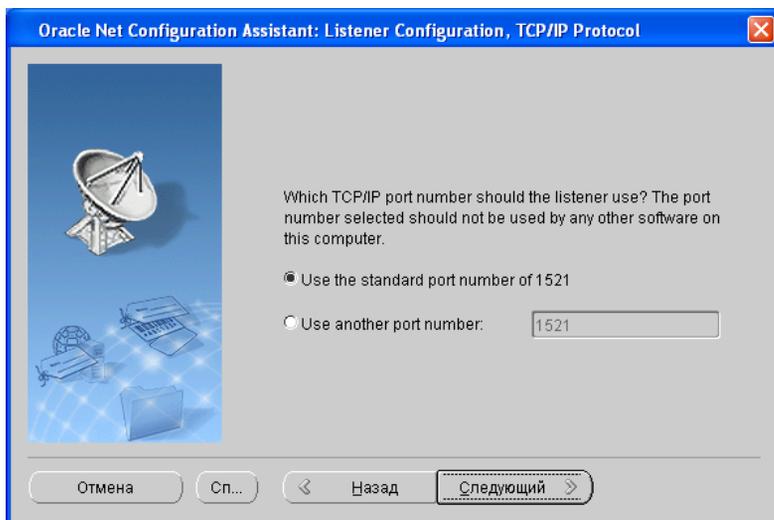
4. Далее выбор сетевых протоколов – оставляем, по умолчанию, TCP.

Рисунок 12 – Диалоговое окно с параметрами создания слушателя (Listener)



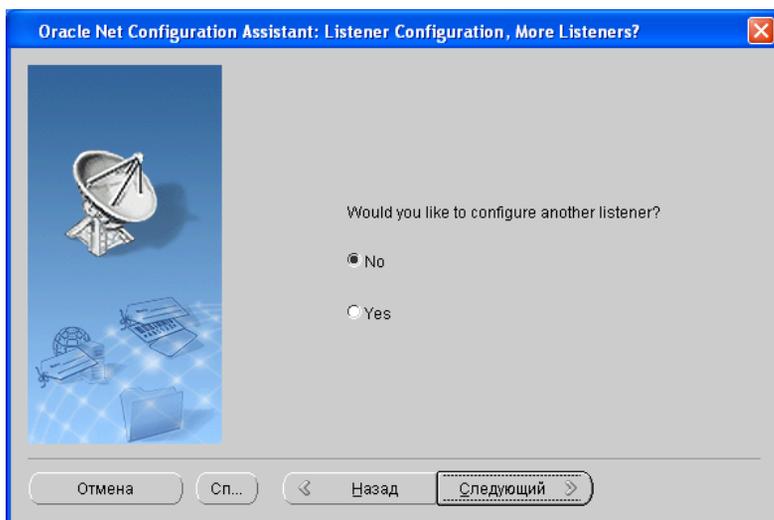
5. Далее выбираем TCP порт – оставляем, по умолчанию 1521. Можно, также указать другой номер порта – это может потребоваться в том случае, если Вы хотите установить несколько Listener-ов на один компьютер (следует помнить – один Listener может обслуживать несколько БД). Указанный порт должен быть открыт в файрволе.

Рисунок 13 – Диалоговое окно с параметрами создания слушателя (Listener)



6. Далее будет задан вопрос нужно ли настраивать другой Listener – отвечаем No.

Рисунок 14 – Диалоговое окно с параметрами создания слушателя (Listener)



7. В итоге создается Listener и выдается сообщение об успешном выполнении операции. Далее жмем кнопку Следующий и выходим из мастера (кнопка Готово или Отмена).

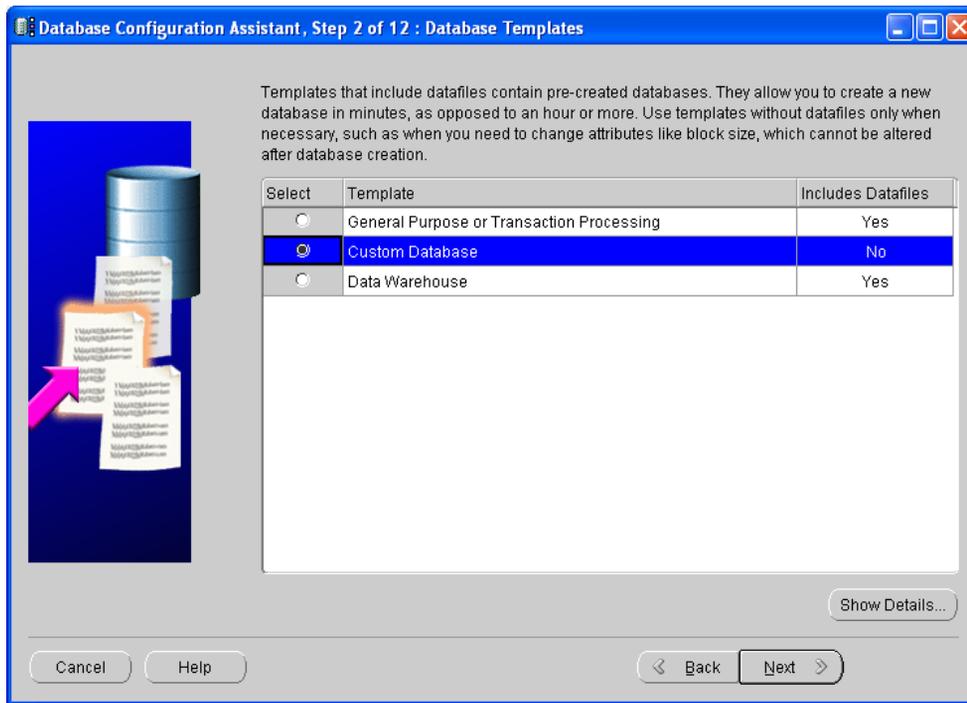
3.4. Создание БД

После установки СУБД нужно будет создать базу данных. Для этого используем утилиту Database Configuration Assistant (она должна появиться в меню "Пуск"). При описании отдельных форм мастера будут пропускаться некоторые дополнительные закладки (они, как правило, предназначены для более тонкой или нестандартной настройки БД).

1. После запуска будет показана форма Welcome – просто жмем кнопку Next. Далее выбираем Create a database (создать базу данных) и жмем Next. Стоит обратить внимание на пункт Delete a database – при помощи данной утилиты также можно удалить созданную БД.

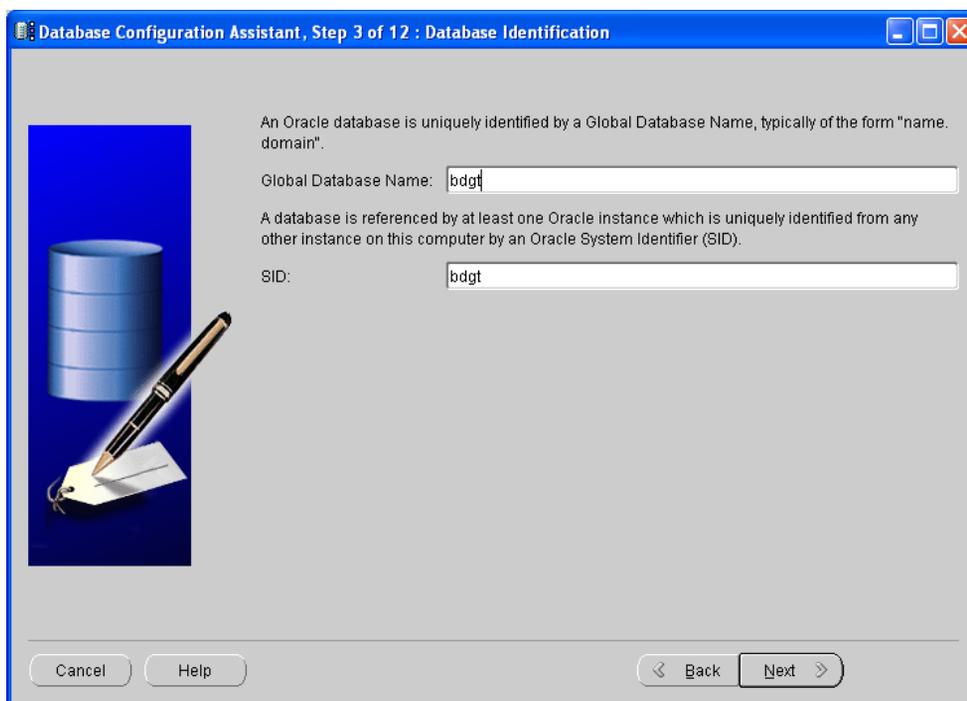
- Далее предлагается выбрать шаблон для создаваемой БД – выбираем Custom Database и жмем кнопку Next.

Рисунок 15 – Диалоговое окно с параметрами создания БД (шаг 2)



- В следующей форме указываем имя создаваемой БД – пишем его в строке Global Database Name (строка SID будет автоматически заполнена тем же значением).

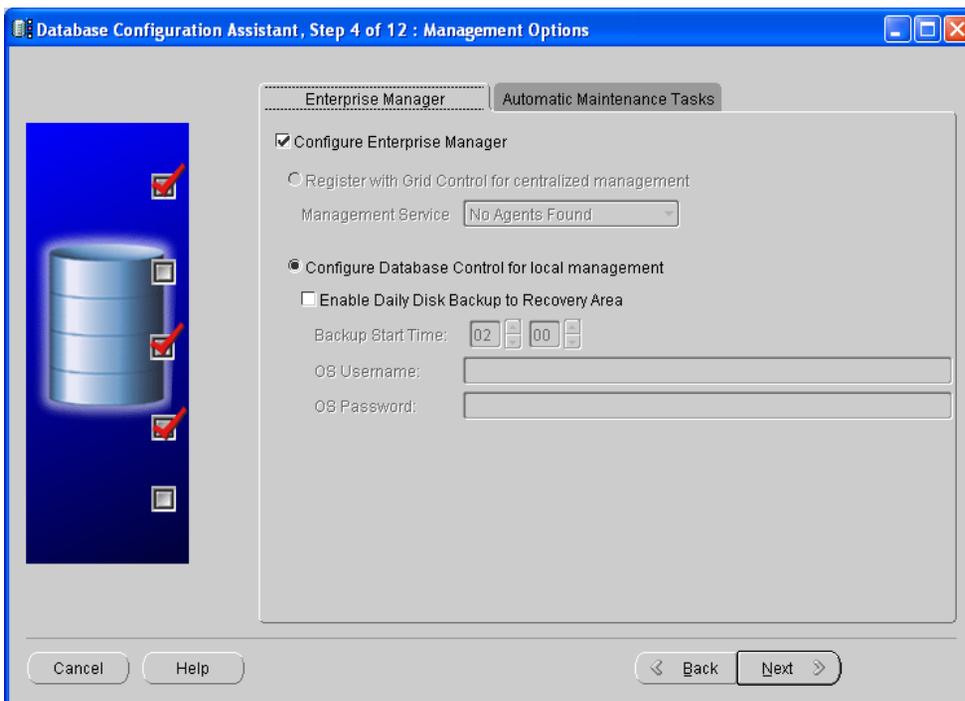
Рисунок 16 – Диалоговое окно с параметрами создания БД (шаг 3)



4. В следующей форме все оставляем по умолчанию (как показано на картинке).

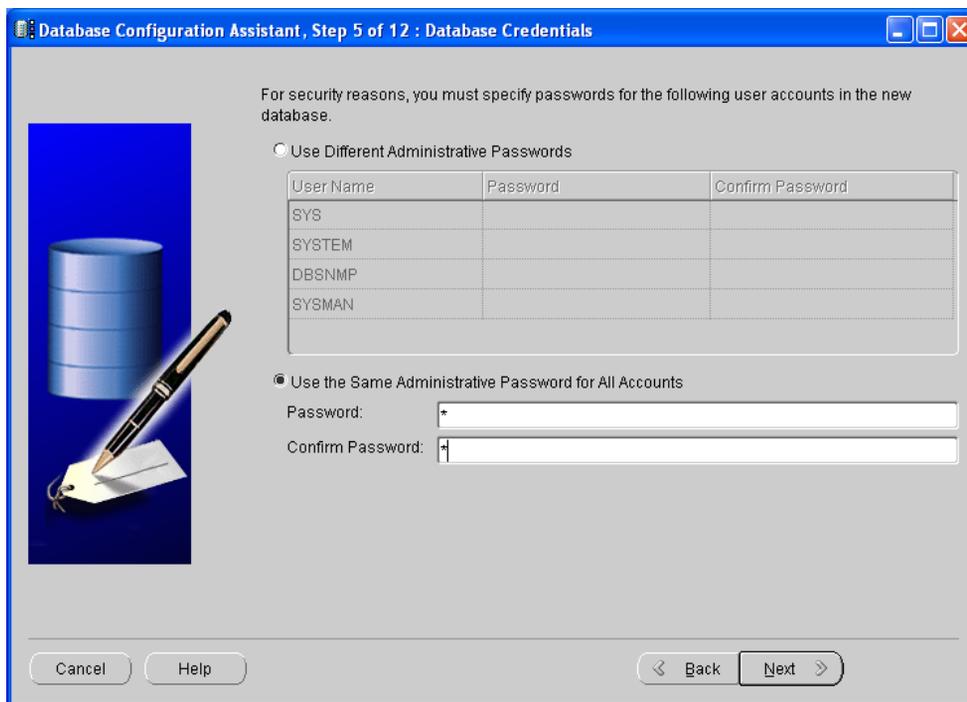
Если Вы хотите настроить резервное копирование БД способом, который рекомендует Oracle, то Вам нужно будет установить галочку **Enable Daily Disk Backup to Recovery Area** (разрешить ежедневное резервное копирование), указать время, в которое будет выполняться резервное копирование, а также имя и пароль пользователя операционной системы обладающего достаточными правами для выполнения операции (как минимум данному пользователю будут нужны права на доступ к каталогу, в котором создается резервная копия). Этот каталог и прочие параметры будут настраиваться на шаге 7 (не забудьте их настроить!).

Рисунок 17 – Диалоговое окно с параметрами создания БД (шаг 4)



5. Далее нужно будет указать пароль для системных пользователей. Можно указывать пароль для каждого пользователя отдельно, а можно задать одинаковый для всех (именно этот вариант и выбран на рисунке). Обязательно запомните этот пароль! Он потребуется в дальнейшем для настройки базы данных.

Рисунок 18 – Диалоговое окно с параметрами создания БД (шаг 5)



6. На следующем шаге, нужно будет выбрать место расположения файлов БД. Есть два варианта способа хранения:
- ♦ файловая система;
 - ♦ ASM – специальная система хранения от Oracle.

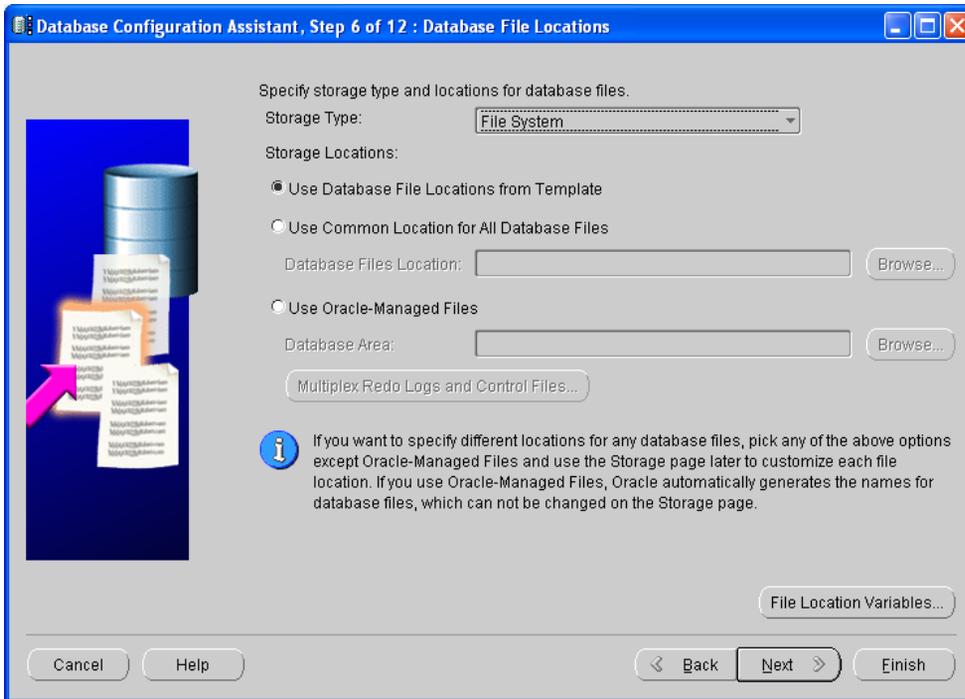
Выбираем файловую систему – это более простой, и, как правило, не менее эффективный вариант.

Из трех возможных вариантов размещения файлов:

- ♦ из шаблона – т.е. разместить файлы так, как описано в шаблоне (шаблон мы выбирали на втором шаге);
- ♦ использовать общее местоположение для всех файлов (нужно указать каталог);
- ♦ использовать файлы управляемые Oracle (нужно будет указать их местоположение, а далее Oracle сам будет создавать файлы и давать им имена).

выбираем вариант из шаблона (Use Database File Location from Template). Здесь не нужно указывать какие конкретно файлы будут созданы, и как они будут называться – это будет сделано чуть позже.

Рисунок 19 – Диалоговое окно с параметрами создания БД (шаг 6)



7. Далее нужно указать местоположение для Fast Recovery Area – специального каталога который Oracle использует в различных целях, как правило, для операций так или иначе связанных с резервным копированием и восстановлением. Оставим все по умолчанию – потом эти настройки можно будет изменить. Если у Вас мало свободного места на диске (меньше чем указано для Fast Recovery Area), то Oracle можно ругнуться по этому поводу – если Вы не собираетесь использовать резервное копирование, то это сообщение можно проигнорировать (или уменьшить размер Fast Recovery Area).

Кроме того нужно указать параметры архивирования журнальных файлов (redo log). Поскольку мы создаем БД без резервного копирования, то галочку Enable Archiving ставить не будем (как уже говорилось выше, если потребуется, то резервное копирование можно будет настроить после).

Если Вы хотите настроить резервное копирование БД, то Вам нужно будет указать корректный размер для Fast Recovery Area (Oracle рекомендует устанавливать этот размер в 2 раза больше чем предполагаемый размер БД), а также ее размещение. При указании размещения можно использовать переменные (например, {ORACLE_BASE}), их значения можно посмотреть в диалоге, который открывается при нажатии кнопки File Location Variables... Резервные копии рекомендуется хранить на физическом носителе отличном от того, на котором размещаются файлы БД, в крайнем случае (при размещении резервной копии и файлов БД на одном диске) следует использовать RAID массивы предусматривающие защиту от сбоев.

Резервное копирование, по варианту, предлагаемому Oracle, выполняется следующим образом – создается полная резервная копия всех файлов данных (в нее могут не входить неиспользуемые блоки, за счет этого размер резервной копии может быть несколько меньше чем размер БД). Далее каждый день создается инкрементальная резервная копия (в которую входят все блоки, изменившиеся с момента последнего резервного копирования (т.е. за сутки), а инкрементальная резервная копия, созданная накануне накатывается поверх полной резервной копии. В итоге после выполнения резервного копирования (допустим, оно выполняется в 2 часа ночи) мы получаем

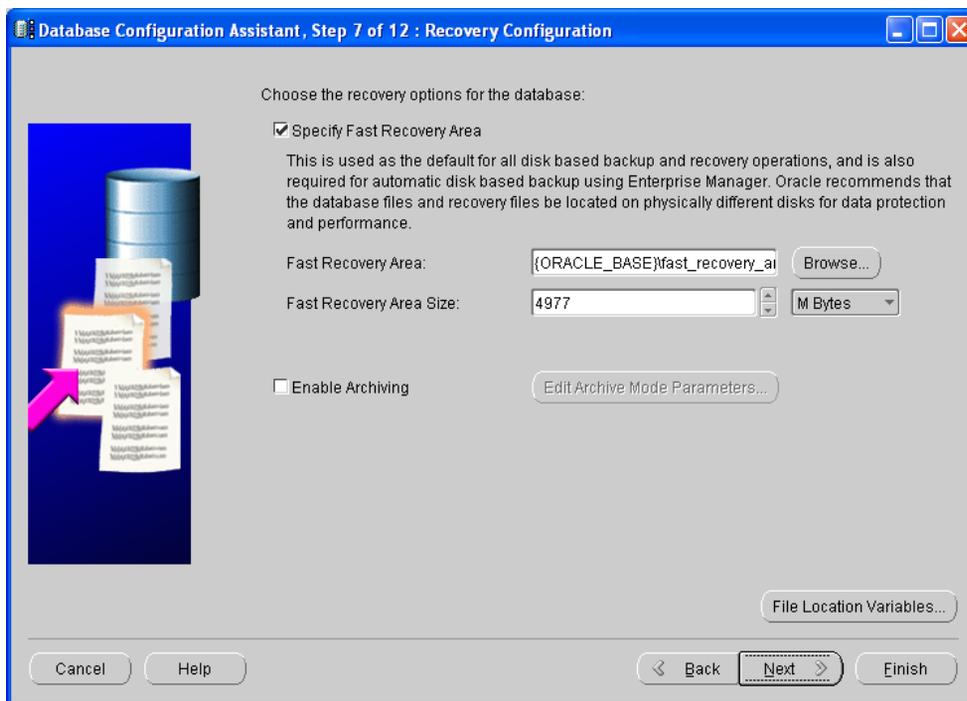
полную резервную копию по состоянию на 2 часа ночи вчера, и инкрементальную резервную копию, включающую все блоки, изменившиеся за 24 часа (с 2-х часов ночи вчера до 2-х часов ночи сегодня). Кроме того нужно хранить все журнальные файлы (используемые и архивные) начиная с 2-х часов ночи сегодня и до текущего момента (а еще лучше с 2-х часов ночи вчера). При восстановлении БД в качестве основы берется полная резервная копия, на которую накатываются инкрементальная резервная копия и логи, поэтому Вы можете восстановить БД не только по состоянию на текущий момент (или момент падения), но и на некоторый момент предшествующий текущему. В принципе, если есть логи с момента создания полной резервной копии до настоящего момента, то этого будет вполне достаточно для восстановления, но использование инкрементальной резервной копии ускоряет процесс (вместо последовательного наката логов используется накат сразу всех изменений). Нужно отметить, что логи могут только накатываться на резервную копию (но не могут откатываться, т.е. Вы не можете восстановить состояние БД на момент времени раньше, чем была создана полная резервная копия).

Журнальные файлы как уже было указано выше делятся на используемые (online) и архивные. Используемые логи – это те, с которыми БД работает в данный момент, после их заполнения они могут быть перезаписаны. Для того чтобы не потерять информацию необходимую для восстановления, перед тем как файл будет перезаписан, его содержимое должно быть скопировано в архив (в итоге мы получим архивный журнальный файл).

