

Базы данных и  
Информационные системы  
14/15 Иерархические запросы. Вопросы  
администрирования БД.

Кузиков Б.О.  
Сумы, СумГУ  
2013



# Содержание

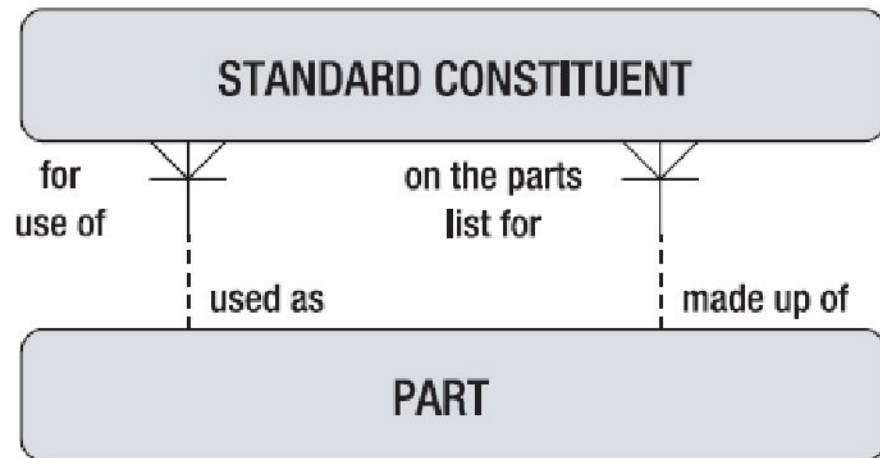
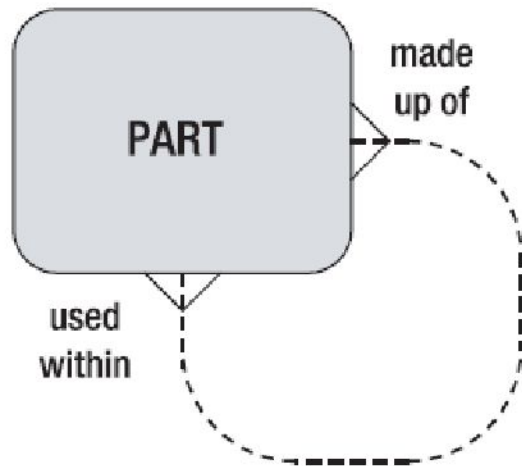
---

▶ После завершения занятия вы должны уметь и знать следующее:

- ▶ Создавать иерархические запросы
- ▶ Использовать псевдостолбцы LEVEL, PRIOR
- ▶ Назначать права на различные объекты БД

# Иерархические запросы

---



# Простой иерархический запрос

---

```
SELECT select_list
FROM table_expression
[ WHERE ... ]
[ START WITH start_expression ]
CONNECT BY
{ PRIOR parent_expr = child_expr
| child_expr = PRIOR parent_expr }
[ ORDER SIBLINGS BY
column1 [ ASC | DESC ] [, column2 [ ASC | DESC ] ] ...
```

- ▶ CONNECT BY определяет *по какому критерию* соединять записи
- ▶ START WITH определяет *с какой записи* начинать цикл
- ▶ PRIOR предоставляет доступ к предыдущей записи
- ▶ SIBLINGS осуществлять сортировку в *пределах уровня*



# Пример

```
select ename, LEVEL
  from emp
START WITH mgr is null
CONNECT BY NOCYCLE PRIOR empno = mgr;
```

```
ENAME LEVEL
```

```
-----
```

```
KING      1
```

```
JONES     2
```

```
SCOTT     3
```

```
ADAMS     4
```

```
FORD      3
```

```
SMITH     4
```

```
...
```

```
14 rows selected.
```

## NO CYCLE

---

- ▶ Ключевое слово **NOCYCLE** в предложении **CONNECT BY** предотвращает бесконечные циклы в иерархических запросах
- ▶ В нашем примере сотрудники не образуют циклов, но осторожность не помешает

## LEVEL, CONNECT\_BY\_ISCYCLE и CONNECT\_BY\_ISLEAF

---

- ▶ **LEVEL** – псевдостолбец содержит уровень строки в иерархическом запросе
- ▶ **CONNECT\_BY\_ISCYCLE** – функция содержит 1 для всех строк которые связаны циклической связью (иначе 0).
- ▶ **CONNECT\_BY\_ISLEAF** – функция содержит 1 если строка является нижним элементом иерархии (листом) иначе - 0.

