

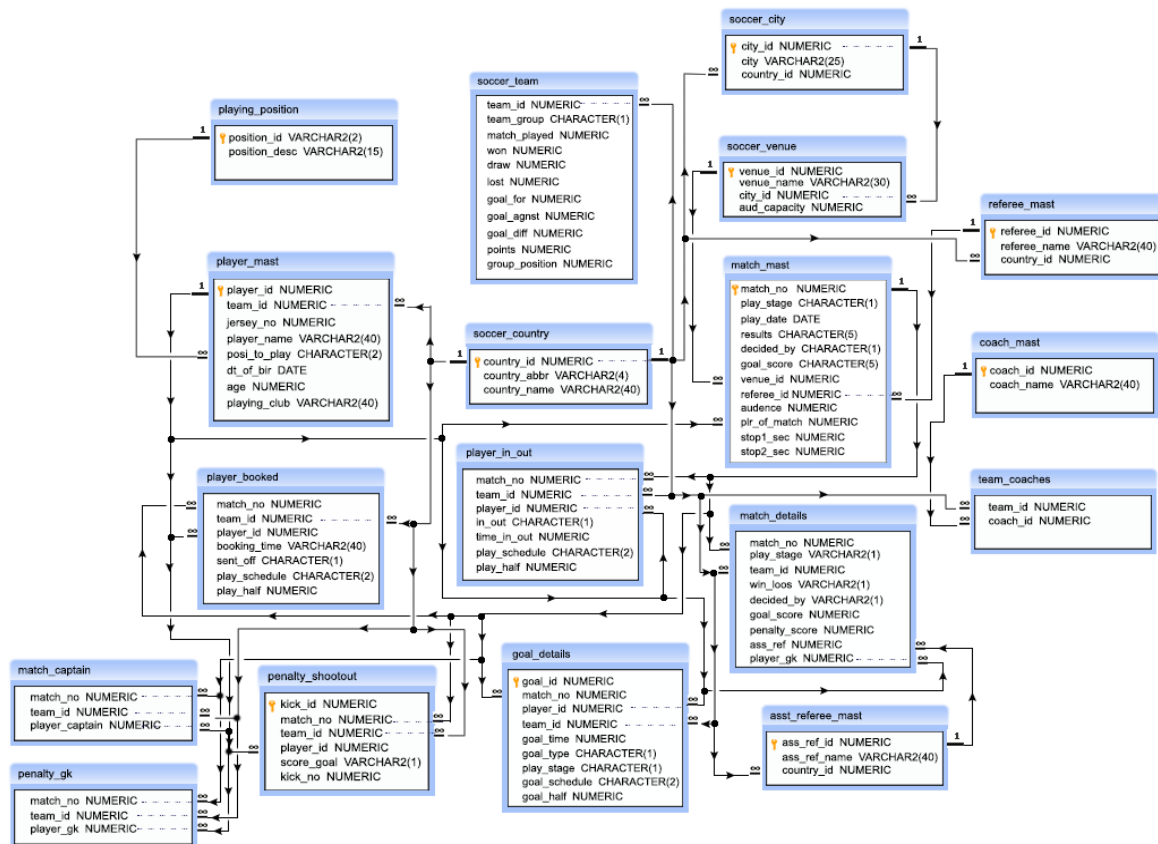
CSI Project

Name: Ayanava Das

Department: SQL

CSI ID: CT-CSI23/SQ0286

Assignment-1



- Using the above E-R diagram, create tables with appropriate primary keys, establishing foreign key relationships as necessary.