Местоположение используемых журнальных файлов определяется тогда же, когда мы будем размещать и остальные файлы БД. А вот архивирование журнальных файлов нужно настроить именно в данной форме. Для этого ставим галочку Enable Archiving. В диалоге Edit Archive Mode Parameters... можно настроить формат имени файлов журнальных логов и их местоположение. Лучше все оставить по умолчанию: галочка Automated archiving установлена, формат имени файла ARC%S_%R.%T, а список мест, куда будут копироваться архивные логи (archiving log destination) пуст. То, что список мест, куда будут копироваться архивные логи, пуст – не должно Вас смущать, т.к. в этом случае, в качестве места для копирования выбирается Fast Recovery Area.

Если Вы настроили резервное копирование БД, то нужно будет регулярно проверять Fast Recovery Area (чтобы там всегда было достаточно места), кроме того рекомендуется регулярно выполнять проверку резервных копий, а также изучить документацию по резервному копированию и восстановлению Oracle.

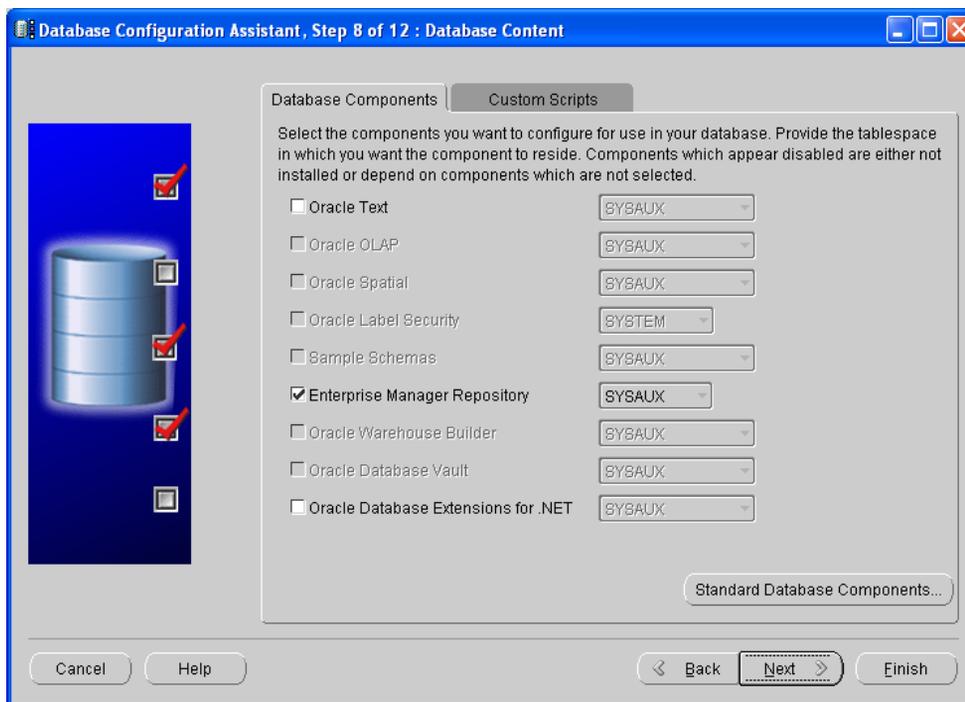
Рисунок 20 – Диалоговое окно с параметрами создания БД (шаг 7)



8. На шаге 8 предлагается указать дополнительные блоки (опции и фичи (features)), которые Вы хотите добавить в БД. Поскольку при установке программного обеспечения был выбран вариант Standard Edition, то большинство пунктов изначально недоступны. У тех, кто ставит Enterprise редакцию может возникнуть вопрос – зачем два раза указывать список устанавливаемых опций – дело в том, что Вы можете создать несколько БД на одном и том же сервере с разными опциями (соответственно при установке ПО Вы должны будете указать все опции которые будут Вами использоваться, а при создании отдельной БД можно указать их подмножество).

Рекомендуемый вариант снять все галочки, кроме Enterprise Manager Repository (чем проще система, тем проще с ней работать). Enterprise Manager Repository – нужен для управления БД, поэтому его оставляем. В диалоге, который открывается по кнопке Standard Database Components... можно отключить установку базовых фич БД (в принципе это можно сделать, но мы так поступать не будем).

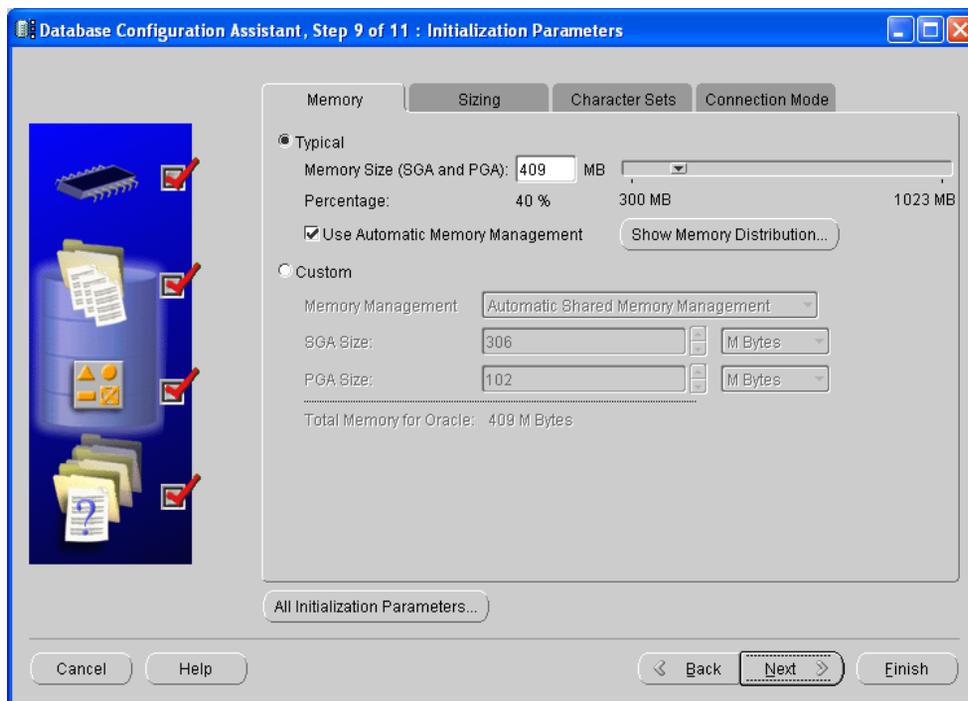
Рисунок 21 – Диалоговое окно с параметрами создания БД (шаг 8)



9. На шаге 9 настраиваются параметры инициализации БД. Те, что представлены на вкладках, в основном отвечают за быстродействие. Кроме того есть кнопка All Initialization Parameters... – в диалоге который появиться можно настроить любые параметры инициализации БД (кстати часть из них мы уже настроили – например, параметры резервного копирования).

Первая закладка – настройка параметров памяти: рекомендуется использовать вариант Typical с автоматическим распределением памяти между SGA и PGA (в данном случае во время работы будет определяться сколько памяти требуется под SGA и PGA и при необходимости память будет автоматически перераспределяться). При указании объема памяти выделяемого под БД следует ориентироваться на общий объем физической памяти на сервере, а также учитывать потребление памяти другими приложениями (и операционной системой). Например, при установке на сервер с 16Гб памяти под сервер Oracle вполне можно выделить 12Гб памяти (предполагается, что речь идет о 64-х разрядной системе и кроме СУБД Oracle на сервере не будет других ресурсоемких приложений), остальные 4Гб отводятся под операционную систему, а также в качестве резерва – ибо при нехватке физической памяти может начаться свопинг, что приведет к резкому падению производительности.

Рисунок 22 – Диалоговое окно с параметрами создания БД (шаг 9)



На последней закладке нужно указать режим работы сервера Oracle с клиентскими сессиями – оставляем по умолчанию Dedicate Server Mode.

Вариант Shared Server Mode имеет смысл использовать только при большом количестве одновременных подключений к серверу. Задуматься о его использовании можно при числе подключений более 500, а если подключений будет более 1000, то режим работы SHARED SERVER будет рекомендуемым.

Здесь нужно пояснить по поводу режимов работы. В Dedicate режиме для каждого подключения выделяется отдельный «процесс» (в Windows нитка, в Linux и Unix – процесс) и оно потребляет ресурсы (в основном память) даже когда неактивно. В Shared режиме есть пул «процессов» называемых серверами (их количество настраивается) и есть диспетчеры, которые знают о том, кто подключен (их количество также настраивается). Когда приходит запрос от клиента, то в Dedicate режиме обработка выполняется ниткой данного клиента, а в Shared режиме диспетчер передает запрос свободному «процессу» из пула (серверу), если свободных нет, то запрос встает в очередь.

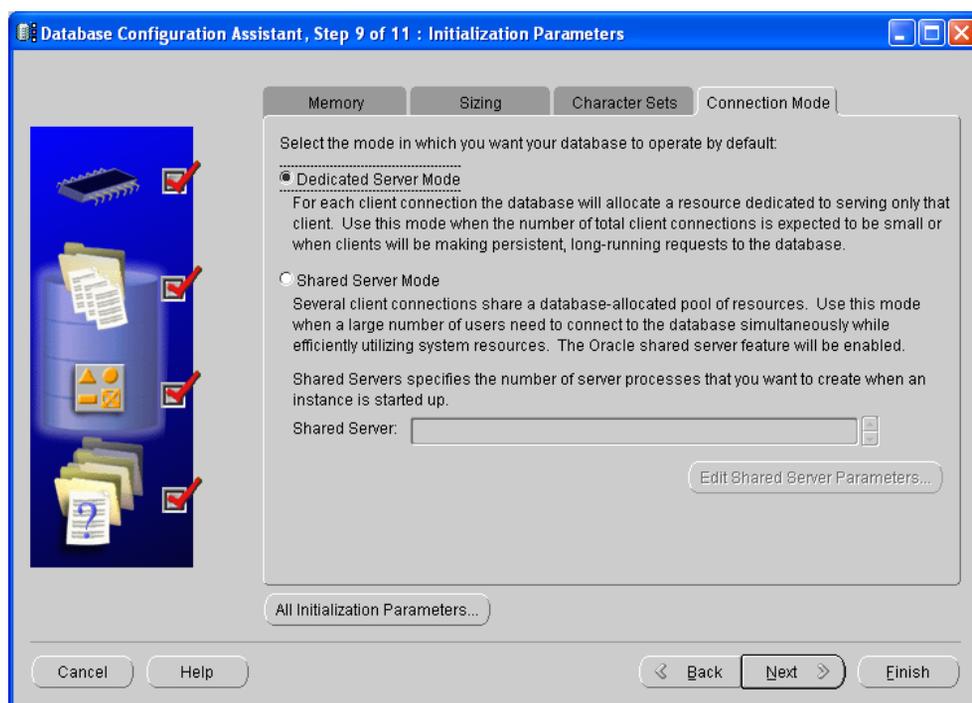
Таким образом, в Shared режиме схема работы получается сложнее (требуются дополнительные расходы на переключение контекста Shared серверов), и при малом числе сессий ее использование невыгодно. При большом числе подключений Shared режим дает выигрыш по памяти (например, вместо 600 «процессов», может остаться 1 «процесс» для диспетчера + 20 «процессов» для серверов + 20 служебных «процессов» итого 41 «процесс»). При определенных условиях это может положительно сказаться на общей скорости работы БД (например, будет меньше переключений контекстов CPU, меньше вытеснения данных в кэшах и т.п.).

Следует помнить, что если Вы выбрали режим Shared, то клиенты по умолчанию будут подключаться через диспетчера (т.е. в shared режиме), но клиента можно настроить и так чтобы он подключался напрямую (т.е. в Dedicate режиме).

Из параметров Shared режима можно обратить внимание на следующие (прочие можно вручную не менять – установятся автоматом):

- ♦ `max_shared_servers` (см. кнопку Edit Shared Server Parameters...) – максимальное число Shared серверов в пуле. Рекомендуется, чтобы это значение было не более чем 2 x (количество потоков CPU). Количество потоков CPU – это, как правило, то число «процессоров» которое видит операционная система в диспетчере задач. В качестве первого приближения можно установить 1,5 x (количество потоков CPU). В дальнейшем если загрузка процессоров будет низкой, при наличии запросов стоящих в очереди, то это значение можно увеличить. Если загрузка CPU будет очень высокой, то это значение можно снизить.
- ♦ `shared_servers` – число постоянно живущих Shared серверов в пуле. Это значение должно быть меньше `max_shared_servers`. В принципе его можно не устанавливать, тогда сервера будут создаваться и удаляться по мере надобности. Однако, в некоторых случаях, это может приводить к небольшому снижению производительности (на создание и удаление серверов будет тратиться время). В первом приближении можно установить это значение равным (количеству потоков CPU).

Рисунок 23 – Диалоговое окно с параметрами создания БД (шаг 9)



На второй вкладке определяется размер блока для хранения данных (пространство под данные в файлах Oracle выделяется блоками) – оставляем предлагаемый по умолчанию вариант 8192 байта (8Кб).

Также здесь нужно указать максимально допустимое количество процессов Oracle. Количество процессов можно рассчитать примерно так:

$Process \geq (N+0,5)*X + N*Y + 25$, где:

- ♦ X – количество рабочих станций с АС «Бюджет» в локальной сети;
- ♦ Y – количество активных удаленных пользователей (УРМ);

- ◆ N – число подключений, используемых одним экземпляром программы (N = от 1 до 3):
 - одно подключение присутствует всегда – это подключение к основной схеме АС «Бюджет»,
 - если удаленный реестр АС «Бюджет» расположен в той же БД, что и основные данные, то это дает еще одно подключение,
 - если крипто-схема расположена в той же БД, что и основные данные, то это дает еще одно подключение;
- ◆ для Бюджета мы берем N+0,5, т.к. иногда abl-макросы создают дополнительное подключение к БД (URM этим не грешит).
- ◆ 25 добавляется на служебные процессы Oracle.

При указании параметра Process лучше добавить некоторый запас, т.к. при попытке создать больше процессов, чем указано Вы получите ошибку. С другой стороны задавать слишком большое (т.е. значительно превосходящее реальные потребности) значение параметра process, также не рекомендуется, т.к. это отрицательно сказывается на работе менеджера PGA памяти (менеджер будет «экономить» память в расчете на большое число процессов и в итоге производительность может снизиться).

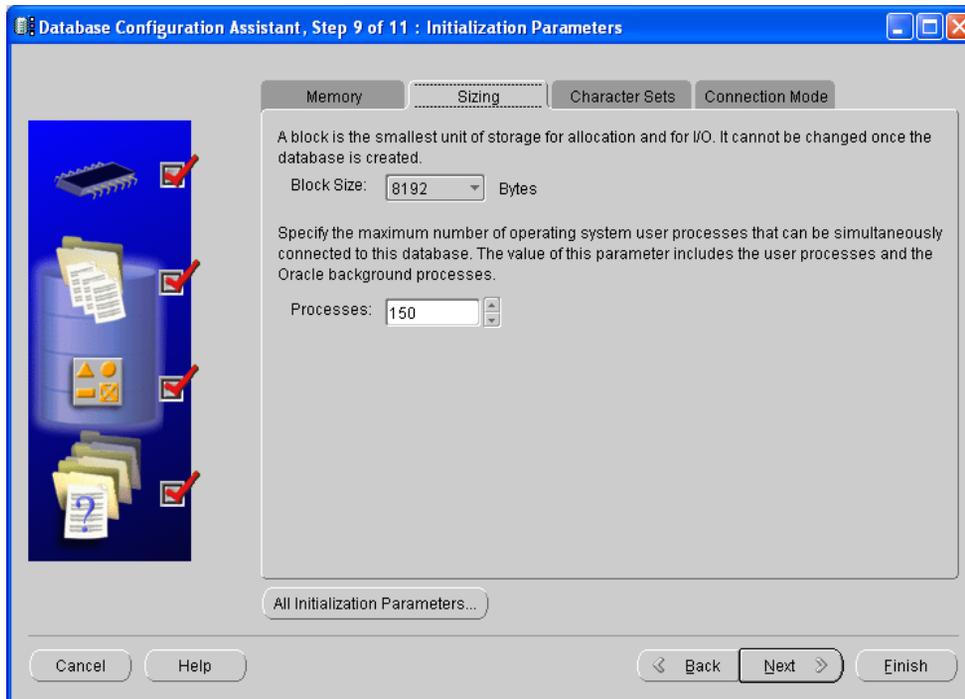
Пусть, например, имеется 100 рабочих станций с АС «Бюджет» и один сервер URM (на котором установлен лимит в 30 подключений). В создаваемой БД будет храниться основная схема АС «Бюджет», а также схема реестра (крипто-схема не используется). В этом случае

$(N+0,5)*X + N*Y + 25 = (2+0,5)*100 + 2*30 + 25 = 335$ добавляем "запас" и в итоге значение параметра Process может быть установлено как 400.

Если был выбран режим работы Shared Server, то количество процессов уже не зависит от числа подключений – оно будет определяться как сложение количества системных процессов (около 30), количества процессов для диспетчеров и shared-серверов, а также количества пользовательских сессий работающих в режиме dedicate (такие сессии могут использоваться для выполнения административных задач). Однако максимально возможное число сессий (параметр SESSIONS) будет по-прежнему определяться максимально возможным числом подключений. Более того Oracle рекомендует устанавливать параметр SESSIONS равным $(1.5 * PROCESSES) + 22$, т.е. в нашем случае это будет $(1,5 * 335) + 22 = 524$. В итоге добавив запас можно установить значение параметра как 550 или 600. Установить значение параметра SESSIONS можно в диалоге, вызываемом по кнопке "All Initialization Parameters...". Если установить значение PROCESSES, но не менять параметр SESSIONS, то Oracle установит параметр SESSIONS в значение по умолчанию (например, если PROCESSES = 100, то SESSIONS будет равен 172, что при использовании Shared Server неверно).

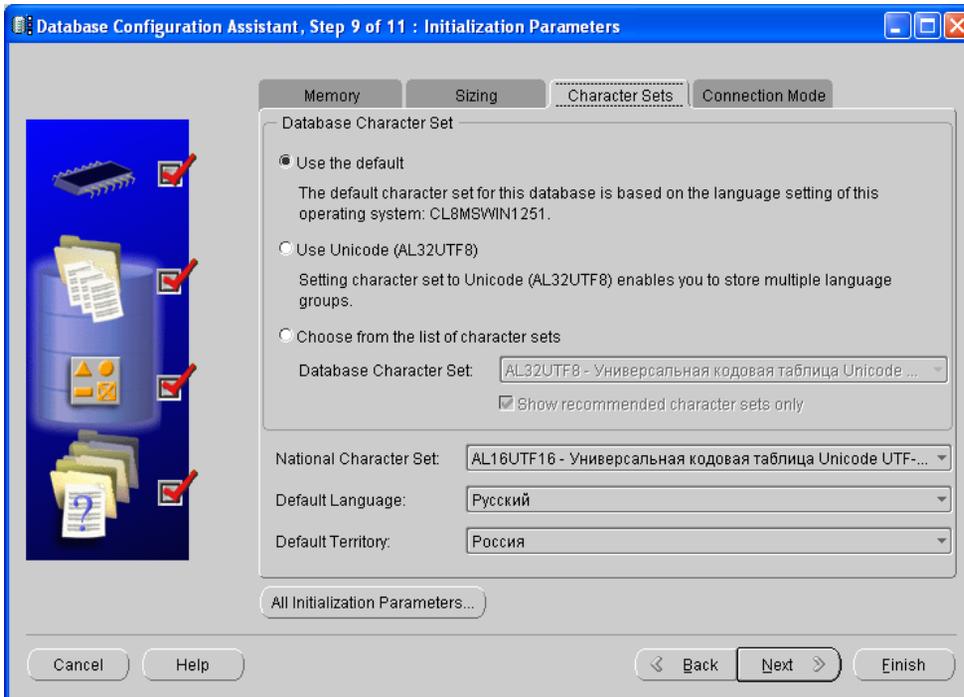
В принципе параметр PROCESSES можно установить также как и для режима DEDICATE (тогда не нужно будет отдельно редактировать параметр SESSIONS).

Рисунок 24 – Диалоговое окно с параметрами создания БД (шаг 9)



На третьей вкладке настраивается используемая кодировка. Обязательно проверьте корректность ее установки, т.к. потом ее практически невозможно будет изменить. Database Character Set должен быть установлен в CL8MSWIN1251 (как правило, это значение предлагается по умолчанию). National Character Set должен быть установлен в AL16UTF16 (тоже, как правило, предлагается по умолчанию). Значения Language и Territory лучше установить аналогично нижеприведенному рисунку, хотя это и не принципиально (можно будет поменять потом).

Рисунок 25 – Диалоговое окно с параметрами создания БД (шаг 9)



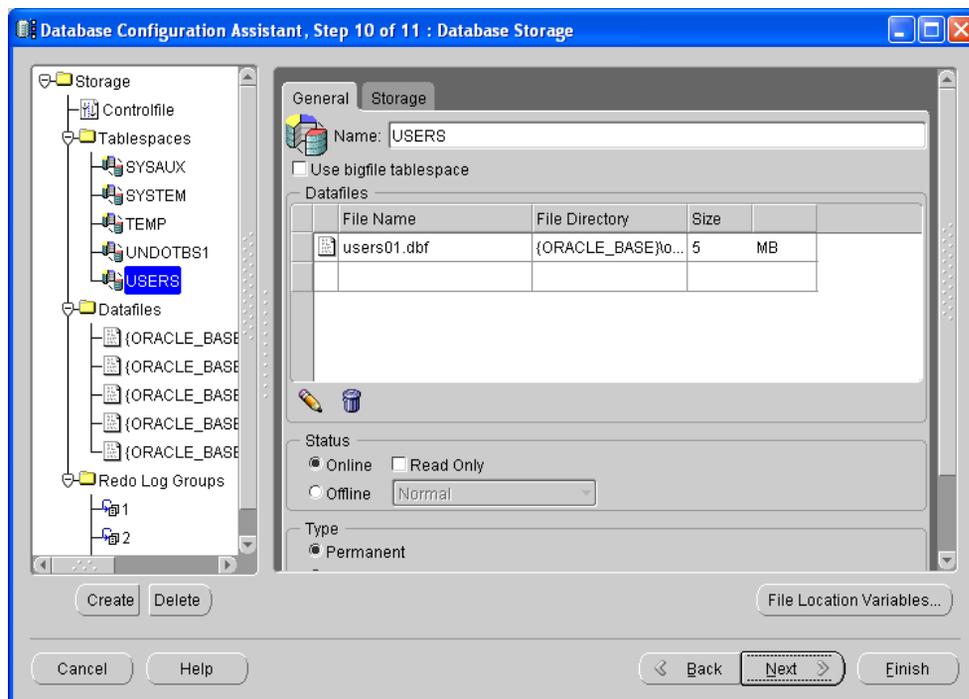
10. Далее предлагается настроить параметры хранения данных (имена и местоположение файлов, их размеры). В принципе можно оставить все по умолчанию или изменить местоположение файлов, если Вы хотите их разместить на другом диске. При указании расположения файлов используются переменные (например, {ORACLE_BASE}), их значения можно посмотреть в диалоге, который открывается при нажатии на кнопку File Location Variables...

