

Министерство цифрового развития, связи и массовых коммуникаций РФ
Федеральное государственное бюджетное образовательное учреждение высшего
образования

Уральский технический институт связи и информатики (филиал) ФГБОУ ВО
"Сибирский государственный университет телекоммуникаций и информатики" в г.
Екатеринбурге (УрТИСИ СибГУТИ)



Уральский технический
институт связи
и информатики



Информационные
системы и
технологии

КАФЕДРА
Информационных систем и технологий
(ИСТ)

ОТЧЕТ

По дисциплине
«Сетевое программирование»
Практическое занятие №4
«Работа с базами данных»

Выполнил:

студент гр. ПЕ-116

Хлебникова Е.А.

Проверил:

преподаватель

Бурумбаев Д.И.

Екатеринбург, 2024

1 Цель работы:

1.3 Научиться работать с базами данных;

1.4 Закрепить знания по теме «Основы работы с базами данных».

2 Перечень оборудования:

2.1 Персональный компьютер;

2.2 СУБД;

2.3 Visual Studio Code.

3.Ход работы:

3.1 Перед началом выполнения работы необходимо ознакомиться с материалами, представленными в приложении А.

3.2 После изучения теоретического материала необходимо выполнить индивидуальное задание, которое получается у преподавателя.

Индивидуальное задание:

Разработка приложения для автоматизации работы информационного агентства

Основная задача любой информационной компании – поиск и публикация информации в виде статей. Нетрудно выделить три основные группы пользователей системы: читатель, главный редактор, журналист. Деятельность каждого из них показана на диаграмме вариантов использования, представленная на рисунке 1.



Рисунок 1 – Диаграмма вариантов использования

Можно выделить следующие основные переменные данной предметной

области:
-Журналисты;
-Статьи.

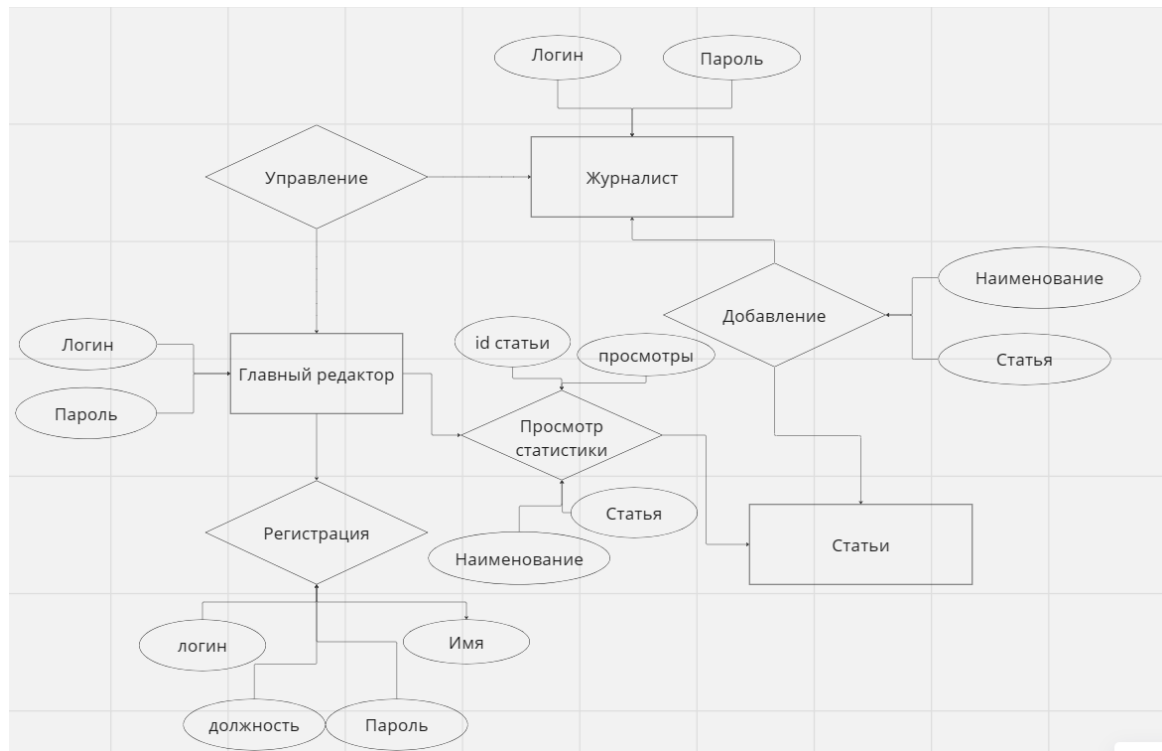


Рисунок 2 – ER-диаграмма

Используя эти данные создаем базу данных и таблицы, которые помогут нам в работе, для работы над индивидуальным заданием.

