Санкт-Петербургское государственное бюджетное

профессиональное образовательное учреждение

«Ижорский колледж»

КУРСОВАЯ РАБОТА

ТЕМА: «Смотровая площадка для стартапов – с идеями и инвестициями»

по ПМ.11 МДК.11.01 Технология разработки и защиты баз данных

Выполнил

обучающийся группы 232с

Вербицкий Максим Дмитриевич

Проверил

преподаватель спец. дисциплин

специальности 09.02.07

Информационные системы и программирование

Венедиктов Д.В.

Санкт-Петербург, 2025ЗАДАНИЕ ДЛЯ ВЫПОЛНЕНИЯ КУРСОВОЙ РАБОТЫ

1. Изучить теоретические основы технологии разработки баз данных, включая этапы проектирования, нормализации, реализации и тестирования.

2. Рассмотреть методы и средства защиты баз данных от несанкционированного доступа, атак и вредоносных программ.

3. Провести анализ существующих систем управления базами данных (СУБД) с точки зрения их функциональности, безопасности и удобства использования.

4. Разработать проект базы данных для конкретной предметной области, учитывая требования к безопасности и конфиденциальности данных.

5. Реализовать разработанный проект в выбранной СУБД.

6. Протестировать разработанную базу данных на предмет соответствия требованиям безопасности и надёжности.

7. Оценить эффективность разработанной системы защиты базы данных и предложить рекомендации по её улучшению.

8. Оформить результаты работы в виде курсовой работы, включающей введение, основную часть, заключение, список использованных источников и приложения.

В основной части курсовой работы необходимо подробно рассмотреть следующие вопросы:

* основные этапы разработки базы данных;
* методы обеспечения целостности и безопасности данных;
* механизмы аутентификации и авторизации пользователей;
* шифрование данных и защита от вредоносного ПО;
* особенности реализации проекта базы данных в выбранной СУБД;
* результаты тестирования и оценка эффективности системы защиты.

Для выполнения задания рекомендуется использовать современные научные источники, а также практический опыт разработки и защиты баз данных.

ПЛАН-ГРАФИК ВЫПОЛНЕНИЯ КУРСОВОЙ РАБОТЫ

|  |  |  |
| --- | --- | --- |
| Дата | Содержание работ | Отметка о выполнении |
| 10.09 | Выбор темы курсовой работы |  |
| 17.09 | Знакомство с методическими рекомендациями по выполнению курсовой работы |  |
| 24.09 | Знакомство с планом-графиком выполнения курсовой работы |  |
| 08.10 | Работа с нормативно-правовыми документами, учебной литературой |  |
| 22.10 | Анализ предметной области. Постановка задачи |  |
| 29.10 | Составление введения к курсовой работе |  |
| 05.11 | Описание предметной области и функции решаемых задач |  |
| 12.11 | Выбор средств для выполнения курсовой работы. Выбор среды разработки |  |
| 19.11 | Концептуально-логическое проектирование. Составление ER-диаграммы |  |
| 10.12 | Создание и заполнение базы данных |  |
| 24.12 | Представления в базе данных |  |
| 21.01 | Процедуры в базе данных |  |
| 04.02 | Создание ролей в базе данных |  |
| 18.02 | Аутентификация пользователей в базе данных |  |
| 25.02 | Импорт и экспорт базы данных |  |
| 04.03 | Разработка стратегии резервного копирования базы данных |  |
| 11.03 | Тестирование базы данных |  |
| 18.03 | Оптимизация базы данных |  |
| 25.03 | Составление заключения к курсовой работе |  |
| 01.04 | Составление списка источников информации, используемых при выполнении курсовой работы |  |
| 08.04 | Подготовка курсовой работы к защите |  |

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 7](#_Toc180594792)

[ГЛАВА 1. ФУНКЦИОНАЛЬНЫЕ ВОЗМОЖНОСТИ БАЗЫ ДАННЫХ 8](#_Toc180594793)

[1.1. Анализ предметной области 8](#_Toc180594794)

[1.2. Анализ технологий для разработки базы данных 8](#_Toc180594795)

[1.3. Постановка задачи курсовой работы 10](#_Toc180594796)

[1.3.1. Цели курсовой работы 10](#_Toc180594797)

[1.3.2. Задачи курсовой работы 11](#_Toc180594798)

[1.4.1. Функциональные требования к системе 11](#_Toc180594799)

[1.4.2. Нефункциональные требования к системе 11](#_Toc180594800)

[1.5. Выбор программных средств для разработки базы данных 11](#_Toc180594801)

[1.6. Выбор среды для разработки базы данных 14](#_Toc180594802)

[ГЛАВА 2. ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ 17](#_Toc180594803)

[2.1. Концептуально-логическое моделирование 17](#_Toc180594804)

[2.1.1 Концептуальное моделирование 17](#_Toc180594805)

[2.1.2. Логическое моделирование 18](#_Toc180594806)

