МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

(ФГБОУ ВО «ВГТУ», ВГТУ)

ФАКУЛЬТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И КОМПЬЮТЕРНОЙ БЕЗОПАСНОСТИ

КАФЕДРА КОМПЬЮТЕРНЫХ ИНТЕЛЛЕКТУАЛЬНЫХ ТЕХНОЛОГИЙ ПРОЕКТИРОВАНИЯ

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №7

По дисциплине: «Системы хранения и обработки данных»

Тема: «Разработка логической структуры базы данных»

Выполнил работу студент группы мИИВТ-231 Костюков К.А.

Принял: Короленко В.В.

Защищена \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Воронеж 2023

**Цель работы:**

изучить основы логического проектирования базы данных, освоить процесс разработки логической структуры базы данных и построения диаграммы «сущность-связь».

**Основные задачи:**

* определение сущности для проекта в соответствии с индивидуальным заданием и их атрибуты;
* выделение ключевых атрибутов;
* определение связей между сущностями и типов связей;
* построение диаграммы сущность-связь для отображения логической структуры базы данных.

**Ход работы:**

Перед основной работой были изучены основные понятия теории баз данных, основные модели данных, а также средство для разработки диаграмм draw.io После этого для 10-го варианта лабораторной (Учет результатов сдачи вступительных экзаменов) были определены основные сущности для разрабатываемой БД, которые логически подходят к данной теме:

Абитуриент – список абитуриентов

Экзаменатор – список экзаменаторов

Экзамен – сопоставление списка абитуриентов и экзаменаторов в список экзаменов с датой проведения, обозначением предмета.

Результат экзамена - сопоставление списка абитуриентов и экзаменаторов в список экзаменов с датой проведения, обозначением предмета.

Оплата экзаменатора - учета оплаты, которую экзаменатор получает за проведение экзаменов.

Далее были выделены ключевые атрибуты сущностей (первичные и внешние ключи):

Таблица "Абитуриент"

Атрибуты:

1. Абитуриент\_ID (первичный ключ)
2. Имя
3. Фамилия
4. Дата рождения
5. Адрес

Таблица "Экзаменатор"

Атрибуты:

1. Экзаменатор\_ID (первичный ключ)
2. Имя
3. Фамилия
4. Специализация
5. Уровень оплаты

Таблица "Экзамен"

Атрибуты:

1. Экзамен\_ID (первичный ключ)
2. Название
3. Дата проведения
4. Аудитория

Таблица "Результат экзамена"

Атрибуты:

1. Результат\_ID (первичный ключ)
2. Оценка
3. Дата сдачи
4. Абитуриент\_ID (внешний ключ, связь с таблицей "Абитуриент")
5. Экзамен\_ID (внешний ключ, связь с таблицей "Экзамен")
6. Экзаменатор\_ID (внешний ключ, связь с таблицей "Экзаменатор")

Таблица "Оплата экзаменатора"

Атрибуты:

1. Оплата\_ID (первичный ключ)
2. Сумма оплаты
3. Дата оплаты
4. Экзаменатор\_ID (внешний ключ, связь с таблицей "Экзаменатор")

Затем, для сущностей были определены связи:

- Между Абитуриентом и Результатом экзамена: Один ко многим (один абитуриент может иметь много результатов).

- Между Экзаменатором и Результатом экзамена: Один ко многим (один экзаменатор может иметь много результатов).

- Между Экзаменом и Результатом экзамена: Один ко многим (один экзамен может иметь много результатов).

- Многие ко многим между Абитуриентом и Экзаменом: Один абитуриент может сдавать много экзаменов, и экзамен может быть сдан многими абитуриентами.

- Между Экзаменатором и Оплатой экзаменатора: Один ко многим (один экзаменатор может иметь много записей об оплате).

- Между Оплатой экзаменатора и Результатом экзамена: Один к одному (каждая запись об оплате соответствует результату одного экзамена).

После необходимо избавиться от связей «многие ко многим» с помощью введения дополнительных сущностей. Добавление таблицы "Участие в экзамене", содержащей внешние ключи ID абитуриента и ID экзамена решает эту задачу. Теперь у нас есть две связи "один ко многим" вместо связи "многие ко многим".

