**PROJECT - SPORT DATABASE MANAGEMENT SYSTEM**

**PROJECT OVERVIEW:** The objective is to create a database schema for a "English Premier League DB" which will be used to evaluate players' performances. Creation of this database helps in deep-dive analysis on matches played by clubs representing various countries and their team performances. Drawing database schema produces the relationships, breakpoints and anomalies across entities to extract data.

| Task | Objective | Action Items |
|---|---|---|
| 0 | To create repository in Oracle Environment. | Determining all the attributes and values for data dictionary and designing it on oracle. |
| 1 | To learn in depth about keys for identification. | Understand the tables, attributes and identify primary and foreign keys. |
| 2 | To create a graphical view of an EER diagram. | Design the EER diagram with correct relationships and cardinalities. |
| 3 | To convert, apply and implement the relational schema. | Design the database on oracle based on the schema developed |
| 4 | To understand normalization and dependencies for the data and execution of the same. | Ensure normalization is achieved and all the dependencies are addressed |
| 5(Queries and output) | To execute the queries and make inference. | Creating repository of EPL data through various tables and perform analysis through extraction of data using SQL. |
| Final review | To check the hierarchy of the process and evaluate the performance of the project. | Perform Quality Check assessments on each tasks and ensure required outcome is achieved. |

**Task 0: Create your complete repository**

**Repository –** It will have metadata of 7 tables where the column names, data type, length, min and max allowable values, and description will be added. The details for each table are listed below:

**players:** Includes the bio details of all players and the clubs they belong to.

| Name | Type | Length | Min | Max | Description |
|---|---|---|---|---|---|
| **player_id** | NUM | 5 | 1 | 5 | The unique id of the player |
| **age** | NUM | 2 | 2 | 2 | Player's age |
| **nationality** | VARCHAR | 4 | 3 | 4 | Player's nationality |
| **club** | VARCHAR | 20 | 1 | 20 | Name of player's club |

**team_stats:** Includes all the players a club owns and their status (healthy, injured).

| Name | Type | Length | Min | Max | Description |
|---|---|---|---|---|---|
| **player_id** | NUM | 5 | 1 | 5 | The unique id of the player |
| **club** | VARCHAR | 20 | 1 | 20 | Name of player's club |
| **status** | VARCHAR | 20 | 1 | 20 | Health status of the player |

**player_stats:** Includes players attributes like goals, assists etc.

| Name | Type | Length | Min | Max | Description |
|---|---|---|---|---|---|
| **player_id** | NUM | 5 | 1 | 5 | The unique id of the player |
| **position** | VARCHAR | 50 | 5 | 50 | Position at which player plays |
| **matches_played** | NUM | 5 | 0 | 5 | Number of matches played |
| **total_goals** | NUM | 4 | 0 | 4 | Number of goals scored |
| **assists** | NUM | 5 | 0 | 5 | Number of assists by the player |
| **passes_attempted** | NUM | 50 | 1 | 5 | Number of passes by the player |
| **perc_passes_attempted** | NUM | 50 | 1 | 5 | Percentage of passes attempted by the player |

**matches:** Includes the match and the results.

| Name | Type | Length | Min | Max | Description |
|---|---|---|---|---|---|
| **match_id** | NUM | 10 | 1 | 10 | The unique id of match |
| **home_team** | VARCHAR | 30 | 3 | 30 | Team playing at home ground |
| **away_team** | VARCHAR | 30 | 3 | 30 | Team playing not at home ground |
| **home_team_goals** | NUM | 4 | 1 | 4 | Goals scored by home team |
| **away_team_goals** | NUM | 4 | 1 | 4 | Goals scored by away team |
| **winner** | VARCHAR | 30 | 3 | 30 | Team which won the match |

**player_matches:** Includes a mapping between the player and the matches the player has played.

| Name | Type | Length | Min | Max | Description |
|---|---|---|---|---|---|
| **match_id** | NUM | 10 | 1 | 10 | The unique id of match |
| **player_id** | NUM | 5 | 1 | 5 | The unique id of the player |

