

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

ОТЧЕТ
по лабораторной работе №2
по дисциплине «Базы данных»
Тема: Реализация базы данных в СУБД PostgreSQL

Студент гр. 1304

Басыров В.А.

Преподаватель

Заславский М.М.

Санкт-Петербург

2023

Цель работы.

Изучение реляционной базы данных на примере PostgreSQL. Написание запросов на создание таблиц, вставки данных в таблиц, а также также написание запросов к базе данных.

Задание.

Постановка задачи.

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 закрепленного 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.

Для реализации запросов к СУБД была выкачана утилита psql на Ubuntu, создана база данных school(скриншот 1), затем выполнено подключение к базе данных school(скриншот 2).

```
postgres=# CREATE DATABASE SCHOOL;
CREATE DATABASE
postgres=# \q
```

Скриншот 1. Создание базы данных.

```
vova@vova-VlvoBook-ASUSLap:~/sql-2023-1304/1304_Basyrov/lab1$ sudo -u postgres psql school
psql (12.16 (Ubuntu 12.16-0ubuntu0.20.04.1))
Type "help" for help.
```

Скриншот 2. Подключение к базе данных.

Далее был выполнен ряд команд для создания таблиц(скришот 3).

```
school=# CREATE TABLE Subject
school=# (
school(#   subject_id SERIAL NOT NULL PRIMARY KEY,
school(#   title VARCHAR(64) NOT NULL UNIQUE
school(# );
CREATE TABLE
school=#
school=#
school=# CREATE TABLE Class
school=# (
school(#   class_id SERIAL NOT NULL PRIMARY KEY,
school(#   number INT NOT NULL,
school(#   paralel VARCHAR(2) NOT NULL
school(# );
CREATE TABLE
school=#
school=# CREATE TABLE Student
school=# (
school(#   student_id SERIAL NOT NULL PRIMARY KEY,
school(#   name VARCHAR(64) NOT NULL,
school(#   surname VARCHAR(64) NOT NULL,
school(#   patronimic VARCHAR(64) NOT NULL,
school(#   class_id INTEGER,
school(#   FOREIGN KEY (class_id)
school(#     REFERENCES Class (class_id)
school(#     ON DELETE CASCADE
school(# );
CREATE TABLE
school=#
school=# CREATE TABLE Mark
school=# (
school(#   mark_id SERIAL NOT NULL PRIMARY KEY,
school(#   mark INTEGER NOT NULL,
school(#   subject_id INTEGER,
school(#   student_id INTEGER,
school(#   FOREIGN KEY (subject_id)
school(#     REFERENCES Subject (subject_id)
school(#     ON DELETE CASCADE,
school(#   FOREIGN KEY (student_id)
school(#     REFERENCES Student (student_id)
school(#     ON DELETE CASCADE
school(# );
CREATE TABLE
school=#
school=# CREATE TABLE Teacher
school=# (
school(#   teacher_id SERIAL NOT NULL PRIMARY KEY,
school(#   name VARCHAR(64) NOT NULL,
school(#   surname VARCHAR(64) NOT NULL,
school(#   patronimic VARCHAR(64) NOT NULL
school(# );
CREATE TABLE
school=# CREATE TABLE Lesson
school=# (
school(#   lesson_id SERIAL NOT NULL PRIMARY KEY,
school(#   teacher_id INTEGER,
school(#   subject_id INTEGER,
school(#   FOREIGN KEY (subject_id)
school(#     REFERENCES Subject (subject_id)
school(#     ON DELETE CASCADE,
school(#   FOREIGN KEY (teacher_id)
school(#     REFERENCES Teacher (teacher_id)
school(#     ON DELETE CASCADE
school(# );
CREATE TABLE
school=#
school=# CREATE TABLE Cabinet(
school(#   cabinet_id SERIAL NOT NULL PRIMARY KEY,
school(#   number INTEGER
school(# );
CREATE TABLE
school=#
)school=#
school=# CREATE TABLE Owner
school=# (
school(#   teacher_id INTEGER,
school(#   cabinet_id INTEGER,
school(#   FOREIGN KEY (cabinet_id)
school(#     REFERENCES Cabinet (cabinet_id)
school(#     ON DELETE CASCADE,
school(#   FOREIGN KEY (teacher_id)
school(#     REFERENCES Teacher (teacher_id)
school(#     ON DELETE CASCADE
school(# );
CREATE TABLE
school=#
school=# CREATE TABLE Schedule
school=# (
school(#   schedule_id SERIAL NOT NULL PRIMARY KEY,
school(#   lesson_id INTEGER,
school(#   class_id INTEGER,
school(#   cabinet_id INTEGER,
school(#   day VARCHAR(32) NOT NULL,
school(#   number_lesson INTEGER NOT NULL,
school(#   FOREIGN KEY (cabinet_id)
school(#     REFERENCES Cabinet (cabinet_id)
school(#     ON DELETE CASCADE,
school(#   FOREIGN KEY (class_id)
school(#     REFERENCES Class (class_id)
school(#     ON DELETE CASCADE,
school(#   FOREIGN KEY (lesson_id)
school(#     REFERENCES Lesson (lesson_id)
school(#     ON DELETE CASCADE
school(# );
CREATE TABLE
```