[2.2. Описание информационных объектов базы данных 20](#_Toc180594807)

[2.3.1. Создание базы данных 25](#_Toc180594808)

[2.3.2. Заполнение базы данных 25](#_Toc180594809)

[2.4. Представления в базе данных 27](#_Toc180594810)

[2.5. Процедуры в базе данных 30](#_Toc180594811)

[2.6. Создание ролей в базе данных 30](#_Toc180594812)

[2.7. Аутентификация пользователей в базе данных 30](#_Toc180594813)

[2.8. Импорт и экспорт базы данных 30](#_Toc180594814)

[2.9. Разработка стратегии резервного копирования базы данных 30](#_Toc180594815)

[2.10.1. Тестирование базы данных 31](#_Toc180594816)

[2.10.2. Оптимизация базы данных 31](#_Toc180594817)

[ЗАКЛЮЧЕНИЕ 32](#_Toc180594818)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 33](#_Toc180594819)

# ВВЕДЕНИЕ

В современном цифровом мире данные представляют собой один из ключевых ресурсов, влияющих на стратегические решения и операционные процессы организаций. Базы данных стали основным инструментом для хранения, обработки и управления информацией, обеспечивая эффективное взаимодействие с данными. Разработка баз данных – это сложный процесс, требующий комплексного подхода, включающего анализ требований, проектирование, реализацию и поддержку.

С учетом растущего объема информации и усложнения бизнес-процессов, значимость качественного проектирования баз данных трудно переоценить. Правильная структура базы данных обеспечивает не только ее функциональность и производительность, но и легкость в управлении данными, что в свою очередь влияет на оперативность принятия решений.

Однако с увеличением объема данных и многообразием технологий, связанных с их хранением, возникает необходимость уделять особое внимание защите данных. Утечки информации и кибератаки могут привести к значительным финансовым потерям и подрыву доверия со стороны клиентов. Поэтому технологии защиты баз данных становятся критически важными для бизнеса всех размеров.

Эта работа сосредоточится на ключевых аспектах разработки и защиты баз данных, включая методы проектирования и элементы безопасности, которые должны быть внедрены для обеспечения надежности и защиты данных в любой организации. Изучение этих тем поможет определить лучшие практики, которые должны применяться для достижения устойчивости и эффективности работы с информацией в условиях динамично меняющейся технологической среды.

# ГЛАВА 1. ФУНКЦИОНАЛЬНЫЕ ВОЗМОЖНОСТИ БАЗЫ ДАННЫХ

## 1.1. Анализ предметной области

Технология разработки и защиты баз данных охватывает широкий спектр процессов, методов и инструментов, необходимых для создания, управления и обеспечения безопасности баз данных. Это область, которая активно развивается в ответ на растущие требования к хранению и защите информации, что делает её актуальной для различных отраслей, таких как финансы, здравоохранение, образование и многие другие.

## 1.2. Анализ технологий для разработки базы данных

Разработка баз данных включает в себя использование различных технологий и инструментов, которые помогают осуществлять проектирование, реализацию, управление и оптимизацию баз данных. Рассмотрим основные технологии и подходы, которые широко применяются в этой сфере.

**Системы управления базами данных (СУБД)**

СУБД являются основными инструментами для создания и управления базами данных. Они могут быть разделены на несколько категорий:

* реляционные СУБД (RDBMS): MySQL, PostgreSQL, Oracle, Microsoft SQL Server;
* не реляционные СУБД (NoSQL): MongoDB, Cassandra, Redis;
* гибридные СУБД: Microsoft Azure Cosmos DB.

**Языки программирования**

* SQL (Structured Query Language): основной язык для работы с реляционными базами данных. Позволяет создавать, изменять, запрашивать и управлять данными;
* Python и JavaScript: часто используются для взаимодействия с базами данных в веб-приложениях. Библиотеки, такие как SQLAlchemy или Sequelize, упрощают работу с базами данных;
* Java и C#: используются в крупных корпоративных приложениях, особенно с реляционными базами данных.

**Инструменты для проектирования баз данных**

* ER-моделирование: Инструменты, такие как Lucidchart, Draw.io или ER/Studio, позволяют создавать диаграммы сущностей и взаимосвязей (ER-диаграммы), которые помогают визуализировать структуру базы данных;
* системы управления версиями: Git и другие системы позволяют контролировать изменения в схемах баз данных и процессах миграции.

**Инструменты для обеспечения безопасности баз данных**

* шифрование: использование технологий шифрования данных (например, AES) для защиты информации как «в покое», так и «в движении»;
* аутентификация и авторизация: применение OAuth, OpenID Connect и других протоколов для управления доступом к базе данных;
* мониторинг и аудит: инструменты, такие как Splunk, могут использоваться для отслеживания запросов и анализа событий безопасности.

