|  |
| --- |
|  |
| The Olympics |
|  |

|  |
| --- |
| Jeffrey Parsons & Daisy McClellan  University of Houston – Clear Lake |

Contents

[Abstract 2](#_Toc26891321)

[Project Overview 3](#_Toc26891322)

[Methodology 4](#_Toc26891323)

[Introduction to Data Warehousing 6](#_Toc26891324)

[The Need for Data Warehousing Applications 7](#_Toc26891325)

[Project Data Set 8](#_Toc26891326)

[Data Cleansing 9](#_Toc26891327)

[Data Transfer to MS Access 11](#_Toc26891328)

[Dimensional Modeling 20](#_Toc26891329)

[Athletes Dimension: 21](#_Toc26891330)

[Events Dimension: 21](#_Toc26891331)

[Games Dimension: 22](#_Toc26891332)

[Medals Dimension: 22](#_Toc26891333)

[Teams Dimension: 23](#_Toc26891334)

[Names Table: 23](#_Toc26891335)

[Sports Table: 24](#_Toc26891336)

[Fact Table: 25](#_Toc26891337)

[Data Transfer to SQL Server 26](#_Toc26891338)

[Cube Development and Deployment in Visual Studio 34](#_Toc26891339)

[Reports 45](#_Toc26891340)

[Report 1: 45](#_Toc26891341)

[Report 2: 46](#_Toc26891342)

[Report 3: 47](#_Toc26891343)

[Report 4: 48](#_Toc26891344)

[Report 5: 49](#_Toc26891345)

[Report 6: 50](#_Toc26891346)

[Conclusion 52](#_Toc26891347)

[References 54](#_Toc26891348)

# Abstract

This project aims to inspect 120 years of results from the Olympic games. To do this, a data warehouse was created. This process involved data cleansing, transferring the data to MS Access, dimensional modeling, transferring to SQL Server, cube development and deployment, and creating reports so the data could be analyzed. Once these steps were complete we found that the females actually didn’t compete in the Olympics until 1920. We also found that the largest sports for medals earned were athletics, rowing, and swimming. The United States came out on top when looking at the total number of medals won. We discovered that the age with the greatest number of medals won is 23 and it seems that males win more gold medals than females. Lastly, we found that females have the best chance to win a medal if they compete in team-based events such as Ice Hockey.

# Project Overview

The goal of our project was to create a data warehouse using a dataset of our choosing. For this project we chose a dataset of different athletes that competed in the past Olympic games, as well as information about them. The information included things such as the athlete’s team, sex, age, height, weight, and the event they competed in. Of course, it also included whether the athlete won a medal or not. To create the data warehouse, we needed to cleanse the data. This involved getting rid of specific columns from the original data that were not required for our analysis. The basis of our data warehouse hinged on being able to use these metrics to compare each athlete, and the event they competed in, with others to see if there were any trends that could provide any insight to how an athlete would perform. However, there were a few errors with some of the records, specifically null, or NA values, for some of the athlete’s height, or weight. This would ruin the results when trying to analyze the data based on these metrics. The solution was to remove these records all together. This was not an issue for us because the data still contained more than 200,000 additional records. From this point we needed to create different dimensions and a fact table. The dimensions we settled on were events, athlete, games, medals, and teams. Based off these dimensions each unique value or combination was assigned an ID. These ids were then used to replace the original data for each row to create the fact table. Once this was finalized, we imported these dimensions into MS Access. This allowed us to create tables for each dimension as well as establish the relationships between them. The next step in creating the data warehouse was to import the database created in MS Access to SQL Server. Once this was complete, SQL queries could be run on the data. A few test queries were run to make sure the results matched that original data. The last step was to create and deploy the cube and generate reports. To do this, Visual Studio was used. The database from SQL Server was used as the data source for the cube. Once the hierarchies were set the cube could then be successfully built and deployed.

# Methodology

**Following are the steps taken in creating our data warehouse:**

* **Data Selection:** First we were tasked with selecting a dataset for the project. The data needed to have enough columns to be able to create at least 5 dimensions and a fact table after the data had been cleansed. In addition to this, it also needed to have enough records to allow us to perform an adequate analysis.
* **Data Cleansing:** During the data cleansing process, any columns that were not useful to our analysis in the original dataset or any values that would cause issues or redundancy were removed. In this step, any reduction in the number of records being analyzed could have been done if the original data contained too many rows.
* **Fact Table and Dimension Creation:** Once the data was cleaned the dimensions and fact table were created in MS Excel.
* **Transfer to MS Access:** After the creation of the dimensions and fact table, the files were imported to MS Access. From here, tables were created from the different dimensions as well as the relationships between them.
* **Transfer to SQL Server:** The database that was created in MS Access was then imported into SQL Server. From here queries could be ran on the data in SQL Server.
* **Cube Generation and Deployment in Visual Studio:** Using Visual Studio, the database was imported from SQL Server. From here the cube was created. After deploying the cube, an Excel spreadsheet was generated allowing for analysis of the data.
* **Creation of Reports:** Lastly, we created reports that were generated from the cube generation and deployment to analyze the data.

# Introduction to Data Warehousing

A Data Warehouse is “designed to support management decision-making process by providing a platform for data cleaning, data integration and data consolidation. A data warehouse contains subject-oriented, integrated, time variant and non-volatile data” (chaitanyashah707, 2019). Data from many sources in compiled and cleansed for quality and consistency. Queries are used to analyze the data.

Data mining is “the process of finding patterns and correlations within large data sets to identify relationships between data.” (chaitanyashah707, 2019). Data mining combines statistics with database management to analyze data sets (Clifton, 2017).

# The Need for Data Warehousing Applications

The need for data warehousing and data mining is to analyze large amounts of data into manageable tables and queries. This is important to business for several reasons. Number one is to ensure consistency in the data. The data warehouse is programmed so that the data in collected in a uniform format and any new data will be inserted in the warehouse in the same format. The second reason is to help businesses make better decisions. Data warehousing and data mining help improve the effectiveness and swiftness of accessing the data. This speed helps businesses to make speedy and efficient decisions to help guide their marketing and business strategies. The quick access to a business’s historical data can ultimately help improve their bottom line. That is, the queries run on the data can help business leaders see if their strategies are working or see where they need to focus more costs (Staff, 2017).

# Project Data Set

The data set we chose is titled “120 Years of Olympic History: Athletes and Results”. It can be found at:

<https://www.kaggle.com/heesoo37/120-years-of-olympic-history-athletes-and-results>

The original author states that the dataset is a “historical dataset on the modern Olympic Games, including all Games from Athens 1896 to Rio 2016” (Rgriffin, 2018).

For our cleansed dataset our columns, with their names, definitions and data types are as follows:

Column Name: Name

Def: The name of the athlete

Data Type: varchar

Column Name: Sex

Def: The gender of the athlete

Data Type: varchar

Column Name: Age

Def: The age of the athlete

Data Type: Int

Column Name: Height

Def: The height of the athlete

Data Type: Int

Column Name: Weight

Def: The weight of the athlete

Data Type: Int

Column Name: Team

Def: The team that the athlete competed for

Data Type: varchar

Column Name: Year

Def: The year the athlete competed

Data Type: Int

Column Name: Season

Def: The season that the Olympics took place

Data Type: varchar

Column Name: City

Def: The city that the Olympics were held the year the athlete competed

Data Type: varchar

Column Name: Sport

Def: The general sport the athlete competed in

Data Type: varchar

Column Name: Event

Def: The actual title of the event that the athlete competed in

Data Type: int

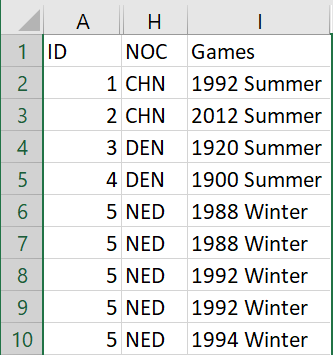
Column Name: Medal

Def: The type of medal that the athlete earned

Data Type: varchar

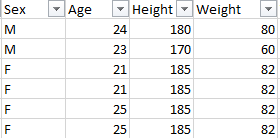
# Data Cleansing

The first step in the data cleansing process was to remove any of the columns in the original dataset that were not required for our analysis. These columns included the id, NOC, and Games columns. The Games column held information that was already in the other columns. Due to this we felt that it was redundant to include it in our cleansed data. These are examples of the columns that were removed from the original data.



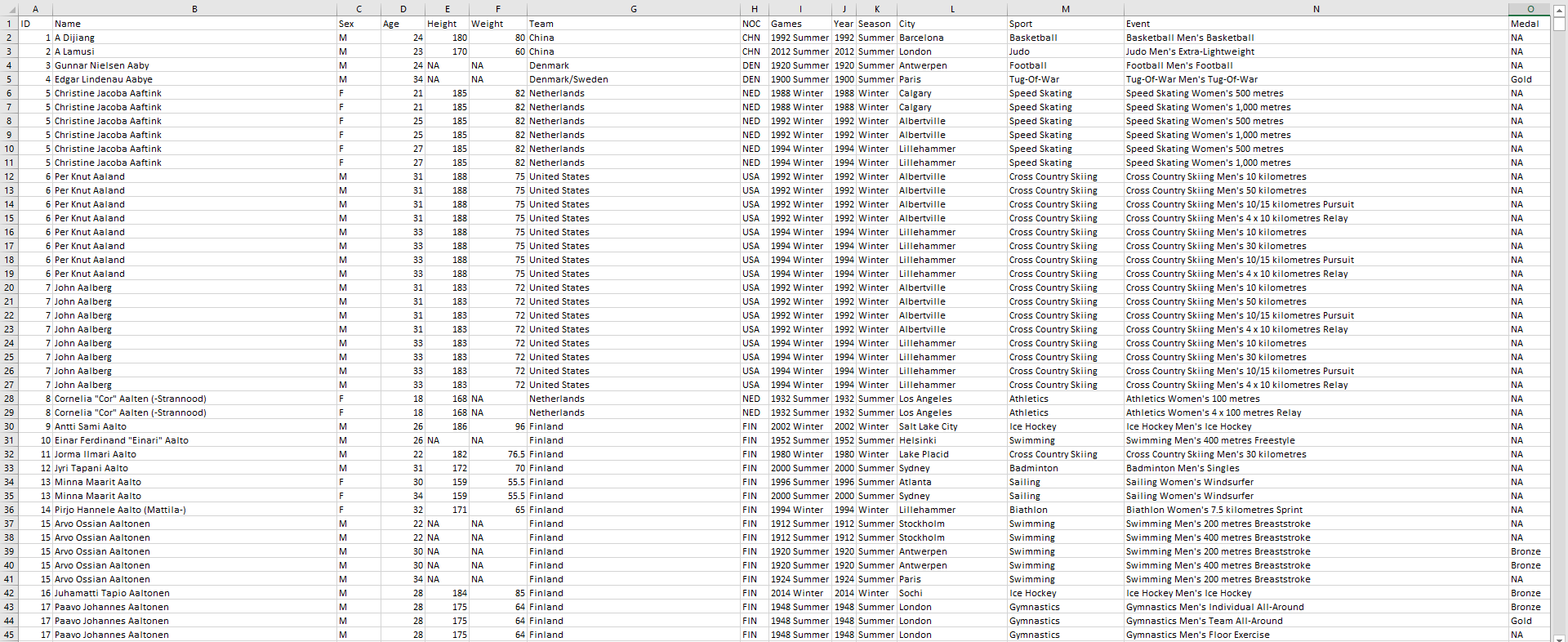
After this, we ran into a few issues with null, or NA, values in the Age, Height, and Weight columns. Since this would ruin our analysis based on these points, we removed any records where these values were null. Below is a before and after example of this issue:

A screenshot of a cell phone

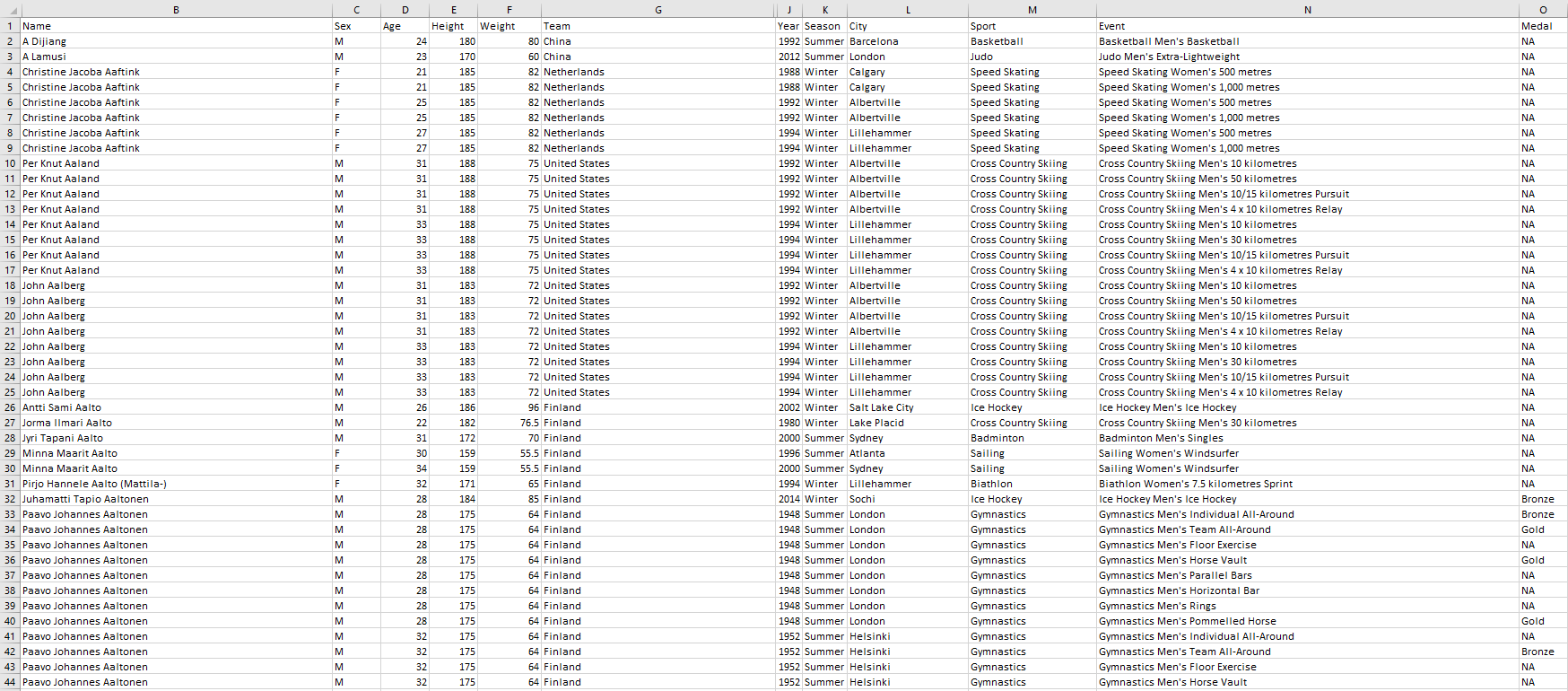
Description automatically generated 

You can also see the difference in the dataset overall after the cleansing was complete. Below is a before and after picture of all columns and some errors that were in the original data that are no longer present in the cleansed dataset.

