**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

**БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**МЕХАНИКО-МАТЕМАТИЧЕСКИЙ ФАКУЛЬТЕТ**

**Кафедра веб-технологий и компьютерного моделирования**

Сокольников

Иван Витальевич

**ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ «АВТОСАЛОН»**

Научный руководитель:

кандидат физ.-мат. наук,

доцент Ю.А. Кремень

Минск, 2025

ОГЛАВЛЕНИЕ

[ВВЕДЕНИЕ 3](#_Toc199376887)

[ГЛАВА 1. ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ «АВТОСТОЯНКА» 4](#_Toc199376888)

[Постановка задачи 4](#_Toc199376889)

[1.2. Инициирование проекта (фаза 0) 4](#_Toc199376890)

[1.3.1. Выделение множеств сущностей 5](#_Toc199376891)

[1.3.2. Описание множеств сущностей 5](#_Toc199376892)

[1.4. Построение модели уровня сущностей (фаза 2) 7](#_Toc199376893)

[1.4.1 Матрица связей 7](#_Toc199376894)

[1.4.2 Описание связей 7](#_Toc199376895)

[1.4.3 Концептуальная схема уровня сущностей 8](#_Toc199376896)

[1.5 Построение модели уровня ключей (фаза 3) 8](#_Toc199376897)

[1.5.1 Определение ключевых атрибутов и доменов 8](#_Toc199376898)

[1.5.2 Концептуальная схема 12](#_Toc199376899)

[1.6 Уточнение модели уровня ключей 13](#_Toc199376900)

[1.7. Построение полноатрибутной модели (фаза 4) 14](#_Toc199376901)

[1.7.1 Явные ограничения целостности 15](#_Toc199376902)

[ГЛАВА 2 ДАТАЛОГИЧЕСКАЯ МОДЕЛЬ. ГЕНЕРАЦИЯ БАЗЫ ДАННЫХ 17](#_Toc199376903)

[2.1 Создание базы данных 17](#_Toc199376904)

[2.2 Заполнение таблиц данными 21](#_Toc199376905)

[2.3 Проверка работы 22](#_Toc199376906)

[ЗАКЛЮЧЕНИЕ 34](#_Toc199376907)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 35](#_Toc199376908)

# ВВЕДЕНИЕ

Разработка базы данных «Автосалон» направлена на создание централизованной системы для управления ключевыми процессами автомобильного бизнеса: учетом автомобилей, поставщиков, клиентов, заявок и заказов. Актуальность проекта обусловлена необходимостью перехода от ручного учета к автоматизированным решениям, снижающим ошибки и ускоряющим операции. Цель — реализация БД, обеспечивающей ввод и редактирование данных об автомобилях (марка, модель, цвет, цена, наличие), поставщиках (название, адрес, контакты) и клиентах (ФИО, паспорт, телефон). Система автоматизирует регистрацию заявок клиентов (покупка/обмен) с отслеживанием статусов (новая/подтвержденная/отклоненная) и заказов поставщикам (новый/выполненный/отмененный).База данных также предоставляет инструменты для поиска информации (автомобили по марке, клиенты по фамилии) и формирования статистики по продажам/поставкам за заданные периоды. Решение станет основой для цифровизации автосалона, оптимизации обслуживания клиентов и повышения управляемости бизнес-процессов.

# ГЛАВА 1. ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ «АВТОСТОЯНКА»

## Постановка задачи

Целью проекта является разработка базы данных Автосалона, которая позволит хранить и обрабатывать информацию о выставке-продаже автомобилей, поставщиках, клиентах, заявках клиентов и заказах поставщикам. База данных должна обеспечивать следующие функции:

Ввод и редактирование данных о выставке-продаже автомобилей, включая номер, марку, модель, цвет, год выпуска, пробег, цену и наличие на складе.

Ввод и редактирование данных о поставщиках, включая номер, название, адрес, телефон и контактное лицо.

Ввод и редактирование данных о клиентах, включая номер, фамилию, имя, отчество, адрес, телефон и паспортные данные.

Регистрация заявок клиентов на покупку или обмен автомобилей, включая номер заявки, номер клиента, номер автомобиля, дату и время заявки и статус заявки (новая, подтвержденная, отклоненная).

Регистрация заказов поставщикам на поставку автомобилей, включая номер заказа, номер поставщика, номер автомобиля, дату и время заказа и статус заказа (новый, выполненный, отмененный).

Поиск и вывод информации по различным критериям, например: список автомобилей по заданной марке или цвету; список поставщиков по заданному названию или адресу; список клиентов по заданной фамилии или телефону; список заявок клиентов по заданному статусу или дате; список заказов поставщикам по заданному статусу или дате; статистика по продажам и обменам автомобилей за заданный период; статистика по поставкам автомобилей за заданный период.