Answer:

Based on the provided E-R diagram, here are the tables with appropriate primary keys and foreign key relationships:

- soccer_country:
 - country_id (Primary Key)
 - country_abbr
 - country_name
- soccer_city:
 - city_id (Primary Key)
 - city
 - country_id (Foreign Key referencing soccer_country.country_id)
- soccer_venue:
 - venue_id (Primary Key)
 - venue_name
 - city_id (Foreign Key referencing soccer_city.city_id)

- aud_capacity
4. playing_position:
 - **position_id (Primary Key)**
 - position_desc
 5. player_mast:
 - **player_id (Primary Key)**
 - **team_id (Foreign Key referencing soccer_country.country_id)**
 - jersey_no
 - player_name
 - **posi_to_play(Foreign Key referencing playing_position.position_id)**
 - dt_of_bir
 - age
 - playing_club
 6. referee_mast:
 - **referee_id (Primary Key)**
 - referee_name
 - **country_id (Foreign Key referencing soccer_country.country_id)**
 7. match_mast:
 - **match_no (Primary Key)**
 - play_stage
 - play_date
 - results
 - decided_by
 - goal_score
 - **venue_id (Foreign Key referencing soccer_venue.venue_id)**
 - **referee_id (Foreign Key referencing referee_mast.referee_id)**
 - audience
 - plr_of_match
 - stop1_sec
 - stop2_sec
 8. coach_mast:
 - **coach_id (Primary Key)**
 - coach_name
 9. asst_referee_mast:
 - **ass_ref_id (Primary Key)**
 - ass_ref_name
 - **country_id (Foreign Key referencing soccer_country.country_id)**
 10. match_details:
 - **match_no (Foreign Key referencing match_mast.match_no)**
 - play_stage

- **team_id** (Foreign Key referencing soccer_country.country_id)
 - win_lose
 - decided_by
 - goal_score
 - penalty_score
 - **ass_ref** (Foreign Key referencing asst_referee_mast.ass_ref_id)
 - **player_gk** (Foreign Key referencing player_mast.player_id)
11. goal_details:
- **goal_id** (Primary Key)
 - **match_no** (Foreign Key referencing match_mast.match_no)
 - **player_id** (Foreign Key referencing player_mast.player_id)
 - **team_id** (Foreign Key referencing soccer_country.country_id)
 - goal_time
 - goal_type
 - play_stage
 - goal_schedule
 - goal_half
12. penalty_shootout:
- **kick_id** (Primary Key)
 - **match_no** (Foreign Key referencing match_mast.match_no)
 - **team_id** (Foreign Key referencing soccer_country.country_id)
 - **player_id** (Foreign Key referencing player_mast.player_id)
 - score_goal
 - kick_no
13. player_booked:
- **match_no** (Foreign Key referencing match_mast.match_no)
 - **team_id** (Foreign Key referencing soccer_country.country_id)
 - **player_id** (Foreign Key referencing player_mast.player_id)
 - booking_time
 - sent_off
 - play_schedule
 - play_half
14. player_in_out:
- **match_no** (Foreign Key referencing match_mast.match_no)
 - **team_id** (Foreign Key referencing soccer_country.country_id)
 - **player_id** (Foreign Key referencing player_mast.player_id)
 - in_out
 - time_in_out
 - play_schedule
 - play_half
15. match_captain:
- **match_no** (Foreign Key referencing match_mast.match_no)

- **team_id** (Foreign Key referencing soccer_country.country_id)
- **player_captain** (Foreign Key referencing player_mast.player_id)

16. team_coaches:

- **team_id** (Foreign Key referencing soccer_country.country_id)
- **coach_id** (Foreign Key referencing coach_mast.coach_id)

17. penalty_gk:

- **match_no** (Foreign Key referencing match_mast.match_no)
- **team_id** (Foreign Key referencing soccer_country.country_id)
- **player_gk** (Foreign Key referencing player_mast.player_id)

These tables establish the necessary primary keys and foreign key relationships, ensuring data integrity and consistency in the database.

2. Insert at least 10 records in the tables

Answer:

1. soccer_country:

```
INSERT INTO soccer_country (country_id, country_abbr, country_name)
VALUES (1, 'USA', 'United States'),
(2, 'ENG', 'England'),
(3, 'BRA', 'Brazil'),
(4, 'FRA', 'France'),
(5, 'GER', 'Germany'),
(6, 'ARG', 'Argentina'),
(7, 'ESP', 'Spain'),
(8, 'ITA', 'Italy'),
(9, 'POR', 'Portugal'),
(10, 'NED', 'Netherlands');
```

