

Министерство образования и науки Украины
Севастопольский национальный технический университет

**ИЗУЧЕНИЕ ОСНОВ ЯЗЫКА МАНИПУЛИРОВАНИЯ
ДАНЫМИ SQL НА БАЗЕ СЕРВЕРА INTERBASE**

Методические указания
к лабораторной работе №2
по дисциплине
“ОРГАНИЗАЦИЯ БАЗ ДАННЫХ”
для студентов специальности 7.080401 –
"Информационные управляющие системы и технологии"
всех форм обучения

Севастополь
2005



Изучение основ языка манипулирования данными SQL на базе сервера Interbase. Методические указания к лабораторной работе №2 по дисциплине “Организация баз данных”, для студентов специальности 7.080401 - "Информационные управляющие системы и технологии" всех форм обучения. /Сост. Ю.В. Доронина, О.Л. Тимофеева, В.Е. Шишкевич. - Севастополь: Изд-во СевНТУ, 2005.-24с.

Цель методических указаний: выработка у учащихся практических навыков по работе с реляционными базами данных.

Методические указания рассмотрены и утверждены на заседании кафедры Информационных Систем. Протокол № 7 от 19 мая 2005 г.

Рецензент: доц. департамента кибернетики и вычислительной техники, к.т.н. Литвинова Л.А.

Допущено учебно-методическим центром в качестве методических указаний.

СОДЕРЖАНИЕ	
Лабораторная работа № 2. SQL. Создание таблицы и элементарные выборки	4
Приложение А. Варианты заданий.	12

ЛАБОРАТОРНАЯ РАБОТА № 2. SQL. СОЗДАНИЕ ТАБЛИЦЫ И ЭЛЕМЕНТАРНЫЕ ВЫБОРКИ

1. Цель лабораторной работы:

Изучить формы оператора CREATE TABLE и простейшие формы оператора SELECT.

2. Основные положения

2.1. Описание базовых операторов для работы с таблицами

2.1.1. CREATE TABLE

CREATE TABLE устанавливает новую таблицу, ее столбцы и ограничения целостности в существующей базе данных. Пользователь, который создает таблицу, становится владельцем таблицы и получает полные привилегии на таблицу. Синтаксис оператора CREATE TABLE приведен в Приложении Б. В общем виде описание оператора можно представить следующим образом:

```
CREATE TABLE <имя таблицы> (<опред_столбца> [,<опред_столбца> |  
<ограничения> ...]);
```

Обязательно надо указать имя таблицы и определить как минимум один столбец.

CREATE TABLE поддерживает следующие возможности для определения столбцов:

Столбцы определяют имя и тип данных для данных, вводимых в столбец. Основанные на доменах столбцы наследуют все характеристики домена, но столбец может переопределить значение по умолчанию, атрибут NOT NULL, дополнительные ограничения CHECK или порядок сортировки.

Вычисляемые столбцы. Значение столбца вычисляется каждый раз при доступе к таблице. Если тип данных не определен, InterBase определяет тип данных результата операции. Столбцы, к которым обращается вычисление, должны быть определены раньше, чем вычисляемый столбец.

Описание типа данных для столбца типа **CHAR**, **VARCHAR** или **BLOB-текст** может включать предложение **CHARACTER SET** определяя специфическую кодировку для одиночного столбца. Иначе столбец использует определенную по умолчанию для базы данных кодировку. Если кодировка базы данных изменена, все столбцы впоследствии определенные имеют новую кодировку, но существующие столбцы не изменяются.

Примеры использования оператора CREATE TABLE

Рассмотрим таблицы, имеющие следующую структуру:

Таблица «Служащий» (EMPLOYEE), представленная на рисунке 1, содержит следующие атрибуты:

Emp_Num	номер служащего, целочисленное, не может принимать значение NULL;
L_Name	фамилия, строка переменной длины с максимальной длиной 20 символов, не может принимать значение NULL.;
F_Name	имя, строка переменной длины с максимальной длиной 10 символов, не может принимать значение NULL.;
Dep_Num	номер отдела, целочисленное, не может принимать значение NULL;
Job_Cod	должность, строка переменной длины с максимальной длиной 10 символов;
Phone Ext	внутренний номер телефона, целочисленное, находится в пределах от 222 до 444 ;
Salary	оклад, значение от 0 до 9999,99.

Emp_Num	L_Name	F_Name	Dep_Num	Job_Cod	Phone Ext	Salary
111	Ivanov	Ivan	10	engineer	226	890,50
112	Petrov	Stepan	10	admin		1456,96
113	Sidorova	Irina	20	engineer	442	920,48

Рисунок 1 – Таблица «Служащий»

Следующая инструкция создает указанную таблицу:

```
CREATE TABLE EMPLOYEE
(EMP_NUM INTEGER NOT NULL,
 L_NAME CHAR(20) NOT NULL,
 F_NAME CHAR(10) NOT NULL,
 DEP_NUM INTEGER NOT NULL,
 JOB_CODE CHAR(10),
 PHONE_EXT INTEGER CHECK (PHONE_EXT BETWEEN 222 AND 444),
 SALARY DECIMAL(6,2) DEFAULT 0
);
```

Таблица «Отдел» (DEPARTMENT), представленная на рисунке 2, содержит следующие атрибуты:

Dep_Num	номер отдела, целочисленное, не может принимать значение NULL;
Dep_Name	наименование отдела, строка переменной длины с максимальной длиной 20 символов;
Dep_Head	номер служащего, руководителя отдела целочисленное;
Budget	бюджет, значение от 0 до 999999,99.
Location	адрес отдела, строка переменной длины с максимальной длиной 20 символов;
Phone_Num	номер телефона, целочисленное.

Dep_Num	Dep_Name	Dep_Head	Budget	Location	Phone_Num
10	Inspector		20000,00	fl 5, of 16-34	454601
20	construct	164	500000,0	fl 6-7	456701
30	security	46	100000,0	fl 10, of 1-22	442101

Рисунок 2 – Таблица «Отдел»

Следующая инструкция создает указанную таблицу:

```
CREATE TABLE DEPARTMENT
(DEP_NUM INTEGER NOT NULL,
 DEP_NAME CHAR(20),
 DEP_HEAD INTEGER,
 BUDGET DECIMAL(8,2) DEFAULT 0
 LOGATION CHAR(20),
 PHONE_NUM INTEGER
);
```

Таблица «Проект» (PROJECT), представленная на рисунке 3, содержит следующие атрибуты:

Pr_ID код проекта, целочисленное, не может принимать значение NULL;

Pr_Name наименование проекта, строка переменной длины с максимальной длиной 80 символов;

Pr_Descr описание проекта,

Leader номер служащего – руководителя проекта, целочисленное, не может принимать значение NULL;

Budget бюджет, значение от 0 до 999999,99.

Pr_Id	Pr_Name	Pr_Descr	Leader	Budget
1112	Data based ...		113	20000,00
1114	Parallel algorithms ...		112	50000,0
2222	Knowledge ...		224	100000,0

Рисунок 3 – Таблица «Проект»

Следующая инструкция создает указанную таблицу:

```
CREATE TABLE PROJECT
(PR_ID INTEGER NOT NULL,
 PR_NAME CHAR(80),
 PR_DESCR CHAR(1000)
 LEADER INTEGER,
 BUDGET DECIMAL(8,2) DEFAULT 0
);
```

Таблица «Проект-Служащий» (PROJECT- EMPLOYEE), представленная на рисунке 4, содержит следующие атрибуты:

Emp_Num	номер служащего, целочисленное, не может принимать значение NULL;
Pr_ID	код проекта, целочисленное, не может принимать значение NULL;
Data_S	содержит дату начала работ
Data_F	содержит дату сдачи проекта

Pr_Id	Emp_Num	Data_S	Data_F
1112	111	25.08.2004	2.04.2005
1112	112	12.06.2004	22.05.2005
2222	111	22.10.2004	22.01.2005
2222	113	2.08.2004	22.08.2005

Рисунок 4 – Таблица «Проект-Служащий»

Следующая инструкция создает указанную таблицу:

```
CREATE TABLE PROJECT_EMPLOYEE
( PR_ID INTEGER NOT NULL,
  EMP_NUM INTEGER NOT NULL,
  DATA_S DATA
  DATA_F DATA
);
```

2.1.2. INSERT

INSERT добавляет одну или более новых строк данных к существующей таблице. Синтаксис оператора INSERT приведен в Приложении Б.. Значения вставляются в столбцы строки по порядку их следования, если не задан факультативный список столбцов. Если список столбцов задан на множестве всех доступных столбцов, то во все не перечисленные столбцы автоматически вставляется или значение по умолчанию, или NULL.

Если факультативный список столбцов упущен, предложение VALUES должно содержать значения для всех столбцов таблицы.

