|  |  |  |
| --- | --- | --- |
|  |  |  |
| МИНОБРНАУКИ РОССИИ | | |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** | | |

Институт Информационных технологий

Кафедра математического обеспечения и стандартизации информационных систем

**Отчет**

**по практической работе №2**

по дисциплине «Проектирование и эксплуатация больших гетерогенных программных систем»

по теме «База данных»

**Выполнили студенты группы** ИКБО-06-17 Лукьянчиков Е.А. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись руководителя)

**Руководитель работы** Григорьев В. К.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись руководителя)

Дата защиты работы «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_ 2020 г.

Москва 2020

Задания:

1. Создать БД H2;
2. Написать SQL-скрипт для создания таблиц в базе данных.

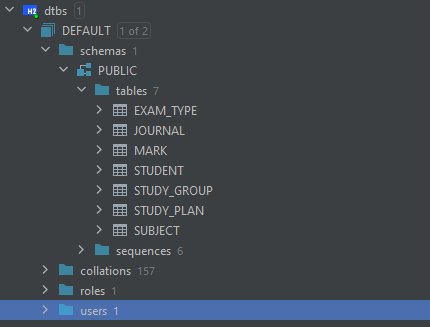


Рисунок 1 – созданная БД

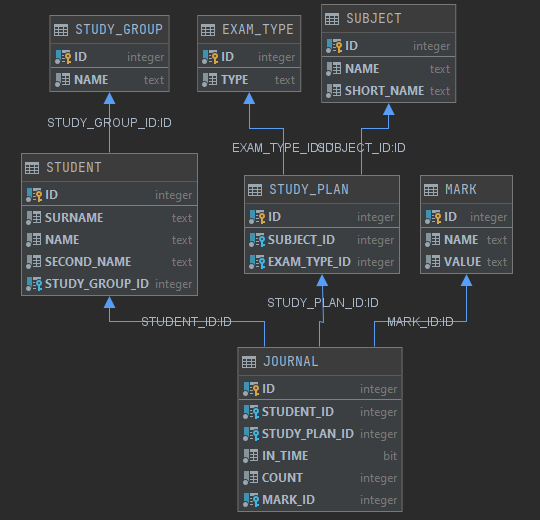


Рисунок 2 – **схема БД**

Код скрипта:

CREATE TABLE IF NOT EXISTS STUDY\_GROUP (  
 id INTEGER AUTO\_INCREMENT PRIMARY KEY NOT NULL,  
 name TEXT NOT NULL  
);  
  
INSERT INTO STUDY\_GROUP VALUES  
(1, 'ИКБО-06-17');  
  
CREATE TABLE IF NOT EXISTS STUDENT (  
 id INTEGER AUTO\_INCREMENT PRIMARY KEY NOT NULL,  
 surname TEXT NOT NULL,  
 name TEXT NOT NULL,  
 second\_name TEXT NOT NULL,  
 study\_group\_id INTEGER NOT NULL  
);  
  
INSERT INTO STUDENT VALUES  
(1, 'Ли', 'Светлана', 'Валерьевна', 1),  
(2, 'Лукьянчиков', 'Егор', 'Анатольевич', 1),  
(3, 'Нестеров', 'Александр', 'Дмитриевич', 1);  
  
CREATE TABLE IF NOT EXISTS SUBJECT (  
 id INTEGER AUTO\_INCREMENT PRIMARY KEY NOT NULL,  
 name TEXT NOT NULL,  
 short\_name TEXT NOT NULL  
);  
  
INSERT INTO SUBJECT VALUES  
(1, 'Проектирование информационных систем', 'ПрИС'),  
(2, 'Системы искусственного интеллекта', 'СИИ'),  
(3, 'Программная инженерия', 'ПИ'),  
(4, 'Национальная система информационной безопасности', 'НСИБ'),  
(5, 'Системный анализ', 'СисАнал'),  
(6, 'Распределенные базы данных', 'РБД'),  
(7, 'Системное программное обеспечение', 'СПО');  
  
  
CREATE TABLE IF NOT EXISTS STUDY\_PLAN (  
 id INTEGER AUTO\_INCREMENT PRIMARY KEY NOT NULL,  
 subject\_id INTEGER NOT NULL,  
 exam\_type\_id INTEGER NOT NULL  
);  
  
INSERT INTO STUDY\_PLAN VALUES  
(1, 1, 1),  
(2, 1, 4),  
(3, 2, 1),  
(4, 3, 1),  
(5, 4, 2),  
(6, 5, 1),  
(7, 6, 2),  
(8, 7, 1);  
  
CREATE TABLE IF NOT EXISTS EXAM\_TYPE (  
 id INTEGER PRIMARY KEY NOT NULL,  
 type TEXT NOT NULL  
);  
  
