Sri Lanka Institute of Information Technology

Assignment 1

IT3021 - Data Warehousing and Business Intelligence

IT22561466

Hettiarachchi D.S.W

Data set selection

I have chosen a dataset that includes information about Indian Premier League (IPL) games. This includes information on every ball played in the IPL. The dataset's link is provided below.

https://www.kaggle.com/datasets/patrickb1912/ipl-complete-dataset20082020?select=IPL+Matches+2008-2020.csv

From this data set, data from 2013 to 2017 was chosen, and additional tables were made taking the relationships into account. To examine player-wise performance, the dataset is converted.

Preparation of data sources

All the data was initially in "csv" format. They were transformed into various data sources, including a text file, two "csv" files, and a database backup file. The tables that were divided into several sources are as follows:

Database(.bak)

1.1. Player
1.2. Venue
1.3. OutType
1.4. ExtraType
1.5. BallingStyle
1.6. BattingStyle
1.7. BallByBall
1.8. BallData

2. Comma Seperated Values(.csv)
   2.1. Country
   2.2. Team

3. Text(.txt)
   3.1. VenueAddress

Data Source Type	Source Name	Column Name	Data Type	Description
Database File (.bak)	dbo. BallByBall	BallDataID	int	Includes facts of the IPL matches ball by ball.
		BallDataID	int	
		TeamBattingID	int	
		TeamBowlingID	int	
		StrikerID	int	
		NonStrikerID	int	
		RunsScored	int	
		ExtraTypeID	int	
		ExtraRuns	int	

	OutTypeID	int	
	OutPlayerID	int	
	IsBowlerWicket	int	
	BowlerID	int	
	FielderID	int	
	MatchDate	datetme	
	VenueID	int	
dbo.BallData	BallDataID	int	

	MatchNo	int	Contains data about the balls in every match as a hierarchy. Ex: Third ball of second over of the first innings, of the fifth match.
	InningsNo	int	
	OverNo	int	
	BallNo	int	
dbo.Venue	VenueID	int	Contains details of grounds where

		VenueName	nvarchar(255)	matches are played.
	dbo.ExtraType	ExtraTypeID	int	Contains data types extras.
		ExtraType	nvarchar(255)	
	dbo.OutType	OutTypeID	int	Contains data about types of wickets.
		OutType	nvarchar(255)	
	dbo.Player	PlayerID	int	Contains details of players.
		PlayeName	nvarchar(255)	
		PlayerNameInitials	nvarchar(255)	
		CountryID	int	
		BattingStyle	int	
		BowlingStyleID	int	
	dbo.BattingStyle	BattingStyleID	int	Contains details of batting styles.
		BattingStyle	nvarchar(255)	
	dbo.BowlingStyle	BowlingStyleID	int	Contains details of batting styles.
		BowlingStyle	nvarchar(255)	
CSV File	Country.csv	CountryID	int	Contains details of countries of players.
		CountryName	varchar(50)	

	Team.csv	TeamID	int	Contains details of teams of the tournament.
		TeanmName	varchar(50)	
Text file	VenueAddress.txt	VenueAddressID	int	Contains details addresses of the venues (grounds)
		CityName	varchar(50)	
		CountryName	varchar(50)	

Solution architecture

Data Sources

The sources that were used to obtain the data are represented by the data sources. CSV, text, and other sources are the three categories. Bak, which stands for database files, text files, and files separated by commas, respectively.

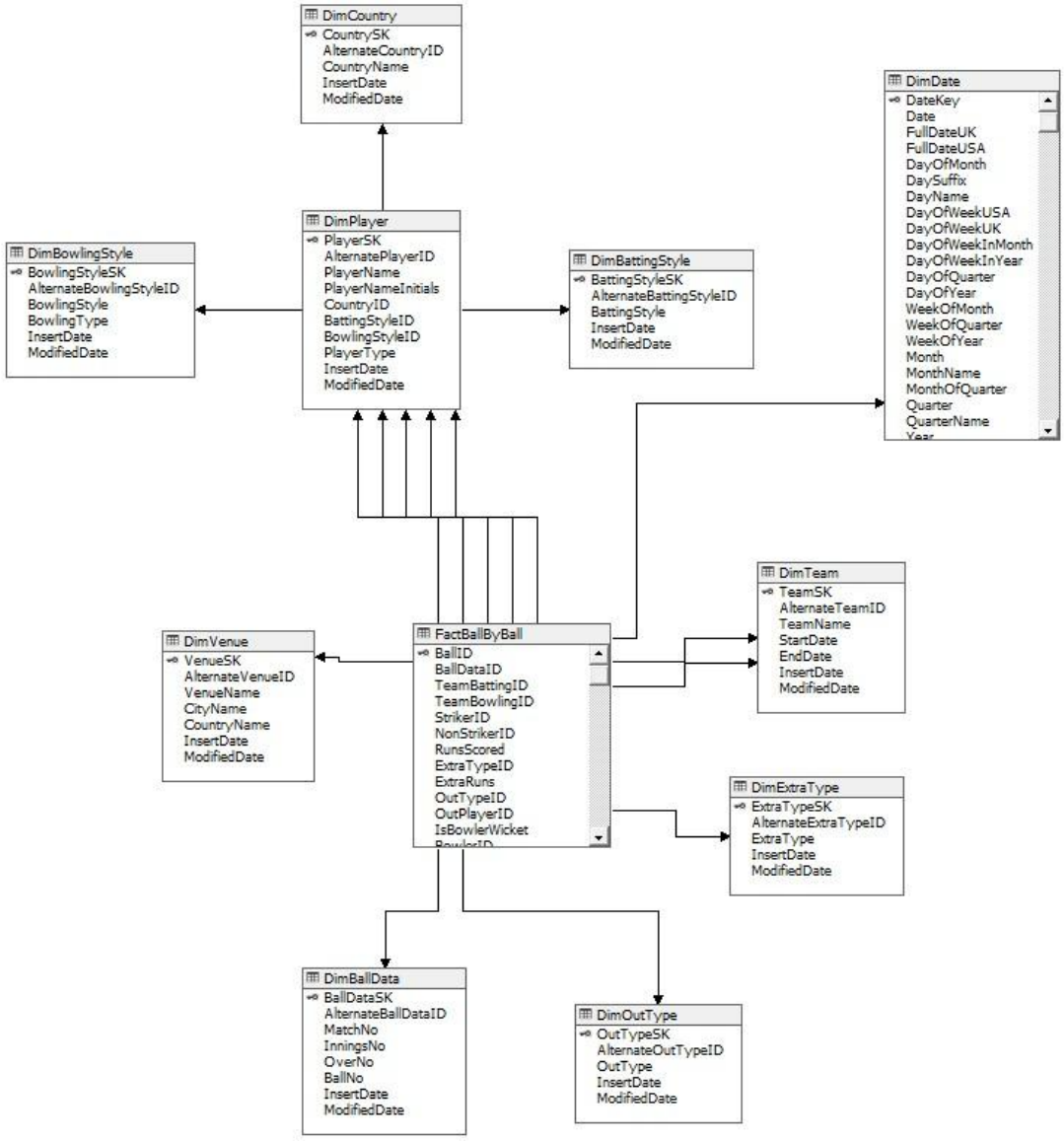
Staging Area

This level represent creating staging level tables using the data which were obtained dy different data sources.

