**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра МОЭВМ**

отчет

**по лабораторной работе №2**

**по дисциплине «Базы данных»**

Тема: Реализация базы данных в СУБД PostgreSQL

|  |  |  |
| --- | --- | --- |
| Студент гр. 1303 |  | Бутыло Е.А. |
| Преподаватель |  | Заславский М.М. |

Санкт-Петербург

2023

**Цель работы.**

Развернуть локально PostgreSQL, написать запросы для создания и заполнения таблиц, написать запросы к БД, отвечающие на вопросы в задания.

**Текст задания**

Вариант 3

Пусть требуется создать программную систему, предназначенную для завуча школы. Она должна обеспечивать хранение сведений о каждом учителе, о предметах, которые он преподаёт, номере закреплённого за ним кабинета, о расписании занятий. Существуют учителя, которые не имеют собственного кабинета. Об учениках должны храниться следующие сведения: фамилия и имя, в каком классе учится, какую оценку имеет в текущей четверти по каждому предмету. Завуч должен иметь возможность добавить сведения о новом учителе или ученике, внести в базу данных четвертные оценки учеников каждого класса по каждому предмету, удалить данные об уволившемся учителе и отчисленном из школы ученике, внести изменения в данные об учителях и учениках, в том числе поменять оценку ученика по тому или иному предмету. В задачу завуча входит также составление расписания. Завучу могут потребоваться следующие сведения:

* Какой предмет будет в заданном классе, в заданный день недели на заданном уроке?
* Кто из учителей преподаёт в заданном классе?
* В каком кабинете будет 5-й урок в среду у некоторого класса?
* В каких классах преподаёт заданный предмет заданный учитель?
* Расписание на заданный день недели для указанного класса?
* Сколько учеников в указанном классе?

