СОДЕРЖАНИЕ

[Введение 3](#_Toc24068437)

[1 Аналитический обзор 4](#_Toc24068438)

[1.1 Описание предметной области 4](#_Toc24068439)

[1.2 Техническое задание 6](#_Toc24068440)

[1.2.1 Функциональные требования 6](#_Toc24068441)

[1.2.2 Минимальные системные требования 6](#_Toc24068442)

[1.2.3 Требование к документации 6](#_Toc24068443)

[1.3 Выбор средств разработки 7](#_Toc24068444)

[1.3.1 Выбор СУБД 7](#_Toc24068445)

[1.3.2 Вывод 8](#_Toc24068446)

[1.3.3 Дополнительные средства разработки 8](#_Toc24068447)

[2 Разработка Базы Данных 10](#_Toc24068448)

[Использованные источники 11](#_Toc24068449)

Введение

База данных – совокупность материалов, систематизированных таким образом, чтобы эти материалы могли быть найдены и обработаны с помощью ЭВМ.

История современных баз данных начинается с 1955 года, когда появилось программируемое оборудование обработки записей. Важный этап связан с появлением в начале 1970-х реляционной модели данных.

1. Аналитический обзор
   1. Описание предметной области

Игра представляет собой Массовую многопользовательскую онлайн игру (ММО) с элементами слешера, шутера от третьего лица (TPS) и ролевых игр (RPG), выполненную в киберпанк сеттинге.

Массовые многопользовательские онлайн игры предполагают одновременное нахождение на одном сервере большого количества игроков.

Перед тем как играть, игрок регистрируется, указывая логин, пароль и почту. Так же игрок может указать никнейм, страну, изображение профиля, а также добавлять других пользователей в список друзей, что упрощает совместную игру.

Каждый игрок имеет как минимум одного персонажа для игры на сервере, каждый из которых имеет свои характеристики, умения, подобранные и экипированные предметы и полигональную сетку (Mesh) для отрисовки персонажа в самой игре. Те же характеристики и умения имеются и у неигровых персонажей, которые появляются в игре с различными целями. К характеристикам персонажей относятся: имя, максимальное здоровье, % от максимального здоровья.

Игрок может изучить любые из умений. Применяться будут только выбранные умения.

Персонажи используют различные предметы. Каждый из предметов имеет название, редкость и свой mesh. Редкость предмета определяет количество экземпляров в мире игры.

Предметы подразделяются на:

* Оружие;
* Одежду;
* Расходуемые предметы;
* Дополнительные устройства.
* Аугментации / Импланты.

Оружие располагается на персонаже в установленных местах. У каждого оружия свои анимации для того, чтобы взять или убрать его.

Оружие подразделяется на ближнее и дальнее.

У ближнего оружия есть свои последовательности ударов (комбо). В комбо определены урон, последовательность, условия (к примеру, может быть выполнено только в воздухе) и воспроизводимая анимация.

У дальнего оружия свои размер магазина, возможные и установленные дополнительные устройства, анимация/кривая отдачи, начальная скорость боеприпасов и возможные и используемые боеприпасы. У боеприпасов свои скорость, урон и тип.

Дополнительные устройства крепятся к дальнему оружию, улучшая характеристики оружия или добавляя ему какие-либо свойства.

Одежда может использоваться игроком и как элемент внешнего вида, и как защита. При любом из применений у одежды есть некоторый коэффициент защиты.

Аугментации и импланты улучшают характеристики персонажа. Заменяют или дополняют какую-либо часть тела персонажа. У каждого свой набор действий. Действия применяются либо пока экипированы, либо в определённый момент.

Все предметы самой высокой редкости имеются в одном единственном экземпляре, а единственный способ их получения – специальные задания.

Так как большинство игр данного жанра представляют собой одиночный сюжет в многопользовательской игре, что воспринимается не самым приятным образом, из заготовленных частей генерируются как побочные, так и сюжетные задания, но не задания для уникальных предметов. Задания для уникальных предметов прорабатываются и создаются вручную из тех же частей.

Сюжетные задания генерируются для конкретного игрока или группы игроков и выполняются только ими.

