СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 3](#_Toc456552148)

[1 ОБЗОР СОСТОЯНИЯ ВОПРОСА 4](#_Toc456552149)

[1.1 Анализ литературы 4](#_Toc456552150)

[1.2 Обзор инструментов для реализации данной задачи 5](#_Toc456552151)

[1.3 Логическая модель базы данных 5](#_Toc456552152)

[1.4 Физическая модель базы данных 7](#_Toc456552153)

[2 РАЗРАБОТКА ПРИЛОЖЕНИЙ ПО РАБОТЕ С БАЗОЙ ДАННЫХ 10](#_Toc456552154)

[2.1 Архитектура приложения 10](#_Toc456552155)

[2.2 Описание приложения 12](#_Toc456552156)

[2.3 Тестирование приложения 12](#_Toc456552157)

[2.4 РУКОВОДСТВО ПОЛЬЗОВАТЕЛЮ 15](#_Toc456552158)

[ЗАКЛЮЧЕНИЕ 19](#_Toc456552159)

[СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ 20](#_Toc456552160)

# ВВЕДЕНИЕ

Производственная практика студентов является важнейшей частью подготовки высококвалифицированных специалистов. Практика в учебных лабораториях университетов, а также на государственных, малых, совместных предприятиях, в научно-исследовательских институтах, банках, конструкторских бюро и вычислительных центрах, на которых используется современная вычислительная техника. Производственная практика является основным получения практических знаний работы по специальности. Целью данной практики является систематизация и закрепление практических и теоретических знаний.

Объектом преддипломной практики является отдел информационных технологий ОАО «Бобруйский машиностроительный завод — крупнейший в СНГ производитель центробежных насосов для нефтеперерабатывающей, нефтехимической, горнодобывающей, металлургической, целлюлозно-бумажной, энергетической, цементной промышленности, а также для перекачивания сточных жидкостей на предприятиях городского, промышленного и сельского водоснабжения.

В ходе производственной практики необходимо получить практические знания в области применения современных информационных систем и средств в производстве.

# 1 ОБЗОР СОСТОЯНИЯ ВОПРОСА

Информационную систему современного предприятия невозможно представить без надежно функционирующей базы данных (БД).

Система управления базой данных (СУБД) должна обеспечить высокую пропускную способность, целостность хранимых данных, безопасность выполнения транзакций и защиту конфиденциальных данных.

Поскольку содержимое БД предприятия является коммерческой тайной, в качестве задания была выбрана задача написания веб-приложения по работе с базой данных. В ходе выполнения работы необходимо произвести оптимизацию созданных хранимых процедур.

Веб-приложения являются наиболее востребованными из-за своей интерактивности и доступности. В разработке корпоративных веб-приложений наиболее часто применяется технология Java EE.

Интерфейс веб-приложения реализуется в виде HTML-страниц с применением каскадных таблиц стилей CSS. При помощи JavaScript возможно создать динамический интерфейс.

Система контроля версий предназначена для облегчения работы при написании кода приложения, а также для организации командной работы над проектом.

1. Анализ литературы

Microsoft SQL Server — [система управления реляционными базами данных (РСУБД)](https://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D0%BB%D1%8F%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D0%A1%D0%A3%D0%91%D0%94), разработанная корпорацией [Microsoft](https://ru.wikipedia.org/wiki/Microsoft). Основной используемый язык запросов — [Transact-SQL](https://ru.wikipedia.org/wiki/Transact-SQL), создан совместно Microsoft и [Sybase](https://ru.wikipedia.org/wiki/Sybase). Используется для работы с базами данных размером от персональных до корпоративных.

Java EE – технология создания серверной платформы веб-приложений для решения задач средних и крупных предприятий. Актуальная версия Java EE имеет номер 7.0.

HTML — стандартный язык разметки документов. Большинство веб-страниц содержат описание разметки на языке HTML. Язык HTML интерпретируется браузерами и отображается в виде документа в удобной для человека форме.

