SQL Queries for transformations and analysis

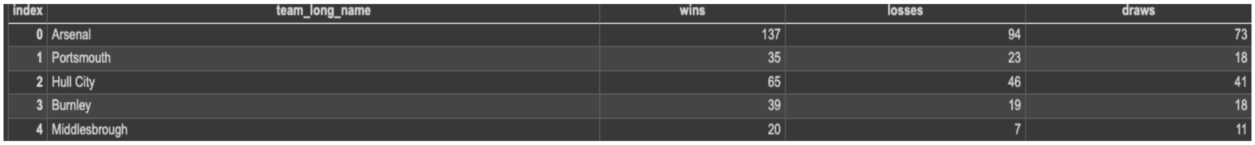
Team:
Anitha Balachandran
Aradhya Alva Rathnakar
Bhavan Kumar Basavaraju
Mahamaya Panda
Shashi Kumar Kadari Mallikarjuna

```
SELECT *
FROM [dbo].[Fact_match] as fm
INNER JOIN [dbo].[matchPlayers] as mp on mp.match_id=fm.match_id


SELECT * FROM [dbo].[Fact_match_ext] as fme
JOIN [dbo].[Fact_match] as fm ON fm.match_id=fme.match_id
```

# The above two join statements are used to help us establish relationships between tables and then visualize utilizing it #

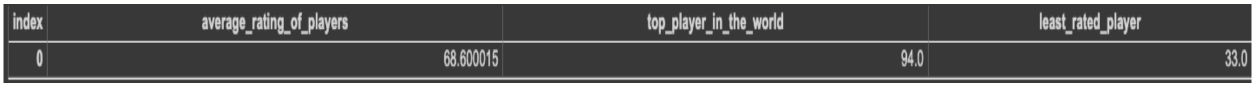
```
SELECT [dt].team_long_name,
       COUNT(CASE WHEN match.home_team_goal > match.away_team_goal THEN 1 ELSE NULL
END) AS wins,
       COUNT(CASE WHEN match.home_team_goal < match.away_team_goal THEN 1 ELSE NULL
END) AS losses,
       COUNT(CASE WHEN match.home_team_goal = match.away_team_goal THEN 1 ELSE NULL
END) AS draws
FROM [dbo].[Fact_match] as MATCH
JOIN [dbo].[DimTeam] as dt ON match.home_team_id = dt.team_id OR match.away_team_id =
dt.team_id
JOIN [dbo].[DimLeague] as dl on dl.league_id=match.league_id
WHERE dl.league_name = 'England Premier League'
GROUP BY dt.team_long_name;
```



index	team_long_name	wins	losses	draws
0	Arsenal	137	94	73
1	Portsmouth	35	23	18
2	Hull City	65	46	41
3	Burnley	39	19	18
4	Middlesbrough	20	7	11

The above code executes to provide us with the report data as per the league and the teams within the league along with the wins, losses and draws over the 9 season period.

