# NYU, Tandon School of Engineering

September 24, 2024

# CS-GY 6083

# Principles of Database Systems

# Section A, Fall 2024

# Homework #1

Submitted by:

Khwaab Thareja

N15911999

Kt3180

&

Ansh Harjai

N14452996

ah7163

Guided By: Prof Phyllis Frankl

**Problem 1**

a. Welcome to New York, home of Broadway Shows. The Tony awards are an annual award ceremony, like the Oscars, but for broadway shows, rather than movies. There are numerous award categories. Design a simple ER diagram modeling data representing which production won which award when. Your ER diagram does not need to model awards for individual actors. There should be entity sets representing Shows and Awards (each with a reasonable primary key and a couple of other attributes), and one or more relationship set, indicating which shows were nominated and which shows won which award in which year. The model doesn't have to indicate that the winner of an award was among the nominees, but it should indicate that there is exactly one winner of each award each year.

b. Show a little sample data for a few shows, a few award categories and which shows were nominated for and won those categories in the 77th annual Tony Awards. You may use any reasonable notation to indicate elements of the relationship set(s), e.g. (entity1, entity2) or lines connecting entities from the participating entity sets, etc

Answer 1.a

While making the ER diagram, Following the instructions provided in the problem statement

- Does not need to model for individual actors
- Must have entity sets representing Shows and Awards (Show primary key and other attributes)
- Must have one or more relationship set and should represent the nomination and winner with their respective years.
- Should indicate exactly one winner of each award each year.

Considering the above pointers, We will make two entity sets [Shows; Awards] and join them two relationship sets [Wins;Nomination] and give '*year*' attribute to both the relationship sets.

So one show can win multiple awards, but one award can have exactly one winner. This establishes a many to one relationship among the show and award, with show winning multiple award.

Also, an Award can have multiple nominations, similarly a show can be nominated for multiple awards. So this establishes a Many to Many nomination relationship between the show and awards entity sets.

Considering all the factors, we get an ER model shown below:

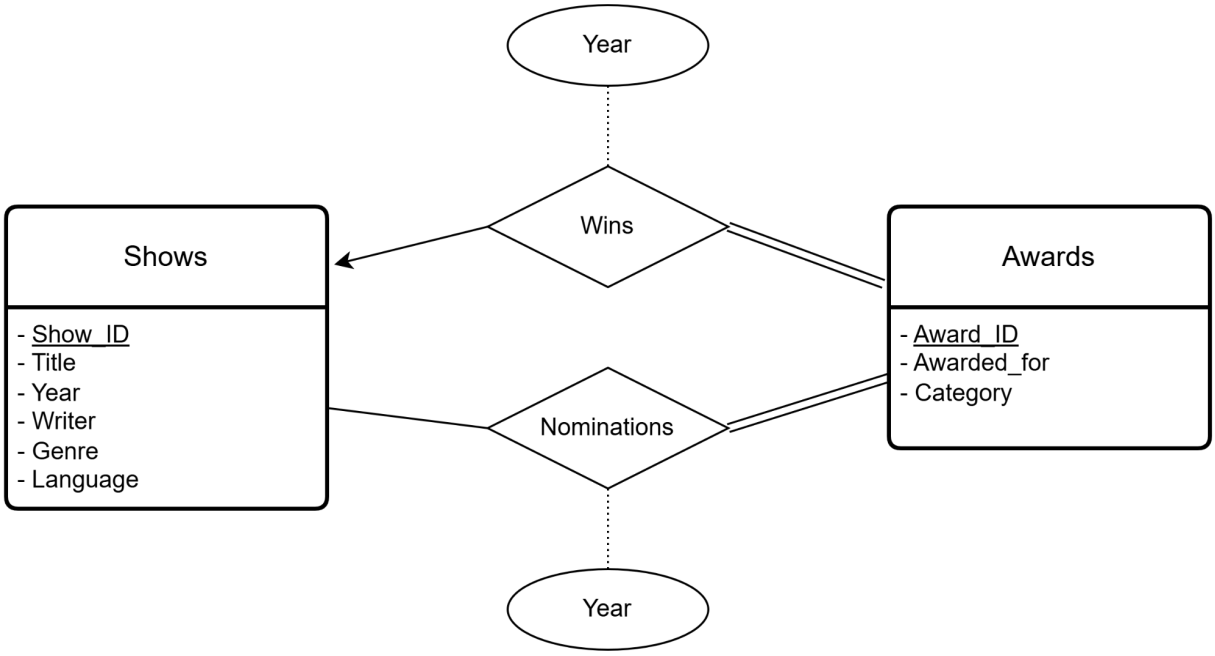



Fig 1.a

Answer 1.b

Taking reference for sample data from 77th annual Tony Awards awards Wikipedia page

