****

**LAB MIDTERM**

**SUBMITTED BY:** Kulsoom Khurshid

**REGISTRATION #:** SP20-BCS-044

**COURSE:** Database Management System-I

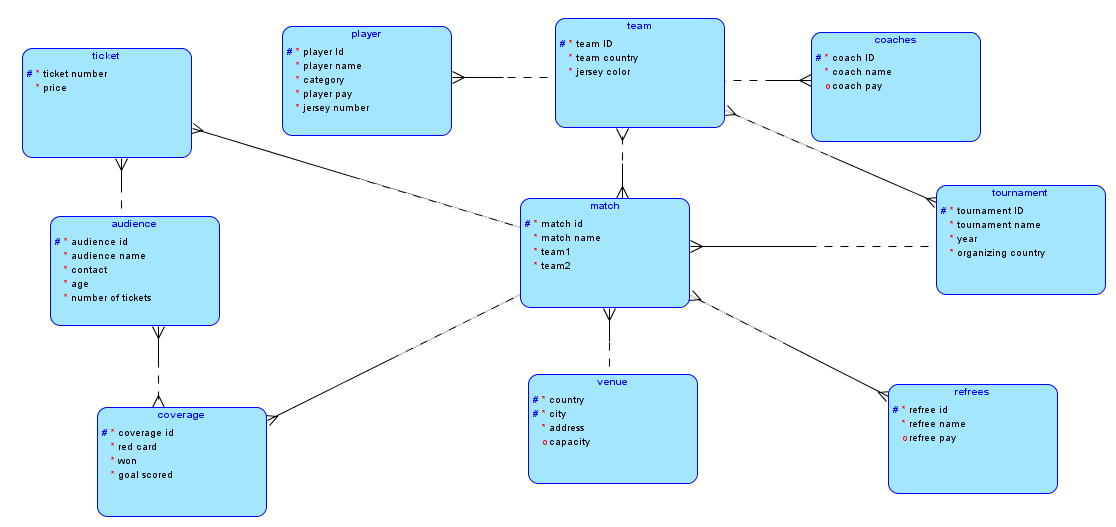
**SUBMITTED TO:** MR. Qasim Malik

**DATE:** 22ndNovember, 2021

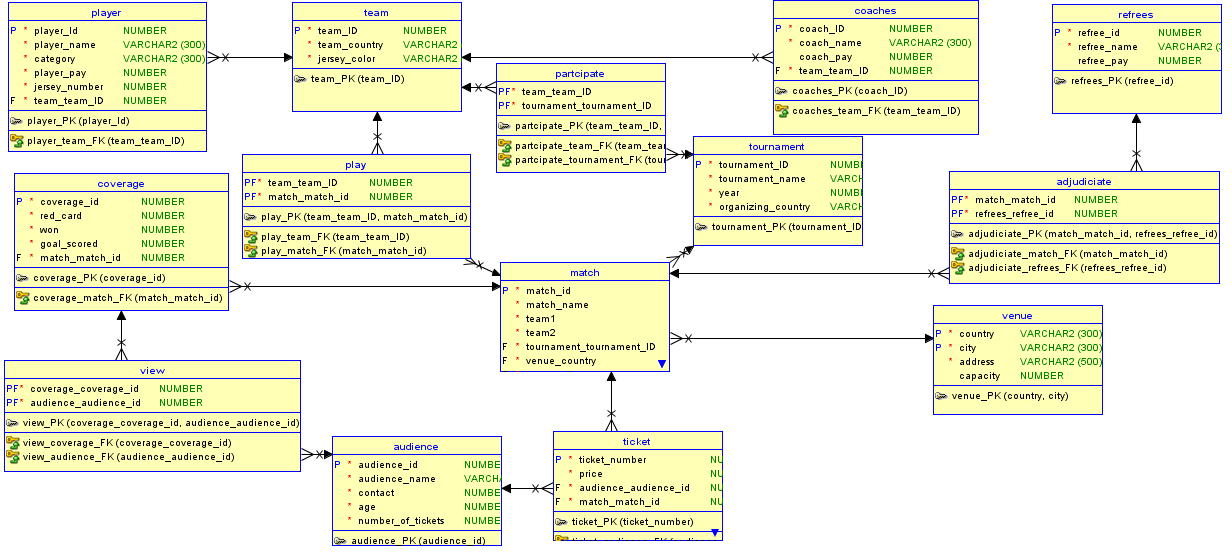
**Part-I [CLO-C3]:** Suppose the database for a sports website needs to be built. The following set of requirements have been identified:

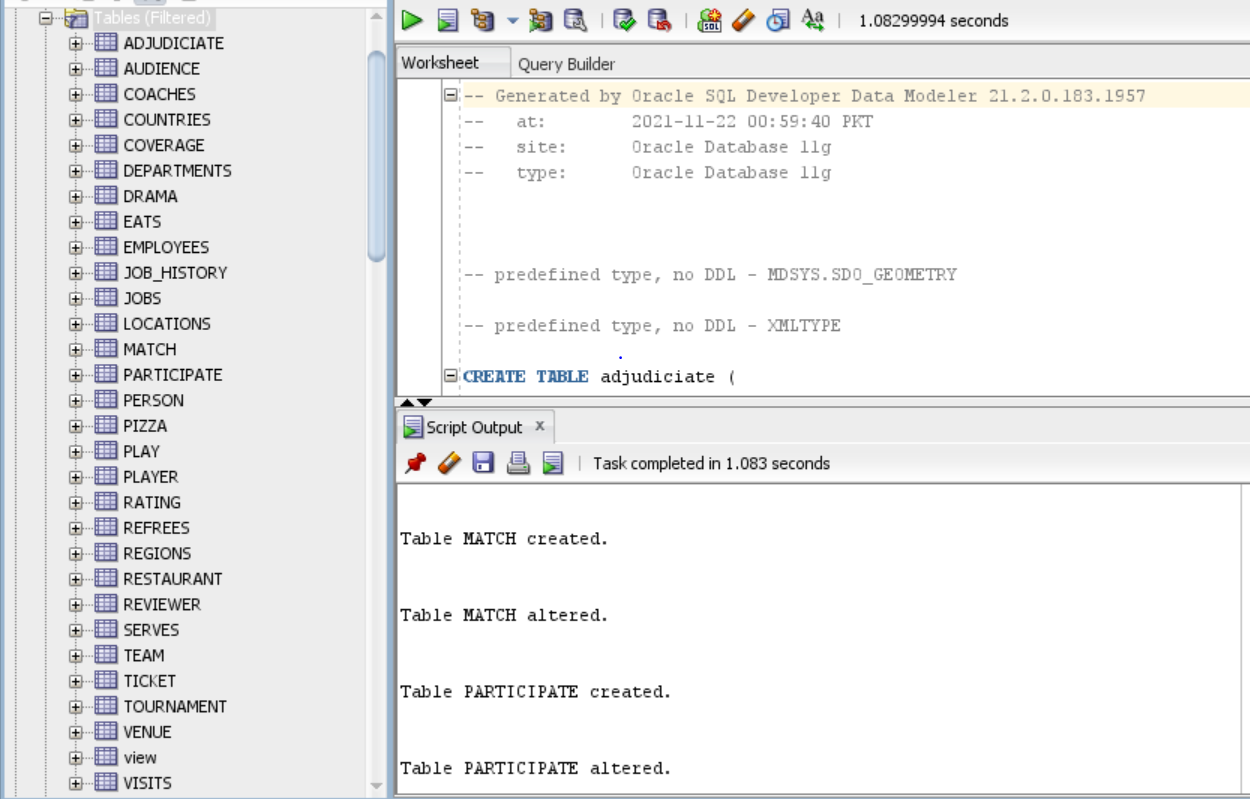
* Players group together to form teams.
* Teams have coaches.
* Teams can participate in tournaments.
* Tournaments comprise of matches.
* Referees adjudicate matches.
* Matches are played at a certain venue.
* A subset of players from a team participates in a certain match.
* For matches, apart from the basic information e.g. won/lost, its event-wise chronological coverage throughout the course of the match.

Given the above scenario, create an Entity Relationship Diagram (ERD) using a modeling tool of your choice. (Oracle’s Data Modeler tool is preferred).

