

УНИВЕРСИТЕТ ИТМО

КАФЕДРА ВТ

КУРСОВАЯ РАБОТА ПО БАЗАМ ДАННЫХ

Этап 4

**Выполнил: Каримов Фаррух
Рахимов Фаррухруз
Факультет: ПИиКТ
Группа: Р3102**

2018 г.

Описание предметной области:

Предметная область- вселенная игры MassEffect. MassEffect- компьютерная игра в жанре ролевого шутера с открытым миром. Отличительной особенностью этой серии игр стала хорошо проработанная история и вселенная. Геймплей позволяет игроку перемещаться между десятками солнечных систем и посещать различные планеты.

В игре существует биотика-способность некоторых форм жизни создавать поля эффекта массы и взаимодействовать с темной энергией, используя частицы нулевого элемента в своих телах. Эти способности можно улучшить с помощью био-усилителей. Биотики могут атаковать врагов издали, поднимать их в воздух, порождать гравитационные вихри и т. п.

Главный герой- капитан Шепард, спасающий галактику от Жнецов[раса разумных машин] вместе со своей командой, которую он собрал со всех концов галактики. Каждый персонаж обладает своей специализацией, навыками и способностями. Перед началом игры можно выбрать имя персонажа, пол, внешность, предысторию и класс. Доступно шесть классов, каждый из которых подразумевает свой стиль ведения боя и позволяет использовать определенные типы оружия. Например, персонаж класса адепт в совершенстве владеет биотикой, но плохо обращается с большей частью оружия, солдат специализируется на применении разнообразного вооружения, а инженер- на взломах систем безопасности и ремонте оборудования. Оставшиеся три класса представляют собой комбинации первых трех.

Выполняя миссии, отряд получает специальные очки опыта, которые необходимы для развития способностей, боевых навыков и пополнения инвентаря.

Как и в большинстве шутеров в MassEffectне обошлось без большого разнообразия брони и оружия. Броня делится на три вида: легкая(упор на защиту от технического/биотического урона, но плохая физическая защита), средняя (средние показатели защиты от любых видов урона) и тяжелая (эффективна только против физического урона и обладает мощными щитами). Каждому классу доступен только определенный уровень брони- например, тяжелая броня доступна только солдату. Также броня одной расы не подходит представителям другой - например, броня люди не могут носить броню сарианцев. Также в бою персонажи защищены силовыми щитами, которые снижают уровень получаемого физического урона. После определенного количества попаданий энергия щита иссякает, и тогда последующий урон отнимает непосредственно очки здоровья. Энергия щита восстанавливается автоматически, если персонаж некоторое время не получает урон. Здоровье же восстанавливается только при использовании меди-геля. Если погибают союзники, то они автоматически воскресают после окончания боя, однако игра заканчивается, если погибает Шепард.

В игре присутствуют такие виды оружия как дробовики, штурмовые и снайперские винтовки, пистолеты, доступны гранаты. Также существуют различные улучшения для оружия и виды боеприпасов, которые можно найти во время миссий или купить в магазине. Оружие не требует перезарядки, однако слишком частая стрельба приводит к его быстрому перегреву, из-за чего необходимо делать паузы между стрельбой.

Перемещение между звездными системами осуществляется с помощью ретрансляторов массы на космическом корабле «Нормандия»(экспериментальный фрегат, созданный учеными Альянса систем(независимое наднациональное правительство, представляющее интересы человечества) совместно с туррианскими специалистами), на борту которого игрок может взаимодействовать с членами экипажа, выбирать экипировку и перемещаться между созвездиями с помощью схематичной карты галактики.

Наземные миссии отряд главного героя выполняет с помощью военного вездехода M35 «Мако», который вооружен пулеметом и 155-мм ускорителем массы.

Данная база данных предназначена для упрощения взаимодействия главного героя с многочисленными торговыми и инвентарными (как собственными, так и команды (находясь на борту «Нормандии»)). Также существует возможность автоматизировать сбор ресурсов и управление персонажами, что снижает исследование тайм-менее мурлы и скучно.

