

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

Дисциплина: Основы построения защищенных баз данных.

**Лабораторная работа №10.
Администрирование Oracle. Шифрование.**

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.

Обзор прозрачного шифрования данных (TDE) в Oracle

- Требуется для защиты информации
- Автоматическое шифрование важной информации:
 - Встроенная возможность базы данных Oracle
 - Не нужно вносить изменений в логику приложений
 - Шифрование значений данных и индексов
- Использование ключа шифрования:
 - Главный ключ для базы данных в целом
 - Хранение в Oracle Wallet

Функциональная возможность Oracle Transparent Data Encryption упрощает шифрование важной персональной информации, например, номеров кредитных карточек и номеров социального страхования. Прозрачное шифрование данных (Transparent Data Encryption – TDE) устраняет необходимость в программах шифрования внутри приложений и существенно снижает стоимость и сложность шифрования. С помощью простых команд можно зашифровать важные данные приложений.

Большинство решений шифрования вызывает необходимость использования вызовов определенных функций шифрования внутри кода приложения. Это дорогие решения, так как они обычно требуют углубленного понимания не только приложения, но и возможностей написания и сопровождения программного обеспечения. Как правило, большинство

организаций не имеют времени и опыта изменения существующих приложений и внесения в них вызовов программ шифрования. Oracle Transparent Data Encryption позволяет решить проблему шифрования с помощью встроенной в базу данных Oracle возможности шифрования.

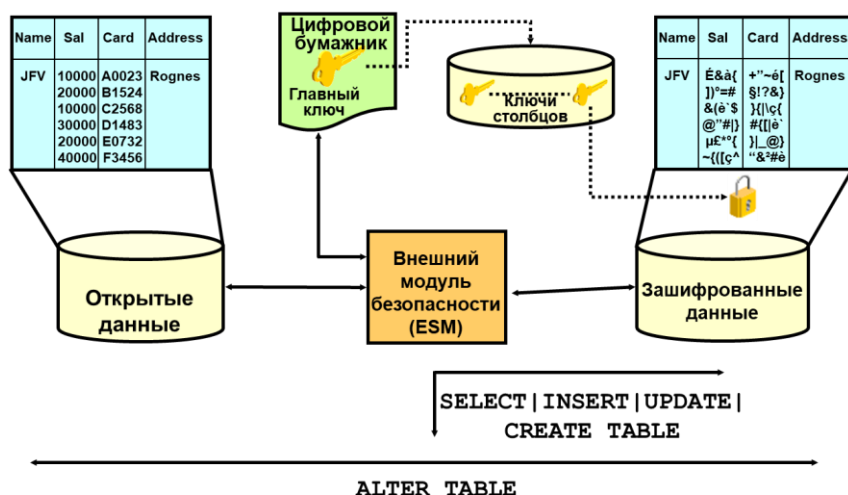
Логические конструкции приложений, реализованные на SQL, продолжают работать без изменений. Так в приложении может использоваться обычный синтаксис команд вставки в таблицы. База данных Oracle автоматически шифрует данные перед записью информации на диск. При выполнении последующих запросов данные прозрачно дешифруются. Поэтому приложение будет продолжать нормально функционировать.

Это важно, поскольку в существующих приложениях обычно выводятся незашифрованные данные. Вывод данных в зашифрованном виде может, как минимум, обескуражить пользователя приложения и даже прервать выполнение существующего приложения.

Шифрование обычно вызывает проблемы для существующих индексов приложения, так как индексные данные не шифруются. Функциональная возможность Oracle Transparent Data Encryption шифрует индексные значения, связанные с данной таблицей приложения. Это означает, что в результате поиск по совпадению ключа лишь незначительно понизит производительность.

Oracle Transparent Data Encryption предоставляет инфраструктуру управления ключами, необходимую для реализации шифрования. Шифрование производится путем передачи открытых текстовых данных вместе с секретным ключом (secret) в программу шифрования. Используя предоставленный ключ, программа шифрует открытые текстовые данные и возвращает зашифрованные данные. Обычно раньше за создание и сопровождение секрета (secret) или ключа отвечало приложение. Oracle Transparent Data Encryption решает эту задачу, автоматически генерируя главный ключ (master key) для всей базы данных. Сразу после старта БД Oracle администратор должен с помощью пароля открыть объект, называемый Oracle Wallet. Пароль должен отличаться от системного и пароля АБД. Объект wallet (цифровой бумажник) использует сертификат от уполномоченной стороны (Certificate Authority). После открытия цифрового бумажника администратор инициализирует главный ключ БД, который генерируется автоматически.

Процесс TDE



Хотя механизмы авторизации и аутентификации эффективно защищают информацию базы данных, они не препятствуют доступу на уровне операционной системы к файлам, в которых хранятся данные. Возможность Transparent Data Encryption позволяет зашифровать важные данные, находящиеся в столбцах БД, и хранить их в зашифрованном виде в файлах операционной системы, делая невозможным выборку данных из файлов в открытом виде.

TDE использует внешний модуль безопасности (External Security Module – ESM) для генерации ключей шифрования, предоставления функций шифрования и дешифрования, а также для безопасного хранения ключей шифрования внутри и вне базы данных.

Для таблицы с шифруемыми столбцами используется единственный ключ столбцов (column key), независимо от количества шифруемых столбцов таблицы. Ключи для всех таблиц хранятся в единственном столбце таблицы словаря базы данных. Этот столбец шифруется с помощью главного ключа (master key) сервера баз данных, что препятствует несанкционированному доступу и использованию этих ключей. Главный ключ хранится в цифровом бумажнике (wallet) вне базы данных. Цифровой бумажник создается с помощью Oracle Wallet Manager, а главный ключ генерируется с помощью ESM.

На рисунке представлена таблица EMPLOYEES, два столбца которой должны быть зашифрованы. Ключ столбцов для таблицы EMPLOYEES поступает от ESM и используется для шифрования этих столбцов. С помощью такого механизма можно шифровать и дешифровать столбцы в базе данных, используя простую команду ALTER TABLE. После того, как произведено шифрование столбцов, их содержимое можно получить в открытом виде, используя обычные команды SELECT (прозрачное дешифрование данных выполняет ESM).