Скришот 3. Создания таблиц

Как видно, после каждого создания выводится «CREATE TABLE», что обозначает успешное создание таблицы.

Затем выполняется вставка в эти таблицы(скришот 4).

```
school=# INSERT INTO Subject(title)
school=# VALUES ('Русский Язык'),
school=# ('Литература'),
school=# ('География'),
school=# ('Математика'),
school=# ('Биология'),
school=# ('Химия'),
school=# ('Информатика');
INSERT 0 7
school=# INSERT INTO Class(number,parallel)
school=# VALUES ('7','a'),
school=# ('8','b'),
school=# ('9','b'),
school=# ('7','b'),
school=# ('9','b');
INSERT 0 5
school=# INSERT INTO Student (name,surname,patronimic,class_id)
school=# VALUES ('Артур','Мусаев','Илдарович',1),
school=# ('Александр','Маркуш','Евгеньевич',1),
school=# ('Михаил','Новицкий','Дмитриевич',2),
school=# ('Егор','Лобанов','Александрович',3),
school=# ('Роман','Поршнев','Александрович',4),
school=# ('Владимир','Басиров','Александрович',2),
school=# ('Дмитрий','Кривоченко','Игоревич',1),
school=# ('Даниил','Павлов','Кузиминичев',3),
school=# ('Никита','Петухов','Евгенич',5),
school=# ('Эдуард','Васильев','Федорович',4),
school=# ('Евгений','Павлов','Андреевич',3),
school=# ('Александр','Петров','Андреевич',3),
school=# ('Даниил','Сидоров','Станиславович',3);
INSERT 0 13
school=#
school=# INSERT INTO Mark(mark,student_id,subject_id)
school=# VALUES (5,1,1),
school=# (4, 1,1),
school=# (4, 2,1),
school=# (3, 2,2),
school=# (2, 3,2),
school=# (5,4,3),
school=# (3,5,4),
school=# (1,6,5),
school=# (4,2,2);
INSERT 0 9
school=#
school=#
school=# INSERT INTO Teacher(name,surname,patronimic)
school=# VALUES ('Евгений','Иванов','Иванович'),
school=# ('Александр','Пожидаев','Евгеньевич'),
school=# ('Ярослав','Коптелов','Юрьевич'),
school=# ('Владимир','Кирияничков','Александрович'),
school=# ('Светлана','Федорова','Александровна'),
school=# ('Наталья','Кузьмина','Николавная');
INSERT 0 6
school=# INSERT INTO Lesson(teacher_id,subject_id)
school=# VALUES (1,1),
school=# (2,1),
school=# (3,3),
school=# (4,2),
school=# (5,2),
school=# (6,3),
school=# (1,4),
school=# (2,5),
school=# (3,6),
school=# (4,3);
INSERT 0 10
school=#
school=# INSERT INTO Cabinet(number)
school=# VALUES (27),
school=# (30),
school=# (55),
school=# (12),
school=# (21),
school=# (29),
school=# (45),
school=# (41),
school=# (5);
INSERT 0 9
school=#
school=# INSERT INTO Owner(teacher_id,cabinet_id)
school=# VALUES (1, 1),
school=# (2, 2),
school=# (4,4),
school=# (5,5),
school=# (6,6);
INSERT 0 5
school=#
school=# INSERT INTO Schedule(lesson_id,class_id,day,number_lesson,cabinet_id)
school=# VALUES
school=# (8,1,'Понедельник',1,2),
school=# (3,2,'Понедельник',1,9),
school=# (5,3,'Понедельник',1,5),
school=# (9,4,'Понедельник',1,8),
school=# (4,5,'Понедельник',1,4),
school=# (5,1,'Понедельник',2,5),
school=# (7,2,'Понедельник',2,1),
school=# (7,3,'Понедельник',2,1),
school=# (2,4,'Понедельник',2,2),
school=# (2,5,'Понедельник',2,2),
school=# (6,1,'Понедельник',3,6),
school=# (3,2,'Понедельник',3,8),
school=# (8,3,'Понедельник',3,2),
school=# (6,4,'Понедельник',3,6),
school=# (8,5,'Понедельник',3,2),
school=# (4,1,'Понедельник',4,4),
school=# (7,2,'Понедельник',4,1),
school=# (7,3,'Понедельник',4,1),
school=# (2,4,'Понедельник',4,2),
school=# (2,5,'Понедельник',4,2),
```

