GAMER

by Arpine Harutyunyan

edX username: Arpinehar

GitHub username: arpharutyunyan

Gyumri, Shirak, Armenia

16 FEB 2024

The purpose of our database is to facilitate the management and organization of gaming-related data for a gaming platform. It serves as a central repository for storing information about players, games, teams, and the results of games played by teams.

The database includes the following entities:

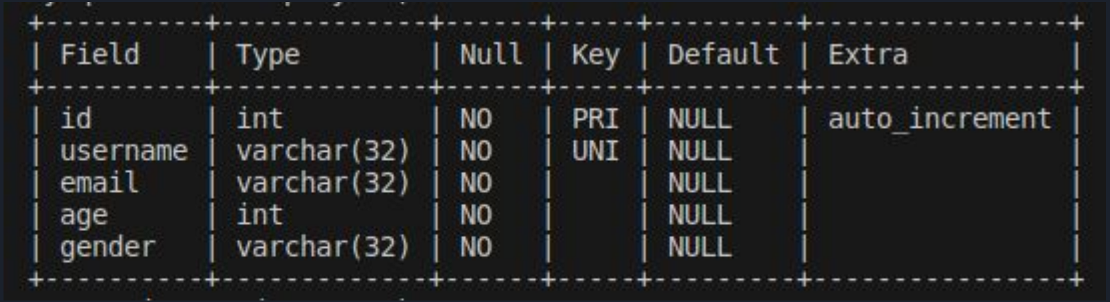
Entities are captured in SQLite tables with the following schema. But screens show with MySQL.

Players

The `players` table includes:

- `id`: Unique ID for the player, represented as an `INTEGER`. This column serves as the `PRIMARY KEY`
- `username`: The username chosen by the player, stored as `TEXT` and `UNIQUE`.
- `email`: Email address of the player, stored as `TEXT`.
- `age`: Age of the player, stored as `INTEGER`.
- `gender`: Gender of the player, stored as `TEXT`.

```sql
CREATE TABLE "players" (
    "id" INTEGER,
    "username" TEXT NOT NULL UNIQUE,
    "email" TEXT NOT NULL,
    "age" INTEGER NOT NULL,
    "gender" TEXT NOT NULL,
    PRIMARY KEY("id")
);
```



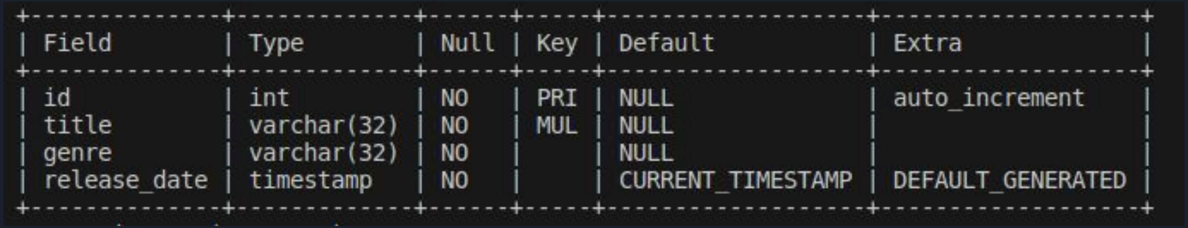
```
+----------+--------------+------+-----+---------+----------------+
| Field    | Type         | Null | Key | Default | Extra          |
+----------+--------------+------+-----+---------+----------------+
| id       | int          | NO   | PRI | NULL    | auto_increment |
| username | varchar(32)  | NO   | UNI | NULL    |                |
| email    | varchar(32)  | NO   |     | NULL    |                |
| age      | int          | NO   |     | NULL    |                |
| gender   | varchar(32)  | NO   |     | NULL    |                |
+----------+--------------+------+-----+---------+----------------+
```

Games

The `games` table includes:

- `id`: Unique ID for the game, represented as an `INTEGER`. This column serves as the `PRIMARY KEY`.
- `title`: The title of the game, stored as `TEXT` and `UNIQUE`.
- `genre`: Genre of the game, stored as `TEXT`.
- `release_date`: The release date of the game, stored as `DATE`.

```
CREATE TABLE "games" (
    "id" INTEGER,
    "title" TEXT NOT NULL UNIQUE,
    "genre" TEXT NOT NULL,
    "release_date" NUMERIC NOT NULL DEFAULT CURRENT_TIMESTAMP,
    PRIMARY KEY("id")
);
```