**Выполнение работы**

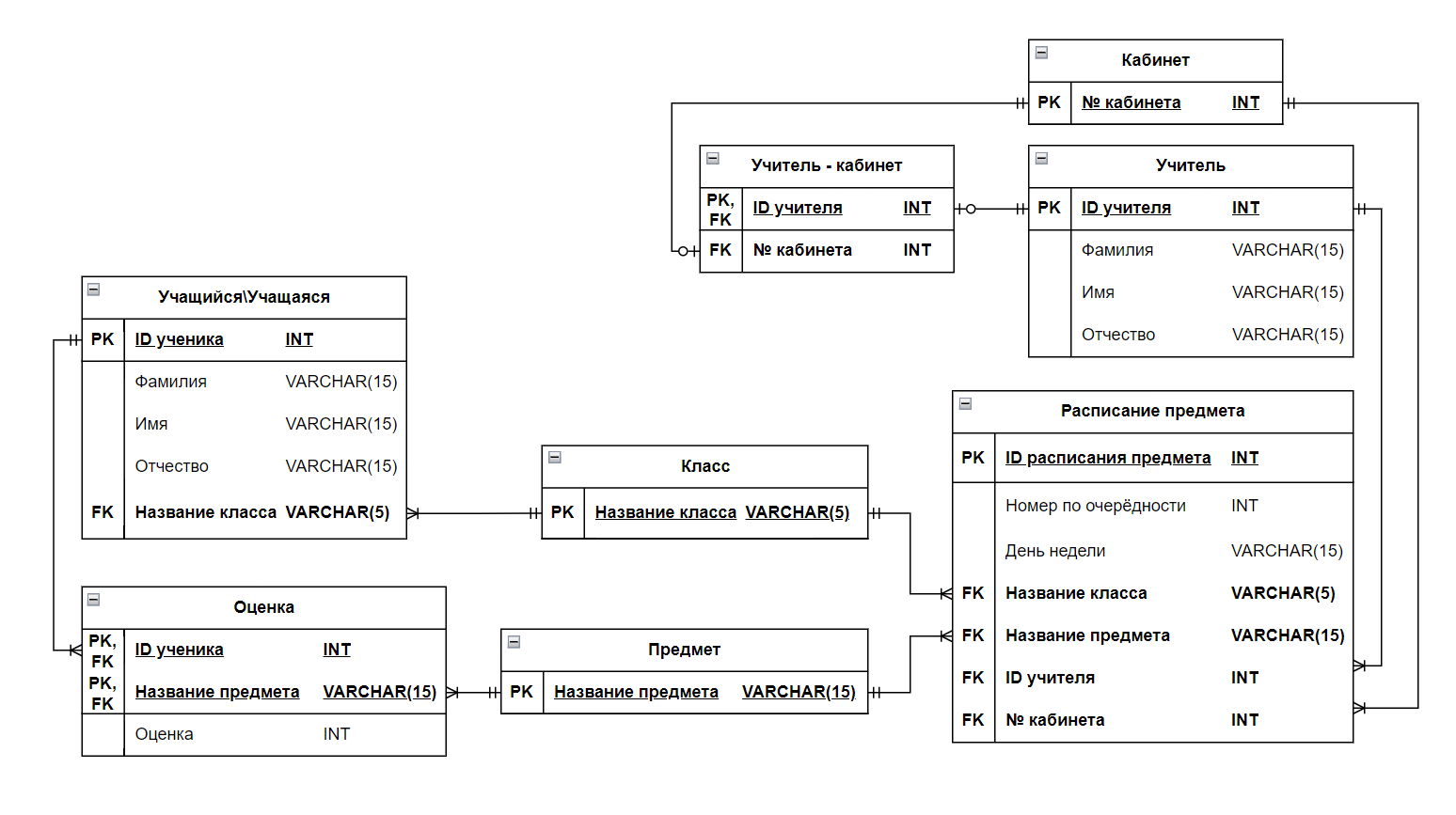


Рисунок 1. – Структура модели БД.

Развернули PostgreSQL локально, создали и заполнили таблицы.

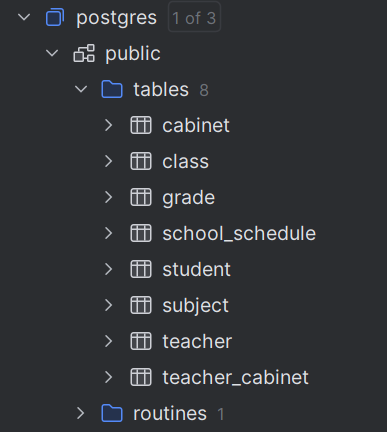


Рисунок 2. – Структура созданной БД.

Выполним запросы к БД предложенные вариантом:

1. Какой предмет будет в заданном классе, в заданный день недели на заданном уроке?

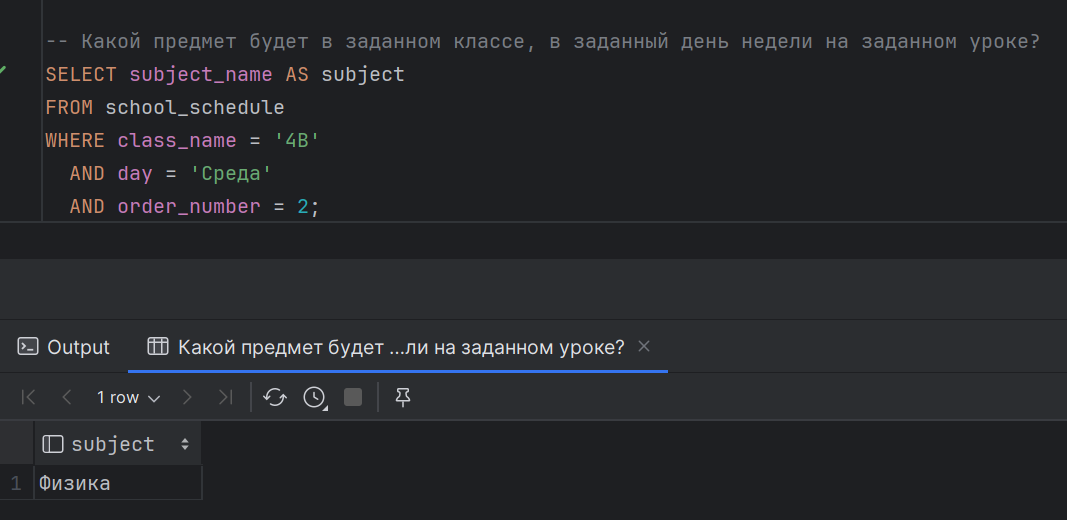


Рисунок 3. – Запрос и результат запроса.

1. Кто из учителей преподаёт в заданном классе?

