Database Management System
(UE20CS301)

E-sports Database Management

Submitted by:

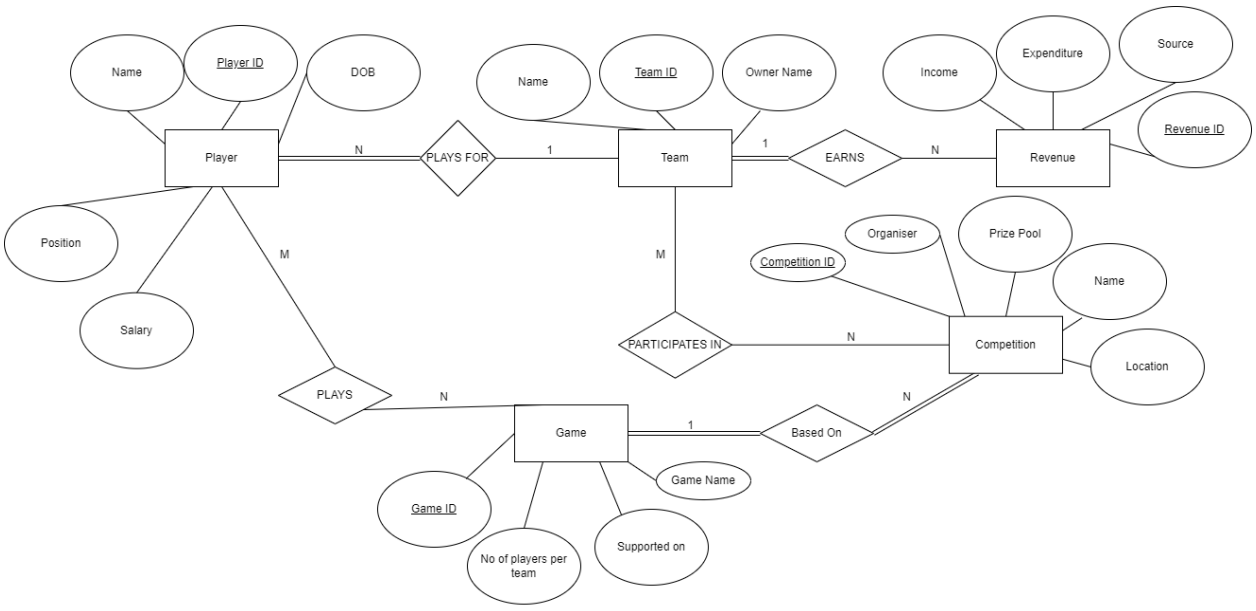
Shishira Bhat O
PES1UG20CS397
5th Sem 'G' Section

Scope Of the Project

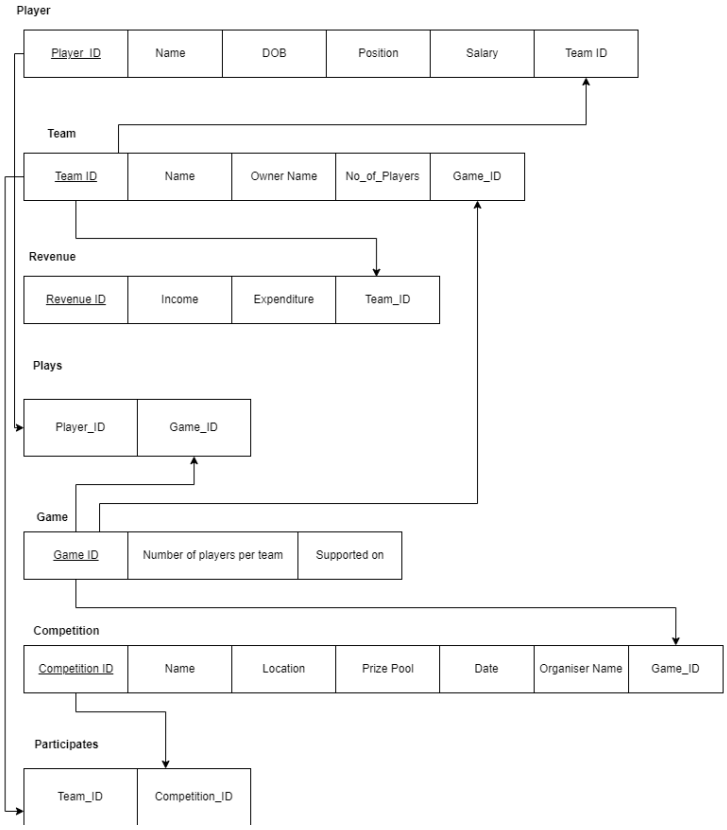
The recent coronavirus pandemic caused many operations to shift to an online mode due to which people invested more into Electronic Gadgets like Mobile Phones and Personal Computers. As a result more people started playing video games and hence as more people began playing these games, the popularity of the games,the popularity of the competitions held for these games and the popularity of the teams that participated in these competitions began to rise.

The data that's being generated thanks to the increase in popularity of the games and the teams requires a proper Database to be stored. This project aims to do the same.This project is a simple implementation of the expected functionalities and structure of such a database.