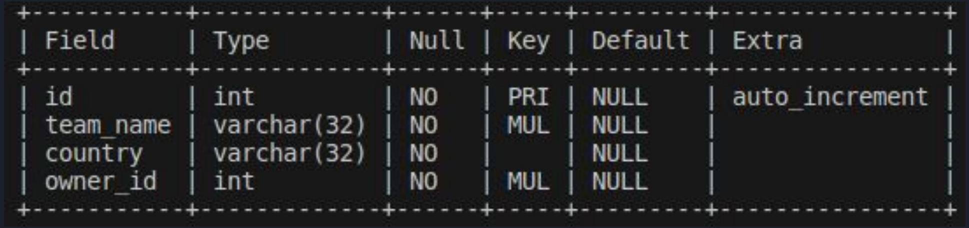
```
+--------------+-------------+------+-----+-------------------+-------------------+
| Field        | Type        | Null | Key | Default           | Extra             |
+--------------+-------------+------+-----+-------------------+-------------------+
| id           | int         | NO   | PRI | NULL              | auto_increment    |
| title        | varchar(32) | NO   | MUL | NULL              |                   |
| genre        | varchar(32) | NO   |     | NULL              |                   |
| release_date | timestamp   | NO   |     | CURRENT_TIMESTAMP | DEFAULT_GENERATED |
+--------------+-------------+------+-----+-------------------+-------------------+
```

Teams

The `teams` table includes:

- `id`: Unique ID for the team, represented as an `INTEGER`. This column serves as the `PRIMARY KEY`.
- `team_name`: The name of the team, stored as `TEXT` and `UNIQUE`.
- `country`: The country of the team, stored as `TEXT`.
- `owner_id`: ID of the owner, stored as `INTEGER`.

```sql
CREATE TABLE "teams" (
    "id" INTEGER,
    "team_name" TEXT NOT NULL UNIQUE,
    "country" TEXT NOT NULL,
    "owner_id" INTEGER,
    PRIMARY KEY("id"),
    FOREIGN KEY("owner_id") REFERENCES "players"("id")
);
```



```
+-----------+-------------+------+-----+---------+----------------+
| Field     | Type        | Null | Key | Default | Extra          |
+-----------+-------------+------+-----+---------+----------------+
| id        | int         | NO   | PRI | NULL    | auto_increment |
| team_name | varchar(32) | NO   | MUL | NULL    |                |
| country   | varchar(32) | NO   |     | NULL    |                |
| owner_id  | int         | NO   | MUL | NULL    |                |
+-----------+-------------+------+-----+---------+----------------+
```

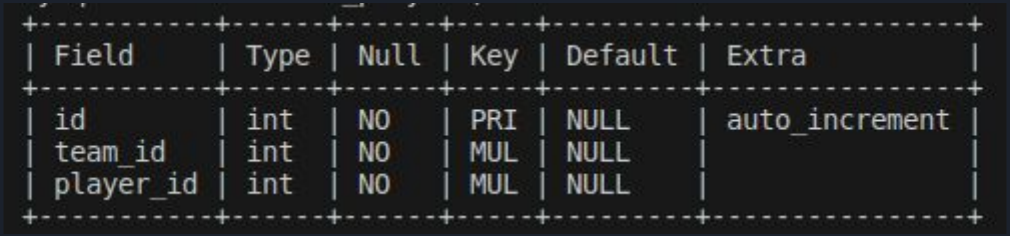
`owner_id` is stored as `INTEGER` and serves as a `FOREIGN KEY` referencing the `id` in the `players` table, ensuring referential integrity

Team Players

The `` `team_players` `` table includes:

- `` `id` ``: Unique ID for the team-player association, represented as an `` `INTEGER` ``. This column serves as the `` `PRIMARY KEY` ``.
- `` `team_id` ``: ID of the team, stored as `` `INTEGER` ``.
- `` `player_id` ``: ID of the player, stored as `` `INTEGER` ``.

```
CREATE TABLE "team_players" (
    "id" INTEGER,
    "team_id" INTEGER,
    "player_id" INTEGER,
    PRIMARY KEY("id"),
    FOREIGN KEY("team_id") REFERENCES "teams"("id"),
    FOREIGN KEY("player_id") REFERENCES "players"("id")
);
```



```
+-----------+------+------+-----+---------+----------------+
| Field     | Type | Null | Key | Default | Extra          |
+-----------+------+------+-----+---------+----------------+
| id        | int  | NO   | PRI | NULL    | auto_increment |
| team_id   | int  | NO   | MUL | NULL    |                |
| player_id | int  | NO   | MUL | NULL    |                |
+-----------+------+------+-----+---------+----------------+
```

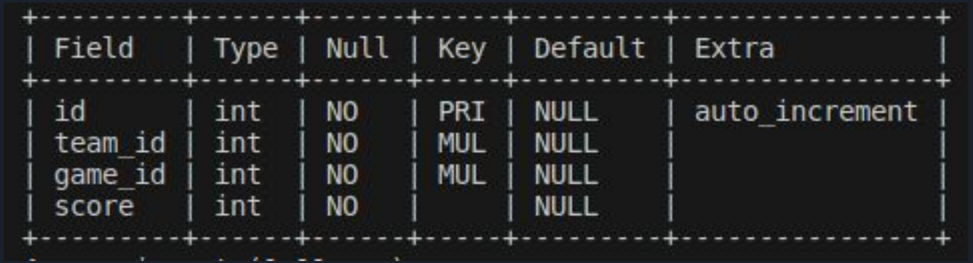
`team_id` and `player_id` are stored as `INTEGER` and serve as `FOREIGN KEY` referencing the `id` columns in the `teams` and `players` tables, respectively

Played Team Games

The `played_team_games` table includes:

- `id`: Unique ID for the played game, represented as an `INTEGER`. This column serves as the `PRIMARY KEY`.
- `team_id`: ID of the team, stored as `INTEGER`.
- `game_id`: ID of the game, stored as `INTEGER`.
- `score`: The score achieved by the team in the game, stored as `INTEGER`.

```
CREATE TABLE "played_team_games" (
    "id" INTEGER,
    "team_id" INTEGER,
    "game_id" INTEGER,
    "score" INTEGER DEFAULT 0,
    PRIMARY KEY("id"),
    FOREIGN KEY("team_id") REFERENCES "teams"("id"),
    FOREIGN KEY("game_id") REFERENCES "games"("id")
);
```



```
+---------+------+------+-----+---------+----------------+
| Field   | Type | Null | Key | Default | Extra          |
+---------+------+------+-----+---------+----------------+
| id      | int  | NO   | PRI | NULL    | auto_increment |
| team_id | int  | NO   | MUL | NULL    |                |
| game_id | int  | NO   | MUL | NULL    |                |
| score   | int  | NO   |     | NULL    |                |
+---------+------+------+-----+---------+----------------+
```

`team_id` and `game_id` are stored as `INTEGER` and serve as `FOREIGN KEY` referencing the `id` in the `teams` table and the `id` in the `games` table, respectively.

Optimizations

- Indexing the team_name column facilitates faster retrieval of teams based on their names. This is particularly useful when searching for specific teams by name or when filtering teams in queries involving team-related operations.

```
CREATE INDEX "team_name_index" ON "teams" ("team_name");
```

- Indexing the title column enables rapid retrieval of games based on their titles. This is particularly beneficial for queries where users search for specific games by their titles or when filtering games based on their names.

