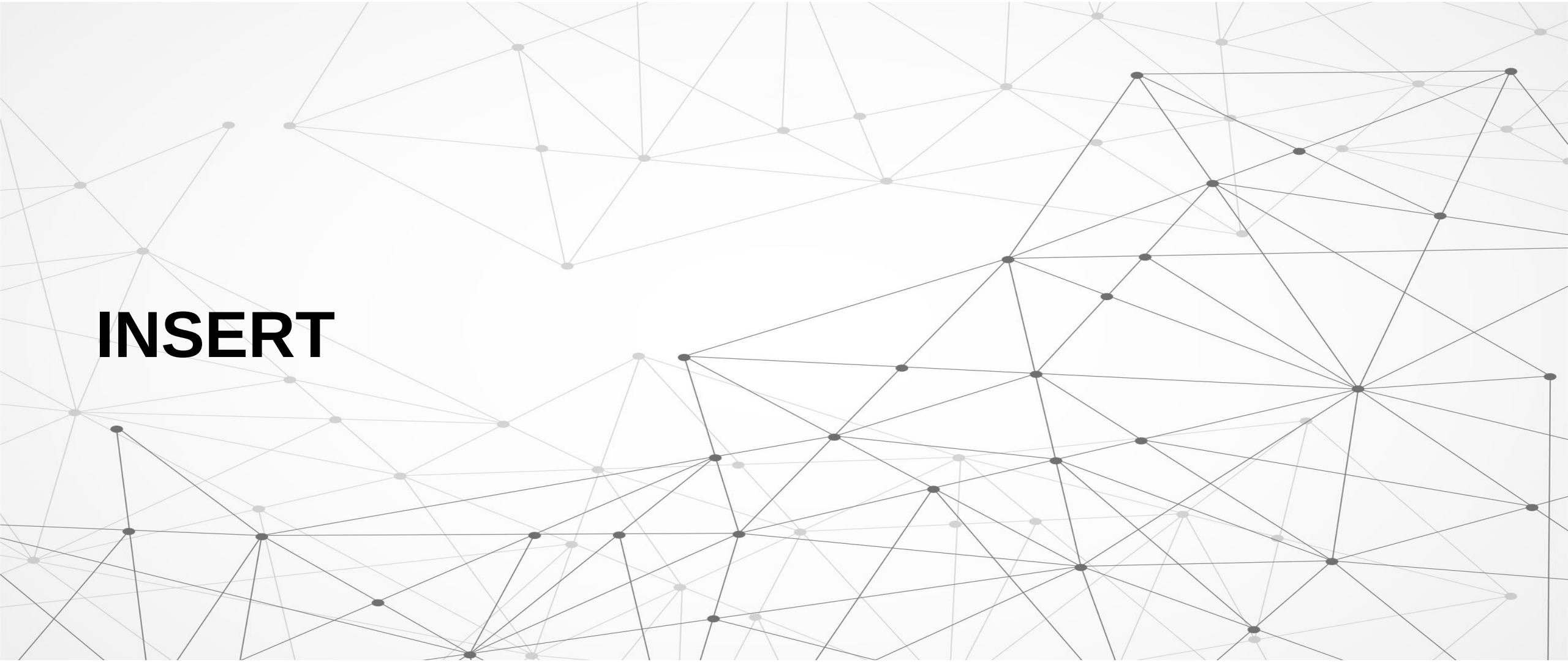
INSERT



Data Science Academy

Темы

• Добавление новых строк в таблицу

–инструкция INSERT

• Изменение данных в таблице

–инструкция UPDATE

• Удаление строк из таблицы:

–инструкция DELETE

–инструкция TRUNCATE

• Управление транзакциями базы данных с помощью инструкций COMMIT, ROLLBACK и SAVEPOINT

• Целостность чтения

• Предложение FOR UPDATE в инструкции SELECT

Data Science Academy

Язык манипулирования данными DML

•Инструкция DML выполняется в следующих ситуациях:

–добавление новых строк в таблицу

–изменение существующих строк в таблице

–удаление существующих строк из таблицы

•**Транзакция** *состоит из набора инструкций DML, образующих логический рабочий блок.*

Data Science Academy

Синтаксис инструкции INSERT

- Добавление нескольких строк в таблицу с помощью инструкции INSERT

```
INSERT INTO   table [(column [, column...])]
VALUES        (value [, value...]);
```

- При использовании этого синтаксиса вставляется только одна строка.