**Методы оптимизации производительности**

* индексация: создание индексов для ускорения выполнения запросов, особенно в реляционных базах данных.
* кэширование: использование кэшей, таких как Redis или Memcached, для ускорения доступа к часто запрашиваемым данным.
* балансировка нагрузки: распределение запросов между несколькими серверами баз данных для повышения производительности и отказоустойчивости.

Выбор технологий для разработки базы данных зависит от специфики проекта, объема данных, ожидаемых нагрузок и требований к безопасности. Комплексный подход к выбору инструментов и методов способствует созданию эффективных и надежных систем управления данными, которые могут адаптироваться к изменяющимся условиям и требованиям.

## 1.3. Постановка задачи курсовой работы

С ростом объемов данных, а также увеличением угроз безопасности информации, эффективная разработка и защита баз данных становятся крайне важными задачами для организаций. Безопасность данных имеет критическое значение для решения вопросов конфиденциальности, целостности и доступности информации.

Успешное выполнение работы будет способствовать более глубокому пониманию актуальных проблем и новых подходов в области безопасного управления данными.

### 1.3.1. Цели курсовой работы

Основная цель заключается в создании эффективной и безопасной базы данных, способной обеспечить надежный доступ к информации и защиту от потери или несанкционированного доступа.

### 1.3.2. Задачи курсовой работы

* проектирование архитектуры баз данных;
* разработка схемы данных и моделей;
* обеспечение безопасности данных через шифрование, аутентификацию и авторизацию;
* реализация механизмов резервного копирования и восстановления;
* мониторинг и аудит доступа к данным.

## 1.4.1. Функциональные требования к системе

* возможность создания, обновления и удаления данных;
* поддержка сложных запросов и отчетов;
* обеспечение многопользовательского доступа.

## 1.4.2. Нефункциональные требования к системе

* высокая производительность и надежность;
* защита от несанкционированного доступа;
* удобный интерфейс для пользователей.

## 1.5. Выбор программных средств для разработки базы данных

Выбор программных средств для разработки баз данных является ключевым этапом, который влияет на функциональность, производительность и безопасность системы.

При выполнении данной курсовой работы я планирую использовать следующие программные средства:

* система управления базами данных – **MySQL;**
* языки программирования – **Python, JavaScript;**
* средства проектирования базы данных – Lucidchart, Liquibase;
* инструменты для обеспечения безопасности данных –Vormetric**,** Splunk**;**
* средства тестирования и оптимизации приложений – Apache Bench**, JMeter,** Database Engine Tuning Advisor.

Выбор программных средств для разработки базы данных зависит от специфических требований проекта, таких как объем данных, тип доступа, требования к безопасности и производительности. Важно учитывать будущую масштабируемость и поддерживаемость решений, чтобы создать надежную и эффективную систему.

## 1.6. Выбор среды для разработки базы данных

Выбор подходящей среды для разработки базы данных является критически важным этапом проекта, который может оказать значительное влияние на эффективность работы команды и качество конечного продукта.

При выполнении данной курсовой работы я планирую использовать следующие среды и инструменты:

* IDE и редактор кода – pgAdmin;
* система контейнеризации – Docker;
* система управления версиями – git;
* фреймворк и библиотека – Hibernate Django ORM.

Выбор среды для разработки базы данных требует комплексного подхода с учетом специфики проекта, требований к производительности, безопасности и интеграции. Правильный выбор инструментария и технологии существенно влияет на успех проекта, поэтому следует тщательно проанализировать свои потребности и доступные решения.

# ГЛАВА 2. ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ

## 2.1. Концептуально-логическое моделирование

Концептуально-логическое моделирование – это важный этап в проектировании базы данных, который позволяет структурировать данные и определить их взаимосвязи. Этот процесс разделяется на два ключевых этапа: концептуальное моделирование и логическое моделирование.

### 2.1.1 Концептуальное моделирование

Целью концептуального моделирования является определение основных сущностей и их связей в предметной области, не вдаваясь в детали реализации.

**Идентификация сущностей**

**«Пользователи»**, **«Стартапы», «Инвестиции», «Проекты», «Аналитика и прогнозирование», «Рынок стартапов».**

**Определение атрибутов**

«Пользователи»: ID пользователя, Имя пользователя, Электронная почта, Пароль, Дата регистрации, Баланс, Статус аккаунта.

«Стартапы»: ID стартапа, Название стартапа, Описание стартапа, Дата основания, Стадия развития, ID пользователя.

«Инвестиции»: ID инвестиции, Сумма инвестиции, Дата инвестиции, ID стартапа, ID пользователя.

«Проекты»: ID проекта, Название проекта, Описание проекта, Дата начала проекта, Дата окончания проекта, ID стартапа.

«Аналитика и прогнозирование»: ID анализа, Метод анализа, Дата проведения анализа, Результаты анализа, Точность прогноза.

«Рынок стартапов»: ID рынка, Тип рынка, Объём торгов, Текущие коэффициенты.

