Министерство образования Республики Беларусь

Учреждение образования «Белорусский государственный университет   
информатики и радиоэлектроники»

Факультет компьютерного проектирования

Кафедра проектирования информационно-компьютерных систем

Дисциплина «Структуры и базы данных»

|  |  |
| --- | --- |
|  | «К ЗАЩИТЕ ДОПУСТИТЬ» |
|  | Руководитель курсового проекта  Магистр техники и технологии, ассистент  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.Д. Сыс |
|  | \_\_\_.\_\_\_\_.2021 |

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

к курсовому проекту

на тему:

**«БАЗА ДАННЫХ ДЛЯ ПОДДЕРЖКИ РАБОТЫ автомагазина»**

БГУИР КП 1-39 03 02 004 ПЗ

|  |  |
| --- | --- |
|  | Выполнил студент группы 913801  Гончар Виталий Витальевич  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись студента) |
|  | Курсовой проект представлен на проверку \_\_\_.\_\_\_\_.2021  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись студента) |

Минск 2021

# СОДЕРЖАНИЕ

[Введение 4](#_Toc72443465)

[1 Анализ исходных данных для проектирования сети 5](#_Toc72443466)

[2 Обоснование топологии компьютерной сети и структуры информационных ресурсов 9](#_Toc72443467)

[3 Выбор и обоснование активного и пассивного сетевого оборудования 15](#_Toc72443468)

[4 Адресация, маршрутизация в проектируемой сети 25](#_Toc72443469)

[5 Расчёт основных числовых характеристик проектируемой сети 32](#_Toc72443470)

[5.1 Поиск и расчёт скорости узких мест 32](#_Toc72443471)

[5.2 Расчёт портовой ёмкости коммутаторов 33](#_Toc72443472)

[5.3 Расчёт мощности сетевых устройств 33](#_Toc72443473)

[6 Мероприятия по обеспечению управления, отказоустойчивости и безопасности сети 34](#_Toc72443474)

[7 Моделирование проектируемой сети в cisco packet tracer 37](#_Toc72443475)

[Заключение 39](#_Toc72443476)

[Список используемых источников 40](#_Toc72443477)

[Приложение А 41](#_Toc72443478)

[Приложение Б 42](#_Toc72443481)

**РЕФЕРАТ**

БГУИР КП 1-39 03 02 006 ПЗ

**Назаренко, Д.И.** База данных для поддержки работы страховой компании : пояснительная записка к курсовому проекту / Д.И. Назаренко. – Минск: БГУИР, 2020. – 34 с.

Пояснительная записка 34 с., 14 рис., 8 источника, 4 приложения, 5 графических материалов.

АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ И ЕЁ ФОРМАЛИЗАЦИИ ДЛЯ ПРОЕКТИРОВАНИЯ БАЗЫ ДАННЫХ, ПРОЕКТИРОВАНИЯ БАЗЫ ДАННЫХ ДЛЯ ОСНОВНОГО ВИДА ДЕЯТЕЛЬНОСТИ РАССМАТРИВАЕОМЙ ОБЛАСТИ, ПРИМЕНЕНИЕ РАЗРАБОТАННОЙ БАЗЫ ДАННЫХ

*Цель* *проектирования*: разработка Android-приложения, проектирование эффективной и безопасной базы данных для поддержания работы страховой компании.

*Методология проведения работы*: В процессе решения поставленных задач использованы принципы системного подхода, теория проектирования базы данных, методы проектирования Android-приложения.

*Результаты работы*: изучены способы хранения информации в базе данных, проведен анализ реляционной модели данных, разработано Android-приложение для страховой компании.

*Область применения результатов*: разработанная база данных и Android-приложение в будущем, возможно применить для создания полноценного коммерческого приложения для обеспечения комфорта работы страховой компании.

# ВВЕДЕНИЕ

//да

# 1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ И ЕЕ ФОРМАЛИЗАЦИЯ ДЛЯ ПРОЕКТИРОВАНИЯ БАЗЫ ДАННЫХ

Проектирование базы данных – это набор процессов, которые облегчают проектирование, разработку, внедрение и обслуживание систем управления корпоративными данными. Правильно спроектированная база данных проста в обслуживании, улучшает согласованность данных и экономична с точки зрения дискового пространства для хранения. Разработчик базы данных решает, как элементы данных соотносятся и какие данные должны быть сохранены.

Основными задачами проектирования баз данных в СУБД являются создание логических и физических моделей дизайна предлагаемой системы баз данных. Логическая модель концентрируется на требованиях к данным и данных, которые должны храниться независимо от физических соображений. Его не волнует, как данные будут храниться или где они будут храниться физически. Модель проектирования физических данных включает в себя перевод логического проекта БД базы данных на физический носитель с использованием аппаратных ресурсов и программных систем, таких как системы управления базами данных (СУБД).

* 1. **Описание предметной области**

Из определения следует, что автомагазин – это бизнес, который продает новые или подержанные автомобили на уровне розничной торговли на основании дилерского договора с автопроизводителем или его дочерней компанией по сбыту. Автомагазин, как правило, имеет в своём штате несколько десятков сотрудников, которые разделены на отделы, а именно: отдел продаж, отдел маркетинга, отдел покупок и так далее. Автомагазин также может предоставлять услуги по техническому обслуживанию автомобилей и нанимать автомобильных специалистов (механиков) для хранения и продажи запасных частей для автомобилей и обработки претензий по гарантии.

Тема продаж и покупок автомобилей очень популярна в наше время, так как сам предмет – автомобиль пользуется большой популярностью у населения. Для облегчения поиска нового автомобиля во многих городах организуются специальные магазины, называемые автомагазином (автосалоном), в которых клиенты могут подобрать себе автомобиль по вкусу. Как известно, новые автомобили поставляются в разных конфигурациях. В таком случае, автомобиль стандартной комплектации отличается от автомобиля максимальной комплектации присутствием той или иной функции или прибора. Поэтому, доукомплектование автомобиля по желанию покупателя будет осуществляться путём выбора наличия той или иной детали автомобиля.

