



**Частное учреждение профессионального образования  
«Высшая школа предпринимательства»  
(ЧУПО «ВШП»)**

**КУРСОВОЙ ПРОЕКТ**

**«База данных для автосалона»**

Выполнил:

студент 3-го курса специальности  
09.02.07 «Информационные системы  
и программирование»

Кудряшов Андрей Антонович

подпись: \_\_\_\_\_

Проверил:

преподаватель дисциплины,  
преподаватель ЧУПО «ВШП»,  
к.ф.н. Ткачев П.С.

оценка: \_\_\_\_\_

подпись: \_\_\_\_\_

## Содержание

Введение.....	3
Глава 1. Основы проектирование баз данных.....	8
1.1 Проектирование баз данных.....	8
1.2 Формулировка требований к базе данных.....	9
1.3 Нормализация данных, минимизация избыточности.....	12
1.4 Запросы и работа с данными в MySQL.....	14
1.5 Базы данных в торговле автомобилями.....	15
1.6 Оптимизация баз данных для эффективной работы с данными.....	17
1.7 Выбор СУБД для реализации базы данных.....	19
1.8 Выбор между MySQL и PostGres.....	22
Глава 2.....	23
2.1 Построение схемы базы данных.....	24
2.2 ER диаграмма базы данных.....	24
2.3 Создание таблиц.....	26
2.4 Связи между таблицами.....	27
Заключение.....	29
Список литературы.....	33

## **Введение**

Актуальность данной темы в условиях стремительного развития цифровых технологий и роста конкуренции в сфере автомобильной торговли эффективное управление данными становится ключевым фактором успеха любого автосалона и не только автосалонов, но и в принципе любой промышленности и не только ее. Современный автомобильный бизнес требует учета огромного количества информации: данных о клиентах, автомобилях (новых и с пробегом), заказах, поставщиках, финансовых операциях, сервисном обслуживании и маркетинговых активностях. Ручная обработка таких массивов данных не только замедляет бизнес-процессы, но и увеличивает риск ошибок, что может привести к финансовым потерям и снижению уровня доверия клиентов. В данный момент в России выбранная тема еще более актуальная, так как из-за санкций большое количество европейских автомобилей перестали завозиться к нам в страну через европейских диллеров, поэтому большая часть автосалонов начала переживать перереформирования и изменения, что так-же затрагивает и базу данных этих салонов.

Внедрение базы данных в работу автосалона позволяет эффективно решать ключевые операционные задачи: существенно сокращать временные затраты на обработку информации, повышать достоверность данных, улучшать уровень клиентского сервиса и оптимизировать бизнес-процессы. Автоматизированная система минимизирует вероятность ошибок, неизбежных при ручном вводе информации, и значительно повышает производительность труда персонала. В современных условиях цифровизации применение информационных систем в сфере автомобильной торговли стало обязательным условием конкурентоспособности бизнеса, поскольку напрямую влияет на качество обслуживания клиентов и финансовые показатели компании. Особую актуальность автоматизация приобретает в условиях активного развития

электронной коммерции и повсеместного внедрения цифровых технологий в традиционные бизнес-модели, что делает базу данных важнейшим инструментом управления автосалоном.

### **Цель работы.**

Разработка эффективной базы данных для автосалона, обеспечивающей комплексную автоматизацию ключевых бизнес-процессов, включая управление автомобильным парком (комплектация авто, марка и.д. ), клиентскими заказами, складским учетом и финансовыми операциями.

### **Задачи работы.**

Для успешного выполнения поставленной цели будут решаться следующие задачи:

- Исследование теоретических основ проектирования баз данных.
- Исследовать существующие аналоги баз данных для автомобильных салонов и определить оптимальные подходы к проектированию.
- Разработать концептуальную модель данных (ER-диаграмму) с выделением основных сущностей и их взаимосвязей.
- Выполнить нормализацию базы данных.
- Спроектировать логическую структуру базы данных с определением таблиц, полей, первичных и внешних ключей.
- Реализовать физическую модель базы данных в выбранной СУБД (MySQL)
- Разработка SQL-запросов для обработки и анализа данных.
- Тестирование базы данных

### **Объект исследования.**

Объект исследования – процесс проектирования и реализации базы данных для автосалона, включающий:

1. Моделирование структуры данных:

- определение ключевых сущностей (автомобили, клиенты, заказы, поставщики)
- проектирование взаимосвязей между объектами

2. Разработку реляционной модели:

- процесс нормализации до оптимальной формы
- создание схемы таблиц и отношений

3. Реализацию функциональности:

- разработку SQL-запросов для основных операций
- создание хранимых процедур и триггеров
- формирование отчетных выборок

4. Оценку работоспособности:

- тестирование на реальных сценариях работы автосалона
- анализ производительности системы
- проверку целостности данных