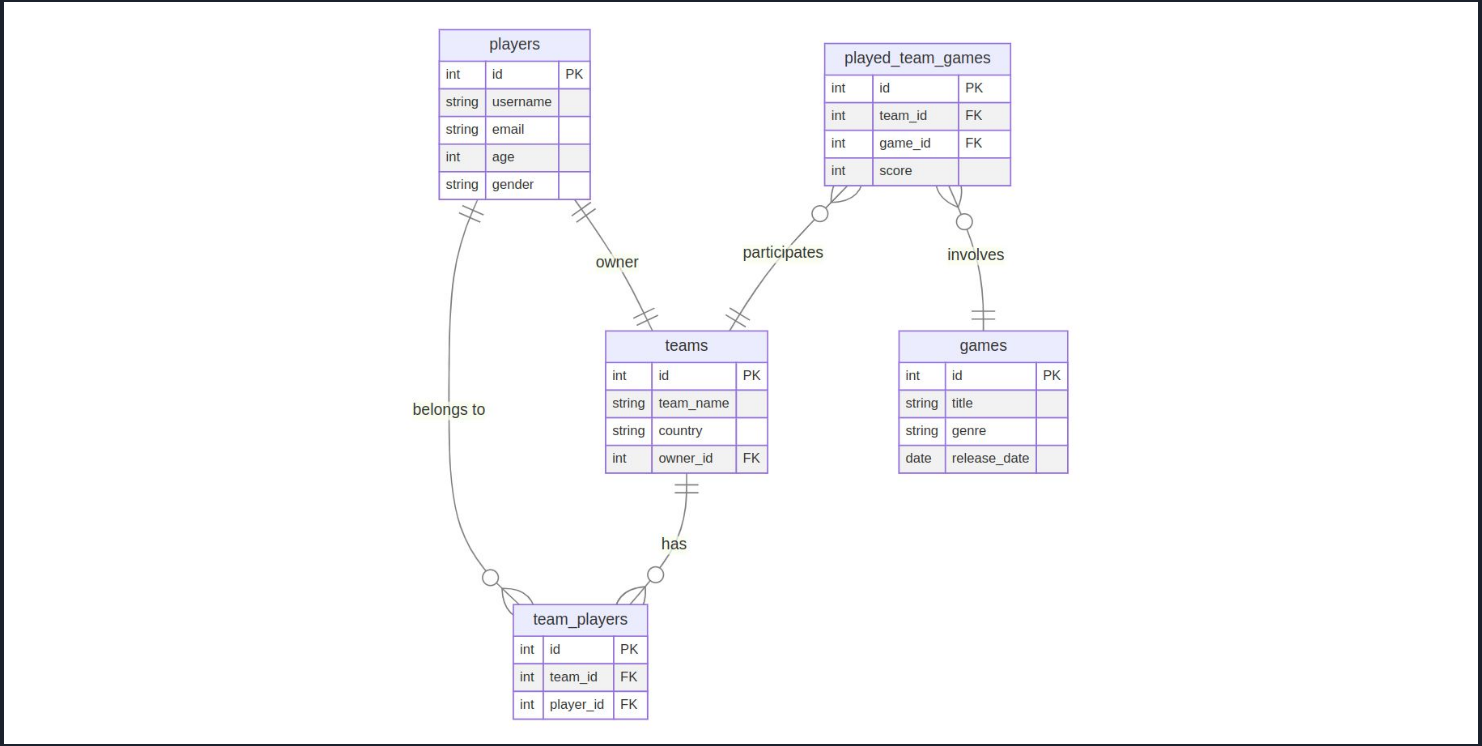
```
CREATE INDEX "game_title_index" ON "games" ("title");
```

- We create a view named team_scores_view that aggregates scores for each team across all games played. This view simplifies querying for total scores of teams, eliminating the need to manually calculate scores by summing individual game scores. By pre-computing and storing aggregated scores, the view improves query performance and reduces the complexity of score-related queries.

```
CREATE VIEW "team scores view" AS
SELECT "team id", SUM("score") AS "total_score"
FROM "played team games"
GROUP BY "team_id";
```

Relationships

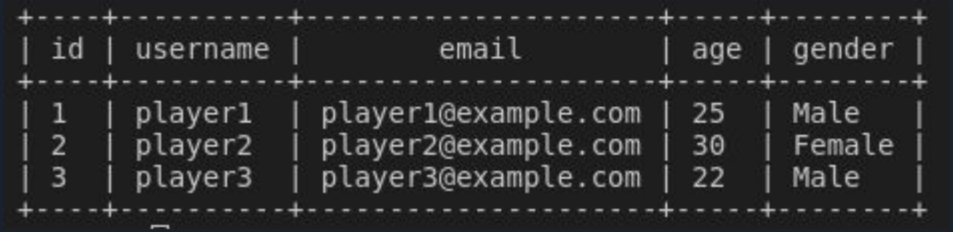
The below entity relationship diagram describes the relationships among the entities in the database.