Before cleansing:



After cleansing:

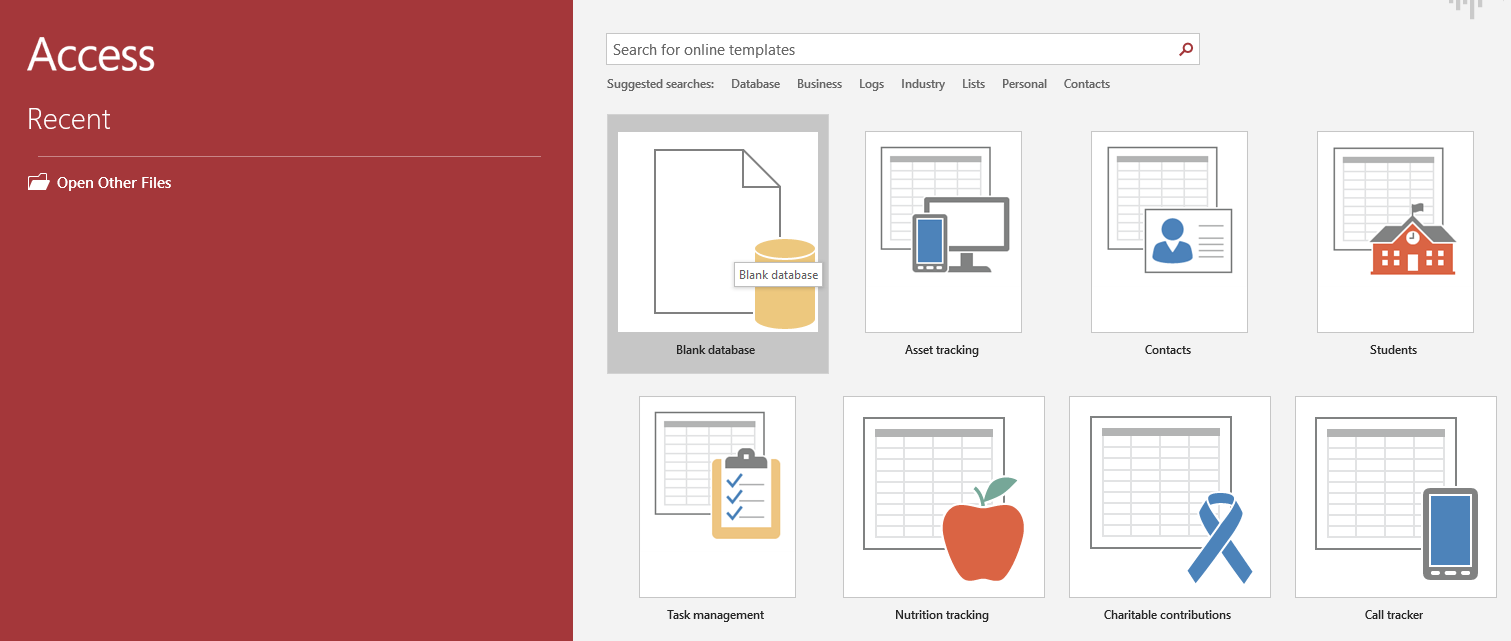


# Data Transfer to MS Access

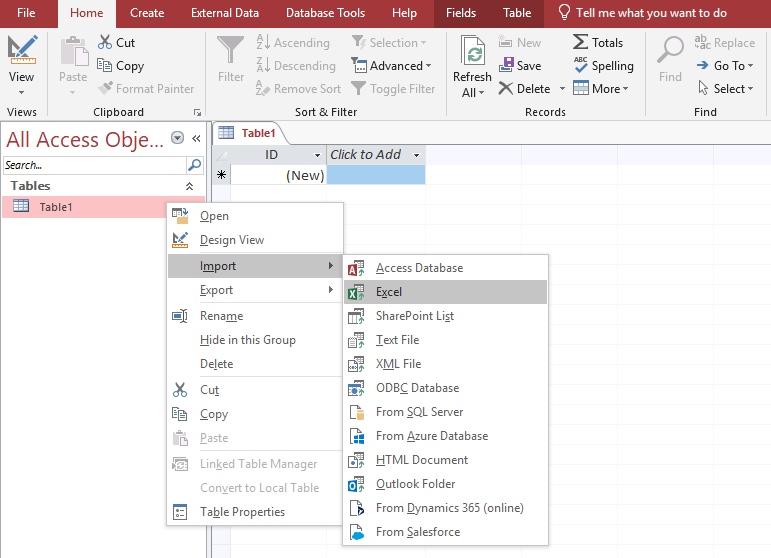
Transferring the dataset to MS Access required that first, the data be cleansed and organized into dimensions. Each dimension was then summarized in a fact table with all the primary keys listed. To transfer the data, we opened Microsoft Access and chose to create a new table by importing data from excel. The first table we chose was our fact table. We selected that the first rows were headers and we had already created primary keys. After the fact table was successfully loaded into Access, we loaded in the dimensions we had which were, Athletes, Events, Medals, Games, Names, Sports and Teams.

Below are the steps that were followed to transfer the data to MS Access:

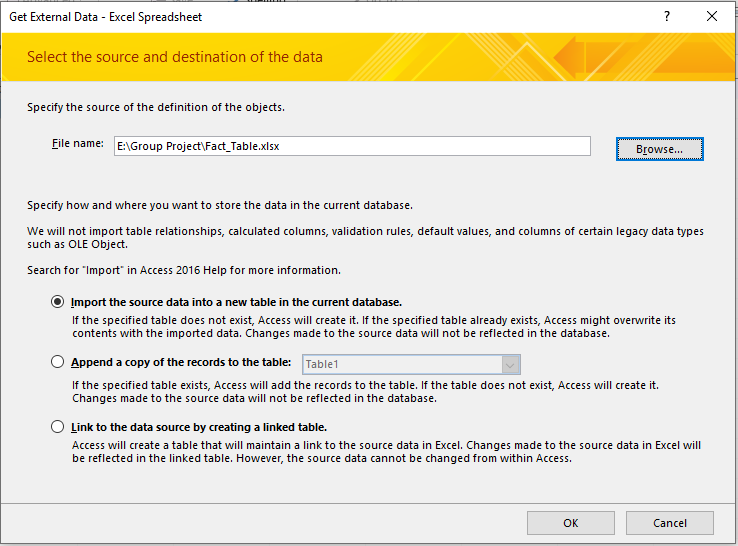
To begin the transfer to Microsoft Access, first open Access and select a “blank database”.



Next, right click on the “Table1” table on the left-hand side. Select “Import” and then “Excel” to import the dataset from the excel files.

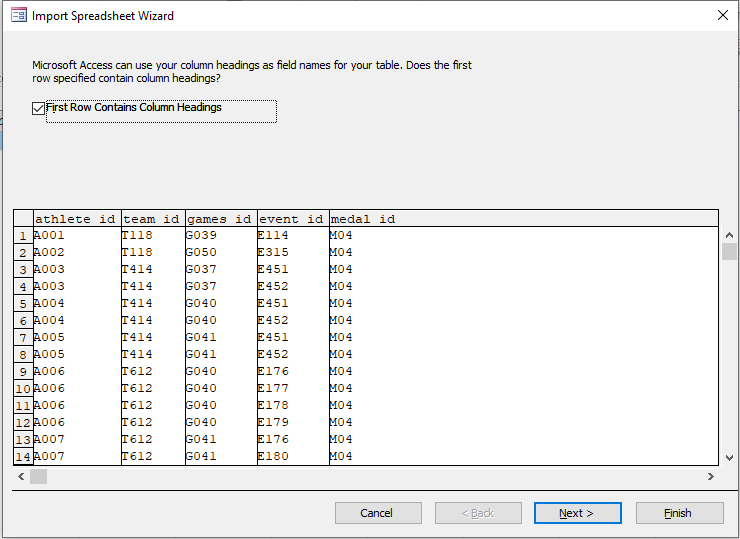


A box should pop-up like the one below. Select the “Browse…” button to search for the Excel file you are wanting to upload.

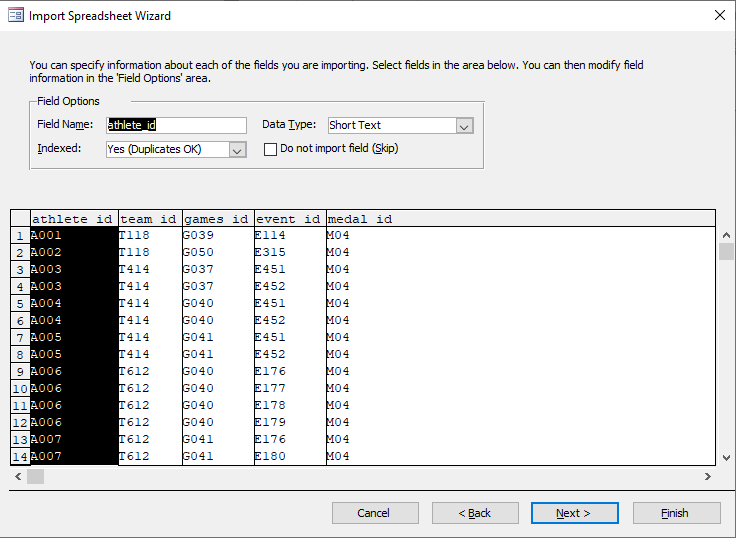


You will keep the selection of “Import the source data into a new table in the current database”. Next press “OK”.

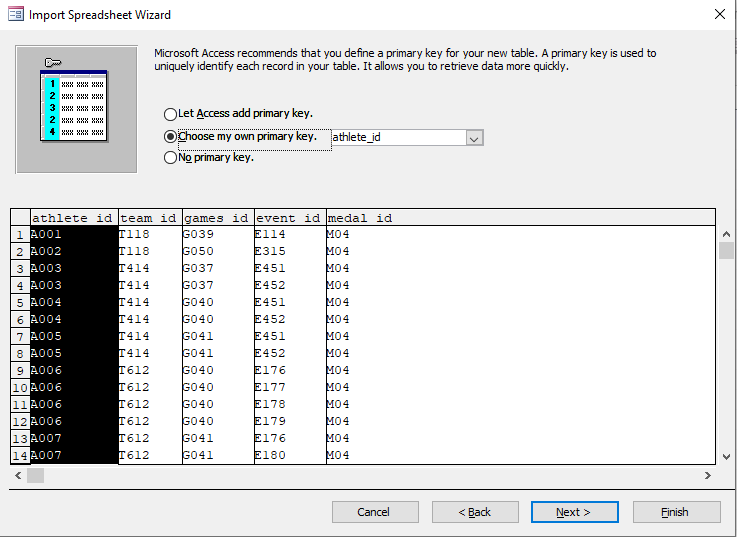
On the next screen, you will select the box that says “First Row Contains Column headings” because the first rows of our Excel file were the column headings. Then press “Next >”.



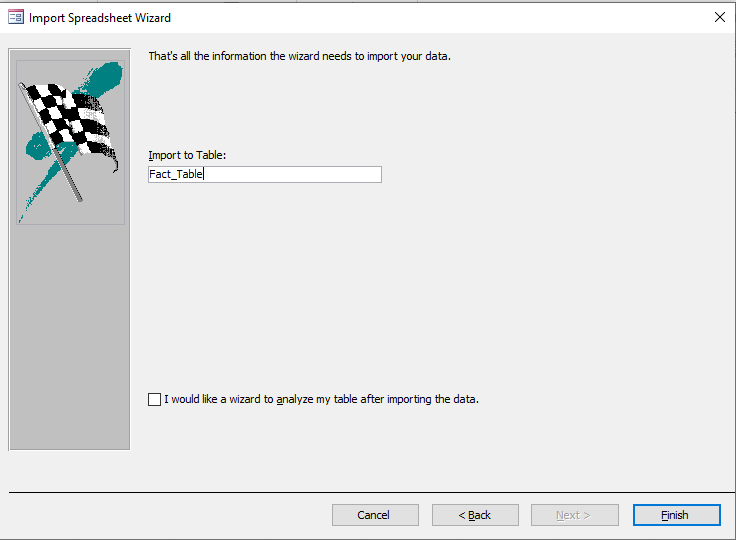
Next, for the box after “Indexed: “, make sure if your rows have repetitive numbers, such as our athlete\_id column, you select “Yes” that it is indexed, but that duplicates are okay. If there are no duplicates, you will select the “no duplicates” option. Then press “Next>”.



