Министерство образования и молодежной политики

Свердловской области

ГАПОУ СО «Екатеринбургский колледж транспортного строительства»

Специальность 09.02.07: «Информационные системы и программирование»

Разработка информационно-поисковой системы для бронирования авиабилетов

**Пояснительная записка**

к курсовому проекту

КР-ПР-41-01-2022-ПЗ

Разработал:

Студент гр. ПР-41 \_\_\_\_\_\_\_\_\_\_\_\_\_\_/М.А. Варганов

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_/С.И. Овчинникова

2022

Министерство образования и молодёжной политики Свердловской области

ГАПОУ СО Екатеринбургский колледж транспортного строительства

Специальность 090207 «Информационные системы и программирование»

**ЗАДАНИЕ**

**на курсовой проект**

**Курс IV**

**Студенту:**

**Фамилия** Варганов

**Имя** Матвей

**Отчество** АлексеевичГруппа ПР-41

Тема курсового проекта: Разработка информационно-поисковой системы

для бронирования авиабилетов

**Курсовой проект должен содержать следующие разделы:**

Титульные листы

Задание на курсовой проект

Содержание

Введение

1. Описание предметной области. Постановка задачи
2. Выбор средств/методологии проектирования. Выбор СУБД
3. Построение инфологической (концептуальной) модели предметной области
4. Проектирование логической структуры БД
5. Формулировка ограничений целостности

5.1 Выявление полного перечня ограниченной целостности, присущего данной предметной области

5.2 Определение перечня ограничений целостности, которые будут контролироваться в данном проекте

5.3 Выбор способа реализации контроля целостности для каждого из ограничений

1. Проектирование физической структуры базы данных
2. Организация ввода данных в БД
3. Организация корректировки БД
4. Реализация запросов пользователя, получение отчетов
5. Разработка пользовательского интерфейса
6. Реализация проекта в среде конкретной СУБД

Условные сокращения

Заключение

Список литературы

Приложения

Приложение А Основные таблицы БД

Приложение Б Схема межтабличных связей

Приложение В Интерфейсные формы диалога пользователей с БД

Приложение Г Образцы отчетов пользователей

Приложение Д Образцы запросов пользователей к БД

Дата выдачи\_\_23 сентября 2022\_\_\_

Дата окончания\_\_\_\_5 декабря 2022\_\_

Преподаватель Овчинникова С.И

Содержание

# Введение

В современных условиях, человек вынужден работать с гигантскими объемами информации. В связи с этим разработка программных продуктов, служащих для автоматизированного учета, весьма актуальна.

Системы обязаны представлять собой мощные средства, способные обрабатывать гигантские потоки данных высокой структурной сложности за минимум затраченного времени, обеспечивая дружественный диалог с пользователем Целью данной выпускной квалификационной работы является создание автоматизированной информационной системы, осуществляющей продажи авиабилетов. Разработка подобной системы весьма актуальна на данный момент. В современном мире самолеты являются не только самым быстрым видом транспорта, но и самым безопасным, в связи с этим авиаперелеты пользуются весьма высокой популярностью. Вследствие этого продаваемые на рейсы билеты востребованы и с высокой вероятностью найдут своего покупателя, при условии, что авиакомпания обеспечила клиенту полноценный доступ к нужной ему информации. Это и есть задача, решаемая современными автоматизированными информационными системами. Существует множество подобных разработок, позволяющих авиакомпаниям реализовывать авиабилеты, а пользователям приобретать их. Однако, зачастую, функциональность таких систем либо весьма ограничена, либо предоставляет достаточное количество информации, жертвуя дружественностью к пользователю.

Для реализации поставленной задачи были выбраны следующие средства разработки: MS Visual Studio 2022, MS SQL Server 2019, веб-платформа Angular, и языки программирования TypeScript, C#, SQL.

# 1 Описание предметной области

Целью данной курсовой работы является создание программного средства, осуществляющего поиск и бронирование авиабилетов. Авиакомпания занимается авиаперевозками пассажиров. Также она устанавливает маршруты полетов. Рейсы осуществляются по установленным маршрутам согласно расписанию. На каждый рейс существует определенное количество билетов. Продажа билета пользователю осуществляется при отправке запроса на бронирование, при условии, что данный билет до сих пор есть в наличии. Приобретя билет, пользователь предоставляет информацию о себе и становится пассажиром. Совершеннолетние пассажиры обязаны иметь занесенные в БД паспортные данные. Несовершеннолетние обязаны иметь занесенные в БД данные из свидетельства о рождении. Администраторы системы могут ограничивать или расширять доступ пользователей и сотрудников к предоставляемой информации.