Data Science Academy

Вставка новых строк

- Вставьте новую строку, которая содержит значения для каждого столбца.
- Перечислите значения в соответствии со стандартным порядком столбцов в таблице.
- Перечислите столбцы в предложении `INSERT` (необязательно).

```
INSERT INTO departments(department_id,
       department_name, manager_id, location_id)
VALUES (70, 'Public Relations', 100, 1700);
```

- Символьные значения и даты заключаются в одиночные кавычки.

```
1 rows inserted
```

Data Science Academy

Вставка строк с пустыми значениями (Null)

- Неявный метод: исключение столбца из списка столбцов.

```
INSERT INTO   departments (department_id,
                           department_name)
VALUES        (30, 'Purchasing');
```

```
1 rows inserted
```

- Явный метод: задание ключевого слова NULL в предло-жении VALUES.

```
INSERT INTO   departments
VALUES        (100, 'Finance', NULL, NULL);
```

```
1 rows inserted
```

Data Science Academy

Копирование строк из другой таблицы

- Запишите инструкцию `INSERT` с подзапросом:

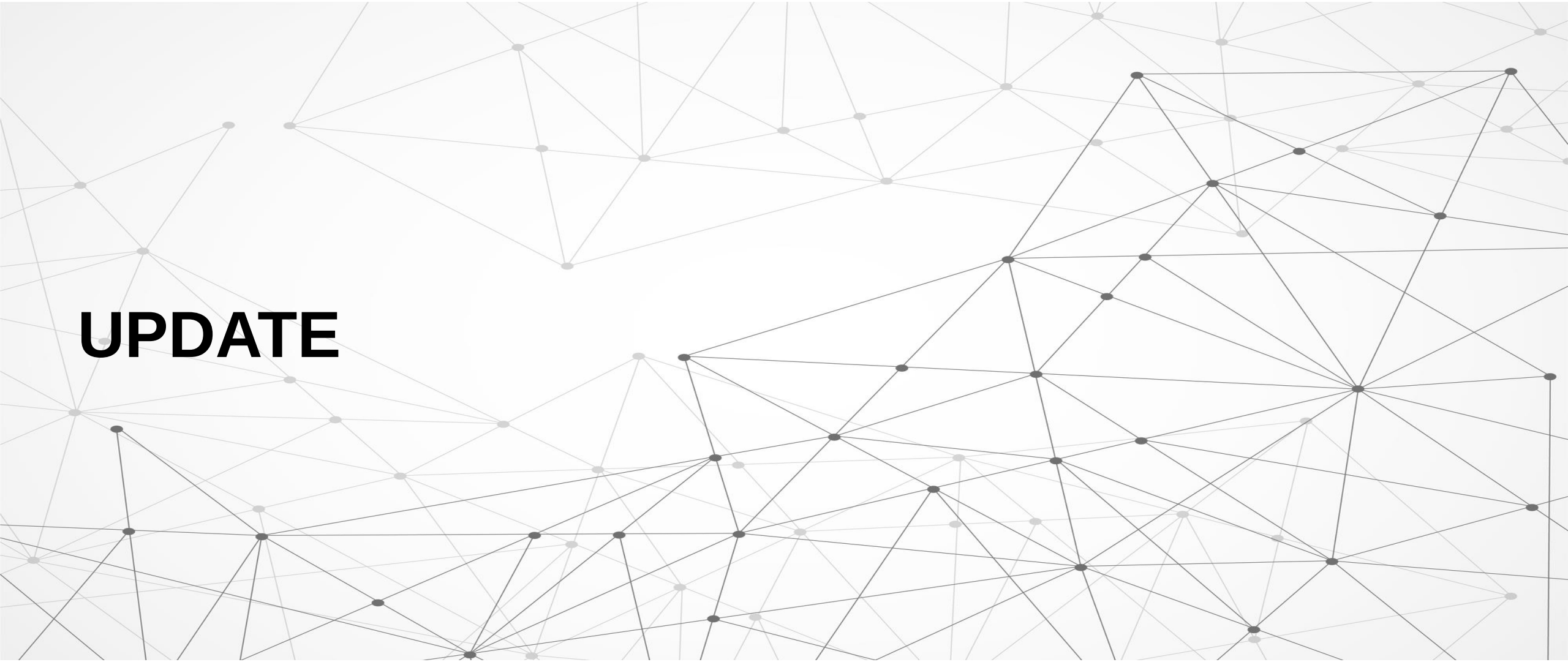
```
INSERT INTO sales_reps(id, name, salary, commission_pct)
  SELECT employee_id, last_name, salary, commission_pct
  FROM   employees
  WHERE  job_id LIKE '%REP%';
```

```
4 rows inserted
```

