Research Project on SQL

1.0 Defining Business Requirements

1.1 Business Objective

A new franchise team is formed and the management of the team has been assigned a project to determine the players that can be auctioned and a balanced team composition be formed.

1.2 Business Requirements

Criteria of selecting the players for the new franchise team in IPL and number of required players needed to build a balanced team composition consisting of 25-member squad.

Requirement #	Player's Skill	Player's Count	Selection Criteria
R1	Top order batsman includes the following – • Opening batsman playing in position 1 and 2 in the batting order • 1st down batsman playing in position 3 in the batting order	Recommend 10 players' list, where top 6 is the first priority in the auction	1. Create a list of 30 Batsman who are the leading run scorers in that said batting position. 2. Filter the list to have batting average of above 20. 3. Filter the list to have deviation of scores of below 28. 4. Filter the list to have strike rate of more than 104 in the first 8 overs. 5. Filter the list to have received at least 5 MOM awards. 6. Set Priority=1 for top 6 in the list ordered by leading run scored with at least 2 left handed batsman. Rest set priority=2.
R2	Middle order batsman includes the following – • Batsman batted in position 4 and 5 in the batting order	Recommend 8 players' list, where top 4 is the first priority in the auction	1. Create a list of 30 Batsman who are the leading run scorers in that said batting position. 2. Filter the list to have batting average of above 12. 3. Filter the list to have deviation of scores of below 23. 4. Filter the list to have strike rate of more than 102 from 7th over to 15th over. 5. Filter the list to have received at least 3 MOM awards. 6. Set Priority=1 for top 4 in the list ordered by leading run scored with at least 1 left handed batsman. Rest set priority=2
R3	Lower middle order batsman includes the following – • Batsman batted in position 6 and 7 in the batting order	Recommend 8 players' list, where top 4 is the first priority in the auction	1. Create a list of 30 Batsman who are the leading run scorers in that said batting position. 2. Filter the list to have batting average of above 16. 3. Filter the list to have deviation of scores of below 18. 4. Filter the list to have strike rate of more than 106 in the last 5 overs. 5. Filter the list to have received at least 1 MOM award. 6. Set Priority=1 for top 4 in the list ordered by leading run scored with at least 1 left handed batsman. Rest set priority=2
R4	Spinners includes the following: • Spinner bowler could be right arm leg break • Spinner bowler could be right arm off break • Spinner bowler could be left-arm orthodox	Recommend 3 players in each category where top 1 of each category is the first priority in the auction as follow: Right-arm leg-break=1 Right-arm off-break=1	1. Create a list of 30 Bowlers who are the leading wicket takers. 2. Filter the list to include 6 bowlers from each said category as per the most wicket takers. 3. Filter the list to include 3 bowlers from each said category as per the lowest economy rate.

	• Spinner bowler could be left-arm chinaman	Left-arm Orthodox=1 Left-arm Chinaman=1	
R5	Pacers includes the following: • Right arm fast • Right arm medium • Left arm fast • Left arm medium	Recommend 3 players in each category where top 1 of each category is the priority in the action as follow: Right-arm fast=1 Left-arm fast=1 Right-arm medium=1 Left-arm medium=1	1. Create a list of 30 Bowlers who are the leading wicket takers. 2. Filter the list to include 6 bowlers from each said category as per the most wicket takers. 3. Filter the list to include 3 bowlers from each said category as per the lowest economy rate.
R6	All-rounder • Batted at any position in the batting order • Any bowling skill [spinner or pacer]	Recommend 6 players' list, where top 3 is the first priority in the auction	1. If batted at position in No-1,2,3, then Selection Criteria of R1 should apply. 2. If batted at position in No-4,5 then Selection Criteria of R2 should apply. 3. If batted at position in No-6,7 then Selection Criteria of R3 should apply. 4. If spinner, then Selection Criteria of R4 should apply 5. If pacer, then Selection Criteria of R5 should apply 6. All-rounder should be in the top 10 highest MOM awards Select the top 3 all-rounder who appears for the most occurrences for the above 6 criteria
R7	Wicket-keeper	Recommend 4 players' list, where top 2 is the first priority in the auction	1. If batted at position in No-1,2,3, then Selection Criteria of R1 should apply. 2. If batted at position in No-4,5 then Selection Criteria of R2 should apply. 3. If batted at position in No-6,7 then Selection Criteria of R3 should apply. Select the top 2 wicket keepers who appears for the most occurrences for the above 3 criteria

1.3 Assumptions:

Following are some of the assumptions considered in this project only [since this is not real project]:

1. Project is created based on the data available from 2008 to 2017. No data is present from 2018 to 2019, so we are simulating a business scenario as if the current year is 2018 i.e. if player's age is a criteria then age will be calculated as of 2018 (and not 2019).
2. All the players are available for auction meaning that we will consider the entire data into this simulated project. There are no players which an existing franchise team is holding on.
3. No manipulation or data skewness is done on top of the extracted data from dataworld.com. The entire data is used as it was extracted. If certain data is mismatch, no correction is done as part of this project such that data originality is retained.
4. Simulated business case is not considering the country and environment to form the team composition since data is not available on the environment such as bouncy pitch, cloudy weather, future schedule in South Africa or India, etc.
5. Umpires are not considered. Umpires data from the extracted ones are removed.

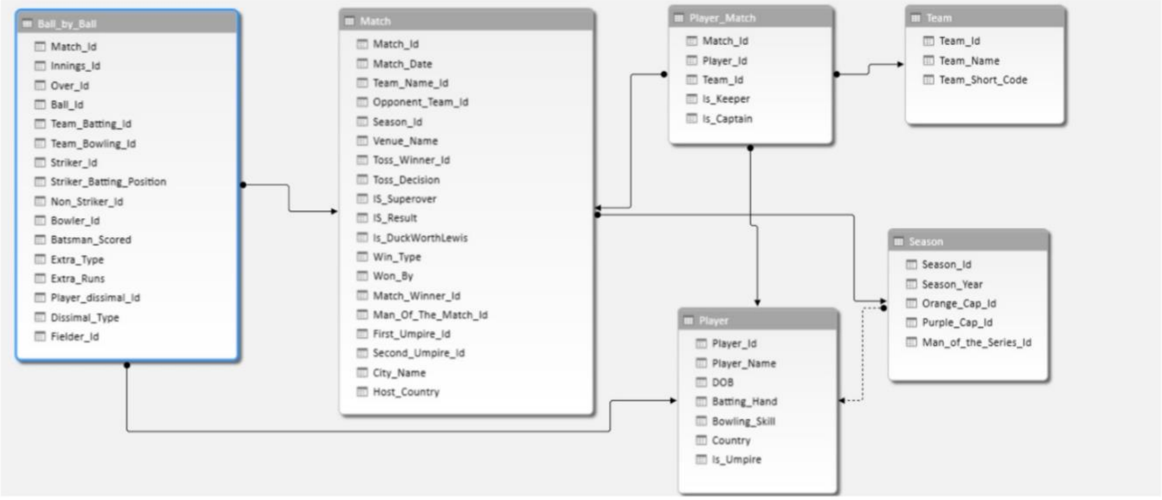
2.0 Data Set-Up

2.1 Data Extraction

IPL data is extracted from the dataworld.com in the form of a zip file. In this zip file, 5 csv files are present namely – Ball_By_Ball.csv, Match.csv, Player.csv, Player_Match.csv and Tean.csv.

Website source: https://data.world/raghu543/ipl-data-till-2017

And the following ER diagram is given in the website for understanding of these csv files as DB tables, its attributes and its relationship among them.