Реализация прозрачного шифрования данных

Для реализации и конфигурирования этой возможности требуется выполнить небольшое число шагов:

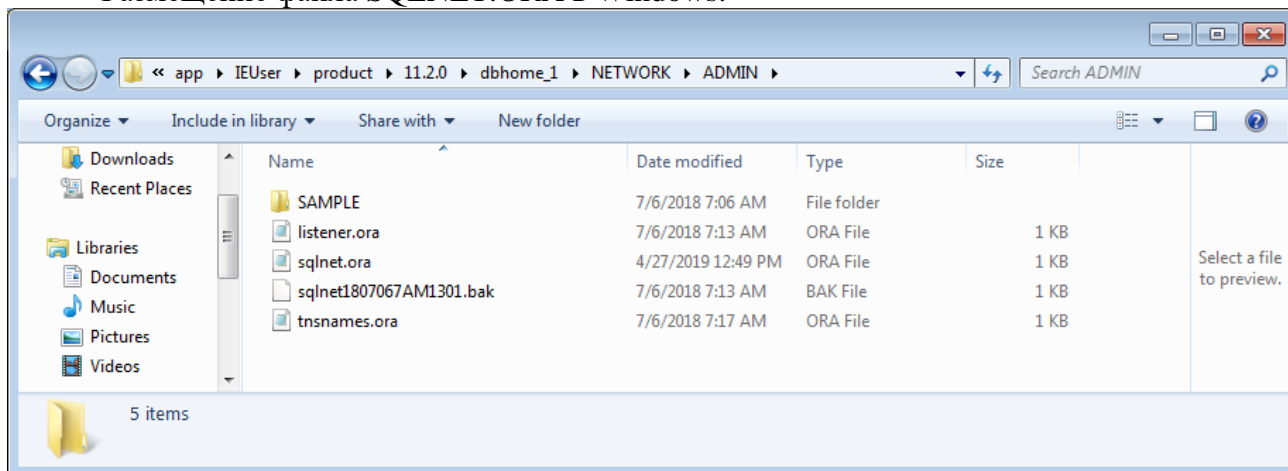
1. Необходимо создать цифровой бумажник (wallet). Это можно сделать либо вручную, используя Oracle Wallet Manager, либо программное обеспечение Transparent Data Encryption (TDE) создаст его автоматически, если директория для цифрового бумажника указана в файле SQLNET.ORA. По умолчанию незашифрованный цифровой бумажник (cwallet.sso) создается при инсталляции базы данных. Однако для TDE используется зашифрованный цифровой бумажник (ewallet.p12).

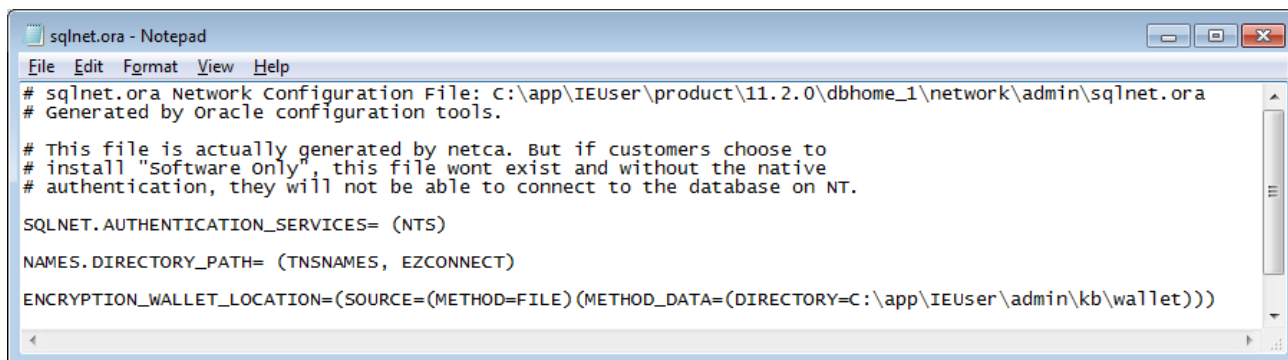


Пример записи в файле SQLNET.ORA:

```
ENCRYPTION_WALLET_LOCATION=(SOURCE=(METHOD=FILE)(METHOD_DATA=
(DIRECTORY=C:\app\IEUser\admin\orcl\wallet)))
```

Размещение файла SQLNET.ORA в Windows:





```
sqlnet.ora - Notepad
File Edit Format View Help
# sqlnet.ora Network Configuration File: C:\app\IEUser\product\11.2.0\dbhome_1\network\admin\sqlnet.ora
# Generated by Oracle configuration tools.

# This file is actually generated by netca. But if customers choose to
# install "Software only", this file wont exist and without the native
# authentication, they will not be able to connect to the database on NT.

SQLNET.AUTHENTICATION_SERVICES= (NTS)

NAMES.DIRECTORY_PATH= (TNSNAMES, EZCONNECT)

ENCRYPTION_WALLET_LOCATION=(SOURCE=(METHOD=FILE) (METHOD_DATA=(DIRECTORY=C:\app\IEUser\admin\kb\wallet)))
```

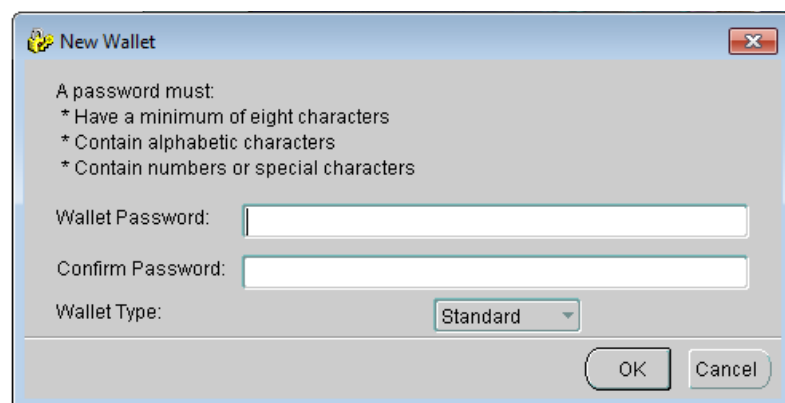
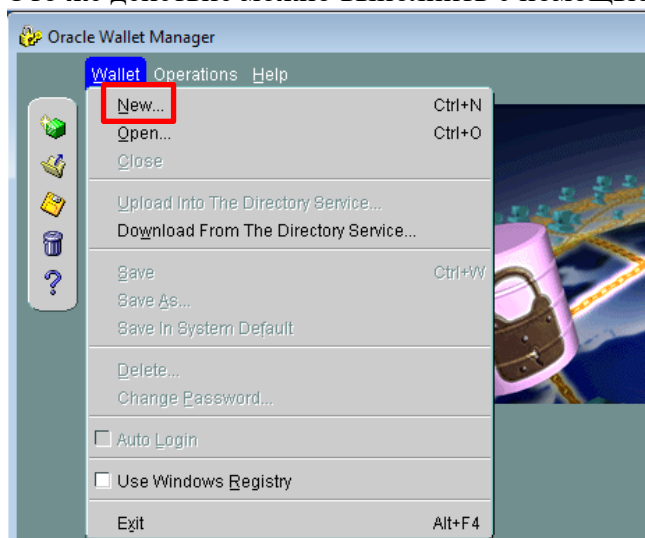
Примечание. В файле sqlnet.ora можно найти две похожих записи: первая содержит параметр WALLET_LOCATION и используется для аутентификации по SSL (Secure Sockets Layer, протокол защищенных сокетов); вторая запись с параметром задается для TDE.

