****

**Частное учреждение профессионального образования**

**«Высшая школа предпринимательства»**

**(ЧУПО «ВШП»)**

КУРСОВОЙ ПРОЕКТ

«Разработка базы данных для магазина колесных дисков»

Выполнил:

студент 3-го курса специальности

09.02.07 «Информационные системы и программирование»  
Зима Дмитрий Владимирович

подпись: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Проверил:

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

оценка: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

подпись: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Содержание**

Введение……………………………………………………………………

**Глава 1.** Теоретические аспекты о разработке базы дынных………….

1.1 Анализ предметной области………………………………………….

**Введение**

**Актуальность**  
В наше время выбор автомобилей очень велик, и спрос на них очень большой. Вместе со спросом на авто, растет спрос на комплектующие, включая диски. Большинство машин продаются на не особо приятных виду дисках.   
Все мы прекрасно понимаем, на сколько большое влияние оказывают на автомобиль его диски. Они не только могут полностью изменить взгляд на авто, но и улучшить его характеристики, ведь колеса для автомобиля, как обувь для человека. Если для повседневной ходьбы в быстром темпе вы выберете дешевые, тяжелые резиновые сапоги, вам будет тяжело преодолеть нужное расстояние и при этом не устать или еще хуже, не повредить ноги. Так происходит и у машин. Чем легче и качественнее диски, которые вы выберете, тем большее влияние они окажут на разгон и тем более управление вашего авто.  
Магазин автодисков поможет людям выбрать те колеса, которые подойдут их автомобилю, а разработанная для этого магазина база данных поможет сотрудникам магазина лишний раз не запутаться в подсчетах.  
**Разработанность**  
База данных позволяет эффективно управлять всей информацией о наличии товаров, их характеристиках, ценах, поставках, клиентах и заказах. Это поможет улучшить процессы управления складом, контроля продаж, а также обеспечит более быструю обработку заказов.

Таким образом, разработка базы данных для магазина колесных дисков не только актуальна, но и является ключевым инструментом для эффективного управления бизнесом и привлечения клиентов.

**Цель**  
Цель данной курсовой работы является разработка базы данных для магазина колесных дисков.  
Исходя из поставленной мною цели проекта, можно составить ряд задач, выполнение которых приведет к достижению цели работы.

**Задачи**

Для достижения цели необходимо решить несколько задач:1. Провести анализ предметной области.  
 2. Спроектировать схему базы данных.

3. Разработать базу данных.  
 4. Заполнить базу данных

5. Проверить работоспособность базы данных. **Объект исследования:**

Объектом исследования является база данных

**Предмет исследования:**

Предметом исследования является база данных для магазина авто дисков

**Метод исследования:**Основные методы исследования, которые будут применяться в данной работе, включают следующие:  
1. Анализ научных источников. Изучение веб-сайтов, статей, форумов и прочего по теме, чтобы лучше разобраться в изучаемом и проще найти подход к разработке базы данных.

2. Проектирование и разработка. Создание модели базы данных с использованием ER-диаграммы, потому что она поможет наглядно увидеть структуру связей внутри базы данных.

**Глава 1. Теоретические аспекты о разработки базы данных  
1.1Анализ предметной области**Моя курсовая работа основана на изучении информации и разработке базыданных, поэтому сначала предлагаю поближе познакомиться с базой данных, и узнать, что это вообще такое. **База данных** — это упорядоченный набор структурированной информации или данных, которые обычно хранятся в электронном виде в компьютерной системе. База данных обычно управляется системой управления бзами данных (СУБД). Данные вместе с СУБД, а также приложения, которые с ними связаны, называются системой баз данных, или, для краткости, просто базой данных.

Данные в наиболее распространенных типах современных баз данных обычно хранятся в виде строк и столбцов формирующих таблицу. Этими данными можно легко управлять, изменять, обновлять, контролировать и упорядочивать. В большинстве баз данных для записи и запросов данных используется язык структурированных запросов (SQL).

Также не стоит забывать, что есть множество различных типов БД, которые мы можем разделить на две группы. Реляционные БД и Нереляционные. Рассмотрим их подробнее…  
  
**Реляционные базы данных** – это тип базы данных, в которой хранятся и систематизируются точки данных с определенными связями для быстрого доступа. В реляционной базе данные организованы в таблицы, содержащие информацию о каждом объекте и представляющие заранее определенные категории через строки и столбцы. Такое структурирование данных делает доступ к ним эффективным и гибким, поэтому реляционные базы данных наиболее распространены.