**Установление связей**

**Пользователи → Инвестиции**

**Инвестиции → Стартапы**

**Стартапы → Проекты**

**Аналитика и прогнозирование ↔ Пользователи, Инвестиции, Стартапы**

**Рынок стартапов ↔ Стартапы**

**Создание ER-диаграммы**

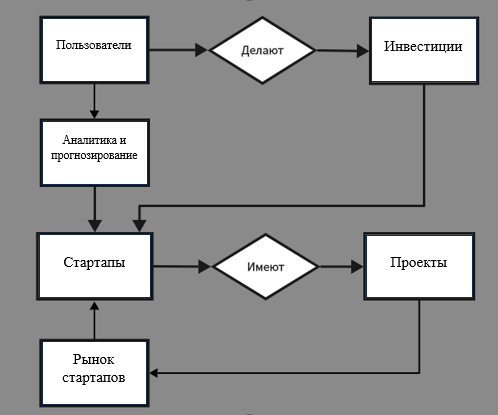


Рисунок 1 – ER-диаграмма разрабатываемой базы данных

### 2.1.2. Логическое моделирование

Целью логического моделирования является перевод концептуальной модели в логическую, с уточнением структуры базы данных и обязательств для хранения данных.

**Определение первичных ключей**

Пользователи – ID пользователя (User ID)

Стартапы – ID стартапы (Startup ID)

Инвестиции – ID инвестиции (Investment ID)

Проекты – ID проекта (Odds ID)

Аналитика и прогнозирование – ID анализа (Analysis ID)

Рынок – ID рынка (Market ID)

**Формализация связей**

**1. Связь «один-ко-многим»:**

**Пользователи и Стартапы:**

В таблице **Стартапы** будет внешний ключ **User ID**, который ссылается на первичный ключ таблицы **Пользователи** (User ID).

**Стартапы и Инвестиции:**

В таблице **Инвестиции** будет внешний ключ **Startup ID**, который ссылается на первичный ключ таблицы **Стартапы** (Startup ID).

**Стартапы и Проекты:**

В таблице **Проекты** будет внешний ключ **Startup ID**, который ссылается на первичный ключ таблицы **Стартапы** (Startup ID). Один стартап может иметь несколько проектов.

**2. Связь «многие-ко-одному»:**

**Инвестиции и Пользователи:**

В таблице **Инвестиции** будет внешний ключ **User ID**, который ссылается на первичный ключ таблицы **Пользователи** (User ID).

**Проекты и Стартапы:**

В таблице **Проекты** будет внешний ключ **Startup ID**, который ссылается на первичный ключ таблицы **Стартапы** (Startup ID).

**3. Связь «многие-ко-многим»:**

**Пользователи и Инвестиции:**

Для реализации связи между **Пользователями и Инвестициями** создадим отдельную таблицу, например,

**User\_Investments**:

**User ID** — внешний ключ, ссылающийся на таблицу, **Пользователи**.

**Investment ID** — внешний ключ, ссылающийся на таблицу **Инвестиции**.

**Стартапы и Проекты:**

Если один проект может быть связан с несколькими стартапами (например, совместные проекты), создадим таблицу

**Startup\_Projects**:

**Startup ID** — внешний ключ, ссылающийся на таблицу **Стартапы**.

**Project ID** — внешний ключ, ссылающийся на таблицу **Проекты**.

**Аналитика и Прогнозирование:**

Если Аналитика может ссылаться на различные сущности, создадим таблицы для каждой связи:

**Analysis\_Users**:

**Analysis ID** — внешний ключ, ссылающийся на таблицу **Аналитика**.

**User ID** — внешний ключ, ссылающийся на таблицу **Пользователи**.

**Analysis\_Startups**:

**Analysis ID** — внешний ключ, ссылающийся на таблицу **Аналитика**.

**Startup ID** — внешний ключ, ссылающийся на таблицу **Стартапы**.

**Analysis\_Investments**:

**Analysis ID** — внешний ключ, ссылающийся на таблицу **Аналитика**.

**Investment ID** — внешний ключ, ссылающийся на таблицу **Инвестиции**. **Analysis\_Projects**:

**Analysis ID** — внешний ключ, ссылающийся на таблицу **Аналитика**.

**Project ID** — внешний ключ, ссылающийся на таблицу **Проекты**.

**Определение типов данных**

**1. Пользователи (Users)**

**ID пользователя** (User ID): INT

**Имя пользователя** (Username): VARCHAR (50)

**Электронная почта** (Email): VARCHAR (100)

**Пароль** (Password): VARCHAR (255)

**Дата регистрации** (Registration Date): DATETIME

**Баланс** (Balance): DECIMAL (10, 2)

**Статус аккаунта** (Account Status): VARCHAR (20)

**2. Стартапы (Startups)**

**ID стартапа** (Startup ID): INT

**Название стартапа** (Startup Name): VARCHAR (100)

**Описание стартапа** (Startup Description): TEXT

**Дата основания** (Founding Date): DATETIME

**Стадия развития** (Development Stage): VARCHAR (50)

**ID пользователя** (User ID): INT (FK)

**3. Инвестиции (Investments)**

**ID инвестиции** (Investment ID): INT

**Сумма инвестиции** (Investment Amount): DECIMAL (15, 2)

**Дата инвестиции** (Investment Date): DATETIME

**ID стартапа** (Startup ID): INT (FK)

**ID пользователя** (User ID): INT (FK)

**4. Проекты (Projects)**

**ID проекта** (Project ID): INT

**Название проекта** (Project Name): VARCHAR (100)

**Описание проекта** (Project Description): TEXT

**Дата начала проекта** (Project Start Date): DATETIME

**Дата окончания проекта** (Project End Date): DATETIME

**ID стартапа** (Startup ID): INT (FK)

**5. Аналитика и прогнозирование (Analytics)**

**ID анализа** (Analysis ID): INT

**Метод анализа** (Analysis Method): VARCHAR (50)

**Дата проведения анализа** (Analysis Date): DATETIME

**Результаты анализа** (Results): TEXT

**Точность прогноза** (Forecast Accuracy): DECIMAL (5, 2)

**6. Рынок стартапов (Startup Market)**

**ID рынка** (Market ID): INT

**Тип рынка** (Market Type): VARCHAR (50)

**Объём торгов** (Trading Volume): DECIMAL (15, 2)

**Текущие коэффициенты** (Current Odds): DECIMAL (5, 2)

**7. Аналитика­\_Пользователи** **(Analytics\_Users)**

**Analysis ID** (FK): INT

**User ID (FK):** INT

**8. Аналитика­\_Инвестиции** (**Analytics**\_**Investments)**

**Analysis ID** (FK): INT

**Investment ID** (FK): INT

**9. Аналитика\_Проекты (Analytics**\_**Projects)**

**Analysis ID** (FK): INT

**Project ID** (FK): INT

**10. Аналитика\_Стартапы (Analytics**\_**Startups)**

**Analysis ID** (FK): INT

**Startup ID** (FK): INT

**Создание логической схемы базы данных**

**1. Пользователи (Users)**

**ID пользователя (User ID)** — INT (первичный ключ)

**Имя пользователя (Username)** — VARCHAR (50)

**Электронная почта (Email)** — VARCHAR (100)

**Пароль (Password)** — VARCHAR (255)

**Дата регистрации (Registration Date)** — DATETIME

**Баланс (Balance)** — DECIMAL (10, 2)

**Статус аккаунта (Account Status)** — VARCHAR (20)

**2. Стартапы (Startups)**

**ID стартапа (Startup ID)** — INT (первичный ключ)

**Название стартапа (Startup Name)** — VARCHAR (100)

**Описание стартапа (Startup Description)** — TEXT

**Дата основания (Founding Date)** — DATETIME

**Стадия развития (Development Stage)** — VARCHAR (50)

**ID пользователя (User ID)** — INT (внешний ключ, ссылается на **Пользователи**)

**3. Инвестиции (Investments)**

**ID инвестиции (Investment ID)** — INT (первичный ключ)

**Сумма инвестиции (Investment Amount)** — DECIMAL (15, 2)

**Дата инвестиции (Investment Date)** — DATETIME

**ID стартапа (Startup ID)** — INT (внешний ключ, ссылается на **Стартапы**)

**ID пользователя (User ID)** — INT (внешний ключ, ссылается на **Пользователи**)

**4. Проекты (Projects)**

**ID проекта (Project ID)** — INT (первичный ключ)

**Название проекта (Project Name)** — VARCHAR (100)

**Описание проекта (Project Description)** — TEXT

**Дата начала проекта (Project Start Date)** — DATETIME

**Дата окончания проекта (Project End Date)** — DATETIME

**ID стартапа (Startup ID)** — INT (внешний ключ, ссылается на **Стартапы**)

**5. Аналитика и прогнозирование (Analytics)**

**ID анализа (Analysis ID)** — INT (первичный ключ)

**Метод анализа (Analysis Method)** — VARCHAR (50)

**Дата проведения анализа (Analysis Date)** — DATETIME

**Результаты анализа (Results)** — TEXT

**Точность прогноза (Forecast Accuracy)** — DECIMAL (5, 2)

**6. Рынок стартапов (Startup Market)**

**ID рынка (Market ID)** — INT (первичный ключ)

**Тип рынка (Market Type)** — VARCHAR (50)

**Объём торгов (Trading Volume)** — DECIMAL (15, 2)

**Текущие коэффициенты (Current Odds)** — DECIMAL (5, 2)

**Связи между таблицами**

**Пользователи** делают много **Стартапов**.

Один пользователь может создать много стартапов.

**Пользователи** делают много **Инвестиций**.

Один пользователь может сделать много инвестиций.

