

CSE412-Phase 2

Group 12 - Yichen Li, Bhavya Patel, Naimish Jayesh Maniya, Juhil Ashok Sojitra

Dataset Link:

<https://www.kaggle.com/datasets/rovnez/fc-26-fifa-26-player-data/data>

1. ER-to-Relational Model Transformation

- Screenshots for DDL scripts:

Query History

```
1  |-- DROP EXISTING TABLES
2  DROP TABLE IF EXISTS player_position CASCADE;
3  DROP TABLE IF EXISTS player_country CASCADE;
4  DROP TABLE IF EXISTS player_club CASCADE;
5  DROP TABLE IF EXISTS RATINGS CASCADE;
6  DROP TABLE IF EXISTS ADDITIONAL_INFO CASCADE;
7  DROP TABLE IF EXISTS PLAYER CASCADE;
8  DROP TABLE IF EXISTS CLUB CASCADE;
9  DROP TABLE IF EXISTS LEAGUE CASCADE;
10 DROP TABLE IF EXISTS POSITION CASCADE;
11 DROP TABLE IF EXISTS COUNTRY CASCADE;
12
13 -- CREATE TABLES (Matching Phase 1 ER Diagram)
14 -- 1. COUNTRY Table
15 CREATE TABLE COUNTRY (
16     Nationality_id INTEGER PRIMARY KEY,
17     Nationality_name VARCHAR(100) NOT NULL
18 );
19
20 COMMENT ON TABLE COUNTRY IS 'Country represents a players nationality';
21
22 -- 2. POSITION Table
23 CREATE TABLE POSITION (
24     Position_Code VARCHAR(10) PRIMARY KEY,
25     Position_Name VARCHAR(50) NOT NULL
26 );
27
28 COMMENT ON TABLE POSITION IS 'Position represents player positions on the field (GK, CB, etc.)';
29
30 -- 3. LEAGUE Table
31 CREATE TABLE LEAGUE (
32     League_id INTEGER PRIMARY KEY,
33     League_name VARCHAR(150) NOT NULL,
34     League_level INTEGER,
35     Country_id INTEGER,
36     CONSTRAINT fk_league_country FOREIGN KEY (Country_id) REFERENCES COUNTRY(Nationality_id)
37     ON DELETE SET NULL
38 );
39
40 COMMENT ON TABLE LEAGUE IS 'League is an organization of clubs that arrange matches';
```

```

42 -- 4. CLUB Table
43 CREATE TABLE CLUB (
44     Club_team_id INTEGER PRIMARY KEY,
45     League_id INTEGER,
46     Club_name VARCHAR(150) NOT NULL,
47     CONSTRAINT fk_club_league FOREIGN KEY (League_id) REFERENCES LEAGUE(League_id)
48     ON DELETE SET NULL
49 );
50
51 COMMENT ON TABLE CLUB IS 'Each club belongs to a specific league';
52
53 -- 5. PLAYER Table (Main Entity)
54 CREATE TABLE PLAYER (
55     Player_id INTEGER PRIMARY KEY,
56     Short_name VARCHAR(100) NOT NULL,
57     Player_position VARCHAR(50),
58     Nationality_id INTEGER,
59     Club_team_id INTEGER,
60     Overall SMALLINT CHECK (Overall >= 0 AND Overall <= 100),
61     CONSTRAINT fk_player_nationality FOREIGN KEY (Nationality_id) REFERENCES COUNTRY(Nationality_id)
62     ON DELETE SET NULL,
63     CONSTRAINT fk_player_club_team FOREIGN KEY (Club_team_id) REFERENCES CLUB(Club_team_id)
64     ON DELETE SET NULL
65 );
66
67 COMMENT ON TABLE PLAYER IS 'Each player uniquely identified by Player_id';
68
69 -- 6. ADDITIONAL_INFO Table (Weak Entity)
70 CREATE TABLE ADDITIONAL_INFO (
71     Player_id INTEGER PRIMARY KEY,
72     Age SMALLINT CHECK (Age >= 15 AND Age <= 50),
73     DOB DATE,
74     Release_clause BIGINT,
75     Preferred_Position VARCHAR(30),
76     Height DECIMAL(5,2),
77     Weight DECIMAL(5,2),
78     Wages DECIMAL(12,2),
79     Weak_foot SMALLINT CHECK (Weak_foot >= 1 AND Weak_foot <= 5),
80     CONSTRAINT fk_additional_info_player FOREIGN KEY (Player_id) REFERENCES PLAYER(Player_id)
81     ON DELETE CASCADE
82 );