Чтобы вставить одну строку данных, должно присутствовать предложение VALUES и содержать определенный список значений.

Чтобы вставить несколько, строк данных, определите **<select_expr>**, которое возвращает уже существующие данные из другой таблицы. Выбранные строки должны соответствовать списку столбцов.

Предупреждение: Допустимо выбирать из той же таблицы, в которую строки вставляются, но эта практика не рекомендуется, так как подобные действия могут привести к бесконечным вставкам строк (зацикливанию).

Примеры использования оператора INSERT

1. Следующая инструкция добавляет строку в таблицу «Отдел»

```
INSERT INTO DEPARTMENT VALUES
(20, 'construct', 164, , 'fl 6-7', 456701);
```

2. Следующая инструкция добавляет строку в таблицу «Служащий», присваивает значения шести столбцам:

```
INSERT INTO EMPLOYEE
(EMP_NUM, L_NAME, N_NAME, DEP_NUM, SALARY, JOB_CODE)
VALUES ( 112, 'Petrov', 'Stepan', 10, 1456.96, 'admin');
```

3. Следующая инструкция определяет значения, чтобы вставить в таблицу, используя инструкцию SELECT:

```
INSERT INTO PROJECTS
SELECT * FROM NEW_PROJECTS
WHERE NEW_PROJECTS.START_DATE > '6-JUN-1994';
```

2.1.3. SELECT

SELECT возвращает данные из таблицы, представления или сохраненной процедуры. Синтаксис оператора SELECT приведен в приложении Б. Различные инструкции SELECT выполняют следующие действия:

- Возвращают одиночную строку или часть строки из таблицы. Это операция упоминается, как **sigleton выбор**.
- Непосредственно возвращают список строк или список частичных строк из таблицы.
- Возвращают связанные строки, или частичные строки из **join** двух или более таблиц.
- Возвращают все строки, или частичные строки из **union** двух или более таблиц.

Любая инструкция SELECT содержит два обязательных предложения (SELECT, FROM) и возможно другие предложения (WHERE, GROUP BY, HAVING, UNION, PLAN, ORDER BY). Предложения SELECT и FROM обязательны и для sigleton, и для multi-row SELECT; все другие предложения перечисленные ниже факультативны. Назначение каждого предложения оператора SELECT приведено в таблице 1.

Таблица 1 - Назначение предложений входящий в оператор SELECT

Предложе- ние	Назначение
SELECT	Список столбцов, которые возвращаются.
FROM	Определяет таблицы, в которых ищутся значения.
WHERE	Определенное условие поиска, которое используется, чтобы выбрать необходимые строки из множества всех строк. Предложение WHERE может содержать инструкцию SELECT, которая упоминается, как подзапрос
GROUP BY	Группирует возвращенные строки, основываясь на общих значениях столбцов. Используется совместно с HAVING.
HAVING	Используется совместно с GRUP BY. Определяет условия, которые ограничивают группировку возвращаемых строк.
UNION	Комбинирует результаты двух или более инструкций SELECT создавая одиночную динамическую таблицу, исключая повторяющиеся строки.
ORDER BY	Определяет порядок сортировки строк возвращенных SELECT, по умолчанию в возрастающем порядке (ASC), или в убывающем порядке (DESC).
PLAN	Определяет план запроса, который будет использоваться оптимизатором запроса вместо обычного выбора.

Примеры использования оператора SELECT

1. Следующая инструкция выбирает столбцы из таблицы «Отдел»:

```
SELECT DEP_NAME, BUDGET, PHONE_NUM FROM DEPARTMENT;
```

2. Следующая инструкция использует шаблон *, что бы выбрать все столбцы и строки из таблицы «Проект»:

```
SELECT * FROM PROJECT ;
```

3. Следующая инструкция использует агрегатную функцию, что бы сосчитать все строки в таблице, которые удовлетворяют условиям поиска определенным в предложении WHERE, количество проектов с бюджетом выше 50000:

```
SELECT COUNT (*) FROM PROJECT
WHERE BUDGET > 50000;
```

4. Следующая инструкция использует составное условие поиска, определенное в предложении WHERE, позволяет определить фамилии и должности сотрудников 10 отдела, имеющих оклад выше 600:

```
SELECT L_NAME, STATE FROM EMPLOYEE
WHERE DEP_NUM=10 AND SALARY>600;
```

5. Следующая инструкция выбирает два столбца и сортирует возвращенные строки по второму столбцу:

```
SELECT L_NAME, STATE FROM EMPLOYEE
ORDER BY STATE;
```

6. Следующая инструкция позволяет оценить в каких проектах служащий принимает участие и вывести информацию с учетом даты окончания служащим работ по проекту

```
SELECT PR_NAME, EMP_NUM, DATA_F FROM PROJECT_EMPLOYEE
WHERE EMP_NUM=112
ORDER BY DATA_F ;
```

7. Следующая инструкция использует в условии поиска конструкцию IN (входит), позволяет определить наименования проектов над которыми работают служащие номера, которых перечислены в списке:

```
SELECT PR_NAME FROM PROJECT_EMPLOYEE
WHERE EMP_NUM IN (111, 113);
```

3. Ход лабораторной работы

1. Определить номер своего варианта. Номер варианта(х) определяется как $x = N \bmod 20$, где N – номер студента в группе.
2. По номеру своего варианта определить имя таблицы, которую надо будет создать при выполнении работы (Приложение А).
3. Ознакомиться с описанием базовых операторов для работы с таблицами.
4. Создать таблицу. Особое внимание надо уделить описанию первичного ключа, значений по умолчанию, описателям NOT NULL и конструкции CHECK.
5. Занести в таблицу образцы данных оператором INSERT INTO. Необходимо занести не менее 10 строк. Внимание! После того, как в таблицу занесены образцы данных, менять структуру таблицы можно только оператором ALTER TABLE.
6. Создать запрос, выводящий все строки таблицы.

7. Создать запрос, задающий порядок столбцов, отличный от исходного.
8. Продemonстрировать действие модификатора DISTINCT.
9. Ограничить вывод запроса, используя WHERE с простым условием.
10. Ограничить вывод запроса, используя WHERE и составное условие.
11. Продemonстрировать действие специальных функций IN, BETWEEN, LIKE, и IS NULL в условии.
12. Продemonстрировать работу специальных функций с условием NOT.

4. Содержание отчета

1. Отчет состоит из титульного листа, цели работы, описания процесса выполнения работы и вывода.
2. Отчет должен содержать:
 - таблицу исходных данных,
 - тексты запросов,
 - результаты их выполнения.

5. Контрольные вопросы

1. Основные понятия реляционных БД: отношение, атрибут, кортеж, домен, схема отношения?
2. Правила задания таблиц?
3. Ключевой атрибут, первичный ключ, дополнительный ключ?
4. Требования к вводу данных оператором INSERT INTO?
5. Унарные операторы над отношениями, операторная форма записи?
6. Оператор SELECT для реализации проекции – примеры?
7. Оператор SELECT для реализации селекции (выбора) – примеры?

ПРИЛОЖЕНИЕ А. Варианты заданий (обязательное)

Вариант 1

На рисунке А.1 изображена структура системы, которая содержит информацию о публикациях по ряду выбранных тем.

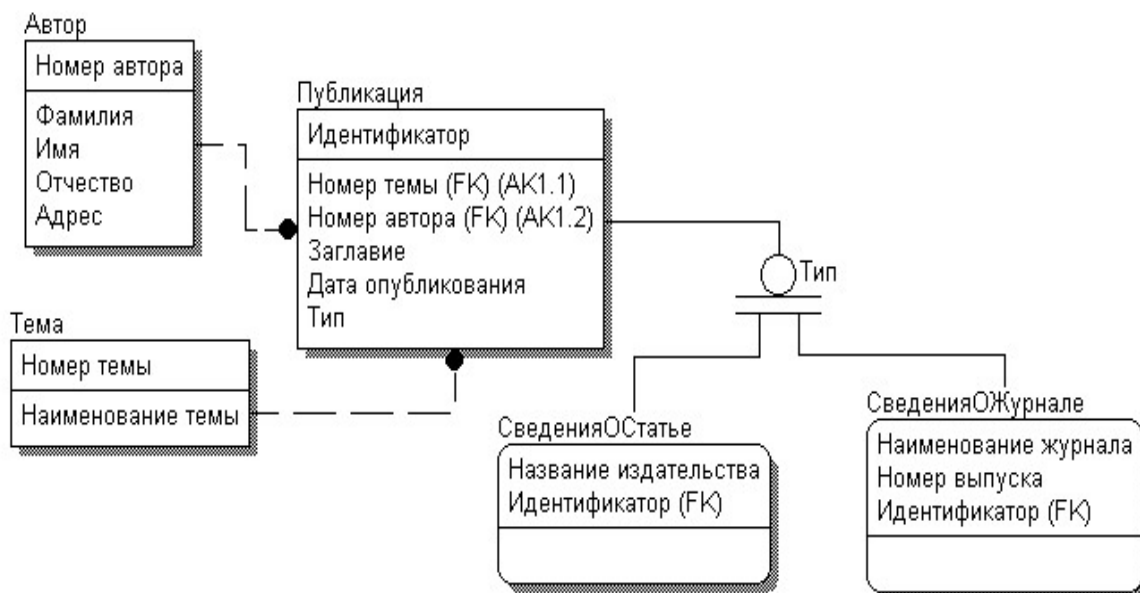


Рисунок А.1.

