МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ

УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«СЕВАСТОПОЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Институт информационных технологий и управления в технических системах

Кафедра «Информационные системы»

**ОТЧЕТ**

о выполнении лабораторной работы №1

по дисциплине “Управление данными”

Выполнил: ст. гр. ИС/б-20-2-о

Белик Г. М.

Проверил: ст. преподаватель каф.

«Информационные системы»

Абрамович А.Ю.

Севастополь

2022

**ЛАБОРАТОРНАЯ РАБОТА №1**

**«Манипулирование базой данных. Реляционная алгебра и SQL»**

**Цель работы**

Изучить основы реляционной алгебры как базового средства манипулирования. Научиться представлять запросы как на реляционной алгебре, так и на SQL.

**Задача**

1. Проанализировать схему БД своего варианта задания (см. приложение А), выделить и классифицировать все существующие связи, определить необходимые ограничения целостности. Номер варианта (х) определяется как x = N mod 20, где N – номер студента в группе.

2. Создать базу данных и все ее таблицы. Особое внимание надо уделить описанию первичного ключа, значений по умолчанию, описателям NOT NULL и конструкции CHECK.

3. Установить связи между таблицами.

4. Занести в таблицу образцы данных оператором INSERT INTO. Необходимо занести не менее 10 строк.

5. Проверить работу ограничений целостности (каскадирование удаления, модификации и др.)

6. Продемонстрировать выполнение простых вычислений в запросе.

7. Использовать простое вычисление как параметр агрегатной функции.

8. Продемонстрировать работу предложения GROUP BY.

9. Продемонстрировать работу предложения HAVING.

10. Ознакомиться с операциями реляционной алгебры.

11. Применить к БД операции селекции и соединения в одном запросе.

12. Создать запрос, использующий операции проекции и деления (в одном запросе).

13. Создать запрос, использующий операции проекции, объединения и конъюнкции (в одном запросе).

14. Создать запрос, использующий операции соединения и деления (в одном запросе).

15. Создать запрос, использующий операции вычитания и дизъюнкции (в одном запросе).

16. Сформулировать и записать запрос на SQL, не реализующийся на РА.