Формализованная организация учета реализации автомобилей в автомагазине предполагает следующие бизнес-процессы:

1 Поступление заказа на какой-либо автомобиль со списком необходимых к установке деталей;

2 Заказ автомобиля автомагазином у компании-поставщика;

3 Покупка деталей у компаний-поставщиков, которые будут устанавливаться в автомобиль согласно заказу;

4 Прием автомобиля – принятие автомобиля на внутренний учет, проведение подготовки, переоборудования и диагностики автомобиля, оповещение покупателя;

5 Реализация автомобиля – осмотр автомобиля покупателем, оформление договора купли-продажи.

Таким образом был описан построен процесс работы автомагазина, однако стоит понимать, что это – упрощённый вариант.

**1.2 Анализ информационных потребностей пользователей и предварительное описание запросов**

Приложение разрабатывается для нескольких групп пользователей, у каждой из которых свой уровень доступа. Пользователь «Администратор» может просматривать все данные (таблицы), которые есть в базе данных, а также изменять, добавлять и удалять поля во всех таблицах. Единственное, чего не может администратор – это добавление новых пользователей. Для этих целей служит окно регистрации. Таким образом достигается конфиденциальность данных для каждого пользователя, который хранится в базе данных.

Стандартный пользователь, которым в 99% случаев будет являться посетитель, имеет ограниченный круг возможностей по взаимодействию с базой данных. Что касается создания и удаления записей, то пользователь может делать вышеперечисленные действия только по отношению к таблице «Заказы». С остальными таблицами пользователь не имеет никаких прямых связей. Просмотр списка товаров (автомобилей, в данном случае) происходит через операцию просмотра представления, которое представительно настроено в базе данных.

**1.3 Определение требований и ограничений к базе данных с точки зрения предметной области**

В соответствии с предметной областью система строится с учётом

следующих особенностей:

– в автомагазине может быть любое количество автомобилей;

– каждый автомобиль имеет уникальный 17-значный номер, который называется VIN;

– каждый автомобиль имеет свою характеристику;

– стоимость автомобиля и стоимость деталей – положительные числа, с не более чем двумя значениями после запятой;

– покупатель может выбрать несколько автомобилей, но на каждый автомобиль оформляется отдельный заказ.

– при создании заказа автомобиль и работник магазина, указанный в заказе, приобретают статус «занят»;

– при подтверждении заказа автомобиль приобретает статус «продан» и больше не показывается в списке, предоставляемом потенциальным покупателям, однако всё еще числится в базе, а работник становится свободным для следующих заказов;

– при удалении заказа автомобиль и работник, указанные в заказе, приобретают статус «свободный».

В соответствии с вышеперечисленными ограничениями будут разрабатываться как база данных, так и программное средство для работы с базой данных.

Качество проектирования базы данных влияет на работу с этой базой данных в будущем. С хорошо спроектированной базой данных легче работать, легче писать запросы.

Таким образом, были сформированы следующие таблицы:

– таблица «userrrs»;

– таблица «orders»;

– таблица «employees»;

– таблица «qualifications»;

– таблица «payment\_types»;

– таблица «order\_types»;

– таблица «vehicle\_models»;

– таблица «manufacturers»;

– таблица «vehicle\_types»;

– таблица «engines»;

– таблица «turbos»;

– таблица «transmissions»;

– таблица «wheels»;

Данного набора сущностей вполне хватит для нормального функционирования автомагазина и взаимодействия пользователей с автомагазином.

**1.4 Постановка решаемой задачи**

Таким образом, в качестве входных данных в данном курсовом проектировании имеются: необходимость в проектировании базы данных, которая будет проста в обслуживании, будет иметь согласованные данные и экономична с точки зрения дискового пространства для хранения. Вместе с этим, база данных должна быть надёжной, обеспечивать целостность данных, содержать не менее 10 сущностей, а также должна быть приведена к третьей нормальной форме. Чтобы обеспечить правильное взаимодействие пользователя с ассортиментом автомагазина, в базе данных будут реализованы так называемые триггеры: например, когда пользователь закажет автомобиль, ему автоматически присвоится статус «занят», поэтому другой пользователь не сможет заказать его до тех пор, пока первый пользователь по какой-либо причине не отменит заказ. Целостность и безопасность данных в базе данных будут реализованы через систему прав доступа, разграничения ролей, а также через использование представлений.

# ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ ДЛЯ ОСНОВНОГО ВИДА ДЕЯТЕЛЬНОСТИ РАССМАТРИВАЕМОЙ ПРЕДМЕТНОЙ ОБЛАСТИ

Правильно спроектированная база данных предоставляет доступ к актуальной и точной информации. Поскольку правильное проектирование важно для достижения целей автомагазина при работе с базой данных, имеет смысл вкладывать время, необходимое для изучения принципов хорошего проектирования.

# Разработка инфологической модели предметной области базы данных

Инфологическая модель применяется на втором этапе проектирования базы данных, когда уже сделано словесное описание предметной области, то есть после этапа постановки задачи.

