IPL CRICKET MATCH DATA INTEGRATION

Objective:

The IPL Cricket Match Data Integration project aims to build a structured and query-ready data warehouse to support analytical and reporting needs using IPL match data. This project utilizes **SQL Server Management Studio (SSMS)** for database management and **SQL Server Integration Services (SSIS)** to perform the **ETL (Extract, Transform, Load)** process on raw cricket data extracted from CSV files. The main goal is to create a scalable and efficient data model to facilitate insightful analysis of player statistics, match outcomes, and team performances across seasons.

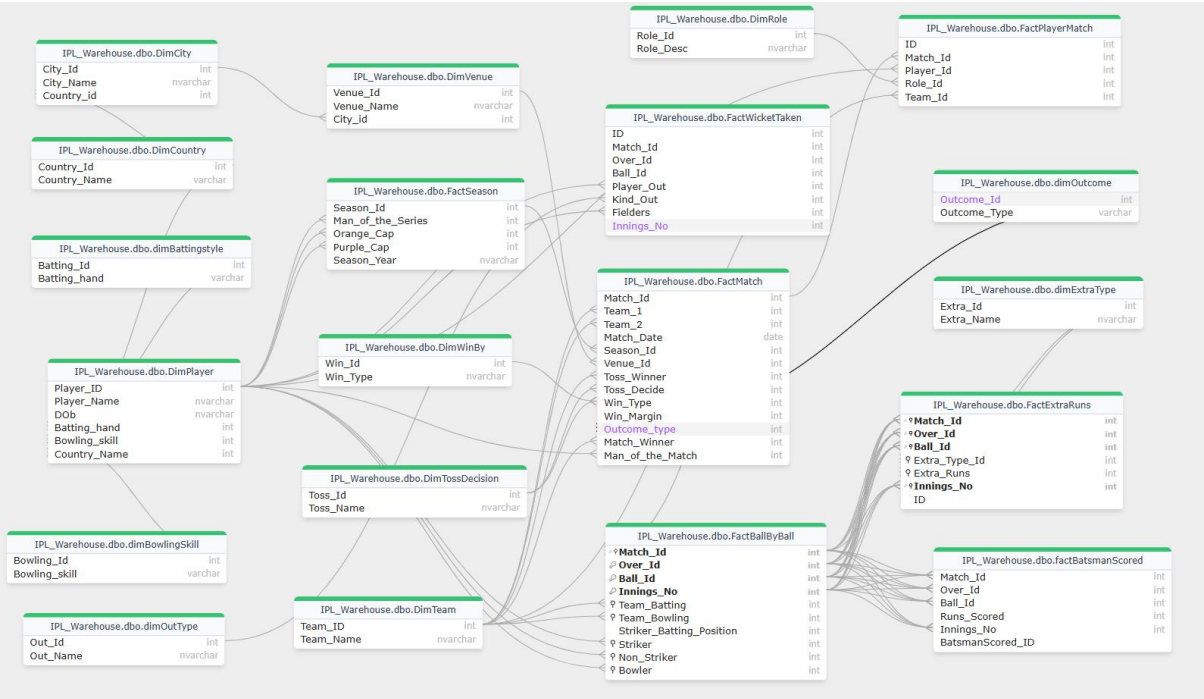
Data Modeling: Star Schema

The **Star Schema** is a widely used data modeling technique in data warehousing. In this project, the **Star Schema** was implemented to organize IPL cricket match data in a way that simplifies querying and enhances performance for analytical tasks.

