

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

Дисциплина: Основы построения защищенных баз данных.

**Лабораторная работа №4.**  
**Администрирование Oracle. Администрирование пользователей,**  
**предоставление привилегий.**

**1. Учебные цели:**

- Отработать вопросы управления экземпляром Oracle в части создания и сопровождения учетных записей пользователей базы данных, предоставления и отмены привилегий, создания и сопровождения ролей.
- Освоить приемы управления учетными записями пользователей, в том числе их аутентификации, привилегий, табличных пространств хранения данных, администрирования ролей.

**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 создания и сопровождения учетных записей пользователей базы данных, предоставления и отмены привилегий, создания и сопровождения ролей.

**3. Перечень материально-технического обеспечения**

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

**4. Краткие теоретические 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.

Далее будут использованы термины:

- *Учетная запись пользователя базы данных (database user account)* – средство упорядочения прав владения и доступа к объектам БД.
- *Пароль (password)* используется при аутентификации пользователей, выполняемой базой данных Oracle.
- *Привилегия (privilege)* – право выполнения команд SQL определенного типа или право доступа к объекту другого пользователя.
- *Роль (role)* – именованная группа связанных привилегий, предоставляемая пользователям или другим ролям.

**Учетные записи пользователей базы данных**

Чтобы получить доступ к базе данных, необходимо указать правильное имя пользователя БД и успешно пройти процедуру аутентификации, как того требует учетная запись этого пользователя. Oracle рекомендует, чтобы каждый пользователь имел свою учетную запись в базе данных. Это лучше всего помогает устранять бреши в безопасности и предоставляет информацию для проведения аудита операций. Однако, в редких случаях.

пользователи могут совместно использовать общую учетную запись в базе данных. При этом операционная система и приложения должны обеспечить соответствующую безопасность базы данных.

Каждая учетная запись пользователя содержит:

- **Уникальное имя пользователя.** Это имя не может содержать более 30 символов, не может содержать специальные символы и должно начинаться с буквы.
- **Метод аутентификации.** Наиболее общий метод аутентификации – это использование пароля. Кроме того, база данных Oracle поддерживает и другие методы аутентификации (биометрический, а также использующие сертификаты и маркеры (token authentication)).
- **Табличное пространство по умолчанию.** В нем пользователи создают свои объекты, если в команде не указывается другое табличное пространство. Отметим, что наличие такого табличного пространства не подразумевает, что пользователь имеет *привилегию* создания объектов в этом табличном пространстве и что у него есть *квота* использования этого табличного пространства при создании объектов. Обе эти возможности предоставляются отдельно.
- **Временное табличное пространство.** Место для создания пользователями временных объектов, например, промежуточных результатов сортировок и временных таблиц.
- **Профиль пользователя.** Набор ограничений пользователя, накладываемых на пароли и системные ресурсы.
- **Группа потребителей.** Используется ресурсным менеджером.
- **Статус блокирования.** Пользователи получают доступ к базе данных только при условии, что их учетная запись «разблокирована».

## Предопределенные учетные записи SYS и SYSTEM

Пользователям SYS и SYSTEM по умолчанию предоставлена роль DBA (роль администратора базы данных).

Кроме того, пользователь SYS имеет все привилегии с атрибутом WITH ADMIN OPTION, а также является владельцем словаря данных. При подсоединении к пользователю SYS необходимо использовать фразу AS SYSDBA. Любой пользователь, которому предоставлена привилегия SYSDBA, может подсоединиться с использованием учетной записи SYS, указав для этого фразу AS SYSDBA. Только “привилегированным” пользователям, обладающим привилегией SYSDBA или SYSOPER, разрешено запускать и останавливать экземпляр базы данных.

По умолчанию пользователю SYSTEM предоставлена роль DBA, но не выделена привилегия SYSDBA.

SYS и SYSTEM – обязательные пользователи базы данных. Их нельзя удалить.

**Подсказка по наилучшему практическому подходу.** В соответствии с принципом выделения наименьших привилегий пользователи SYS и SYSTEM не используются для выполнения рутинных операций. Для пользователей, которым необходимы привилегии АБД, создаются свои отдельные учетные записи и выделяются требуемые привилегии. Например, Джим использует пользователя jim с небольшими привилегиями и “привилегированного” пользователя jim\_dba. Такой подход позволяет применить метод наименьших привилегий, устраняет совместное использование учетных записей и дает возможность осуществлять аудит отдельных действий.

### Создание пользователя.

Oracle Enterprise Manager позволяет сопровождать перечень пользователей, которым разрешен доступ к базе данных. После перехода на страницу Users можно создавать, удалять и изменять установочные параметры пользователя.

Чтобы создать пользователя выберите **Server > Security > Users**, а затем щелкните на кнопке **Create**.

Введите требуемые данные. Звездочкой помечены обязательные элементы, например, имя пользователя (поле Name).

Дополнительные сведения об аутентификации приводятся далее.

Назначайте каждому пользователю табличное пространство по умолчанию и временное табличное пространство. Это позволит контролировать месторасположение создаваемых объектов, когда пользователь не указывает табличное пространство, в котором должен быть создан объект.

Если для создаваемого пользователя не указывается постоянное табличное пространство, тогда используется табличное пространство, определенное в системе по умолчанию. Подобным образом, если не указано временное табличное пространство, пользователю назначается определенное в системе по умолчанию временное табличное пространство.

Database Instance: kb > Users > Logged in As SYS

**Create User**

Show SQL Cancel OK

**General** Roles System Privileges Object Privileges Quotas Consumer Group Privileges Proxy Users

\* Name

Profile DEFAULT

Authentication Password

\* Enter Password

\* Confirm Password

For Password choice, the role is authorized via password.

☐ Expire Password now

Default Tablespace

Temporary Tablespace

Status ☐ Locked ☒ Unlocked

**General** Roles System Privileges Object Privileges Quotas Consumer Group Privileges Proxy Users

Show SQL Cancel OK

## Аутентификация пользователей

*Аутентификация (authentication)* означает проверку идентичности кого-то или чего-то (пользователя, устройства и других объектов), кто хочет использовать данные, ресурсы или приложения. Она производится для установления *доверительного отношения (trust relationship)* перед дальнейшим взаимодействием. Кроме того, аутентификация предоставляет возможность идентификации, позволяя связывать доступ и действия с определенными личностями (identities). После аутентификации процесс *авторизации (authorization)* может предоставить или ограничить уровни доступа и действия, разрешенные данному объекту.

При создании пользователя необходимо решить, какой метод аутентификации будет использоваться. Этот метод позднее может быть изменен.