**countries:** Maintains a mapping between countries and their short codes.

| Name | Type | Length | Min | Max | Description |
|---|---|---|---|---|---|
| **country_code** | VARCHAR | 3 | 3 | 3 | 3-digit unique code given to the country |
| **country_name** | VARCHAR | 30 | 1 | 30 | Name of the country |

**club_countries:** This table is used to refer the country a club belongs to.

| Name | Type | Length | Min | Max | Description |
|---|---|---|---|---|---|
| **club** | VARCHAR | 20 | 1 | 20 | Name of player's club |
| **country_code** | VARCHAR | 3 | 3 | 3 | 3-digit unique code given to the country |

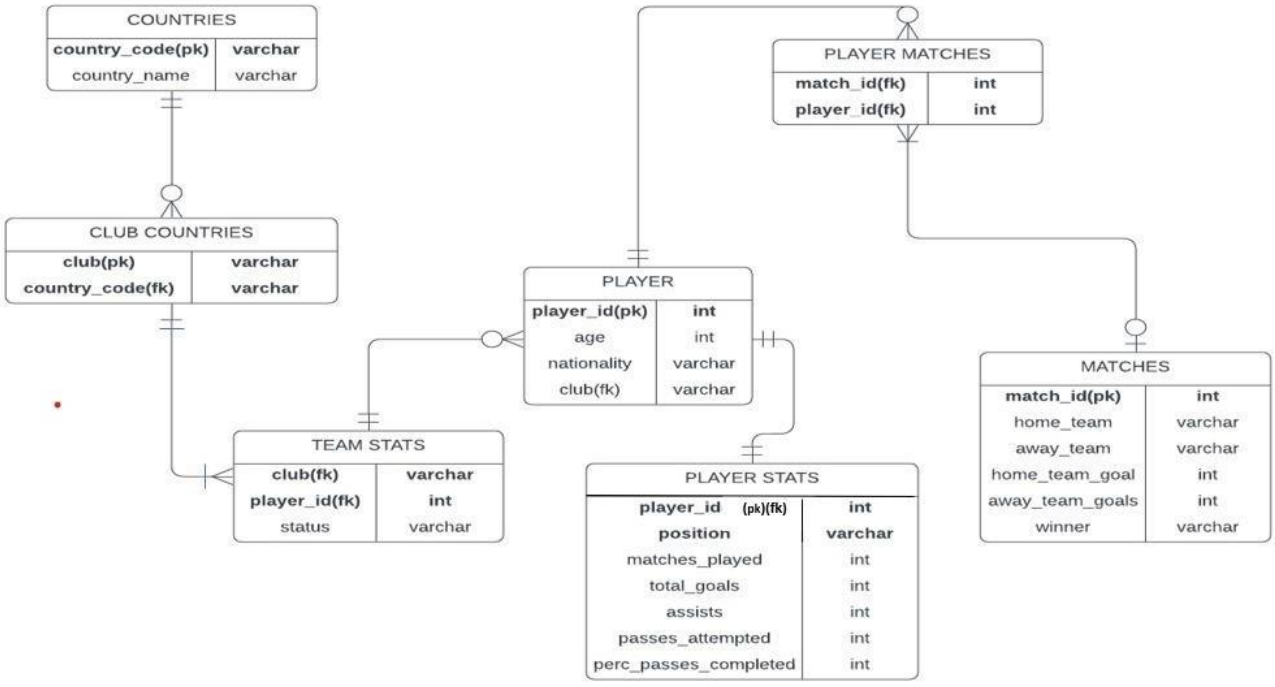
**Business Rules:**

- Player's age cannot be less than 15 and more than 40.
- The position must be among the following-
    - Goalkeeper
    - Right Full-back (or Wingback)
    - Left Full-back (or Wingback)
    - Center-back
    - Center back (or sweeper)
    - Defensive Midfielder
    - Right Midfielder (or Winger)
    - Center Midfielder
    - Center Forward (or Striker)
    - Attacking Midfielder (or Center Forward)
    - Left Midfielder (or Winger)
- *player_id, match_id,* and *country_code* must have unique values.
- Each *player_id* should be associated with only one *club*.
- Each *player_id* must have one *nationality* and every *nationality* must have one *player_id*.
- Each *match_id* should have at least two *player_id*'s participating in the match.
- The player's *status* should be "Active" by default to play.
- The *player_id* in the *player_matches* table should belong to either the *home team* or the *away team* in the *matches* table for the same *match_id*.

