**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. 8383 |  | Киреев К.А. |
| Преподаватель |  | Фомичева Т.Г. |

Санкт-Петербург

2020

**Цель работы.**

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

**Основные теоретические положения.**

Реляционная модель определяет не только принципы хранения данных, но и способы их обработки.

Реляционная модель состоит из трех частей:

* структурной,
* целостной,
* манипуляционной.

В основу манипуляционной части модели положена теория множеств.

Отношение – это множество, поэтому средства манипулирования отношениями могут базироваться на традиционных теоретико-множественных операциях, дополненных некоторыми операциями, специфичными для баз данных.

Набор операций манипуляционной составляющей реляционной модели содержит восемь операций, которые делятся на два класса:

* **теоретико-множественные операции**: объединение, пересечение, разность, декартово произведение;
* **специальные реляционные операции**: селекция, проекция, соединение, деление.

В состав реляционной алгебры включаются также операции присваивания и переименования атрибутов.

Операции объединения, пересечения и разности имеют смысл не для любых двух множеств. Поскольку результат каждой из них - отношение (множество однотипных строк), то операндами должны быть отношения с совместимыми, схемами.

Отношения-операнды должны иметь одинаковую степень, одинаковые типы соответствующих атрибутов. Имена атрибутов могут отличаться, тогда после переименования можно выполнить основную операцию.

Для операции декартово произведение надо, чтобы заголовки столбцов, были разными

Все четыре операции: объединение, пересечение, разность и декартово произведение являются ассоциативными, т.е. (А # В) # С = А # (В # С) = А # В # С, где А, В и С - отношения, # - любая операция.

Все операции, кроме разности, коммутативны, т.е. А # В = В # А.

Восемь выделенных Коддом операций реляционной алгебры определяют эффективность реляционной СУБД.

**Постановка задачи.**

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.

**Ход работы.**

1. Предметная область – гостиница, ее посетители и работники.
2. Было создано 5 таблиц: Гости в данный момент, Праздники, Особые гости за всю историю, Работники и Должность. Структура таблиц представлена на рис. 1-5.

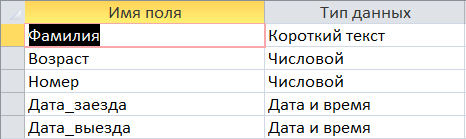


Рисунок 1 – Структура таблицы “Гости в данный момент”

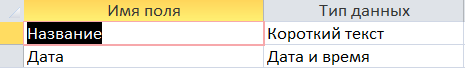


Рисунок 2 – Структура таблицы “Праздники”

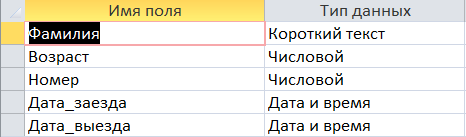


Рисунок 3 – Структура таблицы “Особые гости за всю историю”

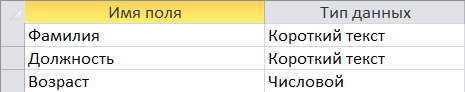


Рисунок 4 – Структура таблицы “Работники”

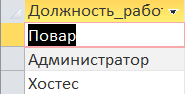


Рисунок 5 – Структура таблицы “Должность”

1. Далее, каждая из таблиц была заполнена тестовыми данными, позволяющими продемонстрировать все 8 операций, данные таблиц представлены на рис. 6-10.

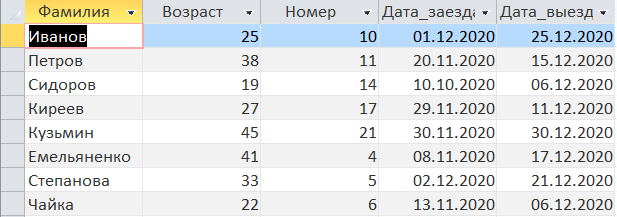


Рисунок 6 – Данные таблицы “Гости в данный момент”

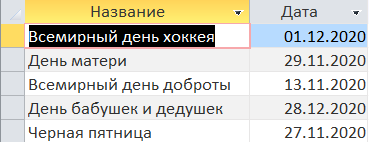


Рисунок 7 – Данные таблицы “Праздники”

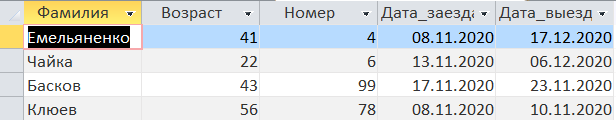


Рисунок 8 – Данные таблицы “Особые гости за всю историю”

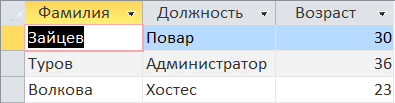


Рисунок 9 – Данные таблицы “Работники”

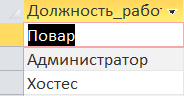


Рисунок 10 – Данные таблицы “Должность”

1. Реализация SQL-кода.