### **Структура работы.**

Курсовая работа имеет четкую структуру, состоящую из двух содержательных глав и заключительной части. Первая глава носит теоретический характер и посвящена всестороннему анализу современных методологий проектирования баз данных с учетом их специфического применения в сфере розничной торговли автомобилями. В данном разделе особое внимание уделяется исследованию существующих решений, анализу их преимуществ и недостатков, а также выявлению оптимальных подходов к организации данных в условиях работы автосалона. Вторая глава представляет собой практическую часть исследования, в которой

подробно излагается авторский подход к созданию специализированной базы данных для автосалона. Этот раздел содержит детальное описание процесса проектирования, включая разработку концептуальной модели, нормализацию структуры данных, реализацию таблиц и установление взаимосвязей между ними. Особую ценность представляет приведенный набор SQL-запросов, демонстрирующих работу системы в реальных условиях - от ведения учета автомобилей до генерации аналитических отчетов. Заключительная часть работы систематизирует полученные результаты, формулирует обоснованные выводы о достижении поставленных целей и предлагает конкретные направления для дальнейшего совершенствования разработанной системы. В частности, рассматриваются перспективы расширения функциональности, пути оптимизации производительности и возможности интеграции с другими бизнес-приложениями, используемыми в автосалоне. Подобная структура обеспечивает логическую последовательность изложения материала - от теоретического осмысления проблемы до ее практического решения.

### **Практическая значимость работы.**

Практическая значимость работы заключается в создании эффективного инструмента для автоматизации ключевых бизнес-процессов современного автосалона. Разработанная база данных позволяет решать актуальные задачи оперативного управления предприятием, обеспечивая:

1. Оптимизацию работы с клиентской базой – систематизированный учет покупателей, истории их обращений и предпочтений способствует повышению качества сервиса и персонализации обслуживания. Автоматическое формирование напоминаний о плановом ТО увеличивает повторные продажи сервисных услуг.

2. Эффективный учет автомобильного парка – точный контроль наличия, характеристик и местоположения транспортных средств на складе и в торговом зале. Интеграция с системами маркировки позволяет отслеживать движение каждого экземпляра.
3. Автоматизацию документооборота – генерация договоров купли-продажи, актов приема-передачи и других документов по стандартизированным шаблонам сокращает временные затраты и минимизирует ошибки оформления.
4. Аналитическую поддержку управленческих решений – встроенные механизмы формирования отчетности предоставляют руководству актуальные данные о продажах, остатках, финансовых показателях и эффективности маркетинговых акций.
5. Масштабируемость и адаптивность – модульная архитектура системы позволяет постепенно расширять функционал (подключение CRM-модуля, интеграция с сайтом, мобильными приложениями), адаптируя решение под растущие потребности бизнеса.

# Глава 1. Основы проектирование баз данных

## 1.1 Проектирование баз данных

Проектирование базы данных представляет собой многоэтапный процесс создания оптимальной структуры для хранения и обработки информации. При разработке системы для автосалона необходимо учитывать специфику автомобильного бизнеса и особенности работы с клиентами, транспортными средствами и заказами.

Концептуальное проектирование начинается с глубокого анализа предметной области. На этом этапе выявляются ключевые сущности: автомобили с их уникальными характеристиками (VIN-коды, комплектации, технические параметры), клиенты, заказы, поставщики и сотрудники. Важной частью является определение атрибутов каждой сущности и установление взаимосвязей между ними. Визуализация этих отношений с помощью ER-диаграмм помогает четко представить структуру будущей базы данных.

Логическое проектирование предполагает преобразование концептуальной модели в реляционную схему. Этот процесс включает нормализацию данных для устранения избыточности и обеспечения целостности информации. Особое внимание уделяется определению первичных и внешних ключей, которые обеспечивают связи между таблицами. На данном этапе важно учитывать предполагаемые запросы к базе данных, чтобы оптимизировать ее структуру для эффективной работы.

Физическое проектирование связано с технической реализацией базы данных. Выбор системы управления базами данных (MySQL, PostgreSQL, MS SQL Server) зависит от требований к производительности,



масштабируемости и безопасности. Определение типов данных для каждого поля, создание индексов для ускорения поиска, разработка триггеров и хранимых процедур – все это составляет содержание данного этапа. Также важно заранее продумать стратегию резервного копирования и восстановления информации.

Реализация и оптимизация завершают процесс проектирования. Написание DDL-скриптов для создания структуры базы данных сопровождается разработкой эффективных SQL-запросов для выполнения основных операций. Тестирование производительности с помощью инструментов анализа запросов позволяет выявить и устранить узкие места. Особое внимание уделяется настройке системы безопасности, включая разграничение прав доступа для разных категорий пользователей.