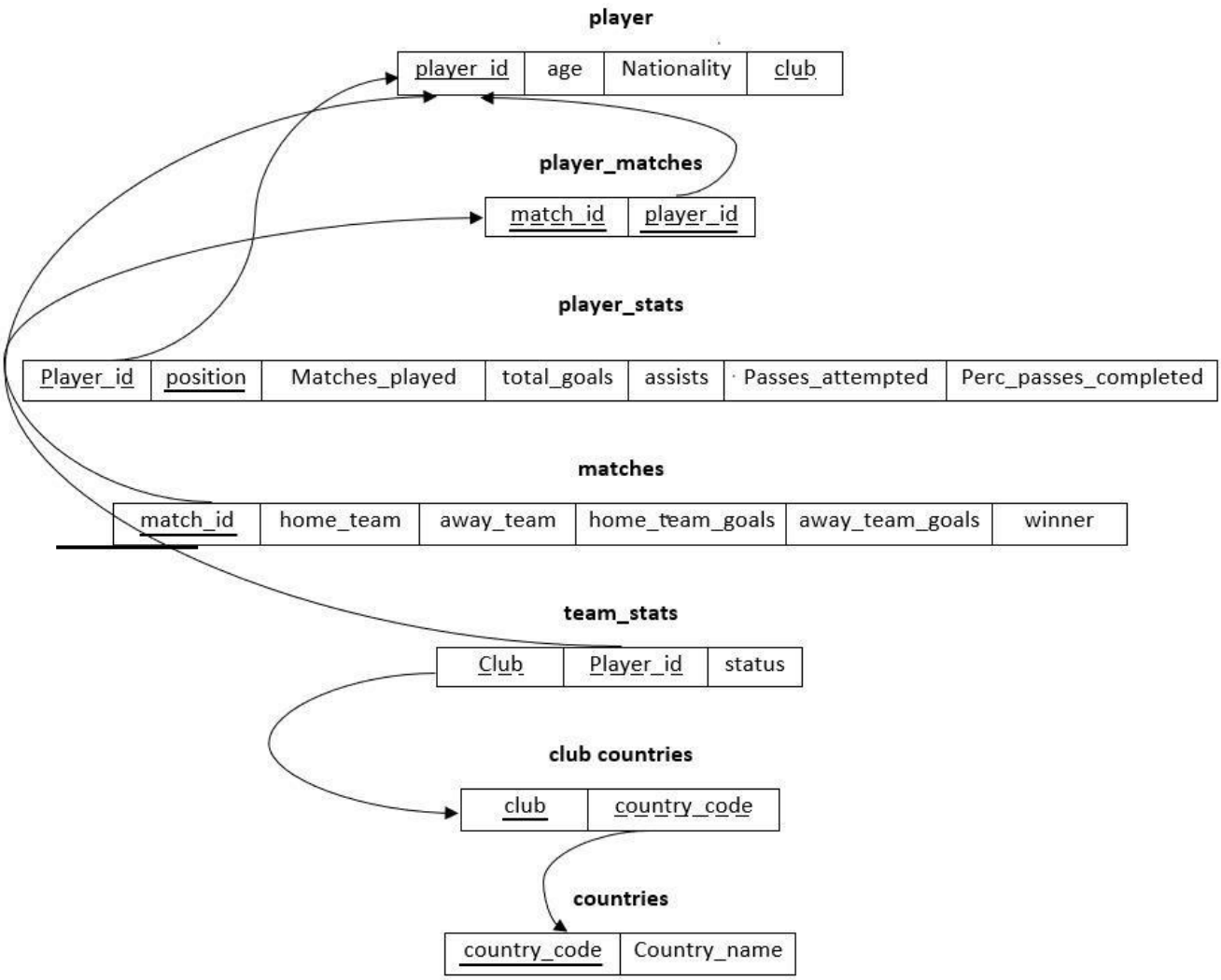
**Task 1: Identify best primary keys and foreign keys for all the seven relations. List out primary key, foreign key of every relation.**

