University of Houston – Clear Lake

Project

IPL Ball-by-Ball 2008-2020

Data Warehousing and Data Mining (ISAM 5332)
Spring (2021)

Instructor:

Charitha Hettiarachchi

Project Group (2):

1) Meher Vishal Malla - 1853791

2) Initha Annamalai Sharavanan - 1894990

3) Sandeep Sagar Muralidhar - 1893894

Contents

Abstract

Data warehousing has become commonplace among associations that attempt to use information technology to gain control. Besides, numerous sellers, having seen this pattern, have started to make different sorts of equipment, programming, and instruments to help data warehouses work all the more successfully. Using data gathered from the dataset which includes **IPL Ball by a ball** from the year **2008** to **2020**, which describes almost every ball bowled by the bowler and the amount of runs batsman scored in particular over again which opponent team. This dataset contains more than **193469** rows and 18 columns.

Keywords: Data warehousing, SQL server 2008 Analysis Services, Star Schema, Dimensional modeling.

Project Overview

The Indian Premier League (IPL) is a Twenty20 cricket league, played by eight different teams based out of eight different Indian cities. The class was established by the Board of Control for Cricket in India (BCCI) in 2007 with funding provided by Zee Entertainment Enterprises.
The IPL is the most-watched cricketing league in the world and 2014 was ranked sixth by average attendance among all sports leagues. In 2010, the IPL became the first sporting event in the world to be broadcast live on Youtube. The brand value of the IPL in 2019 was ₹475 billion, according to Duff & Phelps. According to BCCI, the 2015 IPL season contributed ₹11.5 billion to the GDP of the Indian Economy. There have been 13 seasons so far, where currently Mumbai Indians are champions in the 2020 edition of IPL.

Lastly, we need to provide some Data Mining interviewed to perform analyses and produce reports.

Data Warehousing:

Data Warehousing may be defined as a collection of corporate information and data derived from operational systems and external data sources.

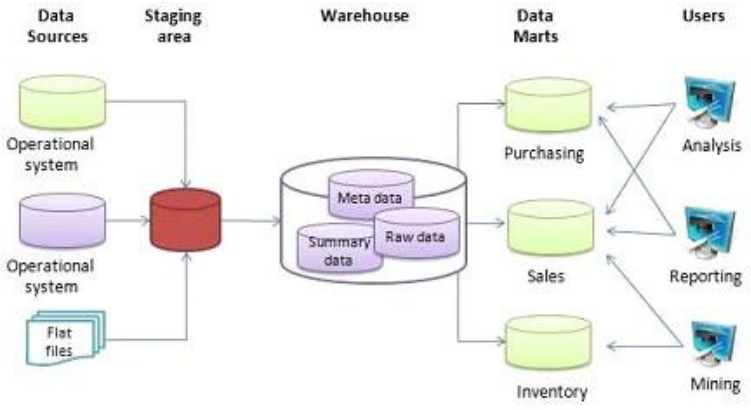


Fig.No. 1: Database Schema

A data warehouse is designed to induce business decisions by allowing data consolidation, analysis, and reporting at different aggregate levels. Data is populated into the DW by extraction, transformation, and loading. Data Warehousing incorporates data stores and conceptual, logical, and physical models to support business goals and end-user information needs. Creating a DW requires mapping data between sources and targets, then capturing the details of the transformation in a metadata repository. The data warehouse provides a single, comprehensive source of current and historical information.

Data Mining:

Data mining is defined as a method or a process that the dimension Non-Striker is used to convert raw data into usable data through which we can analyze information or patterns. It simply analyzes data patterns in large sets of data using software which are particularly designed for processing these kinds of data. Data mining can be used in various industries like science, research, medical, education, and many more. Through data mining, we can obtain information which can be used for business to learn about their customers and serve more efficiently. Moreover, these kinds of techniques include collections of large amounts of data and also need more computer processing. It is also known as Knowledge Discovery in Data (KDD).

The main difference between data mining and data warehousing is that data mining is a process used to process large amounts of data and extract meaningful information where data warehousing is a process of arranging, organizing data in a single database. The two are interconnected, where data warehousing should take place then after data mining can be processed.

The phases involved in Data Mining are as follows:

Fig. No. 2: Data Mining Phases

Introduction:

Difference between Database & Data Warehousing:

I. Database:

A. The data set stores data in columns, rows, and tables, that are maintained by indexes to make them more easily accessible to the users.

B. Databases use Online Transactional Processing (OLTP) to erase, insert, replace, and update a vast number of quick online transactions.

C. Normalization of tables in Database.

D. The database processes everyday transactions within an organization. Databases normally do not contain historical data—current data is all that counts in a normalized relational database.

E. An OLTP (Online Transaction Processing) database supports thousands of concurrent users.

F. There is a one-to-one arrangement with a single application as the source for databases.

II. Data Warehouse

A. A data warehouse is a facility that gathers data from several different sources within an enterprise for reporting and analysis.

B. Data warehouses use Online Analytical Processing (OLAP) which is designed to manage a low number of complex queries on aggregated large historical data sets.

C. DEnormalization of tables in Data Warehouse.

D. Data warehouses usually store historical data by combining copies of transaction data from several sources.

E. Data warehouses support a limited number of concurrent users compared to operational systems.

F. Data warehouses store summarized historical data from a wide variety of applications. There are one or two partnerships between data warehouses and applications that act as data sources.

What is Data Mining?

Data mining is a method of automatically scanning vast data stores to uncover patterns and trends that go beyond basic research. Data mining uses advanced mathematical algorithms to segment data and determine the likelihood of future events and can answer questions that cannot be answered by simple query and reporting techniques. Data mining is often referred to as Information Discovery in Data (KDD).

The key properties of data mining are:

• Automated discovery of a pattern

• Forecasting the possible outcomes
• Production of knowledge that can be used
• Emphasis on broad data sets and databases

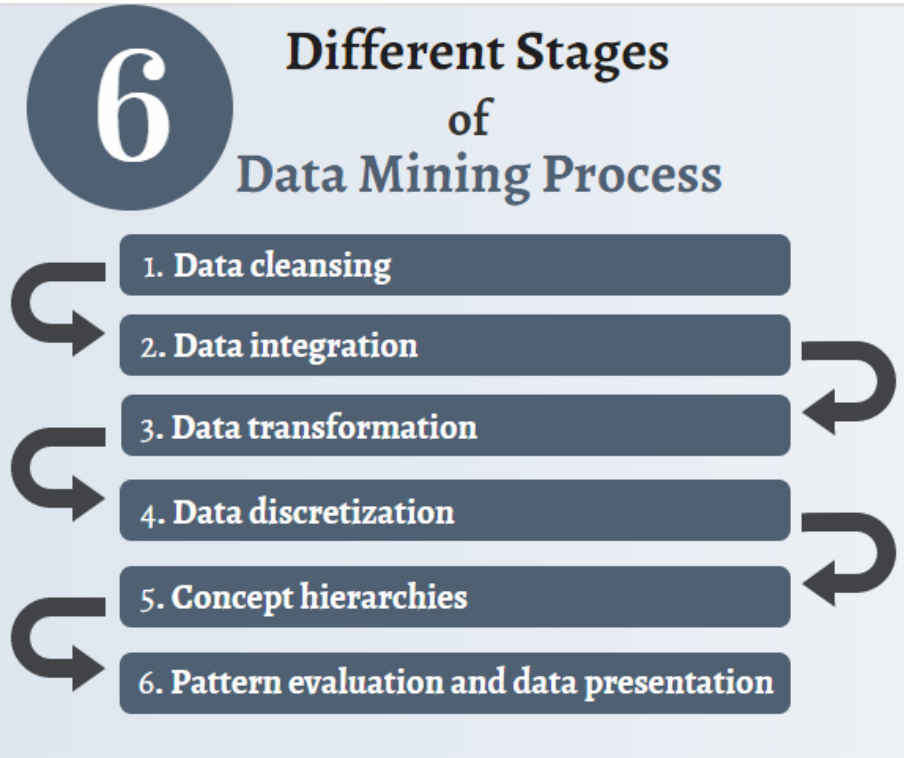


Fig. No. 3: Data Mining Process

What is the Need for Data Warehousing?

A data warehouse is used to store a lot of crude information through which we can analyze the information from that raw data and the data which we store in it can be used uniquely which continuously generates reports. The ordinal database usually performs at a rate of 1/100th of a second while data warehouses have large queries which take minutes to execute.

These are usually used for the execution of huge amounts of data, which uses multiple tables inside it while cross arranging the information which produces dependent reports. The data stored in data warehouses is permanent storage which is used for support, reports, and other BI functions. Some of the needs of a data warehouse are as follows:

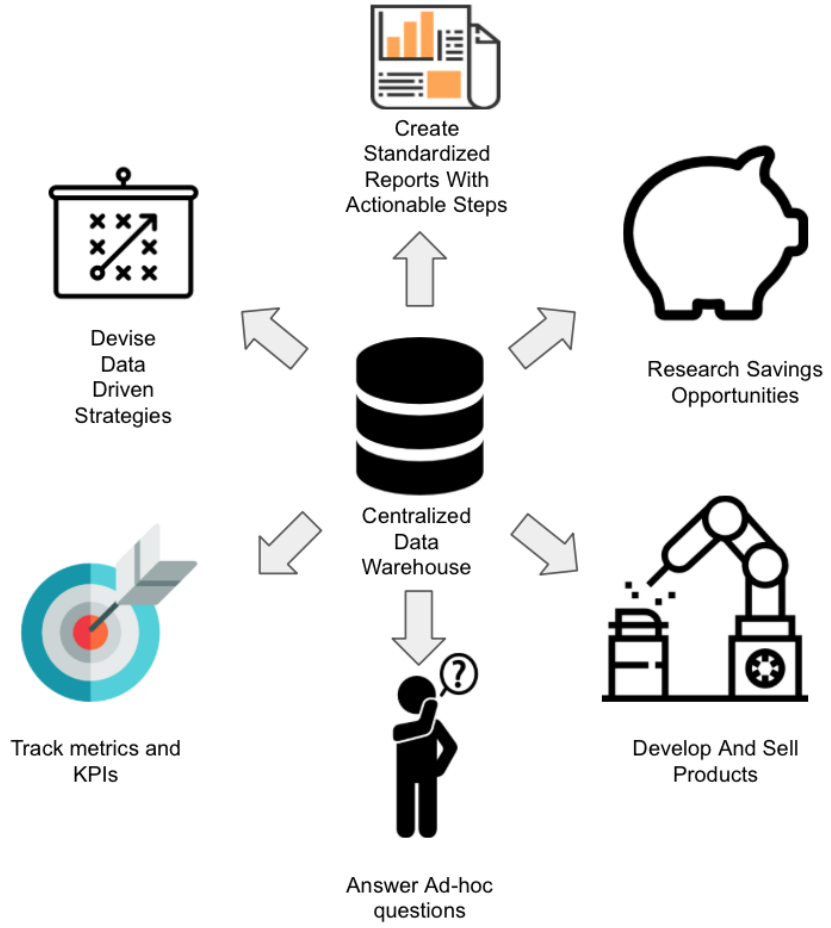


Fig. No. 4: Need for Data Warehouse

1. Drill Down Capability: Enables users to slice and dice to find underlying problems.
2. Data Dashboarding: Data stored in data warehousing can be displayed in interactive dashboards in highly graphical form.
3. Metadata Creation: Storing the data in the warehouse makes it a lot easier for users to understand and interpret the data.
4. Ad Hoc reporting and Analysis: It empowers users to analyze data faster.
5. Data Security: It provides secure access to those that have a legitimate need for specific data.

Methodology

Steps for creating a Data Warehouse:

- **Determine Business Objectives:**

As we observe, as an existing government agency organization, the EAP has strong growth and a good mix of administrative, legislative, collaboration, and support staff. Key decision-makers want to know if the overhead collaborations with various sectors and gathering data can return value to the environment. As the business improves, the leaders need to know if these modes are successful.

- **Collect and Analyze information:**

The only way to collect knowledge about this success is to ask questions. Leaders have sources of knowledge that they use to make decisions. Start with these sources of data. Another part of this compilation and processing step is the understanding of how people gather and process information. A data warehouse can automate a lot of reporting activities, but you can't automate what you haven't found and don't understand.

- **Identify Core Business Processes:**

At this point, we had a good understanding of what business processes we need to correlate with. We defined key performance metrics, such as unit sales, units generated, and gross revenues. Now we need to define the groups that communicate to establish key performance measures. Then we put the main performance metrics into the actual tables and the entities that generate the facts into the dimension tables. Fact tables will share table sizes.

- **Construct a Conceptual Data Model:**

After defining the business processes, we built a conceptual data model. We have defined the subjects that will be expressed as facts tables and the dimensions that will apply to the facts. Clearly define key performance metrics for and business process and decide the format in which the facts are stored. As the information will eventually be aggregated together to form OLAP cubes, the data needs to be in a standardized unit of measure. The method may seem easy, but it's not.

- **Identify Data Sources and Plan Data Transformations:**

As we now have a conceptual data model, we have to identify where critical information lies and how we move it or relate it into a data warehouse structure. Now we transform the data as we move it from one data structure to another. Some transformations are simple mappings to database columns with different names. Some might involve converting the data storage type. We also need to plan when data movement will occur. While the system is accessing the data sources, the performance of those databases will decline precipitously. Schedule the data extraction to minimize its impact on system users (e.g., over a weekend).

- **Set Historical Data Limits:**

When interacting with a data warehouse, we're talking about a vast volume of information that could include historical data. We need to consider and decide how much historical data we want to store in our data warehouse. The data warehouse shall be set up to hold data at varying levels of detail or granularity. This granularity must be consistent across a single data structure, but different data structures with different grains can be connected by shared dimensions.

- **Implementing the plan:**

Following the creation of the schedule, offering that the dimension has a viable basis for forecasting work and scheduling the project. The scope of data warehouse projects is broad, so staggered delivery schedules are essential to keep the project on track. We found that a successful approach is to design the whole warehouse and then execute the data mart component to show what the system is capable of doing.

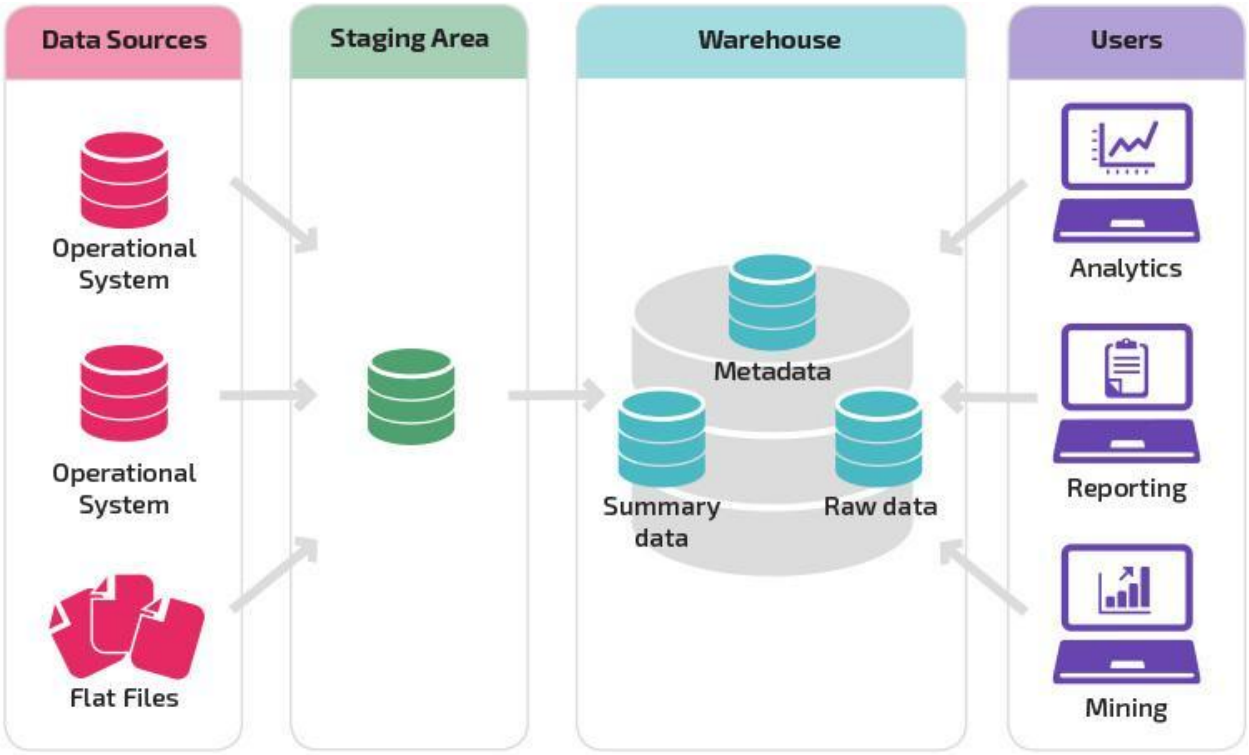


Fig. No. 5: Data Warehouse Architecture

Project Data Set

Using data obtained from **IPL Ball by a ball**, this dataset provides information on different aspects of the over, ball, batsmen, bowler, run_scored, batting_team, bowling_team.
Collected around **July 2020**, it includes almost every ball bowled by each bowler, the player who faced the ball, how much was scored for that ball which helps in keeping track of the runs scored for that particular player for that particular over.

Dataset Link:https: //www.kaggle.com/patrickb1912/ipl-complete-dataset-20082020

This dataset contains more than **193469** rows and **18** columns.

Data Cleansing

Data-cleaning or data-cleaning is the method of identifying and correcting (or deleting) corrupt or incorrect information from a recordset, table, or database and refers to the detection of missing, incorrect, corrupted, or irrelevant parts of the data and the substitution, alteration, or deletion of dirty or gross data. Data cleaning can be done interactively with data wrangling software, or as batch processing by scripting.

After cleansing, a data set should be consistent with other similar data sets in the system. The inconsistencies detected or removed may have been originally caused by user entry errors, by corruption in transmission or storage, or by different data dictionary definitions of similar entities in different stores.

After cleaning, the data set should be compatible with other related data sets in the system. Inconsistencies found or eliminated may have been initially caused by user input errors, transmission or storage corruption, or by different data dictionary meanings of similar entities in different stores. High-quality data needs to follow a set of quality criteria:

• Validity

• Accuracy

• Comprehensiveness

• Accuracy of Uniformity

Data cleaning is an important process that we've had to work on. Data coming from various departments could end up in different formats. In other words, different data sources produce different output types. Until these data are processed or made available in the data center, they must be converted and cleaned. The transformation will allow us to bring data to a common format around the data warehouse. While going through the data in excel and during the extraction process, it was found that many fields were required only for the operating system,

but not in the data warehouse. This phase leads us to clean up the details. We made use of Excel to delete the extraneous data. Extraneous data was deleted and cleaner data was sent to the data warehouse.

id	inning	over	ball	batsman	non_striker	bowler	batsman_runs	extra_runs	total_runs	non_boundary	is_wicket	dismissal_kind	player_dismissed	fielder	extras_type
335982	1	6	5	RT Ponting	BB McCullum	AA Noffke	1	0	1	0	0	NA	NA	NA	NA
335982	1	6	6	BB McCullum	RT Ponting	AA Noffke	1	0	1	0	0	NA	NA	NA	NA
335982	1	7	1	BB McCullum	RT Ponting	Z Khan	0	0	0	0	0	NA	NA	NA	NA
335982	1	7	2	BB McCullum	RT Ponting	Z Khan	1	0	1	0	0	NA	NA	NA	NA
335982	1	7	3	RT Ponting	BB McCullum	Z Khan	1	0	1	0	0	NA	NA	NA	NA
335982	1	7	4	BB McCullum	RT Ponting	Z Khan	1	0	1	0	0	NA	NA	NA	NA
335982	1	7	5	RT Ponting	BB McCullum	Z Khan	1	0	1	0	0	NA	NA	NA	NA
335982	1	7	6	BB McCullum	RT Ponting	Z Khan	1	0	1	0	0	NA	NA	NA	NA
335982	1	8	1	BB McCullum	RT Ponting	JH Kallis	0	0	0	0	0	NA	NA	NA	NA
335982	1	8	2	BB McCullum	RT Ponting	JH Kallis	0	0	0	0	0	NA	NA	NA	NA
335982	1	8	3	BB McCullum	RT Ponting	JH Kallis	0	0	0	0	0	NA	NA	NA	NA
335982	1	8	4	BB McCullum	RT Ponting	JH Kallis	1	0	1	0	0	NA	NA	NA	NA
335982	1	8	5	RT Ponting	BB McCullum	JH Kallis	1	0	1	0	0	NA	NA	NA	NA
335982	1	8	6	BB McCullum	RT Ponting	JH Kallis	2	0	2	0	0	NA	NA	NA	NA
335982	1	9	1	RT Ponting	BB McCullum	SB Joshi	1	0	1	0	0	NA	NA	NA	NA
335982	1	9	2	BB McCullum	RT Ponting	SB Joshi	1	0	1	0	0	NA	NA	NA	NA
335982	1	9	3	RT Ponting	BB McCullum	SB Joshi	1	0	1	0	0	NA	NA	NA	NA
335982	1	9	4	BB McCullum	RT Ponting	SB Joshi	0	0	0	0	0	NA	NA	NA	NA
335982	1	9	5	BB McCullum	RT Ponting	SB Joshi	6	0	6	0	0	NA	NA	NA	NA
335982	1	9	6	BB McCullum	RT Ponting	SB Joshi	1	0	1	0	0	NA	NA	NA	NA
335982	1	10	1	BB McCullum	RT Ponting	JH Kallis	1	0	1	0	0	NA	NA	NA	NA
335982	1	10	2	RT Ponting	BB McCullum	JH Kallis	4	0	4	0	0	NA	NA	NA	NA
335982	1	10	3	RT Ponting	BB McCullum	JH Kallis	0	0	0	0	0	NA	NA	NA	NA
335982	1	10	4	RT Ponting	BB McCullum	JH Kallis	6	0	6	0	0	NA	NA	NA	NA
335982	1	10	5	RT Ponting	BB McCullum	JH Kallis	0	0	0	0	0	NA	NA	NA	NA
335982	1	10	6	RT Ponting	BB McCullum	JH Kallis	0	4	4	0	0	NA	NA	NA	byes
335982	1	11	1	BB McCullum	RT Ponting	SB Joshi	0	0	0	0	0	NA	NA	NA	NA
335982	1	11	2	BB McCullum	RT Ponting	SB Joshi	6	0	6	0	0	NA	NA	NA	NA
335982	1	11	3	BB McCullum	RT Ponting	SB Joshi	2	0	2	0	0	NA	NA	NA	NA
335982	1	11	4	BB McCullum	RT Ponting	SB Joshi	1	0	1	0	0	NA	NA	NA	NA
335982	1	11	5	RT Ponting	BB McCullum	SB Joshi	0	0	0	0	0	NA	NA	NA	NA
335982	1	11	6	RT Ponting	BB McCullum	SB Joshi	1	0	1	0	0	NA	NA	NA	NA

Fig. No. 6.1: Uncleaned Data

Data Cleaning Process Used:

Python is a great language for data processing, mainly because of the fantastic ecosystem of data-centric Python packages. Pandas is one of those bundles, making it much easier to import and analyze data.
Pandas provide data analysts with a way to delete and filter the data frame they use. Drop process (). Use this method to delete rows or columns using the index mark or column name.

General Steps:

1. Import pandas: import pandas as pd
2. Import csv into a Pandas DataFrame object flights = pd.read_csv('filename.csv')
3. Check the shape of your data in (rows, columns) format filename. shape

4. (Optional) Check for all null values in your dataset. This will return a boolean stating if each cell is null. This can take a long time and may not be particularly useful in a very large dataset. filename.IsNull()

5. Explore how many null values are in each column of your dataset filename.IsNull().sum()

6. (Optional) Check how many null values are in a specific column, substituting the name of your column in string form where it says 'col' filename [col].isnull().sum()

Remove rows with empty values:

1. If there are only a few null values and you know that deleting values will not cause adverse effects on your result, remove them from your DataFrame and store that in a new DataFrame* `modified filename = filename. dropna().`

2. Verify that you no longer have any null values by running a that the dimension `modified filename.IsNull(). sum ()`

3. Save your modified dataset to a new CSV, replacing 'modifiedFlights.csv' with whatever you would like to name your new file. `modified filename.to_csv(modifiedFileName.csv',index=False).`

```
In [8]:  1  ipl.isnull().any()
         executed in 132ms, finished 00:53:43 2021-04-30

Out[8]: id                False
        inning            False
        over              False
        ball              False
        batsman           False
        non_striker       False
        bowler            False
        batsman_runs      False
        extra_runs        False
        total_runs        False
        non_boundary      False
        is_wicket         False
        dismissal_kind     True
        player_dismissed   True
        fielder            True
        extras_type        True
        batting_team      False
        bowling_team       True
        dtype: bool
```

Fig. No.7.1: Data Cleaning Process

```
In [4]:
1  ipl.drop(columns=['id','inning','non_boundary','is_wicket'
2                    ,'fielder','extras_type','player_dismissed',
3                    'dismissal_kind'], inplace=True)
executed in 42ms, finished 18:20:52 2021-03-30
```

Fig. No.7.2: Data Cleaning Process

```
In [11]:
1  ipl.head(10)
executed in 23ms, finished 01:03:18 2021-04-30
```

Out[11]:

	over	ball	batsman	non_striker	bowler	batsman_runs	extra_runs	total_runs	batting_team	bowling_team
0	6	5	RT Ponting	BB McCullum	AA Noffke	1	0	1	Kolkata Knight Riders	Royal Challengers Bangalore
1	6	6	BB McCullum	RT Ponting	AA Noffke	1	0	1	Kolkata Knight Riders	Royal Challengers Bangalore
2	7	1	BB McCullum	RT Ponting	Z Khan	0	0	0	Kolkata Knight Riders	Royal Challengers Bangalore
3	7	2	BB McCullum	RT Ponting	Z Khan	1	0	1	Kolkata Knight Riders	Royal Challengers Bangalore
4	7	3	RT Ponting	BB McCullum	Z Khan	1	0	1	Kolkata Knight Riders	Royal Challengers Bangalore
5	7	4	BB McCullum	RT Ponting	Z Khan	1	0	1	Kolkata Knight Riders	Royal Challengers Bangalore
6	7	5	RT Ponting	BB McCullum	Z Khan	1	0	1	Kolkata Knight Riders	Royal Challengers Bangalore
7	7	6	BB McCullum	RT Ponting	Z Khan	1	0	1	Kolkata Knight Riders	Royal Challengers Bangalore
8	8	1	BB McCullum	RT Ponting	JH Kallis	0	0	0	Kolkata Knight Riders	Royal Challengers Bangalore
9	8	2	BB McCullum	RT Ponting	JH Kallis	0	0	0	Kolkata Knight Riders	Royal Challengers Bangalore

```
In [17]:
1  ipl.isnull().sum()
executed in 56ms, finished 18:20:53 2021-03-30

Out[17]: over            0
         ball            0
         batsman         0
         non_striker     0
         bowler          0
         batsman_runs    0
         extra_runs      0
         total_runs      0
         batting_team    0
         dtype: int64
```

Fig. No.7.3: Cleaned Data (Attributes Used)

Data Transfer to MS Access

- Microsoft Access is a Microsoft Database Management System (DBMS) that incorporates the relational Microsoft Jet Database Engine with the graphical user interface and software development tools.
- Using certain steps, we can import different formats into MS Access. The following are the most commonly used import formats for info.
- Microsoft Office Excel
- Microsoft Office Access
- ODBC Databases (For example, SQL Server)
- Text files (delimited or fixed-width)
- XML File

Steps to Import file to MS Access:

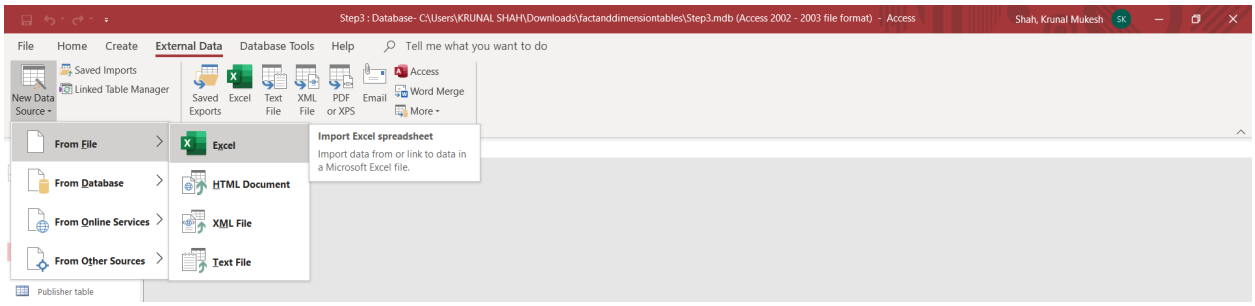


Fig. No. 9: Import Data

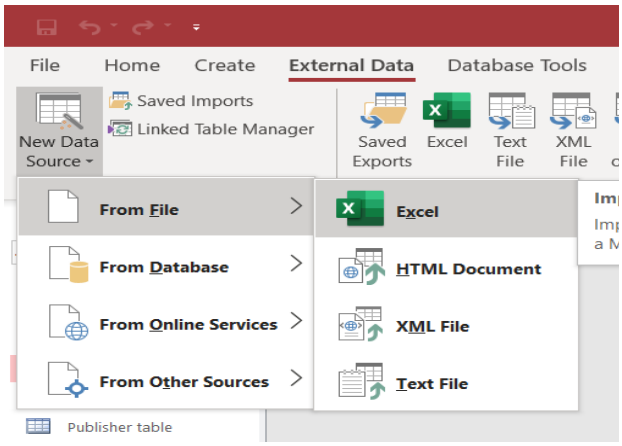


Fig. No. 10: Select Excel File

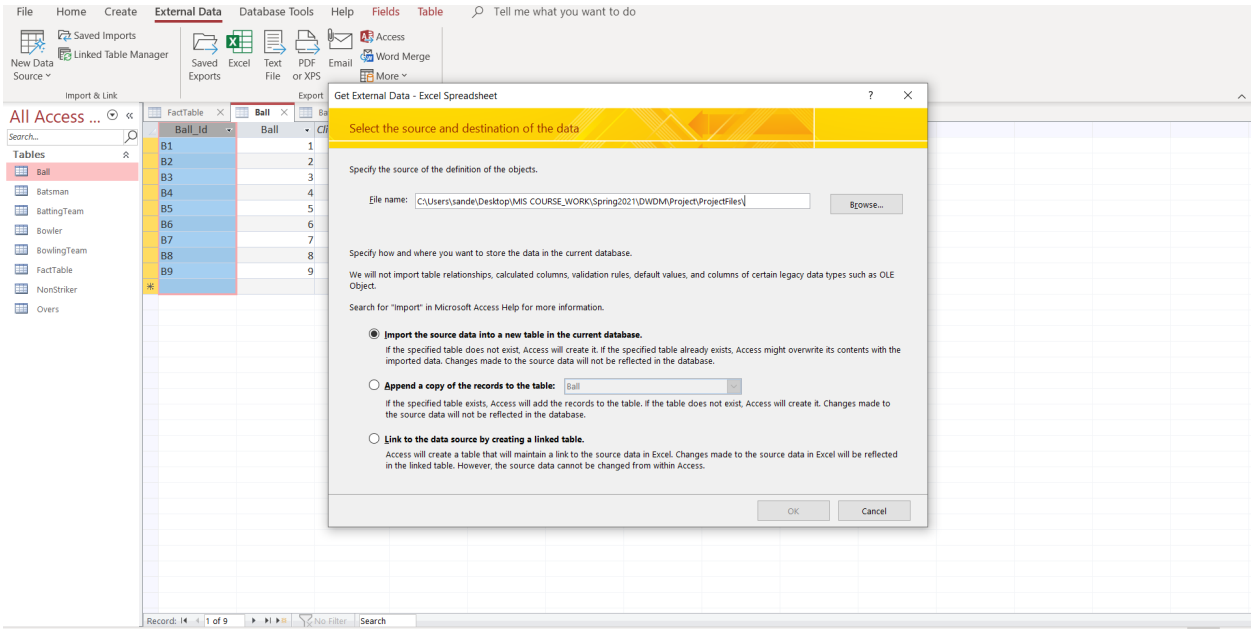
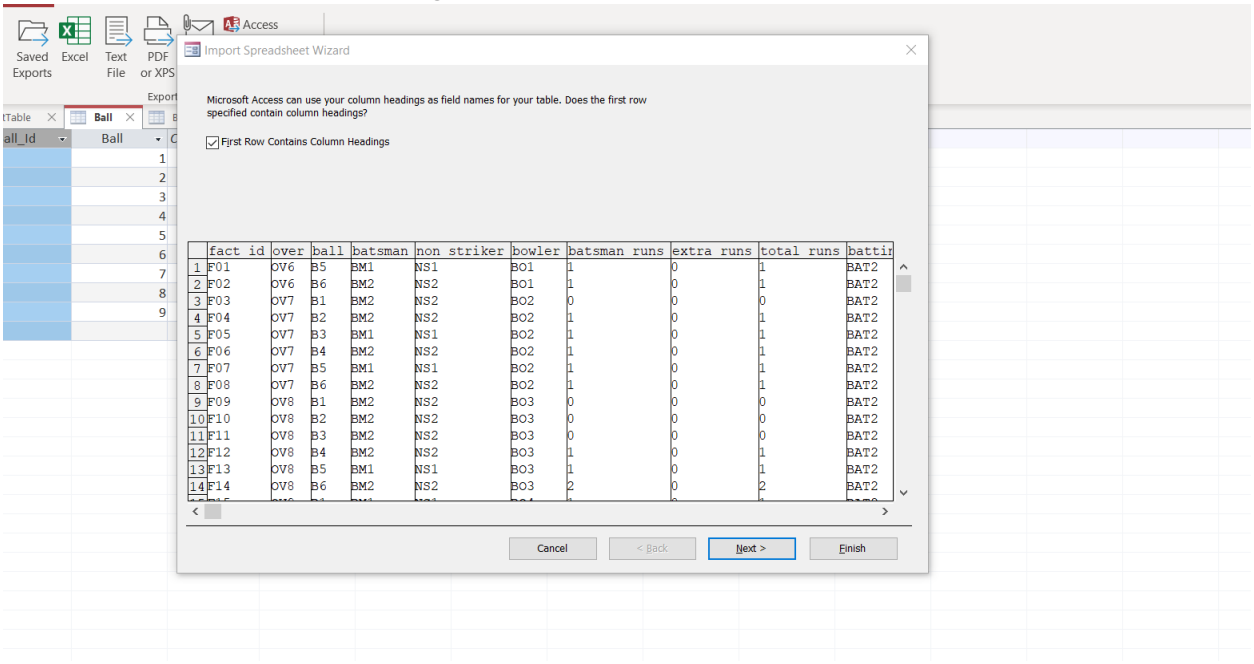


