

Step 1: business story

To start the design process, at first, we need to create a business story.

Introduction: In 2024, the Olympic games are going to be held in the city of Paris. Various athletes from different countries will come to Paris to participate in the coming event. Different types of sports events will be organized and it's very prestigious for any athletes to represent their country. So, Paris will be very busy during this event as there will be many people as a player, as a coach and as a spectator. A committee is formed to maintain everything including accommodation, transportation, for the athletes and coaches, equipment for every sports event. It's also important to keep track of all the schedule of the events, the committee is also focused to maintain the schedule without any error. The committee hired many volunteers to facilitate the whole event as well.

step 2: Database Design

2.1 Identification of entities: Now we can identify the entities from the story, and we need to look for the nouns which can be the entity.

After screening the story, I have decided to keep the following for my entities:

Olympic Games

Athletes

Country

Event

Sports

Committee

Accommodation

Transportation

Volunteers

2.2 : Identification of relationships: Athlete participates in sports.

Athlete represents country.

Athlete, sports and country are part of Olympic Games

Olympic Games organize Event.

Olympic Games manage committee.

Committee organizes accommodation.

Committee organizes transportation.

2.3 Details of relationships: Sport - Participates - Athlete: A sport may have many athletes, and an athlete can participate in many sports, forming a many-to-many relationship.

Country - Represents - Athlete: A country can be represented by many athletes, while an athlete may represent only one country, establishing a one-to-many relationship.

Olympic Games - Event: An Olympic Games event can organize multiple individual events, implying a one-to-many relationship.

Olympic Games - Manages - Committee: Each Olympic Games is managed by one committee, and each committee manages only one Olympic Games, forming a one-to-one relationship.

Event - Organize - Accommodation: An event may organize multiple accommodations, but each accommodation is typically organized for a specific event, demonstrating a one-to-many relationship.

Event - Organize - Transportation: Similarly, an event can organize multiple transportation options, while each transportation option is usually organized for a specific event, indicating a one-to-many relationship.

Committee - Organizes - Accommodation: A committee may organize multiple accommodations, but each accommodation is typically organized by one committee, establishing a one-to-many relationship.

Committee - Organizes - Transportation: Similarly, a committee can organize multiple transportation options, while each transportation option is typically organized by one committee, indicating a one-to-many relationship.

Sport - Participates - Athlete: Reiterating the first relationship, a sport may have many athletes, and an athlete can participate in many sports, forming a many-to-many relationship.

Volunteer - Committee: Each volunteer may be associated with only one committee, while a committee may have multiple volunteers, demonstrating a many-to-one relationship.

2.4 Integrities of entities:

Olympic Games: Game_ID, End Date, Name, Host city, Start Date

Athlete : Athlete_ID, First_Name, Last_Name

Country: Name, Country_ID

Event: Event_ID, End Date, Start Date, Location, Host city.