Для автосалонов критически важна правильная организация данных о характеристиках автомобилей, включая модель, год выпуска, пробег и комплектацию. Не менее значим учет истории обслуживания и ремонтов, движения товарных запасов, взаимодействий с клиентами и финансовых операций. Грамотно спроектированная база данных обеспечивает целостность информации, высокую скорость обработки запросов и возможность интеграции с другими бизнес-системами, такими как CRM или бухгалтерские программы.

Освоение принципов проектирования баз данных особенно важно при создании специализированных решений для автомобильной сферы, где требуется обработка больших объемов структурированной информации с учетом множества взаимосвязанных параметров. Правильно реализованная система становится надежным фундаментом для автоматизации бизнес-процессов и принятия управленческих решений.

## **1.2 Формулировка требований к базе данных**

База данных должна обеспечивать:

- Учет автомобилей:
  - Хранение данных о новых и подержанных автомобилях (VIN, марка, модель, год выпуска, пробег, комплектация, цена, статус "в наличии/продан").
  - Учет технических характеристик (двигатель, коробка передач, тип привода, цвет).
  - Отслеживание истории ТО и ремонтов.
- Управление клиентами:
  - Ведение базы клиентов (ФИО, контакты, история покупок, предпочтения).
  - Фиксация обращений (тест-драйвы, заказы, сервисные запросы).
- Оформление сделок:
  - Фиксация заказов (клиент, автомобиль, дата, цена, менеджер).
  - Учет оплат и финансовых документов.
  - Интеграция с CRM и бухгалтерскими системами.
- Складской учет:
  - Контроль наличия автомобилей и запчастей.
  - Учет поставок от производителей и дилеров.
- Аналитика и отчетность:
  - Формирование отчетов по продажам, остаткам, прибыли.
  - Анализ спроса по моделям и характеристикам.

## 2. Требования к производительности

- Быстрый поиск по ключевым параметрам (марка, модель, цена, пробег) — не более 0.1 сек даже при 50+ тыс. записей.

- Оптимизированные JOIN-запросы для соединения таблиц (автомобили ↔ клиенты ↔ заказы).
- Индексация часто используемых полей (VIN, цена, статус, дата\_поступления).
- Поддержка 50+ одновременных пользователей (менеджеры, администраторы, бухгалтерия).

### 3. Требования к безопасности и надежности

- Резервное копирование (ежедневные бэкапы).
- Разграничение прав доступа:
  - Менеджеры: просмотр/добавление авто, работа с клиентами.
  - Администраторы: редактирование цен, отчеты.
  - Бухгалтерия: доступ к финансовым данным.
- Защита персональных данных (шифрование контактов клиентов).

### 4. Требования к масштабируемости

- Поддержка роста данных (10+ лет истории продаж).
- Гибкость для добавления новых полей (например, данные об электромобилях).
- Возможность интеграции с:
  1. Веб-сайтом (выгрузка каталога авто).
  2. Мобильным приложением для клиентов.
  3. 1C/ERP-системами.

### 5. Технические требования

- СУБД: MySQL 8.0+ / PostgreSQL 14+.

- Нормализация: минимум 3NF (исключение дублирования).
- Индексы для ускорения поиска (VIN, марка-модель-год, цена).
- Триггеры для автоматического обновления статусов (например, при продаже авто).

Итог: База данных должна быть производительной, безопасной и гибкой, чтобы автоматизировать ключевые процессы автосалона — от учета автомобилей до аналитики продаж.

### **1.3 Нормализация данных, минимизация избыточности**

Нормализация данных: принципы и практическое применение

Нормализация представляет собой фундаментальный процесс проектирования реляционных баз данных, направленный на устранение избыточности информации и предотвращение аномалий при операциях вставки, обновления и удаления данных. В контексте разработки базы данных для автосалона этот процесс приобретает особую значимость, учитывая сложную структуру данных о транспортных средствах, клиентах и сделках.

Основные этапы нормализации:

Первая нормальная форма (1NF) требует атомарности данных, что означает невозможность хранения составных значений в одном поле. Например, информация о дополнительном оборудовании автомобиля должна храниться отдельными записями.

Вторая нормальная форма (2NF) дополняет требования 1NF, устанавливая, что все неключевые атрибуты должны полностью зависеть от первичного ключа. В контексте автосалона это означает, например, что технические

характеристики автомобиля должны однозначно определяться его VIN-кодом.

Третья нормальная форма (3NF) устраняет транзитивные зависимости, требуя, чтобы неключевые атрибуты не зависели от других неключевых атрибутов.

Преимущества нормализации в контексте автосалона:

1. Устранение дублирования данных (например, информации о дилерах)
2. Обеспечение согласованности данных при обновлениях
3. Упрощение процедур поддержания целостности информации
4. Оптимизация использования дискового пространства
5. Повышение гибкости структуры для будущих изменений

Нормализация особенно важна для систем учета автосалонов, где требуется точное и согласованное хранение информации о большом количестве взаимосвязанных сущностей: транспортных средствах, их характеристиках, владельцах, сделках и сервисном обслуживании.

