

Урок 3. Создание структуры базы данных

Опишите базу данных для школьного кабинета, в рамках которой можно фиксировать, кто и в какое время сидел за той или иной партой.

— Место ученика — это ряд, парта, вариант.

— В течение одного дня у разных классов бывает несколько уроков.

— Создайте минимум 3 таблицы: «Кабинет», «Ученик» и сводная таблица, где отображаются парты и ученики.

— Используйте поля «началоурока» и «конецурока» с типом данных TimeStamp.

Write for Database Python Code:

Для создания базы данных для школьного кабинета с использованием языка Python и СУБД SQLite можно использовать следующий код:

```
import sqlite3
```

```
# Create a connection to the database
```

```
conn = sqlite3.connect('school_classroom.db')
```

```
# Create the "Classroom" table with fields "id", "name", and "address"
```

```
conn.execute("""CREATE TABLE IF NOT EXISTS Classroom
```

```
    (id INTEGER PRIMARY KEY,
```

```
    name TEXT NOT NULL,
```

```
    address TEXT NOT NULL);""")
```

```
# Create the "Student" table with fields "id", "name", "row", "desk", and "variant"
```

```
conn.execute("""CREATE TABLE IF NOT EXISTS Student
```

```
    (id INTEGER PRIMARY KEY,
```

```
    name TEXT NOT NULL,
```

```
    row INTEGER NOT NULL,
```

```
    desk INTEGER NOT NULL,
```

```
    variant INTEGER NOT NULL);""")
```

```
# Create the "Schedule" table with fields "lesson_id", "classroom_id", "start_time", and "end_time"
```

```
conn.execute("""CREATE TABLE IF NOT EXISTS Schedule
```

```
    (lesson_id INTEGER PRIMARY KEY,
```

```
    classroom_id INTEGER NOT NULL,
```

```
    start_time TIMESTAMP NOT NULL,
```

```
    end_time TIMESTAMP NOT NULL,
```

```
    FOREIGN KEY(classroom_id) REFERENCES Classroom(id));""")
```

```
# Create the "Desks and Students" table with fields "lesson_id", "student_id", "row", "desk", and "variant"
```

```
conn.execute("""CREATE TABLE IF NOT EXISTS DeskStudent
```

```
    (lesson_id INTEGER NOT NULL,
```

```
    student_id INTEGER NOT NULL,
```

```
row INTEGER NOT NULL,  
desk INTEGER NOT NULL,  
variant INTEGER NOT NULL,  
PRIMARY KEY(lesson_id, student_id),  
FOREIGN KEY(lesson_id) REFERENCES Schedule(lesson_id),  
FOREIGN KEY(student_id) REFERENCES Student(id);""
```

```
# Close the connection to the database  
conn.close()
```

Этот код создает 4 таблицы: "Кабинет", "Ученик", "Расписание" и "Парты и ученики".
Таблицы "Кабинет" и "Ученик" содержат информацию о кабинете и ученике соответственно. Таблица "Расписание" содержит информацию о расписании уроков в кабинете, а таблица "Парты и ученики" содержит информацию о том, какие ученики сидели за какими партами во время каждого урока.
Запросы на добавление, изменение и удаление данных из этих таблиц можно выполнять с использованием языка SQL и методов объекта `conn`. Например, чтобы добавить нового ученика в таблицу "Ученик", можно использовать следующий код:

```
conn.execute("INSERT INTO Student (name, row, desk, variant) VALUES (?, ?, ?, ?)", ('John Doe', 1, 2, 3))  
conn.commit()
```

А чтобы получить список всех учеников, можно использовать следующий код:

```
cur = conn.cursor()  
cur.execute("SELECT * FROM Student")  
rows = cur.fetchall()  
for row in rows:  
    print(row)
```

Этот код получает курсор на таблицу "Ученик", выполняет запрос на выборку всех строк из этой таблицы и выводит их на экран.