As described by the diagram:

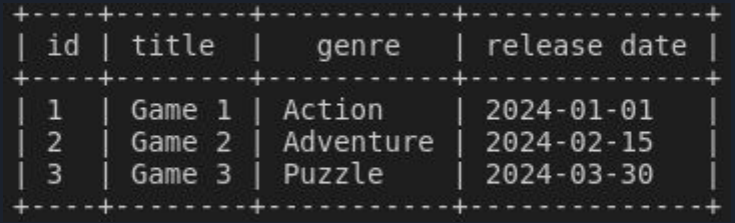
- One-to-One Relationship Between Players and Teams: Each player can own exactly one team, and each team is owned by exactly one player. This relationship is represented by the owner_id foreign key in the teams table, which references the id primary key in the players table.
- Many-to-Many Relationship Between Players and Teams: Each player can belong to 0 or many teams, and each team can have 0 or many players. This relationship is represented by the team_players table, where multiple players can be associated with multiple teams.
- Many-to-Many Relationship Between Teams and Games: Multiple teams can participate in 0 or many games, and each game can involve 0 or many teams. This relationship is captured by the played_team_games table, where multiple teams can play multiple games.

```sql
INSERT INTO "players" ("username", "email", "age", "gender") VALUES
('player1', 'player1@example.com' , 25, 'Male'),
('player2', 'player2@example.com' , 30, 'Female'),
('player3', 'player3@example.com' , 22, 'Male');
```



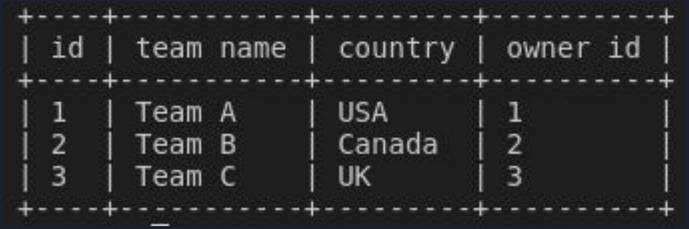
```
+----+----------+---------------------+-----+--------+
| id | username |        email        | age | gender |
+----+----------+---------------------+-----+--------+
| 1  | player1  | player1@example.com | 25  | Male   |
| 2  | player2  | player2@example.com | 30  | Female |
| 3  | player3  | player3@example.com | 22  | Male   |
+----+----------+---------------------+-----+--------+
```

```sql
INSERT INTO "games" ("title", "genre", "release_date") VALUES
('Game 1', 'Action', '2024-01-01'),
('Game 2', 'Adventure', '2024-02-15'),
('Game 3', 'Puzzle', '2024-03-30');
```

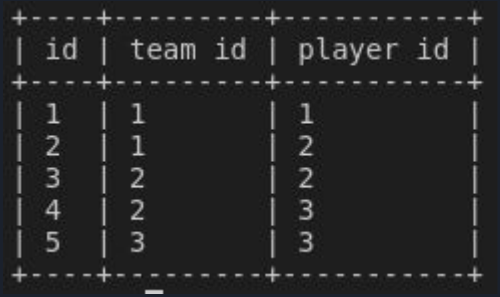


```
+----+--------+-----------+--------------+
| id | title  |   genre   | release date |
+----+--------+-----------+--------------+
| 1  | Game 1 | Action    | 2024-01-01   |
| 2  | Game 2 | Adventure | 2024-02-15   |
| 3  | Game 3 | Puzzle    | 2024-03-30   |
+----+--------+-----------+--------------+
```

```sql
INSERT INTO "teams" ("team_name", "country", "owner_id") VALUES
('Team A', 'USA', 1),
('Team B', 'Canada', 2),
('Team C', 'UK', 3);
```