```

```

84 COMMENT ON TABLE ADDITIONAL_INFO IS 'Stores player details - each player can have at most one detailed record';
85
86 -- 7. RATINGS Table (Weak Entity)
87 CREATE TABLE RATINGS (
88     Player_id INTEGER PRIMARY KEY,
89     Passing SMALLINT CHECK (Passing >= 0 AND Passing <= 100),
90     Defending SMALLINT CHECK (Defending >= 0 AND Defending <= 100),
91     Dribbling SMALLINT CHECK (Dribbling >= 0 AND Dribbling <= 100),
92     Pace SMALLINT CHECK (Pace >= 0 AND Pace <= 100),
93     Shooting SMALLINT CHECK (Shooting >= 0 AND Shooting <= 100),
94     Physic SMALLINT CHECK (Physic >= 0 AND Physic <= 100),
95     CONSTRAINT fk_ratings_player FOREIGN KEY (Player_id) REFERENCES PLAYER(Player_id)
96     ON DELETE CASCADE
97 );
98
99 COMMENT ON TABLE RATINGS IS 'Ratings include FIFAs six core ability values';
100
101 -- 8. player_club Relationship Table
102 CREATE TABLE player_club (
103     Player_id INTEGER PRIMARY KEY,
104     Club_team_id INTEGER,
105     club_position VARCHAR(10),
106     jersey_number SMALLINT CHECK (jersey_number >= 1 AND jersey_number <= 99),
107     contract_until INTEGER,
108     joined_date DATE,
109     CONSTRAINT fk_player_club_player FOREIGN KEY (Player_id) REFERENCES PLAYER(Player_id)
110     ON DELETE CASCADE,
111     CONSTRAINT fk_player_club_club FOREIGN KEY (Club_team_id) REFERENCES CLUB(Club_team_id)
112     ON DELETE CASCADE
113 );
114
115 COMMENT ON TABLE player_club IS 'Many-to-one: Each player belongs to one club';
116
117 -- 9. player_country Relationship Table
118 CREATE TABLE player_country (
119     Player_id INTEGER PRIMARY KEY,
120     Nationality_id INTEGER,
121     nation_position VARCHAR(10),
122     nation_jersey SMALLINT CHECK (nation_jersey >= 1 AND nation_jersey <= 99),
123     CONSTRAINT fk_player_country_player FOREIGN KEY (Player_id) REFERENCES PLAYER(Player_id)
124     ON DELETE CASCADE,
125     CONSTRAINT fk_player_country_country FOREIGN KEY (Nationality_id) REFERENCES COUNTRY(Nationality_id)
126     ON DELETE CASCADE
127 );
128
129 COMMENT ON TABLE player_country IS 'Many-to-one: Each player represents one country';
130
131 -- 10. player_position Relationship Table (Many-to-Many)
132 CREATE TABLE player_position (
133     Player_id INTEGER,
134     Position_Code VARCHAR(10),
135     PRIMARY KEY (Player_id, Position_Code),
136     CONSTRAINT fk_player_position_player FOREIGN KEY (Player_id) REFERENCES PLAYER(Player_id)
137     ON DELETE CASCADE,
138     CONSTRAINT fk_player_position_position FOREIGN KEY (Position_Code) REFERENCES POSITION(Position_Code)
139     ON DELETE CASCADE
140 );
141
142 COMMENT ON TABLE player_position IS 'Many-to-many: A player can play multiple positions';
143
144 -- CREATE INDEXES FOR PERFORMANCE
145 CREATE INDEX idx_player_overall ON PLAYER(Overall DESC);
146 CREATE INDEX idx_player_nationality ON PLAYER(Nationality_id);
147 CREATE INDEX idx_player_club ON PLAYER(Club_team_id);
148 CREATE INDEX idx_player_name ON PLAYER(Short_name);
149 CREATE INDEX idx_club_league ON CLUB(League_id);
150 CREATE INDEX idx_league_country ON LEAGUE(Country_id);

```

- Screenshots for schema reviews

```
fifa26=# \d
      List of relations
 Schema |       Name        | Type  | Owner
-----+-----+-----+-----+
 public | additional_info | table | postgres
 public | club           | table | postgres
 public | country         | table | postgres
 public | league          | table | postgres
 public | player          | table | postgres
 public | player_club    | table | postgres
 public | player_country  | table | postgres
 public | player_position | table | postgres
 public | position         | table | postgres
 public | ratings          | table | postgres
(10 rows)
```

2. Data Population

- Screenshot for Copy csv to import data:

Query History

```

1 -- 1. COUNTRY
2 COPY COUNTRY (Nationality_id, Nationality_name)
3 FROM 'D:/CSE412-project/01_COUNTRY.csv' DELIMITER ',' CSV HEADER;
4
5 -- 2. POSITION
6 COPY POSITION (Position_Code, Position_Name)
7 FROM 'D:/CSE412-project/02_POSITION.csv' DELIMITER ',' CSV HEADER;
8
9 -- 3. LEAGUE
10 COPY LEAGUE (League_id, League_name, League_level, Country_id)
11 FROM 'D:/CSE412-project/03_LEAGUE.csv' DELIMITER ',' CSV HEADER;
12
13 -- 4. CLUB
14 COPY CLUB (Club_team_id, League_id, Club_name)
15 FROM 'D:/CSE412-project/04_CLUB.csv' DELIMITER ',' CSV HEADER;
16
17 -- 5. PLAYER
18 COPY PLAYER (Player_id, Short_name, Player_position, Nationality_id, Club_team_id, Overall)
19 FROM 'D:/CSE412-project/05_PLAYER.csv' DELIMITER ',' CSV HEADER;
20
21 -- 6. ADDITIONAL_INFO
22 COPY ADDITIONAL_INFO (Player_id, Age, DOB, Release_clause, Preferred_Position, Height, Weight, Wages, Weak_foot)
23 FROM 'D:/CSE412-project/06_ADDITIONAL_INFO.csv' DELIMITER ',' CSV HEADER;
24
25 -- 7. RATINGS
26 COPY RATINGS (Player_id, Passing, Defending, Dribbling, Pace, Shooting, Physic)
27 FROM 'D:/CSE412-project/07_RATINGS.csv' DELIMITER ',' CSV HEADER;
28
29 -- 8. player_club
30 COPY player_club (Player_id, Club_team_id, club_position, jersey_number, contract_until, joined_date)
31 FROM 'D:/CSE412-project/08_player_club.csv' DELIMITER ',' CSV HEADER;
32
33 -- 9. player_country
34 COPY player_country (Player_id, Nationality_id, nation_position, nation_jersey)
35 FROM 'D:/CSE412-project/09_player_country.csv' DELIMITER ',' CSV HEADER;
36
37 -- 10. player_position
38 COPY player_position (Player_id, Position_Code)
39 FROM 'D:/CSE412-project/10_player_position.csv' DELIMITER ',' CSV HEADER;
40
```

- Screenshots for table contents:

```

1  SELECT * FROM COUNTRY LIMIT 10;
2  SELECT * FROM POSITION LIMIT 10;
3  SELECT * FROM LEAGUE LIMIT 10;
4  SELECT * FROM CLUB LIMIT 10;
5  SELECT * FROM PLAYER LIMIT 10;
6  SELECT * FROM ADDITIONAL_INFO LIMIT 10;
7  SELECT * FROM RATINGS LIMIT 10;
8  SELECT * FROM PLAYER_CLUB LIMIT 10;
9  SELECT * FROM PLAYER_COUNTRY LIMIT 10;
10 SELECT * FROM PLAYER_POSITION LIMIT 10;
11

```

Sample 1: Country

| | nationality_id [PK] integer | nationality_name character varying (100) |
|----|--------------------------------|---|
| 1 | 14 | England |
| 2 | 60 | Uruguay |
| 3 | 21 | Germany |
| 4 | 129 | Morocco |
| 5 | 27 | Italy |
| 6 | 48 | Türkiye |
| 7 | 52 | Argentina |
| 8 | 45 | Spain |
| 9 | 18 | France |
| 10 | 38 | Portugal |

Sample 2 : Position

Query Query History

```
1  SELECT * FROM POSITION LIMIT 10;
2
```

Data Output Messages Notifications

| | position_code [PK] character varying (10) | position_name character varying (50) |
|----|--|---|
| 1 | GK | Goalkeeper |
| 2 | CB | Center Back |
| 3 | LB | Left Back |
| 4 | RB | Right Back |
| 5 | LWB | Left Wing Back |
| 6 | RWB | Right Wing Back |
| 7 | CDM | Central Defensive Midfield... |
| 8 | CM | Center Midfielder |
| 9 | CAM | Central Attacking Midfielder |
| 10 | LM | Left Midfielder |

Sample 3: League

Query Query History

```
1  SELECT * FROM LEAGUE LIMIT 10;
```

Data Output Messages Notifications

| | league_id [PK] integer | league_name character varying (150) | league_level integer | country_id integer |
|----|---------------------------|--|-------------------------|-----------------------|
| 1 | 53 | La Liga | 1 | 45 |
| 2 | 19 | Bundesliga | 1 | 21 |
| 3 | 16 | Ligue 1 | 1 | 18 |
| 4 | 31 | Serie A | 1 | 27 |
| 5 | 13 | Premier League | 1 | 14 |
| 6 | 350 | Pro League | 1 | 183 |
| 7 | 39 | Major League Soccer | 1 | 95 |
| 8 | 68 | Süper Lig | 1 | 48 |
| 9 | 317 | Hrvatska nogometna liga | 1 | 10 |
| 10 | 353 | Liga Profesional de Fútbol | 1 | 52 |

