

**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 операторов реляционной алгебры на языке SQL”**

Студент гр. 8304

\_\_\_\_\_

Сергеев А.Д.

Преподаватель

\_\_\_\_\_

Фомичева Т.Г.

Санкт-Петербург

2020

## Цель 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.

## Порядок выполнения работы.

### 1. Объединение ( $R3 = R1 \cup R2$ )

Пусть в таблице R1 хранится информация о блюдах в меню ресторана, содержащих продукты животного происхождения, а в таблице R2 – о блюдах, содержащих продукты растительного происхождения. Необходимо получить данные обо всём меню ресторана.

name	kcal	mass
Борщ	430	500
Шашлык с овощами	670	650
Жульен с грибами	340	300
Суши	580	560

Рисунок 1 — Таблица R1(*contains\_animals*)

name	kcal	mass
Гороховый суп	420	470
Шашлык с овощами	670	650
Жульен с грибами	340	300
Запеканка	880	420

Рисунок 2 — Таблица R2 (*contains\_plants*)

SQL запрос: *SELECT \* FROM contains\_plants UNION SELECT \* FROM contains\_animals;*

name ▾	kcal ▾	mass ▾
Борщ	430	500
Гороховый суп	420	470
Жульен с гриб	340	300
Запеканка	880	420
Суши	580	560
Шашлык с овс	670	650

Рисунок 3 — Таблица R3 (меню ресторана)

## 2. Пересечение ( $R3 = R1 \cap R2$ )

Пусть в таблице R1 хранится информация о блюдах в меню ресторана, содержащих продукты животного происхождения, а в таблице R2 – о блюдах, содержащих продукты растительного происхождения. Необходимо получить данные обо всех блюдах смешанного происхождения.

name ▾	kcal ▾	mass ▾
Борщ	430	500
Шашлык с овощами	670	650
Жульен с грибами	340	300
Суши	580	560

Рисунок 4 — Таблица R1(*contains\_animals*)

name ▾	kcal ▾	mass ▾
Гороховый суп	420	470
Шашлык с овощами	670	650
Жульен с грибами	340	300
Запеканка	880	420

Рисунок 5 — Таблица R2 (*contains\_plants*)

SQL запрос: *SELECT contains\_plants.name, contains\_plants.kkal, contains\_plants.mass*  
*FROM contains\_plants, contains\_animals*  
*WHERE contains\_plants.name=contains\_animals.name And*  
*contains\_plants.kkal=contains\_animals.kkal And*  
*contains\_plants.mass=contains\_animals.mass;*

name ▾	kkal ▾	mass ▾
Шашлык с овощами	670	650
Жульен с грибами	340	300

Рисунок 6 — Таблица R3 (сбалансированные блюда в меню)

### 3. Разность ( $R3 = R1 \setminus R2$ )

Пусть в таблице R1 хранится информация о блюдах в меню ресторана, содержащих продукты животного происхождения, а в таблице R2 – о блюдах, содержащих продукты растительного происхождения. Необходимо получить данные обо всех блюдах, содержащих только блюда растительного происхождения.

name ▾	kkal ▾	mass ▾
Борщ	430	500
Шашлык с овощами	670	650
Жульен с грибами	340	300
Суши	580	560

Рисунок 7— Таблица R1(*contains\_animals*)

name ▾	kkal ▾	mass ▾
Гороховый суп	420	470
Шашлык с овощами	670	650
Жульен с грибами	340	300
Запеканка	880	420

Рисунок 8 — Таблица R2 (*contains\_plants*)

SQL запрос: *SELECT contains\_plants.name, contains\_plants.kkal, contains\_plants.mass FROM contains\_plants WHERE NOT EXISTS (SELECT \* FROM contains\_animals WHERE contains\_plants.name = contains\_animals.name AND contains\_plants.kkal = contains\_animals.kkal AND contains\_plants.mass = contains\_animals.mass);*

name	kkal	mass
Гороховый суп	420	470
Запеканка	880	420

Рисунок 9 — Таблица R3 (вегетарианское меню)

#### 4. Декартово произведение ( $R3 = R1 \times R2$ )

Пусть в таблице R1 хранится информация об уборщиках ресторана (каждый из них должен выполнить все действия по уборке ресторана в свои дни недели в одно и то же время суток) и о том, по каким дням они работают, а в таблице R2 – о том, что конкретно и в какое время дня должно быть сделано. Необходимо получить данные о том, кто когда и что убирает.

name	works_on
Сидоров А. И.	понедельник, четверг
Иванова З. К.	вторник, пятница
Петренко В. Л.	среда, суббота

Рисунок 10 — Таблица R1 (staff)

place	timing
Убрать со столиков	13:45
Вытереть пол	19:45

Рисунок 11 — Таблица R2 (cleaning\_duties)

SQL запрос: *SELECT \* FROM staff, cleaning\_duties;*

name	works_on	place	timing
Сидоров А. И.	понедельник, четверг	Убрать со столиков	13:45
Сидоров А. И.	понедельник, четверг	Вытереть пол	19:45
Иванова З. К.	вторник, пятница	Убрать со столиков	13:45
Иванова З. К.	вторник, пятница	Вытереть пол	19:45
Петренко В. Л.	среда, суббота	Убрать со столиков	13:45
Петренко В. Л.	среда, суббота	Вытереть пол	19:45

Рисунок 12 — Таблица R3 (кто что и когда убирает)

