1. A summary of the data
   * Why was it chosen?
   * What does it consist of?
   * How large is it? (File size, number of records)
2. A discussion of the data model
   * Why was it broken it down into those tables?
   * Did students face any difficult choices when deciding on how to set up the model?

* Tricky participation/cardinality ratio decisions could go here

1. A summary of the database
   * List each of the final tables, along with its cardinality and arity
2. A list of the queries implemented in Part 2 (and thus available through the interface in Part 3)
   * Students do not need to write out the SQL code, just a text explanation of what the query returns
3. A summary of the interface. This should include both a description of how it was created (what language was it written in, including any specific libraries) and instructions on how to use it.

Student: Ngoc Pham

Student Number: 7891063

Fall 2021

COMP 3380

Professor: Adam Pazdor

1. **A summary of the data**

The database (***Pham\_Ngoc\_athlete\_events.xlsx***) includes the Olympic events from 1986 to 2016. It was chosen due to the potential of breaking down to small tables to reduce the redundancy of data, and ability to implement the functional dependency and normalization theories that students learned from the course. The original data is in excel format, size is 21,783 KB which comprises of two main tables:

|  |  |  |
| --- | --- | --- |
| **Table** | **Total Records** | **Total Columns (fields)** |
| athlete\_events | 271,117 | 15:  • ID  • Name  • Sex  • Age  • Height  • Weight  • Team  • NOC  • Games  • Year  • Season  • City  • Sport  • Event  • Medal |
| country\_definitions | 231 | 3: NOC, country, notes |

1. **A discussion of the data model**

From the first look, the “athlete\_events” table can be broken down into main five entity tables:

* athlete (**aID**, Name, Sex, Weight, Height) & athlete’s age
* country (**NOC**, Country, Teams)
* Olympic-game (**gID**, Year, Season, Host City)
* event (**eID**, Sport, Event Name)
* medal (**mID**, type)

However, I needed to find a way to connect these tables together. They seemed to be individual tables with its own ID at the moment. I started to draw the ER model and connected entity using relationships:

* **compete** (**eID, aID**): specific athletes competed in the specific sport events
* **playFor** (**aID**, NOC, Team): a specific athlete represents for his/her own country.
* **attend** (**gID, NOC**): specific countries attended the specific Olympics
* **has** (gID, **eID**): specific Olympics had specific sport events
* **win** (**aID, gID, eID**, mID): specific athletes won specific sport events with specific medals at specific Olympic
* **participate** (**aID, gID**, age): specific athletes participated in the specific Olympics at their specific ages

I realized three main entities “athlete”, “Olympic-game”, and “event” are interchangeably related to each other via relationships:

* athlete vs Olympic-game: participate
* athlete vs event: compete
* Olympic-game vs event: has
* Then I needed to assign medal to specific event that the athlete achieved in the specific Olympic: win

The “win” table and “compete” table give the relationship of “athlete” and “event”. But in reality, an athlete is trained to compete in their major (their specialty) in many different years, therefore, duplicates of “compete” will appear in “win” table, distinguished by the gID which is when the medal achievement takes place. “compete” table will have smaller records than “win” table, it’s a subset of “win”. See further below for reason why I kept both tables.

I also needed to break “country (**NOC**, Country, Teams)” to smaller tables, due to multivalued attribute Teams. It will become country(**NOC**, country), and team-list(**tID**, NOC, Team).

playFor(**aID**, NOC, Team) can be merged to athlete (**aID**, Name, Sex, Weight, Height) table using tID because “NOC, Team” are both in team-list table, and both have same primary key.

The rest does not seem like to be able to merge, due to composite key of its table.

I encountered some difficulties while deciding how to break down tables.

* At first, I did not need the “compete” table, because I thought I could derive it from the “win” table due to functional dependency aID, gID, eID→mID holds. However, I realized an athlete can compete in the same sport event in next years. Therefore, primary key of “compete” is different from “win”. This is how I decided to have the “compete” table as well in my database.
* I also came to realized that age cannot be combine in “athlete” table, because athlete’s ages are different at different years that they participated the Olympic-games. Therefore, I assigned age to “participated” table instead. This came to conclusion thanks to professor’s suggestion.
* At first glance, I thought Team attribute was the country name for NOC due to the nature of the content of Team. But I recognized that it’s team names and it just happened to be the same as country name, there are more other team names in one country. A country can have multiple teams playing for it. Hence, “team-list” table is made.

