Министерство образования и науки Российской Федерации

Севастопольский государственный университет

Институт информационных технологий

Кафедра ИС

# ОТЧЕТ

по расчётно-графической работе

по дисциплине «Управление данными»

Вариант - 12

Выполнил:

ст. гр. ИС/б-21-2-о

Мовенко К. М.

Проверил:

Гончаренко Д. Г.

Севастополь

2023

# ЦЕЛЬ РАБОТЫ

Выработать и закрепить практические навыки по работе с реляционными базами данных, изучить основы языка запросов SQL, научиться создавать таблицы и осуществлять элементарные выборки.

# ЗАДАНИЕ №1

1. В СУБД Firebird была создана база данных firma.fdb (Рисунок 1).

create database 'C:\Users\kosta\Documents\firebird\firma.fdb'

user 'SYSDBA' password 'masterkey'

page\_size = 4096

default character set win1251;

1. Была составлена ER-диаграмма для базы данных, с которой будет вестись работа далее (Рисунок 1).

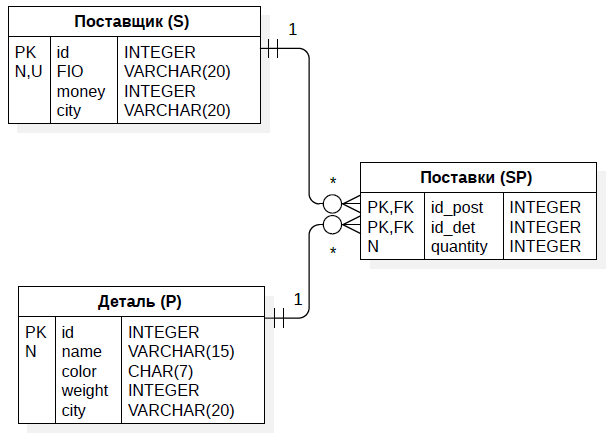


Рисунок 1 – ER-диаграмма базы данных

Была создана и заполнена значениями таблица S, содержащая данные о поставщиках (Рисунок 2).

create table S(

id integer not null check (id > 0),

primary key (id),

FIO varchar(20) not null unique,

money integer,

city varchar(20)

);

insert into S values(1, 'Иванов И.И.', 50000, 'Лондон');

insert into S values(2, 'Петров П.П.', 20000, 'Москва');

insert into S values(3, 'Сидоров С.С.', 150000, 'Париж');

insert into S values(4, 'Васильев В.В.', 300000, 'Лондон');

insert into S values(5, 'Семёнов С.С.', 70000, 'Рим');

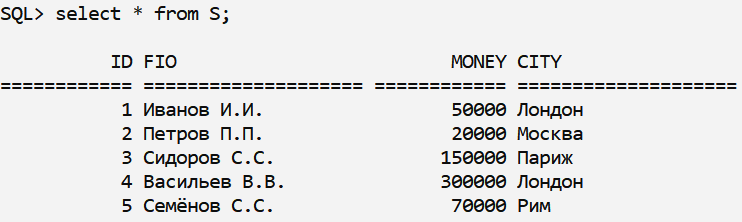


Рисунок 2 – Заполнение таблицы S

Была создана и заполнена значениями таблица P, содержащая данные о поставляемых деталях (Рисунок 3).

create table P(

id integer not null check (id > 0),

primary key (id),

name varchar(15) not null,

color char(7) check (color like '#%'),

weight smallint check (weight > 0),

city varchar(20)

);

insert into P values(1, 'Болт', '#000000', 34, 'Ялта');

insert into P values(4, 'Винт', '#FF0000', 67, 'Ялта');

insert into P values(3, 'Гайка\_мал', '#FFFF00', 18, 'Лондон');

insert into P values(2, 'Гвоздь', '#FFFFFF', 56, 'Париж');

insert into P values(5, 'Петля', '#00FF00', 90, 'Москва');

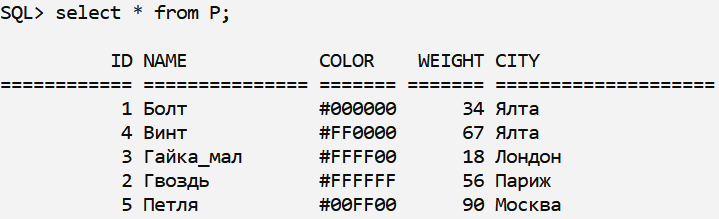


Рисунок 3 – Заполнение таблицы P

Была создана и заполнена значениями таблица SP, содержащая данные о поставках деталей (Рисунок 4).

create table SP(

id\_post integer not null check (id\_post > 0),

id\_det integer not null check (id\_det > 0),

primary key (id\_post, id\_det),

quantity integer not null check (quantity > 0)

);

insert into SP values(2, 3, 120);

insert into SP values(2, 4, 190);

insert into SP values(2, 5, 15);

insert into SP values(1, 4, 18);

insert into SP values(3, 5, 190);

insert into SP values(5, 5, 200);

insert into SP values(1, 5, 100);

insert into SP values(2, 2, 20);

insert into SP values(3, 1, 150);

insert into SP values(4, 3, 30);

insert into SP values(5, 4, 18);

insert into SP values(2, 1, 25);

insert into SP values(1, 3, 14);

insert into SP values(4, 5, 22);

insert into SP values(1, 2, 67);

insert into SP values(3, 3, 15);

insert into SP values(4, 1, 18);

insert into SP values(5, 1, 217);

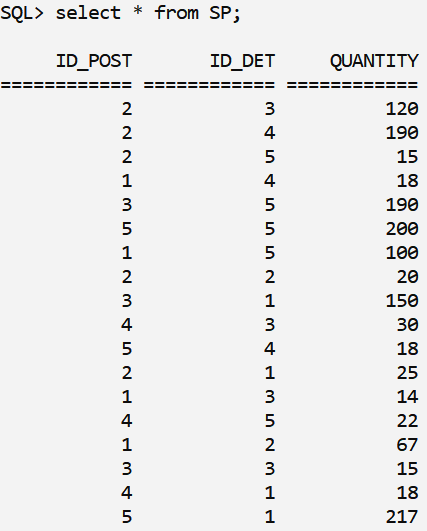


Рисунок 4 – Заполнение таблицы SP

1. Был сделан запрос на вывод номеров всех поставляемых деталей (Рисунок 5).

