IFT 530

Advanced Database Management Systems

Final Project Report

Prof. Robert Rucker

Cricketing Managerial System

Janak Sonalkar

Sahil Sharma

Nihar Shetty

Class #91063

Submission Date: 3rd December 2021

INTRODUCTION

Cricket is a popular game played in many parts of Asia, England, and Australia. The tremendous increase in its popularity has demanded an addition in the variety and the number of tournaments held per year to support the increasing viewership among the fans of this sport. Hosting a single match requires a lot of effort let alone an entire tournament. It can get really tedious to manage ticket booking, track player statistics, match details, stadium availability. Additionally, deciding the format: T20 / ODI / Test, ball type, etc, participating teams, funders, managing stadiums, umpires, top players, commentators, arranging refreshments can take a toll on a team/individual by the end of the tournament.

With the proposed project we intend to reduce the workload of organizers by making a lot of information available with greater ease. The goal of our project is to manipulate all records instantly, thereby reducing employee's workload and improving their accuracy, which will improve their performance.

We propose to develop a system to manage cricket events which include managing stadium bookings, checking availability as well as player stats and tournament scheduler. By using it, cricket organizers can keep track of games in a more convenient and easy way than manually recording them. In the database we can expect to have player name, player age, country name, match details, player statistics, stadium availability etc. As part of our plan, we intend to develop a system which maintains records of various tournaments and tries to monitor every aspect of each tournament, including the teams, players, venues, and the coach tenures.