За побочное задание может любой игрок. Побочные задания даются неигровыми персонажами. Выполнить побочное задание может один игрок или группа игроков. За задания предполагается какая-либо награда.

Специальные задания выполняются в одиночку.

Для каждой из заготовленных частей определены возможные продолжения в виде других (либо той же самой) частей.

Вся вышеописанная информация должна храниться в удобном для редактирования и обработки виде. Единственный способ достичь этого удобства – создать базу данных.

* 1. Техническое задание
     1. Функциональные требования

В базе данных должна храниться информация об игроках: логин, пароль, почта, страна, никнейм, изображение профиля и список друзей. Так же должны храниться персонажи, как игроков, так и неигровые, их характеристики, имеющиеся и используемые умения, имеющиеся у них предметы, сами предметы, их свойства, а также задания.

У персонажа кроме характеристик должны храниться имя, максимальное здоровье, % здоровья, а так же точка, к которой необходимо его возвращать, Mesh и подобранные предметы.

У предметов необходимо хранить название, редкость, mesh, информацию о том, что это за предмет.

Для оружия необходимо хранить место крепления и необходимые анимации, для дальнего – устройства, размер магазина, боеприпасы, начальную скорость боеприпаса и кривую отдачи.

Для ближнего предусмотрены комбо. Они содержат саму последовательность ударов, наносимый урон, анимацию и необходимые для выполнения условия.

У одежды имеется такая информация, как количество подавляемого урона и тип (внешний вид или полноценная защита).

У аугментаций и имплантов необходимо хранить то, как они действуют, что делают и где располагаются на персонаже.

Необходимо хранить задания, их тип, кто принял и кто выполнил.

Задания составляются из частей, поэтому необходимо хранить части, подходящие продолжения и из каких частей состоит конкретное задание.

* + 1. Минимальные системные требования
* Процессор: 64 bit, 2 ядра;
* Операционная система: 64 bit;
* Оперативная память: 2 ГБ;
* RAID 1 и выше;

RAID – технология виртуализации данных, которая объединяет несколько дисков в логический элемент для избыточности и повышения производительности.

RAID 1 – массив из двух (или более) дисков, являющихся полными копиями друг друга.

* + 1. Требование к документации

Необходимо создать руководство администратора базы данных.

* 1. Выбор средств разработки
     1. Выбор СУБД

1. PostgreSQL – объектно-реляционная СУБД.

Преимущества:

* высокопроизводительные и надёжные механизмы транзакций и репликации;
* расширяемая система встроенных языков программирования;
* наследование;
* возможность индексирования геометрических объектов;
* бесплатность.

Недостатки:

* низкая производительность при чтении;
* сложность.

1. MySQL – реляционная СУБД, разрабатывается и поддерживается Oracle. Обычно используется в качестве сервера. Гибкость обеспечивается поддержкой большого количества типов таблиц.

Преимущества:

* простота;
* скорость;
* безопасность;
* масштабируемость – может быть использована для работы и с малыми, и с большими объёмами данных;
* бесплатность.

Недостатки:

* недостаточная надёжность;
* низкая скорость разработки;
* проблемы с многопоточностью.

1. Microsoft SQL Server – Реляционная СУБД, разработанная Microsoft. Используется для работы с базами данных размером от персональных до крупных баз данных масштаба предприятия. Использует Transact-SQL (T-SQL) – реализацию ANSI/ISO по SQL с расширениями.

Преимущества:

* производительность;
* надёжность и безопасность;
* простота;
* масштабируемость.

Недостатки:

* даже при тщательной настройке способен занять все доступные ресурсы.
* высокая цена.
  + 1. Выбор СУБД. Вывод

Для разработки была выбрана СУБД PostgreSQL, так как в ней имеются необходимые возможности для реализации требуемой БД.

* + 1. Дополнительные средства разработки

Так же для создания базы данных были выбраны следующие средства разработки:

* Java – Объектно-ориентированный язык программирования;
* Intellij IDEA – среда разработки для Java;
* Spring framework – универсальный фреймворк с открытым исходным кодом для Java;
* Hibernate – реализация спецификации JPA, предназначенная для решения задач объектно-реляционного отображения (ORM);
* Docker – программное обеспечение для автоматизации развёртывания и управления приложениями в средах с поддержкой контейнеризации;
* Maven – фреймворк для автоматизации сборки проектов на основе описания их структуры в файлах на языке POM.