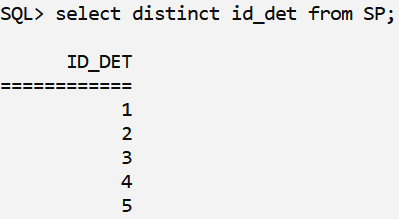


Рисунок 5 – Вывод номеров деталей

1. В таблицу P был добавлен пустой столбец «материал» (Рисунок 6).

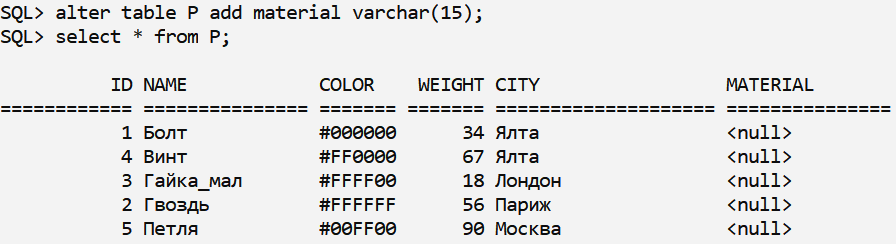


Рисунок 6 – Добавление нового столбца

1. Был сделан запрос на вывод всех поставщиков, не проживающих в Париже (Рисунок 7).

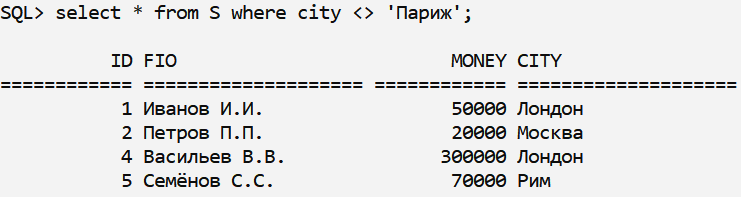


Рисунок 7 – Поставщики, не проживающие в Париже

1. Были выведены номера и состояния всех поставщиков, проживающих в Лондоне (Рисунок 8).

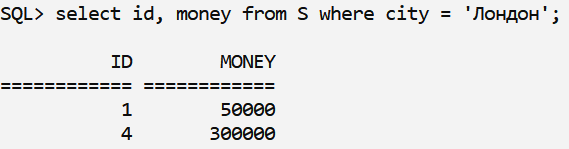


Рисунок 8 – Номера и состояния поставщиков из Лондона

1. Были выведены номера и состояния поставщиков из Москвы, состояние которых меньше 30000$, отсортированные в порядке убывания состояний (Рисунок 9).

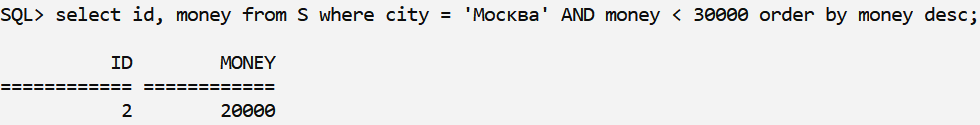


Рисунок 9 – Запрос с условием и сортировкой

1. Были выданы полные характеристики всех поставщиков (Рисунок 2);
2. Были выведены сведения о всех деталях, вес которых находится в промежутке 20-50 грамм (Рисунок 10).

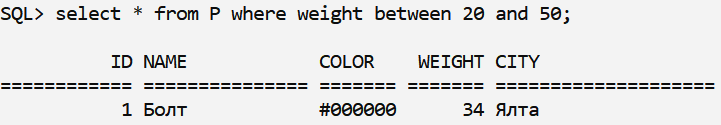


Рисунок 10 – Запрос с условием BETWEEN

1. Был сделан запрос на вывод таблицы P с переводом значений столбца weight из фунтов (454 гр.) в граммы (Рисунок 11).

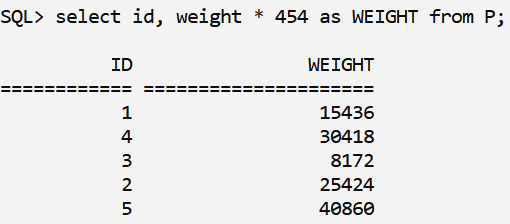


Рисунок 11 – Запрос с операцией над колонкой

1. Был сделан запрос на вывод всех деталей, вес которых 13, 17 или 25 (Рисунок 12). Запрос не выдал результатов, поскольку подобных деталей в таблице нет.



Рисунок 12 – Запрос с условием IN

1. Были выведены детали, начинающиеся на букву «В» (Рисунок 13).
2. Были выведены детали, заканчивающиеся буквой «я» (Рисунок 13).
3. Были выведены детали, содержащие в середине названия букву «а», (Рисунок 13).
4. Были выведены детали, имеющие третьей букву «з» (Рисунок 13). Запрос не дал результатов, т.к. таких деталей нет.
5. Были выведены детали, содержащие в имени символ «\_» (Рисунок 13).

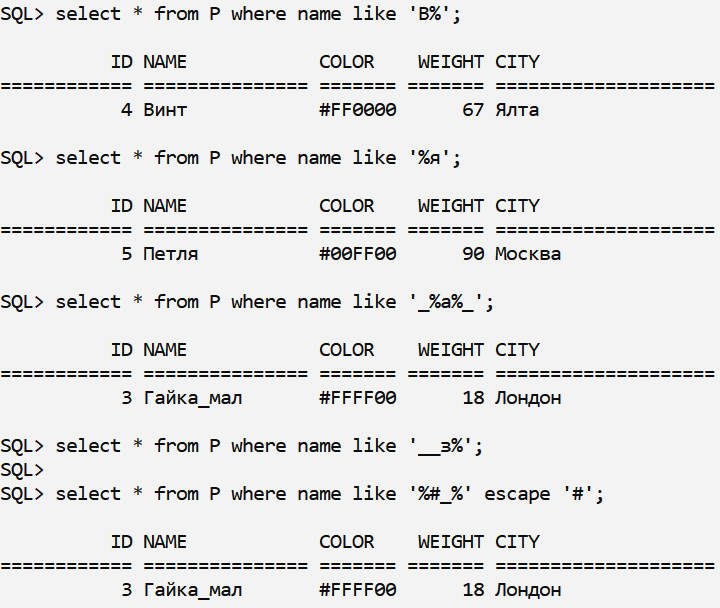


Рисунок 13 – Запросы с условием LIKE

1. Были сделаны запросы на вывод информации о поставщиках и деталях, связанных одним городом, с использованием оператора JOIN (Рисунок 14) и без него (Рисунок 15).