ER Diagram



Relational Schema



DDL Statements - Building the database

1)player table:

CREATE TABLE Player_397(Player_ID varchar(255) NOT NULL, Player_Name varchar(255), Position varchar(255), DOB DATE,Salary int,Team_ID varchar(255), PRIMARY KEY(Player_ID),FOREIGN KEY(Team_ID) REFERENCES Team_397(Team_ID));

2)plays table:
CREATE TABLE Plays_397(Player_ID varchar(255),Game_ID varchar(255),FOREIGN KEY(Player_ID) REFERENCES Player_397(Player_ID),FOREIGN KEY(Game_ID) REFERENCES Game_397(Game_ID),CONSTRAINT game_player_unique UNIQUE (Player_ID, Game_ID));

3)game table:

CREATE TABLE Game_397(Game_ID varchar(255) NOT NULL,Game_Name varchar(255), No_of_players_per_team int,No_of_teams_competing int,No_of_players_worldwide int,Creator varchar(255),PRIMARY KEY(Game_ID));

4)revenue table:

CREATE TABLE Revenue_397(Revenue_ID varchar(255) NOT NULL,Source_name varchar(255), Income int,Team_ID varchar(255),PRIMARY KEY(Revenue_ID),FOREIGN KEY(Team_ID) REFERENCES Team_397(Team_ID));

5)team table:

CREATE TABLE Team_397(Team_ID varchar(255) NOT NULL,Team_name varchar(255), No_of_players int,Owner_name varchar(255),Game_ID varchar(255),PRIMARY KEY(Team_ID),FOREIGN KEY(Game_ID) REFERENCES Game_397(Game_ID));

6)competition table:

CREATE TABLE Competition_397(Competition_ID varchar(255),Competition_Name varchar(255), Location varchar(255),Prize_pool int,Organiser_name varchar(255),Game_ID varchar(255),PRIMARY KEY(Competition_ID),FOREIGN KEY(Game_ID) REFERENCES Game_397(Game_ID));

7) participates table:
CREATE TABLE Participates_397(Team_ID varchar(255),Competition_ID varchar(255),FOREIGN KEY(Team_ID) REFERENCES Team_397(Team_ID),FOREIGN KEY(Competition_ID) REFERENCES competition_397(Competition_ID),CONSTRAINT team_competition_unique UNIQUE (Team_ID, Competition_ID));

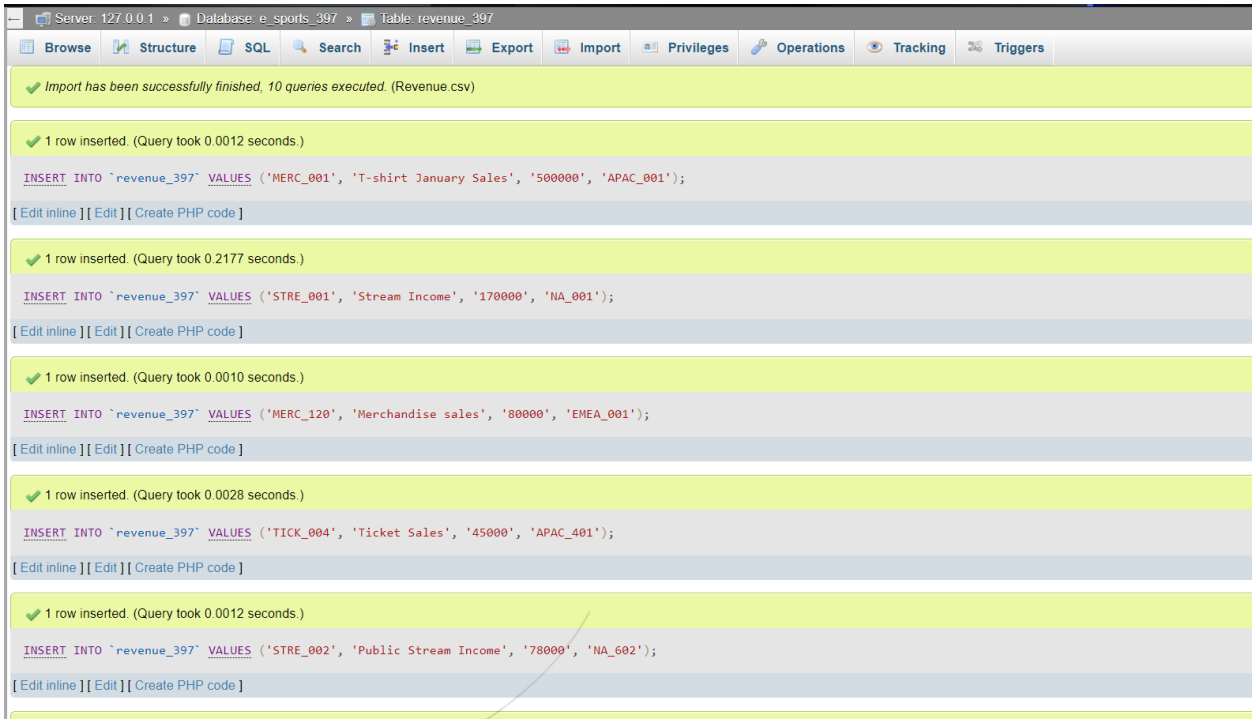
Populating the database

INSERT INTO game_397
(Game_ID,Game_name,Creator,No_of_players_per_team,No_of_teams_competing,No_of_players_worldwide) VALUES
('RIOT_001','Valorant','Riot Games',5,2,700000),
('BLIZ_001','Overwatch 2','Blizzard',5,2,163000),
('RESP_001','Apex Legends','Panic Button Games',3,20,169000),
('EPIC_001','Fortnite','Epic Games',1,99,3000000),
('KRAF_001','PUBG','KRAFTON',4,16,30000000),
('VALV_001','CS:GO','VALVE Corporation',5,2,800000),
('RIOT_002','League of Legends','Riot Games',5,2,2300000),
('PSYO_001','Rocket League','Psyonix',3,2,6000000),
('VALV_002','Dota 2','VALVE Corporation',5,2,666000),
('NAMC_001','Tekken','NAMCO',1,1,5000);

Inserting using query:

```
INSERT INTO game_397 (Game_ID,Game_name,Creator,No_of_players_per_team,No_of_teams_competing,No_of_players_worldwide) VALUES
('RIOT_001','Valorant','Riot Games',5,2,700000),
('BLIZ_001','Overwatch 2','Blizzard',5,2,163000),
('RESP_001','Apex Legends','Panic Button Games',3,20,169000),
('EPIC_001','Fortnite','Epic Games',1,99,3000000),
('KRAF_001','PUBG','KRAFTON',4,16,30000000),
('VALV_001','CS:GO','VALVE Corporation',5,2,800000),
('RIOT_002','League of Legends','Riot Games',5,2,2300000),
('PSYO_001','Rocket League','Psyonix',3,2,6000000),
('VALV_002','Dota 2','VALVE Corporation',5,2,666000),
('NAMC_001','Tekken','NAMCO',1,1,5000);
```

Importing values



Join Queries

1) Natural Join
Get the Competition name along with its prize pool and the game on which the competition was based.

