**СОДЕРЖАНИЕ**

**ПЕРЕЧЕНЬ УСЛОВНЫХ ОБОЗНАЧЕНИЙ, СИМВОЛОВ**

**И ТЕРМИНОВ**

БД – База данных

СУБД ­– Система управления базами данных

*API* – *Application Programming Interface*

*REST* – *Representational state transfer*

*JSON* – *JavaScript Object Notation*

# ВВЕДЕНИЕ

По данным на 2020 год 74 процента людей по всему миру имеют доступ к персональным компьютерам, а в развитых странах эта цифра достигает 92 процентов. В двадцать первом веке, который принято называть веком информационных технологий, жизнь практически каждого человека. из развитых стран напрямую связано с постоянным использование компьютерной техники, как и персональных компьютеров в частности.

Под компьютерными технологиями скрывается большое множество различных понятий, таких как:

­­– мобильные гаджеты;

– интернет-технологии;

– персональные компьютеры;

– различные умные гаджеты и так далее.

Говоря про отрасль производства персональных компьютерных, нельзя не отметить, что в последние годы она стала одной из самых востребованных, так как спрос на всевозможные компьютерные комплектующие значительно превышает предложение, что можно с легкостью отследить по росту цен на них, а также благодаря заявлениям самих компаний-производителей.

Никакая современная компания не обходится без баз данных, и компании-производители компьютерных комплектующих, конечно, не являются исключением, так как базы данных выполняют крайне важные функции, а именно хранение всевозможной информации, администрированию и управлению данными.

Целью данного курсового проекта является проектирование базы данных для поддержки производства персональных компьютеров.

При выполнения курсового проекта были учтены потребности пользователей, связанные с различными взаимодействиями с созданной базой данных: регистрация клиента, создание заказа, просмотр сотрудников, состояний заказов, добавление новых комплектующих, готовых сборок, и другие.

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

Деятельность, направленная на выявление реальных потребностей заказчика, а также на выяснения смысла высказанных требований, называется анализом предметной области. Анализ предметной области – это первый шаг этапа системного анализа, с которого начинается разработка программной системы.

Формализация предметной области заключается в построении ее модели и разработке методов преобразования модели предметной области в модель исполнителя.

## **Описание предметной области**

Предметная область – часть реального мира, подлежащая изучению с   
целью организации управления и в конечном итоге автоматизации. Она содержит в себе информацию, которую необходимо формализовать для внесения в базу данных.

По причине того, что данная база данных предназначена для поддержки

производства персональных компьютеров, то прежде всего нужно узнать, какие потребности есть к данной базе данных. При чем надо учитывать, что база данных реализуется как для клиентов, так и для администраторов, исполнителей заказов, что вносит некоторые коррективы и дополнительные требования к базе данных.

Для полноценного функционирования производства персональных компьютеров в базе данных был создан список сущностей, полностью обеспечивающих функционирование производства персональных компьютеров.

Сущность «Клиенты» необходима для хранения данных о клиентах производства персональных компьютеров. Данная сущность обладает следующими атрибутами:

– уникальный идентификатор;

– имя;

– фамилия;

– электронная почта;

– номер телефона;

– пароль для входа.

Для хранения данных о заказе конкретного клиента была создана сущность «Заказы», обладающая следующими атрибутами:

– уникальный идентификатор;

– адрес;

Для отслеживания выбранного способа доставки была создана сущность «Способы доставки», содержащая следующие атрибуты:

– уникальный идентификатор;

– тип доставки.

Кроме отслеживания выбранного способа доставки, нужно отслеживать выбранные способ оплаты. Для этого была создана сущность «Способы оплаты», содержащая следующая атрибуты:

– уникальный идентификатор;

– тип оплаты.

Сущность «Компьютер» предназначена для хранения информации о компьютере, в неё входят следующие атрибуты:

– уникальный идентификатор;

– название;

– стоимость.

Сущность «Способы сборки» создана для возможности выбора пользователем разных вариантов конфигурации компьютера, и содержит следующие атрибуты:

– уникальный идентификатор;

– тип сборки.

Сущность «Работники» предназначена для хранения информации о работниках производства. Данная сущность содержит следующие атрибуты:

– уникальный идентификатор;

– имя;

– фамилия;

– отчество;

– адрес проживания;

– номер телефона;

– электронная почта;

– пароль для входа.

Так как на любом предприятии есть множество должностей, то их хранения была создана сущность «Должности», в которой содержатся атрибуты:

– уникальный идентификатор;

– название.

Для хранения информации о комплектующих, содержащихся на производстве, была создана сущность «Комплектующие» с следующими атрибутами:

– уникальный идентификатор;

– название;

– характеристики;

– стоимость.

