



Занятие №7



**IT
Education
Academy**

WWW.ITEA.UA



**www.itea.
ua**

Поляков Антон

◆ Python backend developer

Фото
инструктора

КОНТАКТНЫЕ ДАННЫЕ

Telegram: @polyakov1

Базы данных

SQL — язык структурированных запросов. Создан для создания, изменения и управлениями даных в реляционной базе данных.

Основные функции:

- создание в базе данных новой таблицы;

- добавление в таблицу новых записей;

- изменение записей;

- удаление записей;

- выборка записей из одной или нескольких таблиц (в соответствии с заданным условием);

- изменение структур таблиц.

База данных

Представленное в определенной норме хранилище информации. Хранилище должно быть структуризовано и систематизировано, для лучшей возможности поиска и модификации этих же данных.

СУБД (Система управления базами данных) — набор инструментов для работы с БД. Является аналогией компилятора/интерпретатора в контексте программирования.

Запросы

SELECT - extracts data from a database

UPDATE - updates data in a database

DELETE - deletes data from a database

INSERT INTO - inserts new data into a database

CREATE DATABASE - creates a new database

ALTER DATABASE - modifies a database

CREATE TABLE - creates a new table

ALTER TABLE - modifies a table

DROP TABLE - deletes a table

CREATE INDEX - creates an index (search key)

DROP INDEX - deletes an index

Запросы

SELECT Syntax

SELECT field1, field2 FROM TABLENAME;

SELECT * FROM TABLENAME; #Полная выборка по таблице

SELECT DISTINCT Country FROM Customers; #DISTINCT собирает уникальные значения

SELECT COUNT(DISTINCT Country) FROM Customers; #Собираем количество уникальных значений с таблицы

БД

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
1	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209	Germany
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico
3	Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312	México D.F.	05023	Mexico
4	Around the Horn	Thomas Hardy	120 Hanover Sq.	London	WA1 1DP	UK
5	Berglunds snabbköp	Christina Berglund	Berguvsvägen 8	Luleå	S-958 22	Sweden

УСЛОВИЯ

Operator	Description	Example
=	Equal	Try it
>	Greater than	Try it
<	Less than	Try it
>=	Greater than or equal	Try it
<=	Less than or equal	Try it
<>	Not equal. Note: In some versions of SQL this operator may be written as !=	Try it
BETWEEN	Between a certain range	Try it
LIKE	Search for a pattern	Try it
IN	To specify multiple possible values for a column	Try it

Условия

AND Syntax

```
SELECT column1, column2, ...
FROM table_name
WHERE condition1 AND condition2 AND condition3 ...;
```

OR Syntax

```
SELECT column1, column2, ...
FROM table_name
WHERE condition1 OR condition2 OR condition3 ...;
```

NOT Syntax

```
SELECT column1, column2, ...
FROM table_name
WHERE NOT condition;
```



УСЛОВИЯ

```
UPDATE table_name  
SET column1 = value1, column2 = value2, ...  
WHERE condition;
```

DB example:

```
UPDATE Customers  
SET ContactName = 'Alfred Schmidt', City= 'Frankfurt'  
WHERE CustomerID = 1;
```

УСЛОВИЯ

```
SELECT column_name(s)  
FROM table_name  
WHERE condition  
LIMIT number;
```

Limit — ограничивает кол-во возвращаемых данных

```
SELECT column_name(s)  
FROM table_name  
WHERE column_name IN (value1, value2, ...);
```

IN — аналогия оператора в python



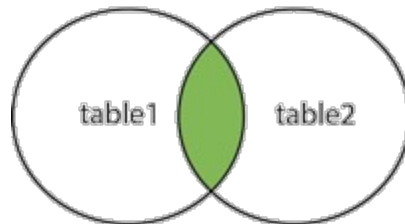
Условия

```
SELECT column_name(s)  
FROM table_name  
WHERE column_name IN (SELECT STATEMENT);
```

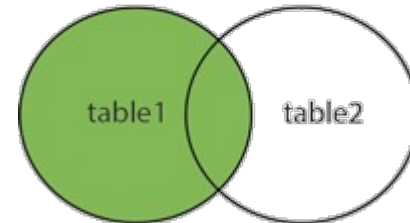
Вложенный запрос в in.

