Московский Авиационный Институт

(Национальный Исследовательский Университет)

Институт информационных технологий и прикладной математики

Кафедра вычислительной математики и программирования

**Лабораторная работа по курсу**

**«Базы данных»**

**Тема: Расписание занятий**

Выполнили: Белоносов К. А.,

Шевлякова С. С., Армишев К. К.

Группа: М8О-308Б-21

Преподаватель: Сеченых П.А.

Дата: \_\_\_\_\_\_\_\_\_\_\_

Оценка: \_\_\_\_\_\_\_\_\_\_\_

Подпись: \_\_\_\_\_\_\_\_\_\_\_

Москва, 2023

1. **Описание таблиц и схема данных**

Таблица learn\_day:

* learn\_day\_id – идентификатор учебного дня (первичный ключ)
* datetime – дата в формате ДД.ММ.ГГГГ



Таблица teacher:

* teacher\_id – идентификатор преподавателя (первичный ключ)
* name – инициалы (имя, отчество) преподавателя
* surname – фамилия преподавателя
* degree – научная степень
* department – кафедра
* experience – стаж работы



Таблица subject:

* subject\_id – идентификатор предмета (первичный ключ)
* name – название предмета

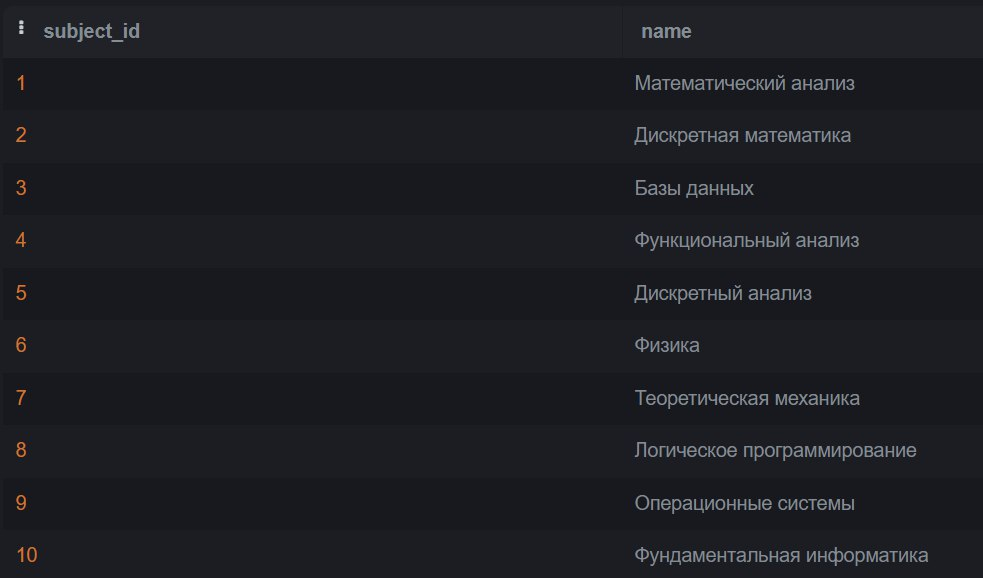


Таблица lesson:

* lesson\_id – идентификатор занятия (первичный ключ)
* date\_id – идентификатор даты проведения занятия (вторичный ключ)
* teacher\_id – идентификатор преподавателя, который ведет занятие (вторичный ключ)
* subject\_id – идентификатор предмета (вторичный ключ)
* class\_id – идентификатор аудитории (вторичный ключ)
* number – порядковый номер пары
* type – вид занятия (семинар, лабораторная работа, лекция)

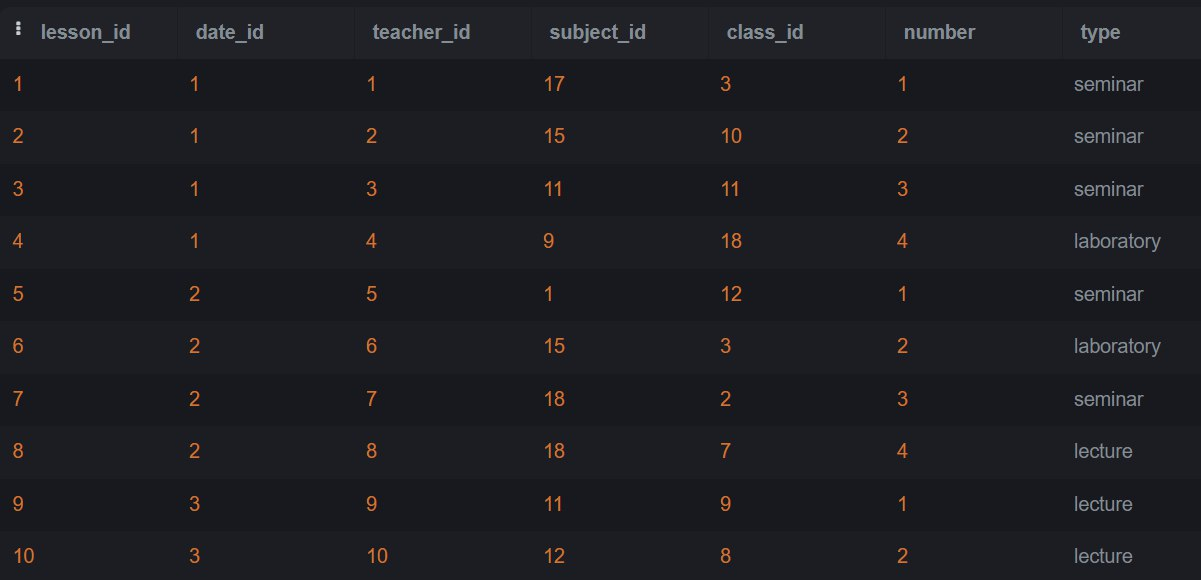


Таблица classroom:

* class\_id – идентификатор аудитории (первичный ключ)
* type – предназначение аудитории (семинарская, лекционная, лаборатория, компьютерный класс)
* capacity – вместительность аудитории
* corpus – корпус
* number – номер аудитории

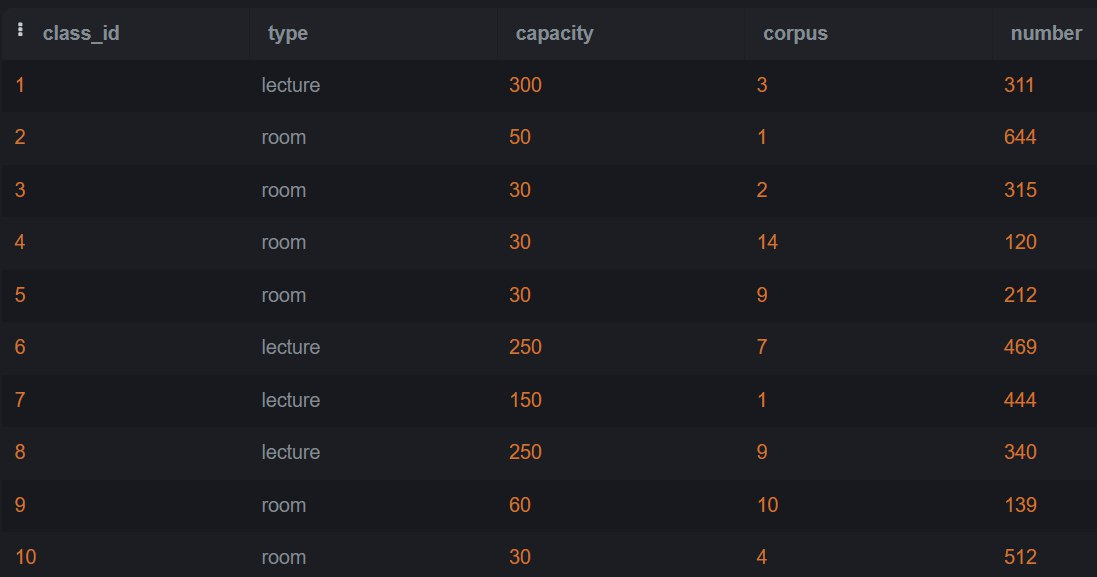


Таблица lesson\_group:

* lesson\_group\_id – идентификатор связи занятия и группы (первичный ключ)
* lesson\_id – идентификатор занятия (вторичный ключ)
* group\_id – идентификатор группы (вторичный ключ)

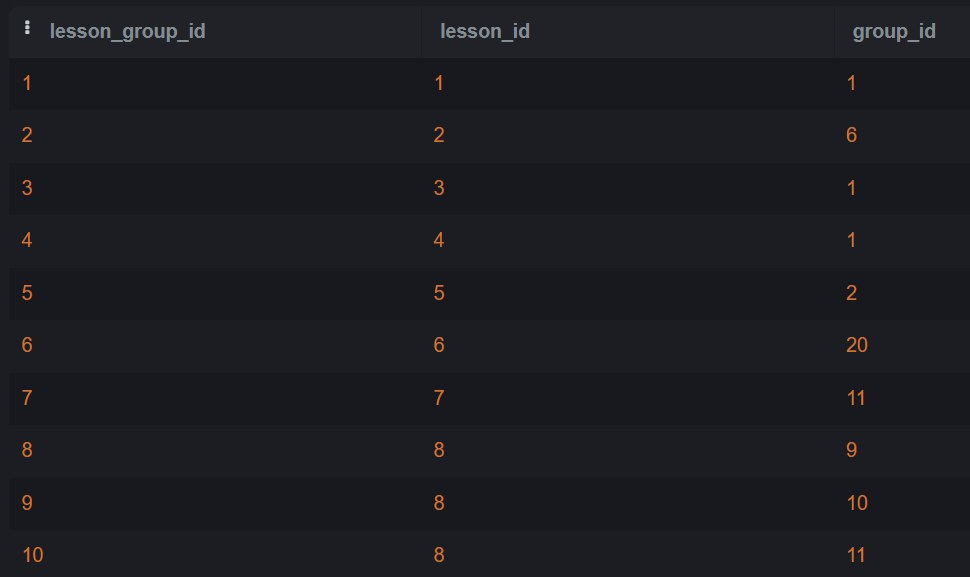


Таблица student\_group:

* student\_group\_id – идентификатор студенческой группы (первичный ключ)
* course – курс
* start\_year – год начала обучения
* speciality – специальность
* size – размер группы
* degree – уровень образования (бакалавриат, магистратура, специалитет, аспирантура)