**Стартапы** могут иметь много **Проектов**.

Один стартап может включать много проектов.

**Инвестиции** относятся к одному **Стартапу**.

Одна инвестиция относится к одному стартапу.

**Аналитика и прогнозирование** используют данные из всех других таблиц.

**Связь многие-ко-многим реализуется через отдельные таблицы**

**Дополнительные таблицы для связи многие-ко-многим**

1. **Аналитика\_Пользователи (Analytics**\_**Users)**

**Analysis ID (FK)** — INT

**User ID (FK)** — INT

2. **Аналитика\_Инвестиции (Analytics**\_**Investments)**

**Analysis ID (FK)** — INT

**Investment ID (FK)** — INT

3. **Аналитика\_Проекты (Analytics**\_**Projects)**

**Analysis ID (FK)** — INT

**Project ID (FK)** — INT

4. **Аналитика\_Стартапы (Analytics**\_**Startups)**

**Analysis ID (FK)** — INT

**Startup ID (FK)** — INT

5. **Аналитика\_Рынки (Analytics**\_**Markets)**

**Analysis ID (FK)** — INT

**Market ID (FK)** — INT

Концептуально-логическое моделирование является важным этапом разработки базы данных, который позволяет углубиться в детализацию структуры данных и обеспечить целостность и согласованность информации. Правильное моделирование позволяет избежать проблем на следующих этапах разработки и эксплуатации базы данных.

## 2.2. Описание информационных объектов базы данных

Информационные объекты базы данных представляют собой основные строительные блоки, на которых базируется структура любой базы данных. Они включают в себя различные сущности и их атрибуты, которые позволяют организовать, хранить и управлять данными.

Таблица 1 – информационные объекты разрабатываемой базы данных

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Объект | Атрибуты | Связи | Ограничения | Тип данных |
| **Пользователи (User)** | ID пользователя, Имя пользователя, Электронная почта, Пароль, Дата регистрации, Баланс, Статус аккаунта | Один пользователь может создавать много стартапов (1: N с Стартапами) | Уникальность ID пользователя, Обязательность имени пользователя | INT, VARCHAR, DATETIME, DECIMAL |
| **Стартапы (Startup)** | ID стартапа, Название стартапа, Описание стартапа, Дата основания, Стадия развития, ID пользователя | Один стартап может иметь много проектов (1: N с Проектами) | Уникальность ID стартапа, Обязательность названия стартапа | INT, VARCHAR, TEXT, DATETIME |
| **Инвестиции (Investment)** | ID инвестиции, Сумма инвестиции, Дата инвестиции, ID стартапа, ID пользователя | Одна инвестиция относится к одному стартапу (N:1 с Стартапами), Один пользователь может делать много инвестиций (1: N с Пользователями) | Уникальность ID инвестиции, Обязательность суммы инвестиции | INT, DECIMAL, DATETIME |
| **Проекты (Project)** | ID проекта, Название проекта, Описание проекта, Дата начала проекта, Дата окончания проекта, ID стартапа | Один проект относится к одному стартапу (N:1 с Стартапами) | Уникальность ID проекта, Обязательность названия проекта | INT, VARCHAR, TEXT, DATETIME |
| **Аналитика и прогнозирование** **(Analytics)** | ID анализа, Метод анализа, Дата проведения анализа, Результаты анализа, Точность прогноза | Связь многие-ко-многим с Пользователями и Инвестициями | Уникальность ID анализа | INT, VARCHAR, DATETIME, TEXT, DECIMAL |
| **Рынок стартапов** **(Startup Market)** | ID рынка, Тип рынка, Объём торгов, Текущие коэффициенты | Один рынок может включать много стартапов (1: N с Стартапами) | Уникальность ID рынка | INT, VARCHAR, DECIMAL |
| **Дополнительные таблицы для связи многие-ко-многим** | **Аналитика\_Пользователи (Analytics\_Users)**  Analysis ID  User ID  **Аналитика\_Инвестиции (Analytics\_Investments)**  Analysis ID (FK)  Investment ID (FK)  **Аналитика\_Проекты (Analytics\_Projects)**  Analysis ID (FK)  Project ID (FK)  **Аналитика\_Стартапы (Analytics\_Startups)**  Analysis ID (FK)  Startup ID (FK)  **Аналитика\_Рынки (Analytics\_Markets)**  Analysis ID (FK)  Market ID (FK) |  |  | INT |

Описание информационных объектов базы данных позволяет структурировать и организовать информацию, обеспечивая эффективное хранение и доступ к данным. Правильное определение сущностей, их атрибутов и связей является ключом к успешному проектированию базы данных и ее функциональной полноте.