SELECT Competition_Name,Game_Name,Prize_pool FROM competition_397 NATURAL JOIN game_397;

2) Inner Join
Get names of games which were played in different competitions.

SELECT Game_Name,Creator FROM game_397 INNER JOIN competition_397 WHERE game_397.Game_ID = competition_397.Game_ID;

3) Left Outer Join
Get the names of players and their associated teams.

SELECT player_397.Player_Name,team_397.Team_Name FROM player_397 LEFT JOIN team_397 ON team_397.Team_ID = player_397.Team_ID;

4) Right Outer Join
Get the names of teams and the name of revenue associated with them.
SELECT team_397.Team_Name,revenue_397.Source_name FROM revenue_397 RIGHT JOIN team_397 ON revenue_397.Team_ID = team_397.Team_ID;

5) Multiple Nested Join
Get names of players and the names of the competitions that they have participated in.

SELECT player_397.Player_Name,Team_397.Team_Name,Competition_Name FROM participates_397 NATURAL JOIN competition_397 JOIN team_397 ON team_397.Team_ID = participates_397.Team_ID JOIN player_397 ON player_397.Team_ID = team_397.Team_ID;

Output Screenshot:

1) Natural Join:

```
MariaDB [E_sports_397]> SELECT Competition_Name,Game_Name,Prize_pool FROM competition_397 NATURAL JOIN game_397;
+-----------------------+--------------------+------------+
| Competition_Name      | Game_Name          | Prize_pool |
+-----------------------+--------------------+------------+
| Grand Masters         | CS:GO              |    1100000 |
| Fortnite Conquerors   | Fortnite           |    1000000 |
| Apex League           | Apex Legends       |    1500000 |
| VCT Champions         | Valorant           |   15000000 |
| VCT Masters           | Valorant           |    5000000 |
| VCT Masters           | Valorant           |     500000 |
| League Champions      | League of Legends  |    2500000 |
| PUBG Champions        | PUBG               |     800000 |
| Dota conquerors       | Dota 2             |    1000000 |
| Tekken warriors       | Tekken             |     150000 |
| Rocket League Tourney | Rocket League      |    2000000 |
| VCT Champions         | Valorant           |   15000000 |
| Overwatch League      | Overwatch 2        |    2500000 |
+-----------------------+--------------------+------------+
13 rows in set (0.001 sec)
```

2) Inner Join:

```
MariaDB [e_sports_397]> SELECT Game_Name,Creator FROM game_397 INNER JOIN competition_397 WHERE game_397.Game_ID = competition_397.Game_ID;
+-------------------+--------------------+
| Game_Name         | Creator            |
+-------------------+--------------------+
| Overwatch 2       | Blizzard           |
| Fortnite          | Epic Games         |
| PUBG              | KRAFTON            |
| Tekken            | NAMCO              |
| Rocket League     | Psyonix            |
| Apex Legends      | Panic Button Games |
| Valorant          | Riot Games         |
| Valorant          | Riot Games         |
| Valorant          | Riot Games         |
| Valorant          | Riot Games         |
| League of Legends | Riot Games         |
| CS:GO             | VALVE Corporation  |
| Dota 2            | VALVE Corporation  |
+-------------------+--------------------+
13 rows in set (0.078 sec)

MariaDB [e_sports_397]>
```

3) Left

```
ariaDB [e_sports_397]> SELECT player_397.Player_Name,team_397.Team_Name FROM player_397 LEFT JOIN team_397 ON team_397.Team_ID = player_397.Team_ID;
+------------------+----------------------+
| Player_Name      | Team_Name            |
+------------------+----------------------+
| Lionel Aguero    | Counter Logic Gaming |
| Daniel Webber    | Elite Esports        |
| Nicholas Stroll  | GodLike              |
| Carlos Alonso    | TSM                  |
| Thomas Reus      | Nigma Galaxy         |
| Rohit Kumar      | Sentinels            |
| Kim Byoung In    | Echo Fox             |
| Sergio Rodriguez | Cloud9               |
| Jeff Musk        | San Francisco Shock  |
+------------------+----------------------+
rows in set (0.041 sec)
```

4) Right

```
MariaDB [e_sports_397]> SELECT team_397.Team_Name,revenue_397.Source_name FROM revenue_397 RIGHT JOIN team_397 ON revenue_397.Team_ID = team_397.Team_ID;
+----------------------+-----------------------------------------+
| Team_Name            | Source_name                             |
+----------------------+-----------------------------------------+
| Paper ReX            | T-shirt January Sales                   |
| GodLike              | Ticket Sales                            |
| Fnatic               | Merchandise sales                       |
| Blood Eagle          | NULL                                    |
| Team Liquid          | Revenue from Advertisements             |
| G2 Esports           | NULL                                    |
| Team Vitality        | NULL                                    |
| Nigma Galaxy         | NULL                                    |
| Tundra Esports       | NULL                                    |
| Optic Gaming         | Stream Income                           |
| Optic Gaming         | Ticket sales for the challengers games. |
| Sentinels            | Twitch Stream Income.                   |
| San Francisco Shock  | NULL                                    |
| Dallas Fuel          | NULL                                    |
| NRG                  | NULL                                    |
| TSM                  | NULL                                    |
| Elite Esports        | Ad Revenue                              |
| Fusion Esports       | NULL                                    |
| Ghost Gaming         | NULL                                    |
| Cloud9               | NULL                                    |
| Evil Geniuses        | NULL                                    |
| 100 Thieves          | NULL                                    |
| Counter Logic Gaming | Public Stream Income                    |
| Immortals            | NULL                                    |
| ROX gaming           | Team jersey sales                       |
| Echo Fox             | NULL                                    |
+----------------------+-----------------------------------------+
26 rows in set (0.000 sec)

MariaDB [e_sports_397]>
```