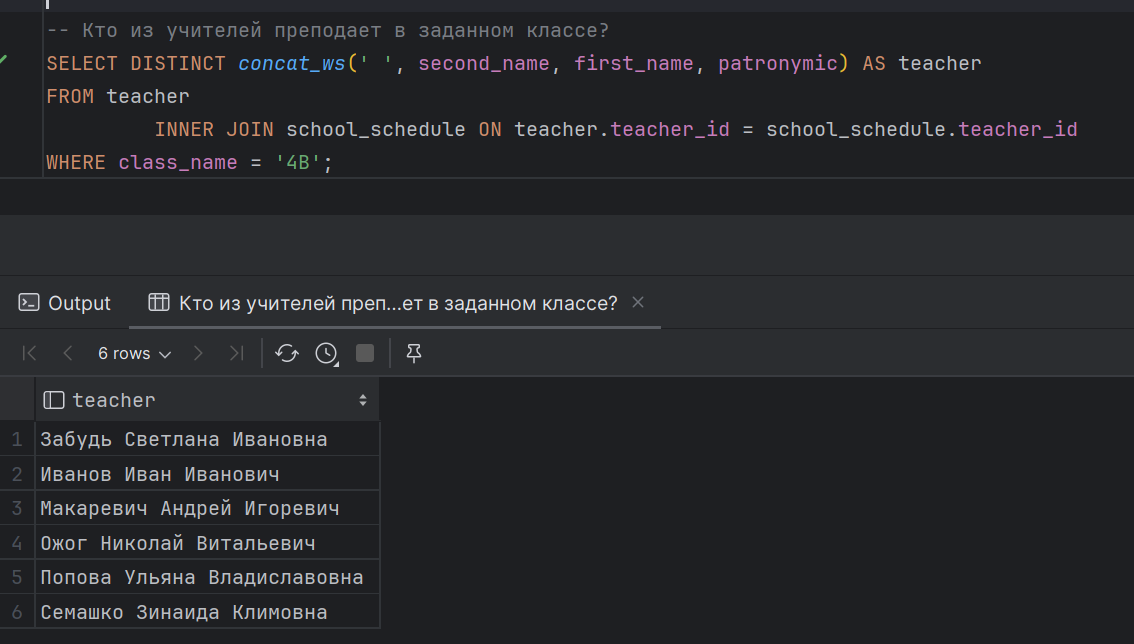


Рисунок 4. – Запрос и результат запроса.

1. В каком кабинете будет 5-й урок в среду у некоторого класса?

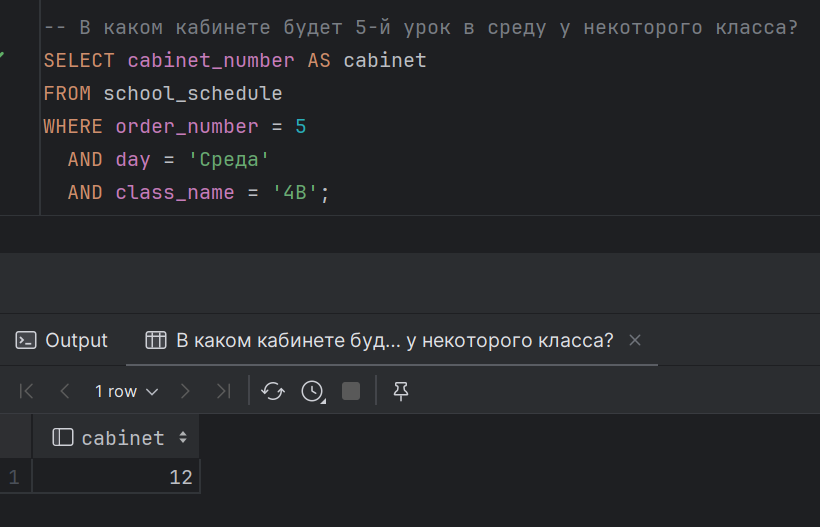


Рисунок 5. – Запрос и результат запроса.

1. В каких классах преподаёт заданный предмет заданный учитель?