Проблема представления семантики давно интересовала разработчиков, и в семидесятых годах было предложено несколько моделей данных, названных семантическими моделями. К ним можно отнести модель функциональную модель данных Шипмана, семантическую модель данных, предложенную Хаммером и Мак-Леоном в 1981 году, модель «сущность—связь», предложенную Ченом в 1976 году и ряд других моделей. У всех моделей были свои положительные и отрицательные стороны, но испытание временем выдержала только последняя. И в настоящий момент именно модель Чена «сущность – связь», или «Entity Relationship», стала фактическим стандартом при инфологическом моделировании баз данных. Общепринятым стало сокращенное название ER-модель, большинство современных CASE-средств содержат инструментальные средства для описания данных в формализме этой модели. Кроме того, разработаны методы автоматического преобразования проекта БД из ER-модели в реляционную, при этом преобразование выполняется в даталогическую модель, соответствующую конкретной СУБД. Все CASE-системы имеют развитые средства документирования процесса разработки БД, автоматические генераторы отчетов позволяют подготовить отчет о текущем состоянии проекта БД с подробным описанием объектов БД и их отношений как в графическом виде, так и в виде готовых стандартных печатных отчетов, что существенно облегчает ведение проекта.

Сущность – это реальный или воображаемый объект, информация о котором представляет интерес. На схемах модели ER сущность представлена ​​в виде прямоугольника (рисунок 1), содержащего имя сущности. В этом случае имя сущности – это имя типа, а не конкретного объекта – экземпляра этого типа. Каждый экземпляр объекта должен отличаться от любого другого экземпляра того же объекта.

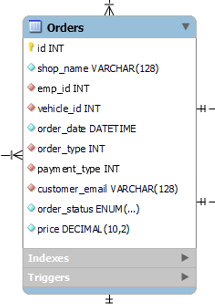


Рисунок 1 – ER-сущность «Заказы»

Отношения – это графическая ассоциация, установленная между двумя объектами. Эта ассоциация всегда является двоичной и может существовать между двумя разными объектами или между объектом и самим собой (рекурсивная связь). В любом отношении различаются два конца (в соответствии с парой связанных сущностей), каждый из которых указывает имя конца отношения, степень конца отношения (сколько экземпляров данной сущности связаны), должен ли какой-либо экземпляр этой сущности участвовать в этом соединении).

Инфологическая модель проектируемой базы данных приведена на рисунке 2.

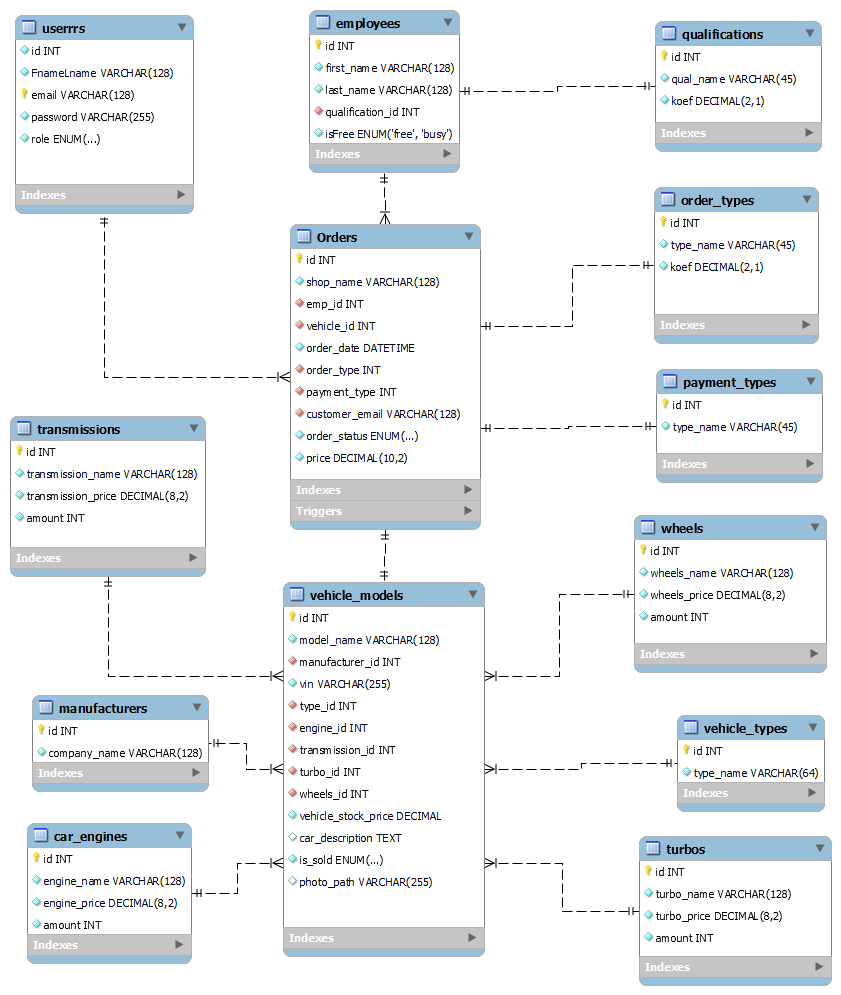


Рисунок 2 – Инфологическая модель базы данных

Стоит также отметить, что важным условием к готовой базе данных является то, что в конце разработки должна получиться база данных в третьей нормальной форме.Метод нормальных форм (НФ) состоит в сборе информации о объектах решения задачи в рамках одного отношения и последующей декомпозиции этого отношения на несколько взаимосвязанных отношений на основе процедур нормализации отношений.

Отношение находится в 1НФ, если все его атрибуты являются простыми, все используемые домены должны содержать только скалярные значения. Не должно быть повторений строк в таблице.

Отношение находится во 2НФ, если оно находится в 1НФ и каждый не ключевой атрибут неприводимо зависит от Первичного Ключа(ПК). Неприводимость означает, что в составе потенциального ключа отсутствует меньшее подмножество атрибутов, от которого можно также вывести данную функциональную зависимость.

Отношение находится в 3НФ, когда находится во 2НФ и каждый не ключевой атрибут нетранзитивно зависит от первичного ключа. Проще говоря, второе правило требует выносить все не ключевые поля, содержимое которых может относиться к нескольким записям таблицы в отдельные таблицы.