Вариант 2

На рисунке А.2 изображена структура системы, которая содержит информацию о внутренней системе обучения в большой промышленной компании.

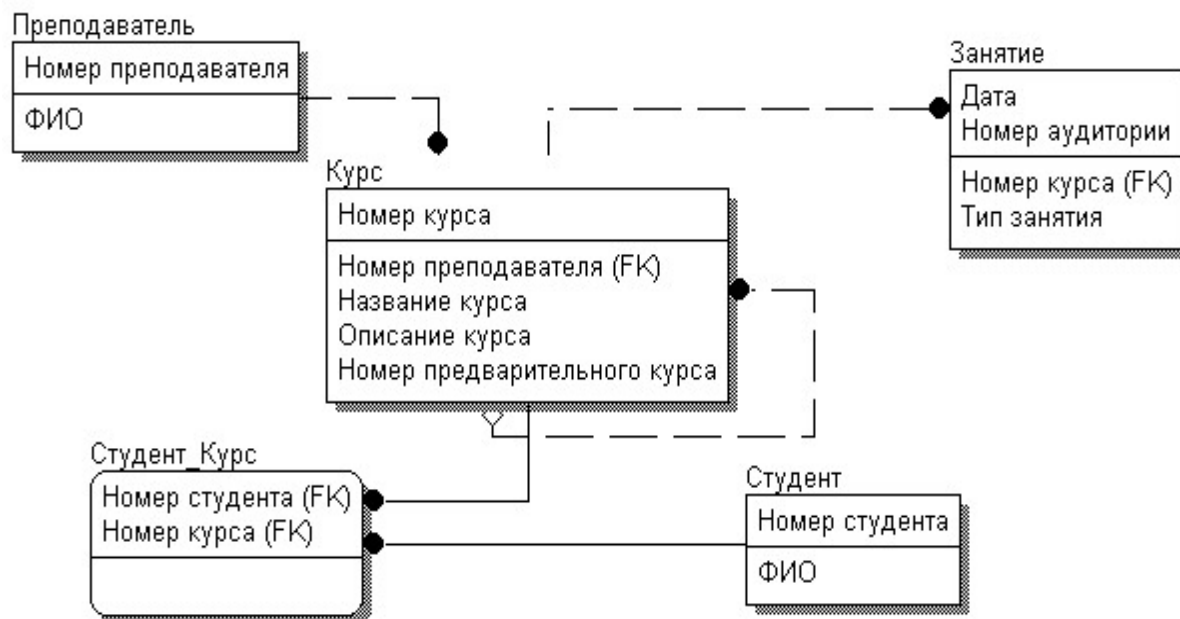


Рисунок А.2.

Вариант 3

На рисунке А.3 изображена структура системы, которая содержит информацию о пациентах клиники.

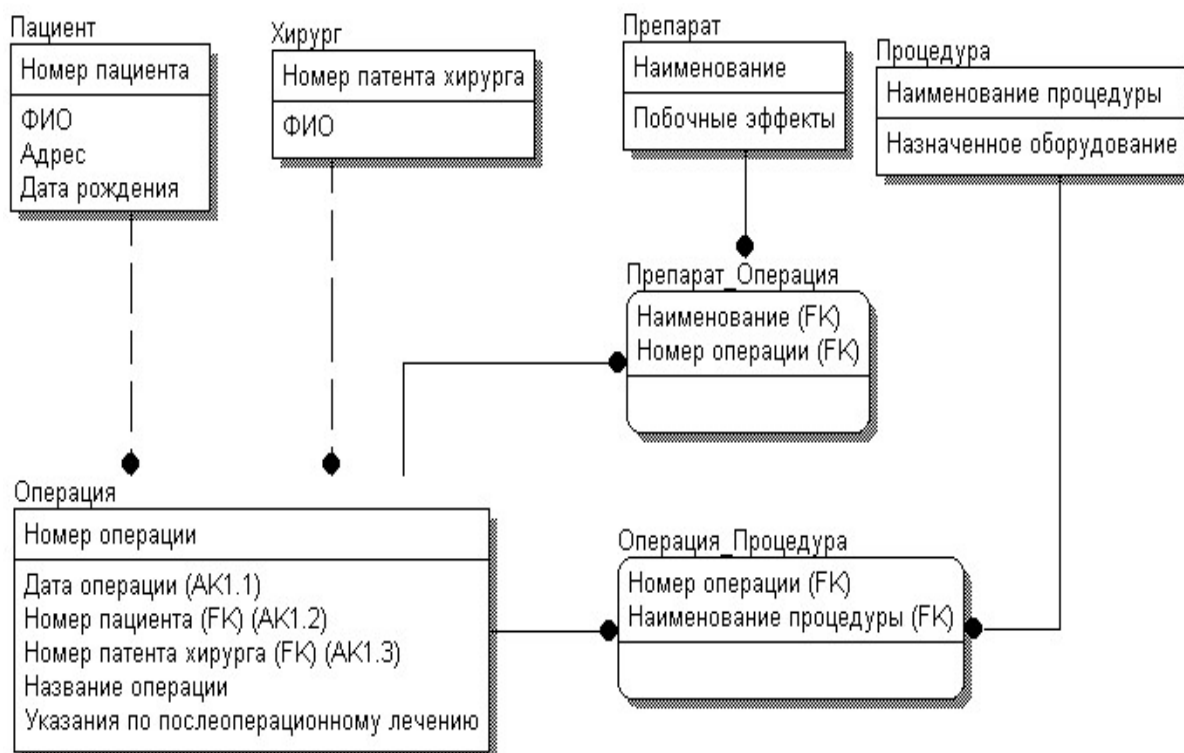


Рисунок А.3.

Вариант 4

На рисунке А.4 изображена структура системы, которая содержит анкетные данные служащих конторы.

Типы отношений: состоят в браке, является родителем

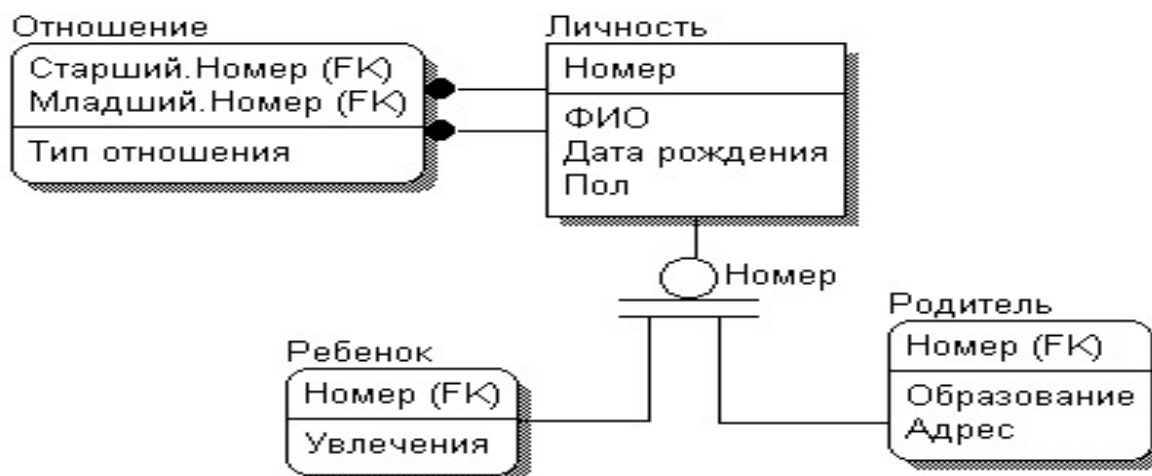


Рисунок А.4.

Вариант 5

На рисунке А.5 изображена структура системы, которая содержит информацию о торговых фирмах.

