

Управление данными

Лекция 10

# Привилегии и транзакции

Свечников С.В.

# **Содержание занятия**

01. Пользователи и привилегии

02. Транзакции

# 01. Пользователи и привилегии

# Привилегии

**Привилегии** – операции определяющие возможность пользователей СУБД выполнять операцию.

Назначение и отмену привилегий обеспечивают два оператора SQL: GRANT и REVOKE

Привилегии делятся на:

- ☐ Системные привилегии – контролируют общий доступ к БД.
- ☐ Объектные привилегии – связаны с конкретным объектом БД.

# Объектные привилегии

**ALTER** – право на изменение таблиц БД

**SELECT** – право на выполнение запросов к таблице

**INSERT** – право на добавление данных в таблицу

**UPDATE** – право на обновление данных в таблице

**DELETE** – право на удаление строк из таблицы

**REFERENCES** – право на определение внешнего ключа

**INDEX** – право на создание индексов таблиц

**DROP** – право на удаление таблиц

Владелец таблицы или представления автоматически получает все эти привилегии и может предоставлять их другим пользователям.

# Предоставление привилегий

Предоставление привилегии пользователю выполняется с использованием команды GRANT.

## Пример

```
GRANT SELECT ON группа TO andrey@localhost;
```

После выполнения команды пользователь andrey@localhost может выполнять запросы к таблице группа.

В одном операторе можно указать сразу несколько привилегий или несколько пользователей.

## Пример

```
GRANT SELECT, INSERT ON группа TO andrey@localhost;
```

```
GRANT SELECT, INSERT ON группа TO andrey@localhost, maria@localhost;
```

# Привилегии на столбцы

Для трех привилегий: UPDATE, INDEX и REFERENCES можно явно указать столбцы, с которыми они будут работать.

## Пример

```
GRANT UPDATE(lname, fname) ON person TO andrey@localhost;
```

# ALL и PUBLIC

Оператор GRANT может иметь аргументы со специальным значением ALL PRIVILEGES или просто ALL и PUBLIC.

**ALL** обозначает все привилегии, которые можно предоставить пользователю для таблицы.

## Пример

```
GRANT ALL ON группа TO andrey@localhost;
```

**PUBLIC** обозначает всех пользователей, которым назначаются привилегии.

## Пример

```
GRANT SELECT ON person TO PUBLIC;
```



# Передача назначения привилегий

Для того, чтобы другие пользователи могли назначать привилегии для таблиц, автором которых они не являются используется предложение **WITH GRANT OPTION**.

## Пример

root – владелец таблицы

root передает права - andrey

andrey передает права - maria

**root:**

```
GRANT SELECT ON person TO andrey@localhost  
WITH GRANT OPTION;
```

**andrey:**

```
GRANT SELECT ON person TO maria@localhost;
```

Пользователь имеющий привилегию с **GRANT OPTION** может передать эту привилегию в том числе и с **GRANT OPTION** другим пользователям.

# Отмена привилегий

Отменить назначенные привилегии можно с помощью команды **REVOKE**.

## **Пример**

```
REVOKE SELECT ON person FROM maria@localhost;
```

Как и в GRANT здесь доступны списки привилегий и пользователей.

## **Пример**

```
REVOKE SELECT, UPDATE ON person FROM maria@localhost;
```

## 02. Транзакции

# Транзакция

**Транзакция** – последовательность команд SQL, которая воспринимается как единой целое.

Иницилируя сеанс работы с SQL вы начинаете транзакцию. Все вводимые команды будут входить в одну транзакцию, пока не будет выполнен оператор фиксирующий транзакцию или отменяющий ее.

**COMMIT WORK** – делает изменения постоянными

**ROLLBACK WORK** – отменяет сделанные изменения

# Откат транзакции

ROLLBACK WORK отменяет сделанные изменения в транзакции.

## Пример

Отключаем автоматическую фиксацию изменений  
`SET AUTOCOMMIT=0;`

Выполняем команду  
`DELETE FROM person WHERE pid=2;`  
`SELECT * FROM person;`

Откатываем транзакцию  
`ROLLBACK WORK;`  
`SELECT * FROM person;`

# Фиксация транзакции

COMMIT WORK делает изменения в транзакции постоянными.

## Пример

Отключаем автоматическую фиксацию изменений  
SET AUTOCOMMIT=0;

Выполняем команду  
DELETE FROM person WHERE pid=3;  
SELECT \* FROM person;

Фиксируем транзакцию  
**COMMIT WORK;**  
SELECT \* FROM person;

# DROP TABLE

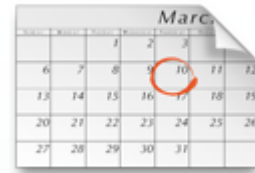
Команды ALTER TABLE и DROP TABLE автоматически завершают транзакции, как если бы после них была выполнена команда COMMIT WORK.

# Вопросы





# Домашнее задание



- ☐ Изучить главы 16 и 17 книги М.Грабера Введение в SQL
- ☐ Подготовиться к контрольной работе по лекции

# Контроль



Для выполнения контрольного теста используем ссылку:

**[app.startexam.com/Center/Web/student](http://app.startexam.com/Center/Web/student)**

Короткая ссылка:

**[clck.ru/2Undm](http://clck.ru/2Undm)**

Пароль на тест:

...