**1 Лабораторная работа №1**

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

**1.1 Цель работы:**

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

**1.2 Постановка задачи**

1.2.1 Проанализировать схему БД (Рисунок 1.1) своего варианта задания (вариант 5), выделить и классифицировать все существующие связи, определить необходимые ограничения целостности.

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

1.2.3 Установить связи между таблицами.

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

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

1.2.6 Продемонстрировать выполнение простых вычислений в запросе.

1.2.7 Использовать простое вычисление как параметр агрегатной функции.

1.2.8 Продемонстрировать работу предложения GROUP BY.

1.2.9 Продемонстрировать работу предложения HAVING.

1.2.10 Ознакомиться с операциями реляционной алгебры.

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

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

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

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

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

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

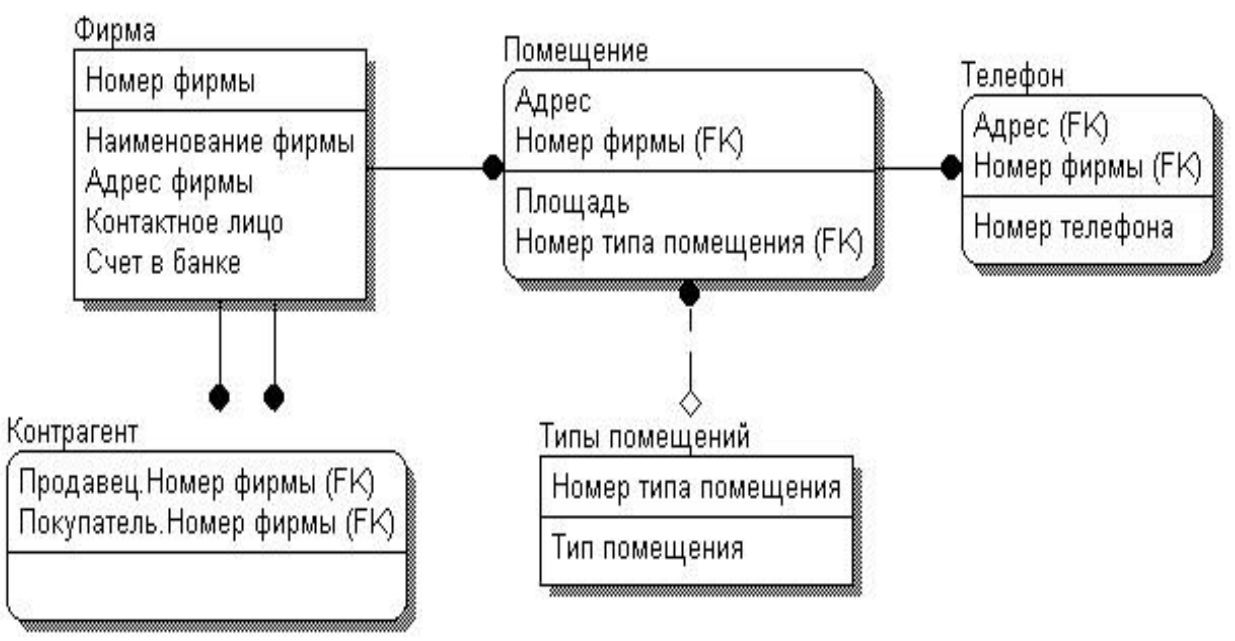


Рисунок 1.1 — Структура системы, согласно варианту

**1.3 Ход работы**

1.3.1 Был запущен Firebird ISQL TOOL, после чего была создана новая база данных:

Создать базу данных “по указанному пути” пользователь ‘SYSDBA’ пароль ‘masterkey’

Размер страницы = 4096 кодировка базы данных по умолчанию win1251;

CREATE DATABASE "C:\Users\Danil\_KHR\Desktop\Git\5\_semester\Databases\Lab\_1\Firms.fdb"

user 'SYSDBA' password 'masterkey'

page\_size = 4096

default character set win1251;

Затем были созданы таблицы, сразу со связями:

Создать таблицу Фирма (

Номер фирмы, целочисленного типа, не нулевое значение, Первичный ключ,

Наименование фирмы, символьного не фиксированного в 20 символов типа, не нулевое значение,

Адресс фирмы, символьного не фиксированного в 100 символов типа, не нулевое значение,

Контактное лицо, символьного не фиксированного в 40 символов типа, не нулевое значение,

Счёт в банке, символьного фиксированного в 20 символов типа, не нулевое значение

);

CREATE TABLE Company (

company\_num INTEGER NOT NULL PRIMARY KEY,

company\_name VARCHAR(20) NOT NULL,

company\_adress VARCHAR(100) NOT NULL,

contact\_person VARCHAR(40) NOT NULL,

bank\_account CHAR(20) NOT NULL

);