1. Устранение избыточности:
  - Удалить дублирующие поля
  - Исправить противоречия в типах данных
2. Улучшить согласованность именования:
  - Единый стиль наименования полей
  - Логичные и понятные названия таблиц и полей
3. Оптимизировать связи:
  - Упростить связи между таблицами
  - Удалить избыточные связи
4. Улучшить целостность данных:
  - Все внешние ключи имеют соответствующие ограничения

- Обязательные поля помечены как NOT NULL

Нормализованная структура обеспечивает:

- Минимальную избыточность данных
- Отсутствие аномалий при обновлении
- Эффективное использование дискового пространства
- Гибкость для будущих изменений
- Простоту поддержки и расширения

## **1.4 Запросы и работа с данными в MySQL**

Основное назначение базы данных заключается в обеспечении эффективного выполнения операций с данными, включая:

1. Оперативную обработку запросов - быстрый поиск, фильтрацию и выборку информации с минимальным временем отклика
2. Эффективное управление данными - надежное выполнение операций вставки, обновления и удаления записей
3. Оптимизацию работы с информацией:
  - Минимизацию времени доступа к данным
  - Снижение нагрузки на систему при обработке сложных запросов
  - Обеспечение стабильной производительности при росте объема данных
4. Поддержку целостности данных - автоматическое соблюдение бизнес-правил и ограничений при всех операциях

Для автосалонов это означает возможность:

- Мгновенно получать информацию о наличии автомобилей с нужными характеристиками
- Быстро формировать договоры и сопроводительные документы

- Оперативно обновлять данные о поставках и продажах
- Анализировать показатели работы в реальном времени

Таким образом, база данных служит высокопроизводительным инструментом для работы с информацией, обеспечивающим скорость, надежность и удобство доступа к данным при любых операциях.

К примеру мы можем вывести Вывести модель автомобиля с самой высокой базовой ценой парами строчками кода:

```
SELECT id, name, base_price  
FROM car_models  
ORDER BY base_price DESC  
LIMIT 1;
```

## 1.5 Базы данных в торговле автомобилями

Базы данных играют центральную роль в современной автомобильной торговле, обеспечивая комплексное управление всеми бизнес-процессами автосалона. Они служат технологическим фундаментом для эффективной работы с информацией и принятия управленческих решений.

В автомобильном бизнесе базы данных выполняют несколько критически важных функций. Прежде всего, они обеспечивают полный учет автомобильного парка, включая хранение технических характеристик, данных о комплектациях, текущем статусе транспортных средств и истории их обслуживания. Это позволяет оперативно получать актуальную информацию о каждом автомобиле в наличии.

Не менее важна функция клиентского менеджмента. База данных хранит полную историю взаимодействий с каждым клиентом, фиксирует его предпочтения и предыдущие покупки, что позволяет выстраивать персонализированный подход и повышать качество обслуживания. Система автоматизирует маркетинговые коммуникации, напоминания о техническом обслуживании и другие сервисные функции.

В сфере управления продажами база данных обеспечивает оформление сделок, генерацию договоров и сопроводительной документации. Она интегрируется с CRM-системами и бухгалтерскими программами, создавая единое информационное пространство для всех подразделений автосалона.

Аналитический модуль предоставляет руководству инструменты для мониторинга ключевых показателей эффективности, анализа спроса по различным параметрам и прогнозирования продаж. Это позволяет принимать обоснованные управленческие решения и оперативно реагировать на изменения рыночной ситуации.

С технической точки зрения современные базы данных для автосалонов отличаются высокой производительностью при работе с большими объемами информации, обеспечивают надежное хранение данных и их защиту от несанкционированного доступа. Они поддерживают многопользовательскую работу с разграничением прав доступа и возможность интеграции с веб-порталами и мобильными приложениями. Внедрение специализированной базы данных приносит автосалону существенные бизнес-преимущества. Практика показывает сокращение времени обработки заказов на 30-40%, значительное уменьшение количества ошибок в учете и повышение конверсии за счет персонализированного подхода к клиентам. Оптимизация складских



запасов и автоматизация рутинных операций позволяют снизить издержки и повысить общую эффективность бизнеса.

Современные системы управления базами данных для автомобильной торговли включают специализированные функции для работы с каталогами запчастей, организации тест-драйвов, планирования сервисного обслуживания и проведения trade-in операций. Они становятся ключевым элементом цифровой инфраструктуры автосалона, обеспечивающим конкурентные преимущества в условиях динамично развивающегося рынка.

