HOMEWORK -1

MIS686 - Enterprise Database Management

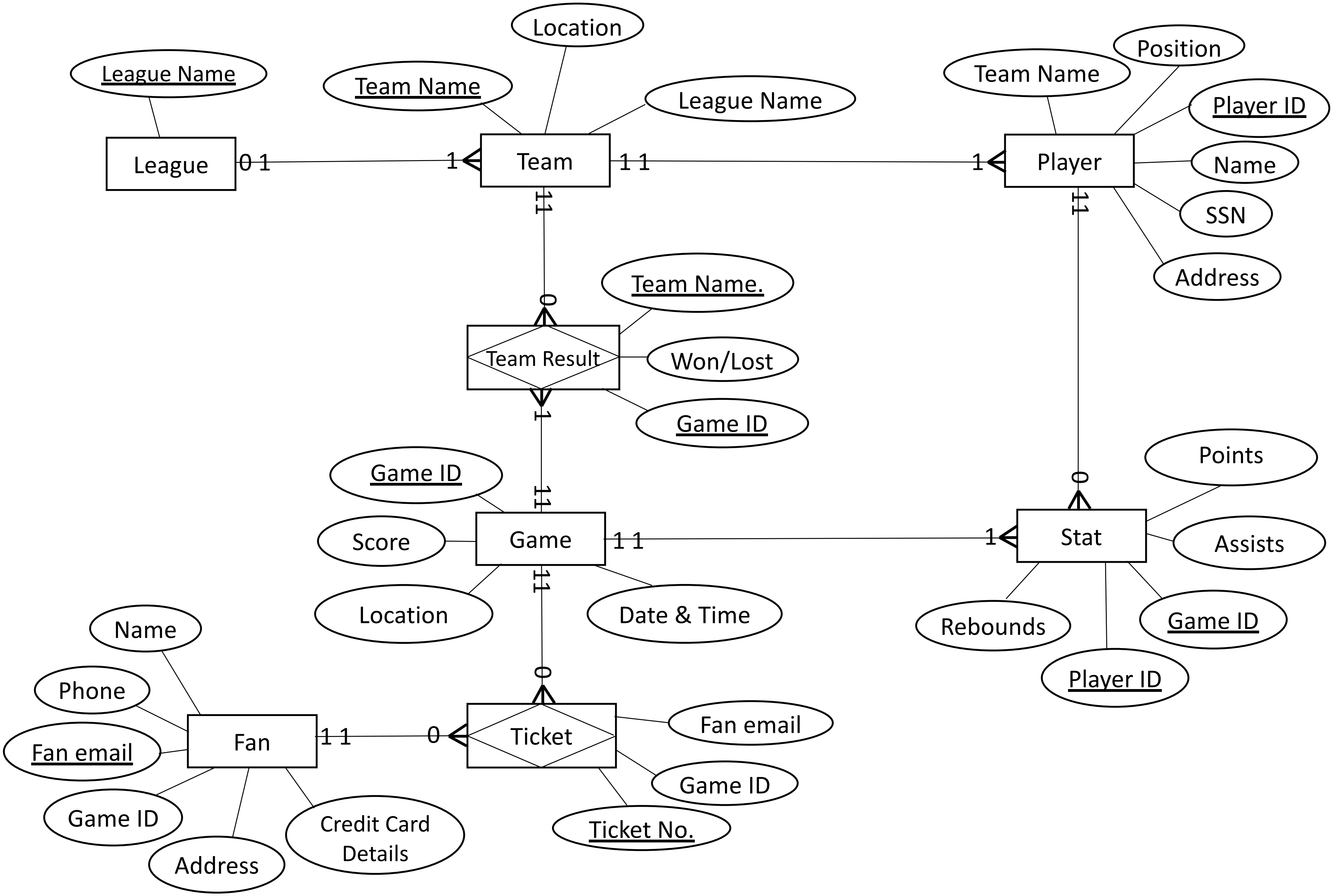
Name: Umadevi Betageri

Red ID: 827194401

Q.No 1. A basketball organization is interested in creating a database to organize its operations. The basketball organization runs multiple local basketball leagues, and the organization would like to store the name for each league. Each league is associated with basketball teams (a league cannot exist without some team in it), and each basketball team plays in only one league. Each basketball team has a name and a location. Basketball teams will hire players that play on the team. Each player has a name, a social security number, a home address, and a position, and each player may only play for a single team. Each basketball team will also participate in basketball games. Each game the teams play is associated with a date/time, a location, and a score. It is possible for multiple games to occur on the same date/time, and it is possible for multiple games to be played at the same location, but only one game can be played at a given combination of date/time and location. Players also play in each basketball game. For each basketball game that they play in, the organization would like to track player stats, which include points, assists, and rebounds. In addition, fans also attend each game. For each fan, a name, a home address, one phone number, one email address, and one credit card number are recorded.