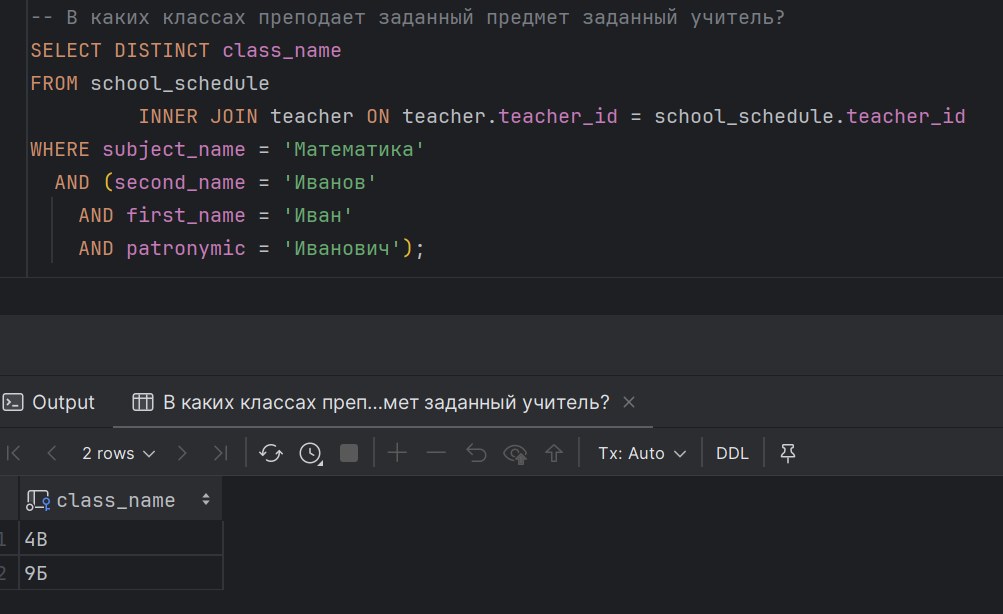


Рисунок 6. – Запрос и результат запроса.

1. Расписание на заданный день недели для указанного класса?

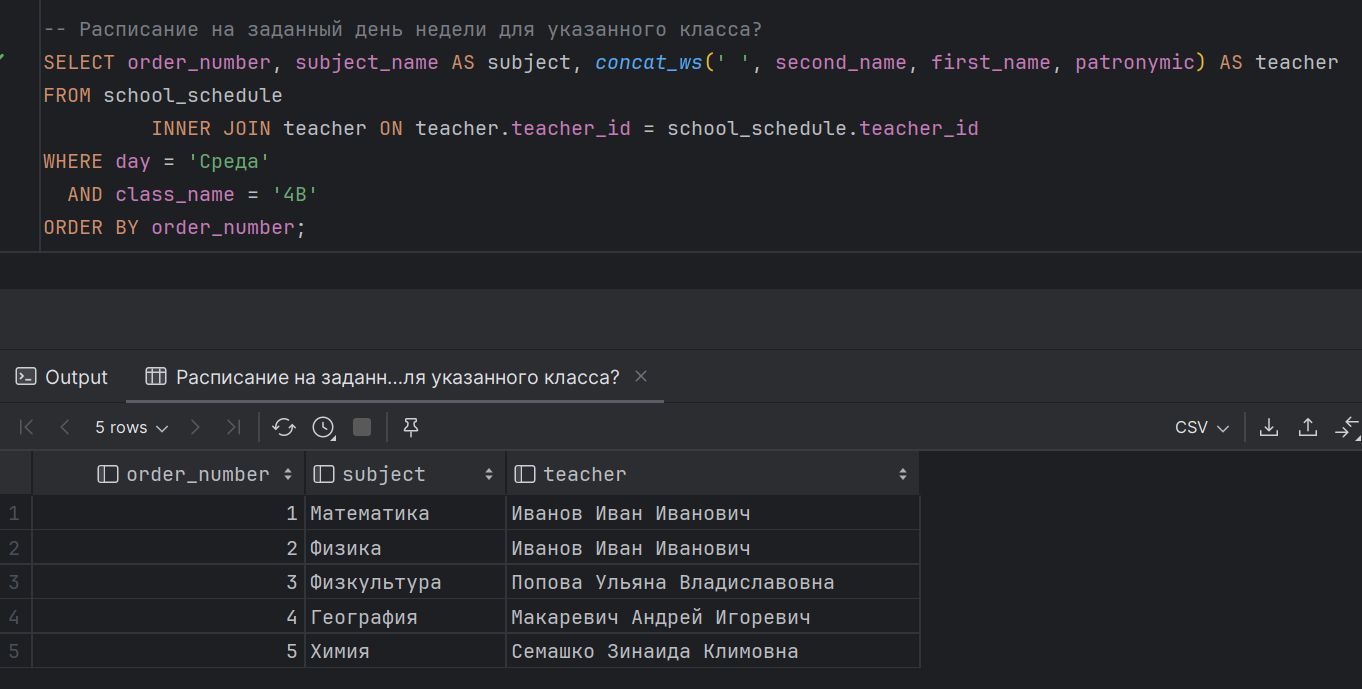


Рисунок 7. – Запрос и результат запроса.

