МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное автономное образовательное учреждение высшего образования

«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ   
АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

КАФЕДРА №34

ОТЧЕТ ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| доцент, к.т.н. |  |  |  | В.А. Мыльников |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

|  |
| --- |
| ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №2  **Разработка структуры базы данных с учетом снижения рисков** |
|  |
| по курсу: БЕЗОПАСНОСТЬ СИСТЕМ БАЗ ДАННЫХ |
|  |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. № | 3843 |  |  |  | А.П.Конева |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург 2021

**Цель работы**

Приобретение навыков по исследованию баз данных на предмет устранения рисков информационной безопасности.

**Задача**

Разработать результирующую логическую и физическую модель диаграммы «сущность-связь» для разрабатываемой информационной системы.

Логическая модель данных позволяет выделить ключевые сущности и связи между ними, атрибуты и их типы данных. Данная диаграмма отображает структуру создаваемой базы данных и используется для точного и полного отображения реальной предметной области.

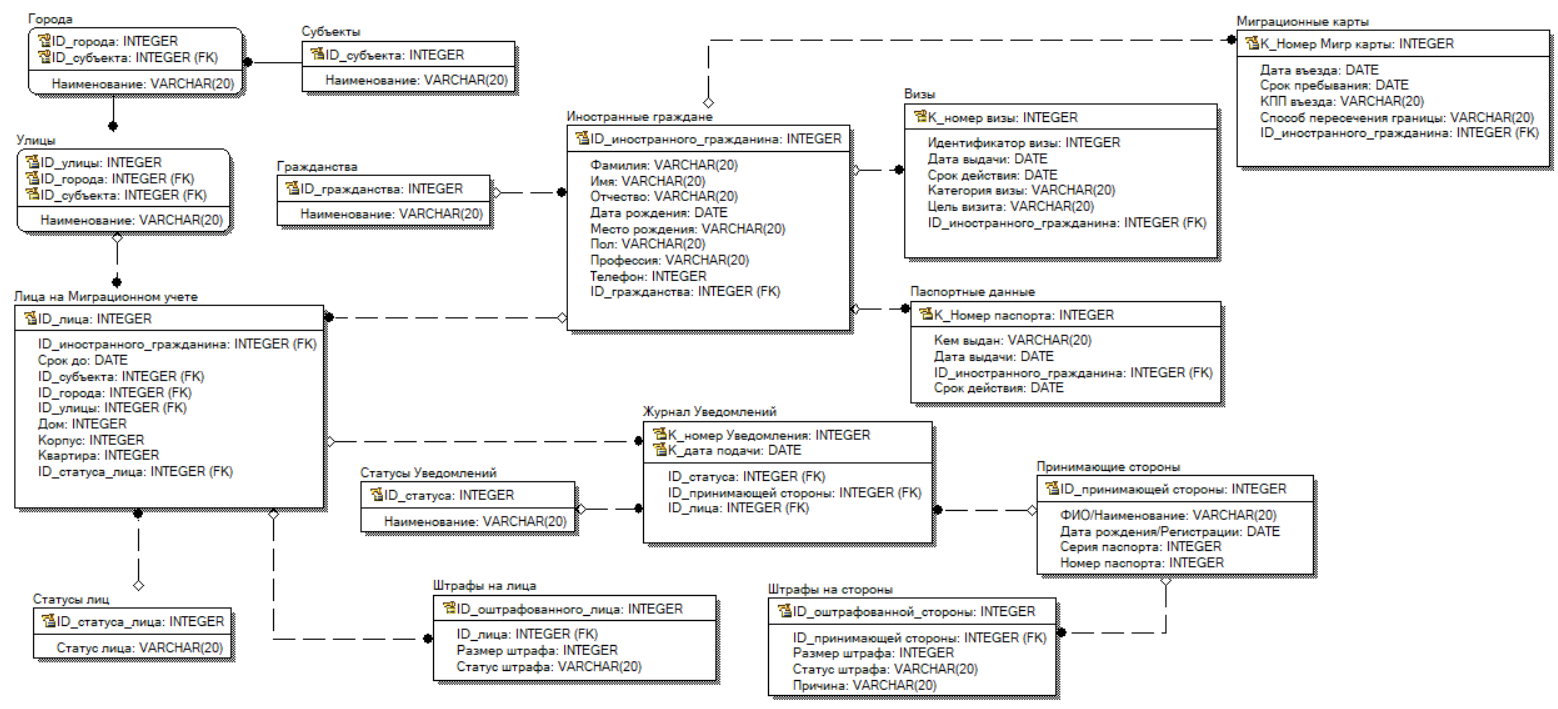
 ER- диаграмма логического уровня представлена на Рис.1.