Создание базы данных:

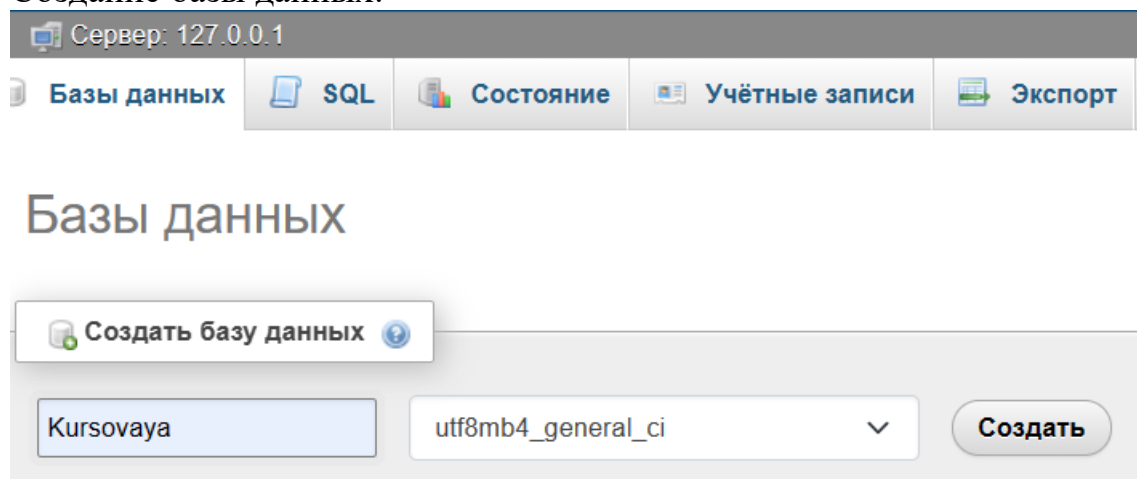


Рисунок 3 – Создание базы данных

Создаем таблицы:

SQL-запрос для создание первой таблицы users с полями id, имя, должность, почта(логин) и пароль:

```
CREATE TABLE users (
    id INT(11) NOT NULL AUTO_INCREMENT,
    name VARCHAR(50) COLLATE utf8mb4_unicode_ci NOT NULL,
    position VARCHAR(50) COLLATE utf8mb4_unicode_ci,
    email VARCHAR(50) COLLATE utf8mb4_unicode_ci NOT NULL,
    password VARCHAR(200) COLLATE utf8mb4_unicode_ci NOT
NULL,
    PRIMARY KEY (id),
    UNIQUE KEY email (email)
) ENGINE=INNODB AUTO_INCREMENT=4 DEFAULT
CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
```

SQL-запрос для создание второй таблицы users с полями id, название публикации, статья , время публикации и посещения:

```
CREATE TABLE combined_table (
    id INT(11) NOT NULL AUTO_INCREMENT,
    title VARCHAR(100) NOT NULL,
    content TEXT NOT NULL,
    published_date DATE NOT NULL,
    views INT(11) NOT NULL DEFAULT 0,
    PRIMARY KEY (id)
) ENGINE=INNODB AUTO_INCREMENT=4 DEFAULT
CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
```

В итоге мы получили готовую базу данных с 2 таблицами:

Таблица	Действие	Строки	Тип	Сравнение	Размер	Фрагментировано
<input type="checkbox"/> combined_table	Обзор Структура Поиск Вставить Очистить Удалить	1	InnoDB	utf8mb4_unicode_ci	16,0 КиБ	-
<input type="checkbox"/> users	Обзор Структура Поиск Вставить Очистить Удалить	4	InnoDB	utf8mb4_unicode_ci	32,0 КиБ	-
2 таблицы	Всего	5	InnoDB	utf8mb4_general_ci	48,0 КиБ	0 Байт