К_Звёзды : таблица описывающая Звёзды во вселенной.		
Атрибуты	Описание	Тип
id	Идентификационный номер звезды.	serial
Название	Название звезды.	varchar
Масса	Масса звезды относительно массы Земли.	real
CREATE TABLE "К_Звёзды" ("id" serial NOT NULL PRIMARY KEY, "Название" varchar NOT NULL, "Масса" real NOT NULL, check("Масса" > 0));		

К_Солнечные_Системы : таблица описывающая солнечные системы.		
Атрибуты	Описание	Тип
id	Идентификационный номер Системы.	serial
Название	Название Солнечной Системы.	varchar
Звезда	Идентификационный номер звезды	int
CREATE TABLE "К_Солнечные_Системы" ("id" serial NOT NULL PRIMARY KEY, "Название" varchar NOT NULL, "Звезда" int NOT NULL REFERENCES "К_Звёзды"("id"));		

К_Месторождения : таблица описывающая Месторождения.		
Атрибуты	Описание	Тип
id	Идентификационный номер месторождения.	serial
Название	Название месторождения.	varchar
Тип	Тип месторождения (спутник или планета)	varchar
Масса	Масса месторождения относительно массы Земли.	real
НаличиеАтмосферы	Да или Нет	boolean
MinТемпература	Минимальная зарегистрированная температура	int
MaxТемпература	Максимальная зарегистрированная температура	int

СолнечнаяСистема	Идентификационный номер Солнечной Системы.	int
<pre>CREATE TABLE "К_Месторождения" ("id" serial NOT NULL PRIMARY KEY, "Название" varchar NOT NULL, "Тип" varchar NOT NULL, "Масса" real NOT NULL, "НаличиеАтмосферы" boolean NOT NULL, "MinТемпература" int NOT NULL, "MaxТемпература" int NOT NULL, "СолнечнаяСистема" int NOT NULL REFERENCES "К_Солнечные_Системы"("id"), check("Масса" > 0), check("Тип" = "Планета" or "Тип" = "Спутник"), check("MinТемпература" <= "MaxТемпература"));</pre>		

К_Полезные_Ископаемые : таблица описывающая полезные ископаемые.		
Атрибуты	Описание	Тип
id	Идентификационный номер минерала	serial
Название	Название минерала	varchar
Описание	Описание минералов	varchar
<pre>CREATE TABLE "К_Полезные_Ископаемые" ("id" serial NOT NULL PRIMARY KEY, "Название" varchar NOT NULL, "Описание" varchar NOT NULL);</pre>		

К_Ископаемые_В_Месторождении : таблица описывающая полезные ископаемые в Месторождениях.		
Атрибуты	Описание	Тип
id	Идентификационный номер.	serial
Месторождение	Идентификационный номер Месторождения.	int
Ресурс	Идентификационный номер ресурса.	int
Количество	Количество единиц ресурса.	bigint
<pre>CREATE TABLE "К_Ископаемые_В_Месторождении" ("id" serial NOT NULL PRIMARY KEY, "Месторождение" int NOT NULL REFERENCES "К_Месторождения"("id"), "Ресурс" int NOT NULL REFERENCES "К_Полезные_Ископаемые"("id"), "Количество" bigint NOT NULL, check("Количество" >= 0));</pre>		

);

К_Ретрансляторы : таблица описывающая ретрансляторы

Атрибуты	Описание	Тип
id	Идентификационный номер ретранслятора	serial
Название	Название ретранслятора	varchar
Местонахождение	Идентификационный номер солнечной системы	int

```
CREATE TABLE "К_Ретрансляторы" (  
    "id" serial NOT NULL PRIMARY KEY,  
    "Название" varchar NOT NULL,  
    "Местонахождение" int NOT NULL REFERENCES  
    "К_Солнечные_Системы"("id")  
);
```

К_Граф_Системы : таблица описывающая перемещения между солнечными системами

Атрибуты	Описание	Тип
id	Идентификационный номер.	serial
Система1	Идентификационный номер первой системы	int
Система2	Идентификационный номер второй системы	int
ПримРаст	Примерная оценка расстояния между СС	int

```
CREATE TABLE "К_Граф_Системы" (  
    "id" serial NOT NULL PRIMARY KEY,  
    "Система1" int NOT NULL REFERENCES "К_Солнечные_Системы"("id"),  
    "Система2" int NOT NULL REFERENCES "К_Солнечные_Системы"("id"),  
    "ПримРаст" int NOT NULL,  
    check("Система1" != "Система2"),  
    check("ПримРаст" >= 0)  
);
```

К_Рассы : таблица описывающая расы персонажей.