2.2 Data Normalization

These 5 csv files as DB tables are normalized and created 16 csv files as DB tables. Following are the actions performed:

1. From TEAM, attribute named as “Team_Short_Code” is removed.
2. From PLAYER, attribute named as “Is_umpire” is removed.
3. Season csv file is not received from extraction, so that DB table is not considered as shown in the above ER diagram.
4. In the MATCH, firstly, attribute named as “Season_id” is not present, and instead “Season_Year” is present. Secondly, three more attributes named as “Is_superover”, “Is_result” and “Is_DuckworthLewis” are missing, instead “Outcome_type” attribute is present holding the same information. Finally, two more attributes named as “First_umpire_id” and “second_umpire_id” are moved
5. MATCH table is normalized into 6 DB tables namely – MATCHES, HOME_TEAM, AWAY_TEAM, MATCHES_TOSS, MATCHES_RESULT, MATCHES_MOM
6. MATCHES_PLAYER_TEAM is renamed from PLAYER_MATCH
7. BALL_BY_BALL table is normalized into 7 DB tables namely – MATCHES_INNINGS_BATTING, MATCHES_INNINGS_BOWLING, BALL_BY_BALL_STRIKER, BALL_BY_BALL_NON_STRIKER, BALL_BY_BALL_BOWLER, BALL_BY_BALL_WICKET, BALL_BY_BALL_OUT_FIELDER. Removed the data duplication that arises after normalization of the original table. No data is tempered.

2.3 Importing CSV into ACCESS

All the 16 csv files are imported into the ACCESS system into their same named DB tables. During the import, following are performed:

1. Attributes that are the candidate of primary and foreign keys are marked as LONG INTEGER.
2. Indexes are set based on the attributes purpose and data, such as, while importing TEAM, Team_ID attribute is set as Index=YES(No Duplicates) while Team_Name attribute is set as Index=No.
3. Primary key is not set up during import. Primary and Foreign keys will be set later using DDL statements.

2.4 Adding Primary Key to each tables using DDL statements

Table Name	Primary Key – DDL Statements

TEAM	`ALTER TABLE Team`
	`    ADD CONSTRAINT pk_TeamID`
	`        PRIMARY KEY(Team_id)`

PLAYER	ALTER TABLE Player ADD CONSTRAINT pk_PlayerId PRIMARY KEY(Player_id)
MATCHES	ALTER TABLE Matches ADD CONSTRAINT pk_MatchId PRIMARY KEY(Match_id)
MATCHES_PLAYER_TEAM	ALTER TABLE Matches_Player_Team ADD CONSTRAINT pk_MatchId_PlayerId_TeamId PRIMARY KEY(Match_Id,Player_Id,Team_Id)
HOME_TEAM	ALTER TABLE Home_Team ADD CONSTRAINT pk_MatchId_TeamId PRIMARY KEY(Match_Id,Team_Id)
AWAY_TEAM	ALTER TABLE Away_Team ADD CONSTRAINT pk_at_MatchId_TeamId PRIMARY KEY(Match_Id,Team_Id)
MATCHES_TOSS	ALTER TABLE Matches_Toss ADD CONSTRAINT pk_mt_MatchId PRIMARY KEY(Match_id)
MATCHES_RESULT	ALTER TABLE Matches_Result ADD CONSTRAINT pk_mr_MatchId PRIMARY KEY(Match_id)
MATCHES_MOM	ALTER TABLE Matches_mom ADD CONSTRAINT pk_mm_MatchId PRIMARY KEY(Match_id)
MATCHES_INNINGS_TEAM_BATTING	ALTER TABLE Matches_Innings_Team_Batting ADD CONSTRAINT pk_mitb_MatchId_InningsNo PRIMARY KEY(Match_Id,Innings_No)
MATCHES_INNINGS_TEAM_BOWLING	ALTER TABLE Matches_Innings_Team_Bowling ADD CONSTRAINT pk_mitbo_MatchId_InningsNo PRIMARY KEY(Match_Id,Innings_No)
BALL_BY_BALL_STRIKER_RUNS	ALTER TABLE Ball_By_Ball_Striker_Runs ADD CONSTRAINT pk_bbbsr_MatchId_InningsNo_OverId_BallId PRIMARY KEY(Match_Id,Innings_No,Over_Id,Ball_Id)
BALL_BY_BALL_NON_STRIKER	ALTER TABLE Ball_By_Ball_Non_Striker ADD CONSTRAINT pk_bbbns_MatchId_InningsNo_OverId_BallId PRIMARY KEY(Match_Id,Innings_No,Over_Id,Ball_Id)

BALL_BY_BALL_BOWLER	ALTER TABLE Ball_By_Ball_Bowler ADD CONSTRAINT pk_bbbb_MatchId_InningsNo_OverId_BallId PRIMARY KEY(Match_Id,Innings_No,Over_Id,Ball_Id)
BALL_BY_BALL_WICKETS	ALTER TABLE Ball_By_Ball_Wickets ADD CONSTRAINT pk_bbbw_MatchId_InningsNo_OverId_BallId PRIMARY KEY(Match_Id,Innings_No,Over_Id,Ball_Id)
BALL_BY_BALL_OUT_FIELDER	ALTER TABLE Ball_By_Ball_Out_Fielder ADD CONSTRAINT pk_bbbof_MatchId_InningsNo_OverId_BallId PRIMARY KEY(Match_Id,Innings_No,Over_Id,Ball_Id)

2.5 Adding Foreign Key to each tables using DDL statements

Table Name	Primary Key – DDL Statements
MATCHES_PLAYER_TEAM	ALTER TABLE Matches_Player_Team ADD CONSTRAINT fk_mpt_MatchId FOREIGN KEY(Match_Id) REFERENCES Matches(Match_Id) ON DELETE CASCADE ON UPDATE CASCADE ALTER TABLE Matches_Player_Team ADD CONSTRAINT fk_mpt_PlayerId FOREIGN KEY(Player_Id) REFERENCES Player(Player_Id) ON DELETE CASCADE ON UPDATE CASCADE ALTER TABLE Matches_Player_Team ADD CONSTRAINT fk_mpt_TeamId FOREIGN KEY(Team_Id) REFERENCES Team(Team_Id) ON DELETE CASCADE ON UPDATE CASCADE
HOME_TEAM	ALTER TABLE Home_Team ADD CONSTRAINT fk_ht_MatchId FOREIGN KEY(Match_Id) REFERENCES Matches(Match_Id) ON DELETE CASCADE ON UPDATE CASCADE ALTER TABLE Home_Team ADD CONSTRAINT fk_ht_TeamNameId FOREIGN KEY(Team_Name_id) REFERENCES Team(Team_Id) ON DELETE CASCADE ON UPDATE CASCADE
AWAY_TEAM	ALTER TABLE Away_Team ADD CONSTRAINT fk_at_MatchId FOREIGN KEY(Match_Id) REFERENCES Matches(Match_Id) ON DELETE CASCADE ON UPDATE CASCADE ALTER TABLE Away_Team ADD CONSTRAINT fk_at_OpponentTeamId FOREIGN KEY(Opponent_Team_id) REFERENCES Team(Team_Id) ON DELETE CASCADE