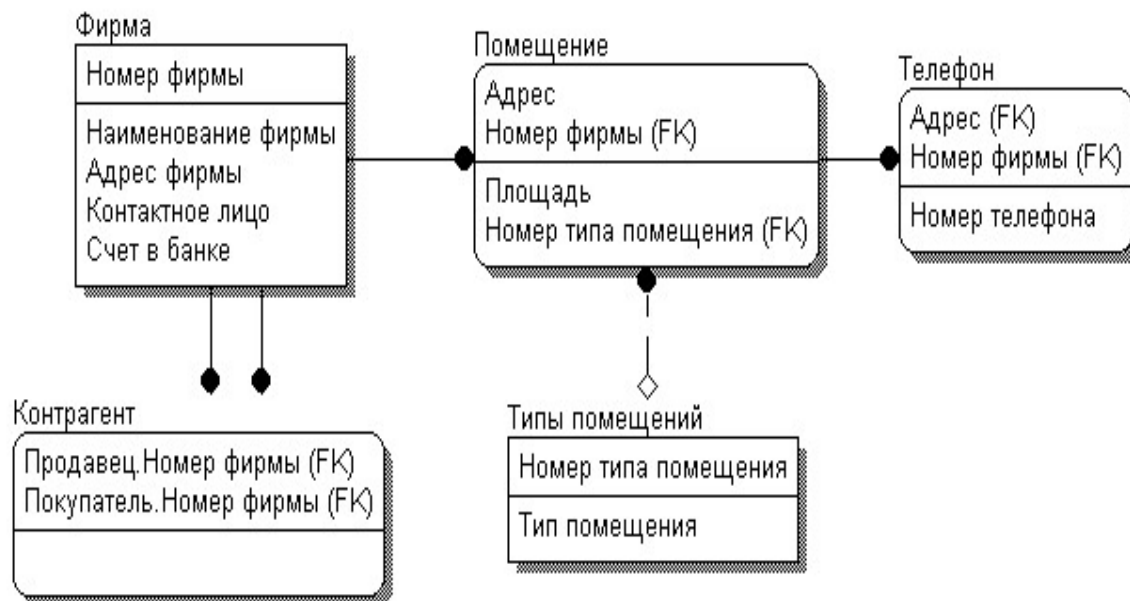


Рисунок А.5.

Вариант 6

На рисунке А.6 изображена структура системы, которая содержит каталог работ.

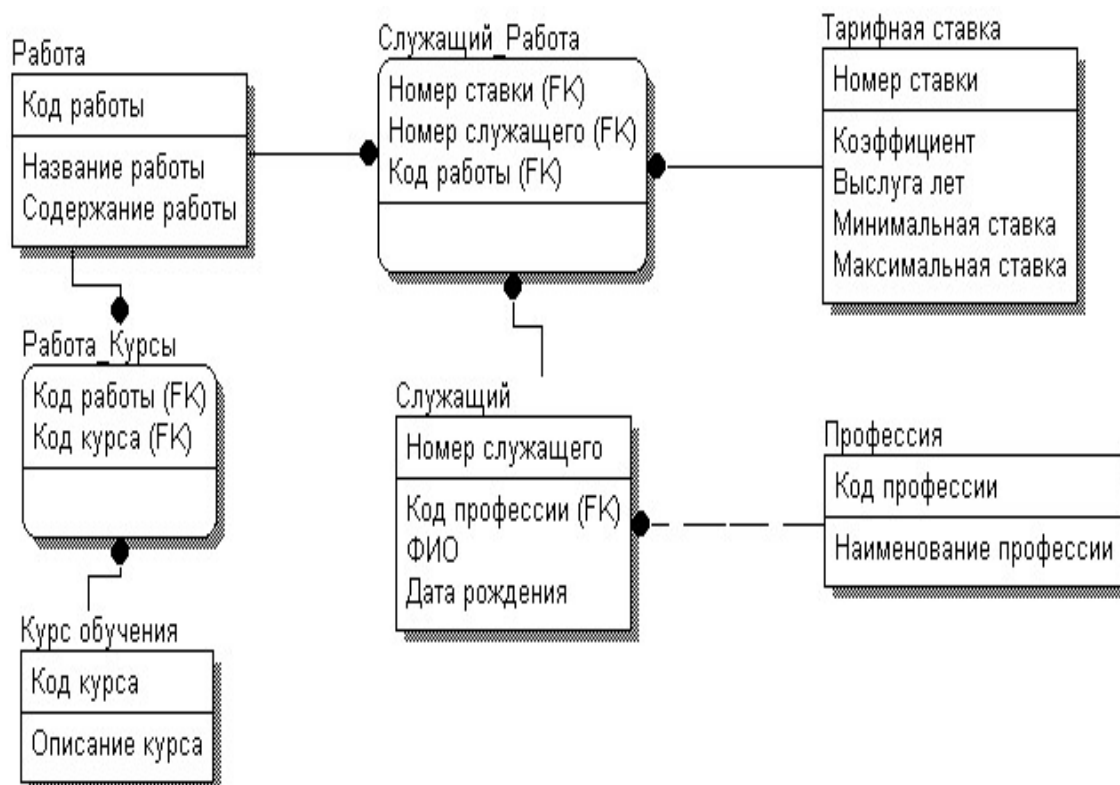
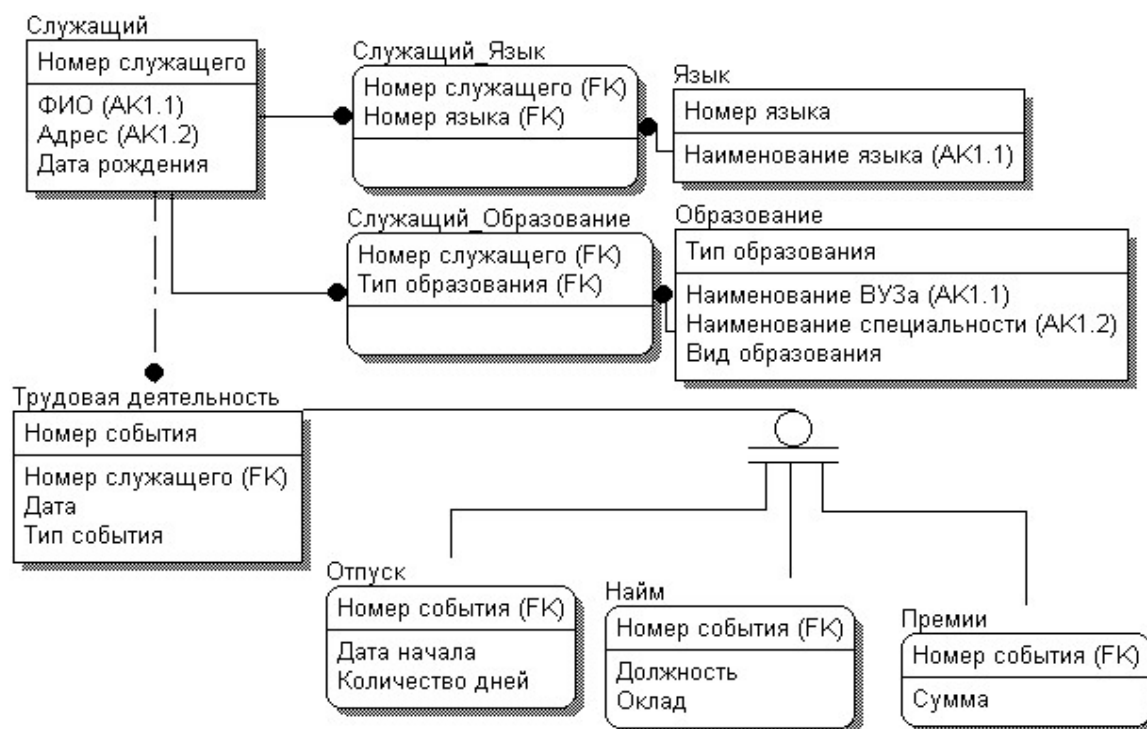


Рисунок А.6.

Вариант 7

На рисунке А.7 изображена структура системы, которая содержит информацию о служащих.



Вид образования - высшее, среднее, неполное высшее. Тип события – отпуск, премирование, события связанные с наймом (в том числе изменение в должности и окладе).

Рисунок А.7.