Скриншот 4. Вставка в таблицы базы данных.

Как видно, операции выполняются успешно, так как после каждой вставки происходит вывод сообщения INSERT(последнее сообщение из-за большого количества записей опущено).

Реализуем каждый из запросов.

1) Выясним, какой предмет будет у 7 «б» класса в понедельник 3 уроком.
(скришот 5)

```

school=# SELECT title FROM Subject
school=# INNER JOIN Lesson USING(subject_id)
school=# INNER JOIN Schedule s USING(lesson_id)
school=# INNER JOIN Class c USING(class_id)
school=# WHERE c.number=7 AND c.paralel='6'
school=# AND s.day='Понедельник' AND s.number_lesson=3;
      title
-----
  География
(1 row)

```

Скриншот 5. Выполнение первого запроса.

2) Далее выясним, какие учителя ведут уроки у 8 «б» класса(скриншот 6).

```

school=# SELECT name,surname,patronimic FROM Teacher
          INNER JOIN Lesson USING(teacher_id)
          INNER JOIN Schedule USING(lesson_id)
          INNER JOIN Class c USING(class_id)
WHERE c.number=8 AND c.paralel='6'
GROUP BY name,surname,patronimic;
   name   | surname | patronimic
-----+-----+-----
 Александр | Пожидаев | Евгеньевич
 Владимир  | Кирьянчиков | Александрович
 Евгений   | Иванов  | Иванович
 Наталья   | Кузьмина | Николаевна
 Светлана  | Федорова | Александровна
 Ярослав   | Коптелов | Юрьевич
(6 rows)

```

Скриншот 6. Выполнение второго запроса.

3) Затем посмотрим, в каком кабинете будут занятия в среду у 8 «б» класса.(скриншот 7)

```

school=# SELECT cab.number FROM Cabinet cab
school=# INNER JOIN Schedule s USING(cabinet_id)
school=# INNER JOIN Class c USING(class_id)
school=# WHERE (c.number=8 AND c.paralel='6' AND s.day='Среда' AND s.number_lesson=5);
      number
-----
         29
(1 row)

```

Скриншот 7. Выполнение третьего запроса.

4) Далее выясним расписание на четверг у 7 «б» класса.(скриншот 8)

```

school=# WHERE (c.number=7 AND c.paralel='a' AND sch.day='Четверг');
number_lesson | title | number
-----+-----+-----
1 | География | 12
2 | География | 12
3 | Математика | 27
4 | География | 29
5 | География | 41
6 | Математика | 27
(6 rows)

```

Скриншот 8. Выполнение четвертого запроса.