## 1.2. Инициирование проекта (фаза 0)

Участники проекта и их функции:

Администратор базы данных:

Управление данными об автомобилях, поставщиках, клиентах.Контроль целостности и безопасности данных.

Менеджер по продажам:

Обработка заявок клиентов. Анализ наличия автомобилей на складе.

Отдел закупок:

Формирование заказов поставщикам. Контроль выполнения заказов.

Клиенты:

Подача заявок на покупку или обмен автомобилей.

Цели проектирования:

Автоматизация учета автомобилей, клиентов и поставщиков. Обеспечение оперативного доступа к данным о заявках и заказах. Формирование отчетности по продажам и поставкам.

Требования к системе:

Отображение текущего наличия автомобилей на складе (атрибут InStock в таблице Car).Регистрация заявок клиентов с возможностью фильтрации по статусу (новая, подтвержденная, отклоненная).Учет заказов поставщикам с отслеживанием статуса (новый, выполненный, отмененный).Ограничение на дублирование паспортных данных клиентов (уникальное поле Passport).Хранение истории всех операций (даты создания заявок и заказов).Определение множеств сущностей (фаза 1)

## 1.3.1. Выделение множеств сущностей

В соответствии с описанием предметной области можно выделить следующие множества сущностей:

Поставщик

Автомобиль

Клиент

Заказ

Заказ поставщику

## 1.3.2. Описание множеств сущностей

Описание выделенных множеств сущностей приведено в табл. 1

Таблица 1. Пул сущностей

|  |  |  |
| --- | --- | --- |
| Имя множества сущностей | Определение множества сущностей | Описание множества сущностей |
| Поставщик | Транспортное средство, представленное в автосалоне для продажи или обмена. | Каждый автомобиль идентифицируется уникальным номером (CarID). Характеризуется маркой (Brand), моделью (Model), цветом (Color), годом выпуска (Year), пробегом (Mileage), ценой (Price) и наличием на складе (InStock). |
| Автомобиль | Компания, осуществляющая поставку автомобилей в автосалон.. | Каждый автомобиль идентифицируется уникальным номером, который может быть номером двигателя или VIN-кодом. Кроме того, для автомобиля указывается его марка, модель, цвет, год выпуска и номерной знак, который используется для идентификации автомобиля на дороге. Как только машина впервые пользуется автостоянкой, создается новый экземпляр сущности |
| Клиент | Физическое или юридическое лицо, оформляющее заявку на покупку или обмен автомобиля. | Клиент идентифицируется уникальным номером (ClientID). Указываются ФИО (LastName, FirstName, MiddleName), адрес (Address), телефон (Phone) и паспортные данные (Passport), которые уникальны для каждого клиента. |
| Заявка | Запрос клиента на покупку или обмен конкретного автомобиля. | Заявка содержит уникальный номер (RequestID), ссылку на клиента (ClientID) и автомобиль (CarID), дату и время создания (DateTime), статус (Status: новая, подтвержденная, отклоненная) и тип операции (Type: покупка, обмен). |
| Заказ поставщику | Заказ на поставку автомобиля от поставщика в автосалон. | Заказ имеет уникальный номер (OrderID), ссылку на поставщика (SupplierID) и автомобиль (CarID), дату и время оформления (DateTime), статус выполнения (Status: новый, выполненный, отмененный). |

## 1.4. Построение модели уровня сущностей (фаза 2)

### 1.4.1 Матрица связей

На основании анализа постановки задачи устанавливаются связи между множествами сущностей. Матрице связей приведена в табл. 2.

Таблица 2. Матрица связей

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Поставщик | Автомобиль | Клиент | Заявка | Заказ |
| Поставщик |  |  |  |  | x |
| Автомобиль |  |  |  |  | x |
| Клиент |  |  |  |  | x |
| Заявка |  |  |  |  | x |
| Заказ | x | x | x | x | x |

### 1.4.2 Описание связей

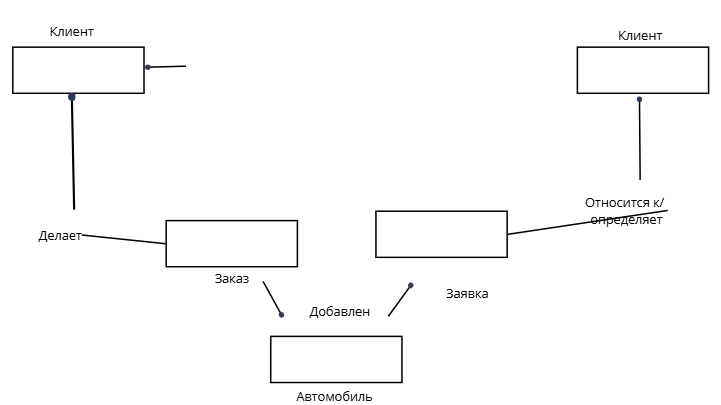
Описание выделенных связей приводится в табл. 3.