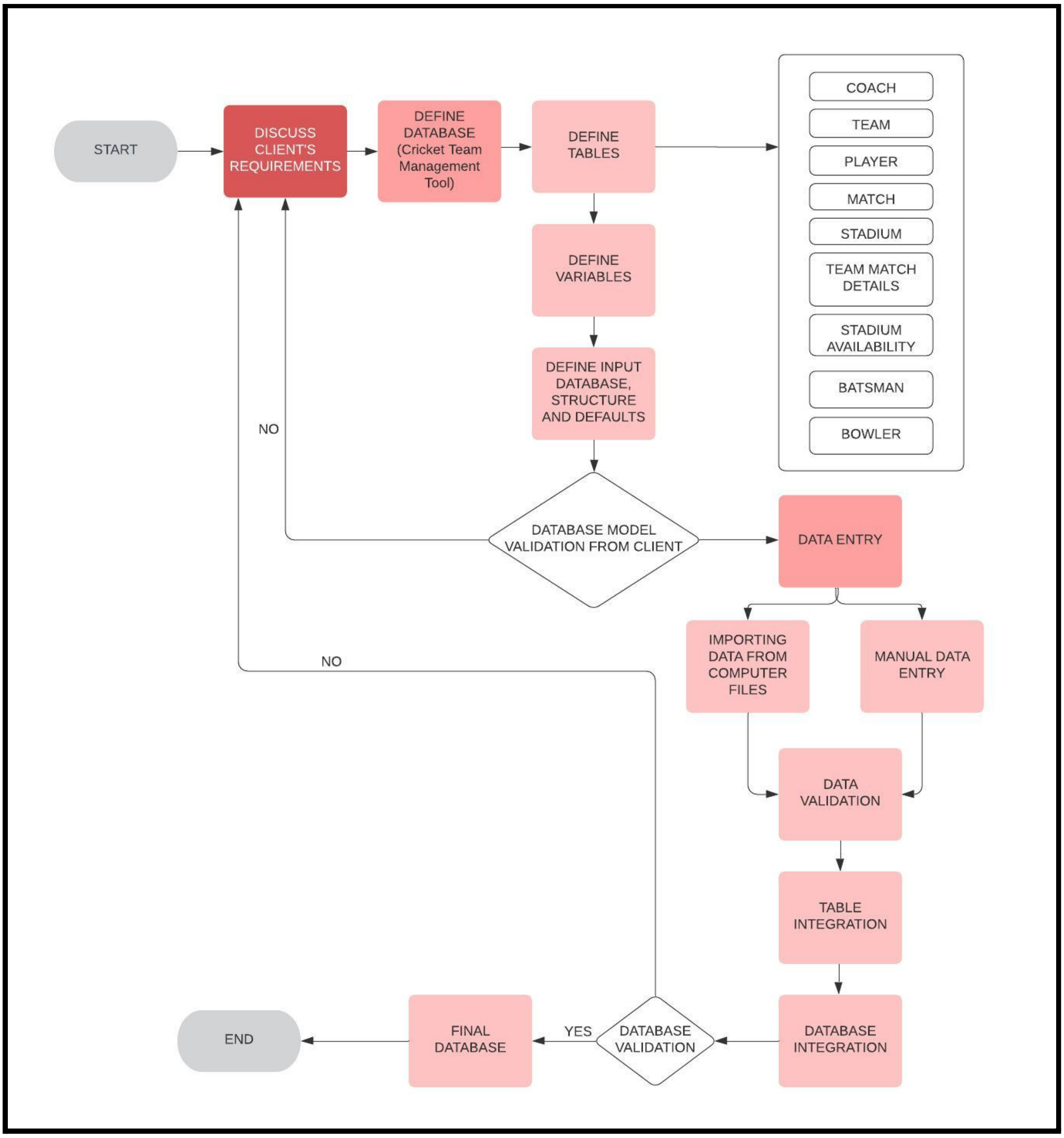
IMPORTANCE

Growing up in India, a cricket loving country, our love for the sport began at quite an early age. The sport has been significantly popular for a very long time now. It is always a topic of great interest when our parents talk about the famous players of their time and compare how the game has evolved since then. The young generation of India always look up to these cricketers and idealize them. The number of emerging young talent coming up in this sport makes it difficult for the selectors to choose from. Even as a spectator the audience is always curious whether their favourite player made the cut into the squad. The proposed project helps us provide the stats of these players in the database. Changes can be made based on the player performance and be used as a criterion for selection purpose.

