

# СОДЕРЖАНИЕ

<b>ВВЕДЕНИЕ</b>	<b>4</b>
<b>1 Аналитический раздел</b>	<b>5</b>
1.1 Определение базы данных . . . . .	5
1.2 Определение СУБД . . . . .	5
1.3 Классификация СУБД по модели данных . . . . .	6
1.4 Классификация СУБД по способу доступа к базе данных . .	7
1.5 Классификация СУБД по организации хранения данных . .	8
1.6 Формализация данных . . . . .	8
1.7 Формализация пользователей . . . . .	9
1.8 Анализ существующих решений . . . . .	9
1.9 Вывод . . . . .	10
<b>2 Конструкторский раздел</b>	<b>11</b>
2.1 Формализация сущностей системы . . . . .	11
2.2 Ролевая модель . . . . .	13
2.3 Разработка триггера и функции . . . . .	13
2.4 Вывод . . . . .	14
<b>СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ</b>	<b>15</b>

# ВВЕДЕНИЕ

Курсовая работа[?] — практическая работа по всему курсу соответствующей научной дисциплины, выполняемая студентом в качестве формы контроля полученных и усвоенных знаний. Ежегодно десятки курсовых проектов по предмету «Базы данных» создаются студентами МГТУ им. Н.Э. Баумана.

Электронное хранение, обработка и анализ информации о курсовых проектах могут быть полезными для различных систем, например автоматизированной проверки на плагиат для выявления студентов, недобросовестно выполняющих практическую работу.

Целью данной работы является разработка программного обеспечения для хранения, редактирования и удаления данных о курсовых проектах по предмету «Базы данных».

Для достижения поставленной цели необходимо выполнить следующие задачи:

- 1) проанализировать существующие методы представления данных и определить подходящий для выполнения работы вариант;
- 2) провести анализ существующих систем управления базами данных и определить подходящий для выполнения работы вариант;
- 3) спроектировать базу данных, описать ее структуру сущностей и связей между ними;
- 4) реализовать описанную базу данных и программное обеспечение для работы с ней.

# 1 Аналитический раздел

В данном разделе представлен анализ существующих систем управления базами данных, проведена формализация информации, подлежащей хранению.

## 1.1 Определение базы данных

**База данных**[1] — это упорядоченный набор структурированных данных, хранящихся в электронном виде.

По способу применения базы данных разделяют на два типа.

- 1) **OLAP**(online analytical processing) — система, используемая для обработки больших объемов данных.
- 2) **OLTP**(online transactional processing) — система, основной упор которой делается на быструю обработку запросов, операции в режиме реального времени.

В связи с необходимостью работать с базой данных в режиме реального времени в разрабатываемом программной обеспечении будет использована технология OLTP.

## 1.2 Определение СУБД

**Система управления базами данных (СУБД)** — это приложение, обеспечивающее создание, хранение, обновление и поиск информации в базе данных.

Существует множество классификация СУБД, использующие разные признаки классификации. Далее приведены некоторые из них.

## 1.3 Классификация СУБД по модели данных

### Дореляционные СУБД

- 1) **Инвертированные списки.** Базы данных, основанные на использовании инвертированных списков представляют собой совокупность файлов, содержащих записи. Инвертированный список — это организованный в вид списка специального вида индекс файла, позволяющий получить всю совокупность указателей на записи, которым соответствует заданное значение ключа индексации. В общем случае ограничения целостности базы данных отсутствуют. В некоторых системах поддерживаются ограничения уникальности значений некоторых полей, но в основном все возлагается на прикладную программу.
- 2) **Иерархическая модель.** Иерархические модели имеют древовидную структуру, где каждому узлу соответствует один сегмент, представляющий собой запись (кортеж полей) базы данных. Каждому сегменту может соответствовать несколько дочерних сегментов, однако запись-потомок должна иметь в точности одного предка.
- 3) **Сетевые СУБД.** Сетевой подход к организации данных является расширением иерархического. На формирование связи ограничений не накладывается, в отличие от иерархической модели, предполагающей ровно одного предка у записи-потомка.

### Реляционные СУБД

Реляционная модель предполагает организацию данных в виде таблиц (отношений), содержащих информацию о сущностях. Каждая запись таблицы содержит уникальный индекс (ключ), используемый для поиска связанных данных в разных таблицах.