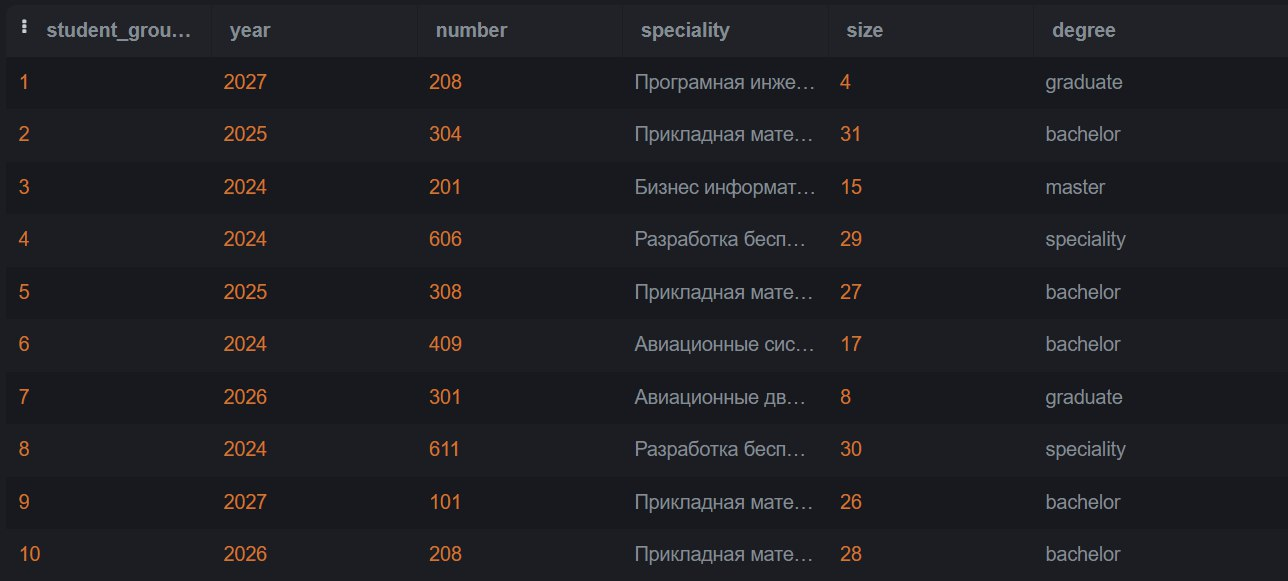
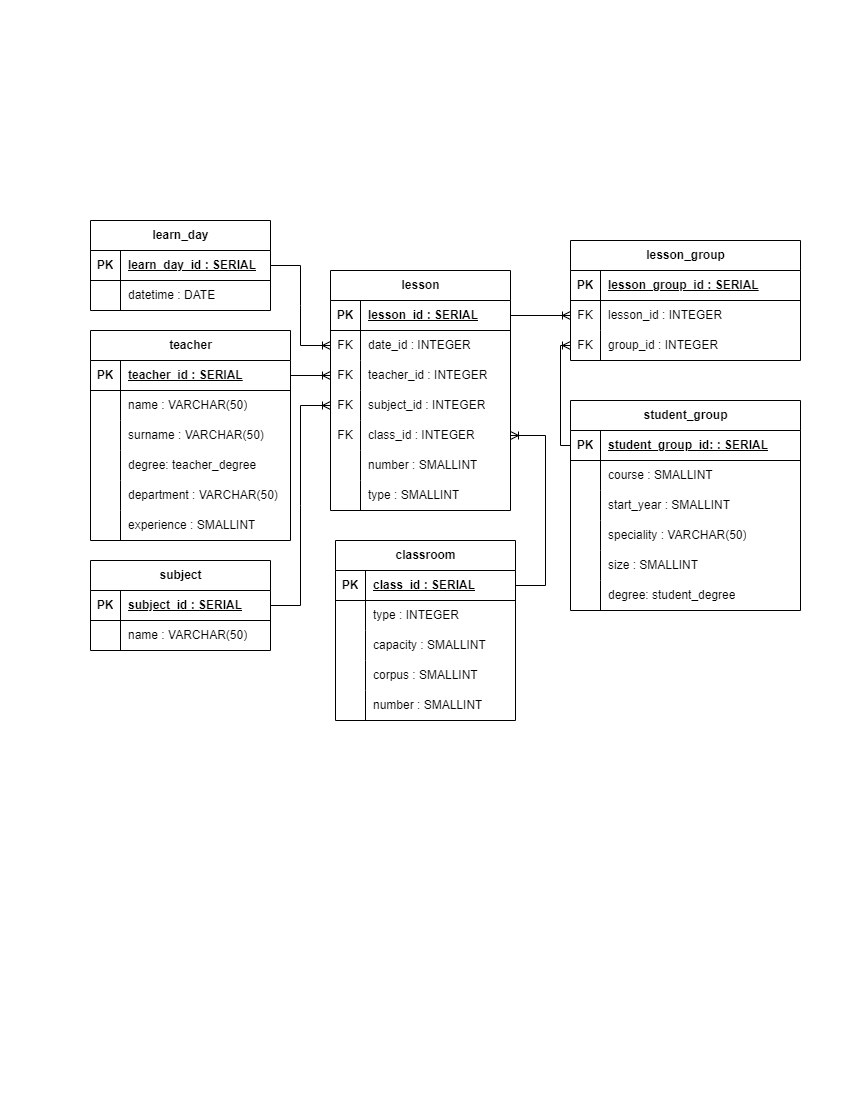


Схема базы данных:

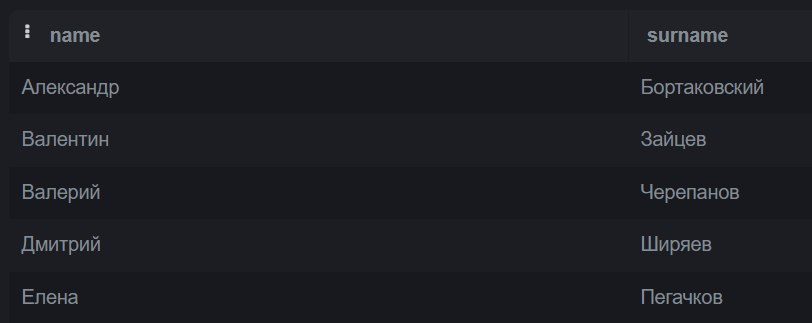


Код для генерации БД в Приложении(1).