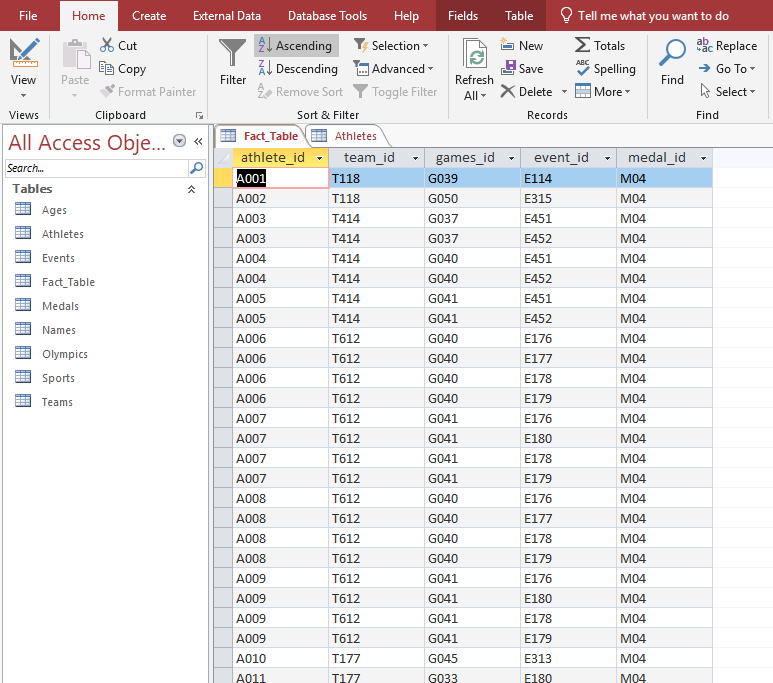
On this screen, it is asking if you want Access to add a Primary Key or if you have already chosen your own. We had already chosen our own Primary Key, so we selected “Choose my own primary key” and selected the column that was our primary key.



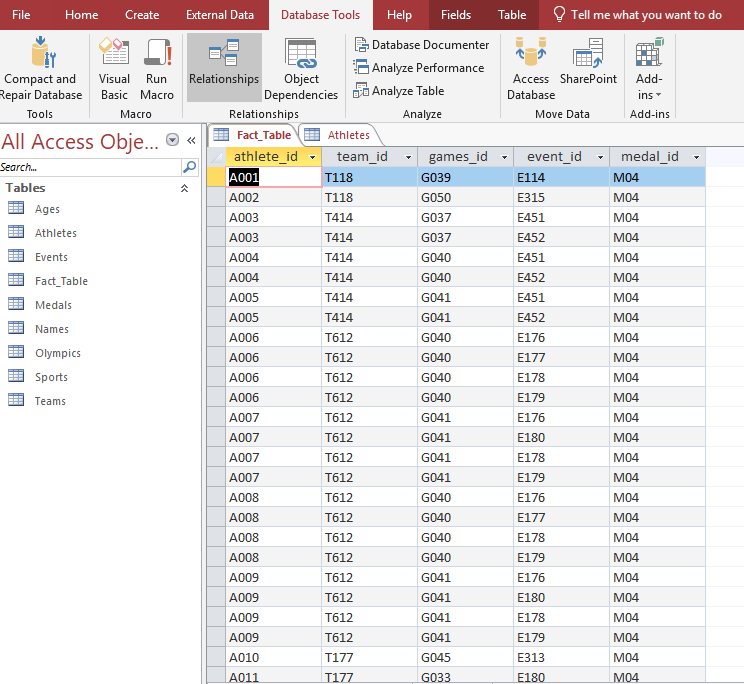
Next, you will name your table that you are adding to Access and select “Finish”.

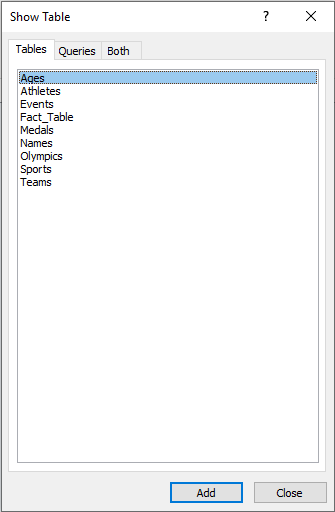


After transferring all the tables in the same manner and deleting the original “Table1” blank table your Access database should look like this.



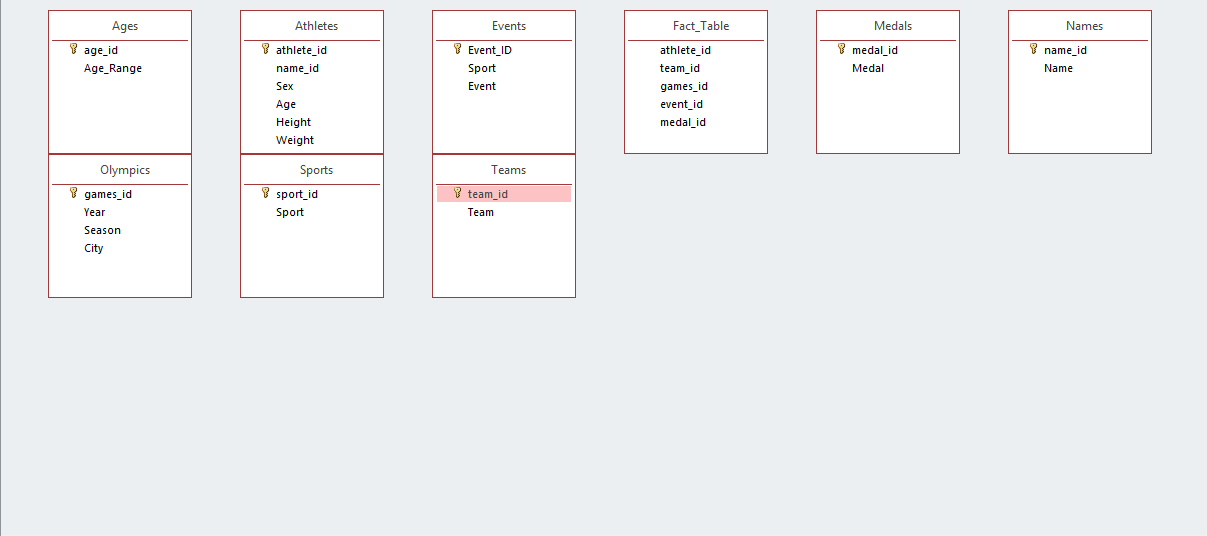
To create the relationship diagram, or the ERD, go to “Database Tools” and select “Relationships”.

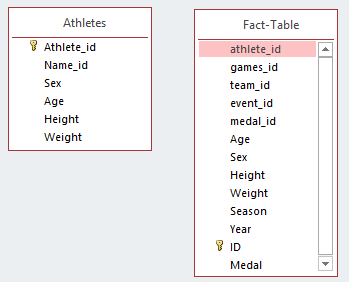


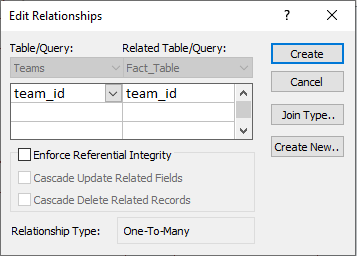


In this box, you will individually select every table and hit “Add” and this will add the tables into the relationship diagram box behind it.

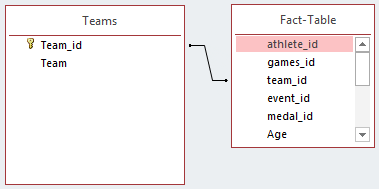
Your tables should look like this below. With no lines or relationships yet.



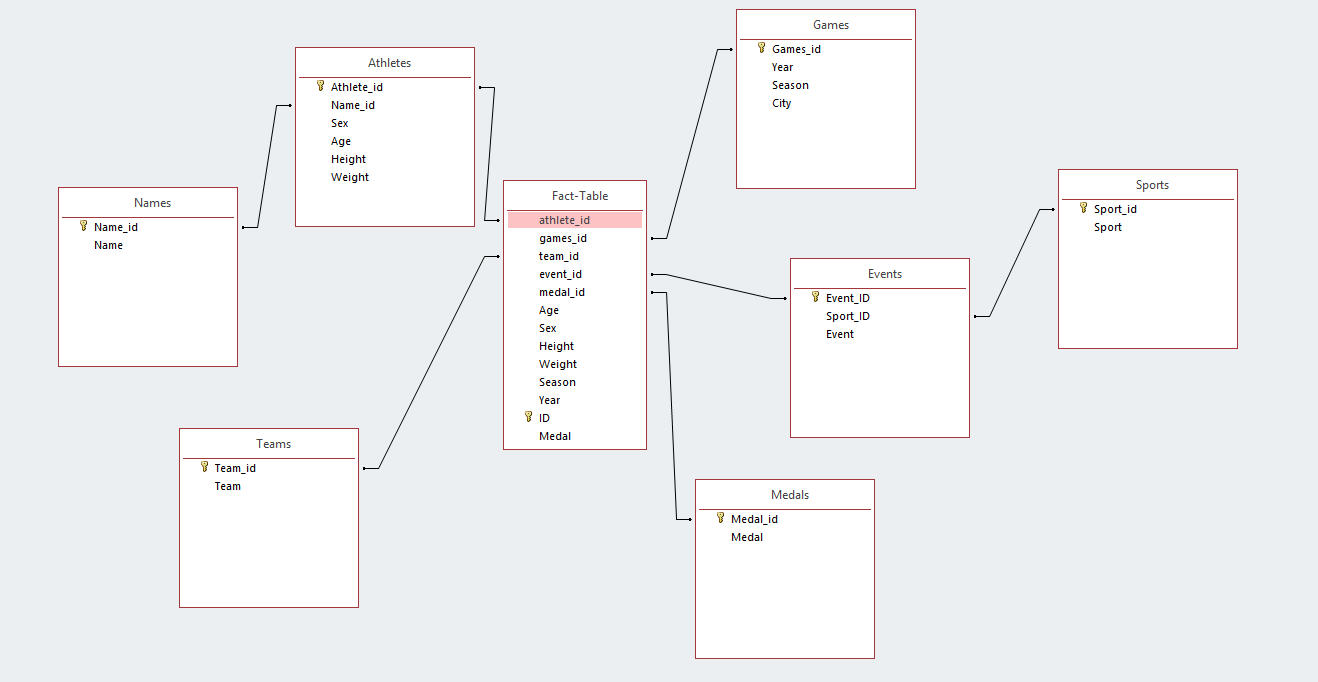


To create the relationships, select the primary key from one table and drag in to match the foreign key in your fact table.

This box is checking the primary keys and foreign keys. If they are accurate in both columns click “Create”

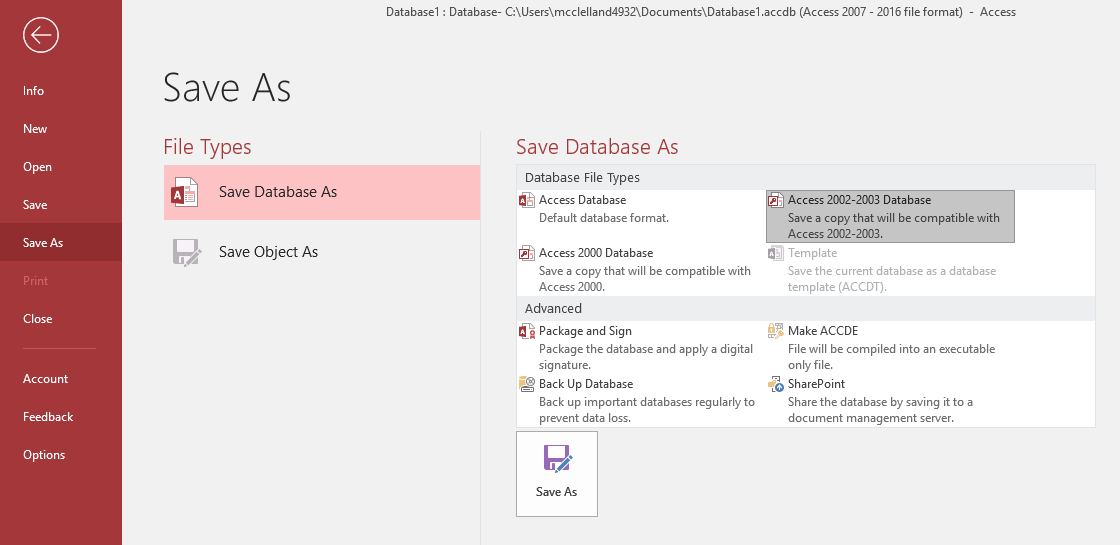


You should now see the line showing the relationship between the child table and the parent table. Continue this action for all of the tables to create the ERD.