5) Multiple Nested Joins

```
MariaDB [E_sports_397]> SELECT player_397.Player_Name,Team_397.Team_Name,Competition_Name FROM participates_397 NATURAL JOIN competition_397 JOIN team_397 ON team_397.Team_ID = participates_397.Team_ID JOIN play
er_397 ON player_397.Team_ID = team_397.Team_ID;
+------------------+----------------------+--------------------+
| Player_Name      | Team_Name            | Competition_Name   |
+------------------+----------------------+--------------------+
| Lionel Aguero    | Counter Logic Gaming | Dota conquerors    |
| Daniel Webber    | Elite Esports        | Fortnite Conquerors |
| Nicholas Stroll  | GodLike              | Apex League        |
| Carlos Alonso    | TSM                  | Apex League        |
| Thomas Reus      | Nigma Galaxy         | VCT Champions      |
| Rohit Kumar      | Sentinels            | VCT Masters        |
| Rohit Kumar      | Sentinels            | VCT Champions      |
| Kim Byoung In    | Echo Fox             | Fortnite Conquerors |
| Sergio Rodriguez | Cloud9               | Grand Masters      |
| Jeff Musk        | San Francisco Shock  | Fortnite Conquerors |
| Jeff Musk        | San Francisco Shock  | Apex League        |
| Jeff Musk        | San Francisco Shock  | VCT Champions      |
| Jeff Musk        | San Francisco Shock  | Overwatch League   |
+------------------+----------------------+--------------------+
13 rows in set (0.001 sec)
```

Aggregate Functions

1) Count
Obtain the count of the sources of revenue which generate an income greater than 100000

```
SELECT COUNT(*) FROM revenue_397 WHERE revenue_397.Income>100000;
```

2) Average
Obtain the average income of various teams from different incomes.

```
SELECT AVG(revenue_397.Income) FROM revenue_397;
```

3) Minimum
Find the lowest amount of salary earned by a player.

```
SELECT MIN(player_397.Salary) FROM player_397;
```

4) Maximum
Find the highest amount of salary earned by a player.

```
SELECT MAX(competition_397.Prize_pool) FROM competition_397;
```

5) Sum
Find the sum of salaries earned by all players in the database.

```
SELECT SUM(player_397.Salary) FROM player_397;
```

Output:

```
MariaDB [e_sports_397]> SELECT COUNT(*) FROM revenue_397 WHERE revenue_397.Income>100000;
+----------+
| COUNT(*) |
+----------+
|        6 |
+----------+
1 row in set (0.107 sec)

MariaDB [e_sports_397]> SELECT AVG(revenue_397.Income) FROM revenue_397;
+-------------------------+
| AVG(revenue_397.Income) |
+-------------------------+
|            229500.0000 |
+-------------------------+
1 row in set (0.028 sec)

MariaDB [e_sports_397]> SELECT MIN(player_397.Salary) FROM player_397;
+------------------------+
| MIN(player_397.Salary) |
+------------------------+
|                  50000 |
+------------------------+
1 row in set (0.047 sec)

MariaDB [e_sports_397]> SELECT MAX(competition_397.Prize_pool) FROM competition_397;
+---------------------------------+
| MAX(competition_397.Prize_pool) |
+---------------------------------+
|                        15000000 |
+---------------------------------+
1 row in set (0.000 sec)

MariaDB [e_sports_397]> SELECT SUM(player_397.Salary) FROM player_397;
+------------------------+
| SUM(player_397.Salary) |
+------------------------+
|                1760000 |
+------------------------+
1 row in set (0.001 sec)

MariaDB [e_sports_397]>
```

Set Operations

1) Union
Get Team ID of teams that are recorded in the teams table or the teams recorded in the revenue table.

SELECT revenue_397.Team_ID FROM revenue_397 UNION SELECT team_397.Team_ID FROM team_397;

2) Intersection

Get Team ID of teams which have a revenue recorded in the revenue_397 table.

SELECT revenue_397.Team_ID FROM revenue_397 INTERSECT SELECT team_397.Team_ID FROM team_397;

3) Set difference

Get Team ID of teams which have not signed a player.

SELECT team_397.Team_ID FROM team_397 EXCEPT SELECT player_397.Team_ID FROM player_397;

4) Cross Join

Get all the possible team and player combinations.

SELECT player_397.player_name,team_397.team_name FROM player_397 CROSS JOIN team_397;

Output:

Union and Intersection:

```
MariaDB [e_sports_397]> SELECT revenue_397.Team_ID FROM revenue_397 UNION SELECT team_397.Team_ID FROM team_397;
+----------+
| Team_ID  |
+----------+
| APAC_001 |
| APAC_401 |
| EMEA_001 |
| EMEA_501 |
| NA_001   |
| NA_002   |
| NA_301   |
| NA_602   |
| NA_901   |
| NA_101   |
| NA_102   |
| NA_302   |
| EMEA_401 |
| NA_401   |
| NA_902   |
| EMEA_701 |
| EMEA_702 |
| NA_201   |
| NA_202   |
| NA_601   |
| NA_501   |
| NA_502   |
| EMEA_801 |
| EMEA_802 |
| NA_801   |
+----------+
25 rows in set (0.001 sec)

MariaDB [e_sports_397]> SELECT revenue_397.Team_ID FROM revenue_397 INTERSECT SELECT team_397.Team_ID FROM team_397;
+----------+
| Team_ID  |
+----------+
| APAC_001 |
| APAC_401 |
| EMEA_001 |
| EMEA_501 |
| NA_001   |
| NA_002   |
| NA_301   |
| NA_602   |
| NA_901   |
+----------+
9 rows in set (0.000 sec)
```