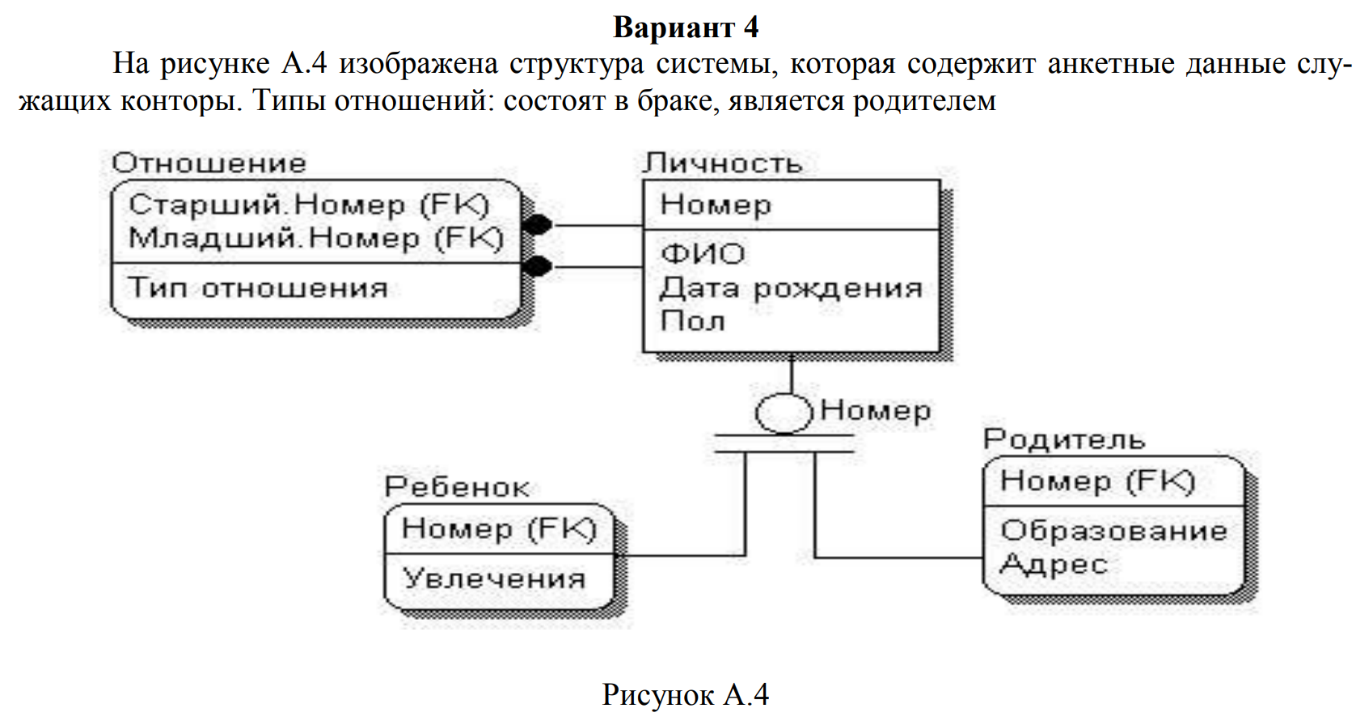


Рисунок 1 – Вариант задания

**Ход работы**

Для начала выполнения работы была создана база данных (Рисунок 2).

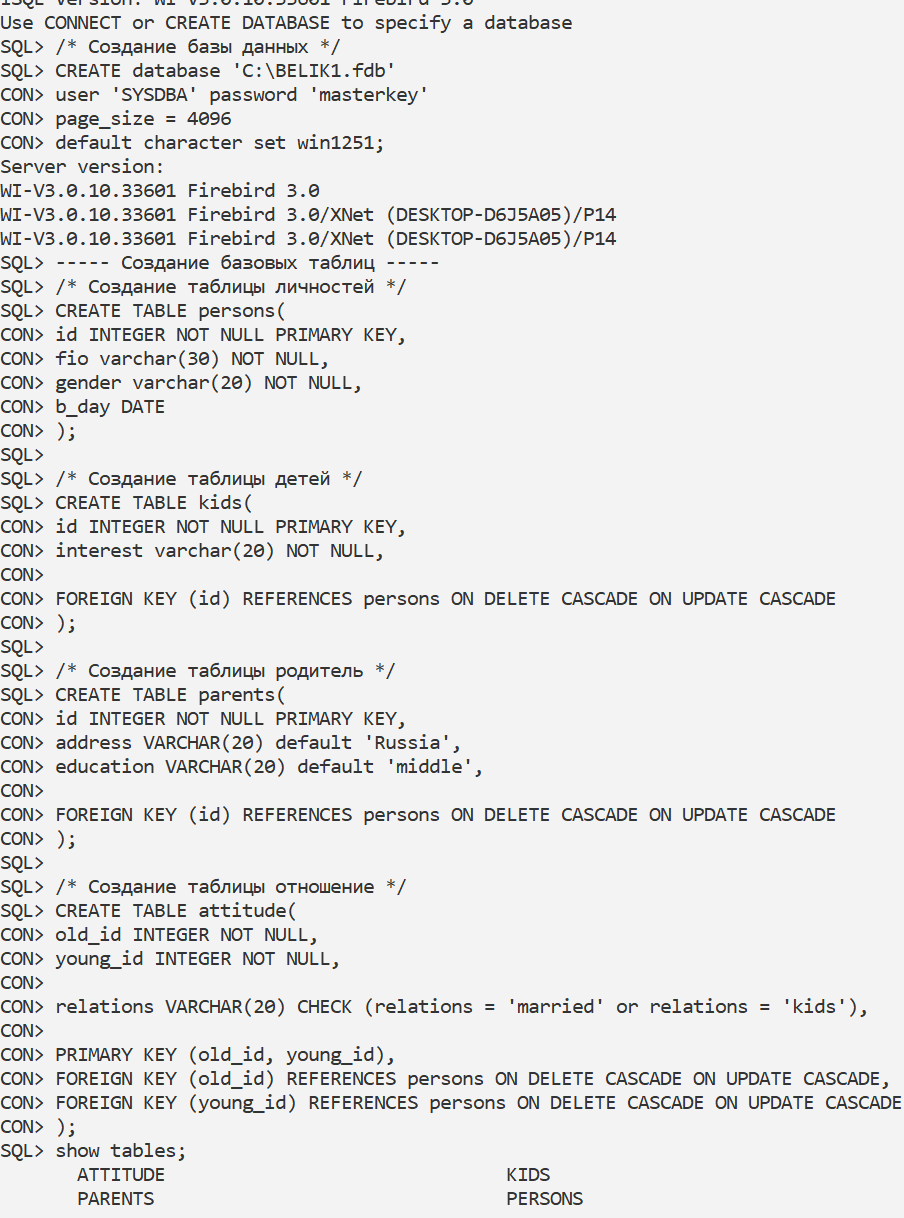


Рисунок 2 – База данных

Затем были созданы все ее таблицы(Рисунок 3).