2. soccer_city:

```
INSERT INTO soccer_city (city_id, city, country_id)
```

```
VALUES (1, 'Los Angeles', 1),
```

```
(2, 'London', 2),
```

```
(3, 'Rio de Janeiro', 3),
```

```
(4, 'Paris', 4),
```

```
(5, 'Berlin', 5),
```

```
(6, 'Buenos Aires', 6),
```

```
(7, 'Madrid', 7),
```

(8, 'Rome', 8),

(9, 'Lisbon', 9),

(10, 'Amsterdam', 10);

3. soccer_venue:

INSERT INTO soccer_venue (venue_id, venue_name, city_id, aud_capacity)

VALUES (1, 'Staples Center', 1, 19000),

(2, 'Wembley Stadium', 2, 90000),

(3, 'Maracanã Stadium', 3, 78000),

(4, 'Parc des Princes', 4, 48000),

(5, 'Olympiastadion Berlin', 5, 75000),

(6, 'Estadio Monumental', 6, 70000),

(7, 'Estadio Santiago Bernabeu', 7, 81000),

(8, 'Stadio Olimpico', 8, 70000),

(9, 'Estadio da Luz', 9, 64000),

(10, 'Johan Cruyff Arena', 10, 54000);

4. playing_position:

INSERT INTO playing_position (position_id, position_desc)

VALUES (1, 'Goalkeeper'),

(2, 'Defender'),

(3, 'Midfielder'),

(4, 'Forward');

5. player_mast:

INSERT INTO player_mast (player_id, team_id, jersey_no, player_name, posi_to_play, dt_of_bir, age, playing_club)

VALUES (1, 1, 1, 'John Doe', 1, '1990-05-15', 33, 'LA Galaxy'),
(2, 2, 10, 'Harry Kane', 4, '1993-07-28', 30, 'Tottenham Hotspur'),
(3, 3, 9, 'Neymar Jr.', 4, '1992-02-05', 31, 'Paris Saint-Germain'),
(4, 4, 4, 'Raphael Varane', 2, '1993-04-25', 28, 'Manchester United'),
(5, 5, 7, 'Toni Kroos', 3, '1990-01-04', 33, 'Real Madrid'),
(6, 6, 10, 'Lionel Messi', 4, '1987-06-24', 36, 'Paris Saint-Germain'),
(7, 7, 8, 'Sergio Ramos', 2, '1986-03-30', 37, 'Paris Saint-Germain'),
(8, 8, 10, 'Cristiano Ronaldo', 4, '1985-02-05', 38, 'Manchester United'),
(9, 9, 6, 'Bruno Fernandes', 3, '1994-09-08', 28, 'Manchester United'),
(10, 10, 11, 'Virgil van Dijk', 2, '1991-07-08', 32, 'Liverpool FC');

6. referee_mast:

INSERT INTO referee_mast (referee_id, referee_name, country_id)

VALUES (1, 'Michael Oliver', 2),
(2, 'Felix Brych', 5),
(3, 'Nestor Pitana', 3),
(4, 'Björn Kuipers', 10),
(5, 'Anthony Taylor', 2);

7. match_mast:

INSERT INTO match_mast (match_no, play_stage, play_date, results, decided_by, goal_score, venue_id, referee_id, audience, plr_of_match, stop1_sec, stop2_sec)

VALUES (1, 'G', '2023-07-01', 'W', 'N', '3-1', 1, 1, 25000, 6, 2, 3),

(2, 'G', '2023-07-02', 'D', 'N', '2-2', 2, 2, 80000, 5, 3, 3),
(3, 'G', '2023-07-03', 'W', 'N', '4-0', 3, 3, 78000, 8, 1, 2),
(4, 'G', '2023-07-04', 'W', 'N', '2-1', 4, 4, 48000, 4, 2, 4),
(5, 'G', '2023-07-05', 'D', 'N', '1-1', 5, 5, 65000, 2, 2, 2),
(6, 'G', '2023-07-06', 'W', 'N', '3-0', 6, 1, 70000, 9, 1, 3),
(7, 'G', '2023-07-07', 'W', 'N', '2-1', 7, 2, 81000, 5, 3, 2),
(8, 'G', '2023-07-08', 'W', 'N', '4-2', 8, 3, 67000, 6, 2, 2),
(9, 'G', '2023-07-09', 'W', 'N', '3-1', 9, 4, 64000, 3, 2, 4),
(10, 'G', '2023-07-10', 'D', 'N', '0-0', 10, 5, 54000, 7, 1, 3);