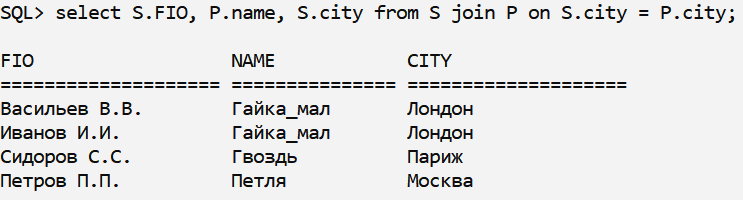


Рисунок 14 – Объединение таблиц с JOIN

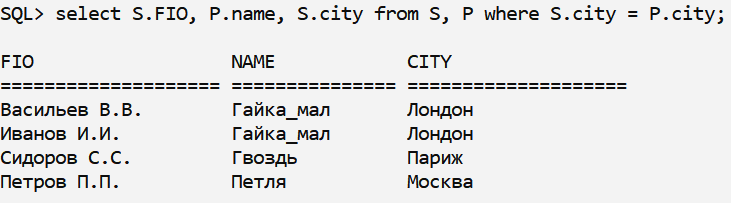


Рисунок 15 – Объединение таблиц без JOIN

1. Цвет детали №2 был изменён на жёлтый (#FFFF00), вес был увеличен на три, значение города установлено на неизвестное (Рисунок 16).

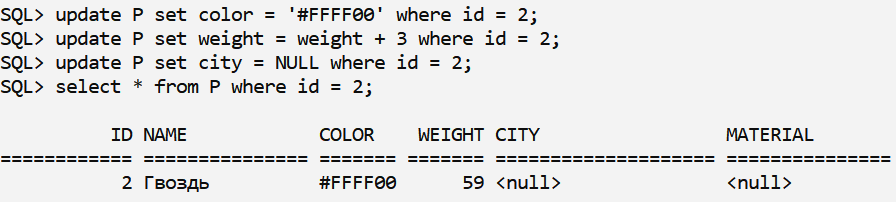


Рисунок 16 – Изменение кортежа таблицы

1. Из таблицы P был удалён столбец «материал» (Рисунок 17).

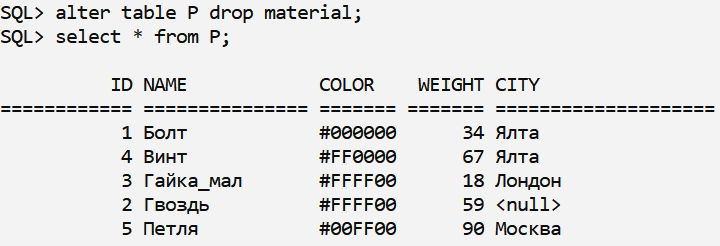


Рисунок 17 – Удаление столбца

1. Состояние всех поставщиков, проживающих в Париже, было увеличено в 4 раза (Рисунок 18).

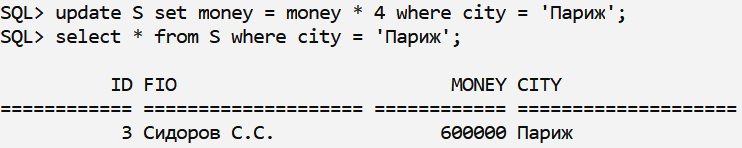


Рисунок 18 – Увеличение состояния поставщиков

1. Из таблицы P были удалены все поставщики из Парижа (Рисунок 19).

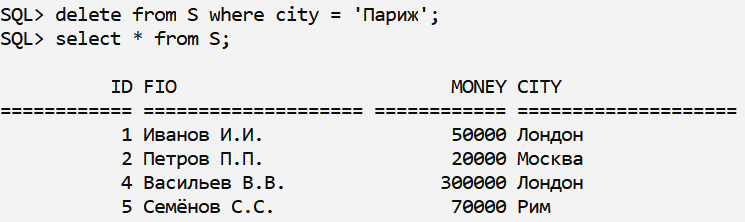


Рисунок 19 – Удаление записей

1. В таблицу P была добавлена новая запись (Рисунок 20).

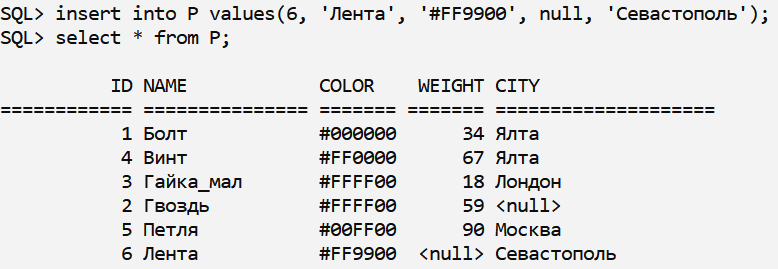


Рисунок 20 – Добавление новой записи

1. Были выведены номера деталей, для которых значение веса не определено (Рисунок 21).

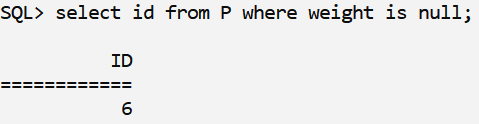


Рисунок 21 – Запрос с IS NULL

1. Было выведено общее количество поставщиков, поставляющих детали на текущий момент (Рисунок 22).