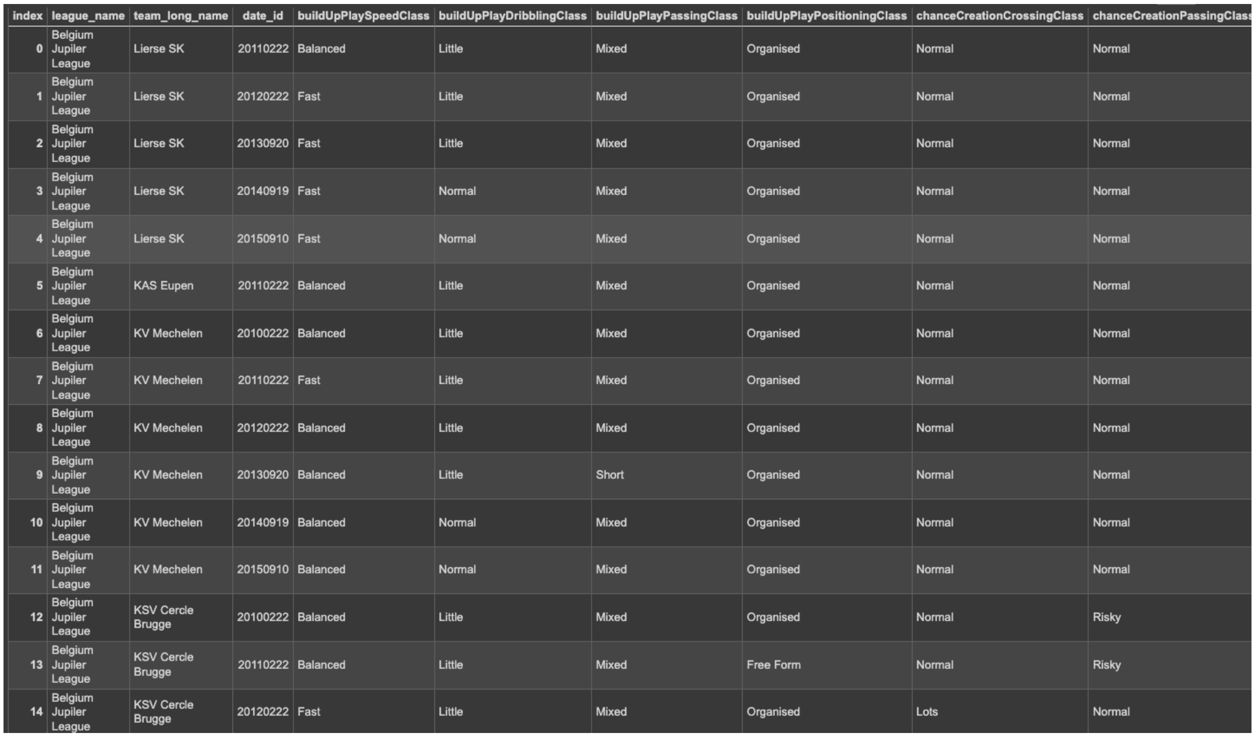
```
SELECT AVG(fpa.overall_rating) AS average_rating_of_players, MAX(fpa.overall_rating)
as top_player_in_the_world, MIN(fpa.overall_rating) as least_rated_player
FROM [dbo].[DimTeam] as dt,[dbo].[Fact_player_attributes] as fpa
WHERE fpa.overall_rating>0
```



index	average_rating_of_players	top_player_in_the_world	least_rated_player
0	68.600015	94.0	33.0

The above piece of code provides us with the average ratings of the players in the professional football world,along with the top player as per rating and least player with the same in the report.

```
select e.league_name,a.team_long_name,b.date_id
,b.buildUpPlaySpeedClass,b.buildUpPlayDribblingClass,b.buildUpPlayPassingClass,b.build
UpPlayPositioningClass,b.chanceCreationCrossingClass,b.chanceCreationPassingClass,b.ch
anceCreationPositioningClass,b.chanceCreationShootingClass,b.defencePressureClass,b.de
fenceAggressionClass,b.defenceDefenderLineClass,b.defenceTeamWidthClass from DimTeam a
join
(select distinct team_id,date_id
,buildUpPlaySpeedClass,buildUpPlayDribblingClass,buildUpPlayPassingClass,buildUpPlayPo
sitioningClass,chanceCreationCrossingClass,chanceCreationPassingClass,chanceCreationPo
sitioningClass,chanceCreationShootingClass,defencePressureClass,defenceAggressionClass
,defenceDefenderLineClass,defenceTeamWidthClass from Fact_team_attributes) b
on a.team_id=b.team_id join (select distinct d.league_name,c.away_team_id from
Fact_match c join DimLeague d on c.league_id=d.league_id) as e
on a.team_id=e.away_team_id
ORDER BY e.league_name;
```

