

## Тема: Создание и работа с многотабличной БД

Принцип работы запросов на объединения таблиц в SQL и реляционных базах данных заключается в том, что внутри одного SQL запроса SELECT выполняется два или более подзапроса (в зависимости от того, сколько мы хотим объединить таблиц), подзапросы разделяются между собой ключевым словом JOIN. У этого JOIN есть ограничение ON, которое называют предикатом объединения. Предикат объединения – это всегда какое-то условие, с помощью которого РСУБД определяет какие строки из двух таблиц ей нужно объединять. А вот с тем, как объединять строки, SQLite разбирается специальным модификатором: INNER, LEFT OUTER или просто LEFT и CROSS.

### Предикат INNER

1. Готовим таблицы для реализации примеров SQL запросов JOIN в базе данных SQLite

```
4 with sq.connect('music.db') as con:
5     con.execute('PRAGMA foreign_keys = ON')
6     cur = con.cursor()
7     cur.execute("""CREATE TABLE IF NOT EXISTS tracks(
8         id INTEGER PRIMARY KEY,
9         title TEXT NOT NULL,
10        second INTEGER NOT NULL,
11        price REAL NOT NULL,
12        album_id INTEGER,
13        FOREIGN KEY (album_id) REFERENCES albums(id) ON DELETE CASCADE ON UPDATE CASCADE
14    )""")
15
16 with sq.connect('music.db') as con:
17     cur = con.cursor()
18     cur.execute("""CREATE TABLE IF NOT EXISTS albums(
19         id INTEGER PRIMARY KEY,
20         title TEXT NOT NULL,
21         artist_id INTEGER,
22         FOREIGN KEY (artist_id) REFERENCES artist(id) ON DELETE CASCADE ON UPDATE CASCADE
23     )""")
24
25 with sq.connect('music.db') as con:
26     cur = con.cursor()
27     cur.execute("""CREATE TABLE IF NOT EXISTS artist(
28         id INTEGER PRIMARY KEY,
29         name TEXT NOT NULL
30     )""")
```

ключевая фраза ON UPDATE CASCADE, которая указывается после FOREIGN KEY говорит СУБД о том, что та должна модифицировать обе таблицы друг за другом – каскадно. Таким образом, указав SQLite, что данные должны модифицироваться каскадом, мы сможем написать только один SQL запрос UPDATE, а SQLite сама обновит обе таблицы.

```
con.execute('PRAGMA foreign_keys = ON')
```

разрешает работу с внешними ключами

## 2. Добавляем информацию в БД

```
create_BD.py × data.py ×
1 info_tracks = [
2     (1, 'Девушка по городу', 193, 26.20, 2),
3     (2, 'Песня идущего домой', 170, 22.10, 2),
4     (3, 'Полковнику никто не пишет', 292, 32.15, 3),
5     (4, 'Мой друг', 291, 27.15, 3),
6     (5, 'Моё сердце', 249, 21.12, 1),
7     (6, 'Линия жизни', 180, 41.12, 1),
8     (7, 'Остаемся зимовать', 218, 17.62, 1),
9 ]
10
11 info_artist = [
12     (1, 'Вячеслав Бутусов'),
13     (2, 'Сплин'),
14     (3, 'Би - 2'),
15 ]
16
17 info_albums = [
18     (1, '25-й кадр', 2),
19     (2, 'Биографика', 1),
20     (3, 'Би - 2', 3),
21 ]
```

## 3. Внутреннее объединение таблиц в базах данных SQLite: INNER JOIN в SQL

```
#Объединение двух таблиц
with sq.connect('music.db') as con:
    cur = con.cursor()
    cur.execute("SELECT tracks.title, second, price, albums.title FROM tracks INNER JOIN "
                "Albums ON tracks.album_id = albums.id")
    result = cur.fetchall()
    print(result)

#Объединение трех таблиц
with sq.connect('music.db') as con:
    cur = con.cursor()
    cur.execute("SELECT tracks.title, albums.title, artist.name FROM tracks INNER JOIN "
                "Albums ON tracks.album_id = albums.id LEFT JOIN "
                "artist ON albums.artist_id = artist.id")
    result = cur.fetchall()
    print(result)
```

После SELECT необходимо перечислить имена столбцов из обеих таблиц, которые будут отображаться в результирующей таблице. Обратите внимание: если при объединение таблиц имена столбцов одинаковы, то следует использовать квалификатор: table\_name.col\_name.

Внутреннее объединение таблиц INNER JOIN в результате дает таблицу, состоящую из строк объединяемых таблиц, для которых результат выполнения предиката объединения ON равен TRUE. Другими словами: в результирующей таблице будут только те строки из левой таблицы, для которых есть связанные строки из правой.

4. Мы можем задавать различные условия выборки данных при помощи WHERE в том случае, когда объединяем таблице предикатом JOIN:

```
SELECT tracks.title, second, price, albums.title FROM tracks INNER JOIN
```

```
Albums ON tracks.album_id = albums.id
```

```
WHERE artist_id = 2
```