1. Сколько учеников в указанном классе?

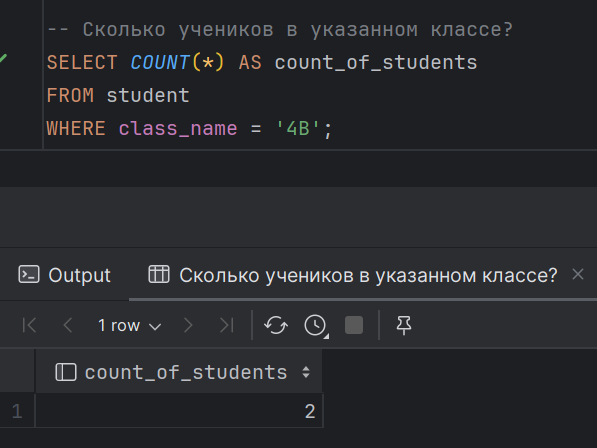


Рисунок 8. – Запрос и результат запроса.

**ПРИЛОЖЕНИЕ А**

Pull request: <https://github.com/moevm/sql-2023-1303/pull/26>

DB-Fiddle: <https://www.db-fiddle.com/f/k3THzfNTupJ6eF5bHc1jRN/3>

**ПРИЛОЖЕНИЕ Б**

Файл lab2.sql:

CREATE OR REPLACE FUNCTION random\_between(low INT, high INT)

RETURNS INT AS

$$

BEGIN

RETURN floor(random() \* (high - low + 1) + low);

END;

$$ language 'plpgsql' STRICT;

CREATE TABLE class

(

class\_name TEXT PRIMARY KEY

);

CREATE TABLE subject

(

subject\_name TEXT PRIMARY KEY

);

CREATE TABLE cabinet

(

cabinet\_number SMALLINT PRIMARY KEY

);

CREATE TABLE teacher

(

teacher\_id SMALLSERIAL PRIMARY KEY,

second\_name TEXT NOT NULL,

first\_name TEXT NOT NULL,

patronymic TEXT NOT NULL

);

CREATE TABLE student

(

student\_id SMALLSERIAL PRIMARY KEY,

class\_name TEXT,

second\_name TEXT NOT NULL,

first\_name TEXT NOT NULL,

patronymic TEXT NOT NULL,

FOREIGN KEY (class\_name) REFERENCES class (class\_name) ON DELETE SET NULL

);

CREATE TABLE grade

(

student\_id SMALLINT NOT NULL,

subject\_name TEXT NOT NULL,

value SMALLINT,

PRIMARY KEY (student\_id, subject\_name),

FOREIGN KEY (student\_id) REFERENCES student (student\_id) ON DELETE CASCADE,

FOREIGN KEY (subject\_name) REFERENCES subject (subject\_name) ON DELETE CASCADE

);

CREATE TABLE teacher\_cabinet

(

teacher\_id SMALLINT PRIMARY KEY,

cabinet\_number SMALLINT NOT NULL,

FOREIGN KEY (teacher\_id) REFERENCES teacher (teacher\_id) ON DELETE CASCADE,

FOREIGN KEY (cabinet\_number) REFERENCES cabinet (cabinet\_number) ON DELETE CASCADE

);

CREATE TABLE school\_schedule

(

subject\_schedule\_id SERIAL PRIMARY KEY,

order\_number SMALLINT NOT NULL,

day TEXT NOT NULL,

class\_name TEXT NOT NULL,

subject\_name TEXT NOT NULL,

teacher\_id SMALLINT,

cabinet\_number SMALLINT,

FOREIGN KEY (class\_name) REFERENCES class (class\_name) ON DELETE CASCADE,

FOREIGN KEY (subject\_name) REFERENCES subject (subject\_name) ON DELETE CASCADE,

FOREIGN KEY (teacher\_id) REFERENCES teacher (teacher\_id) ON DELETE SET NULL,

FOREIGN KEY (cabinet\_number) REFERENCES cabinet (cabinet\_number) ON DELETE SET NULL

);

INSERT INTO cabinet

(cabinet\_number)

VALUES (11),

(12),

(13), -- free

(14),

(15),

(16),

(17);

INSERT INTO teacher

(second\_name, first\_name, patronymic)