Sports: Name, Sports_ID, Athlete_ID

Committee: Committee_ID, Chairperson, Name

Accommodation: Hotel_ID, Name, Address

Transportation: Trasnpoartation_ID, Type, Description

Volunteers: Volunteers_ID, First_Name, Last_Name, Sports_Assign

2.5 Normalisation:

Olympic Games 2NF to 3NF

<u>Game_ID</u>	<u>Name</u>	Start Date	End Date	<u>Host city</u>

Name	Host city
------	-----------

Athlete – 3NF

Country– 3NF

Event– 3NF

Sports– 3NF

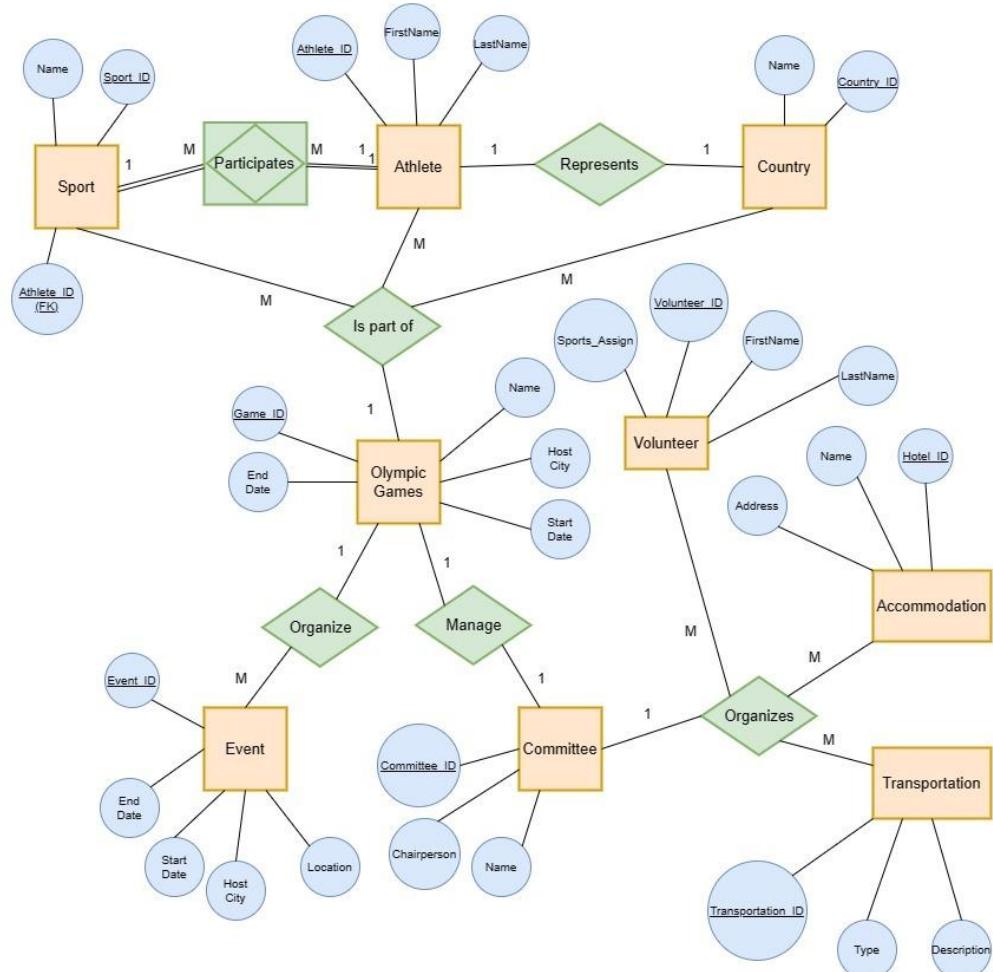
Committee– 3NF

Accommodation– 3NF

Transportation– 3NF

Volunteers– 3NF

2.6 ERD



For part 3.1 and 3.2 , I have taken the entities and made table for each of them using the primary keys and foreign keys in the online SQL editor website and I have put the screenshots for the reference and I have also made their names according to their entity names.

Analysis:

To have some views, I have made some SQL queries, with that I wanted to look for some insights, below I am explaining them.

First one, I looked at the database to find out which countries have the most athlete participation. I used the following SQL quarries to find out it.

```
SELECT Country.Name, COUNT(*) AS
Athlete_Count
FROM Country
JOIN Represents ON Country.Country_ID =
Represents.Country_ID
GROUP BY Country.Name
ORDER BY Athlete_Count DESC;
```

The output:

The screenshot shows the Programiz Online SQL Editor interface. On the left, there are four expandable sections: 'Accommodation [-]', 'Athlete [-]', 'Committee [-]', and 'Country [-]'. The 'Country' section is expanded, showing columns: Country_ID [int], Name [varchar(255)], and Sports_ID [int]. The 'Athlete' section is also expanded, showing columns: Athlete_ID [int], FirstName [varchar(255)], and LastName [varchar(255)]. The 'Accommodation' section and 'Committee' section are collapsed. In the center, the 'Input' tab displays the SQL query:

```
SELECT Country.Name, COUNT(*) AS Athlete_Count
FROM Country
JOIN Represents ON Country.Country_ID = Represents.Country_ID
GROUP BY Country.Name
ORDER BY Athlete_Count DESC;
```

Below the input, the 'Output' tab shows the results of the query:

Name	Athlete_Count
USA	3
Germany	1
China	1

On the right, there are four tables listed under 'Available Tables':

- Accommodation**: Shows three rows with Hotel_ID, Name, and Address.
- Athlete**: Shows five rows with Athlete_ID, FirstName, and LastName.
- Committee**: Shows one row with Committee_ID, Name, Chairperson, and Sports_ID.
- Country**: Shows three rows with Country_ID and Name.

We can see that; USA has the most athlete participation in this event.