CSS — формальный язык описания внешнего вида документа, написанного с использованием языка разметки. CSS используется для задания [цветов](https://ru.wikipedia.org/wiki/%D0%A6%D0%B2%D0%B5%D1%82), [шрифтов](https://ru.wikipedia.org/wiki/%D0%A8%D1%80%D0%B8%D1%84%D1%82), расположения отдельных блоков и других аспектов представления внешнего вида этих веб-страниц.

JavaScript – [прототипно-ориентированный](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D1%82%D0%BE%D1%82%D0%B8%D0%BF%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5) [сценарный](https://ru.wikipedia.org/wiki/%D0%A1%D1%86%D0%B5%D0%BD%D0%B0%D1%80%D0%BD%D1%8B%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) [язык программирования](https://ru.wikipedia.org/wiki/%D0%AF%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F). JavaScript обычно используется как встраиваемый язык для программного доступа к объектам [приложений](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D0%B0%D1%8F_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B0) (DOM). Наиболее широкое применение находит в [браузерах](https://ru.wikipedia.org/wiki/%D0%91%D1%80%D0%B0%D1%83%D0%B7%D0%B5%D1%80) как язык сценариев для придания [интерактивности](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D1%80%D0%B0%D0%BA%D1%82%D0%B8%D0%B2%D0%BD%D0%BE%D1%81%D1%82%D1%8C) [веб-страницам](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D1%82%D1%80%D0%B0%D0%BD%D0%B8%D1%86%D0%B0).

Query — [библиотека JavaScript](https://ru.wikipedia.org/wiki/%D0%91%D0%B8%D0%B1%D0%BB%D0%B8%D0%BE%D1%82%D0%B5%D0%BA%D0%B0_JavaScript), фокусирующаяся на взаимодействии [JavaScript](https://ru.wikipedia.org/wiki/JavaScript) и [HTML](https://ru.wikipedia.org/wiki/HTML). Библиотека jQuery помогает легко получать доступ к любому элементу [DOM](https://ru.wikipedia.org/wiki/Document_Object_Model), обращаться к атрибутам и содержимому элементов [DOM](https://ru.wikipedia.org/wiki/Document_Object_Model), манипулировать ими. Также библиотека jQuery предоставляет удобный [API](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F_%D0%BF%D1%80%D0%B8%D0%BB%D0%BE%D0%B6%D0%B5%D0%BD%D0%B8%D0%B9) для работы с [AJAX](https://ru.wikipedia.org/wiki/AJAX).

AJAX – технология обращения к серверу без перезагрузки страницы За счет этого уменьшается время отклика и веб-приложение по интерактивности больше напоминает десктоп.

Git - распределённая [система управления версиями](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0_%D1%83%D0%BF%D1%80%D0%B0%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D1%8F_%D0%B2%D0%B5%D1%80%D1%81%D0%B8%D1%8F%D0%BC%D0%B8). Ядро Git представляет собой набор утилит командной строки с параметрами. Все настройки хранятся в текстовых файлах конфигурации. Репозиторий Git представляет собой каталог файловой системы, в котором находятся файлы конфигурации, файлы журналов, хранящие операции, выполняемые над репозиторием, индекс, описывающий расположение файлов и хранилище, содержащее собственно файлы.

1.2 Обзор инструментов для реализации данной задачи

Для создания Java EE приложения применяется среда разработки Intellij Idea 2016. С помощью данной среды создаются клиентская и серверная части веб-приложения.

Веб-сервер Wildfly 9.0.2 Final предназначен для развертывания веб-приложения, создания менеджера подключений к SQL серверу, обеспечения механизма безопасности.

Сборка и тестирование проекта осуществляется при помощи среды Apache Maven.

Доступ к базе данных осуществляется при помощи MS SQL Server 2014. Для работы с базой данных используется среда разработки MS SQL Management Studio.

В качестве системы контроля версий используется Git.

1.3 Логическая модель базы данных

Логическая модель определяет для сущностей их атрибуты, описания и ограничения, уточняет состав сущностей и взаимосвязи между ними. Атрибуты и ограничения сущностей приведены в таблице 1.