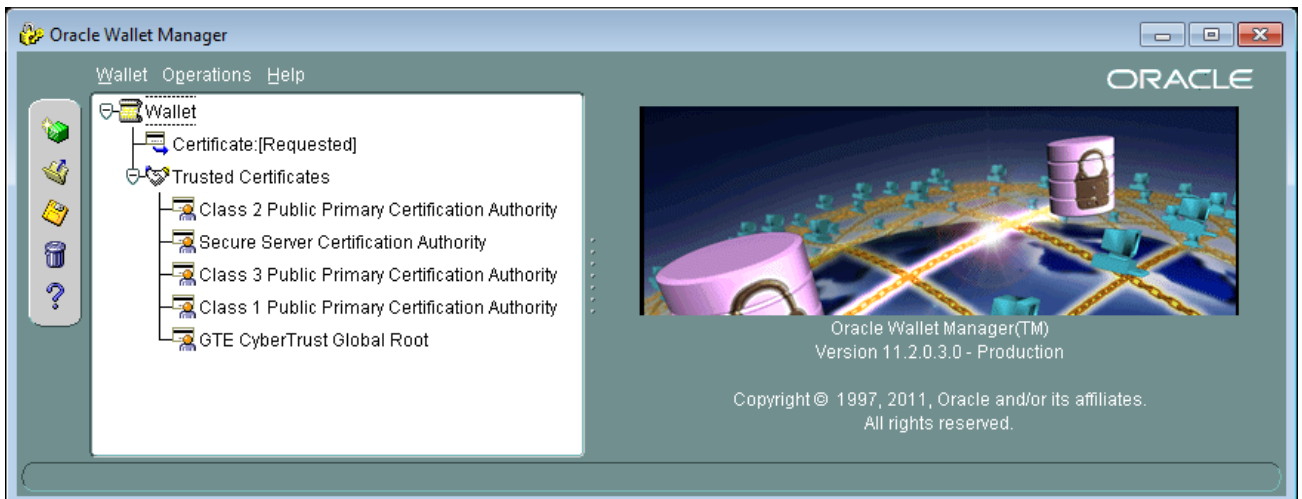
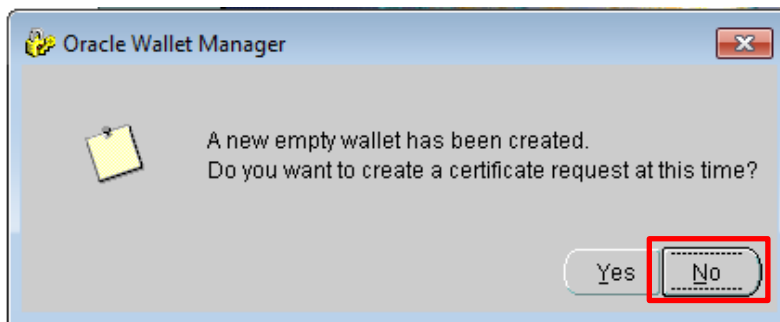
2. Необходимо сгенерировать главный ключ, который хранится в цифровом бумажнике. Главный ключ следует регенерировать только в случае его несанкционированного раскрытия. Частая регенерация главного ключа может привести к отсутствию доступного места в цифровом бумажнике. Вы можете установить или переустановить главный ключ, используя команду ALTER SYSTEM. Если в указанной ENCRYPTION_WALLET_LOCATION директории нет зашифрованного цифрового бумажника, команда создает зашифрованный бумажник (ewallet.p12). Кроме того, команда открывает бумажник, а также создает или пересоздает главный ключ для TDE.

Команда создания главного ключа:

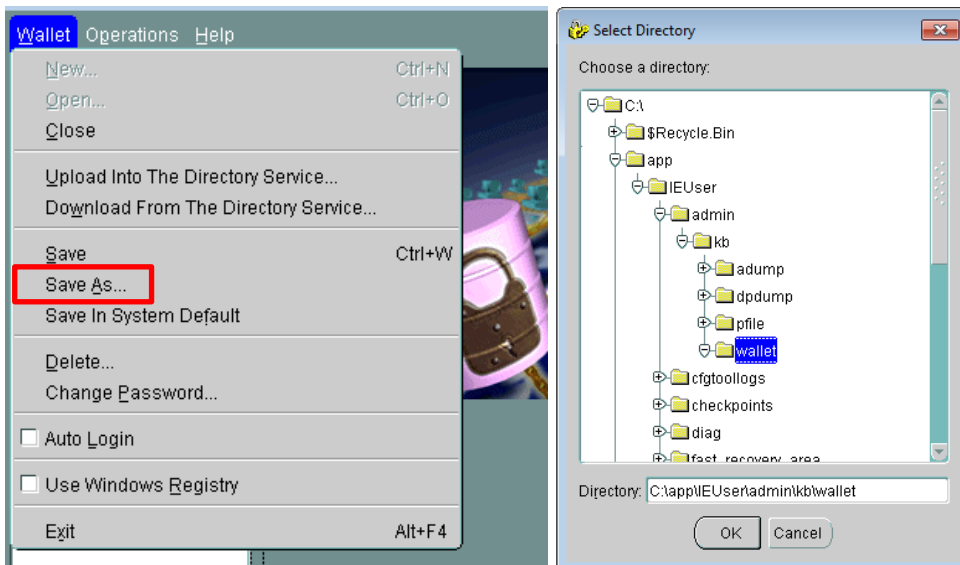
```
ALTER SYSTEM SET ENCRYPTION KEY IDENTIFIED BY <пароль>;
```

Это же действие можно выполнить с помощью приложения Oracle Wallet Manager:





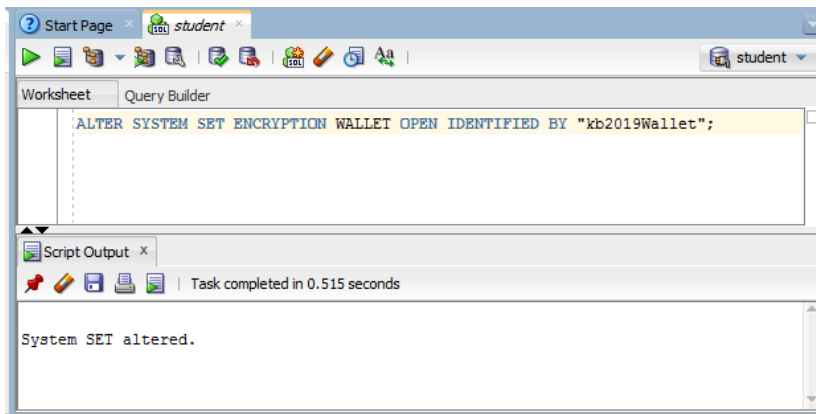
Save wallet



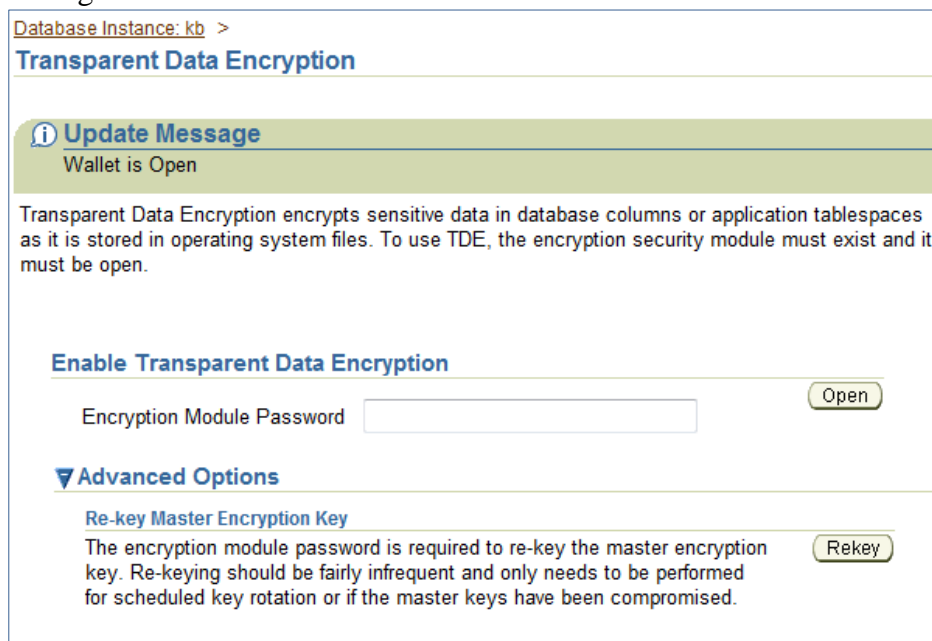
Далее для работы необходимо открыть цифровой бумажник, как описано в п.3.