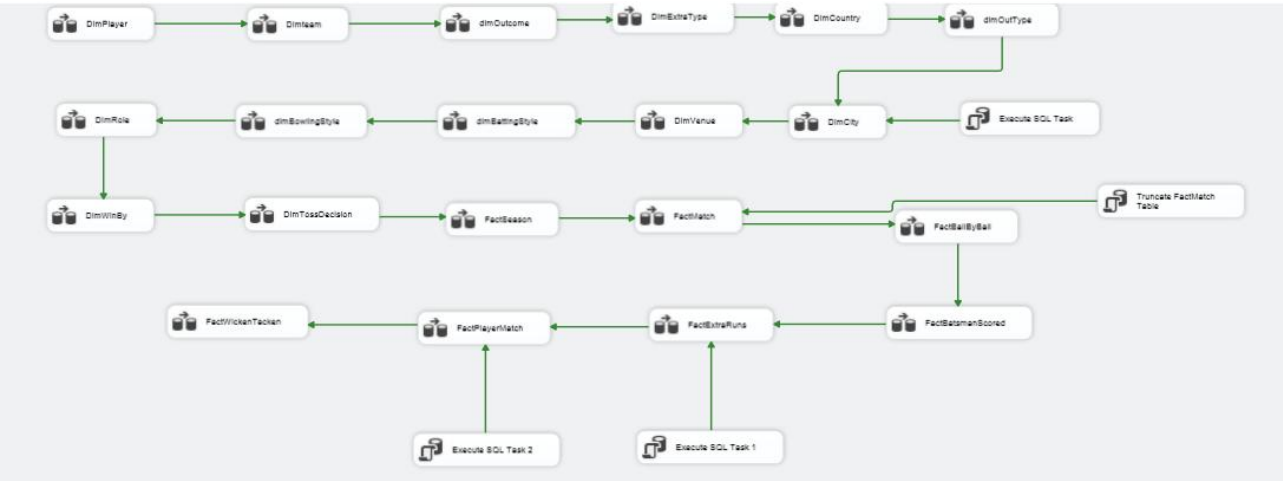
- **Structure of the Star Schema:**
  - The **central table** in a star schema is called the **Fact Table**. It stores measurable, quantitative data like runs scored, wickets taken, and match outcomes.
  - Surrounding the fact tables are **Dimension Tables**, which provide descriptive context (e.g., player details, team names, match venues).
- **Fact Tables in This Project:**
  1. FactMatch: Stores match-level details such as team names, match date, winner, toss result, etc.
  2. FactBallByBall: Captures each ball delivered in the match with bowler and batsman info.
  3. FactExtraRuns: Holds extra runs (like wide, no-ball).
  4. FactWicketTaken: Tracks information about dismissed players.
  5. FactPlayerMatch: Stores which player played which match and in what role.
  6. factBatsmanScored: Contains number of runs scored per ball by batsman.
- **Dimension Tables in This Project:**
  - DimPlayer, DimTeam, DimVenue, DimCity, DimCountry
  - DimBattingStyle, DimBowlingSkill, DimRole
  - DimOutcome, DimWinBy, DimTossDecision, DimOutType, DimExtraType

Each dimension table has a **primary key** that is used as a **foreign key** in one or more fact tables.

ER Diagram:



SSIS Data Flow Process: Table-wise ETL Description



This section provides a detailed explanation of how each table in the IPL Cricket Match Data Integration project was populated using SQL Server Integration Services (SSIS). It highlights the ETL process, tools used, and transformations applied for each dimension and fact table.

1. **Dimplayer:**

- Flat File Source**:**Reads raw player data from a CSV or TXT file containing fields like Player_ID, Player_Name, DOB, Batting_Hand, Bowling_Skill, and Country_Name.

- Derived Column Transformation**:**
  Used to:
  - Standardize or format text fields (e.g., converting DOB to a standard date format or trimming whitespace).
- Data Conversion Transformation**:**Ensures data types match the schema of the destination SQL Server table
- Conditional Split Transformation**:**Used to filter out invalid or incomplete records (e.g., missing Player ID or Name).
- OLE DB Destination**:**Loads the cleaned and transformed data into the DimPlayer table in the SQL Server database.