5) Теперь посмотрим, у каких классов ведет «Русский Язык» Пожидаев Александр Евгеньевич.(скриншот 9)

```

school=# SELECT c.number,c.paralel FROM Class c
school=# INNER JOIN Schedule USING(class_id)
school=# INNER JOIN Lesson USING(lesson_id)
school=# INNER JOIN Teacher t USING(teacher_id)
school=# INNER JOIN Subject s USING(subject_id)
school=# WHERE t.name = 'Александр' AND t.surname='Пожидаев' AND t.patronymic='Евгеньевич'
school=# AND s.title = 'Русский Язык'
school=# GROUP BY c.number,c.paralel;
number | paralel
-----+-----
7 | б
9 | б
9 | в
(3 rows)

```

Скриншот 9. Выполнение 5 запроса

6) И в заключении выполним запрос на выяснение количество ученико в 8 «б» классе.(скриншот 10)

```

school=# SELECT count(*) FROM Student
school=# INNER JOIN Class c USING(class_id)
school=# WHERE c.paralel = 'б' AND c.number=8;
count
-----
2
(1 row)

```

Скриншот 10 Выполнение 6 запроса

Выводы.

Изучены основы работы с реляционной базой данных Postgresql, а также созданы таблицы, вставлены в них данные и выполнены запросы к базе данных.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД И ССЫЛКИ НА РАБОТУ

Lab2.sql

```
CREATE TABLE Subject
```

```
(
    subject_id SERIAL NOT NULL PRIMARY KEY,
    title VARCHAR(64) NOT NULL UNIQUE
);
```

```
CREATE TABLE Class
```

```
(
    class_id SERIAL NOT NULL PRIMARY KEY,
    number INT NOT NULL,
    paralel VARCHAR(2) NOT NULL
);
```

```
CREATE TABLE Student
```

```
(
    student_id SERIAL NOT NULL PRIMARY KEY,
    name VARCHAR(64) NOT NULL,
    surname VARCHAR(64) NOT NULL,
    patronimic VARCHAR(64) NOT NULL,
    class_id INTEGER,
    FOREIGN KEY (class_id)
        REFERENCES Class (class_id)
        ON DELETE CASCADE
);
```

```
CREATE TABLE Mark
```

```
(
    mark_id SERIAL NOT NULL PRIMARY KEY,
    mark INTEGER NOT NULL,
    subject_id INTEGER,
    student_id INTEGER,
    FOREIGN KEY (subject_id)
```

```

        REFERENCES Subject (subject_id)
        ON DELETE CASCADE,
    FOREIGN KEY (student_id)
        REFERENCES Student (student_id)
        ON DELETE CASCADE
);

CREATE TABLE Teacher
(
    teacher_id SERIAL NOT NULL PRIMARY KEY,
    name VARCHAR(64) NOT NULL,
    surname VARCHAR(64) NOT NULL,
    patronimic VARCHAR(64) NOT NULL
);

```

```

CREATE TABLE Lesson
(
    lesson_id SERIAL NOT NULL PRIMARY KEY,
    teacher_id INTEGER,
    subject_id INTEGER,
    FOREIGN KEY (subject_id)
        REFERENCES Subject (subject_id)
        ON DELETE CASCADE,
    FOREIGN KEY (teacher_id)
        REFERENCES Teacher (teacher_id)
        ON DELETE CASCADE
);

```

```

CREATE TABLE Cabinet(

```



```

        cabinet_id      SERIAL          NOT NULL PRIMARY KEY,
        number INTEGER
    );

```

CREATE TABLE Owner

```

(
    teacher_id INTEGER,
    cabinet_id INTEGER,
    FOREIGN KEY (cabinet_id)
        REFERENCES Cabinet (cabinet_id)
        ON DELETE CASCADE,
    FOREIGN KEY (teacher_id)
        REFERENCES Teacher (teacher_id)
        ON DELETE CASCADE
);

```

CREATE TABLE Schedule

```

(
    schedule_id      SERIAL          NOT NULL PRIMARY KEY,
    lesson_id INTEGER,
    class_id INTEGER,
    cabinet_id INTEGER,
    day VARCHAR(32) NOT NULL,
    number_lesson INTEGER NOT NULL,

    FOREIGN KEY (cabinet_id)
        REFERENCES Cabinet (cabinet_id)
        ON DELETE CASCADE,
    FOREIGN KEY (class_id)
        REFERENCES Class (class_id)
        ON DELETE CASCADE,
    FOREIGN KEY (lesson_id)
        REFERENCES Lesson (lesson_id)
        ON DELETE CASCADE
);