Также если предполагается создавать БД большого размера, то в табличное пространство USERS можно добавить файлы. Это связано с тем, что в Oracle есть ограничение на размер файла данных (оно зависит от размера блока): при размере блока 8Кб максимально возможный размер файла будет около 32Гб, при 16Кб около 64Гб и т.д. Для добавления файла нужно войти в пустую строку в списке файлов нужного табличного пространства (в примере на картинке, можно кликнуть мышкой в пустую ячейку в колонке File Name) и начать вводить имя файла. После добавления файла его параметры можно отредактировать в отдельном диалоговом окне (это более удобно, т.к. в таблице очень маленькие ячейки). Для вызова данного диалогового окна нужно сделать двойной клик на иконке рядом с именем файла (в списке файлов табличного пространства).

Смущаться "небольшим" размером файлов, который предлагается по умолчанию не стоит, т.к. у всех файлов с данными (Datafiles) по умолчанию установлено авторасширение (т.е. файл будет автоматически увеличиваться в размере, по мере необходимости, до тех пор, пока не достигнет предельного значения).

Если Вы настроили резервное копирование на диск, отличный от того, на котором хранятся данные, то имеет смысл для Redo Log Groups (групп используемых журнальных файлов) настроить два места размещения файлов. По умолчанию в каждой из трех групп только по одному файлу, если указать в каждой группе по два файла (второй файл должен лежать на другом диске, чтобы обеспечить отказоустойчивость), то Oracle будет писать одни и те же данные в оба файла.

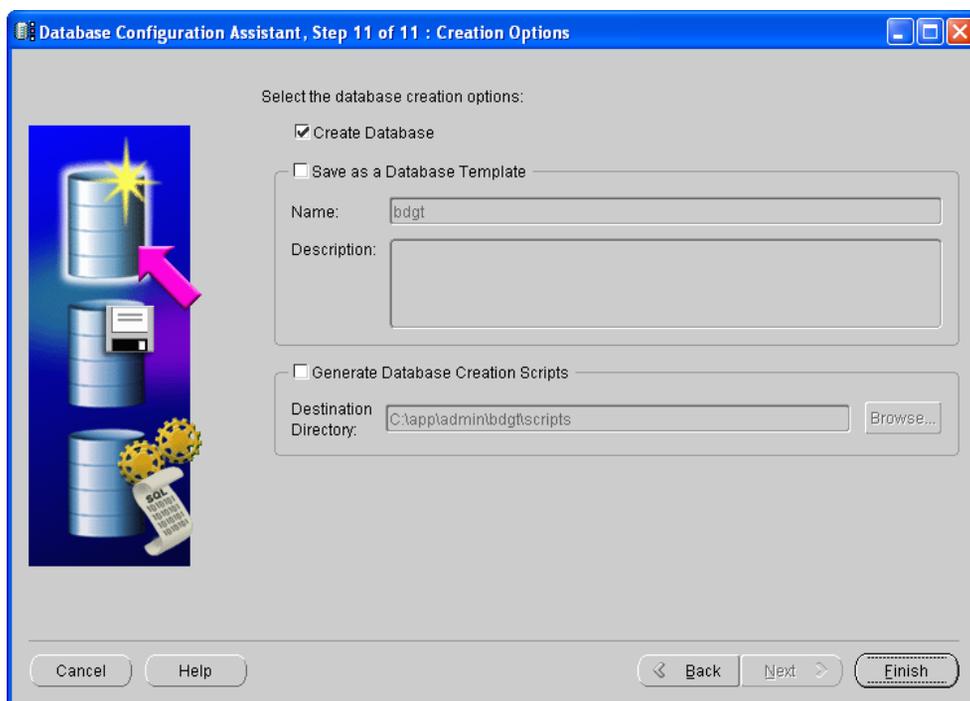
Рисунок 26 – Диалоговое окно с параметрами создания БД (шаг 10)



11. После настройки всех параметров можно будет запустить создание БД нажатием на кнопку Finish. Будет выведено окно с информацией о создаваемой БД (чтобы можно было еще раз убедиться, что все настроено правильно), ждем там ОК и ждем окончания установки.

В конце появится формочка с уведомлением об успешном окончании установки, там, в частности, будет указан адрес для доступа к web-интерфейсу администрирования. В этой формочке нужно нажать Exit и установка будет завершена.

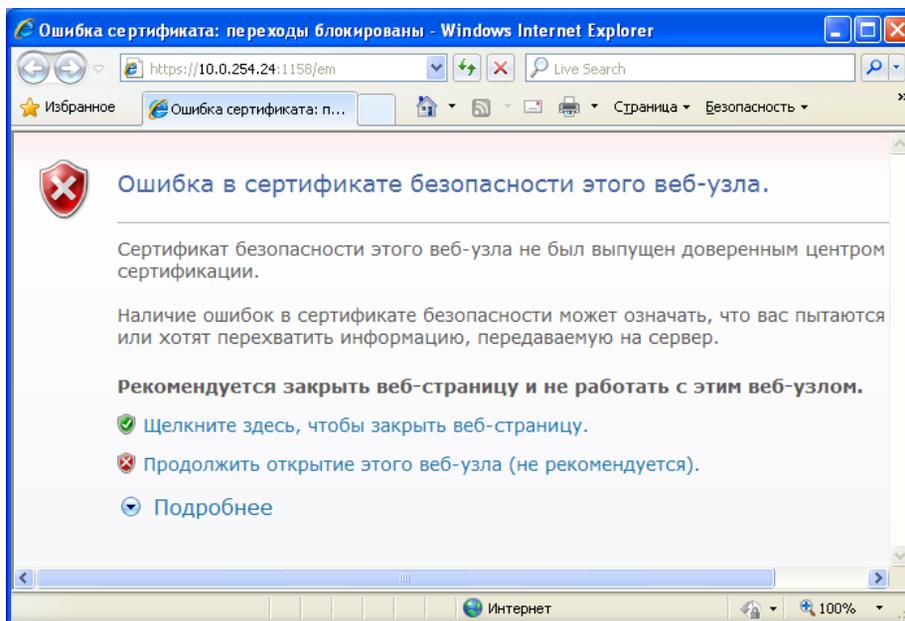
Рисунок 27 – Диалоговое окно с параметрами создания БД (шаг 11)



3.5. Порядок действий после создания БД

1. Следует убедиться, что имеется доступ к web-интерфейсу администрирования – для этого в любом web-браузере (IE, Mozilla, Opera и т.д.) открываем страницу, расположенную по адресу <https://сервер:1158/em/> – должен появиться запрос логина. Поскольку устанавливается защищенное ssl соединение, то Ваш web-браузер может спросить нужно ли доверять предъявленному сертификату. Например, если Вы используете IE, то можете получить приблизительно такое сообщение, как показано на рисунке – нужно просто выбрать пункт "продолжить открытие этого веб-узла". Еще лучше будет установить сертификат у себя и указать что Вы ему доверяете (тогда в следующий раз Вы не получите такого сообщения).

Рисунок 28 – Вид сообщения IE об ошибке сертификата



2. Если Вы не хотите нарушить лицензионное соглашение, то в web-интерфейсе нужно зайти в раздел setup->Managment Pack Access и отключить использование тех пакетов, которые Вы не приобретали.
3. Если Вы хотите подключаться к Oracle из 32-х разрядных программ (АС «Бюджет», OraTools и т.п.) прямо с сервера, и если на сервере установлена 64-х разрядная система, то нужно будет установить 32-х разрядного клиента Oracle. Если был установлен 32-х разрядный сервер Oracle, то установка 32-х разрядного клиента не нужна. Но в любом случае Вам нужно будет настроить клиента (создать Oracle алиасы в файле tnsnames.ora) – это можно сделать с помощью утилиты Net Manager (она расположена там же где и Database Configuration Assistant).
4. Инсталлятор клиента версии 11.2 не всегда корректно указывает используемую кодировку, что может приводить к проблемам при работе с текстовыми данными. Для устранения проблемы следует изменить настройки клиента Oracle – указанную по умолчанию кодировку (как правило, это европейская кодировка WE8MSWIN1252) заменить на русскую (CL8MSWIN1251). Это можно сделать разными способами, например, найти в реестре (среди настроек Oracle) ключ NLS_LANG и изменить его значение. Значение этого ключа состоит из двух частей, первая указывает используемые региональные стандарты (формат даты, знак валюты и т.п.), а вторая кодировку (например, RUSSIAN_CIS.CL8MSWIN1251). При редактировании реестра следует помнить, что в 64-х разрядных версиях Windows настройки 32-х разрядных программ хранятся не в разделе HKLM\Software, а в разделе HKLM\Software\Wow6432Node.
5. Для выполнения sql-команд, скриптов, а также для получения отчетов о состоянии БД можно использовать утилиту SqlDeveloper, которая идет в поставке сервера Oracle. При первом запуске SqlDeveloper запросит местоположение java.exe, поскольку тот SqlDeveloper, что идет в поставке – 32х разрядный, то ему нужна 32х разрядная JDK (именно JDK, а не JRE), т.е. если ставиться серверный дистрибутив x64, то та Java, которая в нем есть, тоже x64 и для SqlDeveloper-а не подходит. SqlDeveloper бесплатен и доступен на сайте Oracle в виде отдельного дистрибутива (таким образом, его можно установить на другие компьютеры).

6. Нужно подключиться под SYS и выполнить скрипт CreatePrimaryObjects.sql (если у Вас его нет, или он слишком старый, то лучше запросить свежую версию в Кристе) – это скрипт создает базовые объекты необходимые для работы АС «Бюджет». Это обязательно нужно сделать до того как Вы начнете создавать схемы.
7. Если устанавливается Enterprise Edition и предполагается использование утилиты exp (утилита для создания дампов старого формата), то нужно отключить отложенное выделение сегментов, установив параметр инициализации DEFERRED_SEGMENT_CREATION в значение False. Дело в том, что при отложенном выделении сегментов первые сегменты для таблицы выделяются не при ее создании, а при первом добавлении данных в нее. Таким образом, могут быть таблицы без сегментов, а утилита exp не записывает в дампы таблицы, у которых нет сегментов – соответственно при восстановлении такого дампа мы можем получить схему с пропущенными таблицами. Параметр DEFERRED_SEGMENT_CREATION влияет только на вновь создаваемые таблицы, для уже существующих таблиц можно выполнить команду, которая создаст сегменты для тех таблиц, у которых их нет (метод materialize_deferred_segments появился в 11.2.0.2):
`dbms_space_admin.materialize_deferred_segments(schema_name => 'ИМЯ_СХЕМЫ');`
8. После этого можно создавать схемы (например, при помощи дампов), а также не забудьте создать (или перенести со старого сервера Oracle) список пользователей Oracle (с их логинами паролями и правами), используемых для подключения в АС «Бюджет».
9. В отличие от предыдущих версий в 11.2 пароли по умолчанию чувствительны к регистру, если Вы считаете что это неудобно, то можете изменить поведение сервера, установив параметр инициализации Oracle SEC_CASE_SENSITIVE_LOGON в FALSE.
10. Некоторые запросы могут работать медленнее, чем в 10-ой версии, из-за новых особенностей планировщика. Для изменения режима работы планировщика нужно изменить значение параметр инициализации OPTIMIZER_FEATURES_ENABLE с 11.2.0.x в значение 10.2.0.5 – в этом случае планировщик станет работать «также как в 10-ой версии». Чтобы эффект проявился сразу после изменения параметра нужно очистить кэш планов – это можно сделать либо перезагрузив сервер Oracle, либо выполнив команду:
`alter system flush shared_pool.`
11. По умолчанию, в 11-ой версии Oracle при создании пользователей используется политика DEFAULT. В этой политике указано, что пароли устаревают через 180 дней – если Вы хотите чтобы пароли не устаревали, то нужно изменить эту политику. Это можно сделать в web-консоли – на закладке Server нужно выбрать Profiles (в разделе Security) – откроется список профилей – в нем нужно выбрать DEFAULT – откроется список свойств профиля – жмем кнопку Edit и редактируем настройку пароля «Expire in (days)», после чего не забываем сохранить.

4. Регистрация пользователей

4.1. Пользователи, права и роли

Каждому пользователю БД Oracle сопоставляется схема с именем, совпадающим с именем пользователя-владельца, которая представляет собой совокупность создаваемых им объектов (таблицы, процедуры, функции, триггеры и т.п.). Кроме того, каждый пользователь БД может создавать объекты и в чужих схемах, сопоставленных другим пользователям (если

такая возможность определена при создании схемы). По умолчанию пользователь БД работает со своей схемой.

Для подключения к базе данных АС «Бюджет» под управлением СУБД Oracle сотрудник должен быть пользователем СУБД и зарегистрирован, как пользователь АС «Бюджет». При регистрации каждый пользователь получает логин и пароль, ему выдаются права доступа на объекты, расположенные в разных схемах, достаточные для работы с АС «Бюджет». При подключении пользователь АС «Бюджет» переключается на ту схему, которую он указал, т.к. в БД может быть несколько схем (например, для хранения данных разных лет).

При создании пустой базы данных Oracle в базе данных уже зарегистрированы пользователи с логинами SYS и SYSTEM, которые дают широкие возможности по администрированию БД.

Пользователь с логином SYS обладает правами SYSDBA (не одно и то же, что пользователь SYSDBA в InterBase (Firebird)). SYSDBA – это особая роль, предназначенная для администрирования БД, которая включает все системные привилегии администратора БД (с флагом ADMIN OPTION) и системную привилегию SYSOPER. Только пользователи с правами SYSDBA могут создавать новую базу данных и подключаться к БД, когда она не открыта, а находится в промежуточном состоянии (например, в состоянии mount (смонтирована)), могут восстанавливать испорченную базу данных, выполнять запуск и остановку базы данных.

Пользователь с логином SYSTEM имеет почти все права администратора БД (роль DBA) для управления уже созданной базой данных и другие привилегии. Он может выделять роли и права другим пользователям (или ролям) или забирать их, создавать пользователей и удалять их, выполнять прочие привилегированные операции.

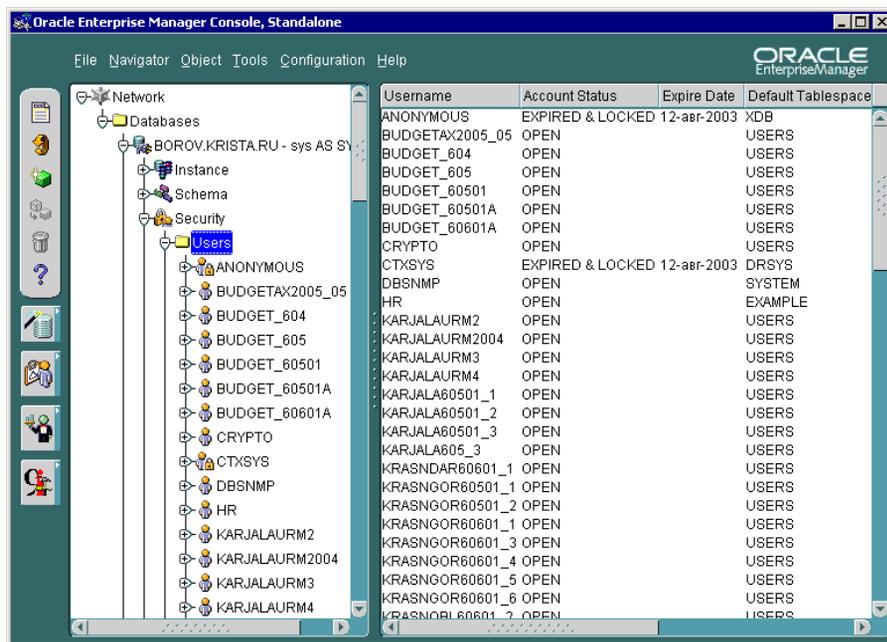
Роль – это именованная группа взаимосвязанных привилегий, которые назначаются администратором БД каждому пользователю СУБД или другой роли. Каждой роли сопоставлен набор прав, в соответствии с которыми пользователь БД будет иметь возможность работать с объектами, использовать утилиты экспорта и/или импорта, быть отключенным или, наоборот, иметь доступ к базе данных при запрете на подключение к базе данных и ограничении сетевого доступа и т.д. Каждому пользователю БД может быть назначено несколько ролей (например, BUDGET_USERS и CONNECT).

4.2. Регистрация пользователей в СУБД Oracle

Для регистрации пользователей в СУБД Oracle можно использовать стандартную утилиту **Oracle Enterprise Manager Console**.

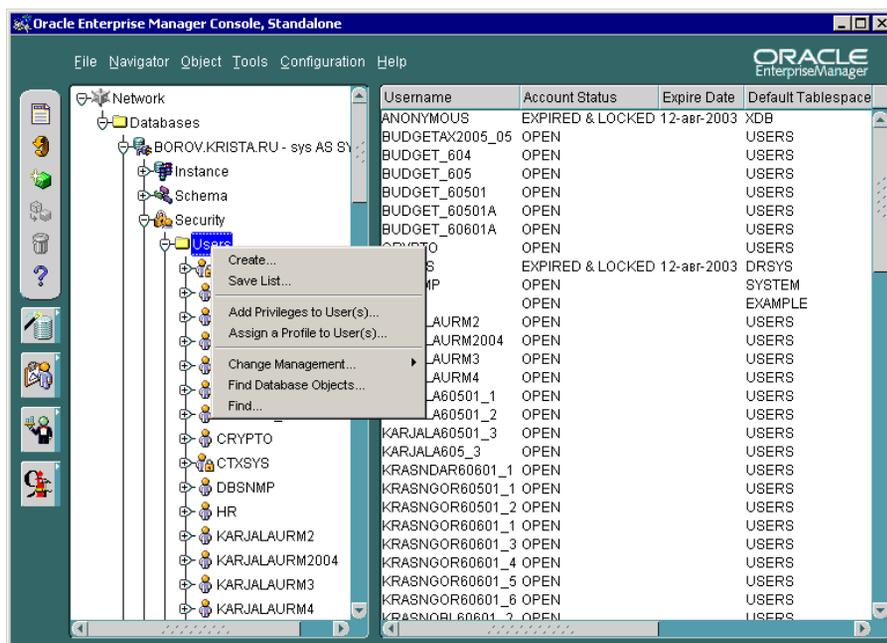
1. Для добавления нового пользователя выберите закладку Security\Users в дереве структуры, расположенном в левой части окна программы «Oracle Enterprise Manager Console».

Рисунок 29 – Окно программы «Oracle Enterprise Manager Console»



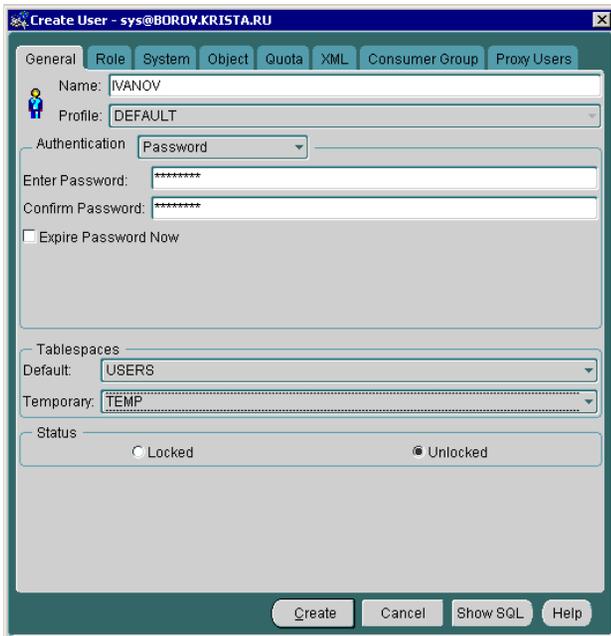
2. С помощью правой кнопки мыши вызовите контекстное меню программы и выберите в нем пункт **Create** («Создать»).

Рисунок 30 – Контекстное меню программы «Oracle Enterprise Manager Console»



3. В появившемся диалоговом окне «Create User» программы выберите закладку General и в обязательном порядке заполните поля:
 - ◆ Name – имя пользователя;
 - ◆ Enter Password – пароль пользователя;
 - ◆ Confirm Password – подтверждение пароля пользователя.

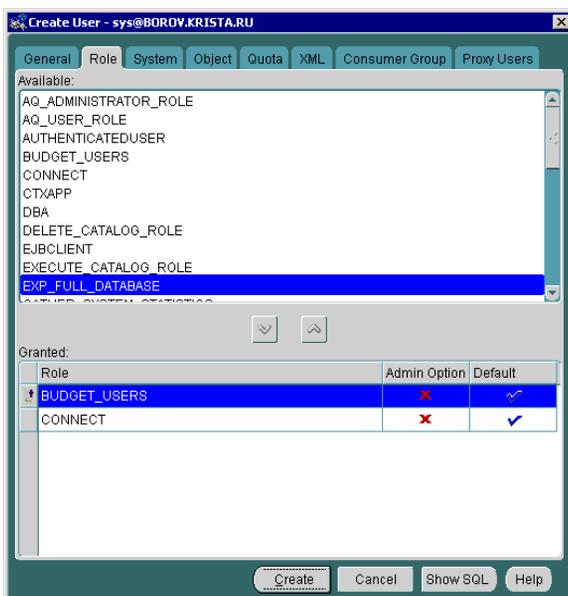
Рисунок 31 – Диалоговое окно «Create User» программы «Oracle Enterprise Manager Console» на закладке «General»



Крайне желательно (хотя и не обязательно) указать временное табличное пространство (место, где будут храниться временные данные пользователя), что влияет только на скорость работы Oracle.

- Выберите в диалоговом окне «Create User» программы закладку Role и укажите роли, доступные пользователю: BUDGET_USERS и CONNECT. Роли выбираются из списка, расположенного в верхней части формы, и перемещаются в нижний список (выбранных ролей) с помощью кнопок, расположенных в центральной части формы, между верхней и нижней таблицей.

Рисунок 32 – Диалоговое окно «Create User» программы «Oracle Enterprise Manager Console» на закладке «Role»

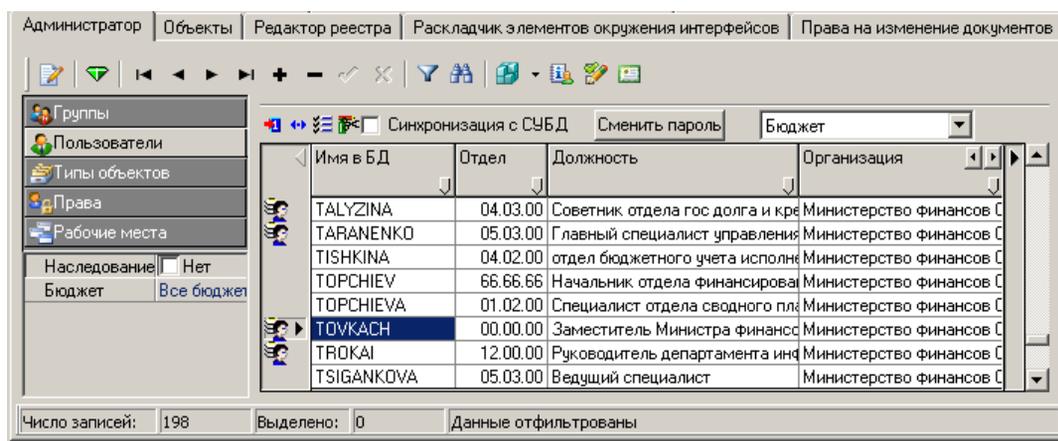


5. Нажмите кнопку **Create** в нижней части диалогового окна.
6. В дальнейшем зарегистрированного в СУБД Oracle пользователя требуется зарегистрировать в АС «Бюджет» в разделе Пользователи интерфейса Администратор РМ Администратор.

