ГУАП

КАФЕДРА № 53

ОТЧЕТ   
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| доц., канд. техн. наук |  |  |  | А. В. Бржезовский |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

|  |
| --- |
| ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ |
| СОЗДАНИЕ ТАБЛИЦ БАЗЫ ДАННЫХ |
| по курсу: МЕТОДЫ И СРЕДСТВА ПРОЕКТИРОВАНИЯ ИНФОРМАЦИОННЫХ СИСТЕМ И ТЕХНОЛОГИЙ |
|  |
|  |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. № | 3641 |  |  |  | Фомин.М.Б |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург 2019

**Описание задачи:**

По аналогии с примерами, приведенными в п. 1.1 – 1.3, произвести создание БД и таблиц, определив правила проверки значений и задав ограничения ссылочной целостности. Структура БД должна обеспечивать хранение сведений, необходимых для выполнения запросов, указанных в варианте задания.

**Вариант задания (вариант 16):**

Создайте базу данных для хранения следующих сведений: город, район, квартира, комната, площадь, цена, агент, продажа. Составьте запросы, позволяющие выбрать:

а) перечень однокомнатных квартир, продаваемых в Московском районе;

б) квартиры, находящиеся на одной улице, но в различных районах; в) двух- и трехкомнатные квартиры, имеющие одинаковую площадь;

г) средняя цена однокомнатной квартиры в городе;

д) районы, в которых продается наибольшее число объектов недвижимости;

е) районы, в которых минимальна стоимость квадратного метра;

ж) улицы, продолжительность которых ограничивается только одним районом; 101

з) районы, в которых не продаются однокомнатные квартиры;

и) районы, в которых продаются квартиры всех строительных серий.

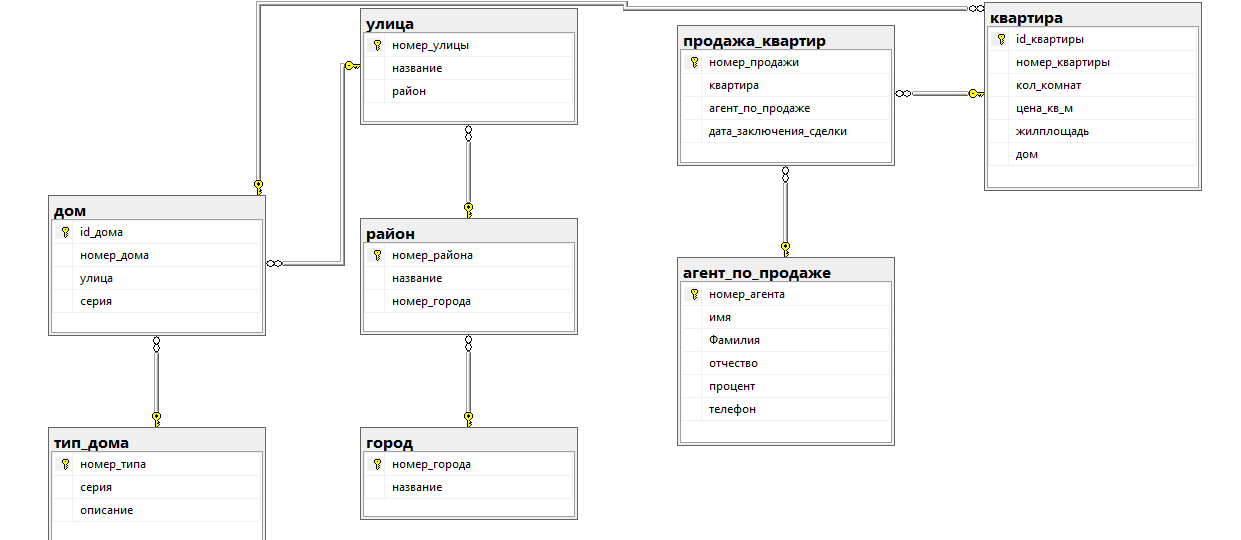


Рис.1 – Схема БД.

**Описание структуры таблиц, ограничений на значения данных, ссылочной целостности, реализованных в БД:**

Данная база данных была создана по структуре агентства по продаже недвижимости. Изучив основные составляющие, было создано 8 таблиц: город,район,улица,дом,квартира,тип\_дома,агент по продаже,продажа квартиры,квартира.