Cross product and Set Difference:

```
MariaDB [e_sports_397]> SELECT team_397.Team_ID FROM team_397 EXCEPT SELECT player_397.Team_ID FROM player_397;
+----------+
| Team_ID  |
+----------+
| NA_102   |
| NA_302   |
| EMEA_401 |
| NA_401   |
| NA_901   |
| EMEA_701 |
| EMEA_702 |
| NA_201   |
| APAC_001 |
| EMEA_001 |
| NA_001   |
| NA_601   |
| EMEA_501 |
| NA_502   |
| EMEA_802 |
| NA_801   |
+----------+
16 rows in set (0.001 sec)

MariaDB [e_sports_397]> SELECT player_397.player_name,team_397.team_name FROM player_397 CROSS JOIN team_397;
+-----------------+---------------------+
| player_name     | team_name           |
+-----------------+---------------------+
| Lionel Aguero   | Paper ReX           |
| Daniel Webber   | Paper ReX           |
| Nicholas Stroll | Paper ReX           |
| Carlos Alonso   | Paper ReX           |
| Thomas Reus     | Paper ReX           |
| Rohit Kumar     | Paper ReX           |
| Kim Byoung In   | Paper ReX           |
| Sergio Rodriguez | Paper ReX           |
| Jeff Musk       | Paper ReX           |
| Lionel Aguero   | GodLike             |
| Daniel Webber   | GodLike             |
| Nicholas Stroll | GodLike             |
| Carlos Alonso   | GodLike             |
| Thomas Reus     | GodLike             |
| Rohit Kumar     | GodLike             |
| Kim Byoung In   | GodLike             |
| Sergio Rodriguez | GodLike             |
```

Nicholas Stroll	G2 Esports
Carlos Alonso	G2 Esports
Thomas Reus	G2 Esports
Rohit Kumar	G2 Esports
Kim Byoung In	G2 Esports
Sergio Rodriguez	G2 Esports
Jeff Musk	G2 Esports
Lionel Aguero	Team Vitality
Daniel Webber	Team Vitality
Nicholas Stroll	Team Vitality
Carlos Alonso	Team Vitality
Thomas Reus	Team Vitality
Rohit Kumar	Team Vitality
Kim Byoung In	Team Vitality
Sergio Rodriguez	Team Vitality
Jeff Musk	Team Vitality
Lionel Aguero	Nigma Galaxy
Daniel Webber	Nigma Galaxy
Nicholas Stroll	Nigma Galaxy
Carlos Alonso	Nigma Galaxy
Thomas Reus	Nigma Galaxy
Rohit Kumar	Nigma Galaxy
Kim Byoung In	Nigma Galaxy
Sergio Rodriguez	Nigma Galaxy
Jeff Musk	Nigma Galaxy
Lionel Aguero	Tundra Esports
Daniel Webber	Tundra Esports
Nicholas Stroll	Tundra Esports
Carlos Alonso	Tundra Esports
Thomas Reus	Tundra Esports
Rohit Kumar	Tundra Esports
Kim Byoung In	Tundra Esports
Sergio Rodriguez	Tundra Esports
Jeff Musk	Tundra Esports
Lionel Aguero	Optic Gaming
Daniel Webber	Optic Gaming
Nicholas Stroll	Optic Gaming
Carlos Alonso	Optic Gaming
Thomas Reus	Optic Gaming
Rohit Kumar	Optic Gaming
Kim Byoung In	Optic Gaming
Sergio Rodriguez	Optic Gaming
Jeff Musk	Optic Gaming
Lionel Aguero	Sentinels
Daniel Webber	Sentinels
Nicholas Stroll	Sentinels
Carlos Alonso	Sentinels
Thomas Reus	Sentinels
Rohit Kumar	Sentinels

Functions and Procedures

1) Functions

Function to display the percentage of payable tax for a particular income in the revenue_397 table.

```
DELIMITER $$
CREATE FUNCTION tax_397(Income INT)
RETURNS INT
DETERMINISTIC
BEGIN
        DECLARE tax INT;
        IF  Income <= 10000 THEN
                SET tax = 0.1*Income;
        ELSEIF Income > 10000 AND income <= 50000 THEN
                SET tax = 0.15*Income;
        ELSE
                SET tax = 0.25*Income;
        END IF;
        RETURN tax;
END; $$
DELIMITER ;
```

Output:

```
MariaDB [E_sports_397]> SELECT tax_397(Income) FROM revenue_397;
+-----------------+
| tax_397(Income) |
+-----------------+
|           75000 |
|           45000 |
|          125000 |
|           20000 |
|           17500 |
|           42500 |
|           19500 |
|          188000 |
|           30000 |
|            6750 |
+-----------------+
10 rows in set (0.001 sec)
```

2) Procedure

Procedure to update the age of players in the player_397 table.

Before calling Procedure:

```
MariaDB [E_sports_397]> ALTER TABLE player_397 ADD age int;
Query OK, 0 rows affected (0.583 sec)
Records: 0  Duplicates: 0  Warnings: 0

MariaDB [E_sports_397]> select * from player_397;
+-----------+------------------+-----------+------------+--------+----------+------+
| Player_ID | Player_Name      | Position  | DOB        | Salary | Team_ID  | age  |
+-----------+------------------+-----------+------------+--------+----------+------+
| ARG_001   | Lionel Aguero    | Attack    | 1996-04-07 | 170000 | NA_602   | NULL |
| AUS_001   | Daniel Webber    | Aggressor | 2004-06-06 |  90000 | NA_301   | NULL |
| CAN_001   | Nicholas Stroll  | Passive   | 2000-01-12 | 110000 | APAC_401 | NULL |
| ESP_001   | Carlos Alonso    | Shield    | 1999-01-20 | 150000 | NA_202   | NULL |
| GER_001   | Thomas Reus      | Defense   | 1998-08-09 | 190000 | EMEA_801 | NULL |
| IND_001   | Rohit Kumar      | Duelist   | 1997-12-09 | 450000 | NA_002   | NULL |
| KOR_001   | Kim Byoung In    | Aggressor | 2001-05-17 |  50000 | NA_902   | NULL |
| MEX_001   | Sergio Rodriguez | Smoker    | 1993-08-02 | 250000 | NA_501   | NULL |
| USA_001   | Jeff Musk        | Support   | 1996-04-25 | 300000 | NA_101   | NULL |
+-----------+------------------+-----------+------------+--------+----------+------+
9 rows in set (0.000 sec)

MariaDB [E_sports_397]> DELIMITER $$
MariaDB [E_sports_397]>
MariaDB [E_sports_397]> CREATE procedure age()
    ->
    -> BEGIN
    ->
    -> UPDATE player_397 SET Age = DATE_FORMAT(FROM_DAYS(DATEDIFF(NOW(),player_397.dob)), '%Y') + 0;
    ->
    -> END; $$
Query OK, 0 rows affected (0.371 sec)

MariaDB [E_sports_397]>
MariaDB [E_sports_397]> DELIMITER ;
MariaDB [E_sports_397]> call age();
Query OK, 9 rows affected (0.139 sec)
```