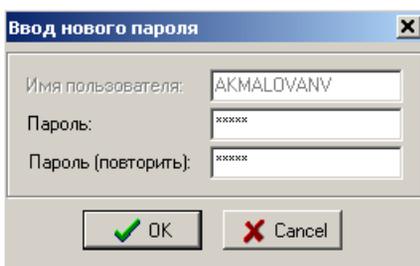
4.3. Регистрация пользователей СУБД Oracle в АС «Бюджет»

Регистрация пользователей СУБД Oracle в АС «Бюджет» производится администратором в разделе Пользователи интерфейса Администратор РМ Администратор в соответствии со стандартным порядком работы.

Рисунок 33 – Вид раздела «Пользователи» интерфейса «Администратор» РМ «Администратор»

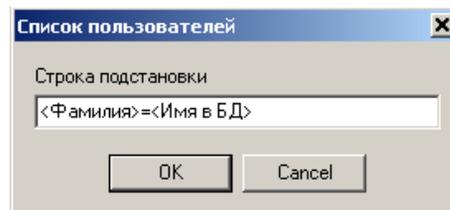


В новой записи заголовка раздела Пользователи интерфейса Администратор вводится сетевое имя пользователя в БД на английском языке. При установленном параметре-флаге **Синхронизация с СУБД** введенное имя нового пользователя АС «Бюджет» будет являться сетевым именем пользователя, одновременно зарегистрированного в используемой СУБД Oracle. При сохранении данных в диалоговом окне устанавливается пароль, который в дальнейшем может быть изменен пользователем с помощью кнопки **Сменить пароль**.



Затем на соответствующих закладках детализации пользователю назначается доступ к рабочим местам, определяются права пользователя на объекты, лицевые счета, типы документов, бюджетную классификацию.

Используя кнопку управления  **Список пользователей**, можно автоматически получить список всех зарегистрированных пользователей АС «Бюджет» и их логинов для настройки дополнительных возможностей ввода в файле **WorkPlace.ini**.



Записи о пользователях, зарегистрированных в СУБД Oracle, отмечаются значком  **Пользователь Oracle**.

5. Миграция с Firebird

Переход с Firebird на Oracle включает в себя несколько этапов.

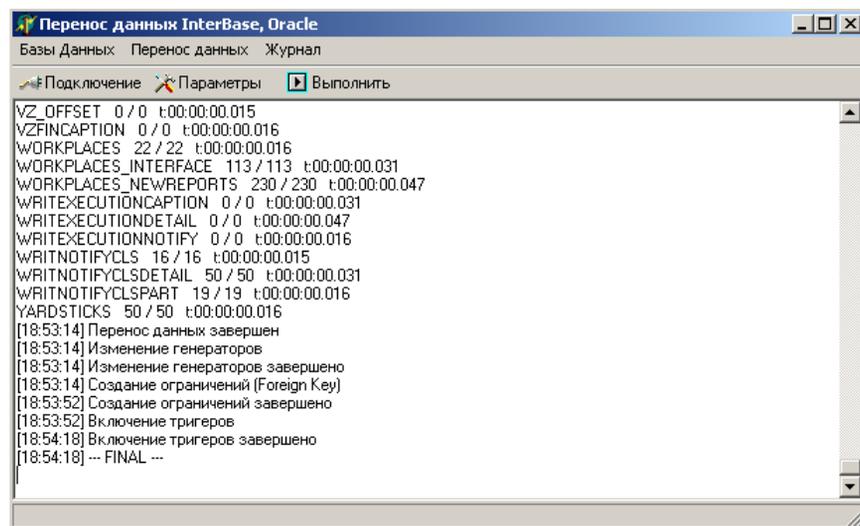
1. Анализ реестра АС «Бюджет» на предмет совместимости пользовательских макросов с Oracle (выполняется специалистами НПО «Криста»).
2. Установка сервера Oracle.
3. Перенос данных из БД Firebird на Oracle.
4. Перенос логинов пользователей с Firebird на Oracle.
5. Изменение файла настроек **workplace.ini** (прописываются подключения к БД).

5.1. Перенос данных из БД Firebird в БД Oracle

❖ Общие сведения

Для переноса данных АС «Бюджет» между БД (схемами) используется утилита DataMigration. Перенос возможен как в существующую, так и в новую схему (схема будет создана утилитой). Новая схема может быть создана по принципу «такая же как...», при этом метаданные будут считаны из схемы указанной в качестве образца, либо по шаблону (в качестве шаблона используется xml-описание метаданных в специальном формате).

Рисунок 34 – Вид окна утилита DataMigration



Алгоритм переноса данных:

1. Определить параметры подключения к БД (см. п.3);
2. Определить параметры переноса данных (см. п. 4);
3. Запустить процесс переноса данных с помощью кнопки  **Выполнить**.

❖ Типовые примеры использования утилиты

Перенос данных из существующей схемы БД одного типа в БД другого типа (Interbase (Firebird) -> Oracle, или Oracle -> Interbase (Firebird)). В этом случае, как правило, подойдет режим «создать новую». Режим «Изменить существующую (данные и метаданные)» может быть использован при переносе данных на Oracle, в том случае, когда нужно создать схему с

нестандартными параметрами: при этом схема создается заранее, а затем в нее переносятся данные при помощи DataMigration. Режим «Изменить существующую (только данные)» может быть использован при переносе в существующую схему Oracle, если эта схема не имеет аналогов, например, в ее метаданные были внесены некие нестандартные изменения (созданы таблицы, добавлены поля или триггера).

При переносе данных между схемами разного типа источником метаданных для создания новой схемы может служить либо файл с описанием структуры метаданных, либо схема того же типа что и изменяемая (целевая). Исходная схема источником метаданных служить не может(!), т.к. структура метаданных сильно зависит от типа СУБД.

Копирование схем (в БД того же типа). Может быть актуально под Oracle, т.к. копирование путем экспорта/импорта потребует двух операций. К тому же при тестировании часто требуется многократно повторять операцию, восстанавливая тестовую схему. В этом случае можно сэкономить на создании метаданных, используя режим «Изменить существующую (только данные)». При выполнении такой операции источником метаданных, для создания новой схемы, как правило, служит схема с данными (хотя можно использовать и другие варианты).

Копирование отдельных таблиц (в целях тестирования или при восстановлении). При этом используется режим «Изменить существующую (только данные)» и задается список переносимых таблиц.

В большинстве случаев рекомендуется выполнять все доступные операции. Однако бывают исключения, например, если при переносе данных возникла проблема с данными одной из таблиц, тогда можно поправить данные в этой таблице, выполнить перенос только этой таблицы, а затем выполнить операции стоящие после переноса данных (перенос генераторов, включение ограничений и триггеров и т.д.).

❖ Настройка подключения к БД

Вызов окна настройки параметров подключения к БД осуществляется при помощи кнопки



Подключение.

Рисунок 35 – Вид окна настройки параметров подключения к БД

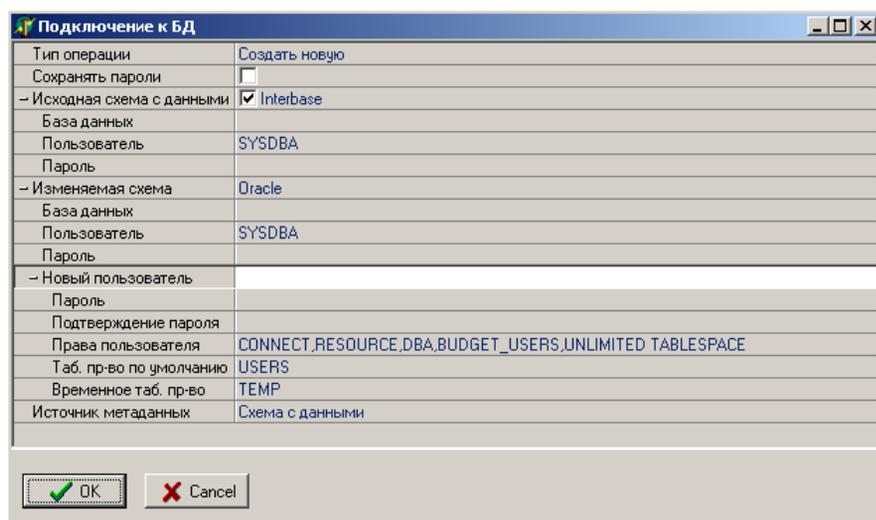


Таблица 1 – Перечень параметров подключения к БД

Название	Описание
----------	----------

Название	Описание
Тип операции	<p>Есть три типа операции:</p> <ul style="list-style-type: none"> Создать новую – создается новая схема и выполняется перенос данных; Изменить существующую (только данные) – выполняется перенос данных в существующую; Изменить существующую (данные и метаданные) – предполагается наличие пустой схемы, в ней создаются метаданные (таблицы, просмотры, триггера и т.д.), а затем выполняется перенос данных
Сохранять пароли	<p>Если поставить галочку, то все введенные в диалоге пароли будут сохраняться в ini-файле. При сохранении пароли преобразуются, т.е. хранятся они не в открытом виде, однако алгоритм преобразования достаточно прост и не гарантирует защиты от взлома. Поэтому хранить пароли рекомендуется только в тестовых средах или в тех случаях, когда есть гарантия того, что файл с настройками не будет доступен злоумышленнику</p>
Исходная схема с данными	<p>Тип исходной СУБД (Interbase или Oracle). Если галочка не установлена, то исходная схема с данными не используется (например, в том случае если мы не хотим выполнять перенос данных, а хотим только создать метаданные, то исходная схема с данными нам и не нужна)</p>
База данных	Файл (для Interbase (Firebird)) или алиас подключения (для Oracle)
Пользователь	Имя пользователя, под которым подключаемся
Пароль	Пароль пользователя, под которым подключаемся
Схема	Название схемы, из которой читаем данные (только для Oracle). Если не задано, то считается равным полю «Пользователь»
Изменяемая схема	Тип СУБД (Interbase (Firebird) или Oracle), над схемой которой выполняются операции
База данных	Файл (для Interbase (Firebird)) или алиас подключения (для Oracle)
Пользователь	Имя пользователя, под которым подключаемся
Пароль	Пароль пользователя, под которым подключаемся
Схема	Название схемы, которую мы изменяем (только для Oracle и только в режимах изменения существующей схемы). Если не задано, то считается равным полю «Пользователь»
Новый пользователь	Название схемы (пользователя), которую мы создаем (только для Oracle в режиме создания новой схемы). Не может быть равно полю «Пользователь» (т.к. он уже создан!)
Пароль	Пароль для создаваемого пользователя (только для Oracle в режиме создания новой схемы)
Подтверждение пароля	Подтверждение пароля создаваемого пользователя (только для Oracle в режиме создания новой схемы). Должно быть идентично полю «Пароль»
Права пользователя	Список прав, которые даются создаваемому пользователю (только для Oracle в режиме создания новой схемы). Права перечисляются через запятую
Таб. пр-во по умолчанию	Табличное пространство по умолчанию для создаваемого пользователя (только для Oracle в режиме создания новой схемы)
Временное таб. пр-во	Временное табличное пространство для создаваемого пользователя (только для Oracle в режиме создания новой схемы)
page_size	Размер страницы в создаваемом файле БД (только для Interbase (Firebird) в режиме создания новой схемы)
Источник метаданных	<p>Тип источника метаданных (только в режиме создания схемы или режиме изменения метаданных):</p> <ul style="list-style-type: none"> Схема с данными – метаданные берутся из той же схемы, что и данные (должна быть задана исходная схема, кроме того тип СУБД исходной схемы должен совпадать с изменяемой); Файл шаблона – файл с описанием структуры метаданных (используется

Название	Описание
	собственный формат файла); <ul style="list-style-type: none"> Другая схема – метаданные берутся из схемы указанной явно (тип СУБД считается равным типу СУБД изменяемой схемы)
Файл шаблона	Имя файла с описанием структуры метаданных (только если источник метаданных «Файл шаблона»)
База данных	Файл (для Interbase (Firebird)) или алиас подключения (для Oracle) (только если источник метаданных «Другая схема»)
Пользователь	Имя пользователя, под которым подключаемся (только если источник метаданных «Другая схема»)
Пароль	Пароль пользователя, под которым подключаемся (только если источник метаданных «Другая схема»)
Схема	Название схемы, из которой читаем данные (только если источник метаданных «Другая схема» и тип изменяемой СУБД Oracle). Если не задано, то считается равным полю «Пользователь»

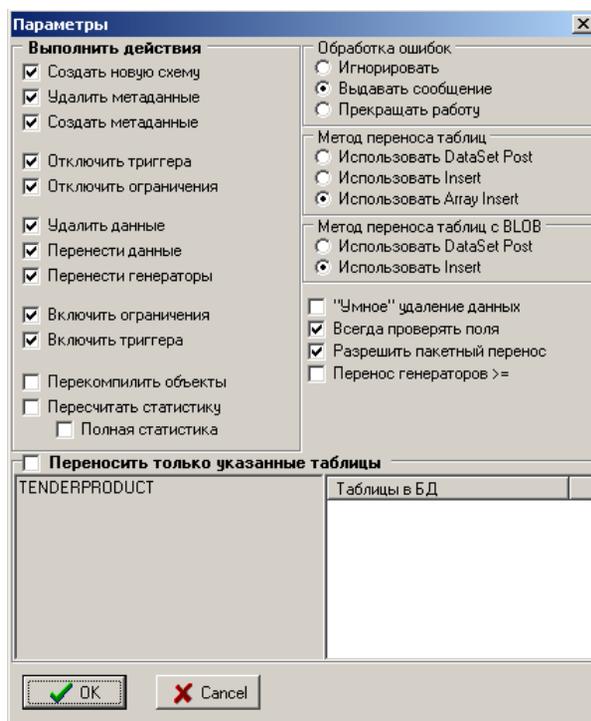
❖ Настройка параметров переноса данных

Вызов окна настройки параметров переноса данных осуществляется при помощи кнопки



Параметры.

Рисунок 36 – Вид окна настройки параметров переноса



1. В списке **Выполнить действия** поставьте галочки у всех пунктов (они перечислены в том порядке, в котором будут выполняться).
2. Настройте **обработку ошибок**. Рекомендуется выбирать пункт «выдавать сообщение», для того чтобы Вы могли принять решение о значимости ошибки.

3. Выберите **метод переноса таблиц**: Dataset Post, Insert или перенос с помощью Array Insert. Рекомендуется использовать Array Insert (т.к. он приблизительно на порядок быстрее) и только при возникновении ошибочных ситуаций рекомендуется применять режим Dataset Post.
4. Выберите **метод переноса таблиц с BLOB**: Dataset Post или Insert. При больших объемах данных рекомендуется использовать Insert (т.к. при большом количестве записей он приблизительно в 2 раза быстрее), при малых объемах или при возникновении ошибочных ситуаций в режиме Insert рекомендуется применять режим Dataset Post.
5. Определите дополнительные настройки:
 - ◆ Умное удаление данных – данные будут удалены только из тех таблиц, которые есть в базе Interbase (Firebird) (эту галку рекомендуется установить, чтобы не были удалены данные в таблицах, которые специфичны только для Oracle).
 - ◆ Всегда проверять поля – дополнительная проверка тождества структур баз данных (если галочка не установлена, то проверка не будет выполняться для пустых таблиц).
 - ◆ Разрешить пакетный перенос – при переносе таблиц, которые содержат более 100 000 записей, загрузка будет выполняться пакетами по 10 000 записей (рекомендуется использовать эту опцию при переносе очень больших таблиц).
 - ◆ Перенос генераторов >= – значения генераторов будут перенесены только в том случае, если их значения больше текущих (настройка предназначена для переноса части данных, при переносе всех данных ее лучше отключить).
6. В нижней части окна настройки можно выбрать таблицы, которые Вы хотите перенести. Слева вводится список таблиц, которые будут переноситься, а справа отображается список таблиц, присутствующих в БД (если Вы подключены к БД). Рекомендуется переносить все таблицы (поэтому галочку Переносить только указанные таблицы рекомендуется снять).

❖ Стандартные ошибки и предупреждения

При работе программы возможна выдача предупреждений (Warning) и сообщений об ошибках (Error). Предупреждения, как правило, не являются критичными для работоспособности полученной схемы.

Таблица 2 – Возможные ошибки и предупреждения

Тип	Текст	Описание	Действие
Warning	Количество пустых строк замененных на текст по умолчанию <NUM>	При переносе пустых строк в Not null поля, во избежание возникновения ошибок, пустые строки заменяются на текст «Не указанное наименование» (текст урезается по длине поля)	Перенос
Warning	В Oracle не найдена таблица <TABLE>	Для переноса данных нужно наличие таблиц в обеих БД. Данное сообщение извещает, что указанная таблица есть в Interbase, но она отсутствует в Oracle	Перенос
Warning	В Interbase не найдена таблица <TABLE>	Для переноса данных нужно наличие таблиц в обеих БД. Данное сообщение извещает, что указанная таблица есть в Oracle, но она отсутствует в Interbase	Перенос
Error	<CODE>, <MESSAGE>	Ошибка возникла при выполнении sql-команды, выводятся номер и описание ошибки, которые были возвращены сервером БД	Везде
Error	В Oracle таблице <TABLE>	Обнаружено различие в структуре таблиц	Перенос

Тип	Текст	Описание	Действие
	не найдено поле <FIELD>		
Error	В Interbase таблице <TABLE> не найдено поле <FIELD>	Обнаружено различие в структуре таблиц	Перенос
Error	В Interbase не найдена таблица <TABLE>	Для переноса данных нужно наличие таблиц в обеих БД. Данное сообщение извещает, что указанная таблица есть в Interbase, но она отсутствует в Oracle. (В отличие от warning данное сообщение выдается, если явно задан список таблиц)	Перенос
Error	В Oracle не найдена таблица <TABLE>	Для переноса данных нужно наличие таблиц в обеих БД. Данное сообщение извещает, что указанная таблица есть в Oracle, но она отсутствует в Interbase. (В отличие от warning данное сообщение выдается, если явно задан список таблиц)	Перенос
Error	(<TABLE>) <MESSAGE>	Неизвестная ошибка при переносе данных из таблицы (перехватывается на try except)	Перенос
Error	(Index: <NUM>, ErrorCode: <CODE>). <MESSAGE>	Ошибка при переносе типа Array Insert. Index - порядковый номер записи в последнем пакете	Перенос

5.2. Перенос логинов пользователей с Interbase (Firebird) на Oracle

Подготовьте полный список пользователей с их паролями. Список пользователей можно взять в **isc4.gdb** (**isc4.gdb** – служебная БД Interbase), в таблице USERS (в этой же таблице хранятся и пароли пользователей, но они хранятся в зашифрованном виде). Если у Вас есть список пользователей без паролей, то пароли нужно сгенерировать каким-либо способом (например, сгенерировать случайные пароли и раздать их пользователям).

При наличии списка логинов (имя пользователя + пароль) можно создать скрипт создания пользователей по шаблону, например с помощью **Excell**:

```
create user ИМЯ identified by «ПАРОЛЬ» default tablespace USERS temporary tablespace TEMP;
grant CONNECT, BUDGET_USERS to ИМЯ.
```

Можно воспользоваться программой **OraTools** (закладка Работа с пользователями, кнопка **Создать списком**), подробнее можно изучить справку к программе **OraTools**.

6. Решение типовых задач по администрированию БД Oracle

Для выполнения каждодневных задач по администрированию и настройке Oracle можно использовать **Oracle Enterprise Manager Console** (входит в поставку Oracle) или его web-версию (в 10-ой и выше версии Oracle), а также утилиту **OraTools** (разработана НПО «Кристалл»). Также для работы с БД можно рекомендовать продукты фирмы QuestSoftware, в частности **Sql Navigator** (удобен для выполнения скриптов при обновлении базы).

Начиная с 10-ой версии Oracle, основным средством администрирования БД является web-консоль. Для выполнения ряда операций (в частности, для остановки и запуска БД) в ОС Windows требуются дополнительные права и привилегии для службы поддержки web-консоли, которая по умолчанию запускается от имени пользователя LocalSystem.

6.1. Использование консоли для задания режима автоматического расширения

1. Запустите консоль: `% oemapp console`.
2. В диалоговом окне при входе выберите автономный режим запуска (launch in standalone).
3. Раскройте требуемую базу данных в папке баз данных.
4. Перейдите на закладку Storage.
5. Раскройте папку **Datafiles**.
6. Выберите файл данных.
7. Измените размер файла.
8. Щелкните на кнопку **Apply**.

Помните, что есть ограничения на размер файла, кроме того, место на диске тоже ограничено.

6.2. Кэширование таблиц Oracle

Объекты БД (таблицы, индексы и т.д.) хранятся в файлах на жестком диске, и при любом обращении к базе данных СУБД подгружает запрашиваемые объекты в оперативную память. Накопление запрашиваемых данных в оперативной памяти с целью их быстрого извлечения по мере надобности называется «кэшированием».

Для увеличения производительности системы можно изменить политику кэширования отдельных таблиц Oracle, установив для них принудительное помещение в буферный кэш (“поднять ряд таблиц в буферный кэш”). В первую очередь, это относится к TB_FACIALFINCAPTION. Если места в буферном кэше достаточно, то можно “поднять” туда и TB_FACIALFINDETAIL.

Изменить политику кэширования можно следующими командами

```
alter table TB_FACIALFINCAPTION cache
/  
alter table TB_FACIALFINDETAIL cache
/
```

Вернуть в прежнее состояние можно следующим образом:

```
alter table TB_FACIALFINCAPTION nocache
/  
alter table TB_FACIALFINDETAIL nocache
/
```

Для ускорения синхронизации АС «УРМ» также рекомендуется изменить режим кэширования у TB_FACIALACC_CLS и REMOTEREADRIGHT – это небольшие, но постоянно используемые таблицы.

Указанные команды можно выполнять во время активной работы с БД (они используют немного ресурсов и меняют, как уже было сказано, только политику кэширования). Также можно изменить политику кэширования других таблиц (например, TB_INCOMES32, TB_PROMISE_CAPTION и TB_PROMISE_DETAIL – если Вы активно работаете с соответствующими интерфейсами), однако следует помнить, что таблицы кэшируются в буферный кэш (Buffer Cache), поэтому он должен иметь достаточный объем (иначе эффект будет отрицательный).