### - Password (метод аутентификации с использованием пароля)

Этот метод также принято называть аутентификацией, выполняемой базой данных Oracle, в которой каждый пользователь создается вместе с паролем. Этот пароль должен указываться пользователем при попытке установления соединения. Когда администратор назначает пароль, он должен сразу же установить для него истечение срока годности (expire). Это заставит пользователя изменить пароль при первом соединении. Однако при этом необходимо позаботиться о том, чтобы пользователь имел возможность изменить пароль. Некоторые приложения не позволяют это сделать.

Пароль всегда автоматически и прозрачно шифруется при установлении сетевого соединения (в архитектуре клиент/сервер и сервер/сервер). Для шифрования пароля перед передачей его по сети применяется модифицированный алгоритм DES (Data Encryption Standard).

### - External (метод внешней аутентификации)

Этот метод также принято называть аутентификацией, выполняемой на уровне операционной системы. При использовании этого метода пользователи могут соединяться с Oracle более просто, без указания имени и пароля. База данных полагается на операционную

систему или на службу сетевой аутентификации, когда ограничивает доступ к учетным записям пользователей БД. Пароль базы данных не используется при этом виде установления соединения. Этот метод аутентификации может использоваться, если это позволяет сделать операционная система или сетевая служба. Когда такое возможно, установите в параметре инициализации OS\_AUTHENT\_PREFIX значение префикса, используемого в именах пользователей Oracle. Этот префикс Oracle подставляет к началу наименования учетной записи каждого пользователя операционной системы. По умолчанию значение параметра – OPS\$, что обеспечивает совместимость с предыдущими версиями сервера Oracle. Когда пользователь ОС пытается подсоединиться, Oracle соединяет префикс с именем пользователя в ОС, а затем сопоставляет полученную строку с именами пользователей в базе данных.

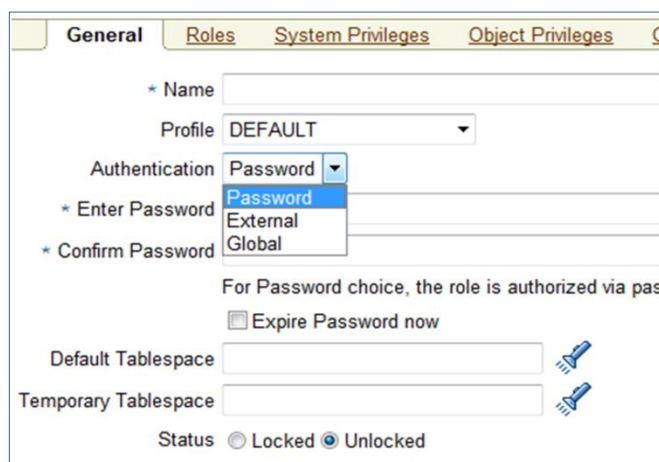
Например, предположим, что значение параметра следующее: OS\_AUTHENT\_PREFIX=OPS\$.

Когда пользователь операционной системы с именем tsmith подсоединяется к базе данных Oracle и аутентифицируется на уровне ОС, Oracle проверяет, что в базе данных есть соответствующий пользователь OPS\$tsmith и, если это так, разрешает соединение пользователя. При всех ссылках на пользователя, аутентифицируемого на уровне операционной системы, необходимо указывать префикс (OPS\$tsmith) .

**Примечание:** текстовое значение параметра OS\_AUTHENT\_PREFIX в некоторых ОС зависит от регистра символов. Дополнительную информацию об этом параметре см. в документации Oracle, относящейся к конкретной ОС.

#### - Global (метод глобальной аутентификации)

Это строгая аутентификация, проводимая с помощью опции Oracle Advanced Security. Глобальная аутентификация позволяет идентифицировать пользователей биометрически, на основе сертификатов x509, маркерных устройств (token devices) и с помощью Oracle Internet Directory.



#### Аутентификация администраторов.

Безопасность на уровне операционной системы

- АБД должны иметь привилегии на уровне ОС для создания и удаления файлов.
- Обычным пользователям базы данных не следует давать привилегии на уровне ОС для создания и удаления файлов.

В UNIX и Linux администраторы базы данных по умолчанию входят на уровне ОС в группу install. Таким образом, им предоставляются привилегии, необходимые для создания и удаления файлов БД.

#### Администраторы и обеспечение безопасности.

Авторизация при подсоединении с привилегией SYSDBA или SYSOPER осуществляется с помощью парольного файла или ОС.

Парольный файл содержит записи с именами пользователей, являющимися администраторами, и используется для их аутентификации.

Для аутентификации на уровне ОС не используются записи об определенных пользователях.

Аутентификация пользователей с привилегией SYSDBA или SYSOPER на уровне ОС имеет более высокий приоритет по сравнению с аутентификацией на основе парольного файла.

Авторизация при подсоединении с привилегией SYSDBA или SYSOPER осуществляется с помощью парольного файла или привилегий и прав доступа на уровне операционной системы. При аутентификации на уровне ОС база данных *не* проверяет введенное имя пользователя и пароль. Аутентификация на уровне ОС применяется, когда отсутствует парольный файл, когда введенное имя пользователя и пароль не находятся в этом файле, а также когда не предоставляются имя пользователя и пароль.

Когда аутентификация с помощью парольного файла завершается успешно, тогда вход в систему производится на основе имени пользователя. Если же успешная аутентификация происходит на основе средств ОС, тогда регистрация соединения вида CONNECT не соотносится с определенным пользователем.

**Примечание.** Аутентификация на уровне ОС имеет более высокий приоритет по сравнению с аутентификацией на основе парольного файла. Поэтому, если вы подсоединены в операционной системе как пользователь, входящий в группу OSDBA или OSOPER (ora\_dba для Windows-систем), тогда установление соединения с привилегией SYSDBA или SYSOPER, осуществляется без учета введенного имени и пароля пользователя.

## Разблокирование учетной записи пользователя и переустановка пароля

В ходе инсталляции сервера, включающей создание БД, или при отдельном создании базы данных можно разблокировать и перенастроить учетные записи многих пользователей Oracle, используемых для поддержки функциональных возможностей. Если это не было сделано при создании БД, разблокировать пользователя и переустановить пароль можно на странице Users. Выберите для этого нужного пользователя и выполните операцию Unlock User.

Если вы находитесь на странице Edit Users, тогда такие действия производятся следующим образом:

1. Укажите новый пароль в полях Enter Password и Confirm Password.
2. Выберите статус Unlocked.
3. Щелкните на кнопке Apply, чтобы переустановить пароль пользователя и разблокировать его учетную запись.