1. **Запросы**

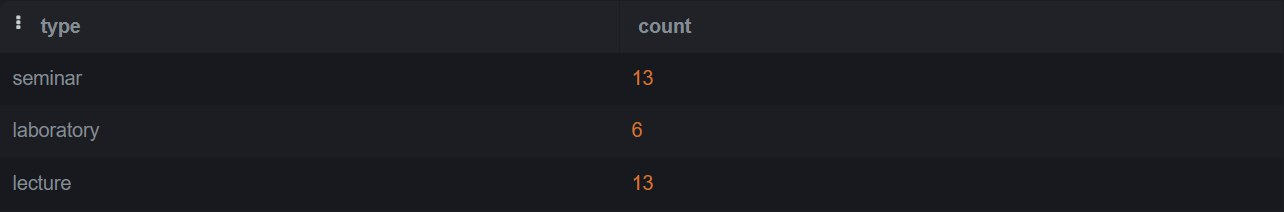
* Вывести имена и фамилии всех учителей, у которых опыт работы не менее 10 лет и ученая степень – профессор, вывод отсортирован по имени:

SELECT "name", surname FROM teacher WHERE experience >= 10 and "degree" = 'professor' ORDER BY "name";



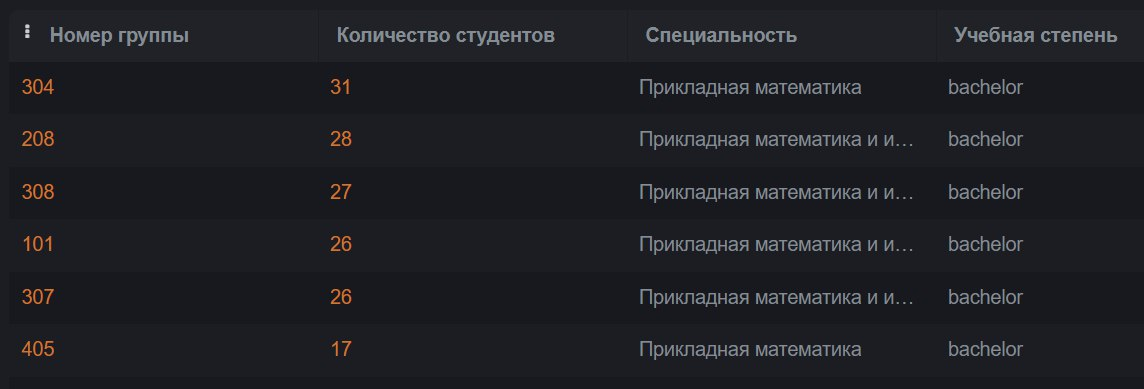
* Вывести количество занятий каждого типа:

SELECT "type", COUNT("type") FROM lesson GROUP BY "type";



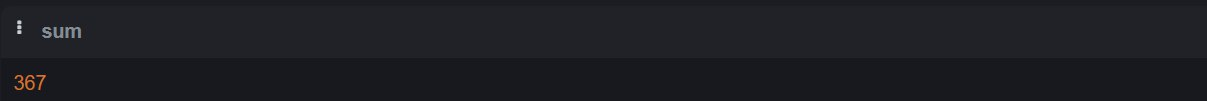
* Выводит именованные поля «Номер группы», «Количество студентов», «Специальность» и «Учебную степень» всех групп, в названии специальности которых есть слово «Прикладная», вывод отсортирован по убыванию размера групп:

SELECT "number" AS "Номер группы", "size" AS "Количество студентов", speciality AS Специальность, "degree" AS "Учебная степень" FROM student\_group WHERE speciality LIKE 'Прикладная%' ORDER BY "size" DESC;



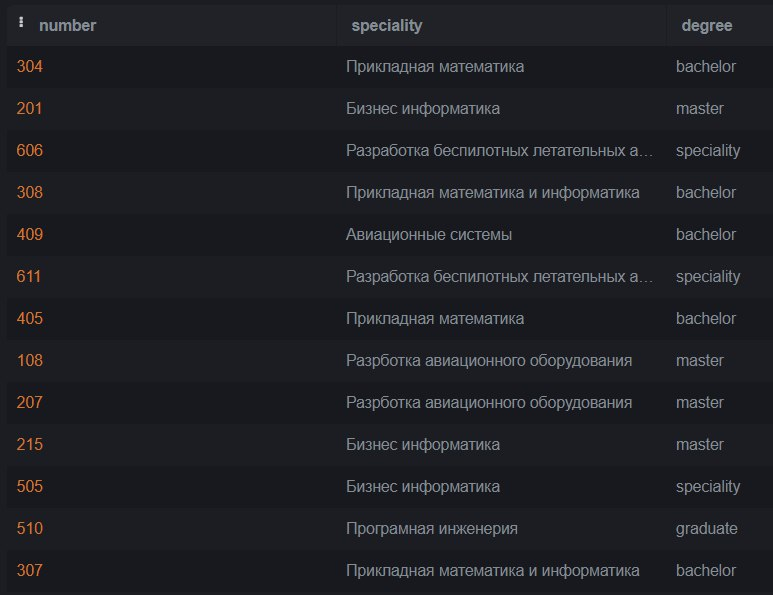
* Выводит суммарный размер всех групп:

SELECT SUM("size") FROM student\_group;



* Выводит номера групп, специальность и уровень образования, в года обучения которых студентов во всех группах было больше 100:

SELECT "number", speciality, "degree" FROM student\_group WHERE "year" IN (SELECT "year" FROM student\_group GROUP BY "year" HAVING SUM("size") > 100);



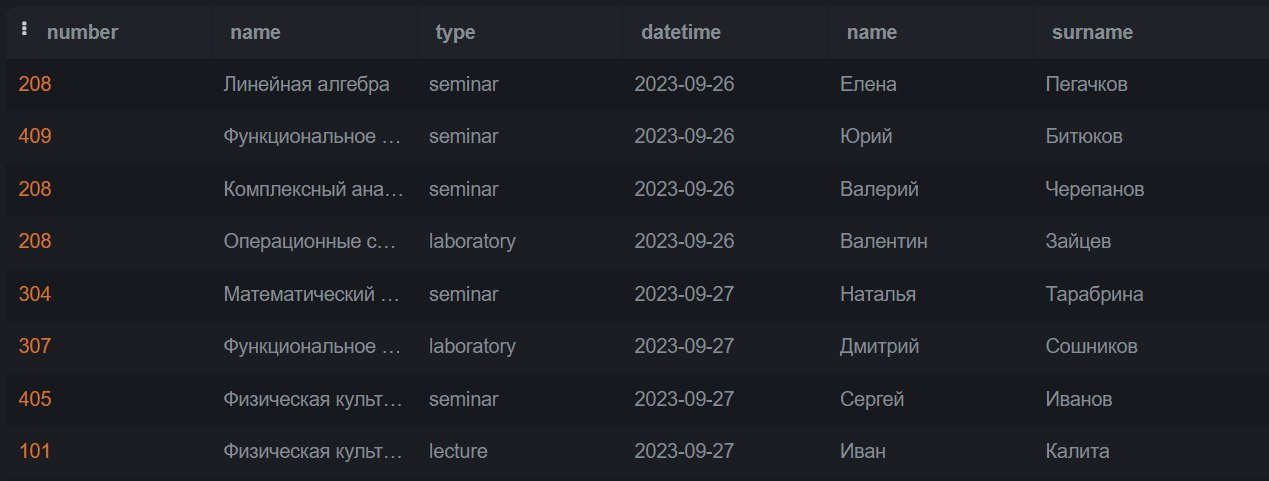
* Выводит имена, фамилии и опыт работы учителей, опыт которых меньше среднего опыта работы всех учителей:

SELECT "name", surname, experience FROM teacher WHERE experience < ALL (SELECT AVG(experience) FROM teacher);



* Соединяет все таблицы в одну большую таблицу:

SELECT student\_group."number", subject."name", "type", datetime, teacher.name, teacher.surname FROM student\_group INNER JOIN lesson\_group ON lesson\_group.group\_id = student\_group.student\_group\_id INNER JOIN lesson ON lesson\_group.lesson\_id = lesson.lesson\_id INNER JOIN learn\_day ON learn\_day.learn\_day\_id = lesson.date\_id INNER JOIN subject ON subject.subject\_id = lesson.subject\_id INNER JOIN teacher ON teacher.teacher\_id = lesson.teacher\_id;



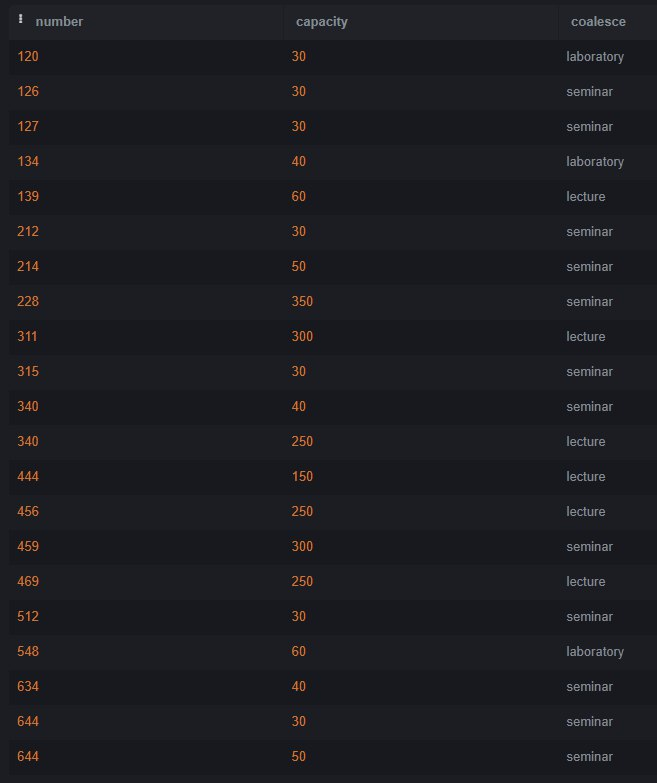
* Выводит имена, фамилии, опыт работы и ученую степень учителей, если опыт работы не менее 20 лет, то степень «Доктор», если опыт не менее 10 лет, то степень «Кандидат», иначе степень «Не имеет ученой степени», вывод отсортирован по возрастанию опыта работы:

SELECT "name", surname, experience, CASE WHEN experience >= 20 THEN 'Доктор' WHEN experience >= 10 THEN 'Кандидат' ELSE 'Не имеет ученой степени' END AS science\_degree FROM teacher ORDER BY experience;



* Выводит номер, вместительность аудитории и тип занятия, если в аудитории не проводилось еще ни одно занятие, то ставится по умолчанию семинар, вывод отсортирован по возрастанию номера аудитории:

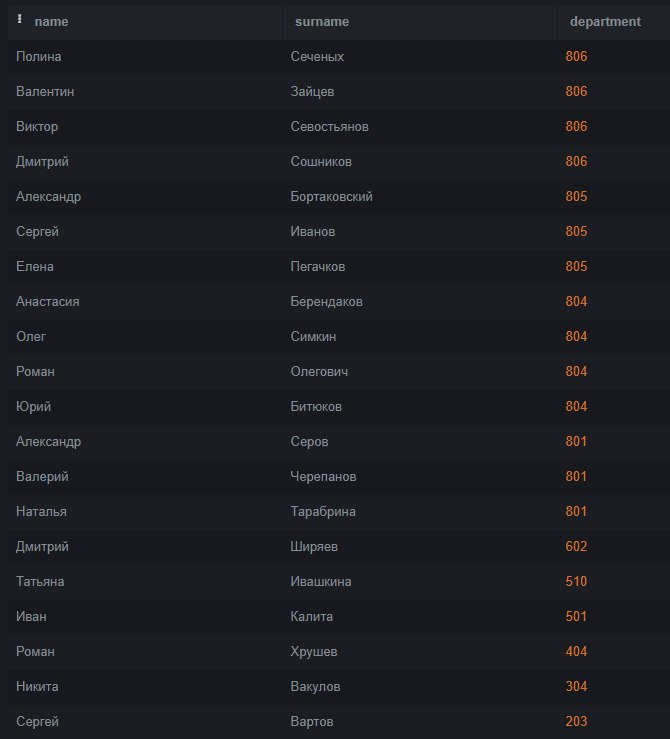
SELECT "number", capacity, COALESCE((SELECT "type" FROM lesson WHERE lesson.class\_id = classroom.class\_id LIMIT 1), 'seminar') FROM classroom ORDER BY "number";



* Всех преподавателей, у которых кафедра начинается с цифры 9, заменяет кафедру на 801. Чтобы показать изменения, второй запрос выводит имя, фамилию и кафедру преподавателя по убывания номера кафедры:

UPDATE teacher SET department = 801 WHERE CAST(department AS TEXT) LIKE '9\_\_';

SELECT "name", surname, department FROM teacher ORDER BY department DESC;



* Этот запрос выводит имя преподавателя, у которого есть хотя бы одно занятие в расписании:

SELECT "name"

FROM teacher

WHERE teacher\_id = ANY (SELECT DISTINCT teacher\_id FROM lesson);

* Этот запрос вернет список имен и фамилий преподавателей с опытом, превышающим опыт хотя бы одного профессора в базе данных.

SELECT name, surname

FROM teacher

WHERE experience > SOME (SELECT experience FROM teacher WHERE "degree" = 'professor');



* Этот запрос выводит номера студенческих групп, у которых есть занятия по предмету "Базы данных":

SELECT "number"

FROM student\_group sg

WHERE EXISTS (

SELECT 1

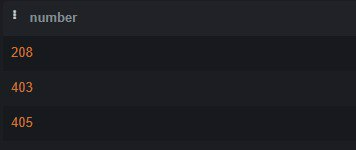
FROM lesson\_group lg

JOIN lesson l ON lg.lesson\_id = l.lesson\_id

WHERE sg.student\_group\_id = lg.group\_id

AND l.subject\_id = (SELECT subject\_id FROM subject WHERE "name" = 'Базы данных')

);



* Этот запрос объединяет занятия по предметам "Базы данных" и "Дискретная математика" и выводит их название, тип и номер:

SELECT "name", "type", "number"

FROM lesson l

JOIN subject s ON l.subject\_id = s.subject\_id

WHERE s."name" = 'Базы данных'

UNION

SELECT "name", "type", "number"

FROM lesson l

JOIN subject s ON l.subject\_id = s.subject\_id

WHERE s."name" = 'Дискретная математика';



* Этот запрос находит пересечение предметов, название которых “Линйеная алгебра”, и предметов, которые преподает Елена Пегачкова:

SELECT class\_id, "number", "type"

FROM lesson

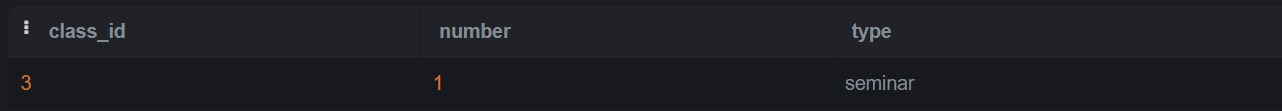
WHERE subject\_id = (SELECT subject\_id FROM subject WHERE "name" = 'Линейная алгебра')

INTERSECT

SELECT class\_id, "number", "type"

FROM lesson

WHERE teacher\_id = (SELECT teacher\_id FROM teacher WHERE "name" = 'Елена' AND surname = 'Пегачков');



* Этот запрос выводит имена и фамилии преподавателей, которые не проводят лекции в расписании:

SELECT "name", "surname"

