

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.

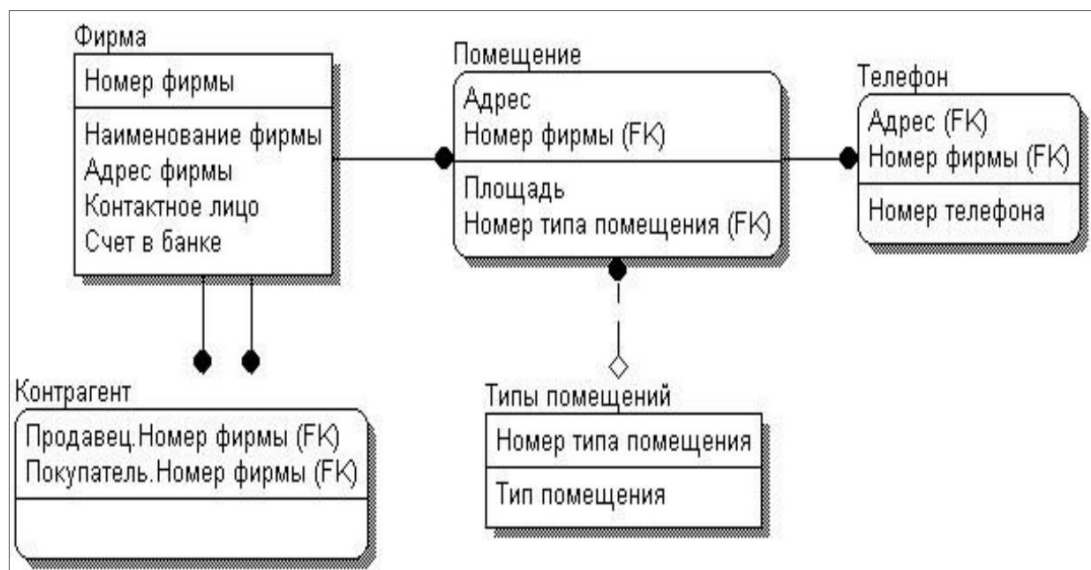


Рисунок 1.1 — Структура системы, согласно варианту

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 (
 Номер телефона, целочисленного типа, Первичный ключ,
 Номер помещения, целочисленного типа, Внешний ключ (ссылается на таблицу «Помещение»
 (Номер помещения)),
 Номер фирмы, целочисленного типа, Внешний ключ (ссылается на таблицу «Фирма»
 (Номер фирмы))

```

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 (
    address      VARCHAR(100) NOT NULL,
    company_num  INTEGER      NOT NULL,
    phone_num    VARCHAR(20),
    FOREIGN KEY (address) REFERENCES Room (address),
    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.

```
SQL> SELECT * FROM ROOM;
```

ADDRESS	COMPANY_NUM	AREA_SIZE	ROOM_TYPE_NUM
129296, Orel region, city of Mytishchi, Chekhov highway, 76	280040223	159	2983562
314799, Ulyanovsk region, Pushkino city, Domodedovo Boulevard, 05	280040223	247	7323131
546499, Arkhangelsk region, Taldom city, Ladygin highway, 73	447095648	116	7323131
733171, Bryansk region, Shatura city, Ladygin descent, 33	317286530	187	7323131
753258, Volgograd region, the city of Pavlovsky Posad, Slava passage, 52	280040223	245	5442937
485934, Arkhangelsk region, Sergiev Posad, Stalins entrance, 07	74792076	167	1473798
211738, Novosibirsk region, the city of Voskresensk, highway Bucharest, 30	74792076	6	8897460
577097, Magadan region, the city of Silver Ponds, Gagarin Square, 63	317286530	257	3563600
022652, Chita region, Ramenskoye city, Bucharest entrance, 17	447095648	204	9077519
766338, Magadan region, Podolsk, Lomonosov str., 24	317286530	183	3563600
860921, Tver region, Domodedovo city, Cosmonauts Square, 91	447095648	193	<null>
588852, Orenburg region, Domodedovo city, 1905 boulevard, 72	74792076	67	9077519
248314, Gogol Ave., Krasnogorsk, Chelyabinsk Region 71	280040223	157	8897460
105518, Sakhalin region, Dorokhovo city, lane. Lomonosov, 22	317286530	296	7323131

Рисунок 3.2 – Вывод кортежей из таблицы «Помещение»