Таблица 1 – Атрибуты и ограничения сущности предметной области.

|  |  |  |  |
| --- | --- | --- | --- |
| Сущность | Атрибуты | Тип данных | Описания и ограничения |
| 1 | 2 | 3 | 4 |
| Автобус | id | int | Первичный ключ, автоинкремент, not-null |
| mark | varchar(30) | not-null, обязательное поле |
| model | varchar(20) | not-null, обязательное поле |
| regNumber | varchar(10) | unique, not-null, обязательное поле |
| state | bit | not-null, обязательное поле |
| category | varchar(2) | not-null, обязательное поле |
| description | varchar(64) | not-null |
| Водитель | id | int | Первичный ключ, автоинкремент, not-null |
| lastName | varchar(50) | not-null, обязательное поле |
| firstName | varchar(50) | not-null, обязательное поле |
| patronymic | varchar(50) | not-null |
| birthDate | date | not-null, обязательное поле |
| address | varchar(256) | not-null |
| telephone | varchar(16) | not-null |
| licenseNumber | varchar(10) | not-null, unique, обязательное поле |
| График | id | int | Первичный ключ, автоинкремент, not-null |
| driverId | int | Внешний ключ, not-null, обязательное поле |
| date | date | not-null, обязательное поле |
| shift | int | not-null, обязательное поле |

Так как атрибуты всех сущностей являются атомарными, то модель соответствует первой нормальной форме.

Так как каждый атрибут, который не входит в состав первичного ключа, функционально полно зависит от первичного ключа, модель соответствует второй нормальной форме.

Так как каждый неключевой атрибут нетранзитивно зависит от первичного ключа, то модель соответствует третьей нормальной форме.

Для обеспечения целостности данных задано свойство внешнего ключа OnUpdate Cascade. В этом случае при удалении базового элемента, все дочерние зависимости будут удалены.

Логическая модель предметной области приведена на рисунке 1.



Рисунок 1 - Логическая модель предметной области

1.4 Физическая модель базы данных

Физическая оргранизация данных оказывает влияние на эксплуатационные характеристики проектируемой базы, так как на этом уровне идет привязка к физической памяти.

Для обеспечения целостности данных были созданы первичные ключи, отношения между сущностями (каскадное обновление и удаление при изменении внешнего ключа), построена логическая модель (рисунок 2). Структуры спроектированных сущностей приведены на рисунках 2 – 4.

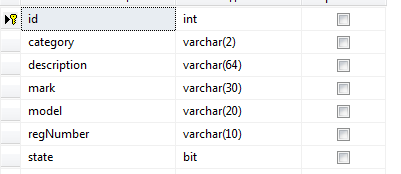


Рисунок 2 – Таблица "Автобус"

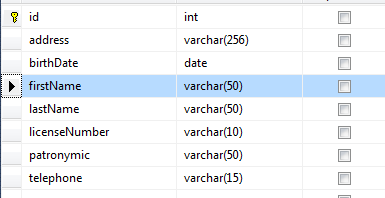


Рисунок 3 – Таблица «Водитель»

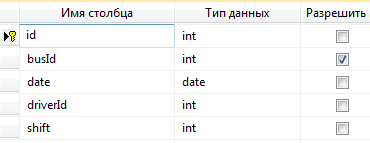


Рисунок 4 – Таблица «График»

Для обеспечения взаимодействия базы данных с приложением-клиентом были созданы хранимые процедуры.

Хранимая процедура – программа на языке SQL, которая хранится на сервере базы данных, используемая для взаимодействия с приложением-клиентом. Достоинством хранимых процедур является независимость от кода клиента, благодаря чему возможно вносить изменения в процедуру без изменения клиентского кода.