**Shows**

| show_id | year | title | writer | genre | language |
|---|---|---|---|---|---|
| S1 | 2023 | "Some Like It Hot" | Matthew López | Musical | English |
| S2 | 2023 | "Leopoldstadt" | Tom Stoppard | Play | English |
| S3 | 2023 | "Kimberly Akimbo" | David Lindsay-Abaire | Musical | English |
| S4 | 2023 | "A Doll's House" | Henrik Ibsen (Adapted) | Play | English |

## Awards

| Award_id | Awarded_for | Category |
|---|---|---|
| A1 | Best Musical | Musical |
| A2 | Best Play | Play |
| A3 | Best Revival of a Play | Revival Play |

**Relationship:**

We have two relationship sets, namely **Wins** and **Nominations.**

**Representing in the form of E1→ R→ E2**

Nominations:

A1→Nomination →S1 // For eg : Best Musical→ Nominated→"Some Like It Hot"

A2→Nomination →S2

Wins:

A2→Wins →S2

Also, we can represent it in tabular form as:

**Nominated**

| show_id | award_id | year |
|---|---|---|
| S001 | A001 | 2023 |
| S003 | A001 | 2023 |
| S002 | A002 | 2023 |
| S004 | A003 | 2023 |

**Wins**

| show_id | award_id | year |
|---|---|---|
| S001 | A001 | 2023 |
| S002 | A002 | 2023 |
| S004 | A003 | 2023 |

**Problem 2**

A shoe store has hired you to design a database to keep track of their inventory and orders. The Inventory is a collection of (pairs of) shoes. Each type of shoe the store carries is identified by its brand, styleID, size, and color; in addition, the inventory keeps track of the quantity of each type of shoe that is currently in stock. Each Customer has a unique email, any number of phone numbers, a number of bonus points, and an address composed of their street address, which in turn is composed of the building number, street name, and apartment number, and a city, state, and zip code.

a. Using the notation studied, design an ER diagram with entity sets representing Customer and Inventory and a relationship set representing Current Orders. The model should enable tracking of the date on which an order is made, the status of the order, the number of (pairs of) each type of shoe in an order, the price paid for each type of shoe, and the status of the order. For part (a), assume that customers cannot have separate current orders of the same shoe type (though they can order several identical pairs in a single order). Your ER diagram for part (a) should be as simple as possible. In particular, it should have only one relationship set and that relationship set should be binary.

b. The store manager realizes that they would also like to keep track of the history of all of the orders that have been made. Briefly explain why the ER diagram from part (a) does not allow representation of a customer who makes multiple orders of the same type of shoe on different dates. Then modify the ER diagram, adding any needed entity sets and replacing Current Orders by a relationship set Orders which can represent past and present orders of the same type of shoe by the same customer.

c. Now the manager realizes that in addition to storing the prices customers paid, they would like to store the current price of each type of shoe and a description of each type of shoe. Furthermore, the current price and the description are determined by the brand and styleID. In other words, shoes that have the same brand and styleID always have the same current price and description, even though they may have different size and/or color. In addition, the manager would like to allow each customer to designate one Favorite shoe type (which should not involve the size or the color). Modify the ER diagram accordingly. HINT: Use a weak entity set; think carefully about which weak or strong entity sets participate directly in each relationship set. [For part c) you may start with the assumptions either in part a) or part b)

d. The store is so successful that they've opened multiple locations. Modify the ER diagram to keep track of the inventories and orders at different locations. Customers can order from any location. Each Customer has a home location.

Answer 2.a

Since we have to keep track of Inventory and orders with the customers, we will make 2 entity sets, namely Inventory and Customers. The Inventory entity set will have Inventory attributes like Shoe_ID[PK], Brand, Style, Size, Color, and Quantity. On the other hand, for the customers, we have the attributes like Customer_ID[PK],Email, phone _no, Bonus_points and Address. **We could have used email_id for the primary key, given that it is unique**, but using customer_id for ease. Similarly, for Inventory, we could have used a composite primary key of [Brand, Style_ID, Size, Color], but we have created a new attribute Shoe_ID for ease. Creating a relationship set amongst these entities namely 'Current Orders' and assigning attributes as stated. We have also used a total participation between current orders and inventory, since all orders will have a shoe from the inventory. Moreover, since a customer can make multiple orders of the same show type, we have not used a directed arrow from Current Orders relation towards Inventory as that would represent that a customer can have at most one order. This relationship set will not have separate current orders for the same type of shoe as stated.