Data Warehouse

Here, data in the staging area are transformed and loaded into the data warehouse as facts and dimensions which is then used for Business Intelligence purposes.

Data warehouse design & development



Above diagram shows how the dimension tables and fact table was combined.

Following were considered when developing the data warehouse dimensional model;

- Snowflake schema type was used.

- **Dimensions**
  - Hierarchical dimensions
    1. Venue – Country name – City name – Venue name

2. BowlingStyle – Bowling Type – Bowling Style
3. BallData – Match number – Innings number – Over number – Ball number
4. Date

- **Slowly changing dimensions**
  1. Team – Team name

- **Fact Table**
  - BallByBall

    This table consists of 12 foreign key columns which are connected to the dimensions of the model.

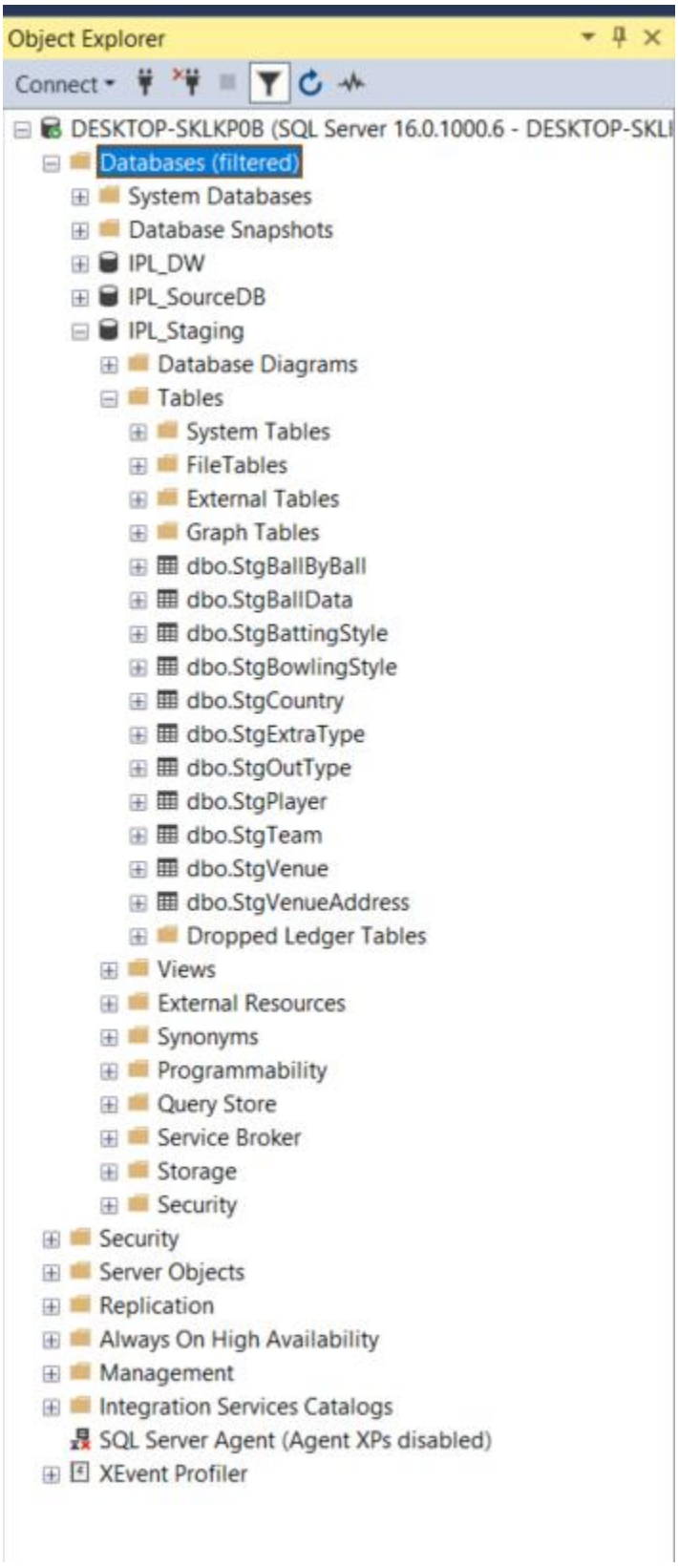
- Assumptions
  - Since the names of the teams are changed by owners when needed, Team was considered as a slowly changing dimension to tack the details of the team names.