```

```

INSERT INTO Subject(title)
VALUES ( 'Русский Язык'),
        ( 'Литература'),
        ( 'География'),
        ( 'Математика'),
        ( 'Биология'),
        ( 'Химия'),
        ( 'Информатика');

INSERT INTO Class(number, paralel)
VALUES ( '7', 'а'),
        ( '8', 'б'),
        ( '9', 'в'),
        ( '7', 'б'),
        ( '9', 'б');

INSERT INTO Student (name,surname,patronimic,class_id)
VALUES ( 'Артур', 'Мусаев', 'Илдарович', 1),
        ( 'Александр', 'Маркуш', 'Евгеньевич', 1),
        ( 'Михаил', 'Новицкий', 'Дмитриевич', 2),
        ( 'Егор', 'Лобанов', 'Александрович', 3),
        ( 'Роман', 'Поршнев', 'Александрович', 4),
        ( 'Владимир', 'Басыров', 'Александрович', 2),
        ( 'Дмитрий', 'Кривоченко', 'Игоревич', 1),
        ( 'Даниил', 'Павлов', 'Кузиминичев', 3),
        ( 'Никита', 'Петухов', 'Евгенич', 5),
        ( 'Эдуард', 'Васильев', 'Федорович', 4),
        ( 'Евгений', 'Павлов', 'Андреевич', 3),
        ( 'Александр', 'Петров', 'Андреевич', 3),
        ( 'Даниил', 'Сидоров', 'Станиславович', 3);

INSERT INTO Mark(mark, student_id, subject_id)
VALUES (5, 1, 1),
        (4, 1, 1),
        (4, 2, 1),
        (3, 2, 2),
        (2, 3, 2),
        (5, 4, 3),
        (3, 5, 4),
        (1, 6, 5),

```

```
(4,2,2);
```

```
INSERT INTO Teacher(name,surname,patronimic)
VALUES ('Евгений','Иванов','Иванович'),
        ('Александр','Пожидаев','Евгеньевич'),
        ('Ярослав','Коптелов','Юрьевич'),
        ('Владимир','Кирияичиков','Александрович'),
        ('Светлана','Федорова','Александровна'),
        ('Наталья','Кузьмина','Николавная');
```

```
INSERT INTO Lesson(teacher_id,subject_id)
VALUES (1,1),
        (2,1),
        (3,3),
        (4,2),
        (5,2),
        (6,3),
        (1,4),
        (2,5),
        (3,6),
        (4,3);
```

```
INSERT INTO Cabinet(number)
VALUES (27),
        (30),
        (55),
        (12),
        (21),
        (29),
        (45),
        (41),
        (5);
```

```
INSERT INTO Owner(teacher_id,cabinet_id)
VALUES (1, 1),
        (2, 2),
        (4,4),
```

(5,5),
(6,6);

INSERT

INTO

Schedule(lesson_id,class_id,day,number_lesson,cabinet_id)

VALUES

(8,1, 'Понедельник' ,1,2),
(3,2, 'Понедельник' ,1,9),
(5,3, 'Понедельник' ,1,5),
(9,4, 'Понедельник' ,1,8),
(4,5, 'Понедельник' ,1,4),
(5,1, 'Понедельник' ,2,5),
(7,2, 'Понедельник' ,2,1),
(7,3, 'Понедельник' ,2,1),
(2,4, 'Понедельник' ,2,2),
(2,5, 'Понедельник' ,2,2),
(6,1, 'Понедельник' ,3,6),
(3,2, 'Понедельник' ,3,8),
(8,3, 'Понедельник' ,3,2),
(6,4, 'Понедельник' ,3,6),
(8,5, 'Понедельник' ,3,2),
(4,1, 'Понедельник' ,4,4),
(7,2, 'Понедельник' ,4,1),
(7,3, 'Понедельник' ,4,1),
(2,4, 'Понедельник' ,4,2),
(2,5, 'Понедельник' ,4,2),
(4,1, 'Понедельник' ,5,4),
(3,2, 'Понедельник' ,5,7),
(7,3, 'Понедельник' ,5,1),
(4,4, 'Понедельник' ,5,4),
(1,5, 'Понедельник' ,5,1),
(3,1, 'Понедельник' ,6,7),
(1,2, 'Понедельник' ,6,1),
(4,3, 'Понедельник' ,6,4),
(4,4, 'Понедельник' ,6,4),
(7,5, 'Понедельник' ,6,1),
(8,1, 'Вторник' ,1,2),
(8,2, 'Вторник' ,1,2),