Система создаётся для обслуживания пользователей, бронирующих билеты на рейсы и осуществляющих поиск.

Абсолютно каждая авиакомпания использует определенную систему дистрибуции. Наиболее развитые используют GDS (глобальные дистрибьюторские системы, которые формируются из основных международных компьютерных систем резервирования). В итоге сервисы продаж авиабилетов при поиске информации пользуются ресурсами глобальных дистрибьюторских систем. Однако доступ к GDS является не бесплатным, поэтому в роли дистрибутивной системы для разрабатываемого продукта будет выступать БД, созданная MS SQL Server 2019.

В БД должна храниться информация о:

* рейсах
* маршрутах
* пользователях
* билетах
* авиакомпаниях
* аэропортах
* самолётах

Разрабатываемая информационная система предназначена для продажи авиабилетов и упрощения доступа к нужной информации. Наличие данной разработки улучшает организационную работу авиаперевозчика за счёт отсутствия бумажной документации, поиск и систематизация которой занимали бы очень большое количество времени.

# 2 Выбор средств/методологии проектирования. Выбор СУБД

Для разработки информационной системы были выбраны язык программирования С#, MS Visual Studio 2022, язык программирования базы данных и запросов к ней SQL и система управления базами данных Microsoft SQL Server 2019, язык программирования TypeScript для разработки веб-интерфейса в фреймворке Angular.

Язык C# является объектно-ориентированным языком программирования. Разработан в 1998—2001 годах группой инженеров под руководством Андерса Хейлсберга в компании Microsoft как основной язык разработки приложений для платформы Microsoft .NET и впоследствии был стандартизирован как ECMA-334 и ISO/IEC 23270. Компилятор с C# входит в стандартную установку самой .NET, поэтому программы на нём можно создавать и компилировать даже без инструментальных средств, вроде Visual Studio.

C# относится к семье языков с C-подобным синтаксисом, из них его синтаксис наиболее близок к C++ и Java. Язык имеет статическую типизацию, поддерживает полиморфизм, перегрузку операторов (в том числе операторов явного и неявного приведения типа), делегаты, атрибуты, события, свойства, обобщённые типы и методы, итераторы, анонимные функции с поддержкой замыканий, LINQ, исключения, комментарии в формате XML.

Переняв многое от своих предшественников — языков C++, Java, Delphi, Модуля и Smalltalk — С#, опираясь на практику их использования, исключает некоторые модели, зарекомендовавшие себя как проблематичные при разработке программных систем: так, C# не поддерживает множественное наследование классов (в отличие от C++).

Среда разработки Microsoft Visual Studio 2022 — это набор инструментов и средств, предназначенных для помощи разработчикам программ любого уровня квалификации в решении сложных задач и создания новаторских решений. Роль Visual Studio заключается в том, чтобы улучшить процесс разработки и упростить разработку высокоэффективных программ. В Visual Studio содержатся инструменты для всех этапов разработки программного обеспечения (разработка, тестирование, развертывание, интеграция и управления). Visual Studio разрабатывается таким образом, чтобы обеспечить высокую надежность и совместимость. Visual Studio обладает удачным сочетанием безопасности, масштабируемости и взаимодействия.

SQL (англ. Structured Query Language — «язык структурированных запросов») – универсальный компьютерный язык, применяемый для создания, модификации и управления данными в реляционных базах данных. SQL основывается на исчислении кортежей. Язык SQL ориентирован на операции с данными, представленными в виде логически взаимосвязанных совокупностей таблиц-отношений. Важнейшая особенность его структур – ориентация на конечный результат обработки данных, а не на процедуру этой обработки. Язык SQL сам определяет, где находятся данные, индексы и даже какие наиболее эффективные последовательности операций следует использовать для получения результата, а потому указывать эти детали в запросе к базе данных не требуется. Основные особенности SQL:

* независимость от конкретной СУБД – несмотря на наличие диалектов и различий в синтаксисе, в большинстве своём тексты SQL-запросов, содержащие DDL и DML, могут быть достаточно легко перенесены из одной СУБД в другую. Существуют системы, разработчики которых изначально ориентировались на применение по меньшей мере нескольких СУБД (например: система электронного документооборота Documentum может работать как с Oracle, так и с Microsoft SQL Server и IBM DB2). Естественно, что при применении некоторых специфичных для реализации возможностей такой переносимости добиться уже очень трудно;
* наличие стандартов и набора тестов для выявления совместимости и соответствия конкретной реализации SQL общепринятому стандарту только способствует «стабилизации» языка. Правда, стоит обратить внимание, что сам по себе стандарт местами чересчур формализован и раздут в размерах (например, Core-часть стандарта SQL:2003 представляет собой более 1300 страниц текста);
* декларативность – с помощью SQL программист описывает только то, какие данные нужно извлечь или модифицировать. То, каким образом это сделать, решает СУБД непосредственно при обработке SQL-запроса. Однако не стоит думать, что это полностью универсальный принцип — программист описывает набор данных для выборки или модификации, однако ему при этом полезно представлять, как СУБД будет разбирать текст его запроса. Чем сложнее сконструирован запрос, тем больше он допускает вариантов написания, различных по скорости выполнения, но одинаковых по итоговому набору данных;
* реляционная основа языка – SQL является языком реляционных БД, поэтому он стал популярным тогда, когда получила широкое распространение реляционная модель представления данных. Табличная структура реляционной БД хорошо понятна, а потому язык SQL прост для изучения.

Microsoft SQL Server — система управления реляционными базами данных (СУБД), разработанная корпорацией Microsoft. Основной используемый язык запросов — Transact-SQL, создан совместно Microsoft и Sybase. Transact-SQL является реализацией стандарта ANSI/ISO по структурированному языку запросов (SQL) с расширениями.

На базе Microsoft SQL Server 2019 могут быть построены решения для компаний малого, среднего и крупного бизнеса. SQL Server 2019 выпускается в двух основных редакциях Standard и Enterprise. На основе последней создана также редакция для разработчиков Developer Edition, лицензия на которую позволяет разрабатывать и тестировать системы и приложения.

Enterprise-версия системы SQL Server 2019 представляет собой комплексную платформу, которая позволяет работать даже с самыми требовательными корпоративными OLTP-системами и хранилищами данных. Она обладает значительной масштабируемостью, возможностью создавать громадные хранилища данных, продвинутыми средствами анализа и усиленной безопасностью, что позволяет использовать ее как основу для критически 23 важных бизнес-приложений. Эта редакция позволяет консолидировать серверы и выполнять большое число OLTP-операций и крупные отчеты. Редакцию Microsoft SQL Server 2019 Enterprise характеризуют:

* высокий уровень доступности – непрерывность бизнес-процессов обеспечивается благодаря защите данных от дорогостоящих человеческих ошибок и максимальному уменьшению сроков аварийного восстановления;
* производительность и масштабируемость – инфраструктура на основе SQL Server 2019 Enterprise позволяет справиться с любыми пиковыми нагрузками;
* безопасность – вопросы конфиденциальности, а также соответствия нормативным требованиям решаются с помощью встроенных средств защиты от несанкционированного доступа;
* управляемость – автоматические диагностика, калибровка и настройка инфраструктуры позволяют управлять огромными объемами данных, значительно сократив издержки на управление и обслуживание;
* бизнес-аналитика. SQL Server 2019 Enterprise помогает легко собрать и проанализировать большие объемы данных из хранилищ или киосков.

Кроме редакций Enterprise и Standard, существуют специализированные редакции SQL Server 2019, одна из которых Express. Редакция Express — также доступная для бесплатной загрузки, эта редакция идеальна для обучения и создания настольных и небольших серверных приложений

Angular представляет фреймворк от компании Google для создания клиентских приложений. Прежде всего он нацелен на разработку SPA-решений (Single Page Application), то есть одностраничных приложений. В этом плане Angular является наследником другого фреймворка AngularJS. В то же время Angular это не новая версия AngularJS, а принципиально новый фреймворк.

Angular предоставляет такую функциональность, как двустороннее связывание, позволяющее динамически изменять данные в одном месте интерфейса при изменении данных модели в другом, шаблоны, маршрутизация и так далее.

Одной из ключевых особенностей Angular является то, что он использует в качестве языка программирования TypeScript.