Вариант 8

На рисунке А.8 изображена структура системы, которая содержит информацию о факультетах вида:

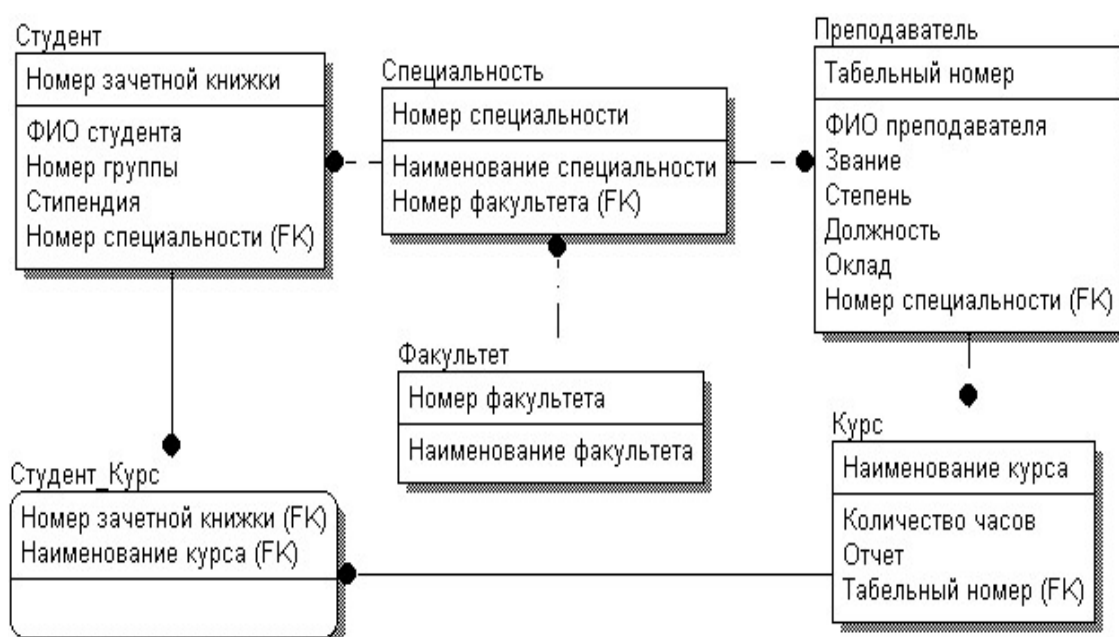


Рисунок А.8.

Вариант 9

На рисунке А.9 изображена структура системы, которая содержит информацию об учебных заведениях, абитуриентах и экзаменах

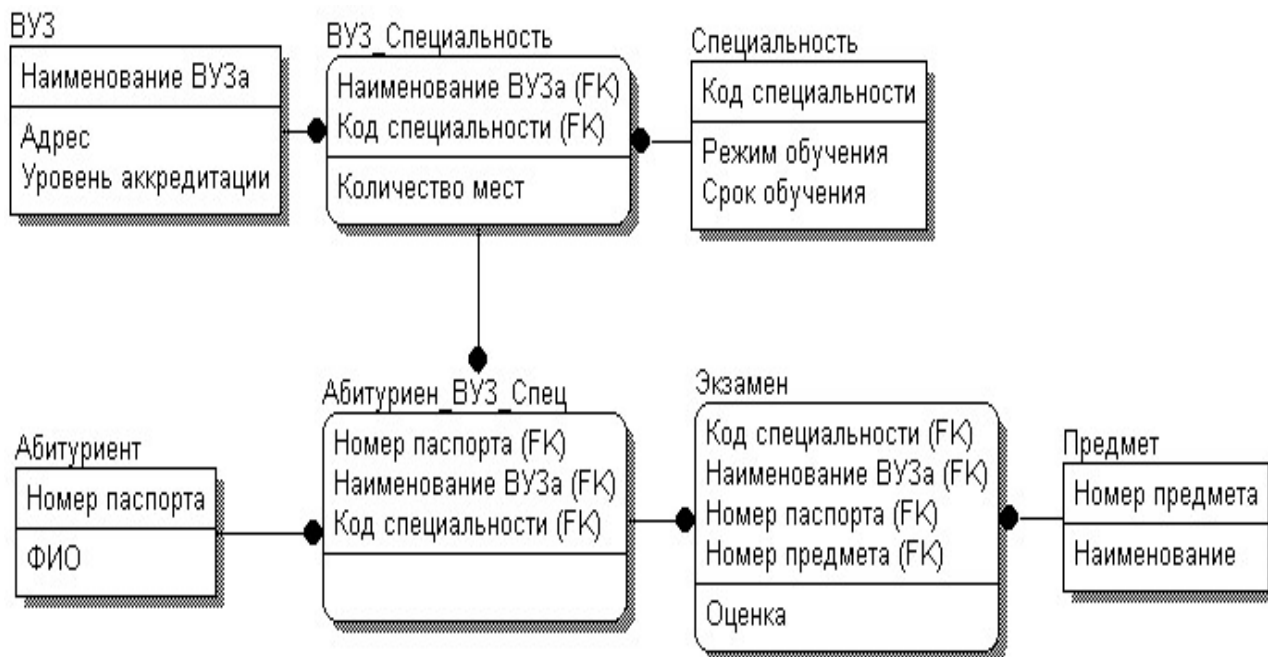
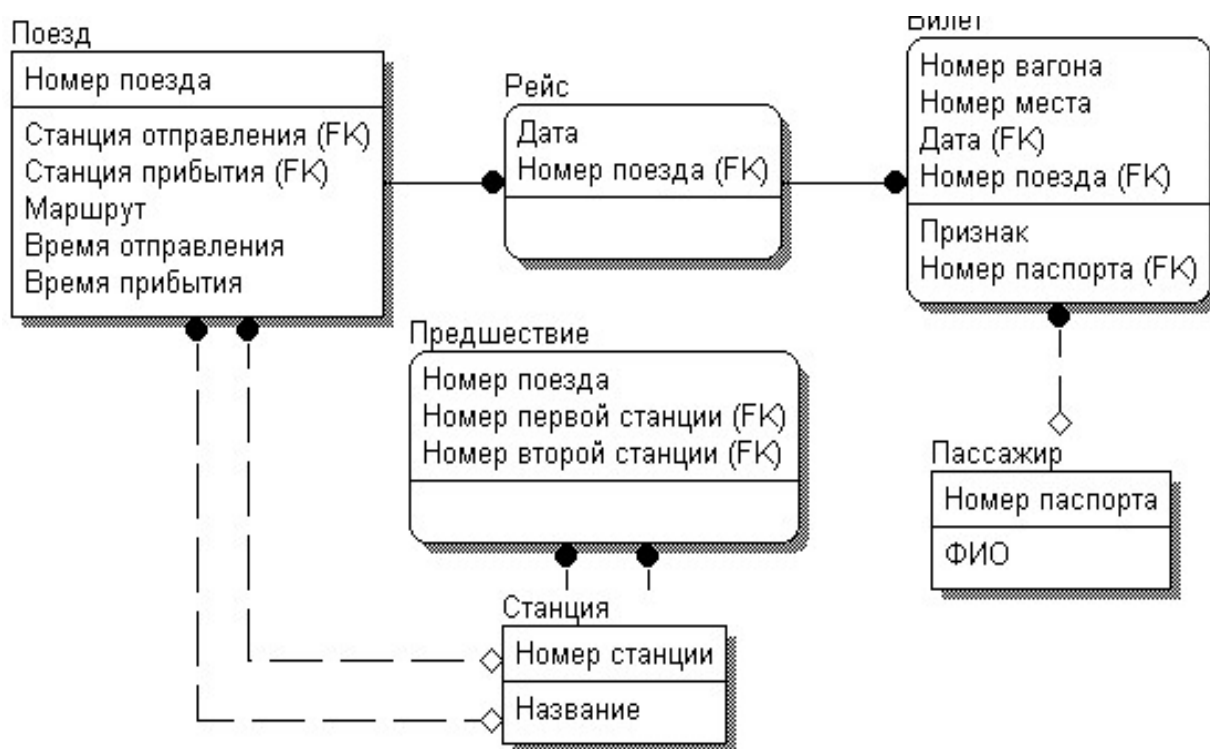


Рисунок А.9.

Вариант 10

На рисунке А.10 изображена структура системы, которая содержит информа-



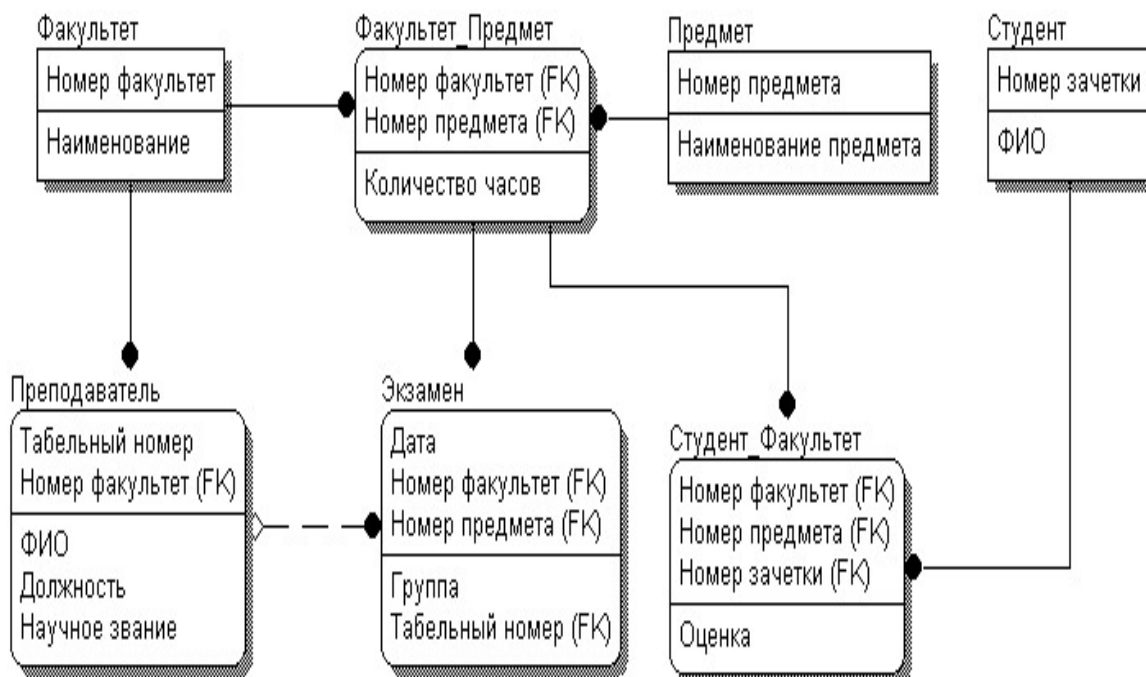
цию о движении поездов.