РСУБД различает следующие типы операций:

Логические: В этом случае приложение указывает, что нужно сделать.   
Например, приложение запрашивает имя сотрудника или добавляет запись о сотруднике в таблицу.

Физические: В этом случае РСУБД определяет, как все должно быть сделано, и выполняет операцию. Например, после того, как приложение запрашивает таблицу, база данных может использовать индекс для поиска запрошенных строк, считывания данных в память и выполнения многих других шагов, прежде чем возвращать результат пользователю. СУБД хранит и извлекает данные таким образом, чтобы физические операции были прозрачны для приложений баз данных.

Код\_продавца Код\_поставщика Код\_изгот.

Продавец Наименование поставщика Изготовитель

Модель фирмы Код\_изготовителя Страна

Код\_поставщика Год

Таблица 1. Пример реляционных баз данных.

Популярные реляционные базы данных: MySQL, PostgreSQL, Microsoft SQL Server, Oracle и SQLite.

В реляционных базах данных приняты следующие обозначения(таблица1):

Отношение – таблица;

Поле – набор однотипных записей для нескольких объектов (столбец);

Кортеж (запись) – строка таблицы, содержащая набор нескольких записей

соответствующих одному объекту;

Атрибут – запись в строке одного поля.

Сущность – любой различимый объект, информация о котором хранится в базе данных

Нереляционные базы данных — тип базы данных, ориентированный на работу с вложенными структурами данных. В отличие от реляционных БД, тут не используют таблицы, строки и столбцы, вместо этого они поддерживают различные модели данных, такие как ключ-значение, столбцовые, документные и графовые.

В нереляционных базах данных применяется модель хранения, оптимизированная под конкретные требования типа хранимых данных. Например, данные могут храниться как простые пары «ключ — значение», документы JSON или граф, состоящий из рёбер и вершин.

Примеры нереляционных СУБД:

— MongoDB - система хранения информации, основанная на формате документа;

— Cassandra - система хранения информации, основанная на колонках;

— Redis - система хранения информации, основанная на ключ-значениях;

— Neo4j - система хранения информации, основанная на графах.

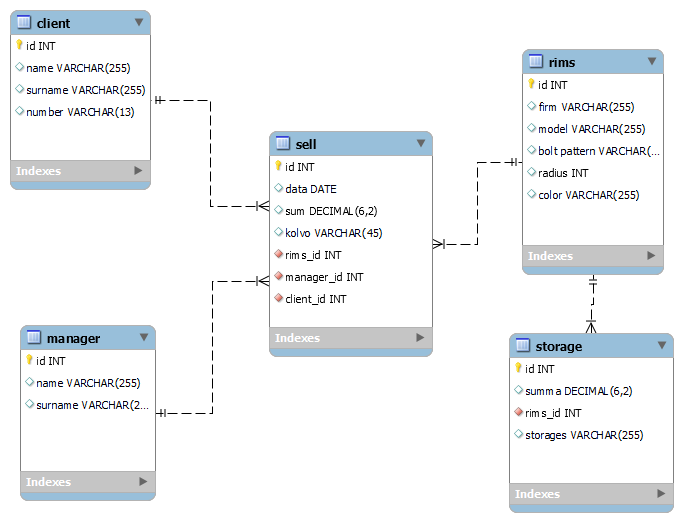
— Amazon DynamoDB - услуга по регулированию баз данных от компании Amazon.  
  
  
1.2 Требования к базе данных   
Реляционные базы данных мне показались самыми подходящими под критерии выбранной мной темой, поэтому для разработки я выбрал именно их.   
Реляционные базы данных обеспечивают структурирование данных, целостность, мощный язык запросов (SQL), гибкость, надежность и масштабируемость.  
  
1.3 Проектировка ERD  
ERD (Entity-Relationship Diagram) — это графическое представление структуры базы данных, которое наглядно показывает нам сущности (entities) и их взаимосвязи (relationships). Оно помогает понять, как данные связаны между собой и как они могут быть организованы в базе данных.  
Построение ERD перед началом разработки базы данных имеет несколько преимуществ:

1. Понимание требований: ERD помогает лучше понять бизнес-требования и взаимосвязи между данными, что позволяет создать более точную модель данных.

2. Идентификация проблем: При построении ERD можно заранее выявить потенциальные проблемы в структуре данных, такие как избыточность или несогласованность.