- Не используйте предложение `VALUES`.
- Число столбцов в предложении `INSERT` и подзапросе должно совпадать.
- Вставьте все строки, возвращенные по подзапросу, в таблицу `sales_reps`.

Data Science Academy

UPDATE



Data Science Academy

Синтаксис инструкции UPDATE

- Измените существующие значения в таблице с помощью инструкции UPDATE:

```
UPDATE      table
SET         column = value [, column = value, ...]
[WHERE      condition];
```

- Обновите сразу несколько строк (при необходимости).

Data Science Academy

Обновление строк в таблице

- При использовании предложения `WHERE` изменяются значения конкретных строк:

```
UPDATE employees
SET    department_id = 50
WHERE  employee_id = 113;
```

```
1 rows updated
```

- При пропуске предложения `WHERE` изменяются значения всех строк в таблице:

```
UPDATE   copy_emp
SET      department_id = 110;
```

```
22 rows updated
```

- Укажите `SET` *`column_name`*`= NULL`, чтобы изменить значение столбца на `NULL`.

Data Science Academy

Обновление двух столбцов с помощью подзапроса

Обновите должность и оклад работника 113, чтобы они совпадали с аналогичными значениями для работника 205.

```
UPDATE    employees
SET       job_id   = (SELECT   job_id
                     FROM     employees
                     WHERE    employee_id = 205),
          salary   = (SELECT   salary
                     FROM     employees
                     WHERE    employee_id = 205)
WHERE     employee_id     =  113;
```

```
1 rows updated
```

