**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра МОЭВМ**

отчет

**по лабораторной работе №10**

**по дисциплине «Базы данных»**

Тема: **Реализация на языке SQL основных операций реляционной алгебры**

|  |  |  |
| --- | --- | --- |
| Студент гр. 6304 |  | Иванов В.С. |
| Преподаватель |  | Фомичева Т.Г. |

Санкт-Петербург

2018

**Цель работы.**

Приобретение навыков написания запросов на языке SQL на примерах реализации набора операций манипуляционной составляющей реляционной модели.

Порядок выполнения работы.

1. Выбрать предметную область и придумать, опираясь на примеры теоретической части методических указаний, собственные примеры для реализации каждой операции. Для операций объединение, пересечение и разность использовать один общий пример. Сформулировать задачу для каждой из 8-ми операций в терминах выбранной предметной области.
2. Создать структуру (определение) каждой таблицы.
3. Заполнить таблицы тестовыми данными.
4. Написать SQL- код, реализующий каждую из 8-ми операций.
5. Выполнить запросы и убедиться, что они реализуют соответствующие операции.
6. Оформить отчёт, по следующей схеме:

* *Название операции*
* *Формулировка задачи*
* *Исходные данные (таблицы)*
* *SQL- код запроса*
* *Таблица – результат*

**Ход работы.**

1. База данных включает в себя таблицы:

* Студенты1 – таблица, содержащая студентов, получающих учебную стипендию (рис.1);

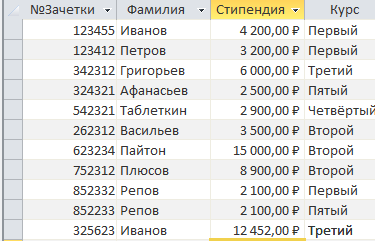


Рисунок 1 – Таблица «Студенты1»

* Студенты2 – таблица, содержащая студентов, получающих неучебную стипендию (рис.2);

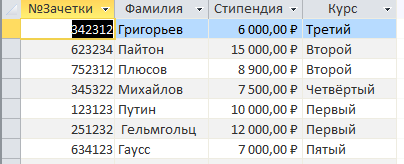


Рисунок 2 – Таблица «Студенты2»

* Экзамены – таблица, содержащая перечень экзаменов и даты их проведения (рис.3);

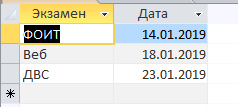


Рисунок 3 – Таблица «Экзамены»

* Курс – таблица, содержащая поле «Курс» (рис.4);

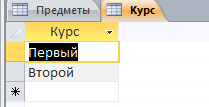


Рисунок 4 – Таблица «Курс»

* Практика – таблица, содержащая информацию о летней практике (рис.5).

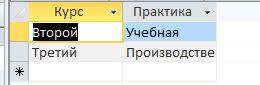


Рисунок 5 – Таблица «Практика»

* Предметы – таблица, содержащая информацию о том, на каком курсе преподаются предметы (рис.6).

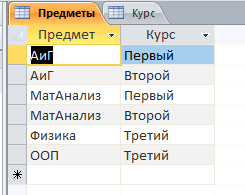


Рисунок 6 – Таблица «Предметы»

1. Описание операций:
2. *Объединение*. Объединение = Студенты1 ∪ Студенты2.

**Задача** – составить список всех студентов, получающих стипендию.

**Исходные** **данные** – таблицы «Студенты1», содержащая информацию о студентах, получающих учебную стипендию, и «Студенты2», содержащая информацию о студентах, получающих неучебную стипендию.