Таблица 3. Описание связей

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1-я сущность | 2-я сущность | Имя связи | Тип связи | Мощность связи | Описание связи |
| Поставщик | Учет оплаты | относится | идент. опред. | N | Каждое Поставщик  может относиться к нулю, одному и более учетам оплаты. Каждый учет оплаты может содержать только одно машино-место |
| Автомобиль | Учет оплаты | использует/учитывается | идент. опред. | N | Каждый автомобиль может использовать ноль или более учетов оплаты.  Каждый учет оплаты может учитывать только один автомобиль |
| Заказ | Учет оплаты | используется/использует | идент. опред | N | Каждый заказ может использоваться нулем и более учетами оплаты.  Каждый учет оплаты может использовать только один тариф |

### 1.4.3 Концептуальная схема уровня сущностей

В соответствии с определенными множествами сущностей и связями между ними строится концептуальная схема уровня сущностей (рис. 1).

Рис. 1. Концептуальная схема уровня сущностей

## 1.5 Построение модели уровня ключей (фаза 3)

### 1.5.1 Определение ключевых атрибутов и доменов

Определение ключевых атрибутов осуществляется одновременно с определением доменов, на которых определяются эти атрибуты. Сначала из анализа задачи для каждого множества сущностей выделяются ключи (простые или составные), позволяющие идентифицировать экземпляры сущностей. В случае если ключей несколько, один выбирается в качестве первичного ключа, а остальные определяются как альтернативные. Связи между множествами сущностей представляются с помощью внешних ключей. Таким образом, определяются все ключи. Для каждого атрибута определяется домен.

В рассматриваемом примере для выделенных сущностей можно предложить следующие ключи и домены:

Автомобиль (Car)

Первичный ключ:

CarID (тип INT, свойство IDENTITY). Уникальный числовой идентификатор, автоматически генерируемый при добавлении записи.

Домены атрибутов:

Brand (марка): NVARCHAR(50), обязательное поле. Пример: Toyota.

Model (модель): NVARCHAR(50), обязательное поле. Пример: Camry.

Color (цвет): NVARCHAR(20). Пример: Красный.

Year (год выпуска): INT. Пример: 2020.

Mileage (пробег): INT (в км). Пример: 15000.

Price (цена): DECIMAL(10,2). Пример: 25000.00.

InStock (наличие): BIT (1 — в наличии, 0 — отсутствует).

Поставщик (Supplier)

Первичный ключ:

SupplierID (тип INT, свойство IDENTITY). Уникальный идентификатор поставщика.

Альтернативный ключ:

Name (название компании): NVARCHAR(100), уникальное значение. Пример: АвтоМир.

Домены атрибутов:

Address (адрес): NVARCHAR(200). Пример: Москва, ул. Ленина, 1.

Phone (телефон): NVARCHAR(20). Пример: +7-999-123-45-67.

ContactPerson (контактное лицо): NVARCHAR(100). Пример: Иванов Иван.

Клиент (Client)

Первичный ключ:

ClientID (тип INT, свойство IDENTITY). Уникальный идентификатор клиента.

Альтернативный ключ:

Passport (паспортные данные): NVARCHAR(20), уникальное значение. Пример: 4508123456.

Домены атрибутов:

LastName (фамилия): NVARCHAR(50), обязательное поле. Пример: Иванов.

FirstName (имя): NVARCHAR(50), обязательное поле. Пример: Петр.

MiddleName (отчество): NVARCHAR(50). Пример: Сергеевич.

Address (адрес): NVARCHAR(200). Пример: Москва, ул. Пушкина, 10.

Phone (телефон): NVARCHAR(20). Пример: +7-495-111-22-33.Учет оплаты

Заявка (Request)

Первичный ключ:

RequestID (тип INT, свойство IDENTITY). Уникальный номер заявки.

Внешние ключи:

ClientID (тип INT): ссылается на Client.ClientID.

CarID (тип INT): ссылается на Car.CarID.

Домены атрибутов:

DateTime (дата создания): DATETIME, значение по умолчанию GETDATE().

Status (статус): NVARCHAR(20), допустимые значения: новая, подтвержденная, отклоненная.

Type (тип операции): NVARCHAR(20), допустимые значения: покупка, обмен.

Заказ поставщику (Order)

Первичный ключ:

OrderID (тип INT, свойство IDENTITY). Уникальный номер заказа.

Внешние ключи:

SupplierID (тип INT): ссылается на Supplier.SupplierID.

CarID (тип INT): ссылается на Car.CarID.