ETL development

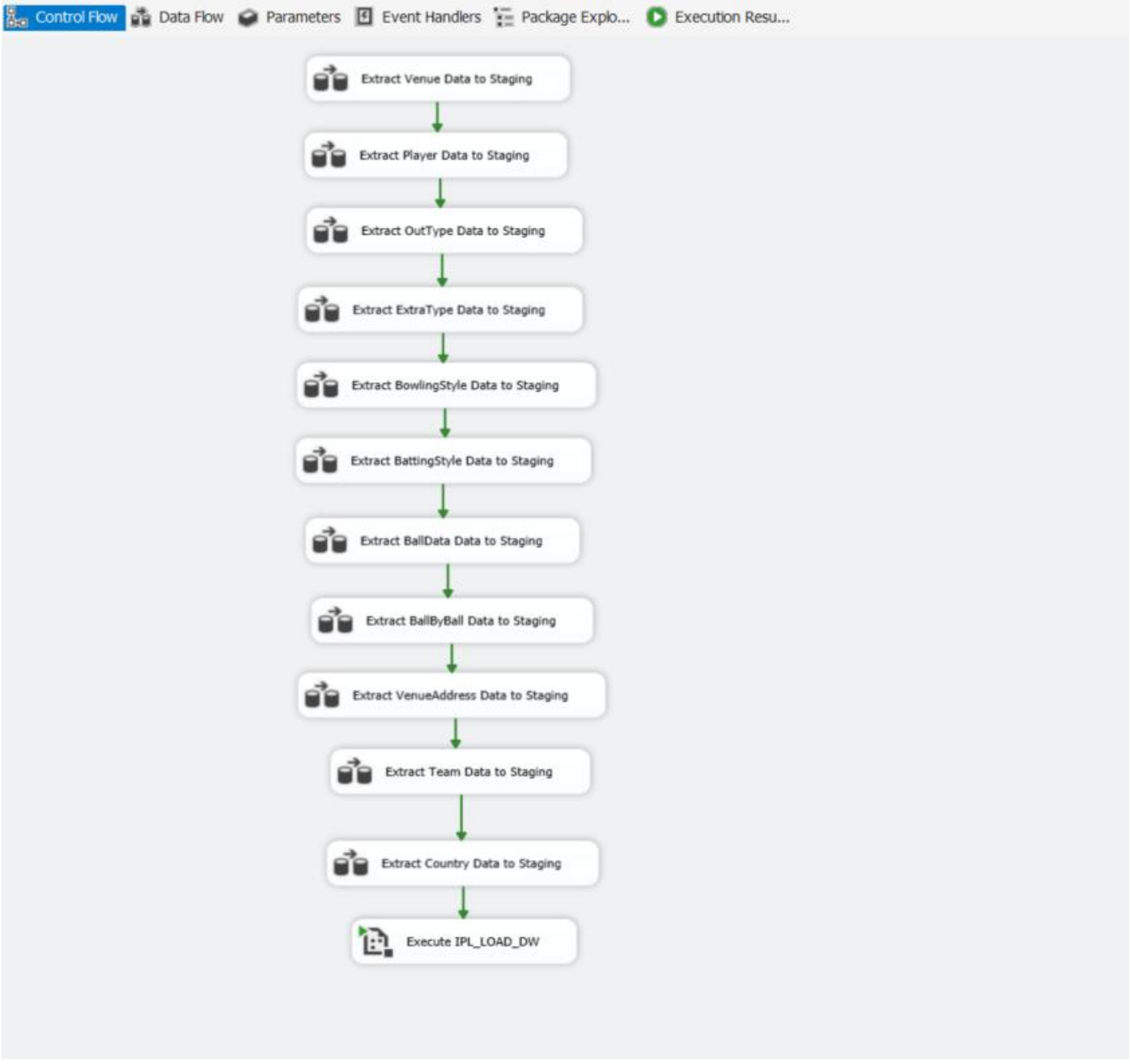
Extract

First, all the data which mentioned in the Preparation of Data Sources step were imported to the staging database (IPL_Staging) by using relevant connections and the sources. Below image shows the tables of the staging database;

SSMS staging database (IPL_Staging)

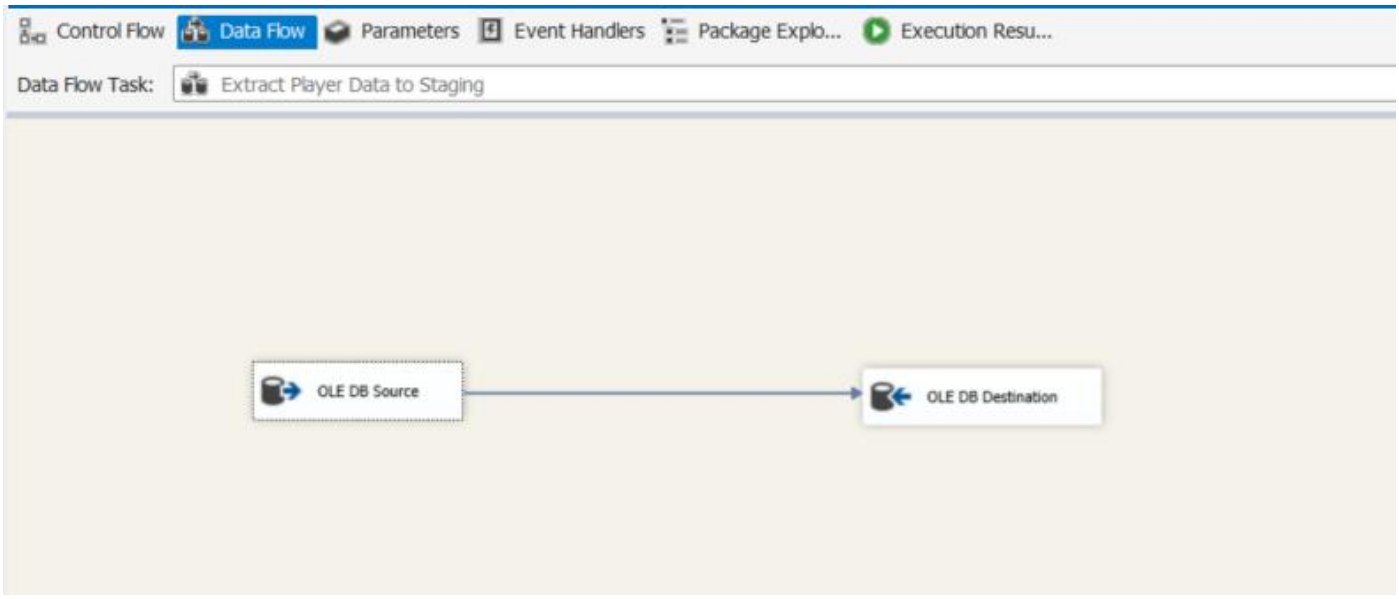


Control flow of extraction

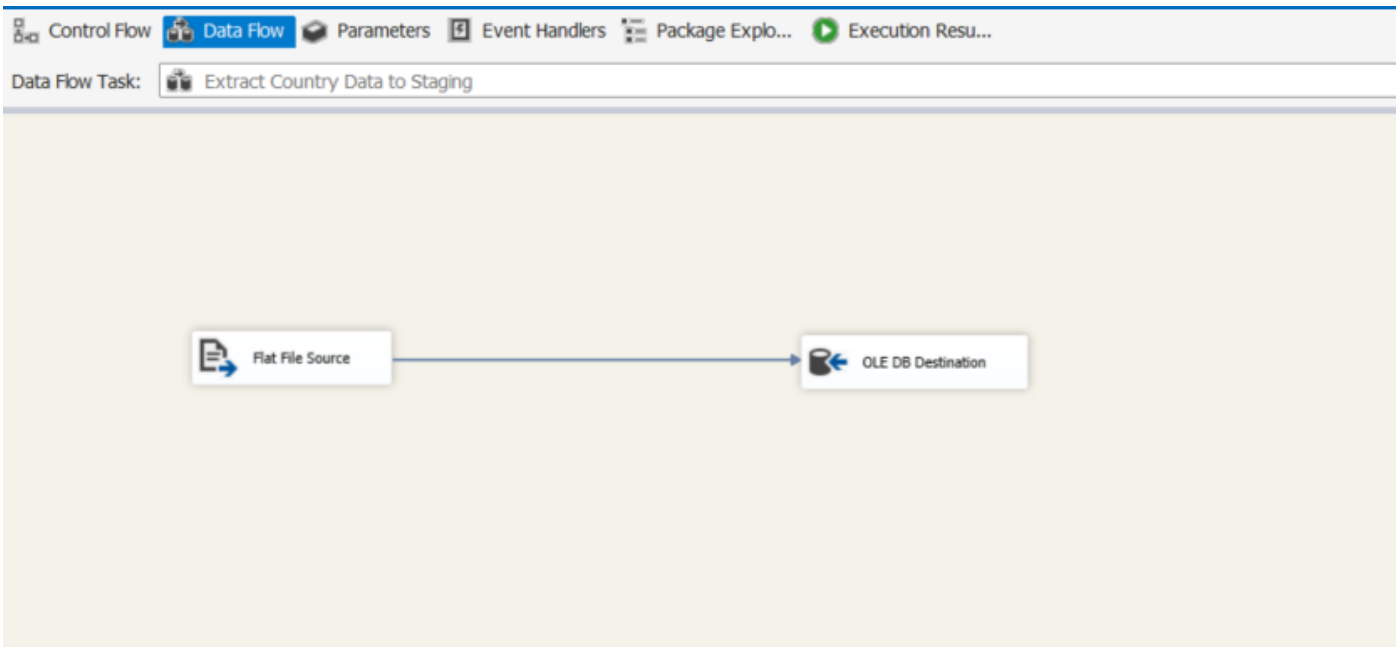


Screeshots of some data flows are given below;