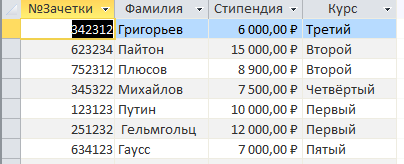
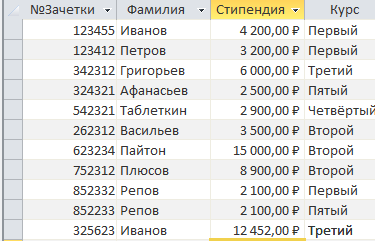


Рисунок 7 – Исходные данные объединения

**SQL-код**:

SELECT \* FROM Студенты1

UNION (SELECT \* FROM Студенты2);

**Результат** - таблица, содержащая информацию о всех студентах, получающих любую стипендию. (рис.8)

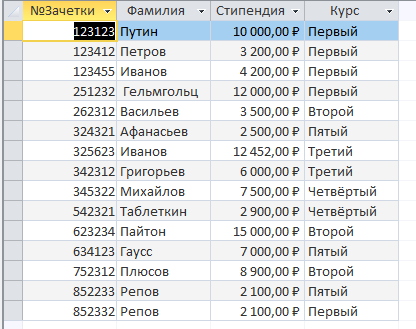


Рисунок 8 – Результат объединения

1. *Пересечение*. Пересечение = Студенты1 ∩ Студенты2

**Задача** – составить список студентов, получающих и учебную, и неучебную стипендии.

**Исходные** **данные** – таблицы «Студенты1», содержащая информацию о студентах, получающих учебную стипендию, и «Студенты2», содержащая информацию о студентах, получающих неучебную стипендию.

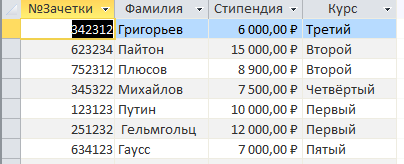
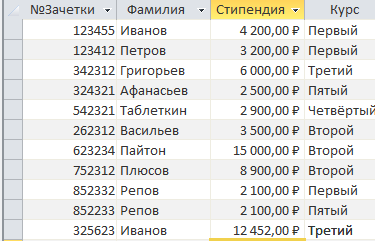


Рисунок 9 – Исходные данные пересечения

**SQL-код**:

SELECT Студенты1.\*

FROM Студенты1, Студенты2

WHERE (((Студенты1.[№Зачетки])=[Студенты2].[№Зачетки]));

**А другие поля?**

**Результат** – таблица, содержащая информацию о студентах, получающих обе стипендии. (рис.10)

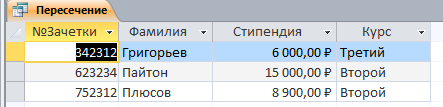


Рисунок 10 – Результат пересечения

1. *Разность*. Разность = Студенты1 \ Студенты2

**Задача** – составить список студентов, получающих только учебную стипендию.

**Исходные данные** – таблицы «Студенты1», содержащая информацию о студентах, получающих учебную стипендию, и «Студенты2», содержащая информацию о студентах, получающих неучебную стипендию.

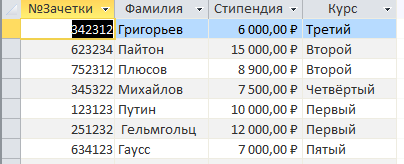
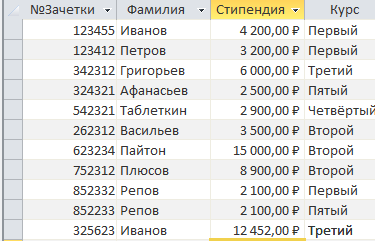


Рисунок 11 – Исходные данные разности

**SQL-код**:

SELECT Студенты1.\*

FROM Студенты1 LEFT JOIN Студенты2 ON Студенты1.[№Зачетки]=Студенты2.[№Зачетки]

WHERE (Студенты2.[№Зачетки] Is Null);

**А другие поля?**

**Результат** – таблица, содержащая студентов, получающих только учебную стипендию. (рис.12)

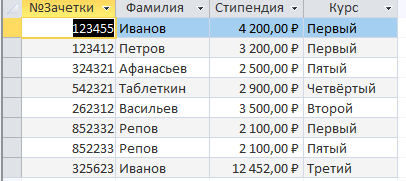


Рисунок 12 – Результат разности

1. *Декартово произведение*.

