<!--1. Создание таблицы "Дисциплины" (Disciplines)

Таблица будет содержать шифр и название дисциплины.

sql

CREATE TABLE Disciplines (

discipline\_code CHAR(8) PRIMARY KEY, -- Шифр дисциплины

name VARCHAR(100) -- Название дисциплины

);

2. Создание таблицы "Преподаватели" (Teachers)

Таблица содержит идентификатор преподавателя, его ФИО и кафедру.

sql

CREATE TABLE Teachers (

teacher\_id INT PRIMARY KEY, -- Идентификатор преподавателя

full\_name VARCHAR(100), -- ФИО преподавателя

department VARCHAR(100) -- Кафедра преподавателя

);

3. Создание таблицы "Сессия" (Exams)

Таблица для хранения расписания экзаменов. Включает факультет, курс, шифр дисциплины, группу,

экзаменатора, аудиторию, дату экзамена и дату консультации.

sql

CREATE TABLE Exams (

faculty VARCHAR(30), -- Факультет

course INT(1), -- Курс

discipline\_code CHAR(8), -- Шифр дисциплины (внешний ключ)

group\_code CHAR(6), -- Группа

examiner\_id INT, -- Экзаменатор (внешний ключ)

classroom INT(3), -- Аудитория

exam\_date DATE, -- Дата экзамена

consultation\_date DATE, -- Дата консультации

PRIMARY KEY (discipline\_code, group\_code), -- Ключевая комбинация полей

FOREIGN KEY (discipline\_code) REFERENCES Disciplines(discipline\_code), -- Связь с дисциплиной

FOREIGN KEY (examiner\_id) REFERENCES Teachers(teacher\_id) -- Связь с преподавателем

);

4. Заполнение данных

Заполнение таблицы "Дисциплины" (Disciplines)

sql

INSERT INTO Disciplines (discipline\_code, name)

VALUES

('MATH101', 'Математика 1'),

('PHYS102', 'Физика 1'),

('CS103', 'Программирование'),

('ENG104', 'Английский язык');

Заполнение таблицы "Преподаватели" (Teachers)

sql

INSERT INTO Teachers (teacher\_id, full\_name, department)

VALUES

(1, 'Иванов Иван Иванович', 'Математика'),

(2, 'Петрова Мария Сергеевна', 'Физика'),

(3, 'Сидоров Петр Алексеевич', 'Программирование'),

(4, 'Алексеева Екатерина Владимировна', 'Языки');

Заполнение таблицы "Сессия" (Exams)

sql

INSERT INTO Exams (faculty, course, discipline\_code, group\_code, examiner\_id, classroom, exam\_date,

consultation\_date)

VALUES

('Физический', 1, 'MATH101', 'Группа1', 1, 101, '2024-12-01', '2024-11-30'),

('Информационный', 2, 'PHYS102', 'Группа2', 2, 102, '2024-12-05', '2024-12-04'),

('Физический', 3, 'CS103', 'Группа1', 3, 103, '2024-12-10', '2024-12-09'),

('Лингвистический', 1, 'ENG104', 'Группа3', 4, 104, '2024-12-15', '2024-12-14');

Л.р. №2. Выборка данных

1. Проверка уникальности комбинаций "Аудитория", "Дата" и "Группа", "Дата"

Для того чтобы проверить уникальность комбинаций "Аудитория", "Дата" и "Группа", "Дата", можно

использовать запрос с группировкой и подсчетом количества записей.

sql

-- Проверка уникальности для "Аудитория" и "Дата"

SELECT classroom, exam\_date, COUNT(\*)

FROM Exams

GROUP BY classroom, exam\_date

HAVING COUNT(\*) > 1;

-- Проверка уникальности для "Группа" и "Дата"

SELECT group\_code, exam\_date, COUNT(\*)

FROM Exams

GROUP BY group\_code, exam\_date

HAVING COUNT(\*) > 1;

Эти запросы показывают, если в какой-то аудитории или группе в определенную дату проводятся несколько

экзаменов.

2. Определение дисциплин, по которым нет экзаменов

Запрос для поиска дисциплин, по которым не назначены экзамены в таблице "Сессия".

sql

SELECT d.discipline\_code, d.name

FROM Disciplines d

LEFT JOIN Exams e ON d.discipline\_code = e.discipline\_code

WHERE e.discipline\_code IS NULL;

Этот запрос вернет дисциплины, для которых не указаны экзамены в таблице "Exams".

3. Создание расписания экзаменов для одной произвольной группы

Для создания расписания экзаменов для одной группы (например, "Группа1"), нужно выбрать все экзамены,

назначенные для этой группы:

sql

SELECT e.faculty, e.course, e.discipline\_code, e.group\_code, t.full\_name AS examiner, e.classroom,

e.exam\_date, e.consultation\_date

FROM Exams e

JOIN Teachers t ON e.examiner\_id = t.teacher\_id

WHERE e.group\_code = 'Группа1';

Этот запрос возвращает все экзамены для группы "Группа1" с информацией о преподавателе, аудитории и

датах.-->