

**Специальность 10.05.01 «Компьютерная безопасность»,
Специализация «Математические методы защиты информации»
Уровень высшего образования – специалитет**

Дисциплина: Основы построения защищенных баз данных.

Лабораторная работа №11.

Администрирование Oracle. Перемещение данных (экспорт и импорт).

1. Учебные цели:

- Отработать вопросы реализации прозрачного шифрования данных Oracle.
- Освоить приемы прозрачного шифрования данных таблиц и отдельных столбцов, шифрования при создании резервных копий.

2. Требования к результатам обучения основной образовательной программы, достигаемые при проведении лабораторной работы:

- Уметь использовать возможности современных систем для решения задач администрирования и защиты баз данных.
- Владеть средствами приложений СУБД Oracle для управления настройками прозрачного шифрования данных таблиц и отдельных столбцов, шифрования при создании резервных копий.

3. Перечень материально-технического обеспечения

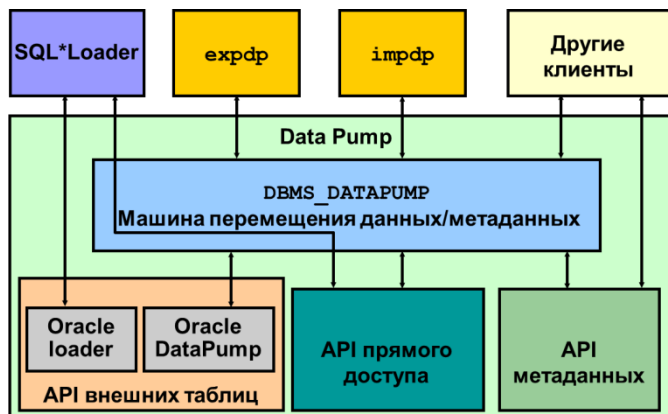
ПЭВМ с проигрывателем виртуальных машин, виртуальная машина с установленной СУБД Oracle.

4. Краткие теоритические сведения и задания на исследование. Задания выделены рамками и синим шрифтом. Результаты лабораторной работы представляются в виде файла, содержащего копии экрана, показывающие этапы выполнения заданий.

Вопросы

- Доступные пути перемещения данных
- Создание и использование объектов directory
- Использование утилиты SQL*Loader для загрузки информации из базы данных, отличной от Oracle (или пользовательских файлов)
- Общая архитектура утилиты Data Pump
- Использование возможностей экспорта и импорта утилиты Data Pump для перемещения информации между базами данных Oracle
- Использование внешних таблиц для перемещения данных с помощью платформо-независимых файлов

Перемещение данных: общая архитектура



На рисунке представлена структурная схема, отражающая основные конструктивные компоненты.

- **DBMS_DATAPUMP**; пакет реализует API (application programming interface, программный интерфейс приложения) для высокоскоростных утилит экспорта и импорта, выполняющих массовое (bulk) перемещение данных и метаданных.
- **API прямого доступа (Direct Path API, DPAPI)**; база данных Oracle поддерживает API прямого доступа, который минимизирует преобразования данных, а также синтаксический разбор во время выгрузки и загрузки.
- **DBMS_METADATA**; рабочие процессы используют данный пакет для выгрузки и загрузки всех метаданных. Большее предпочтение отдается сохранению определений объектов БД на XML по сравнению с их описанием на SQL.
- **API внешних таблиц (External Table API)**; данные можно хранить во внешних таблицах (то есть в платформу-независимых файлах), а для доступа к ним использовать драйверы ORACLE_DATAPUMP и ORACLE_LOADER. Команда SELECT читает внешние таблицы так, как будто они хранятся в БД Oracle.
- **SQL*Loader**; интеграция клиента SQL*Loader с внешними таблицами позволила осуществить автоматическую миграцию управляющих файлов загрузчика в параметры доступа к внешним таблицам.
- **expdp и impdp**; клиенты expdp и impdp – тонкие уровни вызова пакета DBMS_DATAPUMP для запуска и мониторинга операций Data Pump.
- **Другие клиенты** – приложения (например, Database Control, репликация, переносимые табличные пространства, пользовательские приложения), получающие выигрыш от приведенной инфраструктуры. В качестве клиента DBMS_DATAPUMP можно также использовать SQL*Plus и выполнять простые запросы состояния выполняющихся операций.