В рассматриваемой базе данных были использованы следующие ограничения:

* [NOT NULL](https://andreyex.ru/bazy-dannyx/uchebnoe-posobie-po-sql/sql-konstanta-not-null/) – гарантирует , что столбец не может иметь нулевое значение.
* UNIQUE – гарантирует , что все значения в столбце различны.
* Primary Key – идентифицирует уникальность для каждой строки/записи в таблице базы данных.
* CHECK – проверочное ограничение гарантирует, что все значения в столбце удовлетворяют определенным условиям.

Также было использовано ограничение для ссылочной целостности.

Примеры:

Скрипт SQL для таблицы:

Use bd\_estate\_agency;

GO

CREATE TABLE квартира

(

номер\_квартиры int ,

кол\_комнат int not null check (кол\_комнат in ('1', '2', '3', '4', '5')),

цена\_кв\_м money not null,

жилплощадь real not null ,

район int ,

дом int not null,

стр\_серия int,

агент int,

primary key(номер\_квартиры,дом),

foreign key (район) references район(номер\_района),

foreign key (стр\_серия) references стр\_серия(номер\_серии),

foreign key (агент) references агент(номер\_агента),

foreign key (дом) references дом(номер\_дома)

on delete no action

on update cascade)

);

Ограничиваем поле «кол\_комнат» на ввод неудовлетворяющих условию параметров. Ограничение ссылочной целостности запрещает удаление записи в родительской таблице, если на соответствующую запись есть ссылки из дочерней таблицы, и требует автоматической модификации значений в определенном столбце дочерней таблицы, при изменении соответствующих значений в родительской таблице.

Скрипт SQL для таблицы [Город]:

create table Город(

Код\_Города int identity primary key,

Название varchar(30) not null unique)

go

Название города уникально, оно не может повторяться.

**Скрипт SQL для создания таблиц**:

Use bd\_estate\_agency;

GO

CREATE TABLE агент\_по\_продаже

(

номер\_агента int IDENTITY(1,1) primary key,

имя varchar(30) not null,

Фамилия varchar(30) not null,

отчество varchar(30) ,

процент real not null,

);

Use bd\_estate\_agency;

GO

CREATE TABLE город

(

номер\_города int IDENTITY(1,1) primary key,

название varchar (30) not null unique

);

Use bd\_estate\_agency;

GO

CREATE TABLE дом

(

id\_дома int IDENTITY(1,1) primary key,

номер\_дома varchar(10),

улица int ,

серия int not null,

foreign key (улица) references улица(номер\_улицы),

foreign key (серия) references тип\_дома(серия),

);

Use bd\_estate\_agency;

GO

CREATE TABLE продажа\_квартир

(

номер\_продажи int IDENTITY(1,1) primary key,

квартира int not null,

агент\_по\_продаже int not null,

дата\_заключения\_сделки date not null,

foreign key (квартира) references квартира(номер\_квартиры),

foreign key (агент\_по\_продаже) references агент\_по\_продаже(номер\_агента),

);

Use bd\_estate\_agency;

GO

CREATE TABLE район

(

номер\_улицы int IDENTITY(1,1) primary key,

название varchar (30) not null,

номер\_города int,

foreign key (номер\_города) references город(номер\_города),

);

Use bd\_estate\_agency;

GO

CREATE TABLE тип\_дома

(

номер\_типа int IDENTITY(1,1) primary key,

серия varchar (30) not null,

описание varchar(200),

);

Use bd\_estate\_agency;

GO

CREATE TABLE улица

(

номер\_улицы int IDENTITY(1,1) primary key,

название varchar (30) not null,

район int not null,

foreign key (район) references район(номер\_района)

);

Use bd\_estate\_agency;

GO

CREATE TABLE квартира

(

номер\_квартиры int ,

кол\_комнат int not null check (кол\_комнат in ('1', '2', '3', '4', '5')),

цена\_кв\_м money not null,

жилплощадь real not null ,

район int ,

дом int not null,

стр\_серия int,

агент int,

primary key(номер\_квартиры,дом),

foreign key (район) references район(номер\_района),

foreign key (стр\_серия) references стр\_серия(номер\_серии),

foreign key (агент) references агент(номер\_агента),

foreign key (дом) references дом(номер\_дома)

);