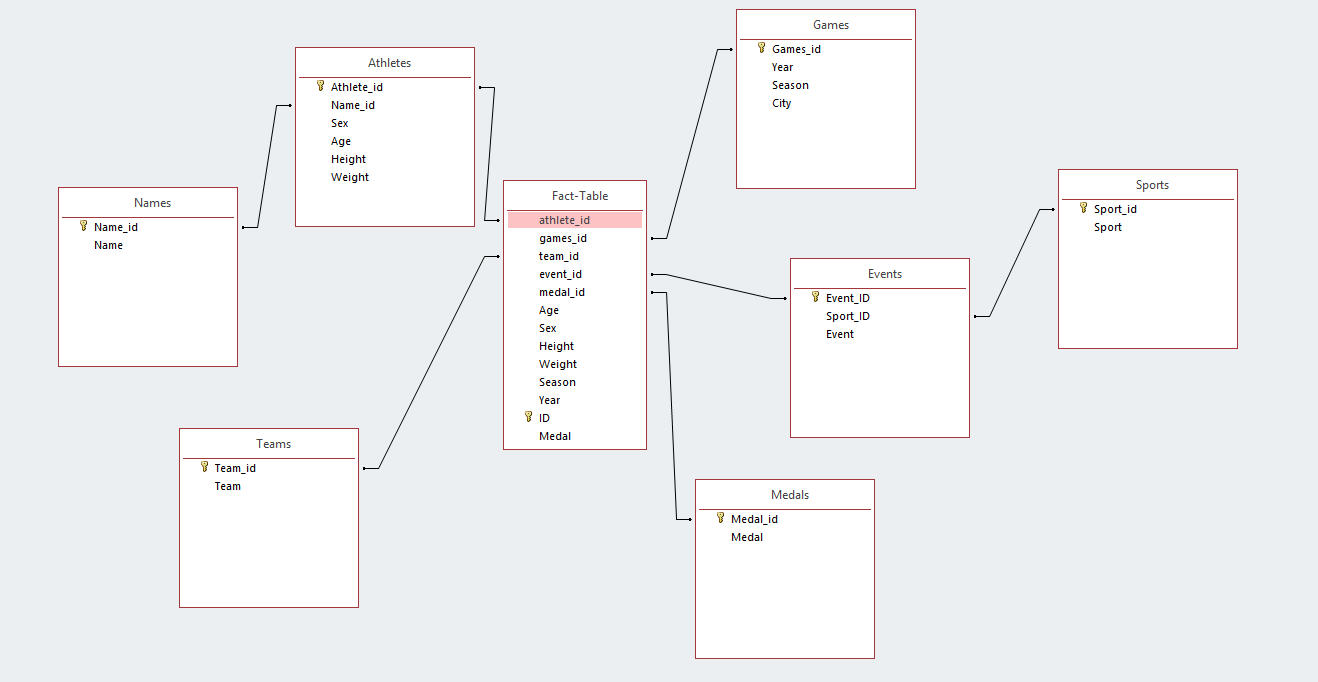
This should be the result. An ERD with relationships to every table. If your tables are not aesthetically pleasing, you can move them around to make the diagram easier to read.

Finally, when saving your database in Access, make sure to save it as an “Access 2002-2003 Database” file. This will make the transfer to SQL Server possible.



# Dimensional Modeling

Once all the tables were imported to Access, we were able to create an ER Diagram that shows how all the tables relate to one another. As you can see from the ER Diagram below, the Fact Table was the central table that every dimension is connected to and holds a reference of the dimensions primary key as a foreign key.



After analyzing our data, we settled on five main dimensions: Games, Events, Medals, Teams, and Athletes. The Athletes and Events tables create a snowflake schema due to there being additional tables that connect directly to them, but not to the fact table itself. We figured this was necessary to prevent duplicate values (that appear to have no correlation) from appearing in the same table. In the case of the Athletes table, the athlete name would appear more than once. This is an issue because the way the original data is set up, the name appears to be referencing the same individual, but since their age, height, weight, or event may be different, they appear more than once and are treated as a completely separate athlete. When the name is pulled out to a separate table and referenced in the Athlete table, this gives us a clearer indication of who the athlete is on an individual level. A similar situation happens in the Events table regarding the sport column. Each event has a sport category that encompasses the individual event. Due to this, each sport can have many different events in its category. A simple solution would have been to get rid of the sport section altogether and just focus on the event. However, instead of cutting out the sport type completely, we decided to leave it in as its own table to allow the possibility to analyze by sport type in general as opposed to a finer analysis by event.

## Athletes Dimension:

For every record in the original data we have a corresponding athlete in the Athlete dimension. This dimension includes a reference to the athlete’s name from the Name table as well as the sex, age, height, and weight of the competitor.

A screenshot of a cell phone

Description automatically generated Attributes:

* Athlete\_id (pk)
* Name\_id (fk)
* Sex
* Age
* Height
* Weight

## Events Dimension:

The Event dimension contains all the information about each event that appears in the data. We removed the duplicates and gave each unique event its own id. It also contains a reference to the sport that event is a part of as well as the title of the event.

A screenshot of a cell phone

Description automatically generated Attributes:

* + Event\_ID (pk)
  + Sport\_ID (fk)
  + Event

## 

## Games Dimension:

The Games dimension includes a unique id for each year, city, and season combination that appears in the original data. Since we had the year, season, and city already in individual columns it was easy to assign an id to each unique combination while eliminating the need for the Games column in the original data. This allows for analysis by any year, season, or city individually instead of all of them at the same time.

A screenshot of a cell phone

Description automatically generated Attributes:

* + Games\_id (pk)
  + Year
  + Season
  + City

## Medals Dimension:

The Medals dimension was very straight forward. The athlete either won a gold, silver, or bronze medal. If they didn’t medal in that event, it is denoted by NA in the original data. All that was needed here was to set a unique id for each different type of medal, or lack thereof, that could be won.

A screenshot of a computer screen

Description automatically generatedAttributes:

* Medal\_id (pk)
* Medal

## Teams Dimension:

For the Teams dimension, once again we took each individual occurrence of a team name and assigned it a unique id. Initially the plan was to use country abbreviations as the unique id, but we quickly found out that not all teams were countries (primarily in the earlier years of the Olympics) and some of the abbreviations were duplicates.

A screenshot of a cell phone

Description automatically generated Attributes:

* + Team\_id (pk)
  + Team

## Names Table:

The Names table contains each individual name that appears in the data. It was then given a unique id that is referenced in the Athletes dimension.

A screenshot of a cell phone

Description automatically generated Attributes:

* + Name\_id (pk)
  + Name

## 

## Sports Table:

The sports table contains the individual sports that each event is a part of. We took all the unique values and gave them an id as well. This id is referenced in the Events dimension.

A screenshot of a cell phone

Description automatically generated Attributes:

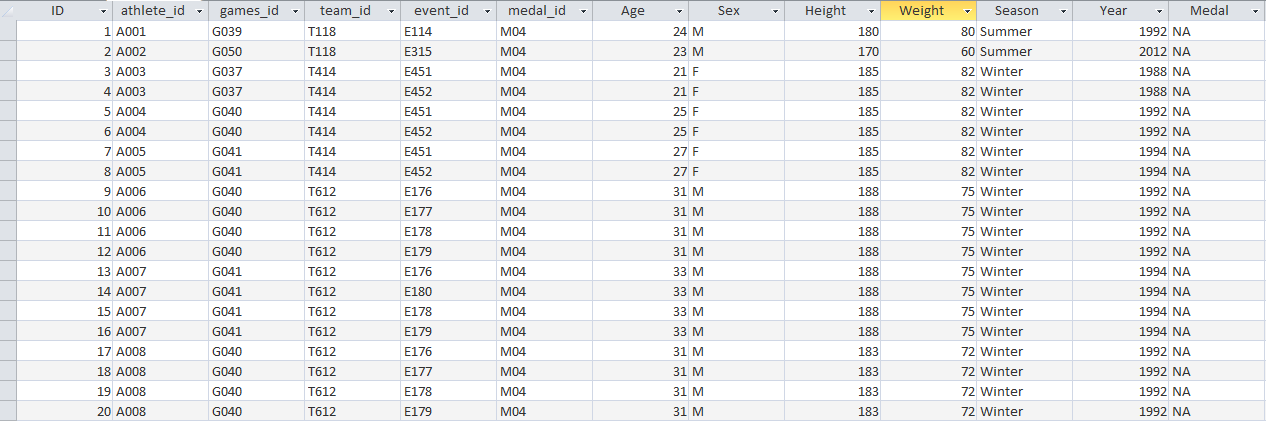
* + Sport\_id (pk)
  + Sport

## Fact Table:

The Fact Table is where all the dimensions come together. The primary keys of each dimension are represented in the Fact Table as foreign keys. This relationship can be seen by examining the previously mentioned ER Diagram above.

Attributes:

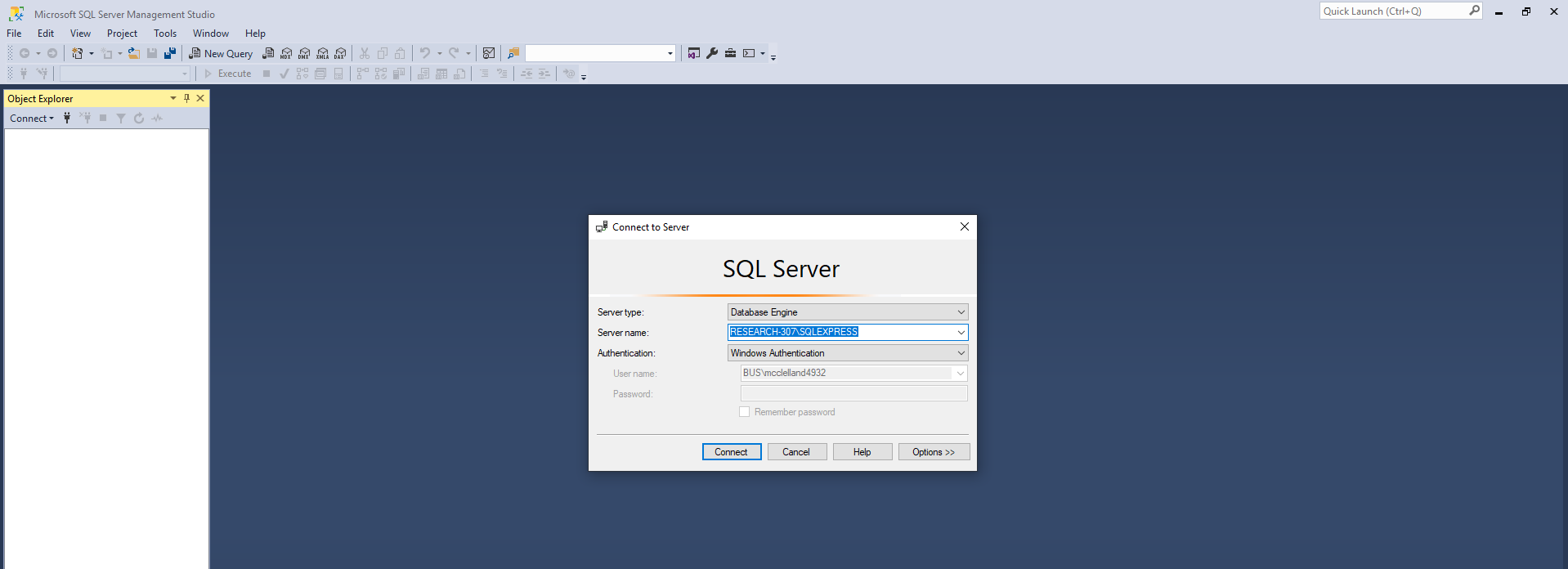
* + ID (pk)
  + Athlete\_id (fk)
  + Team\_id (fk)
  + Games\_id (fk)
  + Event\_id (fk)
  + Medal\_id (fk)
  + Age (fact)
  + Sex (fact)
  + Height (fact)
  + Weight (fact)
  + Season (fact)
  + Year (fact)
  + Medal (fact)



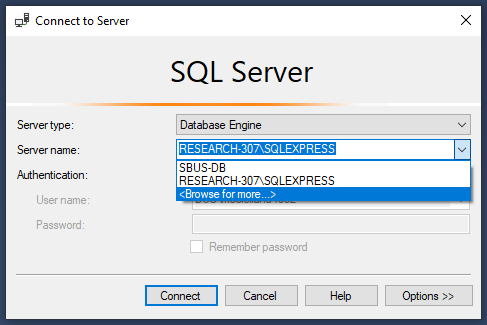
# Data Transfer to SQL Server

The following instructions will follow the steps that must be taken in order to transfer the data from MS Access to SQL server.

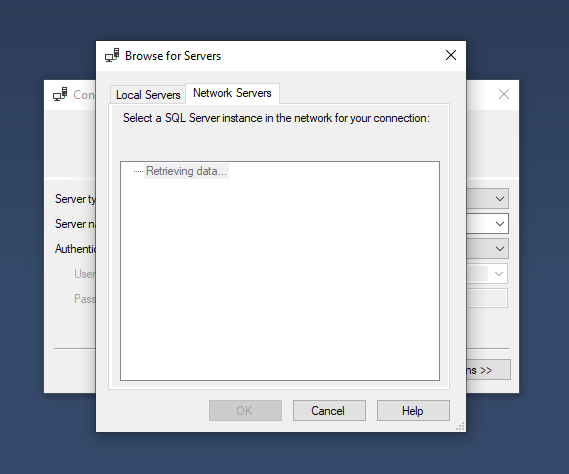
When you first open SQL Server, this is the page you should see. You will need to log into a new server.



