**Онлайн магазин для продажи игровых предметов в игре CS GO**

* Целевой аудиторией нашего сайта являются геймеры
* Площадка будет использоваться для продажи, обмена и покупки игровых товар за реальные деньги.

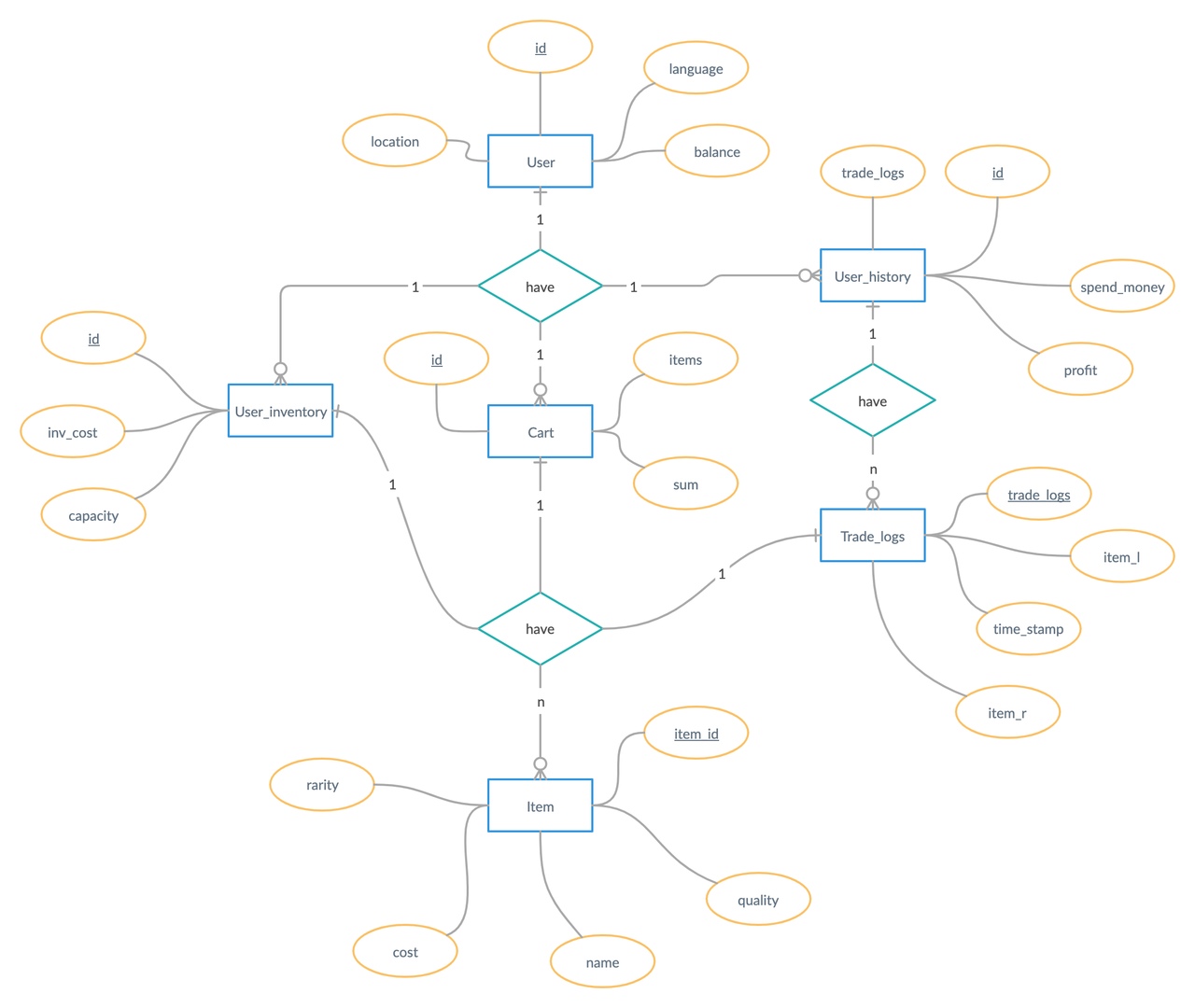
**Функциональные требования**

1. Пользователь, может авторизоваться используя аккаунт steam.
2. Необходимо оповестить пользователя о необходимости открыть инвентарь для совершения обменов и продаж.
3. Любой пользователь с открытым инвентарём должен иметь возможность продавать и обменивать игровые предметы, при этом средства зачисляются на его счет на сайте.
4. Любой пользователь должен иметь возможность приобрести игровые предметы, и получить ее в свой инвентарь на сайте.
5. Любой пользователь должен иметь возможность вывести средства на карту или онлайн кошелек.
6. Ввести запрет на продажу предметов с типом: “сувенирный”.
7. Реализовать функцию поиска конкретных предметов.
8. Реализовать возможность конвертации валют.
9. Реализовать возможность выбора языка.
10. Наличие внутреннего инвентаря у каждого пользователя.
11. Реализовать корзину для покупки предметов.

