СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 3](#_Toc43133937)

[1 Теоретические основы баз данных 4](#_Toc43133938)

[1.1 Основные термины теории баз данных 4](#_Toc43133939)

[1.2 Нормализация Базы Данных 6](#_Toc43133940)

[1.3 Типы данных и команды SQL 7](#_Toc43133941)

[2. Реализация информационных систем 8](#_Toc43133942)

[2.1 Реализация с помощью средств быстрой разработки 8](#_Toc43133943)

[2.2 Реализация с помощью в СУБД SQL Server 2014 8](#_Toc43133944)

[2.3 Создание пользовательского интерфейса для работы с базой данных в Delphi 7 8](#_Toc43133945)

[ЗАКЛЮЧЕНИЕ 9](#_Toc43133946)

[СПИСОК ИСТОЧНИКОВ 10](#_Toc43133947)

## ВВЕДЕНИЕ

Сайты в наше время стали больше, чем просто набор веб-страниц. Это сложные структуры, состоящие из большого количества отдельных компонентов и модулей. Поэтому они создаются не один-двумя людьми, а целыми группами разработчиков. Для того, чтобы много людей могло работать над одним проектом, используются системы контроля версий, или Git. Крупнейшим представителем является сервис GitHub, его уже использует более 40 млн пользователей по всему миру, и он продолжает набирать популярность. Поэтому системы типа Git являются очень популярными и актуальными в наше время.

Тема курсовой работы: разработка ИС для контроля разработки Web-сайтов группой разработчиков.

Цель курсовой работы: автоматизация контроля разработки Web-сайтов группой разработчиков.

Задачи:

1. Наличие системы регистрации и авторизации для защиты от несанкционированного доступа.
2. Хранение, ввод и получение данных о пользователях, их проектов. Хранение собственно базы кода проектов и их изменений. Обеспечение защиты, сохранности и целостности данных.
3. Ввод и вывод файлов проектов в любой момент.
4. Возможность обновления данных.
5. Выполнение статистических и пользовательских запросов и их обработка. Построение графиков и статистических отчётов на основе полученных результатов.
6. Учет всех пользователей и их проектов.

Пояснительная записка состоит из 2 частей: теоретической и практической. В теоретической части производится описание используемых технических терминов, этапов проектирования информационной системы, используемых типов данных и SQL-запросов. В практической части производится постановка алгоритма реализации проекта. Проект на курсовой работе создаётся с помощью встроенной среды разработки Delphi 7 и СУБД MS Server 2014 ( management studio ) через встроенной язык transact sql. Однако я в своей работе буду создавать программу с помощью языка C#, среды разработки Visual Studio 2019 и двух: СУБД MySQL и её встроенного языка SQL для хранения данных о пользователях и СУБД MongoDB для хранения индексации проектов пользователей. Подробнее о причинах смены программного комплекса и полученных преимуществах рассказано в пункте 2.1.

## 1 Теоретические основы баз данных

## Основные термины теории баз данных

База данных ( БД ) – это совокупность специальным образом организованных данных, хранимых в памяти вычислительной системы. При этом данные должны быть непротиворечивы, минимально избыточны и целостны.

Система управления базами данных ( СУБД ) - совокупность программных и лингвистических средств общего или специального назначения, обеспечивающих управление созданием и использованием баз данных.

Модели баз данных - описания содержания, структуры и ограничений целостности, используемые для создания и поддержки базы данных. Выделяют следующие основные модели баз данных:

1. Реляционная ( несколько взаимосвязанных между собой двумерных таблиц )
2. Постреляционная ( расширенная реляционная модель, снимающая ограничение неделимости данных, хранящихся в записях таблиц, с помощью многозначных полей, т.е. полей, значения которых состоят из подзначений )
3. Сетевая ( все данные связаны со всеми - модель обычно делится на уровни, так, чтобы протоколы вышестоящего уровня использовали бы протоколы нижестоящего уровня )
4. Документно-ориентированная ( хранение данных в виде коллекций документов )
5. Иерархическая ( представление базы данных в виде структуры, состоящей из объектов различных уровней )

Таблица – это совокупность связанных данных, хранящихся в структурированном виде в базе данных. Она состоит из столбцов и строк. Таблица содержит определенное число столбцов, но может иметь любое количество строк.