## 2.3.1. Создание базы данных

Создание самой базы данных с помощью соответствующей команды

CREATE DATABASE name\_database;



Рисунок 2 – созданная база данных

## 2.3.2. Заполнение базы данных

1. **Определение объектов и атрибутов**:

На этом этапе необходимо определить, какие объекты будут храниться в базе данных и какие атрибуты будут у этих объектов. Например, для объекта «Пользователь» атрибутами могут быть:

ID пользователя, Имя, Дата регистрации, Баланс, История инвестиций, Статус аккаунта

1. **Выбор типа данных**:

Для каждого атрибута необходимо выбрать соответствующий тип данных. Например:

ID пользователя (User ID) — INT

Имя пользователя (Username) — VARCHAR (50)

Дата регистрации (Registration\_Date) — DATETIME

Баланс (Balance) — DECIMAL (10, 2)

Статус аккаунта (Account\_Status) — VARCHAR (20)

1. **Создание SQL-запросов для вставки данных**:

После определения объектов, атрибутов и типов данных можно создать SQL-запросы для заполнения таблиц данными.



Рисунок 3 – добавление объекта в базу данных

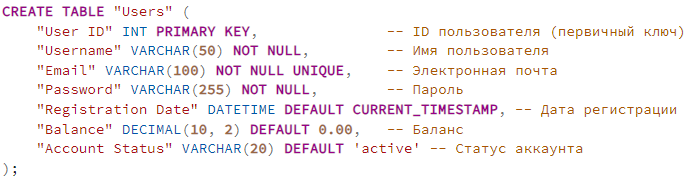


Рисунок 4 – добавление атрибутов объекту в базе данных

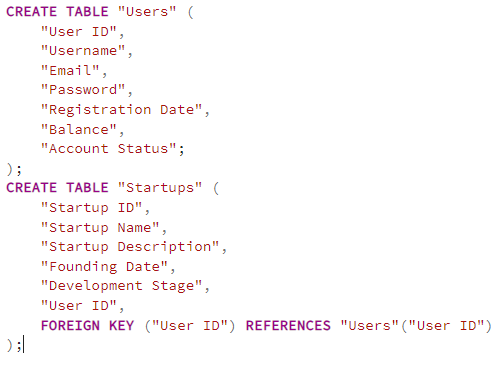


Рисунок 5 – установление связей между объектами в базе данных

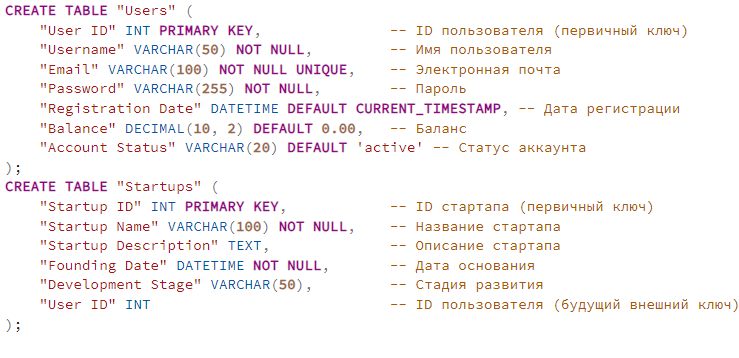


Рисунок 6 – выбор типа данных для объекта в базе данных

## 2.4. Представления в базе данных

Представление в базе данных – это виртуальная таблица, которая формируется на основе результата выполнения SQL-запроса. Она не содержит данных, а лишь определяет, как данные будут отображаться пользователям.

Функции представлений:

* упрощение сложных запросов – позволяют объединять данные из нескольких таблиц и показывать только нужные поля, что делает работу с данными более удобной;
* безопасность – можно ограничить доступ пользователей к определённым столбцам или строкам баз данных, предоставляя доступ только к представлениям;
* логическая независимость – изменения в базах данных не влияют на внешний интерфейс, так как представления могут скрывать изменения структуры таблиц;
* кодовая реиспользуемость – позволяют сократить код, избавляя от необходимости повторно писать одни и те же запросы.

Представление создается с помощью команды CREATE VIEW.

|  |
| --- |
| sql  CREATE VIEW user\_startups AS  SELECT  u."User ID", -- ID пользователя  u."Username", -- Имя пользователя  u."Email", -- Электронная почта  u."Registration Date", -- Дата регистрации  u."Balance", -- Баланс  u."Account Status", -- Статус аккаунта  s."Startup ID", -- ID стартапа  s."Startup Name", -- Название стартапа  s."Startup Description", -- Описание стартапа  s."Founding Date", -- Дата основания  s."Development Stage" -- Стадия развития  FROM  "Users" u -- Таблица Пользователи  LEFT JOIN  "Startups" s ON u."User ID" = s."User ID"; -- Соединение по ID пользователя |

Этот SQL-запрос создает представление (VIEW) под названием user\_startups, которое объединяет данные из двух таблиц: Users (Пользователи) и Startups (Стартапы).