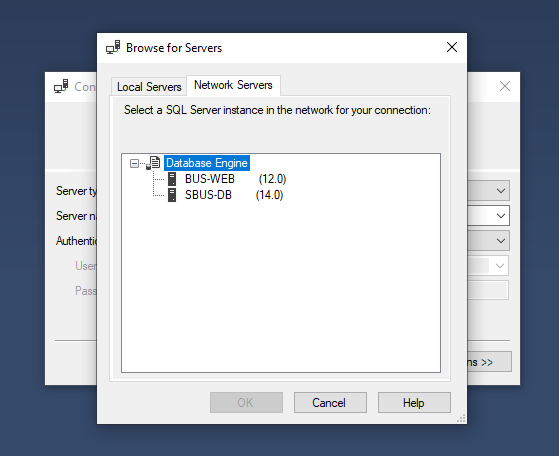
To do this, click the arrow next to “server name:” and select “<Browse for more…>”



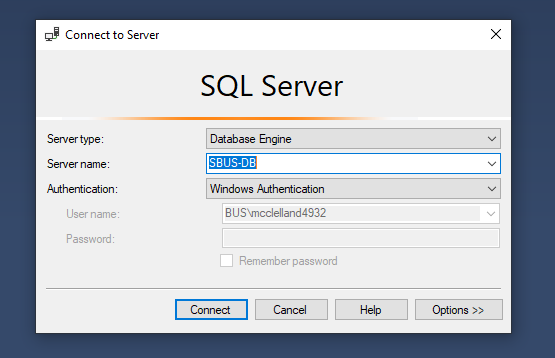
On this page, you will select “Network Servers” instead of “Local Servers”



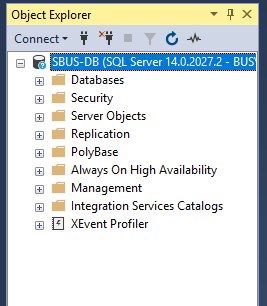
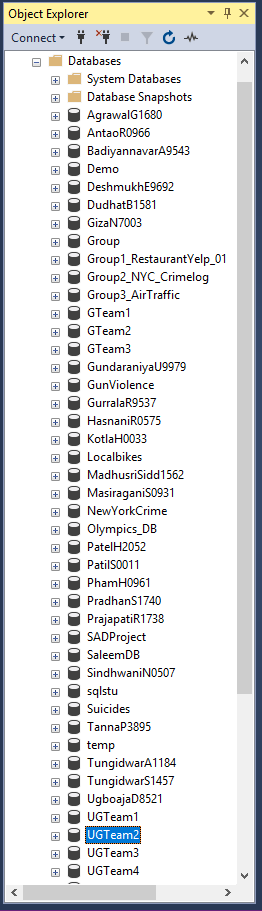
After it loads, you will select “SBUS-DB” and select “OK”. Your screen should now look like this.

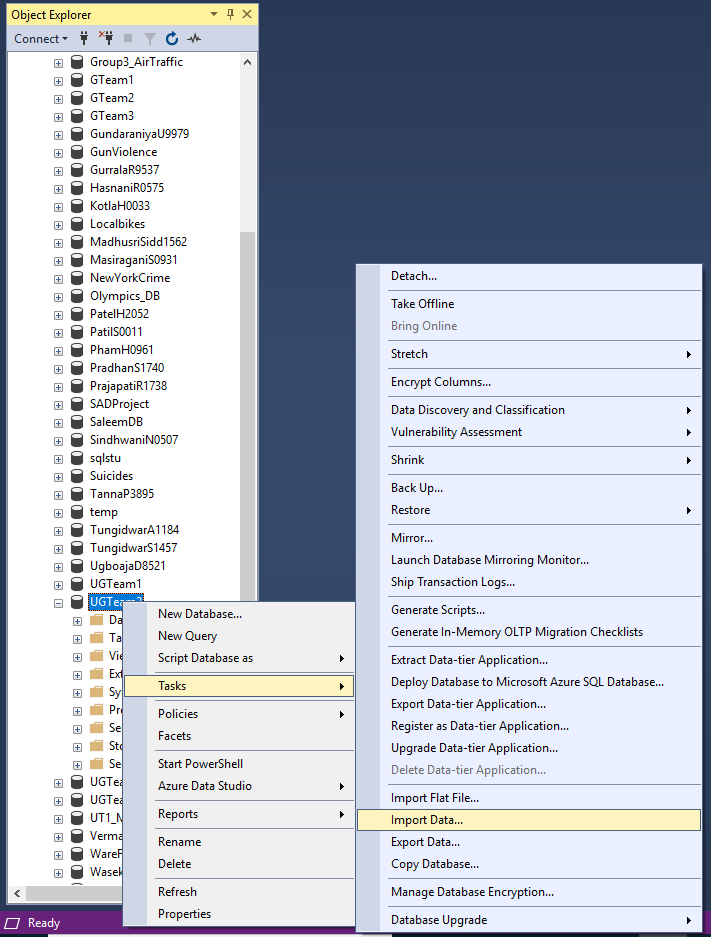


After selecting “Connect”, you will be connected to the SBUS-DB server. Open the folder titled “Databases” and find the name of your assigned group. For this exercise, our group name was “UGTeam2”.

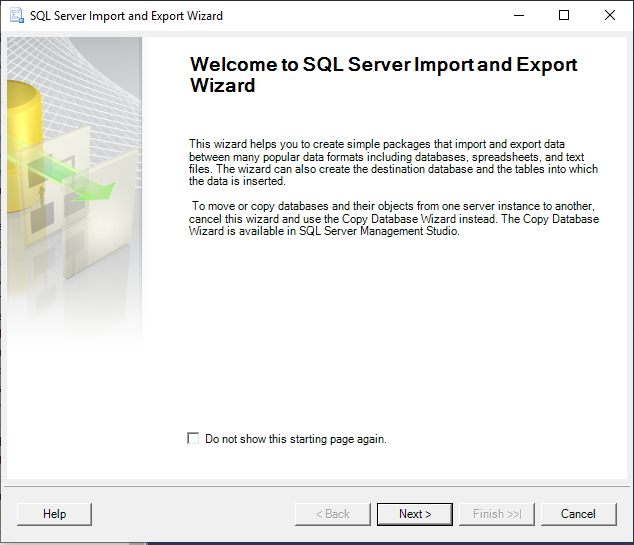


To import the data from Microsoft Access, you will right click on your team name, scroll down to “Tasks” and then down to “import Data…”.

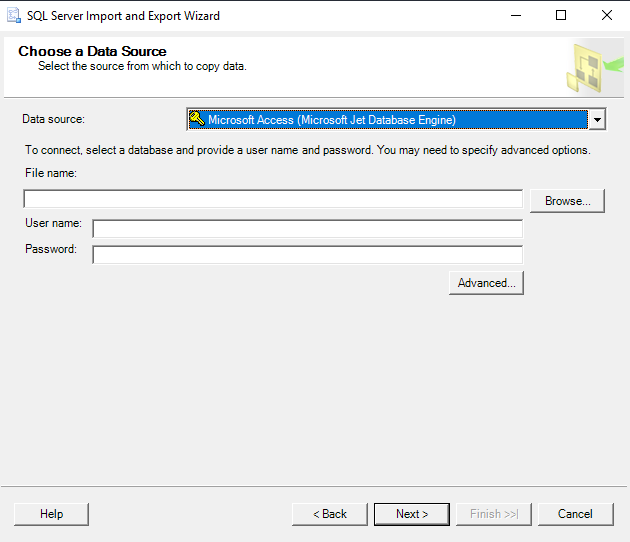
 



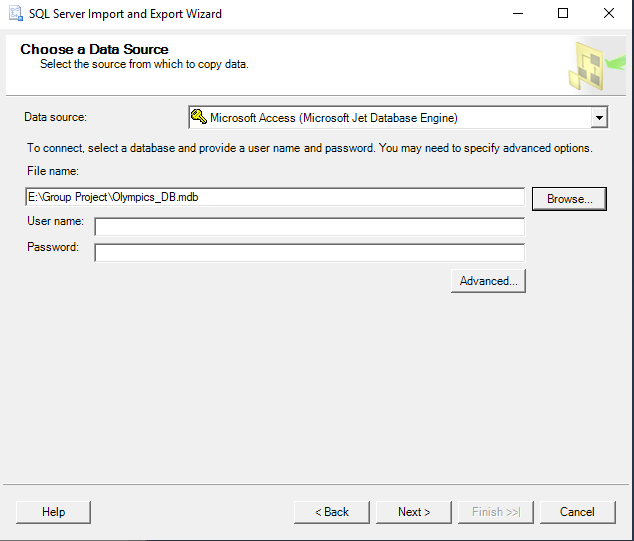
This is the import and export wizard. Select “Next >”.



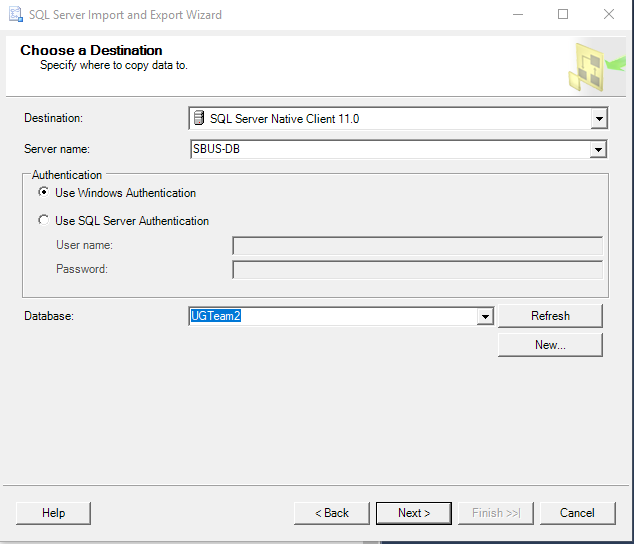
The data source you want to select on the next page is “Microsoft Access (Microsoft Jet Database Engine). Then select “Next>”.



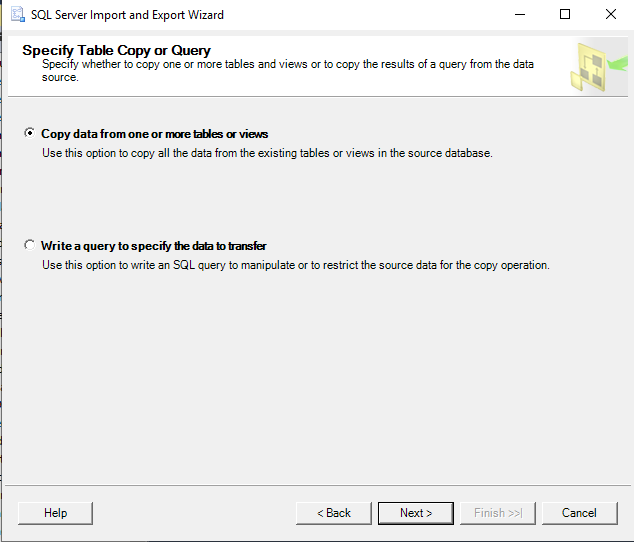
Browse your computer to find the Microsoft Access file that you saved previously to import.



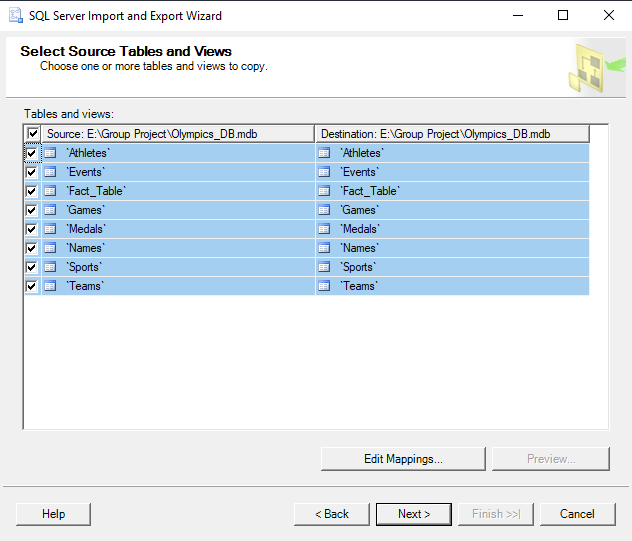
The next page will ask you to choose a destination. You will choose the “SQL Server Native Client 11.0” with the server name of “SBUS-DB” and the database name of your group name. Then select Next.



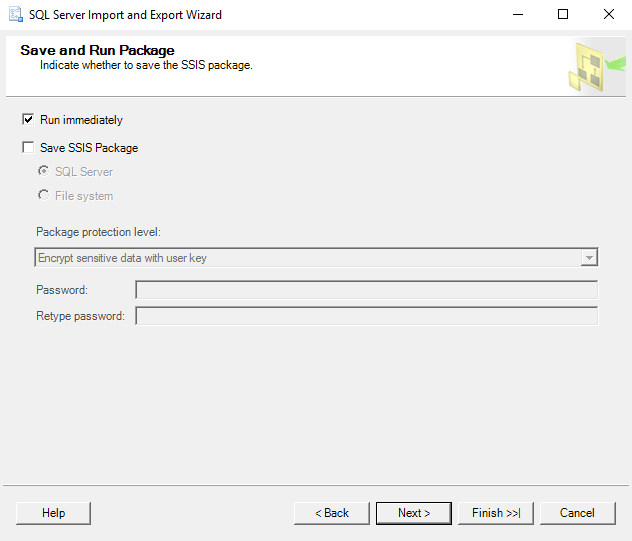
Because you are transferring the entire database from Access you will select “Copy data from one or more tables or views”. Then select “Next”.



