****

**Частное учреждение профессионального образования**

**«Высшая школа предпринимательства»**

**(ЧУПО «ВШП»)**

КУРСОВОЙ ПРОЕКТ

«Разработка базы данных для энциклопедии по игре “Genshin impact”»

Выполнил:

студент 3-го курса специальности

09.02.07 «Информационные системы и программирование»  
Дедов Даниил Романович

подпись: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Проверил:

преподаватель дисциплины,  
преподаватель ЧУПО «ВШП»,  
к.ф.н. Ткачев П.С.

оценка: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

подпись: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Оглавление

[Введение 3](#_Toc166742268)

[Глава 1. Теория о разработке базы данных 6](#_Toc166742269)

[1.1Понятие и структура базы данных 6](#_Toc166742270)

[1.2 Особенности разработки базы данных для энциклопедии и её функций 8](#_Toc166742271)

[1.3 **Обзор существующих баз данных и информационных систем** 9](#_Toc166742272)

[1.4 **Методы разработки баз данных** 10](#_Toc166742273)

1.5 **Выбор технологии для разработки базы данных…………………...12**

**1.6** Требования для базы данных и проектировка схемы……………...14

Глава 2.

**Введение**

**Актуальность темы исследования**

В современном мире компьютерные игры становятся всё более популярными, привлекая миллионы игроков по всему миру. Одной из таких игр является Genshin Impact, которая завоевала огромную аудиторию благодаря своей красочной графике, увлекательному сюжету и разнообразному игровому процессу. Однако, несмотря на свою популярность, игра не имеет полноценной энциклопедии, которая могла бы предоставить игрокам подробную информацию об игровом мире, персонажах, событиях и других аспектах игры.

Разработка базы данных для энциклопедии по игре «Genshin Impact» позволит создать удобный и информативный ресурс, который будет полезен как для игроков, так и для исследователей игры. База данных позволит систематизировать информацию об игре, сделать её доступной и удобной для поиска, а также расширить знания об игре и её влиянии на культуру.

**Разработанность темы**

На данный момент существует ограниченное количество исследований, посвящённых разработке баз данных для игровых энциклопедий. Однако существует множество исследований, посвящённых базам данных и информационным системам в целом. Также существуют исследования, посвящённые игре «Genshin Impact», которые могут быть полезны при разработке базы данных.

**Цель и задачи работы**

**Цель:**

Целью данной работы является разработка базы данных для энциклопедии по игре «Genshin Impact».

**Задачи:**

1. Изучить существующие базы данных и информационные системы, а также методы их разработки.
2. Проанализировать игру «Genshin Impact» и определить, какая информация должна быть включена в энциклопедию.
3. Разработать структуру базы данных, которая будет соответствовать требованиям игры.
4. Реализовать базу данных с использованием выбранной технологии.
5. Протестировать базу данных и убедиться в её работоспособности.

**Объект исследования**

Объектом исследования является игра «Genshin Impact» и её аудитория.

**Методы исследования**

Для достижения поставленной цели и решения задач будут использованы следующие методы исследования:

1. Анализ существующих баз данных и информационных систем.
2. Анализ игры «Genshin Impact» и определение требований к базе данных.
3. Проектирование структуры базы данных.
4. Реализация базы данных с использованием выбранной технологии.
5. Тестирование базы данных.

Таким образом, разработка базы данных для энциклопедии по игре «Genshin Impact» является актуальной и важной задачей, которая позволит создать удобный и информативный ресурс для игроков и исследователей игры.

**Глава 1. Теория о разработке базы данных**

**1.1. Понятие и структура базы данных**

База данных – это организованная коллекция данных, специально разработанная для эффективного хранения, управления и обработки информации. Она обычно состоит из нескольких взаимосвязанных таблиц, каждая из которых содержит информацию о сущностях и их атрибутах.

**Основные компоненты базы данных:**

**Таблицы**: Структурированные наборы данных, представляющие собой сущности и их атрибуты. Например, таблица персонажей может содержать информацию о каждом персонаже, такую как имя, уровень, тип, и т.д.

**Строки**: Каждая строка в таблице представляет собой отдельную запись или элемент данных. Например, каждая строка в таблице персонажей может представлять одного конкретного персонажа.

**Столбцы**: Каждый столбец в таблице представляет собой атрибут или свойство сущности. Например, столбец "Имя" в таблице персонажей содержит имена персонажей.

**Отношения**: Связи между таблицами, определяющие способы взаимодействия между сущностями. Например, отношение между таблицами персонажей и предметов может указывать на то, какие предметы принадлежат определенному персонажу.

**Принципы работы базы данных:**

**Сохранение целостности данных**: Гарантия того, что данные остаются точными, целостными и непротиворечивыми во времени.

