SEPTEMBER 25, 2024

HOMEWORK 2 B

CS 457 B

UMAIR DADA
2O281 - SFBU

1. **(3.16)**
    a. Constraints – The current constraint can make sure that each SECTION in the UNIVERSITY database is unique, so it can be identified easily and prevent any possible conflicts. Here are some Unique Attribute Constraints:
        i. **Classroom, DaysTime, Semester, Year:**
            1. This way we can ensure that one Section will only use a specific classroom during a specific time in a semester or a year. The combination of Croom, DaysTime, Year, and Sem must be unique.
        ii. **Instructor, DaysTime, Semester, Year:**
            1. This way, we can prevent the instructor from being assigned to be teaching multiple Sections at the same time during a semester and the year. IName, DaysTime, Year, and Sem must be unique.
        iii. **Course, Section Number, Semester, Year:**
            1. This will make sure that there are no problems of having same section numbers for each and every course. CCode, SecNo, Year, and Sem need to be unique.
    b. **Other Similar Constraints:**
        i. **Classroom Capacity:**
            1. This will prevent the number of students in a section to ensure the class capacity is not exceeded.
        ii. **Prerequisites:**
            1. To make sure the student has completed the prerequisite courses to attend a specific course, if there any.
        iii. **Time Conflicts:**
            1. Students won't be able to enroll in multiple sections that would overlap in time.
        iv. **Instructor Conflicts:**
            1. To teach like a high level course, the instructor needs to have enough qualifications and experience, so the students receive proper education.

By defining these unique attribute combinations and considering additional constraints, we can ensure the integrity and consistency of the UNIVERSITY database and prevent conflicts in resource allocation.

2. (3.19)

a. Requirements:

i. Airport – Each airport must have an airport code, city, state, and name. The departure and arrival airports are related to the flight legs, which shows where each flight departs and lands.

ii. Flight Leg – Some flights have multiple legs, and each leg is identified uniquely. For each leg, the schedules departure and arrivals are required to be tracked.

iii. Flight Instances – Flights have legs, which can have multiple instances on different times. Each instance of leg is linked to the airline, flight number, operating weekdays, and other info such as restrictions.

iv. Airplanes – They are categorized by their model, which includes info like the max number of seats and the manufactures name or airline name. The schema needs to track the number of seats and like the airplane to its specific type.

v. Reservations – It's associated with a customer with name and phone number. Each reservation includes a seat number and is linked to one specific instance of a flight.

vi. Fares – Each fare is identified by its individual code and has an amount. Fare restrictions are linked with each flight leg.

b. Constraints:

i. Airport Relationships – Airports can be the departure or arrival point for multipe flight legs, but each flight leg can have one departure and arrival airport.

ii. Airplane Type Relationship – Airplanes belong to specific types and each type may have multiple airplanes.

iii. Flight Leg – Flight legs can have multiple instances, each with a specific date to create a difference between them.

iv. Seat Reservation – Seats are reserved for flight leg instances, a customer can have multiple reservations.

v. Instance and Departure/Arrival – Flight leg instances have specific departure and arrival times, and the number of available seats are tracked every time.

vi. Fare Structure – Each flight has multiple fares, which means the same fare can be used for different flights, and each flight can have multiple fares.

c. Additionally

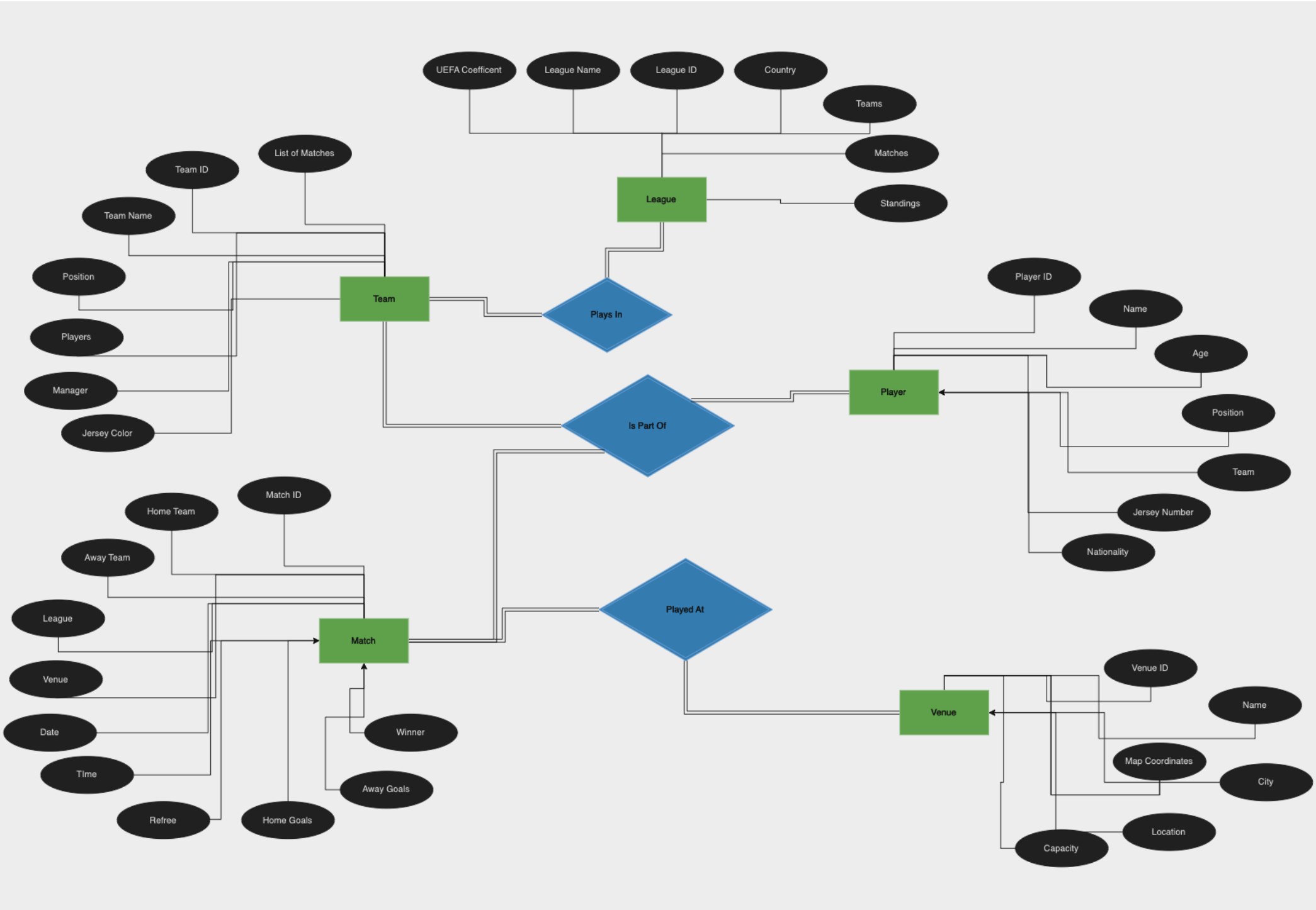
i. A flight leg is a nonstop route between two airports.