Fig. No. 11: Select Excel File Destination



	fact_id	over	ball	batsman	non_striker	bowler	batsman_runs	extra_runs	total_runs	battir
1	F01	OV6	B5	BM1	NS1	BO1	1	0	1	BAT2
2	F02	OV6	B6	BM2	NS2	BO1	1	0	1	BAT2
3	F03	OV7	B1	BM2	NS2	BO2	0	0	0	BAT2
4	F04	OV7	B2	BM2	NS2	BO2	1	0	1	BAT2
5	F05	OV7	B3	BM1	NS1	BO2	1	0	1	BAT2
6	F06	OV7	B4	BM2	NS2	BO2	1	0	1	BAT2
7	F07	OV7	B5	BM1	NS1	BO2	1	0	1	BAT2
8	F08	OV7	B6	BM2	NS2	BO2	1	0	1	BAT2
9	F09	OV8	B1	BM2	NS2	BO3	0	0	0	BAT2
10	F10	OV8	B2	BM2	NS2	BO3	0	0	0	BAT2
11	F11	OV8	B3	BM2	NS2	BO3	0	0	0	BAT2
12	F12	OV8	B4	BM2	NS2	BO3	1	0	1	BAT2
13	F13	OV8	B5	BM1	NS1	BO3	1	0	1	BAT2
14	F14	OV8	B6	BM2	NS2	BO3	2	0	2	BAT2

Fig. No. 12: Select Show Worksheet

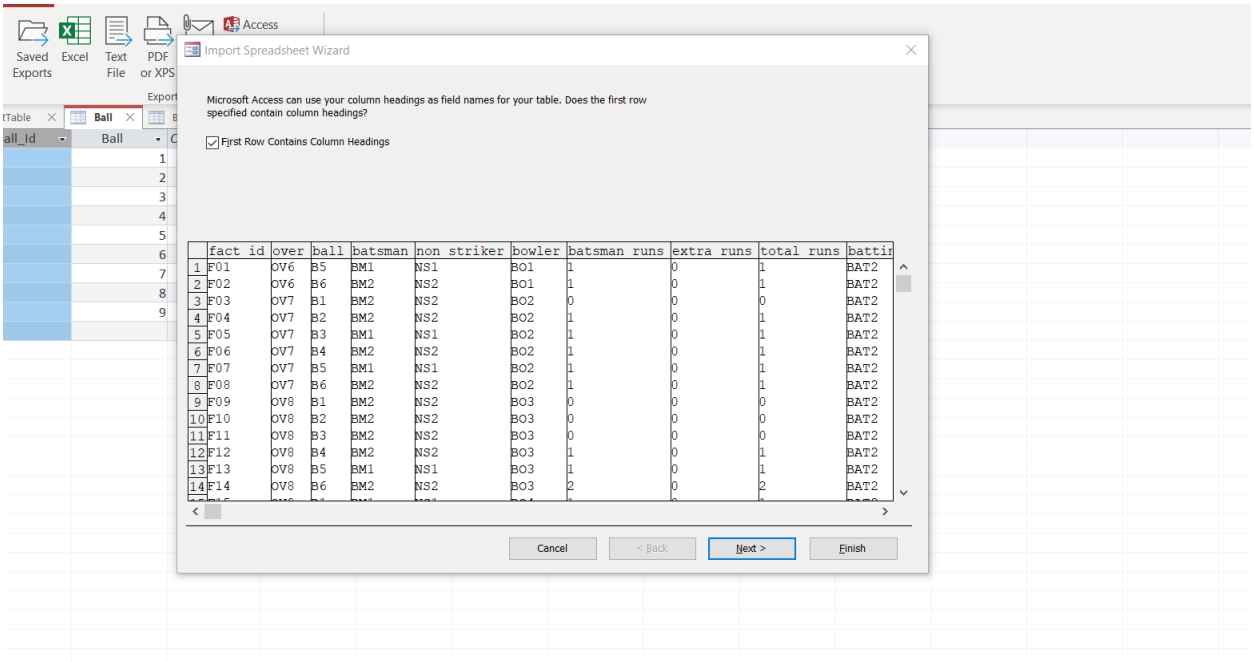
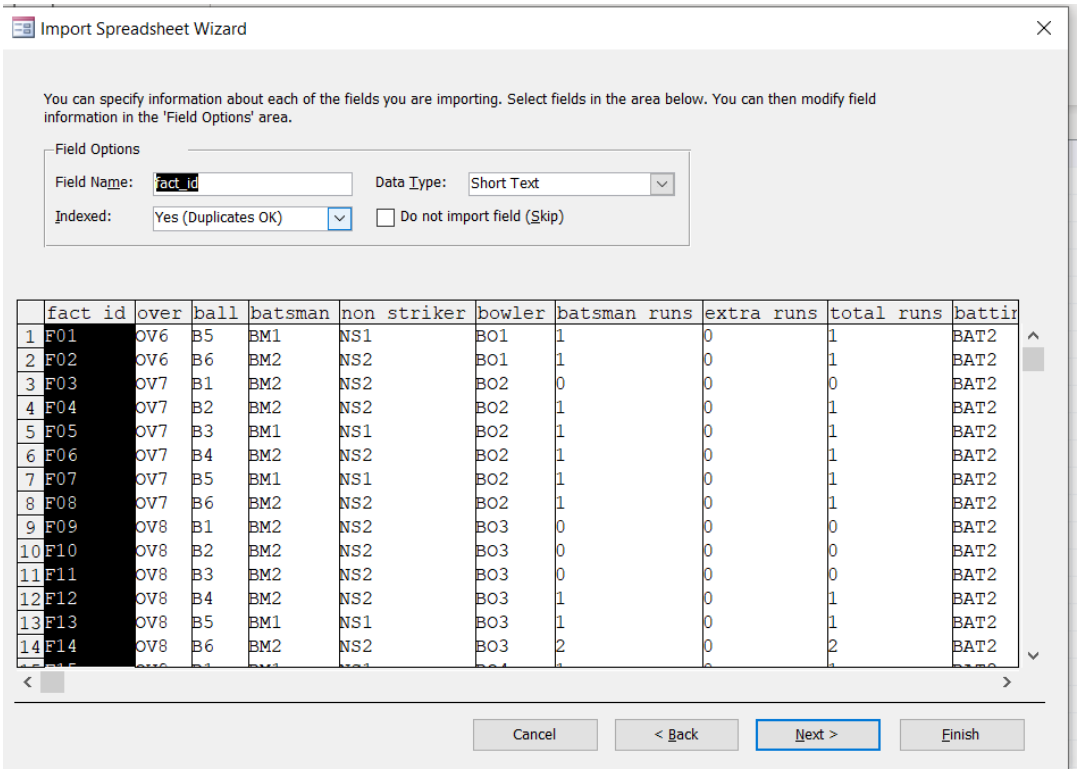
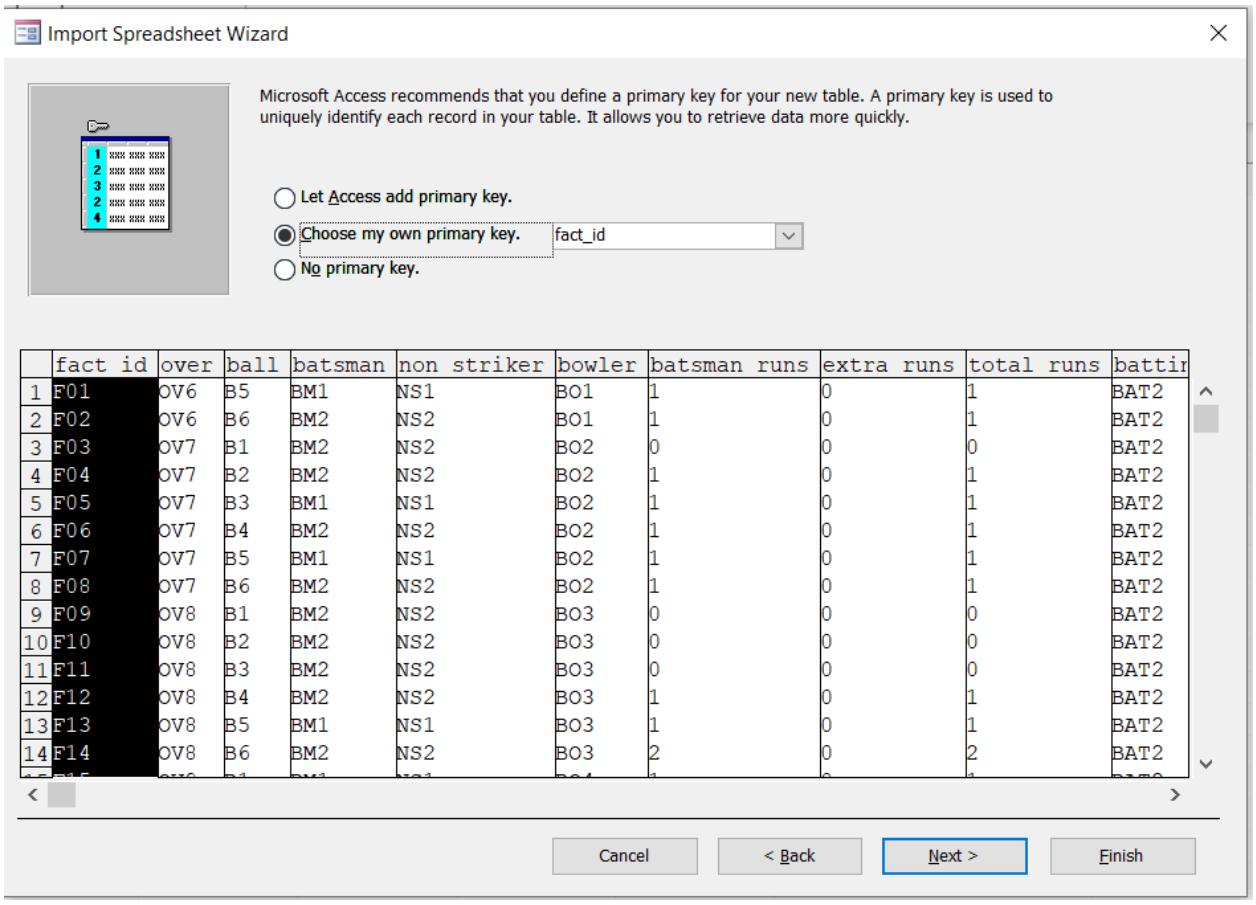


Fig. No. 13: Select First Row as Heading



	fact id	over	ball	batsman	non striker	bowler	batsman runs	extra runs	total runs	battir
1	F01	OV6	B5	BM1	NS1	BO1	1	0	1	BAT2
2	F02	OV6	B6	BM2	NS2	BO1	1	0	1	BAT2
3	F03	OV7	B1	BM2	NS2	BO2	0	0	0	BAT2
4	F04	OV7	B2	BM2	NS2	BO2	1	0	1	BAT2
5	F05	OV7	B3	BM1	NS1	BO2	1	0	1	BAT2
6	F06	OV7	B4	BM2	NS2	BO2	1	0	1	BAT2
7	F07	OV7	B5	BM1	NS1	BO2	1	0	1	BAT2
8	F08	OV7	B6	BM2	NS2	BO2	1	0	1	BAT2
9	F09	OV8	B1	BM2	NS2	BO3	0	0	0	BAT2
10	F10	OV8	B2	BM2	NS2	BO3	0	0	0	BAT2
11	F11	OV8	B3	BM2	NS2	BO3	0	0	0	BAT2
12	F12	OV8	B4	BM2	NS2	BO3	1	0	1	BAT2
13	F13	OV8	B5	BM1	NS1	BO3	1	0	1	BAT2
14	F14	OV8	B6	BM2	NS2	BO3	2	0	2	BAT2

Fig. No. 14: Select Field Options



	fact id	over	ball	batsman	non striker	bowler	batsman runs	extra runs	total runs	battir
1	F01	OV6	B5	BM1	NS1	BO1	1	0	1	BAT2
2	F02	OV6	B6	BM2	NS2	BO1	1	0	1	BAT2
3	F03	OV7	B1	BM2	NS2	BO2	0	0	0	BAT2
4	F04	OV7	B2	BM2	NS2	BO2	1	0	1	BAT2
5	F05	OV7	B3	BM1	NS1	BO2	1	0	1	BAT2
6	F06	OV7	B4	BM2	NS2	BO2	1	0	1	BAT2
7	F07	OV7	B5	BM1	NS1	BO2	1	0	1	BAT2
8	F08	OV7	B6	BM2	NS2	BO2	1	0	1	BAT2
9	F09	OV8	B1	BM2	NS2	BO3	0	0	0	BAT2
10	F10	OV8	B2	BM2	NS2	BO3	0	0	0	BAT2
11	F11	OV8	B3	BM2	NS2	BO3	0	0	0	BAT2
12	F12	OV8	B4	BM2	NS2	BO3	1	0	1	BAT2
13	F13	OV8	B5	BM1	NS1	BO3	1	0	1	BAT2
14	F14	OV8	B6	BM2	NS2	BO3	2	0	2	BAT2

Fig. No. 15: Select our Primary Key

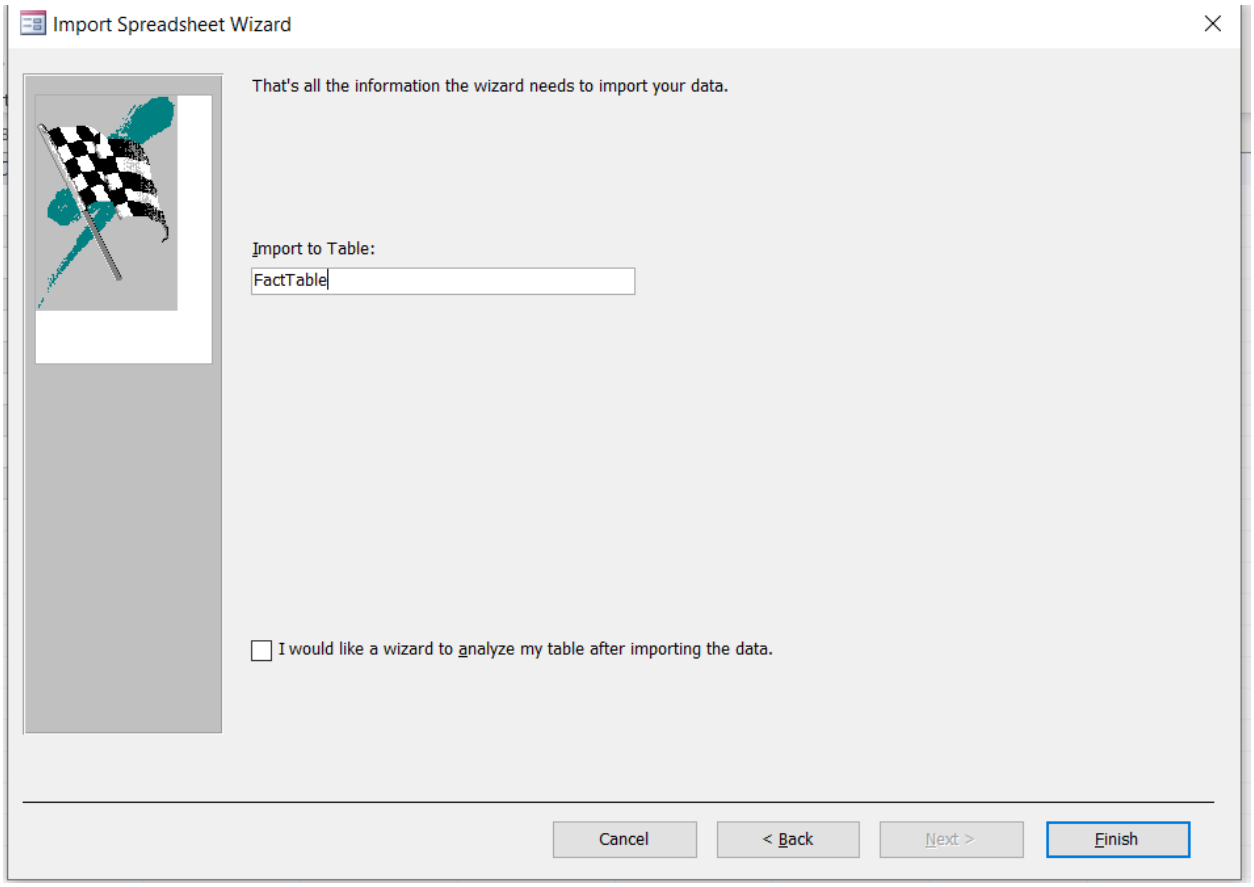


Fig. No. 16: Rename File

Data in MS Access After Import:

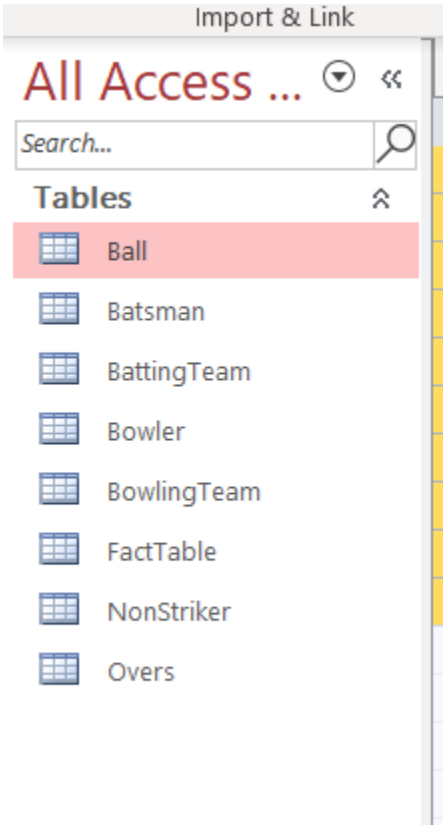


Fig. No. 17: Imported Tables in MS Access

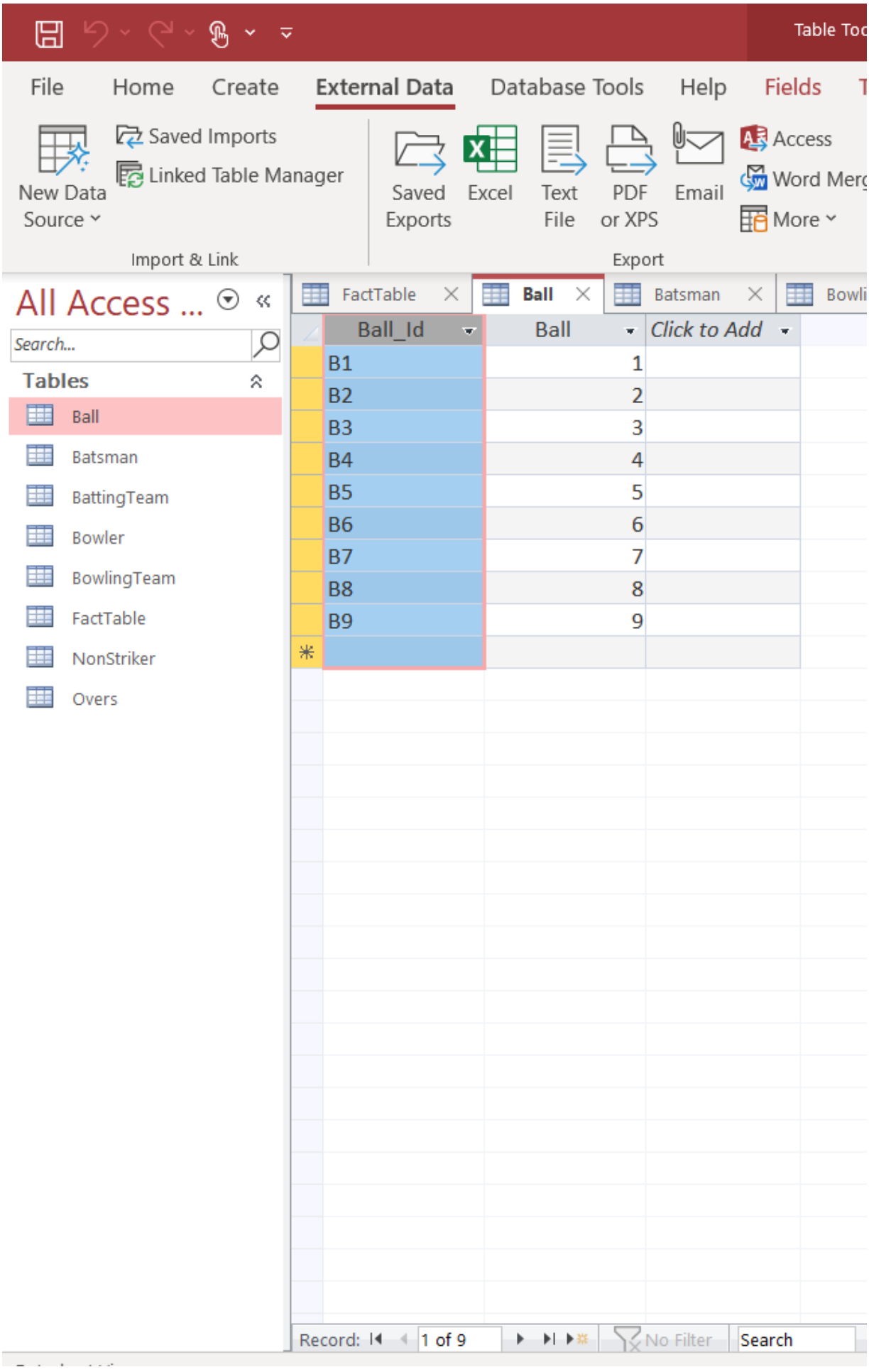


Table Too
File
Home
Create
External Data
Database Tools
Help
Fields
New Data Source
Saved Imports
Linked Table Manager
Import & Link
Saved Exports
Excel
Text File
PDF or XPS
Email
Access
Word Merg
More
Export
All Access ...
Search...
Tables
Ball
Batsman
BattingTeam
Bowler
BowlingTeam
FactTable
NonStriker
Overs
FactTable
Ball
Batsman
Bowli
Ball_Id
Ball
Click to Add
B1 1
B2 2
B3 3
B4 4
B5 5
B6 6
B7 7
B8 8
B9 9
Record: 1 of 9
No Filter
Search