**Обеспечение безопасности и конфиденциальности**: Защита данных от несанкционированного доступа и изменений.

**Эффективное выполнение запросов**: Обеспечение быстрого доступа к данным и выполнения операций над ними, чтобы удовлетворить потребности пользователей.

**1.2. Особенности разработки базы данных для энциклопедии и её функций**

Разработка базы данных для энциклопедии, посвященной игре "Genshin Impact", имеет свои особенности. Важно учитывать специфику игровой информации, такую как персонажи, миры, предметы, задания и т.д. Также необходимо предусмотреть возможность обновления и расширения базы данных с выходом новых обновлений игры.

Основные этапы разработки базы данных включают в себя:

1. **Анализ требований**: определение основных сущностей и их атрибутов на основе информации об игре.
2. **Проектирование структуры**: разработка схемы базы данных, включая определение таблиц, их полей и отношений.
3. **Реализация и наполнение**: создание базы данных и заполнение её информацией, включая данные о персонажах, мирах, предметах и других игровых элементах.
4. **Тестирование и оптимизация**: проверка работоспособности базы данных и оптимизация её производительности

Функции которые должна включать разрабатываемая база данных

1. **Поиск: функция поиска позволит пользователям находить информацию по ключевым словам, категориям и другим параметрам.**
2. **Фильтрация: функция фильтрации позволит пользователям ограничивать результаты поиска по определённым критериям, таким как дата публикации, язык и т.п.**
3. **Сортировка: функция сортировки позволит пользователям упорядочивать результаты поиска по различным параметрам, таким как популярность, релевантность и т.д.**

**1.3. Обзор существующих баз данных и информационных систем**

Для разработки базы данных необходимо провести анализ существующих баз данных и информационных систем. Это позволит определить основные принципы и подходы, которые будут использованы при разработке базы данных для энциклопедии по игре «Genshin Impact».

Существует множество баз данных и информационных систем, которые используются для различных целей. Например, существуют базы данных, которые используются для хранения информации о клиентах, продуктах, заказах и других объектах. Существуют также информационные системы, которые используются для автоматизации бизнес-процессов, таких как управление запасами, управление персоналом и управление финансами.

При анализе существующих баз данных и информационных систем необходимо учитывать следующие факторы:

1. **Цель использования**. Для чего будет использоваться база данных или информационная система?
2. **Структура данных**. Как будут организованы данные в базе данных или информационной системе?
3. **Методы доступа к данным**. Как будут осуществляться запросы к базе данных или информационной системе?
4. **Безопасность данных**. Как будут защищены данные в базе данных или информационной системе?

Анализ существующих баз данных и информационных систем позволит определить основные принципы и подходы, которые будут использованы при разработке базы данных для энциклопедии по игре «Genshin Impact». Это позволит создать удобный и эффективный ресурс, который будет соответствовать требованиям игры.

**1.4. Методы разработки баз данных**

Существует несколько подходов к разработке баз данных для энциклопедий, каждый из которых имеет свои преимущества и недостатки:

**Централизованная база данных**: Одно центральное хранилище информации об игровых элементах, что обеспечивает единообразный доступ к данным, но может привести к проблемам с масштабируемостью и производительностью.

**Децентрализованные базы данных**: Использование отдельных баз данных для разных аспектов игры, что может обеспечить более гибкую структуру, но требует управления несколькими базами данных.

**Использование семантических технологий**: Создание базы данных с использованием онтологий и RDF-троек, что облегчает интеграцию и обмен данными, но может потребовать дополнительных затрат на обучение и разработку.

Для разработки базы данных для энциклопедии по игре «Genshin Impact» необходимо использовать методы, которые позволят создать удобный и эффективный ресурс. Одним из таких методов является использование реляционной модели данных. Реляционная модель данных позволяет представить информацию в виде таблиц, что упрощает её обработку и анализ.

Другим важным методом является использование объектно-ориентированного подхода. Объектно-ориентированный подход позволяет создать гибкую и масштабируемую систему, которая может быть легко адаптирована к изменяющимся требованиям.

Также необходимо использовать методы проектирования баз данных, такие как ER-диаграммы и нормализация. ER-диаграммы позволяют наглядно представить структуру базы данных, а нормализация позволяет избежать избыточности данных и повысить эффективность работы базы данных.

**1.5. Выбор технологии для разработки базы данных**

Для разработки базы данных можно использовать различные технологии, такие как MySQL, PostgreSQL, Oracle, Microsoft SQL Server и другие. Каждая из этих технологий имеет свои преимущества и недостатки. Например, MySQL является бесплатным и открытым исходным кодом, но имеет ограниченные возможности масштабирования. PostgreSQL является более мощным и масштабируемым, но требует больше ресурсов для установки и настройки.