Как можно узнать из схемы, приведённой на рисунке 2, база данных приведена к третьей нормальной форме.

# Выбор и обоснование используемых типов данных и ограничений (доменов)

В MySQL три основных типа данных – строчный (string), числовой (numeric) и тип даты и времени (date and time). Каждый из вышеперечисленных типов имеет как минимум ещё 10 подтипов. Естественно, имеет использовать только те типы данных, которые действительно необходимы.

Строчные типы данных, которые были использованы в данном курсовом проекте:

1 VARCHAR (size) – это строка переменной длины, которая может содержать буквы, числа и специальные символы. Параметр size указывает максимальную длину строки – может быть от 0 до 65535. Данный тип используется для хранения строковых значений, например – имён и фамилий сотрудников, названий моделей автомобилей.

2 TEXT (size) может содержать строку максимальной длиной 65 535 байт. Данный тип используется для хранения описаний автомобилей.

3 ENUM (val1, val2, val3, ...) – это строковый объект, который может иметь только одно значение, выбранное из списка возможных значений. Можно перечислить до 65535 значений. Если вставлено значение, которого нет в списке, будет вставлено пустое значение. Значения отсортированы в том порядке, в котором они введены. Используется для хранения ролей пользователей, а также для хранения занятости работников и занятости автомобилей.

Числовые типы данных, которые были использованы в данном курсовом проекте:

1 INT (size) – это целое число. Диапазон включает числа от -2147483648 до 2147483647. Диапазон без знака – от 0 до 4294967295. Используется, в основном, для хранения идентификаторов в сущностях.

2 DECIMAL (size; d) – точное число с фиксированной точкой. Общее количество цифр указано в параметре size. Количество цифр после десятичной точки указывается в параметре d. Максимальное число size – 65. Максимальное число для d – 30. Значение по умолчанию для size – 10. Значение по умолчанию для d – 0. Используется для хранения стоимостей автомобилей и деталей, а также для значений коэффициентов у типов заказа.

Что касается типов даты и времени, был выбран только один. DATETIME – комбинация даты и времени. Формат: ГГГГ-ММ-ДД чч: мм: сс. Поддерживаемый диапазон: от «1000-01-01 00:00:00» до «9999-12-31 23:59:59». Можно добавлять параметры DEFAULT и ON UPDATE в определение типа для автоматической инициализации и обновления до текущей даты и времени.

Выше были перечислены типы данных, которые были использованы при создании сущностей для базы данных. В проектируемой базе данных использовались лишь необходимые типы, чтобы сделать понятные и обоснованные ограничения полей. Также можно заметить, что в некоторые сущностях есть поля, который имеют связь с фото. Для этих случаев было решено использовать тип VARCHAR и хранить в этих полях путь до директории, в которой хранится необходимое фото.

Что касается ограничений (CONSTRAINTS), то они используются для того, чтобы ограничивать значения в таблице тем или иным способом. Это нужно, чтобы обеспечить целостность базы данных. Ограничения могут быть настроены на уровне столбца и на уровне таблицы в целом. Всего ограничений 6:

1 NOT NULL. В MySQL ограничение NOT NULL позволяет указать, что столбец не может содержать никаких значений NULL(то есть ничего не содержать).

2 UNIQUE. Ограничение UNIQUE в MySQL не позволяет вставлять повторяющееся значение в столбец. Ограничение UNIQUE поддерживает уникальность столбца в таблице. В таблице можно использовать более одного столбца UNIQUE.

3 PRIMARY KEY. Ограничение PRIMARY KEY создает уникальный индекс (первичный ключ) для более быстрого доступа к таблице.

4 FOREIGN KEY. FOREIGN KEY в MySQL создает связь между двумя таблицами по одному определенному столбцу обеих таблиц. Указанный столбец в одной таблице должен быть обозначен как PRIMARY KEY и на него должен ссылаться столбец другой таблицы – FOREIGN KEY.

5 CHECK Ограничение CHECK используется для ограничения диапазона значений в столбце.

6 DEFAULT. Это ограничение означает то, что в таблице MySQL каждый столбец должен содержать какое-либо значение (включая NULL). При вставке данных в таблицу, если в столбец не задано значение, столбец получает значение, установленное как DEFAULT.

В реляционных базах данных практически всегда разные таблицы логически связаны друг с другом. Первичные ключи используются для организации такой связи.

Первичный ключ – это специальный столбец таблицы реляционной базы данных (или комбинация столбцов), предназначенный для уникальной идентификации каждой записи таблицы. Первичный ключ используется как уникальный идентификатор для быстрого анализа данных в таблице. Таблица не может иметь более одного первичного ключа.

Основные характеристики первичного ключа:

– первичный ключ должен содержать уникальное значение для каждой строки данных;

– первичный ключ не может содержать нулевые значения (null);

– каждая строка должна иметь значение первичного ключа;

– первичный ключ может использовать одно или несколько полей, уже присутствующих в базовой модели данных, или может быть создано конкретное дополнительное поле в качестве первичного ключа.

Использование первичных ключей также является важным шагом в приведении базы данных к третьей нормальной форме. Ведь для этого нужно, чтобы база данных уже была приведена ко второй нормальной форме, а перед этим к первой нормальной форме, одним из условий которой является наличие первичных ключей.

Внешний ключ – это столбец или группа столбцов в таблице реляционной базы данных, которая обеспечивает связь между данными в двух таблицах. Он действует как перекрестная ссылка между таблицами, поскольку ссылается на первичный ключ другой таблицы, тем самым устанавливая связь между ними. Большинство таблиц в системе реляционных баз данных придерживаются концепции внешнего ключа. В сложных базах данных и хранилищах данных данные в домене должны добавляться в несколько таблиц, таким образом поддерживая связь между ними. Данный принцип работы с данными позволяет сохранить целостность данных, а также минимизирует избыточность.