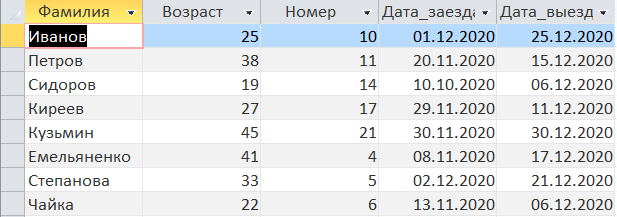
**Объединение (𝑅3 = 𝑅1 ∪ 𝑅2)**

* Постановка задачи

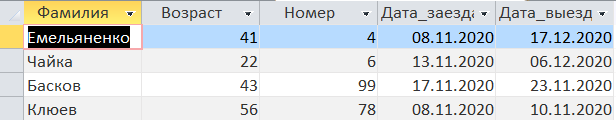
Требуется получить всех людей, проживающих в гостинице в данный момент и всех особых гостей. Тогда R1 – люди, проживающие в данный момент, а R2 – особые гости за всю историю.

* Исходные данные

R1:



R2:

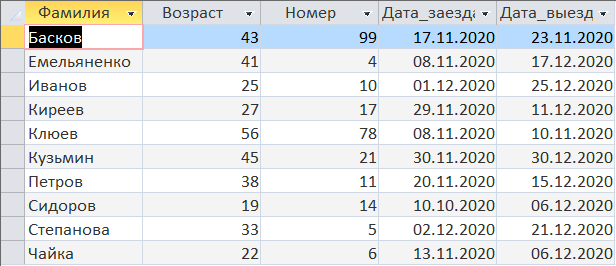


* SQL-код операции

SELECT Гости\_в\_данный\_момент.\* FROM Гости\_в\_данный\_момент

UNION SELECT Особые\_гости\_за\_всю\_историю.\* FROM Особые\_гости\_за\_всю\_историю;

* Результат



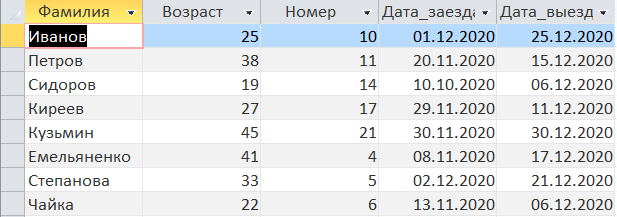
**Пересечение (𝑅3 = 𝑅1 ∩ 𝑅2)**

* Постановка задачи

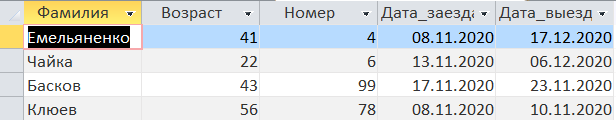
Требуется получить всех людей, проживающих в гостинице в данный момент и являющихся особыми гостями. Тогда R1 – люди, проживающие в данный момент, а R2 – особые гости за всю историю.

* Исходные данные

R1:



R2:



* SQL-код операции

SELECT Гости\_в\_данный\_момент.\*

FROM Гости\_в\_данный\_момент, Особые\_гости\_за\_всю\_историю

WHERE (Гости\_в\_данный\_момент.Фамилия = Особые\_гости\_за\_всю\_историю.Фамилия) AND

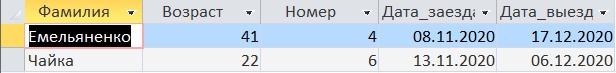
(Гости\_в\_данный\_момент.Возраст = Особые\_гости\_за\_всю\_историю.Возраст) AND

(Гости\_в\_данный\_момент.Номер = Особые\_гости\_за\_всю\_историю.Номер) AND

(Гости\_в\_данный\_момент.Дата\_заезда = Особые\_гости\_за\_всю\_историю.Дата\_заезда) AND

(Гости\_в\_данный\_момент.Дата\_выезда = Особые\_гости\_за\_всю\_историю.Дата\_выезда);