Атрибуты	Описание	Тип
id	Идентификационный номер.	serial
Название	Название расы персонажа	varchar
Описание	Описание особенностей	varchar

```
CREATE TABLE "К_Рассы" (  
    "id" serial NOT NULL PRIMARY KEY,
```

```

"Название" varchar NOT NULL,
"Описание" varchar NOT NULL
);

```

К_Виды_Классов : таблица описывающая виды классов персонажей.		
Атрибуты	Описание	Тип
id	идентификационный номер	serial
Название	Название данного класса	varchar
Описание	Описание особенностей	varchar
<pre> CREATE TABLE "К_Виды_Классов" ("id" serial NOT NULL PRIMARY KEY, "Название" varchar NOT NULL, "Описание" varchar NOT NULL); </pre>		

К_Космические_Станции : таблица описывающая различные космические станции.		
Атрибуты	Описание	Тип
id	Идентификационный номер станции.	serial
Название	Название станции.	varchar
Масса	Масса станции относительно массы Земли.	real
СолнечнаяСистема	Идентификационный номер Солнечной Системы.	int
Кредиты	Количество кредит на станции	int
<pre> CREATE TABLE "К_Космические_Станции" ("id" serial NOT NULL PRIMARY KEY, "Название" varchar NOT NULL, "Масса" real NOT NULL, "СолнечнаяСистема" int NOT NULL REFERENCES "К_Солнечные_Системы"("id"), "Кредиты" int NOT NULL, check("Масса" > 0), check("Кредиты" >= 0)); </pre>		

К_Персонажи : таблица описывающая персонажей.		
Атрибуты	Описание	Тип
id	Идентификационный номер персонажа.	serial
Имя	Имя персонажа	varchar

Расса	Идентификационный номер рассы персонажа	int
Класс	Идентификационный номер класса персонажа.	int
Кредиты	Количество кредитов персонажа (≥ 0)	int
Предыстория	Небольшая предыстория персонажа.	varchar
CREATE TABLE "К_Персонажи" ("id" serial NOT NULL PRIMARY KEY, "Имя" varchar NOT NULL, "Расса" int NOT NULL REFERENCES "К_Рассы"("id"), "Класс" int NOT NULL REFERENCES "К_Виды_Классов"("id"), "Кредиты" int NOT NULL, "Предыстория" varchar NOT NULL, check("Кредиты" ≥ 0));		

К_Тип_Предметов : таблица описывающая тип предметов инвентаря.

Атрибуты	Описание	Тип
id	идентификационный номер	serial
Название	Название данного типа	varchar
Описание	Описание особенностей	varchar
CREATE TABLE "К_Тип_Предметов" ("id" serial NOT NULL PRIMARY KEY, "Название" varchar NOT NULL, "Описание" varchar NOT NULL);		

К_Предметы_Инвентаря : таблица описывающая предметы инвентаря.

Атрибуты	Описание	Тип
id	Идентификационный номер предмета.	serial
Название	Название предмета.	varchar
Тип		int
Характеристика		varchar
CREATE TABLE "К_Предметы_Инвентаря" ("id" serial NOT NULL PRIMARY KEY, "Название" varchar NOT NULL "Тип" int NOT NULL REFERENCES "К_Тип_Предметов"("id") "Характеристика" varchar NOT NULL);		

К_Инвентарь_Персонажа : таблица описывающая инвентарь персонажа.

Атрибуты	Описание	Тип
id	Идентификационный номер.	serial
Персонаж	Идентификационный номер персонажа.	int
Предмет	Идентификационный номер предмета инвентаря.	int
Количество	Количество данного предмета у персонажа	int
CREATE TABLE "К_Инвентарь_Персонажа" ("id" serial NOT NULL PRIMARY KEY, "Персонаж" int NOT NULL REFERENCES "К_Персонажи"("id"), "Предмет" int NOT NULL REFERENCES "К_Предметы_Инвентаря"("id"), "Количество" int NOT NULL, check("Количество" >= 0));		

К_Ископаемые_Персонажа: таблица описывающая полезные ископаемые персонажа.