## **1.6 Оптимизация баз данных для эффективной работы с данными**

После этапов проектирования и нормализации базы данных наступает не менее важный этап - оптимизация производительности системы. Особое значение это приобретает при работе с большими объемами данных, характерными для современных автосалонов. Грамотная оптимизация позволяет значительно ускорить выполнение запросов и повысить общую эффективность работы с информацией. Основные направления оптимизации включают несколько ключевых аспектов. Первый и наиболее значимый - правильное индексирование данных. Индексы служат своеобразными указателями, которые позволяют СУБД быстро находить нужные записи без полного сканирования таблиц. В автомобильной тематике особенно важно создавать индексы для часто запрашиваемых полей, таких как модель автомобиля, VIN-код или дата поступления. Однако следует помнить, что чрезмерное индексирование может замедлить операции вставки и обновления данных, поэтому важно находить разумный баланс.

Важным инструментом анализа производительности является команда EXPLAIN, доступная в большинстве современных СУБД. Она позволяет детально изучить план выполнения запроса, понять, какие индексы используются, в каком порядке соединяются таблицы, и выявить потенциальные узкие места. Это особенно актуально для сложных запросов, включающих соединения нескольких таблиц - например, при выборке автомобилей определенной марки с заданными характеристиками.

Особое внимание при оптимизации следует уделять работе с соединениями таблиц (JOIN-операциям). В контексте автосалона это могут быть запросы, объединяющие информацию об автомобилях, производителях, комплектациях и текущем наличии. Правильный выбор типа соединения и порядка таблиц в запросе может значительно повлиять на его скорость выполнения. Для аналитических задач и генерации отчетов незаменимы агрегатные функции SQL. Они позволяют быстро получать сводные данные - средние цены по моделям, количество автомобилей определенного класса, динамику продаж по периодам. Такие запросы особенно востребованы для анализа эффективности работы автосалона и принятия управленческих решений.

Дополнительные методы оптимизации включают:

- Партиционирование крупных таблиц по определенным критериям
- Регулярное обслуживание базы данных (перестроение индексов, обновление статистики)
- Настройку параметров сервера СУБД под конкретную нагрузку
- Использование кэширования часто запрашиваемых данных
- Оптимизацию структуры хранения данных

Комплексное применение этих методов позволяет создать высокопроизводительную систему, способную эффективно работать даже при интенсивной нагрузке, что особенно важно для современных автосалонов с их большими объемами данных и высокими требованиями к скорости обработки информации. Грамотная оптимизация превращает базу данных из простого хранилища информации в мощный инструмент поддержки бизнес-процессов.

Торговля предполагает работу с большими таблицами. Для быстрого нахождения данных можно использовать оператор JOIN.

JOIN - это оператор SQL, который позволяет объединять данные из двух или более таблиц на основе связанных между ними столбцов. Он используется для выборки информации, которая распределена по разным таблицам, но логически связана (например, заказы и клиенты, товары и категории).

## **1.7 Выбор СУБД для реализации базы данных**

MySQL — это одна из самых популярных реляционных СУБД с открытым исходным кодом, которая хорошо подходит для многих сценариев использования. Вот подробные причины, почему MySQL часто выбирают для реализации баз данных:

### **1. Надежность и стабильность**

1. MySQL разрабатывается с 1995 года и принадлежит Oracle, что гарантирует высокую надежность.
2. Широко используется в крупных проектах (Facebook, Twitter, YouTube, Wikipedia), что подтверждает его стабильность.

3. Поддерживает **ACID** (Atomicity, Consistency, Isolation, Durability) в InnoDB, что критично для транзакционных систем.

## 2. Производительность

1. Оптимизирован для чтения: Быстро выполняет SELECT-запросы благодаря эффективным индексам (B-деревья, hash-индексы в Memory Engine).

2. Кэширование запросов: MySQL кэширует часто выполняемые запросы, ускоряя работу.

3. Поддержка партиционирования: Позволяет разбивать большие таблицы на части для ускорения выборки.

4. Гибкость движков хранения:

5. InnoDB – для транзакций и высокой нагрузки.

6. MyISAM – для быстрого чтения (но без транзакций).

7. Memory – хранение в RAM для сверхбыстрого доступа.

## 3. Масштабируемость

1. Вертикальное масштабирование: MySQL эффективно использует ресурсы сервера (многопоточность, оптимизация запросов).

2. Горизонтальное масштабирование:

3. Репликация (Master-Slave, Master-Master) для распределения нагрузки.

4. Шардинг (разделение данных между серверами) через приложение или инструменты вроде Vitess.

5. Поддержка кластерных решений (MySQL Cluster, Galera Cluster).

## 4. Простота использования и администрирования

- Дружелюбный SQL-синтаксис, близкий к стандарту ANSI SQL.
- Широкие инструменты управления:

phpMyAdmin, MySQL Workbench, DBeaver.

Интеграция с языками программирования (Python, PHP, Java, Node.js).