Select	UserName	Account	Expire	Default Tablespace	Temporary Tablespace	Profile	Created	User Type
<input type="radio"/>	ANONYMOUS	LOCKED	56:52 PM	SYSAUX	TEMP	DEFAULT	Oct 30, 2011 1:15:43 PM PDT	LOCAL
<input type="radio"/>	APEX_030200	LOCKED	56:52 PM	SYSAUX	TEMP	DEFAULT	Oct 30, 2011 1:37:46 PM PDT	LOCAL
<input type="radio"/>	APEX_PUBLIC_USER	EXPIRED & LOCKED	Oct 30, 2011 1:56:52 PM PDT	USERS	TEMP	DEFAULT	Oct 30, 2011 1:37:46 PM PDT	LOCAL
<input type="radio"/>	APPQOSSYS	EXPIRED & LOCKED	Oct 30, 2011 1:05:54 PM PDT	SYSAUX	TEMP	DEFAULT	Oct 30, 2011 1:05:54 PM PDT	LOCAL
<input type="radio"/>	BI	EXPIRED & LOCKED	Jul 6, 2018 7:17:40 AM PDT	USERS	TEMP	DEFAULT	Jul 6, 2018 7:15:22 AM PDT	LOCAL
<input type="radio"/>	CTXSYS	EXPIRED & LOCKED	Oct 30, 2011 1:56:52 PM PDT	SYSAUX	TEMP	DEFAULT	Oct 30, 2011 1:15:00 PM PDT	LOCAL
<input type="radio"/>	DBSNMP	OPEN	Jul 29, 2019 7:12:29 AM PDT	SYSAUX	TEMP	MONITORING_PROFILE	Oct 30, 2011 1:05:52 PM PDT	LOCAL
<input type="radio"/>	DIP	EXPIRED & LOCKED	Oct 30, 2011 12:58:20 PM PDT	USERS	TEMP	DEFAULT	Oct 30, 2011 12:58:20 PM PDT	LOCAL
<input type="radio"/>	EYESYS	EXPIRED &	Oct 30, 2011 1:14:44 PM	SYSAUX	TEMP	DEFAULT	Oct 30, 2011 1:14:44 PM	LOCAL

Блокировка и разблокировка учетной записи, SQL:

```
ALTER USER username ACCOUNT LOCK;
```

```
ALTER USER username ACCOUNT UNLOCK;
```

У каждой учетной записи есть свой статус. Узнать его можно выполнив запрос:

```
SELECT USERNAME, ACCOUNT_STATUS FROM DBA_USERS;
```

Статусы:

- OPEN – учетная запись доступна для использования
- LOCKED – учетная запись заблокирована DBA. Пользователь не может подключаться к базе данных
- EXPIRED – истекло время действия. Пароль может иметь срок действия. Пользователь, у которого истекло время действия пароля, не может подключиться к базе данных пока пароль не будет сброшен
- EXPIRED & LOCKED – учетная запись не только заблокирована, но и истекло время действия пароля
- EXPIRED (GRACE) – сигнализирует о действии дополнительного времени на действие пароля. В этом случае действие пароля не истекает сразу же, а дается время на его смену
- LOCKED (TIMED) – учетная запись заблокирована после нескольких неудачных попыток подключиться к базе данных. Учетная запись может быть настроена на блокирование после указанного числа неудачных попыток авторизации
- EXPIRED & LOCKED (TIMED)
- EXPIRED (GRACE) & LOCKED
- EXPIRED (GRACE) & LOCKED (TIMED)

### Удаление пользователя из базы данных

Для удаления учетной записи из базы данных используется предложение DROP USER, опционально можно указать опцию CASCADE, что позволит удалить все объекты принадлежащие пользователю:

```
DROP USER ALL_ORACLE CASCADE;
```

Удаление пользователя неявно удаляет все объекты, принадлежащие ему, но не роли или привилегии.

### Привилегии

Привилегия – это право на выполнение конкретной команды SQL или право доступа к объектам других пользователей. Oracle предоставляет возможность дифференцированного контроля разрешенных и запрещенных операций пользователя в базе данных. Привилегии делятся на две категории: системные и объектные.

**Системные привилегии** (system privileges). Каждая системная привилегия, предоставленная пользователю, позволяет ему выполнять в базе данных конкретные операции или классы операций. Например, создание табличных пространств – это системная привилегия. Системные привилегии могут быть предоставлены администратором или кем-то, кому явно предоставлены права по сопровождению привилегий. Имеется более 100 системных привилегий. Многие системные привилегии содержат фразу ANY.

**Объектные привилегии** (object privileges). Каждая объектная привилегия, выданная пользователю, позволяет ему выполнять конкретные действия над определенным объектом (например, таблицей, представлением, последовательностью, процедурой, функцией или пакетом). Без специального разрешения пользователи имеют доступ только к своим собственным объектам. Объектные привилегии могут быть предоставлены владельцем объекта либо кем-то, кому явно предоставлено право выдавать привилегии на объект.



## Системные привилегии

Чтобы предоставить системные привилегии, щелкните на ссылке Systems Privileges, расположенной на странице Edit User. Выберите необходимые привилегии из списка доступных привилегий и переместите их, щелкнув на стрелке Move, в окно списка Selected System Privileges.

Предоставление привилегии с фразой ANY означает, что действие привилегии не ограничивается схемой пользователя. Например, привилегия CREATE TABLE позволяет пользователю создавать таблицы, но только в своей собственной схеме. Привилегия SELECT ANY TABLE разрешает пользователю делать запросы к таблицам, принадлежащим другим пользователям.

Отметка, сделанная в поле Admin Option, дает право этому пользователю быть администратором привилегии и выдавать ее другим пользователям.