| Table | Primary Key | Foreign Key |
|---|---|---|
| players | player_id | club |
| team_stats | player_id, club (composite) | club, player_id |
| player_stats | player_id | player_id |
| matches | match_id | - |
| player_matches | player_id, match_id (composite) | player_id, match_id |
| countries | country_code | - |
| club_countries | club | country_code |

**Task 2: Express the relationships in each description graphically with an EER diagram.**




**Task 3: Convert it to a relational schema. Explain briefly about 1) Unary 2) Binary and 3) Ternary relationships. List all the Binary relationships that exists in the above schema. Identify any constraints that may be applicable in EPL Database problem and implement them using database constraints.**




- Unary relationship: A relationship between instances of a single entity type. In the given dataset, there is no entity that is related to itself.
- Ternary relationship: A simultaneous relationship among the instances of three entity types.
- Binary relationship: A relationship between the instances of two entity types.

Based on the above definitions, we see that our database contains 6 binary relationships.

| **Binary relationships** | **Cardinality Constraint** |
|---|---|
| players - player_stats | Mandatory One to Mandatory One |
| team_stats - player | Mandatory One to Optional Many |
| players - player_matches | Mandatory One to Optional Many |
| matches - player_matches | Optional One to Mandatory Many |
| club countries -team_stats | Mandatory One to Mandatory Many |
| countries - club countries | Mandatory One to Optional Many |

**Domain Constraints: NOT NULL, CHECK**
**Key Constraints: PRIMARY KEY, FOREIGN KEY**
**DEFAULT, UNIQUE, INDEX Constraints** can also be used
**Entity Integrity Constraints -** No primary key attribute may be null. All primary key fields must contain data values. Below are the primary keys in the EPL database that should not be null.

| Table | Primary Key |
|---|---|
| players | player_id |
| team_stats | player_id, club (composite) |
| player_stats | player_id |
| matches | match_id |
| player_matches | player_id, match_id (composite) |
| countries | country_code |
| club_countries | club |

**Referential Integrity Constraint:** According to the Referential Integrity rule, any foreign key value must match a primary key value in the relation of the one side.

***Restrict*** - prevent deletion of "parent" side if related rows are present in "dependent" side.

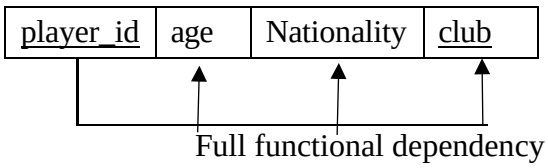
***Cascade*** - automatically remove "dependent" side rows that correspond with the "parent" side row to be deleted.

***Set-to-Null*** - set the foreign key in the dependent side to null if deletion from the parent side.

**<u>Task 4:</u> Provide a list of dependencies for each relation. Explain Normalization? Prove that the above schema is in 1NF, 2NF and 3NF?**

Normalization is the process to eliminate data redundancy and enhance data integrity in the tables. Normalization organizes the columns and tables of a database to ensure that database integrity constraints properly execute their dependencies.

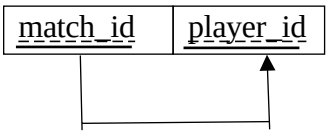
**players**




Player_id->age, nationality, club

A player's age, nationality and club are functionally dependent on that person's player_id. Or there can be only one age, one nationality and one club for a particular player_id.