Если объем буферного кэша превышает объем схемы, то можно выставить флажок cache у всех неаудитных таблиц. Приведенный ниже запрос сформирует скрипт для установки флажка cache у всех неаудитных таблиц в схеме текущего пользователя.

```
select 'alter table ' || OWNER || '.' || TABLE_NAME || ' cache;'
from DBA_ALL_TABLES
where OWNER = USER
and TABLE_NAME NOT LIKE 'A@_%' ESCAPE '@'
```

Для того чтобы оценить, сколько занимает тот или иной объект, можно использовать ниже приведенный запрос. Он выдает размер объектов для текущего пользователя (ограничение по пользователю можно убрать – только этот запрос довольно медленный). Размер выдается в блоках табличного пространства и Кб.

```
select OWNER, SEGMENT_NAME, SEGMENT_TYPE, SUM(BLOCKS) BLOCKS, TRUNC(SUM(BYTES) / 1024) SIZE_KB
from DBA_EXTENTS
where OWNER = USER
group by OWNER, SEGMENT_NAME, SEGMENT_TYPE
```

Следует учитывать, что в кэш “поднимаются” не только пользовательские данные, но и системные таблицы Oracle и индексы. При вытеснении из кэша этих объектов результат также может оказаться противоположным ожидаемому. Рекомендуется, чтобы суммарный объем кэшируемых таблиц не превышал бы 25-30% от объема буферного кэша.

Посмотреть содержимое кэша в текущий момент можно с помощью следующего запроса:

```
select o.owner, o.object_name, o.object_type, count(*) cnt
from v$bh bh left join DBA_OBJECTS o on o.object_id = bh.objid
group by o.owner, o.object_name, o.object_type
order by cnt desc
```

Данный запрос вернет размер объектов в блоках буферного кэша (размер блока буферного кэша совпадает с размером блока табличного пространства, из которого взят объект). Теоретически разные табличные пространства могут иметь разный размер блока (но это используется нечасто).

Если объем буферного кэша слишком мал, но имеется свободная физическая память, то объем буферного кэша может быть увеличен.

6.3. Резервное копирование базы данных

❖ Основные средства резервирования БД в Oracle

Повышение надежности системы обеспечивается применением различных средств резервирования БД (резервного копирования) в Oracle, где, как правило, используется два подхода:

- Логическое резервирование («экспорт») – создание дамп-файла;
- Резервирование на физическом уровне – создание «backup» (бэкап).

Экспорт выполняется утилитой **exp.exe**, а обратная ему операция импорта – утилитой **imp.exe**, которые входят в состав дистрибутива клиента Oracle. Также для этих целей можно рекомендовать использование разработанной в НПО «Кристалл» утилиты **OraTools** (см. [п. 7 «Утилита OraTools»](#)), которая играет роль GUI оболочки над вышеуказанными утилитами. При выполнении экспорта сохраняется описание структур и данных в специальном формате. При экспорте можно резервировать отдельные схемы (и даже таблицы). Экспорт может выполняться прямо во время работы пользователей с базой данных, однако следует отметить, что этот процесс сильно нагружает сервер, поэтому рекомендуется выполнять его, когда замедление работы сервера не является критичным.

При переносе данных между БД различных версий следует соблюдать следующие правила:

- Экспорт надо выполнять утилитой `exp.exe`, версия которой соответствует минимальной версии БД (например, если переносим с 10.2.0.1 на 9.2.0.6, то экспорт выполняем утилитой версии 9.2.0.6).
- Импорт следует выполнять утилитой `imp.exe`, версия которой соответствует версии БД в которую производится импорт.
- При переносе данных с версии 9.2.0.5 и выше (в том числе и 10.x.x.x) на версию 9.2.0.4 и ниже см. При экспорте возникала ошибка EXP-00003, что делать?

Если на машине установлено несколько клиентов Oracle, то при запуске импорта/экспорта следует убедиться, что Вы используете утилиты нужной версии (в OraTools рядом с кнопками импорт и экспорт есть выпадающий список, в котором можно выбрать путь к нужной версии утилит).

В 10-ой версии Oracle появились новые утилиты для импорта и экспорта – `impdp.exe` и `expdp.exe`, соответственно. Эти утилиты обладают большим набором возможностей, чем старые `imp.exe` и `exp.exe`. Однако следует помнить, что дампы, полученные новыми и старыми утилитами, несовместимы по формату (новые утилиты не понимают старый формат, а старые – новый). Начиная с версии 10.2.0.4, старые утилиты для работы с дампами (хотя они еще остаются в составе СУБД) снимаются с тех. поддержки.

Бэкап БД Oracle выполняется либо с помощью команд БД, либо с помощью утилиты **RMAN**. Бэкап также можно выполнить вручную: для этого нужно остановить БД и скопировать все необходимые файлы. Бэкап выполняется на физическом уровне, т.е. происходит копирование файлов (или их блоков). В связи с этим минимальными единицами для бэкапа являются табличные пространства или отдельные файлы. Бэкап схемы, которая является логической единицей, выполнить нельзя.

Различают два вида бэкапов: холодный и горячий. Холодный бэкап выполняется при остановленной БД, горячий бэкап выполняется при работающей БД, с которой могут работать пользователи. Холодный бэкап является согласованным, а горячий – несогласованным. Поэтому, чтобы восстановление БД из горячего бэкапа было возможно, требуется, чтобы БД работала в режиме ARCHIVELOG.

Отличия экспорта и бэкапа:

- Бэкап выполняется быстрее, чем экспорт, поскольку он выполняется на физическом уровне.
- Бэкап (в отличие от экспорта) не проверяет логической целостности резервируемых данных.
- Формат файла экспорта является транспортабельным, т.е. не зависит от операционной системы, обладает прямой совместимостью по версиям Oracle (т.е. возможен импорт в БД более свежей версии), а также зачастую и обратной совместимостью по версиям Oracle.



1. Для экспорта и импорта данных требуется наличие Oracle утилит **exp** и **imp**, соответственно. Кроме того, путь к ним должен быть прописан в переменной окружения PATH. Для выполнения данного условия достаточно установить клиента Oracle с типом установки Administrator.

2. Для работы с утилитами необходимо иметь набор прав, позволяющих выполнять экспорт и импорт (такой набор прав есть у ролей EXP_FULL_DATABASE, IMP_FULL_DATABASE; роль DBA включает в себя обе эти роли).

❖ Рекомендации по организации резервного копирования базы данных АС «Бюджет»

Ответственный за резервное копирование

Сотрудником, ответственным за резервное копирование данных, как правило, является системный администратор, обслуживающий сервер баз данных и непосредственно саму базу данных АС «Бюджет».

Периодичность резервного копирования

Для максимально быстрого и полного восстановления данных после сбоя целесообразно производить резервное копирование базы данных ежедневно либо чаще.

Для того чтобы архивная копия содержала данные в некотором стабильном состоянии, создание резервной копии рекомендуется производить в то время, когда с базой данных АС «Бюджет» никто не работает (как правило, в нерабочее время, например, ночью и/или в обеденный перерыв).

Средства для выполнения резервного копирования

Для создания резервной копии базы данных **не допускается** непосредственное копирование файла базы данных. Следует использовать специализированные инструменты создания резервных копий, поставляемые в составе СУБД.

Процесс создания резервной копии может запускаться как непосредственно сотрудником, ответственным за резервное копирование, так и специализированным сервисом, позволяющим выполнять задания по расписанию (для получения типового bat-файла обратитесь в отдел сопровождения АС «Бюджет» support@krista.ru).

Проверка корректности резервного копирования

Необходимо постоянно следить за тем, чтобы резервное копирование выполнялось корректно. Для этого следует выполнять следующие мероприятия:

- ежедневно отслеживать результат завершения работы утилиты, создающей резервную копию;
- периодически (не реже 1 раза в неделю) проверять объем свободного дискового пространства на устройстве, куда сохраняется резервная копия;
- периодически (не реже 1 раза в неделю) проверять корректность самой резервной копии. Для этого следует восстановить базу данных из резервной копии и убедиться в том, что процесс восстановления завершился успешно, а также в том, что восстановленная база данных является работоспособной.

Количество одновременно хранимых резервных копий базы данных

Следует помнить о том, что процесс резервного копирования базы данных может завершиться ошибкой. Поэтому **не допускается** создание очередной резервной копии базы поверх файла с резервной копией базы за предыдущий день.

Рекомендуется одновременно хранить несколько резервных копий базы за разные даты. Существуют различные стратегии хранения резервных копий. В качестве наиболее простого варианта предлагается хранить резервные копии за неделю: на каждый день недели создавать свой файл резервной копии. Таким образом, в большинстве случаев на каждый момент времени будут доступны резервные копии за последние 5 рабочих дней.

Замена рабочей базы данных на восстановленную базу данных

Для улучшения состояния и производительности базы данных в случае использования СУБД FireBird актуально время от времени (не реже 1 раза в неделю) заменять основную базу данных восстановленной (разумеется, только в случае корректного восстановления).

Именно так работает типовой bat-скрипт резервирования базы данных.

Место хранения резервных копий

Следует помнить о том, что резервные копии подвергаются, по сути, тем же рискам, что и сама база данных. Поэтому следует предпринять меры к тому, чтобы в случае разрушения базы данных резервные копии не пострадали:

- **недопустимо** хранить резервные копии базы данных на том же физическом носителе (жестком диске), на котором находится база данных;
- **крайне нежелательно** хранить резервные копии на другом жестком диске той же машины, на которой хранится сама база данных (при выходе из строя материнской платы или блока питания возможен выход из строя всех жестких дисков, подключенных к машине). Рекомендуется, как минимум, хранить резервные копии на другой машине;
- для ценных данных обычно рекомендуется сохранять резервные копии на внешних носителях, а сами внешние носители хранить в другом помещении (либо другом здании), отдельно от машины, где хранится основная база данных. Это необходимо на случай массового выхода компьютеров из строя ввиду, например, скачка напряжения питания, а также на случай пожаров или стихийных бедствий.

❖ Описание процедуры экспорта для одной схемы

Для экспорта одной схемы на машине, где установлен сервер Oracle, или выполнена администраторская установка клиента Oracle, и настроен алиас для подключения к базе данных АС «Бюджет», следует выполнить команду утилиты **exp** (слова, указанные в <>, следует заменить теми, которые соответствуют Вашему случаю):

```
exp USERID=<USER_NAME>/<PASSWORD>@<ALIAS> LOG=export.log FILE=<DMP_FILE_NAME>.DMP  
OWNER=<SCHEMA_NAME> CONSISTENT=Y GRANTS=Y ROWS=Y CONSTRAINTS=Y INDEXES=Y  
TRIGGERS=Y,
```

где:

- USER_NAME – имя пользователя, под которым выполняется подключение к базе данных (крайне рекомендуется, чтобы оно совпадало со SCHEMA_NAME);
- PASSWORD – пароль пользователя;
- ALIAS – алиас базы данных АС «Бюджет»;
- DMP_FILE_NAME – имя файла, в котором будет сохранен дамп;
- SCHEMA_NAME – имя экспортируемой схемы.

Параметры используемой утилиты **exp**:

- USERID – строка подключения;
- LOG – имя файла для вывода лога;
- FILE – имя файла, в котором будет сохранен дамп;
- OWNER – имя экспортируемой схемы;
- CONSISTENT – обеспечить целостность выбираемых данных;
- GRANTS – сохранять информацию о правах (значения: «Y» или «N»);
- ROWS – сохранять данные (значения: «Y» или «N»);

- CONSTRAINTS – сохранять информацию об ограничениях (значения: «Y» или «N»);
- INDEXES – сохранять индексы (значения: «Y» или «N»);
- TRIGGERS – сохранять триггера (значения: «Y» или «N»).



1. Если указать ROWS=«N», то будут восстановлены только метаданные.
2. Так как указан параметр CONSISTENT=«Y», то выполнять экспорт можно в любое время, независимо от того, работает кто-либо с базой данных в данный момент или нет, при этом будет получена целостная копия данных на момент начала экспорта.
3. Если в имени дампа файла (или в пути к нему) есть пробелы, то нужно использовать кавычки (например: FILE=""C:\Program Files\My dump.dmp"").
4. Утилита **exp** может работать и в других режимах, в этом случае могут использоваться другие параметры. Подробнее смотрите руководство, поставляемое в комплекте с СУБД Oracle.

❖ Описание процедуры импорта для одной схемы

Процедура импорта является несколько более сложной. Ее также следует выполнять на машине, где установлен сервер Oracle или выполнена администраторская установка клиента Oracle и настроен алиас для подключения к базе данных АС «Бюджет». Ниже приводится описание порядка выполнения импорта одной схемы:

1. Создайте нового пользователя:

- с помощью утилиты **Oracle Enterprise Manager Console** (см. [п. 4.2 «Регистрация пользователей в СУБД Oracle»](#));
- при помощи команды sql-скрипта (выполнить скрипт можно через утилиту **sqlplus**):
sqlplus "SYS/password@budserv as sysdba" @user.sql BUDGET_COPY.

Пример скрипта **user.sql** приведен ниже:

```

spool user.lst
create user &&1 identified by masterkey default tablespace users temporary tablespace temp
/
grant connect, resource, dba, budget_users, UNLIMITED TABLESPACE to &&1
/
SPOOL OFF
exit
/

```

2. Импортируйте данные в схему этого пользователя, выполнив команду (слова, указанные в <>, следует заменить теми, которые соответствуют Вашему случаю):

```

imp USERID=<USER_NAME>/<PASSWORD>@<ALIAS> LOG=import.log FILE=<DMP_FILE_NAME>.DMP
FROMUSER=<FROMUSER> TOUSER=<TOUSER> GRANTS=Y ROWS=Y CONSTRAINTS=Y INDEXES=Y,

```

где:

- ♦ USER_NAME – имя пользователя, под которым выполняется подключение к базе данных (крайне рекомендуется, чтобы оно совпадало с TOUSER);
- ♦ PASSWORD – пароль пользователя;
- ♦ ALIAS – алиас базы данных АС «Бюджет»;
- ♦ DMP_FILE_NAME – имя файла, в котором будет сохранен дамп;
- ♦ FROMUSER – имя экспортируемой схемы;
- ♦ TOUSER – имя схемы, в которую выполняется импорт.

Параметры используемой утилиты **imp**:

- ◆ USERID – строка подключения;
- ◆ LOG – имя файла для вывода лога;
- ◆ FILE – имя файла, в котором будет сохранен дамп;
- ◆ FROMUSER – имя экспортируемой схемы;
- ◆ TOUSER – имя схемы, в которую выполняется импорт;
- ◆ GRANTS – сохранять информацию о правах (значения: «Y» или «N»);
- ◆ ROWS – сохранять данные (значения: «Y» или «N»);
- ◆ CONSTRAINTS – сохранять информацию об ограничениях (значения: «Y» или «N»);
- ◆ INDEXES – сохранять индексы (значения: «Y» или «N»).



1. Если указать ROWS=«N», то будут восстановлены только метаданные.
2. Если в имени дампа файла (или в пути к нему) есть пробелы, то нужно использовать кавычки (например, FILE=""C:\Program Files\My dump.dmp"").
3. Утилита **imp** может работать и в других режимах, в этом случае могут использоваться другие параметры. Подробнее смотрите руководство, поставляемое в комплекте с СУБД Oracle.



Из-за сбоев в работе СУБД Oracle, дефектов жестких дисков, сбоев в питании структура базы данных или полученного дампа-файла может быть нарушена. Поэтому, после того, как Вы получите дампа-файл, обязательно убедитесь, что он подлежит восстановлению с помощью импорта.

6.4. Получение данных на момент времени в прошлом, аудит изменений

Возможность получения пользователем данных на некоторый момент в прошлом с помощью запроса называется **FlashBack** и может использоваться для восстановления некорректно измененных данных. Технология **Flashback** использует данные табличного пространства UNDO, где по умолчанию данные хранятся не менее 3 часов (для 9-ой версии, в 10-ой версии Oracle данный интервал был сокращен), поэтому просмотреть старые данные можно только непосредственно после их изменения.

Например, можно выбрать данные на указанный момент времени:

```
SELECT * FROM ACC_CLS AS OF TIMESTAMP TO_TIMESTAMP ('29.12.2006 12:00:00', 'DD.MM.YYYY HH24:MI:SS')
```

Также можно посмотреть, как соответствуют SCN (системные номера контрольных точек) и моменты времени (в smon_scn_time указаны не все SCN, а через определенные промежутки времени с интервалом 10-15 минут):

```
SELECT * FROM sys.SMON_SCN_TIME order by SCN_BAS desc.
```

Зная SCN, можно выбрать данные, указав значение SCN:

```
SELECT * FROM ACC_CLS AS OF SCN 100000.
```

При наличии **Enterprise Edition** возможности **FlashBack** позволяют не только просматривать старые данные в таблицах, но и восстанавливать таблицы после удаления (или выполнения операции TRUNCATE). Начиная с 10-ой версии Oracle доступны следующие возможности:

- Получение SCN для указанного момента времени.

```
SELECT TIMESTAMP_TO_SCN(TO_TIMESTAMP('29.12.2006 12:00:00', 'DD.MM.YYYY HH24:MI:SS')) FROM dual
```

- Обратное преобразование SCN во время выполняется функцией SCN_TO_TIMESTAMP.
- Кроме просмотра данных на конкретный момент времени, можно посмотреть список изменений за указанный временной интервал. Приведенный ниже запрос выбирает все версии записей из предметной таблицы accountplan за последние 3 часа. Обратите внимание на наличие в запросе служебных (виртуальных) столбцов versions_startscn, versions_starttime, versions_endscn, versions_endtime, versions_xid, versions_operation, которые содержат информацию о времени жизни версии и номер транзакции, которая создала эту версию. Нижняя граница в запросе не может быть меньше, чем текущий момент времени минус значение параметра undo_retention:

```
SELECT versions_startscn, versions_starttime, versions_endscn, versions_endtime, versions_xid,
versions_operation, id, name, state, code1, code2, createdate, updatedate FROM accountplan VERSIONS
BETWEEN TIMESTAMP SYSDATE - 3/24 AND SYSDATE.
```

По идентификатору транзакции можно получить дополнительную информацию (в частности, об имени пользователя, который эту операцию выполнил):

```
SELECT * FROM flashback_transaction_query WHERE xid = HEXTORAW ('06002D0055100000')
```

Таблица flashback_transaction_query содержит информацию обо всех изменениях, выполненных над БД. Ее просмотр может быть удобной альтернативой просмотра журналов REDO-логов. Более широкие возможности предоставляет **LogMiner**, содержащий данные о том, с какой машины пришел sql-запрос.

- Получение «conservative upper bound SCN» – виртуального поля ORA_ROWSCN. Значение этого поля равно SCN последнего изменения блока, в котором расположена запись, т.е. если поменять одну запись в блоке, то изменится SCN у всех записей блока. Данное поле можно использовать для обнаружения изменившихся таблиц.

6.5. Перенос пользователей из одной БД Oracle в другую

Благодаря особенностям хранения пользовательских логинов возможен перенос пользователей между СУБД Oracle разных версий.

Рассмотрим пример, иллюстрирующий принцип хранения пользовательских логинов.

Создаем пользователя:

```
fbsd> create user u2 identified by u2_password;
```

User created.

```
fbsd> select username, password from dba_users where username='U2';
```

```
USERNAME          PASSWORD
-----
```

```
U2                A1E4E366FDB8A81A
```

На другой машинке:

```
tst> create user u2 identified by values 'A1E4E366FDB8A81A';
```

User created.

```
tst> grant connect to u2;
```

Grant succeeded.

```
tst> connect u2/u2_password@tst
```

Следует учесть, что хэш пароля зависит от имени пользователя, но не зависит от версии БД. Для переноса пользователей АС «Бюджет» необходимо выполнить 2 скрипта: первый вернет список команд создания пользователей, второй – список команд для раздачи прав.

```
select 'create user ' || username || ' identified by values "' || password
|| '" default tablespace users temporary tablespace temp;'
from dba_users
where username in (
select GRANTEE
from dba_role_privs
where GRANTED_ROLE = 'BUDGET_USERS'
minus
select GRANTEE
from dba_role_privs
where GRANTED_ROLE = 'DBA' )
select 'grant connect, BUDGET_USERS to ' || username || ';'
from dba_users
where username in (
select GRANTEE
from dba_role_privs
where GRANTED_ROLE = 'BUDGET_USERS'
minus
select GRANTEE
from dba_role_privs
where GRANTED_ROLE = 'DBA' )
```

6.6. Запрет/разрешение подключений к базе данных, ограничение сетевого доступа

Ряд операций, связанных с изменениями или анализом структуры схемы, такие, как выполнение SQL-скриптов исправления или обновления базы, должны выполняться только в режиме монопольного использования базы данных, т.е. все пользователи должны быть отключены на это время от базы данных.



Очень распространена ошибочная ситуация, когда администратор просит пользователей отключиться от базы данных, но на своем рабочем месте не закрывает АС «Бюджет», не считая себя пользователем.

Для исключения возможности подключения к базе данных в момент выполнения над ней указанных операций следует перевести ее в монопольный режим использования, а после завершения операции вновь разрешить общий доступ к базе данных.



Недостатком всех описанных ниже методов является то, что пользователи отключаются от всей базы, а не только от нужной схемы.

❖ Ограниченный доступ

ВКЛ.(Alter system enable restricted session)/**ВЫКЛ.** (Alter system disable restricted session).

В этом режиме могут подключаться только пользователи с привилегией Restricted session, но существующие подключения закрыты не будут. Поэтому, если использовать ограниченный доступ, то потребуется отключить все незначащие session после включения режима ограниченного доступа.

Проверить, что база данных находится в режиме ограниченного доступа, можно запросом: select logins from v\$instance. При этом возврат значения «ALLOWED» означает обычный режим, возврат значения «RESTRICTED» означает режим ограниченного доступа.



Как правило, привилегия Restricted session есть у роли DBA, и, следовательно, у пользователей SYS, SYSTEM, SYSDBA, а также у владельцев схем.

❖ Заморозка

ВКЛ. (Alter system quiesce restricted)/**ВЫКЛ.:**(Alter system unquiesce).

После заморозки все запросы к базе данных блокируются (кроме сессий SYS и SYSTEM), т.е. пользователь будет ждать ответа до разморозки базы данных. При переводе в режим заморозки база данных будет ожидать завершения выполняющихся запросов и транзакций и только после их завершения выдаст сообщение об успешном переводе базы данных в режим quiesce.

Проверить, что БД находится в режиме заморозки, можно запросом: select active_state from v\$instance. При этом возврат значения «NORMAL» означает обычный режим; возврат значения «QUIESCING» означает, что ожидается переход в режим заморозки (ждем завершения активных транзакций); возврат значения «QUIESCED» означает, что база данных находится в режиме заморозки.

Определить, в каких сессиях выполняются транзакции, завершение которых ожидается для перехода в режим заморозки, можно запросом:

```
select SID, EVENT, TOTAL_WAITS, TIME_WAITED "TIME WAITED[100 OF SEC]", AVERAGE_WAIT
from V$SESSION_EVENT
where EVENT='wait for possible QUIESCE finish'.
```



Данный режим доступен только в **Enterprise Edition**.