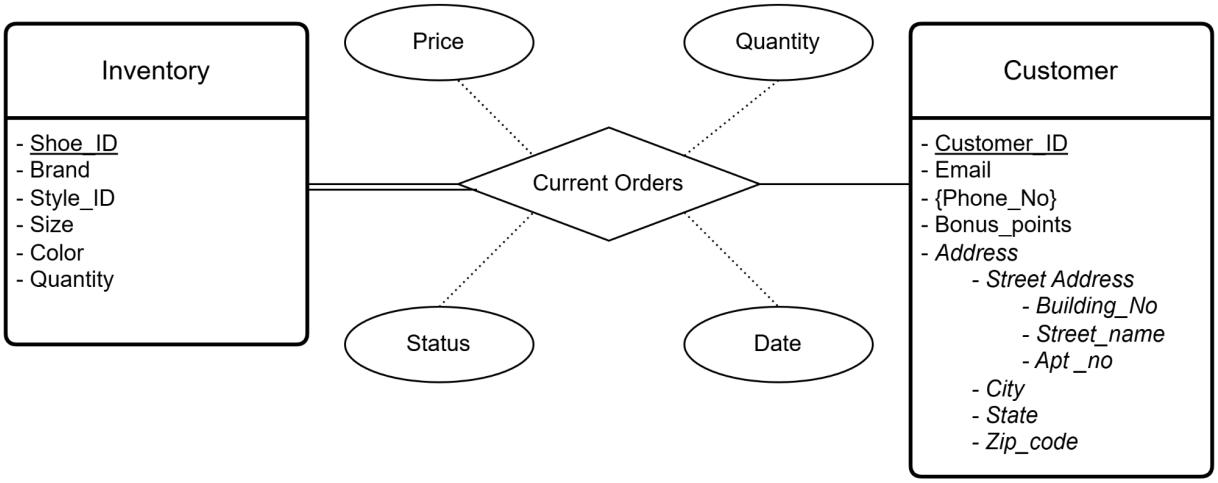



Fig 2.a

Answer 2.b

In part (a), we observed that only a single order will be taken care of for a specific style of shoe. So the above ER model would not capture multiple orders from a customer for the same type even on different dates. This is because the diagram does not store the order information but only stores the information about the inventory and customers and establishes a relation between these two entities.

Also, The model focuses only on current orders and does not store order related information or the past orders. Also if it allows multiple orders of the similar type of shoe, then it would have to foster many to many relation, whereas currently it establishes an one to many relation.

To modify this, so that the model tracks the history as well, we will have a separate entity for tracking orders and will replace the ‘Current Orders’ relationship set with orders. The newly formed order entity set will have attributes like Order_ID [PK], Order_Date, Status,Price paid and quantity of the shoe ordered.