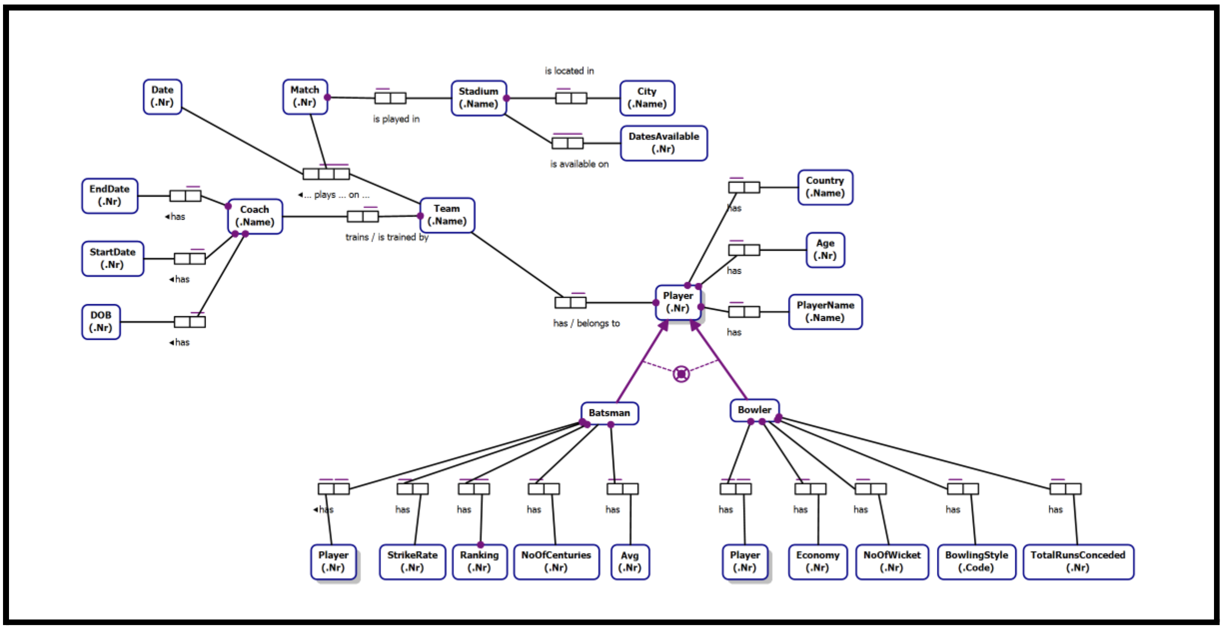
DESIGN REQUIREMENTS

Entity Requirements	**Data Requirements**
Can it provide the stats of a batsman? Can it provide the stats of a bowler? Can it provide the identification details of the players? Can it check for Stadium Availability? Can it provide information about the coach of the team?	**Player**: Player Name, Country, Age. **Batsman**: Strike rate, Ranking, Number of Centuries, Average. **Team**: Team Name, Coach of the team, Dates of matches. **Stadium**: Location, Availability date.

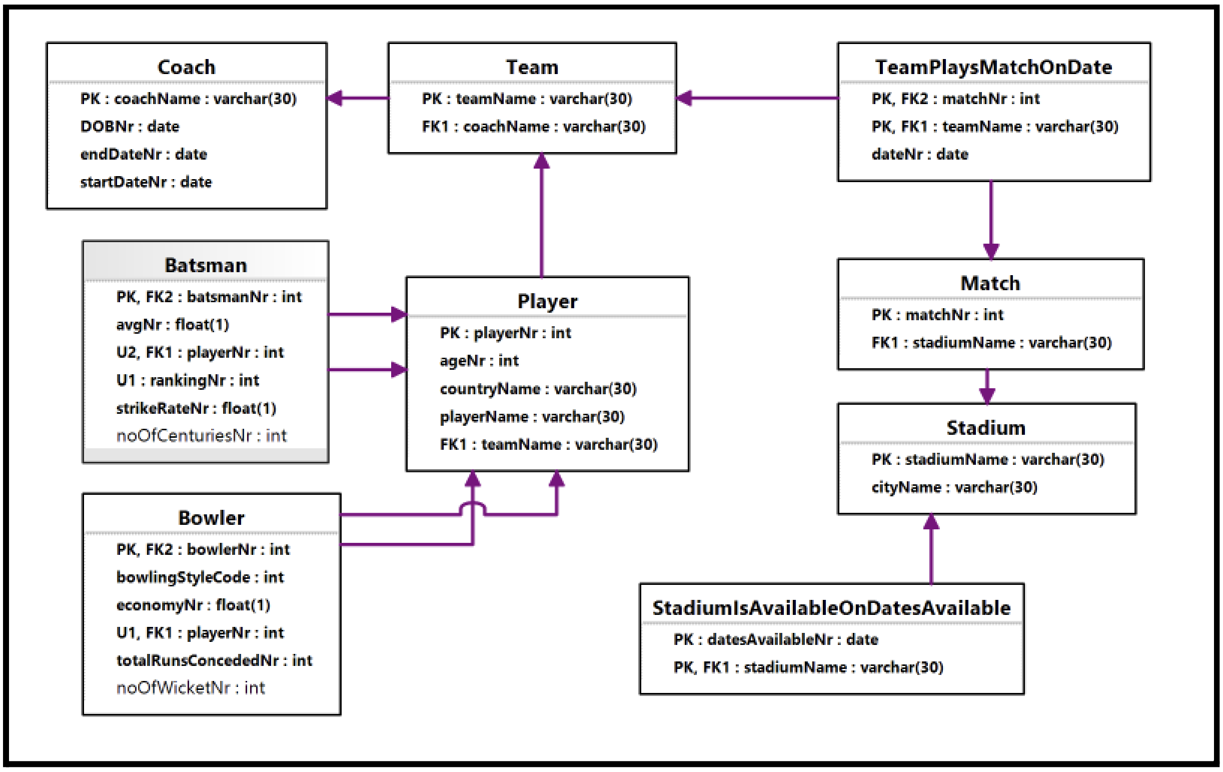
WORKFLOW FLOWCHART



ORM DIAGRAM



RELATIONAL



Coach
PK : coachName : varchar(30)
DOBNr : date
endDateNr : date
startDateNr : date

Team
PK : teamName : varchar(30)
FK1 : coachName : varchar(30)

TeamPlaysMatchOnDate
PK, FK2 : matchNr : int
PK, FK1 : teamName : varchar(30)
dateNr : date