# Пример с Level

```
select lpad(' ',2*level-1)||ename as ename
from emp
start with mgr is null
connect by nocycle prior empno = mgr;
```

```
ENAME
-----
KING
      JONES
                SCOTT
                ADAMS
      FORD
                SMITH
      BLAKE
                ALLEN
                WARD
                MARTIN
                TURNER
                JONES
      CLARK
                MILLER
14 rows selected.
```



# CONNECT\_BY\_ROOT и SYS\_CONNECT\_BY\_PATH

---

- ▶ CONNECT\_BY\_ROOT – оператор позволяет сопоставить строке (независимо от ее уровня) корневой элемент иерархии
- ▶ SYS\_CONNECT\_BY\_PATH – функция позволяет получить полный путь от корня к текущему элементу

# Примеры

---

```
select ename
, connect_by_root ename as manager
, sys_connect_by_path(ename, ' > ') as full_path
from emp
start with job = 'MANAGER'
connect by prior empno = mgr;
```

```
ENAME  MANAGER  FULL_PATH
-----  -
JONES   JONES      > JONES
SCOTT   JONES      > JONES > SCOTT
ADAMS   JONES      > JONES > SCOTT > ADAMS
FORD    JONES      > JONES > FORD
...
MILLER  CLARK      > CLARK > MILLER
13 rows selected.
```

# Сортировка результатов иерархич. запросов

```
select ename
, sys_connect_by_path(ename, '|')
as path
from emp
start with mgr is null
connect by prior empno = mgr
order SIBLINGS by ename;
ENAME  PATH
-----
KING    |KING
BLAKE   |KING|BLAKE
ALLEN   |KING|BLAKE|ALLEN
JONES   |KING|BLAKE|JONES
MARTIN  |KING|BLAKE|MARTIN
TURNER  |KING|BLAKE|TURNER
...
14 rows selected.
```

```
select ename
, sys_connect_by_path(ename, '|')
as path
from emp
start with mgr is null
connect by prior empno = mgr
order by ename;
ENAME PATH
-----
ADAMS  |KING|JONES|SCOTT|ADAMS
ALLEN  |KING|BLAKE|ALLEN
BLAKE  |KING|BLAKE
CLARK  |KING|CLARK
FORD   |KING|JONES|FORD
JONES  |KING|JONES
...
14 rows selected
```



# Выводы

---

- ▶ Иерархически запросы следует использовать если:
  - ▶ необходимо отобразить иерархию объектов
  - ▶ осуществить анализ данных у родителей либо потомков определённого объекта
  
- ▶ Дополнительные материалы:
  - ▶ <http://habrahabr.ru/blogs/sql/43955/>
  - ▶ <http://citforum.ru/database/oracle/recursive/>



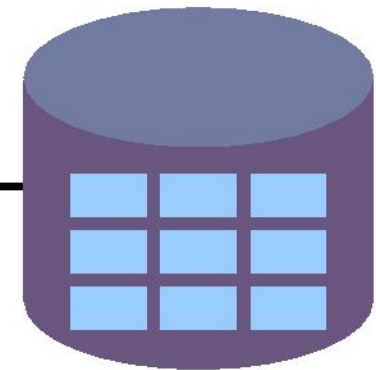
# Контроль прав доступа к БД

---

Администратор  
БД



Имя и пароль
Привелегии



Пользователи



# Разрешения

---

- Безопасность БД:
  - Системные разрешения
  - Разрешения доступа к данным
- Системные разрешения: разрешения на действия внутри базы данных
- Объектные разрешения: разрешения на действия с данными базы данных
- Схема: набор объектов, таких как таблицы, представления, объекты