1. **A summary of the database**

The 11 final tables (post-merged, post-normalization) are: (bold & dark red are relationship tables)

1. athlete (**aID**, Name, Sex, Weight, Height, tID)
2. country (**NOC**, Country)
3. team-list (**tID**, NOC, Team)
4. **athlete-participation** (**aID, gID**, age)
5. Olympic Game (**gID**, Year, Season, Host City)
6. **attend** (**gID, NOC**)
7. **has** (**gID, eID**)
8. event (**eID**, Sport, Event Name)
9. **compete** (**eID, aID**)
10. medal (**mID**, type)
11. **win** (**aID, gID, eID,** mID)

Where:

* **athlete-participation** (**aID, gID**, age)
  + 1-M: an athlete can participate in many Olympic games (this year, next year, etc.)
  + All participation of athlete & Olympic Game because Olympic has to have athletes to be a game of that year, all athletes are trained to participate in the game.
* **attend** (**gID, NOC**)
  + M-N: Many countries attend to many Olympic Games (this year, or next years, etc.)
  + Olympic Game has all participation because all Olympic Games occur with many countries attending.
  + Country has partial participation because not all countries in the world participate in the Olympic.
* **has** (**gID, eID**)
  + 1-M: an Olympic Game holds many sport events in the period.
  + Olympic Game, Event have all participation cardinality because all Olympic Games have to hold all pre-determined sport events.
* **compete** (**eID, aID**)
  + M-N: Many athletes compete in many sport events. For example, a swimming athlete can participate in solo swimming event, or team swimming event; and in a solo/individual swimming event, many swimmers compete to each other.
  + Athlete & Event have all participation cardinality. All athletes at the Olympic are trained and participate in at least an event, all events at the Olympic have athletes compete for their countries.
* **win** (**aID, gID, eID,** mID)
  + M-N: all athletes at least win a medal in a sport event, if they don't win an event that they compete, will be denoted as NA; if they win many medals, it will be from many different sport events (at the specific Olympic Game).
  + Athlete & Event have all participation cardinality. Any (all) sport event always has many ranking gold, silver, bronze. All athletes participate the Olympic with the winning spirit, win at least something (even "NA" medal)

The relationship “playFor” was determined during the process of making EER diagram. This relationship has been merged to “athlete”

* playFor (aID, NOC, Team)
  + 1-1: an athlete represents for his/her own country. An athlete cannot represent for many countries. An athlete can only join one team of his/her country.
  + Athlete has all participation because any athlete in the Olympic is from his/her own country.
  + Country has partial participation because not all countries in the world participate in the Olympic.

I included a ***schema\_data*** folder with excel files and .sql files:

* the source data: ##athlete\_events.csv
* 11 tables in .csv
* create\_tables.sql showing queries how I developed my tables in Microsoft SSMS
* queries.sql showing queries how I developed my 14 queries in Microsoft SSMS. See Section 4.

User can find data of these 11 tables and source tables, along with relationship model in the ***Pham\_Ngoc\_the\_report\_interface.pbix*** as well

Diagram

Description automatically generated

1. **A list of the queries implemented**

Notes: when talking about “won” medals, always means excepting “NA” medals

1. A query returns how many *total countries* “attended” a specific Olympic-game.
2. A query returns how many *total events* a specific Olympic-game “had”.
3. A query returns how many “participated” *athletes* from a specific country in a specific Olympic-game.
4. A query returns how many participated *athletes* from a specific country “won” real medals (except “NA” medal”) in specific Olympic-game.
5. A query returns how many sport *events* the specific country “won” real medals (except “NA” medal) in a specific Olympic-game.
6. A query returns top 5 sport *events* the specific country “won” most real medals (except “NA” medal) in a specific Olympic-game.
7. A query returns count of gold, silver, bronze medals the specific country “won” (except “NA” medal) in a specific Olympic-game.
8. A query returns a total of gold, silver, bronze medals the specific country “won” (except “NA” medal) in a specific Olympic-game.
9. A query returns top 5 *countries* that have “won” (count) most (GOLD) medals (except “NA” medal) in a specific Olympic-game.
10. A query returns top 5 *countries* that have “won” (count) most (GOLD) medals (except “NA” medal) in a specific Olympic-game, based on gender Male of athlete.
11. A query returns top 5 *countries* that have “won” (count) most (GOLD) medals (except “NA” medal) in a specific Olympic-game, based on gender Female of athlete.
12. A query returns total *medals* from 1986 to 2016 of a specific country.
13. A query returns top 5 *athletes* with the most medals “won” from 1986 to 2016 (except “NA” medal).
14. A query returns top 5 *events* with the most medals “won” from 1986 to 2016 (except “NA” medal).