Output:

	Player_ID	Player_Name	DOb	Batting_hand	Bowling_skill	Country_Name
1	1	SC Ganguly	08-07-1972	1	1	1
2	2	BB McCullum	27-09-1981	2	1	4
3	3	RT Ponting	19-12-1974	2	1	5
4	4	DJ Hussey	15-07-1977	2	2	5

2. DimTeam:

Flat File Source**:**Loads the raw team data from a structured CSV or text file containing columns such as Team_ID and Team_Name.

Data Conversion Transformation**:**Ensures compatibility between the data types in the flat file and the SQL Server destination table schema.

OLE DB Destination**:**Transfers the cleansed data into the DimTeam table within the SQL Server database.

Output:

	Team_ID	Team_Name
1	1	Kolkata Knight Riders
2	2	Royal Challengers Bangalore
3	3	Chennai Super Kings
4	4	Kings XI Punjab
5	5	Rajasthan Royals
6	6	Delhi Daredevils
7	7	Mumbai Indians
8	8	Deccan Chargers
9	9	Kochi Tuskers Kerala
10	10	Pune Warriors
11	11	Sunrisers Hyderabad
12	12	Rising Pune Supergiants
13	13	Gujarat Lions

3. DimOutcome:

- Flat File Source:Reads raw outcome type data from a CSV or text file containing columns.
- Data Conversion:Ensures that the data types (especially for Outcome_Id) match the destination schema, typically converting strings or numeric types to SQL-compatible formats like int and varchar.
- Lookup Transformation:Checks whether an Outcome_Id already exists in the DimOutcome table to avoid inserting duplicates.
- Conditional Split:Directs data rows based on conditions.
- OLE DB Destination:Loads cleaned and validated outcome records into the DimOutcome table in SQL Server.

Output:

	Out_Id	Out_Name
1	1	Caught
2	2	Bowled
3	3	Run out
4	4	Lbw
5	5	Retired hurt
6	6	Stumped
7	7	Caught and bowled
8	8	Hit wicket
9	9	Obstructing the field

4. **DimExtraType:**

- Flat File Source:Imports raw data containing Extra_Id and Extra_Name from a structured file format (CSV or text).
- Data Conversion Transformation:Converts the source data types to match the schema of the SQL Server table.
  Example: Ensures Extra_Id is an int and Extra_Name is varchar(100).
- Derived Column Transformation:Used to standardize or enhance column values.
  Example: Trims extra whitespace, capitalizes values, or sets default names if missing.
- OLE DB Destination:Loads the cleaned and transformed data into the DimExtraType table in the database.

Output:

	Extra_Id	Extra_Name
1	1	Legbyes
2	2	Wides
3	3	Byes
4	4	Noballs
5	5	Penalty

5. **DimCountry:**

- Flat File Source:Reads the raw data file containing country information, including Country_ID and Country_Name.
- Data Conversion Transformation:Converts data types from the flat file into SSIS-compatible formats (e.g., converting string Country_ID into INT, or ensuring Country_Name is in NVARCHAR format).
- Derived Column Transformation:Used to create new columns or modify existing ones—for example, trimming whitespace, converting to uppercase/lowercase for consistency, or formatting names.
- OLE DB Destination:Loads the transformed and cleaned data into the DimCountry table in the SQL Server database.

Output:

	Country_Id	Country_Name
1	1	India
2	2	South africa
3	3	U.a.e
4	4	New zealand
5	5	Australia
6	6	Pakistan
7	7	Sri lanka
8	8	West indies
9	9	Zimbabwea
10	10	England
11	11	Bangladesh
12	12	Netherlands

6. **DimOutType:**

- Flat File Source:Imports raw data from a flat file containing Out_ID and Out_Name, representing different modes of player dismissal.
- Data Conversion Transformation:Converts data types from the flat file to match SQL Server schema—for example, ensuring Out_ID is in int format and Out_Name is nvarchar(100).
- Derived Column Transformation:Optionally used to standardize or clean Out_Name values (e.g., trimming spaces, converting to proper case) or to add a custom column if needed during transformation.
- OLE DB Destination:Loads the transformed and cleaned data into the DimOutType table in SQL Server.