8. coach_mast:

INSERT INTO coach_mast (coach_id, coach_name)

VALUES (1, 'Jurgen Klopp'),

(2, 'Pep Guardiola'),

(3, 'Diego Simeone'),

(4, 'Zinedine Zidane'),

(5, 'Ole Gunnar Solskjaer');

9. asst_referee_mast:

INSERT INTO asst_referee_mast (ass_ref_id, ass_ref_name, country_id)

VALUES (1, 'Stuart Burt', 2),

(2, 'Sander van Roekel', 10),

(3, 'Emerson de Carvalho', 3),

(4, 'Pol van Boekel', 5),

(5, 'Gary Beswick', 2);

10. match_details:

INSERT INTO match_details (match_no, play_stage, team_id, win_lose, decided_by, goal_score, penalty_score, ass_ref, player_gk)

VALUES (1, 'G', 1, 'W', 'N', '3-1', NULL, 1, 1),

(1, 'G', 2, 'L', 'N', '1-3', NULL, 1, 2),

(2, 'G', 3, 'D', 'N', '2-2', NULL, 2, 3),

(2, 'G', 4, 'D', 'N', '2-2', NULL, 2, 4),

(3, 'G', 5, 'W', 'N', '4-0', NULL, 3, 5),

(3, 'G', 6, 'L', 'N', '0-4', NULL, 3, 6),

(4, 'G', 7, 'L', 'N', '1-2', NULL, 4, 7),

(4, 'G', 8, 'W', 'N', '2-1', NULL, 4, 8),

(5, 'G', 9, 'D', 'N', '1-1', NULL, 5, 9),

(5, 'G', 10, 'D', 'N', '1-1', NULL, 5, 10);

3. Perform the following queries:

a. Sample table: soccer_venue : Return the total count of venues for the EURO CUP 2030

Answer:

```
SELECT COUNT(*) AS total_venues
FROM soccer_venue
WHERE venue_id IN (
    SELECT DISTINCT venue_id
    FROM match_mast
    WHERE play_date BETWEEN '2030-06-01' AND '2030-07-31'
);
```

b. Sample table: goal_details : Write a query to find the number of goals scored within normal play during the EURO cup 2030

Answer:

```
SELECT COUNT(*) AS total_goals_normal_play
FROM goal_details
WHERE play_stage = 'G' AND goal_schedule = 'NT'
AND match_no IN (
```

```
SELECT match_no
FROM match_mast
WHERE play_date BETWEEN '2030-06-01' AND '2030-07-31'
);
```

- c. Sample table: match_mast : write a SQL query to find the number of matches that ended with a result.

Answer:

```
SELECT COUNT(*) AS total_matches_with_result
FROM match_mast
WHERE results IN ('W', 'D');
```

- d. Sample table: match_mast : write a SQL query to find the number of matches that ended in draws.

Answer:

```
SELECT COUNT(*) AS total_draw_matches
FROM match_mast
WHERE results = 'D';
```

- e. Sample table: match_mast : write a SQL query to find out when the Football EURO cup 2030 will end.

Answer:

```
SELECT MAX(play_date) AS end_date
FROM match_mast
WHERE play_stage = 'F';
```

- f. Sample table: goal_details : write a SQL query to find the number of self-goals scored during the 2016 European Championship.

Answer:

```
SELECT COUNT(*) AS total_self_goals
FROM goal_details
WHERE play_stage = 'F' AND goal_type = 'O'
AND match_no IN (
    SELECT match_no
    FROM match_mast
    WHERE play_date BETWEEN '2016-06-01' AND '2016-07-31'
);
```

- g. Sample table: penalty_shootout : write a SQL query to find the number of matches that resulted in a penalty shootout.