Домены атрибутов:

DateTime (дата заказа): DATETIME, значение по умолчанию GETDATE().

Status (статус): NVARCHAR(20), допустимые значения: новый, выполненный, отмененный.

Описание ключевых атрибутов приведено в табл. 4.

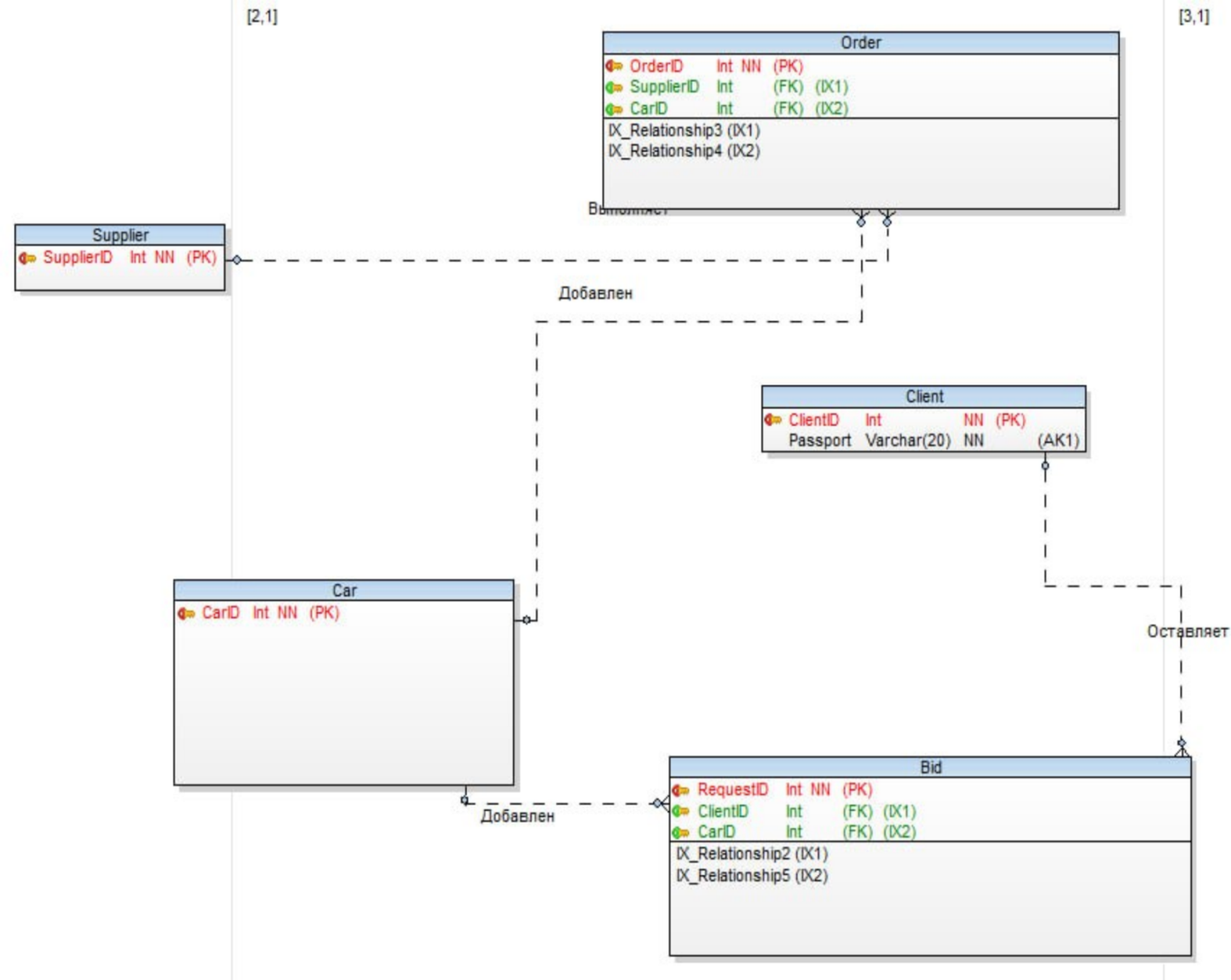
Таблица 4. Описание атрибутов

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Множество сущностей | Имя атрибута | Тип данных | Признак обязательности | Примечание |
| Заказ | Код машины | int | NOT NULL | FK |
|  | Уникальный идентификатор заказа | int |  | PK |
|  | Дата и время создания заказа | Datetime |  |  |
|  | Уникальный идентификатор поставщика | int |  | FK |
|  | Статус заказа | Char(2) |  |  |
| Клиент | Уникальный идентификатор клиента | int | NOT NULL | PK |
| Фамилия | varchar(50) | NOT NULL |  |
|  | Имя | varchar(50) | NOT NULL |  |
|  | Отчество | varchar(50) |  |  |
|  | Адрес | varchar(200) |  |  |
|  | Телефон | varchar(20) |  |  |
|  | Паспортные данные | varchar(20) | NOT NULL | AK |
| Заявка | Уникальный идентификатор заявки | int | NOT NULL | PK |
| Дата и время создания заявки | Datetime |  | AK |
|  | Статус заявки | Char(2) |  |  |
|  | Тип заявки | Char(1) |  |  |
|  | Уникальный идентификатор клиента | int |  | FK |
|  | Код машины | int |  | FK |
| Поставщик | Уникальный идентификатор поставщика | int | NOT NULL | PK |