Для организации работы были созданы следующие хранимые процедуры:

spFindByNumber – поиск автобуса по гаражному номеру;

spFindByRegNumber - поиск автобуса по регистрационному номеру;

spDeleteBus – удаление автобуса;

spSaveBus – добаление автобуса в базу;

spUpdateBus – редактирование данных об автоьбусе;

spGetAllBuses – выборка всех автобусов;

spFindFreeToday – поиск свободных автобусов на сегодняшний день;

spGetCountsBuses – проверка на наличие существующего регистрационного номера;

spGetCountsDrivers – проверка на наличие существующего водительского удостоверения;

spFindById – поиск водителя по табельному номеру;

spFindFreeById – поиск свободного водителя по табельному номеру;

spFindByLastName – поиск водителя по фамилии;

spDeleteDriver – удаление водителя;

spSaveDriver – добавление водителя в базу;

spUpdateDriver – изменение данных о водителе;

spGetAllDrivers – выборка всех водителей;

spFindGraphToday – вывод рабочего графика на дату;

spAddShift – добавление смены в рабочий график;

spDeleteGraphic – удаление графика;

spSelectGraph – поиск рабочего графика указанного водителя;

spSelectMonthFull – вывод графика на месяц;

spSetBus – добавление автобуса в график;

spSetBusNull – удаление автобуса из графика.

Выводы по разделу

В результате анализа технологий можно сделать вывод, что для создания веб-приложения можно использовать такие технологии, как Java EE, Ajax, HTML, CSS, JS. Наличие технической литературы говорит о широком применении данных технологий.

Анализ схемы БД показал, что база является нормализованной и необходимо произвести оптимизацию хранимых процедур, в результате чего будет сокращено время выполнения команд, повысится скорость обработки данных, будет облегчена поддержка SQL – кода.

# 2 РАЗРАБОТКА ПРИЛОЖЕНИЙ ПО РАБОТЕ С БАЗОЙ ДАННЫХ

Целью проекта является создание удобного и простого приложения для хранения информации о транспортных средствах (автобусах), водителях и графиках их работы. Данное приложение должно позволять пользователю вносить, редактировать, просматривать информацию.

Для реализации поставленной цели необходимо решить следующие задачи:

* рассмотреть существующие на данный момент способы реализации поставленной задачи;
* изучить предметную область, выделить сущности для решаемой задачи;
* построить физическую и логическую модели данных;
* разработать web-приложение;
* описать полученные результаты, сформулировать выводы и заключения;
* произвести оптимизацию хранимых процедур.

В системе должен быть представлено два типа пользователей:

* администратор.
* пользователь.

Администратор должен иметь такие возможности – создание, редактирование, удаление предметных областей «Автобус» и «Водитель», создание рабочего графика, его редактирование и удаление. Пользователь имеет возможность просмотра списка транспортных средств и графика работы.

Требования к реализации проекта:

* платформа Java EE;
* язык программирования – Java;
* среда разработки – Intellij Idea 2016;
* система управления базами данных – MS SQL;
* связь приложения с базой данных с помощью технологии JPA-Hibernate;
* клиентская часть – HTML, CSS, Ajax и Javascript.
* аутентификация пользователя с помощью технологии Java EE Security Realm.

Так же система должна включать в себя:

* набор классов-сущностей;
* подключение JPA - Hibernate для расширения функциональности системы;
* EJB – объекты для создания сервисных классов для работы с базой данных;
* создание подключения к базе данных через сервер приложений.

Так же следует отметить то, что система должна представлять собой клиент-серверное приложение и должна как минимум корректно работать на web-сервере WildFly 9.0.2 Final с платформой JRE версии 1.7 и выше и базой данных, которая поддерживает транзакции, триггеры и хранимые процедуры MS SQL.

# СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

«Эффективная работа с СУБД» под ред. Ахаян Р., С. – П. – 1997

Е.В. Мамаев Microsoft SQL Server 2000. – СПб.: БХВ-Петербург, 2002.

«SQL полное руководство» Грофф Дж.Р., Вайнберг П.Н. - Год издания: 2001.

«SQL в примерах и задачах» И. Ф. Астахова. - Год издания: 2005.

«Java. Промышленное программирование» Блинов И.Н. - Год издания: 2013.