Признак билета – забронирован или выкуплен.

Рисунок А.10.

Вариант 11

На рисунке А.11 задана структура системы, которая содержит информацию о



результатах экзаменов по каждой кафедре:

Рисунок А.11.

Вариант 12

На рисунке А.12. изображена структура системы, которая содержит картотеку осужденных лиц:

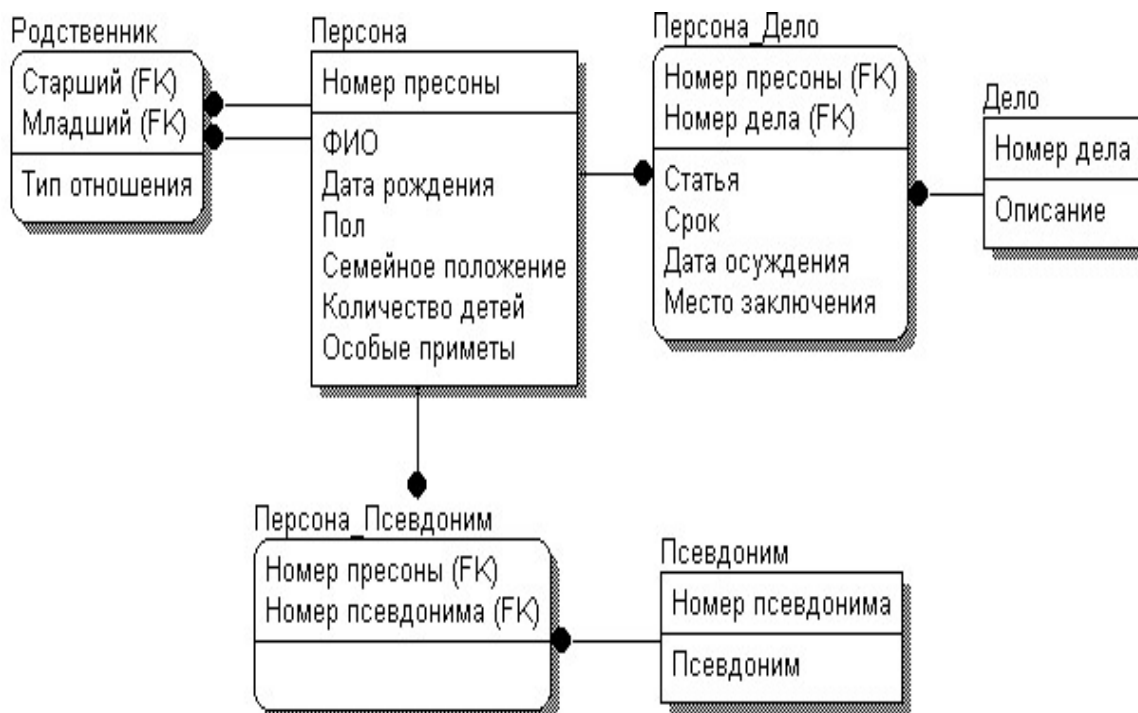


Рисунок А.12.

Вариант 13

На рисунке А.13 изображена структура системы, которая содержит информацию о футбольных командах:

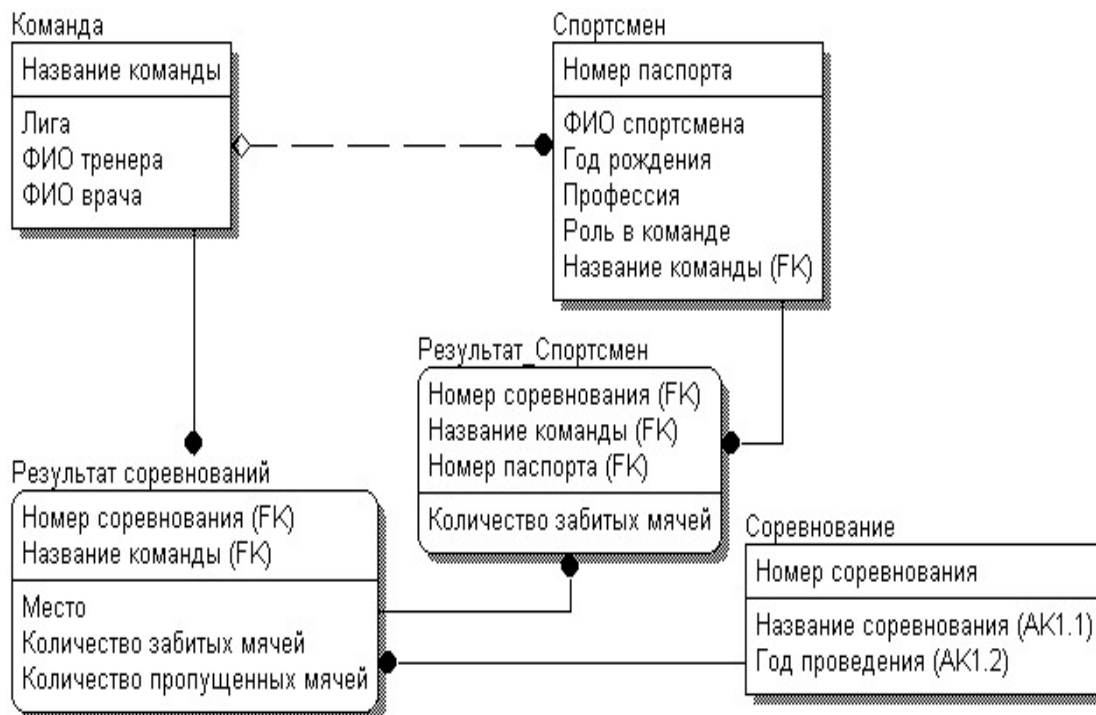
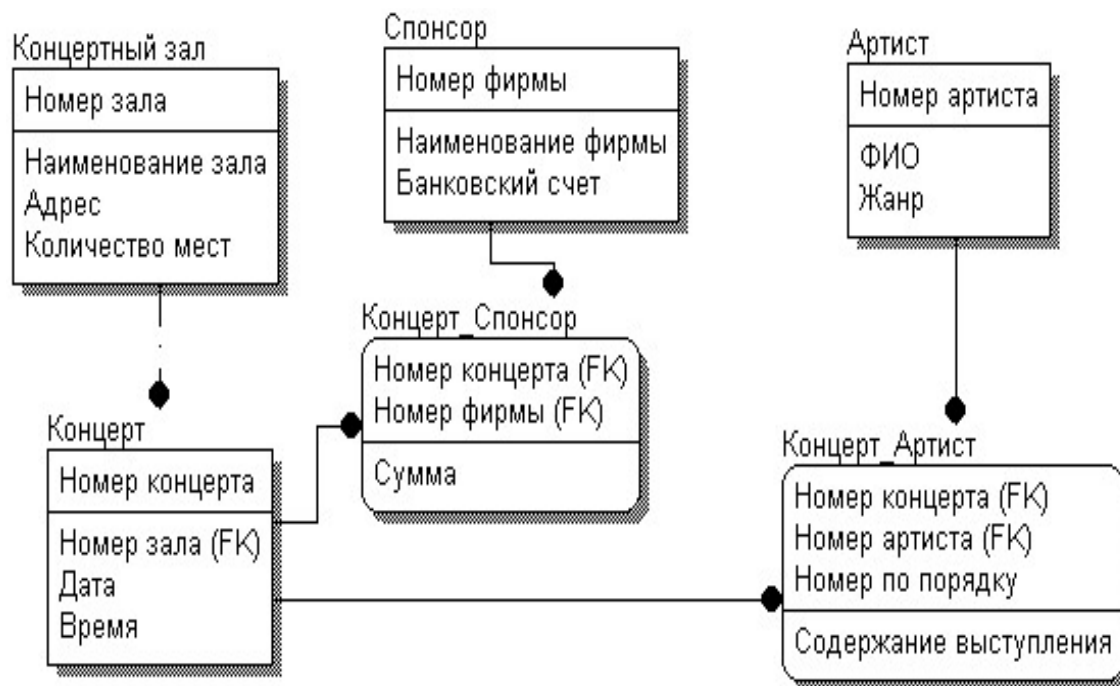


Рисунок А.13.