Then I looked for the sports which have most athlete participation in this event. I used the following SQL queries to find out this.

```

SELECT Sport.Name, COUNT(*) AS Athlete_Count
FROM Sport
JOIN Participates ON Sport.Sports_ID =
Participates.Sports_ID
GROUP BY Sport.Name
ORDER BY Athlete_Count DESC;

```

The output:

The screenshot shows the Programiz Online SQL Editor interface. On the left, there's a sidebar with tables: Accommodation, Athlete, Committee, and Country. The main area has an 'Input' tab containing the SQL query, an 'Output' tab showing the results, and an 'Available Tables' section on the right displaying the four tables with their columns and sample data.

Accommodation	Hotel_ID [int]	Name [varchar(255)]	Address [varchar(255)]
1	Olympic Hotel	123 Main Street, Paris	
2	Winter Lodge	456 Snowy Avenue, Monaco	
3	Summer Resort	789 Sunny Road, Paris	

Athlete	Athlete_ID [int]	FirstName [varchar(255)]	LastName [varchar(255)]
1	John	Doe	
2	Jane	Smith	
3	Michael	Johnson	
4	Emily	Davis	
5	Robert	Brown	

Committee	Committee_ID [int]	Name [varchar(255)]	Chairperson [varchar(255)]	Sports_ID [int]
		Organizing		

Name	Athlete_Count
Football	2
Basketball	2
Swimming	1

We can see that football and basketball have most participants.

After that, I looked for which games have the most volunteers, and I used the following queries to find out that:

```

SELECT Sport.Name, COUNT(*) AS
Volunteer_Count
FROM Sport
JOIN Committee ON Sport.Sports_ID =
Committee.Sports_ID
JOIN Volunteer ON Committee.Committee_ID =
Volunteer.Committee_ID
GROUP BY Sport.Name
ORDER BY Volunteer_Count DESC;

```

The output:

The screenshot shows the Programiz Online SQL Editor interface. On the left, there is a sidebar with a tree view of available tables: Accommodation, Athlete, Committee, and Country. The main area has tabs for 'Input' and 'Output'. The 'Input' tab contains the following SQL query:

```
SELECT Sport.Name, COUNT(*) AS Volunteer_Count
FROM Sport
JOIN Committee ON Sport.Sports_ID = Committee.Sports_ID
JOIN Volunteer ON Committee.Committee_ID = Volunteer.Committee_ID
GROUP BY Sport.Name
ORDER BY Volunteer_Count DESC;
```

The 'Output' tab displays the results of the query:

Name	Volunteer_Count
Swimming	1
Football	1
Basketball	1

On the right side of the interface, there are three tables shown as data grids: 'Accommodation', 'Athlete', and 'Committee'. The 'Accommodation' table has three rows. The 'Athlete' table has five rows. The 'Committee' table has one row.

We can see that, swimming, football, basketball they all have equal volunteer.

Then, I looked for the most transportation method used by the athletes. I used the following queries to find out it.

```
SELECT Transportation.Type, COUNT(*) AS
Athlete_Count
FROM Transportation
JOIN Organizes ON
Transportation.Transportation_ID =
Organizes.Transportation_ID
JOIN Event ON Organizes.Event_ID =
Event.Event_ID
GROUP BY Transportation.Type
ORDER BY Athlete_Count DESC;
```

The output:

The screenshot shows the Programiz Online SQL Editor interface. On the left, there are four tables listed with their columns: Accommodation, Athlete, Committee, and Country. The Accommodation table has columns Hotel_ID, Name, and Address. The Athlete table has columns Athlete_ID, FirstName, and LastName. The Committee table has columns Committee_ID, Name, Chairperson, and Sports_ID. The Country table has columns Country_ID and Name. In the center, the 'Input' section contains the following SQL query:

```
SELECT Transportation.Type, COUNT(*) AS Athlete_Count
FROM Transportation
JOIN Organizes ON Transportation.Transportation_ID = Organizes.Transportation_ID
JOIN Event ON Organizes.Event_ID = Event.Event_ID
GROUP BY Transportation.Type
ORDER BY Athlete_Count DESC;
```

The 'Output' section displays the results of the query:

Type	Athlete_Count
Shuttle Bus	1
Metro	1
Charter Bus	1

On the right, the 'Available Tables' section shows the four tables again. Below the tables, there are three preview sections: 'Accommodation', 'Athlete', and 'Committee', each displaying a small subset of the data.