Общее описание объекта Directory

Объект Directory – логическая структура, представляющая в базе данных физический каталог файловой системы сервера. Такие объекты содержат месторасположение конкретного каталога операционной системы. Объект Directory может быть использован в Enterprise Manager. Он устраняет необходимость задания спецификации пути доступа к каталогу в операционной системе. В результате его применения предоставляется большая гибкость управления файлами. Объекты Directory принадлежат пользователю SYS. Имена директорий (объектов Directory) должны быть уникальными в рамках всей базы данных, поскольку все они располагаются в одном пространстве имен (принадлежат одному пользователю SYS).

Объекты Directory требуются для задания месторасположения файлов при использовании Data Pump, так эта утилита обращается к файлам, находящимся на сервере, а не на клиенте.

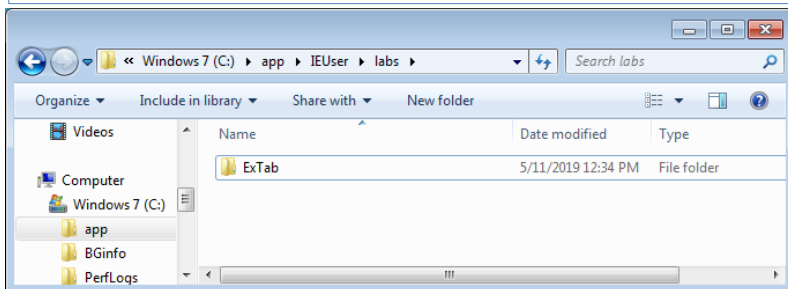
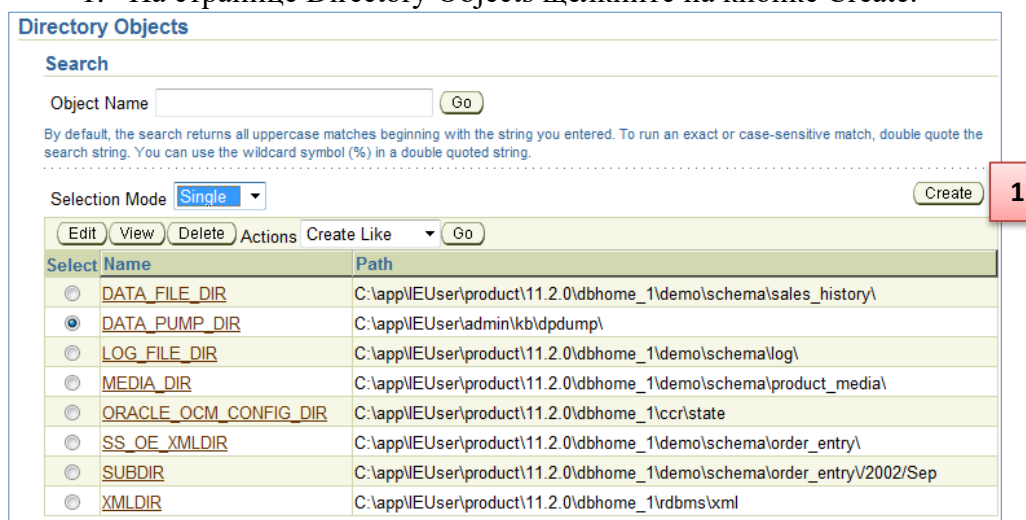
В Enterprise Manager выберите Schema > Directory Objects.



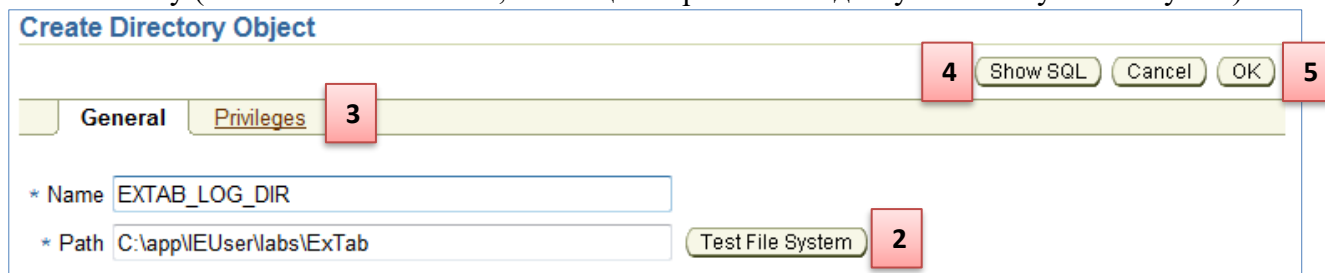
Для редактирования или удаления объекта directory выберите требуемый объект и щелкните на соответствующей кнопке.