Далее с использованием средства для разработки диаграмм draw.io была построена диаграмма сущность-связь для отображения логической структуры проектируемой базы данных (Рисунок 1).

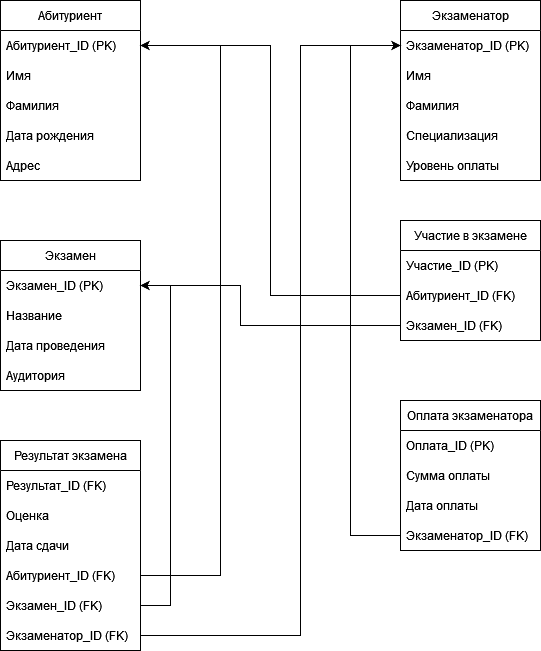


Рисунок 1 – Диаграмма

В базе данных, в которой учтены все сущности, есть следующие связи:

1. Связь "Один ко многим" между Абитуриентом и Участием в экзамене:

- Один абитуриент может участвовать в нескольких экзаменах (но в одном экзамене несколько абитуриентов не участвует).

- Абитуриент\_ID в таблице "Участие в экзамене" является внешним ключом, связанным с первичным ключом Абитуриент\_ID в таблице "Абитуриент".

2. Связь "Один ко многим" между Экзаменом и Участием в экзамене:

- Один экзамен может иметь участие нескольких абитуриентов.

- Экзамен\_ID в таблице "Участие в экзамене" является внешним ключом, связанным с первичным ключом Экзамен\_ID в таблице "Экзамен".

3. Связь "Один ко многим" между Экзаменатором и Результатом экзамена:

- Один экзаменатор может быть ответственным за результаты нескольких экзаменов.

- Экзаменатор\_ID в таблице "Результат экзамена" является внешним ключом, связанным с первичным ключом Экзаменатор\_ID в таблице "Экзаменатор".

4. Связь "Один к одному" между Участием в экзамене и Результатом экзамена:

- Каждое участие в экзамене соответствует одному результату экзамена.

- (Абитуриент\_ID, Экзамен\_ID) в таблице "Участие в экзамене" является составным первичным ключом, абсолютно уникально определяющим участие в конкретном экзамене.

5. Связь "Один ко многим" между Экзаменатором и Оплатой экзаменатора:

- Один экзаменатор может иметь несколько записей об оплате.

- Экзаменатор\_ID в таблице "Оплата экзаменатора" является внешним ключом, связанным с первичным ключом Экзаменатор\_ID в таблице "Экзаменатор".

**Вывод**: в данной работе были изучены основы логического проектирования базы данных, освоен процесс разработки логической структуры базы данных и построения диаграммы «сущность-связь».

**Контрольные вопросы**

1. Дайте определения следующим понятиям: данные, база данных, СУБД, ведение базы данных.

Данные представляют собой факты, информацию или структурированную информацию, которая может быть записана, сохранена и обрабатываться компьютерами. Данные могут быть представлены в различных форматах, включая текст, числа, изображения и звуки.

База данных - это организованная и структурированная коллекция данных, хранящая информацию в компьютере или системе. Она состоит из таблиц, которые содержат записи или строки, представляющие конкретные сущности или объекты, а также атрибуты или столбцы, которые описывают свойства этих сущностей.