Поля – это столбцы таблицы в реляционной БД. Они отвечают за какие-либо характеристики объекта и содержат значения данного свойства

Записи – это строки таблицы в реляционной БД. Они отвечают за собственно объекты, хранимые в БД, и содержат набор значений свойств, размещенных в полях базы данных. Каждая запись однозначно определяется одним или несколькими уникальными значениями.

Реляционная БД – база данных, основанная на реляционной модели данных. Она представляет связанную между собой совокупность таблиц-сущностей базы данных и описывается следующими параметрами:

1. Строка таблицы содержит данные об одном объекте, а столбцы этой таблицы описывают различные характеристики этого объекта.
2. Все строки таблицы имеют одинаковую структуру.
3. В каждой таблице должен быть первичный ключ.
4. Первичный ключ может состоять из нескольких полей.
5. Сам ключ должен быть уникальным и однозначно определять запись.
6. Таблицы могут быть связаны друг с другом через ключи

Реляционная модель данных –

Связь – отношение подчиненности, которое определяет, что для каждой записи главной таблицы может существовать одна или несколько записей во внешней таблице.

Виды связей:

1. Связь один-к-одному — к одному главной первой таблицы соответствует только один атрибут внешней таблицы и наоборот.
2. Связь один-ко-многим — к одному атрибуту главной таблицы соответствует несколько атрибутов внешней таблицы.
3. Связь многие-ко-многим — к одному атрибуту главной таблицы соответствует несколько атрибутов внешней таблицы и наоборот.

Ключевое поле или первичный ключ — это поле, значения которого однозначно определяют записи в таблице и являются уникальными в пределах данной таблицы. Служит как ограничение целостности.

Внешний ключ – поле связи дочерней таблицы, является ограничением целостности - дочерняя таблица не может ссылаться на несуществующие записи главной таблицы.

Представление – это SELECT-запрос, сохранённый для дальнейшего использования.

Индекс — объект базы данных, создаваемый с целью повышения производительности поиска данных. При этом индексы замедляются процессы вставки и удаления данных, т.к. они должны производить перерасчёт индексов.

Триггеры – это процедуры, которые автоматически запускаются при выполнении определенной операции (INSERT/UPDATE/DELETE) – до (BEFORE) или после (AFTER) нее.

Данные — формы представления информации, с которыми имеют дело информационные системы и их пользователи.

## Нормализация Базы Данных

Таблица «vergit\_users» содержит информацию о пользователях данной системы.

Таблица 1 – «vergit\_users»

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| id | login | name | email | pass\_hash | time\_reg | time\_update |
| 1 | Cet | Сергей | cet@mail.ru | \*\*\*\*\*\*\*\* | 2020-01-01 12:00:00 | 2020-01-02 12:00:00 |
| 2 | Lasia | Лация | lasia@mail.ru | \*\*\*\*\*\*\*\* | 2020-01-01 13:00:00 | 2020-01-02 13:00:00 |
| 3 | Test | Тест | test@mail.ru | \*\*\*\*\*\*\*\* | 2020-01-01 14:00:00 | 2020-01-02 14:00:00 |
| 4 | Admin | Админ | adm@mail.ru | \*\*\*\*\*\*\*\* | 2020-01-01 15:00:00 | 2020-01-02 15:00:00 |

Таблица «vergit\_ project» содержит информацию о проектах данной системы.