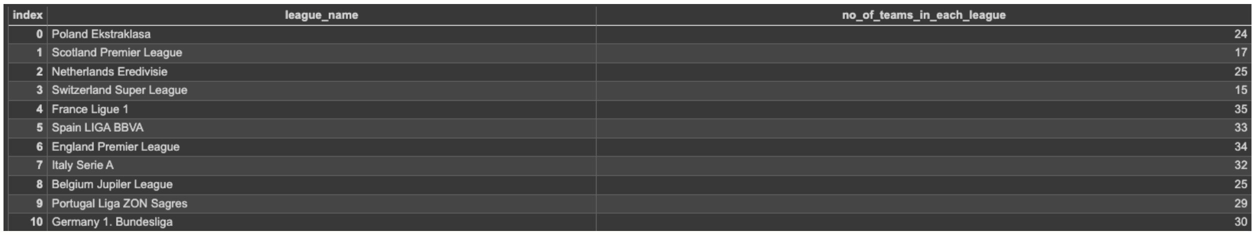


index	league_name	team_long_name	date_id	buildUpPlaySpeedClass	buildUpPlayDribblingClass	buildUpPlayPassingClass	buildUpPlayPositioningClass	chanceCreationCrossingClass	chanceCreationPassingClass
0	Belgium Jupiler League	Lierse SK	20110222	Balanced	Little	Mixed	Organised	Normal	Normal
1	Belgium Jupiler League	Lierse SK	20120222	Fast	Little	Mixed	Organised	Normal	Normal
2	Belgium Jupiler League	Lierse SK	20130920	Fast	Little	Mixed	Organised	Normal	Normal
3	Belgium Jupiler League	Lierse SK	20140919	Fast	Normal	Mixed	Organised	Normal	Normal
4	Belgium Jupiler League	Lierse SK	20150910	Fast	Normal	Mixed	Organised	Normal	Normal
5	Belgium Jupiler League	KAS Eupen	20110222	Balanced	Little	Mixed	Organised	Normal	Normal
6	Belgium Jupiler League	KV Mechelen	20100222	Balanced	Little	Mixed	Organised	Normal	Normal
7	Belgium Jupiler League	KV Mechelen	20110222	Fast	Little	Mixed	Organised	Normal	Normal
8	Belgium Jupiler League	KV Mechelen	20120222	Balanced	Little	Mixed	Organised	Normal	Normal
9	Belgium Jupiler League	KV Mechelen	20130920	Balanced	Little	Short	Organised	Normal	Normal
10	Belgium Jupiler League	KV Mechelen	20140919	Balanced	Normal	Mixed	Organised	Normal	Normal
11	Belgium Jupiler League	KV Mechelen	20150910	Balanced	Normal	Mixed	Organised	Normal	Normal
12	Belgium Jupiler League	KSV Cercle Brugge	20100222	Balanced	Little	Mixed	Organised	Normal	Risky
13	Belgium Jupiler League	KSV Cercle Brugge	20110222	Balanced	Little	Mixed	Free Form	Normal	Risky
14	Belgium Jupiler League	KSV Cercle Brugge	20120222	Fast	Little	Mixed	Organised	Lots	Normal

The above code provides us with the reporting data on what types of changes are made to a team in a league with respect to tactics. This is also combined with the date on which the change is performed. This helps in detailing the probable tactics that can be implemented by a team.

The below code details about how many teams are present in each of the leagues and the report changes with each year alterations performed.

```
select  d.league_name,count(distinct(c.away_team_id)) from Fact_match c join DimLeague
d on c.league_id=d.league_id
group by d.league_name
```



index	league_name	no_of_teams_in_each_league
0	Poland Ekstraklasa	24
1	Scotland Premier League	17
2	Netherlands Eredivisie	25
3	Switzerland Super League	15
4	France Ligue 1	35
5	Spain LIGA BBVA	33
6	England Premier League	34
7	Italy Serie A	32
8	Belgium Jupiler League	25
9	Portugal Liga ZON Sagres	29
10	Germany 1. Bundesliga	30

View to unpivot the player information columns in the match_fact table for analysis:

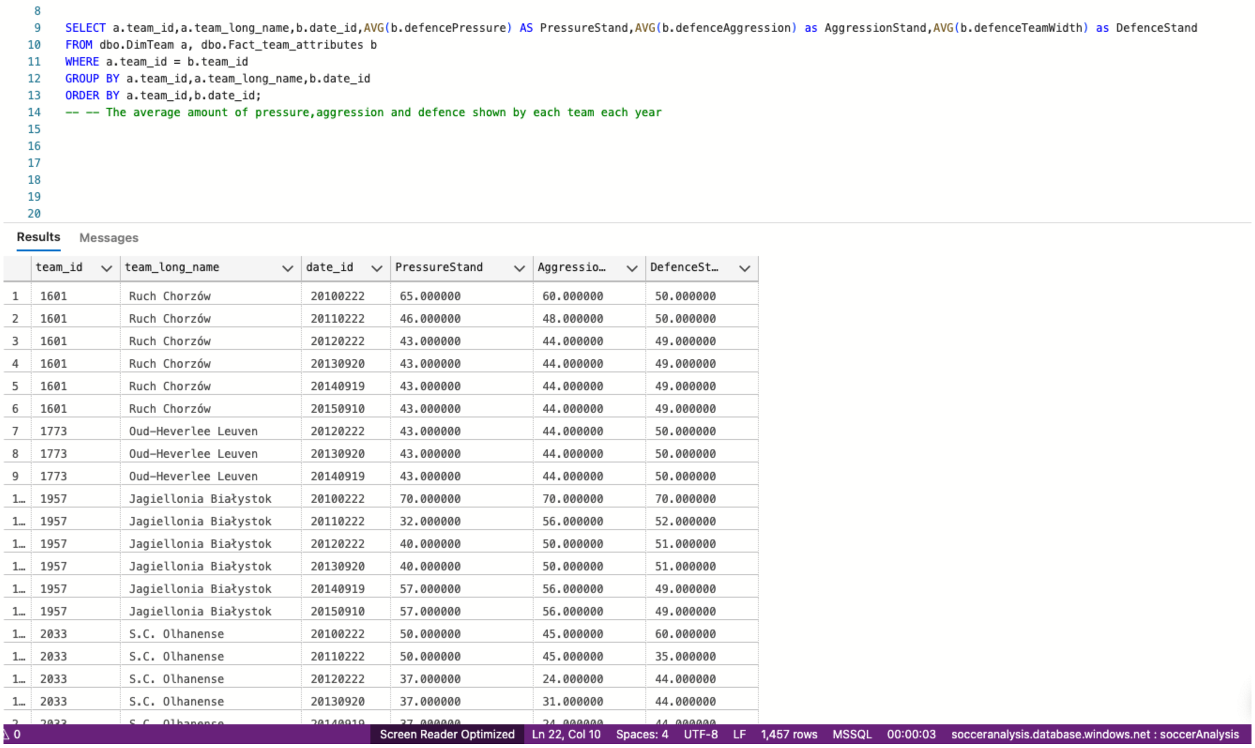
```
CREATE VIEW [dbo].[matchPlayers]
AS SELECT match_id,Player,PlayerID
FROM
   (SELECT match_id
          ,home_player_1
          ,home_player_2
          ,home_player_3
          ,home_player_4
          ,home_player_5
          ,home_player_6
          ,home_player_7
          ,home_player_8
          ,home_player_9
          ,home_player_10
          ,home_player_11
          ,away_player_1
          ,away_player_2
          ,away_player_3
          ,away_player_4
          ,away_player_5
          ,away_player_6
          ,away_player_7
          ,away_player_8
          ,away_player_9
          ,away_player_10
          ,away_player_11
        FROM fact_match) p
UNPIVOT
   (PlayerID FOR Player IN
      (    home_player_1
          ,home_player_2
          ,home_player_3
          ,home_player_4
          ,home_player_5
          ,home_player_6
          ,home_player_7
          ,home_player_8
          ,home_player_9
          ,home_player_10
          ,home_player_11
          ,away_player_1
          ,away_player_2
          ,away_player_3
          ,away_player_4
          ,away_player_5
          ,away_player_6
          ,away_player_7
          ,away_player_8
          ,away_player_9
          ,away_player_10
          ,away_player_11)
)AS unpvt;
```

Stored procedure to load the above view data into a table for faster retrieval:

```
CREATE PROC [dbo].[load_match_players] AS
    truncate table dbo.match_players_ext;
    insert into dbo.match_players_ext select * from matchPlayers
GO;
```

The average amount of pressure,aggression and defence shown by each team each year

```
SELECT a.team_id,a.team_long_name,b.date_id,AVG(b.defencePressure) AS
PressureStand,AVG(b.defenceAggression) as AggressionStand,AVG(b.defenceTeamWidth) as
DefenceStand
FROM dbo.DimTeam a, dbo.Fact_team_attributes b
WHERE a.team_id = b.team_id
GROUP BY a.team_id,a.team_long_name,b.date_id
ORDER BY a.team_id,b.date_id;
```