Вариант 14

На рисунке А. 14 изображена структура системы, которая содержит информа-



цию о концертах:

Предусмотреть жанр «конферансье» - ведущий концерта.

Рисунок А.14.

Вариант 15

На рисунке А.15 изображена структура системы, которая содержит картотеку фирм:

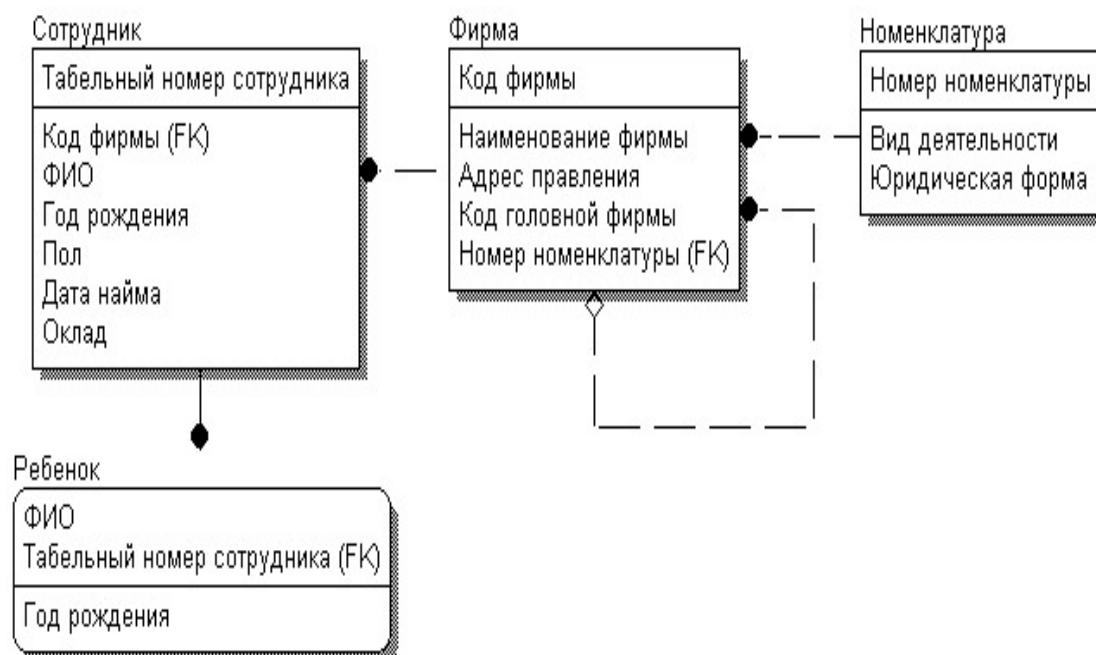
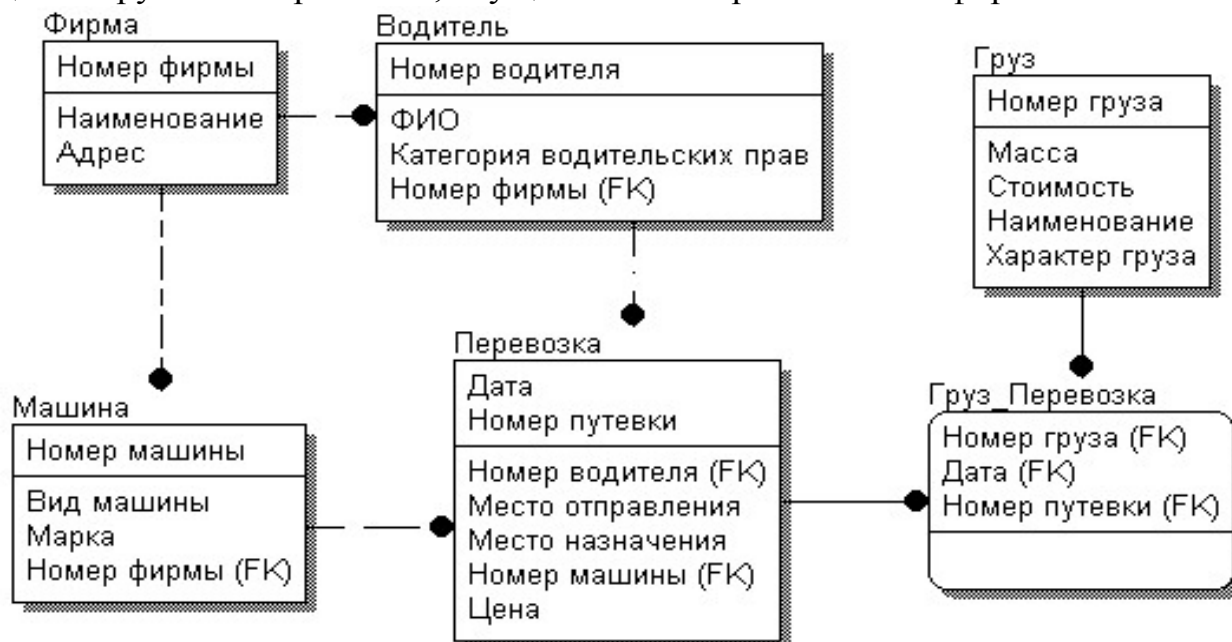


Рисунок А.15.

Вариант 16

На рисунке А.16 изображена структура системы, которая содержит информацию о грузовых перевозках, осуществляемых различными фирмами.



Категория водительских прав –А, В, С

Характер груза - твердый, жидкий и т.д.

Рисунок А.16.

Вариант 17

На рисунке А. 17 изображена структура системы, которая содержит информацию для бронирования мест в гостинице для командированных:

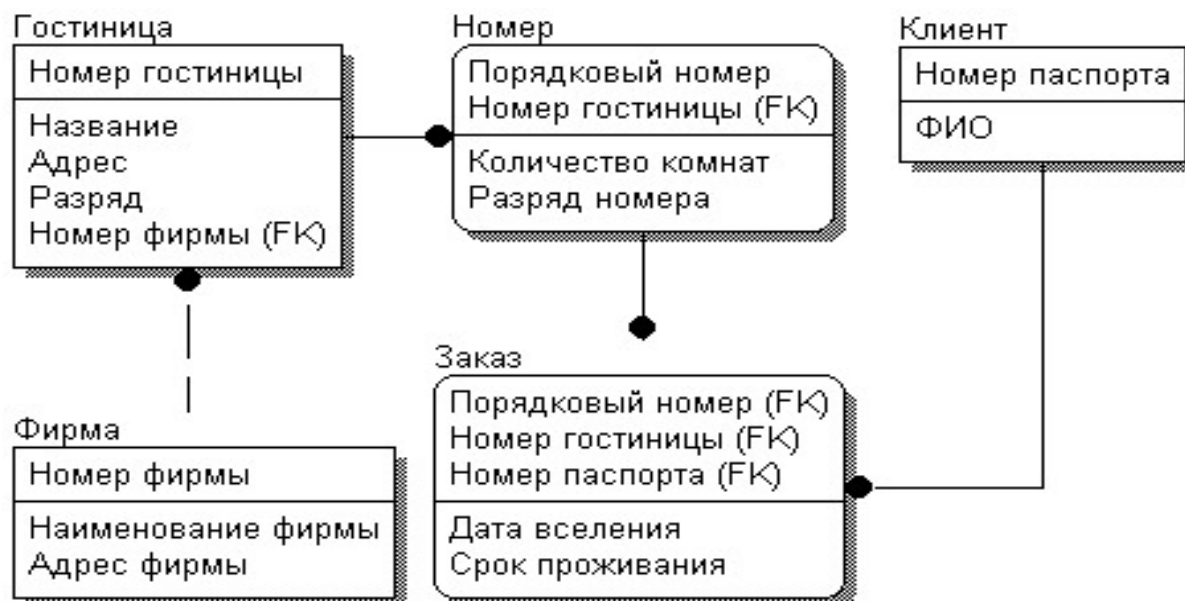


Рисунок А.17.

