–TABLE DDL COMMANDS–

```
CREATE TABLE User(UserId INT Primary Key,
Username VARCHAR(255) NOT NULL,
Email VARCHAR(255) NOT NULL,
Password VARCHAR(255) NOT NULL);

CREATE TABLE Developer (DeveloperID INT Primary Key, Name VARCHAR(255),
Country VARCHAR(255));

CREATE TABLE Game(GameID INT Primary Key, Title VARCHAR(255) NOT NULL,
ReleaseDate VARCHAR(255),
Price FLOAT,
DeveloperID INT,
Foreign Key (DeveloperID) References Developer(DeveloperID));

CREATE TABLE Plays (UserID INT, GameID INT,
Primary Key (UserID, GameID),
Foreign Key (UserID) References User(UserID),
Foreign Key (GameID) References Game(GameID) );

CREATE TABLE Tag(TagID INT Primary Key,
TagName VARCHAR(255) NOT NULL);

CREATE TABLE Recommendation(UserID INT, GameID INT, Rating INT, RecommendDate
VARCHAR(255),
Primary Key (UserID, GameID),
Foreign Key (UserID) References User(UserID), Foreign Key (GameID) References
Game(GameID));

CREATE TABLE GameTags (GameID INT, TagID INT,
Primary Key (GameID, TagID),
Foreign Key (GameID) References Game(GameID),
Foreign Key (TagID) References Tag(TagID));
```
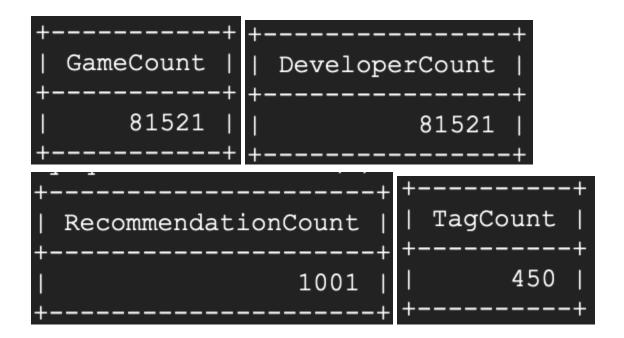
```
mysql> show tables;
+-------------------------+
| Tables_in_GameRecommender |
+-------------------------+
| Developer               |
| Game                    |
| GameTags                |
| Plays                   |
| Recommendation          |
| Tag                     |
| User                    |
| gamefeatures            |
+-------------------------+
8 rows in set (0.02 sec)
```

```
+------------------+    +------------------------+
|   GameCount   |    |   DeveloperCount   |
+------------------+    +------------------------+
|          81521 |    |                 81521 |
+------------------+    +------------------------+
```

```
+----------------------------------+    +--------------+
|   RecommendationCount          |    |   TagCount |
+----------------------------------+    +--------------+
|                           1001 |    |          450 |
+----------------------------------+    +--------------+
```

–QUERIES–

1. Finding Developer/Developer Teams that make above average games
–Output has 3 rows

```
+----------------------+------------+------------+
| Name                 | AvgRating  | GameCount  |
+----------------------+------------+------------+
| Turtle Rock Studios  |    6.7027  |       222  |
| Team Meat            |    6.6085  |       212  |
| Gearbox Software     |    6.5783  |       166  |
+----------------------+------------+------------+
3 rows in set (0.26 sec)
```

```
| -> Limit: 15 row(s)  (actual time=1245.663..1245.664 rows=3 loops=1)
    -> Sort: AvgRating DESC, GameCount DESC  (actual time=1245.663..1245.663 rows=3 loops=1)
      -> Filter: (avg(r.Rating) > (select #2))  (actual time=1245.571..1245.574 rows=3 loops=1)
        -> Table scan on <temporary>  (actual time=1242.037..1242.041 rows=6 loops=1)
          -> Aggregate using temporary table  (actual time=1242.034..1242.034 rows=6 loops=1)
            -> Nested loop inner join  (cost=1737.57 rows=1001) (actual time=736.288..1239.841 rows=1001 loops=1)
              -> Nested loop inner join  (cost=636.47 rows=1001) (actual time=236.637..738.281 rows=1001 loops=1)
                -> Table scan on r  (cost=105.10 rows=1001) (actual time=236.596..735.781 rows=1001 loops=1)
                -> Filter: (g.DeveloperID is not null)  (cost=0.43 rows=1) (actual time=0.002..0.002 rows=1 loops=1001)
                  -> Single-row index lookup on g using PRIMARY (GameID=r.GameID)  (cost=0.43 rows=1) (actual time=0.002..0.002 rows=1 loops=1001)
              -> Single-row index lookup on d using PRIMARY (DeveloperID=g.DeveloperID)  (cost=1.00 rows=1) (actual time=0.501..0.501 rows=1 loops=1001)
      -> Select #2 (subquery in condition; run only once)
        -> Aggregate: avg(r2.Rating)  (cost=205.20 rows=1) (actual time=0.616..0.617 rows=1 loops=1)
          -> Table scan on r2  (cost=105.10 rows=1001) (actual time=0.203..0.479 rows=1001 loops=1)
|
```