Создание объектов Directory

1. На странице Directory Objects щелкните на кнопке Create.



2. Введите имя объекта directory и отображаемый им путь в ОС. Каталог ОС должен быть создан до его использования. Существование каталога ОС можно проверить, щелкнув на кнопке "Test File System". Для выполнения теста надо указать данные для входа в хост-систему (т.е. пользователя ОС, имеющего привилегии доступа к этому каталогу ОС).



3. Следует различать права доступа к объекту directory и предоставляемые в ОС права доступа к физической директории (каталогу) в файловой системе на сервере. Администратор базы данных может осуществлять управление привилегиями доступа к отдельным объектам directory. Такие права повышают уровень безопасности и предоставляют

возможность дифференцированного управления такими объектами. На странице с закладкой Privileges щелкните на кнопке Add, чтобы выбрать пользователя, которому необходимо дать привилегию read или write или же обе эти привилегии.

Create Directory Object Show SQL Cancel OK

General **Privileges**

This page shows the list of users who have privileges for this directory Add

Remove

[Select All](#) | [Select None](#)

Select User Name	Read Access	Write Access
<input type="checkbox"/> HR	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

General **Privileges**

4. Щелкните на кнопке Show SQL, чтобы просмотреть команды, которые будут выполнять действия.

Show SQL Return

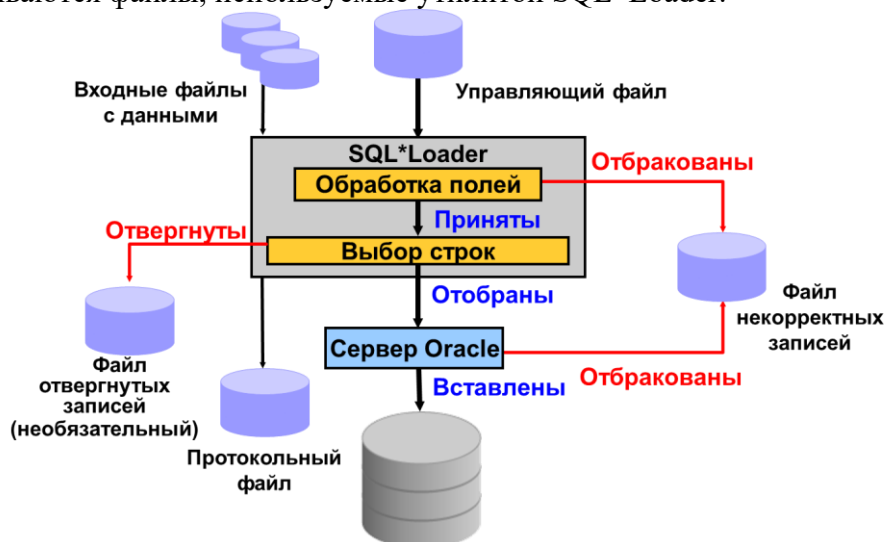
```
CREATE DIRECTORY "EXTAB_LOG_DIR" AS 'C:\app\IEUser\labs\ExTab'
GRANT READ ON DIRECTORY "EXTAB_LOG_DIR" TO "HR"
GRANT WRITE ON DIRECTORY "EXTAB_LOG_DIR" TO "HR"
```

5. Для создания объекта щелкните на кнопке OK.

Создайте средствами ОС папки Labs и ExTab: «C:\app\IEUser\labs\ExTab\». Создайте Directory – объект EXTAB_LOG_DIR, «C:\app\IEUser\labs\ExTab\», пользователю HR предоставьте привилегии read и write.

Обзор утилиты SQL*Loader

SQL*Loader – это утилита, используемая для загрузки данных из внешних файлов в таблицы базы данных Oracle. Эта утилита имеет мощную машину разбора загружаемых данных и накладывает небольшие ограничения на формат данных во внешнем файле. Ниже описываются файлы, используемые утилитой SQL*Loader.