- Легкость развертывания:

Есть облачные версии (Amazon RDS, Google Cloud SQL).

Подходит для небольших проектов и enterprise-решений.

## **5. Безопасность**

1. Гибкая система прав доступа (GRANT/REVOKE, ролевая модель в MySQL 8.0+).
2. Поддержка SSL/TLS для шифрования соединений.
3. Аудит действий (Enterprise Edition + плагины).
4. Защита от SQL-инъекций (подготовленные запросы, экранирование данных).

## **6. Поддержка JSON и NoSQL-функций**

1. С MySQL 5.7+ можно хранить и обрабатывать JSON (функции JSON\_EXTRACT, JSON\_SEARCH).
2. Поддержка документной модели (NoSQL-интерфейс через X DevAPI).

## **7. Экономическая эффективность**

1. Бесплатен для большинства случаев (Community Edition).
2. Низкие требования к ресурсам (работает даже на слабых серверах).
3. Большое комьюнити → дешевая поддержка.

## **8. Кроссплатформенность**

1. Работает на Windows, Linux, macOS.
2. Поддерживается в облаках (AWS, Azure, GCP).

3. Есть встроенные решения для Docker и Kubernetes.

## **9. Резервное копирование и восстановление**

1. Встроенные утилиты (mysqldump, mysqlbackup).
2. Поддержка point-in-time recovery (бинарные логи).
3. Интеграция с Percona XtraBackup для горячего бэкапа.

## **10. Гибкость в проектировании БД**

1. Поддержка внешних ключей, триггеров, хранимых процедур.
2. Возможность кастомизации (плагины, пользовательские функции).

### **1.8 Выбор между MySQL и PostGres**

зависит от конкретных требований проекта. Обе системы управления базами данных обладают значительными возможностями, но имеют разные сильные стороны.

MySQL часто оказывается предпочтительнее в определенных сценариях благодаря своим характеристикам. В высоконагруженных OLTP-системах с короткими транзакциями MySQL с движком InnoDB демонстрирует более высокую производительность на простых операциях выборки, вставки и обновления данных. Это достигается за счет оптимизированного кэширования, менее строгих проверок целостности при необходимости и более простого планировщика запросов.

Система проще в настройке и администрировании по сравнению с PostgreSQL. MySQL требует меньше начальной конфигурации, стабильно работает даже на серверах с ограниченными ресурсами и не нуждается в таких процедурах обслуживания, как vacuum в PostgreSQL. Это делает его удобным выбором для небольших проектов и разработчиков без глубоких знаний администрирования баз данных.

В вопросах масштабирования MySQL предлагает более простые решения для репликации и шардинга. Нативная асинхронная репликация Master-Slave работает стабильно, а для горизонтального масштабирования существуют готовые инструменты вроде Vitess и ProxySQL. MySQL также лучше интегрирован с основными облачными платформами, такими как Amazon RDS, Google Cloud SQL и Azure Database, где доступны функции автоматического масштабирования.

Система более экономично использует ресурсы сервера, особенно на конфигурациях с ограниченной памятью (1-2 ГБ RAM). InnoDB эффективно управляет оперативной памятью, а общие накладные расходы на обслуживание ниже по сравнению с PostgreSQL.

MySQL остается стандартом де-факто для многих популярных CMS и фреймворков, включая WordPress, Joomla, Drupal, Magento, Laravel и Symfony. Хотя PostgreSQL также поддерживается этими системами, его настройка обычно требует больше усилий.

Начиная с версии 8.0, MySQL предлагает улучшенную поддержку работы с JSON-данными, позволяя создавать индексы по JSON-полям и выполнять запросы с использованием специального SQL-синтаксиса. Хотя PostgreSQL по-прежнему мощнее в этой области, MySQL предоставляет достаточные возможности для многих гибридных сценариев.

PostgreSQL становится предпочтительным выбором, когда требуются сложные аналитические запросы с оконными функциями и CTE, работа с геоданными через PostGIS, максимальная ACID-совместимость, поддержка кастомных типов данных или высокая параллельная нагрузка на запись.

## **Глава 2.**

### **2.1 Построение схемы базы данных**

Перед началом работы был выбран официальный инструмент MySQL WorkBench.

MySQL Workbench представляет собой официальную интегрированную среду для разработки и администрирования баз данных MySQL. Этот инструмент предоставляет комплексный набор функций для эффективной работы с данными. И предоставляет нам множество возможностей: масштабируемость, универсальность управления, оптимизированная производительность, кроссплатформенная поддержка, бесплатная лицензия, гибкость хранения, надежная защита.

Данный инструмент сочетает в себе удобный графический интерфейс с мощными функциональными возможностями, что делает его одним из наиболее популярных решений для работы с MySQL.

## **2.2 ER диаграмма базы данных**

Для начало, создадим макет моей базы данных с помощью ER диаграммы.