С целью хранения информации о типах комплектующих была создана сущность «Типы комплектующих», в которой содержатся следующие атрибуты:

– уникальный идентификатор;

– тип комплектующей.

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

Так как разрабатываемое приложение предназначено, как и для сотрудников, так и для клиентов компании, то и функционал будет значительно отличаться.

Клиент имеет доступ к регистрации и просмотру списка заказов.

Для сотрудников с правами доступа уровня администратора функционал будет значительно больше, в него также входят такие функции, как создание, редактирование и удаление всех таблиц в базе данных.

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

База данных представляет собой совокупность структурированных взаимосвязанных данных, относящихся к определенной предметной области и организованных для решения определенных задач разными пользователями.

Особенности каждой предметной области могут накладывать разнообразные требования и ограничения на хранимые в базе данные.

На основе выбранной предметной области на созданную базу данных накладываются следующие ограничения:

– все электронные почты должны быть уникальными;

– все телефонные номера должны быть уникальными;

– названия должностей, типы способов доставки и способы оплаты, сборки должны быть уникальными;

– текстовое поле для ввода номера имеет фиксированное значение по длине в двенадцать символов;

– все цены должны быть указаны в целочисленных значениях;

– все текстовые поля, за исключением текстовых полей для ввода номера телефона и адресов, имеют максимальное значение по длине в двадцать символов.

**1.4** **Постановка решаемой задачи**

Основной целью курсового проекта является проектирование и создание базы данных для поддержки работы производства персональных компьютеров. Проектируемая база данных должна отвечать требованиям надежности, минимальной избыточность, целостности данных и ее схема должна быть приведена к третьей нормальной форме. База данных должна поддерживать основные современные средства для работы и администрирования. Также необходимо реализовать клиентское приложение для взаимодействия с разработанной базой данных.

В качестве клиента для данного курсового проекта было выбрано мобильное приложение на базе системы Android, а в качестве СУБД *MySQL*. При разработке мобильного приложения было нужно реализовать взаимодействие между СУБД и самим приложением благодаря разработанному интерфейсу доступа к данным – *REST API*.

В результате курсового проектирования ожидается создание базы данных для поддержки производства персональных компьютеров, взаимодействие с которой осуществляется при помощи мобильного приложения.

# 2 ПРОЕКТИРОВАНИЕ БАЗЫ ДЛЯ ОСНОВНОГО ВИДА ДЕЯТЕЛЬНОСТИ РАССМАТРИВАЕМОЙ ПРЕДМЕТНОЙ ОБЛАСТИ

2.1 Разработка инфологической модели предметной области базы данных

Проектирование БД – одна из наиболее сложных и ответственных задач, связанных с созданием информационной системы. В результате решения этой задачи должны быть определены содержание БД, эффективный для всех ее будущих пользователей способ организации данных и инструментальные средства управления данными.

Цель инфологического моделирования – обеспечение наиболее естественных для человека способов сбора и представления той информации, которую предполагается хранить в создаваемой базе данных. Храниться данные будут в диаграмме «сущность – связь», которая будет создана с помощью СУБД.

Диаграмма «сущность – связь» (ER-диаграмма) позволяет графически представить все элементы информационной модели согласно простым, интуитивно понятным, но строго определенным правилам – нотациям.

При разработке базы данных проводилась в программе-инструменте для визуального проектирования баз данных *MySQL Workbench*.

Создание новой модели производится при нажатии на пункт «*New Model*» в выпадающем меню пункта «*File*», как показано на рисунке 2.1, либо же благодаря нажатии комбинации клавиш «*Ctrl* + *N*».

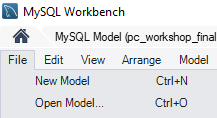


Рисунок 2.1 – Создание новой модели

При создании новой модели, открывается окно создания диаграммы, представленное на рисунке 2.2.

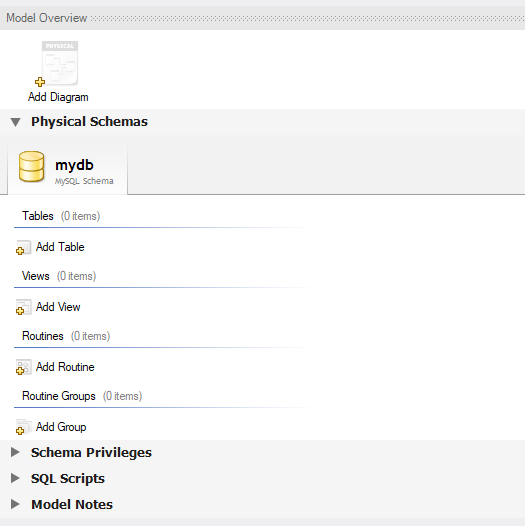


Рисунок 2.2 – Окно создания диаграммы.