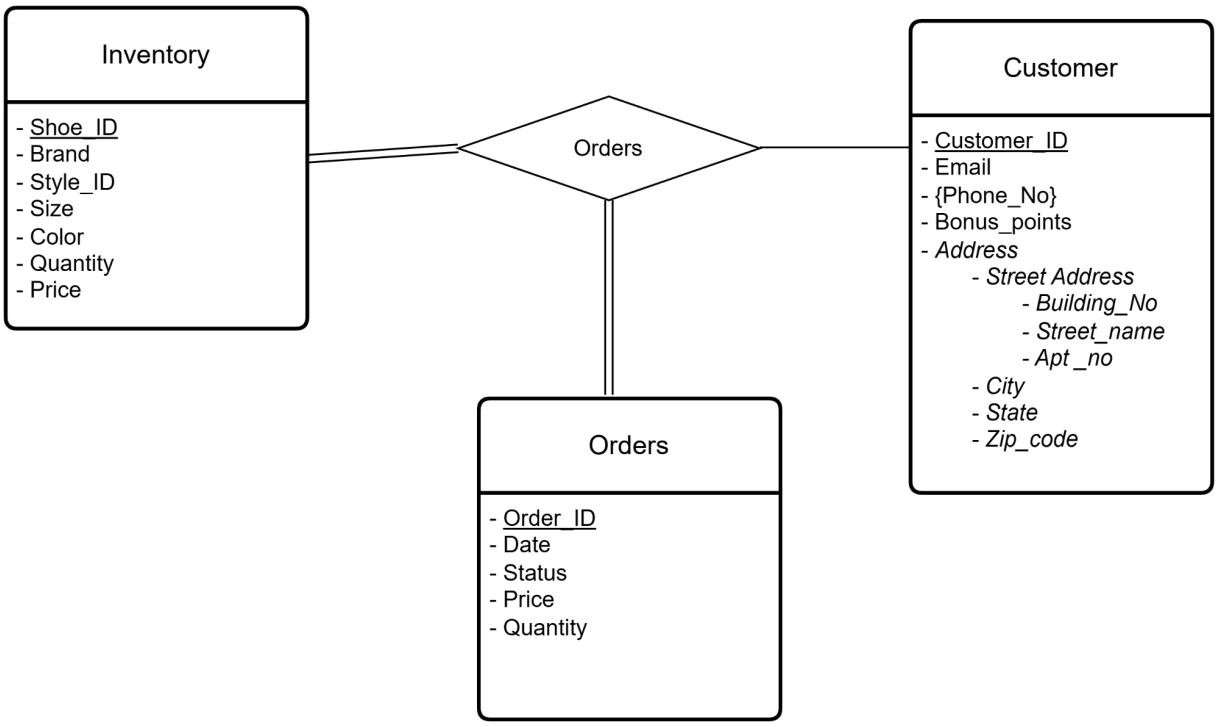
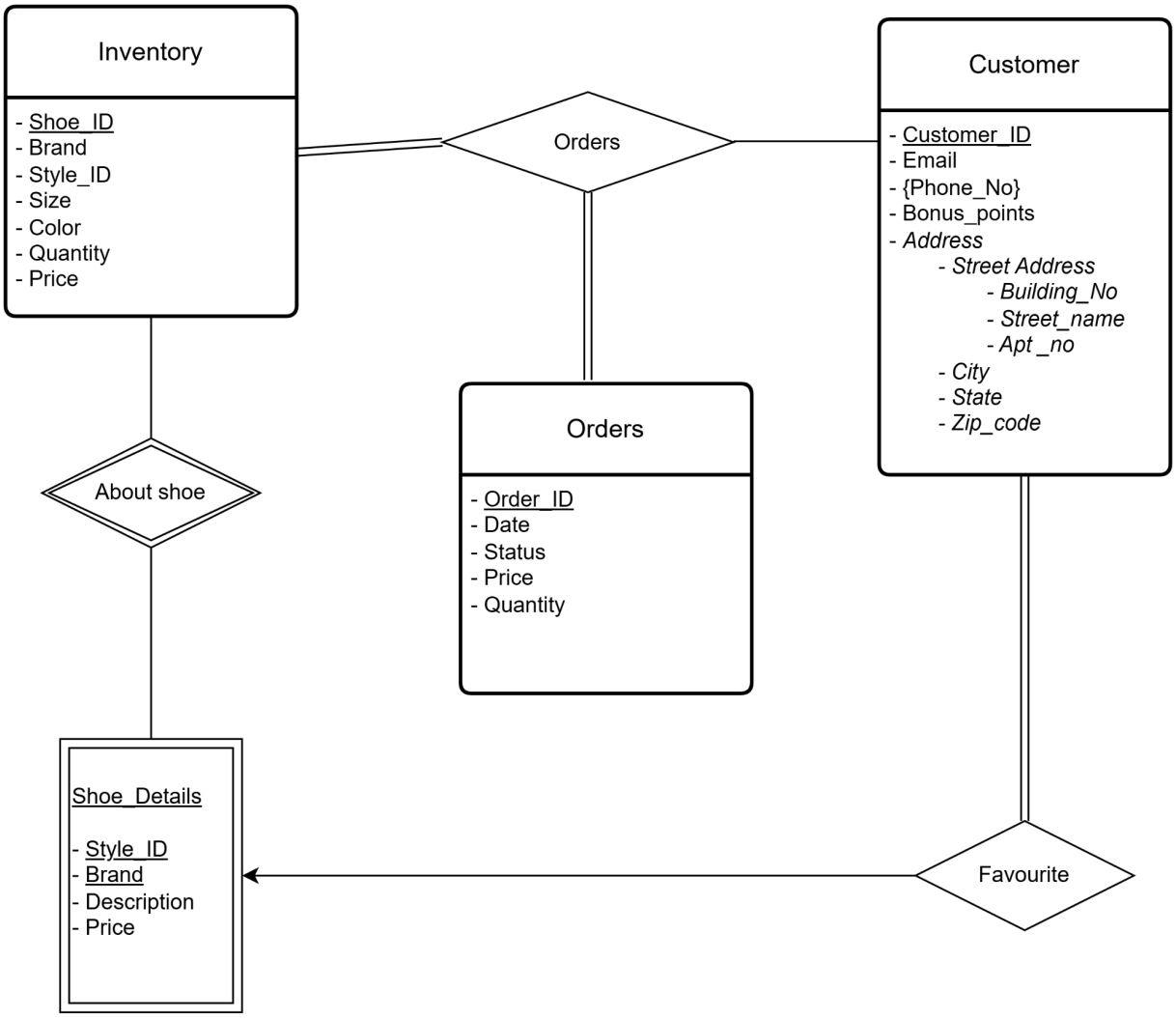