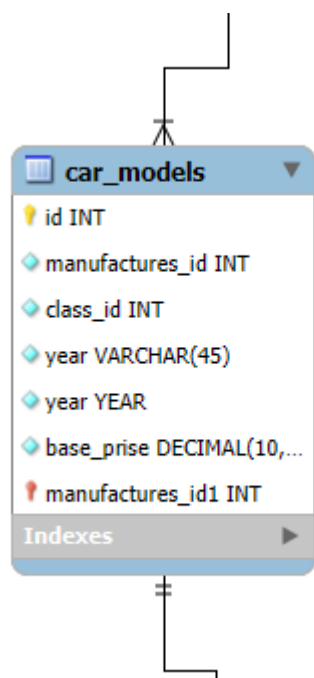
ER-диаграмма (Entity-Relationship Diagram, диаграмма "сущность-связь") — это графическая модель, которая описывает структуру базы данных, отображая сущности, их атрибуты и связи между ними. Она используется на этапе проектирования БД для визуализации данных и их взаимосвязей.

в MySQL ER диаграмма создается сразу в программе и создает уже макет базы данных со связями между таблицами. Главная сложность расставить правильные связи между таблицами для корректной работы базы данных.

## **2.3 Создание таблиц**

1. Создаем таблицу `car_models` и заполняем ее нужными сущностями.