- Screenshots for sample insert commands:

```

Query Query History
1 INSERT INTO PLAYER (Player_id, Short_name, Player_position, Nationality_id, Club_team_id, Overall)
2 VALUES (99999, 'Test Player', 'ST', 1, 1, 99);
3
4 INSERT INTO CLUB (Club_team_id, League_id, Club_name)
5 VALUES (999, 1, 'Test United FC');
6

```

Sample 1 insert a player, Sample 2 insert a club.

Output 1:

| Data Output | | | | | | | Messages | Notifications |
|-------------|---------------------------|---------------------------------------|---|---------------------------|-------------------------|---------------------|----------|---------------|
| | player_id [PK] integer | short_name character varying (100) | player_position character varying (50) | nationality_id integer | club_team_id integer | overall smallint | | |
| 1 | 99999 | Test Player | ST | 1 | 1 | 99 | | |

Output 2:

| Data Output | | | | | | | Messages | Notifications |
|-------------|------------------------------|----------------------|--------------------------------------|--|--|--|----------|---------------|
| | club_team_id [PK] integer | league_id integer | club_name character varying (150) | | | | | |
| 1 | 999 | 1 | Test United FC | | | | | |

3. SQL Queries Covering Application Use Cases

a. Query 1: SELECT - Player Search by Rating and Club

Use Case: Users search for top players from a specific club

```
SELECT
    Short_name,
    Overall,
    Player_position
FROM PLAYER
ORDER BY Overall DESC
LIMIT 10;
```

The screenshot shows a SQL query editor interface. At the top, there are tabs for 'Query' (which is selected) and 'Query History'. Below the tabs is a code editor area containing the SQL query. The code is numbered from 1 to 8. The query itself is:

```
1  SELECT
2      Short_name,
3          Overall,
4              Player_position
5  FROM PLAYER
6  ORDER BY Overall DESC
7  LIMIT 10;
8
```

Below the code editor is a large empty space. At the bottom of the interface, there is a toolbar with icons for file operations (New, Open, Save, etc.) and a 'SQL' button. To the right of the toolbar, it says 'Showing rows: 1 to 10'. Below the toolbar is a table displaying the results of the query. The table has three columns: 'short_name', 'overall', and 'player_position'. The data is as follows:

| | short_name | overall | player_position |
|----|---------------|---------|-----------------|
| 1 | M. Salah | 91 | RM, RW |
| 2 | K. Mbappé | 91 | ST, LW, LM |
| 3 | J. Bellingham | 90 | CAM, CM |
| 4 | Rodri | 90 | CDM, CM |
| 5 | O. Dembélé | 90 | ST, RW, CAM |
| 6 | E. Haaland | 90 | ST |
| 7 | V. van Dijk | 90 | CB |
| 8 | F. Valverde | 89 | CM, CDM, RB |
| 9 | J. Kimmich | 89 | CDM, RB, CM |
| 10 | A. Hakimi | 89 | RB, RM |

b. Query 2: SELECT - Player Comparison with Ratings

Use Case: Users compare multiple players side-by-side with detailed ratings

```

SELECT
    Short_name,
    Overall,
    Player_position
FROM PLAYER
WHERE Player_position LIKE '%ST%'
ORDER BY Overall DESC
LIMIT 5;

```

The screenshot shows a SQL query editor interface. The top section is labeled "Query History" and contains the executed SQL code. The bottom section is labeled "Data Output" and displays the results of the query as a table.

| | short_name | overall | player_position |
|---|------------|---------|-----------------|
| 1 | K. Mbappé | 91 | ST, LW, LM |
| 2 | O. Dembélé | 90 | ST, RW, CAM |
| 3 | E. Haaland | 90 | ST |
| 4 | H. Kane | 89 | ST |
| 5 | F. Wirtz | 89 | CAM, ST, CM |

c. Query 3: INSERT - Add New Player

Use Case: Add a newly signed player to the database

```

-- Insert a new player (example: a young prospect)
INSERT INTO PLAYER (Player_id, Short_name, Player_position, Nationality_id,
Club_team_id, Overall)
VALUES (999998, 'J. Doe', 'CM', 1343, 243, 72);