СУБД (Система Управления Базами Данных) — это программное обеспечение, которое обеспечивает управление базой данных. Она предоставляет функционал для создания, хранения, обновления и извлечения данных из базы данных. СУБД также отвечает за обеспечение безопасности, целостности и доступности данных.

Ведение базы данных означает управление и обслуживание базы данных. Это включает в себя создание, изменение и удаление структуры таблиц, добавление и удаление данных, выполнение запросов, обеспечение безопасности данных и резервное копирование информации. Ведение базы данных также включает мониторинг и оптимизацию производительности базы данных.

1. В чем отличие данных от информации?

Отличие данных от информации заключается в их обработке и значении. В данные входит сырое и необработанное представление фактов или фигурирующих значений, тогда как информация представляет собой осмысленное и полезное содержание, выведенное из этих данных после их анализа и интерпретации. Другими словами, данные представляют собой кусок информации, который еще не прошел стадию интерпретации и контекстуализации.

1. В чем отличие базы данных от банка данных и СУБД?

База данных - это набор структурированных данных, хранящихся на компьютере. Банк данных - это физическое хранилище данных, которое может включать несколько баз данных. СУБД - это программное обеспечение, которое позволяет управлять доступом, сохранением и обработкой данных в базе данных.

1. Назовите основные компоненты банка данных и их назначение.

Банк данных включает следующие основные компоненты: БД, СУБД, администратора базы данных, словарь данных, вычислительную систему, обслуживающий персонал.

База данных – именованная и связная совокупность данных предметной области, как правило – табличный вид. Обновляемая, видоизменяемая, подвергается хранению, архивированию и удалению.

Система управления базой данных - это специальный пакет программ, посредством которого реализуется централизованное управление базой данных и обеспечивается доступ к данным.

Администратор базы данных - это лицо (или группа лиц), реализующее управление базой данных.

Словарь данных- важная структура банка данных, в которой хранится централизованная информация обо всех ресурсах банка данных.

Иногда присутствуют архивы данных и при их наличии есть СУБД архива данных.

1. Классифицируйте АИС по типу хранимых данных.

Автоматизированные информационные системы могут быть классифицированы по типу хранимых данных на следующие категории:

1. Реляционные АИС (РАИС): Они основаны на реляционной модели данных и используют таблицы для хранения данных, а также связи между таблицами для установления отношений.
2. Иерархические АИС (ИАИС): Они структурируют данные в виде иерархической структуры, где каждый элемент имеет родителя и дочерние элементы.
3. Сетевые АИС (САИС): Они похожи на иерархические АИС, но позволяют установление сложных связей и отношений между элементами.
4. Объектно-ориентированные АИС (ООАИС): Они используют объектную модель для организации данных, где данные представлены в виде объектов с методами и свойствами.
5. Графовые АИС (ГАИС): Они представляют данные в виде графа, где узлы представляют сущности, а ребра - связи между ними.
6. Многомерные АИС (МАИС): Они используются для анализа данных, где информация организуется в многомерные структуры, такие как кубы данных.
7. Что понимается под трехуровневой архитектурой ANSI/SPARC?

Трехуровневая архитектура ANSI/SPARC - это модель проектирования и организации баз данных, состоящая из трех уровней: внешнего, концептуального и внутреннего. Внешний уровень описывает данные с точки зрения пользователей, концептуальный уровень определяет общую структуру данных для всей организации, а внутренний уровень определяет способ хранения и доступа к данным. Эта архитектура позволяет разделить представление данных от их физической реализации и облегчает модификацию и расширение базы данных без влияния на пользователей.

1. Дайте определения внешней схеме БД, концептуальной схеме БД, внутренней схеме БД.

Внешняя схема БД – это представление данных организации или пользователями базы данных. Она определяет, как данные могут быть представлены и доступны на уровне приложений. Внешняя схема БД фокусируется на конкретных потребностях пользователей и обеспечивает удобный интерфейс для работы с данными.

Концептуальная схема БД - это репрезентация данных, представляющая общее представление о структуре и связях между данными. Концептуальная схема определяет основные сущности, их атрибуты и связи между ними. Эта схема является независимой от платформы и служит основой для создания внешних и внутренних схем.