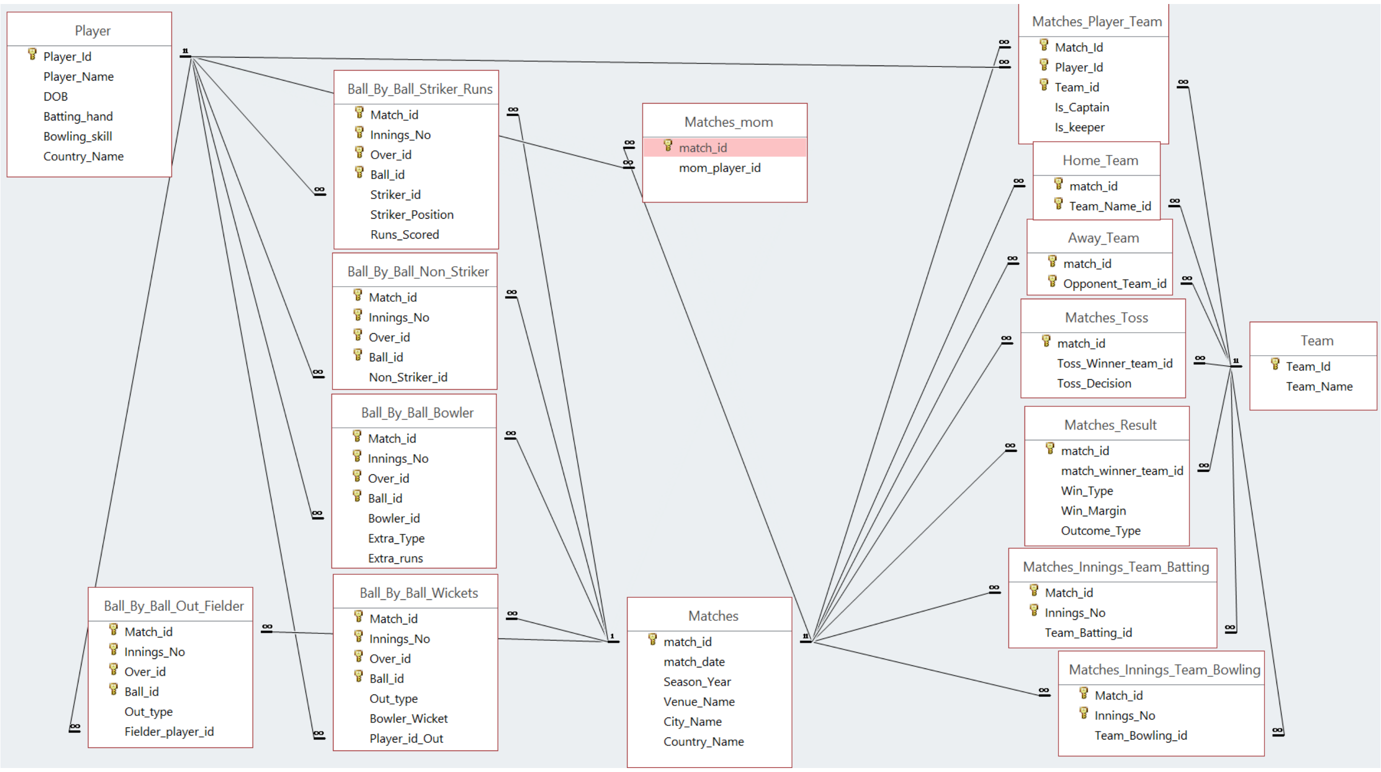
	ON UPDATE CASCADE
MATCHES_TOSS	ALTER TABLE Matches_Toss ADD CONSTRAINT fk_mt_MatchId FOREIGN KEY(Match_Id) REFERENCES Matches(Match_Id) ON DELETE CASCADE ON UPDATE CASCADE ALTER TABLE Matches_Toss ADD CONSTRAINT fk_mt_TossWinnerTeamId FOREIGN KEY(Toss_Winner_team_id) REFERENCES Team(Team_Id) ON DELETE CASCADE ON UPDATE CASCADE
MATCHES_RESULT	ALTER TABLE Matches_Result ADD CONSTRAINT fk_mr_MatchId FOREIGN KEY(Match_Id) REFERENCES Matches(Match_Id) ON DELETE CASCADE ON UPDATE CASCADE ALTER TABLE Matches_Result ADD CONSTRAINT fk_mr_MatchWinnerTeamId FOREIGN KEY(match_winner_team_id) REFERENCES Team(Team_Id) ON DELETE CASCADE ON UPDATE CASCADE
MATCHES_MOM	ALTER TABLE Matches_mom ADD CONSTRAINT fk_mm_MatchId FOREIGN KEY(Match_Id) REFERENCES Matches(Match_Id) ON DELETE CASCADE ON UPDATE CASCADE ALTER TABLE Matches_mom ADD CONSTRAINT fk_mm_MomPlayerId FOREIGN KEY(mom_player_id) REFERENCES Player(Player_Id) ON DELETE CASCADE ON UPDATE CASCADE
MATCHES_INNINGS_TEAM_BATTING	ALTER TABLE Matches_Innings_Team_Batting ADD CONSTRAINT fk_mitb_MatchId FOREIGN KEY(Match_Id) REFERENCES Matches(Match_Id) ON DELETE CASCADE ON UPDATE CASCADE ALTER TABLE Matches_Innings_Team_Batting ADD CONSTRAINT fk_mitb_TeamBattingId FOREIGN KEY(Team_Batting_id) REFERENCES Team(Team_Id) ON DELETE CASCADE ON UPDATE CASCADE
MATCHES_INNINGS_TEAM_BOWLING	ALTER TABLE Matches_Innings_Team_Bowling ADD CONSTRAINT fk_mitbo_MatchId FOREIGN KEY(Match_Id) REFERENCES Matches(Match_Id) ON DELETE CASCADE ON UPDATE CASCADE ALTER TABLE Matches_Innings_Team_Bowling ADD CONSTRAINT fk_mitbo_TeamBowlingId FOREIGN KEY(Team_Bowling_id) REFERENCES Team(Team_Id)

	ON DELETE CASCADE ON UPDATE CASCADE
BALL_BY_BALL_STRIKER_RUNS	ALTER TABLE Ball_By_Ball_Striker_Runs ADD CONSTRAINT fk_bbbsr_MatchId FOREIGN KEY(Match_Id) REFERENCES Matches(Match_Id) ON DELETE CASCADE ON UPDATE CASCADE ALTER TABLE Ball_By_Ball_Striker_Runs ADD CONSTRAINT fk_bbbsr_StrikerId FOREIGN KEY(Striker_Id) REFERENCES Player(Player_Id) ON DELETE CASCADE ON UPDATE CASCADE
BALL_BY_BALL_NON_STRIKER	ALTER TABLE Ball_By_Ball_Non_Striker ADD CONSTRAINT fk_bbbns_MatchId FOREIGN KEY(Match_Id) REFERENCES Matches(Match_Id) ON DELETE CASCADE ON UPDATE CASCADE ALTER TABLE Ball_By_Ball_Non_Striker ADD CONSTRAINT fk_bbbns_NonStrikerId FOREIGN KEY(Non_Striker_id) REFERENCES Player(Player_Id) ON DELETE CASCADE ON UPDATE CASCADE
BALL_BY_BALL_BOWLER	ALTER TABLE Ball_By_Ball_Bowler ADD CONSTRAINT fk_bbbb_MatchId FOREIGN KEY(Match_Id) REFERENCES Matches(Match_Id) ON DELETE CASCADE ON UPDATE CASCADE ALTER TABLE Ball_By_Ball_Bowler ADD CONSTRAINT fk_bbbb_BowlerId FOREIGN KEY(Bowler_id) REFERENCES Player(Player_Id) ON DELETE CASCADE ON UPDATE CASCADE
BALL_BY_BALL_WICKETS	ALTER TABLE Ball_By_Ball_Wickets ADD CONSTRAINT fk_bbbw_MatchId FOREIGN KEY(Match_Id) REFERENCES Matches(Match_Id) ON DELETE CASCADE ON UPDATE CASCADE ALTER TABLE Ball_By_Ball_Wickets ADD CONSTRAINT fk_bbbw_PlayerIdOut FOREIGN KEY(Player_id_Out) REFERENCES Player(Player_Id) ON DELETE CASCADE ON UPDATE CASCADE
BALL_BY_BALL_OUT_FIELDER	ALTER TABLE Ball_By_Ball_Out_Fielder ADD CONSTRAINT fk_bbbof_MatchId FOREIGN KEY(Match_Id) REFERENCES Matches(Match_Id) ON DELETE CASCADE ON UPDATE CASCADE ALTER TABLE Ball_By_Ball_Out_Fielder

	ADD CONSTRAINT fk_bbbof_FielderPlayerId FOREIGN KEY(Fielder_player_id) REFERENCES Player(Player_Id) ON DELETE CASCADE ON UPDATE CASCADE

2.6 ER Diagram

Presented below is the ER diagram after normalization in the system.