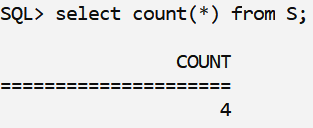


Рисунок 22 – Вывод числа поставщиков

1. Для каждой детали был выведен её объём поставок, т.е. общее число поставленных деталей (Рисунок 23).
2. Для запроса выше было установлено ограничение – не учитывать поставщика под номером 3 (Рисунок 24).

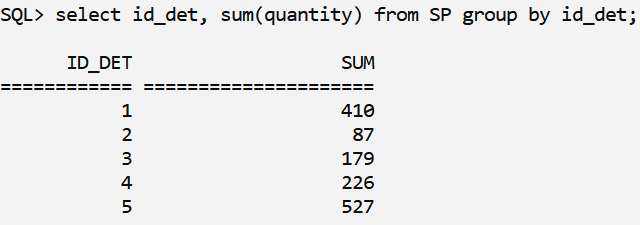


Рисунок 23 – Вывод объёма поставок

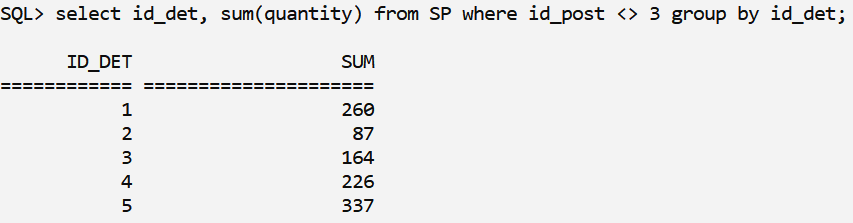


Рисунок 24 – Вывод объёма поставок с ограничением

1. Были выведены номера всех деталей, для которых указано более одного поставщика (Рисунок 25).

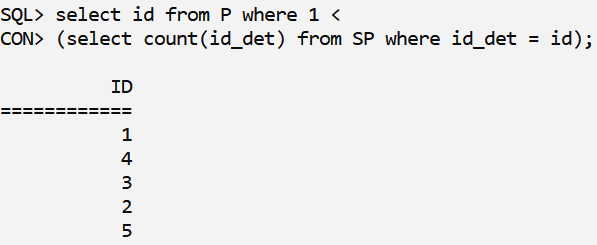


Рисунок 25 – Коррелированный подзапрос

1. Между таблицами S, P и SP были установлены связи с помощью внешних ключей. Для этого в таблицу S был заново введён удалённый ранее поставщик 3.

alter table SP add foreign key (id\_post) references S(id) on delete cascade;

alter table SP add foreign key (id\_det) references P(id) on delete cascade;

1. Были сделаны запросы на вывод информации об именах поставщиков и количестве каждого вида деталей, которые они поставляют, с использованием оператора JOIN (Рисунок 26) и без него (Рисунок 27).
2. Была выведена информация об именах поставщиков и деталях, которые они поставляют (Рисунок 27).

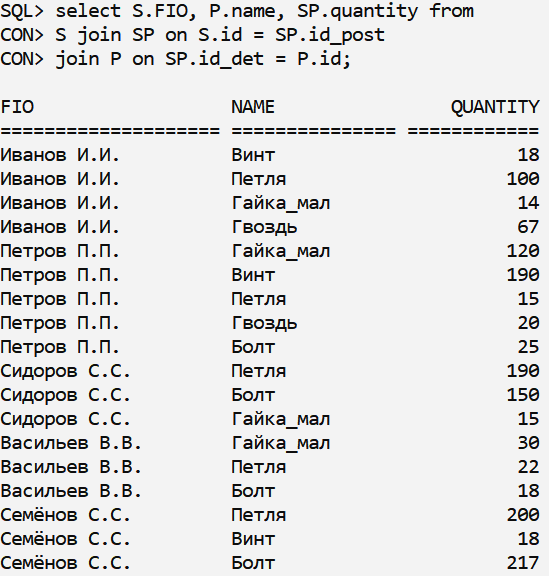


Рисунок 26 – Объединение трёх таблиц через JOIN

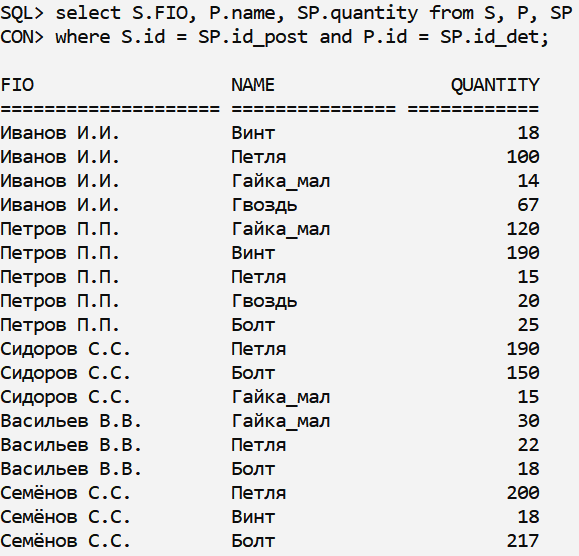


Рисунок 27 – Объединение трёх таблиц через декартово произведение

1. Были сделаны запросы на вывод всех пар поставщиков, проживающих в одном городе, с использованием JOIN (Рисунок 28) и без него (Рисунок 29).

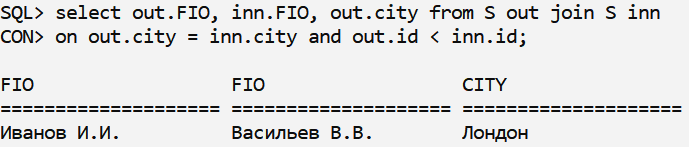


Рисунок 28 – Запрос с псевдонимами и JOIN

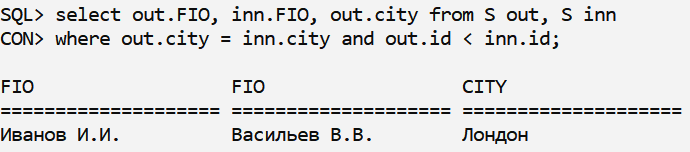


Рисунок 29 – Запрос с псевдонимами без JOIN

1. Были выведены имена всех поставщиков, поставляющих деталь №1 (Рисунок 30).