Создать таблицу Тип помещения (

Номер типа помещения, целочисленного типа, Первичный ключ,

Тип помещения, символьного не фиксированного в 20 символов типа );

CREATE TABLE Room\_type (

room\_type\_num INTEGER PRIMARY KEY,

room\_type VARCHAR(20)

);

Создать таблицу помещение (

Адрес, символьного не фиксированного в 100 символов типа, не нулевое значение, Первичный ключ,

Номер фирмы, целочисленного типа, не нулевое значение,

Площадь, целочисленного типа, не нулевое значение,

Номер типа помещения, целочисленного типа,

Внешний ключ (Номер типа помещения) ссылается на таблицу «Тип помещения» (Номер типа помещения),

Внешний ключ (Номер компании) ссылается на таблицу «Фирма» (Номер фирмы)

);

CREATE TABLE Room (

adress VARCHAR(100) NOT NULL PRIMARY KEY,

company\_num INTEGER NOT NULL,

area\_size INTEGER NOT NULL,

room\_type\_num INTEGER,

FOREIGN KEY (room\_type\_num) REFERENCES Room\_type (room\_type\_num),

FOREIGN KEY (company\_num) REFERENCES Company (company\_num)

);

Создать таблицу телефон (

Адрес, символьного не фиксированного в 100 символов типа, не нулевое значение,

Номер фирмы, целочисленного типа, не нулевое значение,

Номер телефона, символьного не фиксированного в 20 символов типа,

Внешний ключ (адрес) ссылается на таблицу «Помещение» (адрес),

Внешний ключ (номер фирмы) ссылается на таблицу «Фирма» (номер фирмы)

);

CREATE TABLE Telephone (

adress VARCHAR(100) NOT NULL,

company\_num INTEGER NOT NULL,

phone\_num VARCHAR(20),

FOREIGN KEY (adress) REFERENCES Room (adress),

FOREIGN KEY (company\_num) REFERENCES Company (company\_num)

);

Создать таблицу Контрагент (

Продавец. Номер фирмы, целочисленного типа, не нулевое значение,

Покупатель. Номер фирмы, целочисленного типа, не нулевое значение,

Внешний ключ (Продавец. Номер фирмы) ссылается на таблицу «Фирма» (номер фирмы),

Внешний ключ (Покупатель. Номер фирмы) ссылается на таблицу «Фирма» (номер фирмы)

);

CREATE TABLE Counterparty (

seller\_company\_num INTEGER NOT NULL,

buyer\_company\_num INTEGER NOT NULL,

FOREIGN KEY (seller\_company\_num) REFERENCES Company (company\_num),

FOREIGN KEY (buyer\_company\_num) REFERENCES Company (company\_num)

);

1.3.2 В базу данных были занесены записи с помощью INSERT INTO. Конструкция занесения, следующая: «Insert into» «Наименование таблицы» (атрибуты таблицы) «VALUES» (значения таблицы). Среди значений таблицы символьные атрибуты берутся в одинарные кавычки при заполнении. На пример:

INSERT INTO Room\_type (room\_type\_num, room\_type) VALUES (8897460, 'Workshop');

В результате у нас получились 5 таблиц со связями согласно варианту. Результат создания и заполнения таблиц отображен на рисунках 1.2 – 1.6 с помощью фразы «SELECT \* FROM Название таблицы», что означает вывести на экран все кортежи в данной таблице.