Ans:

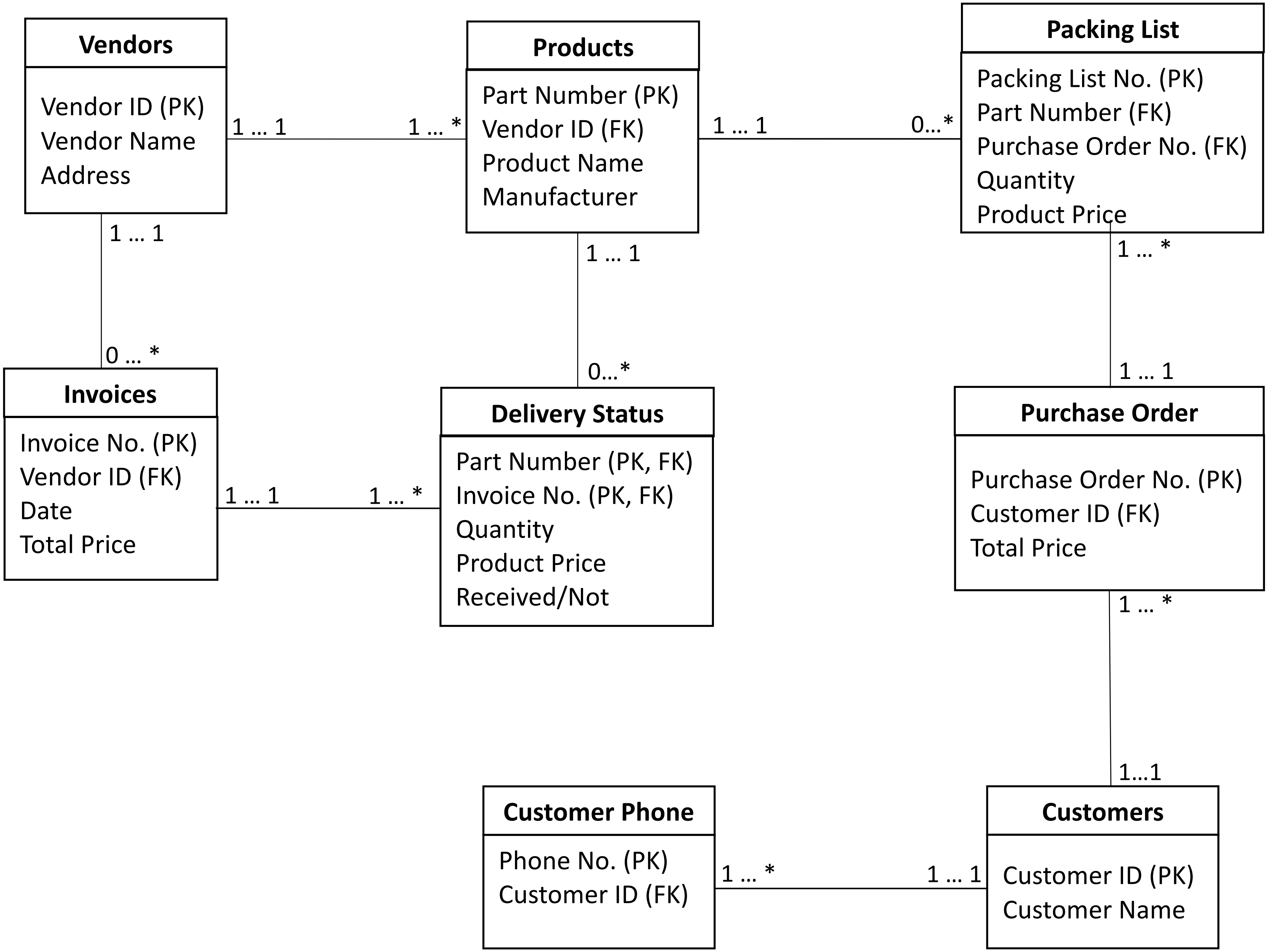
- **Entities: League, Team, Players, Game, Stat and Fans.**
- **Associative Entities: Team Result, Ticket.**
- **Diagram Type: ER Diagram**
- **Assumptions:**
  - For Game I have created a Game ID for each game for given combination of date/time and location as a primary key.
  - Since a game can have more than one team (usually two teams) and a team can play in many games, I have created an associative entity 'Team Result' between game and team. I have assumed that team result is associated with one game for and a team (team won or lost) and if two teams are playing in a game, game can have two team results.
  - For team I choose team name as a primary key considering team names are unique.
  - Since a fan can attend multiple games and each game can have multiple fans, I have created an associative entity as "Ticket" between fan and game with ticket number as primary key, assuming each ticket is associated with a one game and each ticket is only for one fan.
  - For each player I choose player ID as primary key, as names can be same, and I cannot use player SSN for security reason.
  - I have assumed that game may not sale ticket sometimes and a fan may not purchase ticket sometimes. Hence the minimum is set as 0.