**Ограничения на данные**

1. Один пользователь может иметь только одну корзину, один инвентарь, одну историю о его транзакциях.
2. Один инвентарь имеет до 250 предметов.
3. Одна корзина имеет до (250 – инвентарь(количество)) предметов.
4. Одна история транзакций пользователя имеет n транзакций.
5. Одна транзакция имеет до 20 предметов.
6. Одна корзина имеет одного пользователя
7. Одна история транзакций пользователя имеет одного пользователя
8. У одного пользовательского инвентаря существует один пользователь
9. Пользователь может обменять только один предмет за операцию.

**ER**

****

**Создание базы данных**

CREATE DATABASE CSGOMARKET;

CREATE TABLE User (

Id int NOT NULL PRIMARY KEY,

Location VARCHAR(255),

Language VARCHAR(255),

Balance int

);

CREATE TABLE User\_inventory (

Id int NOT NULL PRIMARY KEY,

Inv\_cost int,

Capacity int,

foreign key (id) references User(id)

);

CREATE TABLE Cart (

Id int NOT NULL PRIMARY KEY,

Sum int,

foreign key (id) references User(id)

);

CREATE TABLE User\_history (

Id int NOT NULL PRIMARY KEY,

Spend\_money int,

Profit int,

foreign key (id) references User(id)

);

CREATE TABLE Trade\_logs(

Trade\_log int NOT NULL PRIMARY KEY,

Item\_l VARCHAR(255),

Item\_r VARCHAR (255),

Time\_stamp TIMESTAMP(14)

);

CREATE TABLE Item (

Item\_id int NOT NULL PRIMARY KEY,

Rarity VARCHAR (255),

Cost int,

Name VARCHAR (255),

Quality VARCHAR (255)

);

CREATE TABLE user\_item (

User\_id int NOT NULL PRIMARY KEY,

Item\_id int NOT NULL PRIMARY KEY

);

CREATE TABLE user\_cart (

User\_id int NOT NULL PRIMARY KEY,

Item\_id int NOT NULL PRIMARY KEY

);

CREATE TABLE user\_logs (

User\_id int NOT NULL PRIMARY KEY,

Trade\_log int NOT NULL PRIMARY KEY

);

**Нормализация**

База данных удовлетворяет первой нормальной форме, поскольку выполняются условия:

1. Нет повторяющихся строк
2. Все атрибуты простые
3. Все значения скалярные

База данных удовлетворяет второй нормальной форме, поскольку выполняются условия:

1. Таблица находится в первой нормальной форме
2. У таблицы должен быть первичный ключ
3. Все атрибуты должны описывать первичный ключ целиком, а не какую-то часть первичного блока.

База данных удовлетворяет третьей нормальной форме, поскольку выполняются условия:

1. Таблица находится во второй нормальной форме
2. Не должно быть зависимостей одних не ключевых атрибутов от других. Все атрибуты зависят только от первичного ключа.

**Запросы**

1. *Узнаем какие 10 самых дорогих предмет лежат у пользователя (id = 7 и редкость предметов = rare) в корзине*

select \* from item it

inner join user\_item uit

on id = 7 and type = “choose” and it.rarity = “rare”

order by it.cost

limit 10

1. *Узнать 10 пользователей за октябрь с наибольшим профитом*

select location, id, profit from user

inner join user\_history uh on user.id = uh.id

inner join user\_logs ul on uh.id = ul.id

inner join trade\_logs tl on tl.trade\_log = ul.trade\_log

and month(time\_stamp) = 10

order by profit desc

limit 10

1. *Найти id пользователя совершившего наиболее крупную сделку, вывести id и сумму сделки*

select id, cost from user u

inner join user\_logs ul on u.id = ul.User\_id

inner join Trade\_logs tl on ul.Trade\_log = tl.trade\_logs

inner join Item it on tl.l\_item = it.item\_id

order by it.cost

limit 1

1. *Найти пользователя с id = 3*

select \* fromUser

where id = 3

1. *Найти id пользователя у которого в инвентаре находится самый дорогой предмет и вывести id пользователя, id предмета, цену предмета и его название*

select u.id, it.item\_id, it.cost, it.name from User u

inner join User\_item uit on uit.User\_id = u.id

inner join Item it on it.item\_id = uit.Item\_id

order by it.cost

limit 1