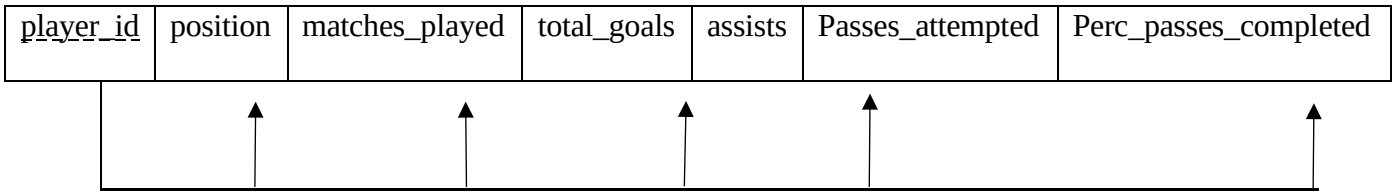
**player_matches**




Match_id->player_id

A player's id is functionally dependent on the match id to know about the player's matches.

**player_stats**




Full dependency

A player's position, matches_played, total_goals, assists, passes_attempted, perc_passes_completed are functionally dependent on the player_id.

In this relational schema, we can see a partial dependency between player_id and position columns. Hence, we normalized the data by creating a different table for the position details.

A new table called "player_position" has been created with the details of player_id and the positions.

All other tables are fully functionally dependent.

Rules for 1NF:

- ✓ It doesn't have 2 columns with same name.
- ✓ Order of rows is not important.
- ✓ It cannot have a multivalued attribute.
- ✓ Data stored in each column should be of same datatype.

Rules for 2NF:

- ✓ It should satisfy 1NF
- ✓ It should not have partial dependencies

Rule for 3NF:

- ✓ It should satisfy 1 NF and 2NF
- ✓ It should not have transitive dependencies

Finally, all the tables after normalization are found to be in 1NF, 2NF and 3NF forms.

## Task 5: Queries and Output

1. **SQL 'CREATE' Query for building every relation with Entity Integrity.**
   **Query**

```
--DROP TABLE PLAYERS;
CREATE TABLE PLAYERS
(
PLAYER_ID INT,
AGE INT,
NATIONALITY   VARCHAR2(100),
CLUB              VARCHAR2(100),
PRIMARY KEY (PLAYER_ID),
FOREIGN KEY (CLUB) REFERENCES CLUB_COUNTRIES (CLUB)
);
```

   **Output:**


   **Query**

```
--DROP TABLE TEAM_STATS;
```

```
CREATE TABLE TEAM_STATS
(
PLAYER_ID INT,
CLUB        VARCHAR2(100),
STATUS      VARCHAR2(100),
PRIMARY KEY (PLAYER_ID),
FOREIGN KEY (CLUB) REFERENCES CLUB_COUNTRIES (CLUB),
FOREIGN KEY (PLAYER_ID) REFERENCES PLAYERS (PLAYER_ID)
);
```

**Output**


**Query**

```
--DROP TABLE PLAYER_STATS;
CREATE TABLE PLAYER_STATS
(
PLAYER_ID INT,
POSITION VARCHAR2(100),
MATCHES_PLAYED      INT,
TEAM_GOALS INT,
ASSISTS                     INT,
PASSES_ATTEMPTED        INT,
PERC_PASSES_COMPLETED INT,
PRIMARY KEY (PLAYER_ID),
FOREIGN KEY (PLAYER_ID) REFERENCES PLAYERS (PLAYER_ID)
);
```

**Output**


**Query**

```
--DROP TABLE MATCHES;
CREATE TABLE MATCHES
(
MATCH_ID INT,
HOME_TEAM   VARCHAR2(100),
AWAY_TEAM   VARCHAR2(100),
HOME_TEAM_GOALS         INT,
AWAY_TEAM_GOALS         INT,
WINNER          VARCHAR2(100),
PRIMARY KEY (MATCH_ID)
);
```