**Edit User: HR**

Actions: Create Like [Go] Show SQL Revert Apply

General Roles **System Privileges** Object Privileges Quotas Consumer Group Privileges Proxy Users

[Edit List]

System Privilege	Admin Option
ALTER SESSION	<input type="checkbox"/>
CREATE DATABASE LINK	<input type="checkbox"/>
CREATE SEQUENCE	<input type="checkbox"/>
CREATE SESSION	<input type="checkbox"/>
CREATE SYNONYM	<input type="checkbox"/>
CREATE VIEW	<input type="checkbox"/>
UNLIMITED TABLESPACE	<input type="checkbox"/>

**Modify System Privileges**

Available System Privileges: ACCESS\_ANY\_WORKSPACE, ADMINISTER\_ANY\_SQL\_TUNING\_SET, ADMINISTER\_DATABASE\_TRIGGER, ADMINISTER\_RESOURCE\_MANAGER, ADMINISTER\_SQL\_MANAGEMENT\_OBJECT, ADMINISTER\_SQL\_TUNING\_SET, ADVISOR, ALTER\_ANY\_ASSEMBLY, ALTER\_ANY\_CLUSTER, ALTER\_ANY\_CUBE

Selected System Privileges: ALTER SESSION, CREATE DATABASE LINK, CREATE SEQUENCE, CREATE SESSION, CREATE SYNONYM, CREATE VIEW, UNLIMITED TABLESPACE

Buttons: Move, Move All, Remove, Remove All, Cancel, OK

## Привилегии SYSDBA и SYSOPER

Эти привилегии разрешают останавливать и запускать экземпляр базы данных, выполнять восстановление, а также решать другие задачи по сопровождению базы данных.

Привилегия SYSOPER позволяет выполнять основные операционные задачи, но без возможности просмотра данных пользователей. Эта привилегия содержит следующие системные привилегии:

- STARTUP и SHUTDOWN
- CREATE SPFILE
- ALTER DATABASE OPEN/MOUNT/BACKUP
- ALTER DATABASE ARCHIVELOG
- ALTER DATABASE RECOVER (только полное восстановление; для неполного восстановления, например, UNTIL TIME | CHANGE | CANCEL | CONTROLFILE требуется подключение с привилегией SYSDBA)
- RESTRICTED SESSION

Системная привилегия SYSDBA дает право выполнять неполное восстановление и удалять базу данных. Пользователь, использующий системную привилегию SYSDBA при подключении, устанавливает соединение как пользователь SYS.



### Примеры системных привилегий:

- `DROP ANY <объект>;` такие привилегии разрешают пользователям удалять объекты, находящиеся в схемах, принадлежащих другим пользователям.

- `CREATE, MANAGE, DROP, ALTER TABLESPACE;` эти привилегии позволяют производить действия по администрированию табличных пространств, включая создание, удаление и изменение их атрибутов.

- `CREATE ANY DIRECTORY;` база данных Oracle предоставляет возможность разработчикам вызывать из PL/SQL внешний код (например, программу из библиотеки C). Средством обеспечения безопасности служит объект типа `DIRECTORY`, представляющий собой виртуальный каталог, с которым должна быть связана директория операционной системы, в которой находится код. Имея привилегию `CREATE ANY DIRECTORY`, пользователь потенциально может вызвать нарушающий безопасность кодовый объект. Привилегия `CREATE ANY DIRECTORY` дает пользователю право создавать объект `DIRECTORY` (с правами доступа `read` и `write`) для любого каталога, к которому может обратиться владелец программного обеспечения Oracle. Это означает, что пользователь может вызвать внешние процедуры из таких директорий. Кроме того, пользователь может попытаться напрямую читать или писать в файл базы данных, например, в оперативный журнал или файлы аудита. Стратегия безопасности, принятая в организации, должна препятствовать неправильному применению подобных привилегий, которые предоставляют такие мощные возможности.

- `GRANT ANY OBJECT PRIVILEGE;` эта привилегия позволяет пользователю предоставлять разрешения на доступ к объектам, которые ему не принадлежат.

- `ALTER DATABASE` и `ALTER SYSTEM;` это очень мощные по предоставляемым возможностям привилегии, позволяющие изменять базу данных и экземпляр Oracle, например, переименовывать файл данных и очищать (`flush`) кэш буферов.

- `ALTER DATABASE;` привилегия на изменение базы данных, например перевод из состояния `MOUNT` в состояние `OPEN` или восстановление базы данных.

- `ALTER SYSTEM;` привилегия управления журнальными файлами (перейти к следующему журналу `redo log group`) и изменения параметров инициализации системы в `SPFILE`.

- `AUDIT SYSTEM;` привилегия на управление операторами аудита (`AUDIT statements`).

- `CREATE DATABASE LINK;` привилегия на создание ссылок на удаление базы данных.

- `CREATE ANY INDEX;` привилегия на создание индекса в схеме любого пользователя; привилегия `CREATE INDEX` предоставляется вместе с `CREATE TABLE` для схемы пользователя.

- `CREATE PROFILE;` привилегия на создание профиля (параметры пароля и использования ресурсов).

- `CREATE PROCEDURE;` привилегия на создание функций, процедур и пакетов в своей схеме.

- `CREATE ANY PROCEDURE;` привилегия на создание функций, процедур и пакетов в схеме любого пользователя.

- `CREATE SESSION;` привилегия на установление соединения с базой данных.

- `CREATE SYNONYM;` привилегия на создание синонима в своей схеме.

- `CREATE ANY SYNONYM;` привилегия на создание синонима в схеме любого пользователя.

- `CREATE PUBLIC SYNONYM;` привилегия на создание синонима, которым могут пользоваться все пользователи.

- `DROP ANY SYNONYM;` привилегия на удаление синонима в схеме любого пользователя.

- `DROP PUBLIC SYNONYM;` привилегия на удаление синонима, которым могут пользоваться все пользователи.

- `CREATE TABLE;` привилегия на создание таблицы в своей схеме.



- `CREATE ANY TABLE`; привилегия на создание таблицы в схеме любого пользователя..
- `CREATE TABLESPACE`; привилегия на создание табличного пространства в базе данных.
- `CREATE USER`; привилегия на создание учетной записи пользователя и соответствующей схемы.
- `ALTER USER`; привилегия на изменение учетной записи пользователя и соответствующей схемы.
- `CREATE VIEW`; привилегия на создание представления в своей схеме.
- `SYSDBA` и `SYSOPER`; эти привилегии разрешают останавливать и запускать экземпляр базы данных, выполнять восстановление, а также решать другие задачи по сопровождению базы данных. Привилегия `SYSOPER` позволяет выполнять основные операционные задачи, но без возможности просмотра данных пользователей.