После настройки первичных и внешних ключей таблицы будут так или иначе связаны, то есть будут находиться в отношениях. Есть определенные правила, которые определяют отношения между таблицами:

– отношение 1:1, когда первичный ключ для одной из таблиц включен в качестве внешнего ключа в другой таблице;

– отношение 1:n, когда первичный ключ из одной таблицы добавляется в качестве внешнего ключа в таблицу n;

– отношение m:n, при котором создается новая таблица (таблица связи), где первичный ключ состоит из первичных ключей двух оригинальных таблиц.

# Проектирование запросов к базе данных

Взаимодействие между базой данных и пользователем будет осуществляться с помощью клиента (разрабатываемого программного средства). Клиент, в свою очередь, обращается к серверу с базой данных с помощью специальных запросов.

Например, та или иная информация о любой сущности может быть получена с помощью SQL запросов.

Общая структура запроса чтения выглядит следующим образом:

– SELECT ('столбцы или \* для выбора всех столбцов; обязательно');

– FROM ('таблица; обязательно');

– WHERE ('условие/фильтрация; необязательно');

– GROUP BY ('столбец, по которому нужно сгруппировать данные; необязательно');

– HAVING ('условие/фильтрация на уровне сгруппированных данных; необязательно');

– ORDER BY ('столбец, по которому нужно отсортировать вывод; необязательно').

Описания элементов запроса:

1 SELECT, FROM – обязательные элементы запроса, которые определяют выбранные столбцы, их порядок и источник данных.

2 WHERE – необязательный элемент запроса, который используется, когда нужно отфильтровать данные по нужному условию. Очень часто внутри данного элемента используются IN / NOT IN для фильтрации столбца по нескольким значениям, AND / OR для фильтрации таблицы по нескольким столбцам.

3 GROUP BY – необязательный элемент запроса, с помощью которого можно задать агрегацию по нужному столбцу (например, если нужно узнать какое количество клиентов живет в каждом из городов). При использовании GROUP BY обязательно, чтобы перечень столбцов, по которым делается разрез, был одинаковым внутри SELECT и внутри GROUP BY, агрегатные функции (SUM, AVG, COUNT, MAX, MIN) должны быть также указаны внутри SELECT с указанием столбца, к которому такая функция применяется.

4 HAVING – необязательный элемент запроса, который отвечает за фильтрацию на уровне сгруппированных данных (по сути, WHERE, но только на уровень выше).

5 ORDER BY – необязательный элемент запроса, который отвечает за сортировку таблицы. Осуществлять сортировку можно и по нескольким столбцам, в этом случае сортировка происходит по порядку указанных столбцов. По умолчанию сортировка происходит по возрастанию для чисел и в алфавитном порядке для текстовых значений. Если нужна обратная сортировка, то в конструкции ORDER BY после названия столбца надо добавить DESC.

SELECT относится к стандартным командам, которые позволяют работать непосредственно с данными. К таким командам также относятся команды CREATE, UPDATE и DELETE, которые используются, соответственно, для создания, обновления и удаления таблиц.

Что касается группы пользователей «администратор», то данная группа выполняет запросы непосредственно к таблицам базы данных. Например, запрос выбора всех полей из таблицы с двигателями автомобилей и результат такого запроса представлены на рисунке 3.



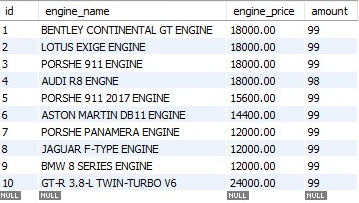
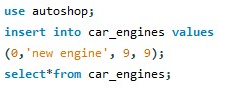


Рисунок 3 – Пример запроса select

Подобным образом происходит выбор всех полей для всех таблиц. Администратор должен быть способен также добавлять новые поля, удалять либо изменять уже существующие.

Добавление записи в таблицу с, например, двигателями к автомобилям, осуществляется через оператор insert и представлено на рисунке 4. Данный оператор используется, например, при регистрации нового пользователя.



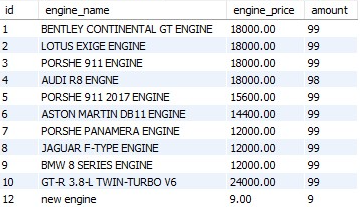
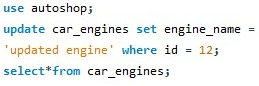


Рисунок 4 – Пример добавления поля

Изменение записей в таблице происходит с помощью оператора update (рисунок 5). При этом важно указать поле, к которому должны быть применены изменения, иначе изменениям подвергнутся все поля в таблице.



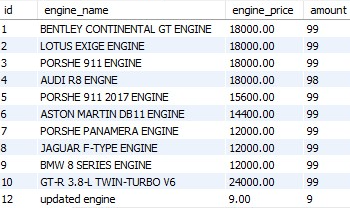
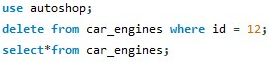


Рисунок 5 – Пример изменения поля

Данный запрос используется, например, для управления ролями администратором, а также для изменения конфигурации автомобиля после создания заказа пользователем.

Запрос на удаление поля можно осуществить с помощью оператора delete (рисунок 6). Как и в случае с update, лучше использовать условие, по которому удалится то или иное поле. Если условие не указать, то все поля удалятся, то есть таблица станет пустой.