3. В последующих сеансах не требуется использовать команду, выполненную на шаге 2, и создавать новый главный ключ. Необходимо, чтобы цифровой бумажник был открыт (он закрывается при остановке базы данных). Открыть цифровой бумажник можно, используя команду:

```
ALTER SYSTEM SET ENCRYPTION WALLET OPEN IDENTIFIED BY <пароль>;
```

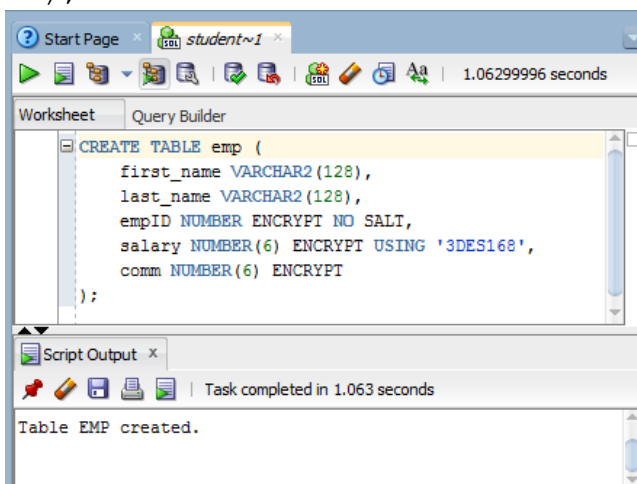


Те же действия можно выполнить на странице «Server – Transparent Data Encryption» Enterprise Manager.



4. Пример команды создания таблицы с шифруемыми столбцами:

```
CREATE TABLE emp (
  first_name VARCHAR2(128),
  last_name VARCHAR2(128),
  empID NUMBER ENCRYPT NO SALT,
  salary NUMBER(6) ENCRYPT USING '3DES168',
  comm NUMBER(6) ENCRYPT
);
```



По умолчанию столбцы шифруются с применением «соли» или «помехи» (salt). Использование «соли» – это метод, повышающий защиту шифруемых данных. Salt – случайная строка, добавляемая к данным перед тем, как они шифруются. Это усложняет для злоумышленника расшифровку данных, которую он выполняет путем сопоставления зашифрованного текста с известными образцами шифруемого текста. Однако «соль» нельзя использовать (NO SALT), если для зашифрованного столбца будет создаваться индекс.

Для столбца, который будет использоваться в индексе или внешнем ключе, необходимо указывать опцию NO SALT. Добавление или удаление «помехи» шифруемого столбца производится по команде ALTER TABLE MODIFY с параметром SALT (по умолчанию) или NO SALT, указанным во фразе ENCRYPT.

```
SQL> ALTER TABLE emp MODIFY (first_name DECRYPT);
```

Для TDE по умолчанию используется алгоритм AES (Advanced Encryption Standard - усовершенствованный стандарт шифрования) с 192-битовым ключом (AES192). Как показано в примере, вы можете выбрать другой алгоритм, например, 3DES (Triple Data Encryption Standard – трехкратное применение алгоритма DES).

5. Шифруемый столбец добавляется в существующую таблицу по команде ALTER TABLE ADD, в которой новый столбец указывается фразой ENCRYPT.

```
SQL> ALTER TABLE emp ADD (ssn VARCHAR2(11) ENCRYPT);
```

6. Существующие незашифрованные столбцы в таблице также можно зашифровать. Для этого используется команда ALTER TABLE MODIFY, в которой для незашифрованных столбцов задается фраза ENCRYPT.

```
SQL> ALTER TABLE emp MODIFY (first_name ENCRYPT);
```

7. Для совместимости с предыдущими версиями или по причинам производительности может понадобиться отключить шифрование. Используйте для этого команду ALTER TABLE MODIFY с фразой DECRYPT.

```
SQL> ALTER TABLE emp MODIFY (first_name DECRYPT);
```

8. Каждая таблица может иметь хотя бы один ключ шифрования для своих столбцов. Этот ключ может быть изменен, когда используется первоначальный алгоритм шифрования, а также путем применения другого алгоритма, указываемого в опции REKEY.

```
SQL> ALTER TABLE emp REKEY USING '3DES168';
```

Примечание: дополнительные сведения о команде ALTER TABLE и ее опциях см. в документе Oracle Database SQL Reference.

Прозрачное шифрование данных: указания

Нельзя шифровать столбцы таблиц, принадлежащих пользователю SYS.

Для данных типа LONG и LOB шифрование не поддерживается.

Любой пользователь, обладающий правом создания таблицы, может создавать таблицы с шифруемыми столбцами. Шифруемые столбцы должны иметь один и тот же ключ шифрования и один и тот же алгоритм. По умолчанию подразумевается алгоритм AES192.

Используемые алгоритмы шифрования:

- 3DES168
- AES128
- AES192
- AES256

Необходимо задавать опцию NO SALT для индексируемых столбцов, например, для столбцов главного или уникального ключа. Опция NO SALT также должна использоваться для столбцов внешнего ключа.

В индексах содержится зашифрованная информация, когда шифруются соответствующие индексируемые столбцы. Поскольку при шифровании данных изменяется их логическая структура, поиск по диапазону невозможен.

Зашифрованные данные должны дешифроваться перед обработкой в выражениях, используемых в запросах и операциях DML (например, должны быть предварительно дешифрованы данные выражений в списке выбора команды select, в ограничении check, в условиях where или when).

Примечание: рекомендуется резервировать цифровой бумажник (wallet) перед переустановкой главного ключа и после этого действия.

1. В параметре в файле SQLNET.ORA укажите путь к папке, где будет храниться цифровой бумажник (wallet). Путь должен полностью существовать.