### Пример назначения системных привилегий командой SQL:

```
GRANT CREATE SESSION TO emi;
GRANT CREATE SESSION TO emi WITH ADMIN OPTION;
```

### В общем виде

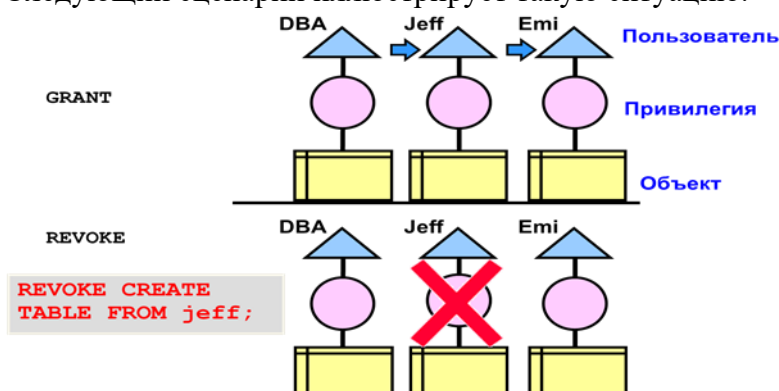
```
GRANT privilege [,privilege] TO user [,user, role, PUBLIC] [WITH ADMIN OPTION];
```

### Отмена системных привилегий

Системные привилегии, которые были предоставлены по команде `GRANT`, могут быть отобраны командой `REVOKE`. Пользователи, получившие системную привилегию с параметром `WITH ADMIN OPTION`, могут отобрать привилегию у любого другого пользователя базы данных. Тот, кто отбирает привилегию, необязательно должен быть тем, кто эту привилегию первоначально выдал.

Отмена системной привилегии не приводит к каскадным отменам, даже если эта привилегия предоставлялась с помощью параметра `WITH ADMIN OPTION`.

Следующий сценарий иллюстрирует такую ситуацию.



### Сценарий

1. Администратор базы данных (DBA) предоставляет пользователю Jeff системную привилегию `CREATE TABLE` с параметром `WITH ADMIN OPTION`.
2. Пользователь Jeff создает таблицу.
3. Пользователь Jeff предоставляет системную привилегию `CREATE TABLE` пользователю Emi.
4. Пользователь Emi создает таблицу.
5. Администратор базы данных (DBA) отбирает у пользователя Jeff системную привилегию `CREATE TABLE`.

### Результат:

- Таблица пользователя Jeff еще существует, однако этот пользователь больше не может создавать новые таблицы.
- У пользователя Emi осталась своя таблица и системная привилегия CREATE TABLE.

### Пример удаления системных привилегий командой SQL:

```
REVOKE CREATE SESSION TO emi;
```

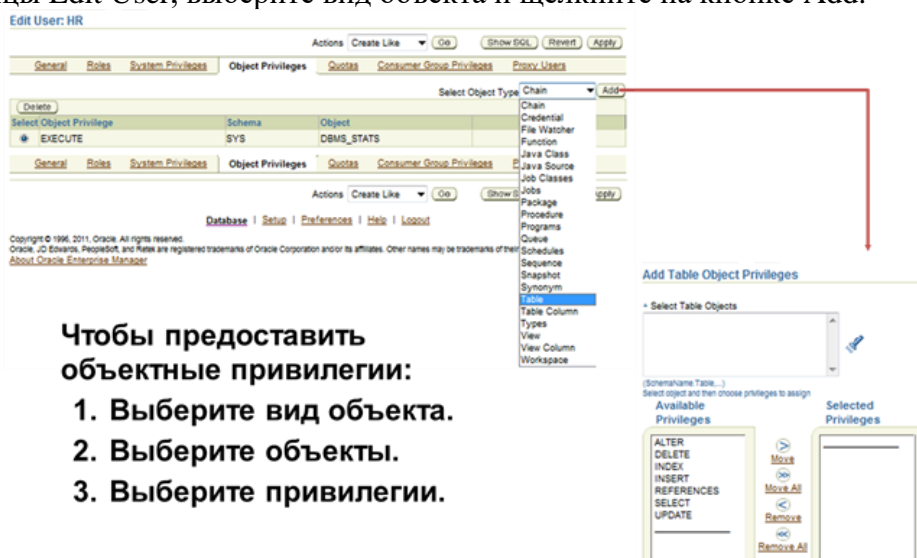
### Объектные привилегии

Объектная привилегия – это привилегия или право на выполнение определенного действия над отдельной таблицей, последовательностью, процедурой или отдельным представлением.

Для разных типов объектов схемы доступны разные объектные привилегии. Пользователь автоматически обладает всеми объектными привилегиями на объекты схемы, содержащиеся в схеме пользователя. Пользователь может предоставить любую объектную привилегию на любой объект схемы, которым он владеет, любому другому пользователю или любой роли.

Объектная привилегия	Таблица	Представление	Последовательность
ALTER	✓		✓
DELETE	✓	✓	
INDEX	✓		
INSERT	✓	✓	
REFERENCES	✓		
SELECT	✓	✓	✓
UPDATE	✓	✓	

Чтобы предоставить объектные привилегии, щелкните на закладке Object Privileges страницы Edit User, выберите вид объекта и щелкните на кнопке Add.



**Чтобы предоставить объектные привилегии:**

1. Выберите вид объекта.
2. Выберите объекты.
3. Выберите привилегии.

Выберите объекты, к которым вы хотите дать привилегии доступа. Для этого либо введите имя *пользователя.имя\_объекта*, либо выберите объекты из списка.

Затем выберите необходимые привилегии в окне списка Available Privileges и щелкните на стрелке Move. После того, как будет закончен выбор привилегий, щелкните на кнопке OK.

Происходит переход обратно на страницу Edit User. Если это необходимо, отметьте поле Grant Option в списке объектных привилегий для того, чтобы разрешить пользователю предоставлять другим пользователям доступ к этому объекту.

### Примеры объектных привилегий:

- ALTER; можно изменять определение таблицы или последовательности.
- DELETE; можно удалять строки из таблицы, представления или материализованного представления.
- EXECUTE; можно выполнять функцию или процедуру, в том числе из пакета.
- DEBUG; разрешено просматривать код PL / SQL в триггерах, определенных в таблице, или операторе SQL, который ссылается на таблицу. Для типов объектов это привилегия позволяет получить доступ ко всем публичным и частным переменным, методам и типам, определенным в типе объекта.
  - FLASHBACK; разрешает ретроспективные запросы к таблицам, представлениям и материализованным представлениям, использует сохраненную информацию об отмене.
  - INDEX; можно создать индекс для таблицы.
  - INSERT; можно вставлять строки в таблицу, представление или материализованное представление.
  - ON COMMIT REFRESH; можно создавать материализованное представление с обновлением при фиксации на основе таблицы.
  - QUERY REWRITE; можно создать материализованное представление для перезаписи запроса на основе таблицы.
  - READ; можно читать содержимое каталога операционной системы, используя словарь Oracle DIRECTORY.
  - REFERENCES; можно создавать ограничение внешнего ключа, которое ссылается на первичный ключ или уникальный ключ другой таблицы.
  - SELECT; можно читать строки из таблицы, представления или материализованного представления в дополнение к чтению текущих или следующих значений из последовательности.
  - UNDER; можно создать представление на основе существующего представления.
  - UPDATE; можно обновлять строки в таблице, представлении или материализованном представлении.
  - WRITE; можно записывать информацию в каталог операционной системы, используя словарь Oracle DIRECTORY.