Fig. No. 18: Ball Dimension Table

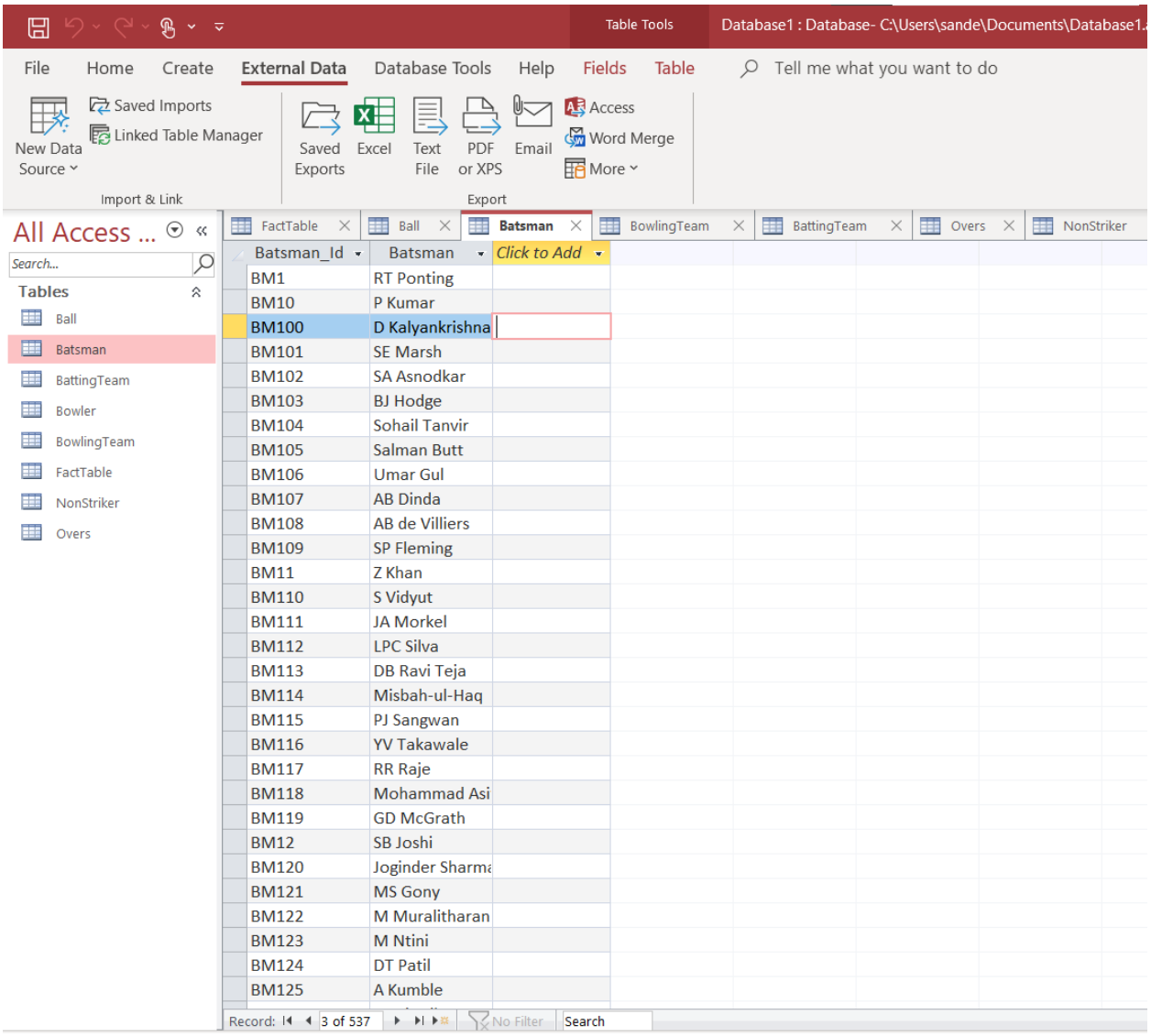


Fig. No. 19: Batsman Dimension Table

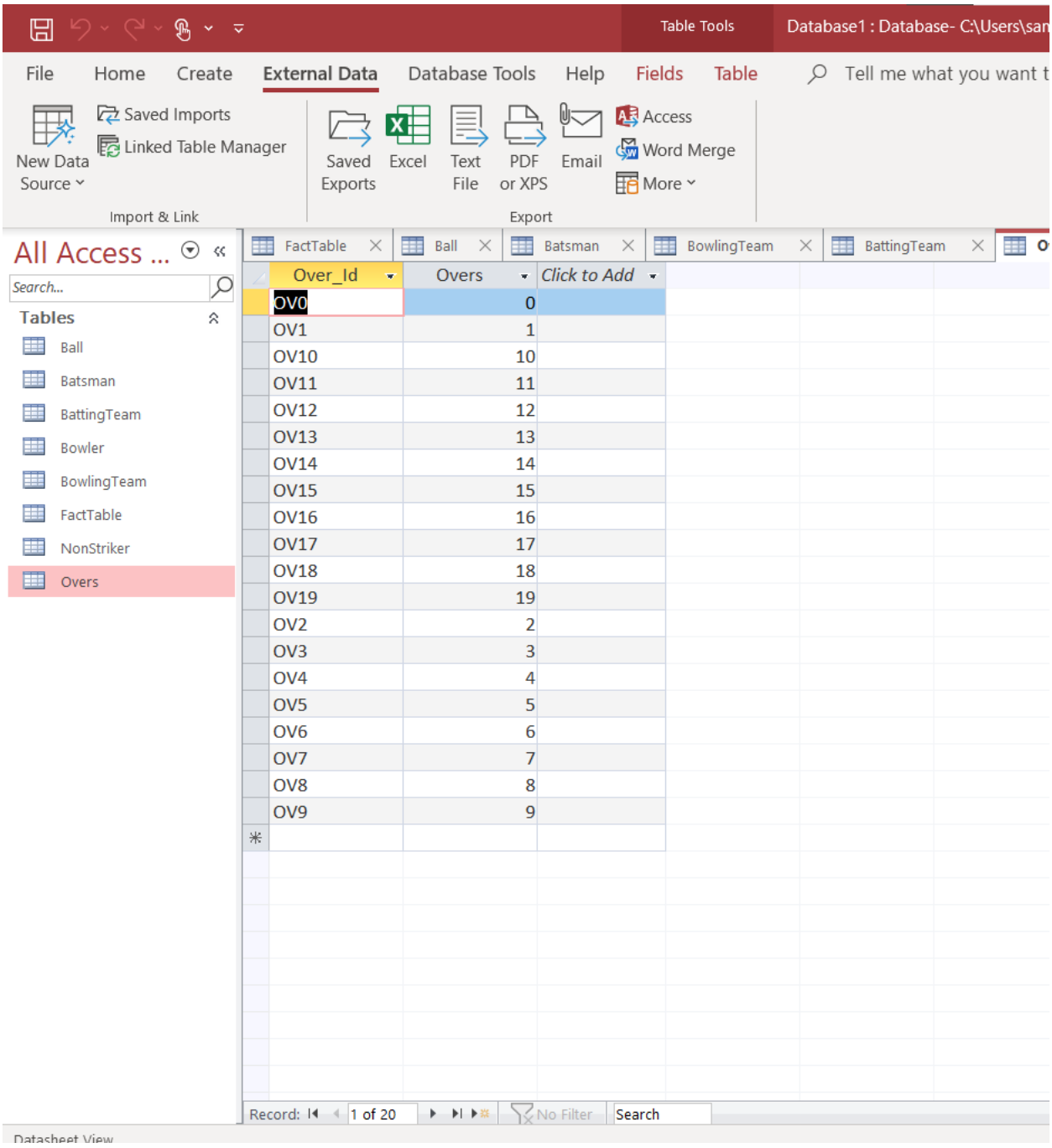


Fig. No. 20: Over Table

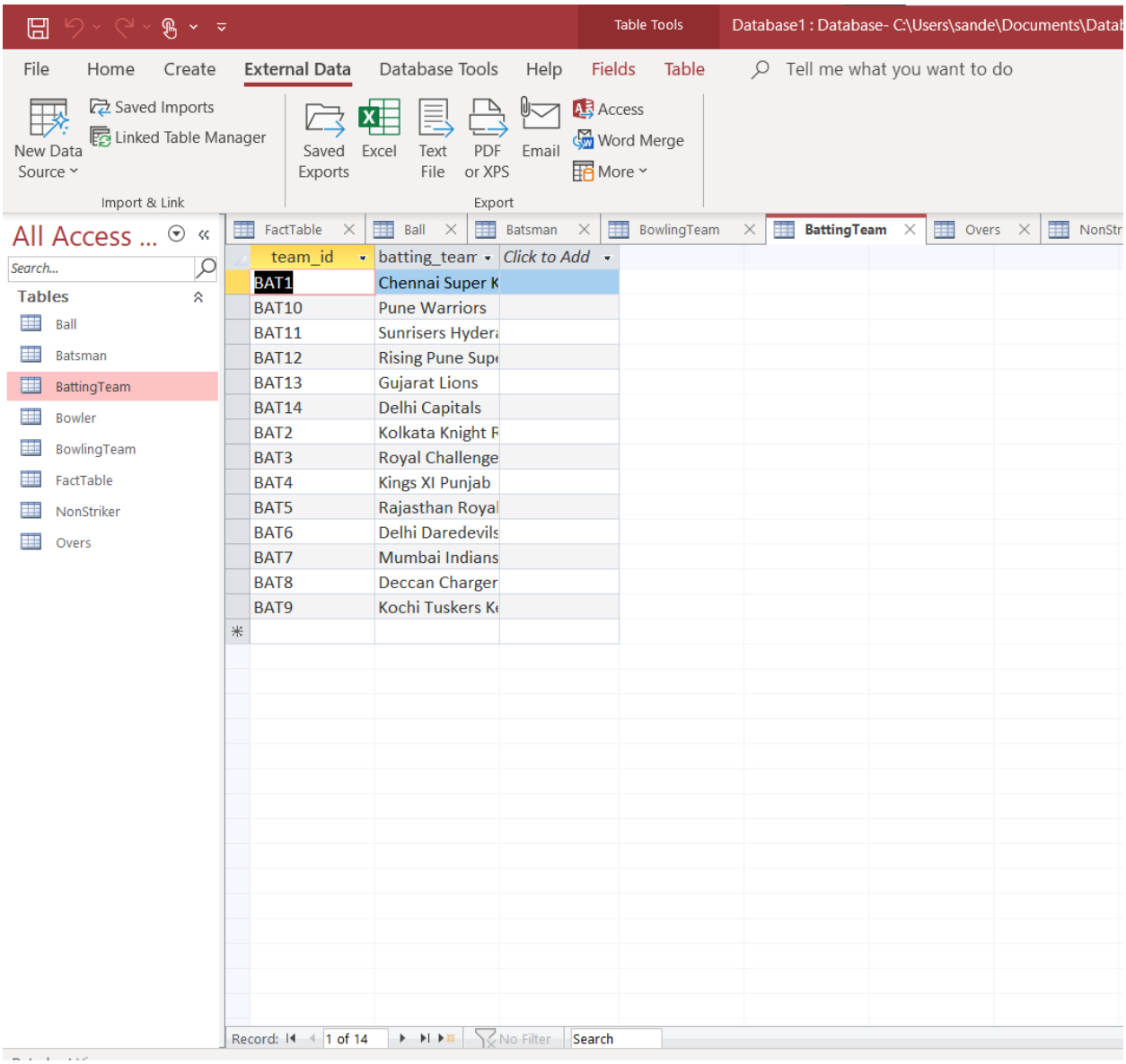


Fig. No. 21: BattingTeam Dimension Table

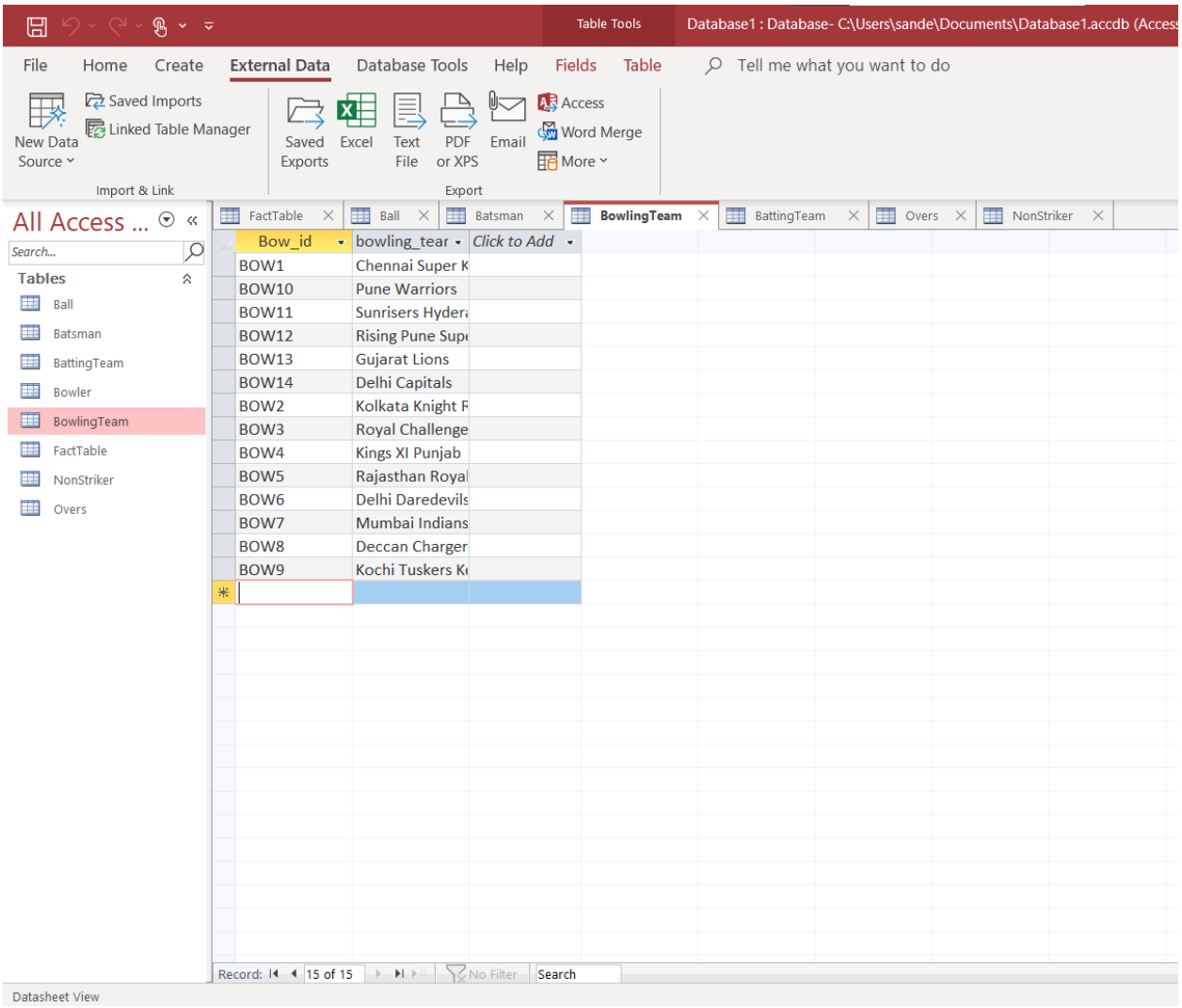


Fig. No. 22: Bowling Dimension Table

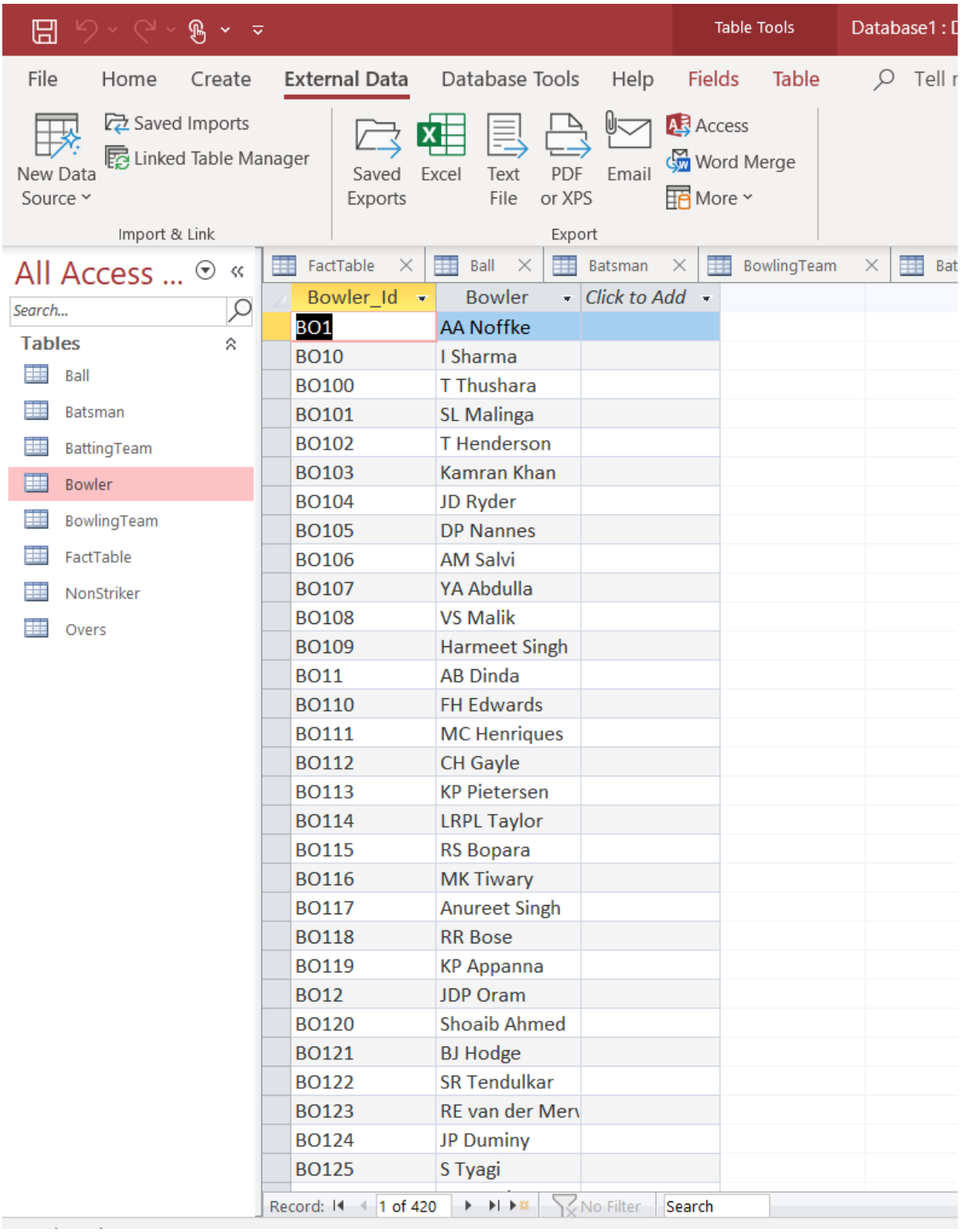


Fig. No. 23: Bowler Dimension Table

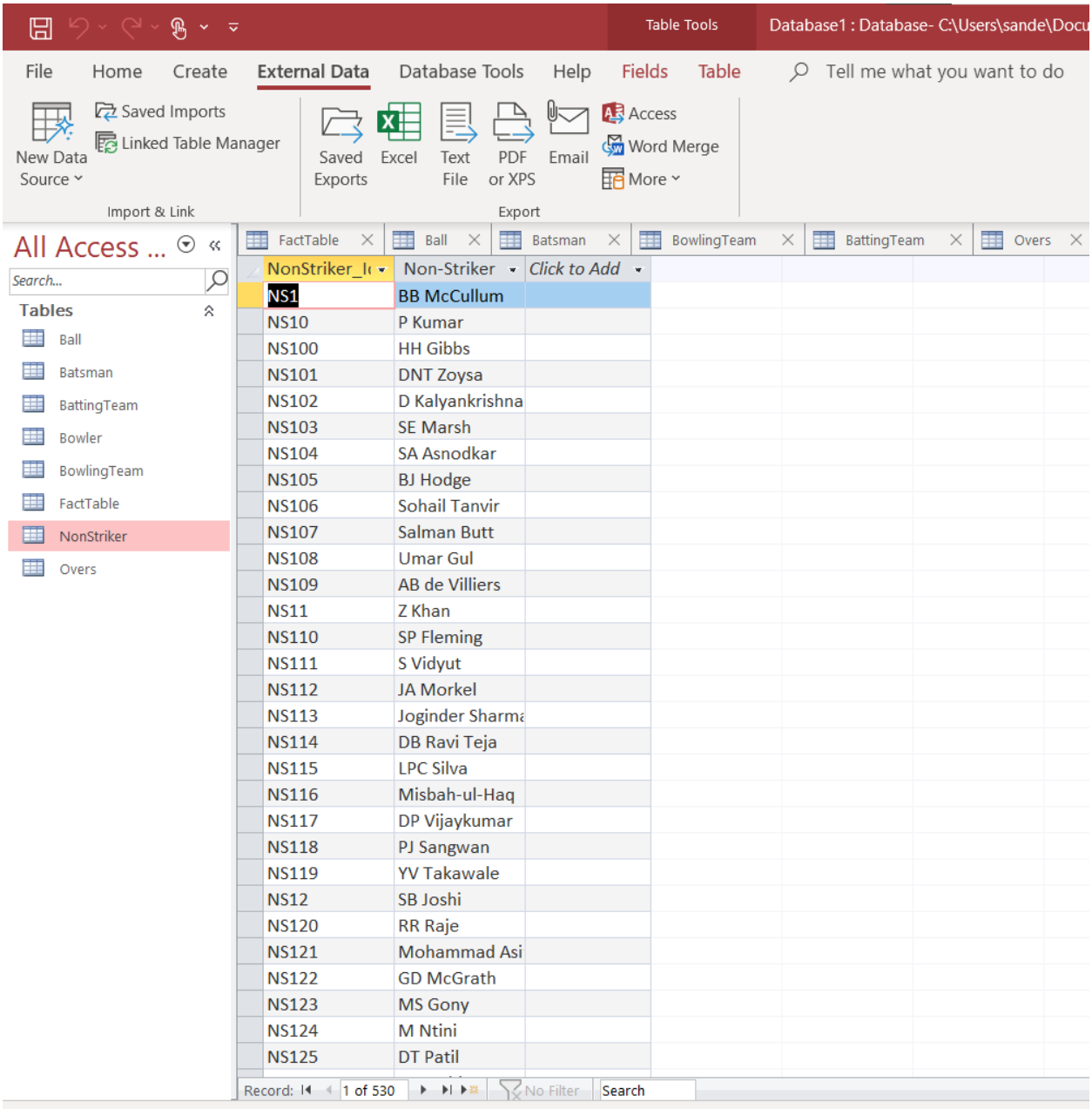


Fig. No. 24: Non Striker Dimension Table

Dimensional Modeling

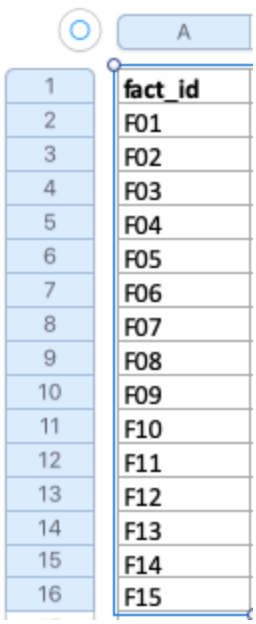
Business dimensions that are to be incorporated into a logical data model give us Dimensional Modeling. Dimensional modeling is a logical design technique that structures the business dimensions and the metrics that are analyzed along these dimensions. This modeling technique is intuitive for that purpose. High performance can be achieved while we query or do analysis using this model. Data in this model is contained in the Fact table and Dimension tables.

A fact table is used to maintain measurements. Each row in a fact table represents data which may relate to a particular method, a particular sample, or parameter used in a particular state. For each entity in an application, there will always be a row associated with that entity in the fact table. Few fields in a fact table may not have data. In an example where there is no method for a particular sample in any given month, the value of it in the fact table may not be present. These types of rows in a fact table may create gaps and hence make other important data to be sparse or scattered.

Fact_Table : fact_id, over, ball, batsman, non_striker, bowler, batsman_runs, extra_runs, total_runs, batting_team, bowling_team

fact_id	over	ball	batsman	non_striker	bowler	batsman_runs	extra_runs	total_runs	batting_team	bowling_team
F01	OV6	B5	BM1	NS1	BO1	1	0	1	BAT2	BOW3
F02	OV6	B6	BM2	NS2	BO1	1	0	1	BAT2	BOW3
F03	OV7	B1	BM2	NS2	BO2	0	0	0	BAT2	BOW3
F04	OV7	B2	BM2	NS2	BO2	1	0	1	BAT2	BOW3
F05	OV7	B3	BM1	NS1	BO2	1	0	1	BAT2	BOW3
F06	OV7	B4	BM2	NS2	BO2	1	0	1	BAT2	BOW3
F07	OV7	B5	BM1	NS1	BO2	1	0	1	BAT2	BOW3
F08	OV7	B6	BM2	NS2	BO2	1	0	1	BAT2	BOW3
F09	OV8	B1	BM2	NS2	BO3	0	0	0	BAT2	BOW3
F10	OV8	B2	BM2	NS2	BO3	0	0	0	BAT2	BOW3
F11	OV8	B3	BM2	NS2	BO3	0	0	0	BAT2	BOW3
F12	OV8	B4	BM2	NS2	BO3	1	0	1	BAT2	BOW3
F13	OV8	B5	BM1	NS1	BO3	1	0	1	BAT2	BOW3
F14	OV8	B6	BM2	NS2	BO3	2	0	2	BAT2	BOW3

Fact Table



	A
1	fact_id
2	F01
3	F02
4	F03
5	F04
6	F05
7	F06
8	F07
9	F08
10	F09
11	F10
12	F11
13	F12
14	F13
15	F14
16	F15

Fig. No. 8.1: Fact_Table (PK)

Dimension tables are used to represent business dimensions along which metrics are analyzed. An important characteristic of a dimensional table is that it is wide. Dimension table consists of many columns and attributes. We may come across dimension tables that have plenty of attributes, and hence that is how they are called wide. A dimensional table when laid out like a normal table with columns and row, it spans out horizontally**.**

Dimension Attributes are the various columns in a dimension table. In the Method dimension, the attributes can be Method ID, Method Name. The dimension attributes also contain one or more hierarchical relationships. Before designing our data warehouse, we need to decide what this data warehouse contains.

Over_Table: Over_Id, Overs
Batsman_Table: Batsman_Id, Batsman
Non_striker_Table : NonStriker_ID, NonStriker
Bowler_Table: Bowler_id, Bowler
Ball_Table: the toBall_id, Ball
BattingTeam_Table Team_id, Batting_Team
BowlingTeam_Table: Bowl_Id, Bowling_Team

Over_Id	Overs
OV0	0
OV1	1
OV2	2
OV3	3
OV4	4
OV5	5
OV6	6
OV7	7
OV8	8
OV9	9
OV10	10
OV11	11
OV12	12
OV13	13
OV14	14
OV15	15
OV16	16
OV17	17
OV18	18
OV19	19

Dimension Table

over
OV6
OV6
OV7
OV7
OV7
OV7
OV7
OV7
OV8
OV8
OV8
OV8
OV8
OV8
OV9
OV9
OV9
OV9
OV9
OV9
OV10

Fact Table

Fig. No. 8.2: Over Table

Batsman_Id	Batsman
BM1	RT Ponting
BM2	BB McCullum
BM3	DJ Hussey
BM4	Mohammad Hafeez
BM5	SC Ganguly
BM6	CL White
BM7	MV Boucher
BM8	B Akhil
BM9	AA Noffke
BM10	P Kumar
BM11	Z Khan
BM12	SB Joshi
BM13	W Jaffer
BM14	JH Kallis
BM15	R Dravid
BM16	V Kohli
BM17	K Goel
BM18	MEK Hussey
BM19	JR Hopes
BM20	SK Raina
BM21	JDP Oram
BM22	S Badrinath
BM23	PA Patel
BM24	ML Hayden
BM25	MS Dhoni
BM26	KC Sangakkara
BM27	Yuvraj Singh

Dimension Table

batsman
BM1
BM2
BM2
BM2
BM1
BM2
BM1
BM2
BM2
BM2
BM2
BM2
BM1
BM2
BM1
BM2
BM1
BM2
BM2
BM2
BM2
BM1
BM1
BM1
BM1
BM1
BM2
BM2
BM2
BM2

Fact Table

Fig. No. 8.3: Batsman Table

NonStriker_Id	Non-Striker
NS1	BB McCullum
NS2	RT Ponting
NS3	DJ Hussey
NS4	Mohammad Hafeez
NS5	SC Ganguly
NS6	MV Boucher
NS7	CL White
NS8	B Akhil
NS9	AA Noffke
NS10	P Kumar
NS11	Z Khan
NS12	SB Joshi
NS13	JH Kallis
NS14	W Jaffer
NS15	R Dravid
NS16	V Kohli
NS17	JR Hopes
NS18	S Badrinath
NS19	K Goel
NS20	SK Raina
NS21	MEK Hussey
NS22	JDP Oram
NS23	ML Hayden
NS24	PA Patel
NS25	MS Dhoni
NS26	KC Sangakkara
NS27	Yuvraj Singh
NS28	SM Katich
NS29	IK Pathan
NS30	YK Pathan
NS31	T Kohli
NS32	G Gambhir

Dimension Table

non_striker
NS1
NS2
NS2
NS2
NS1
NS2
NS1
NS2
NS2
NS2
NS2
NS2
NS1
NS2
NS1
NS2
NS1
NS2
NS2
NS2
NS2
NS1
NS1
NS1
NS1
NS1
NS2
NS2
NS2
NS2
NS1
NS1
NS1
NS3
NS3
NS3

Fact Table

Fig. No. 8.4: NonStriker Table

Bowler_Id	Bowler
BO1	AA Noffke
BO2	Z Khan
BO3	JH Kallis
BO4	SB Joshi
BO5	CL White
BO6	P Kumar
BO7	AB Agarkar
BO8	SC Ganguly
BO9	LR Shukla
BO10	I Sharma
BO11	AB Dinda
BO12	JDP Oram
BO13	JR Hopes
BO14	MS Gony
BO15	K Goel
BO16	PP Chawla
BO17	WA Mota
BO18	IK Pathan
BO19	S Sreesanth
BO20	B Lee
BO21	M Muralitharan
BO22	P Amarnath
BO23	Joginder Sharma
BO24	GD McGrath
BO25	B Geeves
BO26	SK Warne
BO27	YK Pathan
BO28	D Salunkhe
BO29	SR Watson
BO30	SK Trivedi
BO31	DL Vettori
BO32	MF Maharoof
BO33	MM Patel
BO34	R Bhatia
BO35	B Akhil
BO36	R Vinay Kumar
BO37	A Nehra

Dimension Table

bowler
BO1
BO1
BO2
BO2
BO2
BO2
BO2
BO2
BO3
BO3
BO3
BO3
BO3
BO3
BO4
BO4
BO4
BO4
BO4
BO4
BO3
BO3
BO3
BO3
BO3
BO3
BO4
BO4
BO4
BO4
BO4
BO4
BO3
BO3

Fact Table

Fig. No. 8.5: Bowler Table

Ball_Id	Ball
B1	1
B2	2
B3	3
B4	4
B5	5
B6	6
B7	7
B8	8
B9	9

Dimension_table

ball
B5
B6
B1
B2
B3
B4
B5
B6
B1
B2
B3
B4
B5
B6
B1
B2
B3
B4
B5
B6
B1
B2
B3
B4
B5

Fact Table

Fig. No. 8.6: Bowl Table

team_id	batting_team
BAT1	Chennai Super Kings
BAT2	Kolkata Knight Riders
BAT3	Royal Challengers Bangalore
BAT4	Kings XI Punjab
BAT5	Rajasthan Royals
BAT6	Delhi Daredevils
BAT7	Mumbai Indians
BAT8	Deccan Chargers
BAT9	Kochi Tuskers Kerala
BAT10	Pune Warriors
BAT11	Sunrisers Hyderabad
BAT12	Rising Pune Supergiants
BAT13	Gujarat Lions
BAT14	Delhi Capitals

Dimension_Table

	batting_team
1	BAT2
1	BAT2
)	BAT2
1	BAT2
1	BAT2
1	BAT2
1	BAT2
1	BAT2
)	BAT2
)	BAT2
)	BAT2
1	BAT2
1	BAT2
2	BAT2
1	BAT2
1	BAT2
1	BAT2
)	BAT2
5	BAT2
1	BAT2
1	BAT2
4	BAT2
)	BAT2
5	BAT2
)	BAT2
4	BAT2
)	BAT2

Fact Table

Fig. No. 8.7: Batting_Team Table

Bow_id	bowling_team
BOW1	Chennai Super Kings
BOW2	Kolkata Knight Riders
BOW3	Royal Challengers Bangalore
BOW4	Kings XI Punjab
BOW5	Rajasthan Royals
BOW6	Delhi Daredevils
BOW7	Mumbai Indians
BOW8	Deccan Chargers
BOW9	Kochi Tuskers Kerala
BOW10	Pune Warriors
BOW11	Sunrisers Hyderabad
BOW12	Rising Pune Supergiants
BOW13	Gujarat Lions
BOW14	Delhi Capitals

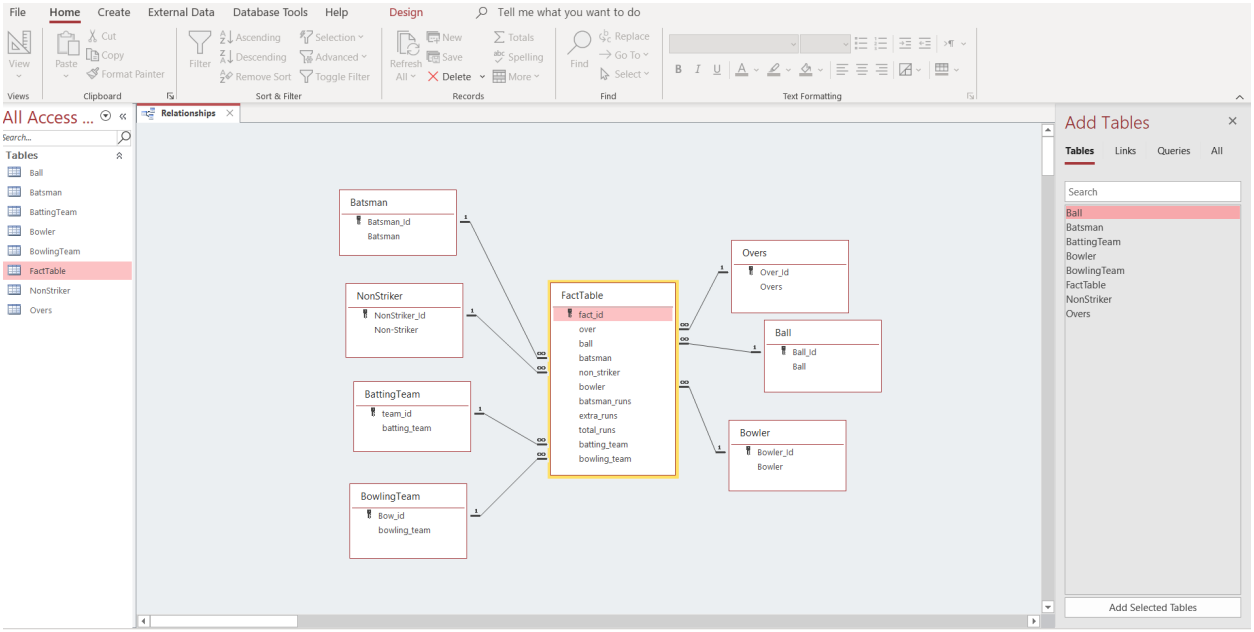
Dimension_table

bowling_team
BOW3
BOW3
BOW3
BOW3
BOW3
BOW3
BOW3
BOW3
BOW3
BOW3
BOW3
BOW3
BOW3
BOW3
BOW3
BOW3
BOW3
BOW3
BOW3
BOW3
BOW3
BOW3
BOW3
BOW3

Fact Table

Fig. No. 8.8: Bowling_Team Table

Dimensional Hierarchies Defines a series of mappings from low-level concepts to higher-level, more general concepts. The dimensional levels in the hierarchy form the structure of the tree. Members at the lowest level are referred to as leaf members and are related to a single member at the highest level. Dimensional hierarchies are those different levels of information within the company dimension. Analysts, data experts, and higher-level management can use the dimensional hierarchies as the paths for drilling down Analysts, data experts and senior management may use dimensional hierarchies as pathways for digging down or rolling up research. Modern software is very useful when designing fact tables, dimensional tables and creating relationships between them. There are two types of schemas that are commonly used in a data warehouse. This is the STAR schema and the Snowflake schema. or rolling up in analysis. Modern software is very useful when designing fact tables, dimensional tables and creating relationships between them. There are two types of schemas that are commonly used in a data warehouse. In the STAR system, the dimensional tables are grouped around a centralized truth table, making it appear like a star. The STAR schema is not structured. Snowflaking is a method of normalizing the size tables in the STAR schema. When normalized, the resulting structure is identical to the STAR scheme. Part tables are often referred to as dimension tables.



Data Transfer to SQL Server

SQL Server Management Studio (SSMS) is a software system that was first published in 2005. The framework offers both script editors and graphical tools that manage server artifacts and functions.

The core feature of SSMS is the Object Explorer, which allows the user to search, select, and act on all objects within the server.