Рисунок 1 - Логическая модель данных

Для корректной работы информационной системы необходимы следующие сущности:

1. «Журнал Уведомлений»
2. «Штрафы»
3. «Лица на Миграционном учете»
4. «Списки лиц, которым запрещен въезд в РФ»
5. «Списки лиц, нарушивших Миграционное законодательство, выявленные в результате проверок граждан сотрудниками Полиции»
6. «Документы от иностранного гражданина»
7. «Принимающие стороны»

Для снижения рисков, связанных с неверной интерпретацией связей между сущностями, с аномалиями технологических операций, истолкования отдельной сущности как атрибута другой, неверного представления многозначных атрибутов, неверной интерпретации рекурсивных связей, приведем все используемые сущности к 3НФ.

Приведем сущность «Документы от иностранного гражданина» к 1НФ. Для этого разобьем атрибут ФИО на атомарные атрибуты Фамилия, Имя, Отчество. Поскольку атрибуты Идентификатор визы, Дата выдачи, Срок действия визы, Категория визы, Цель визита напрямую зависят от Номера визы, перенесем их в новую сущность «Визы», где Первичным ключом является «Номер визы». Поскольку атрибуты Дата въезда, Срок пребывания, КПП въезда и Способ пересечения границы зависят напрямую от Номера Миграционной карты, перенесем их в новую сущность «Миграционные карты», где первичным ключом является «Номер Миграционной карты». Атрибуты Кем выдан паспорт, Дата выдачи паспорта и Срок действия напрямую зависят от атрибута «Номера паспорта», что будет являться первичным ключом в отдельной сущности «Паспортные данные». Создадим сущность «Гражданства», являющуюся выпадающим списком с первичным ключом «ID гражданства» и атрибутом Наименование гражданства. Оставшиеся атрибуты Фамилия, Имя, Отчество, Дата рождения, Место рождения, Номер визы, Пол, Профессия, Телефон выделим в сущность «Иностранные граждане» с первичным ключом «ID иностранного гражданина». Так как перечисленные сущности находятся в 1НФ, и каждый не ключевой атрибут функционально полно зависит от первичного ключа, сущности «Визы» и «Миграционные карты», «Иностранные граждане» и «Паспортные данные» приведены ко 2НФ. Поскольку теперь перечисленные сущности находятся во 2НФ и внутри всех перечисленных сущностей отсутствуют транзитивные зависимости атрибутов от соответствующего первичного ключа, они приведены к 3НФ.

Сущность «Принимающие стороны» приведена к 1НФ, поскольку первичный ключ «ID принимающей стороны» и атрибуты Наименование/ФИО, Дата рождения/Регистрации, Серия паспорта, Номер паспорта атомарны. Так как теперь сущность в 1НФ и каждый не ключевой атрибут функционально полно зависит от первичного ключа «ID принимающей стороны», эта сущность приведена ко 2НФ. Поскольку каждый не ключевой атрибут не транзитивно зависит от первичного ключа, сущность находится в 3НФ.

Приведем сущность «Лица на Миграционном учете» с первичным ключом «ID лица» к 1НФ, для чего разобьем атрибут Адрес на атомарные атрибуты Область, Город, Улица, Дом, Корпус, Квартира. Далее, приводя ко 2НФ, выделим отдельные сущности «Области», «Города», «Улицы», (где атрибутами являются Наименования, а первичными ключами – соответствующие «ID Области», «ID города» и «ID улицы»), которые будут являться выпадающими списками в БД. Поскольку во всех перечисленных сущностях не ключевые атрибуты не транзитивно зависят от соответствующих первичных ключей, сущности находятся в 3НФ.

Чтобы избежать дублирования одной и той же информации (такой как, например, номер Миграционной карты, серия и номер паспорта иностранного гражданина) в нескольких таблицах, объединим информацию из сущностей «Списки лиц, которым запрещен въезд в РФ», «Списки лиц, нарушивших Миграционное законодательство, выявленные в результате проверок граждан сотрудниками Полиции» и сформируем сущность «Статусы лиц» с первичным ключом «ID статуса лица», которая будет представлена в БД в виде выпадающего списка. Поскольку значение атрибута Статус лица атомарно, сущность находится в 1НФ. Так как сущность находится в 1НФ, и каждый не ключевой атрибут функционально полно зависит от первичного ключа, она приведена ко 2НФ, а отсутствие транзитивной зависимости не ключевого атрибута от первичного ключа говорит о 3НФ.

Сущность «Журнал Уведомлений» приведена к 1НФ, поскольку значения первичного ключа «Номер Уведомления» и «Дата подачи» и не ключевого атрибута «Статус Уведомления» и внешнего ключа «ID принимающей стороны» атомарны. Чтобы привести ко 2НФ и создать выпадающий список в БД, выделим сущность «Статусы Уведомлений» с первичным ключом «ID статуса» и атрибутом Наименование. Поскольку в обеих сущностях не ключевые атрибуты не транзитивно зависят от соответствующих первичных ключей, сущности находятся в 3НФ.

Сущность «Штрафы» разделим на сущности «Штрафы на лица» с первичным ключом «ID оштрафованного лица» и атрибутами ID лица, Размер штрафа, Статус штрафа и «Штрафы на стороны» с первичным ключом «ID оштрафованной стороны» и атрибутами ID принимающей стороны, Размер штрафа, Статус штрафа, Причина. Эти сущности приведены к 1НФ, поскольку все атрибуты атомарны. Так как сущности находится в 1НФ, и каждый не ключевой атрибут функционально полно зависит от первичного ключа, они приведены ко 2НФ, а отсутствие транзитивной зависимости не ключевого атрибута от первичного ключа говорит о 3НФ.