Таблица 2 – «vergit\_project»

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| id | id\_creator\_pr | name\_pr | logo | description | is\_open |
| 1 | 1 | Расписание | u | Сайт с расписанием для студентов | TRUE |
| 2 | 2 | Аналитика | u | Аналитическая нейросеть | FALSE |
| 3 | 3 | Просто так | s | Проект для проверки | TRUE |
| 4 | 4 | Админ-панель | u | Удобное управление всем в системе | FALSE |

Таблица «vergit\_logs» содержит логи глобальных действий над проектами, на основе которых мож быть постоена карта проекта.

Таблица 3 – «vergit\_logs»

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| id | id\_project | id\_user | line | id\_action | start\_one | start\_two | end\_one | end\_two |
| 1 |  |  |  |  |  |  |  |  |
| 2 |  |  |  |  |  |  |  |  |
| 3 |  |  |  |  |  |  |  |  |
| 4 |  |  |  |  |  |  |  |  |

Таблица «vergit\_actions» хранит вынесенный список всех возможных глобальных действий над проектами.

Таблица 4 – «vergit\_actions»

|  |  |
| --- | --- |
| id | action |
| 1 | start |
| 2 | parallel |
| 3 | level-up |
| 4 | level-down |
| 5 | merge |
| 6 | fork |
| 7 | end |

Таблица «vergit\_groups» хранит все группы разработки проектов.

Таблица 5 – «vergit\_groups»

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| id | id\_progect | name\_gr | description | avatar\_gr | id\_creator\_gr | time\_create |
| 1 | 1 | SI | SI corporation | u | 1 | 2020-01-02 12:30:00 |
| 2 | 2 | RedBox | Top Secret | u | 2 | 2020-01-02 13:30:00 |
| 3 | 3 | TestGroup | Тестовая группа | s | 3 | 2020-01-02 14:30:00 |
| 4 | 4 | AdminGroup | Администрация | u | 4 | 2020-01-02 15:30:00 |

Таблица «vergit\_ user\_group» содержит соотношения групп и пользователей, т.е. какой пользователь в каких группах находится.

Таблица 6 – «vergit\_user\_group»

|  |  |  |
| --- | --- | --- |
| id | id\_group | id\_user |
| 1 | 1 | 1 |
| 2 | 2 | 1 |
| 3 | 2 | 2 |
| 4 | 3 | 3 |
| 5 | 4 | 1 |
| 6 | 4 | 4 |

Таблица «vergit\_ version» хранит актуальные версии проектов.

Таблица 7 – «vergit\_version»

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| id | id\_project | line | id\_status | number | time\_create |
| 1 |  |  |  |  |  |
| 2 |  |  |  |  |  |
| 3 |  |  |  |  |  |
| 4 |  |  |  |  |  |

Таблица «vergit\_ statuses» содержит все возможные статусы проектов.

Таблица 8 – «vergit\_statuses»

|  |  |
| --- | --- |
| id | status |
| 1 | sketch |
| 2 | pre-alfa |
| 3 | alfa |
| 4 | pre-beta |
| 5 | beta |
| 6 | release |

Рисунок 1 – инфологическая схема БД

## Типы данных и команды SQL

Типы данных, используемые в MySQL:

1. INT – представляет целые числа от -2147483648 до 2147483647, занимает 4 байта.
2. CHAR(n) – представляет стоку фиксированной длины. Отсутствующие символы заменяются пробелами до получения нужной длины. Всегда занимает столько символов, какова длина строки ( вес зависит от кодировки ).
3. VARCHAR – представляет строку переменной длины. Занимает столько символов, сколько внесено и дополнительный байт длины строки.
4. DATETIME – представляет объединённые время и дату ( диапазон с "1000-01-01 00:00:00" до "9999-12-31 23:59:59" ). Формат по умолчанию: "yyyy-mm-dd hh:mm:ss". Занимает 8 байт.
5. TEXT – представляет текст неопределённой длины, максимум до 65 КБ ( конкретное количество символов зависит от кодировки ).