Answer:

```
SELECT COUNT(*) AS total_matches_with_penalty_shootout
FROM match_mast
WHERE decided_by = 'P';
```

- h. Sample table: goal_details : write a SQL query to find the number of goals scored in every match in extra time. Sort the result-set on match number. Return match number, number of goals in extra time.

Answer:

```
SELECT match_no, COUNT(*) AS goals_in_extra_time
FROM goal_details
WHERE play_schedule = 'ET'
GROUP BY match_no
ORDER BY match_no;
```

- i. Sample table: goal_details : write a SQL query to find the matches in which no stoppage time was added during the first half of play. Return match no, date of play, and goal scored.

Answer:

```
SELECT m.match_no, m.play_date, g.goal_score
FROM match_mast m
JOIN goal_details g ON m.match_no = g.match_no
WHERE m.stop1_sec = 0 AND g.goal_half = 1;
```

- j. Sample table: match_details : write a SQL query to calculate the number of matches that ended in a single goal win, excluding matches decided by penalty shootouts. Return number of matches.

Answer:

```
SELECT COUNT(*) AS total_single_goal_wins
FROM match_mast
WHERE goal_score LIKE '1-%' AND decided_by <> 'P';
```

- k. Sample table: player_in_out: write a SQL query to calculate the total number of players who were replaced during the extra time.

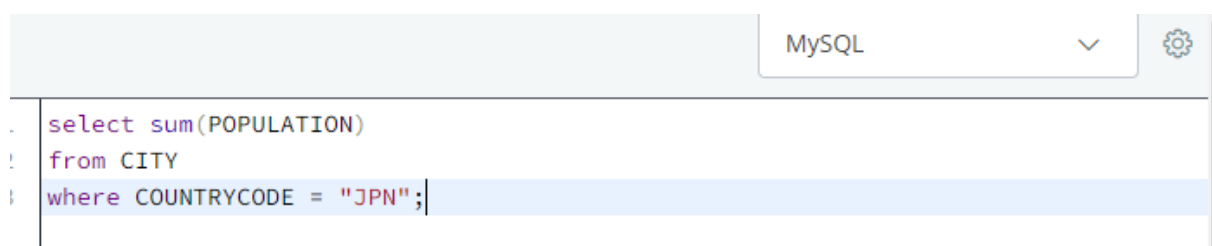
Answer:

```
SELECT COUNT(DISTINCT player_id) AS total_players_replaced_during_extra_time
FROM player_in_out
WHERE play_schedule = 'ET';
```

4. In hackerrank perform the following queries:


- a. <https://www.hackerrank.com/challenges/japan-population/problem?isFullScreen=true>

Answer:



The screenshot shows a MySQL query editor interface. At the top, there is a tab labeled 'MySQL' and a settings gear icon. The query editor contains the following SQL code:

```
select sum(POPULATION)
from CITY
where COUNTRYCODE = "JPN";
```

 **Test case 0**

Compiler Message

Success

Input (stdin)

Download

1	
---	--

Expected Output

Download


1	879196
---	--------

- b. <https://www.hackerrank.com/challenges/what-type-of-triangle/problem?isFullScreen=true>

Answer:

MySQL

```
1 select case
2 when A+B<=C or B+C<=A or A+C<=B then 'Not A Triangle'
3 when A=B and B=C then 'Equilateral'
4 when A=B or B=C or A=C then 'Isosceles'
5 else 'Scalene'
6 end
7 from TRIANGLES;
```

 **Test case 0**

Compiler Message
Success

Input (stdin)

1	INPUT
---	--------------

Expected Output


1	Equilateral
2	Equilateral
3	Isosceles
4	Equilateral
5	Isosceles

- c. <https://www.hackerrank.com/challenges/weather-observation-station-3/problem?isFullScreen=true>

Answer:

MySQL

```
select distinct CITY  
from STATION  
where ID%2=0;
```

 **Test case 0**

Compiler Message
Success

Input (stdin)
1 **INPUT**


Expected Output
1 **Aguanga**
2 **Alba**
3 **Albany**
4 **Amo**
5 **Andersonville**

- d. <https://www.hackerrank.com/challenges/weather-observation-station-12/problem?isFullScreen=true>