После приведения всех сущностей к 3НФ получили следующие сущности:

1. Визы
2. Иностранные граждане
3. Миграционные карты
4. Паспортные данные
5. Гражданства
6. Лица на Миграционном учете
7. Области
8. Города
9. Улицы
10. Статусы лиц
11. Журнал Уведомлений
12. Статусы Уведомлений
13. Принимающие стороны
14. Штрафы на лица
15. Штрафы на стороны

Рассмотрим группу сущностей и взаимосвязи между ними. Сущности «Миграционные карты» и «Иностранные граждане», «Иностранные граждане» и «Паспортные данные», «Визы» и «Иностранные граждане», «Лица на Миграционном учете» и «Иностранные граждане», «Иностранные граждане» и «Гражданства» связаны между собой не идентифицирующей связью.

Сущность «Миграционные карты» представляет собой таблицу, в которой содержится информация из Миграционных карт.

Первичный ключ: Номер Миграционной карты.

Внешние ключи: ID иностранного гражданина.

Атрибуты: Дата въезда, Срок пребывания, КПП въезда, Способ пересечения границы.

Сущность «Визы» представляет собой таблицу, в которой содержится информация из Виз.

Первичный ключ: Номер Визы.

Внешний ключ: ID иностранного гражданина.

Атрибуты: Идентификатор визы, Дата выдачи, Срок действия, Категория визы, Цель визита.

Сущность «Паспортные данные» представляет собой таблицу, в которой содержатся сведения из заграничных паспортов иностранных граждан.

Первичный ключ: Номер паспорта.

Внешний ключ: ID иностранного гражданина.

Атрибуты: Кем выдан, Дата выдачи, Срок действия.

Сущность «Лица на Миграционном учете» представляет собой таблицу, в которой содержатся сведения о лицах, состоящих на Миграционном учете, включая наиважнейшие атрибуты, составляющие адрес временной регистрации иностранного гражданина.

Первичный ключ: ID лица.

Внешние ключи: ID иностранного гражданина, ID субъекта, ID города, ID улицы, ID статуса лица.

Атрибуты: Срок до, Дом, Корпус, Квартира.

Сущность «Иностранные граждане» представляет собой таблицу, в которой содержится информация из документов иностранного гражданина, личные сведения.

Первичный ключ: ID иностранного гражданина.

Внешний ключ: ID гражданства.

Атрибуты: Фамилия, Имя, Отчество, Дата рождения, Место рождения, Пол, Профессия, Телефон.

Сущность «Гражданства» представляет собой выпадающий список, в котором содержатся возможные гражданства, например, Белоруссия, США, Канада и т.д.

Первичный ключ: ID гражданства.

Атрибут: Наименование.

Рассмотрим следующую группу сущностей и взаимосвязи между ними. Сущности «Лица на Миграционном учете» и «Статусы лиц», «Лица на Миграционном учете» и «Улицы», «Лица на Миграционном учете» и «Штрафы на лица», «Лица на Миграционном учете» и «Журнал Уведомлений», «Журнал Уведомлений» и «Принимающие стороны», «Принимающие стороны» и «Штрафы на стороны», «Журнал Уведомлений» и «Статусы Уведомлений» связаны не идентифицирующей связью. Сущности «Улицы» и «Города», «Города» и «Области» – идентифицирующей связью.

Сущность «Статусы лиц» представляет собой выпадающий список, в котором содержатся возможные статусы лиц, состоящих на миграционном учете, например, На учете, Оштрафовано за потерю Уведомления, Оштрафовано за просроченную регистрацию, Оштрафовано за отсутствие регистрации, Въезд запрещен.

Первичный ключ: ID статуса лица.

Атрибут: Статус лица.

Сущность «Субъекты» представляет собой выпадающий список, в котором содержатся наименования возможных областей, например, Архангельская, Ленинградская, т.д.

Первичный ключ: ID субъекта.

Атрибут: Наименование.

Сущность «Города» представляет собой выпадающий список, в котором содержатся возможные города в конкретной области.

Первичный ключ: ID города, ID области.

Атрибут: Наименование.

Сущность «Улицы» представляет собой выпадающий список, в котором содержатся наименования возможных улиц в конкретном городе.

Первичный ключ: ID улицы, ID города, ID области.

Атрибут: Наименование.

Сущность «Принимающие стороны» представляет собой таблицу, в которой содержится информация о принимающей стороне, его ФИО, дата рождения, серия и номер паспорта в случае физического лица или Наименование, дата регистрации - в случае юридического лица.

Первичный ключ: ID принимающей стороны.

Атрибуты: ФИО/Наименование, Дата рождения/Регистрации, Серия паспорта, Номер паспорта.

Сущность «Штрафы на лица» представляет собой таблицу, в которой содержатся сведения об оштрафованном лице, размер штрафа и его статус (оплачен/не оплачен).