VALUES ('Иванов', 'Иван', 'Иванович'),

('Забудь', 'Светлана', 'Ивановна'),

('Макаревич', 'Андрей', 'Игоревич'),

('Семашко', 'Зинаида', 'Климовна'), -- free

('Попова', 'Ульяна', 'Владиславовна'),

('Вечёрко', 'Анастасия', 'Дмитриевна'),

('Ожог', 'Николай', 'Витальевич');

INSERT INTO teacher\_cabinet

(teacher\_id, cabinet\_number)

VALUES (1, 11),

(2, 12),

(3, 14),

(5, 15),

(6, 16),

(7, 17);

INSERT INTO class

(class\_name)

VALUES ('4В'),

('7А'),

('7Б'),

('9Б'),

('11А');

INSERT INTO student

(class\_name, second\_name, first\_name, patronymic)

VALUES ('4В', 'Степанов', 'Василий', 'Анатольевич'),

('4В', 'Авдеев', 'Георгий', 'Фёдорович'),

('7А', 'Виноградов', 'Анатолий', 'Алексеевич'),

('7А', 'Кузнецов', 'Илья', 'Михайлович'),

('7Б', 'Смирнов', 'Николай', 'Денисовчи'),

('7Б', 'Назарова', 'Мария', 'Михайловна'),

('9Б', 'Михеева', 'Елизавета', 'Алексеевна'),

('9Б', 'Ковалева', 'Анастасия', 'Сегреевна'),

('11А', 'Анисимов', 'Ярослав', 'Юрьевич'),

('11А', 'Нечаева', 'Светлана', 'Генадьевна');

INSERT INTO subject

(subject\_name)

VALUES ('Математика'),

('Физика'),

('Русский язык'),

('География'),

('Химия'),

('Биология'),

('Физкультура');

INSERT INTO grade

(student\_id, subject\_name)

SELECT grade\_temp.student\_id, grade\_temp.subject\_name

FROM (student

CROSS JOIN subject) AS grade\_temp;

UPDATE grade

SET value = random\_between(2, 10)

WHERE value IS NULL;

INSERT INTO school\_schedule

(order\_number, day, class\_name, subject\_name, teacher\_id, cabinet\_number)

VALUES (1, 'Понедельник', '4В', 'Математика', 1, 17),

(1, 'Вторник', '4В', 'Физика', 2, 12),

(1, 'Среда', '4В', 'Математика', 1, 17),

(2, 'Среда', '4В', 'Физика', 1, 12),

(3, 'Среда', '4В', 'Физкультура', 5, 16),

(4, 'Среда', '4В', 'География', 3, 14),

(5, 'Среда', '4В', 'Химия', 4, 12),

(1, 'Четверг', '4В', 'Русский язык', 7, 11),

(1, 'Пятница', '4В', 'Математика', 1, 17),

(1, 'Среда', '9Б', 'Математика', 1, 13);

-- Какой предмет будет в заданном классе, в заданный день недели на заданном уроке?

SELECT subject\_name AS subject

FROM school\_schedule

WHERE class\_name = '4В'

AND day = 'Среда'

AND order\_number = 2;

-- Кто из учителей преподает в заданном классе?

SELECT DISTINCT concat\_ws(' ', second\_name, first\_name, patronymic) AS teacher

FROM teacher

INNER JOIN school\_schedule ON teacher.teacher\_id = school\_schedule.teacher\_id

WHERE class\_name = '4В';

-- В каком кабинете будет 5-й урок в среду у некоторого класса?

SELECT cabinet\_number AS cabinet

FROM school\_schedule

WHERE order\_number = 5

AND day = 'Среда'

AND class\_name = '4В';

-- В каких классах преподает заданный предмет заданный учитель?

SELECT DISTINCT class\_name

FROM school\_schedule

INNER JOIN teacher ON teacher.teacher\_id = school\_schedule.teacher\_id

WHERE subject\_name = 'Математика'

AND (second\_name = 'Иванов'

AND first\_name = 'Иван'

AND patronymic = 'Иванович');

-- Расписание на заданный день недели для указанного класса?

SELECT order\_number, subject\_name AS subject, concat\_ws(' ', second\_name, first\_name, patronymic) AS teacher

FROM school\_schedule

INNER JOIN teacher ON teacher.teacher\_id = school\_schedule.teacher\_id

WHERE day = 'Среда'

AND class\_name = '4В'

ORDER BY order\_number;

-- Сколько учеников в указанном классе?

SELECT COUNT(\*) AS count\_of\_students

FROM student

WHERE class\_name = '4В';