Fig 2.b

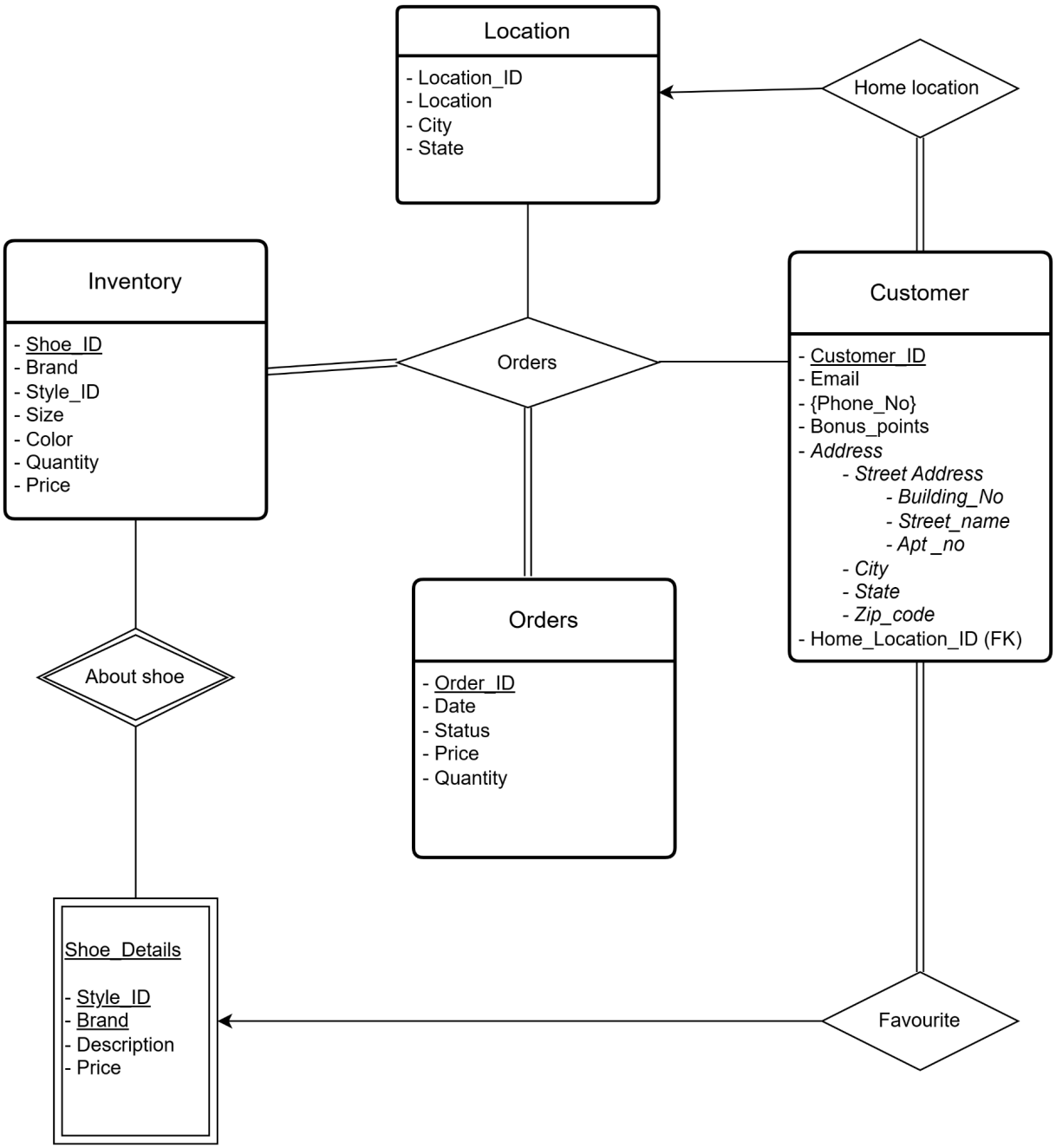
Answer 2.c

To modify this, we have created a new weak entity ‘Shoe_Details’ which is associated with Inventory through ‘About shoe’ relationship. A weak entity set is created to avoid redundancy in our ER diagram. Moreover, to allow the customer to have a favorite shoe, we have connected it to the weak entity set ‘Shoe_Details’ since we don’t need to store the size or color of the favorite pair.



Answer 2.d

To modify this, we have added a new entity called ‘Location’. Now all customers have at most one home location which is represented by the total participation and arrow notations. Apart from that, now each order will have a location. This will help in identifying the location of the inventory and store where the order is placed from.



**Problem 3**

a) Design an ER diagram to model this situation: A tennis league has hired you to design a database for their upcoming tournament. In part (a), you will only consider *singles events, where one individual player plays against another individual player.