| Название компании | varchar(100) |  |  |
| Юридический адрес | varchar(200) |  |  |
| Контактный телефон | varchar(20) |  |  |
|  | ФИО контактного лица | varchar(100) |  |  |
| Автомобиль | Код машины | Int | NOT NULL | PK |
|  | Модель | varchar(50) |  |  |
|  | Бренд | varchar(50) |  |  |
|  | Цвет | varchar(20) |  |  |
|  | Год выпуска | Date |  |  |
|  | Пробег | Int |  |  |
|  | Цена | Decimal(10,2) |  |  |
|  | Наличие | Char(1) |  |  |

### 1.5.2 Концептуальная схема

По результатам определения первичных ключей строится концептуальная схема уровня ключей (рис. 2).

Рисунок 2. Схема уровня ключей



## 1.6 Уточнение модели уровня ключей

Для выполнения дополнительных требований необходимо также добавить таблицу Размеры машино-мест

Размеры машино-мест

Первичный ключ: код размера. Значениями атрибута являются натуральные числа (порядковый номер в перечислении предметов), определенные на типе данных int со свойством IDENTITY для получения автоматически увеличивающегося идентификационного номера.

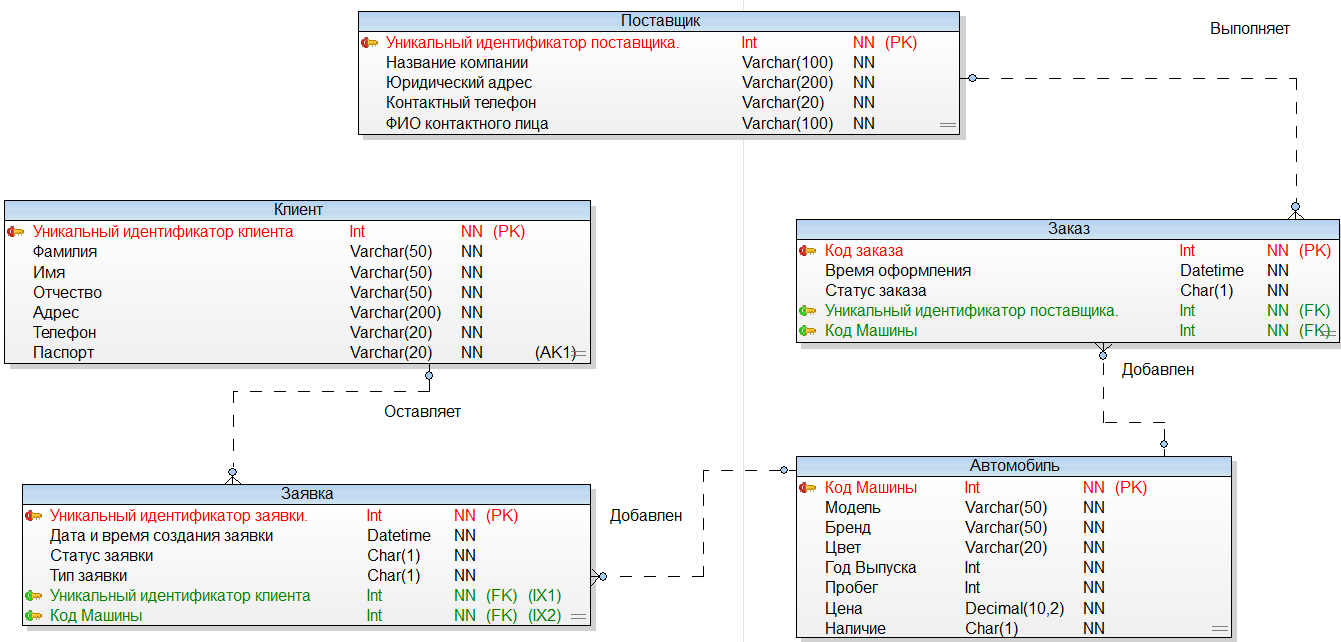
Связь сущности размер машино-места и сущности Машино-место можно реализовать с помощью добавления в машино-место дополнительного атрибута код размера, так как один размер может соответствовать нескольким машино-местам, а одно машино-место может иметь только один размер.