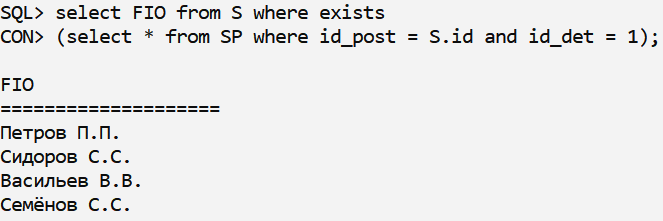


Рисунок 30 – Коррелированный подзапрос с EXISTS

1. Были выведены фамилии поставщиков, поставляющих хотя бы одну красную (#FF0000) деталь (Рисунок 31).

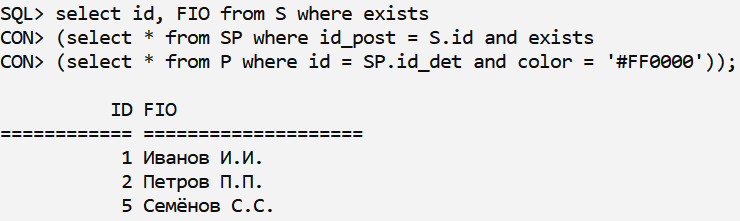


Рисунок 31 – Поставщики, поставляющие красные детали

1. Были выведены поставщики, поставляющие хотя бы одну деталь, поставляемую поставщиком 3, включая его самого (Рисунок 32).

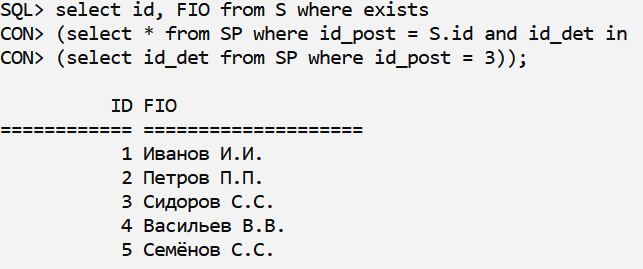


Рисунок 32 – Коррелированный подзапрос с IN

1. Были выведены поставщики, для которых указаны какие-либо поставки (Рисунок 33).
2. Были выведены поставщики, для которых не указано ни одной поставки (Рисунок 33). Подобных найдено не было.

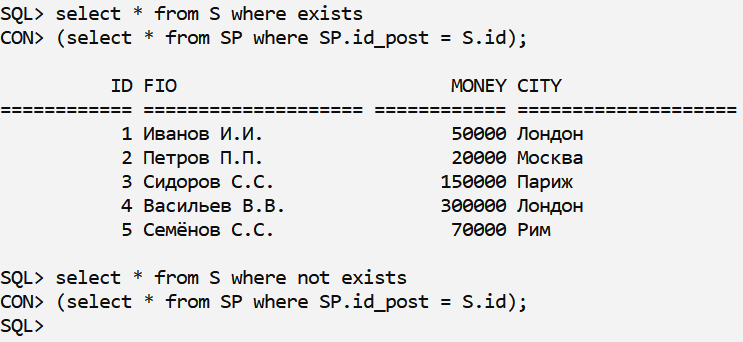


Рисунок 33 – Поставщики, указанные и не указанные в таблице SP

1. Были выведены номера поставщиков, поставляющих все известные детали (Рисунок 34). Для корректности запроса деталь №6 была удалена из таблицы P.

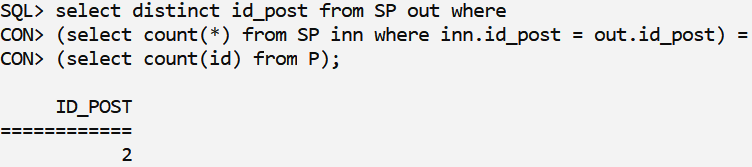


Рисунок 34 – Поставщики, поставляющие все детали

1. Были выведены номера деталей, которые не поставляет поставщик №1 (Рисунок 35).

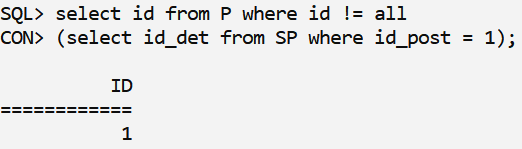


Рисунок 35 – Коррелированный подзапрос с ALL

1. Были выведены номера деталей, вес которых больше 34 или которые поставляются поставщиком №3 (Рисунок 36).

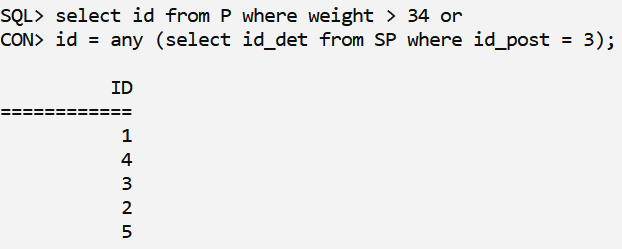


Рисунок 36 – Запрос с условием OR

1. Были выведены названия деталей от поставщика №2 (Рисунок 37).

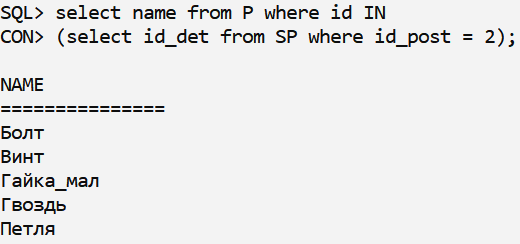


Рисунок 37 – Детали поставщика №2

1. Были выведены номера поставщиков, состояние которых меньше максимального (Рисунок 38).