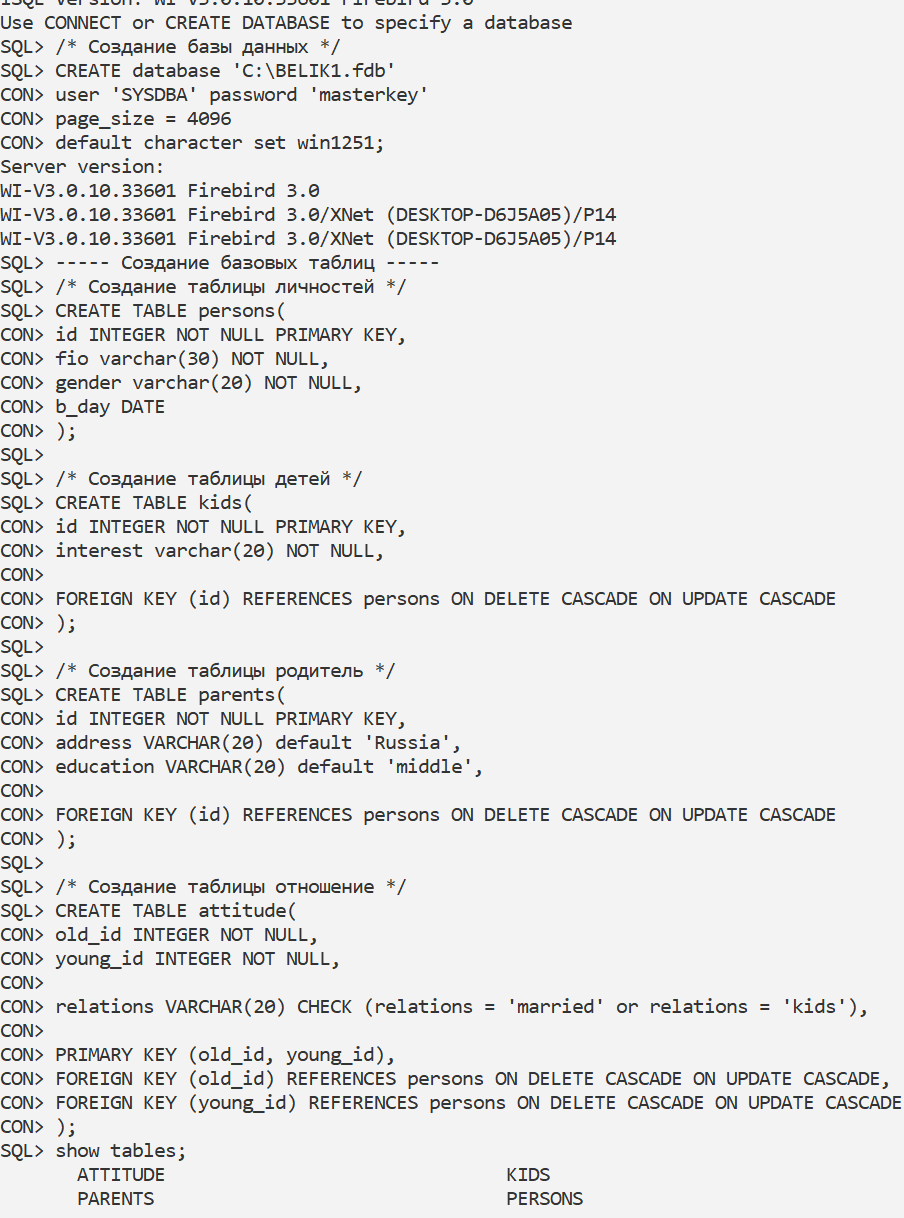
****

Рисунок 3 – Создание таблиц базы данных

После этого были установлены связи между таблицами(Рисунок 4).