(10, 3, 'Вторник', 1, 4),
(4, 4, 'Вторник', 1, 4),
(5, 5, 'Вторник', 1, 5),
(9, 1, 'Вторник', 2, 7),
(5, 2, 'Вторник', 2, 5),
(3, 3, 'Вторник', 2, 8),
(10, 4, 'Вторник', 2, 4),
(4, 5, 'Вторник', 2, 4),
(1, 1, 'Вторник', 3, 1),
(7, 2, 'Вторник', 3, 1),
(7, 3, 'Вторник', 3, 1),
(4, 4, 'Вторник', 3, 4),
(9, 5, 'Вторник', 3, 8),
(3, 1, 'Вторник', 4, 9),
(4, 2, 'Вторник', 4, 4),
(9, 3, 'Вторник', 4, 9),
(10, 4, 'Вторник', 4, 4),
(5, 5, 'Вторник', 4, 5),
(10, 1, 'Вторник', 5, 4),
(4, 2, 'Вторник', 5, 4),
(1, 3, 'Вторник', 5, 1),
(7, 4, 'Вторник', 5, 1),
(4, 5, 'Вторник', 5, 4),
(3, 1, 'Вторник', 6, 9),
(7, 2, 'Вторник', 6, 1),
(2, 3, 'Вторник', 6, 2),
(10, 4, 'Вторник', 6, 4),
(9, 5, 'Вторник', 6, 9),
(4, 1, 'Среда', 1, 4),
(5, 2, 'Среда', 1, 5),
(2, 3, 'Среда', 1, 2),
(2, 4, 'Среда', 1, 2),
(2, 5, 'Среда', 1, 2),
(8, 1, 'Среда', 2, 2),
(1, 2, 'Среда', 2, 1),
(3, 3, 'Среда', 2, 7),
(2, 4, 'Среда', 2, 2),
(2, 5, 'Среда', 2, 2),
(7, 1, 'Среда', 3, 1),

(8, 2, 'Среда', 3, 2),
(2, 3, 'Среда', 3, 2),
(10, 4, 'Среда', 3, 4),
(7, 5, 'Среда', 3, 1),
(10, 1, 'Среда', 4, 4),
(5, 2, 'Среда', 4, 5),
(3, 3, 'Среда', 4, 8),
(8, 4, 'Среда', 4, 2),
(4, 5, 'Среда', 4, 4),
(8, 1, 'Среда', 5, 2),
(6, 2, 'Среда', 5, 6),
(5, 3, 'Среда', 5, 5),
(9, 4, 'Среда', 5, 7),
(2, 5, 'Среда', 5, 2),
(1, 1, 'Среда', 6, 1),
(9, 2, 'Среда', 6, 9),
(7, 3, 'Среда', 6, 1),
(2, 4, 'Среда', 6, 2),
(2, 5, 'Среда', 6, 2),
(10, 1, 'Четверг', 1, 4),
(3, 2, 'Четверг', 1, 8),
(10, 3, 'Четверг', 1, 4),
(10, 4, 'Четверг', 1, 4),
(3, 5, 'Четверг', 1, 8),
(10, 1, 'Четверг', 2, 4),
(10, 2, 'Четверг', 2, 4),
(4, 3, 'Четверг', 2, 4),
(8, 4, 'Четверг', 2, 2),
(4, 5, 'Четверг', 2, 4),
(7, 1, 'Четверг', 3, 1),
(9, 2, 'Четверг', 3, 7),
(5, 3, 'Четверг', 3, 5),
(2, 4, 'Четверг', 3, 2),
(3, 5, 'Четверг', 3, 8),
(6, 1, 'Четверг', 4, 6),
(5, 2, 'Четверг', 4, 5),
(1, 3, 'Четверг', 4, 1),
(5, 4, 'Четверг', 4, 5),
(1, 5, 'Четверг', 4, 1),