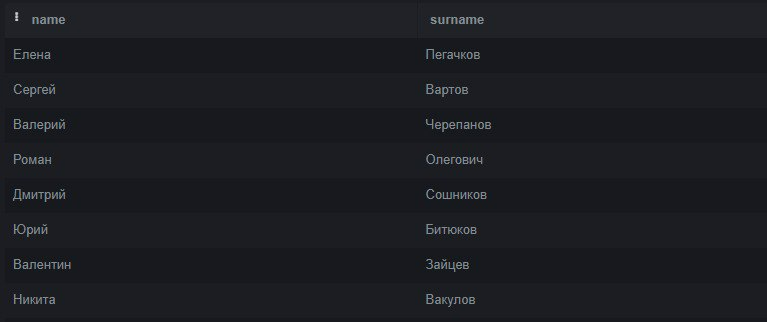
FROM teacher

EXCEPT

SELECT "name", "surname"

FROM teacher

WHERE teacher\_id = ANY (SELECT DISTINCT teacher\_id FROM lesson WHERE "type" = 'lecture');



**Приложение 1**

Создание таблицы:

CREATE TABLE teacher (

teacher\_id SERIAL PRIMARY kEY,

"name" VARCHAR(50),

surname VARCHAR(50),

"degree" degree\_teacher,

department VARCHAR(50),

experience SMALLINT

);

CREATE TABLE student\_group (

student\_group\_id SERIAL PRIMARY kEY,

"year" SMALLINT,

"number" SMALLINT,

speciality VARCHAR(50),

"size" SMALLINT,

"degree" degree\_group

);

CREATE TABLE classroom (

class\_id SERIAL PRIMARY kEY,

"type" audience\_type,

capacity SMALLINT,

corpus SMALLINT,

"number" SMALLINT

);

CREATE TABLE subject (

subject\_id SERIAL PRIMARY kEY,

"name" VARCHAR(50)

);

CREATE TABLE learn\_day (

learn\_day\_id SERIAL PRIMARY kEY,

datetime DATE

);

CREATE TABLE lesson (

lesson\_id SERIAL PRIMARY KEY,

date\_id INTEGER REFERENCES learn\_day(learn\_day\_id),

teacher\_id INTEGER REFERENCES teacher(teacher\_id),

subject\_id INTEGER REFERENCES subject(subject\_id),

class\_id INTEGER REFERENCES classroom(class\_id),

"number" SMALLINT,

"type" lesson\_type

);

CREATE TABLE lesson\_group (

lesson\_group\_id SERIAL PRIMARY kEY,

lesson\_id INTEGER REFERENCES lesson(lesson\_id),

group\_id INTEGER REFERENCES student\_group(student\_group\_id)

);

Создание специальных типов данных:

CREAte TYPE degree\_group AS ENUM('graduate','speciality','master', 'bachelor');

CREATE TYPE degree\_teacher AS ENUM('professor', 'assistant', 'none');

CREATE TYPE audience\_type AS ENUM('lecture','room');

CREATE TYPE lesson\_type AS ENUM('lecture', 'seminar', 'laboratory');

Заполнение таблицы занятия:

INSERT INTO lesson(date\_id, teacher\_id, subject\_id, class\_id, "number", "type")

VALUES

(1, 1, 17, 3, 1, 'seminar'),

(1, 2, 15, 10, 2, 'seminar'),

(1, 3, 11, 11, 3, 'seminar'),

(1, 4, 9, 18, 4, 'laboratory'),

(2, 5, 1, 12, 1, 'seminar'),

(2, 6, 15, 3, 2, 'laboratory'),

(2, 7, 18, 2, 3, 'seminar'),

(2, 8, 18, 7, 4, 'lecture'),

(3, 9, 11, 9, 1, 'lecture'),

(3, 10, 12, 8, 2, 'lecture'),

(3, 11, 13, 12, 3, 'seminar'),

(3, 12, 6, 3, 4, 'seminar'),

(4, 13, 8, 4, 1, 'laboratory'),

(4, 14, 9, 6, 2, 'lecture'),

(4, 15, 9, 1, 3, 'lecture'),

(4, 16, 3, 10, 4, 'laboratory'),

(5, 17, 2, 20, 1, 'seminar'),

(5, 18, 7, 21, 2, 'lecture'),

(5, 19, 8, 6, 3, 'lecture'),

(5, 20, 9, 7, 4, 'lecture'),

(6, 11, 17, 8, 1, 'lecture'),

(6, 12, 13, 10, 2, 'seminar'),

(6, 15, 15, 14, 3, 'laboratory'),

(6, 1, 12, 15, 4, 'seminar'),

(7, 10, 11, 3, 1, 'seminar'),

(7, 7, 4, 8, 2, 'lecture'),

(7, 9, 4, 9, 3, 'seminar'),

(7, 4, 3, 4, 4, 'laboratory'),

(8, 2, 3, 2, 1, 'seminar'),

(8, 5, 6, 6, 2, 'lecture'),

(8, 17, 7, 7, 3, 'lecture'),

(8, 19, 10, 1, 4, 'lecture');

Заполнение таблиц предметов, аудиторий, студенческих групп, учебных дней и преподавателей:

INSERT INTO subject("name")

VALUES

('Математический анализ'),

('Дискретная математика'),

('Базы данных'),

('Функциональный анализ'),

('Дискретный анализ'),

('Физика'),

('Теоретическая механика'),

('Логическое программирование'),

('Операционные системы'),

('Фундаментальная информатика'),

('Комплексный анализ'),

('Вычислительная техника'),

('Искуственный интеллект'),

('Компьютерная графика'),

('Функциональное программирование'),

('Численные методы'),

('Линейная алгебра'),

('Физическая культура');