Например:

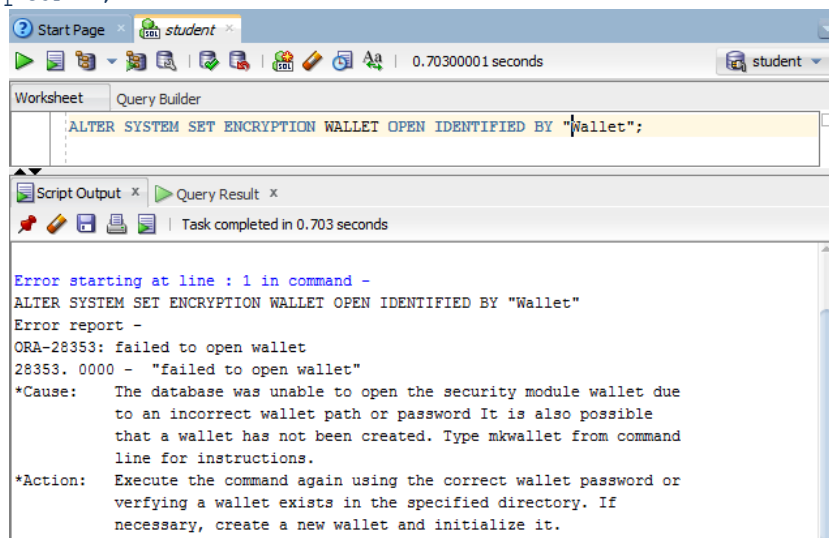
```
ENCRYPTION_WALLET_LOCATION=(SOURCE=(METHOD=FILE)(METHOD_DATA=
(DIRECTORY=C:\app\IEUser\admin\orcl\wallet)))
```

Со стандартными настройками папки wallet в C:\app\IEUser\admin\orcl нет, ее надо создать

2. С помощью приложения Oracle Wallet Manager создайте цифровой бумажник: сгенерируйте главный ключ, сохраните его в папке, указанной в параметре ENCRYPTION_WALLET_LOCATION.

3. В приложении SQL Developer с учетной записью ora1\ora1 попытайтесь открыть бумажник с неверным паролем:

```
ALTER SYSTEM SET ENCRYPTION WALLET OPEN IDENTIFIED BY "неверный_пароль";
```



4. В приложении SQL Developer с учетной записью ora1\ora1 откройте цифровой бумажник.

5. Создайте таблицу с шифруемыми столбцами:

```
CREATE TABLE emp (
  first_name VARCHAR2(128),
  last_name VARCHAR2(128),
  empID NUMBER ENCRYPT NO SALT,
  salary NUMBER(6) ENCRYPT USING '3DES168',
  comm NUMBER(6) ENCRYPT
```

```
);
```

6. Наполните таблицу данными:

```
insert into emp values ('Steven', 'King', 100, 24000, 90);
```

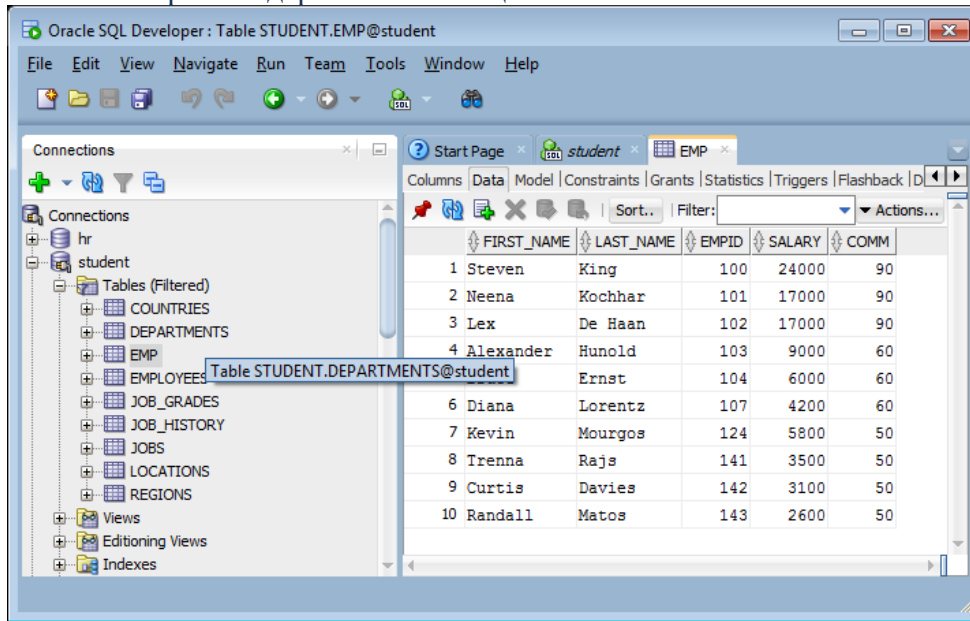


```

insert into emp values ('Neena', 'Kochhar', 101, 17000, 90);
insert into emp values ('Lex', 'De Haan', 102, 17000, 90);
insert into emp values ('Alexander', 'Hunold', 103, 9000, 60);
insert into emp values ('Bruce', 'Ernst', 104, 6000, 60);
insert into emp values ('Diana', 'Lorentz', 107, 4200, 60);
insert into emp values ('Kevin', 'Mourgos', 124, 5800, 50);
insert into emp values ('Trenna', 'Rajs', 141, 3500, 50);
insert into emp values ('Curtis', 'Davies', 142, 3100, 50);
insert into emp values ('Randall', 'Matos', 143, 2600, 50);

```

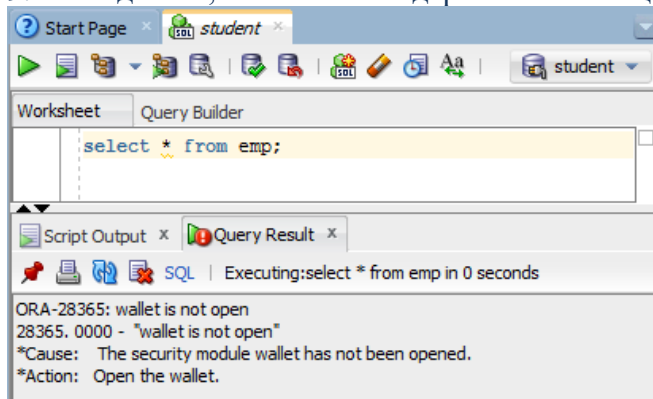
7. Посмотрите содержимое таблицы EMP.



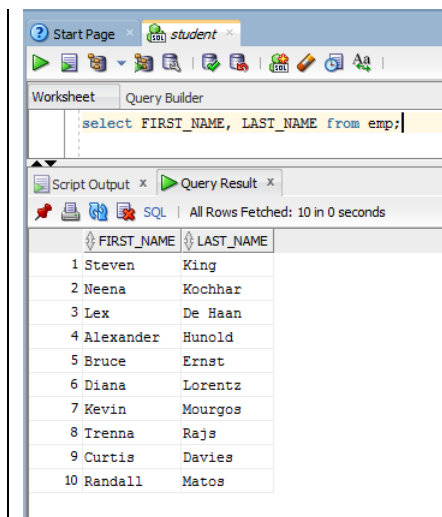
8. Закройте цифровой бумажник.

ALTER SYSTEM SET ENCRYPTION WALLET CLOSE IDENTIFIED BY "пароль";

9. Убедитесь, что полное содержимое таблицы EMP недоступно.



10. Сделайте выборку из незашифрованных столбцов:



Поддержка имен пользователей и паролей в цифровом бумажнике

Парольные мандаты (password credentials, имена пользователей и их пароли) для соединения с базами данных можно хранить на клиентской стороне в Oracle Wallet. Такой цифровой бумажник является безопасным контейнером хранения парольных мандатов для аутентификации и мандатов цифровых подписей (signing credentials).