Player data flow

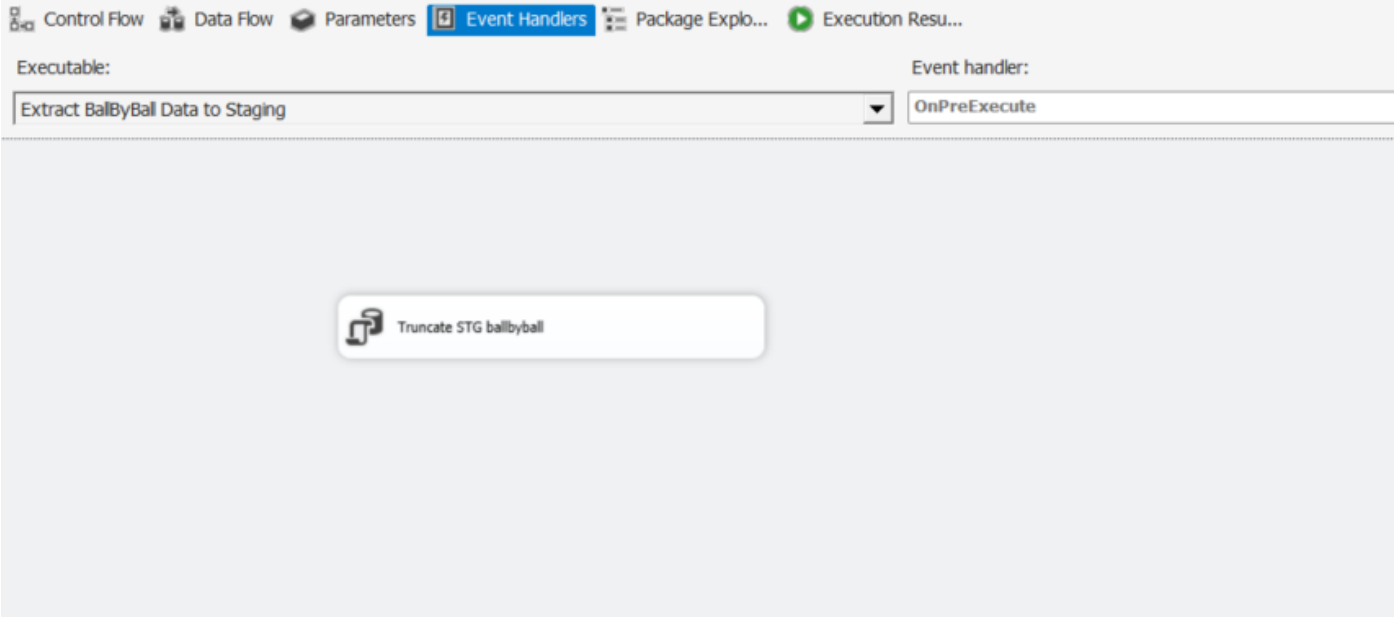


Country data flow

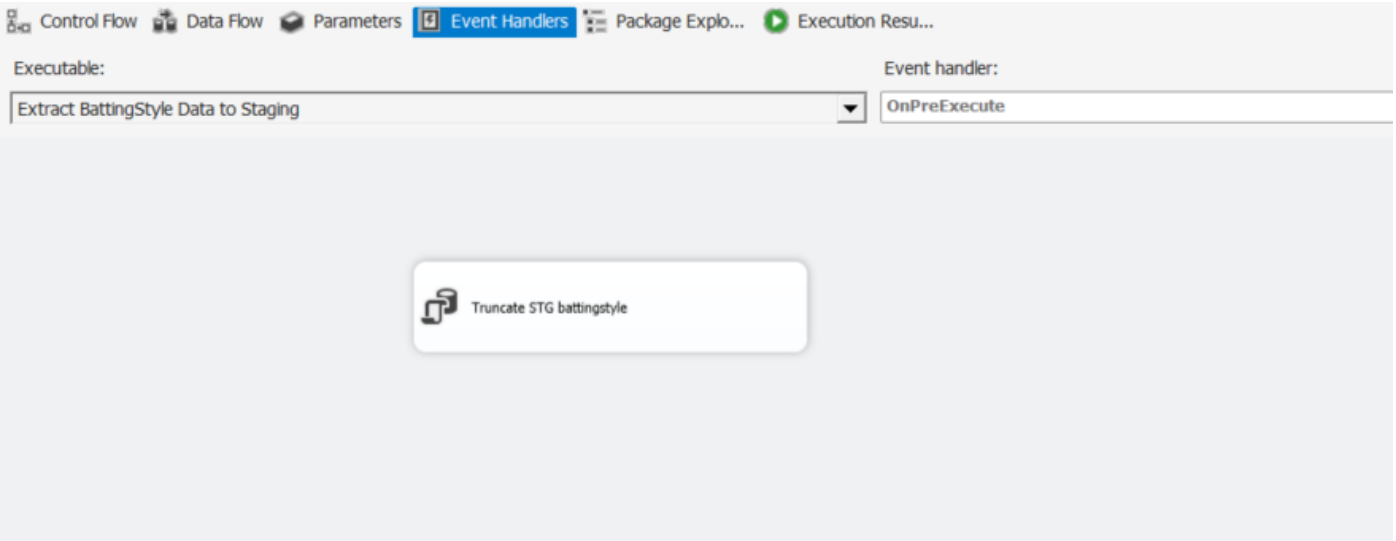


Screeshots of some event handlers are given below;

Truncate BallByBall Staging



Truncate BattingStyle Staging



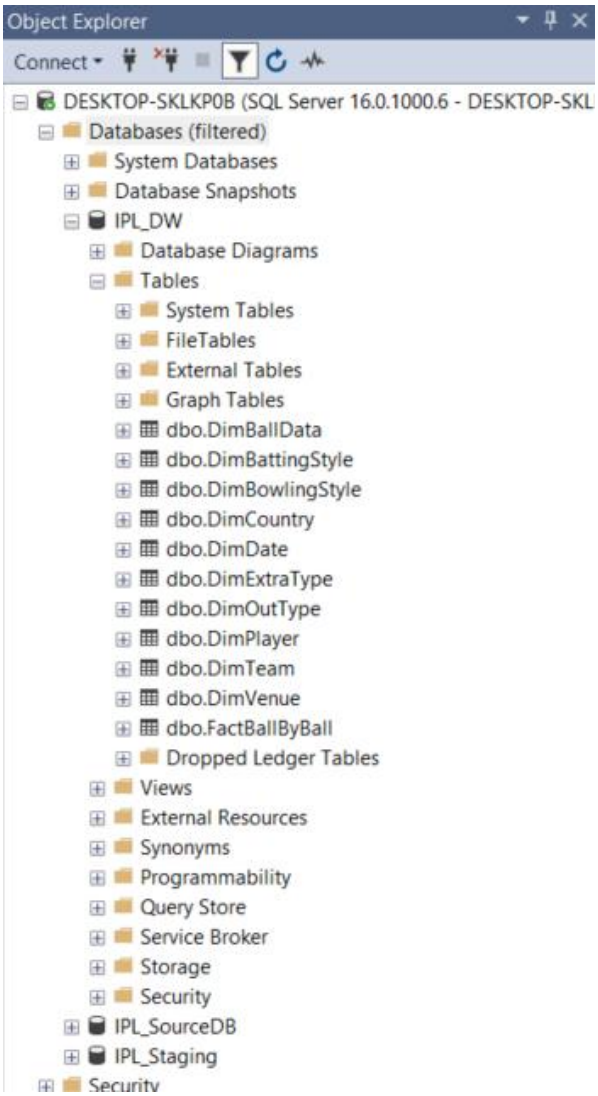
Transform and load

Next the data in the staging area were transformed and loaded in to the data warehouse (IPL_DW). First the dimension tables and the fact table was created and the data were loaded to the in the relevant order.

