**3 ЛАБОРАТОРНАЯ РАБОТА №3**

**«СОЗДАНИЕ СХЕМЫ БД. ССЫЛОЧНАЯ ЦЕЛОСТНОСТЬ»**

**3.1 Цель работы**

Ознакомится с типами связей между отношениями. Узнать требования к ссылочной целостности данных. Изучить ER-диаграммы.

**3.2 Вариант задания – 5**

Требуется проанализировать схему БД своего варианта задания (рисунок 3.1);

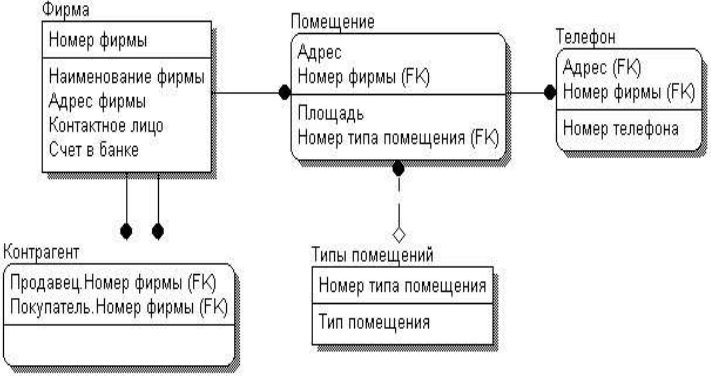


Рисунок 3.1 – Схема базы данных

Создать все еще не созданные таблицы и изменить существующие таким образом, чтобы они могли участвовать в связях;

Установить связи между таблицами;

Проверить работу ограничений целостности;

**3.3 Ход работы**

3.3.1 Был запущен Firebird ISQL TOOL, после чего была создана новая база данных для третьей лабораторной работы.

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

Размер страницы = 4096 кодировка базы данных по умолчанию win1251;

CREATE DATABASE "D:\SevSU\_IS\4\_Sem\Data management\lab\_3\DATABASE.fdb"

user 'SYSDBA' password 'masterkey'

page\_size = 4096

default character set win1251;

Затем были созданы таблицы с использованием primary и foreign keys согласно варианту.

Таблица «Фирма» была создана с помощью следующего запроса:

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

);

Таблица «Тип помещения» была создана с помощью следующего запроса:

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)

);

Таблица «Помещение» была создана с помощью следующего запроса:

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)

);

Таблица «Телефон» была создана с помощью следующего запроса:

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)

);

Таблица «Контрагент» была создана с помощью следующего запроса:

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)

);

3.3.2 Затем с помощью фразы «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 3.2 – 3.6 с помощью фразы «SELECT \* FROM Название таблицы», что означает вывести на экран все кортежи в данной таблице.

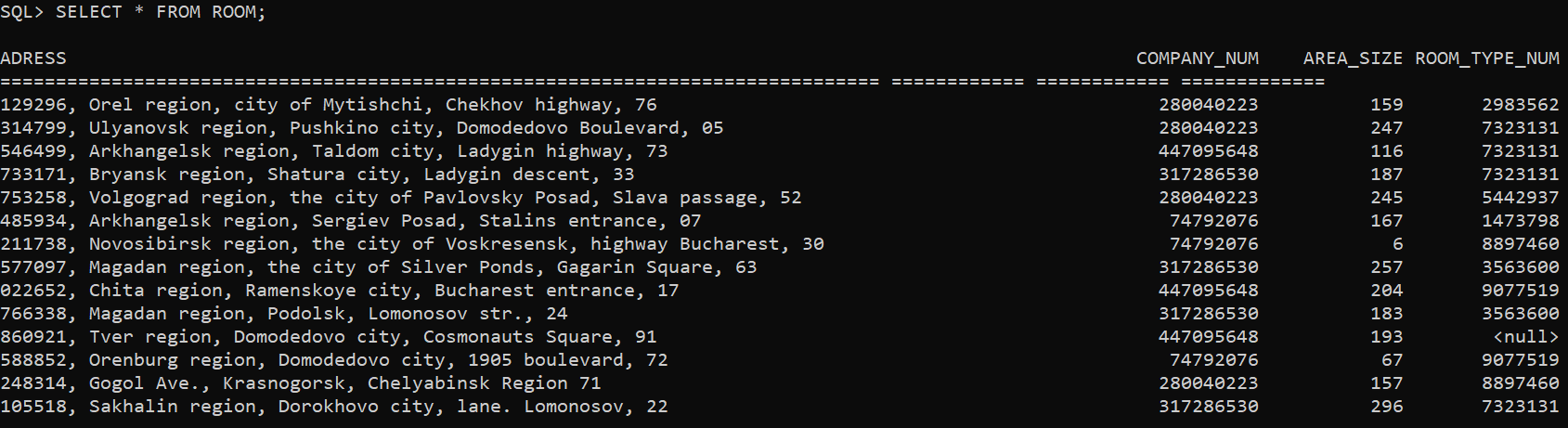
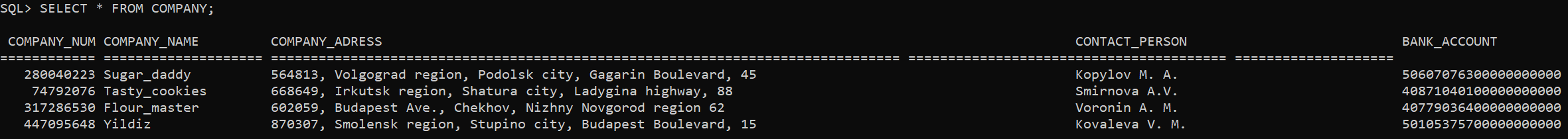
****