❖ Использование скриптов mono_enable.sql и mono_disable.sql

ВКЛ. (выполнить **mono_enable.sql**) /**ВЫКЛ.** (выполнить **mono_disable.sql**)

Скрипт **mono_enable.sql** выполняет следующие действия:

- Отключает от базы данных пользователей с ролью Budget_Users (кроме SYS и SYSTEM);
- Закрывает доступ к базе данных (LOCK) всем пользователям с ролью Budget_Users (кроме SYS и SYSTEM).

```
declare
cursor c$ is
select 'ALTER SYSTEM KILL SESSION "'||to_char(SID)||','||to_char(SERIAL#)||'" IMMEDIATE' as cmd
from v$session
where username in
(
SELECT grantee
```

```

FROM sys.dba_role_privs
where granted_role = 'BUDGET_USERS'
and grantee not in ('SYS', 'SYSDBA')
)
;
begin
for r$ in c$ loop
execute immediate r$.cmd;
end loop;
end;
/
declare
cursor c$ is
SELECT 'ALTER USER ' || grantee || ' ACCOUNT LOCK' as cmd
FROM sys.dba_role_privs
where granted_role = 'BUDGET_USERS'
and grantee not in ('SYS', 'SYSDBA')
;
begin
for r$ in c$ loop
execute immediate r$.cmd;
end loop;
end;
/

```

Скрипт **mono_disable.sql** открывает доступ к базе данных всем пользователям с ролью Budget_Users.

```

declare
cursor c$ is
SELECT 'ALTER USER ' || grantee || ' ACCOUNT UNLOCK' as cmd
FROM sys.dba_role_privs
where granted_role = 'BUDGET_USERS'
and grantee not in ('SYS', 'SYSDBA')
;
begin
for r$ in c$ loop
execute immediate r$.cmd;
end loop;
end;
/

```

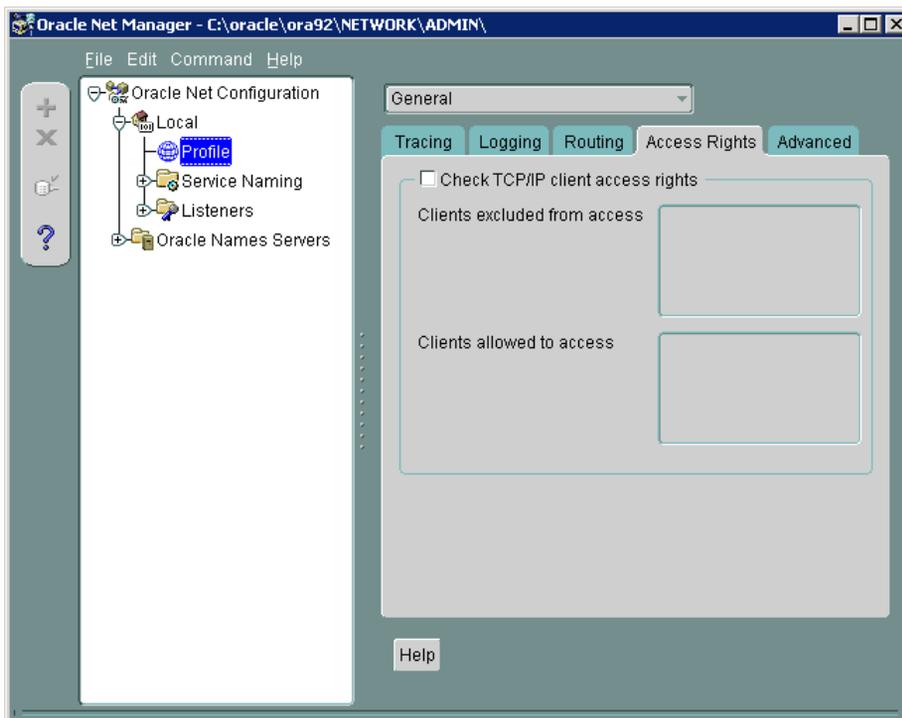


1. Отключение выполняется с опцией **IMMEDIATE**, т.е. скрипт не ждет завершения транзакций. Данный режим доступен только в **Enterprise Edition**.
2. При выполнении скрипта **mono_disable.sql** будет открыт доступ к базе данных всем пользователям с ролью **Budget_Users**, даже тем, доступ которым был закрыт вручную (до вызова скрипта **mono_enable.sql**).
3. Если пользователь, выполняющий скрипт **mono_enable.sql**, не является **SYS** или **SYSTEM**, то он тоже будет отключен (в этом случае вторая часть скрипта, которая закрывает доступ к базе данных, скорее всего не будет выполнена).
4. Пользователи, которые не обладают ролью **Budget_Users**, не будут отключены и смогут работать с БД.

❖ Межсетевой экран

Еще один из вариантов ограничения сетевого доступа – это межсетевой экран (firewall) или настройка службы **Listener** на ограничение по подключению с определенных IP-адресов.

*Рисунок 37 – Окно «Oracle Net Manager» с настройками службы **Listener**-а для ограничения подключений*



7. Утилита OraTools

Утилита **OraTools.exe** – это инструмент администрирования и настройки схем Oracle.

7.1. Запуск программы и подключение к БД

При запуске программы **OraTools.exe** появляется диалоговое окно подключения к БД. Для работы с программой требуется подключиться под пользователем, которому выдана роль **DBA** (т.е. пользователем, у которого есть администраторские права). Ввод логина и пароля

пользователя производится в полях UserName и Password соответственно. Значение поля Database выбирается из выпадающего списка (его также можно ввести вручную) – это имя псевдонима подключения Oracle. Значение поля Connect as для пользователя SYS нужно установить «SYSDBA» (значение выбирается из списка), для всех остальных пользователей, как правило, нужно устанавливать значение «Normal».

Если в поле Database список псевдонимов пуст, или не содержит нужного, то его можно добавить с помощью кнопки с тремя точками  в строке. После нажатия кнопки появится диалоговое окно, в котором следует нажать кнопку  **Добавить подключение** (вверху формы), затем в окне подключения ввести имя подключения (псевдоним), имя компьютера, на котором установлен сервер Oracle (HOST), и имя базы данных (SERVICE_NAME) (не путать с именем схемы!). Если используется нестандартный порт (не 1521), то его можно поменять после добавления нового подключения в этом же диалоговом окне, путем редактирования свойств подключения (см. также **настройки TNS**).

Если при работе программы возникают ошибки, то стек вызовов сохраняется в файле OraTools.trc (этот файл рекомендуется прилагать к сообщению об ошибках программы).



1. Для импорта и экспорта данных требуется наличие Oracle утилит **imp.exe** и **exp.exe**, соответственно. Путь к ним должен быть прописан в переменной окружения PATH. Для выполнения данного условия достаточно установить клиента Oracle с типом установки Administrator. Следует помнить, что при переносе данных путем импорта/экспорта с БД одной версии на БД другой версии следует правильно выбирать версию утилит **imp/exp** (подробнее см. документацию по Oracle).
2. Для работы с программой требуется подключаться под пользователем, которому выдана роль **DBA**.
3. В дальнейшем будем считать что схема – это пользователь, у которого есть объекты.

❖ Пароли и безопасность

Часто при работе с БД бывает важно обеспечить секретность пароля. В связи с этим следует отметить, что пароли, вводимые в настройках экспорта и импорта (и только они), сохраняются в ini-файле (пароли хранятся в преобразованном виде, но их декодирование является простой задачей). В связи с этим можно порекомендовать, во-первых, изменять пароль пользователя после завершения операции импорта. Во-вторых, не хранить пароль в настройках экспорта: при выполнении данной операции пароль будет запрошен и его можно будет ввести.

При решении обратной задачи (т.е. для облегчения дальнейшей работы) можно рекомендовать указать пароль, запрашиваемый для подключения к БД при запуске программе, в ini-файле (здесь следует отметить, что пароль подключения не сохраняется автоматически, но считывается, если он указан). Добавить пароль можно в секцию Connect, следующим образом: **Password=masterkey**.



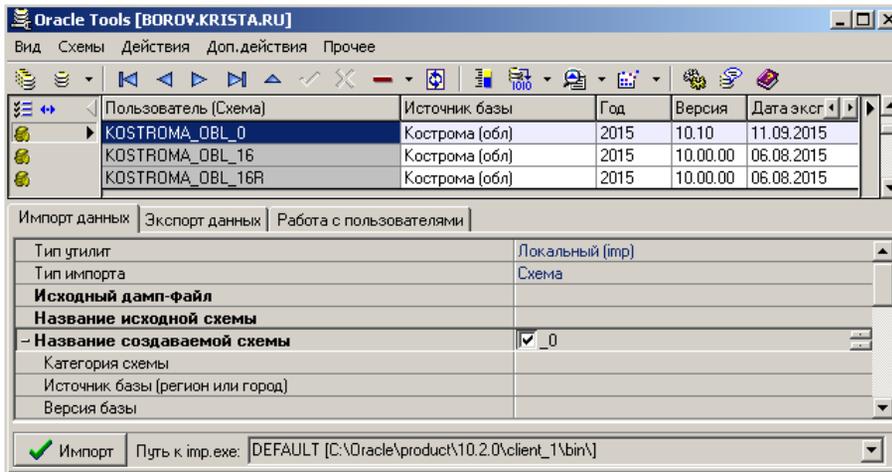
Пример

```
[Connect]
Database=TEST1.KRISTA.RU
Username=SYSDBA
Password=masterkey
ConnectAs=0
```

7.2. Структура окна программы и принципы работы

В самой верхней части окна программы расположена строка заголовка, в которой слева направо размещены кнопка вызова стандартного меню действий над окном программы  (свернуть, переместить, восстановить, закрыть), название программы, имя псевдонима подключения Oracle.

Рисунок 38 – Вид окна программы OraTools.exe на закладке «Импорт данных»



Ниже заголовка расположены главное меню программы, панель кнопок управления, рабочая область, включающая таблицу для просмотра информации о пользователях и схемах, и 4 закладки для переключения между страницами программы: Импорт данных, Экспорт данных, Работа с пользователями. При этом большинство функций можно вызвать как из главного меню, так и через панель кнопок управления.

В верхней таблице отражается перечень соответствия существующих схем и пользователей, а закладки предназначены для настройки параметров экспорта, импорта и т.п. В поле статуса таблицы выводятся значки, отражающие информацию из системных таблиц ( **Со схемой работают [N]**,  **Пользователю назначена роль [Budget_Users]**,  **Стандартный пользователь Oracle**, др.). Строки таблицы окрашены в разные цвета для выделения пользователей, имеющих схемы, пользователей без схем, а также наделенных различными правами.

❖ Главное меню

Главное меню окна программы **OraTools.exe** предназначено для выполнения различных действий над схемами, доступно на всех закладках и содержит набор следующих команд: **Вид**, **Схемы**, **Действия**, **Доп. действия**, **Прочее**. Выбор любой команды производится щелчком мыши по ее названию, при этом появляется выпадающий список команд, обеспечивающих непосредственно работу с программой.

- Пункт меню **Вид** содержит следующие команды, регулирующие режим отображения пользователей в зависимости от наличия схем и назначенных прав:
 - ♦  **Пользователи имеющие схемы** – отображение в рабочей области только тех пользователей, которые имеют схемы (соответствует функции кнопки  **Тип пользователя: Пользователи имеющие схемы**);
 - ♦  **Пользователи без схем** – отображение только пользователей без схем (соответствует функции кнопки  **Тип пользователя: Пользователи без схем**);

- ♦  **Все пользователи** – отображение всех пользователей (соответствует функции кнопки  **Тип пользователя: Все пользователи**);
- ♦  **Показывать все схемы** – включение/отключение отображения всех пользователей, включая системных пользователей (соответствует функции одноименной кнопки);
- Пункт меню **Схемы** содержит следующие команды, отражающие разновидности режима удаления (соответствуют командам одноименной кнопки):
 - ♦  **Удаление выделенных пользователей** – стандартный режим удаление выделенных в рабочей области пользователей одной командой (при этом пользователь либо удалится полностью, либо не изменится);
 - ♦  **Быстрое удаление выделенных пользователей** – режим удаления, при котором сначала выполняется удаление таблиц, а затем удаление пользователя (если возникнут ошибки, то в базе может остаться частично удаленная схема);
 - ♦  **Удаление схемы (пользователь не удаляется)** – удаление всех объектов, принадлежащих пользователю (сам пользователь при этом не удаляется);
 - ♦  **Удаление аудита в АС «Бюджет»** – удаление аудита АС Бюджет (данные и метаданные), если он есть в схеме (остальные объекты не изменяются);
 - ♦  **Удаление схем по списку...** – одновременное удаление нескольких схем, путем задания списка на удаление в появившемся диалоговом окне;
 - ♦  **Обновить (F5)** – обновление информации о схемах, выводимую в рабочей области, данная команда присутствует только в меню программы.
- Пункт меню **Действия** содержит следующие команды, соответствующие функциям одноименных кнопок или их команд, которые производятся для выделенных в таблице пользователей:
 - ♦  **Подсчет размера схем** – расчет места занимаемого объектами схем. Результат выводится в колонку Размер, МБ таблицы для выделенного пользователя;
 - ♦  **Простая компиляция** – компиляция всех invalid-объектов в порядке, не учитывающем возможные зависимости между ними (по умолчанию);
 - ♦  **Oracle компиляция** – компиляция объектов (view, procedure, triggers, package, function) с помощью стандартной процедуры Oracle;
 - ♦  **Статистика типа GATHER AUTO** – сбор статистики только для тех объектов, которые изменились (по умолчанию);
 - ♦  **Оценочная статистика** – сбор статистики для таблиц с большим количеством записей выполняется не по всем записям, а по некоторой выборке (ее размер Oracle определяет сам);
 - ♦  **Полная статистика** – сбор статистики по всем записям;
 - ♦  **Настраиваемая статистика...** – вызов диалогового окна с возможностью выбора режима сбора статистики, где, кроме вышеуказанных, доступно еще **Удаление статистики** и **Сбор статистики командой ANALYZE**. В данном диалоге можно указать процент сбора для оценочной статистики, а также строку параметров (для оценочной и полной статистики).
- Пункт меню **Доп. действия** содержит следующие команды:

- ◆  **Смена табличного пространства** – замена имен табличных пространств (ТП) в dmp-файле (файл при этом изменяется, поэтому лучше сделать копию). Допускается замена только на более короткие имена (например, BUD2010 можно заменить на USERS, а наоборот нельзя). Операция нужна для того, чтобы можно было импортировать дампы с блобами, которые были экспортированы из табличного пространства, отсутствующего в целевой БД. В появившемся диалоговом окне следует заполнить поле Имя dmp-файла, выбрав путь к дамп-файлу в стандартном окне поиска. При нажатии кнопки  **Скан** окна выполнится проверка дампа, и в строке Исходное ТП появится выпадающий список с именами табличных пространств, которые встречаются в дампе. При нажатии кнопки  **Менять** выполнится замена имени табличного пространства Исходное ТП на Новое ТП (если Исходное ТП не указано, то будет производиться замена всех имен табличных пространств в файле на Новое ТП). Выполнять сканирование перед сменой ТП не обязательно.
- ◆  **Экспорт описания в Excel** – осуществляет экспорт в Excel таблицы программы с перечнем пользователей и схем;
- ◆  **Блокировки** – вызов модального окна с информацией о блокировках на текущий момент времени. В окне присутствуют четыре закладки:
 - Заблокированные объекты – список объектов БД, которые блокируются пользователями. Как правило, объект блокируется не полностью (например, в таблице может быть заблокирована только часть записей, а не вся таблица);
 - Заблокированные сессии – список сессий, которые ждут завершения транзакции в другой сессии, чтобы продолжить свою работу;
 - Блокировки (DBA_LOCK) – список блокировок БД из просмотра DBA_LOCK;
 - Блокировки (by AskTom) – список блокировок БД полученный с помощью запроса из конференции AskTom.

Кнопка управления  **Закреть** предназначена для закрытия окна «Блокировки»;

Кнопка  **Обновить** служит для обновления информации в окне.

- Пункт меню **Прочее** содержит следующие команды:
 - ◆  **Настройки программы** – вызов диалогового окна «Настройки», в котором осуществляется определение основных параметров работы программы (подробнее смотрите п. 7.7 [«Настройка программы OraTools.exe»](#));
 - ◆  **Информация к размышлению** – вызов диалогового окна, содержащего информацию о версии программы и Oracle, а также три закладки с полезной информацией. В разделе общей информации присутствует параметр Первичный ORA_HOME – он указан в виде <имя>=<путь>. Если на компьютере установлено более одного клиента Oracle, то первичный HOME можно менять, выбирая нужный из выпадающего списка (только для WIN NT систем, для WIN 9x – такая возможность не предусмотрена).

На закладке Табличные пространства отображается объем табличных пространств и объем свободного места в них. Если свободного места в табличных пространствах мало, то это (при отсутствии автоматического увеличения) может привести к сообщениям о недостатке свободного места (особенно при импорте).

На закладке Сессии отображается список подключений (сессии). В данном окне не отображаются служебные сессии Oracle. В нижней части информационного окна приводятся данные о количестве запущенных сессий (первое число – это количество отображаемых, а в квадратных скобках дается полное количество сессий),

количестве активных сессий и числе «процессов» Oracle. Там, при наличии администраторских прав, можно выполнить принудительное отключение пользователя из меню, вызываемого по правой кнопке мыши. Также можно отправить сообщение выбранным пользователям с помощью команды net send.

На закладке Отчеты предоставляется возможность генерации отчетов в виде HTML-файлов, содержащих информацию о характеристиках работы, настройках БД и запросах. При возникновении проблем с производительностью БД следует сформировать отчеты и выслать их специалистам НПО «Криста». На закладке располагаются следующие кнопки формирования соответствующих доступных отчетов:

-  **Общая информация** – отчет, содержащий общую информацию о настройках БД и ее работе с момента старта БД. Поэтому если с момента запуска БД прошло несколько дней, то данный отчет можно формировать в любое время. Если БД была перезагружена недавно, то для сбора данных требуется несколько часов работы пользователей с БД в обычном режиме;
-  **Список сессий** – отчет, содержащий сведения о нагрузке, создаваемой пользовательскими сессиями в текущий момент (поэтому его нужно формировать в момент работы пользователей с БД);
-  **Самые "толстые" SQL** – отчет, содержащий информацию о наиболее ресурсоемких sql-запросах, выполненных за последнее время. При его формировании крайне желательно установить флаг-галочку у параметра План выполнения запроса для получения более подробной информации. Поскольку данные о старых запросах со временем вытесняются из памяти, то формировать этот отчет лучше после продолжительной работы БД в интересующем Вас режиме;
-  **Мониторинг событий** – отчет, содержащий информацию о нагрузке БД в указанный период времени. Формировать этот отчет нужно во время работы БД в интересующем Вас режиме. Период сбора данных должен быть более 30 минут. Для получения мониторинга событий необходимо нажать данную кнопку дважды: первый раз – для начала сбора данных, а затем через некоторое время еще раз – для прекращения сбора данных, после этого отчет сохраняется в файл;
-  **Размер объектов БД** – отчет, содержащий информацию о размере объектов БД или выбранной схемы (справа от кнопки находится параметр-флаг Только выбранная схема и выпадающий список с именами схем). При наличии большого числа схем выборка данных по всей БД может выполняться достаточно медленно.

Внизу на закладке Отчеты присутствуют дополнительные параметры формирования отчетов:

- Сохранять отчеты в каталог – путь к каталогу, в который автоматически будут сохраняться сформированные отчеты (имя файла при этом формируется автоматически). Если флаг-галочка не установлена, то после завершения формирования отчета будет появляться диалог с запросом имени файла для сохранения;
- Показывать результаты – если установлена флаг-галочка, то полученный HTML-файл будет открыт для просмотра сразу после завершения формирования отчета.

- ♦  **Настройки TNS** – вызов диалога редактирования TNS–подключений (псевдонимов для подключения к Oracle). Это диалоговое окно также можно вызвать при подключении к Oracle. Информация о подключениях хранится в файле tnsnames.ora. Основными параметрами подключения являются:
 - Имя подключения (или псевдоним), которое в дальнейшем используется для указания на данное подключение (со всеми его настройками);
 - PROTOCOL – сетевой протокол, используемый для связи с сервером (в большинстве случаев это TCP);
 - HOST – имя компьютера, на котором установлен сервер Oracle (это параметр протокола TCP, для других протоколов он может отсутствовать);
 - PORT – TCP-порт, на котором функционирует прослушиватель (Listener), как правило, 1521 (это параметр протокола TCP, для других протоколов он может отсутствовать);
 - SERVICE_NAME – имя базы данных (не путать с именем схемы!).
- ♦  **Справка о программе** – вывод справочной информации и описания программы;
- ♦ **Директория с логами...** – открытие каталога, в котором хранятся лог-файлы;
- ♦  **Задания...** – при нажатии кнопки выводится диалог, в котором можно посмотреть информацию о выполняющихся в данный момент заданиях DataPump и выполнить ряд операций:
 -  **Подключиться к процессу DataPump** – подключение к процессу и слежение за ходом его выполнения;
 -  **Удалить остатки процесса DataPump** – удаление «мусора» в БД, оставшегося в результате остановки работающего процесса (если это не удастся, то можно попробовать убить (kill) блокирующий процесс);
 -  **Удалить процесс DataPump** – остановка работающего процесса;
 -  **Удалить все остатки процессов DataPump** – удаление остатков всех неработающих процессов DataPump;
 -  **Обновить информацию о процессах DataPump** – обновление списка работающих процессов;
 -  **Закреть диалог.**

❖ Панель кнопок управления и дополнительные элементы управления

Панель кнопок управления, помимо кнопок навигатора и кнопок, аналогичных кнопкам управления главного меню, содержит следующие элементы управления:

-  **Перестройка индексов** – перестроение (rebuild) индексов. Операция выполняется над пользователями, выделенными в таблице рабочей области. В списке доступны два варианта:
 - ♦  **Перестройка индексов** – перестроение индексов с целью их балансировки и уменьшения занимаемого места;
 - ♦  **Изменение табл. пр-ва индексов...** – производится перенос индексов в новое табличное пространство (оно выбирается в диалоге). Если часть индексов уже располагалась в указанном табличном пространстве, то их перестроение не выполняется.

При щелчке правой клавиши мыши в области таблицы появляется контекстное меню команд действий над записями таблицы:

- 
Проверка схемы (простая) – выполнение проверки текущей схемы на соответствие заданным правилам, доступно только администратору программы. На данный момент проверяется наличие инвалидных (INVALID) и деактивированных (DISABLE) объектов. Результат проверки выводится в модальном окне;
- 
Генерация скрипта для схемы – генерация sql-скрипта (или нескольких sql-скриптов, по одному для каждого типа объектов), которые описывают создание объектов схемы.

7.3. Закладка «Импорт данных»

Закладка Импорт данных в нижней части окна программы предназначена для осуществления настроек импорта схем Oracle.