Основные операторы и запросы SQL, необходимые для работы с БД через методологию CRUD ( Create, Read, Update, Delete ):

1. Cоздание новой базы данных:

CREATE DATABASE name\_db;

1. Cоздание новой таблицы c первичным ключом с автоинкрементом, дополнительным полем и указанием кодировок:

CREATE TABLE name\_table (

id INT NOT NULL AUTO\_INCREMENT,

field\_table VARCHAR(60),

PRIMARY KEY (id)

) ENGINE=InnoDB AUTO\_INCREMENT=1 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci;

1. Вставка данных в таблицу:

INSERT INTO name\_table ( `id`, `field\_table` ) VALUES ( 10, “data” );

1. Выборка данных из таблицы:

SELECT `name\_table`.`field\_table` FROM `name\_table` WHERE id = 10;

1. Обновление данных в таблице:

UPDATE `name\_table` SET `field\_table` = “new data” WHERE id = 10;

1. Удаление данных из таблицы:

DELETE FROM `name\_table` WHERE id = 10;

1. Переименование таблицы:

ALTER TABLE `name\_table` RENAME TO `new\_name\_table`;

1. Добавить поле в таблицу:

ALTER TABLE `name\_table` ADD num INT;

1. Удалить поле из таблицы:

ALTER TABLE `name\_table` DROP COLUMN num;

1. Переименовать поле в таблице:

ALTER TABLE `name\_table` CHANGE `field\_table` `rename\_field` VARCHAR(60);

1. Полностью очистить таблицу от данных:

TRUNCATE TABLE `name\_table`;

1. Удалить таблицу:

DROP TABLE `name\_table`;

1. Удалить базу данных:

DROP DATABASE `name\_table`;

## 2. Реализация информационных систем

## 2.1 Реализация с помощью средств быстрой разработки

## 2.2 Реализация с помощью в СУБД SQL Server 2014

## 2.3 Создание пользовательского интерфейса для работы с базой данных в Delphi 7

## ЗАКЛЮЧЕНИЕ

В курсовой работе можно подвести итоги. Поставленные задачи были успешно выполнены в полном объёме.

В процессе работы была выполнена следующая цель: автоматизирован контроль разработки Web-сайтов группой разработчиков при помощи прикладного программного обеспечения. Цель считаю полностью сделанной.

Для выполнение данной цели были разработаны и решены следующие задачи:

1. Написана пояснительная записка.
2. Продумана тема курсовой работы, проведена декомпозиция на составляющие блоки будущей информационной системы.
3. Собрана дополнительная информация про работу систем контроля версий, нужная для начала работы по данному проекту.
4. Создана схема и проведенная нормализация БД.
5. Проведено развёртывание СУБД и создание собственно БД
6. Внесены тестовый набор данных и постоянные значения в БД
7. Продуманы внешний вид, дизайн и интерфейс программы
8. Проведено развёртывание IDE, подключены необходимые модули и библиотеки
9. Подключена БД. Созданы системы регистрации и авторизации.
10. Обеспечены хранение, ввод и получение данных о пользователях и их проектах.
11. Создана система хранения собственно кодовой базы проектов.
12. Для всей системы хранения была обеспечена защита, сохранность и целостность данных, что сильно повысило общую безопасность системы.
13. Сделана возможность ввода и вывода любых файлов проектов в любой момент.
14. Осуществлена возможность обновления данных.
15. Для администраторов сети сделана возможность выполнения статистических и пользовательских запросов с их последующей обработкой.
16. Сделан учёт всех пользователей и их проектов.
17. Полностью и пошагово задокументированная проведённая работа.

# СПИСОК ИСТОЧНИКОВ

1. ГОСТ 2.104 ЕСКД для всех специальностей.
2. ГОСТ 7.1-2003 Система стандартов по информации.
3. ГОСТ 19.781-90 «Единая система программной документации. Программное обеспечение систем обработки информации. Термины и определения».