```
SQL> SELECT * FROM COMPANY;
```

COMPANY_NUM	COMPANY_NAME	COMPANY_ADDRESS	CONTACT_PERSON	BANK_ACCOUNT
280040223	Sugar_daddy	564813, Volgograd region, Podolsk city, Gagarin Boulevard, 45	Kopylov M. A.	56607076300000000000
74792076	Tasty_cookies	668649, Irkutsk region, Shatura city, Ladygina highway, 88	Smirnova A.V.	48871040100000000000
317286530	Flour_master	602059, Budapest Ave., Chekhov, Nizhny Novgorod region 62	Voronin A. M.	48779036400000000000
447095648	Yildiz	870307, Smolensk region, Stupino city, Budapest Boulevard, 15	Kovaleva V. M.	50105375700000000000

Рисунок 3.3 – Вывод кортежей из таблицы «Фирма»

```
SQL> SELECT * FROM TELEPHONE;
```

ADDRESS	COMPANY_NUM	PHONE_NUM
129296, Orel region, city of Mytishchi, Chekhov highway, 76	280040223	+7 (917) 921-43-15
314799, Ulyanovsk region, Pushkino city, Domodedovo Boulevard, 05	280040223	+7 (936) 293-61-29
546499, Arkhangelsk region, Taldom city, Ladygin highway, 73	447095648	+7 (953) 243-45-28
733171, Bryansk region, Shatura city, Ladygin descent, 33	317286530	+7 (989) 785-16-63
753258, Volgograd region, the city of Pavlovsky Posad, Slava passage, 52	280040223	+7 (957) 469-96-49
485934, Arkhangelsk region, Sergiev Posad, Stalins entrance, 07	74792076	+7 (973) 622-62-47
211738, Novosibirsk region, the city of Voskresensk, highway Bucharest, 30	74792076	+7 (998) 814-35-97
577097, Magadan region, the city of Silver Ponds, Gagarin Square, 63	317286530	+7 (954) 675-54-12
022652, Chita region, Ramenskoye city, Bucharest entrance, 17	447095648	+7 (990) 670-11-83
766338, Magadan region, Podolsk, Lomonosov str., 24	317286530	+7 (960) 893-55-81
860921, Tver region, Domodedovo city, Cosmonauts Square, 91	447095648	<null>
588852, Orenburg region, Domodedovo city, 1905 boulevard, 72	74792076	<null>
248314, Gogol Ave., Krasnogorsk, Chelyabinsk Region 71	280040223	+7 (913) 849-82-92
105518, Sakhalin region, Dorokhovo city, lane. Lomonosov, 22	317286530	+7 (925) 618-64-30

Рисунок 3.4 – Вывод кортежей из таблицы «Телефон»

```
SQL> SELECT * FROM COUNTERPARTY;
```

SELLER_COMPANY_NUM	BUYER_COMPANY_NUM
280040223	74792076
317286530	74792076
447095648	74792076

Рисунок 3.5 – Вывод кортежей из таблицы «Контрагент»

```
SQL> SELECT * FROM ROOM_TYPE;
```

ROOM_TYPE_NUM	ROOM_TYPE
2983562	Couloir
8897460	Workshop
5442937	Shop
6276751	Electrical panel
9077519	Office
3563600	Warehouse
7323131	Technical under...
2916324	<null>
1473798	<null>

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.

```
SQL> DELETE FROM Company Where Company_num = 317286530;  
Statement failed, SQLSTATE = 23000  
violation of FOREIGN KEY constraint "INTEG_14" on table "ROOM"  
-Foreign key references are present for the record  
-Problematic key value is ("COMPANY_NUM" = 317286530)  
SQL>
```

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.