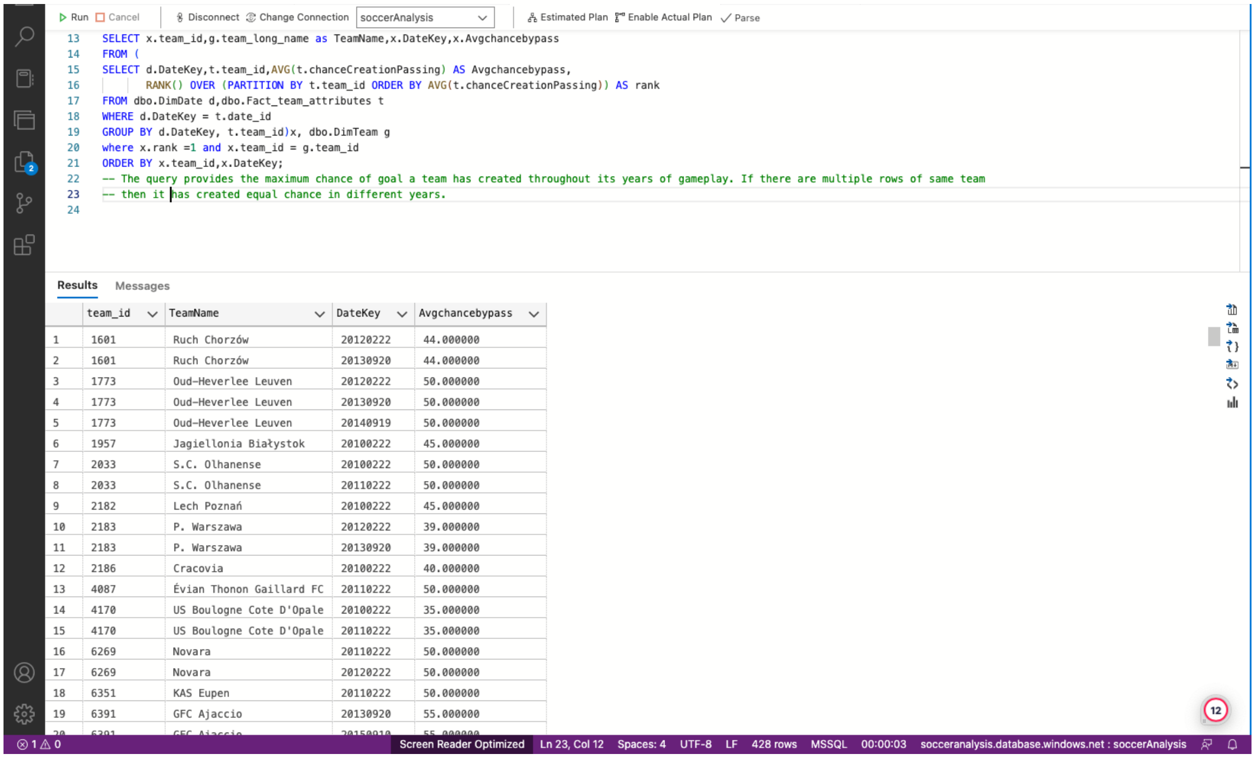


```sql
SELECT a.team_id,a.team_long_name,b.date_id,AVG(b.defencePressure) AS PressureStand,AVG(b.defenceAggression) as AggressionStand,AVG(b.defenceTeamWidth) as DefenceStand
FROM dbo.DimTeam a, dbo.Fact_team_attributes b
WHERE a.team_id = b.team_id
GROUP BY a.team_id,a.team_long_name,b.date_id
ORDER BY a.team_id,b.date_id;
-- -- The average amount of pressure,aggression and defence shown by each team each year
```