### Пример назначения объектных привилегий командой SQL:

```
GRANT select,insert ON employees TO emi;  
GRANT update (deptmnt_name, location_id) ON deptmnts TO scott,  
manager;
```

В следующем примере администратор баз данных предоставляет пользователю SCOTT полный доступ к таблице HR.EMPLOYEES, но позволяет пользователю SCOTT передавать другим пользователям только объектные привилегии SELECT.

```
SQL> grant insert, update, delete on hr.employees to scott;  
grant select on hr.employees to scott with grant option;
```

### В общем виде

```
GRANT object_priv [(columns)] ON object TO {user|role|PUBLIC} [WITH  
ADMIN OPTION];
```

В следующем примере администратор баз данных предоставляет пользователю SCOTT доступ на изменение отдельных столбцов таблицы HR.EMPLOYEES:

```
SQL> grant update (employee_id, first_name, last_name, email,  
    phone_number, hire_date, job_id, commission_pct,  
    manager_id, department_id)  
    on hr.employees to scott;
```

Пользователь SCOTT сможет обновить все столбцы в таблице HR.EMPLOYEES, кроме столбца SALARY:

```
SQL> update hr.employees set first_name = 'Stephen' where  
employee_id = 100;
```

1 row updated.

```
SQL> update hr.employees set salary = 50000 where employee_id =  
203;
```

```
update hr.employees set salary = 50000 where employee_id = 203
```

\*

ERROR at line 1:

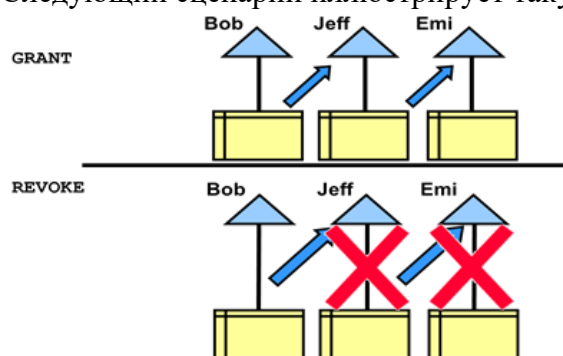
ORA-01031: insufficient privileges

### Отмена объектных привилегий

Каскадный эффект наблюдается, когда отменяется системная привилегия, от которой зависит операция DML. Например, привилегия SELECT ANY TABLE была предоставлена пользователю, и затем пользователь создал процедуры, которые используют таблицы из других схем. После отмены привилегии все процедуры в схеме пользователя должны быть повторно перекомпилированы перед их использованием.

Если объектные привилегии предоставлялись с параметром WITH GRANT OPTION, то отмена этих привилегий будет каскадной.

Следующий сценарий иллюстрирует такую ситуацию.



### Сценарий

1. Пользователю Jeff объектная привилегия SELECT была предоставлена с параметром WITH GRANT OPTION.
2. Пользователь Jeff предоставил пользователю Emi привилегию SELECT на таблицу EMPLOYEES.
3. Если затем привилегия SELECT отбирается у пользователя Jeff, то эта привилегия также каскадно отбирается у пользователя Emi.

### Пример удаления объектных привилегий командой SQL:

```
REVOKE select,insert ON employees FROM emi;
```

### В общем виде

```
REVOKE {object_priv [,object_priv]|ALL} ON object TO  
{user|role|PUBLIC} [CASCADE CONSTRAINTS];
```

### Роли

Роли – это именованные группы связанных привилегий, которые предоставляются пользователям или другим ролям. Они разработаны для облегчения сопровождений привилегий в базе данных и улучшают безопасность.

В большинстве систем выделение по отдельности необходимых пользователю привилегий может занять слишком много времени и вызвать с высокой вероятностью появление ошибок. Oracle предоставляет возможность простого и контролируемого сопровождения привилегий с помощью ролей.

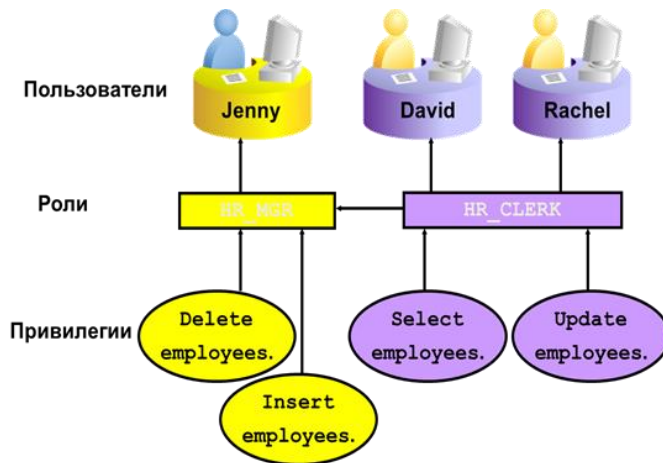
#### Преимущества ролей:

- **Упрощение сопровождения привилегий.** Роли используются для облегчения сопровождения привилегий. Вместо того, чтобы повторно предоставлять один и тот же набор привилегий нескольким пользователям, можно выдать эти привилегии роли, а затем выдать эту роль каждому пользователю.
- **Динамическое сопровождение привилегий.** Если привилегии, связанные с ролью, изменяются, то на всех пользователей, которым предоставлена данная роль, автоматически и немедленно распространяются эти изменения.
- **Избирательная доступность привилегий.** Для временного предоставления или отмены привилегий можно включать или выключать роли. Включение роли также используется для проверки, предоставлена ли пользователю данная роль.

#### Характеристики ролей

- Привилегии выдаются и отбираются у ролей с помощью тех же команд, которые используются для выдачи и отмены привилегий пользователю.
- Роли подобно системным привилегиям могут быть выданы и отобраны у любого пользователя или другой роли.
- Могут состоять как из системных, так и из объектных привилегий.
- Могут включаться или выключаться для любого пользователя, которому эта роль предоставлена.
- Могут требовать пароль для включения.
- Роли никому не принадлежат; не находятся ни в какой схеме.