Декартово произведение = Студенты1×Экзамены

З**адача** – составить экзаменационную ведомость для студентов, получающих учебную стипендию. Ваш результат - это не экзаменационная ведомость. И почему только для них? В какой ситуации это может понадобиться?

**Исходные** **данные** – таблицы «Студенты1», содержащая информацию о студентах, получающих учебную стипендию, и «Экзамены», содержащая список экзаменов и их даты.

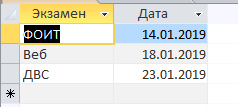
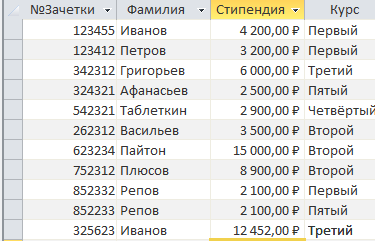


Рисунок 13 – Исходные данные декартово произведения

**SQL-код**:

SELECT Студенты1.[№Зачетки], Студенты1.Фамилия, Экзамены.\*

FROM Студенты1, Экзамены;

2-х последних столбцов таблицы Студенты1 в коде не вижу.

**Результат** – таблица, содержащая информацию о том, когда какой экзамен сдаёт студент. (рис.14)

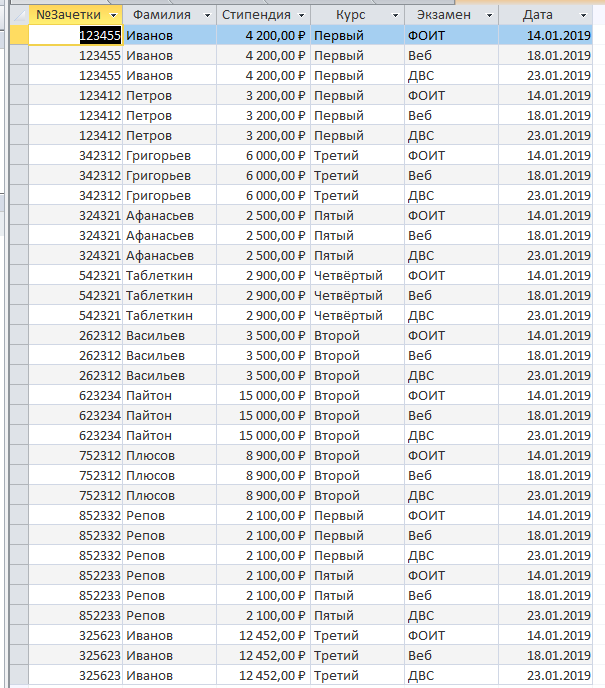


Рисунок 14 – Результат декартова произведения

1. *Деление*. Деление = R1 ÷ R2

**Задача** – составить список предметов, которые преподаются на всех нужных курсах. Кому и для чего нужных?

**Исходные данные** – таблицы «Предметы», содержащая информацию о том, на каком курсе преподаются предметы, и «Курс», содержащая список нужных курсов.

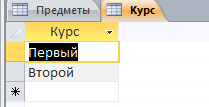
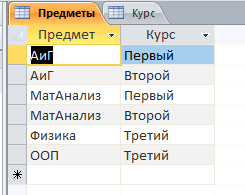


Рисунок 15 – Исходные данные деления

**SQL-код**:

SELECT DISTINCT Предметы.Предмет

FROM Предметы

WHERE EXISTS (SELECT \* FROM Курс WHERE EXISTS (SELECT \* FROM Предметы As X WHERE (X.Курс = Курс.Курс AND Предметы.Предмет = X.Предмет) ) );

**Деление не реализовано! Тот же результат вы получите, если предмет будет читаться и на одном из этих курсов.**

**Результат** – таблица, содержащая список предметов, которые преподаются и на первом, и на втором курсах. (рис.16)

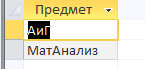


Рисунок 16 – Результат деления

1. *Проекция*. Проекция = πi1 i2… in (Студенты1)

**Задача** – составить список, показывающий размер получаемой каждым студентом, который получает учебную стипендию, стипендии.