**Output**


**Query**

```
--DROP TABLE PLAYER_MATCHES;
CREATE TABLE PLAYER_MATCHES
(
MATCH_ID    INT    NOT    NULL,
PLAYER_ID INT NOT NULL,
PRIMARY KEY (MATCH_ID,PLAYER_ID),
FOREIGN KEY (PLAYER_ID) REFERENCES PLAYERS1 (PLAYER_ID), FOREIGN
KEY (MATCH_ID) REFERENCES MATCHES1 (MATCH_ID)
);
```

**Output**


**Query**

```
--DROP TABLE COUNTRIES;
CREATE TABLE COUNTRIES
(
COUNTRY_CODE  VARCHAR2(100),
COUNTRY_NAME  VARCHAR2(100),
PRIMARY KEY (COUNTRY_CODE)
);
```

**Output**


**Query**

```
--DROP TABLE CLUB_COUNTRIES;
CREATE TABLE CLUB_COUNTRIES
(
CLUB                    VARCHAR2(100),
COUNTRY_CODE       VARCHAR2(100),
PRIMARY KEY (CLUB),
FOREIGN KEY (COUNTRY_CODE) REFERENCES COUNTRIES (COUNTRY_CODE)
);
```

**Output**


**2. SQL 'UPDATE' QUERY**

We found that in Countries for the country "Cotedlovire" was represented in lower case while other countries were presented in upper case. To update the country "Cotedlovire" to upper case we have used the below query.

**Data example before update query**

| COUNTRY_CODE | COUNTRY_NAME |
|---|---|
| ENG | ENGLAND |
| CIV | CotedIvoire |

2 rows returned in 0.00 seconds Download

**Query**

```
UPDATE COUNTRIES
SET  COUNTRY_NAME  =  UPPER(COUNTRY_NAME)
WHERE COUNTRY_CODE = 'CIV';
```

**Output after updation**

| COUNTRY_CODE | COUNTRY_NAME |
|---|---|
| CIV | COTEDIVOIRE |

1 rows returned in 0.00 seconds Download

3. **Find the age of the oldest player whose total_goals > 50.**

**Query**

```
SELECT MAX(AGE)
FROM
(
SELECT A.PLAYER_ID,A.AGE,SUM(B.TOTAL_GOALS)
FROM PLAYERS A
LEFT JOIN PLAYER_STATS B
ON  A.PLAYER_ID  =  B.PLAYER_ID
GROUP  BY  A.PLAYER_ID,  A.AGE
HAVING  SUM(B.TOTAL_GOALS)  >  50
ORDER BY A.PLAYER_ID, A.AGE DESC
);
```

**Output**

| MAX(AGE) |
| --- |
| - |

1 rows returned in 0.00 seconds Download

4. **Select the club that has made more goals.**

**Query**

```
SELECT A.CLUB, SUM(B.TOTAL_GOALS) AS CLUB_GOALS
FROM TEAM_STATS A
LEFT JOIN PLAYER_STATS B
ON A.PLAYER_ID = B.PLAYER_ID
GROUP BY A.CLUB
ORDER BY CLUB_GOALS DESC;
```

**Output**

| CLUB | CLUB_GOALS |
| --- | --- |
| Manchester City | 104 |
| Leicester City | 86 |
| West Ham United | 85 |

5. **Write a Query Using SUB-QUERY.**

We have sub-query to fetch player information such as Age and Matches played with respect to each player id under 'Liverpool FC' club.

**Query**

```
SELECT PLAYER_ID, AGE, MATCHES_PLAYED
FROM
(
SELECT B.*, A.AGE, A.NATIONALITY, A.CLUB
FROM PLAYERS A
LEFT JOIN PLAYER_STATS B
ON A.PLAYER_ID = B.PLAYER_ID
)
```

WHERE CLUB = 'Liverpool FC' ;

**Output**

| PLAYER_ID | AGE | MATCHES_PLAYED |
|---|---|---|
| 264 | 20 | 9 |
| 265 | 28 | 10 |
| 266 | 19 | 9 |
| 267 | 25 | 10 |
| 268 | 23 | 7 |
| 269 | 28 | 14 |
| 269 | 28 | 14 |
| 270 | 29 | 24 |