2.7 Database Table and its attributes: Description, purpose and relationship

Following is the tabular format showcasing the names of the database tables, purpose of DB tables, attributes, purpose of the attributes, primary key of DB tables, foreign key of DB tables and the relationship between DB tables.

DB Table Name	DB Table Purpose	DB Table Attributes	Attributes Description	Primary Key	Foreign Key	Relationship
TEAM	It holds all the names of the franchise teams played till date from 2008.	Team_id	Unique identification number of each franchise team	Y		1. One-2-Many relationship with the following DB tables: - Matches_Player_Team - Home_Team - Away_Team - Matches_Toss - Matches_Result - Matches_Innings_Team_Batting
		Team_ Name	Names of the franchise teams			

						- Matches_Innings_Team_Bowling
PLAYER	It holds all the names of the player from 2008 to 2017 irrespective of the teams. Player list contains captains, wicket keepers, batman and bowlers.	Player_id	Unique identification number of each player	Y		1. One-2-Many relationship with the following DB tables: - Matches_Player_Team - Matches_mom - Ball-By_Ball_Striker_Runs - Ball_By_Ball_Non-Striker - Ball_By_Ball_Bowler - Ball_By_Ball_Wickets - Ball_By_Ball_Out_Fielder
		Player_ Name	Names of the players			
		DOB	Date of Birth of the players - can use used to determine the age of the player as of current date			
		Batting_ Hand	Holds the player's batting hand - left handed or right handed			
		Bowling_ Skill	Holds the player's bowling skills: - pacers or spinners - fast, medium, fast-medium pacers - legbreak, offbreak, orthodox or chinaman - right arm bowler or left arm bowler			
		Country_ Name	Player's Nationality - applicable for the overseas player business rules			
MATCHES	It holds all the matches played from 2008 to 2017.	Match_id	Unique identification number of each match played	Y		1. One-2-One relationship with the following DB tables: - Matches_Toss - Matches_Result - Matches_mom 2. One-2-Many relationship with the following DB tables: - Matches_Player_Team - Home_Team - Away_Team - Matches_Innings_Team_Batting - Matches_Innings_Team_Bowling - Ball-By_Ball_Striker_Runs - Ball_By_Ball_Non-Striker - Ball_By_Ball_Bowler - Ball_By_Ball_Wickets - Ball_By_Ball_Out_Fielder
		Match_Date	Date on which individual matches were played			
		Season_ Year	Year on which matches were played - can be used to filter out matches played recently vs. in the past			
		Venue_ Name	Name of the ground where matches were played including the grounds of South Africa and UAE			
		City_Name	Name of the City where matches were played			
		Country_ Name	Name of the country where matches were played			
MATCHES_ PLAYER_ TEAM [junction table]	It holds which player is playing a match for which franchise team.	Match_id	Unique identification number of each match played	Y	Y	One-2-Many relationship with Matches
		Player_id	Unique identification number of each player	Y	Y	One-2-Many relationship with Player
		Team_id	Unique identification number of each franchise team	Y	Y	One-2-Many relationship with Team
		Is_Captain	Determine which player is the captain for that particular match			
		Is_Keeper	Determine which player is the wicker keeper for that particular match			
HOME_ TEAM [junction table]	It holds the information of which team is one of the team (named as home team) playing that particular match	Match_id	Unique identification number of the match to extract information about a particular match	Y	Y	One-2-Many relationship with Matches
		Team_Name_ id	Holds the unique identification number of the team to determine the name of one of the team among which match was played	Y	Y	One-2-Many relationship with Team

AWAY_TEAM [junction table]	It holds the information of which team is the other team (named as visiting team) playing that particular match	Match_id	Unique identification number of the match to extract information about a particular match	Y	Y	One-2-Many relationship with Matches
		Opponent_Team_ id	Holds the unique identification number of the team to determine the name of the other team among which match was played	Y	Y	One-2-Many relationship with Team
MATCHES_TOSS	It holds the information of which team is the toss winner for that particular match and what is the decision taken during toss.	Match_id	Unique identification number of the match to extract information about a particular match	Y	Y	One-2-One relationship with Matches
		Toss_Winner_Team_ id	Holds the unique identification number of the team to determine which team has won the toss for that match		Y	One-2-Many relationship with Team
		Toss_Decision	Holds the information taken during the toss			
MATCHES_RESULT	It holds the information about the result of the matches: - Won/Lost/Tie - Winner team - Win Margin	Match_id	Unique identification number of the match to extract information about a particular match	Y	Y	One-2-One relationship with Matches
		Match_Winner_Team_ id	Holds the unique identification number of the team to determine which team is the winner for that match		Y	One-2-Many relationship with Team
		Win_ Type	Holds the information if won or tie or no result. If won, then if won by runs or wickets			
		Win_ Margin	Holds the information of the number of runs or wickets if won, otherwise 0 for all other cases			
		Outcome_Type	Holds the information if that match resulted in a result (i.e. either win or lose) or abandoned or super over			
MATCHES_MOM	It holds the information which player was the man-of-the-match of that particular match	Match_id	Unique identification number of the match to extract information about a particular match	Y	Y	One-2-One relationship with Matches
		Mom_Player_id	Holds the unique identification number of the player who was declared man of the match for that particular match		Y	One-2-Many relationship with Player
MATCHES_INNINGS_TEAM_BATTING [junction table]	It holds the information which team was batting in which innings for that particular match. Innings can 1, 2, 3, 4 (3 or 4 in case of super over)	Match_id	Unique identification number of the match to extract information about a particular match	Y	Y	One-2-Many relationship with Matches
		Innings_No	Holds the Innings number for that particular match	Y		
		Team_Batting_ id	Holds the unique identification number of the team who was batting during a particular innings of a particular match		Y	One-2-Many relationship with Team
MATCHES_INNINGS_TEAM_BOWLING	It holds the information which team was bowling in which innings for that	Match_id	Unique identification number of the match to extract information about a particular match	Y	Y	One-2-Many relationship with Matches
		Innings_No	Holds the Innings number for that particular match	Y		