```
SQL> UPDATE ROOM_TYPE  
CON> SET ROOM_TYPE_NUM = 1234567  
CON> WHERE ROOM_TYPE_NUM = 9077519;  
Statement failed, SQLSTATE = 23000  
violation of FOREIGN KEY constraint "INTEG_13" on table "ROOM"  
-Foreign key references are present for the record  
-Problematic key value is ("ROOM_TYPE_NUM" = 9077519)  
SQL>
```

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.

COMPANY_NUM	HALF_AREA_SIZE
280040223	79.5
280040223	123.5
447095648	58.0
317286530	93.5
280040223	122.5
74792076	83.5
74792076	3.0
317286530	128.5
447095648	102.0
317286530	91.5
447095648	96.5
74792076	33.5
280040223	78.5
317286530	148.0

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.

AVG_AREA_SIZE_X2
354

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;
```

COMPANY_NUM	MAX
=====	=====
74792076	167
280040223	247
317286530	296
447095648	204

Рисунок 1.11 – Демонстрация работы предложения GROUP BY

COMPANY_NUM	MAX
=====	=====
280040223	247
317286530	296
447095648	204

Рисунок 1.12 – Демонстрация работы предложения HAVING

1.3.7 Произошло ознакомление с операциями реляционной алгебры.

1.3.8 Были применены операции селекции и соединения, составлен следующий запрос: вывести типы помещения, размер которых больше 200 квадратных метров.

```
SELECT DISTINCT ROOM_TYPE FROM Room_type, Room
WHERE Room_type.ROOM_TYPE_NUM = Room.ROOM_TYPE_NUM and Room.AREA_SIZE > 200;
```

Фраза “WHERE Room_type.ROOM_TYPE_NUM = Room.ROOM_TYPE_NUM” отвечает за присутствие соединения. Фраза “WHERE Room.AREA_SIZE > 200” отвечает за присутствие селекции. Результат работы запроса представлен на рисунке 1.13.

ROOM_TYPE
=====
Office
Shop
Technical under...
Warehouse

Рисунок 1.13 – Результат выполнения запроса с селекцией и соединением

1.3.9 Не был создан запрос, использующий операции проекции и деления, так как не удалось придумать запрос используемый требуемые операции.

1.3.10 Был создан запрос, использующий операции проекции, объединения и конъюнкции: Вывести номер типа помещения – офис и номера типа помещения у

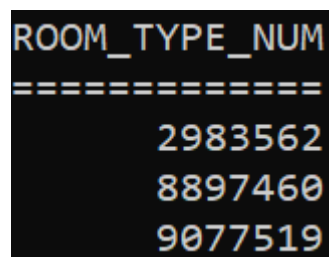
которых номер компании владельца этого помещения = 280040223 и размер этого помещения меньше 200 квадратных метров.

```
SELECT ROOM_TYPE_NUM FROM Room_type where ROOM_TYPE like 'Office'
```

```
UNION
```

```
SELECT ROOM_TYPE_NUM FROM Room where COMPANY_NUM = 280040223 AND AREA_SIZE < 200;
```

Фраза “UNION” отвечает за присутствие объединения. Фраза “AND” отвечает за присутствие конъюнкции. С проекцией очевидно связан выбор одного атрибута кортежа из многих в таблице. Результат работы запроса представлен на рисунке 1.14.



ROOM_TYPE_NUM
2983562
8897460
9077519

Рисунок 1.14 – Результат выполнения запроса с проекцией, объединением и конъюнкцией

1.3.11 Были бы рады создать запрос, использующий операции соединения и деления в одном запросе, да вот только с делением возникли проблемы: не удаётся составить запрос, использующий деление. В целом не совсем понятно как деление работает.

Выводы

В ходе выполнения данной лабораторной работы были изучены основы реляционной алгебры как базового средства манипулирования. Научились представлять запросы как на реляционной алгебре, так и на SQL.

ОТВЕТЫ НА КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Поясните действие операции проекции.

Операция проекции – это операция выбора столбцов из таблицы, представляющей отношение, по какому-либо признаку. А именно машина выбирает те атрибуты (т. е. буквально те столбцы) исходного отношения-операнда, которые были указаны в проекции.

2. Приведите пример операции селекции.

`SELECT * FROM <таблица> WHERE какой_либо_атрибут = чему_нибудь;`

3. Чем отличаются операции РА соединение и объединение.

Разница между соединением и объединением заключается в том, что при соединении будет выполнено горизонтальное соединение колонок разных таблиц. А при объединении будет выполнено вертикальное объединение строк разных таблиц, количество колонок останется без изменений.

4. Продемонстрируйте на примере, как выразить операцию соединения через декартово произведение.

`SELECT таблА.*, таблВ.* FROM таблА, таблВ`