CREATE DATABASE academy; - создание базы данных

\c academy; - переход к базе данных для работы

CREATE TABLE Students ( s\_id INT PRIMARY KEY, name VARCHAR NOT NULL, start\_year INT ); - создание таблицы Students

CREATE TABLE Courses (c\_no INT PRIMARY KEY,title VARCHAR NOT NULL,hours INT); - Создание таблицы Courses

CREATE TABLE Exams (s\_id INT,c\_no INT,score INT ,PRIMARY KEY (s\_id, c\_no),FOREIGN KEY (s\_id) REFERENCES Students(s\_id) ON DELETE CASCADE, FOREIGN KEY (c\_no) REFERENCES Courses(c\_no) ON DELETE CASCADE); - создание таблицы Exams

INSERT INTO Students (s\_id,name, start\_year) VALUES (1,‘Artyom Shabanov’, 2024); - добавление студентов(Изменять при заполнении)

INSERT INTO Courses (c\_no.title, hours) VALUES (1,'Математика', 100),(2,'Русский язык', 120),(3,'Физика', 90);-заполнение таблицы Courses

INSERT INTO Exams (s\_id, c\_no, score) VALUES (2, 1, 90),(2,2,100), (2,3,70)- Заполнение таблицы Exams.При заполнении менять s\_id и score (Первые и третьи значения в скобках после Values)

SELECT \* FROM Students WHERE s\_id NOT IN (SELECT s\_id FROM Exams); - запрос, который возвращает всех студентов, которые еще не сдали ни одного экзамена.

SELECT Students.name , COUNT (Exams.s\_id) AS exam\_count FROM Students JOIN Exams ON Students.s\_id = Exams.s\_id GROUP BY Students.s\_id , Students.name HAVING COUNT (Exams.s\_id)>0;-запрос, который возвращает список студентов и количество сданных им экзаменов. Только для студентов, у которых есть сданные экзамены.

SELECT Courses.c\_no , Courses.title, AVG(Exams.score) AS avg\_score FROM Courses JOIN Exams ON Courses.c\_no = Exams.c\_no GROUP BY Courses.c\_no, Courses.title ORDER BY avg\_score DESC; - список курсов со средним баллом по экзамену. Список отсортирован по убыванию среднего балла.