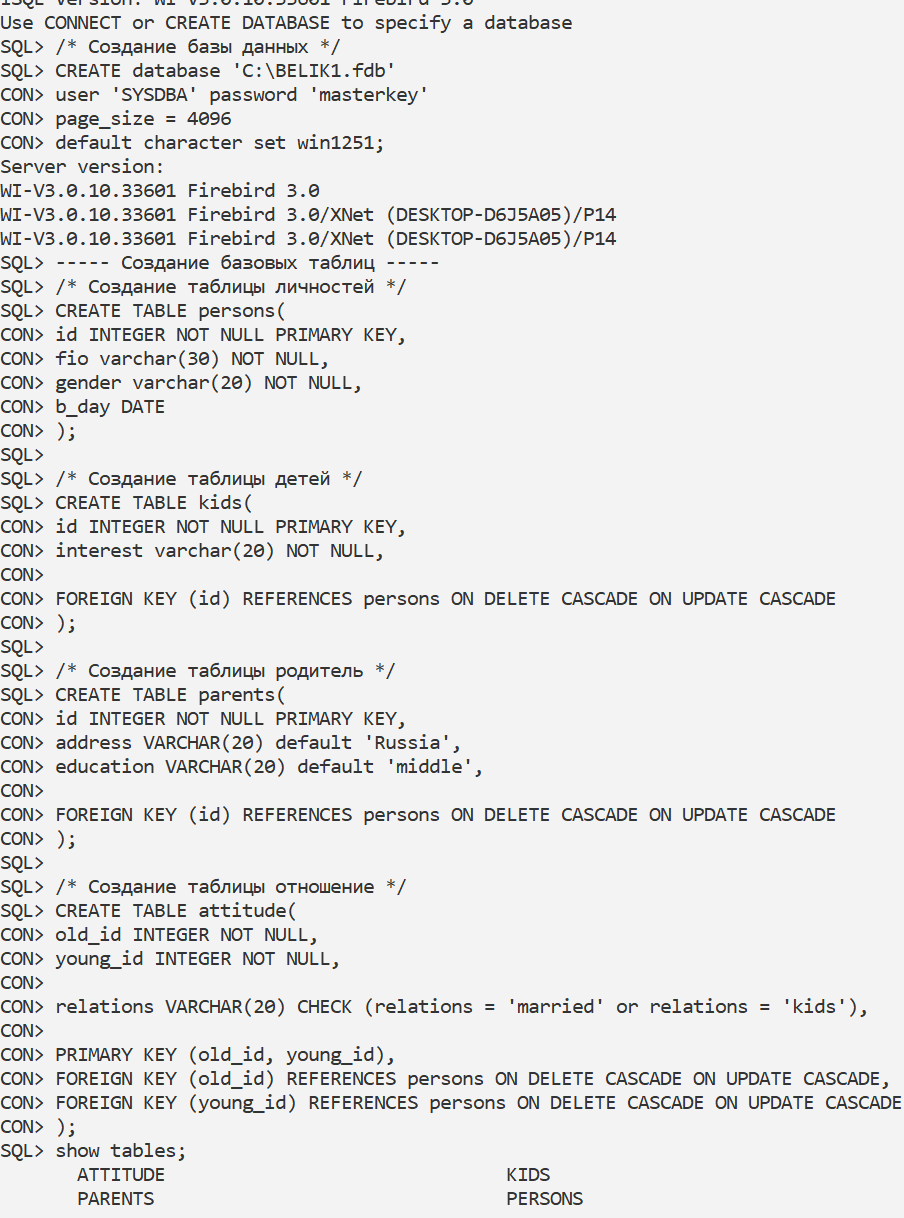


Рисунок 4 – Установленные связи между таблицами

Далее образцы данных были занесены в таблицу оператором INSERT INTO. (Рисунок 5).