# Системные разрешения

---

- Более 100 типов разрешений.
- Примеры основных разрешений:
  - Создать пользователя
  - Удалить пользователя
  - Удалить таблицу
  - Архивировать таблицу



# Создание пользователя

---

DBA может создать нового пользователя командой CREATE USER

```
CREATE USER user  
IDENTIFIED BY password;
```

```
CREATE USER demo  
IDENTIFIED BY demo;
```





# Назначение привилегий

---

После того, как пользователь создан, ему можно дать определенные разрешения:

```
GRANT privilege [, privilege...]  
TO user [, user | role, PUBLIC...];
```

Например разработчику можно дать разрешения на:

- CREATE SESSION
- CREATE TABLE
- CREATE SEQUENCE
- CREATE VIEW
- CREATE PROCEDURE



## Пример

---

DBA предоставляет пользователю demo привилегии на создание некоторых объектов.

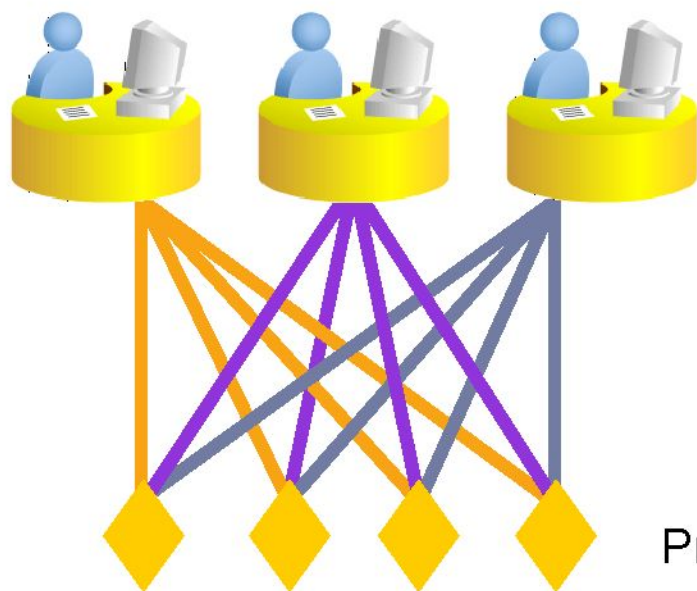
```
GRANT  create session, create table,  
       create sequence, create view  
TO     demo;
```



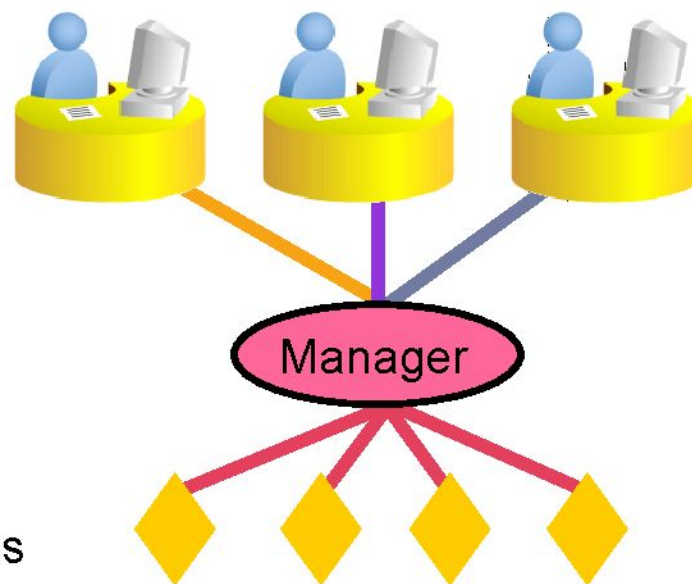
# Использование ролей

---

Пользователи



Назначение привилегий без ролей



Назначение привилегий с ролями



# Использование ролей

---

- Создание роли

```
CREATE ROLE manager;
```

- Предоставление разрешения роли

```
GRANT create table, create view  
TO manager;
```