Атрибуты	Описание	Тип
id	Идентификационный номер.	serial
Ресурс	Идентификационный номер ресурса.	int
Персонаж	Идентификационный номер персонажа.	int
Количество	Количество данного ресурса у персонажа	int
CREATE TABLE "К_Ископаемые_Персонажа" ("id" serial NOT NULL PRIMARY KEY, "Ресурс" int NOT NULL REFERENCES "К_Полезные_Ископаемые"("id"), "Персонаж" int NOT NULL REFERENCES "К_Месторождения"("id"), "Количество" bigint NOT NULL, check("Количество" >= 0));		

К_Предметы_Станции : таблица описывающая предметы инвентаря на станции.

Атрибуты	Описание	Тип
id	Идентификационный	serial

	номер.	
Станция	Идентификационный номер станции.	int
Предмет	Идентификационный номер предмета инвентаря.	int
Цена	Цена предмета на этой станции в кредитах.	int
Количество	Количество данного предмета на станции	int
<pre>CREATE TABLE "К_Предметы_Станции" ("id" serial NOT NULL PRIMARY KEY, "Станция" int NOT NULL REFERENCES "К_Космические_Станции"("id"), "Предмет" int NOT NULL REFERENCES "К_Предметы_Инвентаря"("id"), "Цена" int NOT NULL, "Количество" int NOT NULL, check("Количество" >= 0), check("Цена" >= 0));</pre>		

Необходимо автоматизировать :

А. Сбор ресурсов и минералов с поверхности планет и спутников (что делает исследование планет и спутников менее нудным и скучным).

Для этого нам необходимо знать какие ресурсы и минералы (далее полезные ископаемые) есть в этой предметной области (далее ПОб).

Создаём таблицу К_Полезные_Ископаемые, в ней будем хранить названия и описания полезных ископаемых.

Также нам необходимо знать, на каких планетах и спутниках — какие полезные ископаемы находятся, то создадим таблицы : К_Планеты и К_Спутники, а также вспомогательные таблицы К_Ископаемые_На_Планете и К_Ископаемые_На_Спутнике;

Так как в данной ПОб возможны межзвёздные перелёты, то у нас появляются различные солнечные системы, и тем самым создадим таблицы К_Звёзды и К_СолнечныеСистемы.

О солнечной системы нам необходимо знать : название солнечной системы, а также информацию о её звезде — название звезды, масса звезды.

О планетах и спутниках — необходимо знать их названия, их массу, наличие атмосферы, минимальную и максимальную зарегистрированную температуру. А также о планетах — в какой солнечной системе они находятся, а о спутниках — спутником какой планеты они являются.

Вспомогательные таблицы К_Ископаемые_На_Планете и К_Ископаемые_На_Спутнике — также содержат информацию о количествах полезных ископаемых на планетах и спутниках соответственно.

В. Взаимодействие главного героя с многочисленными торговцами и инвентарём.

Для этого нам необходимо иметь информацию о предметах инвентаря; для этого создадим таблицу К_Предметы_Инвентаря , в которой будем хранить названия предметов инвентаря.

Торговцы находятся на космических станциях; для этого создадим таблицы К_Космические_Станции и К_Предметы_Станции.

О космической станции нам интересно знать — её массу, название и в какой солнечной системе она находится и количество кредитов на станции (кредит — торговая валюта в данной ПОБ).

К_Предметы_Станции — вспомогательная таблица — хранящая информацию о предметах инвентаря которые находятся на станциях, их количество и их стоимость в кредитах.

О персонажах интересно знать — их имена, количество их кредитов и предысторию; для этого создадим таблицу К_Персонажи, где будем хранить выше упомянутые данные.

Также нам важно хранить информацию о предметах инвентаря персонажей; для этого создадим вспомогательную таблицу К_Инвентарь_Персонаж — в которой будем хранить эту информацию.

*** Предметы инвентаря бывают различных типов — шиты, аптечки и оружия. Создадим и для них таблицы : К_Тип_Предметов в которой будем хранить название типа и его описание.

В данной ПОБ — персонажи являются представителями различных Рас и Классов; по этому — создадим таблицы К_Рас и К_Виды_Классов, и добавим в таблицу К_Персонажи — такие поля как Раса и Класс.

С. Поиск оптимального маршрута для перемещения из одной СС — в другую СС (есть возможности выбора кратчайшего пути или минимального количества перелётов между СС).