Входные файлы данных (Input Data Files). SQL*Loader читает данные из одного или более файлов (или эквивалентных файлам объектов операционной системы), имена которых заданы в управляющем файле. С позиции SQL*Loader информация в файле данных располагается в виде записей (records). Файл данных может быть в одном из трех форматов и в зависимости от этого содержать записи фиксированного формата, переменного формата или потокового формата. Формат записи может быть задан в параметре INFILE управляющего файла. Если ни один формат не указан, по умолчанию предполагается потоковый формат записей.

Управляющий файл (Control File) – это текстовый файл, содержащий инструкции языка, понимаемого утилитой SQL*Loader. Управляющий файл загрузчика сообщает утилите SQL*Loader, где найти загружаемые данные, как анализировать и интерпретировать данные, куда их вставлять и многое другое. Хотя это точно не определено, можно сказать, что управляющий файл загрузчика имеет три секции.

- Первая секция содержит общую информацию о сеансе, например:
 - глобальные опции: имя входного файла данных, количество пропускаемых записей во входном файле данных (SKIP) и т.д.;
 - предложение INFILE для указания местонахождения входных данных;
- Вторая секция состоит из одного или более блоков INTO TABLE. Каждый из этих блоков содержит информацию о таблице, в которую загружаются данные (имя таблицы и имена ее столбцов).
- Третья секция необязательная и, если она имеется, то содержит входные данные.

Журнальный файл (Log File) создается утилитой SQL*Loader, когда она начинает выполняться. Если файл нельзя создать, выполнение прекращается. Журнальный файл содержит детальную информацию о загрузке, в том числе и описание ошибок, возникших в ходе загрузки.

Файл некорректных записей (Bad File) содержит записи, которые были отбракованы либо при проверке утилитой SQL*Loader, либо при вставке сервером Oracle. Записи файла данных отвергаются утилитой SQL*Loader, если формат входных данных неверен. Если запись файла данных принята утилитой SQL*Loader, она передается для вставки в таблицу в виде строки таблицы. База данных Oracle проверяет правильность строки. Если строка правильная, она вставляется в таблицу; в противном случае, запись отвергается и SQL*Loader записывает ее в файл некорректных записей.

Файл отвергнутых записей (Discard File) создается, если только он необходим и если было задано его создание. Этот файл содержит не подходящие для загрузки записи, которые не удовлетворили хотя бы одному из критериев отбора, заданных в управляющем файле.

Дополнительную информацию о SQL*Loader см. в документе Oracle Database Utilities.

Загрузка данных с помощью SQL*Loader

Для загрузки данных из плоского файла в базу данных Oracle используется мастер, вызываемый по ссылке Load Data from User Files. Вызвать мастер загрузки можно по ссылке в Enterprise Manager «Data Movement > Move Row Data > Load Data from User Files. Далее надо выполнить предлагаемые мастером шаги, т.е. подготовить и передать на выполнение задание загрузки данных

Move Row Data

[Export to Export Files](#)
[Import from Export Files](#)
[Import from Database](#)
[Load Data from User Files](#)
[Monitor Export and Import Jobs](#)

Load Data: Generate Or Use Existing Control File

Database **kb** Cancel Continue

☒ Automatically Generate Control File
A control file will be generated after you define the structure of the data file.

☐ Use Existing Control File
Allows you to use an existing control file that defines the structure of the data file.

Host Credentials

* Username
* Password
☒ Save as Preferred Credential

[Data Files](#)
[Table and File Format](#)
[File Format Attributes](#)
[Load Method](#)
[Options](#)
[Schedule](#)
[More](#)

Load Data: Data Files

Database: **kb** Cancel Step 1 of 7 Next

☐ Data File is Located on Database Server Machine
 Server Data File:
Provide the full path and name of an existing data file on the database server machine.

☒ Data File is Located on Your Browser Machine
Data File will be uploaded to the database server machine.
 Browser Data File: Browse...
Provide the full path and name of an existing data file on your browser machine. NOTE: file size must be less than 10MB; files containing binary or multibyte data may not transfer correctly.

Temporary File Location:
Provide the full directory path on the database server machine where files are to be copied.
☒ **TIP** There may be a delay when you click Next if you have specified a data file on your browser machine. It will be uploaded to the database server machine when you click Next.