We can see that; athletes are using all the transportation available to them equally.

Then I looked for the most used accommodations by the athletes, and I used the following queries to find out it.

```
SELECT Accommodation.Name, COUNT(*) AS Athlete_Count
FROM Accommodation
JOIN Organizes ON Accommodation.Hotel_ID = Organizes.Hotel_ID
JOIN Event ON Organizes.Event_ID = Event.Event_ID
GROUP BY Accommodation.Name
ORDER BY Athlete_Count DESC;
```

The output:

The screenshot shows the Programiz Online SQL Editor interface. On the left, there are four tables listed with their columns: Accommodation, Athlete, Committee, and Country. The central area has an 'Input' section containing the following SQL query:

```
SELECT Accommodation.Name, COUNT(*) AS Athlete_Count
FROM Accommodation
JOIN Organizes ON Accommodation.Hotel_ID = Organizes.Hotel_ID
JOIN Event ON Organizes.Event_ID = Event.Event_ID
GROUP BY Accommodation.Name
ORDER BY Athlete_Count DESC;
```

Below the input is an 'Output' table:

Name	Athlete_Count
Winter Lodge	1
Summer Resort	1
Olympic Hotel	1

On the right, there are three tables displayed as results:

- Accommodation**: Shows three rows with Hotel_ID 1 (Olympic Hotel), Hotel_ID 2 (Winter Lodge), and Hotel_ID 3 (Summer Resort) at their respective addresses.
- Athlete**: Shows five rows of athletes with their Athlete_ID, FirstName, and LastName.
- Committee**: Shows one row for the organizing committee with Committee_ID 1, Name 'Organizing', Chairperson null, and Sports_ID null.

We can see that the three hotels are used by the athletes.

I also looked for the location which has the most events happening using the following SQL queries.

```
SELECT Event.Host_City, COUNT(*) AS Event_Count
FROM Event
GROUP BY Event.Host_City
ORDER BY Event_Count DESC;
```

The output:

The screenshot shows the Programiz Online SQL Editor interface. On the left, there are four tables listed with their columns: Accommodation, Athlete, Committee, and Country. The central area has an 'Input' section containing the following SQL query:

```
SELECT Event.Host_City, COUNT(*) AS Event_Count
FROM Event
GROUP BY Event.Host_City
ORDER BY Event_Count DESC;
```

Below the input is an 'Output' table:

Host_City	Event_Count
Paris	3

On the right, there are three tables displayed as results:

- Accommodation**: Shows three rows with Hotel_ID 1 (Olympic Hotel), Hotel_ID 2 (Winter Lodge), and Hotel_ID 3 (Summer Resort) at their respective addresses.
- Athlete**: Shows five rows of athletes with their Athlete_ID, FirstName, and LastName.
- Committee**: Shows one row for the organizing committee with Committee_ID 1, Name 'Organizing', Chairperson null, and Sports_ID null.

Paris is hosting most events we can see from the above output.

Then, I looked for which sports have committee members allocated to them and I used the following queries to find it out.

```
SELECT c.Name AS Committee_Name,  
c.Chiperson, s.Name AS Sport_Name  
FROM Committee c  
JOIN Sport s ON c.Sports_ID = s.Sports_ID;
```

The output:

The screenshot shows an online SQL editor interface. On the left, there are several tables listed under categories: 'Participates [-]', 'Represents [-]', 'Sport [-]', 'Transportation [-]', and 'Volunteer [-]'. The 'Sport [-]' category lists 'Sports_ID [int]' and 'Name [varchar(255)]'. The 'Transportation [-]' category lists 'Transportation_ID [int]', 'Type [varchar(255)]', and 'Description [varchar(255)]'. The 'Volunteer [-]' category lists 'Volunteer_ID [int]', 'FirstNames [varchar(255)]', 'LastNames [varchar(255)]', 'Sports_Acces [varchar(255)]', and 'Com [varchar(255)]'. In the center, the 'Input' section contains the SQL query: `SELECT c.Name AS Committee_Name, c.Chiperson, s.Name AS Sport_Name
FROM Committee c
JOIN Sport s ON c.Sports_ID = s.Sports_ID;`. Below the input is the 'Output' section, which displays the results of the query. The results are presented in a table with columns: 'Committee_Name', 'Chairperson', and 'Sport_Name'. The data shows three rows: 'Organizing Committee for the Olympic Games' with 'Chairperson' 'John Johnson' and 'Sport_Name' 'Football'; 'International Olympic Committee' with 'Chairperson' 'Alice Anderson' and 'Sport_Name' 'Basketball'; and 'National Olympic Committee' with 'Chairperson' 'Michael Smith' and 'Sport_Name' 'Swimming'. To the right of the output, there are two tables: 'Sport' and 'Transportation'. The 'Sport' table has columns 'Sports_ID' and 'Name', with data for rows 1 through 5: Football, Basketball, Swimming, Tennis, and Gymnastics. The 'Transportation' table has columns 'Transportation_ID', 'Type', and 'Description', with data for rows 1 through 3: Shuttle Bus (Airport Shuttle Service), Metro (Public Transportation System), and Charter Bus (Private Bus Service). A button labeled 'Run SQL' is located at the top of the input area.