Использование цифрового бумажника может упростить широкомасштабную задачу развертывания, решение которой основывается на использовании парольных мандатов для установления соединений с базами данных. Если такая возможность сконфигурирована, код приложения, пакетные задания и скрипты могут не содержать внутри себя имена пользователей и их пароли. Это повышает безопасность, так как пароли не предоставляются в открытом виде. Кроме того, проще осуществлять политику управления паролями, не модифицируя код приложений при изменении имен пользователей или их паролей.

Когда на клиенте сконфигурировано безопасное внешнее хранение паролей, подсоединение приложений к базе данных производится с помощью команды следующего вида, в которой не указываются имя и пароль для входа в базу данных: `connect /@db_connect_string`.

Мандаты баз данных помещаются в цифровой бумажник Oracle (Oracle Wallet), созданный для их безопасного хранения. Возможность автоподсоединения (autologin feature) при использовании Oracle Wallet включена, поэтому в системе не требуется указывать пароль для открытия цифрового бумажника.

Чтобы сконфигурировать описанную возможность, необходимо создать Oracle Wallet на стороне клиента с помощью команды `mkstore`. Затем следует добавить имя пользователя и пароль для определенной строки соединения. Это действие также выполняется с помощью утилиты `mkstore`. После этого необходимо убедиться в том, что в файле `sqlnet.ora` в параметре `WALLET_LOCATION` находится правильный указатель месторасположения бумажника.

Утилита Data Pump и прозрачное шифрование данных

Следующие два фактора необходимо учитывать при экспорте таблиц с зашифрованными столбцами. Во-первых, важные данные транспортируются не в открытом виде. Во-вторых, авторизованные пользователи могут дешифровать такие данные после импорта в целевую БД.

Так как ключ для дешифрования локально хранится на сервере, где первоначально размещаются таблицы, их дешифрование на принимающей стороне возможно только с использованием ключа принимающей стороны. Поэтому перед экспортом администратор переустанавливает ключи для таблиц с помощью парольного ключа (password key), который затем тайно сообщает администратору принимающей стороны. При импорте администратор указывает этот пароль. Соответствующие столбцы в ходе импорта дешифруются, что позволяет принимающему серверу немедленно повторно шифровать эти столбцы с помощью

ключа локального сервера. После этого использование столбцов на новом месте основывается на обычном механизме авторизации.

```
ENCRYPTION_PASSWORD = <пароль>
```

Похожий метод применяется для внешних таблиц, использующих драйвер доступа ORACLE_DATAPUMP. Для шифрования определенных столбцов внешней таблицы при описании этих столбцов указывается фраза ENCRYPT. В результате случайным образом генерируется ключ, используемый для шифрования столбцов.

Однако при переносе внешней таблицы ключ недоступен в новом месторасположении. Поэтому для такой таблицы администратор должен указать свой собственный пароль, чтобы зашифровать столбцы. Затем после переноса данных можно воспользоваться этим же паролем для регенерации ключа. В результате можно получить доступ к зашифрованным столбцам в новом месторасположении.

```
CREATE TABLE emp_ext (
    first_name, last_name, empID,
    salary ENCRYPT IDENTIFIED BY "xIcf3T9u" )
ORGANIZATION EXTERNAL
( TYPE ORACLE_DATAPUMP
  DEFAULT DIRECTORY "D_DIR"
  LOCATION('emp_ext.dat') )
REJECT LIMIT UNLIMITED
as select * from employees;
```

Обзор создания шифруемых резервных копий RMAN

Для повышения безопасности можно шифровать резервные копии, получаемые с помощью утилиты RMAN. Зашифрованные резервные копии не могут быть прочитаны обычным образом.

RMAN предоставляет три режима шифрования:

Прозрачный режим (transparent mode). Используя прозрачное шифрование, можно создавать и восстанавливать зашифрованные резервные копии без дополнительного вмешательства, пока доступна требуемая инфраструктура управления ключами Oracle. Прозрачное шифрование наилучшим образом подходит для операций ежедневного резервирования, производимых с целью последующего восстановления в том же самом месте. В RMAN по умолчанию используется прозрачный режим шифрования.

Парольный режим (password mode). В этом режиме при создании и восстановлении зашифрованных резервных копий необходимо предоставить пароль. Так при восстановлении зашифрованной подобным образом резервной копии указывается пароль, который использовался при создании этой копии. Парольное шифрование полезно применять для безопасной передачи резервных копий в удаленные месторасположения и их последующего восстановления. Для парольного шифрования невозможно создать постоянную конфигурацию. Поскольку парольное шифрование будет использоваться единолично, не требуется конфигурировать Oracle Wallet.

Двойной режим (dual mode). Резервные копии в этом режиме могут быть восстановлены либо прозрачным образом, либо с предоставлением пароля. Резервные копии, полученные в двойном режиме, обычно полезны при последующем восстановлении на сайте, использующем цифровой бумажник, а также, если иногда требуется восстановить Резервные копии в месторасположении, на котором недоступен цифровой бумажник. В ходе восстановления зашифрованной в двойном режиме резервной копии можно использовать либо Oracle Wallet, либо пароль для выполнения дешифрования.

Настройка прозрачного режима

Для того чтобы изменить существующую среду резервирования таким образом, чтобы все резервные копии RMAN шифровались в прозрачном режиме, выполните следующие шаги:

1. Сконфигурируйте место хранения Oracle Wallet
2. Задайте в экземпляре главный ключ:

```
ALTER SYSTEM SET ENCRYPTION KEY IDENTIFIED BY <пароль>;
```

3. Откройте цифровой бумажник, используя команду ALTER SYSTEM:

```
ALTER SYSTEM SET ENCRYPTION WALLET OPEN IDENTIFIED BY <пароль>;
```

4. Сконфигурируйте **RMAN** для использования прозрачного шифрования:

```
CONFIGURE ENCRYPTION FOR DATABASE ON
```

После этих шагов все резервные наборы RMAN, создаваемые БД, шифруются, если только вы либо временно не переопределите такой режим работы в сеансе RMAN с помощью команды SET ENCRYPTION OFF, либо не измените постоянную установку командой CONFIGURE ENCRYPTION FOR DATABASE OFF. В командах BACKUP, создающих шифруемые резервные копии, не требуется менять аргументы. Шифрование производится на основе установки, заданной командой CONFIGURE ENCRYPTION или SET ENCRYPTION.

RMAN автоматически дешифрует содержимое резервных наборов в ходе операции restore. Пока открыт и доступен Oracle Wallet, не требуется никакого вмешательства при восстановлении прозрачно зашифрованных резервных копий.

Примечание: при потере цифрового бумажника Oracle невозможно выполнить восстановление прозрачно зашифрованных резервных копий.

Установка парольного режима

По соображениям безопасности невозможно постоянным образом изменить существующую среду резервирования таким образом, чтобы все резервные копии RMAN шифровались с использованием парольного режима. Только внутри вашего сеанса RMAN можно задать создание резервных копий, шифруемых в парольном режиме. Для этого в скриптах RMAN используется команда SET ENCRYPTION ON IDENTIFIED BY пароль ONLY. Это команда действует только в течение вашего сеанса RMAN.