INSERT INTO EXAM\_TYPE VALUES  
(1, 'Экзамен'),  
(2, 'Зачет'),  
(3, 'Зачет с оценкой'),  
(4, 'Курсовая');  
  
CREATE TABLE IF NOT EXISTS JOURNAL (  
 id INTEGER AUTO\_INCREMENT PRIMARY KEY NOT NULL,  
 student\_id INTEGER NOT NULL,  
 study\_plan\_id INTEGER NOT NULL,  
 in\_time BIT NOT NULL,  
 count INTEGER NOT NULL,  
 mark\_id INTEGER NOT NULL  
);  
  
CREATE TABLE IF NOT EXISTS MARK (  
 id INTEGER AUTO\_INCREMENT PRIMARY KEY NOT NULL,  
 name TEXT NOT NULL,  
 value TEXT NOT NULL  
);  
  
  
INSERT INTO MARK VALUES  
(1, 'Отлично', 5),  
(2, 'Хорошо', 4),  
(3, 'Удовлетворительно', 3),  
(4, 'Неудовлетворительно', 2),  
(5, 'Зачет', 'з'),  
(6, 'Незачет', 'н'),  
(7, 'Неявка', '');  
  
alter table STUDENT add foreign key (study\_group\_id) references STUDY\_GROUP(id);  
alter table STUDY\_PLAN add foreign key (subject\_id) references SUBJECT(id);  
alter table STUDY\_PLAN add foreign key (exam\_type\_id) references EXAM\_TYPE(id);  
alter table JOURNAL add foreign key (student\_id) references STUDENT(id);  
alter table JOURNAL add foreign key (study\_plan\_id) references STUDY\_PLAN(id);  
alter table JOURNAL add foreign key (mark\_id) references MARK(id);

Контрольные вопросы:

1) Что подразумевается под таблицей и полем в SQL?

2) Какие есть типы связей в базах данных?

**Связь один-к-одному** создается в случае, если ключевой столбец (идентификатор) принадлежит другой таблице, для которой также является ключевым или его уникальность задается свойствами столбца (значения в разных строках не могут повторяться).

На практике такая связь используется редко. Например, связь «один-к-одному» может использоваться, если нужно выполнить разделение данных одной таблицы на некоторое число отдельных таблиц для обеспечения безопасности.

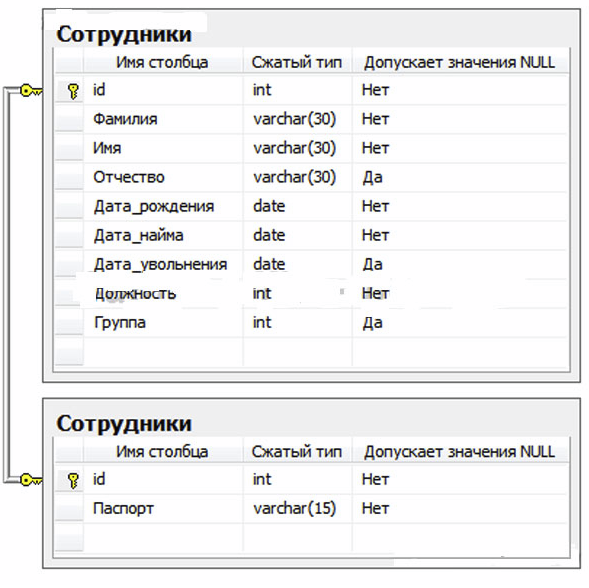


Рисунок 3 – INNER JOIN

3) Что такое соединения (JOIN) в SQL? Каких видов они бывают?

**Оператор языка SQL JOIN** предназначен для соединения двух или более таблиц базы данных по совпадающему условию. Этот оператор существует только в реляционных базах данных. Именно благодаря JOIN реляционные базы данных обладают такой мощной функциональностью, которая позволяет вести не только хранение данных, но и их, хотя бы простейший, анализ с помощью запросов.

Для соединения двух таблиц оператор **SQL JOIN** имеет следующий синтаксис:

SELECT ИМЕНА\_СТОЛБЦОВ (1..N)

FROM ИМЯ\_ТАБЛИЦЫ\_1 JOIN ИМЯ\_ТАБЛИЦЫ\_2

ON УСЛОВИЕ

**INNER JOIN (внутреннее соединение)**

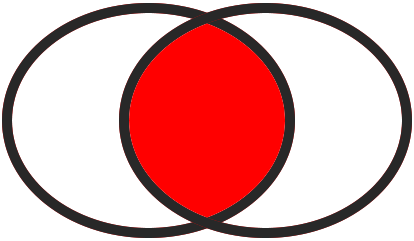


Рисунок 3 – INNER JOIN

**OUTER JOIN (внешнее соединение)**

Внешнее присоединение. Различают LEFT OUTER JOIN и RIGHT OUTER JOIN, и обычно опускают слово «OUTER».