Steps to import data file to SQL Server:

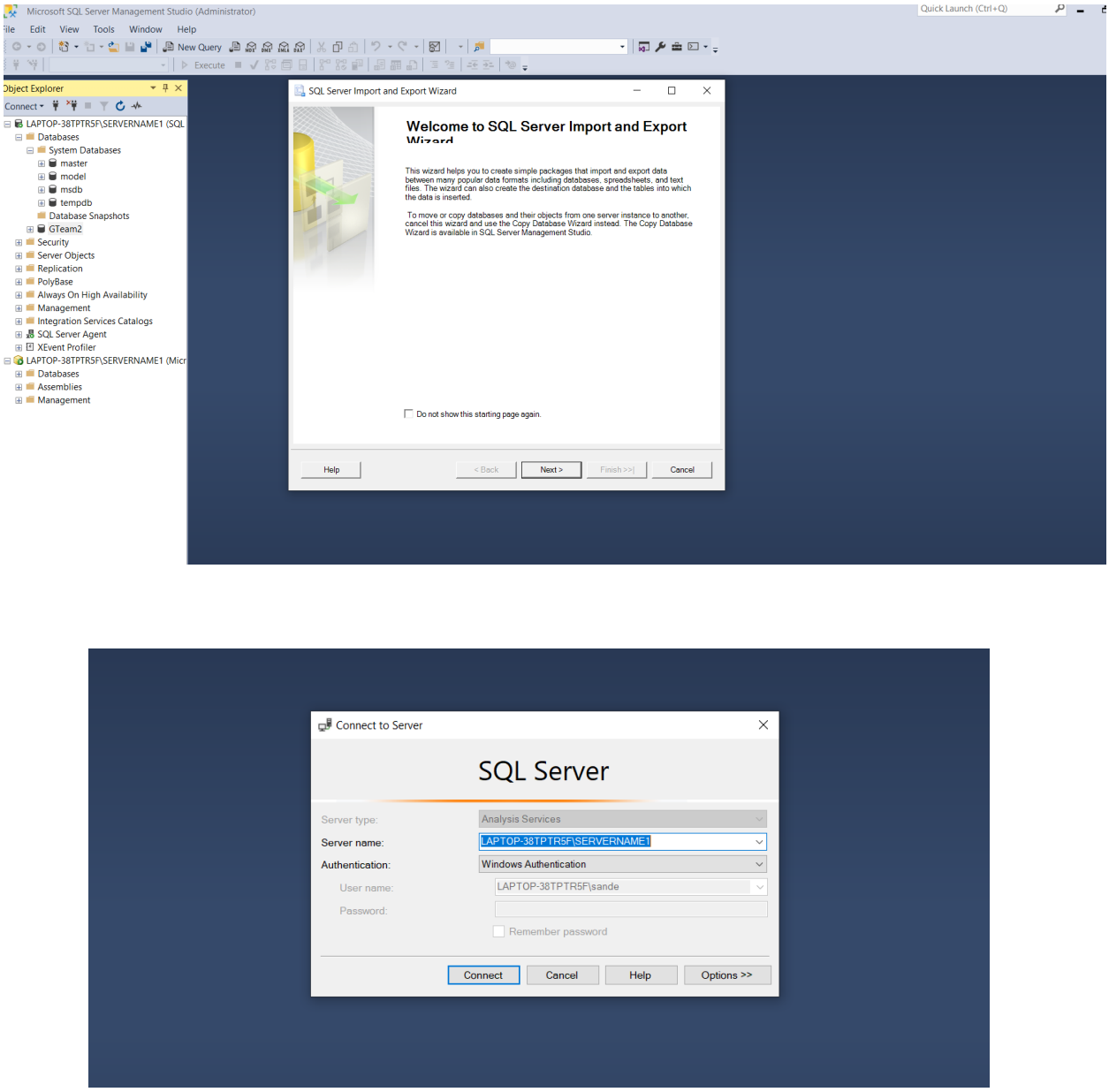


Fig 25: Connection to SQL Server

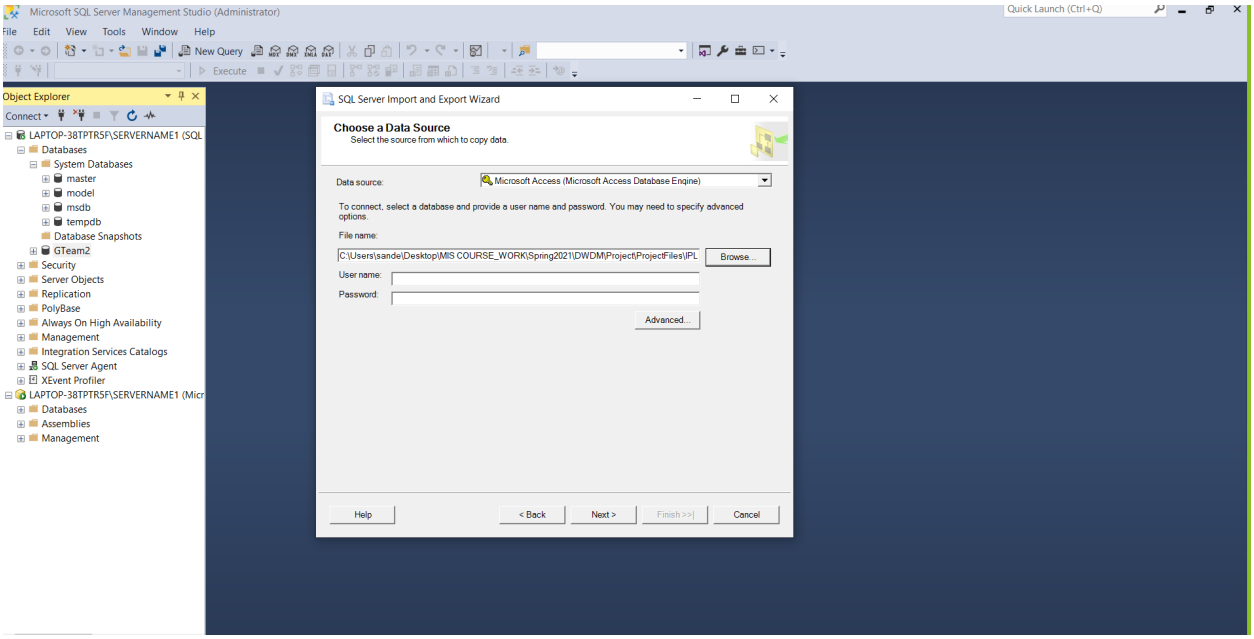


Fig 26: Choose Data Source

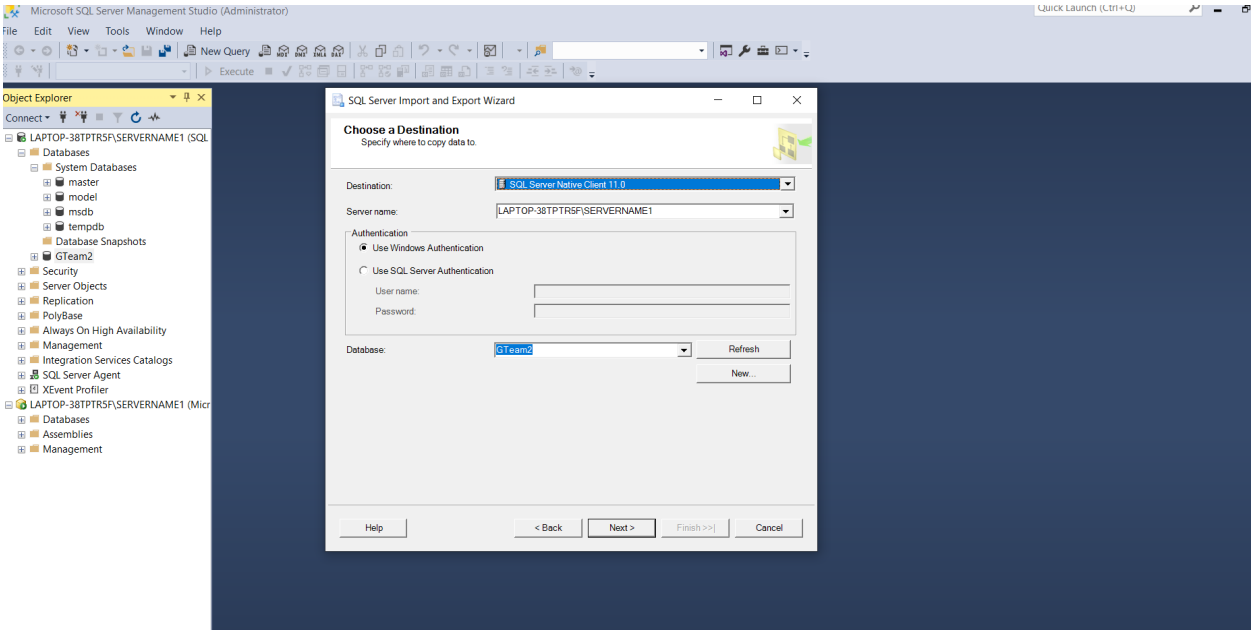


Fig 27: Choose Destination

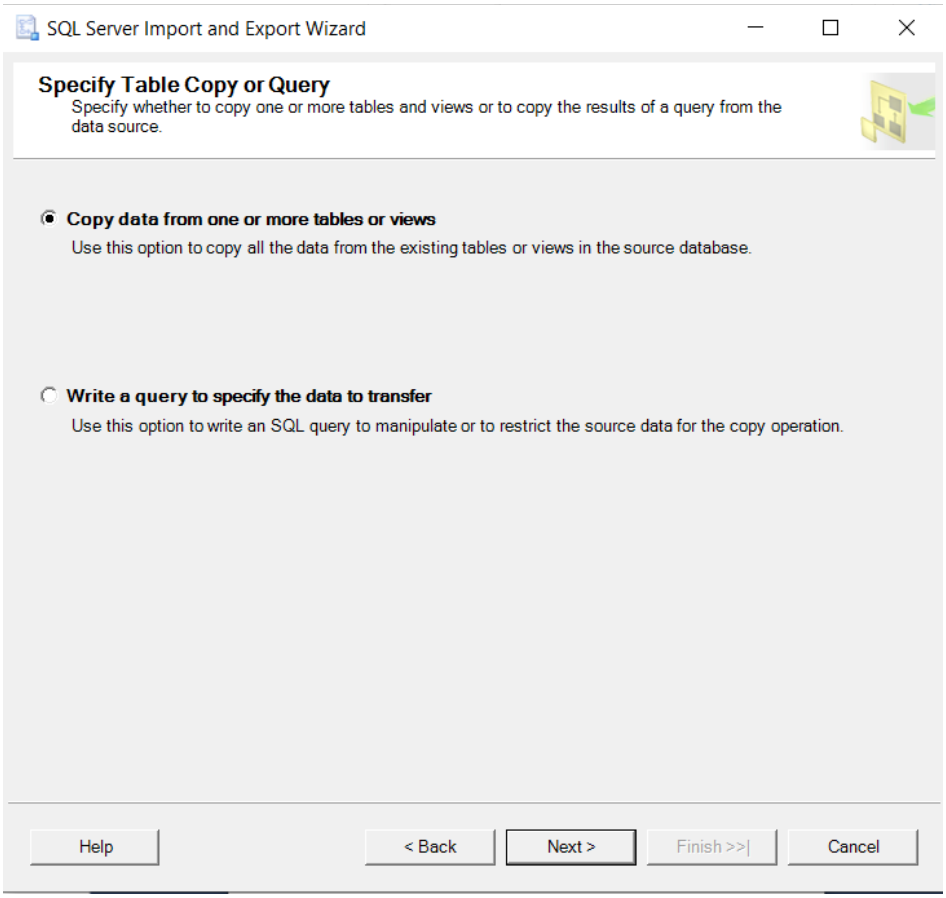


Fig 28: Specify Table Copy or Query

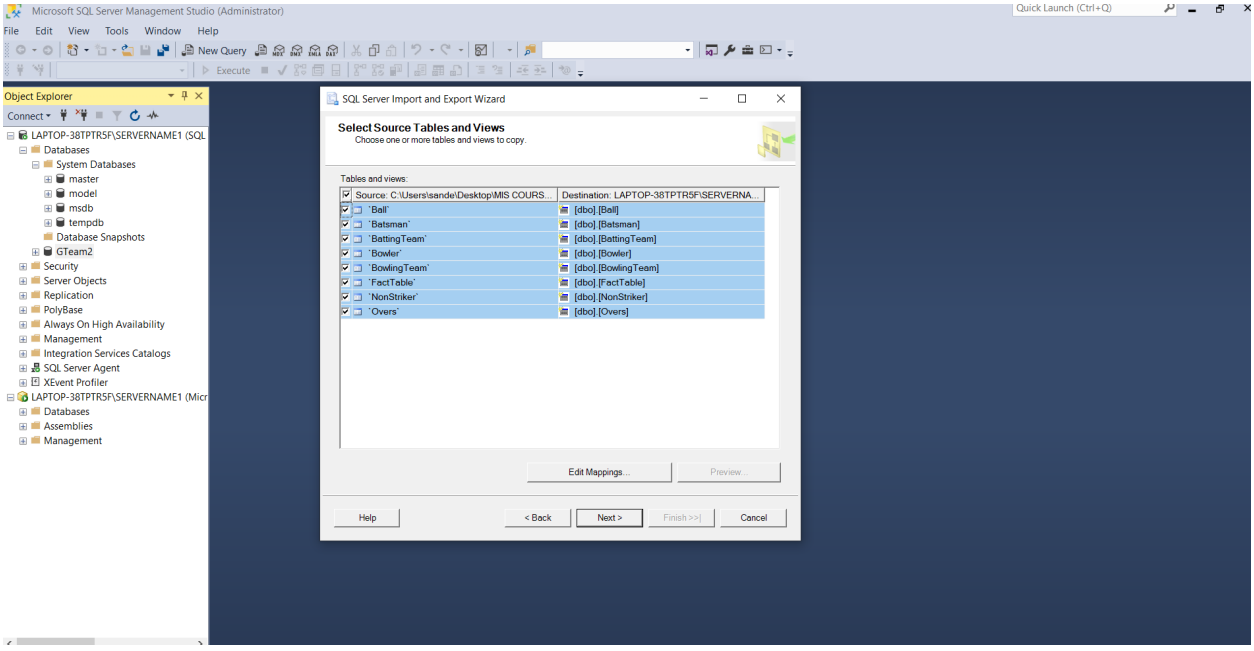


Fig 29: Select Source Tables and Views

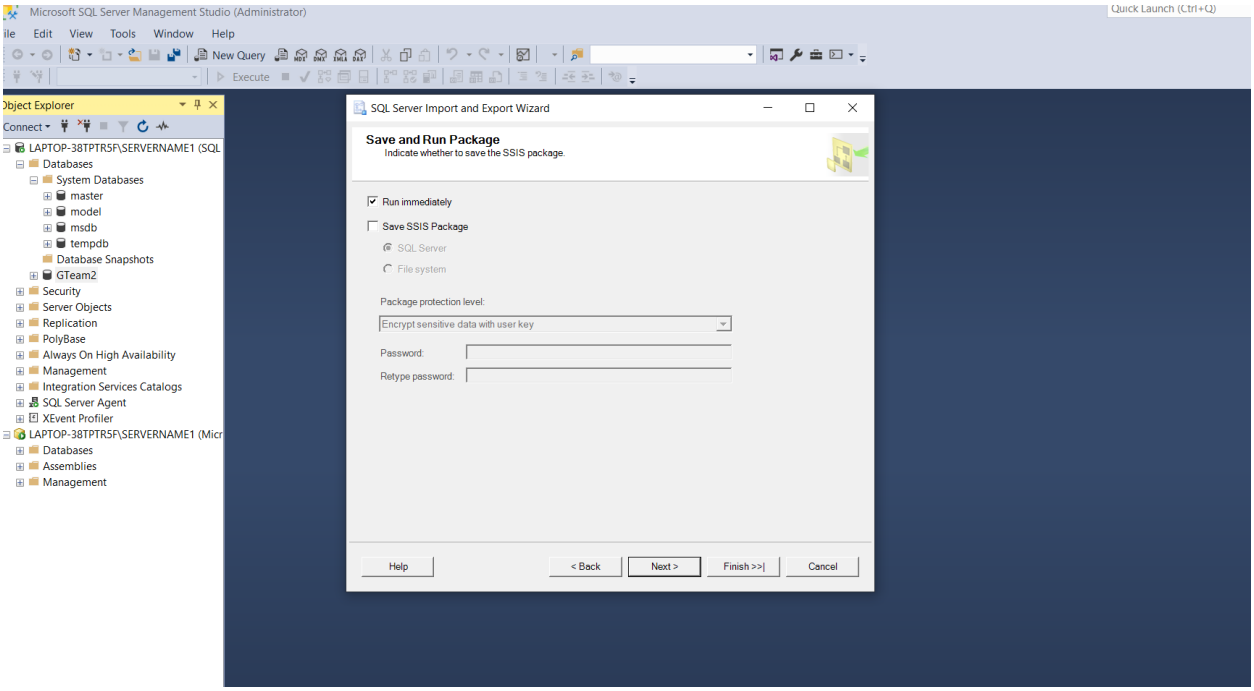


Fig 30:Save and Run Package

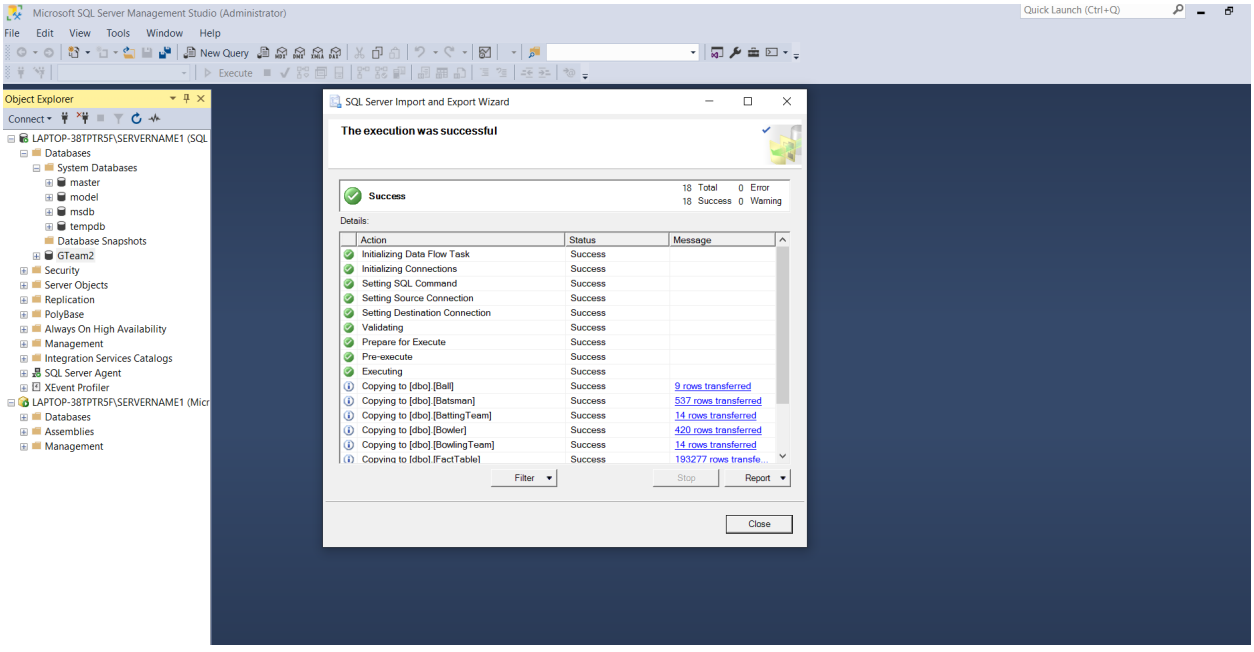


Fig 31: Execution Successful

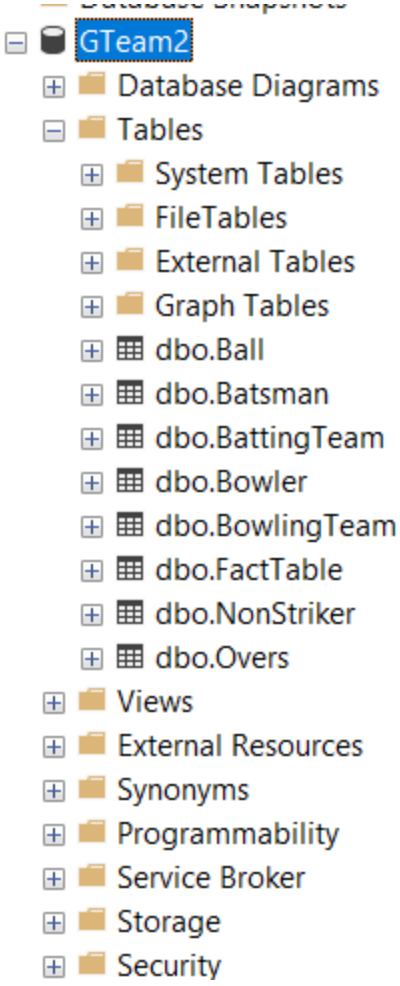


Fig 32: Tables in SQL Server

Cube Development and Deployment

Based on the GTeam2 database, we can upgrade OLAP to the information we've recovered using SQL server information devices. To do this we need to move the data from Microsoft to the SQL server database. ToNon-Striker dimensions that the dimension Non-Striker carries out the OLAP cube for our knowledge, the following steps should be taken:
Build an OLAP database on the SQL server that uses information from Microsoft as a source of information.

- To construct an Information Cube Schema, connect the current information tables in the same way.
- Build the reality and measurements you want for the truth table.
- Define the relation of the trait between the measurements.
- Process the cube with open details so that it can be separated into various hierarchies.

Steps to Cube Development and Deployment in Visual Studio:

Fig 33: Creating Project in Visual Studio

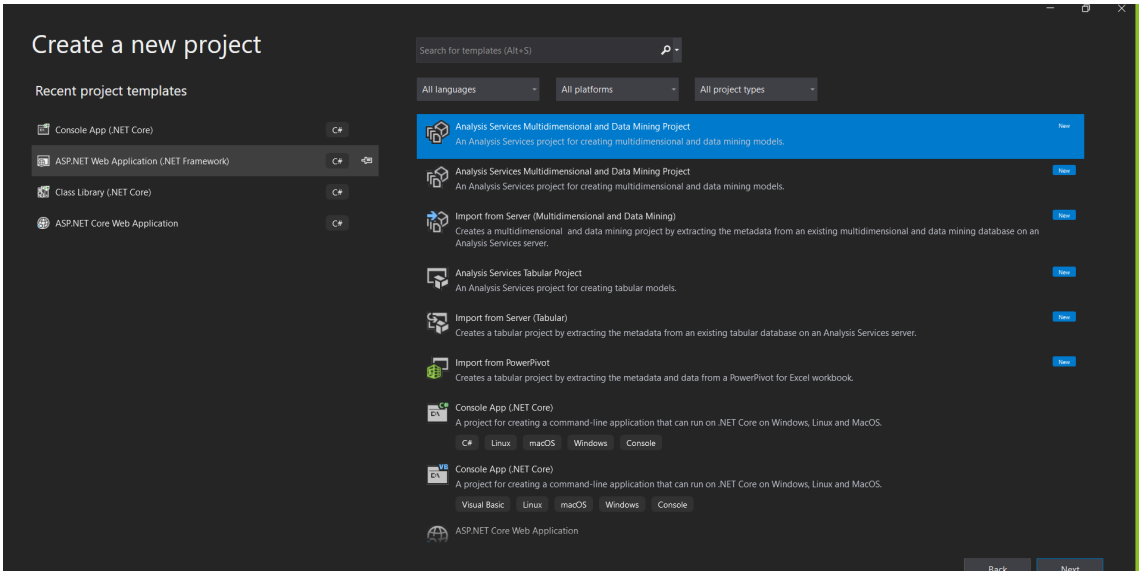


Fig 34: Selecting Analysis Services Template

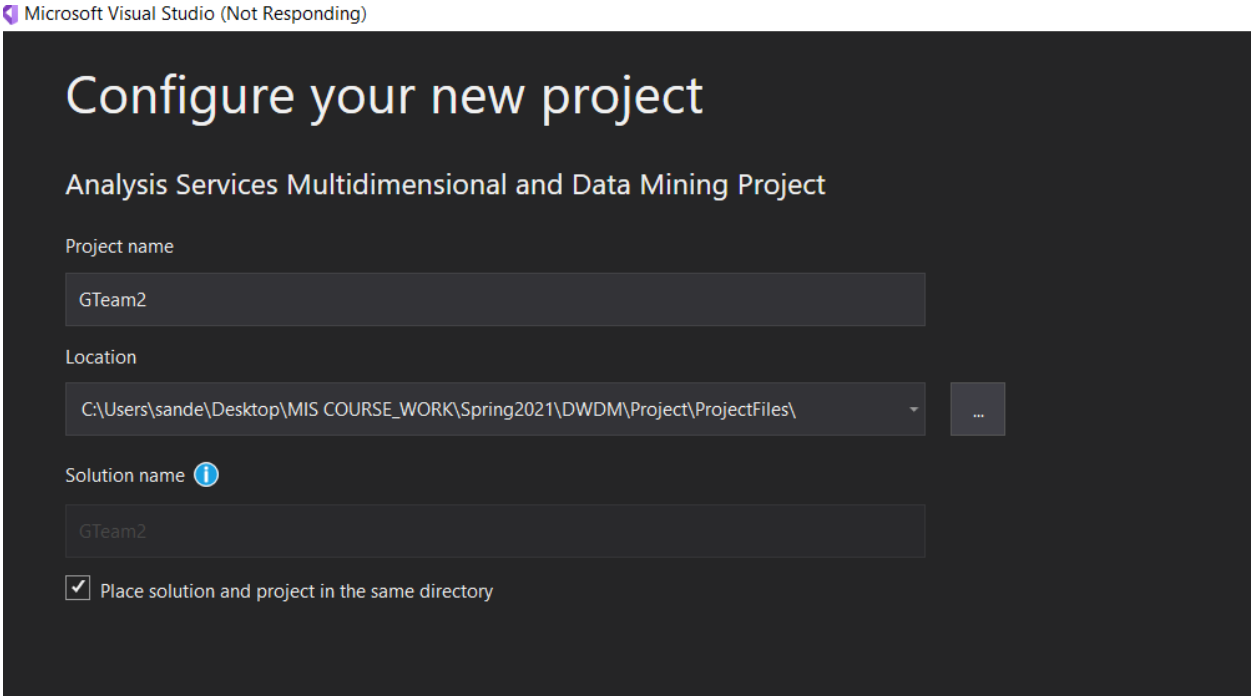


Fig 35: Configuring New Project

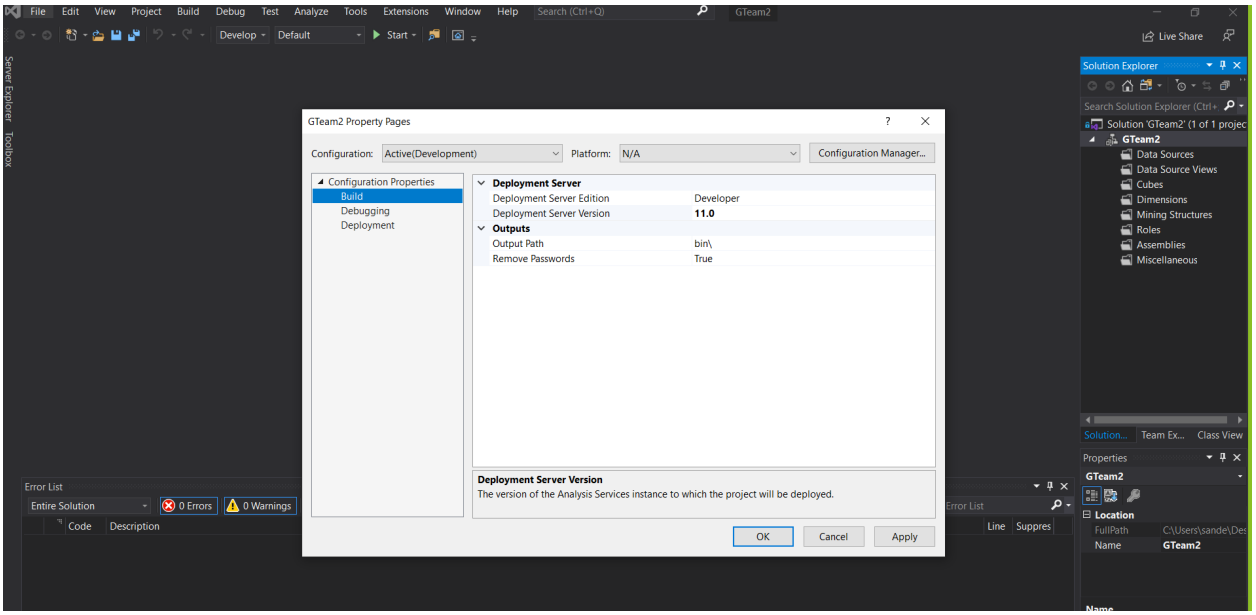


Fig 36: Deployment Server Version from 15.0 to 11.0

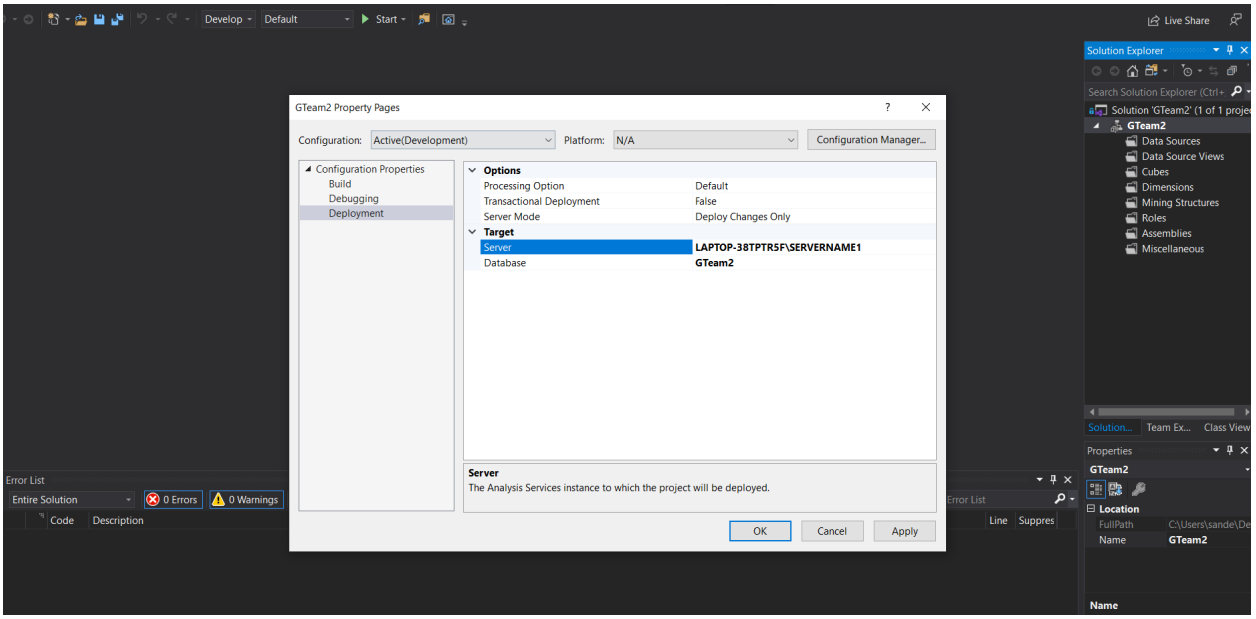


Fig 37: Set Target Server to SBUS-DB

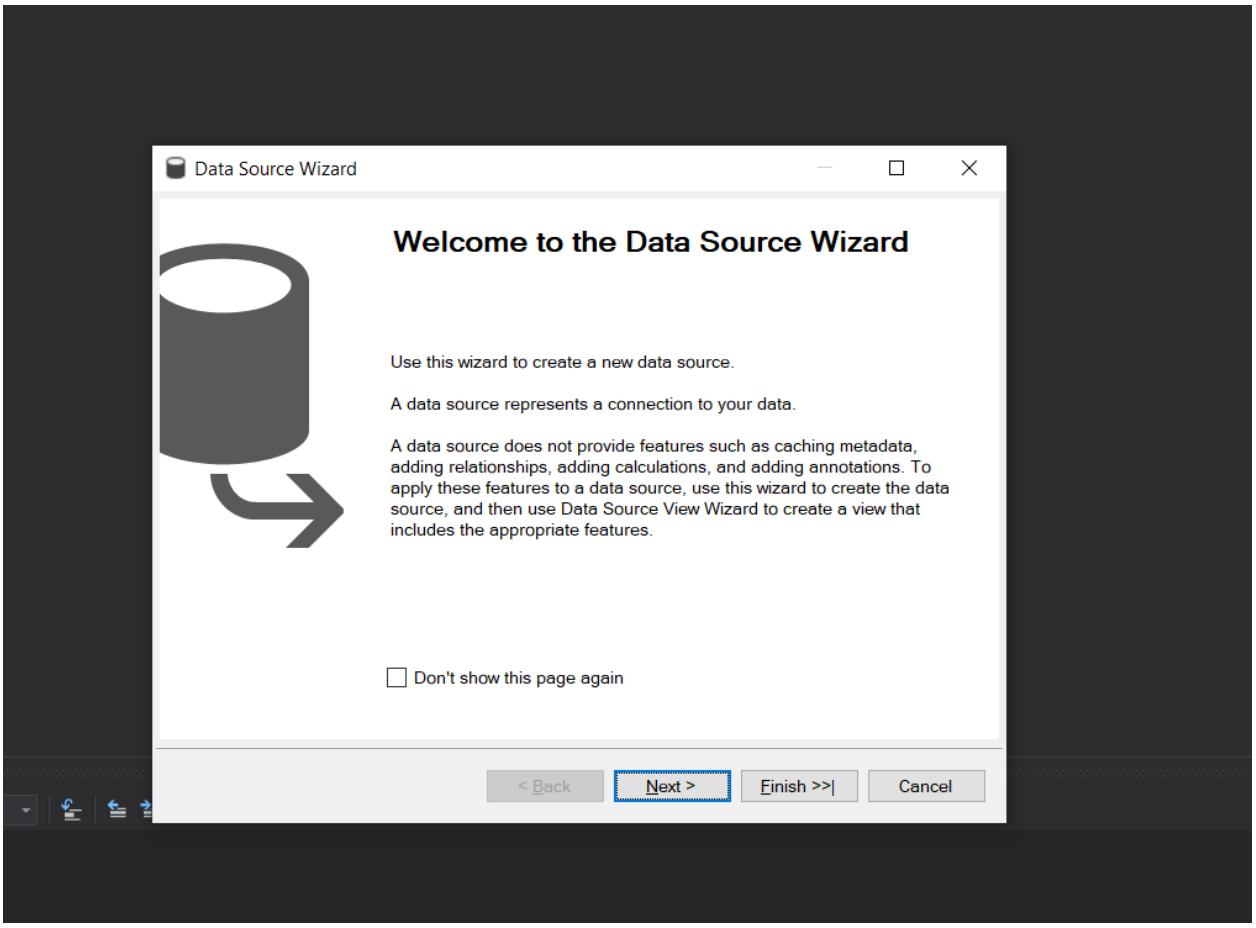


Fig 38: Data Source Wizard Installation

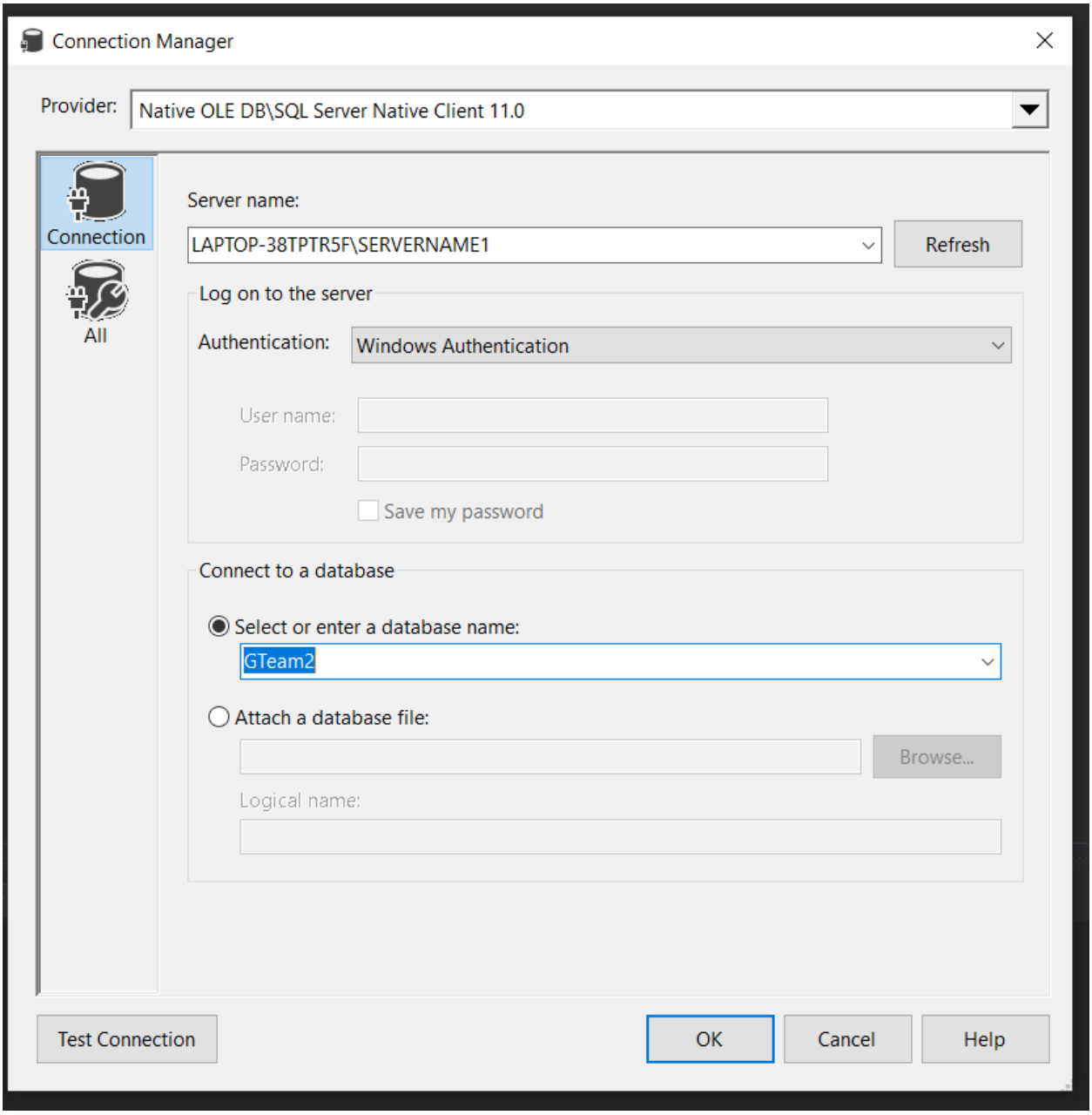


Fig 39: Database Connection

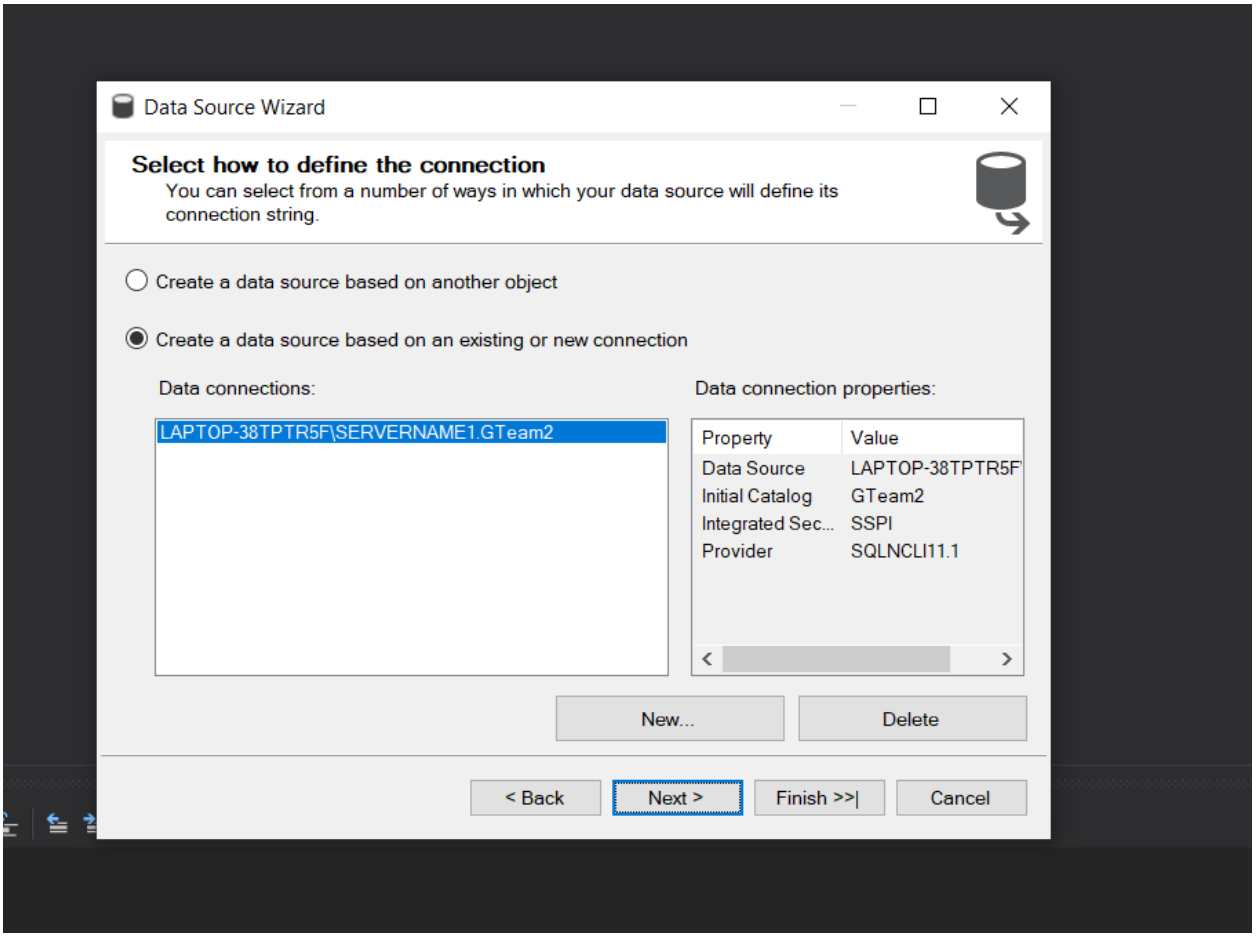


Fig 40: Defining Connection String

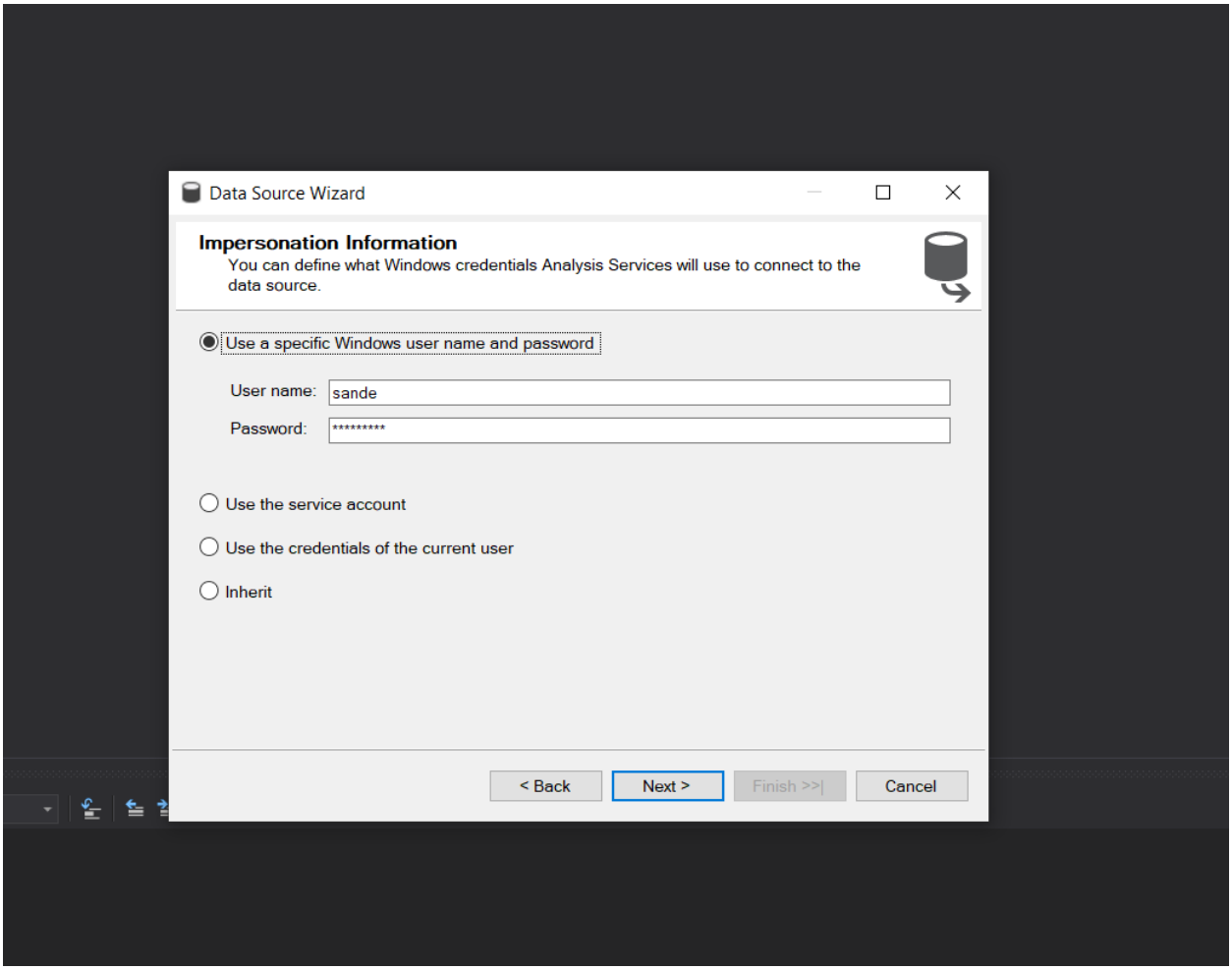


Fig 41: Entering Credentials for connecting data source

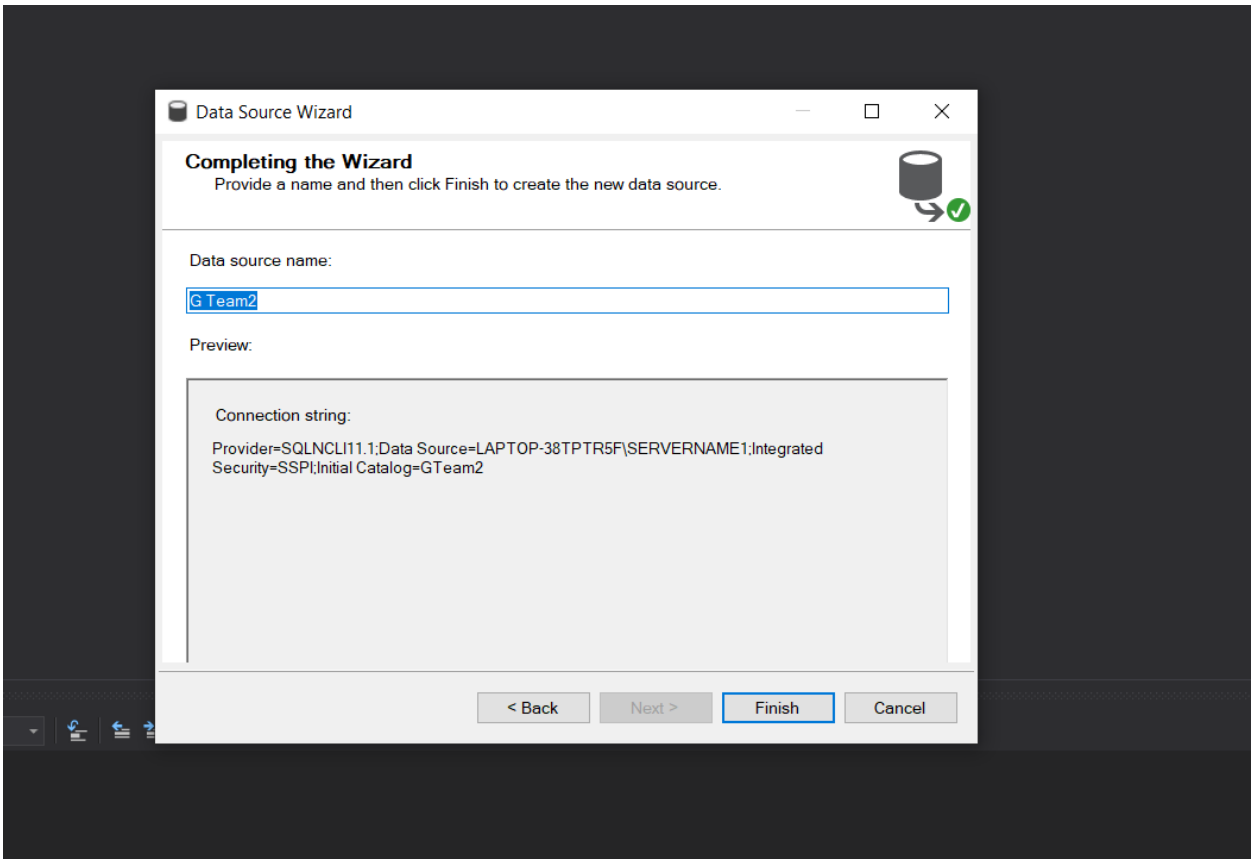


Fig 42: Entering Data Source Name

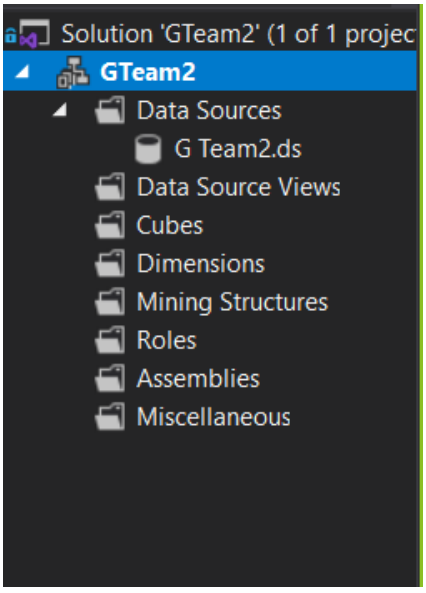


Fig 43: View Data Source

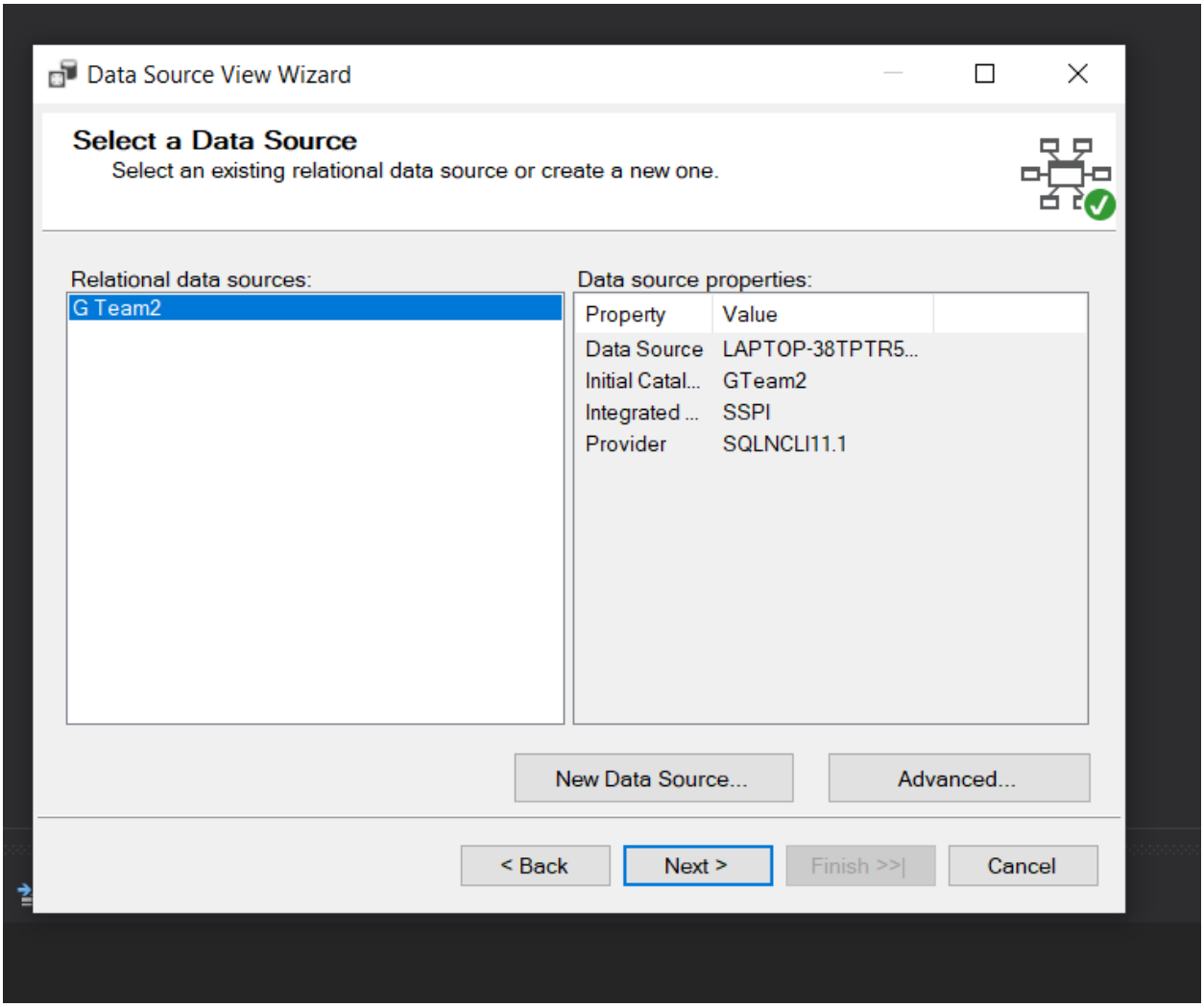


Fig 44: Including Available objects in View Wizard

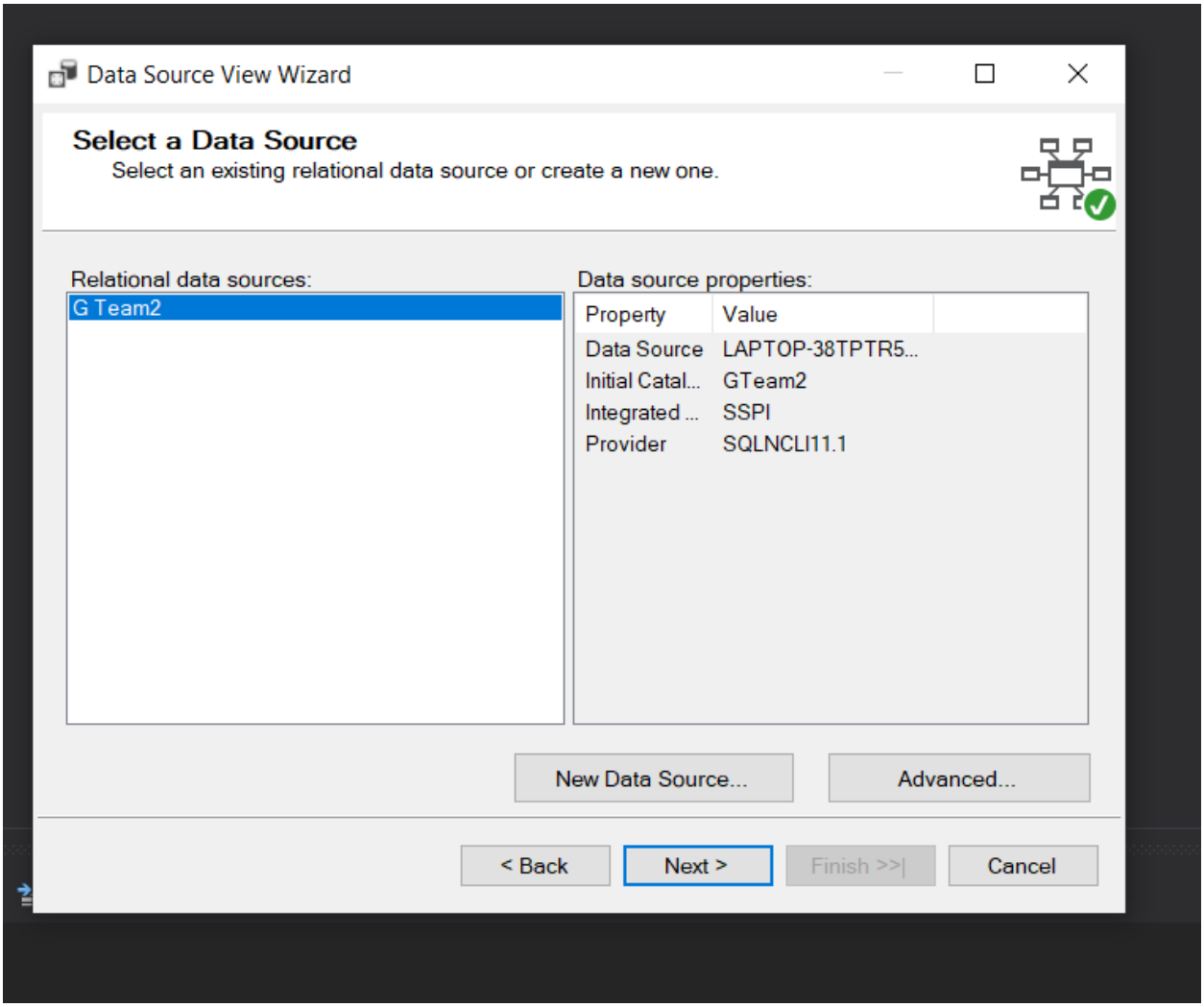


Fig 45: Selecting the Data Source

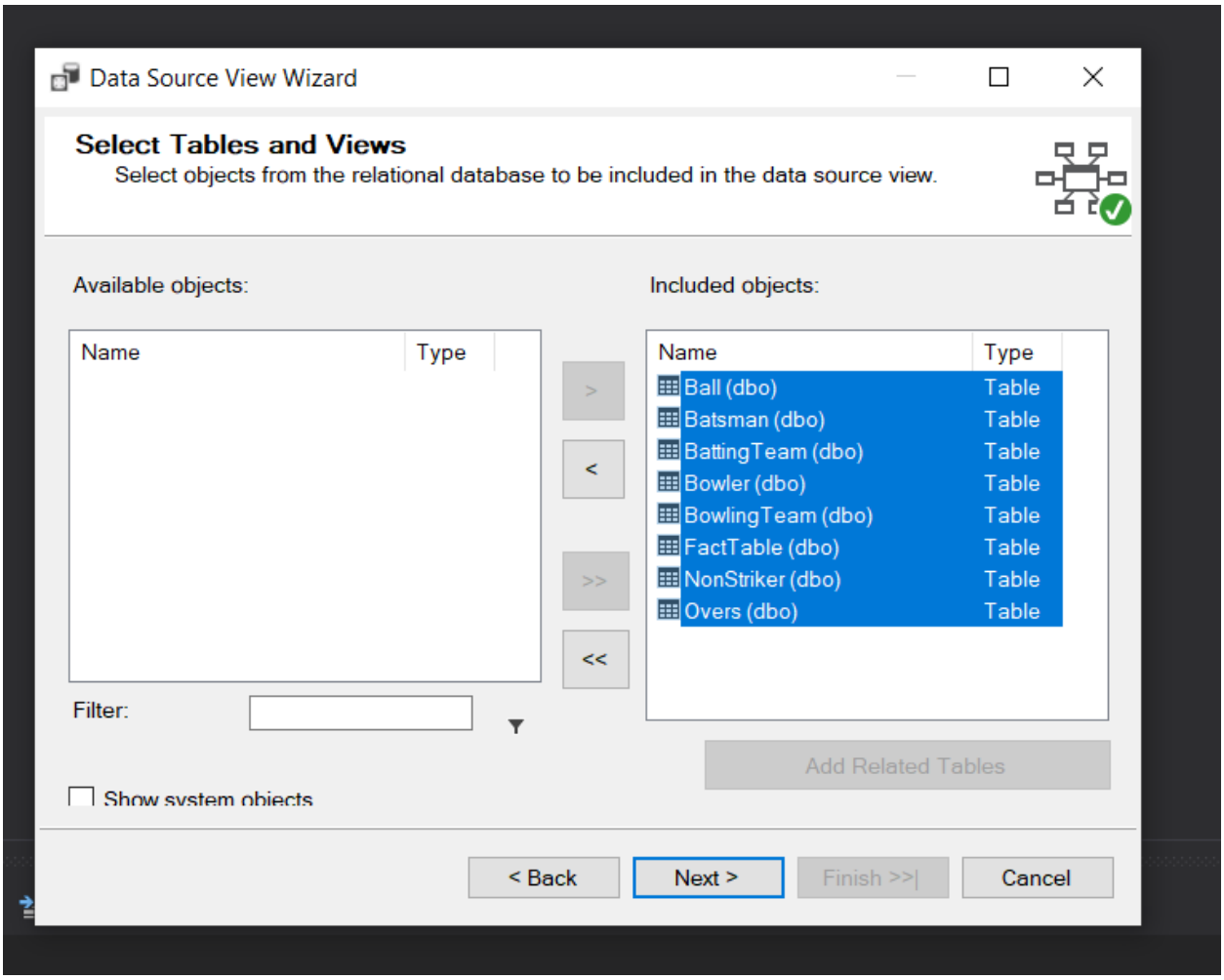


Fig 46: Including Available objects in View Wizard

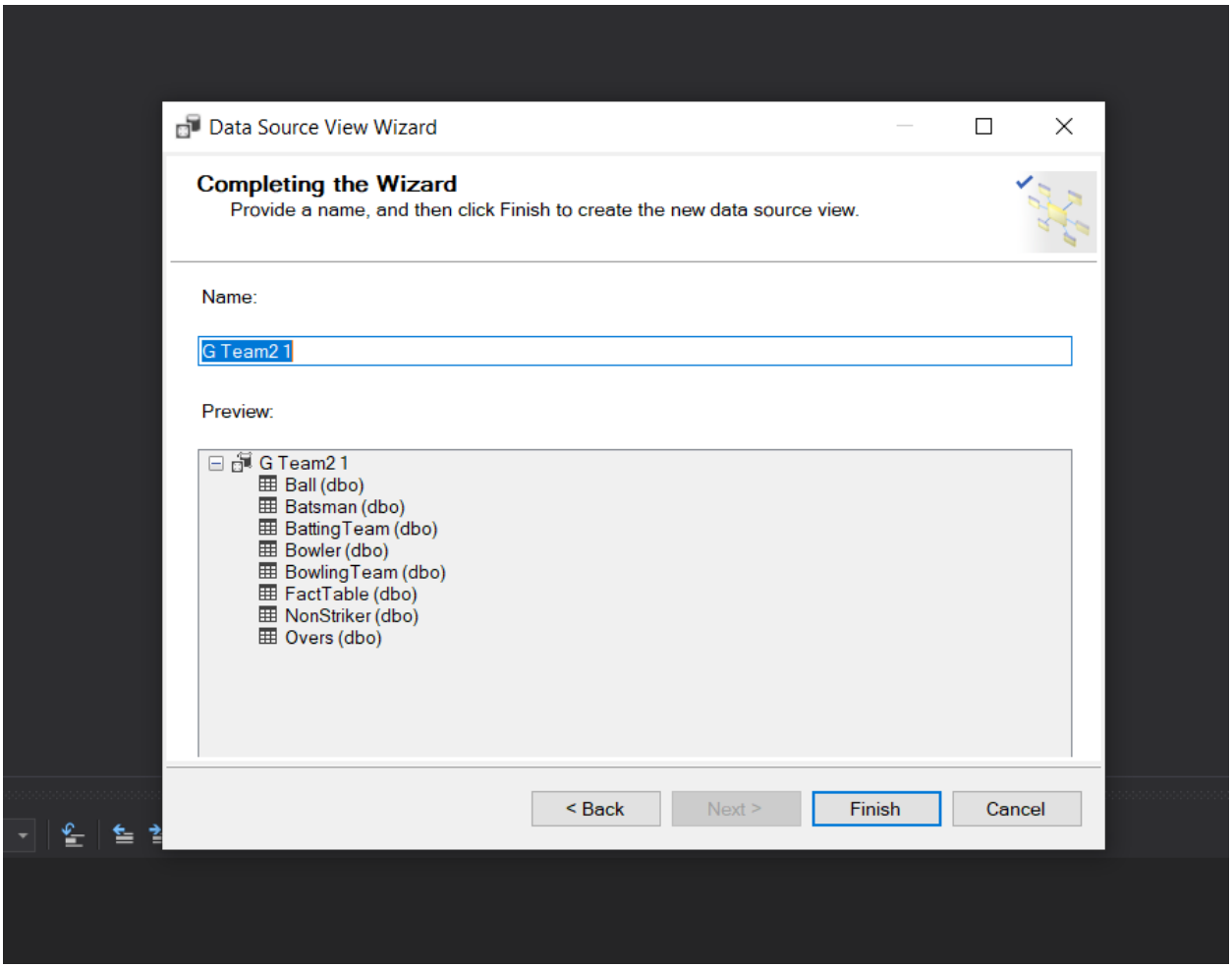


Fig 47: Data Source View Wizard

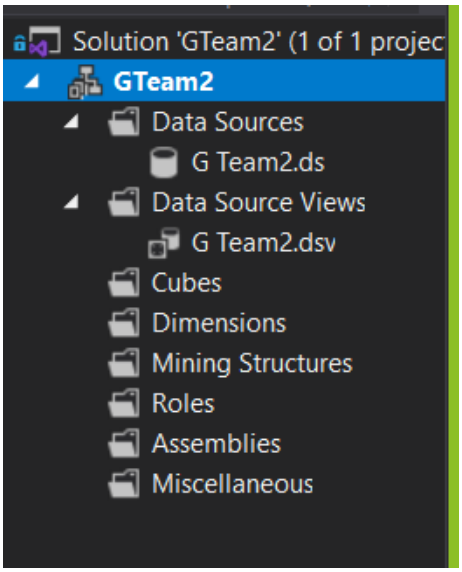


Fig 48: Solution Explorer Screen with Data Source View

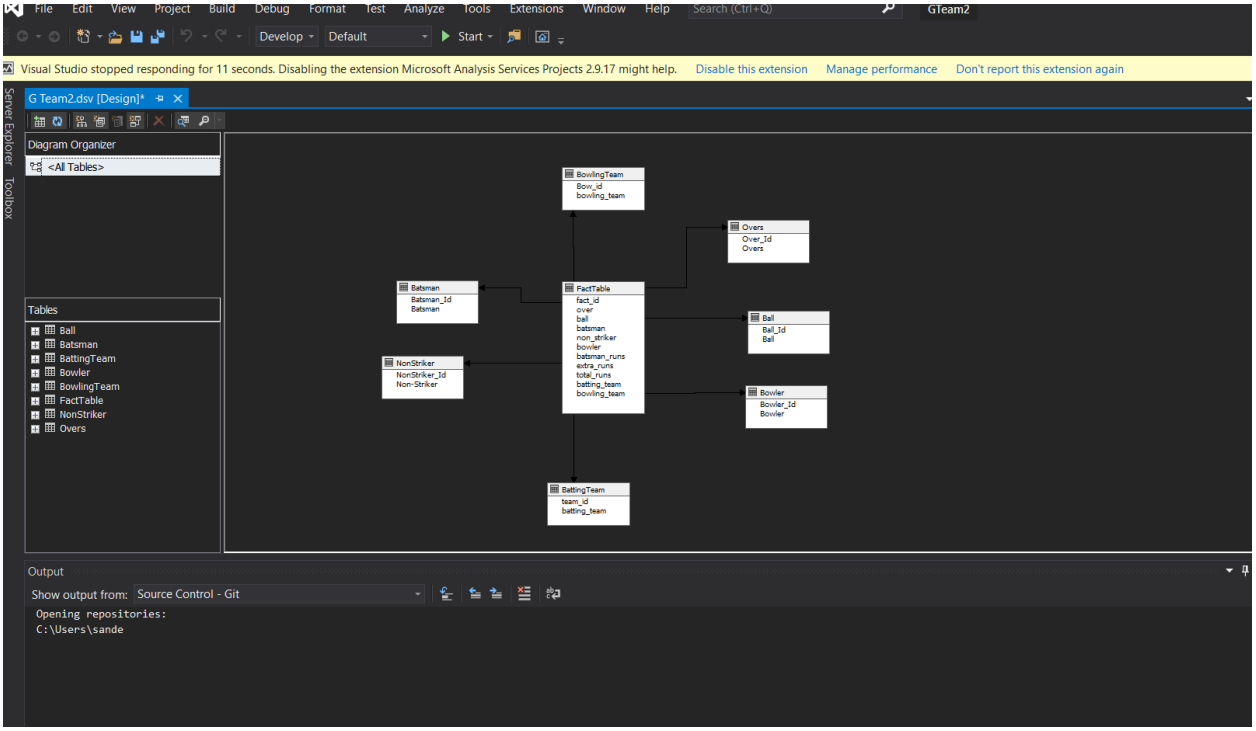


Fig 49: Relationship established in Data Source View

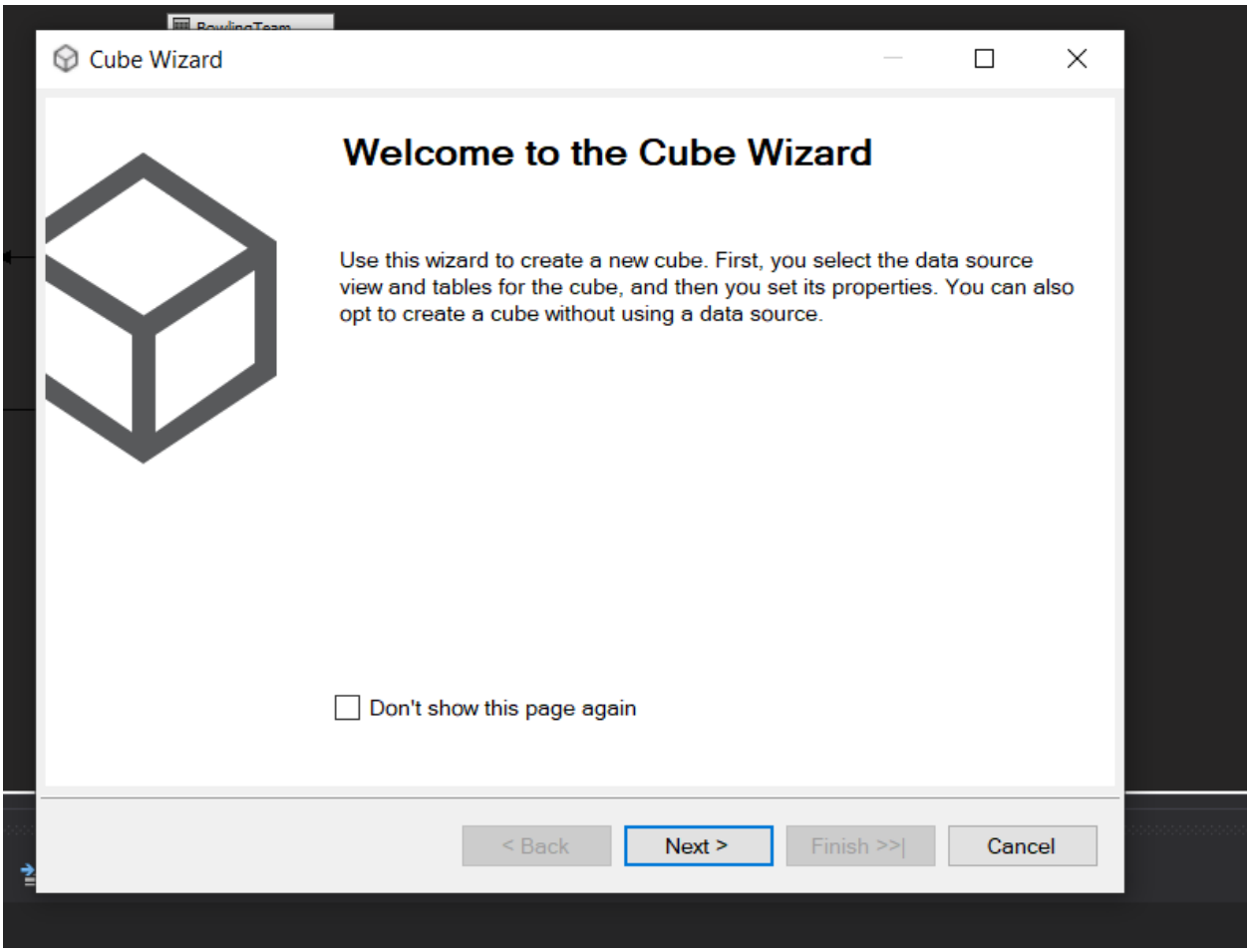


Fig 50: Cube Deployment Wizard

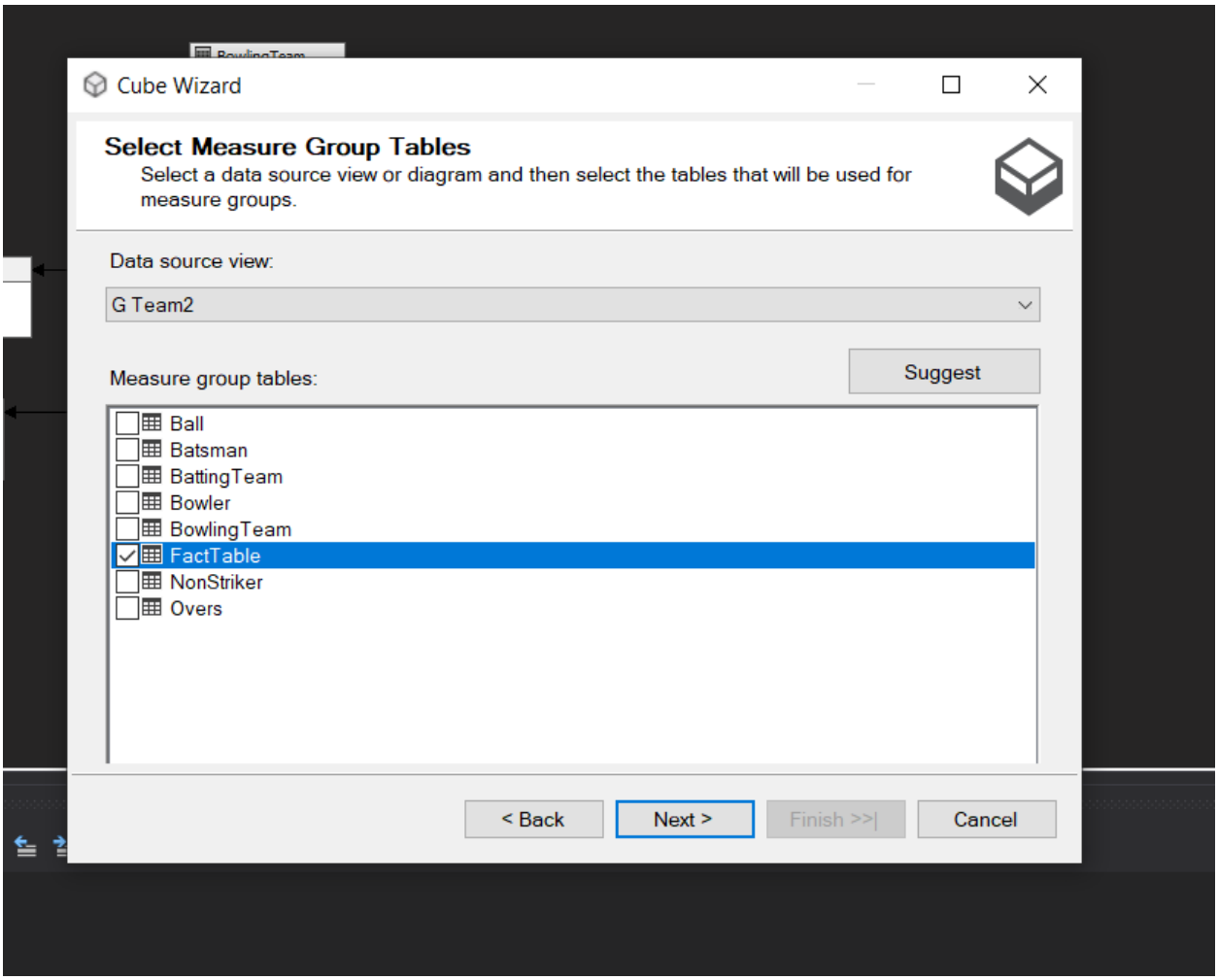


Fig 51: Checking Fact table as Measure Group Tables

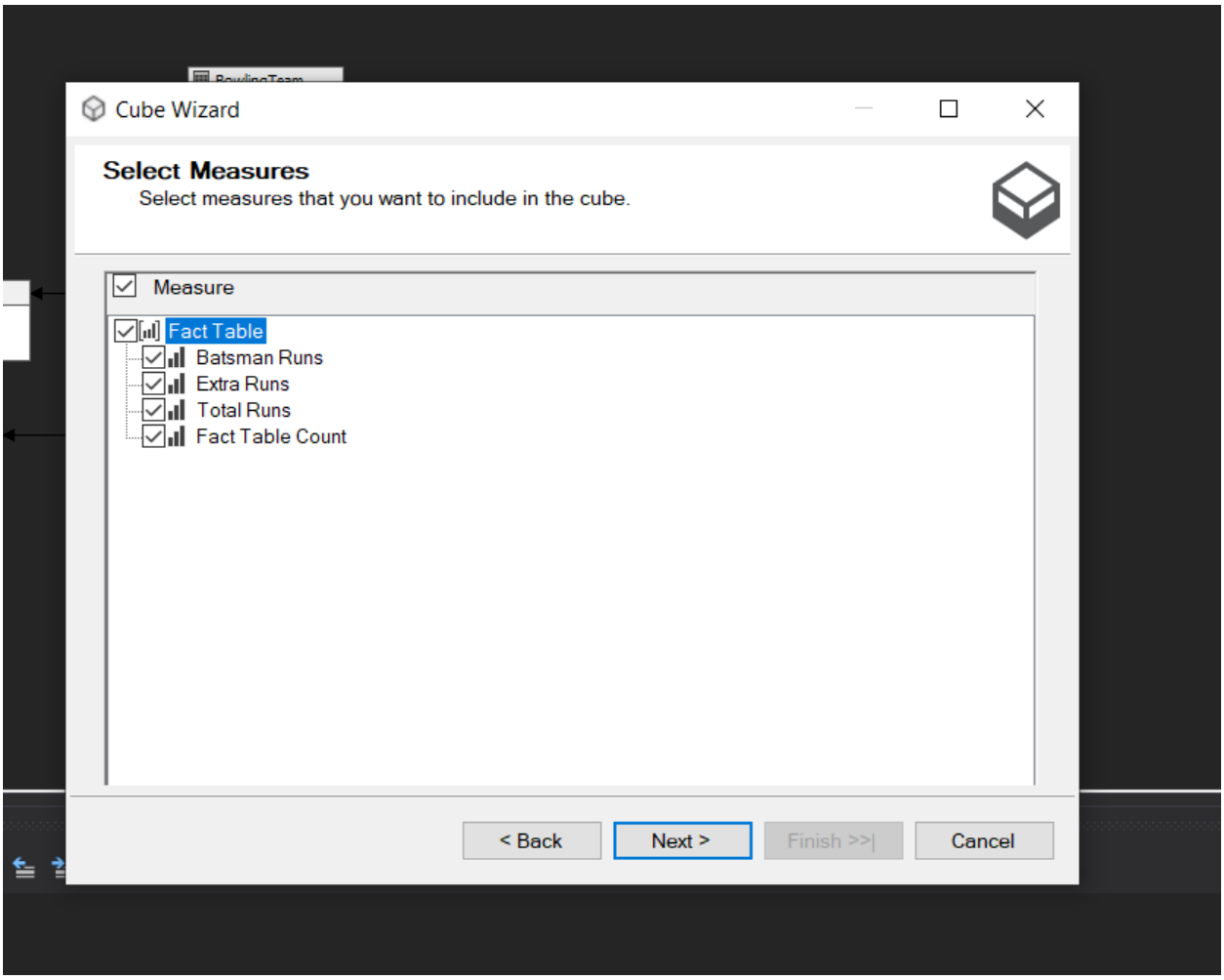


Fig 52: Checking Dimensions

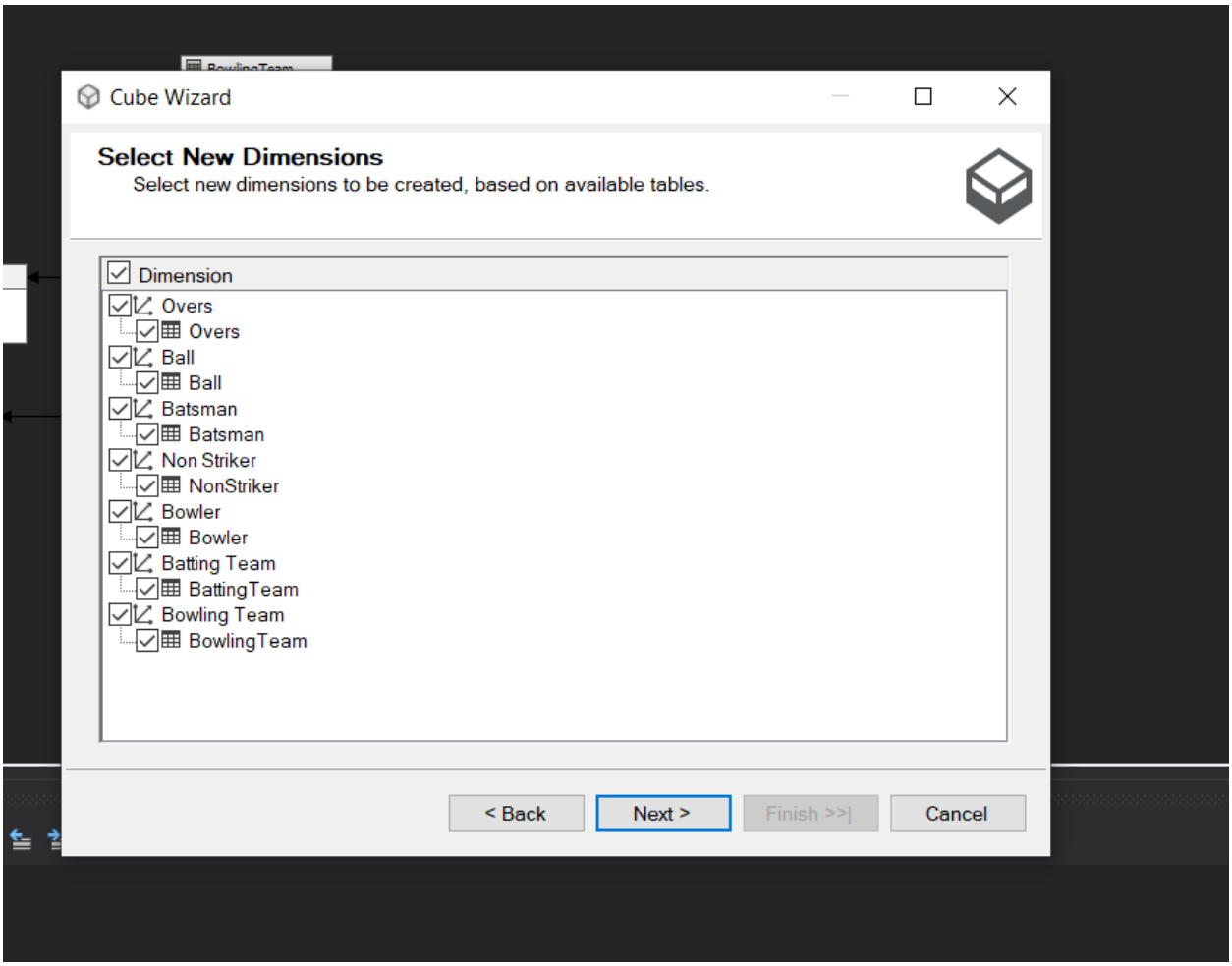


Fig 53: Checking New Dimensions

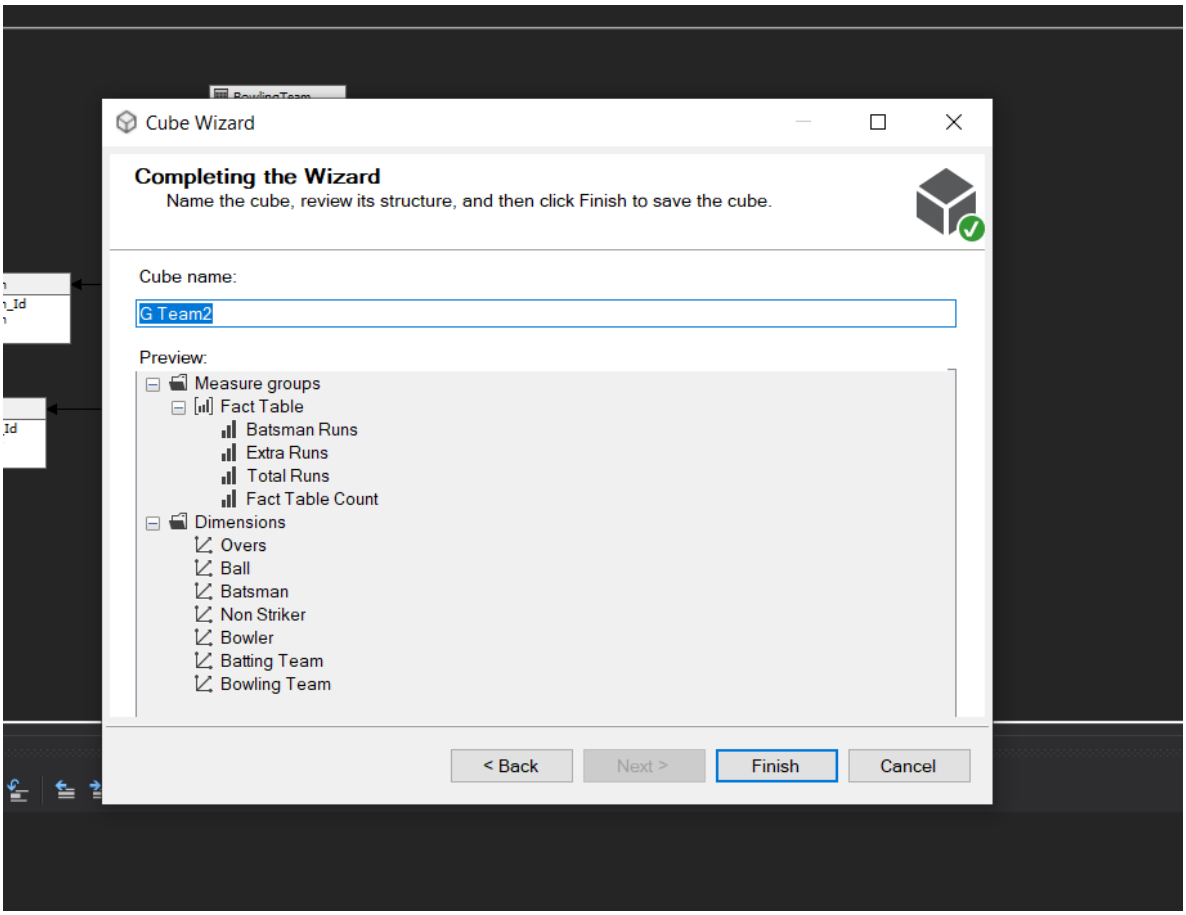


Fig 54: Completing the Wizard

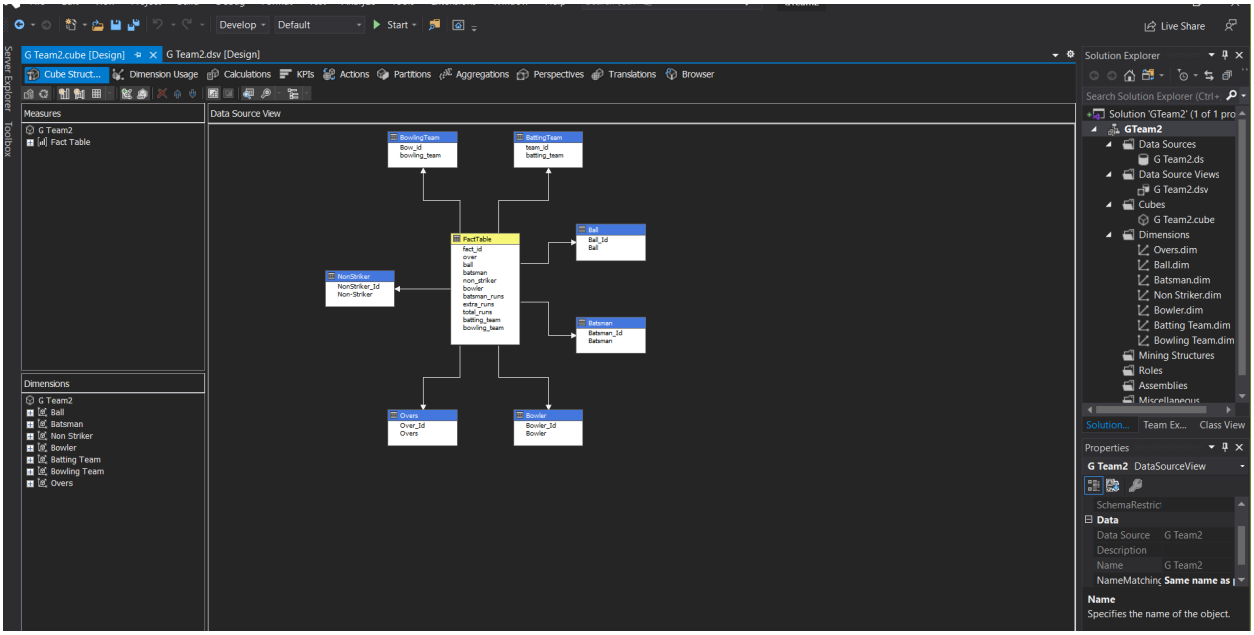


Fig 55: Cube Design View

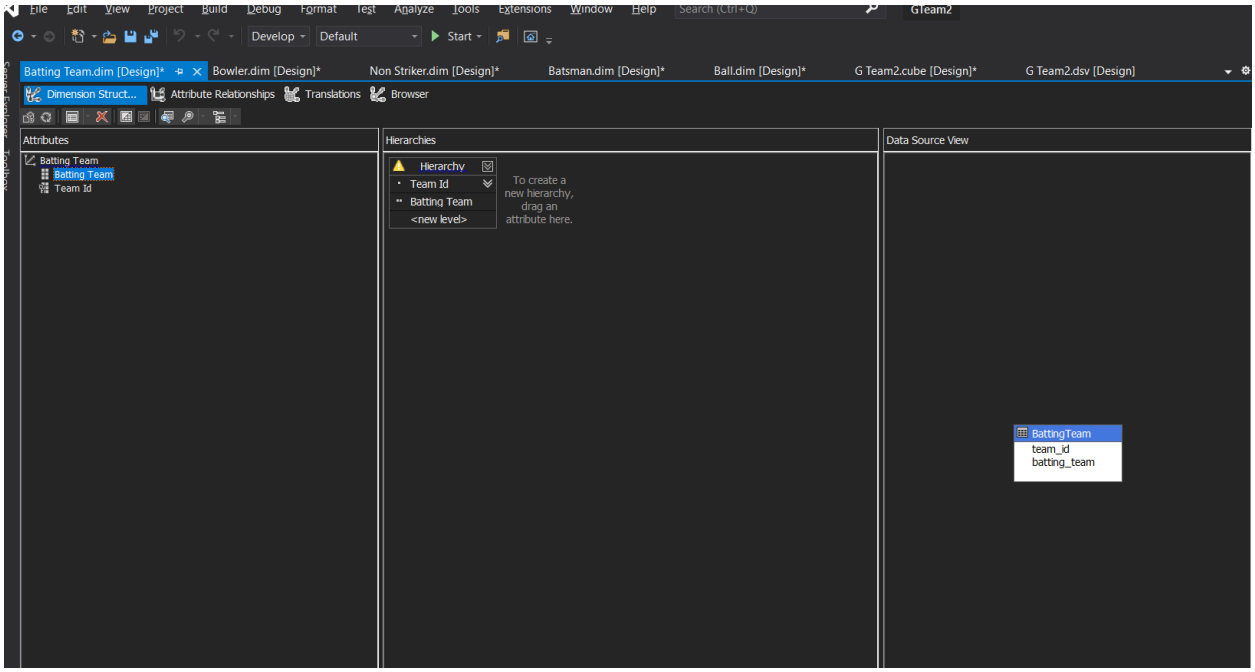


Fig 56: Batting Dimension

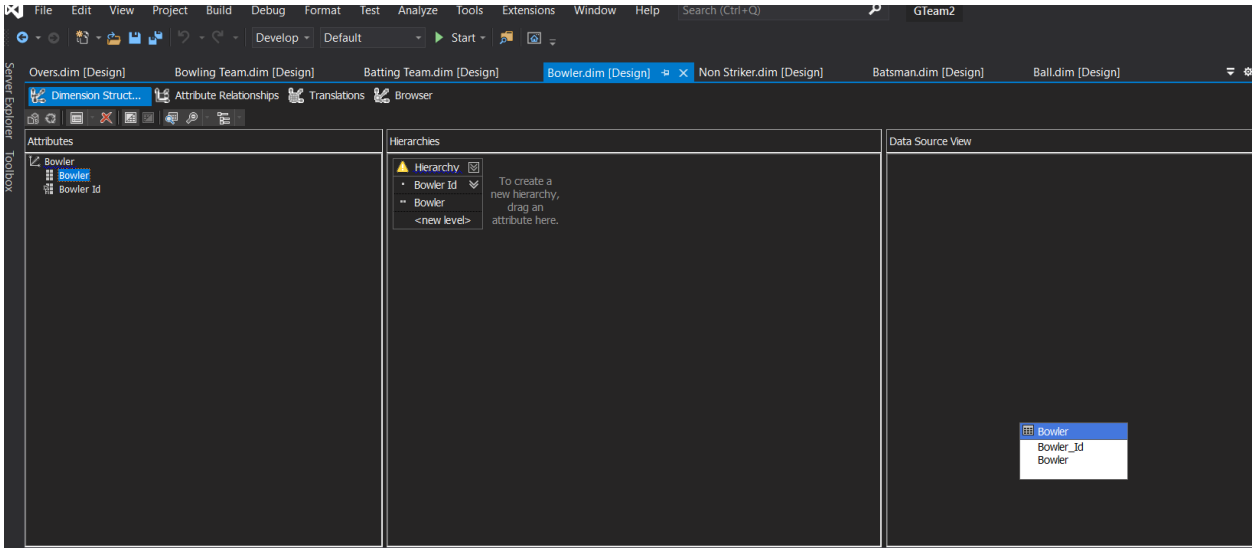


Fig 57: Bowler Dimension

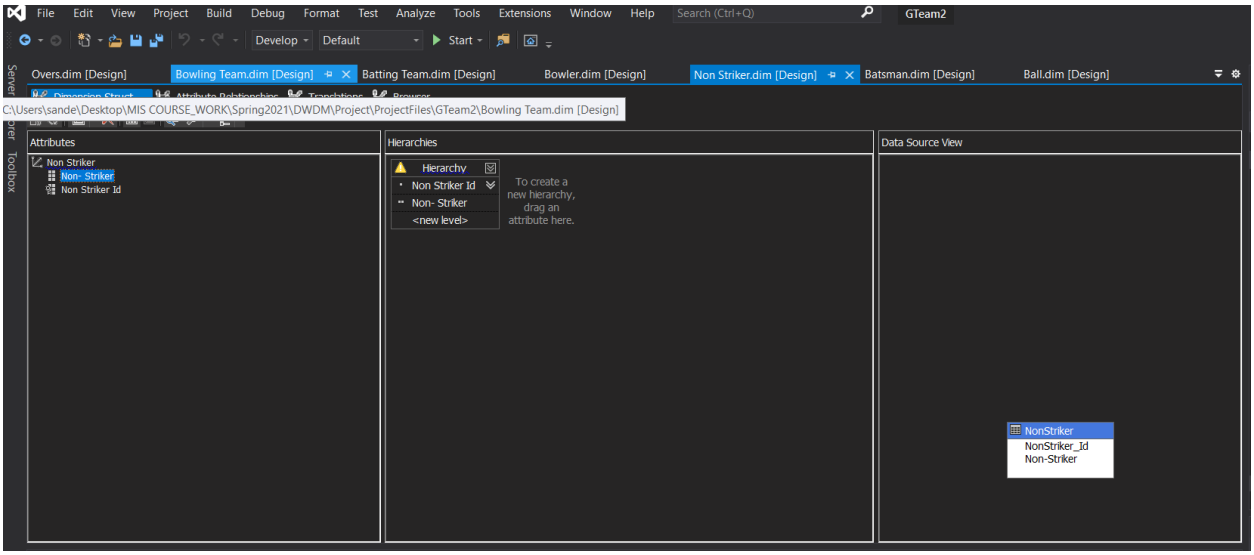


Fig 58: Non-Striker Dimension

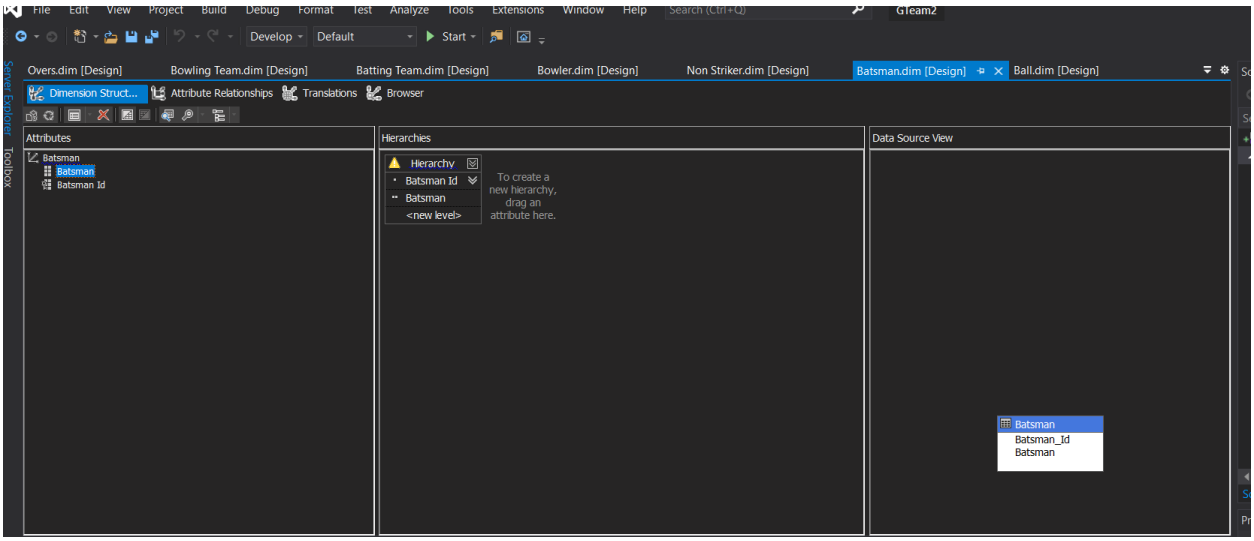


Fig 59: Batsman Dimension

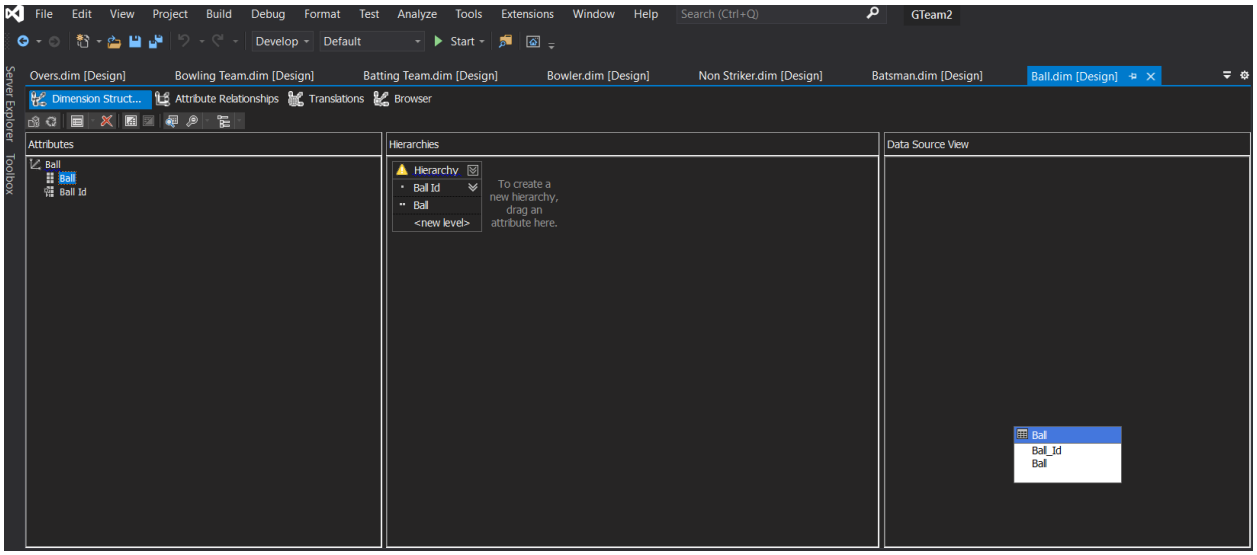