Первичный ключ: ID оштрафованного лица.

Внешний ключ: ID лица.

Атрибуты: Размер штрафа, Статус штрафа.

Сущность «Штрафы на стороны» представляет собой таблицу, в которой содержатся сведения об оштрафованной стороне, размер штрафа и его статус (оплачен/не оплачен).

Первичный ключ: ID оштрафованной стороны.

Внешний ключ: ID принимающей стороны.

Атрибуты: Размер штрафа, Статус штрафа, Причина.

Сущность «Журнал Уведомлений» представляет собой таблицу, в которой содержится информация о поданных Уведомлениях, его статус и номер, сведения о принимающей стороне.

Первичный ключ: Номер Уведомления, Дата подачи.

Внешние ключи: ID статуса, ID принимающей стороны, ID лица.

Сущность «Статусы Уведомлений» представляет собой выпадающий список, в котором содержатся возможные статусы Уведомлений, например, Принято, На рассмотрении, Отказано в приеме, Сведения не прошли проверку.

Первичный ключ: ID статуса.

Атрибут: Наименование.

Для данной лабораторной работы была выбрана СУБД Microsoft SQL Server. Рассмотрим основные преимущества выбранной СУБД:

* Масштабирование системы. Взаимодействовать с ней можно как на простых ноутбуках, так и на ПК с мощным процессором, который способен обрабатывать большой объем запросов.
* Автоматизация рутинных административных задач. Например, управление блокировками и памятью, редактура размеров файлов. В программе продуманы настройки, можно создавать профили пользователей.
* Удобный поиск. Его можно осуществлять по фразам, словам, тексту либо создавать ключевые индексы.

К минусам можно отнести высокую стоимость.

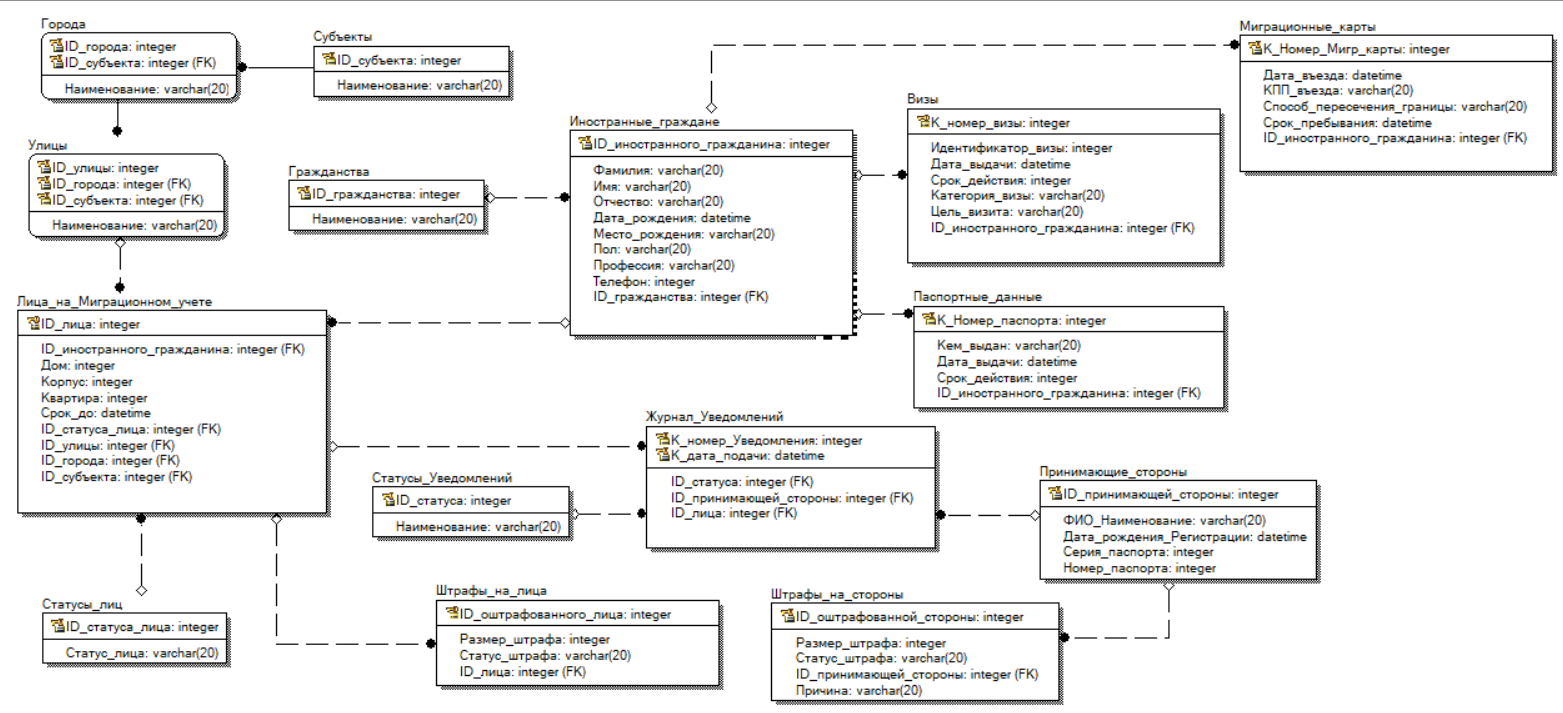
С учетом особенностей выбранной СУБД разработали ER-диаграмму физического уровня, которая представлена на Рис.2.