Таблица 5. Описание атрибутов сущности размер машино-места

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Множество сущностей | Имя атрибута | Тип данных | Признак обязательности | Примечание |
| Машино-место | Код машино-места | int | NOT NULL | PK |
| Код размера | Int | NOT NULL | FK |
| Размер машино-места | Код размера | Int | NOT NULL | PK |
| Название размера | nvarchar(64) | NOT NULL | AK |

Таким образом, уточнённая концептуальная схема уровня ключей с учётом сущности **Размеры машино-мест** представлена на рис. 3.

Рисунок 3. Уточнённая концептуальная схема



## 1.7. Построение полноатрибутной модели (фаза 4)

На этом этапе в соответствии с проведенным анализом предметной области определяются все необходимые атрибуты и домены, на которых эти атрибуты определены. Соответствующим образом модифицируются таблица описания доменов и атрибутов (табл. 6) и строится полноатрибутная концептуальная схема (рис. 4).

Таблица 6. Таблица описания доменов и атрибутов

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Множество сущностей | Имя атрибута | Тип данных | Признак обязательности | Примечание |
| Машино-место | Код машино-места | int | NOT NULL | PK |
| Стоимость аренды в сутки | money | NOT NULL |  |
| Наличие крыши | BIT | NOT NULL |  |
| Код размера | Int | NOT NULL | FK |
| Автомобиль | Код автомобиля | int | NOT NULL | PK |
| Государственный номер | varchar(16) | NOT NULL | AK |
| Бренд | Varchar(32) | NULLABLE |  |
| Модель | Varchar(32) | NULLABLE |  |
| Год выпуска | int | NULLABLE |  |
| Тариф | Код тарифа | int | NOT NULL | PK |
| Название тарифа | nvarchar(64) | NOT NULL | AK |
| Скидка | decimal(5,2) | NULLABLE |  |
| Минимальный период аренды | int | NULLABLE |  |
| Учет оплаты | Код учета оплаты | int | NOT NULL | PK |
| Код машино-места | Int | NOT NULL | FK1 |
| Код автомобиля | Int | NOT NULL | FK2 |
| Код тарифа | Int | NOT NULL | FK3 |
| Начало периода | Date | NOT NULL |  |
| Конец периода | Date | NOT NULL |  |
| Размер машино-места | Код размера | Int | NOT NULL | PK |
| Название размера | nvarchar(64) | NOT NULL | AK |

Рисунок 4. Полноатрибутная концептуальная схема

### 1.7.1 Явные ограничения целостности

Из анализа поставленной задачи можно выделить следующие дополнительные ограничения целостности:

Год выпуска автомобиля должен быть больше 1900 и меньше либо равен текущему году

Конец периода учета оплаты должен быть позже, чем начало периода

Стоимость аренды в сутки для машино-места должна быть больше нуля

Скидка (при её присутствии) должна быть больше нуля.

Минимальный период оплаты (при его присутствии) должен быть больше нуля

# ГЛАВА 2 ДАТАЛОГИЧЕСКАЯ МОДЕЛЬ. ГЕНЕРАЦИЯ БАЗЫ ДАННЫХ

## 2.1 Создание базы данных

Создадим базу данных Банк / Bank

USE master;

GO

DROP DATABASE IF EXISTS School;

GO

CREATE DATABASE School;

GO

USE School;

GO

В соответствии с приведенной структурой, внутренняя схема разрабатываемой базы данных представлена в табл. 8 –12.

Таблица 8.Автомобиль/Car

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Имя атрибута** | **Имя столбца** | **Тип данных** | **Признак обязательности** | **Ключи** |
| Код машины | CarID | int | Not null | PK |
| Бренд | Brand | Varchar(50) | Not null |  |
| Модель | Model | Varchar(50) | Not null |  |
| Цвет | Color | Varchar(20) |  |  |
| Год выпуска | Year | int |  |  |
| Пробег | Mileage | int |  |  |
| Цена | Price | Decimal(10,2) |  |  |
| Наличие | InStock | Char(1) |  |  |

Описание таблицы на SQL:

CREATE TABLE Car (

CarID INT PRIMARY KEY IDENTITY(1,1),

Brand NVARCHAR(50) NOT NULL,

Model NVARCHAR(50) NOT NULL,

Color NVARCHAR(20),

Year INT,

Mileage INT,

Price DECIMAL(10, 2),

InStock BIT

);