Join

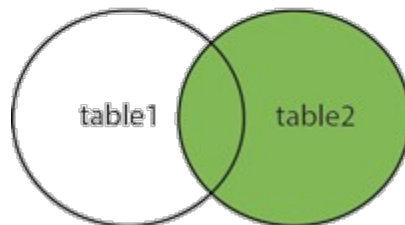
INNER JOIN



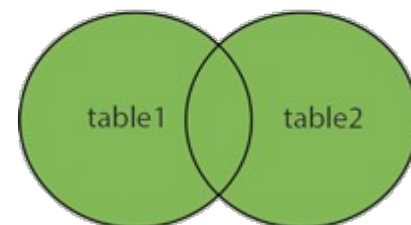
LEFT JOIN



RIGHT JOIN

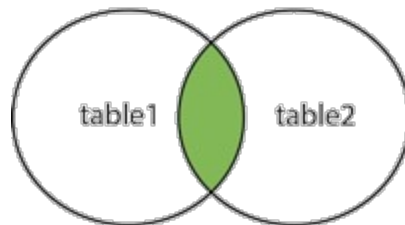


FULL OUTER JOIN

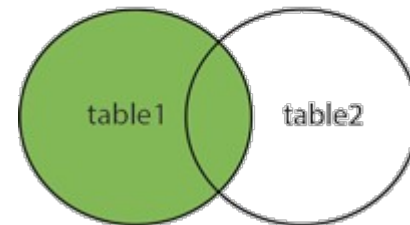


Join

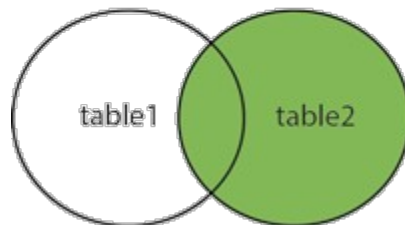
INNER JOIN



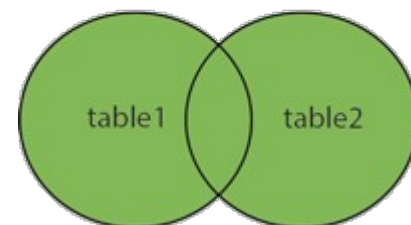
LEFT JOIN



RIGHT JOIN



FULL OUTER JOIN





Join

```
SELECT column_name(s)
FROM table1
INNER JOIN table2
ON table1.column_name = table2.column_name;
```

Задачи

- 1) Написать контекстный менеджер для работы с SQLite DB.
- 2) Создать базу данных студентов. У студента есть факультет, группа, оценки, номер студенческого билета. Написать программу, с двумя ролями: Администратор, Пользователь. Администратор может добавлять, изменять существующих студентов. Пользователь может получать список отличников, список всех студентов, искать студентов по номеру студенческого, получать полную информацию о конкретном студенте (включая оценки, факультет)

SQLite3 connector

```
import sqlite3
```

```
conn = sqlite3.connect("mydatabase.db") # или :memory: чтобы  
сохранить в RAM
```

```
cursor = conn.cursor()
```

```
# Создаем курсор - это специальный объект который делает  
запросы и получает их результаты
```

```
# Создание таблицы
```

```
cursor.execute("""CREATE TABLE albums  
                (title text, artist text, release_date text,  
                 publisher text, media_type text)  
                """)
```

Example

```
import sqlite3

conn = sqlite3.connect("mydatabase.db")
cursor = conn.cursor()
sql = "SELECT * FROM albums WHERE artist=?"
cursor.execute(sql, [("Red")])
print(cursor.fetchall()) # or use fetchone()

print("Here's a listing of all the records in the table:")
for row in cursor.execute("SELECT rowid, * FROM albums ORDER BY
artist"):
    print(row)
print("Results from a LIKE query:")
sql = "SELECT * FROM albums WHERE title LIKE 'The%'"
cursor.execute(sql)
print(cursor.fetchall())
```

SECURITY

Never do this -- insecure!

```
symbol = 'RHAT'
```

```
c.execute("SELECT * FROM stocks WHERE symbol = '%s'" % symbol)
```

Do this instead

```
t = ('RHAT',)
```

```
c.execute('SELECT * FROM stocks WHERE symbol=?', t)
```

```
print(c.fetchone())
```

Larger example that inserts many records at a time

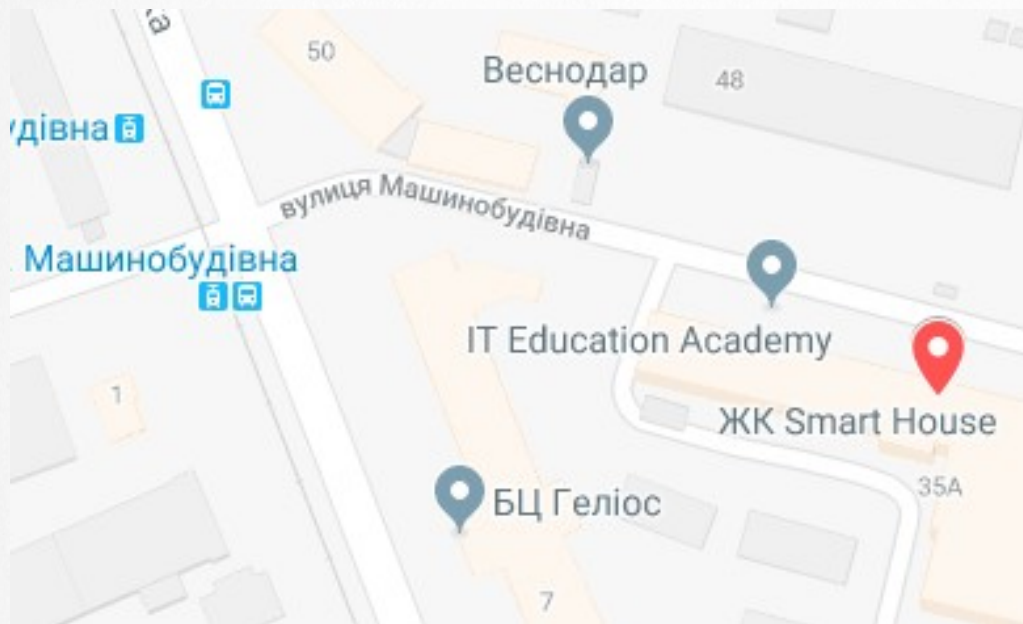
```
purchases = [('2006-03-28', 'BUY', 'IBM', 1000, 45.00),
```

```
              ('2006-04-05', 'BUY', 'MSFT', 1000, 72.00),
```

```
              ('2006-04-06', 'SELL', 'IBM', 500, 53.00),
```

```
]
```

```
c.executemany('INSERT INTO stocks VALUES (?, ?, ?, ?, ?)', purchases)
```



КОНТАКТНЫ Е ДАННЫЕ ITEA

ул. Машиностроительная, 41,
ЖК «Smart House», Киев

ул. Срибнокильская, 1, офис
269, Киев

пр. Академика Глушкова, 1,
корп.17, Киев

+38 (044) 599-01-79
facebook.com/Itea
info@itea.ua
itea.ua