**LOGICAL ERD: **

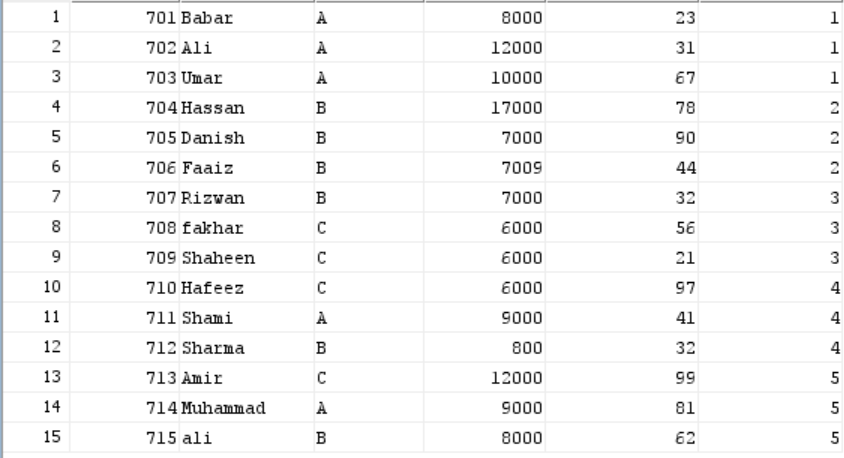
**RELATIONAL ERD:**

****

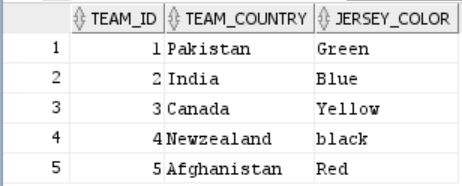
**Part-II [CLO-C2]:** Export the DDL script, corresponding to the relational schema of the ERD created in Part-I, to Oracle Database Management System (DBMS). 

**Part-III [CLO-C2]:** Insert a few meaningful tuples in the resultant relations.

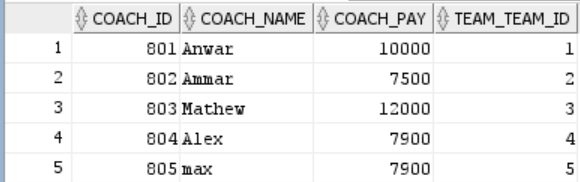
PLAYERS:

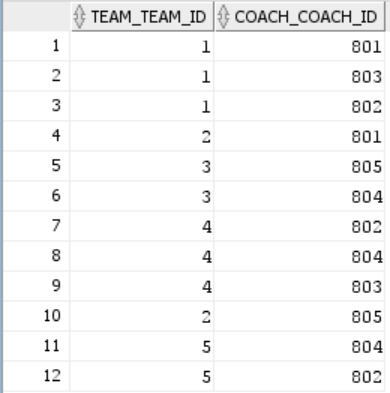


TEAM:

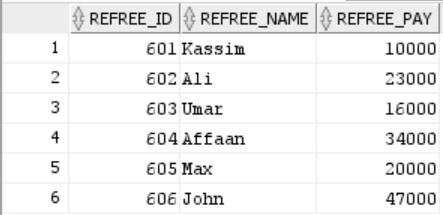


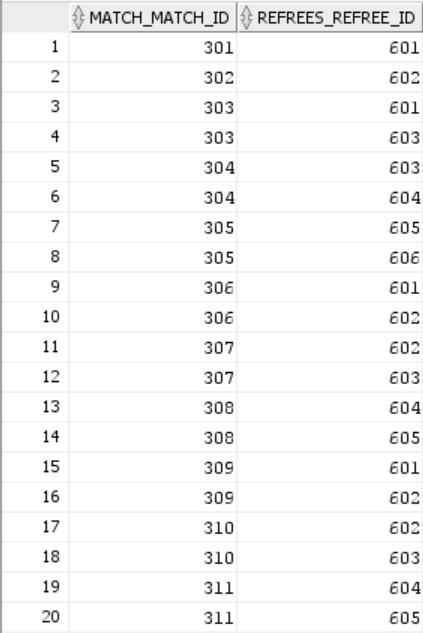
COACHES:

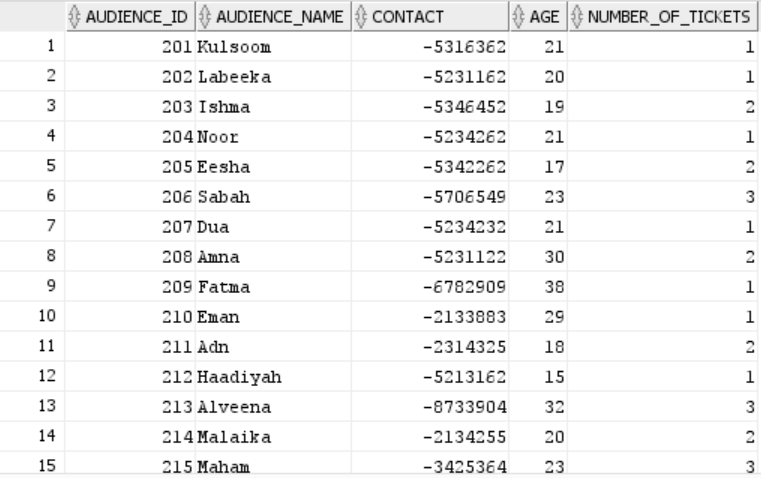


COACHED:

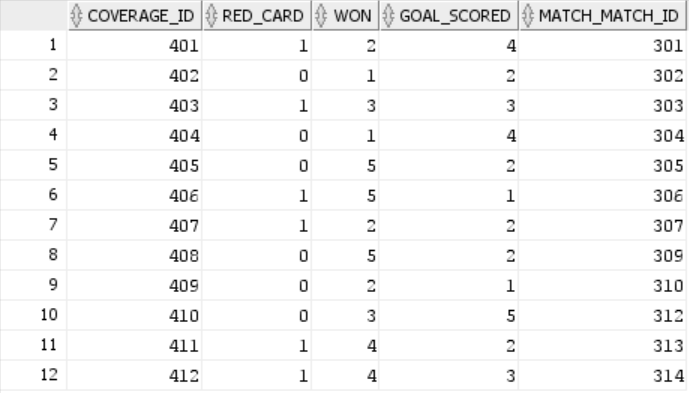
REFREE:

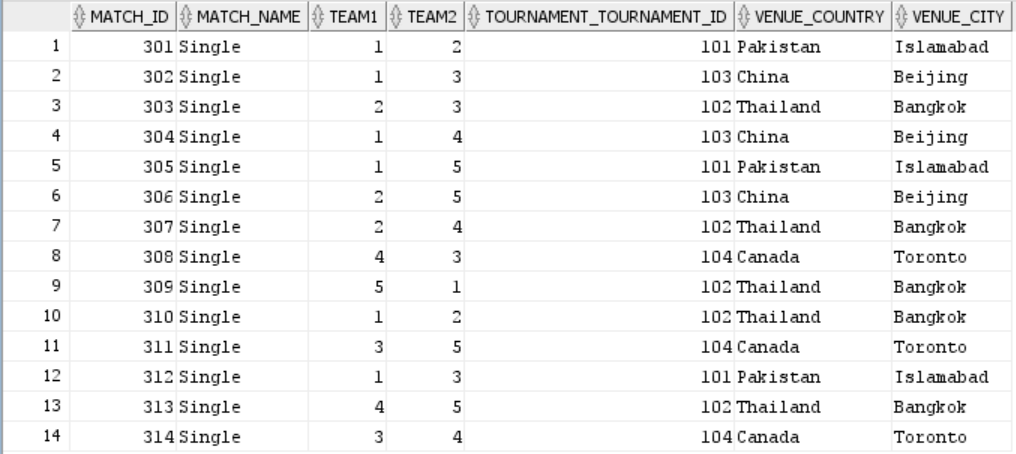
  
ADJUDICIATE:

  
AUDIENCE:

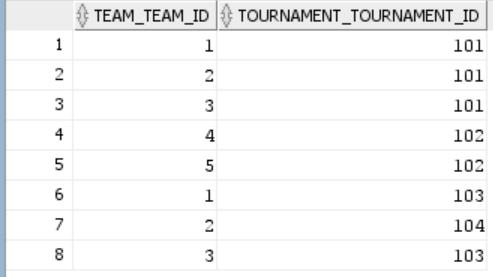


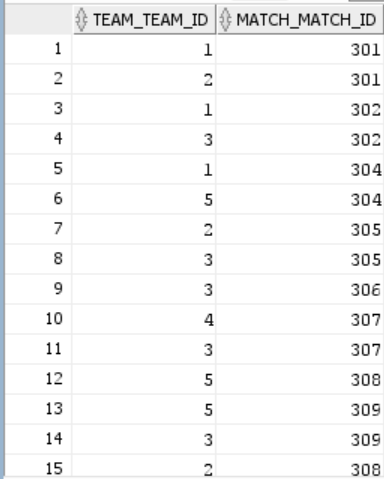
COVERAGE:

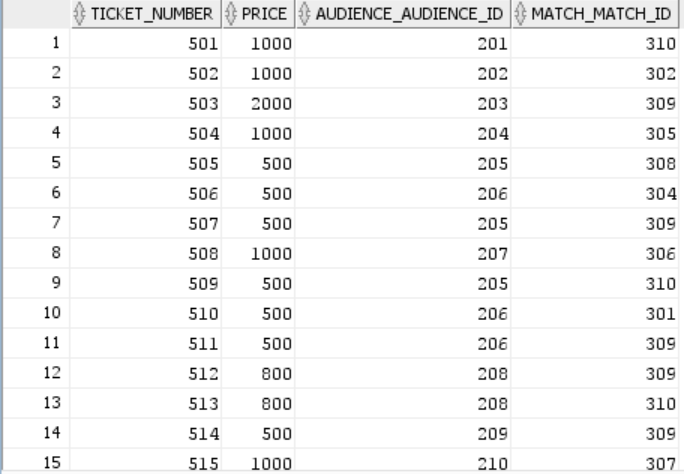
  
MATCH:



PARTICIPATE:

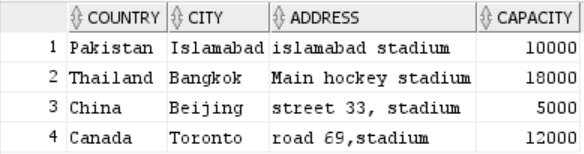
  
PLAY:

  
TICKET:

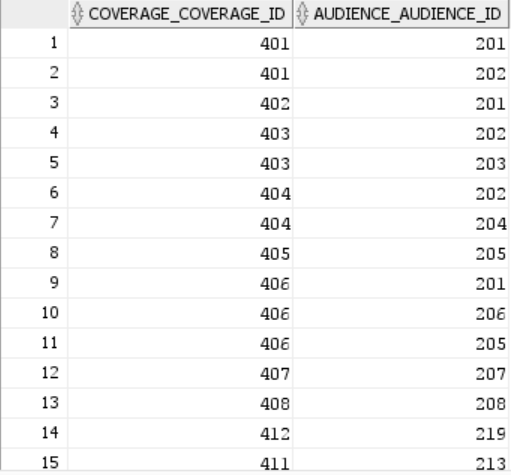


TOURNAMENT:

  
VENUE:



VIEW:



**Part-IV [CLO-C2]:** At this stage you would have put a lot of thoughts to the scenario in question and have an ERD that models all the given data requirements. You would now have a very good understanding of the system and may also anticipate the kind of queries that your database might need to answer in the future. Now in this part, you are going to first create a meaningful query in English and then in SQL for each of the following situations:

* Think of a need that will require joining at least 3 tables.

**English query:**

Display the name of the team that scored the maximum goals in matches along with the number of goals scored.

**SQL query:**

SELECT DISTINCT team.team\_country, coverage.goal\_scored

FROM (match NATURAL JOIN coverage NATURAL JOIN team)

WHERE coverage.goal\_scored = (SELECT MAX(goal\_scored) FROM coverage) AND coverage.won = team\_id;



* Think of a need that will require the use of join with grouping and aggregation.

**English query:**

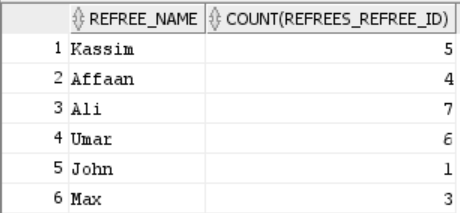
Display the name of the refrees and the number of matches that are adjudicates by each of them.

**SQL query:**

SELECT refree\_name, COUNT(refrees\_refree\_id)

FROM (match NATURAL JOIN adjudiciate NATURAL JOIN refrees)

WHERE match\_id = adjudiciate.match\_match\_id and refree\_id = adjudiciate.refrees\_refree\_id

GROUP BY refree\_name;

* Think of a need that involves the use of a sub-query.

**English query:**

Display the player id, name , pay and jersey number whose pay is more than the pay of Fakhar.

**SQL query:**

SELECT player\_id, player\_name, player\_pay, Jersey\_number

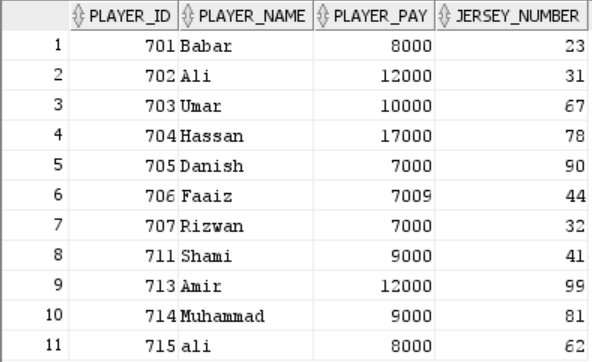
FROM player

WHERE player\_pay > (

SELECT player\_pay

FROM player

WHERE player\_name = 'fakhar' );



* Think of a need that involves the need of using any of the set operators.

**English query:**

Display player id, name, pay and jersey number who is paid the highest and lowest.

**SQL query:**

(SELECT player\_id, player\_name, player\_pay, Jersey\_number

FROM player

WHERE player\_pay = (

SELECT MIN(player\_pay)

FROM player

))

UNION

(SELECT player\_id, player\_name, player\_pay, Jersey\_number

FROM player

WHERE player\_pay = (

SELECT MAX(player\_pay)

FROM player));