3. Оптимизация структуры: ERD позволяет оптимизировать структуру базы данных, улучшая производительность и эффективность запросов к данным.

4. Согласованность данных: Построение ERD помогает обеспечить согласованность данных и уменьшить вероятность возникновения ошибок при разработке базы данных.

Таким образом, построение ERD перед началом разработки базы данных способствует созданию более качественной и эффективной структуры данных, соответствующей потребностям бизнеса.  
  
  
1.4 Практическое построение ERD на примере диаграммы, построенной мною.  
  
Моя схема ERD, представленная ниже: Готовая база данных, основанная на требованиях выбранной темы  


В базе данных находятся 5 таблиц: client, manager, sell, rims, storage  
Рассмотрим каждую подробнее:  
1. Client-Содержит информацию о клиентах, такую как (Имя, Фамилия, Номер телефона)  
2. Manager-Содержит информацию о менеджерах (продавцах) (Имя и Фамилия)  
3. Rims-Содержит информацию о всем товаре, который мы имеем. (фирма, модель, разболтовка колеса (PCD), радиус продаваемого диска и цвет)  
4. Storage-Содержит информацию о складах (количество товара, информацию о товаре, и какой конкретно склад)  
5. Sell-Грубо говоря основная таблица, таблица продажи. В которую входит информация из таблиц rims-manager-client.

Глава 2. Разработка базы данных для магазина колесных дисков наглядно  
  
2.1 Отдельно и детально  
Сейчас предлагаю рассмотреть каждую таблицу отдельно и более детально. Начнем с таблицы клиент.  
1. Таблица Client состоит состоит из id, name, surname, number Id-в котором тип данных INT, также используется первичный ключ (Primary Key), также она не может быть не заполненной (Not Null), также используется Auto Increment для создания столбца в таблице, который автоматически увеличивается на единицу при каждой вставке новой записи.  
Name-Имя клиента, в котором тип данных VARCHAR(255), количество символов я  решил оставить «по госту». Тоже самое я сделал и с таблицей surname  
Number-Таблица, в которой будет храниться номер телефона клиента, ей я присвоил тип данных VARCHAR(13). Почему я выбрал максимальное значение символов 13? Потому что это самый длинный возможный номер телефона в мире. Всегда есть вероятность что в магазин придет клиент с номером телефона Китая или Нигерии. Вероятность крайне мала, но не равна нулю.  
2. Таблица Manager состоит из id (INT), name VARCHAR(255), surname VARCHAR(255)  
3. Таблица sell включает в себя столбцы id(NN, PK,AI) , data, sum, kolvo, а также включает связи с таблицами rims, manager, client. Для столбца data я выбрал тип данных DATE, потому что он поддерживает только значения даты, и ничего лишнего. Для столбца sum (Итого сделки), я выбрал DECIMAL (6.2), 6.2 означает 6 целых цифр, и 2после запятой, допустим сумма сделки:123321.12. Для kolvo я выбрал VARCHAR(45), потому что посчитал что так будет удобнее вводить значения.   
  
Связи с таблицами rims, manager, client, отвечают за то, чтобы мы понимали через какого менеджера, какой клиент, купил какие диски.  
4. Таблица rims (id, firm, model, bolt pattern, radius, color).  
Id по‑прежнему содержит PK, NN, AI. Столбец firm (фирма производитель) имеет тип данных VARCHAR(255). Столбец model(модель диска) также использует VARCHAR(255), Для таблицы bolt pattern (разболтовка колеса, она же PCD) уже сложнее было выбрать тип данных, ведь разболтовка выражается двумя числами, например, 5x112. Первое число указывает на количество болтов, а второе — на расстояние между ними (измеренное в миллиметрах). Также существуют разболтовки не из целых чисел, например 5х114,3. Поэтому, для простоты действий я выбрал VARCHAR(255).  
!!Для радиуса выбрал неправильное!!  
Для столбца Color логично было выбрать VARCHAR(255), так как в наше время цвета зачастую обозначают не как просто purple, light-blue-gray, red, green.  
Производители зачастую стараются дать своим цветам красивые названия, такие как frozen purple, nardo gray, rosso corsa, british racing green, также не исключаются названия в численном значении.  
5. Таблица storage (int, summa, storages), а также связь с rims, которая показывает информацию о товаре на складе.  
sum (6.2)  
storages VARCHAR(255)