Преимущества Angular заключается в обилии возможностей, он помогает привязывать компоненты приложения друг к другу, передавать данные, анимировать интерфейсы и пр. Для простых проектов его функциональность может быть избыточной, но для сложных SPA-приложений она незаменима.

Фреймворк позволяет создавать не только веб-приложения. С его помощью можно писать код, который может быть адаптирован под другую среду. Например, приложение сможет работать в мобильной или десктопной операционной системе. С помощью Angular можно создать даже приложение для дополненной реальности.

Разработчики Angular — сотрудники Google, а поддержка большой корпорации помогает фреймворку развиваться. При этом благодаря свободной лицензии и открытому исходному коду развивать его могут и сторонние разработчики.

3 Построение концептуальной модели предметной области

Был проведён подробный анализ предметной области, и на основе результатов этого анализа были выделены следующие сущности:

* регистрация
* рейс
* маршрут
* рейс-маршрут
* аэропорт
* самолёт

авиалиния

* город
* страна
* клиент
* паспорт
* контактные данные
* тариф

В процессе функционирования сущности взаимодействуют друг с другом. В концептуальной модели взаимодействие между сущностями выражается с помощью связей, основными из которых являются следующие:

Связь Регистрация – Рейс (Рисунок 3.1). Показывает, что регистрация содержит рейс. Связь 1 ко многим, так как на регистрацию может быть несколько клиентов с одним и тем же рейсом.



Рис. 3.1 – Связь Регистрация – Рейс

Связь Регистрация – клиент (Рисунок 3.2). Показывает, что регистрация содержит клиента. Связь 1 к М, так как Клиент может зарегистрироваться на несколько рейсов.



Рис. 3.2 – Связь Гражданин – права

Связь Маршрут – Рейс-Маршрут – Рейс (Рисунок 3.4). Данные таблицы построены данным образом для связи М к М. В ходе изучения предметной области выявилась нужда в такой модели, так как у рейса могут быть пересадки и смены самолётов. Клиент покупает билет на рейс, а сам рейс может содержать несколько маршрутов.

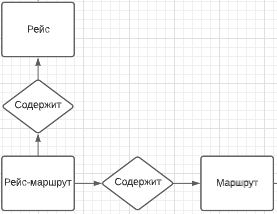


Рис. 3. – Маршрут – Рейс-Маршрут – Рейс

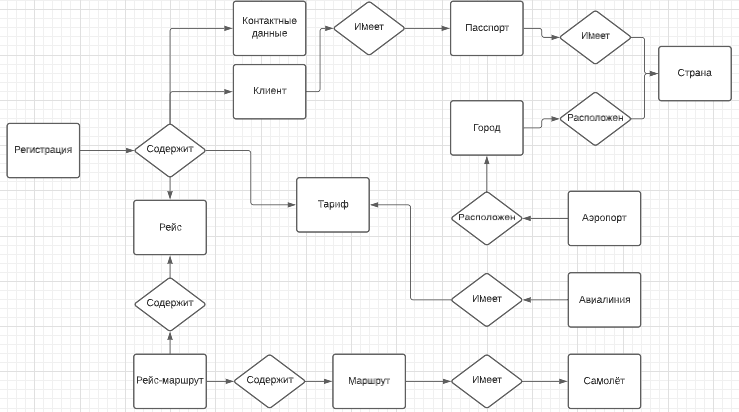


Рис. 3.5 – Концептуальная модель предметной области

Ранее были рассмотрены основные связи с итоговой таблицей Регистрация, далее будет рассмотрена вся модель.

Связь Аэропорт – Город. Храня данные о аэропорте нам нужно знать где он находится, а точнее в каком городе. Таблица Город является таблице списком.

Связь Город – Страна. Каждый город имеет страну, и важно знать не только в какой город приземляется или вылетает самолёт, ещё важно знать страну посадки или высадки.

Связь Авиалиния – Тариф. У каждой авиалинии есть свои тарифы, по которым летают клиенты, у каждого тарифа свои бонусы и наценка, которая влияет на итоговую сумму в регистрации.

Связь Маршрут – Самолёт. Каждый маршрут имеет свой самолёт. Стоит помнить, что маршрут не является рейсом. Клиент покупает билет на рейс, который состоит из маршрутов, так как рейс может быть с пересадками.

Связь Клиент – Паспорт. Для каждого клиента есть колонка с его паспортными данными. Вынесена таблица Паспорт для удобства и безопасности.