**Исходные данные** – таблица «Студенты1», содержащая информацию о студентах, получающих учебную стипендию.

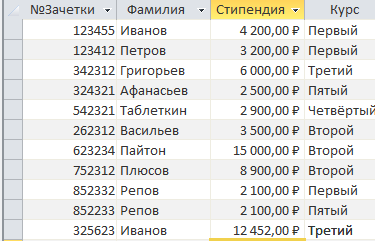


Рисунок 17 – Исходные данные проекции

**SQL-код**:

SELECT DISTINCT Студенты1.Стипендия, Студенты1.[№Зачетки]

FROM Студенты1;

**Результат** – таблица, в которой каждому, получающему учебную стипендию, студенту (по его зачётке) соответствует размер его стипендии. (рис.18)

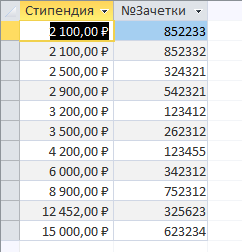


Рисунок 18 – Результат проекции

1. *Селекция* (выбор). Селекция = σF (Студенты1), где F –условие

**Задача** – составить список студентов, получающих учебную стипендию и обучающихся на третьем курсе.

**Исходные** **данные** – таблица «Студенты1», содержащая информацию о студентах, получающих учебную стипендию.

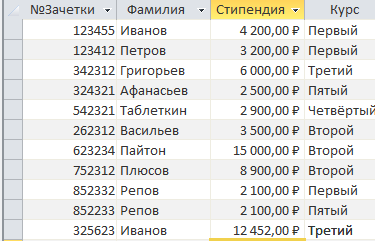


Рисунок 19 – Исходные данные селекции

**Условие** – «Третий» курс

**SQL-код:**

SELECT \*

FROM Студенты1

WHERE Курс="Третий";

**Результат** **–** таблица, содержащая студентов, получающих учебную стипендию и обучающихся на третьем курсе. (рис.20)

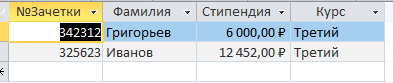


Рисунок 20 – Результат селекции

1. *Соединение*. Соединение = Студенты1 ⊲⊳Практика

**Задача** – выбрать студентов, получающих учебную стипендию, и распределить их по практикам.

**Исходные** **данные** – таблицы «Студенты1», содержащая информацию о студентах, получающих учебную стипендию, и «Практика», содержащая информацию о распределении студентов (по курсу) по практикам. Таблицы Практика не наблюдаю.

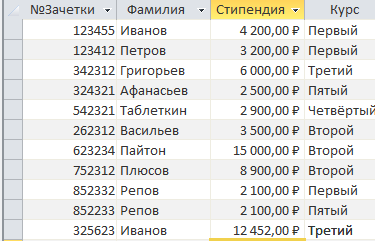


Рисунок 21 – Исходные данные соединения не все

**SQL-код**:

SELECT Студенты1.\*, Практика.Практика

FROM Студенты1, Практика

WHERE Студенты1.Курс=Практика.Курс;

**Результат** – таблица, содержащая распределение студентов, получающих учебную стипендию, по летним практикам. (рис. 22)

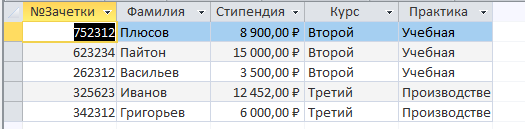


Рисунок 22 – Результат соединения

**Вывод.**

В процессе выполнения данной лабораторной работы была изучена реализация на языке SQL основных операций реляционной алгебры, созданы и выполнены соответствующие основным операциям запросы.