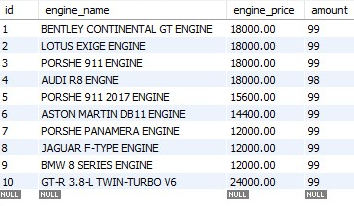


Рисунок 6 – Удаление поля из таблицы

Данный запрос используется, например, пользователем, когда необходимо удалить заказ.

Таким образом была создана связь между сервером с базой данных и программным средством, которое разрабатывается в данном курсовом проекте.

# Программная реализация и документирование базы данных

Программное средство для поддержки работы автомагазина разрабатывалось целиком при помощи языка программирования JavaScript. JavaScript – это кроссплатформенный объектно-ориентированный язык сценариев, используемый для создания интерактивных веб-страниц (например, со сложной анимацией, интерактивными кнопками, всплывающими меню и т. Д.). Существуют также более продвинутые серверные версии JavaScript, такие как Node.js, которые позволяют добавлять на веб-сайт больше функций, чем просто загрузка файлов (например, совместная работа между несколькими компьютерами в реальном времени). Внутри веб-браузера JavaScript может быть связан с объектами своей среды, чтобы обеспечить программный контроль над ними.

JavaScript содержит стандартную библиотеку объектов, таких как Array, Date и Math, а также основной набор языковых элементов, таких как операторы, управляющие структуры и инструкции. Базовый JavaScript можно расширить для различных целей, добавив в него дополнительные объекты.

Например:

1 Клиентский JavaScript расширяет базовый язык, предоставляя объекты для управления браузером и его объектной моделью документа (DOM). Например, клиентские расширения позволяют приложению размещать элементы в HTML-форме и реагировать на пользовательские события, такие как щелчки мыши, ввод формы и навигация по страницам.

2 Серверный JavaScript расширяет базовый язык, предоставляя объекты, относящиеся к запуску JavaScript на сервере. Например, серверные расширения позволяют приложению обмениваться данными с базой данных, обеспечивать непрерывность информации от одного вызова до другого приложения или выполнять манипуляции с файлами на сервере.

Это означает, что в браузере JavaScript может изменить внешний вид веб-страницы (DOM). Точно так же Node.js JavaScript на сервере может отвечать на пользовательские запросы из кода, написанного в браузере.

В качестве среды разработки использовался Visual Studio Code – редактор исходного кода, созданный Microsoft (рисунок 7). Visual Studio Code включает встроенную поддержку автозавершения кода IntelliSense, глубокого понимания семантического кода и навигации, а также рефакторинга кода. Отладка часто является той функцией, которую разработчики больше всего упускают из виду при более компактном кодировании, поэтому Visual Studio Code включает интерактивный отладчик, поэтому можно выполнять пошаговое выполнение исходного кода, проверять переменные, просматривать стеки вызовов и выполнять команды в консоли. VS Code также интегрируется со средствами сборки и создания сценариев для выполнения общих задач, ускоряя повседневные рабочие процессы. VS Code поддерживает Git, поэтому можно работать с системой управления версиями, не выходя из редактора, включая просмотр изменений в ожидающих изменениях.