Таблица 3 – Перечень параметров настройки импорта на закладке «Импорт данных»

Название параметра	Назначение параметра	Способ ввода значения
Тип утилит	Определяет инструментарий для создания дампов. Может принимать значения: - «Локальный (imp)»; - «Серверный (DataPump)»	
Тип импорта	Определяет тип импорта. Может принимать значения: - «Схема»/«SCHEMA»; - «Таблицы»/«TABLE»; - «Комплексный». Подробное описание приводится ниже	Выбор из выпадающего списка значений
Исходный дамп-файл	Файл с дампом (при выборе файла из диалога определяется версия дампа и выводится рядом с именем файла)	Ввод с клавиатуры или с помощью дополнительного окна «Открыть», по кнопке  , расположенной справа в данной строке
Название исходной схемы	Имя схемы, которая была экспортирована (нужно для утилиты импорта), автоматически определяется по исходному файлу. В случае, если производится попытка поднятия дампа с несколькими схемами (в частности FULL дампа), то автоматическое определение имени схемы, естественно, невозможно (также возможен случай, когда имя определяется некорректно). В этих случаях имя исходной схемы нужно указать вручную	Ввод с клавиатуры
Название создаваемой схемы	Имя схемы, в которую будет выполняться импорт	Установка/снятие флага-галочки, ввод с клавиатуры (при снятом флаге-галочке), при установленном флаге-галочке можно увеличить последнюю цифру в имени схемы с помощью кнопок увеличения/уменьшения
Каталог (параметр)	«Каталог Oracle» и путь к нему с клиентского компьютера. Указывается в диалоге (кнопка с точками) в	Ввод с помощью дополнительного окна

Название параметра	Назначение параметра	Способ ввода значения
<i>отображается если Тип утилит = «Серверный (DataPump)»</i>	диалоге выводятся данные по «каталогам Oracle», а также пути к ним с клиентского компьютера (LocalPath). Пути с клиентского компьютера задаются пользователем. LocalPath – это единственное редактируемое поле, оно может быть белым – если у пользователя достаточно прав на использование «каталога Oracle», или желтым – если прав недостаточно (остальные поля серые)	«Каталоги сервера Oracle», по кнопке  , расположенной справа в данной строке
Переименовать ТП <i>(параметр отображается если Тип утилит = «Серверный (DataPump)»)</i>	В диалоге можно задать новые имена для табличных пространств. Исходные имена ТП вводятся вручную. Новые имена можно как вводить вручную, так и выбирать из выпадающего списка (ограничений по длине имени нет, главное чтобы такое ТП уже было в БД). Также можно указать вручную в формате OLDNAME1>NEWNAME1,OLDNAME2>NEWNAME2	Ввод с клавиатуры или с помощью дополнительного окна «Переименование табличных пространств», по кнопке  , расположенной справа в данной строке
Без аудита (*MF) <i>(параметр отображается если Тип утилит = «Серверный (DataPump)»)</i>	Если флаг установлен, то добавляется набор фильтров запрещающих импорт объектов Бюджетного аудита.	Установка/снятие флага-галочки
Настройки импорта		
<i>Параметры отображаются при Тип утилит = «Локальный (imp)»</i>		
Имя лог-файла (без пути)	Имя файла для вывода лога (располагается в каталоге, где размещается текущая программа)	Ввод с клавиатуры
Импортировать данные	Определяет необходимость импорта данных схемы	Установка/снятие флага-галочки
Импортировать права	Определяет необходимость импорта прав схемы	
Импортировать индексы	Определяет необходимость импорта индексов схемы	
Импортировать ограничения	Определяет необходимость импорта ограничений схемы	
Не импортировать статистику	Определяет необходимость импорта статистики	
Размер буфера (Buffer) в байтах	Размер используемого при импорте буфера	Установка/снятие флага-галочки, ввод с клавиатуры
Использовать быстрое удаление	Активирует применение механизма быстрого удаления (если нужно удалить уже существующего пользователя)	Установка/снятие флага-галочки
Компилировать объекты	Определяет необходимость компиляции invalid-объектов (простая компиляция)	Установка/снятие флага-галочки
Сбор статистики	Активирует режим сбора статистики для оптимизатора запросов, на больших схемах может работать долго (без собранной статистики могут замедляться/зависать запросы)	Установка/снятие флага-галочки
Проверка схемы	Активирует механизм поиск отключенных и нерабочих объектов (если таковые будут найдены, то выводится соответствующее сообщение)	Установка/снятие флага-галочки
<i>Параметры отображаются при Тип утилит = «Серверный (DataPump)»</i>		
INCLUDE_ROWS	Определяет необходимость обработки данных. Если флаг не установлен, то будет выполнен экспорт (или	Установка/снятие флага-галочки

Название параметра	Назначение параметра	Способ ввода значения
	импорт) только метаданных	
INCLUDE_METADATA	Определяет необходимость обработки метаданных. Если флаг не установлен, то будет выполнен импорт только данных	Установка/снятие флага-галочки
Секции storage segment	Определяет необходимость добавления параметров хранения. Если флаг не установлен, то эти параметры добавляются	Установка/снятие флага-галочки
METRICS	Вывод дополнительной информации в лог при выполнении операции. Если флаг установлен, то дополнительная информация выводится	Установка/снятие флага-галочки
Настройки создаваемой схемы		
Табличное пространство по умолчанию	Табличное пространство, в котором по умолчанию будут создаваться объекты пользователя. При импорте они создаются либо в пространстве, в котором они хранились изначально (до экспорта), если оно есть, либо в пространстве по умолчанию	Выбор из выпадающего списка значений
Временное табличное пространство	Пространство для временных объектов, рекомендуется указывать	Выбор из выпадающего списка значений
Пароль	Пароль создаваемого пользователя	Ввод с клавиатуры
Права пользователя	Перечень прав создаваемого пользователя	Ввод с клавиатуры

Программа поддерживает следующие типы импорта:

- **Схема** – тип импорта по умолчанию, производится импорт схемы. При этом используется один файл с дампом. При данном типе импорта выполняются следующие действия:
 - ◆ Если существует пользователь с таким именем, то он удаляется;
 - ◆ Создается новый пользователь;
 - ◆ Выполняется импорт;
 - ◆ Компиляция invalid-объектов;
 - ◆ Сбор статистики;
 - ◆ Проверка схемы.
- **Таблицы** – производится импорт таблиц в существующую схему (если таблиц нет, то они создаются, если есть, то просто заполняются данными). При этом используется один файл с дампом и файл со списком таблиц (он должен иметь такое же имя, как и файл с дампом, но расширение должно быть *.txt). При данном типе импорта выполняются следующие действия:
 - ◆ Отключение триггеров и внешних ключей;
 - ◆ Удаление данных из таблиц, которые будут импортированы;
 - ◆ Выполняется импорт;
 - ◆ Включение триггеров и внешних ключей;
 - ◆ Компиляция invalid-объектов;
 - ◆ Сбор статистики;
 - ◆ Проверка схемы.

- **Комплексный** – производится импорт схемы без данных, а затем производится импорт таблиц в созданную схему (если таблиц нет, то они создаются, если есть, то просто заполняются данными). При этом используются два файла с дампом и файл со списком таблиц. Имя первого файла, который содержит метаданные, задается пользователем, имя дампа-файла с таблицами получается прибавлением к исходному имени суффикса «_DATA», имя файла со списком таблиц совпадает с именем дампа-файла с таблицами, но имеет расширение *.txt. Например: BUDGET.dmp, BUDGET_DATA.dmp, BUDGET_DATA.txt. При данном типе импорта выполняются следующие действия:
 - ♦ Если существует пользователь с таким именем, то он удаляется;
 - ♦ Создается новый пользователь;
 - ♦ Выполняется импорт метаданных (из первого файла с дампом, данные игнорируются);
 - ♦ Отключение триггеров и внешних ключей;
 - ♦ Выполняется импорт таблиц (из второго файла с дампом, используется список таблиц из текстового файла, ошибки игнорируются IGNORE=Y, все остальные объекты, кроме самих таблиц и их данных тоже игнорируются);
 - ♦ Включение триггеров и внешних ключей;
 - ♦ Компиляция invalid-объектов;
 - ♦ Сбор статистики;
 - ♦ Проверка схемы.

Файл со списком таблиц передается в качестве параметра утилите imp, поэтому его формат должен соответствовать требованиям, определенным в документации по использованию данной утилиты. Например, текст файла может выглядеть так:

```
TABLES=(ACC_CLS, ORGANIZATIONS, BANKS)
```

❖ Порядок проведения импорта

1. Перейдите на закладку Импорт данных;
2. Введите параметры (большая часть параметров заполнена значениями по умолчанию);
3. Если у Вас установлено несколько клиентов Oracle (разных версий), то можно выбрать путь к нужному из выпадающего списка, который расположен рядом с кнопкой;
4. Нажмите кнопку  **Импорт**. После этого закладки будут скрыты, а на их месте покажется окно лога, куда будет выводиться информация о выполненных действиях. Внизу окна располагается кнопка  **Прервать работу**, нажатие которой прерывает выполнение операции. Рядом с ней находится строка статуса, в которой выводится текущее выполняемое действие;
5. Во время выполнения операции иконка приложения в панели задач изменится. После завершения операции иконка станет прежней, в окне статуса будет выведено ОК, активизируется кнопка  **Закреть лог**;
6. Нажатие на кнопку  **Закреть лог** закроет окно лога, а на его месте появятся закладки.



Рядом с кнопкой импорта находится выпадающий список, содержащий пути к утилите imp.exe. Если у Вас установлено две версии клиента Oracle, то при выполнении операции импорта Вам следует выбирать ту, которая соответствует версии сервера (следует отметить, что при обычном подключении версия клиента не важна!).

7.4. Закладка «Экспорт данных»

Закладка Экспорт данных предназначена для осуществления настроек экспорта схемы Oracle.

Таблица 4 – Перечень параметров настройки экспорта на закладке «Экспорт данных»

Название параметра	Назначение параметра	Способ ввода значения
Тип утилит	Определяет инструментарий для создания дампов. Может принимать значения: - «Локальный (imp)»; - «Серверный (DataPump)»	Выбор из выпадающего списка значений
Тип импорта	Определяет тип экспорта. Может принимать значения: - «Схема»/«SCHEMA»; - «Таблицы»/«TABLE»; - «Комплексный»/«FULL». Подробное описание приводится ниже	Выбор из выпадающего списка значений
Владелец экспортируемой схемы	Владелец экспортируемой схемы (при нажатии кнопки в окне параметра значение берется из выделенной строки рабочей области)	Выбор из выпадающего списка значений, ввод с клавиатуры
Пользователь (по умолчанию, владелец схемы)	Имя пользователя, под которым выполняется подключение к БД для выполнения экспорта (если не указано, то берется имя владельца схемы)	Выбор из выпадающего списка значений
Пароль	Пароль владельца экспортируемой схемы; он должен соответствовать имени пользователя, под которым производится подключение (если не указан пользователь, то подключение выполняется под именем владельца исходной схемы)	Ввод с клавиатуры
Список таблиц	Список таблиц, которые подлежат экспорту (требуется только для экспорта таблиц и комплексного экспорта). Список задается либо при помощи фильтров, либо вручную	Ввод с помощью дополнительного окна «Выбор таблиц», по кнопке  , расположенной справа в данной строке
Без аудита (*MF) (параметр отображается если Тип утилит = «Серверный (DataPump)»)	Если флаг установлен, то добавляется набор фильтров запрещающих экспорт объектов Бюджетного аудита.	Установка/снятие флага-галочки
Создаваемый дамп-файл	Файл, в который будет выполняться экспорт	Ввод с клавиатуры или с помощью дополнительного окна «Сохранить как», по кнопке  , расположенной справа в данной строке
Каталог (параметр отображается если Тип утилит = «Серверный (DataPump)»)	«Каталог Oracle» и путь к нему с клиентского компьютера. Указывается в диалоге (кнопка с точками) в диалоге выводятся данные по «каталогам Oracle», а также пути к ним с клиентского компьютера (LocalPath). Пути с клиентского компьютера задаются	Ввод с помощью дополнительного окна «Каталоги сервера Oracle», по кнопке  , расположенной справа

Название параметра	Назначение параметра	Способ ввода значения
	пользователем. LocalPath – это единственное редактируемое поле, оно может быть белым – если у пользователя достаточно прав на использование «каталога Oracle», или желтым – если прав недостаточно (остальные поля серые)	в данной строке
Настройки экспорта		
<i>Параметры отображаются при Тип утилит = «Локальный (imp)»</i>		
Имя лог-файла (без пути)	Имя файла для вывода лога (располагается в каталоге программы)	Ввод с клавиатуры
Экспортировать данные	Определяет необходимость экспорта данных схемы	Установка/снятие флага-галочки
Экспортировать права	Определяет необходимость экспорта прав схемы	
Экспортировать индексы	Определяет необходимость экспорта индекса схемы	
Экспортировать ограничения	Определяет необходимость экспорта ограничений схемы	
Экспортировать триггера	Определяет необходимость экспорта триггеров схемы	
Обеспечить целостность данных	При установленном флаге-галочке данные для экспорта берутся по состоянию на начало его выполнения (за счет этого можно получить согласованный дамп, даже при параллельной работе нескольких пользователей)	
Размер буфера (Buffer) в байтах	Размер используемого при экспорте буфера	Установка/снятие флага-галочки, ввод с клавиатуры
<i>Параметры отображаются при Тип утилит = «Серверный (DataPump)»</i>		
INCLUDE_ROWS	Определяет необходимость обработки данных. Если флаг не установлен, то будет выполнен экспорт (или импорт) только метаданных	Установка/снятие флага-галочки
INCLUDE_METADATA	Определяет необходимость обработки метаданных. Если флаг не установлен, то будет выполнен экспорт только данных	Установка/снятие флага-галочки
Секции storage segment	Определяет необходимость добавления параметров хранения. Если флаг не установлен, то эти параметры добавляются	Установка/снятие флага-галочки
METRICS	Вывод дополнительной информации в лог при выполнении операции. Если флаг установлен, то дополнительная информация выводится	Установка/снятие флага-галочки

Программа поддерживает следующие типы экспорта:

- **Схема** – тип экспорта по умолчанию, производится экспорт схемы. При этом создается один файл с дампом.
- **Таблицы** – производится экспорт таблиц и их данных (а также, если заданы соответствующие настройки: триггеров, индексов, ограничений, прав). При этом получается один файл с дампом и файл со списком таблиц (он должен иметь такое же имя, как и файл с дампом, но расширение должно быть *.txt).

- **Комплексный** – производится экспорт схемы без данных, а затем производится экспорт таблиц и их данных. При этом получаются два файла с дампом и файл со списком таблиц. Имя первого файла, который содержит метаданные, задается пользователем, имя дампа-файла с таблицами получается прибавлением к исходному имени суффикса «DATA», имя файла со списком таблиц совпадает с именем дампа-файла с таблицами, но имеет расширение *.txt. Например: BUDGET.dmp, BUDGET_DATA.dmp, BUDGET_DATA.txt.

❖ Порядок проведения экспорта

1. Перейдите на закладку Экспорт данных.
2. Введите параметры (большая часть параметров заполнена значениями по умолчанию).
3. Если у Вас установлено несколько клиентов Oracle (разных версий), то можно выбрать путь к нужному клиенту из выпадающего списка, расположенного рядом с кнопкой.
4. Нажмите кнопку  **Экспорт**. После этого закладки будут скрыты, а на их месте покажется окно лога, куда будет выводиться информация о выполненных действиях. Внизу окна располагается кнопка  **Прервать работу**, нажатие которой прерывает выполнение операции. Рядом с ней находится строка статуса, в которой выводится текущее выполняемое действие.
5. Во время выполнения операции иконка приложения в панели задач изменится. После завершения операции иконка станет прежней, в окне статуса будет выведено ОК, активизируется кнопка  **Закреть лог**.
6. Нажатие на кнопку  **Закреть лог** закрывает окно лога, а на его месте появятся закладки.



Рядом с кнопкой экспорта находится выпадающий список, содержащий пути к утилите exp.exe. Если у Вас установлено две версии клиента Oracle, то при выполнении операции экспорта Вам следует выбирать ту, которая соответствует версии сервера (следует отметить, что при обычном подключении версия клиента не важна!).

7.5. Закладка «Работа с пользователями»

Закладка Работа с пользователями предназначена для создания новых пользователей (схем), создания пользователей списком, а также изменения паролей существующих пользователей. Просмотреть, произвести фильтрацию и удалить существующих пользователей можно в верхней таблице рабочей области программы (см. п. 7.2 [«Структура окна программы и принципы работы»](#)).

1. Для создания нового пользователя укажите:
 - ♦ Имя пользователя;
 - ♦ Пароль, назначаемый пользователю;
 - ♦ Подтверждение пароля;
 - ♦ Таб. прост-во по умолчанию – табличное пространство, в котором будут создаваться объекты пользователя;
 - ♦ Временное таб. прост-во – табличное пространство, в котором будут храниться временные данные пользователя;
 - ♦ Права пользователя – права, назначаемые пользователю.

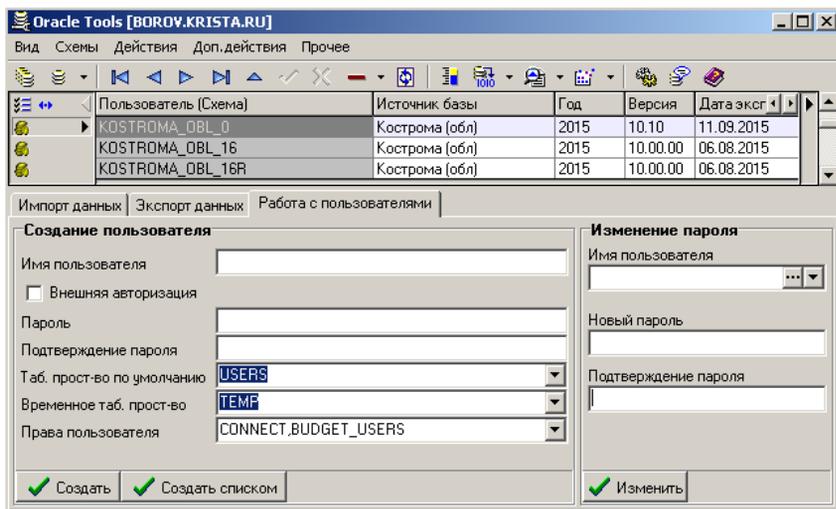
Затем нажмите кнопку  **Создать**.

2. Для одновременного создания сразу нескольких пользователей можно воспользоваться кнопкой  **Создать списком** и в появившемся диалоговом окне задать список пользователей с паролями. Список должен выглядеть следующим образом:

 *Пример*

Ivanov 123
Petrov ah5
Sidorov dsfsd

Рисунок 39 – Вид окна программы OraTools.exe на закладке «Работа с пользователями»



3. Для смены пароля выделенного курсором пользователя в группе параметров Изменение пароля (в правой части закладки) укажите имя этого пользователя, затем новый пароль и подтверждение пароля, затем нажмите кнопку  **Изменить**.



При необходимости создания пользователя с именем, в котором содержатся нестандартные символы, имя пользователя должно быть заключено в двойные кавычки (имя при этом становится чувствительным к регистру). То же самое относится к смене пароля.

7.6. Инструментарий для создания дампов «DataPump»

Начиная с 10-ой версии Oracle появился новый инструмент для создания дампов (в дальнейшем именуемый DataPump, т.к. он основан на использовании пакета DBMS_DATAPUMP).



Из-за большого числа ошибок DataPump в ранних версиях (10.2, 11.1) мог сильно тормозить и нестабильно работать. Поэтому его использование можно рекомендовать, только начиная с версии Oracle 11.2.

Изменился не только инструмент, но и формат дампов – новые инструменты понимают только свой формат дампов (т.е. поднять старый дамп новыми инструментами не получится), старые тоже не понимают нового формата дампов.

DataPump имеет следующие особенности:

- DataPump представляет собой набор пакетов в БД, т.е. с ним можно работать через sql-запросы, без использования клиентских утилит.
- DataPump выполняется как процесс на сервере (ранее импорт и экспорт были клиентскими процессами). Поэтому DataPump сохраняет дамп-файл в каталог, который виден из СУБД. «Каталог Oracle» – это алиас для некоторого каталога операционной системы. На «каталоги Oracle» выдаются права пользователям Oracle. Т.е. для записи данных из Oracle в каталог операционной системы он должен быть зарегистрирован как «каталог Oracle» и пользователю Oracle (под которым выполняется операция) должны быть выданы права (внутри СУБД) на запись в этот «каталог Oracle». Кроме того, разумеется, сам процесс Oracle (рассматриваемый как объект операционной системы) должен иметь права на запись в соответствующий каталог операционной системы.
- В связи с тем, что DataPump выполняется как процесс на сервере, то клиент может отсоединиться, при этом процесс экспорта или импорта дампа будет продолжаться на сервере. Фактически клиент при использовании DataPump отдает серверу команду и далее может наблюдать за ее выполнением, а может и не наблюдать.

Для работы с DataPump из OraTools предварительно необходимо выполнить следующие действия:

1. Зарегистрировать «каталог Oracle» и назначить соответствующие права. Как правило, такой каталог автоматически создается с именем DATA_PUMP_DIR, при создании БД.
2. Поскольку каталог операционной системы, в который помещаются дампы при экспорте (или из которого берутся дампы при импорте), как правило, располагается на сервере, то при необходимости получения дампа на другом компьютере (клиентском), необходимо чтобы каталог операционной системы, который соответствует «каталогу Oracle», был доступен с клиентской машины. Для этого администратор должен его расшарить и назначить пользователям соответствующие права.

Пример 1

Пусть на сервере с именем OraServer имеется «каталог Oracle» DATA_PUMP_DIR, которому соответствует каталог операционной системы «C:\app\admin\bdtg\dpdump\». Допустим, мы хотим получить дамп на клиентском компьютере с именем OraUser. Тогда администратор может расшарить папку «C:\app\admin\bdtg\dpdump\» под именем dpdump и дать на нее права пользователям сети. В этом случае с клиентского компьютера (и из OraTools) данный каталог будет виден под именем «\\OraServer\dpdump».

Таким образом один и тот же физический каталог будет иметь 3 имени: имя для сервера Oracle (DATA_PUMP_DIR), имя в операционной системе компьютера OraServer (C:\app\admin\bdtg\dpdump\), имя в операционной системе компьютера OraUser (\\OraServer\dpdump). При использовании OraTools требуется указать 2 из них: имя для сервера Oracle (DATA_PUMP_DIR) – будет использовано для выполнения sql-запросов, и имя в операционной системе компьютера OraUser (\\OraServer\dpdump) – будет использовано для файловых операций.

Пример 2

Пусть на сервере с именем OraServer имеется «каталог Oracle» DATA_PUMP_DIR, которому соответствует каталог операционной системы «C:\app\admin\bdtg\dpdump\». Допустим, мы хотим получить дамп на той же самой машине. Тогда администратору можно ничего не расшаривать. В этом случае с клиентского компьютера (и из OraTools) данный каталог будет виден под именем «C:\app\admin\bdtg\dpdump\» (т.к. клиентский компьютер и сервер – это один и тот же компьютер).

При использовании OraTools потребуется указать: имя для сервера Oracle (DATA_PUMP_DIR) – будет использовано для выполнения sql-запросов, и имя в операционной системе компьютера клиента (C:\app\admin\bdtg\dpdump\)) – будет использовано для файловых операций.

В СУБД Oracle, используя пакет UTL_FILE, можно читать, писать и удалять файлы из файловой системы сервера. Но в СУБД Oracle нельзя получить список файлов и каталогов файловой системы сервера. Поэтому теоретически возможна работа с файлами, без создания на сервере дополнительных расшаренных каталогов. Однако при таком подходе на сервере, скорее всего, будет происходить накопление «забытых» (клиент отключился от процесса и забыл забрать результат или стереть исходный дамп) и «потерянных» (разрыв соединения, выключение клиентского компьютера) файлов (дампы и логи), которые клиенты не смогут удалить, т.к. не «увидят» (нет возможности посмотреть список файлов). К тому же с точки зрения безопасности права на использование пакета UTL_FILE (а также пакетов UTL_TCP, UTL_HTTP, UTL_SMTP и других) желательно ограничивать. Поэтому OraTools не использует данную возможность.