Реляционная модель состоит из трех частей:

- 1) **структурная часть** фиксирует, что база данных использует только одну структуру данных —  $n$ -арное отношение;

- 2) **целостная часть** описывает ограничения, накладываемые на отношения реляционной модели;
- 3) **манипуляционная часть** описывает два эквивалентных способа манипулирования реляционными данными — реляционную алгебру и реляционное исчисление.

## Постреляционные СУБД

Постреляционная модель представляет собой обобщение реляционной модели. Она допускает многозначные поля таблиц, каждое из которых рассматривается как самостоятельное отношение, встроенное в главное отношение.

### 1.4 Классификация СУБД по способу доступа к базе данных

- 1) **Файл-серверные СУБД.** Значительная часть вычислений выполняется на стороне клиента, сервер отвечает только за извлечение данных из файлов и отправку пользователю.
- 2) **Клиент-серверные СУБД.** Основная вычислительная нагрузка ложится на сервер базы данных, задача клиента заключается в предварительной обработке данных и организации доступа к серверу.
- 3) **Встраиваемые СУБД.** СУБД представляет собой библиотеку, позволяющую структурировать и хранить большие объемы данных на локальной машине.
- 4) **Сервис-ориентированные СУБД.** База данных представляет собой хранилище сообщений и метаданных о сервисах и очередях сообщений.
- 5) **Другие СУБД.**

## 1.5 Классификация СУБД по организации хранения данных

- 1) **Локальные СУБД.** Все части СУБД располагаются на одной машине
- 2) **Распределенные СУБД.** Части СУБД распределены на двух или более машинах.

## 1.6 Формализация данных

База данных состоит из следующих сущностей:

- 1) таблица курсовых проектов Project;
- 2) таблица литературных источников Source;
- 3) таблица академических групп Group;
- 4) таблица студентов Student;
- 5) таблица тем курсовых проектов Theme;

На рисунке 1 представлена ER-диаграмма сущностей в нотации Чена.

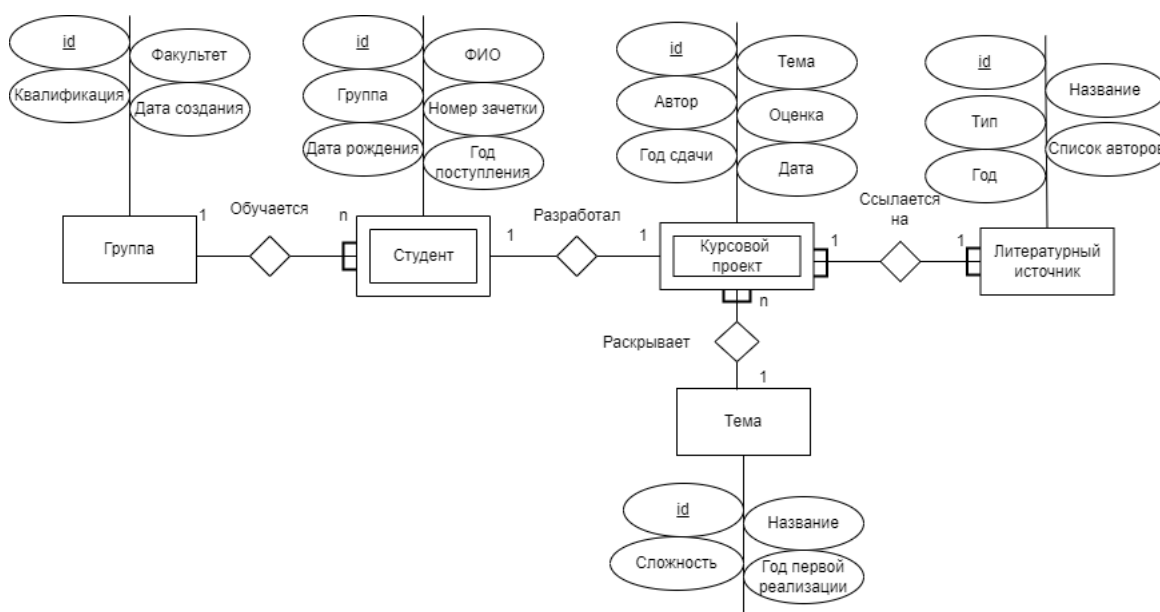


Рисунок 1 – ER-диаграмма в нотации Чена

## 1.7 Формализация пользователей

В системе присутствуют три уровня пользователей.

- 1) **Студент** — пользователь, обладающий возможностями просмотра сущностей академических групп и тем курсовых проектов.
- 2) **Преподаватель** — пользователь, обладающий возможностями просмотра всех сущностей, перечисленных в разделе «Формализация данных».
- 3) **Администратор** — пользователь, обладающий возможностями изменения сущностей и полей базы данных, просмотра всех сущностей.