|  |  |
| --- | --- |
| --Query 1: Total countries attended 2010 Winter Olympic  SELECT COUNT([NOC]) AS [Q1: Total Countries]  FROM attend a  LEFT JOIN olympic\_game g  ON a.gID = g.gID  WHERE g.gYear = 2010 --result: 82 countries occurred in 2010 Winter Olympic |  |
| --Query 2: Total sport events that 2010 Winter Olympic had  SELECT COUNT([eID]) AS [Q2: Total Events]  FROM has h  LEFT JOIN olympic\_game g  ON h.gID = g.gID  WHERE g.gYear = 2010 --result: 86 sport events occurred in 2010 Winter Olympic |  |
| --Query 3: # Canadian athletes participated in 2010 Winter Olympic  SELECT COUNT(ap.[aID]) AS [Q3: # Participated athletes]  FROM athlete\_participation ap  LEFT JOIN olympic\_game g  ON ap.gID = g.gID  LEFT JOIN athlete a  ON ap.aID = a.aID  LEFT JOIN team\_list tl  ON a.tID = tl.tID  WHERE tl.NOC = 'CAN' AND g.gYear = 2010 --result: 198 Canadian athletes |  |
| --Query 4: # Canadian athletes participated in 2010 Winter Olympic that got medals  SELECT COUNT(DISTINCT w.[aID]) AS [Q4: # Participated athletes got medals]  FROM win w  LEFT JOIN olympic\_game g  ON w.gID = g.gID  LEFT JOIN athlete a  ON w.aID = a.aID  LEFT JOIN team\_list tl  ON a.tID = tl.tID  LEFT JOIN medal m  ON w.mID = m.mID  WHERE tl.NOC = 'CAN' AND g.gYear = 2010 AND m.type <> 'NA' --result: 85 Canadian athletes won medals in 2010 Winter Olympic |  |
| --Query 5: Total sport events that Canadian athletes won real medals  SELECT COUNT(DISTINCT w.eID) AS [Q5: Total sport events]  FROM win w  LEFT JOIN olympic\_game g  ON w.gID = g.gID  LEFT JOIN athlete a  ON w.aID = a.aID  LEFT JOIN team\_list tl  ON a.tID = tl.tID  LEFT JOIN medal m  ON w.mID = m.mID  WHERE tl.NOC = 'CAN' AND g.gYear = 2010 AND m.type <> 'NA' --result: 24 sport events won by Canadian athletes in 2010 Winter Olympic  --Query 5 bonus: Name of these sport events  SELECT e.eName, a.aName, m.type  FROM win w  LEFT JOIN olympic\_game g  ON w.gID = g.gID  LEFT JOIN athlete a  ON w.aID = a.aID  LEFT JOIN team\_list tl  ON a.tID = tl.tID  LEFT JOIN medal m  ON w.mID = m.mID  LEFT JOIN olympic\_event e  ON w.eID = e.eID  WHERE tl.NOC = 'CAN' AND g.gYear = 2010 AND m.type <> 'NA'  ORDER BY e.[eName] --result: 24 sport events won by Canadian athletes in 2010 Winter Olympic |  |
| --Query 6: A query returns top 5 sport events the specific country “won” most real medals (except “NA” medal) in a specific Olympic-game.  SELECT TOP 5 e.[eName], COUNT(w.mID) AS [Count medals]  FROM win w  LEFT JOIN olympic\_game g  ON w.gID = g.gID  LEFT JOIN athlete a  ON w.aID = a.aID  LEFT JOIN team\_list tl  ON a.tID = tl.tID  LEFT JOIN medal m  ON w.mID = m.mID  LEFT JOIN olympic\_event e  ON w.eID = e.eID  WHERE tl.NOC = 'CAN' AND g.gYear = 2010 AND m.type <> 'NA' --results: top 5 sport events Canadian athletes won most medals in 2010 Winter Olympic  GROUP BY e.[eName]  ORDER BY [Count medals] DESC |  |
| --Query 7: A query returns count of gold, silver, bronze medals the specific country “won” (except “NA” medal) in a specific Olympic-game.  SELECT m.type AS [Medal], COUNT(w.mID) AS [Count medals]  FROM win w  LEFT JOIN olympic\_game g  ON w.gID = g.gID  LEFT JOIN athlete a  ON w.aID = a.aID  LEFT JOIN team\_list tl  ON a.tID = tl.tID  LEFT JOIN medal m  ON w.mID = m.mID  LEFT JOIN olympic\_event e  ON w.eID = e.eID  WHERE tl.NOC = 'CAN' AND g.gYear = 2010 AND m.type <> 'NA' --results: Gold/Silver/Bronze medal counts that Canadian athletes won in 2010 Winter Olympic  GROUP BY m.type |  |