Для перемещений на большие расстояния — в данной ПОБ есть ретрансляторы. Ретрансляторы — это устройства которые позволяют телепортироваться от одного ретранслятора — к другому. В одной СС — может быть только одна (или 0). По

этому создадим таблицу К_Ретрансляторы в которой будем хранить информацию о ретрансляторах (их название и месторождения — в какой СС они находятся).

Также создадим таблицу К_Граф_Системы в которой будем хранить какое примерно расстояние необходимо пролететь между двумя СС.

Этап 4

Возможные варианты запросов в нашу базу данных

- 1) Какие полезные ископаемые - есть на поверхности данной планеты/спутника, и в каком количестве ?
- 2) На каких плантах/спутниках (+ солнечная система) есть данный ресурс ?
- 3) На каких плантах/спутниках - можно найти ресурс - в количестве не меньше чем x ?
- 4) Собрать x единиц ресурса у с планеты/спутника. (персонаж и планеты/спутника - должны находиться в одной солнечной системе)
- 5) Предметы данной станции ?
- 6.1) На каких станциях можно купить данный предмет (в каком-то количестве) + цена?
- 6.2) На каких станциях можно купить данный предмет + цена (упорядочить по возрастанию цены) ?
- 7) Купить x единиц предмета на станции у.
(персонаж и станция - должны находиться в одной солнечной системе)
- 8) Продать x единиц данного предмета - станции у.
(персонаж и станция - должны находиться в одной солнечной системе)
- 9) Примерная оценка для перелёта из СС1 в СС2 (выбор кратчайшего пути)
- 10) Примерная оценка для перелёта из СС1 в СС2 (выбор минимального количества перелётов между СС)
- 11) Через какие СС/ретрансляторы - надо проходить для перелёта из СС1 в СС2 ?

Комбинированные запросы

- 12) На каких планетах/спутниках (+ солнечная система) есть данный ресурс
Упорядочить по количеству или оценке перемещения ?
- 13) На каких станциях можно купить данный предмет + цена (упорядочить по возрастанию цены)
Упорядочить по оценке перемещения ?
- 12) На какой СС можно найти данные ресурсы (в пределах одной СС)
Упорядочить по оценке перемещения ?

К_Звёзды;
 К_Солнечные_Системы;
 К_Месторождения;
 Hash Index (Название) (запрос 1)
 CREATE INDEX index_name ON К_Месторождения USING hash(Название);
 К_Полезные_Ископаемые;
 Hash Index (Название) (запрос 2)
 CREATE INDEX index_name ON К_Полезные_Ископаемые USING hash(Название);
 К_Ископаемые_В_Месторождении;
 B-Tree (Количество) (запрос 3)
 CREATE INDEX index_name ON К_Ископаемые_В_Месторождении USING
 btree(Количество);
 К_Ретрансляторы;
 К_Граф_Системы;
 К_Рассы;
 К_Виды_Классов;
 К_Космические_Станции;
 Hash Index (Название) (запрос 5)
 CREATE INDEX index_name ON К_Космические_Станции USING hash(Название);
 К_Персонажи;
 К_Тип_Предметов;
 К_Предметы_Инвентаря;
 Hash Index (Название) (запрос 6)
 CREATE INDEX index_name ON К_Предметы_Инвентаря USING hash(Название);
 К_Инвентарь_Персонажа;
 К_Ископаемые_Персонажа;
 К_Предметы_Станции;
 B-Tree (Цена) (запрос 6.1)
 CREATE INDEX index_name ON К_Предметы_Станции1 USING btree(Цена);
 B-Tree (Количество) (запрос 6.2)
 CREATE INDEX index_name ON К_Предметы_Станции2 USING btree(Количество);

Создадим такие индексы.

```
CREATE INDEX index_Месторождения
  ON К_Месторождения USING hash(Название);

CREATE INDEX index_Полезные_Ископаемые
  ON К_Полезные_Ископаемые USING hash(Название);

CREATE INDEX index_Ископаемые_В_Месторождении
  ON К_Ископаемые_В_Месторождении USING btree(Количество);
```

```
CREATE INDEX index_Космические_Станции
    ON K_Космические_Станции USING hash(Название);

CREATE INDEX index_Предметы_Станции1
    ON K_Предметы_Станции USING btree(Цена);

CREATE INDEX index_Предметы_Инвентаря
    ON K_Предметы_Инвентаря USING hash(Название);

CREATE INDEX index_Предметы_Станции2
    ON K_Предметы_Станции USING btree(Количество);
```