После анализа преимуществ и недостатков различных технологий для разработки базы данных, был сделан вывод, что для разработки базы данных для энциклопедии по игре «Genshin Impact» наиболее подходящей является технология MySQL. Эта технология является бесплатной и открытой, что позволяет использовать её для разработки некоммерческого проекта. Кроме того, MySQL имеет множество инструментов и библиотек, которые могут быть использованы при разработке базы данных.

Одним из ключевых аспектов разработки базы данных для энциклопедии "Genshin Impact" является обеспечение её гибкости и масштабируемости. С учетом динамичной природы игрового контента и постоянного обновления игровых механик и персонажей необходимо создать информационную систему, способную легко адаптироваться к изменениям и расширяться по мере необходимости.

Для достижения этой цели важно использовать современные подходы к проектированию баз данных, такие как нормализация данных, использование распределенных систем хранения данных и механизмы кэширования. Такие методы позволяют обеспечить эффективное управление ресурсами и улучшить производительность базы данных даже при большом объеме информации и высоких нагрузках.

Кроме того, важно предусмотреть возможность интеграции с внешними источниками данных, такими как игровые API, социальные сети и форумы сообщества игроков. Это позволит обогатить базу данных актуальной информацией и обеспечить её полноту и достоверность.

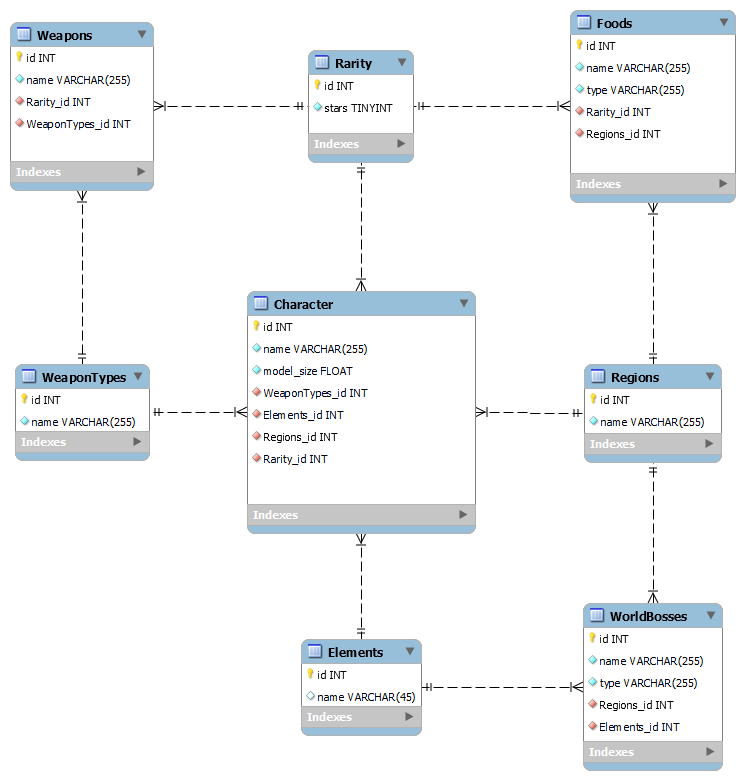
Таким образом, гибкость и масштабируемость базы данных играют ключевую роль в обеспечении её актуальности и функциональности, что в конечном итоге способствует удовлетворению потребностей как игроков, так и разработчиков игры "Genshin Impact".

**1.6 Требования для базы данных и проектировка схемы**

Основываясь на проведённом анализе различных типов баз данных, я выбираю реляционные базы данных. Они хорошо подходят для создания таблиц и установления связей между объектами.

База данных должна отражать основные данные персонажа, наименования элементов, характеристики оружий и еды, мировых боссов, регионов а также редкости для того, у чего она есть.

Перед началом разработки базы данных, стоит построить схему которая будет отражать связи между сущностями. Эта схема называется ERD (Entity-Relationship Diagram). ER-диаграмма - это разновидность блок-схемы, где показано, как разные «сущности» (люди, объекты, концепции и так далее) связаны между собой внутри системы.



**Рис. 1**

Для составления схемы базы данных я воспользовался функцией создания моделей в MySQL Workbench.