| --Query 8: Total medals Canadian athletes won in 2010 Winter Olympic  SELECT COUNT(w.mID) AS [Q8: Total medals]  FROM win w  LEFT JOIN olympic\_game g  ON w.gID = g.gID  LEFT JOIN athlete a  ON w.aID = a.aID  LEFT JOIN team\_list tl  ON a.tID = tl.tID  LEFT JOIN medal m  ON w.mID = m.mID  WHERE tl.NOC = 'CAN' AND g.gYear = 2010 AND m.type <> 'NA' --result: 89 medals won by Canadian athletes in 2010 Winter Olympic |  |
| --Query 9: A query returns top 5 countries that have “won” (count) most (GOLD) medals (except “NA” medal) in a specific Olympic-game.  SELECT TOP 5 c.country, COUNT(w.mID) AS [Count medals]  FROM win w  LEFT JOIN olympic\_game g  ON w.gID = g.gID  LEFT JOIN athlete a  ON w.aID = a.aID  LEFT JOIN team\_list tl  ON a.tID = tl.tID  LEFT JOIN country c  ON tl.NOC = c.NOC  LEFT JOIN medal m  ON w.mID = m.mID  WHERE g.gYear = 2010 AND m.type = 'Gold' --results: top 5 countries won most GOLD medals in 2010 Winter Olympic  GROUP BY c.country  ORDER BY [Count medals] DESC |  |
| --Query 10: A query returns top 5 countries that have “won” (count) most (GOLD) medals (except “NA” medal) in a specific Olympic-game, based on gender Male of athlete.  SELECT TOP 5 c.country, COUNT(w.mID) AS [Count medals]  FROM win w  LEFT JOIN olympic\_game g  ON w.gID = g.gID  LEFT JOIN athlete a  ON w.aID = a.aID  LEFT JOIN team\_list tl  ON a.tID = tl.tID  LEFT JOIN country c  ON tl.NOC = c.NOC  LEFT JOIN medal m  ON w.mID = m.mID  WHERE g.gYear = 2010 AND m.type = 'Gold' AND a.asex = 'M' --results: top 5 countries won most GOLD medals in 2010 Winter Olympic based on gender  GROUP BY c.country  ORDER BY [Count medals] DESC |  |
| --Query 11: A query returns top 5 countries that have “won” (count) most (GOLD) medals (except “NA” medal) in a specific Olympic-game, based on gender Female of athlete.  SELECT TOP 5 c.country, COUNT(w.mID) AS [Count medals]  FROM win w  LEFT JOIN olympic\_game g  ON w.gID = g.gID  LEFT JOIN athlete a  ON w.aID = a.aID  LEFT JOIN team\_list tl  ON a.tID = tl.tID  LEFT JOIN country c  ON tl.NOC = c.NOC  LEFT JOIN medal m  ON w.mID = m.mID  WHERE g.gYear = 2010 AND m.type = 'Gold' AND a.asex = 'F' --results: top 5 countries won most GOLD medals in 2010 Winter Olympic based on gender  GROUP BY c.country  ORDER BY [Count medals] DESC |  |
| --Query 12: A query returns total medals from 1986 to 2016 of a specific country.  SELECT c.country AS [Country], g.gYear AS [Year], g.season AS [Season], COUNT(w.mID) AS [Count medals]  FROM win w  LEFT JOIN olympic\_game g  ON w.gID = g.gID  LEFT JOIN athlete a  ON w.aID = a.aID  LEFT JOIN team\_list tl  ON a.tID = tl.tID  LEFT JOIN country c  ON tl.NOC = c.NOC  LEFT JOIN medal m  ON w.mID = m.mID  WHERE m.type <> 'NA' --results: Count medals from 1986 to 2016 of countries.  GROUP BY c.country, g.gYear, g.season  ORDER BY [Country], [Year], [Season] |  |
| --Query 13: A query returns top 5 athletes with the most medals “won” from 1986 to 2016 (except “NA” medal).  SELECT TOP 5 a.aName AS [Athleth Name], COUNT(w.mID) AS [Count medals]  FROM win w  LEFT JOIN athlete a  ON w.aID = a.aID  WHERE w.mID <> 4  GROUP BY a.aname  ORDER BY [Count medals] DESC |  |
| --Query 14: A query returns top 5 events with the most medals “won” from 1986 to 2016 (except “NA” medal).  SELECT TOP 5 e.sport AS [Athleth Name], COUNT(w.mID) AS [Count medals]  FROM win w  LEFT JOIN olympic\_event e  ON w.eID = e.eID  WHERE w.mID <> 4  GROUP BY e.sport  ORDER BY [Count medals] DESC |  |