Fig 60: Ball Dimension

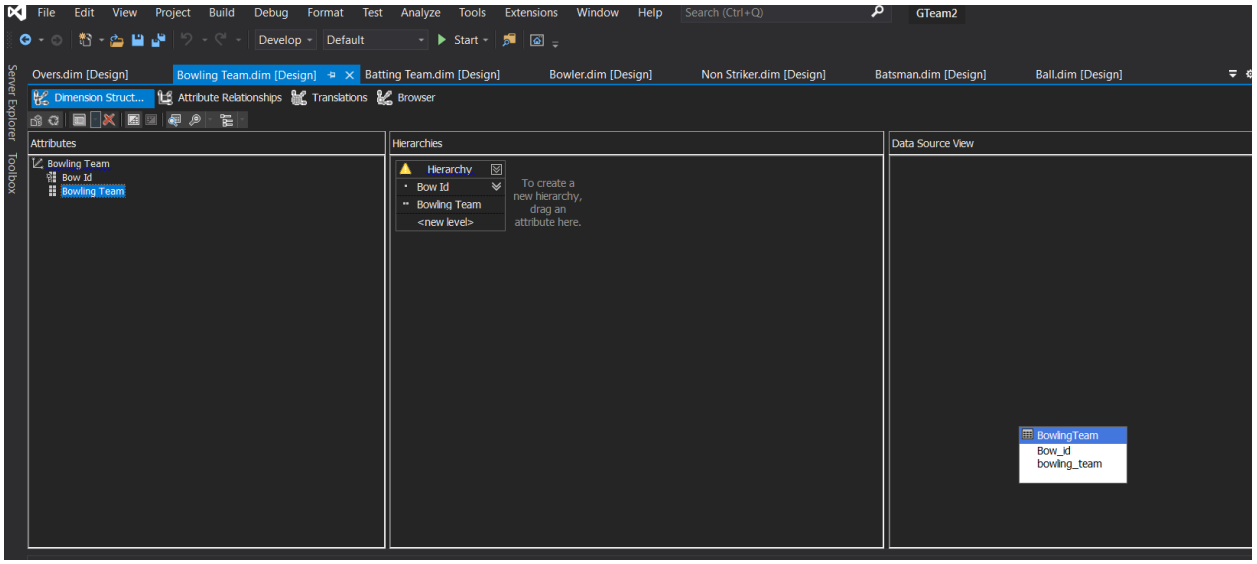


Fig 61: BowlingTeam Dimension

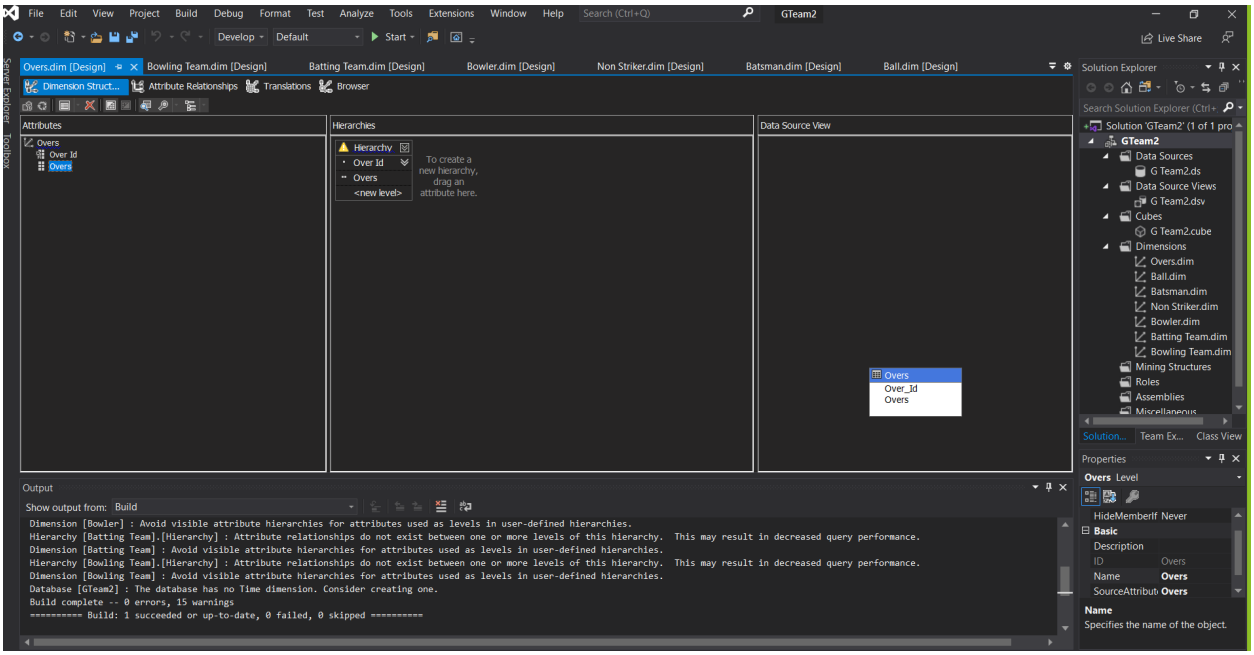
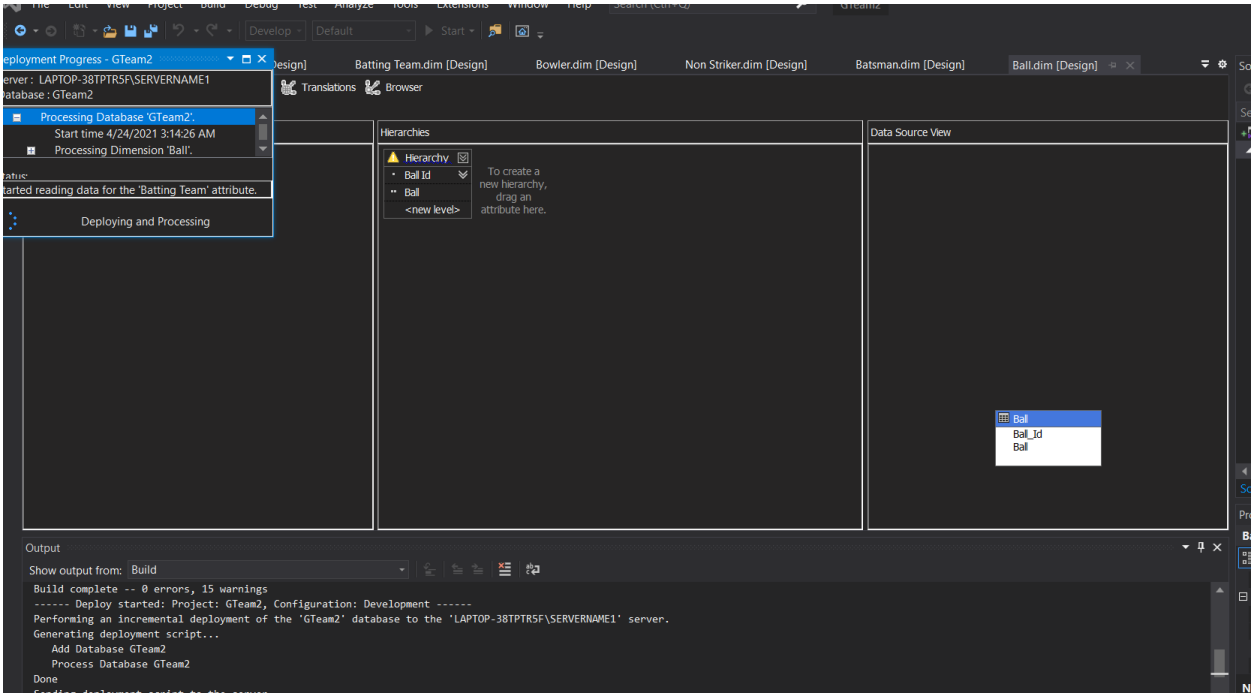


Fig 62: Over Dimension



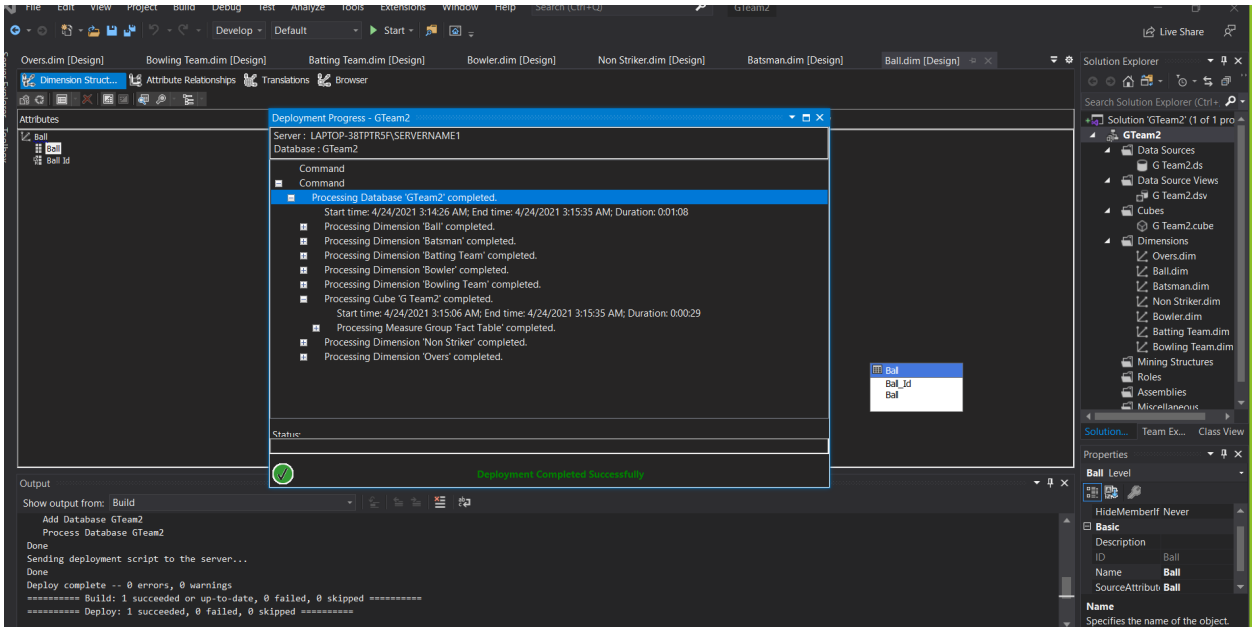


Fig 63: Deploy Successful

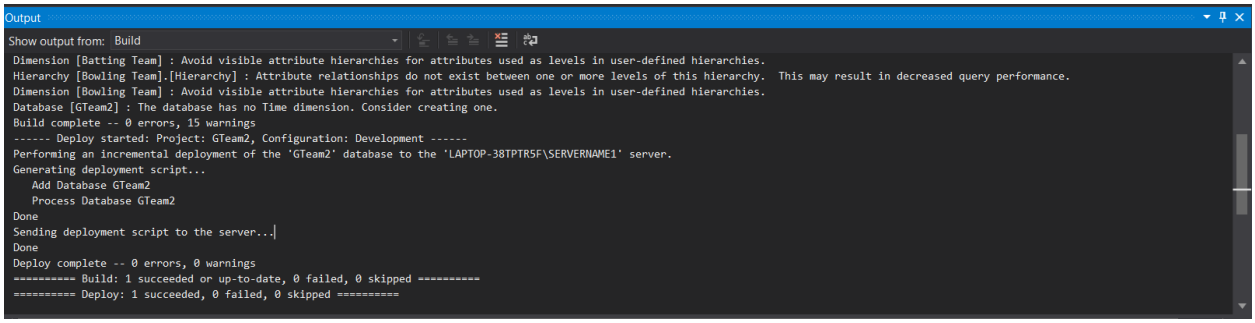


Fig 64: Build Successful

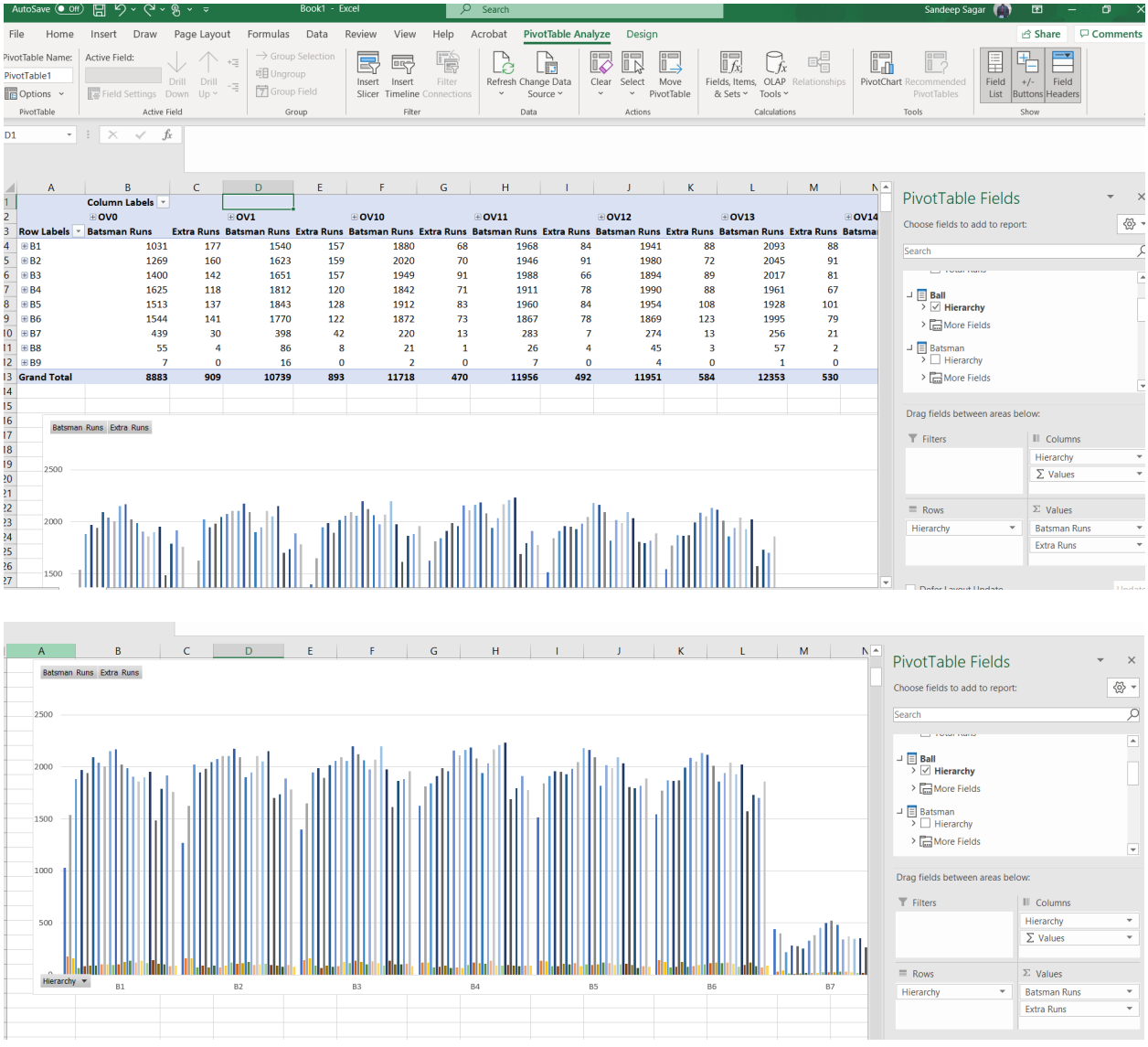
Reports:

Finally, the data we processed using data mining and data warehousing is to extract information and to make choices. The information which we generated is used to make certain vital reports. Working with the cube to process this information is a challenging task for administration, individuals in the IT industry can understand these reports or conclusions but the administrative officials have no idea what this information is so, now in order to make it clear the reports generated are discernible and can be drawn upon by the officials of the administration. This can be done by cube.

From the data we processed in this project, we can conclude some certain analysis and reports which are as follows:

Analysis - 1:

- The below report depicts how many runs were scored by the batsman for ball number in each over of all the ipl seasons.
- And also it also gives the total number of extra runs were given in each over in each game.
- From the analysis maximum runs scored by batsman is in Over No-17.
- And ball no 4 has conceded the most number of runs in an over in overall seasons.

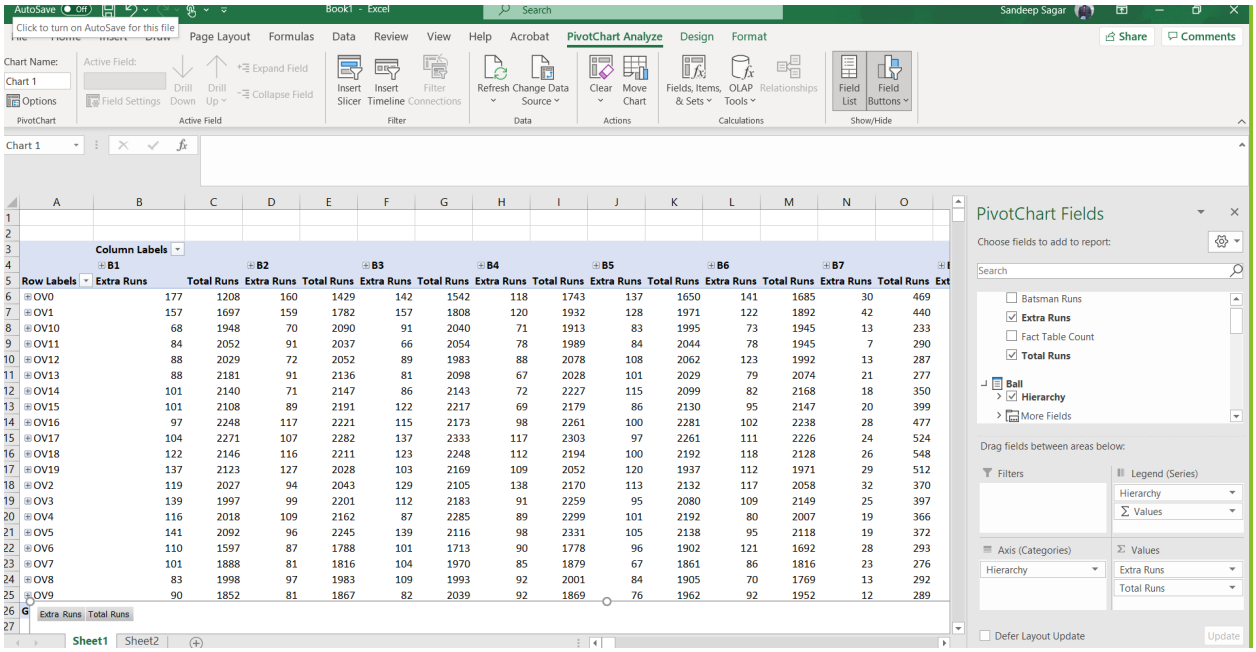


Column Labels	OV0		OV1		OV10		OV11		OV12		OV13		OV14	
Row Labels	Batsman Runs	Extra Runs	Batsman Runs	Extra Runs	Batsman Runs	Extra Runs	Batsman Runs	Extra Runs	Batsman Runs	Extra Runs	Batsman Runs	Extra Runs	Batsma	
B1	1031	177	1540	157	1880	68	1968	84	1941	88	2093	88		
B2	1269	160	1623	159	2020	70	1946	91	1980	72	2045	91		
B3	1400	142	1651	157	1949	91	1988	66	1894	89	2017	81		
B4	1625	118	1812	120	1842	71	1911	78	1990	88	1961	67		
B5	1513	137	1843	128	1912	83	1960	84	1954	108	1928	101		
B6	1544	141	1770	122	1872	73	1867	78	1869	123	1995	79		
B7	439	30	398	42	220	13	283	7	274	13	256	21		
B8	55	4	86	8	21	1	26	4	45	3	57	2		
B9	7	0	16	0	2	0	7	0	4	0	1	0		