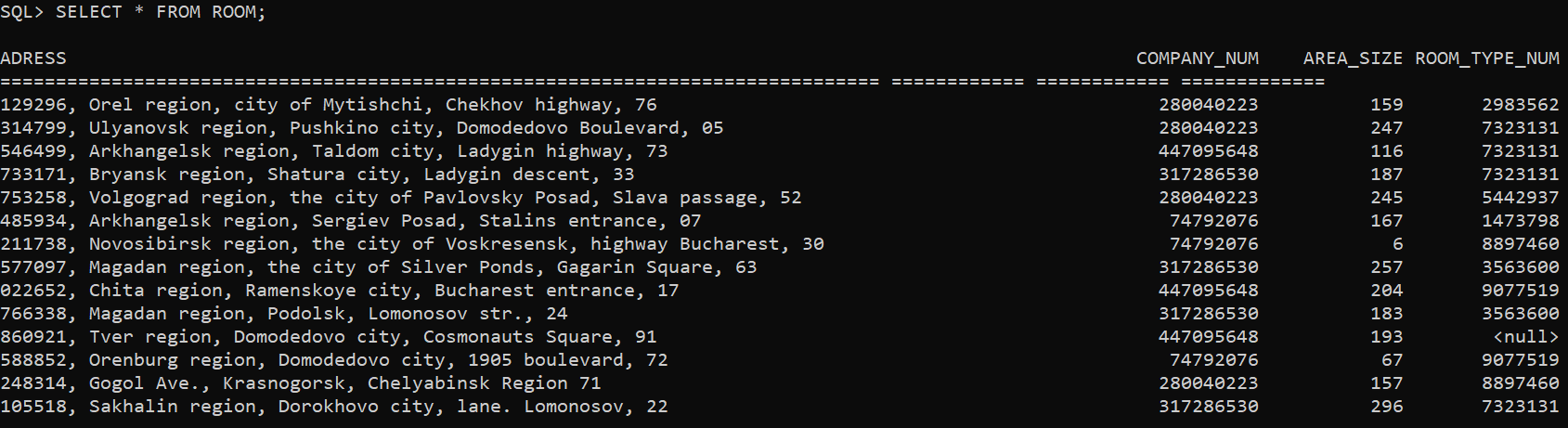
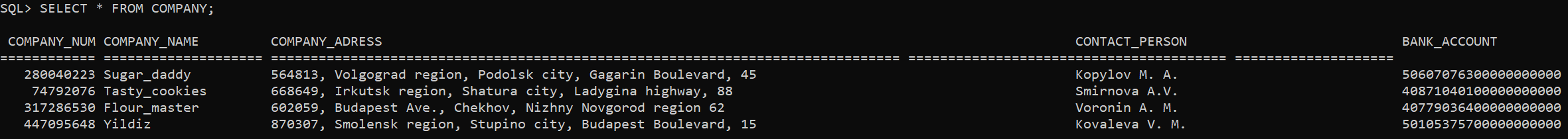
****

Рисунок 3.2 – Вывод кортежей из таблицы «Помещение»

****Рисунок 3.3 – Вывод кортежей из таблицы «Фирма»

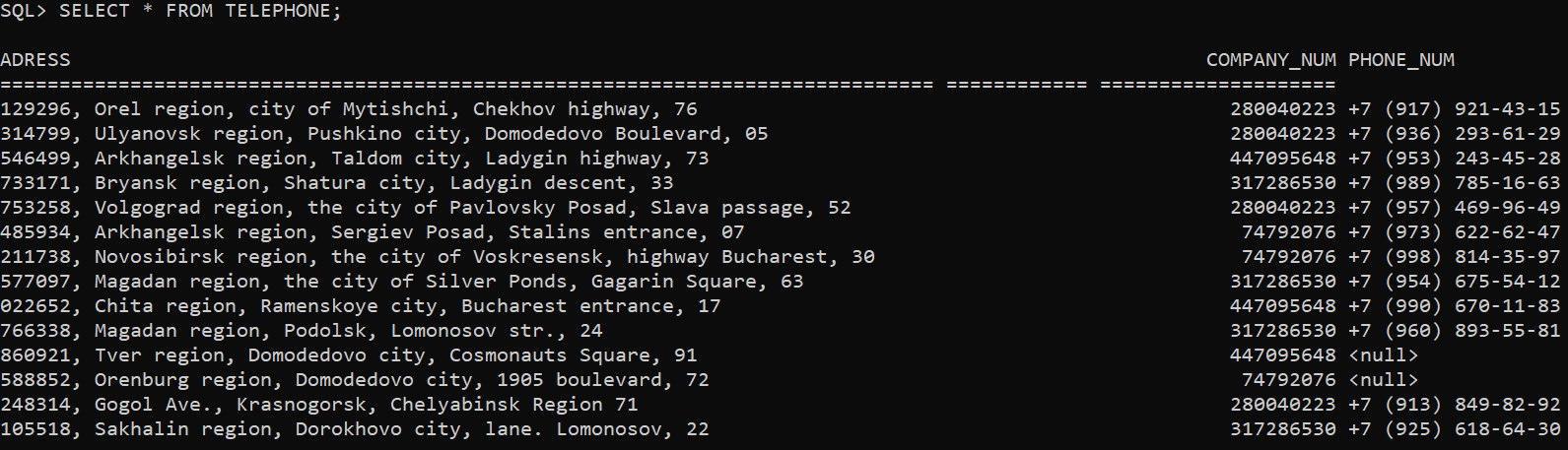
****

Рисунок 3.4 – Вывод кортежей из таблицы «Телефон»

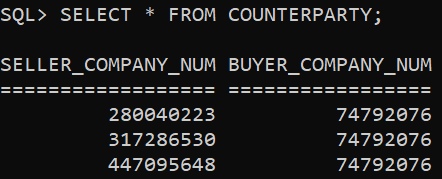
****

Рисунок 3.5 – Вывод кортежей из таблицы «Контрагент»