После создания представления его можно использовать так же, как и таблицу:

|  |
| --- |
| sql  SELECT \*, CONCAT (InvestmentAmount, ‘ RUB’)  AS InvestmentAmountRUB  FROM user\_bets  WHERE BetAmount > 1000; --Инвестиции в рублях |

Запрос возвращает все строки из таблицы user\_bets, где сумма ставки (BetAmount) превышает 1000, и добавляет новый столбец InvestmentAmountRUB, который отображает сумму инвестиций с указанием валюты (рубли).

Представления могут быть обновляемыми или не обновляемыми. Обновляемые представления позволяют выполнять операции INSERT, UPDATE и DELETE, при этом изменения будут применяться к базовой таблице.

|  |
| --- |
| UPDATE Пользователи  SET AccountStatus = 'Неактивен'  WHERE UserID = 5; |

Запрос делает аккаунт пользователя с UserID = 5 неактивным, устанавливая его статус (AccountStatus) в значение 'Неактивен'.

Представления можно изменять с помощью команды CREATE OR REPLACE VIEW или удалять с помощью DROP VIEW

|  |
| --- |
| CREATE OR REPLACE VIEW user\_startups AS  SELECT  u."User ID", -- ID пользователя  u."Username", -- Имя пользователя  u."Email", -- Электронная почта  u."Registration Date", -- Дата регистрации  s."Startup ID", -- ID стартапа  s."Startup Name", -- Название стартапа  s."Startup Description", -- Описание стартапа  s."Founding Date", -- Дата основания  s."Development Stage", -- Стадия развития  u."Account Status" AS "UserStatus" -- Статус аккаунта пользователя  FROM  "Пользователи" u -- Таблица Пользователи  LEFT JOIN  "Стартапы" s ON u."User ID" = s."User ID" -- Соединение по ID пользователя  WHERE  u."Account Status" = 'active'; -- Только активные пользователи |

Этот SQL-запрос создает или заменяет представление (VIEW) под названием user\_startups.

Представление user\_startups содержит информацию о активныхпользователях и их стартапах (если они есть). Это удобно для анализа данных, связанных с активными аккаунтами.

|  |
| --- |
| sql  DROP VIEW IF EXISTS employee\_details; |

Этот SQL-запрос выполняет удаление представления (VIEW) под названием employee\_details.

Представления в базах данных являются мощным инструментом для работы с данными. Они обеспечивают большую гибкость, безопасность и легкость в использовании, способствуя более эффективной организации и обработке информации. Понимание работы с представлениями – важный аспект для разработчиков и администраторов баз данных.

## 2.5. Процедуры в базе данных

(2 курс 2 семестр)

## 2.6. Создание ролей в базе данных

(2 курс 2 семестр)

## 2.7. Аутентификация пользователей в базе данных

(2 курс 2 семестр)

## 2.8. Импорт и экспорт базы данных

(2 курс 2 семестр)

## 2.9. Разработка стратегии резервного копирования базы данных

(2 курс 2 семестр)

## 2.10.1. Тестирование базы данных

(2 курс 2 семестр)

## 2.10.2. Оптимизация базы данных

(2 курс 2 семестр)

# ЗАКЛЮЧЕНИЕ

В ходе изучения технологии разработки и защиты баз данных было выявлено, что успешная работа с данными в современных условиях требует комплексного подхода, включающего как аспекты проектирования, так и обеспечения безопасности. Принимая во внимание стремительное развитие информационных технологий, применение современных методов и инструментов становится неизбежным. Это позволяет не только оптимизировать процессы хранения и обработки данных, но и защитить их от возможных угроз.

Особое внимание следует уделить вопросам шифрования данных, аутентификации пользователей и резервного копирования. Эффективная защита баз данных требует активного мониторинга и регулярного обновления систем безопасности.

Таким образом, подходы к разработке и защите баз данных должны быть гибкими и адаптивными, чтобы соответствовать быстро меняющимся требованиям и угрозам. В будущем важно продолжать исследовать и внедрять новые технологии, что позволит обеспечить надежность и безопасность информации на всех уровнях.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Волик, М.В. Разработка базы данных в Access / М.В. Волик. – Издательство «Прометей», 2021. – 88 с.
2. Даккет, Дж. PHP и MySQL. Серверная веб-разработка / Дж. Даккет. – Издательство «Эксмо», 2023. – 688 с.
3. Дьяков, И.А. Базы данных. Язык SQL : учебное пособие / И.А. Дьяков. – Тамбов : Тамбовский государственный технический университет (ТГТУ), 2022. – 82 с.
4. Ла Рокка, М. Продвинутые и структуры данных. / М. ла Рокка. – Издательство «Питер», 2024. – 848 с.
5. Никсон, Р. Создаём динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5. 6-е изд. / Р. Никсон. – Издательство «Питер»: Санкт-Петербург, 2023. – 832 с.
6. https://www.figma.com/ (дата обращения: 15.03.2025).
7. https://www.github.com/ (дата обращения: 16.03.2025).

ПРИЛОЖЕНИЯ

(Скриншоты разработанной базы данных)