LAB MIDTERM

Name: Kulsoom Khurshid

Reg #: SP20-BCS-044

Course: Database Management

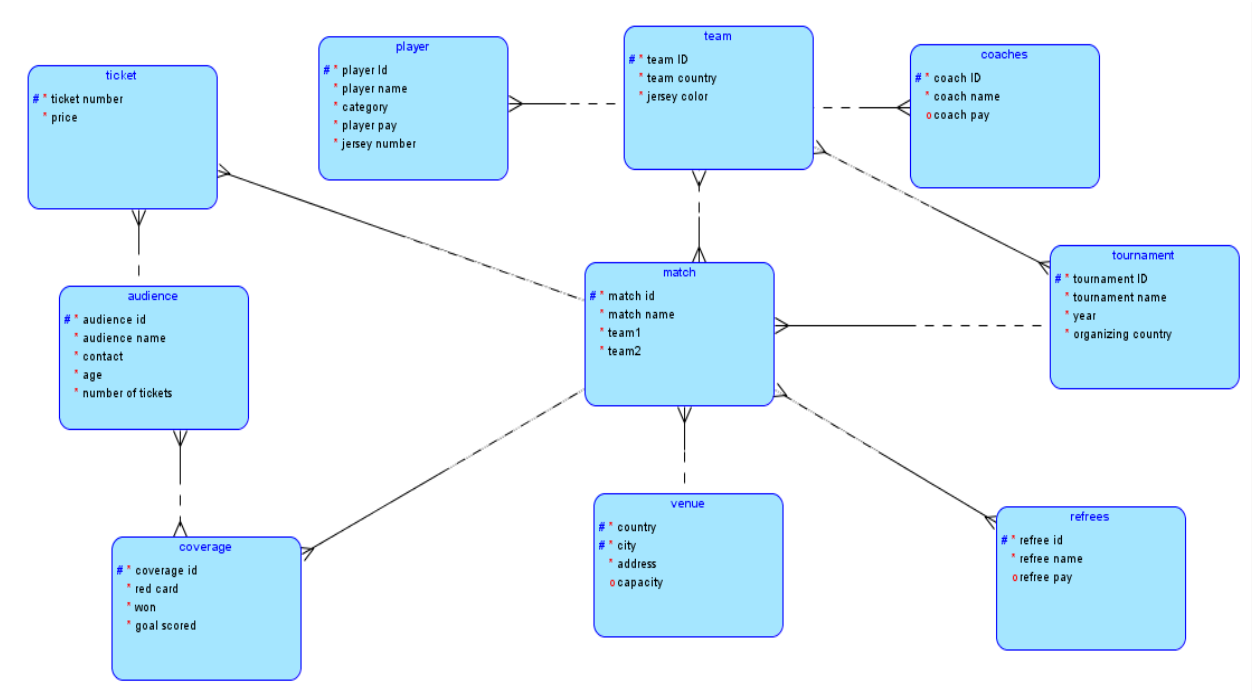
Instructor: Mr. Qasim Malik

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

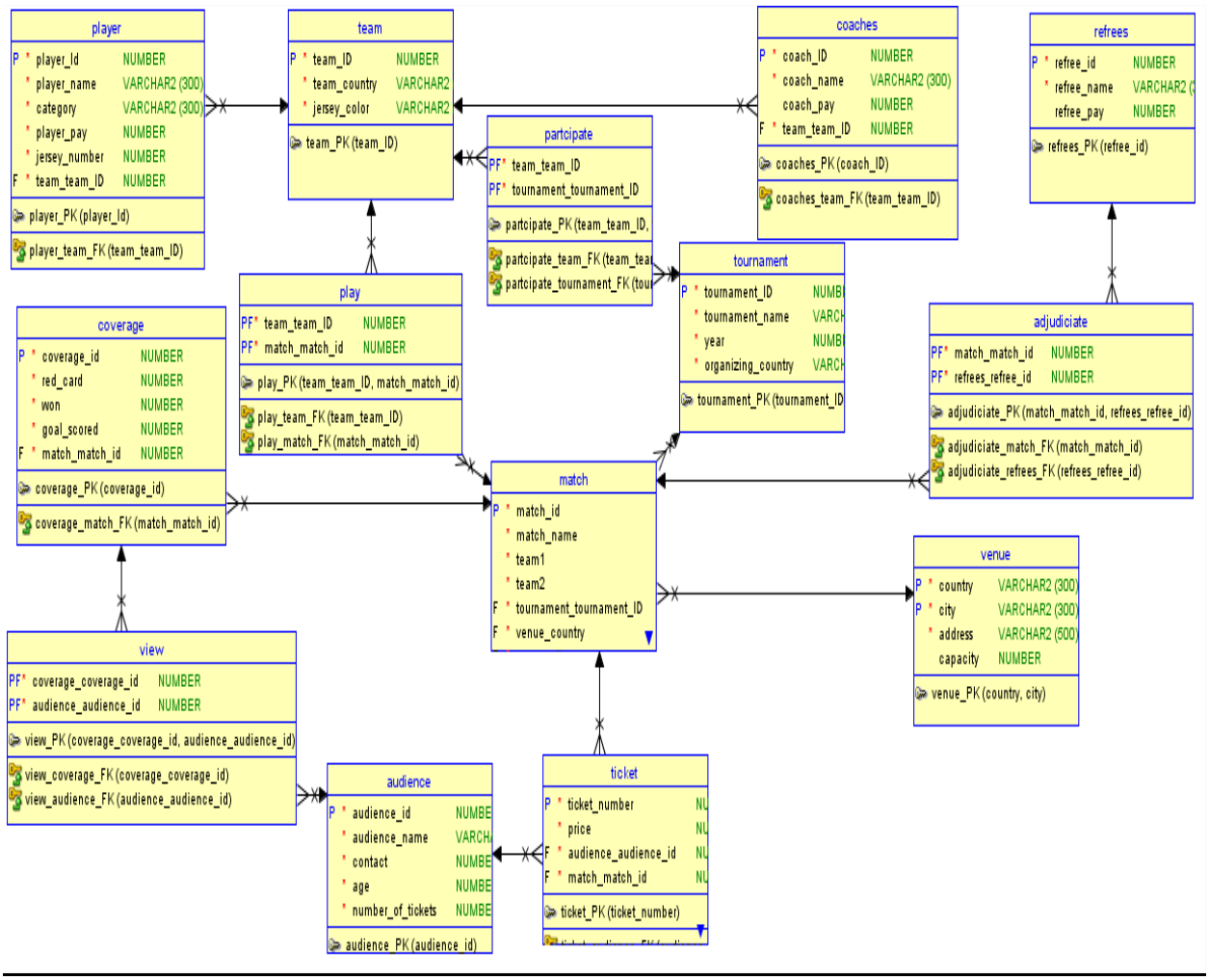
- Players group together to form teams.
- Teams have coaches.