Рисунок 4 – Готовая база данных

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
<input type="checkbox"/> 1	id	int(11)			Нет	Нем		AUTO_INCREMENT	Изменить Удалить Ещё
<input type="checkbox"/> 2	name	varchar(50)	utf8mb4_unicode_ci		Нет	Нем			Изменить Удалить Ещё
<input type="checkbox"/> 3	position	varchar(50)	utf8mb4_unicode_ci		Да	NULL			Изменить Удалить Ещё
<input type="checkbox"/> 4	email	varchar(50)	utf8mb4_unicode_ci		Нет	Нем			Изменить Удалить Ещё
<input type="checkbox"/> 5	password	varchar(200)	utf8mb4_unicode_ci		Нет	Нем			Изменить Удалить Ещё

Рисунок 5 – Первая таблица «user»

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
<input type="checkbox"/> 1	id	int(11)			Нет	Нет		AUTO_INCREMENT	Изменить Удалить Ещё
<input type="checkbox"/> 2	title	varchar(100)	utf8mb4_unicode_ci		Нет	Нет			Изменить Удалить Ещё
<input type="checkbox"/> 3	content	text	utf8mb4_unicode_ci		Нет	Нет			Изменить Удалить Ещё
<input type="checkbox"/> 4	published_date	date			Нет	Нет			Изменить Удалить Ещё
<input type="checkbox"/> 5	views	int(11)			Нет	0			Изменить Удалить Ещё

Рисунок 6 – Вторая таблица «combined_table»

4. Контрольные вопросы:

4.1 Пояснить термин «реляционная база данных».

Реляционная база данных - это тип базы данных, организованный в соответствии с реляционной моделью данных. В реляционной модели данные представлены в виде таблиц, состоящих из строк и столбцов. Каждая таблица представляет собой отдельный набор данных, а каждая строка в таблице представляет отдельную запись или кортеж, а каждый столбец представляет отдельное поле данных. В реляционных базах данных данные между таблицами связываются посредством ключей или отношений, что позволяет эффективно организовывать, хранить и извлекать данные.

4.2 Для чего необходима система разграничения прав пользователей?

Система разграничения прав пользователей - это механизм, позволяющий администраторам управлять доступом пользователей к данным и функциям системы. Этот механизм позволяет определить, какие пользователи имеют доступ к каким данным и какие действия они могут выполнять с этими данными. Система разграничения прав позволяет обеспечить безопасность данных, предотвращая несанкционированный доступ и изменение данных, а также ограничивая пользовательские привилегии в соответствии с их ролями и обязанностями.

4.3 Какие основные привилегии разрешают доступ к БД?

Основные привилегии, разрешающие доступ к базе данных, включают:

- **SELECT:** Позволяет выполнять операции чтения данных из таблиц.
- **INSERT:** Разрешает добавление новых записей в таблицу.
- **UPDATE:** Позволяет изменять существующие записи в таблице.
- **DELETE:** Разрешает удаление записей из таблицы.
- **CREATE:** Позволяет создавать новые таблицы и базы данных.
- **DROP:** Разрешает удалять существующие таблицы и базы данных.
- **ALTER:** Позволяет изменять структуру существующих таблиц.
- **GRANT OPTION:** Предоставляет пользователю возможность предоставлять привилегии другим пользователям.

4.4 Для чего применяются ключевые поля?

Ключевые поля применяются для уникальной идентификации каждой

записи в таблице. Они обеспечивают уникальность и целостность данных, а также позволяют эффективно выполнять операции поиска и сортировки. Ключевые поля могут состоять из одного или нескольких столбцов и обычно выбираются из основных атрибутов, которые наиболее четко идентифицируют каждую запись.

4.5 Сколько ключевых полей и полей ссылок может содержать таблица?

В таблице может быть только одно ключевое поле или комбинация полей, образующих составной ключ. Что касается полей ссылок, то в таблице может быть неограниченное количество полей, связывающих ее с другими таблицами.

4.6 Каким способом можно поменять структуру таблицы без применения команды ALTER?

Структуру таблицы можно изменить без применения команды ALTER с помощью инструментов администрирования базы данных или средств для работы с моделями данных, таких как MySQL Workbench или PHPMyAdmin. Эти инструменты обеспечивают графический интерфейс для добавления, изменения или удаления столбцов, индексов и ограничений таблицы.

4.7 В чем различие операторов INSERT и LOAD DATA?

Оператор INSERT используется для добавления новых записей в таблицу, где каждое значение указывается явно в команде INSERT. В то время как оператор LOAD DATA используется для загрузки данных из внешнего файла или источника непосредственно в таблицу базы данных, что может быть более эффективным для загрузки больших объемов данных.