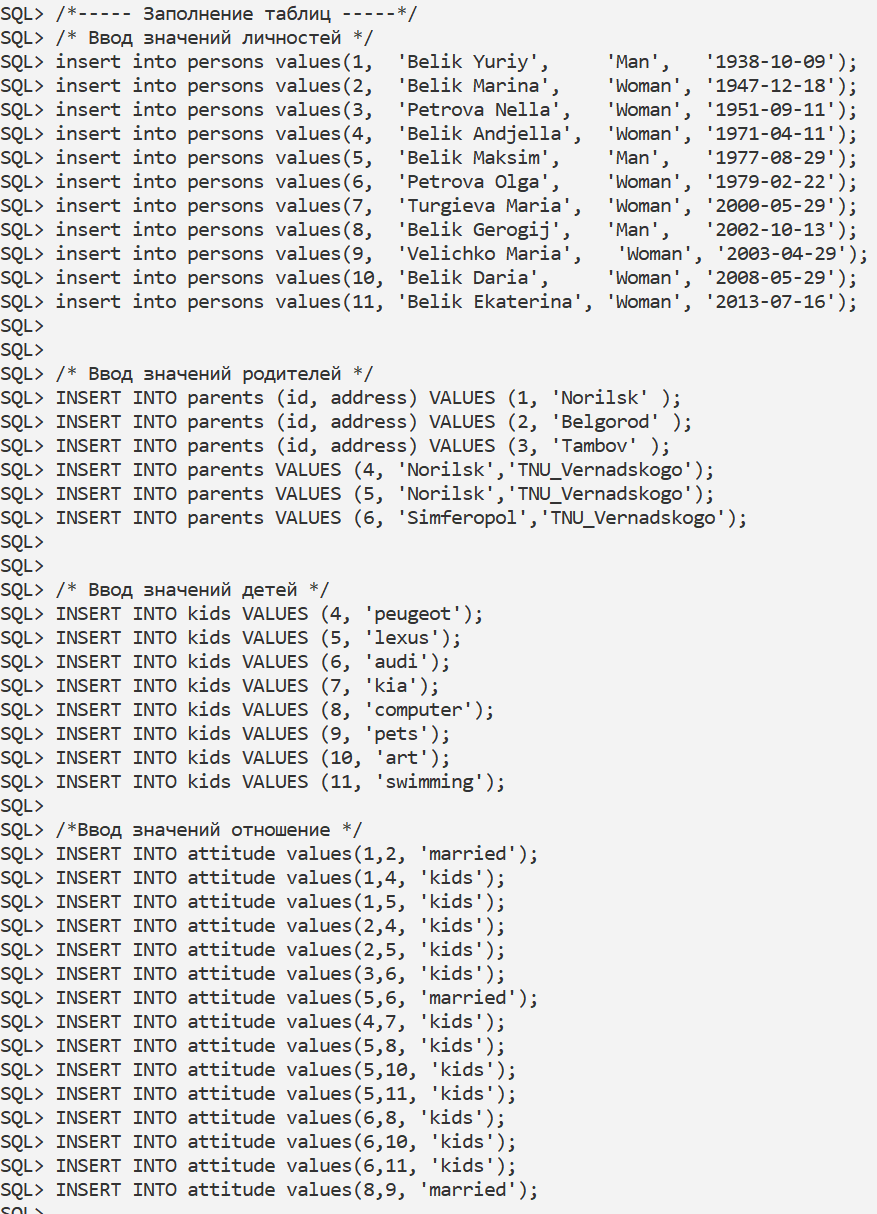


Рисунок 5 – Заполнение таблиц данными

Затем произвели проверку работы ограничений целостности (Рисунок 6).

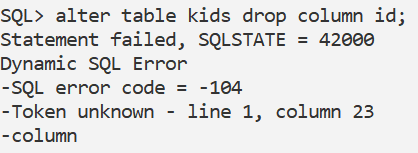


Рисунок 6 – Проверка ограничений целостности.

Далее были выполнены простые вычисления в запросе как параметр агрегатной

функции.

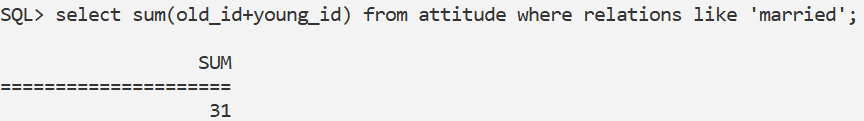


Рисунок 7 – Выполнение простых вычислений в запросе.

После этого произвелась работа предложения GROUP BY(Рисунок 8).

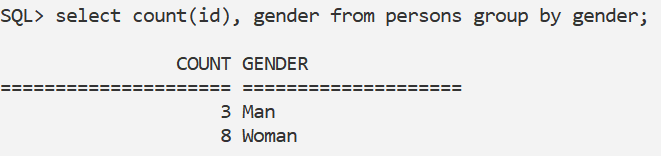


Рисунок 8 – Демонстрация работы Group By.

Затем использовали предложение HAVING(Рисунок 9).

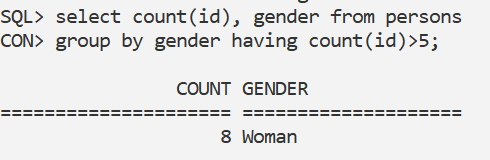


Рисунок 9 – Работа предложения HAVING.

Далее к БД были произведены операции селекции и соединения в одном запросе –

Соединения таблиц персон и родителей, у которых среднее образование(Рисунок 10).

Вывести всю информацию о личностях и родителях, где образование родителей =’middle’.