Рисунок 2 - Физическая модель данных

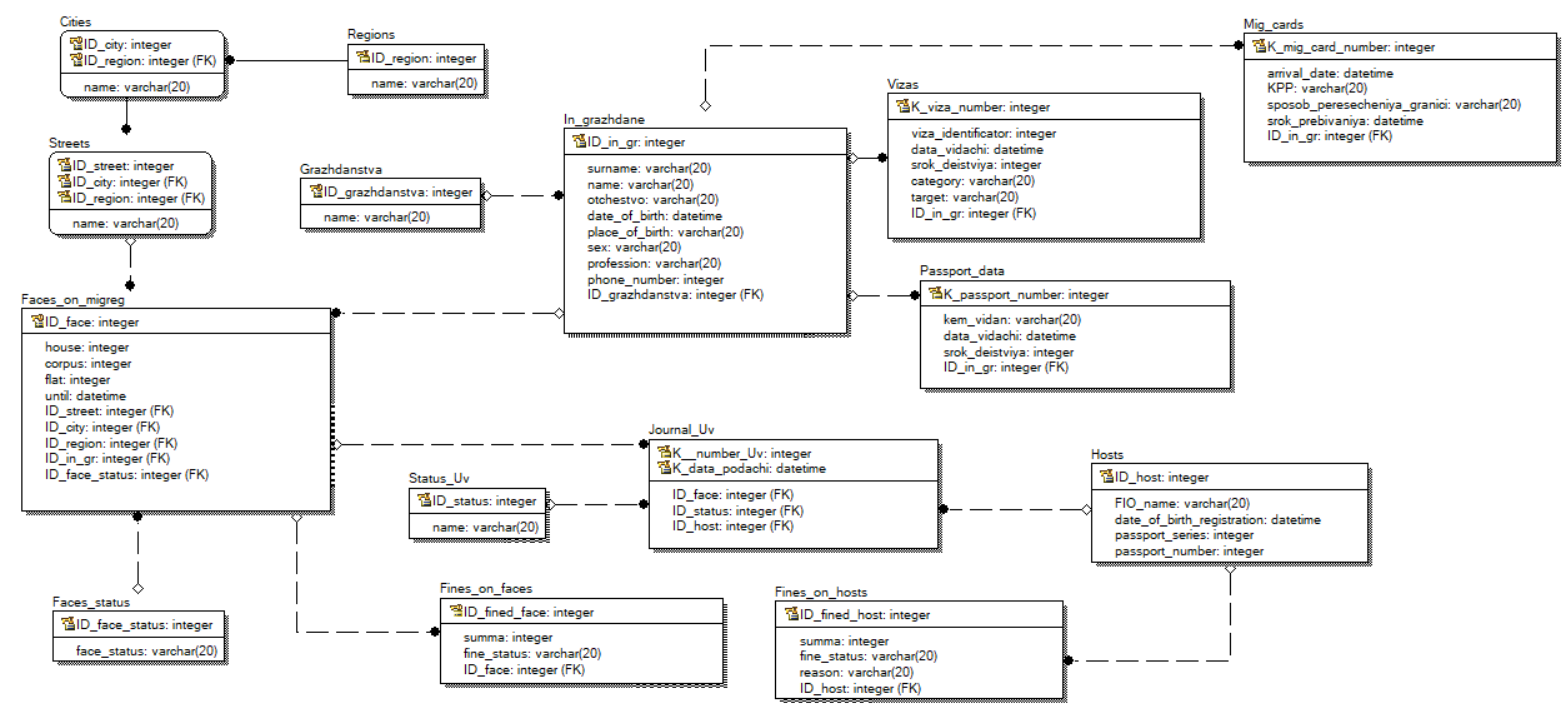
Физическая модель данных, латинизированная для взаимодействия с MS SQL Server, представлена на Рис.3.

Рисунок 3 - Физическая модель данных, латинизированная для взаимодействия с MS SQL Server

Для удобства работы с базой данных были созданы представления, указанные в Таблице 1.