Data Science Academy

Обновление строк на основе другой таблицы

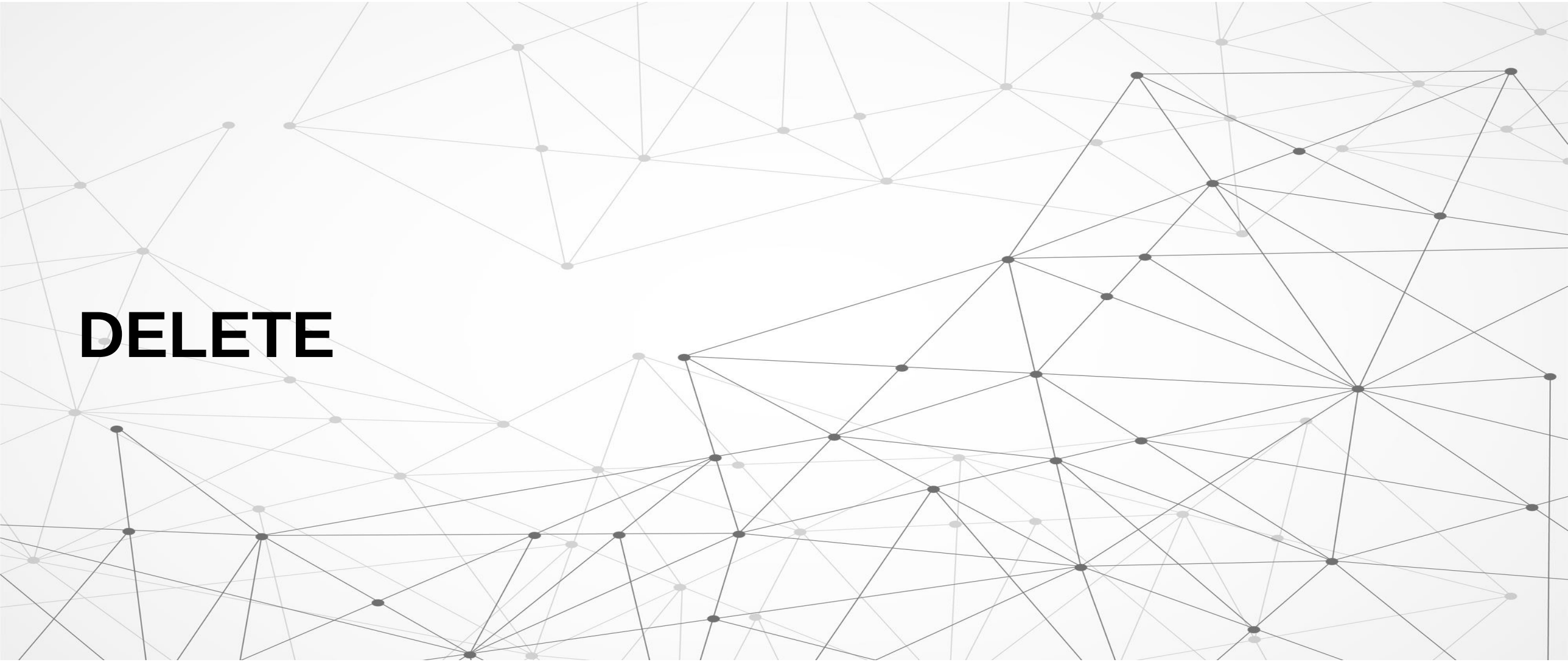
Использование подзапросов в инструкциях `UPDATE` позволяет обновлять значения строк в таблице на основе значений из другой таблицы:

```
UPDATE   copy_emp
SET      department_id  =  (SELECT department_id
                           FROM employees
                           WHERE employee_id = 100)
WHERE    job_id         =  (SELECT job_id
                           FROM employees
                           WHERE employee_id = 200);
```

```
1 rows updated
```

Data Science Academy

DELETE



Data Science Academy

Инструкция DELETE

Инструкция DELETE позволяет удалить существующие строки из таблицы:

```
DELETE [FROM]   table
[WHERE          condition];
```

Data Science Academy

Удаление строк из таблицы

- Использование предложения WHERE позволяет удалить заданные строки:

```
DELETE FROM departments
WHERE  department_name = 'Finance';
```

```
1 rows deleted
```

- При отсутствии предложения WHERE удаляются все строки в таблице:

```
DELETE FROM  copy_emp;
```

```
22 rows deleted
```