* There are a large number of players, each with a unique *playerID* and a few other attributes (your choice.) Each player is enrolled in one or more *Events* (for example women's singles, men's singles, juniors, etc. – you don't need to know the specific events to design the model.) Each *Event* has a unique *eventID,* a *description*, and some other attributes

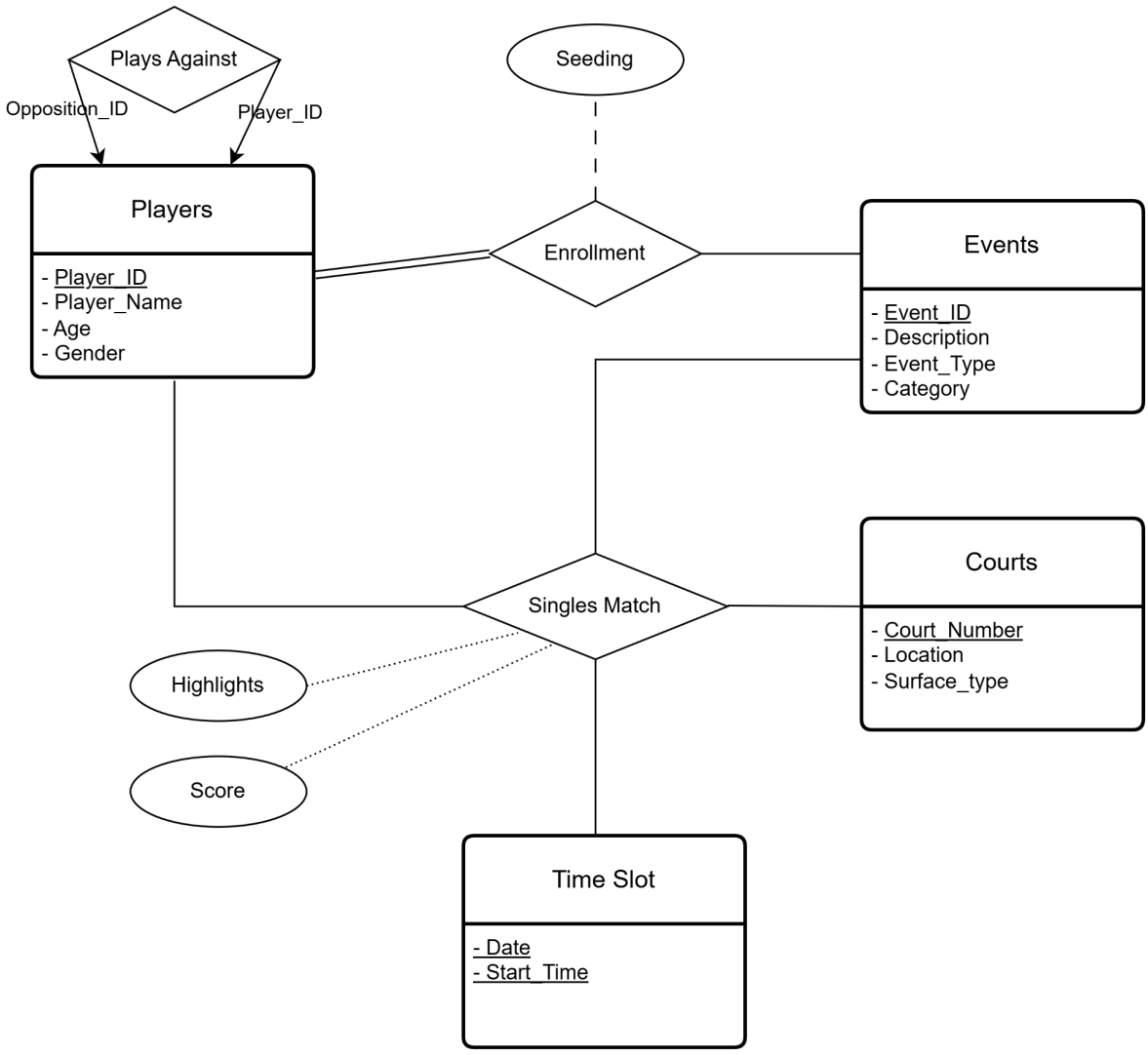
(again, your choice). Each player is enrolled in at least one event; some of the enrolled players are *seeded*, i.e. assigned a number indicating that they have a high ranking among all players in a given event. (Seeding is used in tournaments to arrange the schedule so the top players are unlikely to meet one another in early rounds.) The club has multiple *Courts*, each with a unique *courtNumber* and some other attributes. There are multiple *timeSlots* at which matches are played, each of which has a unique *date* and *startTime.*