League Name
League
Location
Team Name
League Name
Team
0 1
1
1 1
Team Name
Position
Player ID
Player
Name
SSN
Address
1
Team Name.
Team Result
Won/Lost
Game ID
Game ID
Score
Game
Location
Date & Time
Points
Assists
Stat
Rebounds
Game ID
Player ID
1 1
1
0
Name
Phone
Fan email
Fan
Game ID
Address
Credit Card Details
1 1
0
Ticket
Fan email
Game ID
Ticket No.

Q.No. 2 An auto parts store is interested in creating a databased to organize its inventory. The auto parts store sells a variety of products, each of which is sourced from a vendor. Each product has a name, a part number, and a manufacturer (the database does not need to store any other information about manufacturers). While the product name and/or manufacturer may not be unique, the part number is unique. Each vendor may supply multiple products. The database should also store the names and addresses of the vendors. Whenever a vendor ships a product to the store, they send an invoice. The invoice should list the product(s) being shipped, the number of each product shipped, the price of each product, an invoice number, which is unique, and a date. Finally, the store would also like to keep information on its customers. Each customer has a name and potentially several phone numbers. When a customer purchases a product, they do so by submitting a purchase order. The purchase order contains a purchase order number, which is unique, which products are purchased, how many of each product are purchased, and the price of each product being purchased.

Ans

- **Entities: Vendor, Product, Invoice, Customer, Purchase Order.**
- **Associative Entities: Delivery Status, Packing List, Customer Phone.**
- **Diagram Type: UML Diagram**
- **Assumptions:**
    - For each invoice no., there can be many products and for each product, there can be invoices. Hence, I have created an associative entity as “Delivery Status”, assuming that each delivery status is associated with an invoice and product.
    - Combination of invoice no. and part number is chosen as primary key for delivery status.
    - Similarly, for each purchase order, there can be many products and for each product, there can be purchase orders. Hence, I have created an associative entity as “Packing List”, assuming that each packing list is associated with an purchase order and product.
    - A packing list ID is generated for each combination of purchase order no. and part number.
    - Since for each customer there are several phone numbers, an additional entity is created for storing customer phone numbers.
    - I have assumed that the store, stores the vendor details when vendor supplied at least one product and similarly customer details are stored when a customer sends at least one purchase order.



Vendors

Vendor ID (PK)
Vendor Name
Address

Products

Part Number (PK)
Vendor ID (FK)
Product Name
Manufacturer

Packing List

Packing List No. (PK)
Part Number (FK)
Purchase Order No. (FK)
Quantity
Product Price

Invoices

Invoice No. (PK)
Vendor ID (FK)
Date
Total Price

Delivery Status

Part Number (PK, FK)
Invoice No. (PK, FK)
Quantity
Product Price
Received/Not

Purchase Order

Purchase Order No. (PK)
Customer ID (FK)
Total Price

Customer Phone

Phone No. (PK)
Customer ID (FK)