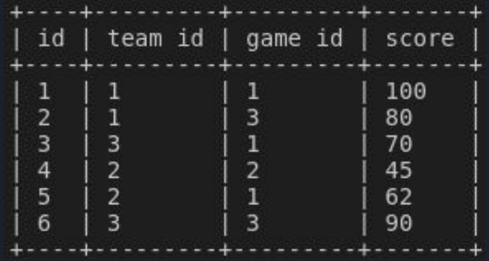


```
+----+-----------+---------+----------+
| id | team name | country | owner id |
+----+-----------+---------+----------+
| 1  | Team A    | USA     | 1        |
| 2  | Team B    | Canada  | 2        |
| 3  | Team C    | UK      | 3        |
+----+-----------+---------+----------+
INSERT INTO "team_players" ("team_id", "player_id") VALUES
(1, 1),
(1, 2),
(2, 2),
(2, 3),
(3, 3);
```



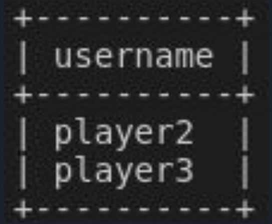
```
+----+---------+-----------+
| id | team id | player id |
+----+---------+-----------+
| 1  | 1       | 1         |
| 2  | 1       | 2         |
| 3  | 2       | 2         |
| 4  | 2       | 3         |
| 5  | 3       | 3         |
+----+---------+-----------+
```

```
INSERT INTO "played_team_games" ("team_id", "game_id", "score") VALUES
(1, 1, 100),
(1, 3, 80),
(3, 1, 70),
(2, 2, 45),
(2, 1, 62),
(3, 3, 90);
```



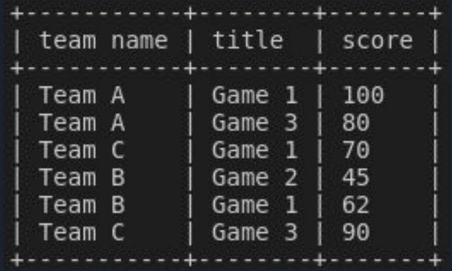
```
+----+---------+---------+-------+
| id | team id | game id | score |
+----+---------+---------+-------+
| 1  | 1       | 1       | 100   |
| 2  | 1       | 3       | 80    |
| 3  | 3       | 1       | 70    |
| 4  | 2       | 2       | 45    |
| 5  | 2       | 1       | 62    |
| 6  | 3       | 3       | 90    |
+----+---------+---------+-------+
```

```sql
-- Find the usernames of players who are members of teams from a specific country:
SELECT "username" FROM "players"
WHERE "id" IN (
   SELECT "player_id" FROM "team_players"
   WHERE "team_id" IN (
       SELECT "id" FROM "teams"
       WHERE "country"='Canada'
   )
);
```



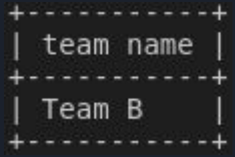
```
+----------+
| username |
+----------+
| player2  |
| player3  |
+----------+
```

```sql
-- Retrieve the details of games played by teams along with the team names:
SELECT "teams"."team_name", "games"."title", "played_team_games"."score" FROM
"played_team_games"
JOIN "teams" ON "teams"."id"="played_team_games"."team_id"
JOIN "games" ON "games"."id"="played_team_games"."game_id";
```



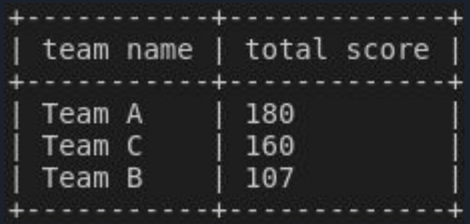
```
+-----------+--------+-------+
| team name | title  | score |
+-----------+--------+-------+
| Team A    | Game 1 | 100   |
| Team A    | Game 3 | 80    |
| Team C    | Game 1 | 70    |
| Team B    | Game 2 | 45    |
| Team B    | Game 1 | 62    |
| Team C    | Game 3 | 90    |
+-----------+--------+-------+
```

```sql
-- Find the team name of teams who played the game 'Game 2'
SELECT "team_name" FROM "teams"
WHERE "id" IN (
   SELECT "game_id" FROM "played_team_games"
   WHERE "game_id" IN (
       SELECT "id" FROM "games"
       WHERE "title"='Game 2'
   )
);
```



```
+-----------+
| team name |
+-----------+
| Team B    |
+-----------+
```