Sports_ID	Name
1	Football
2	Basketball
3	Swimming
4	Tennis
5	Gymnastics

Transportation_ID	Type	Description
1	Shuttle Bus	Airport Shuttle Service
2	Metro	Public Transportation System
3	Charter Bus	Private Bus Service

We can see that out of five, three sports have committee members allocated to them, tennis and gymnastics don't have committee members allocated to them at this moment.

Recommendation:

As from the SQL query, we can see that, USA have the most athletes participating in this event, so it will be better to allocate those athletes together for accommodation and transportation, it will be more efficient and cost effective as the people from same country always try to stay together when they are outside of their own country for an event like this.

From the second SQL query, we can see that football and basketball are the most participated sport by the athletes. That's why the committee should plan and put staff and volunteers for these two sports, it can be said that many spectators will also come to see these two sports as well. The committee can also put more charge in this sports ticket price to gain some more profit and, they can earn money from the sponsors for these two sports as these two will be most viewed sports based on the assumption that most of the athletes are participating in these two games.

From the third SQL query, we can see that, three volunteers are assigned for three sports swimming, basketball, and football on equal level, but the participants number isn't equal, that's why we need to put volunteers where the number of participants is higher like basketball and football game. Again, these two games are team games, that means many players will be taking part in the game at the same time, so without more volunteers, these two sports won't be able to run smoothly, and it will affect the reputation of the organizing committee.

From the fourth SQL query, we can see athletes are using all three type of transportation available to them, they are using those equally, so instead of having two types of bus transportation, the committee should run one bus transportation, this will help to minimize the cost and they should include some flights available for the team sports like basketball and football, where it will open up the space in the bus transportation for individual sport like swimming's and tennis's athlete. It will save cost and time both.

From the fifth SQL query, we can see which hotels the athletes are living. We can see that, they are using the three hotels equally, here, the committee should keep the athlete from the same nation together, we can also see that, these three hotels are the most used accommodation by the athletes, so committee can put more transportation available near these hotels for the athletes and they can also cut cost in other hotels where the number of the athletes who are living there are not that high.

From the sixth SQL query, we can see that Paris is the location where most of the events are happening. From this query, the committee can make the arrangements, as most of the events are happening here, it will be a busy area, the committee have to make the transportation arrangements earlier than normal time in order to avoid any delays, the spectators amount would be also higher than the other area as well, so the committee also have to make arrangements for their spectators as well, and as the committee can predict earlier from the from the SQL query as here most of the events are going to held, that's why they can make their move long before which will give them enough time to rely on any last minutes action and eventually that will be more time and cost effective.

From the last SQL query, we can see which committee is allocated to which sports. We can see that three sports have their allocated committee at this moment, the rest two are still waiting for their committee, so it's important to give them their allocated committee so that they can run their activities without any obstacle. Here they can create a joint committee for Tennis and Gymnastic as the participants numbers are not that much higher in those two games compare to football and basketball. So, if they create a joint committee for Tennis and Gymnastics, it will save time and money, both at the same time fulfilling the purposes as well.

Conclusion:

Finally, the creation of the "Olympic_Games_2024" database marks a significant milestone in the planning process for the upcoming Paris 2024 Olympic Games. The primary goal of this database is to provide a solid platform for data-driven decision-making, with a specific emphasis on cost savings associated with transportation, accommodations, and committee management. Furthermore, the database is useful in leveraging user data to inform strategic decisions. It is apparent that the database, when combined with these queries, is a powerful tool for optimizing the Olympic Games system and ensuring the event's resounding success.