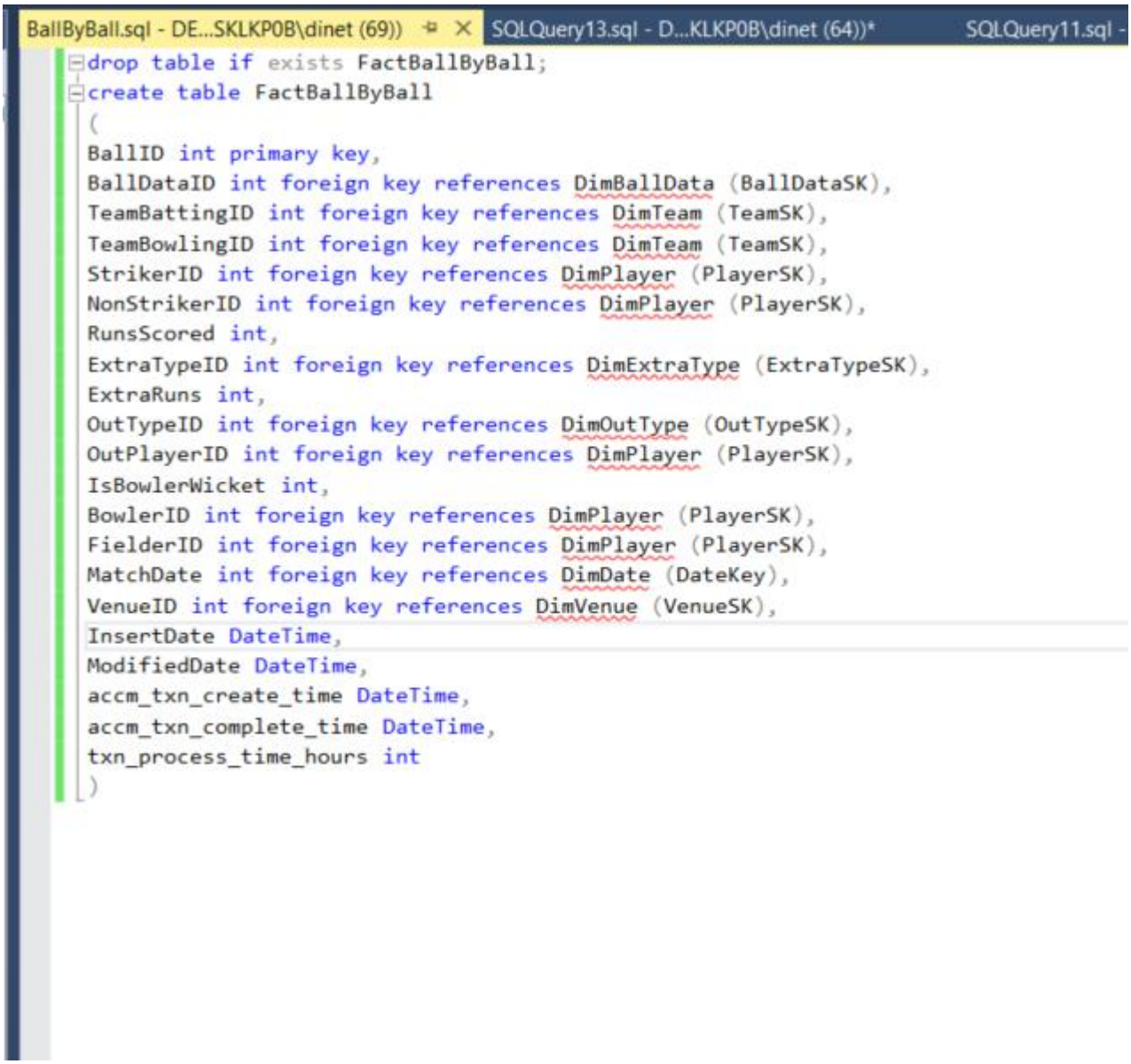
Tasks such as merge join, lookup, derived columns, and sort were used in transforming and loading data to data warehouse.

Below image shows the tables of the staging database;

SSMS data warehouse (IPL_DW)

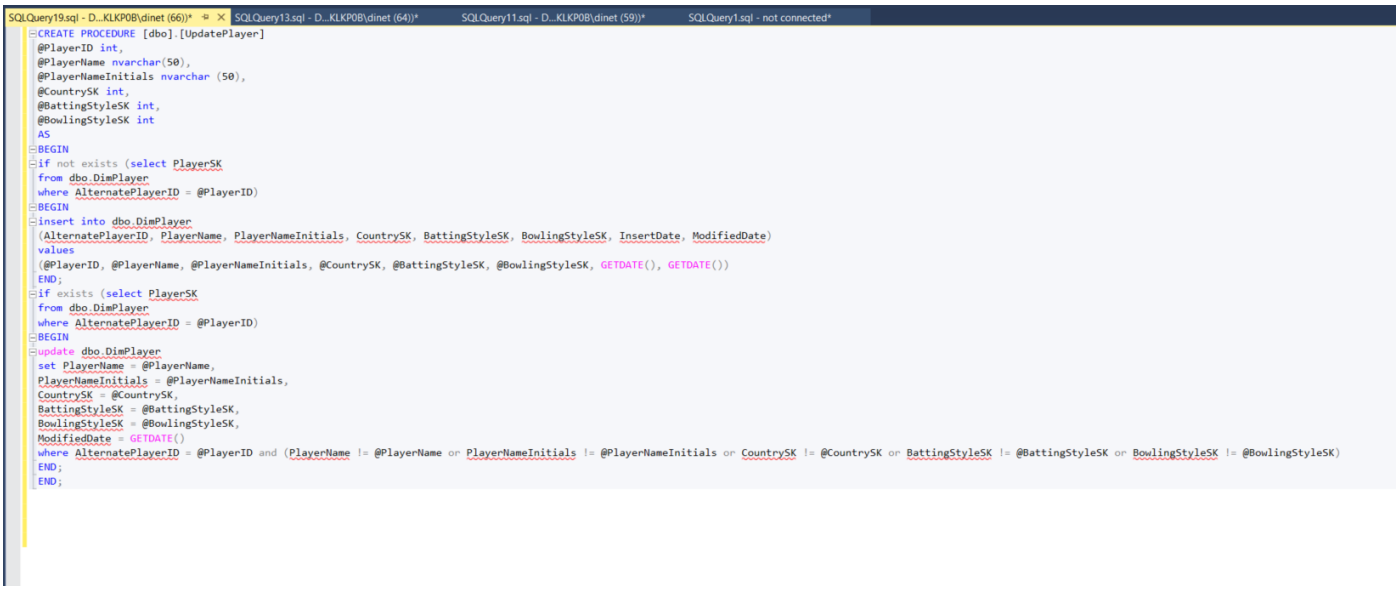


Sql Query to create fact table;