GO

Таблица 9.Поставщик/Supplier

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Имя атрибута** | **Имя столбца** | **Тип данных** | **Признак обязательности** | **Ключи** |
| Уникальный индентификатор поставщика | SupplierID | int | Not null | PK |
| Название компании | Name | Varchar(100) | Not null |  |
| Юридический адрес | Address | Varchar(200) | Not null |  |
| Контактный телефон | Phone | Varchar(20) |  |  |
| ФИО Контактного лица | ContactPerson | Varchar(100) |  |  |

Описание таблицы на SQL:

-- Таблица "Поставщик"

CREATE TABLE Supplier (

SupplierID INT PRIMARY KEY IDENTITY(1,1),

Name NVARCHAR(100) NOT NULL,

Address NVARCHAR(200),

Phone NVARCHAR(20),

ContactPerson NVARCHAR(100)

);

GO

Таблица 10.Клиент/Сlient

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Имя атрибута** | **Имя столбца** | **Тип данных** | **Признак обязательности** | **Ключи** |
| Уникальный индентификатор клиента | ClientID | int | Not null | PK |
| Фамилия | LastName | Varchar(50) | Not null |  |
| Имя | FirstName | Varchar(50) | Not null |  |
| Отчество | MedlleName | Varchar(50) |  |  |
| Телефон | Phone | Varchar(20) |  |  |
| Паспортные данные | Passport | Varchar(20) | Not null | AK1 |

Описание таблицы на SQL

-- Таблица "Клиент"

CREATE TABLE Client (

ClientID INT PRIMARY KEY IDENTITY(1,1),

LastName NVARCHAR(50) NOT NULL,

FirstName NVARCHAR(50) NOT NULL,

MiddleName NVARCHAR(50),

Address NVARCHAR(200),

Phone NVARCHAR(20),

Passport NVARCHAR(20) UNIQUE

);

GO

Таблица 11.Заявка/Bid

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Имя атрибута** | **Имя столбца** | **Тип данных** | **Признак обязательности** | **Ключи** |
| Уникальный индентификатор заявки | RequestID | int | Not null | PK |
| Дата и время | DateName | Dateime | Not null |  |
| Статус заявки | Status | Char(2) | Not null |  |
| Тип заявки | Type | Char(1) |  |  |
| Уникальный идентификатор клиента | ClientID | Int | Not null | FK |
| Код машины | CarID | Int | Not null | FK |

Описание таблицы на SQL

-- Таблица "Заявка"

CREATE TABLE Request (

RequestID INT PRIMARY KEY IDENTITY(1,1),

ClientID INT,

CarID INT,

DateTime DATETIME DEFAULT GETDATE(),

Status NVARCHAR(20) CHECK (Status IN ('новая', 'подтвержденная', 'отклоненная')),

Type NVARCHAR(20) CHECK (Type IN ('покупка', 'обмен')),

FOREIGN KEY (ClientID) REFERENCES Client(ClientID) ON DELETE CASCADE,

FOREIGN KEY (CarID) REFERENCES Car(CarID) ON DELETE SET NULL

);

GO

Таблица 12.Заказ/Order

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Имя атрибута** | **Имя столбца** | **Тип данных** | **Признак обязательности** | **Ключи** |
| Уникальный индентификатор  аказа | OrderID | int | Not null | PK |
| Дата и время | DateTime | Dateime | Not null |  |
| Статус заявки | Status | Char(2) | Not null |  |
| Уникальный идентификатор клиента | SupplierId | Int | Not null | FK |
| Код машины | CarID | Int | Not null | FK |

Описание таблицы на SQL

-- Таблица "Заказ поставщику"

CREATE TABLE SupplierOrder (

OrderID INT PRIMARY KEY IDENTITY(1,1),

SupplierID INT,

CarID INT,

DateTime DATETIME DEFAULT GETDATE(),

Status NVARCHAR(20) CHECK (Status IN ('новый', 'выполненный', 'отмененный')),

FOREIGN KEY (SupplierID) REFERENCES Supplier(SupplierID) ON DELETE CASCADE,

FOREIGN KEY (CarID) REFERENCES Car(CarID) ON DELETE SET NULL

);

GO

## 2.2 Заполнение таблиц данными

-- 1. Добавление автомобилей

INSERT INTO Car (Brand, Model, Color, Year, Mileage, Price, InStock)

VALUES

(N'Toyota', N'Camry', N'Красный', 2020, 15000, 25000.00, 1),

(N'BMW', N'X5', N'Черный', 2022, 5000, 60000.00, 1),

(N'Audi', N'A4', N'Белый', 2019, 30000, 28000.00, 0);

-- 2. Добавление поставщиков

INSERT INTO Supplier (Name, Address, Phone, ContactPerson)

VALUES

(N'АвтоМир', N'Москва, ул. Ленина, 1', N'+7-999-123-45-67', N'Иванов Иван'),

(N'ЕвроАвто', N'Санкт-Петербург, Невский пр., 100', N'+7-911-222-33-44', N'Петрова Мария');