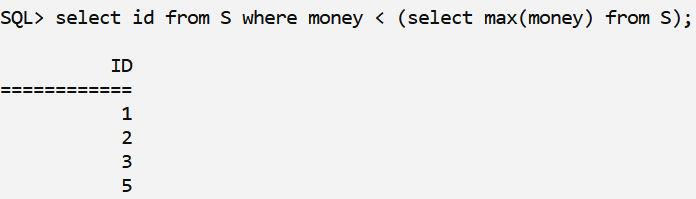


Рисунок 38 – Подзапрос с оператором MAX

1. Были выведены поставщики, состояние которых больше или равно среднему (Рисунок 39).

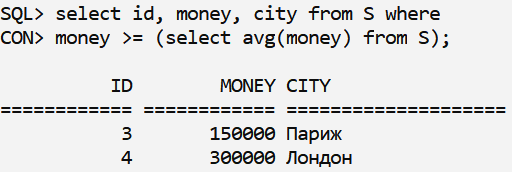


Рисунок 39 – Подзапрос оператором AVG

1. Была создана таблица Q для хранения общего объёма поставок для каждой детали. В неё были внесены значения из таблицы SP (Рисунок 40).

create table Q(

id integer not null,

primary key(id),

foreign key (id) references P(id) on delete cascade,

amount integer not null check (amount > 0)

);

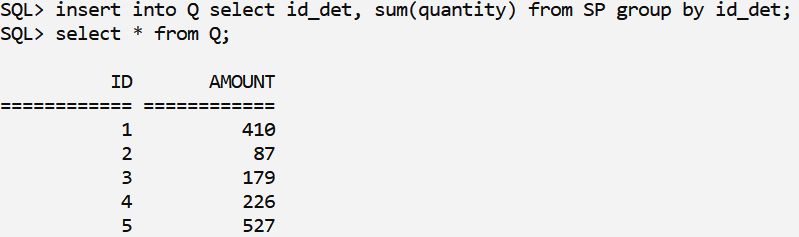


Рисунок 40 – Заполнение таблицы результатами запроса

1. Были выведены имена поставщиков, название деталей с наибольшим суммарным объёмом поставки и количество этих деталей, поставленное каждым из поставщиков. Для этого были использованы запросы с использованием JOIN (Рисунок 41) и без него (Рисунок 42).

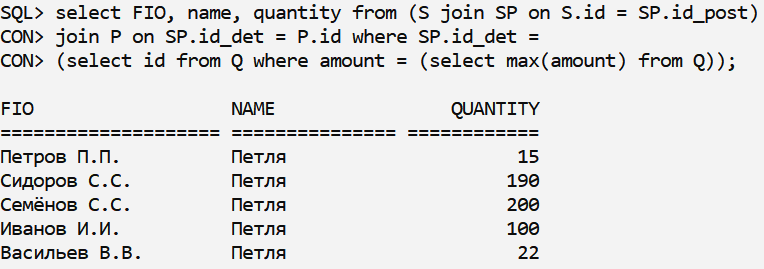


Рисунок 41 – Поставки деталей с наибольшим общим объёмом (с JOIN)

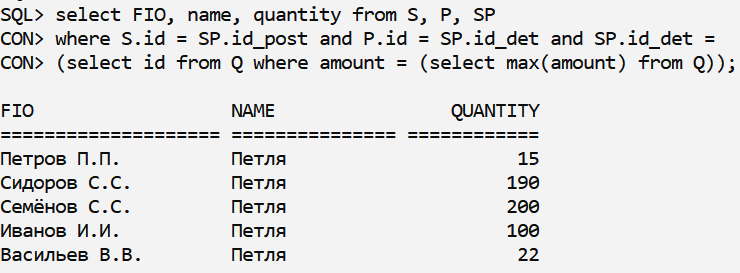


Рисунок 42 – Поставки деталей с наибольшим объёмом (без JOIN)

# ЗАДАНИЕ №2

1. Были выведены все детали, в названии которых не содержится символ «\_» (Рисунок 43).

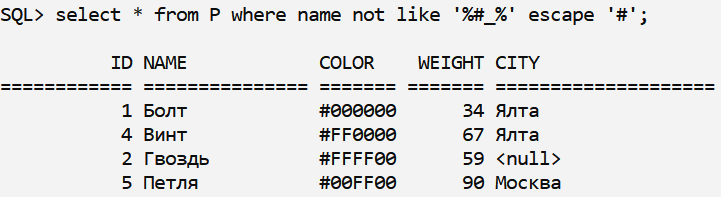


Рисунок 43 – Запрос с условием NOT LIKE

В таблицы БД был внесён ряд изменений для обеспечения корректной работы последующих запросов. Для поставщика №5 городом был указан Краков, в таблицу деталей был добавлен «Цилиндр», для первого поставщика были указаны поставки деталей №1 и №6.

update S set city = 'Краков' where id = 5;

insert into P values (6, 'Цилиндр', '#000000', 45, 'Москва');

insert into SP values (1, 1, 80);

insert into SP values (1, 6, 25);

1. В отношение поставщиков из Кракова были введены санкции – объём поставки каждого товара был уменьшен в 7 раз (Рисунок 44).

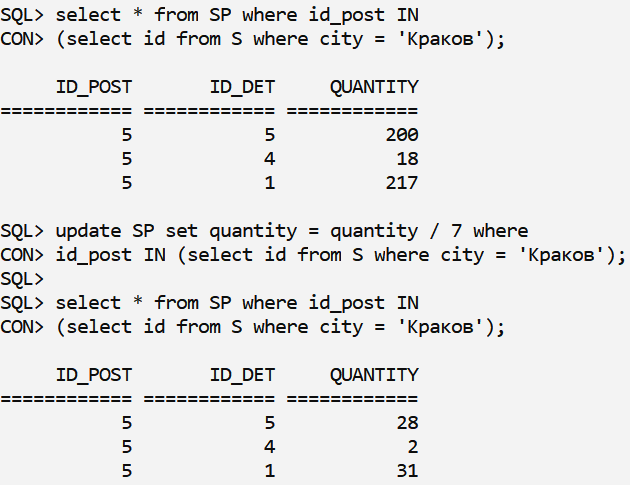


Рисунок 44 – Поставки из Кракова до и после изменения

1. Было выведено количество деталей, поставляемых из Москвы, за исключением белых (цвет #FFFFFF) деталей и цилиндров (Рисунок 45).