Grand Total	8883	909	10739	893	11718	470	11956	492	11951	584	12353	530		

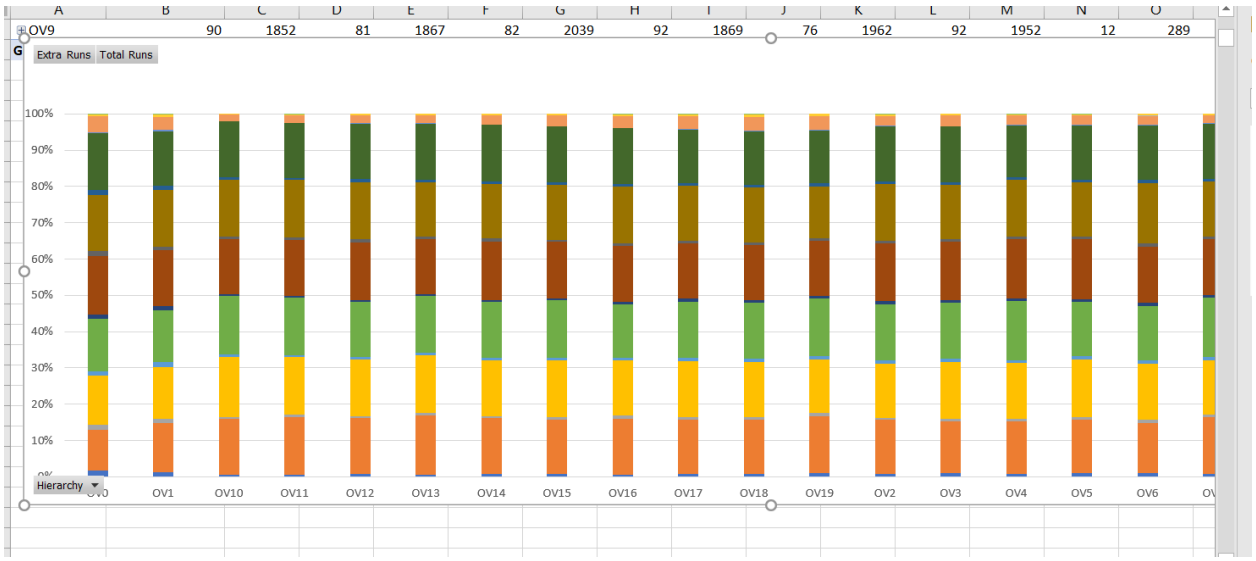
Fig: Analysis 1

Analysis - 2:

- From the below report represents how many runs and extra runs were scored in every over for every ball for all the seasons by the batsmans.
- We can say that Over 0 is the least in conceding runs and extra runs.
- And Ball 5 in every over is considered to be the most expensive one with 41485 runs.

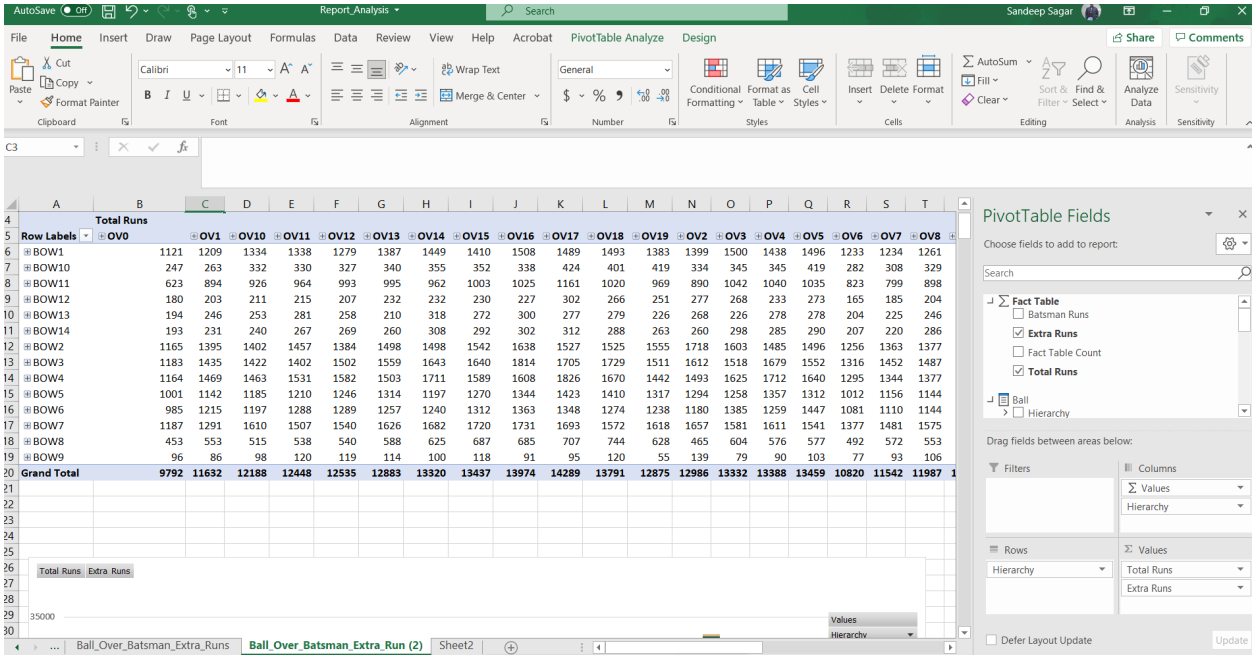


Row Labels	B1 Extra Runs	B1 Total Runs	B2 Extra Runs	B2 Total Runs	B3 Extra Runs	B3 Total Runs	B4 Extra Runs	B4 Total Runs	B5 Extra Runs	B5 Total Runs	B6 Extra Runs	B6 Total Runs	B7 Extra Runs	B7 Total Runs
OV0	177	1208	160	1429	142	1542	118	1743	137	1650	141	1685	30	469
OV1	157	1697	159	1782	157	1808	120	1932	128	1971	122	1892	42	440
OV10	68	1948	70	2090	91	2040	71	1913	83	1995	73	1945	13	233
OV11	84	2052	91	2037	66	2054	78	1989	84	2044	78	1945	7	290
OV12	88	2029	72	2052	89	1983	88	2078	108	2062	123	1992	13	287
OV13	88	2181	91	2136	81	2098	67	2028	101	2029	79	2074	21	277
OV14	101	2140	71	2147	86	2143	72	2227	115	2099	82	2168	18	350
OV15	101	2108	89	2191	122	2217	69	2179	86	2130	95	2147	20	399
OV16	97	2248	117	2221	115	2173	98	2261	100	2281	102	2238	28	477
OV17	104	2271	107	2282	137	2333	117	2303	97	2261	111	2226	24	524
OV18	122	2146	116	2211	123	2248	112	2194	100	2192	118	2128	26	548
OV19	137	2123	127	2028	103	2169	109	2052	120	1937	112	1971	29	512
OV2	119	2027	94	2043	129	2105	138	2170	113	2132	117	2058	32	370
OV3	139	1997	99	2201	112	2183	91	2259	95	2080	109	2149	25	397
OV4	116	2018	109	2162	87	2285	89	2299	101	2192	80	2007	19	366
OV5	141	2092	96	2245	139	2116	98	2331	105	2138	95	2118	19	372
OV6	110	1597	87	1788	101	1713	90	1778	96	1902	121	1692	28	293
OV7	101	1888	81	1816	104	1970	85	1879	67	1861	86	1816	23	276
OV8	83	1998	97	1983	109	1993	92	2001	84	1905	70	1769	13	292
OV9	90	1852	81	1867	82	2039	92	1869	76	1962	92	1952	12	289

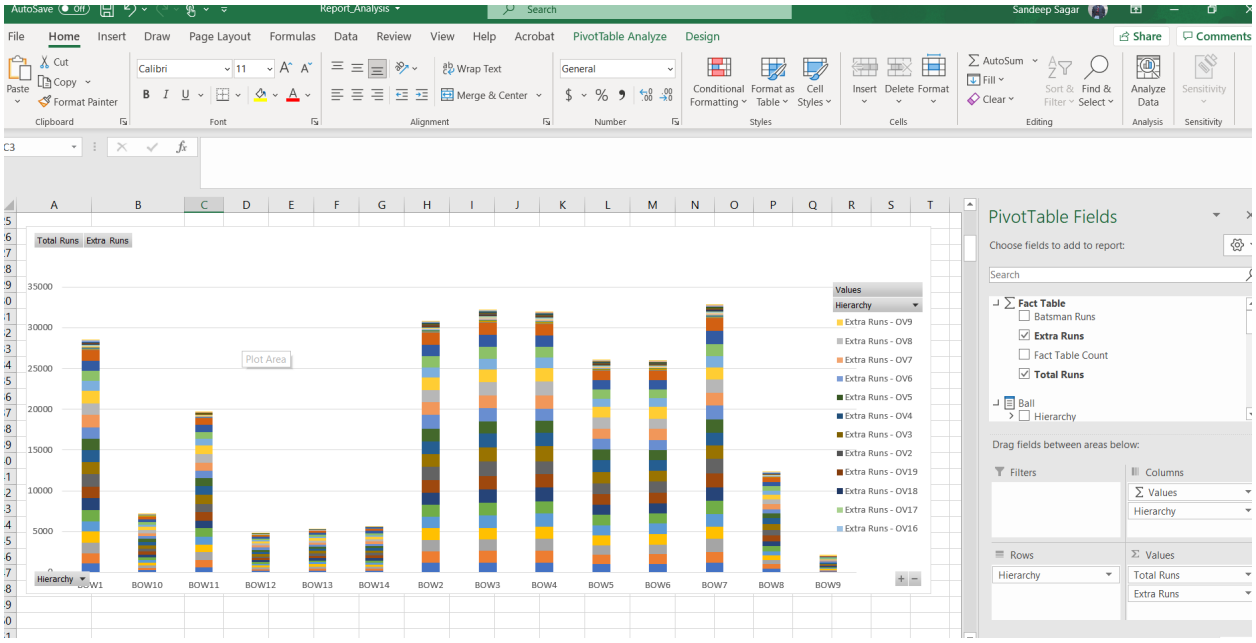


Analysis - 3:

- The below reports gives which bowler conceded many runs in each of the over he bowled in the entire IPL.
- Over-17 is the highest runs given over by every bowler who bowled that over with 14289 runs.