After calling Procedure:

```
MariaDB [E_sports_397]> call age();
Query OK, 9 rows affected (0.139 sec)

MariaDB [E_sports_397]> SELECT * FROM player_397;
+-----------+------------------+-----------+------------+--------+----------+------+
| Player_ID | Player_Name      | Position  | DOB        | Salary | Team_ID  | age  |
+-----------+------------------+-----------+------------+--------+----------+------+
| ARG_001   | Lionel Aguero    | Attack    | 1996-04-07 | 170000 | NA_602   |   26 |
| AUS_001   | Daniel Webber    | Aggressor | 2004-06-06 |  90000 | NA_301   |   18 |
| CAN_001   | Nicholas Stroll  | Passive   | 2000-01-12 | 110000 | APAC_401 |   22 |
| ESP_001   | Carlos Alonso    | Shield    | 1999-01-20 | 150000 | NA_202   |   23 |
| GER_001   | Thomas Reus      | Defense   | 1998-08-09 | 190000 | EMEA_801 |   24 |
| IND_001   | Rohit Kumar      | Duelist   | 1997-12-09 | 450000 | NA_002   |   24 |
| KOR_001   | Kim Byoung In    | Aggressor | 2001-05-17 |  50000 | NA_902   |   21 |
| MEX_001   | Sergio Rodriguez | Smoker    | 1993-08-02 | 250000 | NA_501   |   29 |
| USA_001   | Jeff Musk        | Support   | 1996-04-25 | 300000 | NA_101   |   26 |
+-----------+------------------+-----------+------------+--------+----------+------+
9 rows in set (0.000 sec)
```

Triggers and Cursors

1) Trigger

Trigger to check if the new player being added is above the age of 18. An error message informing his ineligibility to play a game as a professional is raised if s/he is below the age of 18.

Code:

```
DELIMITER $$
CREATE TRIGGER age_checks_397
BEFORE INSERT
ON player_397 FOR EACH ROW
BEGIN
DECLARE error_msg VARCHAR(50);
DECLARE age INT;
SET error_msg = "Not eligible to play as a professional.";
SELECT DATE_FORMAT(FROM_DAYS(DATEDIFF(NOW(),new.DOB)), '%Y')
 + 0 INTO age;

IF age < 18 THEN
SIGNAL SQLSTATE '45000'
SET MESSAGE_TEXT = error_msg;
END IF;
END $$
DELIMITER ;
```

Output:

```
MariaDB [E_sports_397]> DELIMITER $$
MariaDB [E_sports_397]> CREATE TRIGGER age_checks_397
    -> BEFORE INSERT
    -> ON player_397 FOR EACH ROW
    -> BEGIN
    -> DECLARE error_msg VARCHAR(50);
    -> DECLARE age INT;
    -> SET error_msg = "Not eligible to play as a professional.";
    -> SELECT DATE_FORMAT(FROM_DAYS(DATEDIFF(NOW(),new.DOB)), '%Y')
    ->  + 0 INTO age;
    ->
    -> IF age < 18 THEN
    -> SIGNAL SQLSTATE '45000'
    -> SET MESSAGE_TEXT = error_msg;
    -> END IF;
    -> END $$
Query OK, 0 rows affected (0.352 sec)

MariaDB [E_sports_397]> DELIMITER ;
MariaDB [E_sports_397]> INSERT INTO player_397 VALUES ('XYZ_001','Shashank','Defuser','2010-01-28',60000,'NA_101');
ERROR 1644 (45000): Not eligible to play as a professional.
MariaDB [E_sports_397]>
```

2) Cursor

Cursor to find the players who play for a given team.

```
DELIMITER $$

CREATE FUNCTION Player_Finder ( In_Team_ID VARCHAR(50) )
RETURNS VARCHAR(50) READS SQL DATA

BEGIN

  DECLARE done INT DEFAULT FALSE;
  DECLARE PlayerName VARCHAR(50);

  DECLARE cursor_player CURSOR FOR
    SELECT Player_Name
    FROM player_397
    WHERE Team_ID = In_Team_ID;

  DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = TRUE;

  OPEN cursor_player;
  FETCH cursor_player INTO PlayerName;

  CLOSE cursor_player;
```

RETURN PlayerName;