Screenshots of some sql procedures are given below;

Procedure for dimPlayer

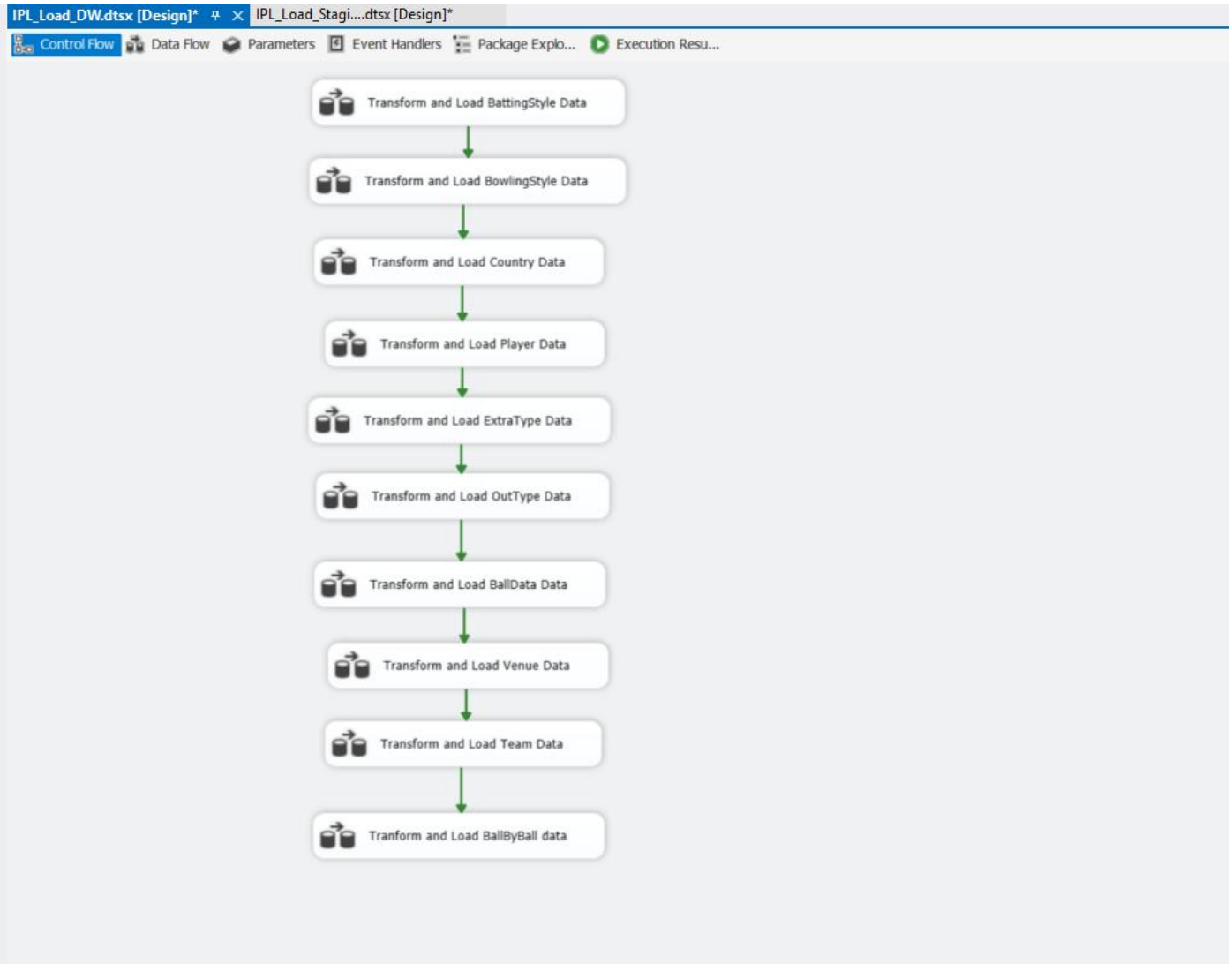


```
CREATE PROCEDURE [dbo].[UpdatePlayer]
@PlayerID int,
@PlayerName nvarchar(50),
@PlayerNameInitials nvarchar (50),
@CountrySK int,
@BattingStyleSK int,
@BowlingStyleSK int
AS
BEGIN
if not exists (select PlayerSK
from dbo.DimPlayer
where AlternatePlayerID = @PlayerID)
BEGIN
insert into dbo.DimPlayer
(AlternatePlayerID, PlayerName, PlayerNameInitials, CountrySK, BattingStyleSK, BowlingStyleSK, InsertDate, ModifiedDate)
values
(@PlayerID, @PlayerName, @PlayerNameInitials, @CountrySK, @BattingStyleSK, @BowlingStyleSK, GETDATE(), GETDATE())
END;
if exists (select PlayerSK
from dbo.DimPlayer
where AlternatePlayerID = @PlayerID)
BEGIN
update dbo.DimPlayer
set PlayerName = @PlayerName,
PlayerNameInitials = @PlayerNameInitials,
CountrySK = @CountrySK,
BattingStyleSK = @BattingStyleSK,
BowlingStyleSK = @BowlingStyleSK,
ModifiedDate = GETDATE()
where AlternatePlayerID = @PlayerID and (PlayerName != @PlayerName or PlayerNameInitials != @PlayerNameInitials or CountrySK != @CountrySK or BattingStyleSK != @BattingStyleSK or BowlingStyleSK != @BowlingStyleSK)
END;
END;
```