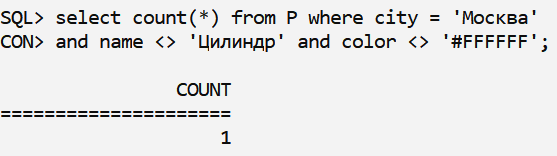


Рисунок 45 – Детали из Москвы с исключениями

1. Было подсчитано количество поставщиков из Лондона, поставляющих все детали (Рисунок 46).

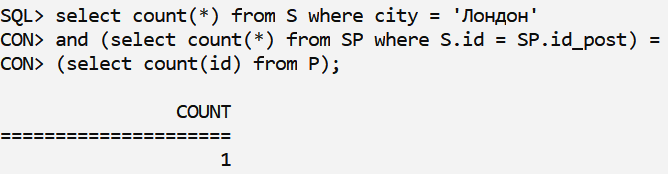


Рисунок 46 – Запрос со сравнением результатов подзапросов

# ЗАДАНИЕ №3

Была составлена ER-диаграмма базы данных для хранения информации о рыбах и аквариумах (Рисунок 47).

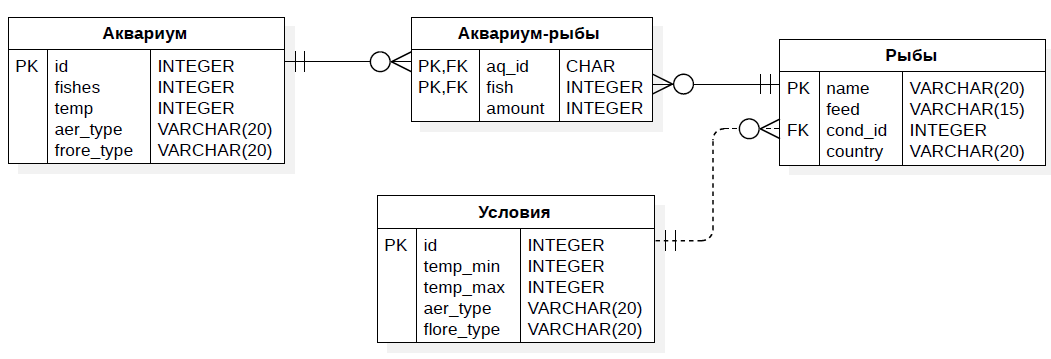


Рисунок 47 – ER-диаграмма базы данных

Была создана и заполнена таблица aquarium, содержащая информацию об аквариумах (Рисунок 48).

create table aquarium(

id integer check(id > 0),

primary key (id),

fishes integer check (fishes > 0),

temp integer,

aer\_type varchar(20),

flore\_type varchar(20)

);

SQL> insert into aquarium values (1, 20, 5, 'Безнапорная', 'Кладофора');

SQL> insert into aquarium values (2, 50, 10, 'Напорная', 'Гигрофила');

SQL> insert into aquarium values (3, 5, 8, 'Напорная', 'Валлиснерия');

SQL> insert into aquarium values (4, 7, 6, 'Инжекционная', 'Кладофора');

SQL> insert into aquarium values (5, 2, 5, 'Безнапорная', 'Гигрофила');

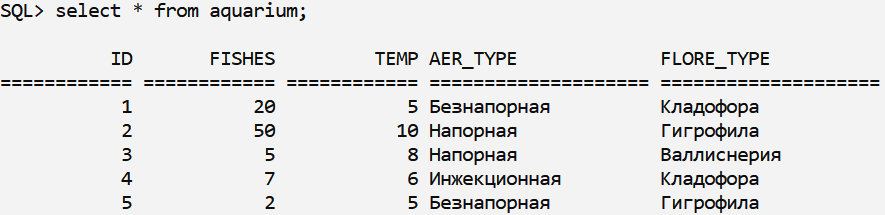


Рисунок 48 – Таблица aquarium

Была создана и заполнена таблица conditions, содержащая информацию о различных условиях содержания рыб (Рисунок 49).

create table conditions(

id integer check (id > 0),

primary key (id),

temp\_min integer,

temp\_max integer,

aer\_type varchar(20),

flore\_type varchar(20)

);

insert into conditions values (1, 2, 10, 'Напорная', 'Валлиснерия');

insert into conditions values (2, 5, 15, 'Напорная', 'Гигрофила');

insert into conditions values (3, 5, 7, 'Инжекционная', 'Кладофора');

insert into conditions values (4, 3, 8, 'Безнапорная', 'Гигрофила');

insert into conditions values (5, 0, 10, 'Безнапорная', 'Кладофора');

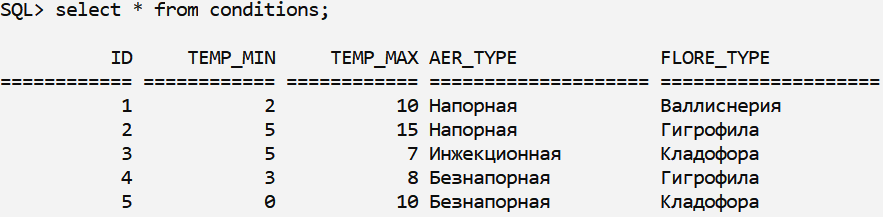


Рисунок 49 – Таблица conditions

Была создана и заполнена таблица fishes, содержащая информацию о различных видах рыб (Рисунок 50).

create table fishes(

name varchar(20),

primary key (name),

feed varchar(15),

cond\_id integer,

foreign key (cond\_id) references conditions(id) on delete set null,

country varchar(20)

);

SQL> insert into fishes values ('Гуппи', 'сухой', 3, 'Австралия');

SQL> insert into fishes values ('Гурами', 'сухой', 1, 'Индия');

SQL> insert into fishes values ('Лабео', 'растения', 2, 'Австралия');

SQL> insert into fishes values ('Меченосец', 'живой', 5, 'Испания');

SQL> insert into fishes values ('Моллинезия', 'живой', 4, 'Япония');

