Государственное бюджетное профессиональное образовательное учреждение города Москвы "Московский государственный колледж электромеханики и информационных технологий"

Практическая работа №3

Работу выполнил:

студент 3 курса 3ИП-11-19

Клейменов Д. Д.

Преподаватель:

Басыров С. А.

**ВВЕДЕНИЕ**

Цель – Научиться формировать требования к базе данных

(программному продукту)

Задачи:

1) Обосновать выбор СУБД;

2) Описать основные сущности и их характеристики, которые будут

храниться в БД (в каком виде и как);

3) Описать основные сценарии использования СУБД с помощью use-

case;

4) Составить список API методов, предоставляемой БД для работы «из

вне»;

5) Определить список пользователей их роли.

**ОСНОВНАЯ ЧАСТЬ**

**1 Обосновать выбор СУБД;**

Система управления базами данных (далее СУБД) - система управления базами данных (БД) (DBMS) отвечает за агрегирование данных и их последующее хранение и обработку.

СУБД управляется на языках работы с БД, например, SQL (Structured Query Language). СУБД основаны на реляционной модели данных. Реляционная модель - представление БД в виде таблиц для действий над записями на языке SQL. Реляционные системы - это системы «автоматической навигации». SQL - более абстрактный язык, чем C, т.к. способ запроса остается на выбор оптимизатора СУБД. «Постреляционная СУБД» - наличие в реляционной СУБД файлов управления данными, не вписывающихся в реляционную модель, т.е. объектов. Ранее данные хранились только в алфавитно-цифровой форме, классифицировались по стандартным типам (строки, целые числа и т.д.). Теперь сюда включаются и бинарные объекты: изображения, видео и большие фрагменты текста, по которым может происходить поиск. Выбор СУБД зависит от тех приложений, которыми она будет управляться, т.е. выбор СУБД - прерогатива разработчика, а не пользователя. Ведущие поставщики СУБД: IBM, Oracle и Microsoft. При выборе СУБД необходимо руководствоваться такими показателями, как масштабируемость, быстродействие (как в выборе транзакций, так и в построении сложных аналитических выборок), работа с XML и кластерные решения.

Помимо универсальности и распространенности СУБД MySQL обладает целым комплексом важных преимуществ перед другими системами. В частности следует отметить такие качества как:

Простота в использовании. MySQL достаточно легко инсталлируется, а наличие множества плагинов и вспомогательных приложений упрощает работу с базами данных.

Обширный функционал. Система MySQL обладает практически всем необходимым инструментарием, который может понадобиться в реализации практически любого проекта.

Безопасность. Система изначально создана таким образом, что множество встроенных функций безопасности в ней работают по умолчанию.

Масштабируемость. Являясь весьма универсальной СУБД, MySQL в равной степени легко может быть использована для работы и с малыми, и с большими объемами данных.

Скорость. Высокая производительность системы обеспечивается за счет упрощения некоторых используемых в ней стандартов.

Как и любой программный продукт, система MySQL имеет определенные ограничения в своем функционале, что не позволяет использовать ее для работы с приложениями, имеющими некоторые специфические требования. К недостаткам этой СУБД относятся:

Недостаточная надежность. В вопросах надежности некоторых процессов по работе с данными (например, связь, транзакции, аудит) MySQL уступает некоторым другим СУБД.

Низкая скорость разработки. Как и многим другим программным продуктам с открытым кодом, MySQL не достает некоторого технического совершенства, что порой сказывается на эффективности процессов разработки.

Сам факт того, что MySQL является самой популярной СУБД на рынке, говорит об ее универсальности и предпочтительности в большинстве ситуаций. В частности ее используют в следующих ситуациях:

при распределённых операциях, когда функционала SQLite (другая популярная система) не хватает;

когда требуется обеспечить высокий уровень безопасности, что MySQL делает с успехом;

для работы с интернет-страницами и веб-приложениями, поскольку MySQL является наиболее удобной СУБД для этой сферы применения;

при работе со специфическим проектом, где функционал MySQL дает оптимальный результат.

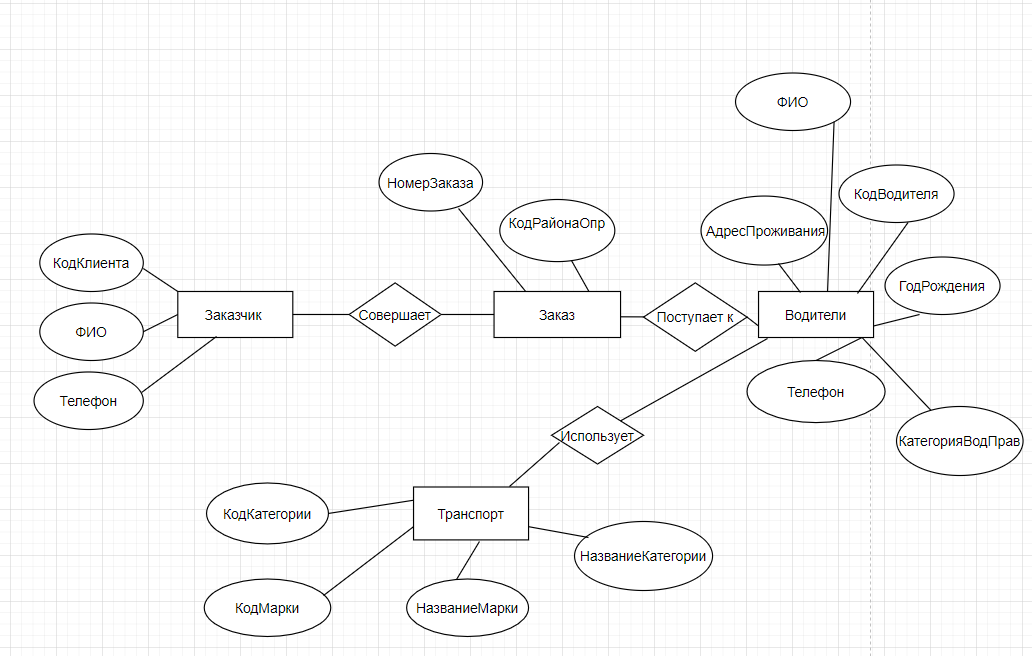
Тем не менее, существуют ситуации, когда от MySQL всё же следует отказаться в пользу других СУБД. Например, она не годится, если:

нужно соответствовать стандарту SQL, который данная система поддерживает лишь частично;

проект предусматривает многопоточность данных, поскольку при осуществлении параллельных операций чтения/записи у MySQL могут возникать проблемы;

имеющийся функционал MySQL не способен обеспечить весь набор возможностей работы с базой данных.

**2 Как создавать диаграмму сущностей**



­­­­­­­­­

Рисунок 1 – Диаграмма сущностей

Мы можем упростить описание сущности благодаря нашей диаграмме.­­­­­­

**3 Основные сценарии использования СУБД с помощью use-case**

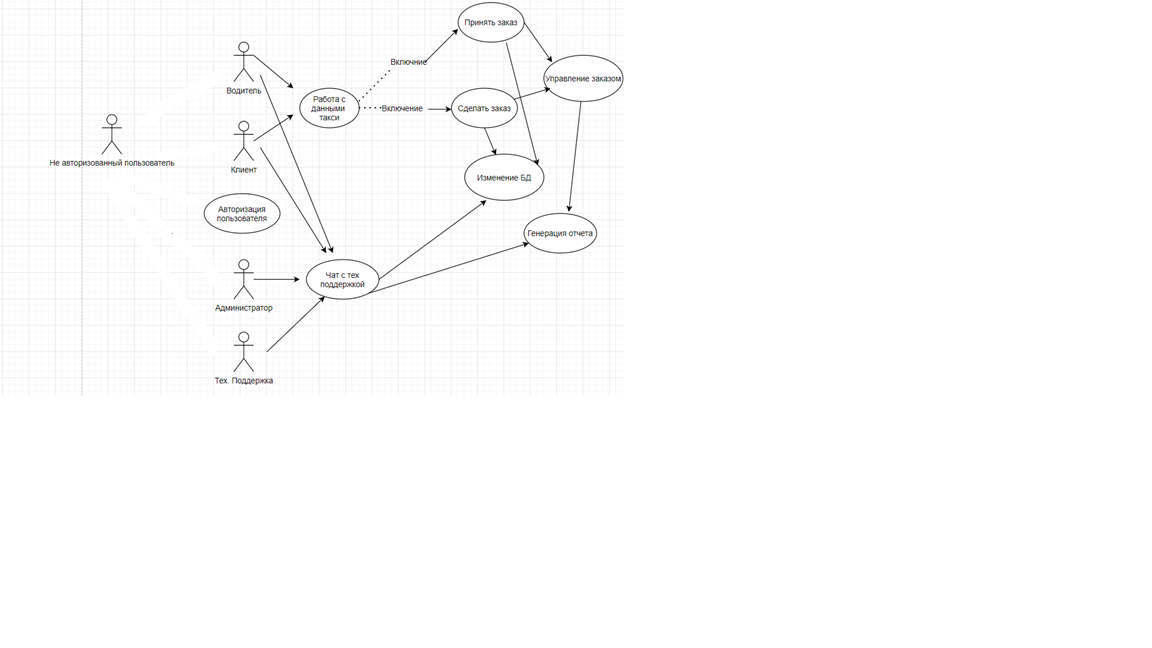


Рисунок 1 – Диаграмма сущностей

На основании нашей схемы будет использоваться UML то есть use-case.

Используется она для того чтобы мы могли представить функционал разрабатываемой системы.

**4 API методы представляемые БД для работы «из вне»**

|  |  |  |
| --- | --- | --- |
| Метод | Описание | Аргументы |
| Добавить заказ | Доступ: клиент | Код района, код заказчика |
| Принять заказ | Доступ: водитель | Код водителя |
| Завершить заказ | Доступ: водитель, администратор, тех. поддержка | Код заказа |
| Добавить транспорт | Доступ: администратор | Код категории, код марки |
| Изменить транспорт | Доступ: администратор | Код категории, код марки |
| Удалить транспорт | Доступ: администратор | Код транспорта |
| Регистрация заказчика | Доступ: не авторизированный пользователь | ФИО, телефон |
| Изменение заказчика | Доступ: администратор | Код заказчика, ФИО, телефон |
| Добавить водителя | Доступ: администратор | ФИО, адрес проживания, категория прав, телефон |
| Изменить водителя | Доступ: администратор | Код водителя, ФИО, адрес проживания, категория прав, телефон |
| Удалить водителя | Доступ: администратор | Код водителя |
| Изменить заказ | Доступ: администратор | Код заказа, Код района, ко заказчика, код водителя |

Таблица 1 – Список методов

**5 Список пользователей и их роли**

Будут представлены следующие роли:

Клиент – совершает заказ имеет полный доступ к базе данных

Водитель – исполняет заказ.

**ЗАКЛЮЧЕНИЕ**

В выполнении данной работы был выполнен анализ требований с базой данных. Также я провел анализ с основными объектами, функциями и разработки баз данных.