Data Science Academy

Удаление строк на основе другой таблицы

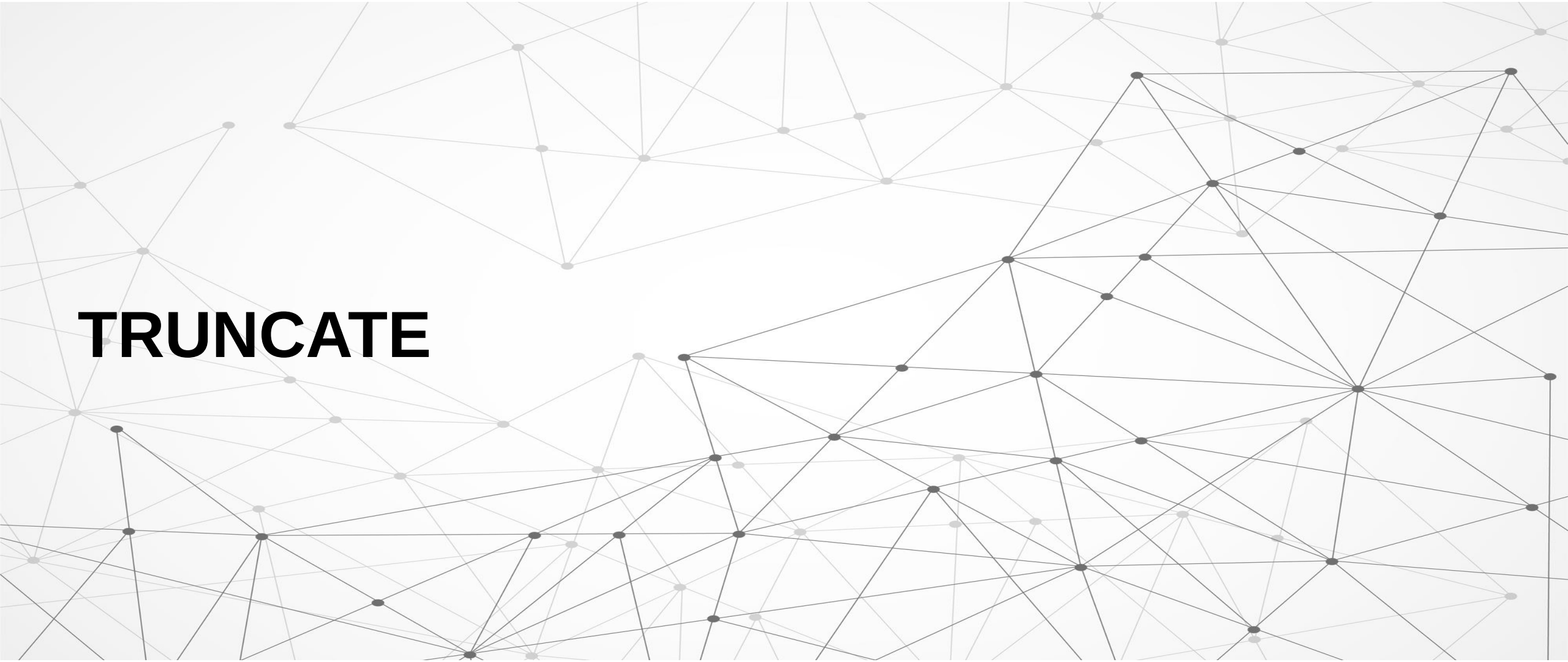
Использование подзапросов в инструкциях DELETE для удаления строк в таблице на основе значений из другой таблицы:

```
DELETE FROM employees
WHERE  department_id =
                (SELECT department_id
                 FROM   departments
                 WHERE  department_name
                        LIKE '%Public%');
```

```
1 rows deleted
```

Data Science Academy

TRUNCATE



Data Science Academy

Инструкция TRUNCATE

- Удаляет все строки из таблицы, оставляя ее пустой и сохраняя структуру таблицы
- Является инструкцией языка определения данных (DDL), а не DML; практически не подлежит отмене
- Синтаксис:

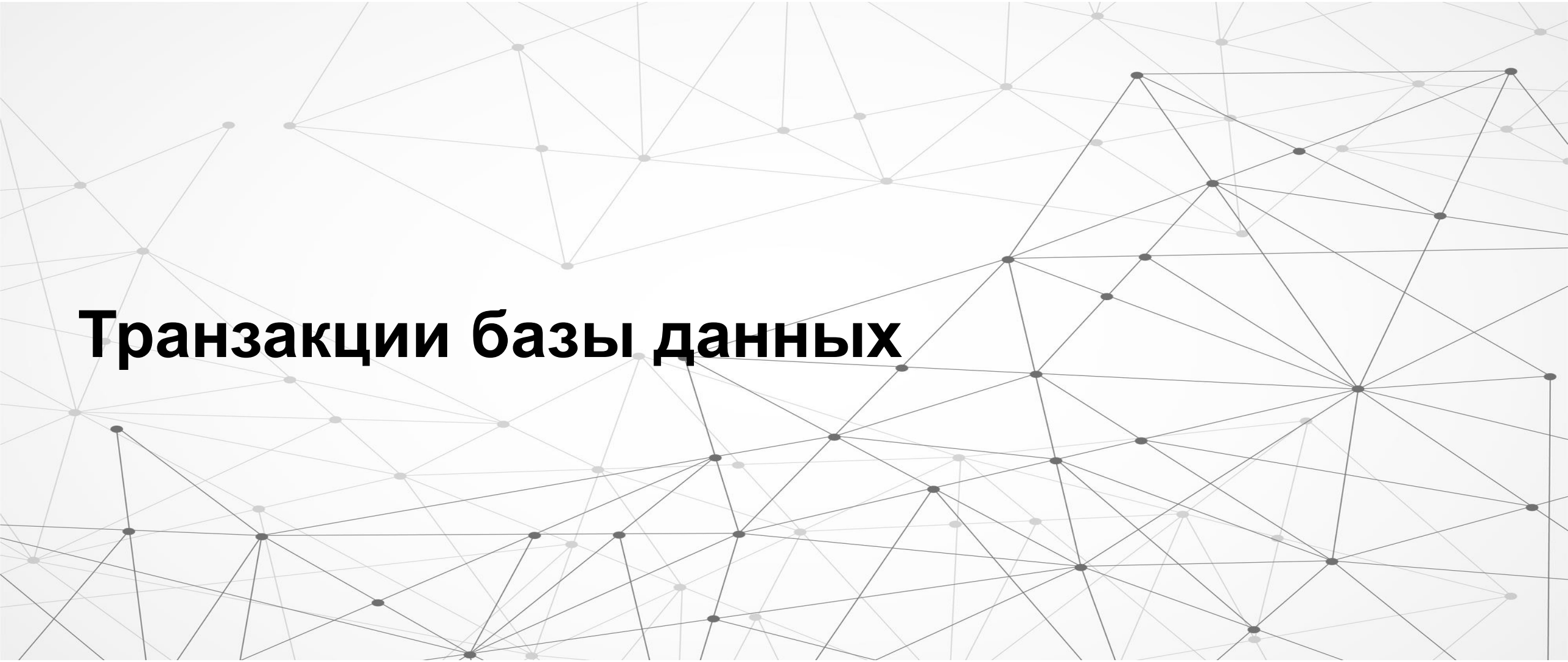
```
TRUNCATE TABLE table_name;
```

- Пример:

```
TRUNCATE TABLE copy_emp;
```

Data Science Academy

Транзакции базы данных



Data Science Academy

Транзакции базы данных

Состав транзакции базы данных:
•инструкции DML, составляющие одно согласованное изменение данных
•одна инструкция DDL
•одна инструкция языка управления данными (DCL)

Транзакции базы данных: начало и завершение

- Начинается при выполнении первой инструкции SQL DML.
- Завершается одним из следующих событий:
  - Запуск инструкции COMMIT или ROLLBACK.
  - Выполнение инструкции DDL или DCL (автоматическая фиксация).
  - Завершение пользователем работы SQL Developer или SQL*Plus.
  - Отказ системы.