END; $$

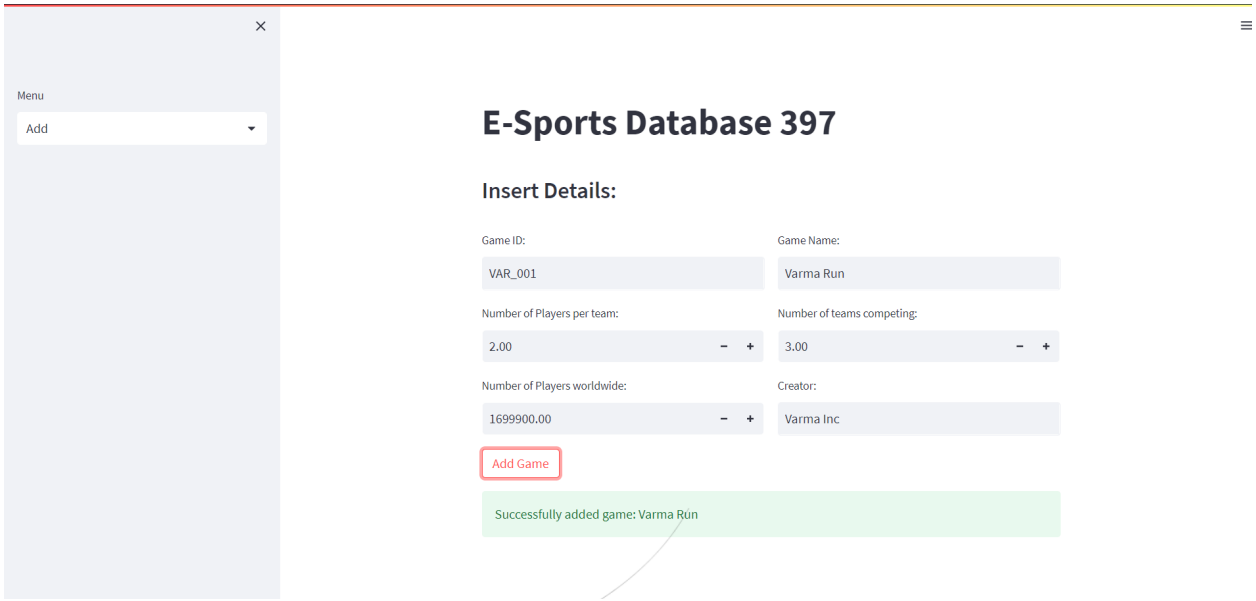
DELIMITER ;

Output:

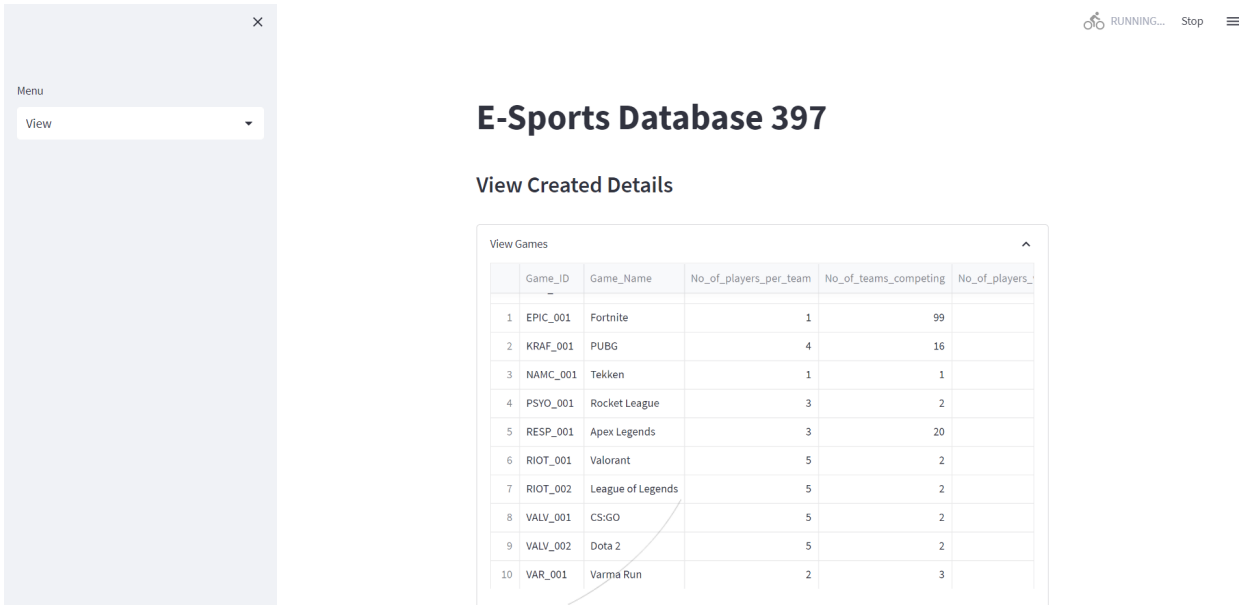
```
MariaDB [E_sports_397]> CREATE FUNCTION Player_Finder ( In_Team_ID VARCHAR(50) )
    -> RETURNS VARCHAR(50) READS SQL DATA
    ->
    -> BEGIN
    ->
    ->    DECLARE done INT DEFAULT FALSE;
    ->    DECLARE PlayerName VARCHAR(50);
    ->
    ->    DECLARE cursor_player CURSOR FOR
    ->      SELECT Player_Name
    ->      FROM player_397
    ->      WHERE Team_ID = In_Team_ID;
    ->
    ->    DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = TRUE;
    ->
    ->    OPEN cursor_player;
    ->    FETCH cursor_player INTO PlayerName;
    ->
    ->    CLOSE cursor_player;
    ->
    ->    RETURN PlayerName;
    ->
    -> END; $$
Query OK, 0 rows affected (0.162 sec)

MariaDB [E_sports_397]> DELIMITER ;
MariaDB [E_sports_397]> SELECT Player_Finder('NA_101');
+-------------------------+
| Player_Finder('NA_101') |
+-------------------------+
| Jeff Musk               |
+-------------------------+
1 row in set (0.157 sec)
```