Output:

	Out_Id	Out_Name
1	1	Caught
2	2	Bowled
3	3	Run out
4	4	Lbw
5	5	Retired hurt
6	6	Stumped
7	7	Caught and bowled
8	8	Hit wicket
9	9	Obstructing the field

7. **DimCity:**

- Flat File Source: Reads city data from a specified flat file.
- Derived Column: [You can describe here what derived columns you created, e.g., "Creates a new column combining city name and state for a unique identifier."]
- Data Conversion: [Specify any data type conversions performed, e.g., "Converts the city name column to Unicode string data type."]
- Sort: [Explain the sorting applied, e.g., "Sorts the data by city name before loading."]
- OLE DB Destination: Loads the transformed data into the `dimCity` table in the destination database.

Output:

	City_Id	City_Name	Country_id
1	1	Bangalore	1
2	2	Chandigarh	1
3	3	Delhi	1

8. **DimVenue:**

- Derived Column: Used to create new columns or modify existing ones based on expressions.
- Data Conversion: Employed to ensure that the data types of the incoming columns match the data types of the destination table columns.
- Sort: Utilized to order the data based on specific columns, which can be beneficial for performance or meeting specific requirements of the destination.
- OLE DB Destination: The final transformed data is loaded into the `DimVenue` table using an OLE DB Destination.

Output:

	Venue_Id	Venue_Name	City_id
1	1	M Chinnaswamy Stadium	1
2	2	Punjab Cricket Association Stadium	2
3	3	Feroz Shah Kotla	3

9. DimBattingStyle:

- Flat File Source: The Flat File Source component is used to read data from plain text files, such as CSV (Comma Separated Values) or TXT files.
  - Data Conversion: The Data Conversion transformation is used to change the data type of one or more columns in your data flow.
  - Lookup: The Lookup transformation allows you to retrieve additional columns from a related table
  - Destination: A Destination component is used to write data from your data flow to a data store.

Output:

	Batting_Id	Batting_hand
1	1	Left-hand bat
2	2	Right-hand bat

10. DimBowlingskill:

- Flat File Source:The process begins by importing raw bowling skill data from a CSV file using the Flat File Source component.
- Derived Column (Data Cleaning & Standardization):A derived column transformation is used to clean and format the Bowling_Skill values.
- Sort Transformation:Data is sorted on Bowling_Skil
- Data Conversion:Data types are converted to match the destination table schema.
- Lookup: If the process includes mapping Bowling_Skill to an existing dimension
- OLE DB Destination:The final cleaned, transformed, and validated records are loaded into the DimBowlingSkill table in the SQL Server database.

Output:

	Bowling_Id	Bowling_skill
1	1	Right-arm medium
2	2	Right-arm offbreak
3	3	Right-arm fast-medium
4	4	Legbreak googly
5	5	Right-arm medium-fast

11. DimRole:

- Flat File Source:The process begins by importing role-related data from a CSV file using the Flat File Source component.
- Data Conversion: The incoming data types from the flat file are converted to match the target table schema.
- Derived Column: Common transformations include Removing extra spaces using TRIM().
- OLE DB Destination: The transformed and validated role data is finally inserted into the DimRole table in the SQL Server database.

Output:

	Role_Id	Role_Desc
1	1	Captain
2	2	Keeper
3	3	Player
4	4	CaptainKeeper

12. DimWinType:

- Flat File Source:Data is read from a CSV file containing raw Win_Type values.
- Data Conversion: Converts the Win_Type column to a uniform data type (DT_WSTR) to allow string manipulation in the transformation phase.
- Derived Column: Transformation logic ensures Removal of unnecessary spaces using TRIM() Example:" run " → "Run"
- OLE DB Destination:The transformed data is loaded into the DimWinType table in the SQL Server database.