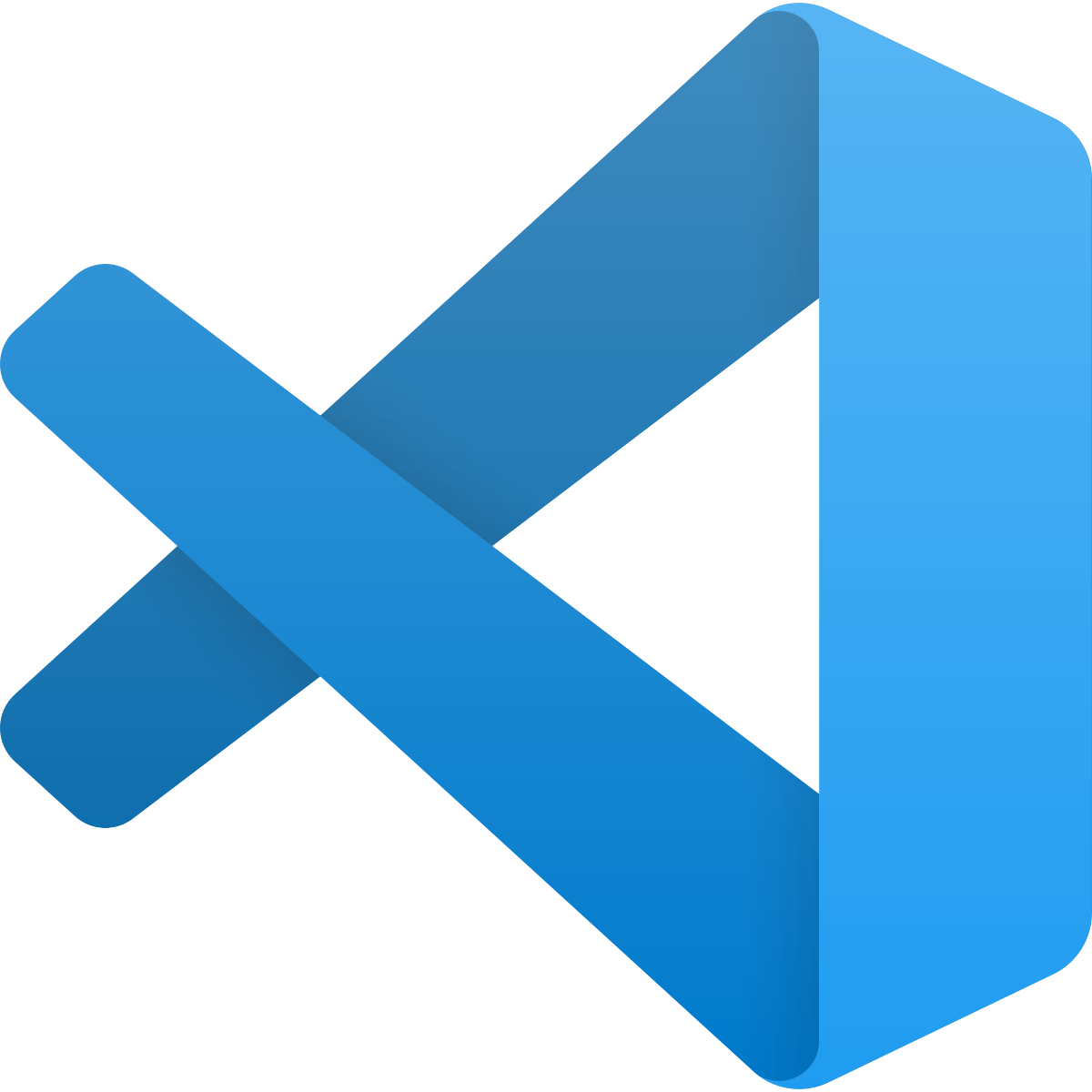


Рисунок 7 – Visual Studio Code

Что касается документации базы данных, есть несколько подходов. Конечно, в контексте данного курсового проекта, в качестве средства документации можно использовать ER-диаграмму, однако в общей практике – это не самый удачный способ, потому что такая диаграмма может быть очень запутанной и мало кто сможет из неё что-либо узнать. Таким образом, хорошая и удобная к чтению человеком документация будет включать в себя для каждой таблицы:

– описание значения таблицы и ее функционального использования (в пользовательском интерфейсе и т. д.);

– описание того, что означает каждый атрибут, если это не очевидно;

– объяснение отношений (внешних ключей) от этой таблицы к другим, и наоборот;

– объяснения дополнительных ограничений и / или триггеров;

– дополнительное объяснение основных представлений и процессов, которые касаются таблицы, если они еще не хорошо документированы.

Учитывая всё вышеперечисленное следует сказать, что не нужно делать документацию ради документации, потому что повторение очевидного будет только сбивать с правильных мыслей. Вместо этого нужно сосредоточиться на том, что может быть неочевидно, и потратить несколько минут на написание действительно четких и кратких объяснений.

Документацию можно проводить как вручную, так и с помощью сторонних программных средств. Инструменты документации баз данных действуют как каталог данных, который фиксирует всю информацию базы данных в упрощенном табличном формате. Чаще всего он используется в качестве словаря данных, который предоставляет подробные сведения о схеме таблицы, или в качестве единственного источника достоверной информации, который обеспечивает обзор всех взаимосвязей внутри базы данных.

# ПРИМЕНЕНИЕ РАЗРАБОТАННОЙ БАЗЫ ДАННЫХ

**3.1 Руководство пользователя**

Взаимодействие пользователя и клиента начинается с окна входа, в котором есть форма как форма для входа, так и форма для регистрации. Форма для входа представлена на рисунке 8.

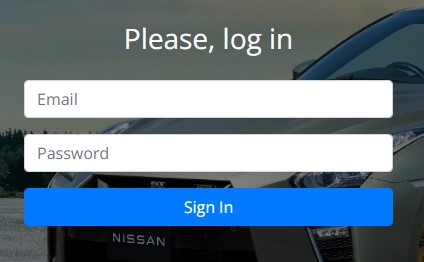


Рисунок 8 – Форма входа в учетную запись

При правильном вводе данных произойдёт перенаправление на домашнюю страницу. Если же ввод почты или пароля произведён неверно, то страница входа просто перезагрузится.

При отсутствии аккаунта предусмотрена форма регистрации (рисунок 9). При успешной регистрации будет выведено информирующее сообщение об успешности операции, после которого будет возможен вход.

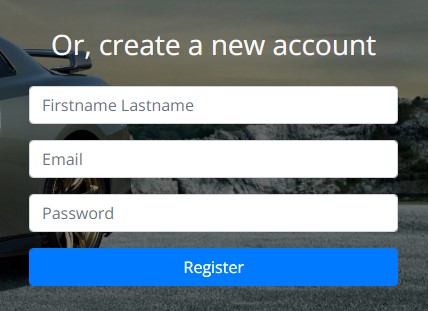


Рисунок 9 – Форма регистрации

Пустыми поля оставить не получится, будет выведено предупреждение о необходимости ввести данные рядом с пустым полем (рисунок 10).

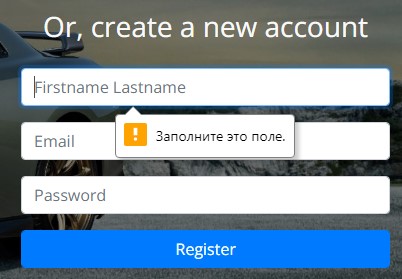


Рисунок 10 – Предупреждение о пустом поле

# ЗАКЛЮЧЕНИЕ

После того, как было проанализировано условие задания, а также были прочитаны статьи и лекции о компьютерных сетях, в эмуляторе сетей Cisco Packet Tracer была создана рабочая компьютерная сеть.

В конечном итоге сеть состоит из 59 рабочих мест (ПК) и 29 прочих устройств и имеет древовидную топологию. Были реализован доступ к указанным в задании сервисам в сети: FTP, DHCP, DNS, E-MAIL, WEB. Были обеспечены: комфортное управление сетью (с помощью SSH), безопасность (через сегментацию, списки доступа, сетевые фильтры), отказоустойчивость основных компонентов сети (с помощью резервирования каналов, дублирования оборудования, использования источников бесперебойного питания). Была создана Wi-Fi зона для сотрудников компании, которая покрывает всю площадь офиса, и Wi-Fi сеть для гостей (DMZ). Построенная сеть обладает высоким показателем быстродействия, так как используются патч-корды с пропускной способностью 10gbit между сетевыми устройствами и 1gbit к оконечным. Таким образом, вероятность возможного простоя сведена к минимуму.