* Результат



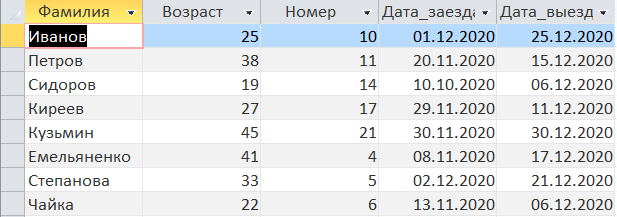
**Разность (𝑅3 = 𝑅1 \ 𝑅2)**

* Постановка задачи

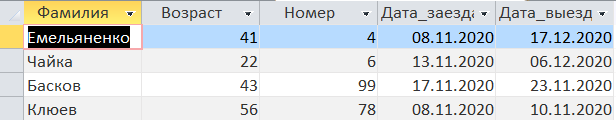
Требуется получить всех людей, проживающих в гостинице в данный момент, которые не являются особыми гостями. Тогда R1 – люди, проживающие в данный момент, а R2 – особые гости за всю историю.

* Исходные данные

R1:



R2:



* SQL-код операции

SELECT Гости\_в\_данный\_момент.\*

FROM Гости\_в\_данный\_момент LEFT JOIN Особые\_гости\_за\_всю\_историю ON (Гости\_в\_данный\_момент.Дата\_выезда = Особые\_гости\_за\_всю\_историю.Дата\_выезда) AND (Гости\_в\_данный\_момент.Дата\_заезда = Особые\_гости\_за\_всю\_историю.Дата\_заезда) AND (Гости\_в\_данный\_момент.Номер = Особые\_гости\_за\_всю\_историю.Номер) AND (Гости\_в\_данный\_момент.Возраст = Особые\_гости\_за\_всю\_историю.Возраст) AND (Гости\_в\_данный\_момент.Фамилия = Особые\_гости\_за\_всю\_историю.Фамилия)

WHERE (Особые\_гости\_за\_всю\_историю.Фамилия IS NULL) AND

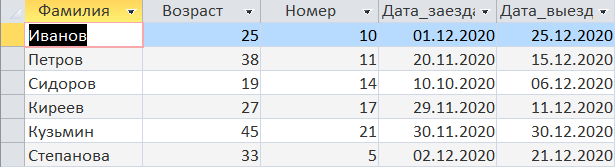
(Особые\_гости\_за\_всю\_историю.Возраст IS NULL) AND

(Особые\_гости\_за\_всю\_историю.Номер IS NULL) AND

(Особые\_гости\_за\_всю\_историю.Дата\_заезда IS NULL) AND

(Особые\_гости\_за\_всю\_историю.Дата\_выезда IS NULL);

* Результат



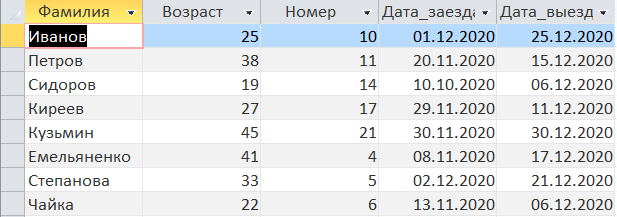
**Декартово произведение (𝑅3 = 𝑅1 × 𝑅2)**

* Постановка задачи

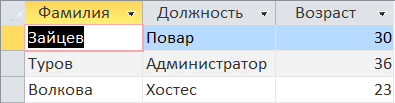
Требуется получить полный список обслуживающего персонала для каждого посетителя гостиницы. Обслуживающий персонал одинаковый для всех клиентов. Тогда R1 – люди, проживающие в данный момент, а R2 – работники.

* Исходные данные

R1:



R2:

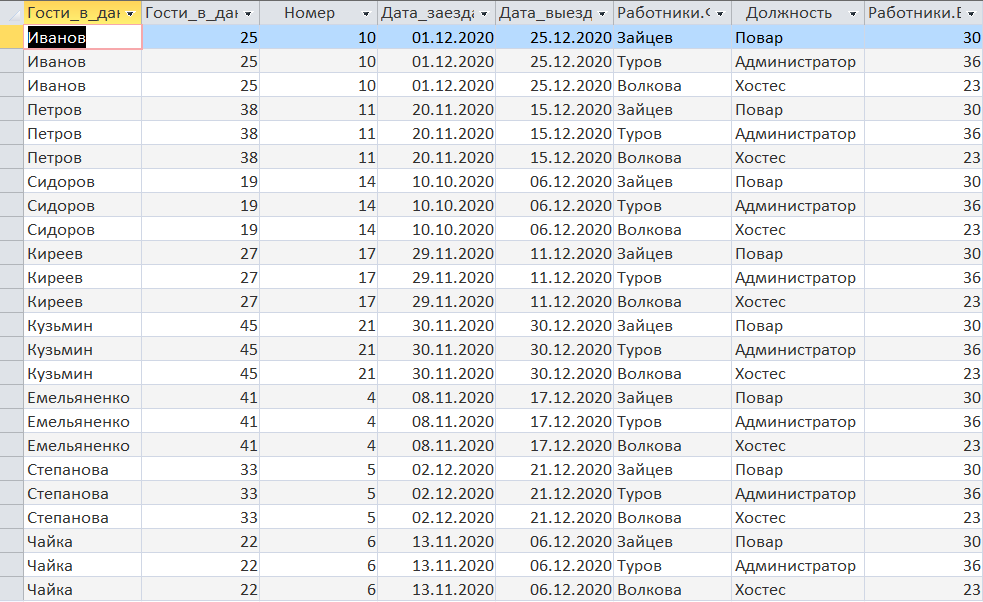


* SQL-код операции

SELECT Гости\_в\_данный\_момент.\*, Работники.\*

FROM Гости\_в\_данный\_момент, Работники;

* Результат



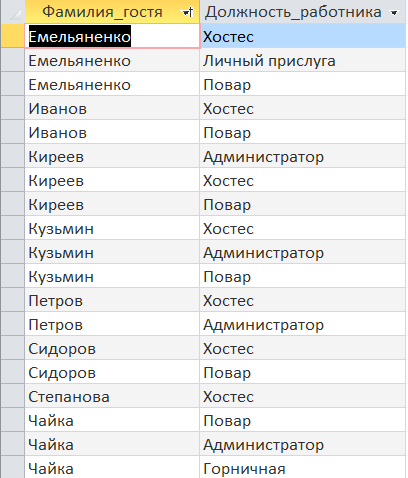
**Деление (𝑅3 = 𝑅1 ÷ 𝑅2)**

* Постановка задачи

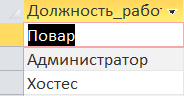
Модернизируем таблицу из прошлого пункта и удалим некоторых работников для нескольких посетителей и оставим только столбцы с фамилией гостя и должностью работника, который его обслуживает, полученную таблицу назовем “Предоставление сервиса гостиницы”. Также добавим гостям Емельяненко и Чайка новых работников, не входящих в список делителя: личный прислуга и горничная соответственно. Требуется получить список имен всех гостей, которых обслуживают все работники из списка должностей. Тогда R1 – “Предоставление сервиса гостиницы”, а R2 – должности работников.

* Исходные данные

R1:



R2:



* SQL-код операции

SELECT DISTINCT Фамилия\_гостя

FROM Предоставление\_сервиса\_гостиницы AS X

WHERE NOT EXISTS

(SELECT Должность\_работника FROM Должность

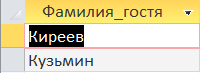
WHERE Должность\_работника NOT IN

(SELECT Должность\_работника

FROM Предоставление\_сервиса\_гостиницы AS Y

WHERE X.Фамилия\_гостя = Y.Фамилия\_гостя));