7.7. Настройка программы OraTools.exe

Вызов модального окна настройки программы осуществляется выбором подпункта **Настройки** пункта главного меню **Прочее**. Настройка сводится к определению параметров сохранения настроек в реестре и выбору используемого типа кодировки. Кодировки (языковые настройки) влияют на отображение текста выводимых сообщений, а также на перекодировку вводимых данных при сохранении на сервере. Рекомендуется использовать одинаковые кодировки на клиенте и сервере – тогда не потребуются дополнительной перекодировки.

Таблица 5 – Перечень параметров настройки программы OraTools

Название параметра	Назначение параметра	Способ ввода значения
Сохранять настройки (в реестр)	Группа параметров настроек сохранения, определяющая необходимость сохранения в системном реестре настроек программы, а также задания видов паролей, которые следует сохранять	Установка/снятие флага-галочки
Сохранять пароль импорта		
Сохранять пароль экспорта		
NLS_LANG при импорте/экспорт	<p>Влияет на кодировку выдаваемых сообщений, а также на перекодировку вводимых данных при сохранении на сервере. Рекомендуется использовать одинаковые кодировки на клиенте и сервере - тогда перекодировки не будет. Если не указано, то используется настройка клиента Oracle, выполняющего операцию импорта/экспорта.</p> <p>Может принимать следующие значения:</p> <ul style="list-style-type: none"> - RUSSIAN_CIS.CL8MSWIN1251 – русский язык в кодировке win1251; - AMERICAN_CIS.CL8MSWIN1251 – английский язык в кодировке win1251; - NA – значение не определено (если указать данное значение, то переменная окружения будет очищена) 	Выбор из выпадающего списка значений

8. Перечень возможных ошибок, возникающих при работе с Oracle

- [При запуске возникает ошибка ORA-00604: ошибка на рекурсивном SQL-уровне 1.](#)

- [При запуске возникает ошибка "Точка входа в процедуру longjmp не найдена в библиотеке DLL orautils.dll".](#)
- [Возникает ошибка ORA-00031. Не убиваемая сессия.](#)
- [Возникает ошибка ORA-12571 во время создания базы данных.](#)
- [Возникает ошибка TNS-12541 TNS:no listener.](#)
- [ORA-12154 и другие проблемы клиента.](#)
- [Возникает ошибка ORA-00600, что делать?](#)
- [Возникает ошибка ORA-03113, или ORA-03114, что делать?](#)
- [При экспорте возникла ошибка EXP-00003, что делать?](#)
- [Возникает ошибка ORA-01036: illegal variable name/number, что делать?](#)
- [При импорте возникает ошибка IMP-00017 + ORA-00959, что делать?](#)
- [Возникает ошибка ORA-01653, ORA-01658, ORA-01659, ORA-01691, что делать?](#)
- [У пользователей возникла ошибка с DBMS_LOCK, что делать?](#)
- [Возникла ошибка ORA-00060: deadlock detected while waiting for resource, что делать?](#)
- [Возникла ошибка ORA-00054: указан занятый ресурс и его получение с опцией NOWAIT, что делать?](#)
- [Запрос висит очень долго...](#)
- [При работе с Oracle возникла ошибка не описанная в данном FAQ, что делать?](#)
- [Какие лог-файлы, следует смотреть при возникновении ошибок?](#)
- [При экспорте схемы OraTools выдал предупреждение: "Вы пытаетесь изменить схему системного пользователя. Вы УВЕРЕНЫ в том, что хотите СДЕЛАТЬ именно ЭТО?"](#)
- [ORA-01445: нельзя выбрать ROWID из связ.представления без таблицы, сохраняющей ключ.](#)
- [ORA-28040: No matching authentication protocol.](#)

❖ **Вопрос: При запуске возникает ошибка ORA-00604: ошибка на рекурсивном SQL-уровне 1**

Ответ: Такая ошибка может возникать при подключении 11-ым клиентом к серверу 9-ой версии. Способ решения - для подключения к серверу 9-ой версии нужно использовать клиента 9-ой или 10-ой версии. Если у Вас на компьютере одновременно установлено несколько клиентов, то для подключения используется тот что установлен, как используемый по умолчанию. Изменить используемый по умолчанию клиент можно с помощью

- Oracle Home Selector для 9-ой версии Oracle (находится в папке Oracle Installation Products);
- Universal Installer для 10-ой, 11-ой версии Oracle (находится в папке Oracle Installation Products). Далее необходимо нажать кнопку Installed Products и в открывшемся окне перейти на вкладку Environment;
- OraTools (нужно вызвать диалог «Информация к размышлению» и там выбрать из выпадающего списка Первичный ORA_HOME). В Vista и Win7 Вам нужно запускать программу с правами администратора – иначе Вам не дадут доступ к реестру.

❖ **Вопрос: При запуске возникает ошибка "Точка входа в процедуру longjmp не найдена в библиотеке DLL orautils.dll"**

Ответ: Такая ошибка может возникать при попытке подключиться клиентом версии 10.2.0.1 к серверу версии 11.2. Для ее устранения нужно установить патч версии 10.2.0.3 или 10.2.0.4 на клиента 10.2.0.1.

❖ **Вопрос: Возникает ошибка ORA-00031. Не убиваемая сессия?**

Ответ: Вообще сессия может висеть некоторое время в состоянии marked for kill до тех пор, пока Oracle не закончит обязательные процедуры, связанные с ее остановкой, о чем сказано в документации. Поэтому сначала необходимо подождать. Если время ожидания закрытия сессии превышает разумные пределы, то перезагрузите сервер, когда это будет возможно. Существует также консольная утилита **OraKill**, но ее использовать следует только в случае крайней необходимости.

❖ **Вопрос: Возникает ошибка ORA-12571 во время создания базы данных**

Ответ: Пришлось пройтись по скриптам на создание базы данных (..\admin\SQLNET.AUTHENTICATION_SERVICES = (NONE) После чего инсталляция успешно прошла. В общем же случае нужно проверить корректность **SQLNET.ORA, TNSNAMES.ORA, LISTNER.ORA**

❖ **Вопрос: Возникает ошибка TNS-12541 TNS:no listener?**

Ответ: Ошибка обычно возникает при попытке подключения, тестировании соединения, иногда непосредственно сразу после установки Oracle. Означает она, что по каким-либо причинам отсутствует процесс-слушатель, который должен обслуживать соединения клиентов с БД. В этом случае полезно проверить наличие службы с именем ...**Listener** и ее состояние, попробовать запустить, изменить тип запуска на «Автоматический». Если службы нет (что бывает даже сразу после установки), то воспользуйтесь консольной утилитой **lsnrctl** (входит в поставку Oracle), набрав в командной строке **lsnrctl start**. После этой команды появится ошибка, которую более внятно можно посмотреть в **ora92\network\log\listener.log**. Кроме того, после исполнения команды служба может появиться в списке с типом запуска «manual». Это может наблюдаться при старте нескольких слушателей одновременно для разных инстансов. Например, если установлен одновременно Oracle9 и Oracle10.

❖ **Вопрос: ORA-12154 и другие проблемы клиента**

Ответ: Возникает, поскольку OCI не может разрешить указанный идентификатор соединения согласно network\ADMIN\tnsnames.ora. Проверьте содержимое этого файла или воспользуйтесь Net Manager или другой утилитой для его настройки. Для идентификации tns предпочтительно использовать имена в доменной нотации, например, borov.krista.ru (а не borov).

Часто проблемы с подключением не оканчиваются правкой tnsnames.ora. Причиной этому может быть несколько установленных клиентов, неверная разрядность или поврежденный клиент. В случае использования стандартного дистрибутива следует выбирать установку либо Runtime, либо Administrator. Прежде всего необходимо разобраться какие клиенты Oracle установлены и какой из них активен.

Первое смотрится в реестре, примерно, HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\ORACLE

Второе – в переменной окружения path, где путь к библиотекам активного клиента должен быть раньше других.

Активный клиент должен быть в адекватном состоянии (визуально, есть bin\oci.dll и много других библиотек, network\ADMIN\tnsnames.ora и т.д. – можно сравнить с достоверно работающим эталоном), причем для работы 32-бит приложений нужен именно 32 бит клиент (особенно, если копировали поверх).

Иногда имеет смысл раздать все права на все каталоги клиента.

Избегать кириллицы, пробелов, () во всех путях.

❖ **Вопрос: Возникает ошибка ORA-00600, что делать?**

Ответ: Это внутренняя ошибка Oracle, напоминающая синий экран в Windows. Проблема может быть временной, т.е. через некоторое время самоустранится. Можно попробовать следующие действия с БД: перезапустить сервер, выполнить экспорт, а затем путем импорта создать новую схему с теми же данными, собрать статистику, перестроить индексы и т.п. Если ничего не помогло, тогда обратитесь в техподдержку Oracle, а также сообщите разработчикам программы.

❖ **Вопрос: Возникает ошибка ORA-03113, или ORA-03114, что делать?**

Ответ: Если данная ошибка возникает во время работы регулярно на одном и том же запросе, то это падение oracle-сессии (по причине внутренних ошибок Oracle). Такая ошибка зависит от БД (т.е. в другой БД или даже в другой схеме, тот же запрос может работать безошибочно). Поэтому рекомендации такие же как и в случае ORA-00600.

Если ошибка возникает при создании БД, то, возможно, у Вас не хватает каких-либо прав. В частности Oracle очень чувствителен к вхождению компьютера в домен (например, под доменным пользователем с правами администратора такая ошибка может появиться, а под локальным администратором исчезнет).

❖ **Вопрос: При экспорте возникла ошибка EXP-00003, что делать?**

Ответ: Если вы ”поднимаете” дампы, сделанный с сервера версии 9.2.0.5 или выше, на сервер версии 9.2.0.4 или ниже, то это закономерно. Для того чтобы обойти данную ошибку, требуются определенные усилия, поэтому мы настоятельно не рекомендуем клиентам ставить на версию 9.2 патчи 9.2.0.5 и выше.

❖ **Вопрос: Возникает ошибка ORA-01036: illegal variable name/number, что делать?**

Ответ: Как правило, данная ошибка возникает при ”прямом” доступе к БД (т.е. при доступе без использования BDE). Если возникла данная ошибка, то скорее всего Вы использовали в качестве имени переменной зарезервированное слово. При обращении через BDE имена переменных заменяется на безопасные и ошибка не возникает. Для исправления данной ошибки следует изменить имя переменной.

❖ **Вопрос: При импорте возникает ошибка IMP-00017 + ORA-00959, что делать?**

Например, текст ошибки может выглядеть так:

IMP-00017: при выполнении следующей команды ORACLE возвращает ошибку 959:

```
"CREATE TABLE "UTEMPLATES" ("ID" NUMBER(10, 0) NOT NULL ENABLE, "TYPEDFORM" "  
"NUMBER(10, 0) NOT NULL ENABLE, "DETAILINDEX" NUMBER(10, 0), "NUMSTR" NUMBER"  
"(10, 0), "NOTESTR" VARCHAR2(255), "NOTEBLOB" BLOB) PCTFREE 10 PCTUSED 40 I"  
"NITRANS 1 MAXTRANS 255 STORAGE(INITIAL 65536 FREELISTS 1 FREELIST GROUPS 1)"
```

```
" TABLESPACE "BUD2006" LOGGING NOCOMPRESS LOB ("NOTEBLOB") STORE AS (TABLES"  
"PACE "BUD2006" ENABLE STORAGE IN ROW CHUNK 8192 PCTVERSION 10 NOCACHE STOR"  
"AGE(INITIAL 65536 FREELISTS 1 FREELIST GROUPS 1))"
```

IMP-00003: ORACLE выдал ошибку 959 «ORA-00959: раздел 'BUD2006' не существует».

Ответ: В базе данных, куда выполняется импорт, отсутствует табличное пространство с указанным именем (в примере это «BUD2006»). Обычно Oracle может сам решить такую проблему, создав объект в табличном пространстве по умолчанию. Однако это работает в ситуации с блоб полями (и еще некоторыми не используемыми в АС «Бюджет» типами объектов). Существует два варианта: либо создать необходимое табличное пространство, либо изменить дамп-файл. Изменение дамп-файла сводится к замене имен табличных пространств путем редактирования дамп-файла: вручную или используя **OraTools**. Если использовать **OraTools**, то исходное имя не должно быть короче того, на которое заменяем (например, «BUD2006» можно заменить на «USERS», а наоборот уже нельзя). В **OraTools** смотрите главное меню: Доп.действия -> Смена табличного пространства....

❖ **Вопрос: Возникает ошибка ORA-01653, ORA-01658, ORA-01659, ORA-01691, что делать?**

Ответ: Это значит, что в табличном пространстве, куда Вы пытаетесь "поднять" данные, закончилось место. Чаще всего это бывает табличное пространство «USERS», однако проблемы с местом могут быть в любом табличном пространстве. В данной ситуации нужно либо увеличить место, выделенное под табличное пространство, либо удалить ненужные данные.

❖ **Вопрос: У пользователей возникла ошибка с DBMS_LOCK, что делать?**

Ответ: Чаще эти ошибки возникают при обращении к пакету DBMS_LOCK в случае, когда у пользователя нет соответствующих прав. Выдача этих прав выполняется в скрипте **CreatePrimaryObjects.sql**, который нужно выполнить до создания схем АС «Бюджет». Выдавать права на этот пакет следует командой (выполнять надо под пользователем SYS):

```
GRANT EXECUTE ON SYS.DBMS_LOCK TO BUDGET_USERS
```

/

В случае если у Вас используется старая версия скрипта, рекомендуем запросить новую версию **CreatePrimaryObjects.sql**, чтобы при последующих созданиях БД ошибка не появилась снова.

❖ **Вопрос: Возникла ошибка ORA-00060: deadlock detected while waiting for resource, что делать?**

Ответ: Проблема во взаимной блокировке двух сессий. Рассмотрим, например, такую последовательность действий (это всего лишь пример, возможны и другие сценарии, приводящие к ORA-00060).

- Первая сессия вошла в транзакцию (это важно, если блокировки будут в разных транзакциях, то ничего подобного не произойдет) и изменила запись в таблице заголовка (в результате до конца транзакции данная запись будет заблокирована от изменения, но читать ее по-прежнему можно).
- Вторая сессия вошла в транзакцию и заблокировала запись в таблице детализации (для того же самого документа, что и первая сессия).
- Первая сессия пытается изменить запись в таблице детализации, но поскольку запись заблокирована второй сессией, то она ждет освобождения ресурса (смотрите вопрос «Запрос висит очень долго...»).

- Вторая сессия пытается изменить запись в таблице заголовка – но поскольку запись заблокирована первой сессией, то она тоже будет ждать освобождения ресурса.
- Очевидно, что в данной ситуации сессии ждут друг друга, т.е. их ожидание может длиться бесконечно – Oracle, видя такую ситуацию, выдает ошибку ORA-00060 в одной из сессий. В данном случае, вероятнее всего, в первой (хотя это и не важно). В результате мы получаем первую сессию с ошибкой и незавершенной транзакцией, и вторую сессию, которая ждет завершения первой. Соответственно, если первая сессия завершит транзакцию (например, rollback), то вторая успешно закончится (а до тех пор она будет висеть и ждать первую).

Блокировка записи возникает при изменении данных (insert, update, delete) или при их явной блокировке (select ... for update).

При получении такой ошибки нужно попросить администратора посмотреть **alert_<SID>.log**, где должна появиться запись, аналогичная приведенной ниже:

```
Tue Dec 22 11:28:57 2009
ORA-00060: Deadlock detected. More info in file
c:\oracle\admin\bdgt\udump\bdgt_ora_3372.trc.
```

В этой строке указывается, что создан файл c:\oracle\admin\bdgt\udump\bdgt_ora_3372.trc, содержащий подробное описание ситуации, в которой возник deadlock (в частности, запросы в сессиях). Эти данные (а также описание действий, которые выполнялись пользователями программы и привели к такой ситуации) нужно переслать разработчикам программы, чтобы они устранили причину возникновения данной ошибки.

❖ **Вопрос: Возникла ошибка ORA-00054: указан занятый ресурс и его получение с опцией NOWAIT, что делать?**

Ответ: Как правило, такая проблема возникает при попытке изменить состояние документа, если документ заблокирован другим пользователем. В данной ситуации следует дождаться, когда другой пользователь закончит редактирование и разблокирует документ. Следует помнить, что перед тем как Вы начнете его изменять, Вам следует обновить данные документа, чтобы увидеть изменения, внесенные другим пользователем.

Если документ блокируется длительное время, то можно попробовать узнать, кто его блокирует. Для этого нужно посмотреть блокировки на объекты БД. Сделать это можно с помощью **OraTools** (пункт меню Блокировки -> закладка Заблокированные объекты) или sql-запросом к БД. Например, запрос может быть таким:

```
select obj.owner, obj.object_name, obj.object_type,
       lo.SESSION_ID, lo.oracle_username, lo.os_user_name, lo.process,
       decode(lo.xidusn + lo.xidslot + lo.xidsqn, 0, 'WAIT FOR LOCK', 'LOCK') state
from v$locked_object lo, ALL_OBJECTS obj
where obj.object_id = lo.OBJECT_ID
```

Следует отметить, что мы не увидим, какие именно строки таблицы заблокированы пользователем – ведь пользователь, как правило, блокирует не всю таблицу! Поэтому в ситуации, когда нужную нам таблицу (или таблицы) блокируют несколько пользователей, сложно определить, кто же реально мешает выполнить команду. В этом случае для точного установления источника блокировки попробуйте выполнить запрос вида: select * from <table> where ID = <value> for update. Данный запрос повиснет, т.к. он запущен без опции NOWAIT, и мы сможем определить, кого же именно он ждет. О том, как это сделать, смотрите следующий вопрос «Возникла ошибка ORA-00060: deadlock detected while waiting for resource, что делать?»

❖ **Вопрос: Запрос висит очень долго...**

Ответ: Возможно, проблема в блокировках, т.е. выполнение одной операции блокируется другой операцией. Если установлен **Enterprise Edition**, то посмотреть блокировки можно следующим образом: запустить **Oracle Enterprise Manager Console** (слева в окне будет дерево). Затем в дереве выбираем нужную БД и подключаемся к ней, открываем закладку <имя БД>\Instance\Locks. Внизу будет переключатель:

- All Locks – показывать все блокировки (будет показано много блокировок, в том числе служебных, которые нас мало интересуют);
- BLOCKING/WAITING locks – показывать блокировки, которые ждут других блокировок (это уже явная проблема);
- User Lock Types – показывать блокировки пользователя (если блокировка висит долго, то это может приводить к проблемам).

Для обновления данных на странице используется кнопка **Refresh**, расположенная слева внизу.

Также список блокирующих друг друга сессий можно посмотреть в **OraTools** (пункт меню Блокировки -> закладка Заблокированные сессий).

«Убить» сессию, которая нам мешает можно через **Oracle Enterprise Manager Console**. Открываем закладку <имя БД>\Instance\Sessions, в списке сессий выбираем нужную, в выпадающем (по правой кнопке мыши) меню выбираем пункт Kill Session, там будут опции «Immediate» (сразу), «Post Transaction» (после завершения транзакции). Естественно, в данном случае следует выбрать «Immediate».

❖ **Вопрос: При работе с Oracle возникла ошибка, не описанная в данном FAQ, что делать?**

Ответ: Следует сообщить в техотдел. При этом нужно указать следующую информацию:

1. Обязательно нужен **номер** ошибки (например, ORA-02001 или TNS-12561), также желательно добавить и ее текст.
2. Версию Oracle (полностью, а не просто 10 или 11), также неплохо указать регион – если ошибка пришла от клиентов.
3. Описание ситуации, при которой ошибка возникла (скриншоты приветствуются, но надо помнить, что они ничего не говорят о том, как Вы получили ошибку)
4. Если у вас есть sql-запрос, выполнение которого вызывает ошибку, его также следует прислать.

Если ошибка возникла при работе в АС «Бюджет», то также следует указать:

1. Версию программы (в том числе версию схемы БД и версии программных модулей).
2. Название и GUID интерфейса, на котором ошибка возникла.

Также приветствуются лог-файлы (их лучше пересылать в виде приложенных к письму файлов). В общем случае нужно стремиться к тому, чтобы информации хватило для воспроизведения ошибки другим человеком, который работает на другом компьютере и с другой БД. Следует помнить, что чем точнее будет описана ошибка, тем быстрее можно будет найти причину ее возникновения и устранить.

❖ **Вопрос: Какие лог-файлы следует смотреть при возникновении ошибок?**

Ответ: На клиентской машине ошибки записываются в файл **sqlnet.log**. Данный файл, как правило, размещен в каталоге с программой, которая вызвала ошибку.

На сервере есть каталог \$ORACLE_HOME\network\log, там содержится файл **listener.log** – лог листнера, а также логи других служб. В каталоге admin\ORACLE_SID\bdump размещен файл **alert_<SID>.log** (основной лог работы сервера), остальные файлы в данном каталоге и других подкаталогах admin\ORACLE_SID – это трассировочные файлы, содержащие дополнительную информацию о работе различных служб Oracle.

❖ **Вопрос: При экспорте схемы OraTools выдал предупреждение: "Вы пытаетесь изменить схему системного пользователя. Вы УВЕРЕНЫ в том, что хотите СДЕЛАТЬ именно ЭТО?"**

Ответ: Это предупреждение означает, что Вы пытаетесь изменить системные данные Oracle. Случайное изменение этих данных может привести к непредсказуемым последствиям – вплоть до падения сервера. Изменение схемы системного пользователя можно сравнить с ручной правкой системных файлов в ОС Windows. Для выполнения такого рода изменений Вы должны хорошо разбираться в Oracle и четко представлять для чего Вам эти изменения нужны – во всех остальных случаях такие изменения делать недопустимо.

Причиной появления данного предупреждения является то, что в параметре "название исходной схемы" указано имя системной схемы (например: OUTLN, SYS, SYSMAN, SYSTEM). Если это имя подставляется автоматически при выборе дампа, то чаще всего это означает что был выполнен FULL экспорт (экспорт всей БД).

❖ **Вопрос: ORA-01445: нельзя выбрать ROWID из связ. представления без таблицы, сохраняющей ключ**

Ответ: Возможно Вы именно это и пытаетесь сделать – то есть, выбрать ROWID от туда, где его не может быть.

Если запрос содержит большое количество join, то возможно это ошибка Oracle 9 (и ранних релизов 10й версии). Исправлено в 9.2.0.8.

❖ **Вопрос: ORA-28040: No matching authentication protocol**

Ответ: Клиент 9й версии с настройками по умолчанию получает такую ошибку при подключении к серверу 12й версии. При этом клиент 10й версии уже подключается нормально. Можно поэкспериментировать с sqlnet.ora параметр SQLNET.ALLOWED_LOGON_VERSION, но все же рекомендуется сменить клиента на более подходящий.