-- 3. Добавление клиентов

INSERT INTO Client (LastName, FirstName, MiddleName, Address, Phone, Passport)

VALUES

(N'Иванов', N'Петр', N'Сергеевич', N'Москва, ул. Пушкина, 10', N'+7-495-111-22-33', N'4508123456'),

(N'Сидорова', N'Анна', NULL, N'Казань, ул. Гагарина, 5', N'+7-843-444-55-66', N'4510987654');

-- 4. Добавление заявок

INSERT INTO Request (ClientID, CarID, Status, Type)

VALUES

(1, 1, N'новая', N'покупка'),

(2, 3, N'подтвержденная', N'обмен');

-- 5. Добавление заказов

INSERT INTO SupplierOrder (SupplierID, CarID, Status)

VALUES

(1, 2, N'новый'),

(2, 3, N'выполненный');

## 2.3 Проверка работы

--------------------------------------------

-- Проверочные запросы:

--------------------------------------------

-- 1. Все автомобили в наличии

SELECT Brand, Model, Price

FROM Car

WHERE InStock = 1;

-- 2. Клиенты из Москвы

SELECT LastName, FirstName, Phone

FROM Client

WHERE Address LIKE N'%Москва%';

-- 3. Заявки с статусом "новая"

SELECT RequestID, DateTime, Status

FROM Request

WHERE Status = N'новая';

-- 4. Заказы, выполненные за последний месяц

SELECT OrderID, DateTime, Status

FROM SupplierOrder

WHERE Status = N'выполненный'

AND DateTime >= DATEADD(MONTH, -1, GETDATE());

-- 5. Общая стоимость автомобилей в наличии

SELECT SUM(Price) AS TotalPrice

FROM Car

WHERE InStock = 1;

-- 6. Все данные о заявках (с именем клиента и маркой авто)

SELECT

R.RequestID,

C.LastName + ' ' + C.FirstName AS ClientName,

Car.Brand + ' ' + Car.Model AS Car,

R.DateTime,

R.Status

FROM Request R

JOIN Client C ON R.ClientID = C.ClientID

JOIN Car ON R.CarID = Car.CarID;

-- 7. Поставщики и их активные заказы

SELECT

S.Name AS Supplier,

SO.DateTime AS OrderDate,

SO.Status

FROM SupplierOrder SO

JOIN Supplier S ON SO.SupplierID = S.SupplierID;

# ЗАКЛЮЧЕНИЕ

Разработка базы данных «Автосалон» успешно завершена, достигнув всех целей проекта. Реализованная система обеспечивает централизованное управление ключевыми процессами автосалона: от учета автомобилей, поставщиков и клиентов до автоматизации заявок на покупку/обмен и заказов поставщикам. База данных соответствует техническому заданию, поддерживая ввод и редактирование данных, регистрацию операций с контролем статусов, а также гибкий поиск и формирование статистики. Проверка работоспособности на тестовых данных подтвердила корректность проектных решений, включая целостность связей между сущностями и эффективность запросов. Внедрение системы устранит ручной учет данных, сократит ошибки и ускорит обработку операций, создав основу для цифровизации бизнес-процессов. Для дальнейшего развития рекомендуется расширение функционала аудита изменений и оптимизация сложных аналитических запросов. Проект готов к промышленной эксплуатации и станет инструментом повышения эффективности работы автосалона.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Ильяшенко, О. Ю. Инновационное развитие ИТ-архитектуры предприятия посредством внедрения системы бизнес-аналитики / О.Ю. Ильяшенко, И.В. Ильин, А.А. Лепехин // Наука и бизнес: пути развития. – 2017. – №8. – С. 59-66.

2. Феррари, А. Шаблоны DAX: руководство / А. Феррари, М. Руссо; перевод с английского А. Ю. Гинько. – Москва: ДМК Пресс, 2021. – 408

3. Основные сведения о DAX в Power BI Desktop [Электронный ресурс] // Microsoft – Редмонд, Вашингтон, США, 23.03.2023. – Режим доступа: https://learn.microsoft.com/ru-ru/power-bi/transform-model/desktopquickstart-learn-dax-basics. – Дата обращения: 05.10.2023.

4. Power BI [Электронный ресурс] // Википедия, свободная энциклопедия, 2018. – Режим доступа: https://ru.wikipedia.org/wiki/Power\_BI. – Дата обращения: 05.10.2023.

5. AdventureWorks sample databases [Electronic resource] // Microsoft – Redmond, Washington, U.S., 2017. –. Mode of access: https://github.com/Microsoft/sql-server-samples/releases/tag/adventureworks. – Date of access: 05.10.2023