A *SinglesMatch* takes place between two players enrolled in the same event in a particular time slot on a particular court; at the end the *score* is recorded along with a description of the *highlights.*

b) **Extra Credit:** Doubles anyone? The tournament also includes doubles events, in which two-player teams play one another (e.g. men's doubles, women's doubles, mixed doubles) . Augment your ER diagram to model doubles events and their matches. (You do not have to show all of the details for entity sets and relationship sets from part b) unless they're relevant here. Highlight the parts that are new.) There are several reasonable ways to approach this. In addition to designing the ER, comment on which aspects of doubles events and matches are or are not represented by your diagram.

Answer 3A:

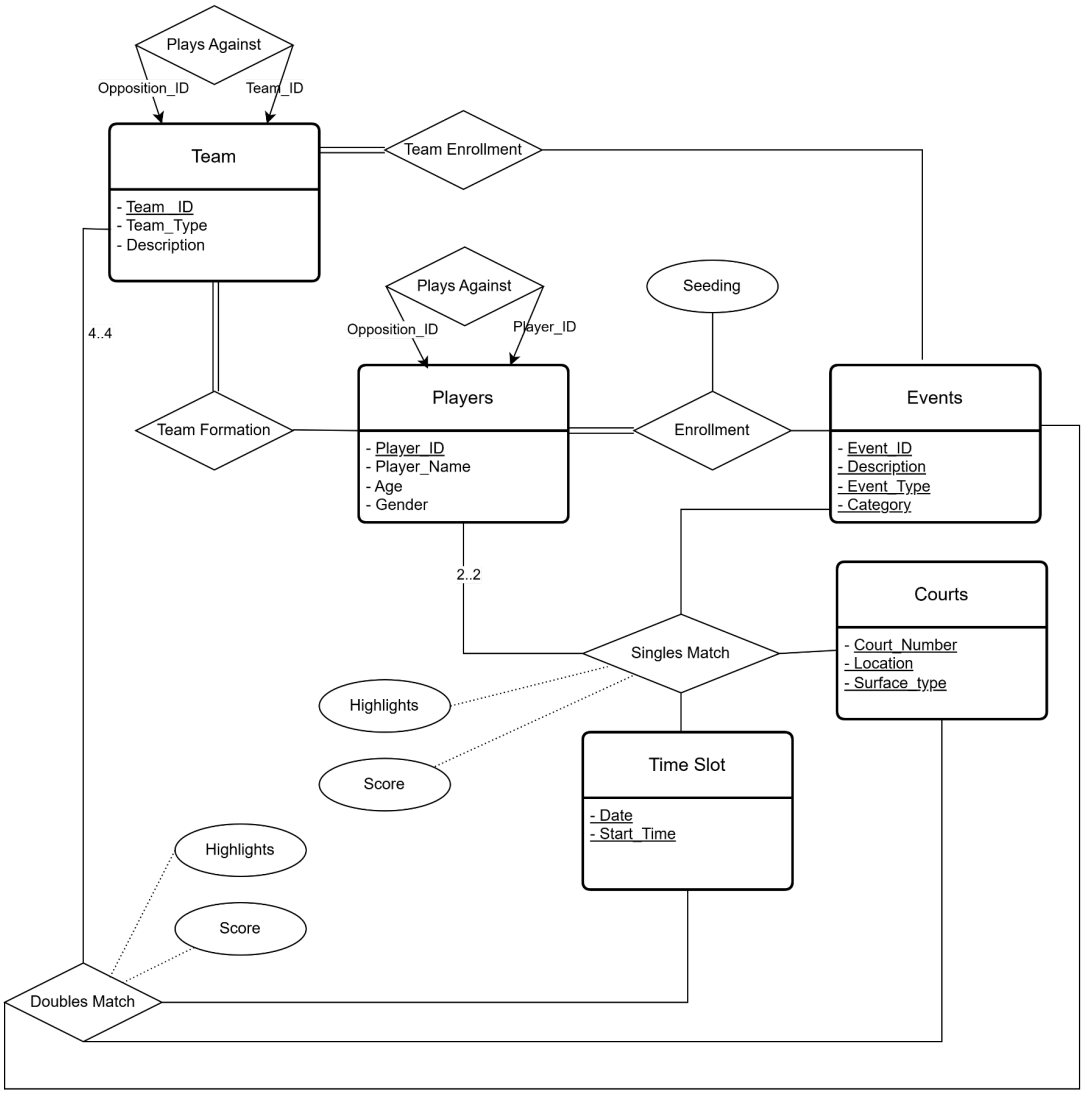
To solve this problem, we have used a self-referential relationship for players to decide which players play against each other. We have also used arrow notation in the self-referential relationship to represent that at most one player can play against at most one player. Then since all players are enrolled in at least one event, we have used the total participation notation between players and enrollment. Moreover, since seeding is an attribute of players and is used in events, we have used seeding as an attribute of the relationship enrollment. Lastly, we have used a quaternary relationship Singles Match, which is connected with Players, Courts, Events and TimeSlot. This is because Players play the Singles match of an Event on courts at a fixed time-slot.