ii. A leg instance is a specific occurrence of that route on a particular day.

3. (3.22)

a. Diagram (ON NEXT PAGE)

b. This ER diagram provides a foundation for a database that can store information about teams, players, games, and player participation in Football (UEFA).


UEFA Coefficent
League Name
League ID
Country
Teams
Matches
Standings
League
List of Matches
Team ID
Team Name
Position
Players
Manager
Jersey Color
Team
Plays In
Player ID
Name
Age
Position
Team
Jersey Number
Nationality
Player
Is Part Of
Match ID
Home Team
Away Team
League
Venue
Date
TIme
Refree
Home Goals
Away Goals
Winner
Match
Played At
Venue ID
Name
Map Coordinates
City
Location
Capacity
Venue

4. **(3.24)**

   a. **CONSTRAINTS.**

      i. EMPLOYEE to WORKS_IN:

         1. The employee may be working for 2 different departments or not in any department at all.
         2. Minimum = 0 / Maximum = 2 (EMPLOYEE side)

      ii. DEPARTMENT to WORKS_IN:

         1. There is no limit on how many employees could be assinged to a single department but all departments must have atleast 1 employee.
         2. Minimum = 1 / Maximum = Many (DEPARTMENT side)

      iii. DEPARTMENT to CONTAINS:

         1. A department needs to have at least 1 phone number and maximum 3 phone numbers.
         2. Minimum = 1 / Maximum = 3 (DEPARTMENT side)

      iv. PHONE to CONTAINS:

         1. Each phone is assigned to one department, since it's not shown that the departments can share the same phone number.
         2. Minimum = 1 / Maximum = 1 (PHONE side)

      v. EMPLOYEE to HAS_PHONE

         1. The diagram is not showing any indication of how many phones the employee can have so I am assuming there is not limit or requirement.
         2. Minimum = 0 / Maximum = Many (EMPLOYEE side)

      vi. PHONE to HAS_PHONE:

         1. It's not necessary that the phone number must belong to an employee and phones can be shared or personal.
         2. Minimum = 0 / Maximum = 1 (PHONE side)

b. REDUNDANCY.

i. The Phone Numbers are associated with the departments, there is no indication that employees have a direct phone number, so I am going to assume there is not option to contact directly. Based on my assumption, the CONTAINS relationship between DEPARTMENT and PHONE should be enough to get the phone information, and I believe that the employees can access indirectly through the departments.

ii. If the concern is only with the department phone numbers then the HAS_PHONE relationship not be useful anymore.

5. (3.27)

a. The table below indicates the cardinality ratios for each pair of entities, along with the assumptions based on my abilities:

Entity 1	Entity 2	Cardinality Ratio and Assumption.
STUDENT	SOCIAL_SECURITY_CARD	1 TO 1 – A student can have only 1 Social Security Number
STUDENT	TEACHER	MANY TO MANY – Multiple Teachers can be teaching multiple student. Students can be taking lectures from multiple teachers.
CLASSROOM	WALL	MANY TO MANY – Walls are shared between 2 or more classes.
COUNTRY	CURRENT_PRESIDENT	1 TO 1 – A country can have only 1 president at time.
COURSE	TEXTBOOK	MANY TO MANY – Each course could have multiple number of textbooks. Some textbooks can be used for multiple courses as well.

ITEM	ORDER	MANY TO MANY – The order can have multiple items. Different orders can have the same item.
STUDENT	CLASS	MANY TO MANY – A student can enroll into multiple classes. A class can have multiple students.
CLASS	INSTRUCTOR	1 TO 1 – An instructor can only teach one class during specific hours.
INSTRUCTOR	OFFICE	1 TO 1 – An instructor can have only one office. Each office would belong to only one instructor
EBAY_AUCTION_ITEM	EBAY_BID	1 TO MANY – Multiple bids can be placed on one single item. A bid cannot have multiple items.