Batsman
PK, FK2 : batsmanNr : int
avgNr : float(1)
U2, FK1 : playerNr : int
U1 : rankingNr : int
strikeRateNr : float(1)
noOfCenturiesNr : int

Player
PK : playerNr : int
ageNr : int
countryName : varchar(30)
playerName : varchar(30)
FK1 : teamName : varchar(30)

Match
PK : matchNr : int
FK1 : stadiumName : varchar(30)

Stadium
PK : stadiumName : varchar(30)
cityName : varchar(30)

Bowler
PK, FK2 : bowlerNr : int
bowlingStyleCode : int
economyNr : float(1)
U1, FK1 : playerNr : int
totalRunsConcededNr : int
noOfWicketNr : int

StadiumIsAvailableOnDatesAvailable
PK : datesAvailableNr : date
PK, FK1 : stadiumName : varchar(30)

SQL CODE

Tables created

I. The team table stores various information such as the team’s name and coach name. Here the team’s name is defined as the primary key.

II. A player table is created which stores the various attributes of the player such as player name, player number, country name, team name, player age and team name. The player number is defined as the primary key in this table.

III. The coach table stores the information regarding the coach’s name, date of birth, start date and end date.

IV. We have a batsman table for storing all batsmen in the cricket management system. The table provides the stats of each batsman which are batsman number, ranking, strike rate and no of centuries. The batsman number would be the primary key. The batsman ranking should be unique.

V. Similar to the batsman table the bowler table would store and display the stats of the bowler. The bowler stats would be bowler number, bowling style, bowling economy, number of wickets and the total number of runs conceded. Here the bowler number would be the primary key. The bowler ranking would be unique.

VI. A stadium table displays information of the stadium in which the matches would be played throughout the tournament. The table shows the stadium name and available dates. Here the stadium name and dates available are set as the primary key.