1. **A summary of the interface**

Interface is implemented using Power BI, a Microsoft tool represents data in meaningful visuals (charts, graphs, plots, etc.). I used proprietary M-language and DAX to develop the report. Download & intall Power BI [here](https://www.microsoft.com/en-us/download/details.aspx?id=58494).

User can filter multiple aspects based on the filter on the right of each report tab. This is an expansion from the queries in section 4, so that user does not have to be constrained to Canada as country and 2010 as Olympic year. However, it is set/saved to be Canada & 2010 when user opens the report for initial comparison. User can filter any country and any Olympic year as user wishes.

The interface (***Pham\_Ngoc\_the\_report\_interface.pbix***) includes 6 report tabs (last tab is intentionally blank):

1. Participation by O-game:
   1. Showing filter by year of the Olympic game on the right. Medal is auto-filtered to exclude “NA” medal.
   2. By selecting a year that a specific Olympic game occurs, the stats of query 1, 2 (mentioned in Section 4) will display.
   3. In addition, a map shows host city for the O-game that year, and a map shows the participated countries with its medal stats.
   4. Be mindful that some years, we have Summer Olympic & Winter Olympic in the same year.
   5. **Query used: 1, 2**
2. Medal Stats by Country, Event:
3. Showing the filters of O-game, country on the right. Medal is auto-filtered to exclude “NA” medal.
4. By selecting the proper filter, user can find medal stats, event stats of the filtered country in the filtered O-game.
5. **Query used: 3, 4, 5, 6, 7, 8**
6. Top medals ranking by Country:
7. Showing filter by year of the Olympic game, and medal types on the right.
8. User can filter specific medal type and see the stats based on Country (donut chart), and gender-specific of that country.
9. User can tell which countries are in top 5.
10. **Query used: 9, 10, 11**
11. Medal stats by Country, Gender
12. Showing filter by country, and medal types on the right.
13. A different representation of Report tab 3. This report is in stacked column chart.
14. The medal stats are in gender-specific, or medal-type-specific, filtered by country.
15. User cannot tell which countries are in top 5, due to this is filtered by country.
16. **Query used: 9, 10, 11**
17. Total Medals by Country
18. Showing filter by country. Medal is auto-filtered to exclude “NA” medal.
19. Count all medals by countries over the year 1986-2016.
20. Pre-filtered to Canada, Japan, Russia, USA. Select more than 4 countries, user needs to scroll down to view it.
21. **Query used: 12**
22. Top Athletes & Top Sports:
23. Medal is auto-filtered to exclude “NA” medal.
24. Using stacked bar chart, the report displays the top 5 athletes that won most medals over the year 1986-2016, and top 5 sport events that have most medals over the year 1986-2016.
25. **Query used: 13, 14**