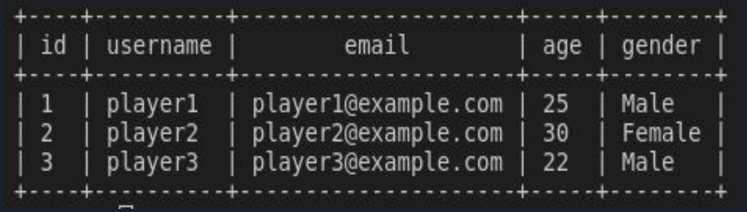
```sql
-- Retrieve the top 3 teams based on their total scores using team_scores_view VIEW
SELECT "teams"."team_name", "total_score" FROM "team_scores_view"
JOIN "teams" ON "teams"."id"="team_scores_view"."team_id"
ORDER BY "total_score" DESC
LIMIT 3;
```



```
+-----------+-------------+
| team name | total score |
+-----------+-------------+
| Team A    | 180         |
| Team C    | 160         |
| Team B    | 107         |
+-----------+-------------+
```

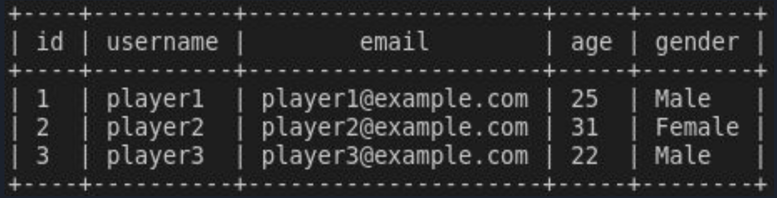
```sql
-- Update the age of player with email 'player2...' to 31
UPDATE "players" SET "age"='31'
WHERE "email" LIKE 'player2%';
```

OLD



```
+----+----------+--------------------+-----+--------+
| id | username |       email        | age | gender |
+----+----------+--------------------+-----+--------+
| 1  | player1  | player1@example.com | 25  | Male   |
| 2  | player2  | player2@example.com | 30  | Female |
| 3  | player3  | player3@example.com | 22  | Male   |
+----+----------+--------------------+-----+--------+
```

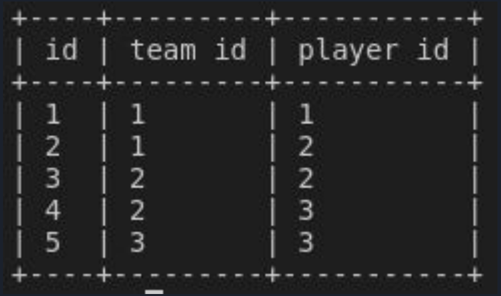
Updated



```
+----+----------+--------------------+-----+--------+
| id | username |       email        | age | gender |
+----+----------+--------------------+-----+--------+
| 1  | player1  | player1@example.com | 25  | Male   |
| 2  | player2  | player2@example.com | 31  | Female |
| 3  | player3  | player3@example.com | 22  | Male   |
+----+----------+--------------------+-----+--------+
```

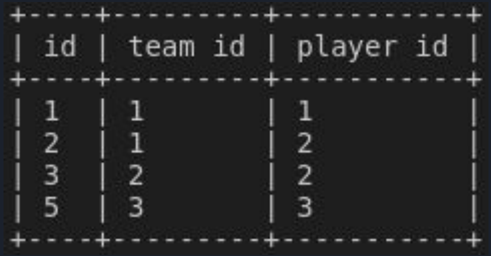
```sql
-- Delete player with username 'player3' from 'Team B' team
DELETE FROM "team_players"
WHERE "player_id"=(
    SELECT "id" FROM "players"
    WHERE "username"='player3'
)
AND "team_id"=(
    SELECT "id" FROM "teams"
    WHERE "team_name"='Team B'
);
```

Old



```
+----+---------+-----------+
| id | team id | player id |
+----+---------+-----------+
| 1  | 1       | 1         |
| 2  | 1       | 2         |
| 3  | 2       | 2         |
| 4  | 2       | 3         |
| 5  | 3       | 3         |
+----+---------+-----------+
```

Updated



```
+----+---------+-----------+
| id | team id | player id |
+----+---------+-----------+
| 1  | 1       | 1         |
| 2  | 1       | 2         |
| 3  | 2       | 2         |
| 5  | 3       | 3         |
+----+---------+-----------+
```