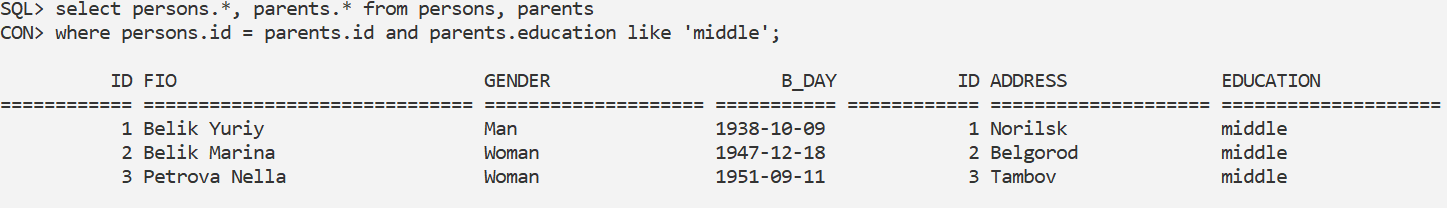


Рисунок 10 – Демонстрация работы операций селекции и соединения.

PA:

После создали запрос, использующий операции проекции и деления (Рисунок 11).

Определить ФИО личностей, которые являются всеми детьми.

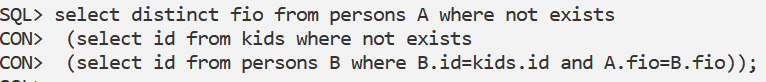


Рисунок 11 – Созданный запрос, использующий операции деления и проекции.

РА:

Далее был создан запрос, использующий операции проекции, объединения и

конъюнкции (Рисунок 12).

Выбрать номера и фио людей у которых номер больше 3 либо они являются детьми

и их номер меньше 10 и интересы детей, у которых номер меньше 10.

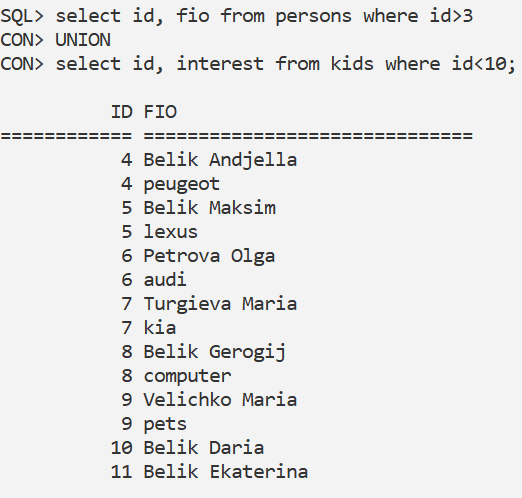


Рисунок 12 – Запрос, использующий операции проекции, объединения и конъюнкции.

РА:

Затем был создан запрос, использующий операции соединения и деления

(Рисунок 13).

Выбрать ФИО и номера людей, которые не являются детьми.

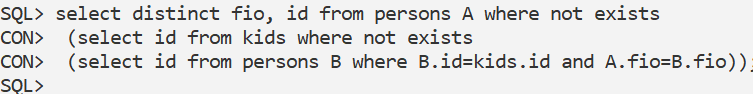


Рисунок 13 – Запрос, использующий операции деления и соединения.

РА:

После создали запрос, использующий операции вычитания и дизъюнкции

(Рисунок 13).

Определить личностей, которые являются детьми, но их номер не равен 4.

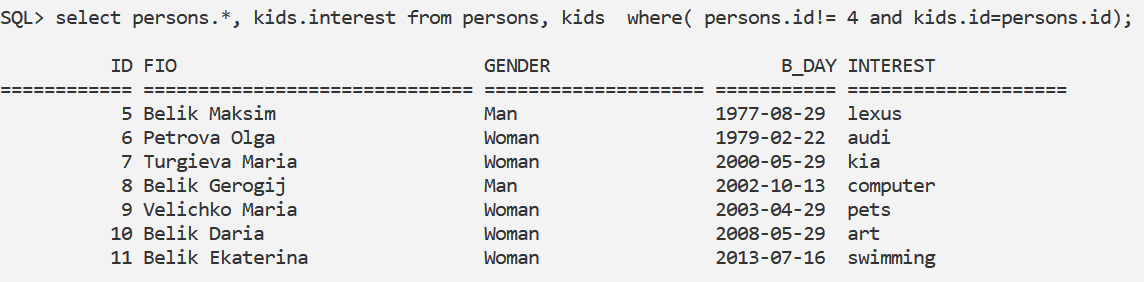


Рисунок 14 – Запрос, использующий операции вычитания и дизъюнкции.