1. Разработка Базы Данных
   1. Создание структуры
      1. Начальная структура

Начальная структура базы данных представлена на рисунке 1.

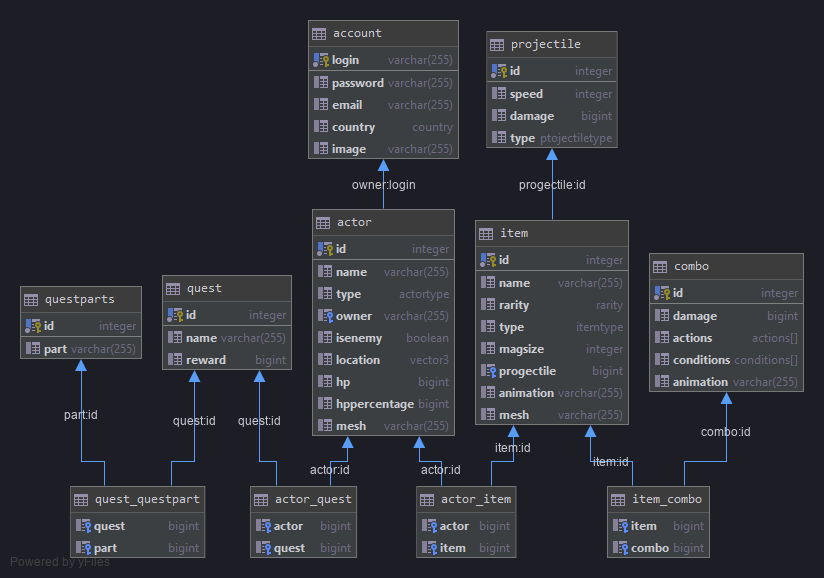


Рисунок 1 – начальная структура.

Проблема данной базы данных в избыточности данных в таблицах, лишних полях, которые могут никогда не использоваться конкретным предметом и неудобстве определения предмета.

База данных будет разрабатываться на Java для удобства как разработки, так и последующей работы с ней.

* + 1. Нормализация.
    2. Конечная структура базы данных
  1. Разработка SQL запросов
     1. Создание таблиц

Account.java

@Entity  
@Table(name = "Account")  
public class Account {  
@Id  
@Column(nullable = false)  
private String login**;**@Column(nullable = false)  
private String password**;**@Column  
private String name**;**@ManyToMany  
private List<Account> friends**;**@Column  
private String image**;**@Enumerated(EnumType.STRING)  
@Column  
private Country country = Country.NOT\_SELECTED**;**

// Конструкторы, геттеры и сеттеры

}

* + 1. Запросы на выборку

Использованные источники

Postgresql <https://ru.wikipedia.org/wiki/PostgreSQL>

Недостатки postgresql <https://tproger.ru/translations/sqlite-mysql-postgresql-comparison/>

Достоинства и недостатки MySQL [https://depix.ru/articles/sistema \_upravleniya\_bazami\_dannyh\_mysql](https://depix.ru/articles/sistema%20_upravleniya_bazami_dannyh_mysql), <http://drach.pro/blog/hi-tech/item/145-db-comparison>

MySQL <https://ru.wikipedia.org/wiki/MySQL>

Microsoft SQL Server <https://ru.wikipedia.org/wiki/Microsoft_SQL_Server>

Java <https://ru.wikipedia.org/wiki/Java>

Itellij IDEA <https://ru.wikipedia.org/wiki/IntelliJ_IDEA>

Spring framework <https://ru.wikipedia.org/wiki/Spring_Framework>

Hibernate <https://ru.wikipedia.org/wiki/Hibernate_(библиотека)>

Docker <https://ru.wikipedia.org/wiki/Docker>

Maven <https://ru.wikipedia.org/wiki/Apache_Maven>

База данных <https://ru.wikipedia.org/wiki/База_данных>