Procedure for dimVenue

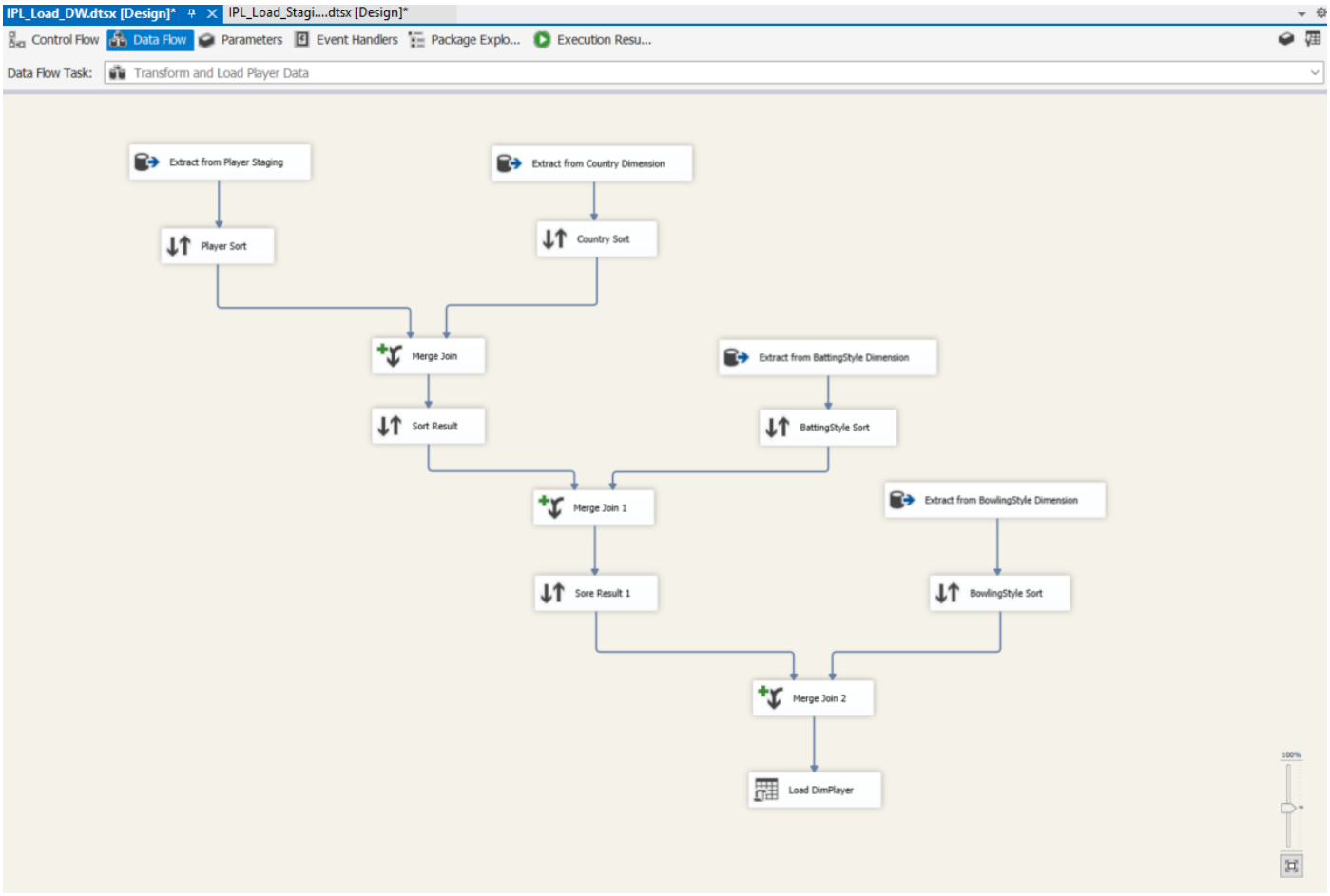
```
CREATE PROCEDURE dbo.UpdateVenue
@VenueID int,
@VenueName nvarchar(100),
@CityName nvarchar(50),
@CountryName nvarchar(50)
AS
BEGIN
if not exists (select VenueSK
from dbo.DimVenue
where AlternateVenueID = @VenueID)
BEGIN
insert into dbo.DimVenue
(AlternateVenueID, VenueName, CityName,  CountryName, InsertDate, ModifiedDate)
values
(@VenueID, @VenueName, @CityName, @CountryName, GETDATE(), GETDATE())
END;
if exists (select VenueSK
from dbo.DimVenue
where AlternateVenueID = @VenueID)
BEGIN
update dbo.DimVenue
set VenueName = @VenueName,
CityName = @CityName,
CountryName = @CountryName,
ModifiedDate = GETDATE()
where AlternateVenueID = @VenueID and (VenueName != @VenueName or CityName != @CityName or CountryName != @CountryName)
END;
END;
```

Control flow extraction

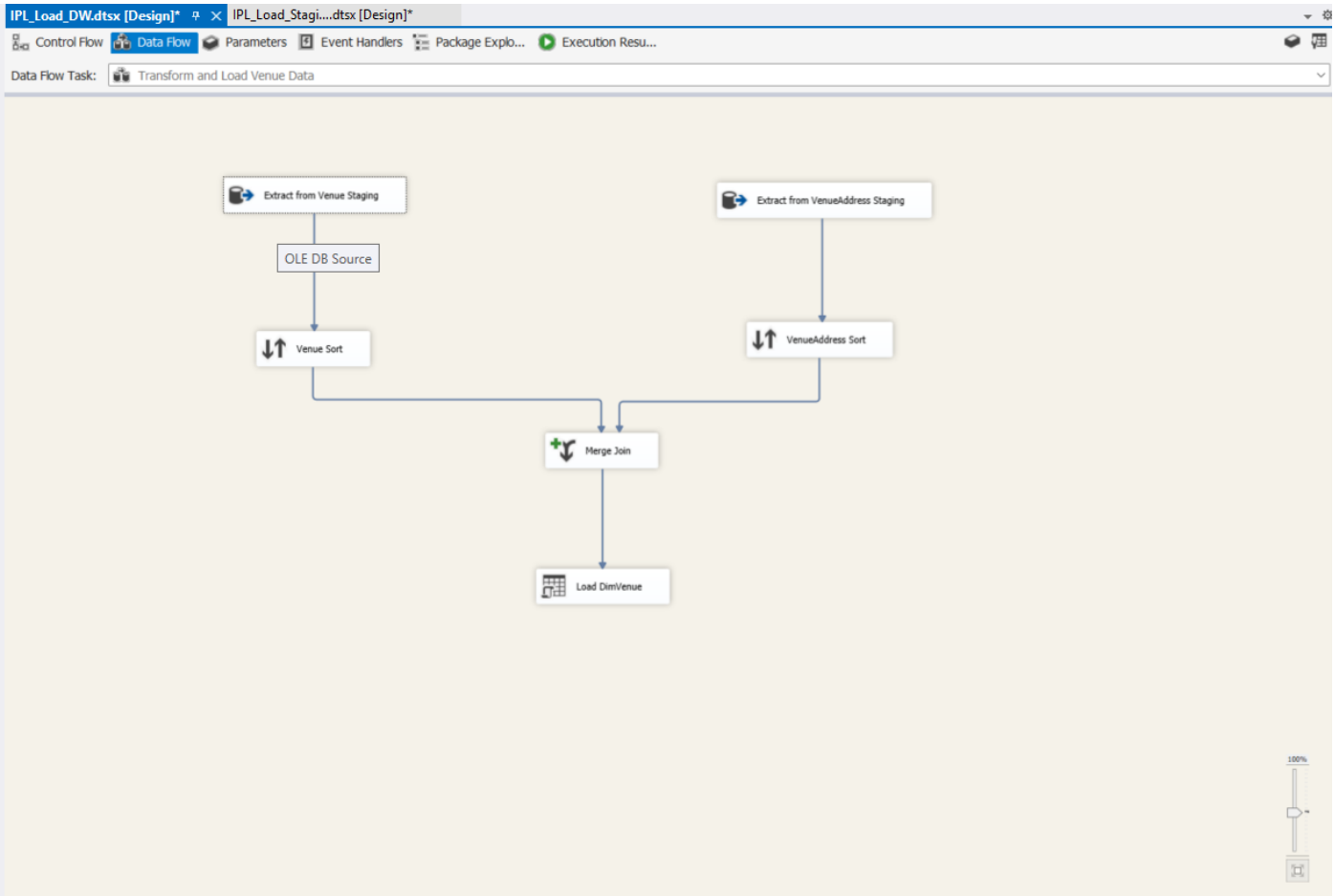


Screeshots of some data flows are given below;