- Связывание роли и пользователя

```
GRANT manager TO BELL, KOCHNAR;
```



# Изменение пароля пользователя

---

- DBA создает пользователей и задает им пароли
- Изменить пароль можно используя `ALTER USER`

```
ALTER USER demo  
IDENTIFIED BY employ;
```



# Разрешения и объекты БД

---

Действие	Таблица	View	Последовательность
ALTER	✓		✓
DELETE	✓	✓	
INDEX	✓		
INSERT	✓	✓	
REFERENCES	✓		
SELECT	✓	✓	✓
UPDATE	✓	✓	



# Разрешения на уровне объектов

---

- ▶ Для разных объектов можно назначить разные разрешения.
- ▶ Владелец объекта имеет все разрешения по отношению к объекту.
- ▶ Владелец может устанавливать разрешения доступа к объекту для других пользователей .

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



# Пример передачи разрешений

---

- Назначение разрешения на чтение:

```
GRANT  select
ON     emp
TO     demo;
```

- Назначение разрешений на отдельные столбцы определенным пользователям и ролям:

```
GRANT  update (department_name, location_id)
ON     departments
TO     demo, manager;
```





# Передача разрешений

---

- ▶ Установка разрешений с возможностью их передачи:

```
GRANT  select, insert
ON     dept
TO     demo
WITH   GRANT OPTION;
```

- ▶ Передача разрешений на чтение из псевдонимов таблицы:

```
GRANT  select
ON     alice.dept
TO     PUBLIC;
```



# Как проверить разрешения

---

Представление	Описание
ROLE_SYS_PRIVS	Системные разрешения переданные роли
ROLE_TAB_PRIVS	Разрешения на уровне таблиц, переданные роли
USER_ROLE_PRIVS	Каким ролям принадлежит пользователь
USER_SYS_PRIVS	Системные разрешения переданные пользователю
USER_TAB_PRIVS_MADE	Объектные разрешения на объекты пользователя
USER_TAB_PRIVS_RECD	Объектные разрешения на пользователя
USER_COL_PRIVS_MADE	Объектные разрешения на столбцы объектов пользователя
USER_COL_PRIVS_RECD	Объектные разрешения на столбцы для пользователя



# Отзыв привилегий

---

- Для отзыва привилегии используйте предложение REVOKE
- Так же можно отозвать привилегии WITH GRANT OPTION.

```
REVOKE {privilege [, privilege...]|ALL}
ON      object
FROM    {user[, user...]|role|PUBLIC}
[CASCADE CONSTRAINTS];
```



## Пример

---

Отзвы привелегий на вставку и чтение пользователю demo из таблицы DEPT.

```
REVOKE select, insert  
ON dept  
FROM demo;
```



# Вопрос

---

## Какие из утверждений справедливы?

1. После того, как пользователь создан ему могут быть переданы любые разрешения и использованием `GRANT`.
2. Пользователь может создать роль используя `CREATE ROLE` и передать через роль все свои системные и объектные разрешения другому пользователю.
3. Пользователи могут менять свои пароли.
4. Пользователь может просматривать свои разрешения и разрешения на свои объекты.



# Выводы

---

Бывают системные и объектные привилегии.

Привилегии можно передавать.

Предложение	Действие
CREATE USER	Создать пользователя (DBA)
GRANT	Дать разрешение
CREATE ROLE	Создать роль (DBA)
ALTER USER	Изменить пароль пользователя
REVOKE	Отзыв разрешений



## Домашнее чтение

---

- ▶ Лекции по дисциплине "Администрирование баз данных и приложений"