В примере на рисунке роль HR\_CLERK содержит объектные привилегии SELECT и UPDATE для доступа и изменения данных в таблице employees. Роль HR\_MGR содержит привилегии DELETE и INSERT, позволяющие удалять и вставлять данные в таблицу employees, а также роль HR\_CLERK. Поэтому менеджер, которому выделена роль HR\_MGR может выбирать, удалять, вставлять и изменять данные в таблице employees.



## Предопределенные роли

Существует несколько ролей, определяемых автоматически в процессе выполнения скрипта создания базы данных Oracle. Роль CONNECT выделяется автоматически любому пользователю, создаваемому с помощью Enterprise Manager. В предыдущих версиях БД (до Oracle Database 10g версии 2) роль CONNECT включала больше привилегий, например, CREATE TABLE и CREATE DATABASE LINK, которые были удалены по соображениям безопасности.

**Примечание:** следует иметь в виду, что при выделении роли RESOURCE одновременно предоставляется системная привилегия UNLIMITED TABLESPACE.

### Примеры:

CONNECT	CREATE SESSION
RESOURCE	CREATE CLUSTER, CREATE INDEXTYPE, CREATE OPERATOR, CREATE PROCEDURE, CREATE SEQUENCE, CREATE TABLE, CREATE TRIGGER, CREATE TYPE
SCHEDULER_ ADMIN	CREATE ANY JOB, CREATE EXTERNAL JOB, CREATE JOB, EXECUTE ANY CLASS, EXECUTE ANY PROGRAM, MANAGE SCHEDULER
DBA	Большинство системных привилегий, некоторые другие роли. Не предоставляйте пользователям, не являющимся АБД.
SELECT_CATALOG_ROLE	Ни одной системной привилегии, но роль HS_ADMIN_ROLE и более 1700 объектных привилегий доступа к словарию данных.

- CONNECT содержит только одну привилегию - CREATE SESSION
- RESOURCE содержит привилегии CREATE CLUSTER, CREATE INDEXTYPE, CREATE OPERATOR, CREATE PROCEDURE, CREATE SEQUENCE, CREATE TABLE, CREATE TRIGGER, CREATE TYPE. Эти привилегии обычно используются разработчиками приложений, которые могут программируют процедуры и функции PL / SQL.
- DBA содержит все системные привилегии WITH ADMIN OPTION. Позволяет администратору базы данных предоставлять системные привилегии другим пользователям.
- DELETE\_CATALOG\_ROLE не содержит никаких системных привилегий, а только объектные привилегии (DELETE) для SYS.AUD\$ и FGA\_LOG\$. Другими словами, эта роль позволяет пользователю удалять записи аудита из журнала аудита (регулярного или детального).
- EXECUTE\_CATALOG\_ROLE позволяет выполнять процедуры из системных пакетов, процедур и функций, таких как DBMS\_FGA и DBMS\_RLS.
- SELECT\_CATALOG\_ROLE содержит объектные привилегии на чтение системных таблиц словаря данных (более 2400 таблиц).

- EXP\_FULL\_DATABASE содержит в себе роли EXECUTE\_CATALOG\_ROLE, SELECT\_CATALOG\_ROLE, а также системные привилегии такие как BACKUP ANY TABLE и RESUMABLE. Позволяет пользователю с этой ролью экспортировать все объекты базы данных.
- IMP\_FULL\_DATABASE содержит все необходимые привилегии для импорта ранее экспортированных объектов базы данных.
- RECOVERY\_CATALOG\_OWNER используется для создания пользователя, владеющего каталогом восстановления для резервного копирования и восстановления RMAN.
- SCHEDULER\_ADMIN предоставляет доступ к пакету DBMS\_SCHEDULER вместе с привилегиями для создания пакетных заданий.

Роли CONNECT, RESOURCE и DBA предоставляются в основном для совместимости с предыдущими версиями Oracle; они могут не существовать в будущих версиях Oracle. Администратор базы данных должен создавать собственные роли, используя права, предоставленные этим ролям, в качестве отправной точки.

Системные представления, в которых хранится информация о ролях, предоставленных пользователям:

Представление словаря данных	Описание
DBA_TAB_PRIVS	Объектные привилегии по доступу к таблицам, предоставленные ролям и пользователям. Включает пользователя, который предоставил привилегию роли или пользователю с использованием или без использования GRANT OPTION
DBA_COL_PRIVS	Объектные привилегии по доступу к столбцам, предоставленные ролям или пользователям, содержащие имя столбца и тип привилегии для столбца
SESSION_PRIVS	Все системные привилегии, действующие для указанного пользователя в сеансе, предоставленные напрямую или через роль
ROLE_TAB_PRIVS	Привилегии, предоставленные таблицам через роли, для текущего сеанса

### Роли для поддержки дополнительных функций

Другие роли позволяют выполнять администрирование специальных возможностей, если они установлены. Например, роль XDBADMIN содержит привилегии, необходимые для сопровождения установленной базы данных XML (Extensible Markup Language). Роль AQ\_ADMINISTRATOR\_ROLE предоставляет привилегии для администрирования *функциональных возможностей обработки сообщений (advanced queuing)*. Роль HS\_ADMIN\_ROLE содержит привилегии для администрирования гетерогенных служб. Не следует вносить изменения в привилегии этих ролей без консультации со службой технической поддержки Oracle, поскольку вы можете непреднамеренно отключить необходимые функциональные возможности.

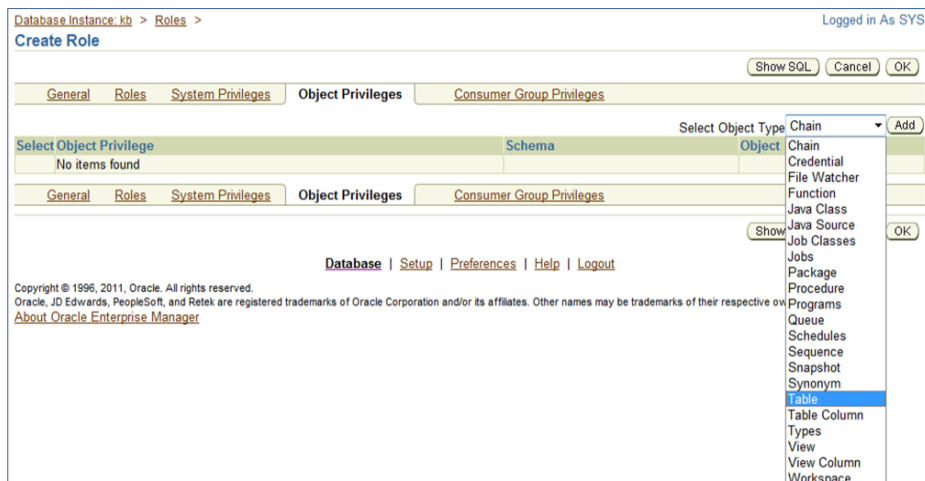
### Создание роли

Для создания роли выполните следующие шаги:

1. В Enterprise Manager Database Control выберите Server > Security > Roles.
2. Щелкните на кнопке Create.
3. Введите имя роли.