Output:

	Win_Id	Win_Type
1	1	Runs
2	2	Wickets
3	3	No result
4	4	Tie

13. DimTossDecision:

- Flat File Source:Data is imported from a CSV file that contains toss decision values.
- Data Conversion: A Data Conversion transformation is applied to convert the string data to a consistent DT_WSTR type.
- Derived Column: A Derived Column transformation is used to clean and standardize the Toss_Name field. The expression handles: Null or empty strings.
- OLE DB Destination: The cleaned and standardized toss decision values are inserted into the DimTossDecision table in the SQL Server database.

Output:

	Toss_Id	Toss_Name
1	1	Field
2	2	Bat

14. FactSeason:

- Flat File Source: Season data is sourced from a flat file (CSV), which contains season-related attributes such as Match_ID, Season_Year, etc.
- Data Conversion: This step ensures that incoming string dates or IDs are converted to appropriate SSIScompatible data types such as DT_I4 for integers and DT_WSTR for strings.
- Sort: The data is sorted based on Match_ID to ensure compatibility.
- Lookup: A Lookup transformation is used to match Match_ID or other foreign keys from the source data with existing records in relevant dimension tables (such as FactMatch or DimSeason, if applicable).
- Destination: The final cleaned and enriched dataset is loaded into the FactSeason table in the SQL Server database.

Output:

	Season_Id	Man_of_the_Series	Orange_Cap	Purple_Cap	Season_Year
1	1	32	100	102	2008
2	2	53	18	61	2009
3	3	133	133	131	2010

15. FactMatch:

- Flat file source: Source data is extracted from a structured CSV file containing match-level details.
- Data Conversion: Match_Id converted to integer Match_Date converted to DT_DBDATE or DT_DBTIMESTAMP.
- Derived Column: ISNULL(Win_Margin) || TRIM(Win_Margin) == "" ? NULL(DT_I4) : (DT_I4)Win_Margin, (ISNULL(Match_Winner) || TRIM(Match_Winner) == "") ? (DT_I4)-1 : (DT_I4)Match_Winner, (ISNULL(Man_of_the_Match) || TRIM(Man_of_the_Match) == "") ? (DT_I4)-1 : (DT_I4)Man_of_the_Match.
- Sort: A Sort transformation is used to organize data based on a key column typically Match_Id.
- Lookup: Multiple Lookup components are used to replace textual fields (like Team_Name, Venue_Name, etc.)
- Destination: The transformed, cleaned, and fully referenced data is loaded into the FactMatch table in the SQL Server database.

Output:

	Match_Id	Team_1	Team_2	Match_Date	Season_Id	Venue_Id	Toss_Winner	Toss_Decide	Win_Type	Win_Margin	Outcome_type	Match_Winner	Man_of_the_Match
1	335987	2	1	2008-04-18 00:00:00	1	1	2	1	1	140	1	1	2
2	335988	4	3	2008-04-19 00:00:00	1	2	3	2	1	33	1	3	19
3	335989	6	5	2008-04-19 00:00:00	1	3	5	2	2	9	1	6	90

16. FactBallByBall:

- Flat File Source: Data is extracted from a CSV file containing ball-by-ball match records.
- Data Conversion: A Data Conversion transformation converts string-based fields and numerics into appropriate SSIS-compatible data types (e.g., DT_I4 for integers, DT_WSTR for text).
- Sort: Records are sorted by primary keys or keys used in Lookup operations, such as Match_Id, Player_Id, or Over_Id.

- Conditional Split: it split data into to destination based on InningId 1 and inning Id 2
- Destination: The fully transformed and validated data is loaded into the FactBallByBall SQL Server table.