В конечном итоге, выполнение всех заданных требований позволило построить полностью рабочую модель сети.

Курсовой проект выполнен самостоятельно, проверен в системе «Антиплагиат». Процент оригинальности составляет 82,37%. Цитирования обозначены ссылками на публикации, указанными в «Списке использованных источников». Скриншот приведен в приложении.

# СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

Первичный ключ [Электронный ресурс]. – Режим доступа: https://www.techopedia.com/definition/5547/primary-key. – Дата доступа: 11.11.2021.

Типы данных MySQL [Электронный ресурс]. – Режим доступа: https://www.w3schools.com/sql/sql\_datatypes.asp. – Дата доступа: 15.11.2021.

Фронтенд [Электронный ресурс]. – Режим доступа: https://www.seobility.net/en/wiki/Frontend. – Дата доступа: 01.12.2021.

Нормализация отношений [Электронный ресурс]. – Режим доступа: https://habr.com/ru/post/254773/. – Дата доступа: 21.11.2021.

JavaScript [Электронный ресурс]. – Режим доступа: https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Introduction. – Дата доступа: 05.11.2021.

VS Code [Электронный ресурс]. – Режим доступа: https://code.visualstudio.com/docs/editor/whyvscode. – Дата доступа: 05.11.2021.

[1] routebig [Электронный ресурс]. – Режим доступа:

https://routebig.com/types-of-network-topology/. – Дата доступа: 23.02.2021.

[2] На сети [Электронный ресурс]. – Режим доступа:

https://naseti.com/o-routerah/dlya-ofisa.html. – Дата доступа: 26.02.2021.

[3] Маршрутизатор Mikrotik CCR1016-12G. [Электронный ресурс]. – Режим доступа: https://catalog.onliner.by/switch/mikrotik/ccr1016\_12g. – Дата доступа: 01.03.2021.

[4] Коммутатор L3 Cisco C9200-24PXG-A. [Электронный ресурс]. – Режим доступа: https://www.router-switch.com/c9200-24pxg-a.html. – Дата доступа: 01.03.2021.

[5] Коммутатор L2 Huawei S5700-28X-PWR -LI-AC. [Электронный ресурс]. – Режим доступа: https://www.router-switch.com/s5700-28x-pwr-li-ac-p-15918. – Дата доступа: 01.03.2021.

[6] Точка доступа Huawei AP5030DN. [Электронный ресурс]. – Режим доступа: https://www.router-switch.com/ap5030dn-p-20997.html. – Дата доступа: 01.03.2021.

[7] Сетевая розетка Schneider SEDNA SDN5000121. [Электронный ресурс]. – Режим доступа: https://deal.by/p145176082-rozetka-kompyuternaya2xrj45.html. – Дата доступа: 02.03.2021.

[8] Сетевая розетка Legrand VALENA INMATIC. [Электронный ресурс]. – Режим доступа: https://deal.by/p138369239-valena-inmatic-rozetka.html. – Дата доступа: 02.03.2021.

[9] Патч-корд 1Гбит/c GCR LSZH UTP Cat5e. [Электронный ресурс]. – Режим доступа: https://deal.by/p143680338-setevoj-kabel-gcr.html. – Дата доступа: 03.03.2021.

[10] Патч-корд 10Гбит/c GCR LSZH STP Cat6. [Электронный ресурс]. – Режим доступа:

https://www.techly.com/s-ftp-roll-cable-cat-6-cca-100m-stranded-pimf-lsoh.html. – Дата доступа: 03.03.2021.

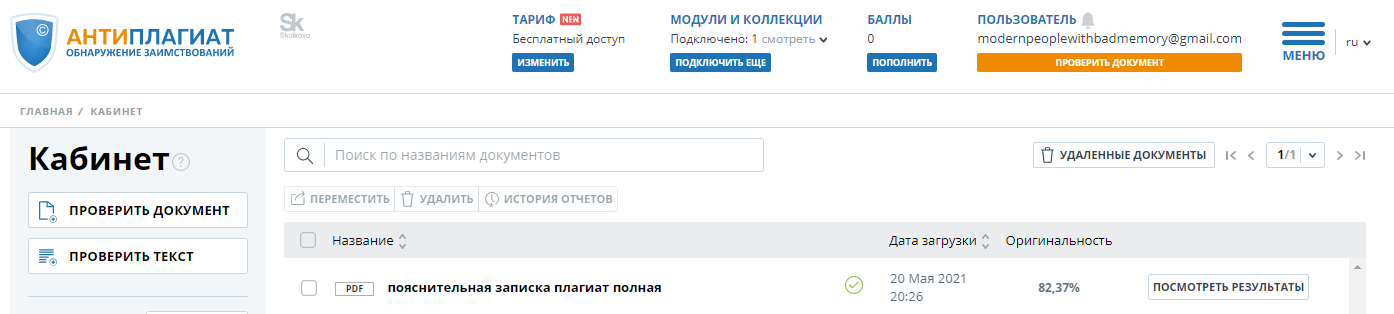
[11] Патч-панель Exegate EX256752RUS UTP Cat5e. [Электронный ресурс]. – Режим доступа: https://deal.by/p58697126-patch-panel-utp.html. – Дата доступа: 03.03.2021.

[12] Коммутационный шкаф SYSMATRIX WP 6406.710. [Электронный ресурс]. – Режим доступа: https://sysmatrix.ru/catalog/. – Дата доступа: 03.03.2021.

# ПРИЛОЖЕНИЕ А

# (обязательное)

# Проверка на оригинальность



# ПРИЛОЖЕНИЕ Б

# (обязательное)

# Ведомость документов