Связь Паспорт – Страна. В паспорте есть страна выдачи и гражданство, эта связь сделана для удобства

# 4 Проектирование логической структуры БД

На основе сущностей концептуальной модели предметной области были сформированы следующие таблицы:

Таблица 4.1 – Страна

|  |  |
| --- | --- |
| Имя столбца | Тип данных |
| id | int |
| name | nvarchar(50) |

Атрибуты таблицы:

* id – уникальный идентификатор для страны
* name – название страны

Таблица 4.2 – Таблица город

|  |  |
| --- | --- |
| Имя столбца | Тип данных |
| id | int |
| name | nvarchar(50) |
| country\_id | int |

Атрибуты таблицы:

* id – уникальный идентификатор для страны;
* name – название страны;
* country\_id – ключ страны;

|  |  |
| --- | --- |
| Имя столбца | Тип данных |
| id | int |
| birthdate | nvarchar(50) |
| citizenship\_id | int |
| number | nvarchar(50) |
| country\_of\_issue\_id | int |
| validity\_period | date |

Таблица 4.3 – Таблица Паспорт

Атрибуты таблицы:

* id – уникальный идентификатор паспорта;
* birthdate – день рождения;
* citizenship\_id – название страны гражданства;
* number – уникальный идентификатор паспорта (серия и номер);
* country\_of\_issue\_id – название страны выдачи;
* validity\_period – дата, до которой годен документ;

|  |  |
| --- | --- |
| Имя столбца | Тип данных |
| id | int |
| name | nvarchar(50) |
| surname | int |
| lastname | nvarchar(50) |
| passport\_id | date |

Таблица 4.4 – Таблица Клиент

Атрибуты таблицы:

* id – уникальный идентификатор клиента;
* name – имя клиента;
* surname – фамилия клиента;
* lastname – отчество клиента;
* passport\_id – ключ паспорта;

|  |  |
| --- | --- |
| Имя столбца | Тип данных |
| id | int |
| code | nvarchar(50) |
| name | nvarchar(50) |
| city\_id | int |

Таблица 4.5 - Таблица Аэропорт

Атрибуты таблицы:

* id – уникальный идентификатор;
* сode – сокращённое название аэропорта;
* name – название аэропорта;
* city\_id – ключ города;

|  |  |
| --- | --- |
| Имя столбца | Тип данных |
| id | int |
| name | nvarchar(100) |

Таблица 4.6 – Таблица Авиалиния

Атрибуты таблицы:

* id – уникальный идентификатор;
* name – название аэропорта;

|  |  |
| --- | --- |
| Имя столбца | Тип данных |
| id | int |
| model | nvarchar(100) |
| business\_seats | int |
| economy\_seats | int |

Таблица 4.7 – Таблица Самолёт

Атрибуты таблицы:

* id – уникальный идентификатор;
* model – модель аэропорта;
* business\_seats – количество мест в бизнес классе;
* economy\_seats – количество мест в эконом классе;

|  |  |
| --- | --- |
| Имя столбца | Тип данных |
| id | int |
| start\_date | datetime |
| end\_date | datetime |
| airline\_id | int |
| price | money |
| code | nvarchar(50) |

Таблица 4.8 – Таблица Рейс

Атрибуты таблицы:

* id – уникальный идентификатор;
* start\_date – дата начала полёта;
* end\_date – дата конца рейса;
* airline\_id – ключ авиакомпании;
* price – цена без наценки;
* code – код рейса;

|  |  |
| --- | --- |
| Имя столбца | Тип данных |
| id | int |
| aircraft\_id | int |
| boarding\_gate | nvarchar(30) |
| start\_time | datetime |
| end\_time | datetime |
| start\_airport\_id | int |
| end\_airport\_id | id |

Таблица 4. – Таблица Маршрут

Атрибуты таблицы:

* id – уникальный идентификатор;
* aircraft\_id – ключ самолёта;
* boarding\_gate – посадочные ворота;
* start\_time – время начала полёта;
* end\_time – время конца полёта;
* start\_airport\_id – ключ начального аэропорта;
* end\_airport\_id – ключ конечного аэропорта;

|  |  |
| --- | --- |
| Имя столбца | Тип данных |
| id | int |
| route\_id | int |
| flight\_id | int |

Таблица 4. – Таблица Рейс-Маршрут

Атрибуты таблицы:

* id – уникальный идентификатор;
* route\_id – ключ рейса;
* flight\_id – ключ маршрута;

|  |  |
| --- | --- |
| Имя столбца | Тип данных |
| id | int |
| airline\_id | int |
| name | nvarchar(50) |
| hand\_luggage | bit |
| luggage | bit |
| rebooking | bit |
| refund | bit |
| markup | int |
| insurance | bit |

Таблица 4. – Таблица Тариф

Атрибуты таблицы:

* id – уникальный идентификатор;
* airline\_id – ключ авиалинии;
* name – название;
* hand\_luggage – наличие ручного багажа;
* luggage – наличие багажа;
* rebooking – перебронирование билета;
* refund – возврат билета;
* markup – наценка тарифа для цены на рейс;

|  |  |
| --- | --- |
| Имя столбца | Тип данных |
| id | int |
| email | nvarchar(100) |
| number | nvarchar(30) |

Таблица 4.12 – Таблица Контактная информация

Атрибуты таблицы:

* id – уникальный идентификатор;
* email –электронная почта;
* name – телефонный номер;

|  |  |
| --- | --- |
| Имя столбца | Тип данных |
| id | int |
| route\_id | int |
| client\_id | int |
| contact\_id | int |
| tariff\_id | int |
| seat | nvarchar(10) |
| total\_price | money |

Таблица 4.13 – Таблица Регистрация

Атрибуты таблицы:

* id – уникальный идентификатор;
* route\_id –ключ маршрута;
* client\_id – ключ клиента;
* contact\_id – почта и номер телефона для отправки билетов;
* tariff\_id –ключ тарифа;
* seat – место пассажира;
* total\_price – конечная цена билета;

5 Формулировка ограничений целостности

Целостность – это актуальность и непротиворечивость информации, соответствие имеющейся информации в базе данных внутренней логике и структуре, а также защищенность от несанкционированных изменений и разрушений.

Ограничение целостности – это некоторое утверждение или правило, которое описывает допустимые значения отдельных информационных единиц и связей между ними. Они делятся на:

\* явные;

\* неявные.

За соответствие ограничениям целостности следит СУБД в процессе своего функционирования. Она проверяет ограничения целостности во всех случаях, когда они могут быть нарушены (например, при редактировании данных).

5.1 Выявление полного перечня ограниченной целостности, присущего данной предметной области

В процессе проектирования для предметной области были выделены следующие ограничения целостности;

\* Поля номера СТС, паспорта, водительских прав, VIN кода машины имеют фиксированную длину, состоят только из цифр и не могут повторяться;

\* Поля номеров протоколов и постановлений не могут повторяться, состоят из цифр и букв латинского алфавита;

\* Поля регистрационных номеров машин могут состоять только из цифр и определённых букв, и не могут повторяться в 1 регионе;

\* Поле владельца транспортного владельца в СТС может включать либо человека, либо организацию, и не может не включать ничего.

5.2 Определение перечня ограничений целостности, которые будут контролироваться в данном проекте

В этом проекте будут учитываться ограничения целостности, связанные с ограничением максимальной длины значения, с уникальностью значений и с взаимоисключающими значениями.

5.3 Выбор способа реализация контроля целостности для каждого из ограничений

Все ограничения целостности будут реализованы с использованием стандартных функций СУБД:

\* Ограничения поля по длине достигается выбором типа данных с ограничением максимальной длины;

\* Уникальность значений достигается использованием команды UNIQUE в настройках поля таблицы, либо назначением поля первичным ключом, который не может повторяться;

\* Ситуация, когда 2 поля таблицы взаимосвязаны так, что значение в 1 поле исключает значение во 2 поле, решается использованием команды CHECK, которая будет проверять поля, и при взаимоисключающих значениях не будет заносить изменения в таблицу.

6 пРОЕКТИРОВАНИЕ ФИЗИЧЕСКОЙ СТРУКТУРЫ БАЗЫ ДАННЫХ

**Физическая модель базы данных** — это модель данных, которая определяет, каким образом представляются данные, и содержит все детали, необходимые СУБД для создания базы данных. Физическая структура определяет, тип и свойства данных, которые будут записаны в память компьютера.

Опишем схемы для основных таблиц из логической модели данных. Схема сущности будет содержать полное описание атрибутов-характеристик этой сущности и их свойств на языке целевой СУБД, с учетом связей между сущностями и ограничениями целостности.