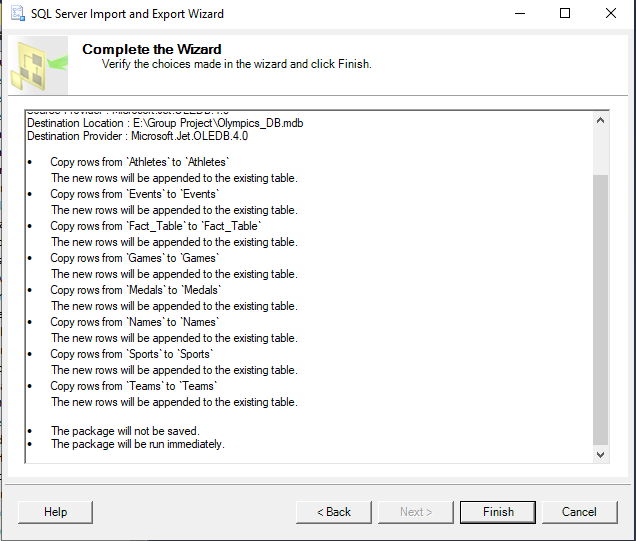
This page shows all the tables and views (if you have any) from your Access database. Make sure all the tables you want to transfer and selected and press Next.



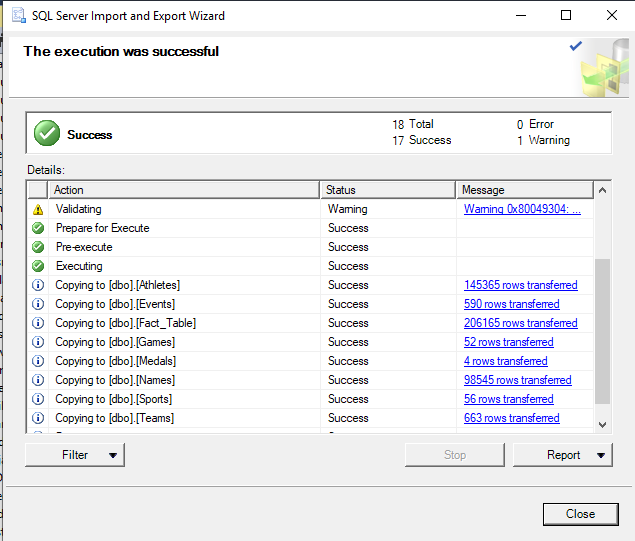
Make sure “Run immediately” is selected and press “Next”.



This page summarizes what will be happening with your database. Select “Finish”.

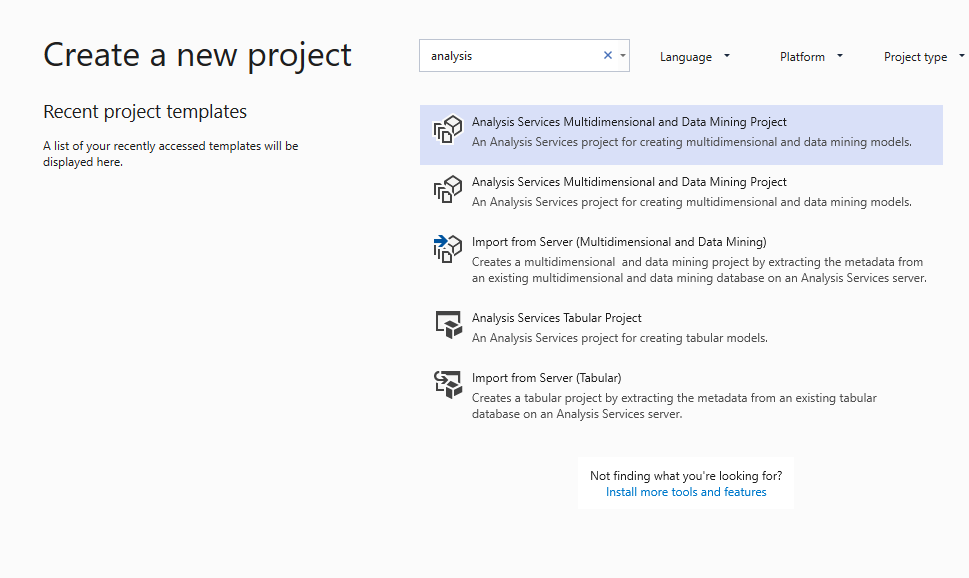


When you are done all columns should have the green check mark next to them. When the statuses show “success”, select close. Your database should now be fully loaded into your server, under your group name.

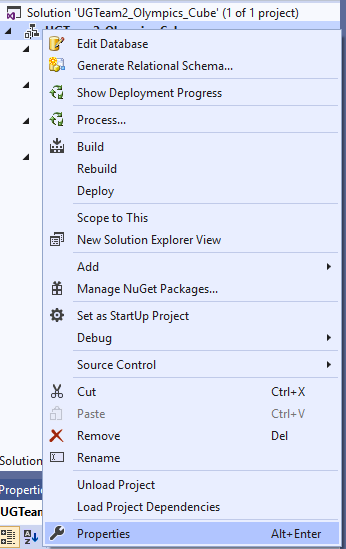


# Cube Development and Deployment in Visual Studio

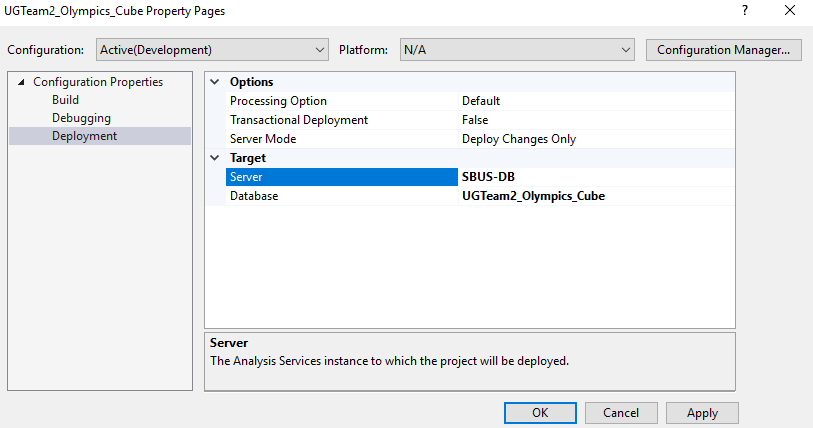
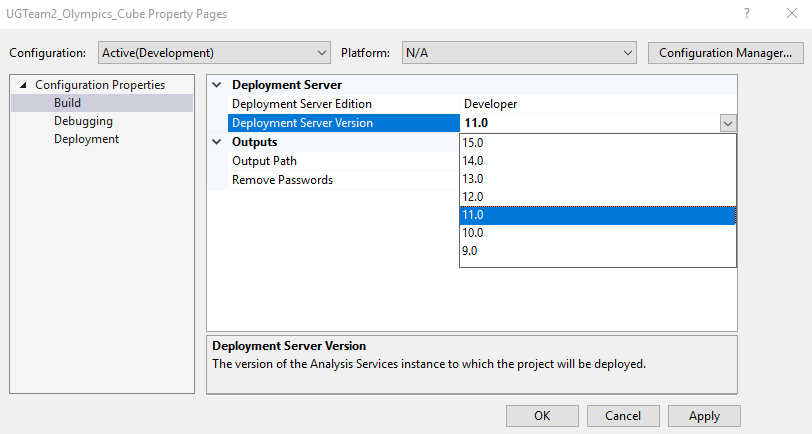
For the development and deployment of the cube we used Visual Studio. The first step in creating the cube is to make a new Analysis Services Multidimensional and Data Mining Project.



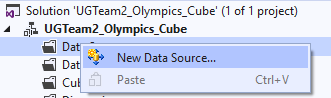
Next, we needed to change the version of the deployment server to 11.0 from 15.0. To do this, changes to the project’s properties need to be made.



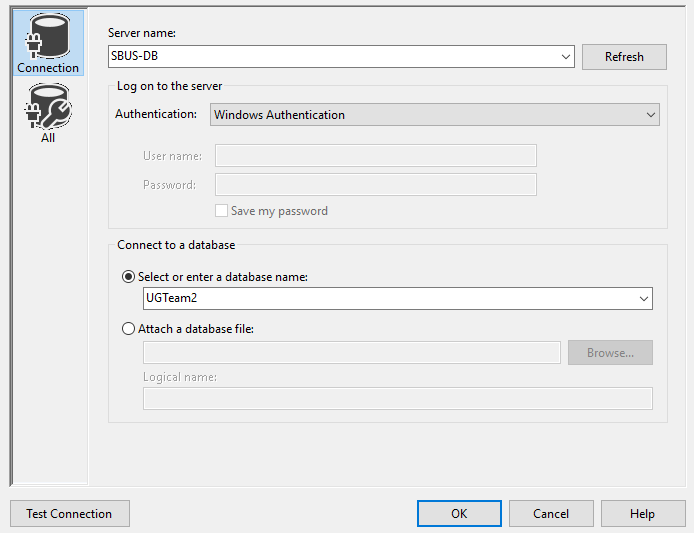
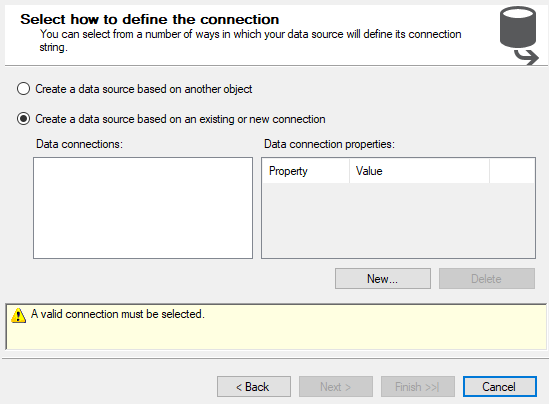
Under the Build section we changed the Deployment Server Version from 15.0 to 11.0. Then, under the Deployment section we changed the server name to SBUS-DB.



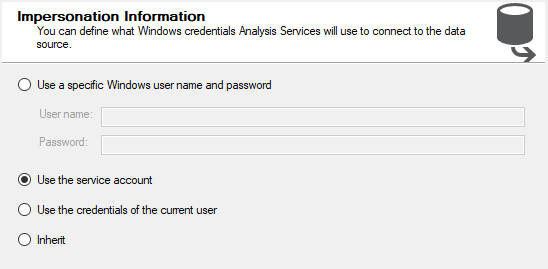
Now we needed to add our data source so that we could create the cube. To do this, right click on Data Sources in the Solution Explorer and select New Data Source.



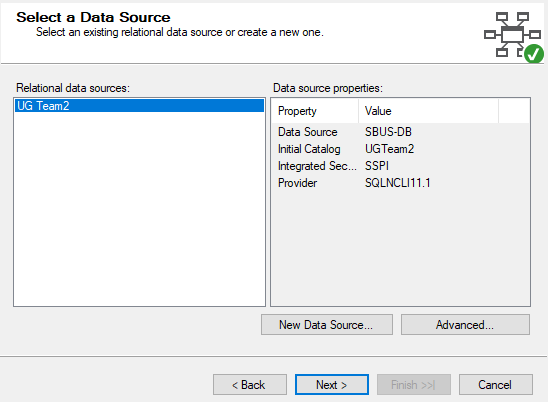
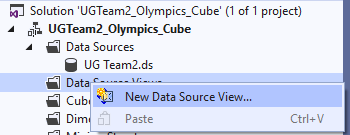
From here we’ll create a new data source based on an existing or new connection. Then select the server where our database is located as well as the database itself.



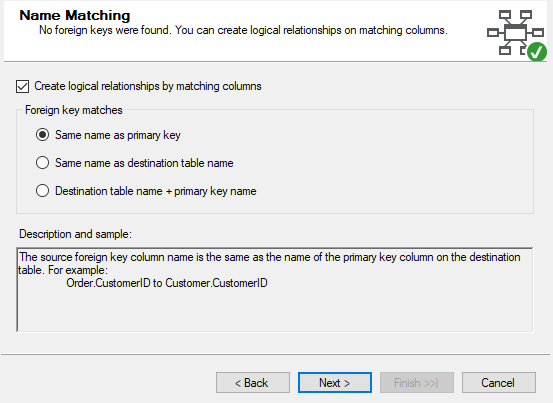
Lastly for our data source, we needed to make sure that we check the “Use this service account” option.



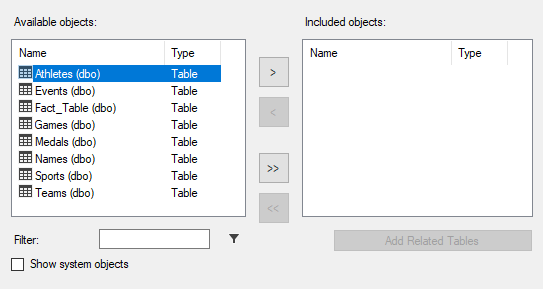
The next step was to set up our Data Source Views. We started by right clicking on Data Source Views in the Solution Explorer and selecting New Data Source View. Then confirmed that our database was selected as the data source.

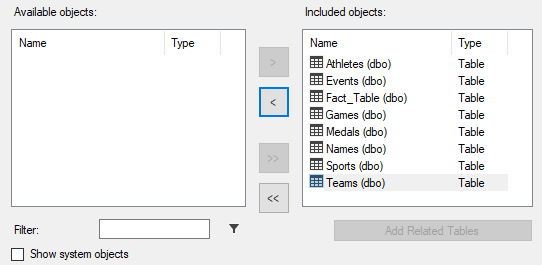


From here we want to create logical relationships by matching columns and make sure that the foreign key matches the same name as the primary key.

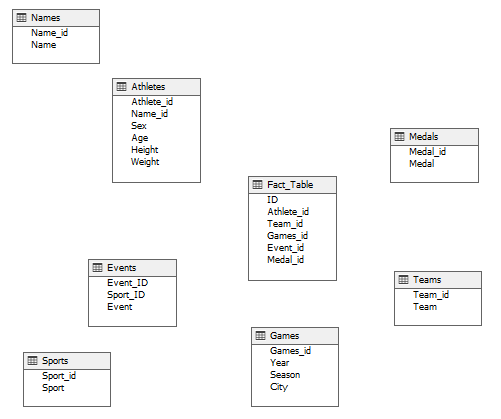


Then we needed to move all of the tables from the left column over to the right. This will allow the tables to be used in the development of the cube.

