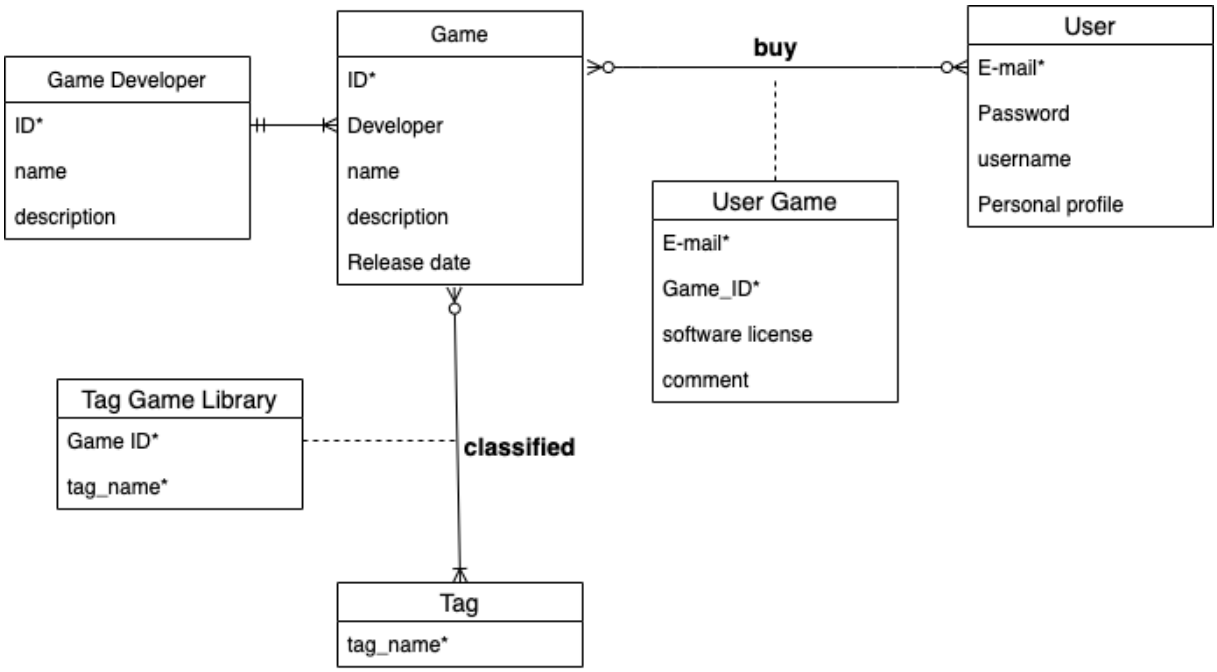
Question 1

Describe in your own words some of the features of the application/website that you have chosen. You do not need to describe all the site's features, just the ones that you will work with in the other points of this exercise.

What I want to describe is the **Steam** gaming platform by Valve. Game Developers can publish their work on the steam platform. Users can purchase computer games though Steam Store. Games can be classified with tags so that users can quickly find the games they are interested in Steam. Once the game is bought, an unique software license is permanently attached to the user's Steam account, allowing them to download the software on any compatible device. The Steam platform also allows users to comment on games they have purchased.

Question 2

Draw an ER diagram



Question 3

Write a create/drop script for the exacttables in your model.

Listing 1: Steam_model

```
drop database if exists Steam_model;
create database Steam_model;
use Steam_model;

drop table if exists 'Tag Gmae Library';
drop table if exists 'User Game';
drop table if exists 'tag';
drop table if exists 'Game';
drop table if exists 'Game Developer';
drop table if exists 'User';
```

Question 3 continued on next page...

```
11
12 create table User(
13     'E-mail' varchar(100) not null primary key,
14     'password' varchar(100) not null,
15     'username' varchar(100) not null,
16     'Personal profile' varchar(500),
17     CONSTRAINT user_name UNIQUE ('username')
18 );
19
20 create table 'Game Developer'(
21     'ID' INTEGER not null primary key auto_increment,
22     'name' varchar(100) not null ,
23     'description' varchar(500),
24     CONSTRAINT game_dev_name UNIQUE ('name')
25 );
26
27 create table 'Game'(
28     'Game_ID' INTEGER not null primary key auto_increment,
29     'Developer' varchar(100) not null,
30     'name' varchar(100) not null,
31     'description' varchar(500),
32     'Release date' DATE
33     CONSTRAINT game_dev foreign key ('Developer') references 'Game Developer'(name)
34 );
35
36 create table 'User_Game'(
37     'E-mail' varchar(100) not null,
38     'Game_ID' INTEGER not null,
39     'Software license' varchar(100) not null unique,
40     'comment' varchar(500),
41     CONSTRAINT UG_email foreign key ('E-mail') references  User('E-mail'),
42     CONSTRAINT UG_gid foreign key ('Game_ID') references Game('Game_ID'),
43     CONSTRAINT primary key ('E-mail','Game_ID')
44 );
45
46 create table 'tag'(
47   'tag name' varchar(100) not null primary key
48 );
49
50 create table 'Tag Game Library'(
51     'Game_ID' INTEGER not null,
52     'tag name' varchar(100) not null,
53     CONSTRAINT TGL_gid foreign key ('Game_ID') references Game('Game_ID'),
54     CONSTRAINT TGL_tag foreign key ('tag name') references tag('tag name'),
55     CONSTRAINT primary key ('Game_ID','tag name')
56 );
```

Question 4

Describe two use cases of your chosen application or website in terms of SQL queries that involve some form of JOIN.

(a) Query the list of games released by a game developer, in descending order of release time:

Listing 2: Answer

```
1 SELECT G.name,G.description,G.Developer,G.'Release date' FROM 'Game' G
2 INNER JOIN 'Game Developer' GD ON GD.name=G.Developer
3 where GD.name=?
4 ORDER BY G.'Release date' DESC ;
```

(b) Query all games with the same tag as all games owned by a player

Listing 3: Answer

```
1 SELECT G.name,G.description,G.Developer,G.'Release date' FROM 'Game' G
2 INNER JOIN 'Tag Game Library' TGLL on G.Game_ID = TGLL.Game_ID
3 INNER JOIN 'Tag Game Library' TGLR on TGLL.'tag name'=TGLR.'tag name'
4 INNER JOIN User_Game UG on TGLR.Game_ID = UG.Game_ID
5 where UG.'E-mail'=?;
```