(3,1, 'Четверг', 5,8),
(5,2, 'Четверг', 5,5),
(3,3, 'Четверг', 5,8),
(3,4, 'Четверг', 5,7),
(9,5, 'Четверг', 5,7),
(7,1, 'Четверг', 6,1),
(4,2, 'Четверг', 6,4),
(1,3, 'Четверг', 6,1),
(10,4, 'Четверг', 6,4),
(2,5, 'Четверг', 6,2),
(7,1, 'Пятница', 1,1),
(3,2, 'Пятница', 1,9),
(2,3, 'Пятница', 1,2),
(7,4, 'Пятница', 1,1),
(6,5, 'Пятница', 1,6),
(7,1, 'Пятница', 2,1),
(5,2, 'Пятница', 2,5),
(9,3, 'Пятница', 2,7),
(5,4, 'Пятница', 2,5),
(8,5, 'Пятница', 2,2),
(9,1, 'Пятница', 3,7),
(7,2, 'Пятница', 3,1),
(6,3, 'Пятница', 3,6),
(4,4, 'Пятница', 3,4),
(3,5, 'Пятница', 3,8),
(9,1, 'Пятница', 4,8),
(5,2, 'Пятница', 4,5),
(7,3, 'Пятница', 4,1),
(8,4, 'Пятница', 4,2),
(3,5, 'Пятница', 4,7),
(7,1, 'Пятница', 5,1),
(1,2, 'Пятница', 5,1),
(4,3, 'Пятница', 5,4),
(4,4, 'Пятница', 5,4),
(8,5, 'Пятница', 5,2),
(10,1, 'Пятница', 6,4),
(4,2, 'Пятница', 6,4),
(6,3, 'Пятница', 6,6),
(7,4, 'Пятница', 6,1),

```
(10,5, 'Пятница',6,4);
```

```
SELECT title FROM Subject
      INNER JOIN Lesson USING(subject_id)
      INNER JOIN Schedule s USING(lesson_id)
      INNER JOIN Class c USING(class_id)
WHERE c.number=7 AND c.paralel='б'
AND s.day='Понедельник' AND s.number_lesson=3;
```

```
SELECT name,surname,patronimic FROM Teacher
      INNER JOIN Lesson USING(teacher_id)
      INNER JOIN Schedule USING(lesson_id)
      INNER JOIN Class c USING(class_id)
WHERE c.number=9 AND c.paralel='в'
GROUP BY name,surname,patronimic;
```

```
SELECT cab.number FROM Cabinet cab
      INNER JOIN Schedule s USING(cabinet_id)
      INNER JOIN Class c USING(class_id)
WHERE (c.number=8 AND c.paralel='б' AND s.day='Среда' AND
s.number_lesson=5);
```

```
SELECT sch.number_lesson,sub.title,cab.number FROM Schedule sch
      INNER JOIN Class c USING(class_id)
      INNER JOIN Cabinet cab USING(cabinet_id)
      INNER JOIN Lesson USING(lesson_id)
      INNER JOIN Subject sub USING(subject_id)
WHERE (c.number=7 AND c.paralel='а' AND sch.day='Четверг')
```



```
SELECT c.number,c.paralel FROM Class c
      INNER JOIN Schedule USING(class_id)
      INNER JOIN Lesson USING(lesson_id)
      INNER JOIN Teacher t USING(teacher_id)
      INNER JOIN Subject s USING(subject_id)
      WHERE t.name = 'Александр' AND t.surname='Пожидаев' AND
t.patronimic='Евгеньевич'
      AND s.title = 'Русский Язык'
GROUP BY c.number,c.paralel;
```

```
SELECT count(*) FROM Student
      INNER JOIN Class c USING(class_id)
      WHERE c.paralel = '6' AND c.number=8;
```

Ссылка на www.db-fiddle.com

<https://www.db-fiddle.com/f/d3CzevXGZg1eAGRCXVosyu/6>

Ссылка на pull request

<https://github.com/moevm/sql-2023-1304/pull/41>