Внутренняя схема БД - это физическое представление данных в БД, которое определяет, как данные хранятся и обрабатываются на физическом уровне. Внутренняя схема определяет структуру данных, методы доступа и организацию данных на диске. Эта схема обычно невидима для конечных пользователей и ориентирована на оптимизацию производительности системы.

1. Каковы особенности иерархической модели организации данных?

Основные особенности иерархической модели организации данных:

- Иерархическая структура данных, организованная в виде древовидной иерархии.

- Узлы данных имеют родительские и дочерние узлы, создавая иерархию.

- Организация данных по принципу «один-к-многим».

- Быстрый доступ к данным, благодаря ссылкам между узлами.

- Хорошо подходит для организации данных с ясной иерархией и зависимостями между ними.

1. Каковы особенности сетевой модели организации данных?

Сетевая модель организации данных является одной из классических моделей баз данных. Она основана на структуре "родитель-потомок", где каждый элемент данных имеет одного родителя и может иметь несколько потомков.

Основные особенности:

* 1. Гибкость: сетевая модель позволяет представлять сложные взаимосвязи между данными, такие как многоуровневые иерархии, вложенность и связи многие-ко-многим.
  2. Быстрота доступа: в сетевой модели данные могут быть связаны напрямую друг с другом, что позволяет быстро и эффективно получать информацию из различных связанных записей.
  3. Поддержка множественного доступа: Сетевая модель позволяет множеству пользователей одновременно получать доступ к данным и выполнять операции чтения и записи.
  4. Несогласованность структуры: Изменение структуры данных в сетевой модели требует обновления всех связанных записей, что может привести к несогласованности данных.

1. Каковы особенности многомерной модели организации данных?

Многомерная модель организации данных представляет данные в виде многомерных кубов с использованием измерений и иерархической структуры. Она позволяет эффективно анализировать и организовывать данные, учитывая различные характеристики и связи между ними.

1. Каковы особенности постреляционной модели организации данных?

Постреляционная модель организации данных, также известная как NoSQL (Not only SQL), предлагает гибкую схему данных, которая позволяет хранить и обрабатывать неструктурированные и полуструктурированные данные. В постреляционных базах данных отсутствует жесткая схема и предлагается распределенная архитектура, поддерживающая горизонтальное масштабирование. Это позволяет эффективно обрабатывать большие объемы данных и обеспечивает высокую производительность при работе с нагруженными системами. Однако постреляционные базы данных могут быть менее подходящими для сложных запросов и операций, которые требуют жесткой связи между данными.

1. Что относится к неструктурированным данным?

Неструктурированные данные относятся к информации, которая не имеет четко определенной и организованной структуры. Они не соответствуют традиционным моделям данных, таким как таблицы и столбцы в реляционных базах данных. Неструктурированные данные могут быть представлены в различных форматах, таких как текстовые файлы, видео- и аудиозаписи, изображения, документы PDF, социальные медиа-публикации, веб-страницы и другие. Обработка неструктурированных данных в постреляционных базах данных может включать технологии, такие как текстовый анализ, машинное обучение и распознавание образов.

1. В чем преимущество использования колоночной СУБД по сравнению с реляционной?

Преимущества использования колоночной СУБД по сравнению с реляционной включают лучшую производительность чтения данных, более эффективное сжатие данных, быстрые аналитические запросы и поддержку больших объемов данных.

1. Каким образом осуществляется связь между таблицами в реляционной СУБД?

В реляционной СУБД связь между таблицами осуществляется с помощью внешних ключей. Внешний ключ представляет собой поле или набор полей в одной таблице, которые связаны с первичным ключом в другой таблице. Это позволяет установить отношение или связь между данными в разных таблицах.

1. Каким образом на этапе проектирования решается проблема дублирующих записей в таблице?

Проблему можно решить использованием первичных ключей. Каждая таблица должна иметь первичный ключ, который уникально идентифицирует каждую запись в таблице. Использование первичных ключей помогает предотвратить вставку дублирующих записей.