	team_id	team_long_name	date_id	PressureStand	Aggressio…	DefenceSt…
1	1601	Ruch Chorzów	20100222	65.000000	60.000000	50.000000
2	1601	Ruch Chorzów	20110222	46.000000	48.000000	50.000000
3	1601	Ruch Chorzów	20120222	43.000000	44.000000	49.000000
4	1601	Ruch Chorzów	20130920	43.000000	44.000000	49.000000
5	1601	Ruch Chorzów	20140919	43.000000	44.000000	49.000000
6	1601	Ruch Chorzów	20150910	43.000000	44.000000	49.000000
7	1773	Oud-Heverlee Leuven	20120222	43.000000	44.000000	50.000000
8	1773	Oud-Heverlee Leuven	20130920	43.000000	44.000000	50.000000
9	1773	Oud-Heverlee Leuven	20140919	43.000000	44.000000	50.000000
1…	1957	Jagiellonia Białystok	20100222	70.000000	70.000000	70.000000
1…	1957	Jagiellonia Białystok	20110222	32.000000	56.000000	52.000000
1…	1957	Jagiellonia Białystok	20120222	40.000000	50.000000	51.000000
1…	1957	Jagiellonia Białystok	20130920	40.000000	50.000000	51.000000
1…	1957	Jagiellonia Białystok	20140919	57.000000	56.000000	49.000000
1…	1957	Jagiellonia Białystok	20150910	57.000000	56.000000	49.000000
1…	2033	S.C. Olhanense	20100222	50.000000	45.000000	60.000000
1…	2033	S.C. Olhanense	20110222	50.000000	45.000000	35.000000
1…	2033	S.C. Olhanense	20120222	37.000000	24.000000	44.000000
1…	2033	S.C. Olhanense	20130920	37.000000	31.000000	44.000000
2…	2033	S.C. Olhanense	20140919	37.000000	24.000000	44.000000

The query provides the maximum chance of goal a team has created throughout its years of gameplay. If there are multiple rows of same team then it has created equal chance in different years.

```sql
SELECT x.team_id,g.team_long_name as TeamName,x.DateKey,x.Avgchancebypass
FROM (
SELECT d.DateKey,t.team_id,AVG(t.chanceCreationPassing) AS Avgchancebypass,
      RANK() OVER (PARTITION BY t.team_id ORDER BY AVG(t.chanceCreationPassing)) AS
rank
FROM dbo.DimDate d,dbo.Fact_team_attributes t
WHERE d.DateKey = t.date_id
GROUP BY d.DateKey, t.team_id)x, dbo.DimTeam g
where x.rank =1 and x.team_id = g.team_id
ORDER BY x.team_id,x.DateKey;
```



```sql
SELECT x.team_id,g.team_long_name as TeamName,x.DateKey,x.Avgchancebypass
FROM (
SELECT d.DateKey,t.team_id,AVG(t.chanceCreationPassing) AS Avgchancebypass,
        RANK() OVER (PARTITION BY t.team_id ORDER BY AVG(t.chanceCreationPassing)) AS rank
FROM dbo.DimDate d,dbo.Fact_team_attributes t
WHERE d.DateKey = t.date_id
GROUP BY d.DateKey, t.team_id)x, dbo.DimTeam g
where x.rank =1 and x.team_id = g.team_id
ORDER BY x.team_id,x.DateKey;
-- The query provides the maximum chance of goal a team has created throughout its years of gameplay. If there are multiple rows of same team
-- then it has created equal chance in different years.
```

	team_id	TeamName	DateKey	Avgchancebypass
1	1601	Ruch Chorzów	20120222	44.000000
2	1601	Ruch Chorzów	20130920	44.000000
3	1773	Oud-Heverlee Leuven	20120222	50.000000
4	1773	Oud-Heverlee Leuven	20130920	50.000000
5	1773	Oud-Heverlee Leuven	20140919	50.000000
6	1957	Jagiellonia Białystok	20100222	45.000000
7	2033	S.C. Olhanense	20100222	50.000000
8	2033	S.C. Olhanense	20110222	50.000000
9	2182	Lech Poznań	20100222	45.000000
10	2183	P. Warszawa	20120222	39.000000
11	2183	P. Warszawa	20130920	39.000000
12	2186	Cracovia	20100222	40.000000
13	4087	Évian Thonon Gaillard FC	20110222	50.000000
14	4170	US Boulogne Cote D'Opale	20100222	35.000000
15	4170	US Boulogne Cote D'Opale	20110222	35.000000
16	6269	Novara	20110222	50.000000
17	6269	Novara	20120222	50.000000
18	6351	KAS Eupen	20110222	50.000000
19	6391	GFC Ajaccio	20130920	55.000000
20	6391	GFC Ajaccio	20150910	55.000000

Screen Reader Optimized Ln 23, Col 12 Spaces: 4 UTF-8 LF 428 rows MSSQL 00:00:03 socceranalysis.database.windows.net : soccerAnalysis