[junction table]	particular match. Innings can 1, 2, 3, 4 (3 or 4 in case of super over)	Team_ Bowling_ id	Holds the unique identification number of the team who was bowling during a particular innings of a particular match		Y	One-2-Many relationship with Team
BALL_BY_ BALL_ STRIKER_ RUNS [junction table]	It holds the following information for each individual balls being bowled from 2008 till 2017 in every matches being played: - who was the striker for that ball - what was the striker position - how many runs scored in that ball	Match_ id	Unique identification number of the match to extract information about a particular match	Y	Y	One-2-Many relationship with Matches
		Innings_ No	Holds the Innings number for that particular match	Y		
		Over_ id	Holds the information of the over number	Y		
		Ball_ id	Holds the information of the ball number for a particular over	Y		
		Striker_ id	Holds the unique identification number of the player who was the striker for that particular ball		Y	One-2-Many relationship with Player
		Striker_ position	Holds the position number according to the batting order where striker was playing that ball in that match innings			
		Runs_ Scored	Holds the information on the amount of the runs being scored by the striker off that particular ball in that match innings			
BALL_BY_ BALL_ NON_ STRIKER [junction table]	It holds the following information for each individual balls being bowled from 2008 till 2017 in every matches being played: - who was the non striker for that ball	Match_ id	Unique identification number of the match to extract information about a particular match	Y	Y	One-2-Many relationship with Matches
		Innings_ No	Holds the Innings number for that particular match	Y		
		Over_ id	Holds the information of the over number	Y		
		Ball_ id	Holds the information of the ball number for a particular over	Y		
		Non_ Striker_ id	Holds the unique identification number of the player who was the non-striker for that particular ball		Y	One-2-Many relationship with Player
BALL_BY_ BALL_ BOWLER [junction table]	It holds the following information for each individual balls being bowled from 2008 till 2016 in every matches being played: - who was the bowler for that ball - If extra happened, what type of extra and how extra runs	Match_ id	Unique identification number of the match to extract information about a particular match	Y	Y	One-2-Many relationship with Matches
		Innings_ No	Holds the Innings number for that particular match	Y		
		Over_ id	Holds the information of the over number	Y		
		Ball_ id	Holds the information of the ball number for a particular over	Y		
		Bowler_ id	Holds the unique identification number of the player who was the bowler for that particular ball		Y	One-2-Many relationship with Player
		Extra_ Type	Holds the information on the type of extras happened in that ball of that match innings			
		Extra_ Runs	Holds the information on the amount of the extra runs scored off that particular ball in that match innings			

BALL_BY_ BALL_ WICKETS [junction table]	It holds the information of that particular ball of that over in that match innings in which wicket fell along with dismissal type, if that wicket belongs to bowler or not and the player dismissed [Note: this table does not have the information of all the balls where wickets didn't fell]	Match_ id	Unique identification number of the match to extract information about a particular match	Y	Y	One-2-Many relationship with Matches
		Innings_ No	Holds the Innings number for that particular match	Y		
		Over_ id	Holds the information of the over number	Y		
		Ball_ id	Holds the information of the ball number for a particular over	Y		
		Out_ Type	Holds the information of the type of the dismissal			
		Bowler_ Wicket	Holds the information if the wicket belongs to the bowler or not			
		Player_id_ out	Holds the unique identification number of the player who was dismissed in that ball of that match innings		Y	One-2-Many relationship with Player
BALL_BY_ BALL_ OUT_ FIELDER [junction table]	It holds the information of that particular ball of that over in that match innings in which a fielder was involved in the dismissal& dismissal type. [Note: this table does not have the information of all the balls where wickets didn't fell and where no fielder was involved in the dismissal such as bowled]	Match_ id	Unique identification number of the match to extract information about a particular match	Y	Y	One-2-Many relationship with Matches
		Innings_ No	Holds the Innings number for that particular match	Y		
		Over_ id	Holds the information of the over number	Y		
		Ball_ id	Holds the information of the ball number for a particular over	Y		
		Out_ Type	Holds the information of the type of the dismissal			
		Fielder_ Player_ id	Holds the unique identification number of the player who was the fielder involved in that dismissal in that ball of that match innings		Y	One-2-Many relationship with Team

3.0 SQL Query for Business Requirements AND Output

3.1 SQL Query for Requirement #1

Following tabular format showcase the steps performed to fulfil each criteria under requirement #1 along with the SQL queries

3.1.1 SQL Query and Steps for Criteria #1

Criteria Description: Create a list of 30 Batsman who are the leading run scorers in that said batting position

Step #	Step Description	SQL Query

A	Creating a list of 30 batsman sorted by total runs scored in descending order and stored in temp table	SELECT TOP 30 p.Player_name, SUM(bbbsr.Runs_scored) AS Total_Runs
		INTO Req1_Cri1_TotalRuns
		FROM Ball_By_Ball_Striker_Runs AS bbbsr INNER JOIN
		Player AS p ON
		bbbsr.Striker_id=p.Player_id

		WHERE bbbsr.Striker_position IN (1,2,3)
		GROUP BY p.Player_name
		ORDER BY 2 DESC

3.1.2 SQL Query and Steps for Criteria #2

Criteria Description: Filter the list to have batting average of above 20

Step #	Step Description	SQL Query

A	Create a temp table and store total runs scored by Striker batted at position # 1, 2 or 3	SELECT p.Player_Name, SUM(Runs_Scored) AS Total_Runs
		INTO Buffer_Total_Runs
		FROM Ball_By_Ball_Striker_Runs AS bbbsr INNER JOIN
		     Player AS p ON
		     bbbsr.Striker_id = p.Player_id
		WHERE Striker_position IN (1,2,3)
		GROUP BY p.Player_Name

B	Create another temp table and store dismissal count of the Striker batted at position # 1, 2 or 3	SELECT p.Player_Name, COUNT(*) AS Out_Count
		INTO Buffer_Dismissal_Count
		FROM Ball_By_Ball_Wickets AS bbbw INNER JOIN
		     Player AS p ON
		     bbbw.Player_id_out = p.Player_id
		GROUP BY p.Player_Name

C	Create a third temp table by joining the 1st and 2nd temp table and store batting average (runs scored / dismissal count)	SELECT btr.Player_name AS Player_Name,
		       ROUND(btr.Total_Runs/bdc.Out_count,2) AS Bat_Avg
		INTO Buffer_Bat_avg
		FROM Buffer_Total_Runs AS btr INNER JOIN
		     Buffer_Dismissal_Count AS bdc ON
		     btr.Player_name = bdc.Player_name

D	Create a final table for criteria #2 based on the final table created for criteria #1 filtering the min. batting average of 20	SELECT r1c1tr.Player_name AS Player_name,
		       r1c1tr.Total_Runs AS Total_Runs,
		       bba.Bat_Avg AS Bat_Avg
		INTO Req1_Cri2_TotalRuns_BatAvg
		FROM Req1_Cri1_TotalRuns AS r1c1tr INNER JOIN
		     Buffer_Bat_Avg AS bba ON
		     r1c1tr.Player_name = bba.Player_name
		WHERE bba.Bat_avg >= 20
		ORDER BY 2 DESC, 3 DESC

E	Remove all the temp tables created for criteria #2. Keep the final table only	DROP TABLE Buffer_Total_Runs DROP TABLE Buffer_Dismissal_Count DROP TABLE Buffer_Bat_Avg

3.1.3 SQL Query and Steps for Criteria #3

Criteria Description: Filter the list to have deviation of scores of below 28

Step #	Step Description	SQL Query

A	Create a temp table and store the standard deviation of the match scores of	SELECT Player_name,
		       STDEV(Match_runs) AS Runs_Dev
		INTO Buffer_Runs_Dev
		FROM

	each Striker batted in position 1, 2 or 3	(SELECT p.Player_name AS Player_name, bbbsr.Match_id AS Match_id, SUM(bbbsr.Runs_scored) AS Match_Runs FROM Ball_By_Ball_Striker_Runs AS bbbsr INNER JOIN Player AS p ON bbbsr.Striker_id = p.Player_id GROUP BY p.Player_name, bbbsr.Match_id) GROUP BY Player_name HAVING STDEV(Match_runs) > 0