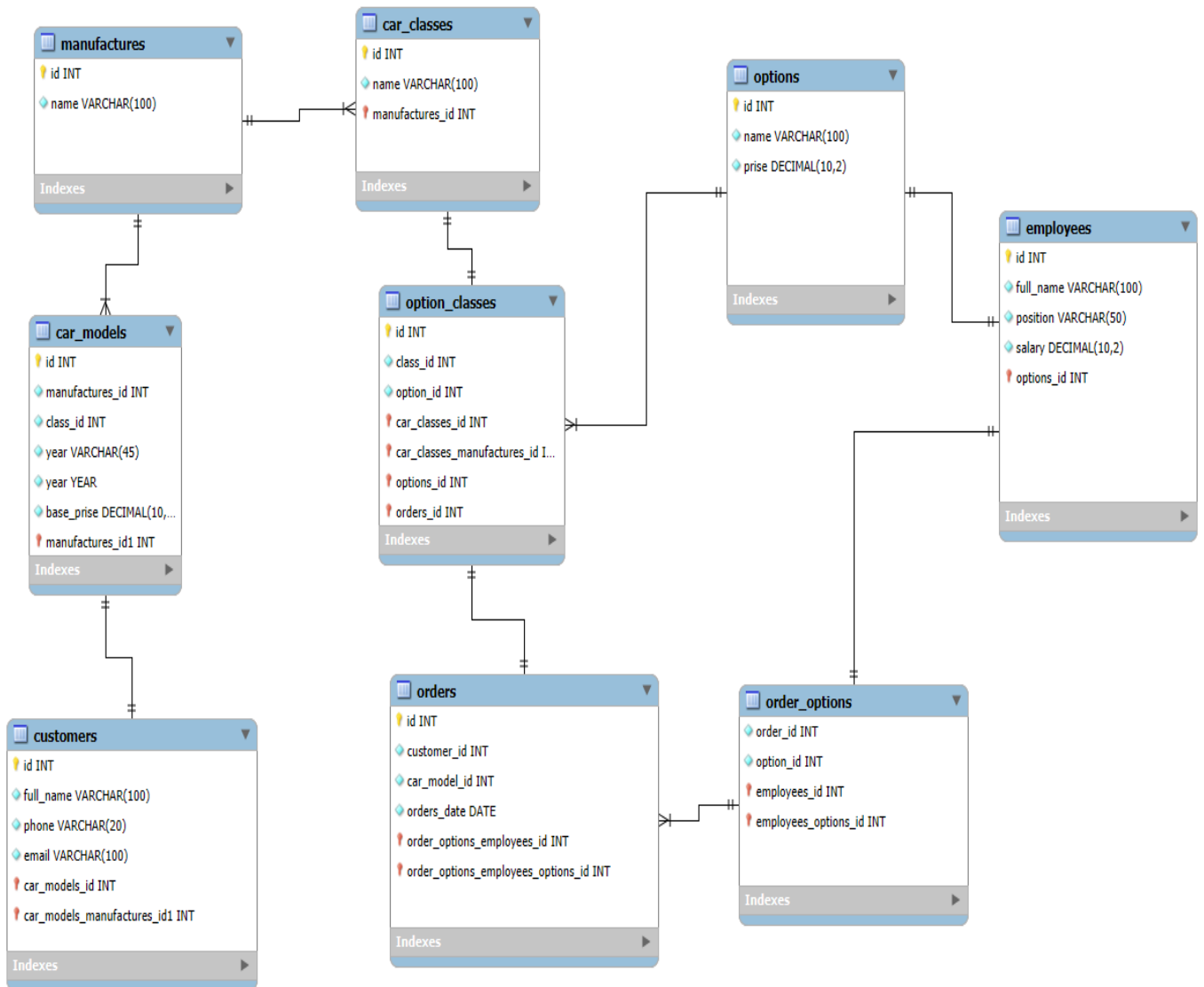


Пример кода для создания данной таблицы

```
CREATE TABLE IF NOT EXISTS `mydb`.`car_models` (  
  `id` INT UNSIGNED NOT NULL AUTO_INCREMENT,  
  `manufactures_id` INT UNSIGNED NOT NULL,  
  `class_id` INT UNSIGNED NOT NULL,  
  `year` VARCHAR(45) NOT NULL,  
  `year` YEAR NOT NULL,  
  `base_prise` DECIMAL(10,2) UNSIGNED NOT NULL,  
  `manufactures_id1` INT UNSIGNED NOT NULL,  
  PRIMARY KEY (`id`, `manufactures_id1`),  
  INDEX `fk_car_models_manufactures1_idx` (`manufactures_id1` ASC) VISIBLE,  
  CONSTRAINT `fk_car_models_manufactures1`  
    FOREIGN KEY (`manufactures_id1`)  
      REFERENCES `mydb`.`manufactures` (`id`)  
      ON DELETE NO ACTION  
      ON UPDATE NO ACTION)  
ENGINE = InnoDB
```

Так же по аналогии создаем и другие таблицы, с другими данными, которые нам потребуются для создания базы данных для автосалона.

Так выглядит готовая ER диаграмма со связями



## 2.4 Связи между таблицами

### 1. Связь "один к одному" (1:1)

Характеристика:

- Каждой записи в таблице А соответствует ровно одна запись в таблице В
- Обратное также верно: запись в В относится только к одной записи в А

### 2. Связь "один ко многим" (1:N)

### **Характеристика:**

- Одна запись в таблице А может ссылаться на множество записей в таблице В
- Каждая запись в В относится только к одной записи в А

### 3. Связь "многие ко многим" (N:M)

#### **Характеристика:**

- Одна запись в А может относиться ко многим записям в В
- Одна запись в В может относиться ко многим записям в А

### Реализация бизнес процессов на уровне СУБД

Для наглядности стоит огласить перечень инструментов MYSQL, подходящий для реализации бизнес процессов в искомой базе данных:

#### 1. Стандартные запросы:

##### 1.1 Стандартный запрос

SELECT выбирает отдельные столбцы или всю таблицу полностью.

FROM указывает, из какой таблицы получить данные.

WHERE - условие, по которому SQL выбирает данные.

GROUP BY столбец, по которому данные будут группироваться.

HAVING условие, по которому сгруппированные данные будут отфильтрованы.

ORDER BY столбец, по которому данные будут отсортированы.

2. Триггеры в MySQL представляют собой специальные хранимые процедуры, которые автоматически выполняются при наступлении определенных событий в базе данных. Эти события могут быть связаны с операциями изменения данных: вставкой (INSERT), обновлением (UPDATE) или удалением (DELETE) записей в таблице.

Основные характеристики триггеров:

1. Автоматическое выполнение - триггеры срабатывают без явного вызова пользователем
2. Привязка к таблице - каждый триггер связан с конкретной таблицей
3. Временные точки срабатывания:
  - a. BEFORE (перед выполнением операции)
  - b. AFTER (после выполнения операции)
4. Управление данными - могут проверять и модифицировать данные перед их сохранением
5. Логирование - часто используются для записи изменений в журналы

3. Представления:

Представление — это виртуальная таблица, которая не хранит данные физически, а представляет собой сохранённый SQL-запрос. Когда вы обращаетесь к представлению, MySQL выполняет этот запрос и возвращает результат.

## **Заключение**

MySQL — это проверенная временем, стабильная и производительная СУБД, которая хорошо подходит для большинства проектов. Благодаря простоте настройки, хорошей документации и широкому распространению, она остается одним из лучших выборов для разработчиков.

## **Цель работы**

Цель работы - создать рабочую базу данных для автосалон.

## **Список литературы**

1. Википедия (Электронный ресурс)

<https://ru.wikipedia.org/wiki/>

2.Кафедра It VSHP (Электронный ресурс)

<https://it.vshp.online/>

3.Skillbox (Электронный ресурс)

<https://skillbox.ru/>

4.Skypro (Электронный ресурс)

<https://sky.pro/wiki/sql/rabota-s-bazami-dannyh-mysql-rukovodstvo-dlya-nachinayushih/>

5.You Tube (Электронный ресурс)

<https://www.youtube.com/?app=desktop&hl=ru>