В данной схеме (рис. 1) готовая спроектированная база данных составления с учётом требований описанных ранее

В схеме находится 8 таблиц, каждая имеет свою логическую ценность при выполнение поставленной задачи

1. Weapons – Таблица которая содержит список оружий, используемых персонажами.
2. Rarity – Таблица которая содержит количество звёзд, которые отражают редкость некоторых элементов игры.
3. Foods – Таблица которая содержит еду, используемую персонажами в различных целях.
4. WeaponTypes – Таблица которая отражает типы оружия(меч, лук и т.д), эта таблица нужна из-за того что, все персонажи используют разный тип оружия.
5. Character – Таблица которая содержит, информацию про персонажей
6. Regions – Таблица которая содержит наименование регионов.
7. Elements – Таблица которая содержит наименование элементов к которым принадлежат некоторые аспекты игры.
8. WorldBosses – Таблица которая содержит информацию о мировых боссах.

**Глава 2. Разработка базы данных.**

При проектировании базы данных важно тщательно продумать типы данных для каждого столбца, поскольку MySQL предлагает широкий спектр опций. Каждый тип данных занимает разный объем памяти, и правильный выбор может существенно повлиять на общий размер таблицы.

Создание таблиц по схеме (Рис. 1), разработанной ранее.

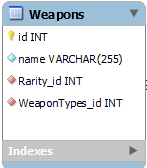
1. Таблица Weapons содержит столбцы: id, name, rarity\_id, weaponTypes\_id.

1. Для столбца id используется тип данных INT, использую Primary Key, также Not Null и Auto Increment. Тип данных INT используется чтобы в будущем не было упора в диапазон другого типа данных. Primary Key используется как уникальный идентификатор каждой записи в таблице, Auto increment используется для того, чтобы при записи новых данных id присваивался автоматически. Not null – используется потому что id не может быть равен null.

2. Для столбца name используется тип данных VARCHAR(255) 255, что означает – он сможет вместить запись не длиннее 255 символов. Так же используется Not Null.

3. Rarity\_id используется для создания связи между таблицами Weapons и Rarity. Используемая связь один ко многим (1:n) , где Rarity это 1, a Weapons n.

4. WeaponTypes\_id также служит для создания связи между таблицами Weapons и WeaponTypes. . Используемая связь один ко многим (1:n) , где WeaponTypes это 1, a Weapons n.



**Рис. 2**

Так выглядит готовая таблица Weapons(Рис. 2)

1. Таблица Rarity содержит столбцы: id, stars.

1. Для id используется Primary Key, Not Null, Auto Increment – так как этот столбец нужен для идентификации записей.

2. Для Stars использую TINYINT, так как это самая маленькая переменна и не требующая много места, потому-что количество звёзд не может быть больше 5.



**Рис. 3**

Так выглядит готовая таблица Rarity (Рис. 3)

1. Таблица Foods содержит столбцы: id, name, type, Rarity\_id, Region\_id.

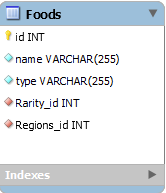
1. Для id используется Primary Key, Not Null, Auto Increment – так как этот столбец нужен для идентификации записей.

2. Для Name используется тип VARCHAR(255), такого размера будет достаточно чтобы при добавлении новых позиций не возникало проблем.

3. Для Type используется тип VARCHAR(255), опять же, такой же размер используется во избежание проблем при добавлении новых позиций.

4. Rarity\_id используется для создания связи между таблицами Foods и Rarity. Используемая связь один ко многим (1:n) , где Rarity это 1, a Foods n.

5. Region\_id используется для создания связи между таблицами Foods и Region. Используемая связь один ко многим (1:n) , где Region это 1, a Foods n.



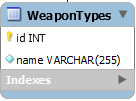
**Рис. 4**

Так выглядит готовая таблица Foods (Рис. 4)

1. Таблица WeaponTypes содержит столбцы: id, name.

1. Для id используется Primary Key, Not Null, Auto Increment – так как этот столбец нужен для идентификации записей.

2.Для Name используется тип VARCHAR(255), такого размера будет достаточно чтобы при добавлении новых позиций не возникало проблем.



**Рис. 5**

Так выглядит готовая таблица WeaponTypes (Рис. 5)

1. Таблица Character содержит столбцы: id, name, model\_size, WeaponTypes\_id, Elements\_id, Regions\_id, Rarity\_id.

1. Для id используется Primary Key, Not Null, Auto Increment – так как этот столбец нужен для идентификации записей.

2. Для Name используется тип VARCHAR(255), такого размера будет достаточно чтобы при добавлении новых позиций не возникало проблем.

3. Для model\_size используется тип VARCHAR(255), такого размера будет достаточно чтобы при добавлении новых позиций не возникало проблем.

4. Столбец WeaponTypes\_id служит для создания связи между таблицами Character и WeaponTypes и использует тип связи один ко многим (1:n) где WeaponTypes это - 1, a character - n

5. Столбец Elements\_id служит для создания связи между таблицами Character и Elements и использует тип связи один ко многим (1:n) где Elements это - 1, a character - n