B	Create a final table for criteria #3 based on the final table created for criteria #2 filtering max. score deviation of 28	SELECT r1c2trba.Player_name AS Player_name,
		r1c2trba.Total_runs AS Total_Runs,
		r1c2trba.Bat_Avg AS Bat_Avg,
		ROUND(brd.Runs_dev,2) AS Score_Dev
		INTO Req1_Cri3_TotalRuns_BatAvg_ScoreDev
		FROM Req1_Cri2_TotalRuns_BatAvg AS r1c2trba INNER JOIN
		Buffer_Runs_Dev AS brd ON
		r1c2trba.Player_name = brd.Player_name
		WHERE brd.Runs_dev <= 28
		ORDER BY 2 DESC, 3 DESC, 4 DESC

C	Remove all the temp tables created for criteria #3. Keep the final table only	DROP TABLE Buffer_Runs_Dev

3.1.4 SQL Query and Steps for Criteria #4

Criteria Description: Filter the list to have strike rate of more than 104 in the first 8 overs

Step #	Step Description	SQL Query

A	Create a temp table and store the batting strike rate ((runs scored / balls faced) *100) in the first 8 overs for the striker batted in position 1, 2 or 3	SELECT p.Player_name,
		ROUND(((SUM(bbbsr.Runs_scored))/(COUNT(bbbsr.Ball_id)))*100,2) AS Bat_Strike_Rate_First_8_Overs
		INTO Buffer_Bat_Strike_Rate_First_8_Overs
		FROM Ball_By_Ball_Striker_Runs AS bbbsr INNER JOIN
		Player AS p ON
		bbbsr.Striker_id = p.Player_id
		WHERE bbbsr.Striker_position IN (1,2,3) AND
		bbbsr.Over_id < 9
		GROUP BY p.Player_name

B	Create a final table for criteria #4 based on the final table created for criteria #3 filtering min. batting average of first 8 overs of 104	SELECT r1c3.Player_name AS Player_name, r1c3.Total_Runs AS Total_Runs, r1c3.Bat_Avg AS Bat_Avg, r1c3.Score_Dev AS Score_Dev, Buf.Bat_Strike_Rate_First_8_Overs AS Bat_Strike_Rate_First_8_Overs INTO Req1_Cri4_TotalRuns_BatAvg_ScoreDev_BatStrikeRate FROM Req1_Cri3_TotalRuns_BatAvg_ScoreDev AS r1c3 INNER JOIN Buffer_Bat_Strike_Rate_First_8_Overs AS Buf ON r1c3.Player_name = Buf.Player_name WHERE Buf.Bat_Strike_Rate_First_8_Overs >= 104 ORDER BY 2 DESC, 3 DESC, 4 DESC, 5 DESC

C	Remove all the temp tables created for criteria #4. Keep the final table only	DROP TABLE Buffer_Bat_Strike_Rate_First_8_Overs

3.1.5 SQL Query and Steps for Criteria #5

Criteria Description: Filter the list to have received at least 5 MOM awards

Step #	Step Description	SQL Query

A	Create a temp table and store the count of mom awards for each striker	SELECT p.Player_name, COUNT(mm.match_id) AS mom_count
		INTO Buffer_mom_count
		FROM Matches_mom as mm INNER JOIN
		     Player AS p ON
		     mm.mom_player_id = p.Player_id
		GROUP BY p.Player_name

B	Create a final table for criteria #5 based on the final table created for criteria #4 filtering min. count of mom awards of 5	SELECT r1c4.Player_name AS Player_name,
		       r1c4.Total_runs AS Total_Runs,
		       r1c4.Bat_avg AS Bat_Avg,
		       r1c4.Score_dev AS Score_Dev,
		       r1c4.Bat_Strike_Rate_First_8_Overs AS Bat_Strike_Rate_First_8_Overs,
		       Buf.mom_count AS mom_count
		INTO Req1_Cri5_TotalRuns_BatAvg_ScoreDev_BatStrikeRate_MomCount
		FROM Req1_Cri4_TotalRuns_BatAvg_ScoreDev_BatStrikeRate AS r1c4 INNER JOIN
		         Buffer_mom_count AS Buf ON
		         r1c4.Player_name = Buf.Player_name
		WHERE Buf.mom_count >= 5
		ORDER BY 2 DESC, 3 DESC, 4 DESC, 5 DESC, 6 DESC

C	Remove all the temp tables created for criteria #5. Keep the final table only	DROP TABLE Buffer_mom_count

3.1.6 Finalize the final probable list of 10 batsman in opening and 1st down position

Step #	Step Description	SQL Query

A	Create a table to store the final probable list of 10 batsman	SELECT r1c5.Player_name AS Player_name,
		       p.Batting_hand AS Bat_Style,
		       r1c5.Total_runs AS Total_Runs,
		       r1c5.Bat_avg AS Bat_Avg,
		       r1c5.Score_dev AS Score_Dev,
		       r1c5.Bat_Strike_Rate_First_8_Overs AS Bat_Strike_Rate_First_8_Overs,
		       r1c5.mom_count AS mom_count
		INTO Req1_Probable_10_Batsman_Pos123
		FROM Req1_Cri5_TotalRuns_BatAvg_ScoreDev_BatStrikeRate_MomCount AS r1c5
		     INNER JOIN Player AS p
		     ON r1c5.Player_name = p.Player_name

B	Remove all the intermediate table created at each criteria level. Keep the	DROP TABLE Req1_Cri1_TotalRuns DROP TABLE Req1_Cri2_TotalRuns_BatAvg DROP TABLE Req1_Cri3_TotalRuns_BatAvg_ScoreDev DROP TABLE Req1_Cri4_TotalRuns_BatAvg_ScoreDev_BatStrikeRate DROP TABLE Req1_Cri5_TotalRuns_BatAvg_ScoreDev_BatStrikeRate_MomCount

	final one for this requirement.	
C	Add Priority Column in the final table	ALTER TABLE Req1_Probable_10_Batsman_Pos123 ADD COLUMN Priority INTEGER

D	Check at least 2 left handed batsman are present in the top 6 of the final list. If yes, Update Priority=1 for the top 6 in the final table. If no, discard it.	UPDATE Req1_Probable_10_Batsman_Pos123
		SET Priority = 1
		WHERE Player_Name IN
		      (SELECT TOP 6 Player_name
		      FROM Req1_Probable_10_Batsman_Pos123
		      ORDER BY Total_Runs DESC)
		      AND
		      (SELECT COUNT(*)
		       FROM
		           (SELECT TOP 6 Bat_Style
		            FROM Req1_Probable_10_Batsman_Pos123
		            ORDER BY Total_Runs DESC)
		      WHERE Bat_Style = "Left-hand bat") >= 2

E	Update Priority=2 for the rest	UPDATE Req1_Probable_10_Batsman_Pos123
		SET Priority = 2
		WHERE Priority IS NULL

3.2 SQL Query for Requirement #2

In the same way, following tabular format showcase the steps performed to fulfil each criteria under requirement #2 using the same SQL queries used for requirement #1

3.2.1 SQL Query and Steps for Criteria #1

Criteria Description: Create a list of 30 Batsman who are the leading run scorers in that said batting position

Step #	Step Description	SQL Query

A	Creating a list of 30 batsman sorted by total runs scored in descending order and stored in temp table	SELECT TOP 30 p.Player_name, SUM(bbbsr.Runs_scored) AS Total_Runs
		INTO Req2_Cri1_TotalRuns
		FROM Ball_By_Ball_Striker_Runs AS bbbsr INNER JOIN
		     Player AS p ON
		     bbbsr.Striker_id=p.Player_id
		WHERE bbbsr.Striker_position IN (4,5)
		GROUP BY p.Player_name
		ORDER BY 2 DESC