Вариант 18

На рисунке А.18 изображена структура системы, которая содержит информацию о лечении различных заболеваний:

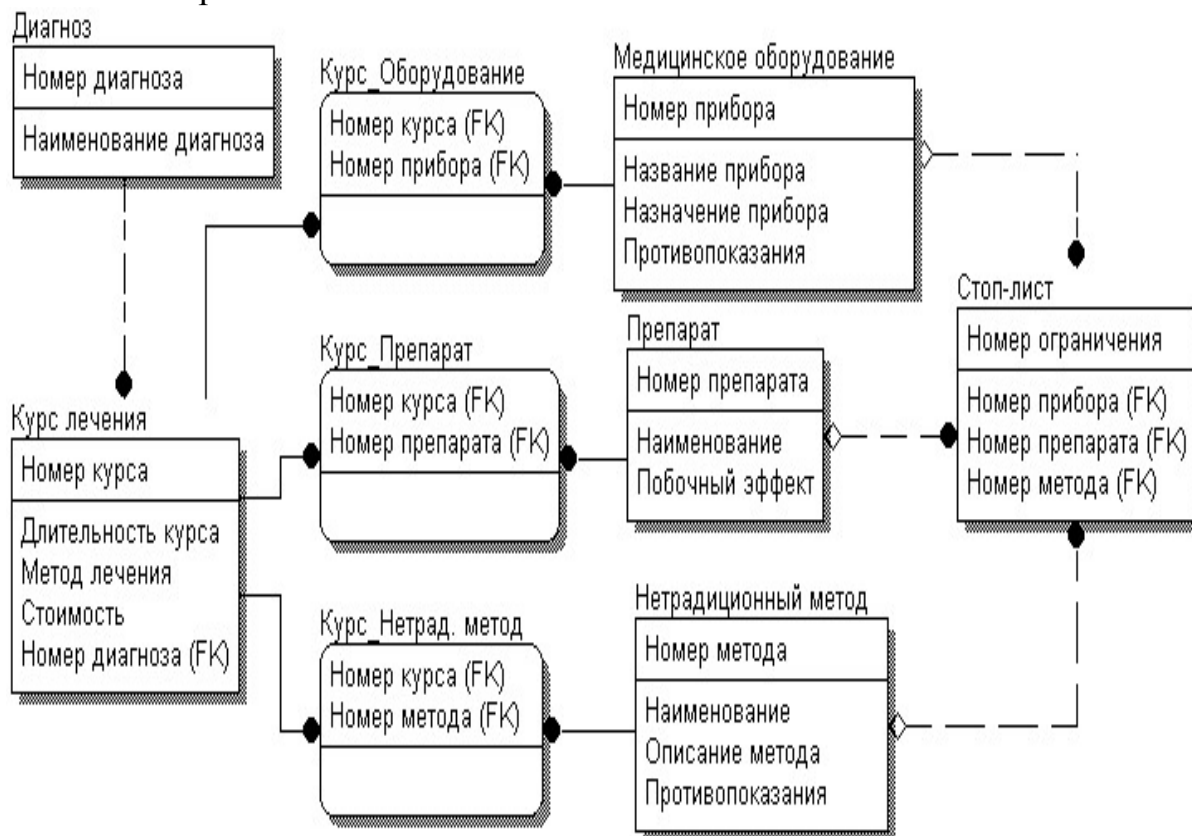


Рисунок А.18.

Вариант 19

На рисунке А. 19 изображена структура системы, которая содержит информацию о вычислительных центрах города:

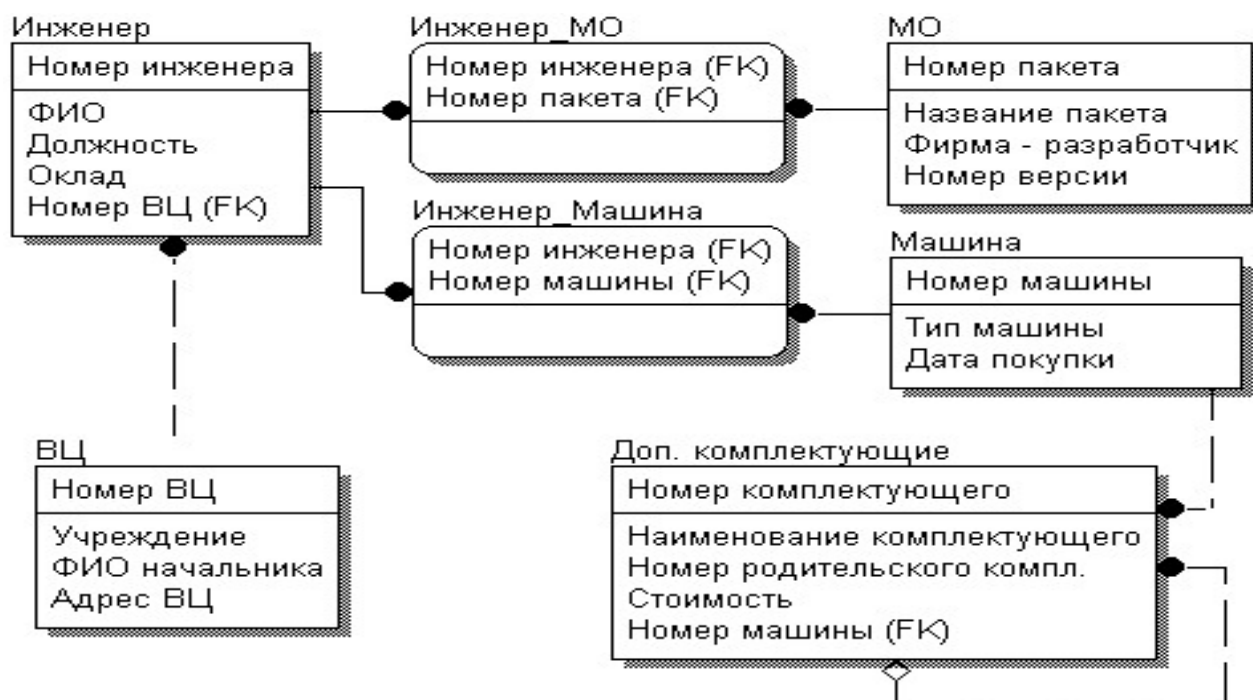


Рисунок А.19.

Вариант 20

На рисунке А.20 изображена структура системы, которая содержит информацию о политических партиях:

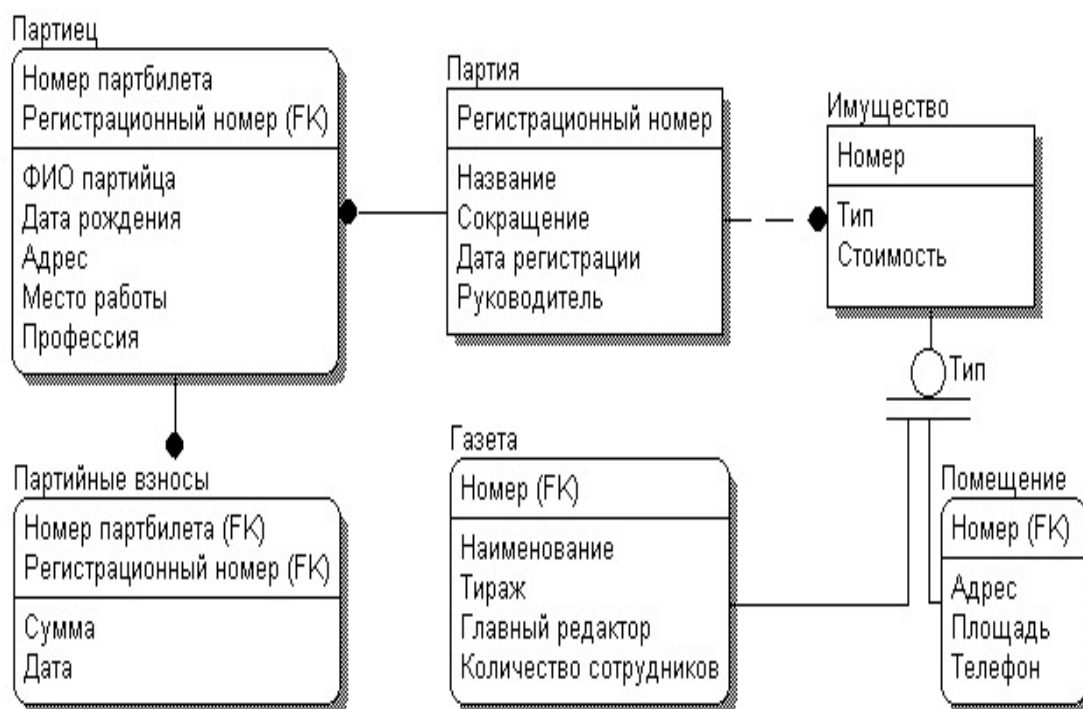


Рисунок А.20.