После установки пароля по команде SET ENCRYPTION можно использовать обычно выполняемые команды BACKUP. Все получаемые резервные наборы шифруются в парольном режиме.

Чтобы выполнить операцию restore для резервных копий, полученных в парольном режиме, необходимо ввести пароль шифрования, используя команду

```
SET DECRYPTION IDENTIFIED BY пароль1 {, пароль2,..., парольn}.
```

Для восстановления из резервных копий, полученных с различными паролями, требуется задать все необходимые пароли в команде SET DECRYPTION. RMAN автоматически использует правильный пароль для каждого резервного набора.

Примечание: если потерял пароль, использовавшийся при получении зашифрованной резервной копии, невозможно восстановление из этой резервной копии.

Настройка двойного режима

1. Сконфигурируйте место хранения Oracle Wallet. Задайте в экземпляре главный ключ:

```
ALTER SYSTEM SET ENCRYPTION KEY IDENTIFIED BY <пароль>;
```

2. Откройте цифровой бумажник, используя команду ALTER SYSTEM:

```
ALTER SYSTEM SET ENCRYPTION WALLET OPEN IDENTIFIED BY <пароль>;
```

3. Задайте в сеансе **RMAN** использование двойного шифрования:

SET ENCRYPTION ON IDENTIFIED BY password

4. После этого можно создавать резервные копии в том же самом сеансе, в котором был установлен пароль. Ваши команды резервирования изменять не требуется.

5. Позднее, когда потребуется дешифровать бэкапы этого вида, можно будет либо воспользоваться цифровым бумажником, не задавая дополнительной команды, либо использовать правильный пароль, указав его в вашем сеансе RMAN в команде:

SET DECRYPTION IDENTIFIED BY пароль {, пароль2,..., парольn}

6. Ваши команды восстановления изменять не требуется.

Шифруемые резервные копии RMAN: указания

Можно шифровать любые резервные копии RMAN, получаемые в виде резервных наборов. Копии образов зашифровать нельзя.

Для использования шифрования в RMAN необходимо в экземпляре целевой БД задать значение параметра инициализации COMPATIBLE, равное, по крайней мере, 10.2.0.

Представление V\$RMAN_ENCRYPTION_ALGORITHMS содержит список алгоритмов шифрования, поддерживаемых RMAN. Если не указан никакой алгоритм, по умолчанию предполагается AES с ключом шифрования в 128-бит. Вы можете поменять алгоритм, используя команды:

```
CONFIGURE ENCRYPTION ALGORITHM 'algorithmname'  
SET ENCRYPTION ALGORITHM 'algorithmname'
```

Шифрование резервных копий возможно только в Oracle Database Enterprise Edition.

База данных Oracle использует новый ключ шифрования для каждой шифруемой резервной копии. Ключ шифрования резервной копии затем шифруется либо с использованием пароля, либо главного ключа базы данных, либо с использованием обоих методов в зависимости от выбранного режима шифрования. Отдельные ключи шифрования или пароли никогда не хранятся в открытом виде.

Шифрование может негативным образом сказаться на производительности операции резервирования на диск. Так как при получении шифруемых резервных копий требуется больше ресурсов CPU, чем при нешифруемом резервировании, повысить производительность таких операций можно, используя больше каналов RMAN.

Поскольку инфраструктура управления ключами Oracle архивирует все предыдущие главные ключи в цифровом бумажнике, изменение или переустановка текущего главного ключа базы данных не влияет на возможность восстановления из зашифрованных резервных копий, полученных при старых значениях главного ключа. Администратор может переустановить главный ключ БД в любой момент, и RMAN будет в состоянии восстановить зашифрованную резервную копию, когда-либо полученную в данной БД.

<p>Создайте зашифрованную резервную копию табличного пространства Users. В Enterprise Manager на странице Availability в разделе Manage выберите ссылку Schedule Backup.</p>
--

Schedule Backup

Oracle provides an automated backup strategy based on your disk and/or tape configuration. Alternatively, you

Oracle-Suggested Backup

Schedule a backup using Oracle's automated backup strategy.

[Schedule Oracle-Suggested Backup](#)

This option will back up the entire database. The database will be backed up on daily and weekly intervals.

Customized Backup

Select the object(s) you want to back up.

[Schedule Customized Backup](#)

- ☐ Whole Database
- ☒ Tablespaces
- ☐ Datafiles
- ☐ Archived Logs
- ☐ All Recovery Files on Disk
Includes all archived logs and disk backups that are not already backed up to tape.

Host Credentials

To perform a backup, supply operating system login credentials to access the target database.

* Username
* Password
☐ Save as Preferred Credential

Выполните шаги резервирования. При использовании опции «Backups will be encrypted using the Oracle Encryption Wallet» цифровой бумажник должен быть открыт.

●

○○●○●○
Tablespaces Options Settings Schedule Review

Schedule Customized Backup: Tablespaces

Database kb [Cancel](#) [Step 1 of 5](#) [Next](#)

Backup Strategy Customized Backup

Object Type Tablespaces

Populate this table with the tablespaces you want to back up.

[Add](#)

Select Tablespace Name	Tablespace Number	Status	Contents
No Items Selected			

[Return to Schedule Backup](#) [Cancel](#) [Step 1 of 5](#) [Next](#)

Search Results

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

Select	Tablespace Name	Tablespace Number	Status	Contents
<input type="checkbox"/>	SYSTEM	0	ONLINE	PERMANENT
<input type="checkbox"/>	SYSAUX	1	ONLINE	PERMANENT
<input type="checkbox"/>	UNDOTBS1	2	ONLINE	UNDO
<input checked="" type="checkbox"/>	USERS	4	ONLINE	PERMANENT
<input type="checkbox"/>	EXAMPLE	6	ONLINE	PERMANENT

[Cancel](#) [Select](#)

Schedule Customized Backup: Options

Database kb
Backup Strategy Customized Backup
Object Type Tablespaces

Cancel Back Step 2 of 5 Next

Backup Type

- ☒ Full Backup
- ☐ Use as the base of an incremental backup strategy
- ☒ Incremental Backup
- A level 1 cumulative incremental backup includes all blocks changed since the most recent level 0 backup.
- ☐ Refresh the latest datafile copy on disk to the current time using the incremental backup

Advanced

- ☒ Also back up all archived logs on disk
- ☐ Delete all archived logs from disk after they are successfully backed up
 - ☐ Delete obsolete backups
Delete backups that are no longer required to satisfy the retention policy.
 - ☐ Use proxy copy supported by media management software to perform a backup
If proxy copy of the selected files is not supported, a conventional backup will be performed.

Maximum Files per Backup Set

Section Size KB

Backs up large files in parallel, using sections of the specified size. (This parameter overrides Maximum Backup Piece Size in Backup Settings.)

Encryption

Encrypt the backup using the Oracle Encryption Wallet, a user-supplied password, or both, to protect sensitive data.