CODE

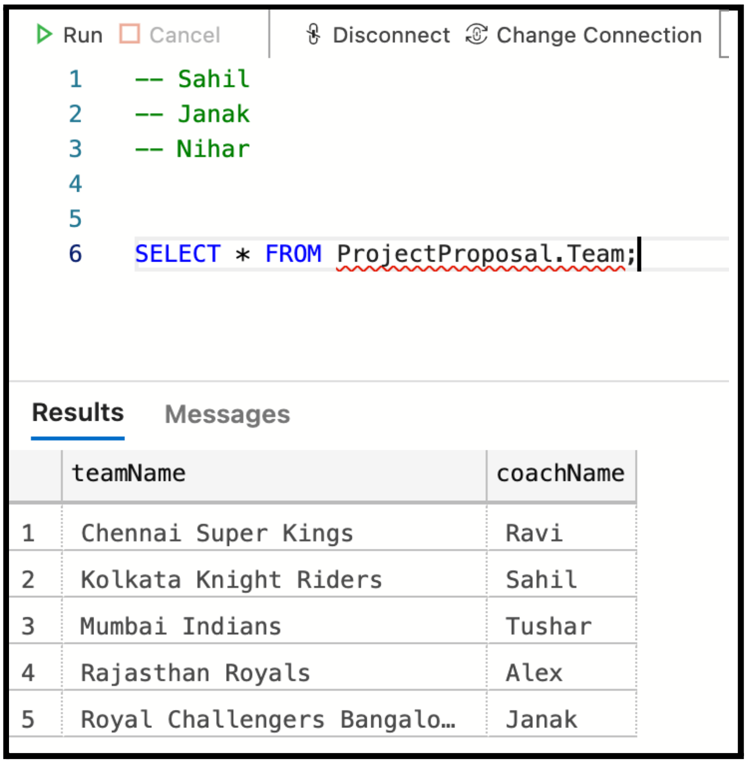
```
CREATE SCHEMA ProjectInitia
GO
GO
CREATE TABLE ProjectInitial.Team
(
  teamName nvarchar(30) NOT NULL,
  coachName nvarchar(30) NOT NULL,
  CONSTRAINT Team_PK PRIMARY KEY(teamName)
)
GO
CREATE TABLE ProjectInitial.Player
(
  playerNr int CHECK (playerNr >= 0) NOT NULL,
  age int CHECK (age >= 0) NOT NULL,
  countryName nvarchar(30) NOT NULL,
  playerName nvarchar(30) NOT NULL,
  teamName nvarchar(30) NOT NULL,
  CONSTRAINT Player_PK PRIMARY KEY(playerNr)
)
GO
CREATE TABLE ProjectInitial.Coach
(
  coachName nvarchar(30) NOT NULL,
  DOB date NOT NULL,
  startDate date NOT NULL,
  endDate date NOT NULL,
  CONSTRAINT Coach_PK PRIMARY KEY(coachName)
)
GO
CREATE TABLE ProjectInitial.Batsman
(
  batsmanNr int CHECK (batsmanNr >= 0) NOT NULL,
  avg float(1) NOT NULL,
  playerNr int CHECK (playerNr >= 0) NOT NULL,
  ranking int CHECK (ranking >= 0) NOT NULL,
  strikeRate float(1) NOT NULL,
  noOfCenturies int CHECK (noOfCenturies >= 0),
  CONSTRAINT Batsman_UC1 UNIQUE(ranking),
  CONSTRAINT Batsman_UC2 UNIQUE(playerNr),
  CONSTRAINT Batsman_PK PRIMARY KEY(batsmanNr)
)
GO
CREATE TABLE ProjectInitial.Bowler
(
  bowlerNr int CHECK (bowlerNr >= 0) NOT NULL,
  bowlingStyle varchar(30) NOT NULL,
  economy float(1) NOT NULL,
  playerNr int CHECK (playerNr >= 0) NOT NULL,
totalRunsConceded int CHECK (totalRunsConceded >= 0) NOT NULL,
  noOfWicket int CHECK (noOfWicket >= 0),
  CONSTRAINT Bowler_UC UNIQUE(playerNr),
  CONSTRAINT Bowler_PK PRIMARY KEY(bowlerNr)
)
GO
CREATE TABLE ProjectInitial."Match"
(
  matchNr int CHECK (matchNr >= 0) NOT NULL,
  stadiumName nvarchar(30) NOT NULL,
  CONSTRAINT Match_PK PRIMARY KEY(matchNr)

)
GO
CREATE TABLE ProjectInitial.TeamPlaysMatchOnDate
(
  matchNr int CHECK (matchNr >= 0) NOT NULL,
  teamName nvarchar(30) NOT NULL,
  date date NOT NULL,
CONSTRAINT TeamPlaysMatchOnDate_PK PRIMARY KEY(teamName, matchNr)
)
GO
CREATE TABLE ProjectInitial.Stadium
(
  stadiumName nvarchar(30) NOT NULL,
  cityName nvarchar(30) NOT NULL,
  CONSTRAINT Stadium_PK PRIMARY KEY(stadiumName)
)
GO
CREATE TABLE ProjectInitial.StadiumIsAvailableOnDatesAvailable
(
  stadiumName nvarchar(30) NOT NULL,
  datesAvailable date NOT NULL,
CONSTRAINT StadiumIsAvailableOnDatesAvailable_PK PRIMARY KEY(stadiumName, datesAvailable)
)
GO
ALTER TABLE ProjectInitial.Team ADD CONSTRAINT Team_FK FOREIGN KEY (coachName) REFERENCES ProjectInitial.Coach (coachName) ON DELETE NO ACTION ON UPDATE NO ACTION
GO
ALTER TABLE ProjectInitial.Player ADD CONSTRAINT Player_FK FOREIGN KEY (teamName) REFERENCES ProjectInitial.Team (teamName) ON DELETE NO ACTION ON UPDATE NO ACTION



GO
ALTER TABLE ProjectInitial.Batsman ADD CONSTRAINT Batsman_FK1 FOREIGN KEY (playerNr) REFERENCES ProjectInitial.Player (playerNr) ON DELETE NO ACTION ON UPDATE NO ACTION
GO
ALTER TABLE ProjectInitial.Batsman ADD CONSTRAINT Batsman_FK2 FOREIGN KEY (batsmanNr) REFERENCES ProjectInitial.Player (playerNr) ON DELETE NO ACTION ON UPDATE NO ACTION
GO
```

```
ALTER TABLE ProjectInitial.Bowler ADD CONSTRAINT Bowler_FK1 FOREIGN KEY (playerNr)
REFERENCES ProjectInitial.Player (playerNr) ON DELETE NO ACTION ON UPDATE NO ACTION

GO

ALTER TABLE ProjectInitial.Bowler ADD CONSTRAINT Bowler_FK2 FOREIGN KEY (bowlerNr)
REFERENCES ProjectInitial.Player (playerNr) ON DELETE NO ACTION ON UPDATE NO ACTION

GO

ALTER TABLE ProjectInitial."Match" ADD CONSTRAINT Match_FK FOREIGN KEY
(stadiumName) REFERENCES ProjectInitial.Stadium (stadiumName) ON DELETE NO ACTION
ON UPDATE NO ACTION

GO

ALTER TABLE ProjectInitial.TeamPlaysMatchOnDate ADD CONSTRAINT
TeamPlaysMatchOnDate_FK1 FOREIGN KEY (teamName) REFERENCES ProjectInitial.Team
(teamName) ON DELETE NO ACTION ON UPDATE NO ACTION

GO

ALTER TABLE ProjectInitial.TeamPlaysMatchOnDate ADD CONSTRAINT
TeamPlaysMatchOnDate_FK2 FOREIGN KEY (matchNr) REFERENCES ProjectInitial."Match"
(matchNr) ON DELETE NO ACTION ON UPDATE NO ACTION

GO

ALTER TABLE ProjectInitial.StadiumIsAvailableOnDatesAvailable ADD CONSTRAINT
StadiumIsAvailableOnDatesAvailable_FK FOREIGN KEY (stadiumName) REFERENCES
ProjectInitial.Stadium (stadiumName) ON DELETE NO ACTION ON UPDATE NO ACTION

GO

GO
```