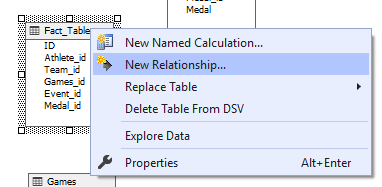


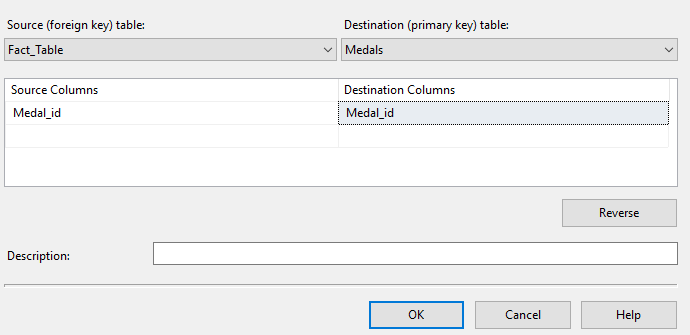


The next thing to was to create the relationships between all of the tables. This was done by opening the data source view we had just created.

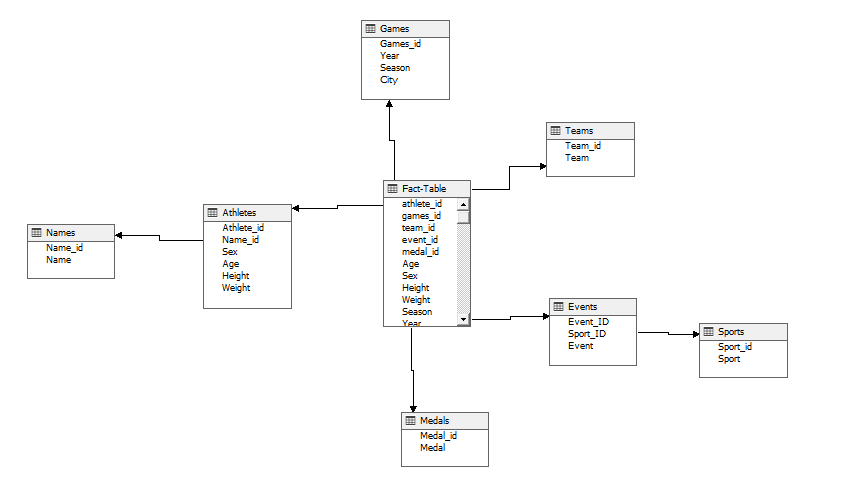


To set up the relationships between the dimensions and the fact table, right click on the fact table and select New Relationship. In the popup that follows, we set the fact table as the foreign key and selected the dimension table we wanted to set the relationship for. At this point we made sure that the corresponding columns that match are selected for each table.

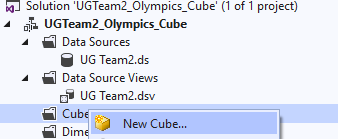




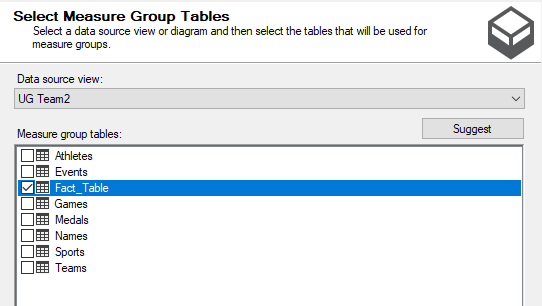
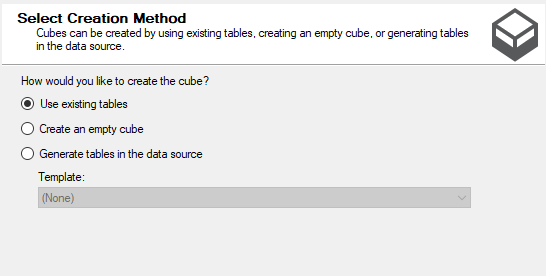
Once this is completed for all tables, the relationships were set, and it was time to generate the cube.



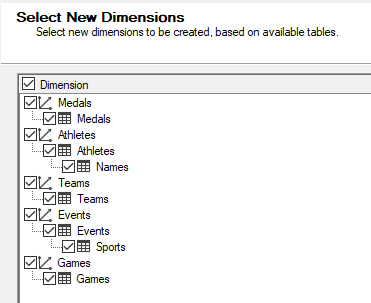
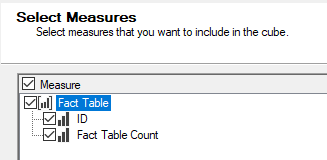
We had to create a new cube by right clicking on Cubes in the Solution Explorer and selecting New Cube.

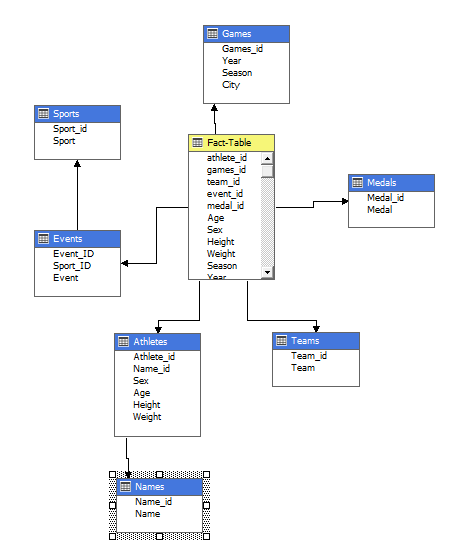


Next, we selected the use existing tables option and made sure to only select the fact table in the Measure Group Tables section.

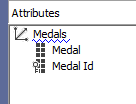
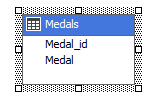
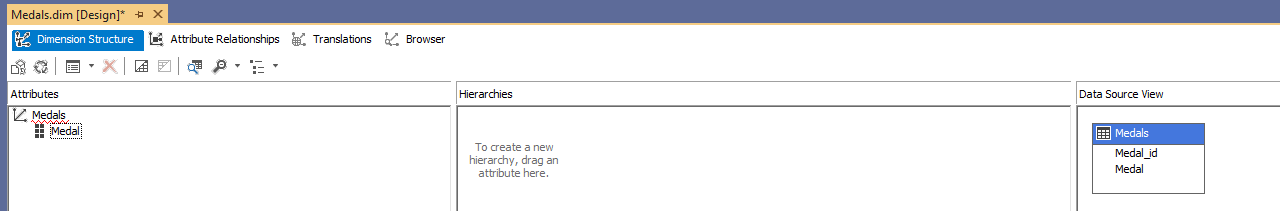


At this point we selected our measures and dimensions, along with their primary keys. Once this was finished our cube was complete.



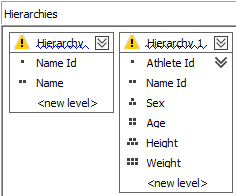


From here, it was time to set up the hierarchies within each dimension. This involved dragging each column from the cube into the attributes panel. Then, the attributes were dragged into the appropriate hierarchy in order to resemble the correct relationship structure. The screenshots below describe the process of correctly setting up the hierarchy.

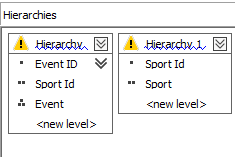


The final hierarchy for each of our dimensions is listed below.

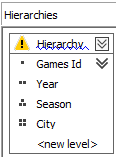
Athletes:



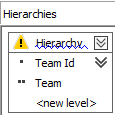
Events:



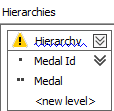
Games:



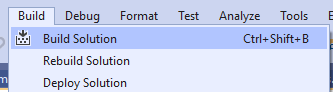
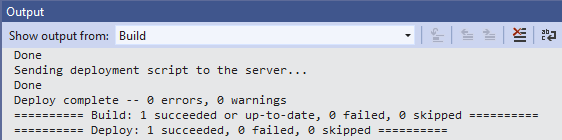
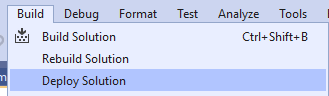
Teams:



Medals:



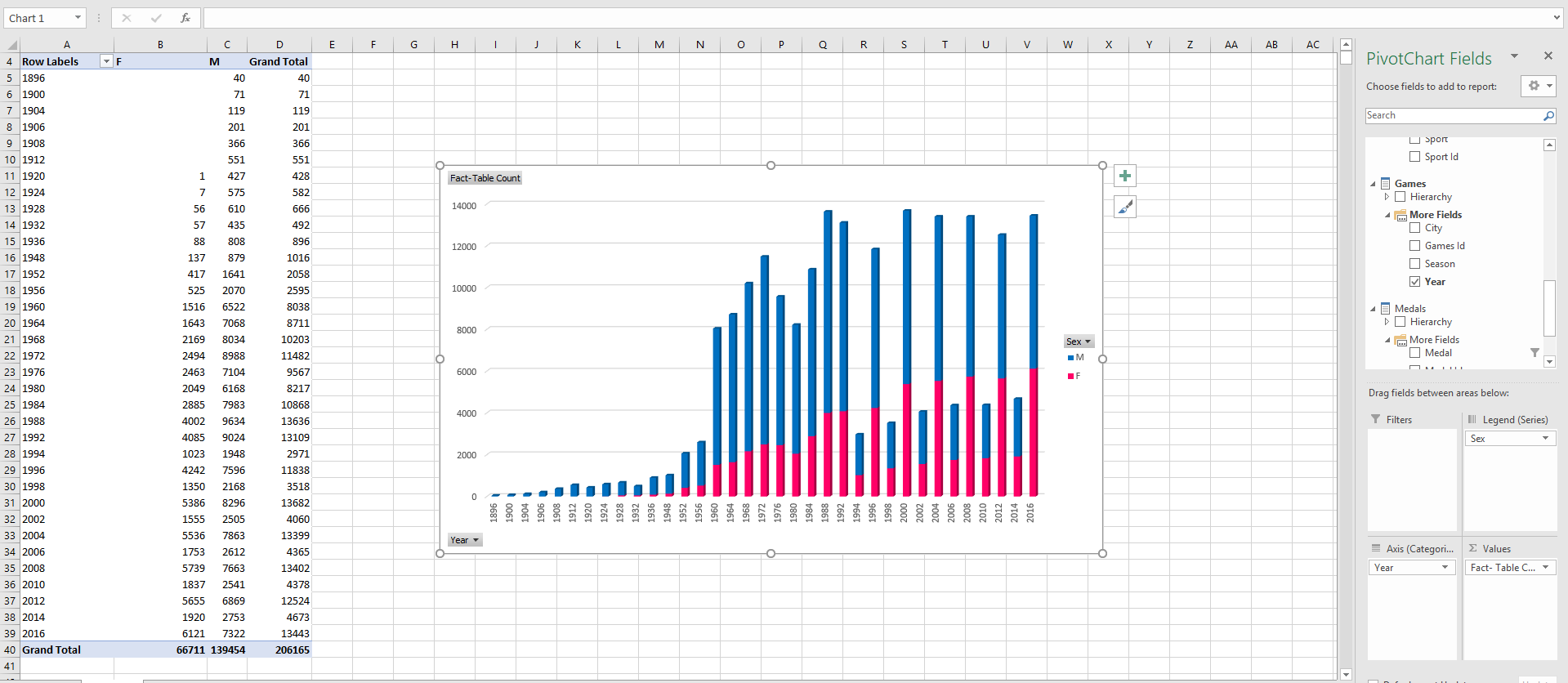
Now that the cube is all set up, the last step was to build and deploy the cube.

# Reports

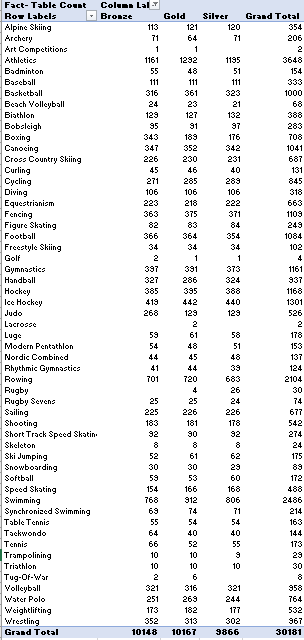
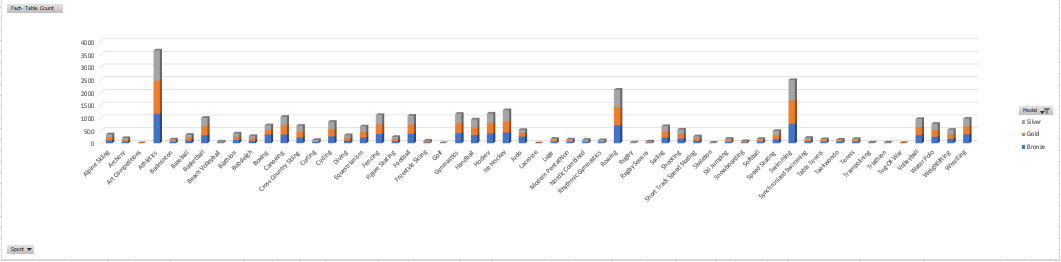
## Report 1:

The first aspect of our database I wanted to explore was the number of participants by gender and year. Looking at the table, you can see that no females competed in Olympic games until 1920. Also, on the graph you can see a sharp decline in the total number of participants in the games starting in 1994.