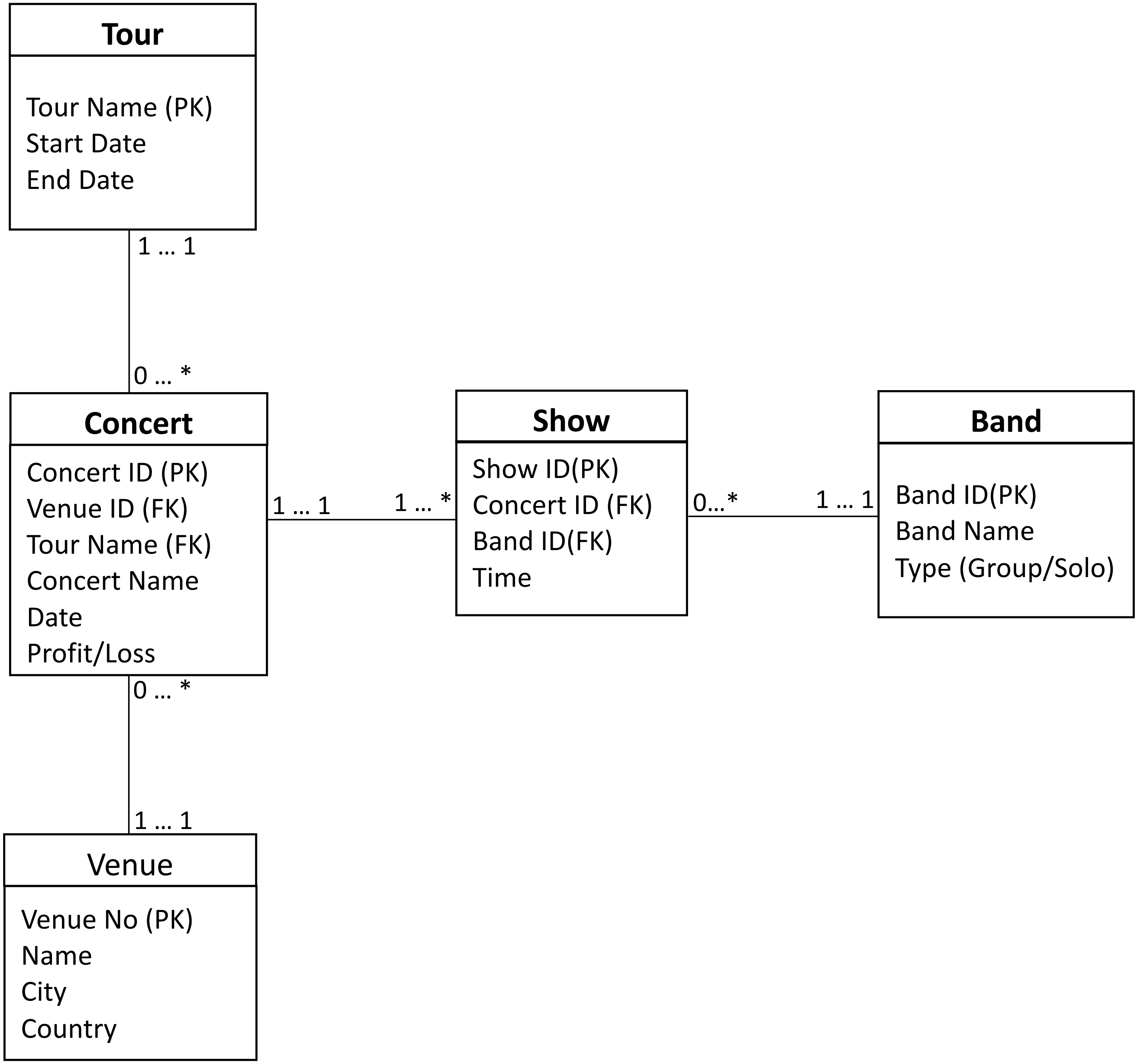
Customers

Customer ID (PK)
Customer Name

Q.No. 3 A concert promoter is interested in creating a database to organize its tours and concerts. The promoter works for several different tours, which each have a name, which is unique, a start date, and an end date. Each tour will ultimately consist of several different concerts, but it is possible that the tour will be entered into the database before any specific concerts are scheduled. Each concert will be part of exactly one tour. Each concert will have a name, date, and profit (or loss). Each concert will take place at a venue, which has a name, a city, and a country. These three pieces of data are unique in combination, but they are not unique on their own. Each concert will take place at exactly one venue, but a venue may be stored in the database before it has held any concerts. Finally, bands will be hired to play at the concerts. It is possible for a band to play at multiple different concerts, and it is possible for multiple bands to play at the same concert. For each band, the promoter would like to store the band name and whether the band is a solo act or a group.

Ans:

- **Entities: Tour, Concert, Venue, Bands.**
- **Associative Entity: Show**
- **Diagram Type: UML Class Diagram**
- **Assumptions:**
  - For the given condition, since a band can play in multiple concerts and a concert can have more than one band, I have created a associative entity as "Show", assuming that a show is associated with one band and a concert.


Tour
Tour Name (PK)
Start Date
End Date
1 … 1
0 … *
Concert
Concert ID (PK)
Venue ID (FK)
Tour Name (FK)
Concert Name
Date
Profit/Loss
1 … 1
1 … *
Show
Show ID(PK)
Concert ID (FK)
Band ID(FK)
Time
0…*
1 … 1
Band
Band ID(PK)
Band Name
Type (Group/Solo)
0 … *
1 … 1
Venue
Venue No (PK)
Name
City
Country