4. На вкладках назначьте для выбранной роли системные и объектные привилегии.



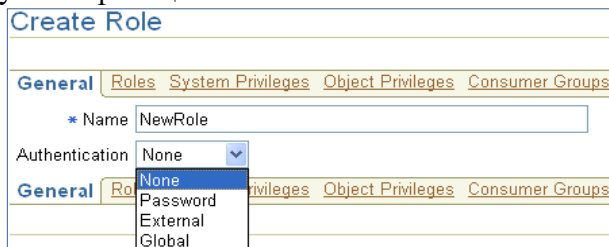
### Пример SQL-команды

```
CREATE ROLE hr_clerk IDENTIFIED BY pa$$w0rd;
```

### Защита ролей

Роли обычно включены по умолчанию. Это означает, что, когда роль предоставлена пользователю, он может воспользоваться привилегиями этой роли. Возможно и следующее:

- *Сделать роль не по умолчанию.* При выделении роли пользователю уберите отметку в поле DEFAULT. Теперь пользователь должен явно включить роль перед тем, как станут доступны привилегии роли.
- *Потребовать дополнительную аутентификацию роли.* По умолчанию дополнительная аутентификация отключена (NONE), однако можно сделать так, чтобы перед динамическим включением роли обязательно производилась дополнительная аутентификация.



- *Создать защищаемую роль приложения (secure application role), безопасное включение которой возможно только в результате успешного выполнения процедуры на PL/SQL.* В этой процедуре можно, например, проверить сетевой адрес пользователя, имя программы, выполняемой пользователем, время дня или что-либо другое для обеспечения должной безопасности при выделении группы полномочий.

### Пример:

```
CREATE ROLE защищаемая_роль_приложения IDENTIFIED USING  
<имя_процедуры_защиты>;
```

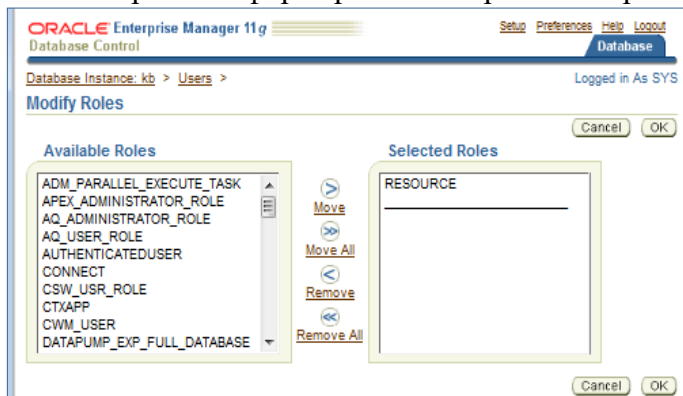
### Предоставление ролей пользователям

Роль – набор привилегий, который может быть предоставлен пользователям и другим ролям. Роли используются для администрирования привилегий базы данных. Привилегии можно добавить в роль, а затем выделить роль пользователю. После этого пользователь может включить роль и воспользоваться привилегиями, входящими в роль. Роль содержит все привилегии, предоставленные этой роли, а также все привилегии других ролей, входящих в эту роль.

По умолчанию Enterprise Manager автоматически предоставляет новому пользователю роль CONNECT. Это позволяет пользователям подключаться к базе данных и после получения других привилегий создавать объекты базы данных в своей схеме.

Для того, чтобы выделить роль пользователю, выполните следующие шаги:

1. В Enterprise Manager Database Control выберите Server > Security > Users.
2. Выберите пользователя, а затем щелкните на кнопке Edit.
3. Щелкните на закладке Roles, а затем на кнопке Edit List.
4. Выберите необходимую роль в списке Available Roles и переместите ее в перечень Selected Roles.
5. После завершения формирования перечня выбранных ролей щелкните на кнопке ОК.



### Пример:

```
GRANT oe_clerk TO scott WITH ADMIN OPTION;
REVOKE oe_clerk FROM scott;
```

Создайте роль HRCLERK с привилегиями SELECT и UPDATE на все таблицы схемы HR.  
Создайте роль HRMANAGER с привилегиями INSERT и DELETE на все таблицы схемы HR.  
Выдайте роль HRCLERK роли HRMANAGER.

Создайте средствами ЕМ пользователя DHAMBY и назначьте ему роли согласно таблице. При создании сохраните SQL-команду в файле script.sql.

имя	Учетная запись	описание	параметры	роли
David Hamby	DHAMBY	A new HR Clerk	Authentication: Password Password: newuser Profile: HRPROFILE Default Tablespace: USERS Temporary Tablespace: TEMP +установите параметр: Password expired	Connect HRCLERK

Отредактируйте script.sql для создания пользователей RPANDYA и JGOODMAN согласно приведенному описанию. Выполните скрипт(ы) средствами SQL Plus или SQL Developer.

имя	Учетная запись	описание	параметры	роли
Rachel Pandya	RPANDYA	A new HR Clerk	Authentication: Password Password: newuser Profile: HRPROFILE Default Tablespace: USERS Temporary Tablespace: TEMP +установите параметр: Password expired	Connect HRCLERK

имя	Учетная запись	описание	параметры	роли
Jenny Goodman	JGOODMAN	A new HR Manager	Authentication: Password Password: newuser Profile: HRPROFILE Default Tablespace: USERS Temporary Tablespace: TEMP +установите параметр: Password expired	Connect HRMANAGER

Протестируйте новых пользователей в SQL Plus. Подключитесь в БД как DHAMBY. Установите новый пароль kb2023 (команда password). Выберите строку с EMPLOYEE\_ID=197 из таблицы HR.EMPLOYEES. Затем попробуйте удалить ее. Должна появиться ошибка “insufficient privileges”.

Повторите тест для пользователя JGOODMAN. Установите новый пароль kb2023. После удаления строки выполните ее восстановление командой rollback.

В SQL Plus подключитесь как пользователь RPANDYA. Измените пароль на kb2023. Оставьте это подключение до конца занятия. HRPROFILE ограничивает время неактивности сессий в 15 минут. Позднее убедитесь, что сессия была отключена.

**Время на выполнение лабораторной работы – 2 часа.**