Output:

	Match_Id	Over_Id	Ball_Id	Innings_No	Team_Batting	Team_Bowling	Striker_Batting_Position	Striker	Non_Striker	Bowler
1	335987	1	1	1	1	2	1	1	2	14
2	335987	1	1	2	2	1	1	6	7	106
3	335987	1	2	1	1	2	2	2	1	14

17. FactBastmanScored:

- Flat File Source: The raw data is ingested from a CSV file containing columns like Match_Id, Player_Id, and Runs_Scored
- Data Conversion: Data Conversion transformation is applied to convert source data types to compatible formats (DT_I4, DT_WSTR, etc.)
- Sort: A Sort transformation is used to organize the data based on Match_Id or other keys, ensuring that the lookup operations that follow are optimized and efficient.
- Conditional Split: it split data into to destination based on InningId 1 and inning Id 2
- Destination: The fully transformed and validated data is loaded into the FactBatsmanScored table in the SQL Server data warehouse.

Output:

	Match_Id	Over_Id	Ball_Id	Runs_Scored	Innings_No	BatsmanScored_ID
1	335987	1	1	0	1	1
2	335987	1	2	0	1	2
3	335987	1	4	0	1	3

18. FactExtraRuns:

- Flat File Source: The raw data is ingested from a CSV file containing columns like Match_Id, Player_Id, and Runs_Scored
- Data Conversion: Data Conversion transformation is applied to convert source data types to compatible formats (DT_I4, DT_WSTR, etc.)
- Derived Column: (ISNULL(Match_Id) || TRIM(Match_Id) == "") ? (DT_I4)-1 : (DT_I4)Match_Id
- Sort: A Sort transformation is used to organize the data based on Match_Id or other keys, ensuring that the lookup operations that follow are optimized and efficient.
- Conditional Split: it split data into to destination based on InningId 1 and inning Id 2
- Destination: The fully transformed and validated data is loaded into the FactExtraRuns table in the SQL Server data warehouse.

Output:

	Match_Id	Over_Id	Ball_Id	Extra_Type_Id	Extra_Runs	Innings_No	ID
1	335987	1	1	1	1	1	1
2	335987	9	5	2	1	2	2
3	335987	10	5	1	1	2	3

19. FactPlayerMatch:

- Flat File Source: The raw data is ingested from a CSV file containing columns like Match_Id, Player_Id, and Runs_Scored
- Data Conversion: Data Conversion transformation is applied to convert source data types to compatible formats (DT_I4, DT_WSTR, etc.)
- Sort: A Sort transformation is used to organize the data based on Match_Id or other keys, ensuring that the lookup operations that follow are optimized and efficient.
- Conditional Split: it split data into to destination based on InningId 1 and inning Id 2
- Destination: The fully transformed and validated data is loaded into the FactPlayerMatch table in the SQL Server data warehouse.

Output:

	ID	Match_Id	Player_Id	Role_Id	Team_Id
1	1	335987	1	1	1
2	2	335987	2	3	1
3	3	335987	3	3	1

20. FactWicketTaken:

- Flat File Source: The raw data is ingested from a CSV file containing columns like Match_Id, Player_Id, and Runs_Scored
- Data Conversion: Data Conversion transformation is applied to convert source data types to compatible formats (DT_I4, DT_WSTR, etc.)
- Sort: A Sort transformation is used to organize the data based on Match_Id or other keys, ensuring that the lookup operations that follow are optimized and efficient.
- Conditional Split: it split data into to destination based on Fielders Null Value and Not Null Value.
- Destination: The fully transformed and validated data is loaded into the FactWicketTaken table in the SQL Server data warehouse.

Output:

	ID	Match_Id	Over_Id	Ball_Id	Player_Out	Kind_Out	Fielders	Innings_No
1	1	335987	2	1	6	2	NULL	2
2	2	335987	3	2	8	2	NULL	2
3	3	335987	5	5	9	1	83	2
4	4	335987	6	2	1	1	9	1
5	5	335987	6	2	7	1	3	2
6	6	335987	8	5	11	1	83	2