* Результат



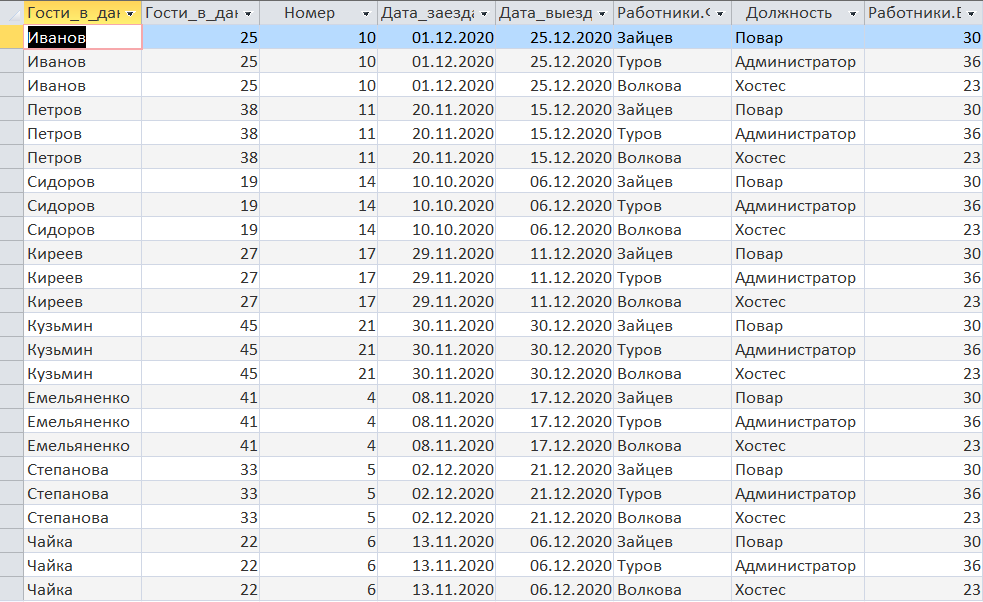
**Проекция (𝑅3 = 𝜋𝑖1, 𝑖2 ... 𝑖3 (𝑅1))**

* Постановка задачи

Требуется получить всех посетителей гостиницы в текущий момент и номера, в которых они проживают. Тогда R1 – таблица, полученная декартовым произведением выше, содержащая сведения (все атрибуты) о каждом госте. Применяемая операция над R1: **𝜋1, 3**.

* Исходные данные

R1:

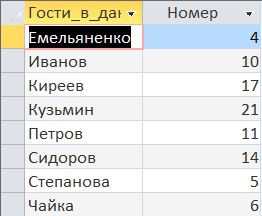


* SQL-код операции

SELECT DISTINCT Полный\_сервис.Гости\_в\_данный\_момент\_Фамилия, Полный\_сервис.Номер

FROM Полный\_сервис;

* Результат



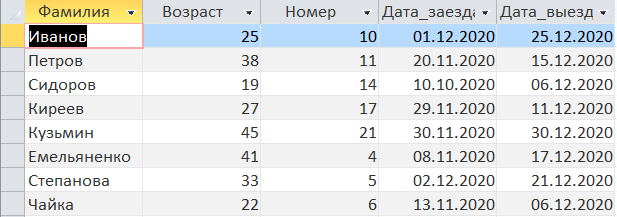
**Селекция (выбор) (𝑅3 = 𝜎𝐹 (𝑅1))**

* Постановка задачи

Требуется получить список всех гостей, старше 35 лет. Тогда R1 – люди, проживающие в данный момент. Применяемая операция над R1: **𝜎𝐹**, где **𝜎𝐹** – атрибут «Возраст» больше 35.

* Исходные данные

R1:



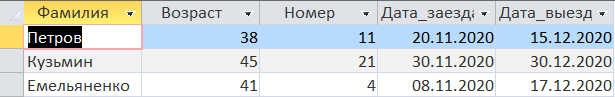
* SQL-код операции

SELECT Гости\_в\_данный\_момент.\*

FROM Гости\_в\_данный\_момент

WHERE Гости\_в\_данный\_момент.Возраст > 35;

* Результат



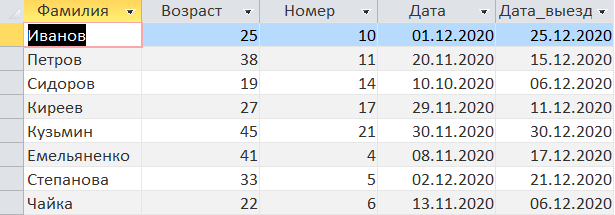
**Эквисоединение (𝑅3 = 𝑅1 ⊲ i=j ⊳ 𝑅2)**

* Постановка задачи

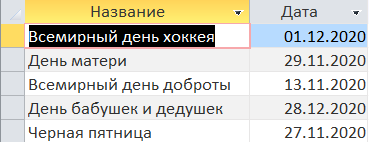
Требуется получить список всех гостей, дата заезда которых совпадает с одним из праздников. Тогда R1 – люди, проживающие в данный момент. R2 – праздники. Условие соединения таблиц – равенство даты заезда посетителя и даты праздника. В результирующем отношении появляется два атрибута с одинаковыми значениями.

* Исходные данные

R1:



R2:

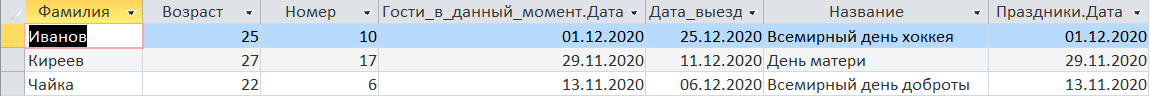


* SQL-код операции

SELECT Гости\_в\_данный\_момент.\*, Праздники.\*

FROM Праздники INNER JOIN Гости\_в\_данный\_момент ON Праздники.Дата = Гости\_в\_данный\_момент.Дата;

* Результат



**Выводы.**

В ходе выполнения данной работы были изучены основы составления SQL запросов и закреплены знания реляционной алгебры.