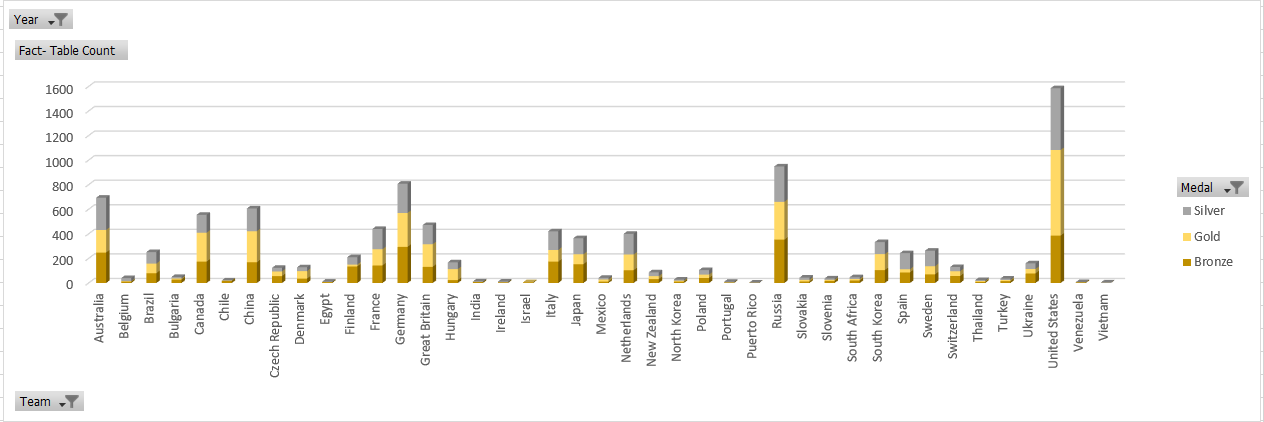


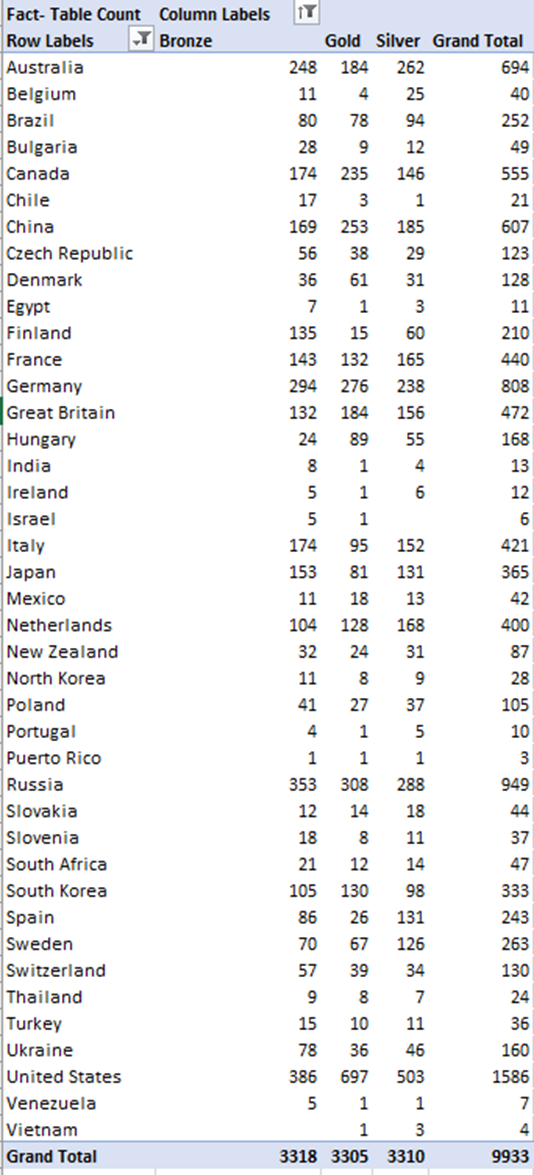
In 1994, they started holding the Winter and Summer Olympics every 2 years, with the season itself happening every 4. 1994 was the first year for the Winter Olympics. Previously, the Olympics were held on the same year every four years, instead of staggered by 2 years like they are now.

## Report 2:

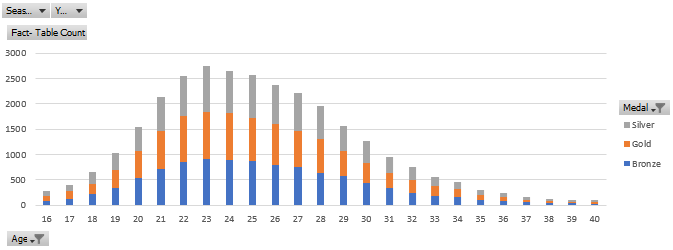
Next, we wanted to look at the medal distribution by event. We were curious to see if there was a correlation between the most popular events and the number of medals awarded. According to this chart, you can see a few large spikes in the data. Those spiked belong to the Athletics event, Rowing and Swimming. Those three events, over the lifetime of the Olympics have given out at least 2000 medals each, with the Athletics event giving out 3,648. These events were the first Olympic events. Examples include the Men’s 10 miles Walk, or the Men’s Stone Throw. Because these events were the only events when the Olympics first began, it makes since that there is a lot of medals awarded. We expect to see this number start to dwindle in the years to come though as events like Gymnastics, Swimming, Rowing, Snowboarding and Speed Skating continue.

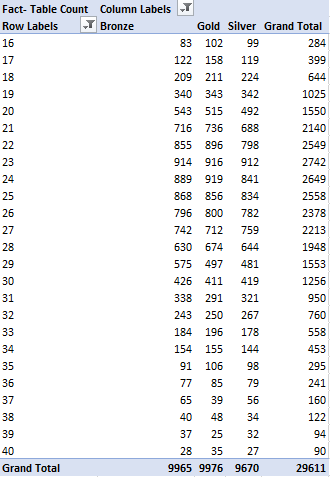
## Report 3:

We also looked at medals by Country. We were looking to find a correlation between the size and population of a country and their participants winning medals in Olympic games.

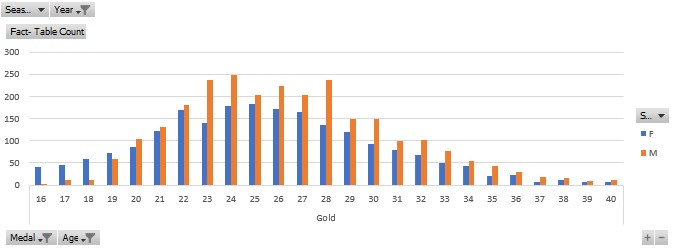
Based on this chart, we can see that the United States has the highest number of total medals awarded over the life of the Olympic Games. You can also see other countries such as Russia, Germany, China, Canada, Australia, Great Britain, France, Italy, Japan and the Netherlands have had many medals awarded. This result was not surprising to us. However, since we were looking at Country size to the number of medals, we were surprised by India. The population of India is over 1 billion people and the number of medals awarded to that country in the history of the Olympics is 13.

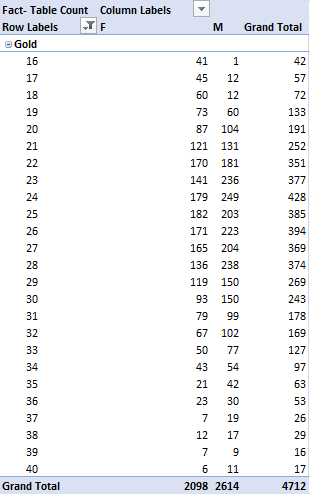
## Report 4:

As part of our analysis we also looked at the total number of medals, and their type, that are won by age. This included both seasons, summer and winter, as well as all male and female athletes. The age range was narrowed down to any athletes that were in between the ages of 16 and 40.

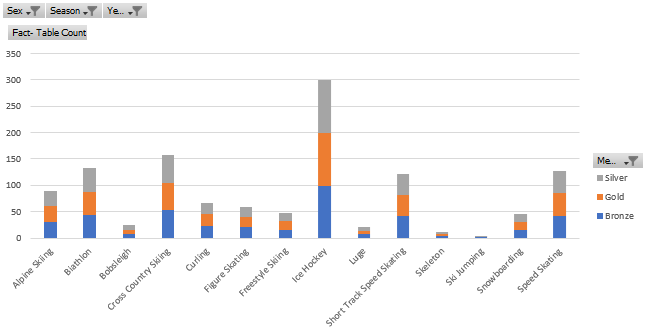
As can be seen by the chart, the age with the greatest number of medals won is 23. Although 23 years old has the greatest number of medals won, the ages just above and below are very close contenders. It is interesting to note that while athletes age 23 have won the most medals, that doesn’t mean that they win the most of every type of medal. Looking at the table we can see that athletes 24 years old have won 919 gold medals compared to 916 gold medals won by those 23 years old.

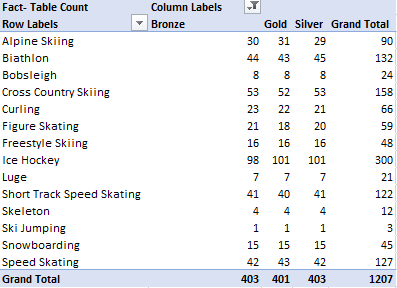
## Report 5:

We also wanted to examine the number of gold medals won by gender while also looking at their age. For this we used the same age range of 16 – 40 years old as we did in Report 4. In addition to this we wanted to look at the Olympics that have taken place more recently due to the fact that there weren’t many females that competed in the earlier Olympic games. To do this we limited the year range to anything that took place in 1994 or later.

 Within this year range, the top age for gold medals was 24 with a total of 428. This including both male and female athletes. To see the exact break down of the number of gold medals won by gender and age you can look at the table that has been provided. In general, at the peak of the curve, or the ages more prone to winning medals as seen in Report 4, males tend to win more gold medals than females. Although these charts don’t hint to a reason, one likely explanation to this could be that there are simply more events for males to compete in. In ages 20 – 22 you can see that while males still have the edge when it comes to gold medals won, the females are not far behind. From this data one could conclude that males have a higher chance to win a gold medal on the basis of age at 24, and the females have a higher chance at the age of 25.

## Report 6:

Lastly, we thought it would be interesting to see which sport women earn the most medals in during the winter Olympics. For this analysis we limited the athlete sex to females only. In addition, the year range was kept at anything taking place 1994 or later. The last restriction was to limit the results to only medals and sports that take place in the winter Olympics.

After examining the chart, it was clear that Ice Hockey has the greatest number of medals won with a total of 300. The next closest was Cross Country Skiing at 158. Popular events for the winter Olympics such as Bobsleigh and Figure Skating were significantly lower than we expected. After some thought, it is assumed that this is due to them being team events with a lower number of athletes per team. A sport like Ice Hockey has many athletes on one team, where Figure Skating can be done solo. Something we found interesting is that there weren’t an equal number of bronze, silver, and gold medals issued for each of the same events. This could be due some of the athletes that earned this medal were removed during the data cleansing process. The biggest takeaway from this analysis is that females stand a greater chance at winning a medal in the winter Olympics if they are competing in a team sport such as Ice Hockey.

# Conclusion

Data warehouses are an efficient way to analyze historical data and present it in a way that can be easily understood. It provides a way to visually see the data that is being worked with. Throughout the process of creating the data warehouse for this project many changes were made. It became clear very early on that any changes made during one step of the process had a great impact on the outcome of the steps to come after. As the process went on, our knowledge of data warehouses increased which opened our eyes to issues that needed to be resolved. This was very evident towards the end of the process when we realized we had a serious issue with our Fact Table that prevented us from being able to create reports. Situations like this happened periodically requiring us to step back and really take a look at the design of our data warehouse. In the end it wasn’t perfect, but lot was learned along the way and we were eventually able to create reports and analyze the data. Six reports were generated looking at different aspects of the original data set that could not have been concluded just by looking at the raw data file. For instance, one of the more interesting things we learned is that women didn’t start to compete in the Olympics until 1920. It also appears that there are significantly less athletes that compete in the Winter Olympics when compared to the Summer Olympics. When looking at the medal distribution across different sports we saw that the sports that have been in the Olympics since the beginning, such as Athletics events, have the highest medals earned. This, in conjunction with the findings in report 6 can possibly tell us a couple of things. First, since these sports have been around the longest, there have been many more competitions for them resulting in the larger number of medals earned. Also, it could be that these sports have events that are team based. This would mean that when a medal is won it could be given to multiple athletes for the same event. A situation like this could lead to inflated numbers of medals won. Most likely it is a combination of the two different scenarios. We also learned that the United States is the most winningest team in the Olympics when looking at total medal counts. The data also shows that the ideal age for winning a medal of any kind was anywhere from 22 – 25 years old, with 23 being the age with the most medals won. When looking at just gold medal winners’ ages by gender we found that the results were slightly different, but still very similar. The age with the most gold medals for males was 24, and 25 years old for females.

# References

chaitanyashah707. “Difference between Data Warehousing and Data Mining.” *GeeksforGeeks*, 19 Aug. 2019, <https://www.geeksforgeeks.org/difference-between-data-warehousing-and-data-mining/>.

Clifton, Christopher. “Data Mining.” *Encyclopædia Britannica*, Encyclopædia Britannica, Inc., 26 Sept. 2017, <https://www.britannica.com/technology/data-mining>.

Rgriffin. “120 Years of Olympic History: Athletes and Results.” *Kaggle*, 15 June 2018, https://www.kaggle.com/heesoo37/120-years-of-olympic-history-athletes-and-results.

Staff, Herzing. “What Is Data Warehousing and Why Is It Important?” *Herzing University*, Herzing University, 14 Nov. 2017, <https://www.herzing.edu/blog/what-data-warehousing-and-why-it-important>.