- ☒ Use Recovery Manager encryption
- Encryption Algorithm AES128
- Encryption Mode ☒ Backups will be encrypted using the Oracle Encryption Wallet
- ☒ Backups will be encrypted using the following password
- ☒ TIP Checking both encryption modes will provide the flexibility of restoring a backup using either the Oracle Encryption Wallet or a password.

Password

Confirm Password

Tablespaces Options Settings Schedule Review

Schedule Customized Backup: Settings

Database kb
Backup Strategy Customized Backup
Object Type Tablespaces

Cancel Back Step 3 of 5 Next

Select the destination media for this backup. You can also override the default backup settings.

☒ Disk

Disk Backup Location C:\app\IEUser\fast_recovery_area

☐ Tape

Media Management Vendor (MMV) Library Parameters Not specified

View Default Settings

Override Default Settings

Changed settings will only apply to the current backup.

Return to Schedule Backup

Cancel Back Step 3 of 5 Next

Tablespaces Options Settings Schedule Review

Schedule Customized Backup: Schedule

Database kb
Backup Strategy Customized Backup
Object Type Tablespaces

Cancel Back Step 4 of 5 Next

Job

Job Name BACKUP_KB_000081

Job Description Tablespaces Backup

Schedule

Type ☒ One Time (Immediately) ☐ One Time (Later) ☐ Repeating

Return to Schedule Backup

Cancel Back Step 4 of 5 Next

Tablespaces
Options
Settings
Schedule
Review

Schedule Customized Backup: Review

Database **kb** Cancel Edit RMAN Script Back Step 5 of 5 Submit Job

Backup Strategy **Customized Backup**
 Object Type **Tablespaces**

Settings

Destination	Disk
Backup Type	Full Backup
Backup Mode	Online Backup
Encryption Algorithm	AES128
Encryption Mode	Oracle Encryption Wallet, Password
Fast Recovery Area	C:\app\IEUser\fast_recovery_area

Tablespaces

USERS

RMAN Script

The RMAN script below is generated based on previous input.

```

set encryption on for all tablespaces algorithm 'AES128' identified by '%PASSWORD';
backup device type disk tag '%TAG' tablespace 'USERS' ;
backup device type disk tag '%TAG' archivelog all not backed up;
  
```

[Return to Schedule Backup](#)
Cancel Edit RMAN Script Back Step 5 of 5 Submit Job

The job has been successfully submitted.

Status

The job has been successfully submitted.
 You can view the status of the job by clicking on the View Job button.

View Job OK

Восстановите из зашифрованной резервной копии табличное пространство Users.
 В Enterprise Manager на странице Availability в разделе Manage выберите ссылку Perform Recovery.

Perform Recovery

Oracle Advised Recovery

Oracle did not detect any failures. Advise and Recover

User Directed Recovery

Recovery Scope Tablespaces Recover

Operation Type

- ☐ Recover to current time or a previous point-in-time
Datafile will be restored as required.
- ☒ **Restore tablespaces**
Specify Time, SCN or log sequence. The backup taken at or prior to that time will be used. No recovery will be performed in this operation.
- ☐ Recover from previously restored tablespaces

Decrypt Backups

If objects are being recovered from password-encrypted backups, supply a comma-separated list of passwords.

Password
 Confirm Password

Host Credentials

To perform recovery, supply operating system login credentials to access the target database.

Username
 Password
☐ Save as Preferred Credential

● ○ ○ ○ ○
 Tablespaces Restore Rename Schedule Review

Perform Object Level Recovery: Tablespaces

Database kb Cancel Step 1 of 5 Next
 Recovery Scope Tablespaces
 Operation Type Restore Only

Populate this table with the tablespaces you want to restore.

Add

Remove

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

Select	Tablespace Name	Tablespace Number	Status	Contents
<input type="checkbox"/>	USERS	4	ONLINE	PERMANENT

[Return to Perform Recovery](#) Cancel Step 1 of 5 Next

○ ● ○ ○ ○
 Tablespaces Restore Rename Schedule Review

Perform Object Level Recovery: Restore

Database kb Cancel Back Step 2 of 5 Next
 Recovery Scope Tablespaces
 Operation Type Restore Only

Backup Selection

Specify a backup from which to restore your files.

☐ The most recent backup
☐ The backup taken at or just prior to the specified point-in-time
☒ The backup specified by a tag

Backup Validation

☐ Validate the specified backup without restoring the datafiles.
If selected, the "Rename" step will be skipped. Datafiles and tablespaces will remain online during the validation.

[Return to Perform Recovery](#) Cancel Back Step 2 of 5 Next

[Database](#) | [Setup](#) | [Preferences](#) | [Help](#) | [Logout](#)

○ ○ ● ○ ○
 Tablespaces Restore Rename Schedule Review

Perform Object Level Recovery: Rename

Database kb Cancel Back Step 3 of 5 Next
 Recovery Scope Tablespaces
 Operation Type Restore Only

Do you want to restore the files to a different location?

☒ No. Restore the files to the default location.
☐ Yes. Restore the files to a new, common location.
This option will execute an RMAN 'rename' operation.
☒ Update the control file to use the renamed files
This is applicable only if you selected "Yes" above.

[Return to Perform Recovery](#) Cancel Back Step 3 of 5 Next

○ ○ ○ ○ ●
 Point-in-time Tablespaces Rename Schedule Review

Perform Object Level Recovery: Review

Database kb Cancel Edit RMAN Script Back Step 5 of 5 Submit
 Recovery Scope Tablespaces
 Operation Type Restore and Recover

Click on the Edit RMAN Script button to view or edit the RMAN script before submitting the operation.

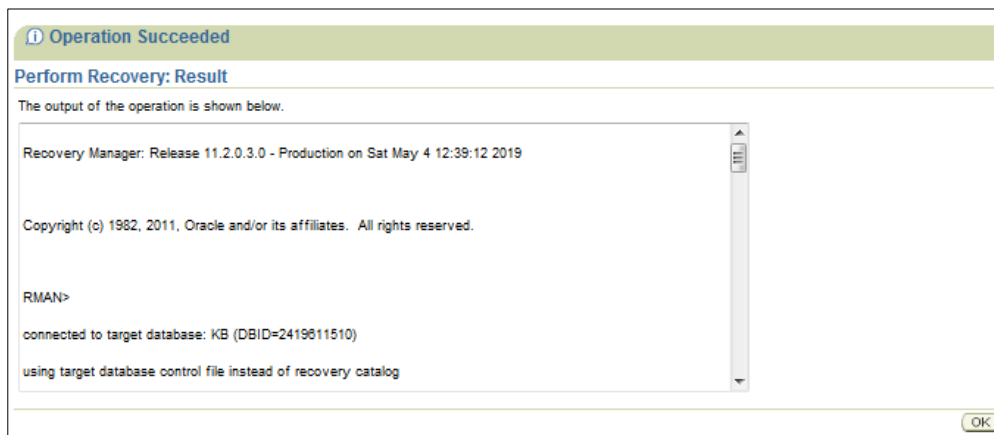
Options

Point-in-time Recover to the current time

Tablespaces

USERS

[Return to Perform Recovery](#) Cancel Edit RMAN Script Back Step 5 of 5 Submit



Проверьте возможность восстановления при закрытом цифровом бумажнике и неправильном пароле.

Время на выполнение лабораторной работы – 2 часа.