РА: R1 = π .\* (persons)

R2 = π interest kids.id= persons.id && id!=4 (kids)

R =R1 – R2

Далее был сформулирован запрос на SQL, не реализующийся на РА(Рисунок 15).

Для примера был выбран запрос с агрегатной функцией.

В начале определили сколько раз использовались дети в таблице отношений (сумма).

Затем были определена сумма детей, использованных в таблицу отношений без повторей.

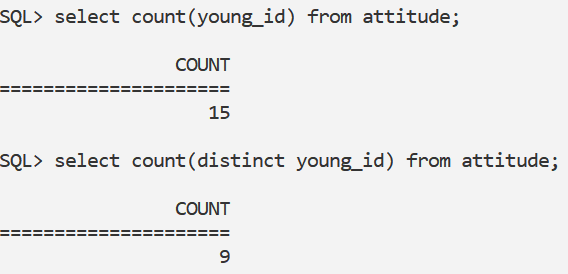


Рисунок 15 – Запрос на SQL, не реализующийся на РА.

SQL возвращает таблицу (возможно с дубликатами), которые нужно удалять явно

помощью distinct. А РА возвращает набор (без дубликатов по определению).

**Выводы**

В ходе выполнения данной лабораторной работы были изучены основы реляционной алгебры как базового средства манипулирования, получены навыки представления запросов как на реляционной алгебре, так и на SQL.

**ПРИЛОЖЕНИЕ А**

**Тексты запросов**

/\* Создание базы данных \*/

CREATE database 'C:\BELIK1.fdb'

user 'SYSDBA' password 'masterkey'

page\_size = 4096

default character set win1251;

/\*----- Создание базовых таблиц -----\*/

/\* Создание таблицы личностей \*/

CREATE TABLE persons(

id INTEGER NOT NULL PRIMARY KEY,

fio varchar(30) NOT NULL,

gender varchar(20) NOT NULL,

b\_day DATE

);

/\* Создание таблицы детей \*/

CREATE TABLE kids(

id INTEGER NOT NULL PRIMARY KEY,

interest varchar(20) NOT NULL,

FOREIGN KEY (id) REFERENCES persons ON DELETE CASCADE ON UPDATE CASCADE

);

/\* Создание таблицы родитель \*/

CREATE TABLE parents(

id INTEGER NOT NULL PRIMARY KEY,

address VARCHAR(20) default 'Russia',

education VARCHAR(20) default 'middle',

FOREIGN KEY (id) REFERENCES persons ON DELETE CASCADE ON UPDATE CASCADE

);

/\* Создание таблицы отношение \*/

CREATE TABLE attitude(

old\_id INTEGER NOT NULL,

young\_id INTEGER NOT NULL,

relations VARCHAR(20) CHECK (relations = 'married' or relations = 'kids'),

PRIMARY KEY (old\_id, young\_id),

FOREIGN KEY (old\_id) REFERENCES persons ON DELETE CASCADE ON UPDATE CASCADE,

FOREIGN KEY (young\_id) REFERENCES persons ON DELETE CASCADE ON UPDATE CASCADE

);

/\*----- Заполнение таблиц -----\*/

/\* Ввод значений личностей \*/

insert into persons values(1, 'Belik Yuriy', 'Man', '1938-10-09');

insert into persons values(2, 'Belik Marina', 'Woman', '1947-12-18');

insert into persons values(3, 'Petrova Nella', 'Woman', '1951-09-11');

insert into persons values(4, 'Belik Andjella', 'Woman', '1971-04-11');

insert into persons values(5, 'Belik Maksim', 'Man', '1977-08-29');

insert into persons values(6, 'Petrova Olga', 'Woman', '1979-02-22');

insert into persons values(7, 'Turgieva Maria', 'Woman', '2000-05-29');

insert into persons values(8, 'Belik Gerogij', 'Man', '2002-10-13');

insert into persons values(9, 'Velichko Maria', 'Woman', '2003-04-29');

insert into persons values(10, 'Belik Daria', 'Woman', '2008-05-29');

insert into persons values(11, 'Belik Ekaterina', 'Woman', '2013-07-16');

/\* Ввод значений родителей \*/

INSERT INTO parents (id, address) VALUES (1, 'Norilsk' );

INSERT INTO parents (id, address) VALUES (2, 'Belgorod' );

INSERT INTO parents (id, address) VALUES (3, 'Tambov' );

INSERT INTO parents VALUES (4, 'Norilsk','TNU\_Vernadskogo');

INSERT INTO parents VALUES (5, 'Norilsk','TNU\_Vernadskogo');

INSERT INTO parents VALUES (6, 'Simferopol','TNU\_Vernadskogo');