SQL*Loader: управляющий файл

Управляющий файл SQL*Loader – это текстовый файл, содержащий инструкции языка определения данных (DDL). DDL используется для управления следующими аспектами сеанса SQL*Loader :

- где SQL*Loader ищет данные для загрузки;
- в каком формате ожидает SQL*Loader обрабатываемые входные данные;
- в какой конфигурации SQL*Loader загружает данные (управление памятью, отбраковка записей, прерывание процесса обработки и т.д.);
- как SQL*Loader изменяет загружаемые данные.

Методы загрузки



Обычная загрузка	Прямая загрузка
Использует COMMIT	Использует <i>сохранение данных</i> (более быстрая операция)
Всегда генерируются журнальные записи	Генерация журнала только при определенных условиях
Обеспечивает выполнение всех ограничений	Выполнение только ограничений PRIMARY KEY, UNIQUE и NOT NULL
Срабатывают триггеры INSERT	Триггеры INSERT не срабатывают
Можно выполнять загрузку в кластеризованные таблицы	Нельзя выполнять загрузку в кластеризованные таблицы
Другие пользователи могут вносить изменения в таблицы	Другие пользователи не могут вносить изменения в таблицы

Сравнение прямой и обычной загрузки

Способ сохранения данных

В обычном режиме загрузки (conventional path loads) для сохранения данных используется обработка SQL и фиксация транзакций в базе данных. За вставкой массивов записей следует операция фиксации транзакции. Каждая загрузка данных может включать несколько транзакций.

Для записи блоков данных в файлы данных Oracle при загрузке в прямом режиме (direct path load) используется сохранение данных. Поэтому прямой режим загрузки быстрее обычного. Отличие сохранения данных от фиксации транзакции состоит в следующем:

- во время сохранения данных в базу данных записываются только полные блоки данных;
- они записываются выше верхней границы таблицы (отметки максимального заполнения, high-water mark, HWM);
- после сохранения данных эта отметка перемещается;
- после сохранения данных внутренние ресурсы не освобождаются;
- сохранение данных не завершает транзакцию;

индексы не обновляются при каждом сохранении данных.

Примечание: прямая загрузка (direct path load) и параллельная прямая загрузка (parallel direct path load) очень похожи в части операций DML, поэтому они отдельно не выделяются в этом сравнении.

Журнализация изменений

Обычная загрузка данных генерирует журнальные записи, как любой оператор DML. При прямой загрузке журнальные записи не генерируются, если:

база данных находится в режиме NOARCHIVELOG;

база данных находится в режиме ARCHIVELOG, однако журнализация отключена (журнализация отключается при установке для таблицы атрибута NOLOGGING или при использовании в управляющем файле фразы UNRECOVERABLE).

Обеспечение выполнения ограничений

Во время загрузки в обычном режиме обеспечивается выполнение всех включенных ограничений, как в любой операции DML.

Во время загрузки в прямом режиме ограничения обрабатываются следующим образом:

- Ограничения NOT NULL проверяются во время построения массивов.
- Ограничения FOREIGN KEY и CHECK отключаются и могут быть включены в конце процесса загрузки путем использования соответствующих команд в управляющем файле. Ограничения FOREIGN KEY отключаются, потому что они ссылаются на другие строки или таблицы, а ограничения CHECK – потому что могут использовать функции SQL. Если в большую таблицу вставляется небольшое число строк, используйте обычный режим загрузки.
- Ограничения PRIMARY KEY и UNIQUE проверяются во время и в конце загрузки, и в случае нарушения их можно отключить.

Срабатывание триггеров INSERT

При обычной загрузке триггеры INSERT срабатывают, а при прямой загрузке они отключаются перед выполнением и снова включаются в конце выполнения. В случае, когда в конце выполнения ссылочный объект недоступен, эти триггеры могут остаться отключенными. Для загрузки данных в таблицы с включенными триггерами INSERT рекомендуется использовать обычный режим.

Загрузка в кластеризованные таблицы

Прямой режим не может использоваться для загрузки данных в кластеризованные таблицы. Данные в кластеризованные таблицы загружаются только в обычном режиме.

Блокирование

Во время выполнения загрузки в прямом режиме другие транзакции не могут вносить изменения в загружаемые таблицы. Единственным исключением из этого правила является случай, когда одновременно выполняется несколько параллельных сеансов прямой загрузки.