В окне создания диаграммы можно, как и создать саму диаграмму, так и выставить определенные параметры для неё.

Для начала работы с диаграммой необходимо дважды нажать на пункт «*Add Diagram*», после чего откроется окно проектирования диаграммы, где можно осуществить множество различных действий:

– создание новых таблиц;

– редактирование существующих таблиц;

– изменение свойств таблицы;

– создание текстовых заметок.

Для создания таблиц в окне проектирования диаграммы стоит выбрать инструмент «*Place a New Table*» (см. рисунок 2.3).

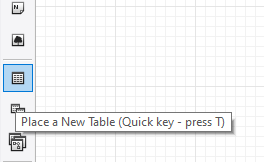


Рисунок 2.3 – Создание новой таблицы

После создания таблицы, необходимо добавить к ней атрибуты. Для выполнения данного действия стоит дважды кликнуть по созданной таблице. При исполнении этих действий откроется окно просмотра и редактирования свойств таблицы (см. рисунок 2.4).

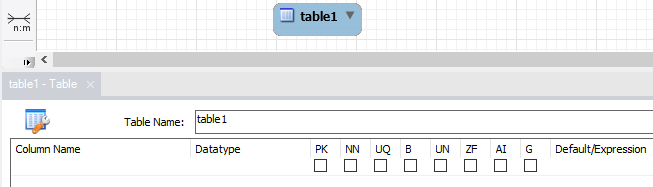


Рисунок 2.4 – Просмотр и редактирование свойств таблицы

В представленном на рисунке выше окне можно выполнять следующие действия:

– изменять имя таблицы;

– добавлять атрибуты к таблице;

– выбирать свойств для атрибутов.

Таким образом нужно создать все необходимые таблицы для базы данных и выставить связи между ними (см. рисунок 2.5).

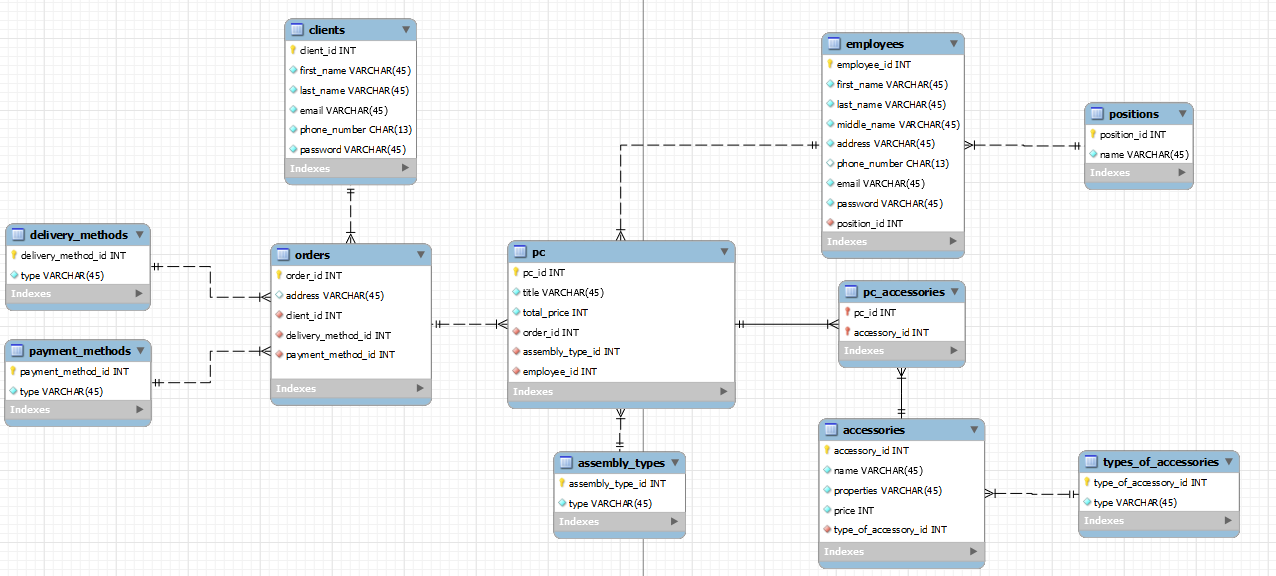


Рисунок 2.5 – Созданная диаграмма «сущность – связь»

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

В *MySQL* существует множество типов данных, которые можно разделить на шесть групп: целые, вещественные, строковые, бинарные, даты и времени, перечисления и множества.