**LEFT JOIN**

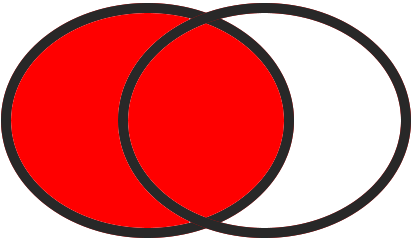
****

Рисунок 4 – LEFT JOIN

**RIGHT JOIN**

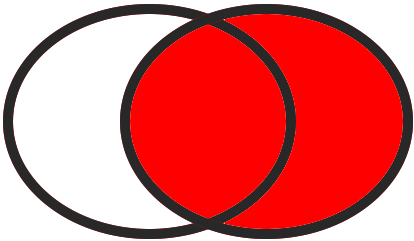
****

Рисунок 5 – RIGHT JOIN

**Полное множество**

MySQL не знает соединения FULL OUTER JOIN.

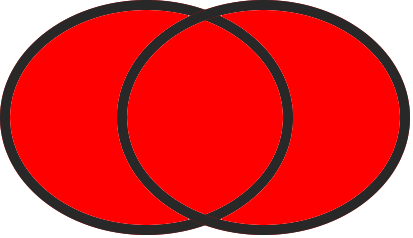
****

Рисунок 6 – Полное множество

**Первый способ** — объединение запросов LEFT и RIGHT.

**Второй способ** — объединение LEFT и RIGHT, но в одном из запросов мы исключаем часть, соответствующую INNER. А объединение задаём как UNION ALL, что позволяет движку SQL обойтись без сортировки.

**Левое подмножество**

LEFT JOIN ограничиваем проверкой, что данных из второй таблицы нет.

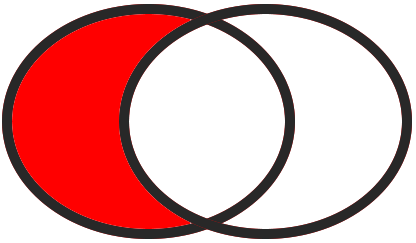
****

Рисунок 7 – Левое подмножество

**Правое подмножество**

RIGHT JOIN ограничиваем проверкой, что данных из второй таблицы нет.

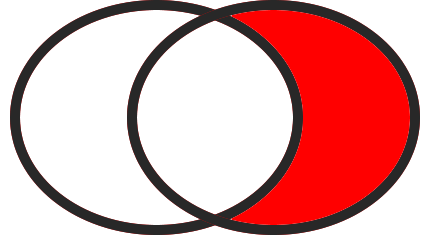
****

Рисунок 8 – правое подмножество

**Всё кроме пересечения**

Его можно сложить из двух предыдущих запросов через UNION ALL (т.к. подмножества не пересекаются).

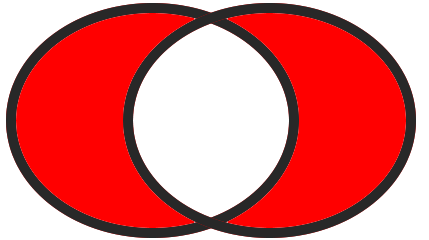
****

Рисунок 9 – все кроме пересечения

4) Напишите запрос на выборку всех предметов с типом экзаменов “Экзамен”.

SELECT [S.name](https://vk.com/away.php?utf=1&to=http%3A%2F%2FS.name), ET.type FROM EXAM\_TYPE ET   
JOIN STUDY\_PLAN SP on [ET.id](https://vk.com/away.php?utf=1&to=http%3A%2F%2FET.id) = SP.EXAM\_TYPE\_ID   
JOIN SUBJECT S on SP.subject\_id = [S.ID](https://vk.com/away.php?utf=1&to=http%3A%2F%2FS.ID) WHERE ET.type = 'Экзамен'

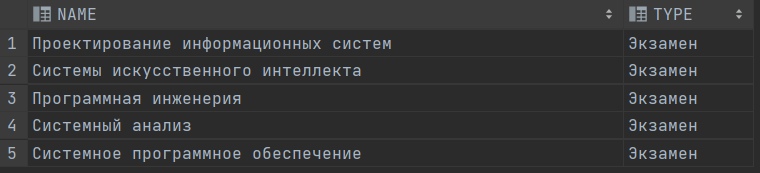


Рисунок 10 – **Результат запроса**