Answer 3.b

To modify this, we have added a new relationship called Team Formation, and a new entity called Team. Through this relationship, players are allowed to make teams. Then after forming a team of two players, there is a self-referential relationship on the team entity, so that two teams can compete against each other. The arrow notation again signifies that at most one team can play against at most one team. Then the two teams can finally enroll in an event through Team Enrollment relationship. Apart from that, there is a new relationship called Doubles Matches for team events.

Another idea to solve this problem is to generate fixtures of teams, where we directly generate fixtures or matches of multiple teams who play against each other instead of first forming teams, then creating two teams who play each other.



Answer 4.a

1) Which of the following aspects of the data are represented in this model?

(a) Each district has exactly one representative.

Ans: **Yes**, Shown in model. “Rep_of” relationship has arrows towards both one to one double line representing total participation , i.e it has representatives

(b) Each state has exactly two senators

Ans: **No**, Not shown in model, No constraint of no of senator being 2

(c) Each state has at least one senator

Ans: **Yes.** Each state must participate in senator of relationship due to double line, which represents total participation. No constraint on person side so the person may or may not be senator and there may be more than 1 senator of same state.

(d) A person cannot be senator of two different states

Ans: **Yes.** Arrow or state side of “Senator of” relationship means relationship is one- only on state side, i.e the person can be the senator of only one state.

(e) A person cannot be congressperson of two different districts .

Ans: **Yes**. Arrow or district side of “Rep of” relationship indicates person can represent only one district

(f) A person can be the congressperson of two different districts, as long as they're in different states

Ans: **No.** Because arrow on person side of “Rep of” relationship, each person can participate at most once in “Rep of” relationship

(g) A person can be the congressperson of two different districts, as long as they have different numbers

Ans: **No.** Because of arrow for person of “Rep of” relationship, each person can participate at most once in  in “Rep of” relationship

Answer 4.b

Person( lname, fname, DOB, gender, url, email)

District (District_ID, num, population)

State(sname, nickname, flag, statePopulation() )

Repof( Relationship between Person and District)

lname foreign key references Person(lname, DOB)

District_ID foreign key references District(District_ID)

In (Relationship between District and State)

District_ID foreign key references District(District_ID)

sname foreign key references State(sname)

SenatorOf (Relationship between Person and State)

lname foreign key references Person(lname, DOB)

sname foreign key references State(sname)