2. Find top-rated games through average rating
–Output has 6 rows

```
+--------+-------------------------+---------------------------------------------+-----------+
| GameID | Title                   | Developer                                   | AvgRating |
+--------+-------------------------+---------------------------------------------+-----------+
|      3 | Evolve Stage 2          | Turtle Rock Studios                         |    6.7027 |
|      1 | Super Meat Boy          | Team Meat                                   |    6.6085 |
|      4 | Borderlands 3           | Gearbox Software                            |    6.5783 |
|      2 | DCS World Steam Edition | Eagle Dynamics SA                           |    6.4398 |
|      5 | BioShock Infinite       | Irrational Games,Virtual Programming (Linux)|    6.3110 |
|      0 | Title                   | Name                                        |    0.0000 |
+--------+-------------------------+---------------------------------------------+-----------+
```

```
 -> Limit: 15 row(s)  (actual time=1013.965..1013.967 rows=6 loops=1)
   -> Sort: AvgRating DESC, limit input to 15 row(s) per chunk  (actual time=1013.964..1013.966 rows=6 loops=1)
     -> Table scan on <temporary>  (actual time=1011.970..1011.974 rows=6 loops=1)
       -> Aggregate using temporary table  (actual time=1011.968..1011.968 rows=6 loops=1)
         -> Nested loop inner join  (cost=1706.70 rows=1001) (actual time=765.752..1008.875 rows=1001 loops=1)
           -> Nested loop inner join  (cost=605.60 rows=1001) (actual time=498.723..739.592 rows=1001 loops=1)
             -> Table scan on r  (cost=105.10 rows=1001) (actual time=498.614..736.635 rows=1001 loops=1)
             -> Filter: (g.DeveloperID is not null)  (cost=0.40 rows=1) (actual time=0.003..0.003 rows=1 loops=1001)
                 -> Single-row index lookup on g using PRIMARY (GameID=r.GameID)  (cost=0.40 rows=1) (actual time=0.002..0.002 rows=1 loops=1001)
           -> Single-row index lookup on d using PRIMARY (DeveloperID=g.DeveloperID)  (cost=1.00 rows=1) (actual time=0.269..0.269 rows=1 loops=1001)
|
```

3. Find a specific game using a specific tag for both Adventure and Indie

4. Recommend Game based on games played

```
 -> Limit: 15 row(s)  (actual time=1013.965..1013.967 rows=6 loops=1)
   -> Sort: AvgRating DESC, limit input to 15 row(s) per chunk  (actual time=1013.964..1013.966 rows=6 loops=1)
     -> Table scan on <temporary>  (actual time=1011.970..1011.974 rows=6 loops=1)
       -> Aggregate using temporary table  (actual time=1011.968..1011.968 rows=6 loops=1)
         -> Nested loop inner join  (cost=1706.70 rows=1001) (actual time=765.752..1008.875 rows=1001 loops=1)
           -> Nested loop inner join  (cost=605.60 rows=1001) (actual time=498.723..739.592 rows=1001 loops=1)
             -> Table scan on r  (cost=105.10 rows=1001) (actual time=498.614..736.635 rows=1001 loops=1)
             -> Filter: (g.DeveloperID is not null)  (cost=0.40 rows=1) (actual time=0.003..0.003 rows=1 loops=1001)
                 -> Single-row index lookup on g using PRIMARY (GameID=r.GameID)  (cost=0.40 rows=1) (actual time=0.002..0.002 rows=1 loops=1001)
           -> Single-row index lookup on d using PRIMARY (DeveloperID=g.DeveloperID)  (cost=1.00 rows=1) (actual time=0.269..0.269 rows=1 loops=1001)
```