Таблица Человек:

|  |  |  |
| --- | --- | --- |
| Поле | Тип данных | Макс. вес в байтах |
| **Код человека** | **Числовой** | **4** |
| Полное имя | Текстовый | 200 |
| Дата рождения | Дата | 3 |
| Серия и номер паспорта | Текстовый | 10 |
| Фактический адрес | Текстовый | 200 |
| Место рождения | Текстовый | 200 |
| Адрес по прописке | Текстовый | 200 |

Таблица Звание:

|  |  |  |
| --- | --- | --- |
| Поле | Тип данных | Макс. вес в байтах |
| **Код звания** | **Числовой** | **4** |
| Звание | Текстовый | 70 |

Таблица Человек-Звание:

|  |  |  |
| --- | --- | --- |
| Поле | Тип данных | Макс. вес в байтах |
| **Код записи** | **Числовой** | **4** |
| **Код человека** | **Числовой** | **4** |
| **Код звания** | **Числовой** | **4** |
| Дата получения звания | Дата | 3 |

Таблица Транспортное средство:

|  |  |  |
| --- | --- | --- |
| Поле | Тип данных | Макс. вес в байтах |
| **VIN код ТС** | **Текстовый** | **34** |
| Наименование ТС | Текстовый | 100 |
| **Код марки ТС** | **Числовой** | **4** |
| Дата изготовления | Дата | 3 |
| **Код категории вождения ТС** | **Числовой** | **4** |
| Максимальный вес | Числовой | 4 |
| Вес с завода | Числовой | 4 |
| **Код типа ТС** | **Числовой** | **4** |

Таблица Марка ТС:

|  |  |  |
| --- | --- | --- |
| Поле | Тип данных | Макс. вес в байтах |
| **Код марки тс** | **Числовой** | **4** |
| Название марки | Текстовый | 60 |
| Название фото марки | Текстовый | 60 |

Таблица Тип ТС:

|  |  |  |
| --- | --- | --- |
| Поле | Тип данных | Макс. вес в байтах |
| **Код типа тс** | **Числовой** | **4** |
| Тип ТС | Текстовый | 50 |

Таблица Категория вождения:

|  |  |  |
| --- | --- | --- |
| Поле | Тип данных | Макс. вес в байтах |
| **Код категории вождения** | **Числовой** | **4** |
| Название категории | Текстовый | 6 |

Таблица СТС:

|  |  |  |
| --- | --- | --- |
| Поле | Тип данных | Макс. вес в байтах |
| **Код СТС** | **Текстовый** | **20** |
| **Код организации** | **Числовой** | **4** |
| **Код человека** | **Числовой** | **4** |
| **VIN код ТС** | **Текстовый** | **34** |
| **Код отделения ГИБДД** | **Текстовый** | **7** |
| **Код региона** | **Текстовый** | **2** |
| Регистрационный номер ТС | Текстовый | 20 |
| Дата регистрации | Дата | 3 |

Таблица Организация:

|  |  |  |
| --- | --- | --- |
| Поле | Тип данных | Макс. вес в байтах |
| **Код организации** | **Числовой** | **4** |
| Наименование организации | Текстовый | 80 |
| Адрес организации | Текстовый | 200 |
| Название фото организации | Текстовый | 60 |

Таблица Регион:

|  |  |  |
| --- | --- | --- |
| Поле | Тип данных | Макс. вес в байтах |
| **Код региона** | **Текстовый** | **2** |
| Название региона | Текстовый | 100 |
| Дополнительный код | Текстовый | 3 |

Таблица Право на управление ТС:

|  |  |  |
| --- | --- | --- |
| Поле | Тип данных | Макс. вес в байтах |
| **Код водительских прав** | **Текстовый** | **10** |
| **Код региона** | **Текстовый** | **2** |
| **Код человека** | **Числовой** | **4** |
| **Код отделения ГИБДД** | **Текстовый** | **7** |
| Дата выдачи | Дата | 3 |
| Дата конца действия | Дата | 3 |

Таблица Отделение ГИБДД:

|  |  |  |
| --- | --- | --- |
| Поле | Тип данных | Макс. вес в байтах |
| **Код отделения ГИБДД** | **Текстовый** | **7** |
| Адрес отделения | Текстовый | 200 |
| Название отделения | Текстовый | 200 |

# Заключение

# Список использованной литературы