Developing a Frontend

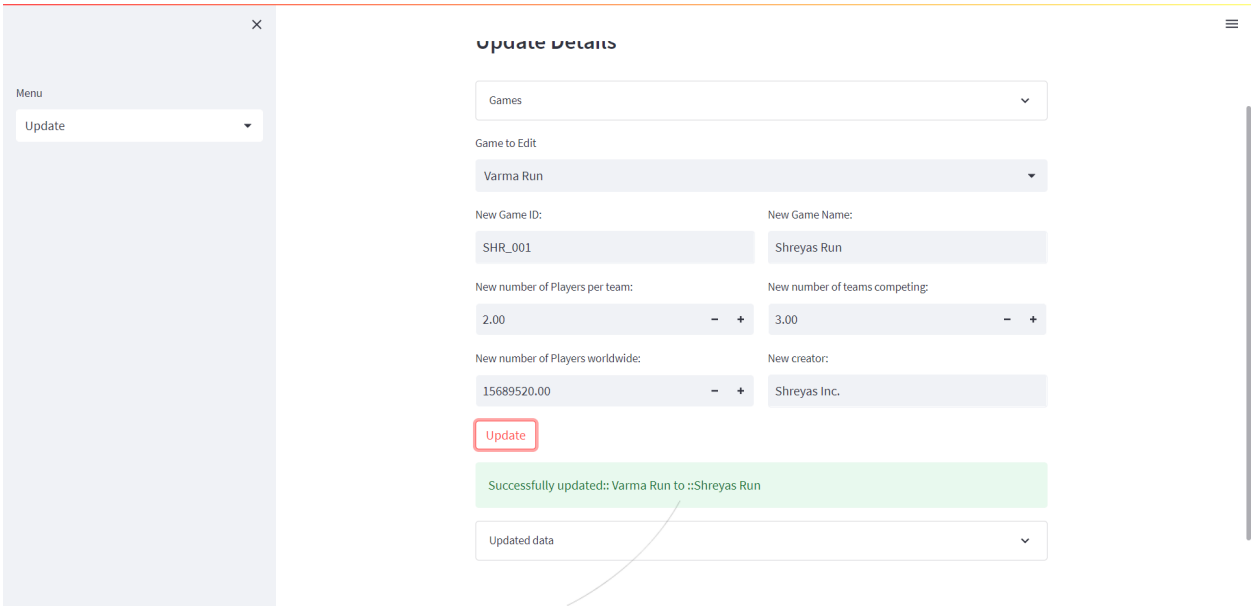
1) Insert



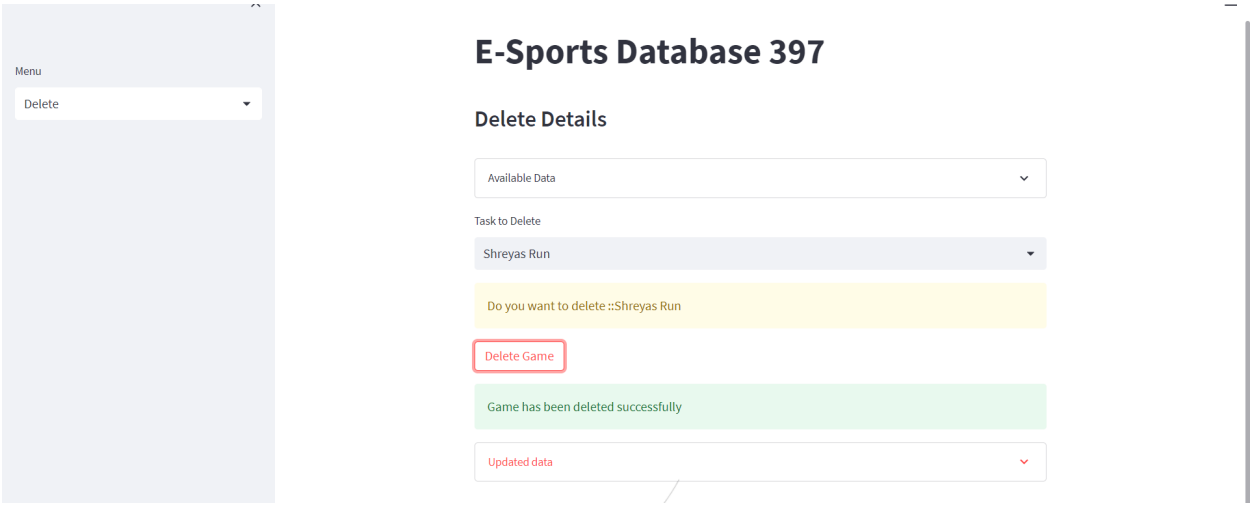
2) View



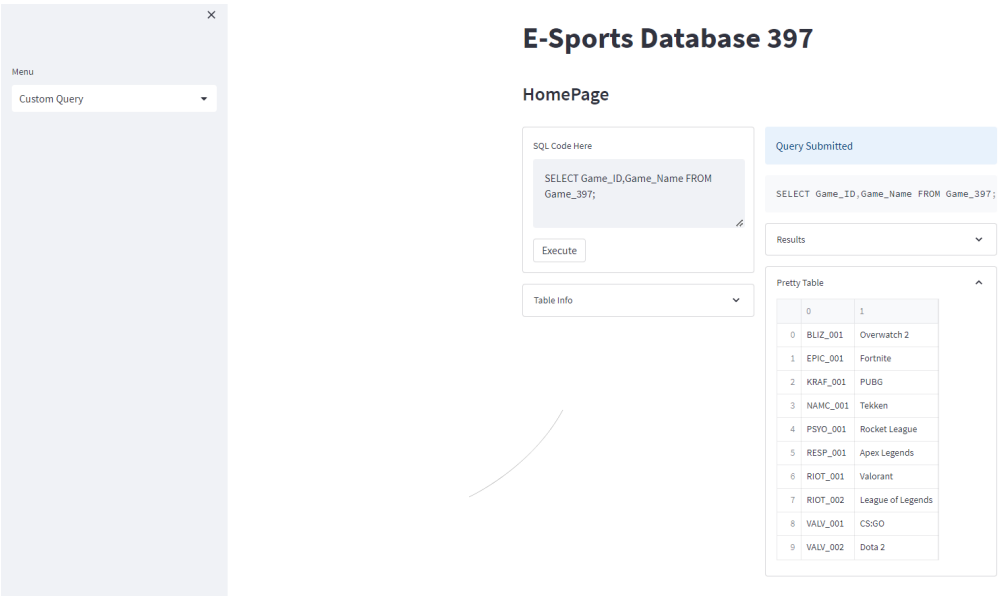
3) Update



4) Delete



5) Custom Query



6) Pie Chart

Pie chart showing the distribution of players worldwide across various games.