## 1.8 Анализ существующих решений

Среди уже существующих проектов были выделены 3 аналога, частично решающие поставленную задачу. Их сравнительный анализ представлен в таблице 1.

Для сравнения были выбраны следующие критерии:

- 1) группы — возможность просмотра информации об академических группах студентов;
- 2) антиплагиат — невозможность приобретения готового проекта другого человека;
- 3) темы — возможность просмотра тем уже существующих работ для поиска вдохновения.

Таблица 1 – Существующие решения поставленной задачи

Название проекта	Группы	Антиплагиат	Темы
ЭУ МГТУ[?]	+	+	-
Studynote [?]	-	+	+
Workspay [?]	-	-	+

Из таблицы можно сделать вывод, что ни один из перечисленных аналогов не обладает функционалом, удовлетворяющим требованиям или обладает лазейками для недобросовестного выполнения курсовой работы.

Создаваемое программное обеспечение будет предоставлять описанный функционал, являясь некоммерческим продуктом и не предоставляя возможности для плагиата.

## 1.9 Вывод

В данном разделе представлен анализ существующих систем управления базами данных, проведена формализация информации, подлежащей хранению.



## 2 Конструкторский раздел

В данном разделе рассматриваются сущности проектируемой базы данных, спроектированы триггер и функция.

### 2.1 Формализация сущностей системы

На основе ER-диаграммы (рис. 1) спроектированы таблицы базы данных.

#### Таблица Group

Содержит информацию об академических группах и включает следующие поля

- 1) id — идентификатор группы, являющийся первичным ключом;
- 2) faculty — название факультета, символьный тип;
- 3) qualif — квалификация (бакалавр, специалист и т.д.), символьный тип;
- 4) date — дата создания группы, тип-дата.

#### Таблица Student

Содержит информацию о студентах и включает следующие поля:

- 1) id — идентификатор студента, являющийся первичным ключом;
- 2) group — группа студента, внешний ключ;
- 3) FIO — ФИО студента, символьный тип;

- 4) record — номер зачетной книжки, целочисленный тип;
- 5) birth — дата рождения, тип-дата;
- 6) admission — год поступления, целочисленный тип.

## Таблица Theme

Содержит информацию о темах курсовых проектов и включает следующие поля:

- 1) id — идентификатор темы, являющийся первичным ключом;
- 2) name — название темы, символьный тип;
- 3) complex — сложность темы по 10-бальной шкале, целочисленный тип;
- 4) first — год первой реализации темы, целочисленный тип;

## Таблица Source

Содержит информацию о литературных источниках курсовых проектов и включает следующие поля:

- 1) id — идентификатор источника, являющийся первичным ключом;
- 2) name — название источника, символьный тип;
- 3) type — тип источника (учебник, статья и т.д.), символьный тип;
- 4) author — список авторов, символьный тип;
- 5) year — год создания источника, целочисленный тип;

## Таблица Project

Содержит информацию о курсовых проектах и включает следующие поля:

- 1) id — идентификатор проекта, являющийся первичным ключом;

- 2) theme — тема проекта, внешний ключ;
- 3) author — автор проекта, внешний ключ;
- 4) mark — оценка проекта, целочисленный тип;
- 5) date — дата сдачи проекта, тип-дата;

## 2.2 Ролевая модель

Для обеспечения работы пользователей с системой управления базами данных, выделена следующая ролевая модель.

### **Студент StUser**

- 1) SELECT над таблицей Theme.
- 2) SELECT над таблицей Group.

### **Преподаватель TeUser**

SELECT над таблицами Theme, Group, Student, Source, Project.

### **Администратор AdmUser**

Все права над Theme, Group, Student, Source, Project.

## 2.3 Разработка триггера и функции

В разрабатываемой базе данных предусмотрена функция, выполняющая ...

Также в системе представлен INSERT триггер, который при добавлении в базу нового курсового проекта, проверяет, что год его реализации больше или равен году первой реализации соответствующей темы. В противном случае триггер меняет год первой реализации данной темы на год создания проекта.

## 2.4 Вывод

В данном разделе рассматриваются сущности проектируемой базы данных, спроектированы триггер и функция.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- [1] DATABASE SYSTEMS The Complete Book Second Edition Hector Garcia-Molina Jeffrey D. Ullman.