В ходе проектирования базы данных используются следующие типы данных из *MySQL*:

– *VARCHAR*;

– *INT*.

*VARCHAR* – строковые данные переменного размера. Используются значения для определения размера строки в байтах (допускаются значения от 1 до 8000) или указание предельного размера столбца, вплоть до максимального размера хранилища, что составляет 2^31-1 байт (2 ГБ).

*INT* – целочисленный тип данных, один из простейших и самых распространённых типов данных в базах данных и программировании. Служит для представления целых чисел. [2]

2.3 Проектирование запросов к базе данных

Запрос – объект базы данных, который используется для извлечения информации из одной или нескольких таблиц или для выполнения определенных действий с данными [3].

Запрос (команда) строится на основе одной или нескольких взаимосвязанных таблиц, позволяя комбинировать содержащуюся в них информацию. При этом могут использоваться как таблицы БД, так и сохраненные таблицы, полученные в результате выполнения других запросов. Кроме того, запрос может строиться непосредственно на другом запросе.

Чтобы просмотреть данные из таблицы следует воспользоваться оператором *SELECT*. На рисунке 2.6 приведен пример просмотра таблицы клиентов.



Рисунок 2.6 – Запрос данных для таблицы клиентов

Результат выполнения запроса представлен на рисунке 2.7.

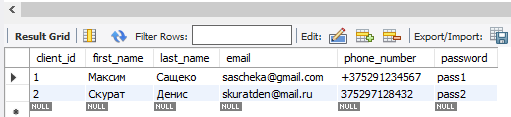


Рисунок 2.7 – Выполнения запроса вывода информации о клиентах

Для получения информации из таблицы по определенному значению используется оператор *SELECT*, где дополнительно нужно указать желаемый атрибут (см. рисунок 2.8).



Рисунок 2.8 – Запрос на получение имен из таблицы клиентов

Результат выполнения запроса представлен на рисунке 2.9.

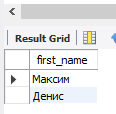


Рисунок 2.9 – Выполнения запроса вывода имен клиентов

Для добавления данных в таблицу используется оператор *INSERT INTO*. На рисунке 2.10 приведен пример добавления данных в таблицу клиентов.



Рисунок 2.10 – Добавление данных в таблицу клиентов

В результате выполнения операции таблица клиентов заполнена введенными значениями (см. рисунок 2.11)

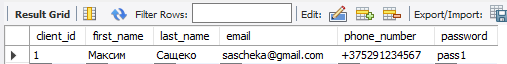


Рисунок 2.11 – Заполненная таблица клиентов

Для обновления данных в таблице используется оператор *UPDATE* (см. рисунок 2.12).



Рисунок 2.12 – Выполнения запроса на редактирование таблицы клиентов

Результат выполнения представлен на рисунке 2.13.

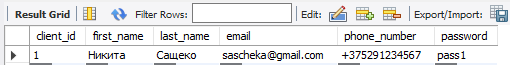


Рисунок 2.13 – Измененная таблица клиентов

Для удаления данных из таблицы используется оператор *DELETE*. На рисунке 2.14 приведен пример удаления данных из таблицы клиентов.



Рисунок 2.14 – Удаление данных из таблицы клиентов

После этого из таблицы клиентов были удалены выбранные данные (см. рисунок 2.15)



Рисунок 2.15 – Таблица клиентов после выполнения операции удаления

На основе представленных выше запросов были созданы все остальные запросы к базе данных производства персональных компьютеров.

2.4 Программная реализация и документирование базы данных

Так как данная курсовая работа разрабатывалась под операционную систему *ANDROID*, то выбор языков программирования был не столь большим, а именно *JAVA* и *KOTLIN*.

Язык программирования *KOTLIN* впервые был представлен в 2011 году, более чем на 20 лет после появления языка *JAVA*. В 2017 году компания *GOOGLE* в своей официальной документации признала *KOTLIN* наиболее подходящим языком для создания *ANDROID-*приложений и рекомендовала его к использованию программистам.

Основными преимуществами языка программирования *KOTLIN* можно смело назвать следующие пункты:

– полная совместимость с Java;

– чистота;

– безопасность;

– простота синтаксиса.

Если говорить про преимущества языка *JAVA* перед *KOTLIN* то таковыми стоит отметить:

– б

[2] <https://ru.wikipedia.org/wiki/Целое_(тип_данных)>

[3] http://lab314.brsu.by/roleg/bio/bio/bit/access/lr3tnew.html