Таблица 1. Список представлений для разрабатываемой системы

|  |  |  |
| --- | --- | --- |
| Название представления | Сотрудник | Участвующие элементы |
| Информация о лицах, состоящих на Миграционном учете | Инспектор УВМ МВД;  Уполномоченный сотрудник УВМ МВД | Лица на Миграционном учете  Иностранные граждане |
| Информация о лицах, которым запрещен въезд в РФ | Инспектор УВМ МВД | Иностранные граждане  Паспортные данные  Статус лица |
| Информация об оштрафованных лицах | Инспектор УВМ МВД;  Уполномоченный сотрудник УВМ МВД | Лица на Миграционном учете  Штрафы на лица  Иностранные граждане  Паспортные данные |
| Информация об оштрафованных Принимающих сторонах | Инспектор УВМ МВД;  Уполномоченный сотрудник УВМ МВД | Штрафы на стороны  Принимающие стороны |
| Информация об Уведомлениях | Уполномоченный сотрудник УВМ МВД | Журнал Уведомлений  Лица на Миграционном учете  Иностранные граждане  Принимающие стороны |

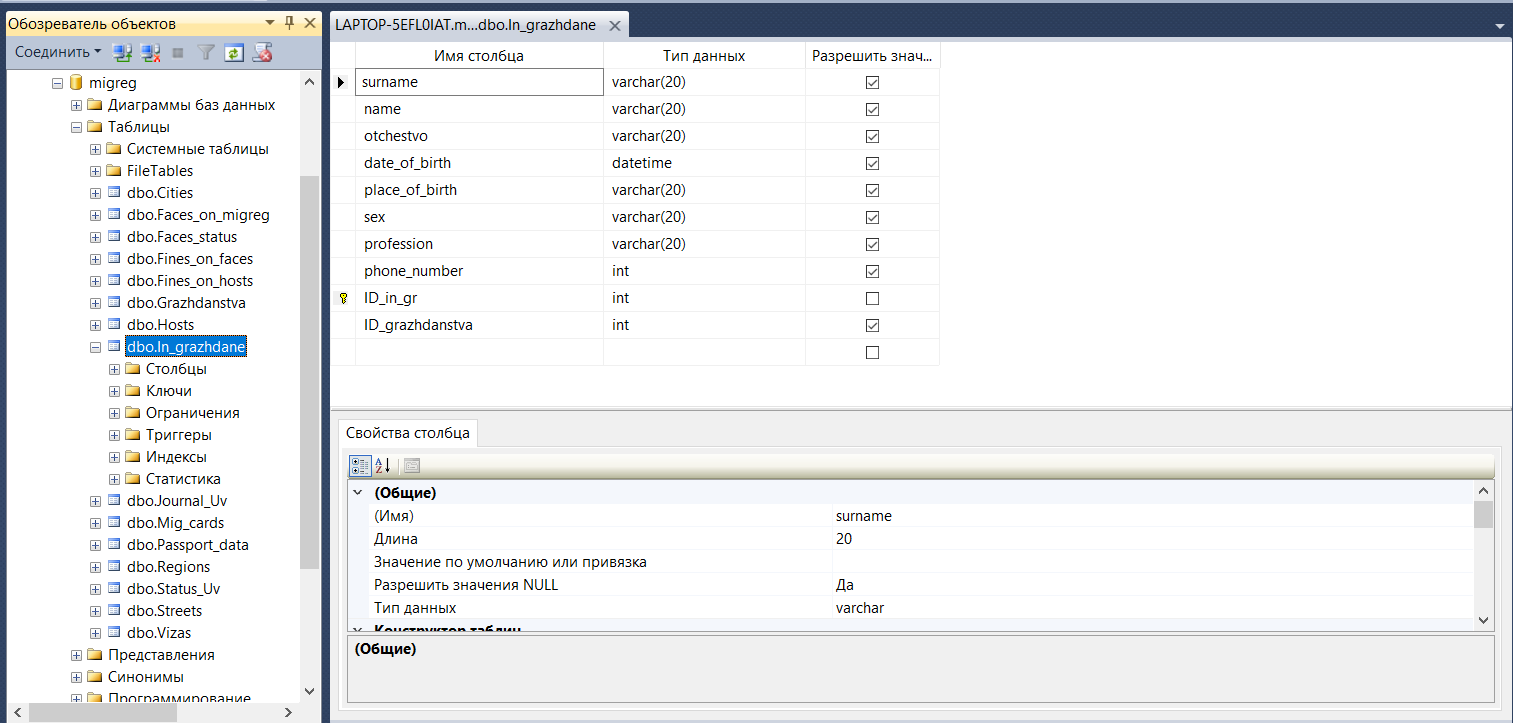
На Рис.4 представлен результат генерации SQL-сценария, а именно: созданная база данных migreg в SQL Server и соответствующие таблицы.

Рисунок 4 - Результат генерации SQL-сценария

**Сгенерированный SQL-сценарий для реализации структуры базы данных:**

CREATE TABLE Hosts

(

FIO\_name varchar(20) NULL ,

ID\_host integer NOT NULL ,

date\_of\_birth\_registration datetime NULL ,

passport\_series integer NULL ,

passport\_number integer NULL ,

CONSTRAINT XPKHosts PRIMARY KEY NONCLUSTERED (ID\_host ASC)

)

go

CREATE TABLE Fines\_on\_hosts

(

ID\_fined\_host integer NOT NULL ,

summa integer NULL ,

fine\_status varchar(20) NULL ,

reason varchar(20) NULL ,

ID\_host integer NULL ,

CONSTRAINT XPKFines\_on\_hosts PRIMARY KEY CLUSTERED (ID\_fined\_host ASC),

CONSTRAINT R\_100 FOREIGN KEY (ID\_host) REFERENCES Hosts(ID\_host)