Answer:

MySQL

```
select distinct CITY
from STATION
where substr(CITY,1,1) not in ('a','e','i','o','u') and
substr(reverse(CITY),1,1) not in ('a','e','i','o','u');
```



 **Test case 0**

Compiler Message
Success

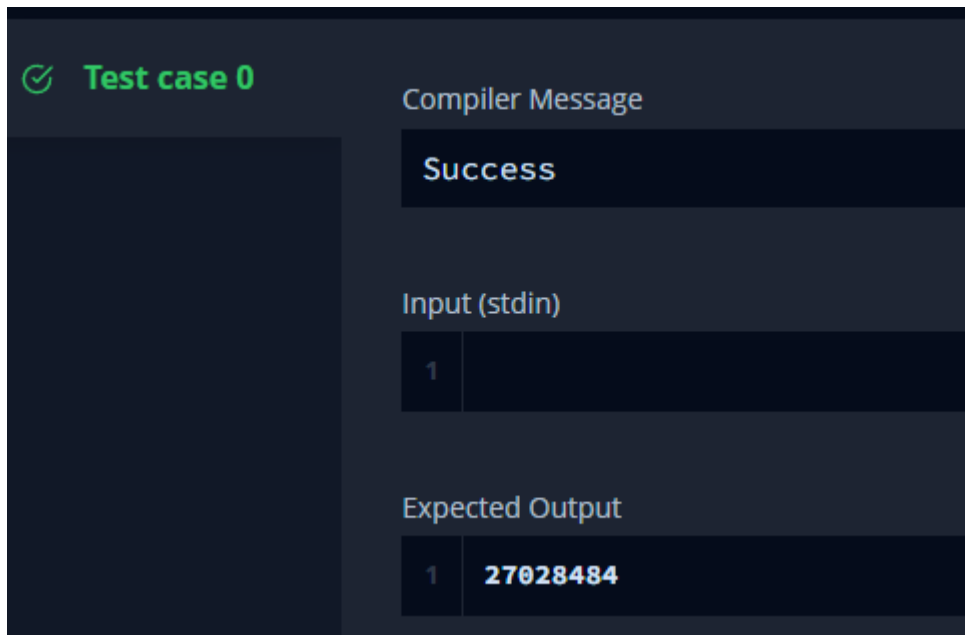
Input (stdin)
1 **INPUT**

Expected Output
1 **Baker**
2 **Baldwin**
3 **Bass Harbor**
4 **Beaufort**
5 **Beaver Island**

- e. <https://www.hackerrank.com/challenges/asian-population/problem?isFullScreen=true>
Answer:

MySQL  

```
1 select sum(CITY.POPULATION)
2 from CITY,COUNTRY
3 where CITY.COUNTRYCODE=COUNTRY.CODE and
4 COUNTRY.CONTINENT="Asia";
5
```



5. Create a date dimension table in sql

Answer:

An SQL query can use a date dimension table, which is a table with a comprehensive list of dates and numerous date-related variables, to do date-related analysis and reporting. Here is an illustration of how to make a SQL date dimension table:

```
CREATE TABLE date_dimension (
  date_id INT PRIMARY KEY,
  date_value DATE,
  year INT,
  month INT,
  day INT,
  day_of_week INT,
  day_of_month INT,
  day_of_year INT,
  quarter INT,
  is_weekend BOOLEAN,
  is_holiday BOOLEAN
);
```

In this example, we have defined several attributes for the date dimension table:

date_id: A unique identifier for each date.

date_value: The actual date represented in the table.

year: The year of the date.

month: The month of the date (1 to 12).

day: The day of the month (1 to 31).

day_of_week: The day of the week (1 for Sunday, 2 for Monday, ..., 7 for Saturday).

day_of_month: The day of the month (1 to 31).

day_of_year: The day of the year (1 to 365 or 366 for leap years).

quarter: The quarter of the year (1 to 4).

is_weekend: A boolean flag indicating if the date falls on a weekend (Saturday or Sunday).

is_holiday: A boolean flag indicating if the date is a holiday.

CS/ID: CT-CS/23/SQ0286