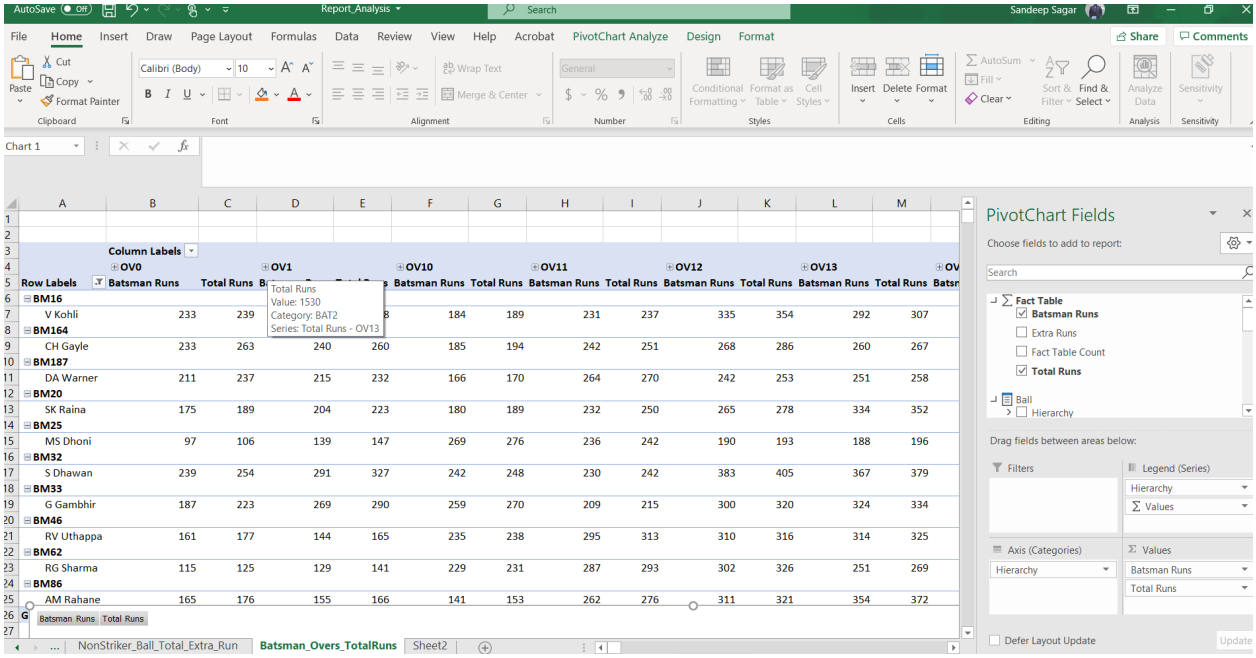


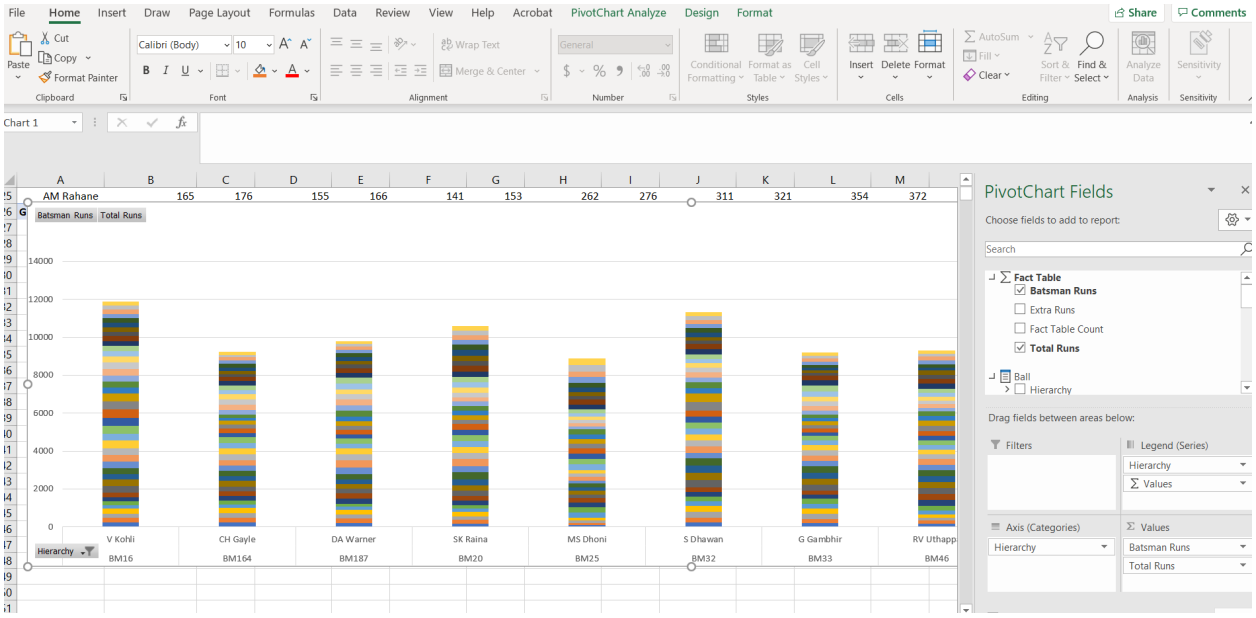
Row Labels	OV0	OV1	OV10	OV11	OV12	OV13	OV14	OV15	OV16	OV17	OV18	OV19	OV2	OV3	OV4	OV5	OV6	OV7	OV8
BOW1	1121	1209	1334	1338	1279	1387	1449	1410	1508	1489	1493	1383	1399	1500	1438	1496	1233	1234	1261
BOW10	247	263	332	330	327	340	355	352	338	424	401	419	334	345	345	419	282	308	329
BOW11	623	894	926	964	993	995	962	1003	1025	1161	1020	969	890	1042	1040	1035	823	799	898
BOW12	180	203	211	215	207	232	232	230	227	302	266	251	277	268	233	273	165	185	204
BOW13	194	246	253	281	258	210	318	272	300	277	279	226	268	226	278	278	204	225	246
BOW14	193	231	240	267	269	260	308	292	302	312	288	263	260	298	285	290	207	220	286
BOW2	1165	1395	1402	1457	1384	1498	1498	1542	1638	1527	1525	1555	1718	1603	1485	1496	1256	1363	1377
BOW3	1183	1435	1422	1402	1502	1559	1643	1640	1814	1705	1729	1511	1612	1518	1679	1552	1316	1452	1487
BOW4	1164	1469	1463	1531	1582	1503	1711	1589	1608	1826	1670	1442	1493	1625	1712	1640	1295	1344	1377
BOW5	1001	1142	1185	1210	1246	1314	1197	1270	1344	1423	1410	1317	1294	1258	1357	1312	1012	1156	1144
BOW6	985	1215	1197	1288	1289	1257	1240	1312	1363	1348	1274	1238	1180	1385	1259	1447	1081	1110	1144
BOW7	1187	1291	1610	1507	1540	1626	1682	1720	1731	1693	1572	1618	1657	1581	1611	1541	1377	1481	1575
BOW8	453	553	515	538	540	588	625	687	685	707	744	628	465	604	576	577	492	572	553
BOW9	96	86	98	120	119	114	100	118	91	95	120	55	139	79	90	103	77	93	106
Grand Total	9792	11632	12188	12448	12535	12883	13320	13437	13974	14289	13791	12875	12986	13332	13388	13459	10820	11542	11987



Analysis - 4:

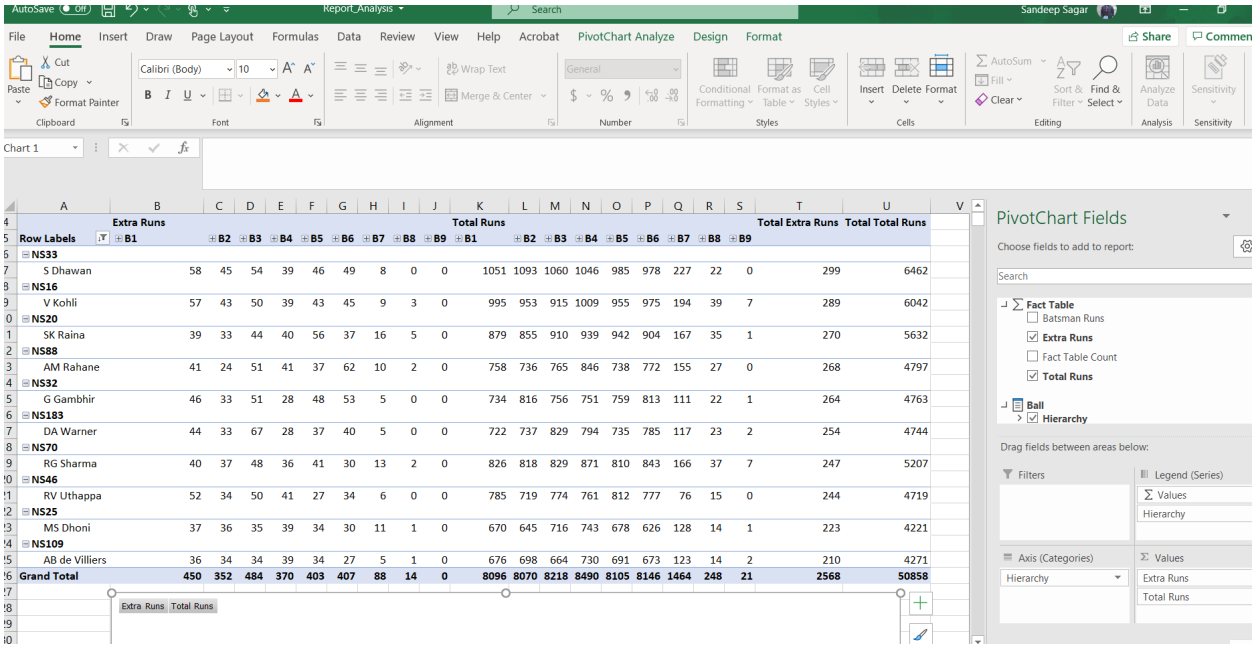
- The below analysis shows graphs between total numbers of runs scored by batsmans in all seasons of IPL where it also says how many runs were scored in an over by batsman in their entire career.
- Virat Kohil has scored the most number of runs among all other batsmans in the IPL.





Analysis - 5:

- The reports conclude that how many runs were scored by the non striker batsman in all of the matches.
- Where S Dhawan as a Non Striker, batsman has scored maximum runs of 1051 which concludes that he is a good non striker batsman among all other batters in the IPL.



Row Labels	Extra Runs B1	B2	B3	B4	B5	B6	B7	B8	B9	Total Runs B1	B2	B3	B4	B5	B6	B7	B8	B9	Total Extra Runs	Total Total Runs
NS33																				
S Dhawan	58	45	54	39	46	49	8	0	0	1051	1093	1060	1046	985	978	227	22	0	299	6462
NS16																				
V Kohli	57	43	50	39	43	45	9	3	0	995	953	915	1009	955	975	194	39	7	289	6042
NS20																				
SK Raina	39	33	44	40	56	37	16	5	0	879	855	910	939	942	904	167	35	1	270	5632
NS88																				
AM Rahane	41	24	51	41	37	62	10	2	0	758	736	765	846	738	772	155	27	0	268	4797
NS32																				
G Gambhir	46	33	51	28	48	53	5	0	0	734	816	756	751	759	813	111	22	1	264	4763
NS183																				
DA Warner	44	33	67	28	37	40	5	0	0	722	737	829	794	735	785	117	23	2	254	4744
NS70																				
RG Sharma	40	37	48	36	41	30	13	2	0	826	818	829	871	810	843	166	37	7	247	5207
NS46																				
RV Uthappa	52	34	50	41	27	34	6	0	0	785	719	774	761	812	777	76	15	0	244	4719
NS25																				
MS Dhoni	37	36	35	39	34	30	11	1	0	670	645	716	743	678	626	128	14	1	223	4221
NS109																				
AB de Villiers	36	34	34	39	34	27	5	1	0	676	698	664	730	691	673	123	14	2	210	4271
Grand Total	450	352	484	370	403	407	88	14	0	8096	8070	8218	8490	8105	8146	1464	248	21	2568	50858

Data Mining Queries

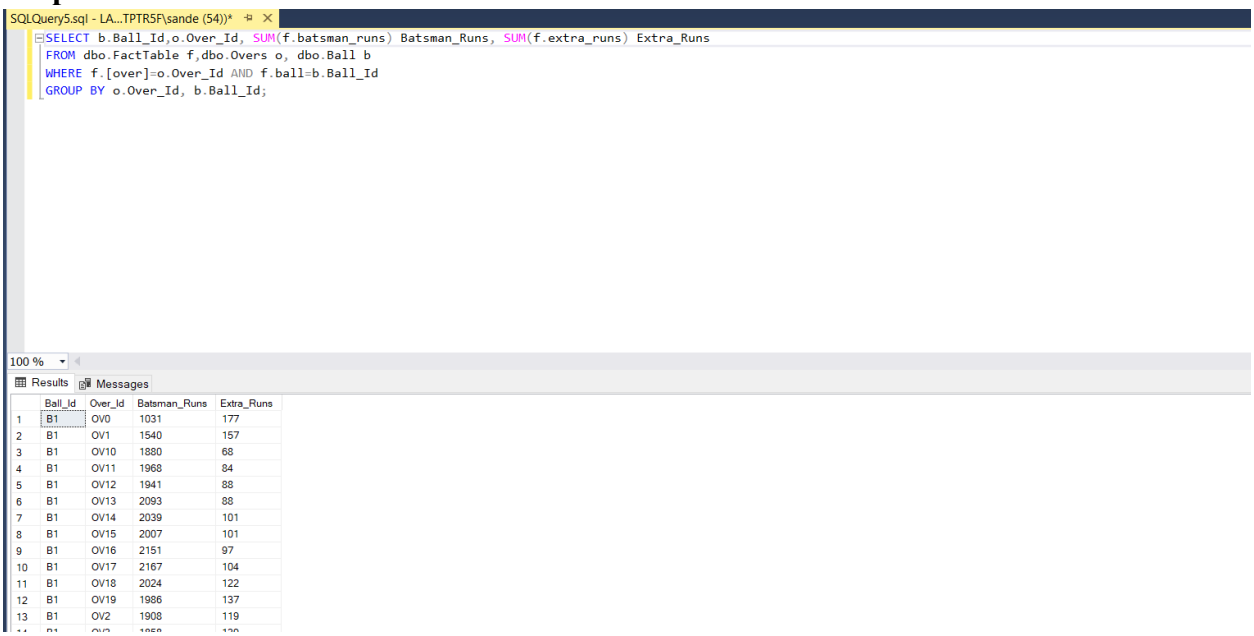
A desired feature of data mining systems is the ability to support ad hoc and interactive data mining to facilitate flexible and effective knowledge discovery. Data mining queries can be designed to support that such a feature. The primitive of data mining query language are:

- More flexible user interaction.
- Graphical explanation of complex data for clear views.
- Can retrieve cases that match patterns.
- Can analyze or predict from the current data.

Query – 1

```
SELECT b.Ball_Id,o.Over_Id, SUM(f.batsman_runs) Batsman_Runs, SUM(f.extra_runs) Extra_Runs
FROM dbo.FactTable f,dbo.Overs o, dbo.Ball b
WHERE f.[over]=o.Over_Id AND f.ball=b.Ball_Id
GROUP BY o.Over_Id, b.Ball_Id;
```

Output:



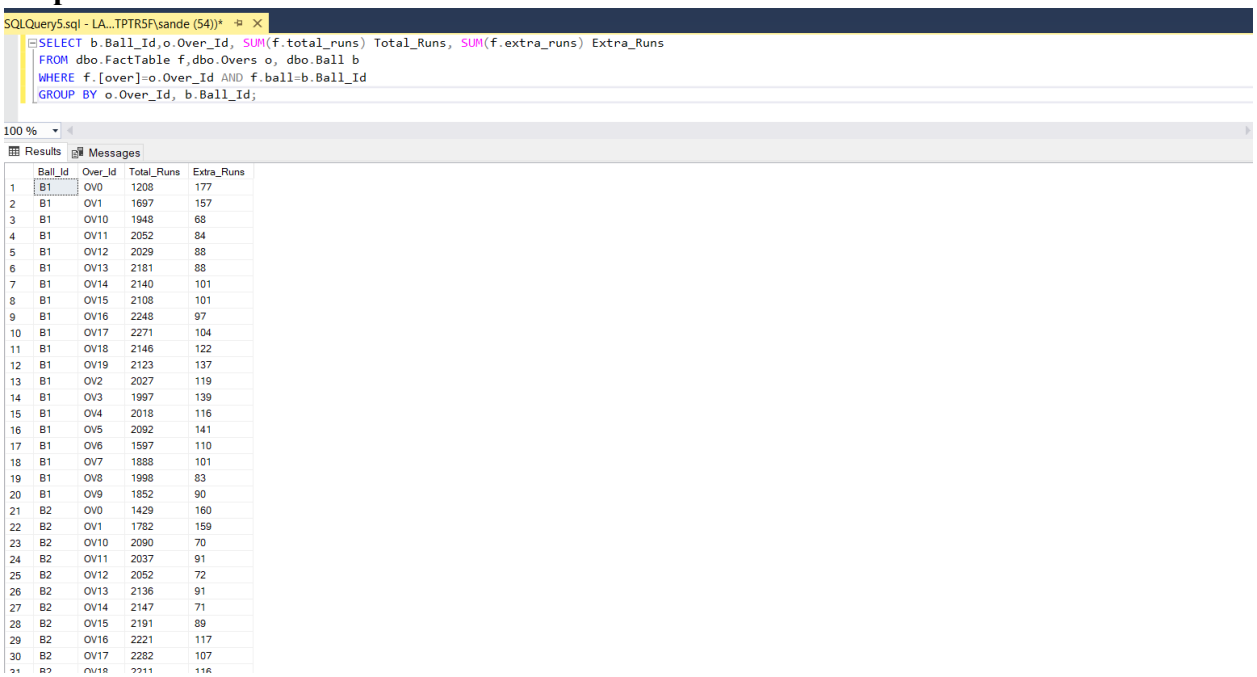
	Ball_Id	Over_Id	Batsman_Runs	Extra_Runs
1	B1	OV0	1031	177
2	B1	OV1	1540	157
3	B1	OV10	1880	68
4	B1	OV11	1968	84
5	B1	OV12	1941	88
6	B1	OV13	2093	88
7	B1	OV14	2039	101
8	B1	OV15	2007	101
9	B1	OV16	2151	97
10	B1	OV17	2167	104
11	B1	OV18	2024	122
12	B1	OV19	1986	137
13	B1	OV2	1908	119

Fig. No: Query Output 1

Query – 2

```
SELECT b.Ball_Id,o.Over_Id, SUM(f.total_runs) Total_Runs, SUM(f.extra_runs) Extra_Runs
FROM dbo.FactTable f,dbo.Overs o, dbo.Ball b
WHERE f.[over]=o.Over_Id AND f.ball=b.Ball_Id
GROUP BY o.Over_Id, b.Ball_Id;
```

Output:



SQLQuery5.sql - LA...TPTR5F\sande (54))*

```
SELECT b.Ball_Id,o.Over_Id, SUM(f.total_runs) Total_Runs, SUM(f.extra_runs) Extra_Runs
FROM dbo.FactTable f,dbo.Overs o, dbo.Ball b
WHERE f.[over]=o.Over_Id AND f.ball=b.Ball_Id
GROUP BY o.Over_Id, b.Ball_Id;
```

100 %

Results | Messages

	Ball_Id	Over_Id	Total_Runs	Extra_Runs
1	B1	OV0	1208	177
2	B1	OV1	1697	157
3	B1	OV10	1948	68
4	B1	OV11	2052	84
5	B1	OV12	2029	88
6	B1	OV13	2181	88
7	B1	OV14	2140	101
8	B1	OV15	2108	101
9	B1	OV16	2248	97
10	B1	OV17	2271	104
11	B1	OV18	2146	122
12	B1	OV19	2123	137
13	B1	OV2	2027	119
14	B1	OV3	1997	139
15	B1	OV4	2018	116
16	B1	OV5	2092	141
17	B1	OV6	1597	110
18	B1	OV7	1888	101
19	B1	OV8	1998	83
20	B1	OV9	1852	90
21	B2	OV0	1429	160
22	B2	OV1	1782	159
23	B2	OV10	2090	70
24	B2	OV11	2037	91
25	B2	OV12	2052	72
26	B2	OV13	2136	91
27	B2	OV14	2147	71
28	B2	OV15	2191	89
29	B2	OV16	2221	117
30	B2	OV17	2282	107

Fig. No: Query Output 2

Query – 3

```
SELECT b.team_id,o.Over_Id, SUM(f.total_runs) Total_Runs, SUM(f.extra_runs) Extra_Runs
FROM dbo.FactTable f,dbo.Overs o, dbo.BattingTeam b
WHERE f.[over]=o.Over_Id AND f.batting_team = b.team_id
GROUP BY o.Over_Id, b.team_id;
```

Output:

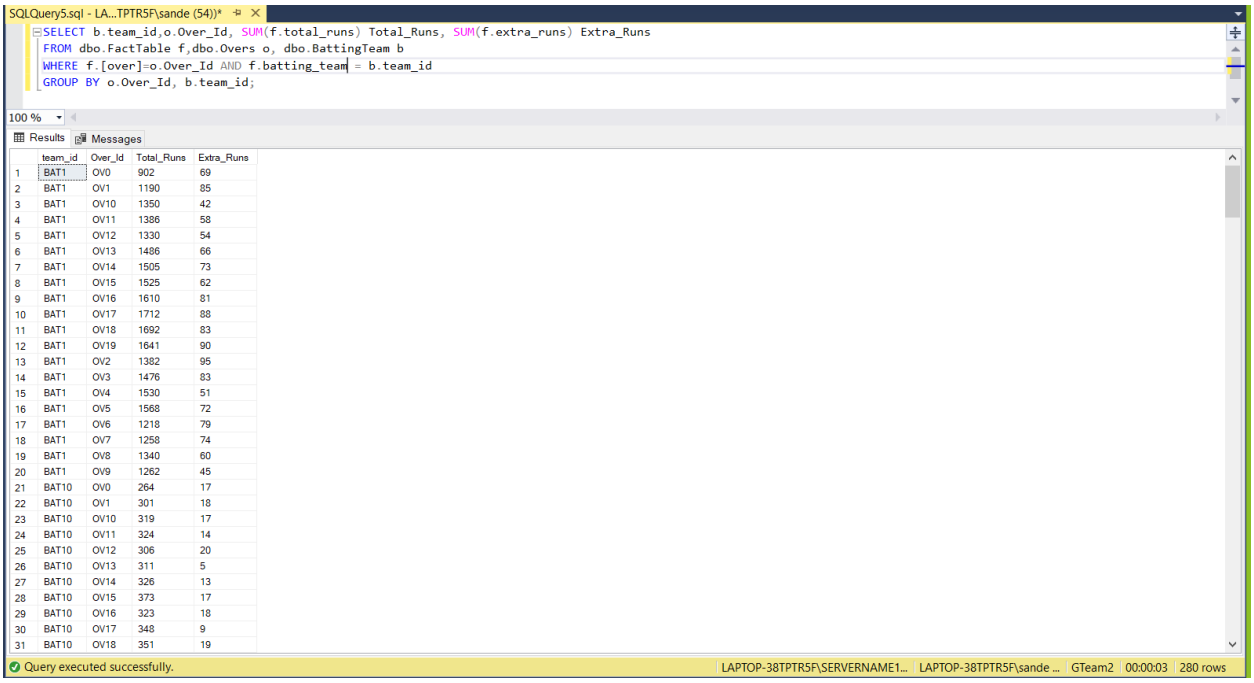


Fig. No: Query Output 3

Query – 4

```
SELECT b.Batsman_Id,o.Over_Id, SUM(f.total_runs) Total_Runs, SUM(f.batsman_runs) Batsman_Runs
FROM dbo.FactTable f,dbo.Overs o, dbo.Batsman b
WHERE f.[over]=o.Over_Id AND f.batsman = b.Batsman_Id
GROUP BY o.Over_Id, b.Batsman_Id
ORDER BY Total_Runs desc;
```

Output:

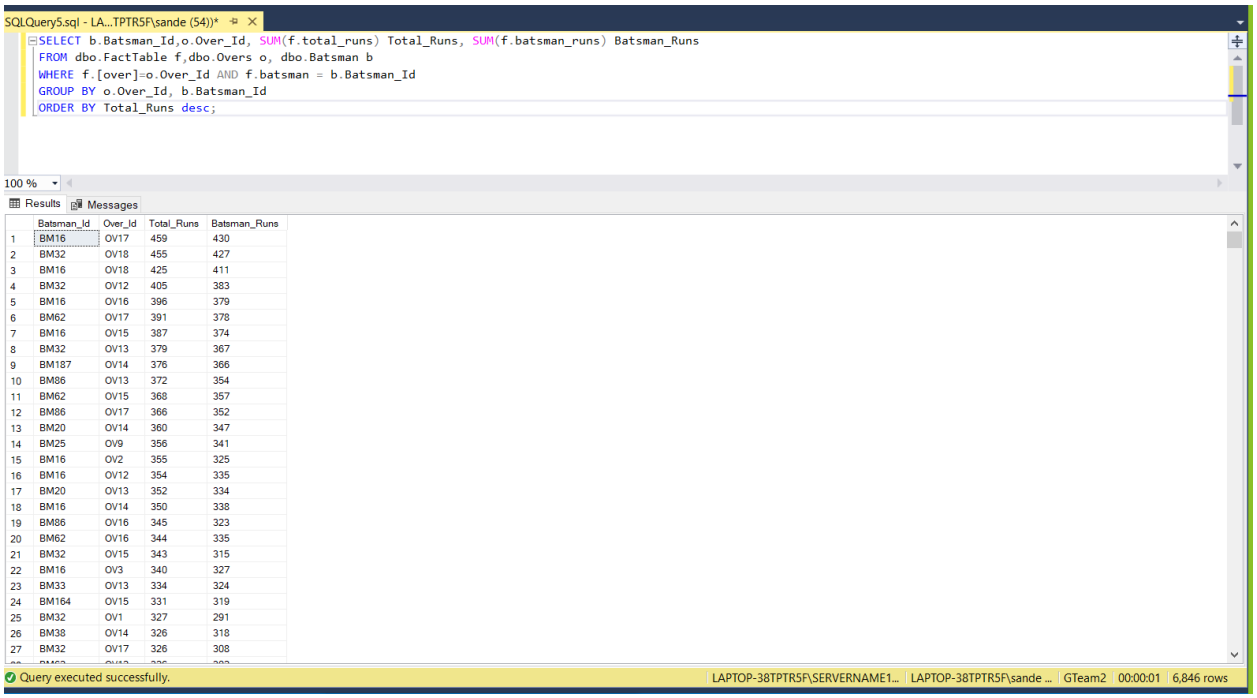
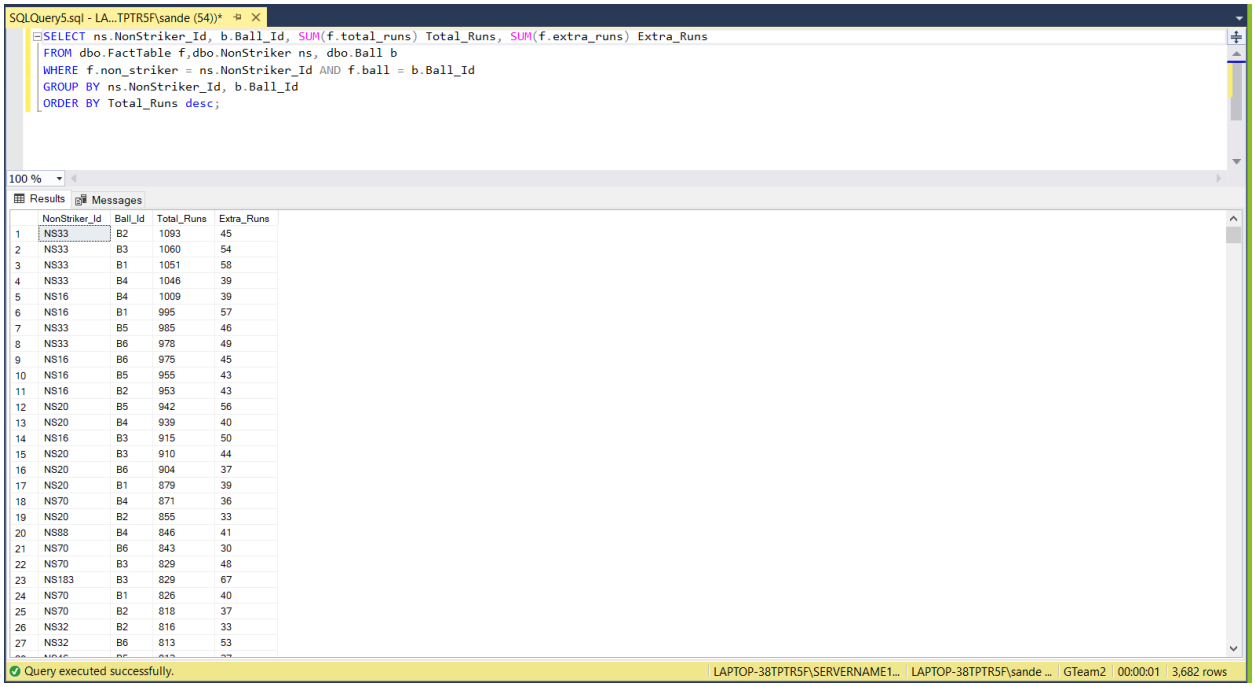


Fig. No: Query Output 4

Query – 5

```
SELECT ns.NonStriker_Id, b.Ball_Id, SUM(f.total_runs) Total_Runs, SUM(f.extra_runs) Extra_Runs
 FROM dbo.FactTable f, dbo.NonStriker ns, dbo.Ball b
 WHERE f.non_striker = ns.NonStriker_Id AND f.ball = b.Ball_Id
 GROUP BY ns.NonStriker_Id, b.Ball_Id
 ORDER_BY Total_Runs desc;
```

Output:



	NonStriker_Id	Ball_Id	Total_Runs	Extra_Runs
1	NS33	B2	1093	45
2	NS33	B3	1060	54
3	NS33	B1	1051	58
4	NS33	B4	1046	39
5	NS16	B4	1009	39
6	NS16	B1	995	57
7	NS33	B5	985	46
8	NS33	B6	978	49
9	NS16	B6	975	45
10	NS16	B5	955	43
11	NS16	B2	953	43
12	NS20	B5	942	56
13	NS20	B4	939	40
14	NS16	B3	915	50
15	NS20	B3	910	44
16	NS20	B6	904	37
17	NS20	B1	879	39
18	NS70	B4	871	36
19	NS20	B2	855	33
20	NS88	B4	846	41
21	NS70	B6	843	30
22	NS70	B3	829	48
23	NS183	B3	829	67
24	NS70	B1	826	40
25	NS70	B2	818	37
26	NS32	B2	816	33
27	NS32	B6	813	53

Fig. No: Query Output 5

Conclusion

- From our dataset analysis, we were able to deduce how many runs, extra runs were scored by batsmans for each over for a particular ball individually in all of the IPL Seasons.
- Among all the batters, we found that Virat Kohil has the highest runs among all the batsmans in entire IPL seasons.
- Additionally, we also found which ball of each over bowled is conceding the highest percentage of runs.
- Concluding the fact that, each over and each ball is an important parameter in this dataset through which we can find many factors which tells about the batsmans, bowlers and the economy rates and also strike rates of the batsmans.

References

- https://www.congruentsoft.com/data-warehousing-development.aspx
- https://digitaltransformationpro.com/data-mining-steps/
- https://www.kaggle.com/patrickb1912/ipl-complete-dataset-20082020
- https://en.wikipedia.org/wiki/Indian_Premier_League
- https://realpython.com/python-data-cleaning-numpy-pandas/
- https://dev.mysql.com/doc/
- https://www.astera.com/type/blog/data-warehouse-architecture/
- https://docs.microsoft.com/en-us/analysis-services/multidimensional-tutorial/lesson-2-5-deploying-an-analysis-services-project?view=asallproducts-allversions

Vishal - Data Cleaning and Creating fact and dimension tables,Executed Data Mining Queries(3)

Initha - Installation and Data transfer to MS Access, Data Transfer to SQL Server and Executed Data Mining Queries(2)

Sandeep - Cube development and deployment,Dimensional Modelling, Reports and Graph Generation.

Final Documentation is divided among the three individuals..