ON DELETE NO ACTION

ON UPDATE NO ACTION

)

go

CREATE NONCLUSTERED INDEX XIF1Fines\_on\_hosts ON Fines\_on\_hosts

(

ID\_host ASC

)

go

CREATE TABLE Regions

(

ID\_region integer NOT NULL ,

name varchar(20) NULL ,

CONSTRAINT XPKRegions PRIMARY KEY NONCLUSTERED (ID\_region ASC)

)

go

CREATE TABLE Cities

(

ID\_city integer NOT NULL ,

name varchar(20) NULL ,

ID\_region integer NOT NULL ,

CONSTRAINT XPKCities PRIMARY KEY CLUSTERED (ID\_city ASC,ID\_region ASC),

CONSTRAINT R\_84 FOREIGN KEY (ID\_region) REFERENCES Regions(ID\_region)

ON DELETE NO ACTION

ON UPDATE NO ACTION

)

go

CREATE NONCLUSTERED INDEX XIF1Cities ON Cities

(

ID\_region ASC

)

go

CREATE TABLE Streets

(

ID\_street integer NOT NULL ,

name varchar(20) NULL ,

ID\_city integer NOT NULL ,

ID\_region integer NOT NULL ,

CONSTRAINT XPKStreets PRIMARY KEY CLUSTERED (ID\_street ASC,ID\_city ASC,ID\_region ASC),

CONSTRAINT R\_85 FOREIGN KEY (ID\_city,ID\_region) REFERENCES Cities(ID\_city,ID\_region)

ON DELETE NO ACTION

ON UPDATE NO ACTION

)

go

CREATE NONCLUSTERED INDEX XIF1Streets ON Streets

(

ID\_city ASC,

ID\_region ASC

)

go

CREATE TABLE Grazhdanstva

(

name varchar(20) NULL ,

ID\_grazhdanstva integer NOT NULL ,

CONSTRAINT XPKGrazhdanstva PRIMARY KEY NONCLUSTERED (ID\_grazhdanstva ASC)

)

go

CREATE TABLE In\_grazhdane

(

surname varchar(20) NULL ,

name varchar(20) NULL ,

otchestvo varchar(20) NULL ,

date\_of\_birth datetime NULL ,

place\_of\_birth varchar(20) NULL ,

sex varchar(20) NULL ,

profession varchar(20) NULL ,

phone\_number integer NULL ,

ID\_in\_gr integer NOT NULL ,

ID\_grazhdanstva integer NULL ,

CONSTRAINT XPKIn\_grazhdane PRIMARY KEY CLUSTERED (ID\_in\_gr ASC),

CONSTRAINT R\_86 FOREIGN KEY (ID\_grazhdanstva) REFERENCES Grazhdanstva(ID\_grazhdanstva)

ON DELETE NO ACTION

ON UPDATE NO ACTION

)

go

CREATE NONCLUSTERED INDEX XIF1In\_grazhdane ON In\_grazhdane

(

ID\_grazhdanstva ASC

)

go

CREATE TABLE Faces\_status

(

ID\_face\_status integer NOT NULL ,

face\_status varchar(20) NULL ,

CONSTRAINT XPKFaces\_status PRIMARY KEY NONCLUSTERED (ID\_face\_status ASC)

)

go

CREATE TABLE Faces\_on\_migreg

(

ID\_face integer NOT NULL ,

house integer NULL ,

corpus integer NULL ,

flat integer NULL ,

until datetime NULL ,

ID\_street integer NULL ,

ID\_city integer NULL ,

ID\_region integer NULL ,

ID\_in\_gr integer NULL ,

ID\_face\_status integer NULL ,

CONSTRAINT XPKFaces\_on\_migreg PRIMARY KEY CLUSTERED (ID\_face ASC),

CONSTRAINT R\_87 FOREIGN KEY (ID\_street,ID\_city,ID\_region) REFERENCES Streets(ID\_street,ID\_city,ID\_region)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT R\_88 FOREIGN KEY (ID\_in\_gr) REFERENCES In\_grazhdane(ID\_in\_gr)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT R\_89 FOREIGN KEY (ID\_face\_status) REFERENCES Faces\_status(ID\_face\_status)

ON DELETE NO ACTION

ON UPDATE NO ACTION

)

go

CREATE NONCLUSTERED INDEX XIF1Faces\_on\_migreg ON Faces\_on\_migreg

(

ID\_street ASC,

ID\_city ASC,

ID\_region ASC

)

go

CREATE NONCLUSTERED INDEX XIF2Faces\_on\_migreg ON Faces\_on\_migreg

(

ID\_in\_gr ASC

)

go

CREATE NONCLUSTERED INDEX XIF3Faces\_on\_migreg ON Faces\_on\_migreg

(

ID\_face\_status ASC

)

go

CREATE TABLE Fines\_on\_faces

(

ID\_fined\_face integer NOT NULL ,

summa integer NULL ,

fine\_status varchar(20) NULL ,

ID\_face integer NULL ,

CONSTRAINT XPKFines\_on\_faces PRIMARY KEY CLUSTERED (ID\_fined\_face ASC),

CONSTRAINT R\_96 FOREIGN KEY (ID\_face) REFERENCES Faces\_on\_migreg(ID\_face)