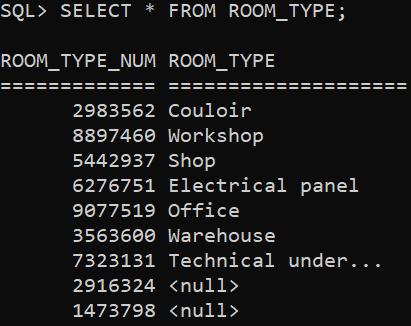
****

Рисунок 3.6 – Вывод кортежей из таблицы «Типы помещений»

1.3.3 Для проверки работы «Первичных ключей» было решено удалить кортеж содержащий первичный ключ. В результате firebird isql tool не позволил этого сделать, так как поле «Номер фирмы» содержит первичный ключ. Запрос был следующий:

Удалить из таблицы «Фирма» кортеж, где номер фирмы = 317286530;

Результат выполнения запроса отображен на рисунке 1.7. В то же время если удалить кортеж содержащий внешний ключ, это успешно удастся сделать.

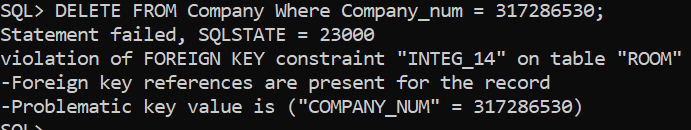


Рисунок 1.7 – Ошибка при удалении кортежа с первичным ключом

Аналогично при попытке модифицировать значения атрибутов с Первичным ключом происходит такая же ошибка, как и при попытке удаления. Запрос был следующий:

Модифицировать в таблице «Тип помещения», поставить номер = 1234567, где номер был = 9077519;

Результат выполнения запроса отображен на рисунке 1.8.

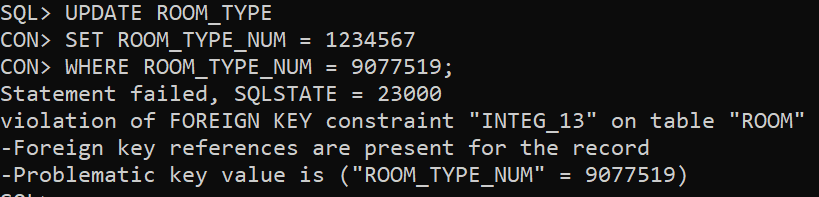


Рисунок 1.8 – Ошибка при модификации записи атрибута с первичным ключом

1.3.4 Для демонстрации простых вычислений в запросе был сделан следующий запрос:

Вывести номер фирмы, половину размера помещения из таблицы помещение;

SELECT COMPANY\_NUM, AREA\_SIZE\*0.5 AS half\_area\_size FROM ROOM;

Результат работы запроса представлен на рисунке 1.9.

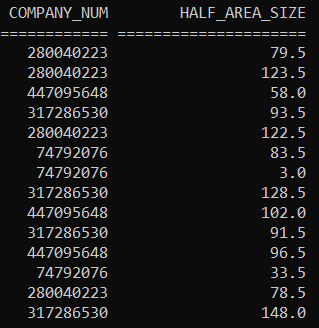


Рисунок 1.9 – Результат выполнения простого вычисления в запросе

1.3.5 Использовалось простое вычисление как параметр агрегатной функции с помощью следующего запроса:

Вывести среднее значение размера помещения умноженное на2 среди всех помещений в таблице помещение;

SELECT AVG(AREA\_SIZE\*2) AS AVG\_AREA\_SIZE\_X2 FROM ROOM;

Результат работы запроса представлен на рисунке 1.10.

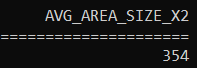


Рисунок 1.10 – Демонстрация использования простого вычисления как параметра агрегатной функции

1.3.6 Для демонстрации работы предложения GROUP BY и HAVING были сделаны следующие запросы и результаты из работы представлены на рисунках 1.11 – 1.12:

Вывести название фирмы и соответствующее ей максимально большое помещение сортируя по названию фирмы;

SELECT COMPANY\_NUM, MAX(AREA\_SIZE) FROM ROOM GROUP BY COMPANY\_NUM;

Аналогичный запрос, в конце дополнительно выводящий только те фирмы, у которых максимально большое помещение больше 200м­­2;

SELECT COMPANY\_NUM, MAX(AREA\_SIZE) FROM ROOM GROUP BY COMPANY\_NUM HAVING MAX(AREA\_SIZE) > 200;

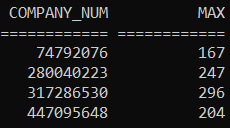


Рисунок 1.11 – Демонстрация работы предложения GROUP BY

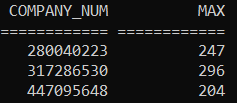


Рисунок 1.12 – Демонстрация работы предложения HAVING

1.3.7 Произошло ознакомление с операциями реляционной алгебры.