### 5. Деление ( $R3 = R1 \div R2$ )

Пусть не все уборщики справились со своим графиком и теперь в таблице R1 хранится информация о том, кто из уборщиков что и когда успел сделать, а в таблице R2 – о том, что конкретно и в какое время нужно было сделать. Необходимо получить данные о том из уборщиков справился со своими задачами вовремя, чтобы выдать ему премию.

name	works_on	place	timing
Сидоров А. И.	понедельник, четверг	Убрать со столиков	13:45
Сидоров А. И.	понедельник, четверг	Вытереть пол	19:58
Иванова З. К.	вторник, пятница	Убрать со столиков	13:45
Иванова З. К.	вторник, пятница	Вытереть пол	19:45
Петренко В. Л.	среда, суббота	Убрать со столиков	13:56
Петренко В. Л.	среда, суббота	Вытереть пол	19:45

Рисунок 13 — Таблица R1 (actual\_times)

place	timing
Убрать со столиков	13:45
Вытереть пол	19:45

Рисунок 14 — Таблица R2 (cleaning\_duties)

SQL запрос: *SELECT actual\_times.name, actual\_times.works\_on*  
*FROM actual\_times, cleaning\_duties*  
*WHERE actual\_times.timing=cleaning\_duties.timing And*  
*actual\_times.place=cleaning\_duties.place*

*GROUP BY actual\_times.name, actual\_times.works\_on*  
*HAVING COUNT(actual\_times.place) = (SELECT COUNT(place) FROM*  
*cleaning\_duties);*

name	works_on
Иванова З. К.	вторник, пятница

Рисунок 15 — Таблица R3 (кто из уборщиков справился с графиком)

## 6. Проекция ( $R2 = \pi_{i_1, i_2, \dots, i_n}(R1)$ )

Пусть в таблице R1 хранится информация о том, кто из уборщиков что и когда успел сделать. Необходимо получить данные о том, в какое время суток в каких частях ресторана проходила уборка (вне зависимости от дня недели).

name	works_on	place	timing
Сидоров А. И.	понедельник, четверг	Убрать со столиков	13:45
Сидоров А. И.	понедельник, четверг	Вытереть пол	19:58
Иванова З. К.	вторник, пятница	Убрать со столиков	13:45
Иванова З. К.	вторник, пятница	Вытереть пол	19:45
Петренко В. Л.	среда, суббота	Убрать со столиков	13:56
Петренко В. Л.	среда, суббота	Вытереть пол	19:45

Рисунок 16 — Таблица R1 (actual\_times)

SQL запрос: *SELECT place, timing FROM actual\_times GROUP BY place, timing;*

place	timing
Вытереть пол	19:45
Вытереть пол	19:58
Убрать со столиков	13:45
Убрать со столиков	13:56

Рисунок 17 — Таблица R2 (в какое время дня где была уборка)

## 7. Селекция ( $R2 = \sigma_F(R1)$ , где $F$ — условие)

Пусть в таблице R1 хранится информация о сумме заказа каждого из клиентов. Необходимо получить данные о том, кто из клиентов заказал больше, чем на 3000 рублей для того, чтобы обслужить их первыми.

name	price
Вейсман А. Д.	1205
Ожегов С. И.	5907
Петрученко О.	2785
Кузищин В. И.	9638
Дворецкий И. Х.	3078
Мусселиус В.	12895

Рисунок 18 — Таблица R1 (orders)

SQL запрос: *SELECT \* FROM orders WHERE price>3000;*

name	price
Ожегов С. И.	5907
Кузищин В. И.	9638
Дворецкий И. Х.	3078
Мусселиус В.	12895

Рисунок 19 — Таблица R2 (кто из клиентов заказал больше, чем на 3000)

#### 8. Соединение ( $R3 = R1 \triangleright_{i\theta j} \triangleleft R2$ )

Пусть в таблице R1 хранится информация о сумме заказа каждого из клиентов, а в R2 — информация о том, начиная с какой суммы заказа какой комплимент полагается клиенту. Необходимо получить данные о том, какому из клиентов следует принести какой комплимент от ресторана.



name	price
Вейсман А. Д.	1205
Ожегов С. И.	5907
Петрученко О.	2785
Кузищин В. И.	9638
Дворецкий И. Х.	3078
Мусселиус В.	12895

Рисунок 20 — Таблица R1 (orders)

price	compliment
1000	скромный
5000	приятный
10000	щедрый

Рисунок 21 — Таблица R2 (compliments)

SQL запрос: *SELECT orders.name, MAX(compliments.price) AS compliment\_price, LAST(compliments.compliment) AS compliment, orders.price FROM orders INNER JOIN compliments ON orders.price >= compliments.price GROUP BY name, orders.price;*

name	compliment_price	compliment	price
Вейсман А. Д.	1000	скромный	1205
Дворецкий И. Х.	1000	скромный	3078
Кузищин В. И.	5000	приятный	9638
Мусселиус В.	10000	щедрый	12895
Ожегов С. И.	5000	приятный	5907
Петрученко О.	1000	скромный	2785

Рисунок 22 — Таблица R2 (кому из клиентов полагается какой комплимент)

## Вывод.

В ходе данной лабораторной работы были приобретены навыки написания запросов на языке SQL на примерах реализации таких операций как объединение, пересечение, разность, декартово произведение, деление, проекция, селекция, соединение.