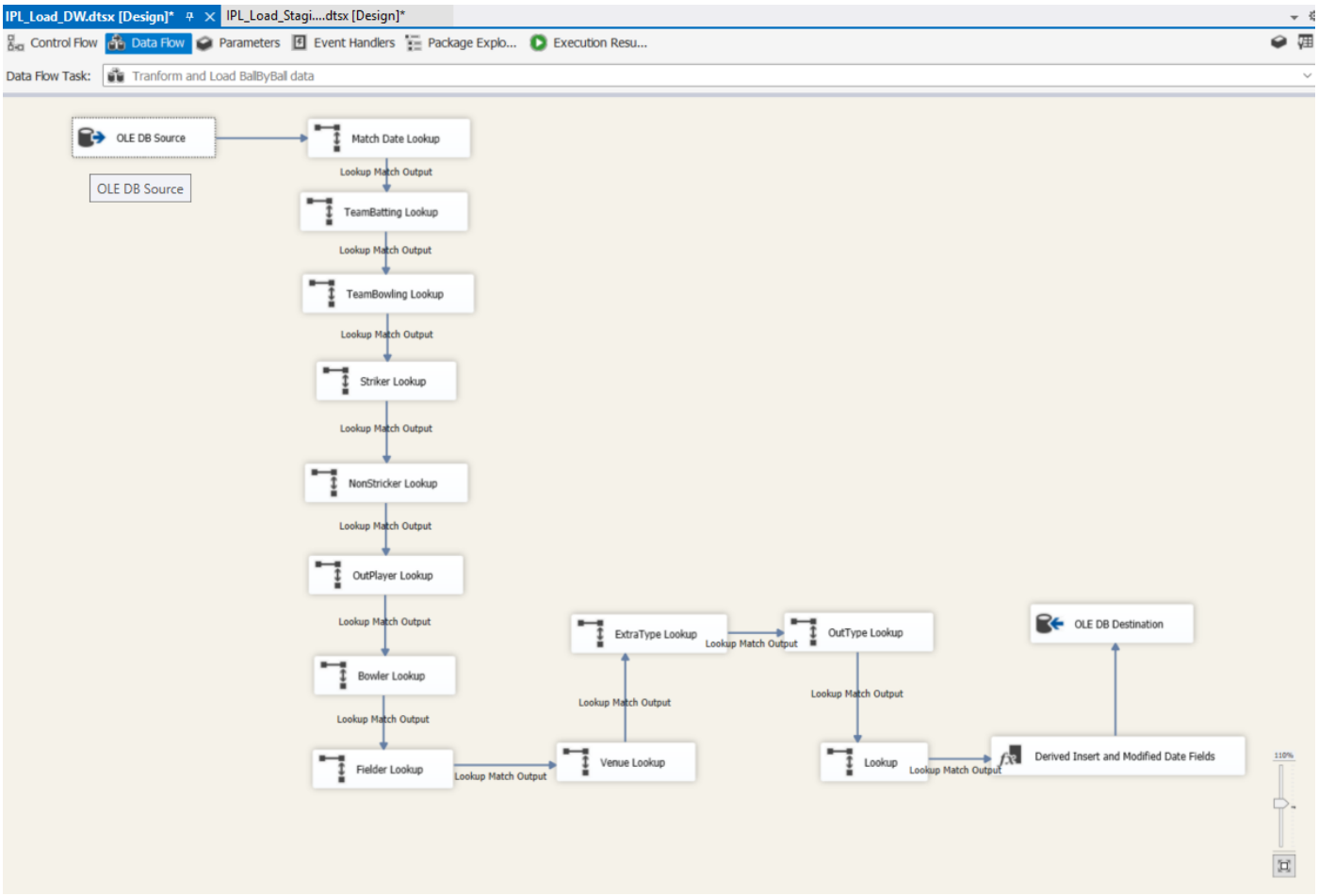
DimPlayer transform and load



Dim Venue transform and load



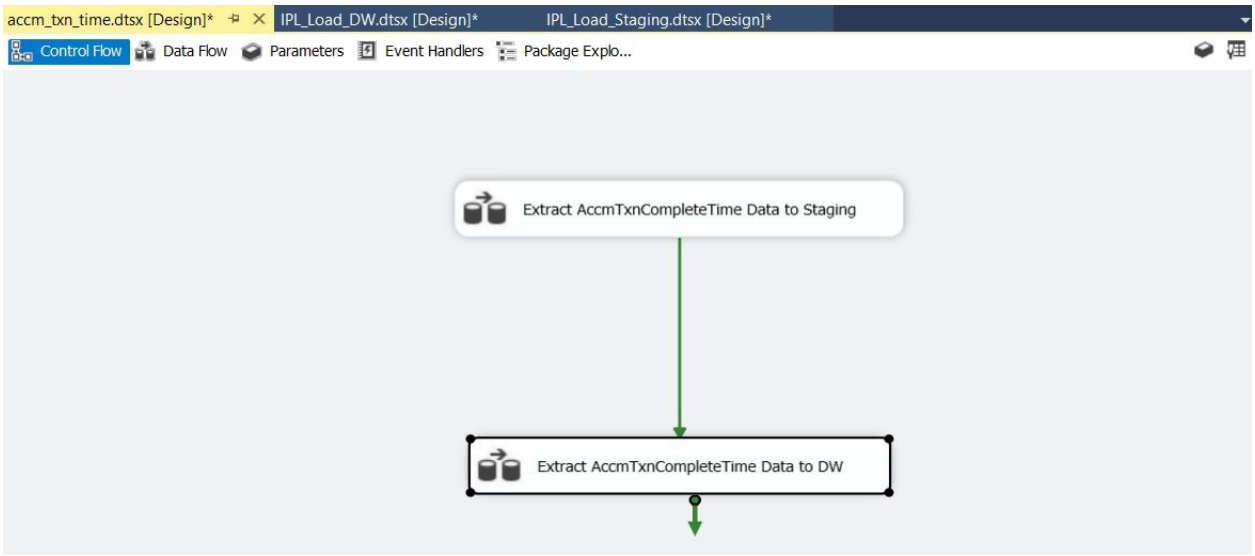
Fact BallByBall transform and load



ETL development – Accumulating fact tables

An external csv data source was used for this step and relevant coulombs were created in fact table. Data was transformed and loaded to these fields using separate ETL process.

Control flow extraction



Transform and load to Fact BallByBall