Data Science Academy

Преимущества инструкций COMMIT и ROLLBACK

Инструкции COMMIT и ROLLBACK позволяют выполнять следующие задачи:

- обеспечивать согласованность данных
- просматривать изменения данных перед их сохранением
- группировать логически связанные операции

Data Science Academy

Фиксация данных

- Внесите изменения:

```
DELETE FROM employees
WHERE  employee_id = 99999;
```

```
1 rows deleted
```

```
INSERT INTO departments
VALUES (290, 'Corporate Tax', NULL, 1700);
```

```
1 rows inserted
```

- Зафиксируйте изменения:

```
COMMIT;
```

```
COMMIT succeeded.
```

Data Science Academy

Состояние данных после отката

Отмена всех отложенных изменений с помощью инструк-ции отката ROLLBACK:

- Изменения данных отменяются.
- Восстанавливается предыдущее состояние данных.
- Затронутые строки разблокируются.

```
DELETE FROM copy_emp;
ROLLBACK ;
```

Data Science Academy

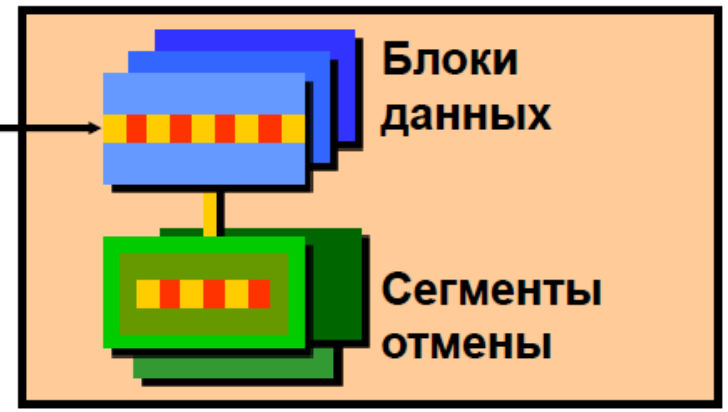
Пример состояния данных после отката

```
DELETE FROM test;
25000 rows deleted.

ROLLBACK;
Rollback complete.

DELETE FROM test WHERE  id = 100;
1 row deleted.

SELECT * FROM   test WHERE  id = 100;
No rows selected.

COMMIT;
Commit complete.
```

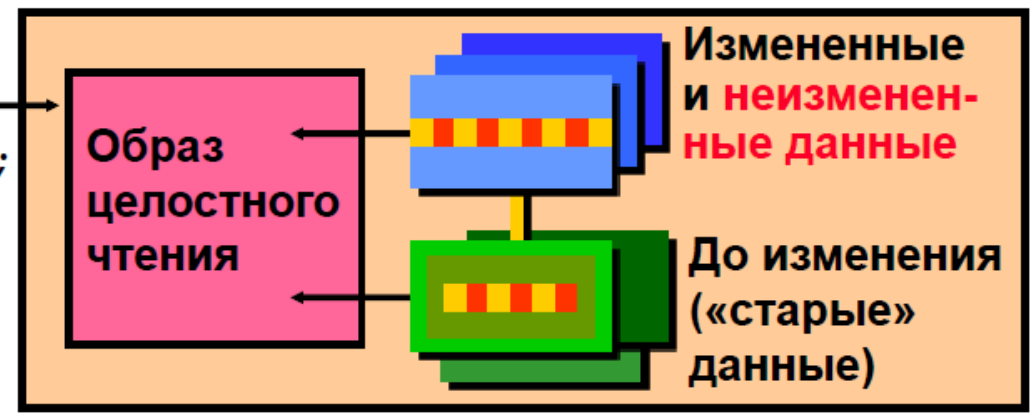
Data Science Academy

Пользователь A

```
UPDATE employees
SET    salary = 7000
WHERE  last_name = 'Grant';
```



```
SELECT  *
FROM userA.employees;
```



Пользователь B

Data Science Academy