INSERT INTO classroom("type", capacity, corpus, "number")

VALUES

('lecture', 300, 3, 311),

('room', 50, 1, 644),

('room', 30, 2, 315),

('room', 30, 14, 120),

('room', 30, 9, 212),

('lecture', 250, 7, 469),

('lecture', 150, 1, 444),

('lecture', 250, 9, 340),

('room', 60, 10, 139),

('room', 30, 4, 512),

('room', 40, 5, 340),

('room', 50, 11, 214),

('room', 30, 6, 644),

('room', 40, 8, 134),

('room', 30, 1, 127),

('lecture', 350, 1, 228),

('lecture', 300, 4, 459),

('room', 60, 2, 548),

('room', 30, 5, 126),

('room', 40, 7, 634),

('lecture', 250, 10, 456);

INSERT INTO student\_group("year", "number", speciality, "size", "degree")

VALUEs

(2027, 208, 'Програмная инженерия', 4, 'graduate'),

(2025, 304, 'Прикладная математика', 31, 'bachelor'),

(2024, 201, 'Бизнес информатика', 15, 'master'),

(2024, 606, 'Разработка беспилотных летательных аппаратов', 29, 'speciality'),

(2025, 308, 'Прикладная математика и информатика', 27, 'bachelor'),

(2024, 409, 'Авиационные системы', 17,'bachelor'),

(2026, 301, 'Авиационные двигатели', 8,'graduate'),

(2024, 611, 'Разработка беспилотных летательных аппаратов', 30, 'speciality'),

(2027, 101, 'Прикладная математика и информатика', 26, 'bachelor'),

(2026, 208, 'Прикладная математика и информатика', 28, 'bachelor'),

(2024, 405, 'Прикладная математика', 17, 'bachelor'),

(2025, 108, 'Разрботка авиационного оборудования', 11,'master'),

(2024, 207, 'Разрботка авиационного оборудования', 13,'master'),

(2024, 215, 'Бизнес информатика', 17, 'master'),

(2026, 403, 'Иностранные языки', 23,'speciality'),

(2025, 505, 'Бизнес информатика', 6,'speciality'),

(2028, 209, 'Разработка беспилотных летательных аппаратов', 32,'speciality'),

(2026, 311, 'Биоинформатика', 2, 'graduate'),

(2024, 510, 'Програмная инженерия', 5, 'graduate'),

(2025, 307, 'Прикладная математика и информатика', 26, 'bachelor');

INSERT INTO learn\_day(datetime)

Values

('09/26/2023'),

('09/27/2023'),

('09/28/2023'),

('09/29/2023'),

('09/30/2023'),

('10/01/2023'),

('10/02/2023'),

('10/03/2023'),

('10/04/2023'),

('10/05/2023'),

('10/06/2023'),

('10/07/2023'),

('10/08/2023'),

('10/09/2023'),

('10/10/2023'),

('10/11/2023'),

('10/12/2023'),

('10/13/2023'),

('10/14/2023'),

('10/15/2023'),

('10/16/2023');

INSERT INTO teacher("name", surname, "degree", department, experience)

VALUES

('Елена', 'Пегачков', 'professor', 805, 10),

('Юрий', 'Битюков', 'assistant', 804, 15),

('Валерий', 'Черепанов', 'professor', 901, 20),

('Валентин', 'Зайцев', 'professor', 806, 30),

('Наталья', 'Тарабрина', 'none', 914, 8),

('Дмитрий', 'Сошников', 'assistant', 806, 11),

('Сергей', 'Иванов', 'professor', 805, 9),

('Иван', 'Калита', 'none', 501, 8),

('Татьяна', 'Ивашкина', 'none', 510, 20),

('Анастасия', 'Берендаков', 'assistant', 804, 4),

('Олег', 'Симкин', 'assistant', 804, 6),

('Роман', 'Олегович', 'assistant', 804, 7),

('Никита', 'Вакулов', 'none', 304, 17),

('Дмитрий', 'Ширяев', 'professor', 602, 25),

('Роман', 'Хрушев', 'assistant', 404, 10),

('Сергей', 'Вартов', 'none', 203, 21),

('Полина', 'Сеченых', 'assistant', 806, 7),

('Александр', 'Бортаковский', 'professor', 805, 28),

('Александр', 'Серов', 'none', 901, 17),

('Виктор', 'Севостьянов', 'assistant', 806, 3);

Заполнение связей между занятиями и учебными группами:

INSERT INTO lesson\_group (lesson\_id, group\_id)

Values

(1, 1),

(2, 6),

(3, 1),

(4, 1),

(5, 2),

(6, 20),

(7, 11),

(8, 9),

(8, 10),

(8, 11),

(9, 9),

(9, 20),

(9, 16),

(9, 11),

(10, 8),

(11, 7),

(12, 2),

(13, 5),

(14, 5),

(14, 10),

(15, 6),

(15, 7),

(15, 3),

(16, 1),

(17, 8),

(18, 8),

(19, 8),

(19, 20),

(19, 18),

(20, 11),

(20, 8),

(21, 3),

(21, 4),

(21, 7),

(22, 7),

(23, 8),

(24, 3),

(25, 1),

(26, 7),

(26, 14),

(26, 12),

(26, 17),

(26, 18),

(27, 19),

(28, 11),

(29, 15),

(30, 4),

(30, 8),

(30, 9),

(31, 12),

(32, 4),

(32, 5),

(32, 19),

(32, 13);