- Teams can participate in tournaments.
- Tournaments comprise of matches.
- Referees adjudicate matches.
- Matches are played at a certain venue.
- A subset of players from a team participates in a certain match.
- For matches, apart from 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).

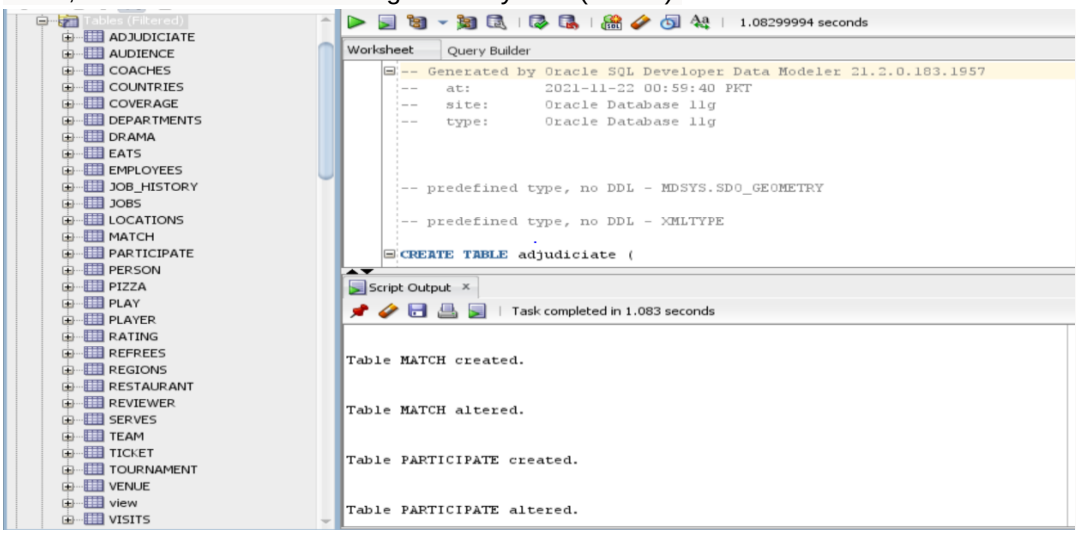
LOGICAL ERD:



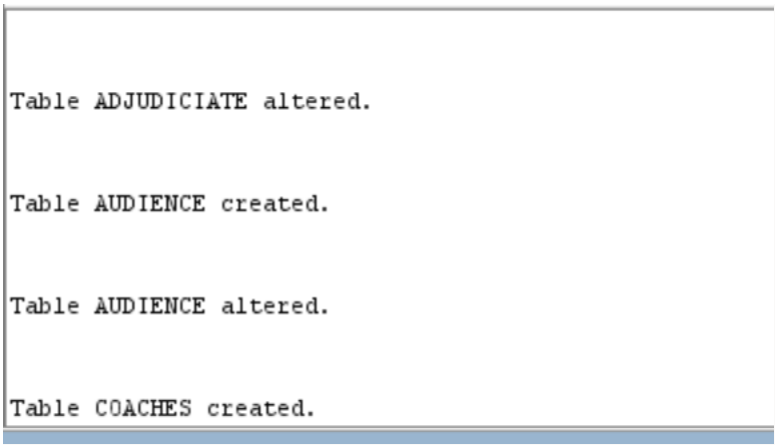
RELATIONAL ERD:

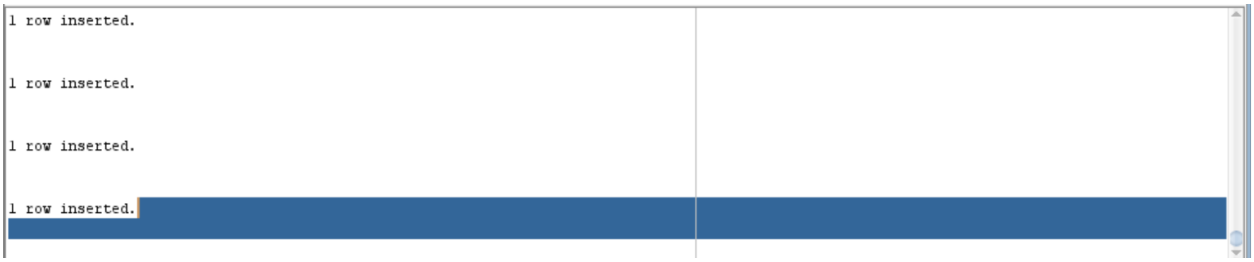


Part-II [CLO-C2]: Export the DDL script, corresponding to the relational schema of the ERD created in Part-I, to Oracle Database Management System (DBMS).



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





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:
Matches played by a particular player. For instance we need to find it for a player BABAR.
SQL query:
SELECT distinct player_name , match.team1, match.team2
FROM (player NATURAL JOIN team NATURAL JOIN match)
WHERE player_name = 'Babar' and player.team_team_id = team.team_id and (team.team_id = match.team1 or team.team_id = match.team2);

	PLAYER_NAME	TEAM1	TEAM2
1	Babar	1	4
2	Babar	5	1
3	Babar	1	3
4	Babar	1	5
5	Babar	1	2

- Think of a need that will require the use of join with grouping and aggregation.
English query:
Number of tickets bought by a particular audience.
SQL query:
SELECT audience.audience_name, COUNT(audience_audience_id)
FROM (audience NATURAL JOIN ticket)
WHERE audience.audience_id = ticket.audience_audience_id
GROUP BY audience.audience_name;

	AUDIENCE_NAME	COUNT(AUDIENCE_AUDIENCE_ID)
1	Labeeka	1
2	Sabah	3
3	Eman	1
4	Maham	3
5	Ehtisham	1
6	Alveena	3
7	Haadiyah	1
8	Noor	1
9	Fatma	1
10	Hammad	2
11	Manahil	1
12	Malaika	2
13	Hamza	2
14	Ishma	1
15	Eesha	3
16	Amna	2
17	Kulsoom	1
18	Dua	1
19	Adn	2

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

From all the audience, retrieve the youngest audience along with age.

SQL query:

```
(SELECT audience_name, age
FROM audience
WHERE audience.age = (
                SELECT MIN(age)
                FROM audience));
```

	AUDIENCE_NAME	AGE
1	Haadiyah	15

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

English query:

Combine the youngest and eldest audience.

SQL query:

```
(SELECT audience_name, age
FROM audience
WHERE audience.age = (
                SELECT MIN(age)
                FROM audience))
UNION
(SELECT audience_name, age
FROM audience
WHERE audience.age = (
                SELECT MAX(age)
                FROM audience));
```

	AUDIENCE_NAME	AGE
1	Fatma	38
2	Haadiyah	15