Рисунок 3.2 – Вывод кортежей из таблицы «Помещение»

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

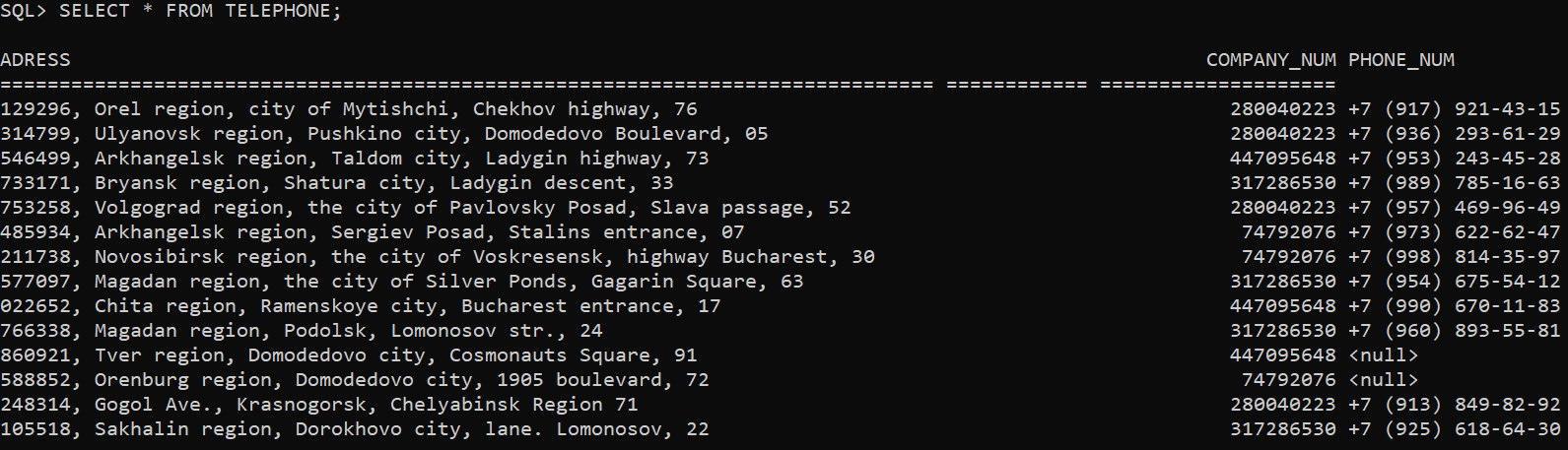
****

Рисунок 3.4 – Вывод кортежей из таблицы «Телефон»

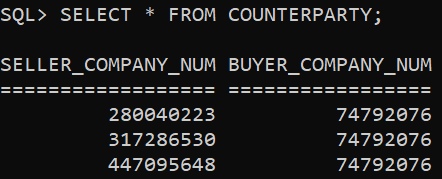
****

Рисунок 3.5 – Вывод кортежей из таблицы «Контрагент»

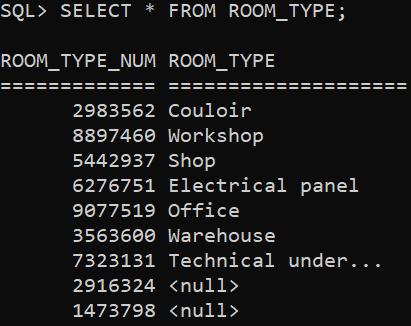
****

Рисунок 3.6 – Вывод кортежей из таблицы «Типы помещений»

3.3.3 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 3.7. В то же время если удалить кортеж содержащий внешний ключ, это успешно удастся сделать.

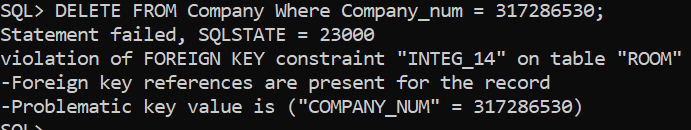


Рисунок 3.7 – Ошибка при удалении кортежа с первичным ключом

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 3.8.

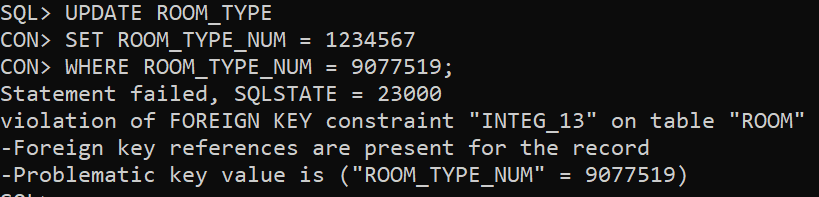


Рисунок 3.8 – Ошибка при модификации записи атрибута с первичным ключом

**Выводы**

При выполнении данной лабораторной работы были получены навыки работы с первичными и вторичными ключами. Изучены ER-диаграммы. Был сделан вывод, что при попытке удаления или модификации атрибута таблицы, содержащего первичный ключ, действия закончистя не успехом, для того чтобы не нарушалась целостность таблицы. Полученные навыки и знания помогут в будущем эффективнее создавать базы данных и работать с ними.