3.2.2 SQL Query and Steps for Criteria #2

Criteria Description: Filter the list to have batting average of above 12

Step #	Step Description	SQL Query

A	Create a temp table and store total runs scored by Striker	SELECT p.Player_Name, SUM(Runs_Scored) AS Total_Runs
		INTO Buffer_Total_Runs
		FROM Ball_By_Ball_Striker_Runs AS bbbsr INNER JOIN
		     Player AS p ON

	batted at position # 4 or 5	`     bbbsr.Striker_id = p.Player_id`
		`WHERE Striker_position IN (4,5)`
		`GROUP BY p.Player_Name`

B	Create another temp table and store dismissal count of the Striker batted at position # 4 or 5	`SELECT p.Player_Name, COUNT(*) AS Out_Count`
		`INTO Buffer_Dismissal_Count`
		`FROM Ball_By_Ball_Wickets AS bbbw INNER JOIN`
		`     Player AS p ON`
		`     bbbw.Player_id_out = p.Player_id`
		`GROUP BY p.Player_Name`

C	Create a third temp table by joining the 1st and 2nd temp table and store batting average (runs scored / dismissal count)	`SELECT btr.Player_name AS Player_Name,`
		`       ROUND(btr.Total_Runs/bdc.Out_count,2) AS Bat_Avg`
		`INTO Buffer_Bat_avg`
		`FROM Buffer_Total_Runs AS btr INNER JOIN`
		`     Buffer_Dismissal_Count AS bdc ON`
		`     btr.Player_name = bdc.Player_name`

D	Create a final table for criteria #2 based on the final table created for criteria #1 filtering the min. batting average of 12	`SELECT r2c1tr.Player_name AS Player_name,`
		`       r2c1tr.Total_Runs AS Total_Runs,`
		`       bba.Bat_Avg AS Bat_Avg`
		`INTO Req2_Cri2_TotalRuns_BatAvg`
		`FROM Req2_Cri1_TotalRuns AS r2c1tr INNER JOIN`
		`     Buffer_Bat_Avg AS bba ON`
		`      r2c1tr.Player_name = bba.Player_name`
		`WHERE bba.Bat_avg >= 12`
		`ORDER BY 2 DESC, 3 DESC`

E	Remove all the temp tables created for criteria #2. Keep the final table only	`DROP TABLE Buffer_Total_Runs`
		`DROP TABLE Buffer_Dismissal_Count`
		`DROP TABLE Buffer_Bat_Avg`

3.2.3 SQL Query and Steps for Criteria #3

Criteria Description: Filter the list to have deviation of scores of below 23

Step #	Step Description	SQL Query

A	Create a temp table and store the standard deviation of the match scores of each Striker batted in position 4 or 5	`SELECT Player_name,`
		`       STDEV(Match_runs) AS Runs_Dev`
		`INTO Buffer_Runs_Dev`
		`FROM`
		`(`
		`   SELECT p.Player_name AS Player_name,`
		`          bbbsr.Match_id AS Match_id,`
		`          SUM(bbbsr.Runs_scored) AS Match_Runs`
		`   FROM Ball_By_Ball_Striker_Runs AS bbbsr INNER JOIN`
		`        Player AS p ON`
		`        bbbsr.Striker_id = p.Player_id`
		`   GROUP BY p.Player_name, bbbsr.Match_id`
		`)`
		`GROUP BY Player_name`
		`HAVING STDEV(Match_runs) > 0`

B	Create a final table for criteria #3 based on	`SELECT r2c2trba.Player_name AS Player_name,`
		`       r2c2trba.Total_runs AS Total_Runs,`
		`       r2c2trba.Bat_Avg AS Bat_Avg,`

	the final table created for criteria #2 filtering max. score deviation of 28	ROUND(brd.Runs_dev,2) AS Score_Dev
		INTO Req2_Cri3_TotalRuns_BatAvg_ScoreDev
		FROM Req2_Cri2_TotalRuns_BatAvg AS r2c2trba INNER JOIN
		Buffer_Runs_Dev AS brd ON
		R2c2trba.Player_name = brd.Player_name
		WHERE brd.Runs_dev <= 23
		ORDER BY 2 DESC, 3 DESC, 4 DESC

C	Remove all the temp tables created for criteria #3. Keep the final table only	DROP TABLE Buffer_Runs_Dev

3.2.4 SQL Query and Steps for Criteria #4

Criteria Description: Filter the list to have strike rate of more than 102 from 7th over to 15th over

Step #	Step Description	SQL Query

A	Create a temp table and store the batting strike rate ((runs scored / balls faced) *100) from 7th over to 15th over for the striker batted in position 4 or 5	SELECT p.Player_name,
		ROUND(((SUM(bbbsr.Runs_scored))/(COUNT(bbbsr.Ball_id)))*100,2) AS Bat_Strike_Rate_7th_15th_Over
		INTO Buffer_Bat_Strike_Rate_7th_15th_Over
		FROM Ball_By_Ball_Striker_Runs AS bbbsr INNER JOIN
		Player AS p ON
		bbbsr.Striker_id = p.Player_id
		WHERE bbbsr.Striker_position IN (4,5) AND
		bbbsr.Over_id BETWEEN 7 AND 15
		GROUP BY p.Player_name

B	Create a final table for criteria #4 based on the final table created for criteria #3 filtering min. batting average from 7th over to 15th over of 104	SELECT r2c3.Player_name AS Player_name, r2c3.Total_Runs AS Total_Runs, r2c3.Bat_Avg AS Bat_Avg, r2c3.Score_Dev AS Score_Dev, Buf.Bat_Strike_Rate_7th_15th_Over AS Bat_Strike_Rate_7th_15th_Over INTO Req2_Cri4_TotalRuns_BatAvg_ScoreDev_BatStrikeRate FROM Req2_Cri3_TotalRuns_BatAvg_ScoreDev AS r2c3 INNER JOIN Buffer_Bat_Strike_Rate_7th_15th_Over AS Buf ON R2c3.Player_name = Buf.Player_name WHERE Buf.Bat_Strike_Rate_7th_15th_Over >= 102 ORDER BY 2 DESC, 3 DESC, 4 DESC, 5 DESC
C	Remove all the temp tables created for criteria #4. Keep the final table only	DROP TABLE Buffer_Bat_Strike_Rate_7th_15th_Over

3.2.5 SQL Query and Steps for Criteria #5

Criteria Description: Filter the list to have received at least 3 MOM awards

Step #	Step Description	SQL Query

A	Create a temp table and store the count of	SELECT p.Player_name, COUNT(mm.match_id) AS mom_count
		INTO Buffer_mom_count
		FROM Matches_mom as mm INNER JOIN
		Player AS p ON

	mom awards for each striker	mm.mom_player_id = p.Player_id
		GROUP BY p.Player_name

B	Create a final table for criteria #5 based on the final table created for criteria #4 filtering min. count of mom awards of 5	SELECT r2c4.Player_name AS Player_name, r2c4.Total_runs AS Total_Runs, r2c4.Bat_avg AS Bat_Avg, r2c4.Score_dev AS Score_Dev, r2c4.Bat_Strike_Rate_First_8_Overs AS Bat_Strike_Rate_7th_15th_Over, Buf.mom_count AS mom_count INTO Req2_Cri5_TotalRuns_BatAvg_ScoreDev_BatStrikeRate_MomCount FROM Req2_Cri4_TotalRuns_BatAvg_ScoreDev_BatStrikeRate AS r2c4 INNER JOIN Buffer_mom_count AS Buf ON R2c4.Player_name = Buf.Player_name WHERE Buf.mom_count >= 3 ORDER BY 2 DESC, 3 DESC, 4 DESC, 5 DESC, 6 DESC
C	Remove all the temp tables created for criteria #5. Keep the final table only	DROP TABLE Buffer_mom_count