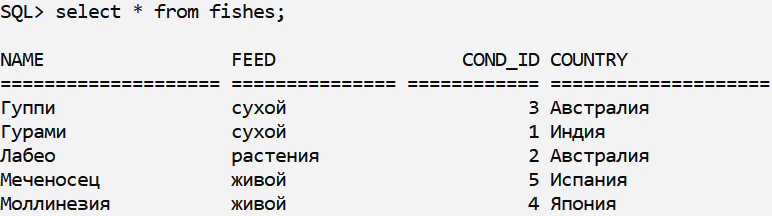


Рисунок 50 – Таблица fishes

Была создана таблица aqfish, связывающая первичные ключи таблиц aquarium и fishes отношением М:М (Рисунок 51).

create table aqfish(

aq\_id integer,

fish varchar(20),

primary key (aq\_id, fish),

foreign key (aq\_id) references aquarium (id) on delete cascade,

foreign key (fish) references fishes (name) on delete cascade,

amount integer check (amount > 0)

);

SQL> insert into aqfish values (1, 'Меченосец', 12);

SQL> insert into aqfish values (2, 'Лабео', 25);

SQL> insert into aqfish values (3, 'Гурами', 5);

SQL> insert into aqfish values (4, 'Гуппи', 5);

SQL> insert into aqfish values (5, 'Моллинезия', 2);

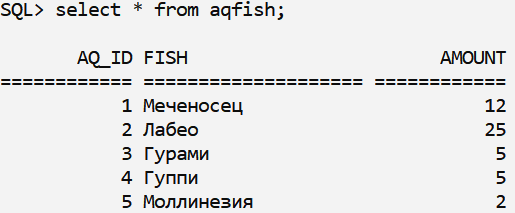


Рисунок 51 – Таблица aqfish

1. Для каждого вида рыбы были выведены его условия содержания, т.е. в запросе были связаны таблицы fishes и conditions. Запрос был сделан сначала с помощью JOIN (Рисунок 52), затем без него (Рисунок 53).

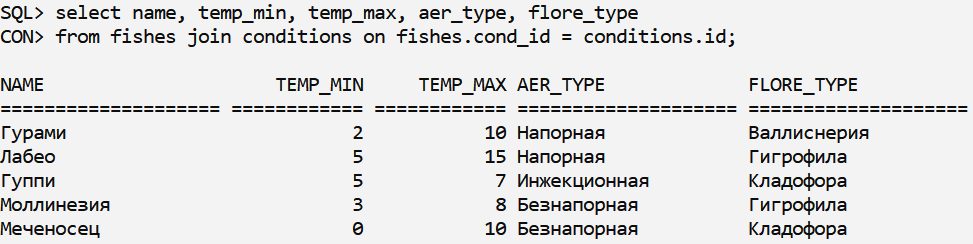


Рисунок 52 – Связывание таблиц fishes и conditions с JOIN

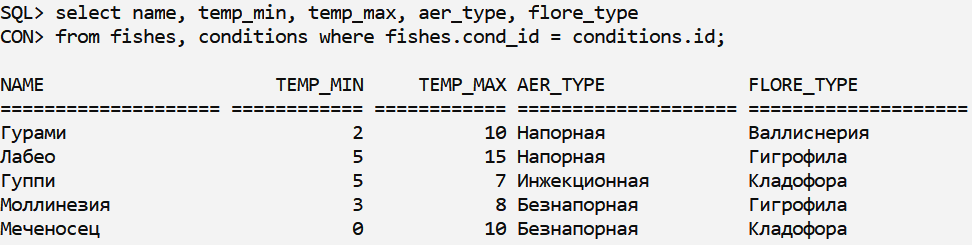


Рисунок 53 - Связывание таблиц fishes и conditions без JOIN

1. Был получен список аквариумов, количество рыб в которых превышает среднее по всем аквариумам (Рисунок 54). Для проверки результата было выведено среднее количество рыб в аквариумах и количество рыб в каждом из аквариумов.

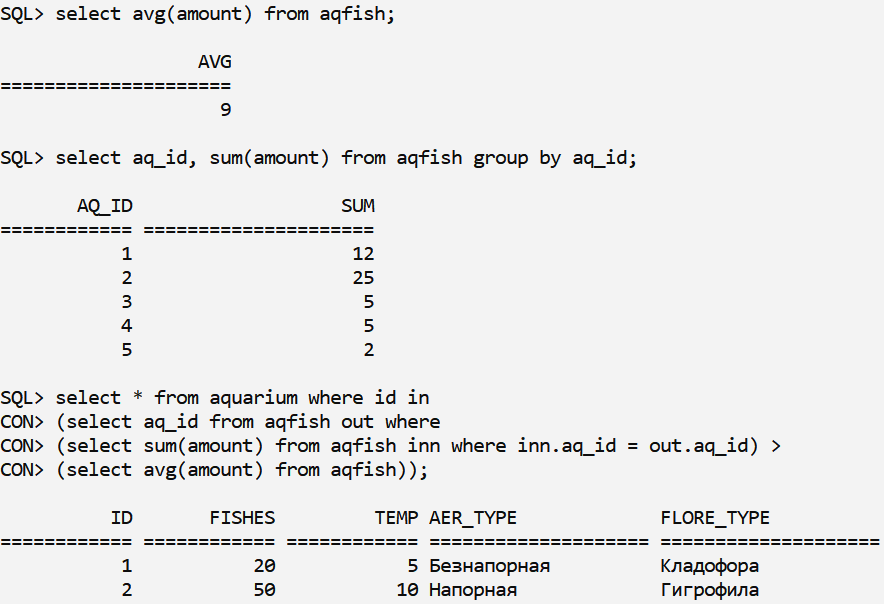


Рисунок 54 – Количество аквариумов, где рыб больше среднего

# ВЫВОД

В ходе работы были продемонстрированы и закреплены основные навыки работы с реляционными базами данных на языке запросов SQL. Было создано и использовано в работе несколько баз данных через СУБД Firebird.

Среди освоенных навыков имеются: создание баз данных, создание таблиц, работа с типами данных, ограничениями, первичными и внешними ключами, написание простых и корреляционных запросов, применение агрегатных функций, объединение таблиц различными методами и др.