/\* Ввод значений детей \*/

INSERT INTO kids VALUES (4, 'peugeot');

INSERT INTO kids VALUES (5, 'lexus');

INSERT INTO kids VALUES (6, 'audi');

INSERT INTO kids VALUES (7, 'kia');

INSERT INTO kids VALUES (8, 'computer');

INSERT INTO kids VALUES (9, 'pets');

INSERT INTO kids VALUES (10, 'art');

INSERT INTO kids VALUES (11, 'swimming');

/\*Ввод значений отношение \*/

INSERT INTO attitude values(1,2, 'married');

INSERT INTO attitude values(1,4, 'kids');

INSERT INTO attitude values(1,5, 'kids');

INSERT INTO attitude values(2,4, 'kids');

INSERT INTO attitude values(2,5, 'kids');

INSERT INTO attitude values(3,6, 'kids');

INSERT INTO attitude values(5,6, 'married');

INSERT INTO attitude values(4,7, 'kids');

INSERT INTO attitude values(5,8, 'kids');

INSERT INTO attitude values(5,10, 'kids');

INSERT INTO attitude values(5,11, 'kids');

INSERT INTO attitude values(6,8, 'kids');

INSERT INTO attitude values(6,10, 'kids');

INSERT INTO attitude values(6,11, 'kids');

INSERT INTO attitude values(8,9, 'married');

----- Запросы -----

Alter table kids drop column id;

----- Запросы -----

Select sum(old\_id + young\_id) from attitude where relations like 'married';

----- Запросы -----

Select count(id), gender from persons group by gender;

----- Запросы -----

Select count(id), gender from persons group by gender having count(id) > 5;

----- Запросы -----

Select persons.\*, parents.\* from persons, parents where persons.id = parents.id and parents.education like 'middle';

----- Запросы -----

Select distinct fio from persons A where not exist (select id from kids where not exists( select id from persons B where B.id = kids.id and A.fio = B.fio));

----- Запросы -----

Select id, interest from persons where id > 3

UNION

Select id, interest from kids where id < 10;

----- Запросы -----

Select distinct fio, id from persons A where not exists

(select id from kids where not exists

(select id from persons B where B.id=kids.id and A.fio=B.fio));

----- Запросы -----

Select persons.\*, kids.interest from persons, kids where (persons.id != 4 and kids.id = persons.id);

----- Запросы -----

Select count(young\_id) from attitude;

----- Запросы -----

Select count(distinct young\_id) from attitude);

**Контрольные вопросы**

1) Поясните действие операции проекции:  
Проекция – выборка из отношения данных по заданным атрибутам.  
  
2) Приведите пример операции селекции:  
Вывести фамилии студентов старше 16 лет.  
РА: R = π surname σ age > 16 (student)  
SQL: SELECT surname FROM student WHERE age > 16;  
  
3) Чем отличаются операции РА соединение и объединение?  
Операция соединения допустима только, если отношения содержат атрибуты, реализованные на одном домене. Реализацию операции соединения можно рассматривать, как операцию декартова произведения, при наличии условия ограничения на общий атрибут отношений, по которому реализовано соединение.  
Операция допустима только над совместными множествами-доменами (одной природы и структуры). Т.е. операция допустима для отношений, имеющих одинаковые схемы. Если отношение имеют разные схемы, но в схемах присутствуют атрибуты, реализованные на одном домене, то операция Объединение допустима для таких отношений только при приведении этих отношений операцией проекцией (над каждым) к одинаковым схемам.  
  
4) Продемонстрируйте на примере, как выразить операцию соединения через декартово произведение:  
РА: R = (course ⋈ teacher)  
SQL: SELECT course.\*, teacher.\* FROM course, teacher  
WHERE course.teacher\_id = [teacher.id](https://vk.com/away.php?to=http%3A%2F%2Fteacher.id&cc_key=);  
  
5) Сформулировать и записать запрос на РА, не реализующийся на SQL.  
Определить дату, когда осуществляют рейсы все возможные типы самолетов.  
РА: flight / plane  
SQL: SELECT DISTINCT date FROM flight A WHERE NOT EXISTS  
(SELECT plane\_type FROM plane WHERE NOT EXISTS  
(SELECT plane\_type FROM flight B WHERE B.plane\_type = plane.type AND [A.date](https://vk.com/away.php?to=http%3A%2F%2FA.date&cc_key=" \t "_blank)=[B.date](https://vk.com/away.php?to=http%3A%2F%2FB.date&cc_key=" \t "_blank)));