3.2.6 Finalize the final probable list of 10 batsman in the middle order for position #4 and #5

Step #	Step Description	SQL Query
A	Create a table to store the final probable list of 8 batsman	SELECT r2c5.Player_name AS Player_name, p.Batting_hand AS Bat_Style, r2c5.Total_runs AS Total_Runs, r2c5.Bat_avg AS Bat_Avg, r2c5.Score_dev AS Score_Dev, r2c5.Bat_Strike_Rate_First_8_Overs AS Bat_Strike_Rate_7th_15th_Over, r2c5.mom_count AS mom_count INTO Req2_Probable_8_Batsman_Pos45 FROM Req2_Cri5_TotalRuns_BatAvg_ScoreDev_BatStrikeRate_MomCount AS r2c5 INNER JOIN Player AS p ON r2c5.Player_name = p.Player_name
B	Remove all the intermediate table created at each criteria level. Keep the final one for this requirement.	DROP TABLE Req2_Cri1_TotalRuns DROP TABLE Req2_Cri2_TotalRuns_BatAvg DROP TABLE Req2_Cri3_TotalRuns_BatAvg_ScoreDev DROP TABLE Req2_Cri4_TotalRuns_BatAvg_ScoreDev_BatStrikeRate DROP TABLE Req2_Cri5_TotalRuns_BatAvg_ScoreDev_BatStrikeRate_MomCount
C	Add Priority Column in the final table	ALTER TABLE Req2_Probable_8_Batsman_Pos45 ADD COLUMN Priority INTEGER

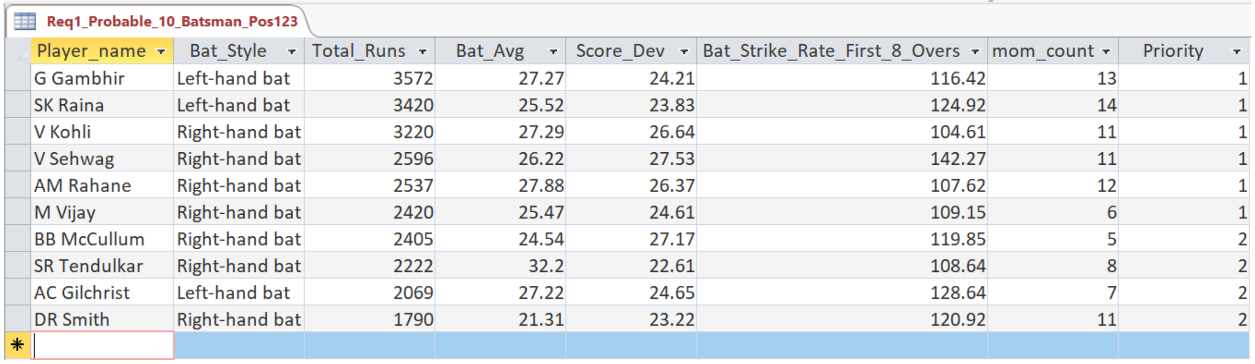
D	Check at least 1 left handed batsman is present in the top 4 of the final list. If yes, Update Priority=1 for top 4 in the final table. If no, discard it.	UPDATE Req2_Probable_8_Batsman_Pos45
		SET Priority = 1
		WHERE Player_Name IN
		(SELECT TOP 4 Player_name
		FROM Req2_Probable_8_Batsman_Pos45
		ORDER BY Total_Runs DESC)
		AND
		(SELECT COUNT(*)
		FROM
		(SELECT TOP 4 Bat_Style
		FROM Req2_Probable_8_Batsman_Pos45

		`ORDER BY Total_Runs DESC)`
		`WHERE Bat_Style = "Left-hand bat") >= 1`

E	Update Priority=2 for the rest	`UPDATE  Req2_Probable_8_Batsman_Pos45`
		`SET Priority = 2`
		`WHERE Priority IS NULL`

4.0 Output

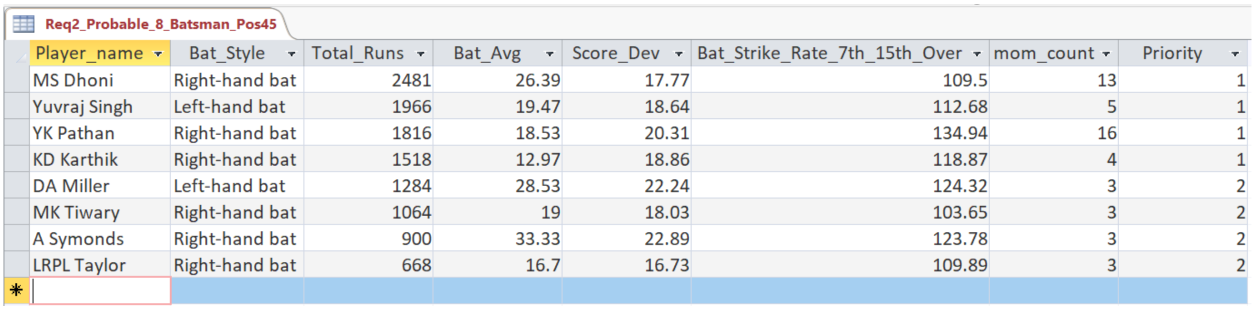
4.1 Output of Requirement #1



Req1_Probable_10_Batsman_Pos123

Player_name	Bat_Style	Total_Runs	Bat_Avg	Score_Dev	Bat_Strike_Rate_First_8_Overs	mom_count	Priority
G Gambhir	Left-hand bat	3572	27.27	24.21	116.42	13	1
SK Raina	Left-hand bat	3420	25.52	23.83	124.92	14	1
V Kohli	Right-hand bat	3220	27.29	26.64	104.61	11	1
V Sehwag	Right-hand bat	2596	26.22	27.53	142.27	11	1
AM Rahane	Right-hand bat	2537	27.88	26.37	107.62	12	1
M Vijay	Right-hand bat	2420	25.47	24.61	109.15	6	1
BB McCullum	Right-hand bat	2405	24.54	27.17	119.85	5	2
SR Tendulkar	Right-hand bat	2222	32.2	22.61	108.64	8	2
AC Gilchrist	Left-hand bat	2069	27.22	24.65	128.64	7	2
DR Smith	Right-hand bat	1790	21.31	23.22	120.92	11	2

4.1 Output of Requirement #2



Req2_Probable_8_Batsman_Pos45

Player_name	Bat_Style	Total_Runs	Bat_Avg	Score_Dev	Bat_Strike_Rate_7th_15th_Over	mom_count	Priority
MS Dhoni	Right-hand bat	2481	26.39	17.77	109.5	13	1
Yuvraj Singh	Left-hand bat	1966	19.47	18.64	112.68	5	1
YK Pathan	Right-hand bat	1816	18.53	20.31	134.94	16	1
KD Karthik	Right-hand bat	1518	12.97	18.86	118.87	4	1
DA Miller	Left-hand bat	1284	28.53	22.24	124.32	3	2
MK Tiwary	Right-hand bat	1064	19	18.03	103.65	3	2
A Symonds	Right-hand bat	900	33.33	22.89	123.78	3	2
LRPL Taylor	Right-hand bat	668	16.7	16.73	109.89	3	2

5.0 Disclosure

We can achieve the output of all the requirements by solving the individual criteria under each requirement via the already described steps in requirement #1 and #2 through SQL queries.

Closing the project with requirement #1 and #2 as solved and the rest of the requirements to be solved in future.

************** THANK YOU **************