```

-- Also insert their ratings

```

INSERT INTO RATINGS (Player_id, Passing, Defending, Dribbling, Pace, Shooting,
Physic)
VALUES (999998, 68, 65, 70, 72, 60, 69);

```

-- And additional info

```

INSERT INTO ADDITIONAL_INFO (Player_id, Age, DOB, Height, Weight, Wages,
Weak_foot)
VALUES (999998, 19, '2006-03-15', 178.0, 72.0, 5000.0, 3);

```

-- Search for him

```

SELECT * FROM PLAYER p
JOIN RATINGS r ON p.Player_id = r.Player_id
JOIN ADDITIONAL_INFO a ON p.Player_id = a.Player_id
WHERE p.Player_id = 999998;

```

The screenshot shows a SQL query history and a results grid. The query history contains the following code:

```

1 -- Insert a new player (example: a young prospect)
2 INSERT INTO PLAYER (Player_id, Short_name, Player_position, Nationality_id, Club_team_id, Overall)
3 VALUES (999998, 'J. Doe', 'CM', 45, 243, 72);
4
5 -- Also insert their ratings
6 INSERT INTO RATINGS (Player_id, Passing, Defending, Dribbling, Pace, Shooting, Physic)
7 VALUES (999998, 68, 65, 70, 72, 60, 69);
8
9 -- And additional info
10 INSERT INTO ADDITIONAL_INFO (Player_id, Age, DOB, Height, Weight, Wages, Weak_foot)
11 VALUES (999998, 19, '2006-03-15', 178.0, 72.0, 5000.0, 3);
12
13
14 SELECT * FROM PLAYER p
15 JOIN RATINGS r ON p.Player_id = r.Player_id
16 JOIN ADDITIONAL_INFO a ON p.Player_id = a.Player_id
17 WHERE p.Player_id = 999998;
18
19

```

The results grid shows one row of data corresponding to the inserted player:

| | player_id | short_name | player_position | nationality_id | club_team_id | overall | player_id | passing | defending | dribbling | pace | shooting | physic | player_id | age | dob |
|---|-----------|------------|-----------------|----------------|--------------|---------|-----------|---------|-----------|-----------|------|----------|--------|-----------|-----|--------|
| 1 | 999998 | J. Doe | CM | 45 | 243 | 72 | 999998 | 68 | 65 | 70 | 72 | 60 | 69 | 999998 | 19 | 2006-0 |

d. Query 4: UPDATE - Modify Player Rating

Use Case: Update player's rating after performance improvement

-- update their improved ratings

UPDATE RATINGS

SET Passing = 75,

Dribbling = 78,

Pace = 76

WHERE Player_id = 999999;

```
-- search for it
SELECT * FROM RATINGS WHERE Player_id = 999999;
```

The screenshot shows a database interface with two tabs: 'Query' and 'Query History'. The 'Query' tab contains the following SQL code:

```
1 -- update their improved ratings
2 UPDATE RATINGS
3 SET Passing = 75,
4     Dribbling = 78,
5     Pace = 76
6 WHERE Player_id = 999999;
7
8 SELECT * FROM RATINGS WHERE Player_id = 999999;
```

The 'Data Output' tab displays the results of the last query. The table has columns: player_id [PK] integer, passing smallint, defending smallint, dribbling smallint, pace smallint, shooting smallint, and physic smallint. There is one row with values: 999999, 75, 65, 78, 76, 60, and 69.

| | player_id [PK] integer | passing smallint | defending smallint | dribbling smallint | pace smallint | shooting smallint | physic smallint |
|---|---------------------------|---------------------|-----------------------|-----------------------|------------------|----------------------|--------------------|
| 1 | 999999 | 75 | 65 | 78 | 76 | 60 | 69 |

e. Query 5: DELETE - Remove Player

Use Case: Remove a retired player from the database

```
-- delete a player who retired
DELETE FROM PLAYER
WHERE Player_id = 999999;

-- search for it
SELECT * FROM RATINGS WHERE Player_id = 999999;
```

Query Query History

```
1 -- Delete a player who retired
2 DELETE FROM PLAYER
3 WHERE Player_id = 999999;
4
5 SELECT * FROM RATINGS WHERE Player_id = 999999;
```

Data Output Messages Notifications

| | player_id [PK] integer | passing smallint | defending smallint | dribbling smallint | pace smallint | shooting smallint | physic smallint |
|--|---------------------------|---------------------|-----------------------|-----------------------|------------------|----------------------|--------------------|
|--|---------------------------|---------------------|-----------------------|-----------------------|------------------|----------------------|--------------------|

4. Video

<https://youtu.be/40y-hUiPals>