

# Практика

Выполнил **Белик Игорь** и **Павлов Артур**.  
**Павлов Артур** - анализ, структурирование, создание таблиц  
**Белик Игорь** - реализация базы данных и *Sql*-запросов

## Цели.

- Структурировать данные
- Обеспечить гибкость
- Реализовать запросы

## Этап 1. Структура.

- Группы
- Дисциплины
- Преподаватели
- Часы
- Формы занятий(лекции, лабораторные, практические)
- Учебный год

## Этап 2. Структура базы данных.

### Таблица: Groups.

Поле	Тип	Комментарий
id	INT	Ключ
name	VARCHAR	Название группы

### Таблица: Subjects.

Поле	Тип	Комментарий
id	INT	Ключ
name	VARCHAR	Название дисциплины

### Таблица: Teachers.

Поле	Тип	Комментарий
id	INT	Ключ
full_name	VARCHAR	ФИО преподавателя

### Таблица: AcademicYears.

Поле	Тип	Комментарий
id	INT	Ключ
year	VARCHAR	Год

### Таблица: Lessons.

Поле	Тип	Комментарий
id	INT	Ключ
group_id	INT	Ссылается на <a href="#">Groups</a>
subject_id	INT	Ссылается на <a href="#">Subjects</a>
teacher_id	INT	Ссылается на <a href="#">Teachers</a>
year_id	INT	Ссылается на <a href="#">AcademicYears</a>
lecture_hours	INT	Уроки
lab_hours	INT	Лабораторные
practice_hours	INT	Практические
total_hours	INT	Сумма всех часов

## Этап 3. Создание таблиц и SQL-запросов.

### SQL: Создание таблиц.

```
-- Таблица: Группы
CREATE TABLE Groups (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    name VARCHAR(50) NOT NULL UNIQUE
);

-- Таблица: Дисциплины
CREATE TABLE Subjects (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    name VARCHAR(255) NOT NULL UNIQUE
);

-- Таблица: Преподаватели
CREATE TABLE Teachers (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    full_name VARCHAR(255) NOT NULL UNIQUE
);

-- Таблица: Учебные года
CREATE TABLE AcademicYears (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    year VARCHAR(20) NOT NULL UNIQUE
);

-- Таблица: Тарификация / Нагрузка
CREATE TABLE Lessons (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    group_id INTEGER NOT NULL,
    subject_id INTEGER NOT NULL,
    teacher_id INTEGER NOT NULL,
    year_id INTEGER NOT NULL,
    lecture_hours INTEGER DEFAULT 0,
    lab_hours INTEGER DEFAULT 0,
    practice_hours INTEGER DEFAULT 0,
    total_hours INTEGER GENERATED ALWAYS AS (
        lecture_hours + lab_hours + practice_hours
    ) STORED,

    FOREIGN KEY (group_id) REFERENCES Groups(id),
    FOREIGN KEY (subject_id) REFERENCES Subjects(id),
    FOREIGN KEY (teacher_id) REFERENCES Teachers(id),
    FOREIGN KEY (year_id) REFERENCES AcademicYears(id)
);
```

### SQL: Создание запросов.

```
SELECT
    s.name AS subject_name,
    t.full_name AS teacher_name
FROM Lessons l
JOIN Groups g ON l.group_id = g.id
JOIN Subjects s ON l.subject_id = s.id
JOIN Teachers t ON l.teacher_id = t.id
JOIN AcademicYears y ON l.year_id = y.id
WHERE g.name = 'П-23' AND y.year = '2024-2025';
```

Получить список дисциплин и преподавателей по группе на 2024–2025 год

```
SELECT
    t.full_name,
    g.name AS group_name,
    s.name AS subject_name,
    l.lecture_hours,
    l.lab_hours,
    l.practice_hours,
    l.total_hours
FROM Lessons l
JOIN Groups g ON l.group_id = g.id
JOIN Subjects s ON l.subject_id = s.id
JOIN Teachers t ON l.teacher_id = t.id
JOIN AcademicYears y ON l.year_id = y.id
WHERE t.full_name = 'Иванов И.И.' AND y.year = '2024-2025';
```

Получить нагрузку преподавателя по всем группам и предметам

```
SELECT
    t.full_name,
    SUM(l.total_hours) AS total_hours
FROM Lessons l
JOIN Teachers t ON l.teacher_id = t.id
JOIN AcademicYears y ON l.year_id = y.id
WHERE y.year = '2024-2025'
GROUP BY t.full_name;
```

Получить общее количество часов на преподавателя