ON DELETE NO ACTION

ON UPDATE NO ACTION

)

go

CREATE NONCLUSTERED INDEX XIF1Fines\_on\_faces ON Fines\_on\_faces

(

ID\_face ASC

)

go

CREATE TABLE Status\_Uv

(

ID\_status integer NOT NULL ,

name varchar(20) NULL ,

CONSTRAINT XPKStatus\_Uv PRIMARY KEY NONCLUSTERED (ID\_status ASC)

)

go

CREATE TABLE Journal\_Uv

(

K\_\_number\_Uv integer NOT NULL ,

K\_data\_podachi datetime NOT NULL ,

ID\_face integer NULL ,

ID\_status integer NULL ,

ID\_host integer NULL ,

CONSTRAINT XPKJournal\_Uv PRIMARY KEY CLUSTERED (K\_\_number\_Uv ASC,K\_data\_podachi ASC),

CONSTRAINT R\_97 FOREIGN KEY (ID\_face) REFERENCES Faces\_on\_migreg(ID\_face)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT R\_98 FOREIGN KEY (ID\_status) REFERENCES Status\_Uv(ID\_status)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT R\_99 FOREIGN KEY (ID\_host) REFERENCES Hosts(ID\_host)

ON DELETE NO ACTION

ON UPDATE NO ACTION

)

go

CREATE NONCLUSTERED INDEX XIF1Journal\_Uv ON Journal\_Uv

(

ID\_face ASC

)

go

CREATE NONCLUSTERED INDEX XIF2Journal\_Uv ON Journal\_Uv

(

ID\_status ASC

)

go

CREATE NONCLUSTERED INDEX XIF3Journal\_Uv ON Journal\_Uv

(

ID\_host ASC

)

go

CREATE TABLE Passport\_data

(

K\_passport\_number integer NOT NULL ,

kem\_vidan varchar(20) NULL ,

data\_vidachi datetime NULL ,

ID\_in\_gr integer NULL ,

srok\_deistviya integer NULL ,

CONSTRAINT XPKPassport\_data PRIMARY KEY CLUSTERED (K\_passport\_number ASC),

CONSTRAINT R\_95 FOREIGN KEY (ID\_in\_gr) REFERENCES In\_grazhdane(ID\_in\_gr)

ON DELETE NO ACTION

ON UPDATE NO ACTION

)

go

CREATE NONCLUSTERED INDEX XIF1Passport\_data ON Passport\_data

(

ID\_in\_gr ASC

)

go

CREATE TABLE Vizas

(

K\_viza\_number integer NOT NULL ,

data\_vidachi datetime NULL ,

category varchar(20) NULL ,

target varchar(20) NULL ,

viza\_identificator integer NULL ,

ID\_in\_gr integer NULL ,

srok\_deistviya integer NULL ,

CONSTRAINT XPKVizas PRIMARY KEY CLUSTERED (K\_viza\_number ASC),

CONSTRAINT R\_92 FOREIGN KEY (ID\_in\_gr) REFERENCES In\_grazhdane(ID\_in\_gr)

ON DELETE NO ACTION

ON UPDATE NO ACTION

)

go

CREATE NONCLUSTERED INDEX XIF1Vizas ON Vizas

(

ID\_in\_gr ASC

)

go

CREATE TABLE Mig\_cards

(

K\_mig\_card\_number integer NOT NULL ,

KPP varchar(20) NULL ,

sposob\_peresecheniya\_granici varchar(20) NULL ,

arrival\_date datetime NULL ,

srok\_prebivaniya datetime NULL ,

ID\_in\_gr integer NULL ,

CONSTRAINT XPKMig\_cards PRIMARY KEY CLUSTERED (K\_mig\_card\_number ASC),

CONSTRAINT R\_90 FOREIGN KEY (ID\_in\_gr) REFERENCES In\_grazhdane(ID\_in\_gr)

ON DELETE NO ACTION

ON UPDATE NO ACTION

)

go

CREATE NONCLUSTERED INDEX XIF1Mig\_cards ON Mig\_cards

(

ID\_in\_gr ASC

)

go

**Вывод**

Разработали результирующую логическую модель данных диаграммы «сущность-связь» (Рис.1) для разрабатываемой информационной системы. Привели ее к 3НФ, определились с составом первичных ключевых полей. Выбрали СУБД MS SQL Server. С учетом особенностей выбранной СУБД разработали ER-диаграмму физического уровня, которая представлена на Рис.2. Для удобства работы с базой данных были созданы представления, указанные в Таблице 1. На Рис.4 представлен результат генерации SQL-сценария, а именно: созданная база данных migreg в MS SQL Server и соответствующие таблицы.