INSERTING VALUES INTO THE TABLES

```
insert into ProjectInitial.Coach

values

('tushar','1987-02-18','2015-11-09','2021-05-08'),

('ravi','1963-02-19','2012-10-09','2022-05-08'),

('alex','1972-02-16','2001-06-09','2021-05-08'),

('sahil','1984-02-15','2005-02-09','2023-05-08'),

('janak','1967-02-12','2017-12-09','2023-05-08'),

('joe','1963-04-01','2019-12-09','2022-05-08'),

('charlie','1966-02-10','2020-12-09','2021-05-08'),

('simon','1962-02-23','2007-12-09','2022-05-08'),

('dunken','1977-02-28','2018-12-09','2023-04-08'),
('chappel','1987-02-11','2017-12-09','2022-05-08');

insert into ProjectInitial.Team
values
('mumbai indians','tushar'),
('chennai super kings','ravi'),
('rajasthan royals','alex'),
('kolkata knight riders','sahil'),
('royal challengers bangalore','janak'),
('punjab kings','joe'),
('delhi capitals','charlie'),
('pune warriors','simon'),
('new york 11','dunken'),
('phoenix lions','chappel');

insert into ProjectInitial.Player
values
('10','23','India','Sachin','mumbai indians'),
('7','25','India','Dhoni','chennai super kings'),
('5','32','England','Butler','rajasthan royals'),
('16','34','West Indies','Narine','kolkata knight riders'),
('8','34','South Africa','AB De Villiers','royal challengers
bangalore'),
('89','27','India','Zaheer','mumbai indians'),
('34','28','West Indies','Bravo','chennai super kings'),
('19','29','England','Jofra Archer','rajasthan royals'),
('99','31','West Indies','Russel','kolkata knight riders'),
('56','34','Australia','Glenn Maxwell','royal challengers
bangalore'),
('11','24','Pakistan','ashwin','punjab kings'),
('71','26','Afghanistan','kl rahul','punjab kings'),

('55','33','England','rishab','delhi capitals'),

('66','35','West Indies','rabada','delhi capitals'),

('3','35','South Africa','ganguly','pune warriors'),

('88','28','Bangladesh','chahal','pune warriors'),

('35','29','Spain','gilchrist','new york 11'),

('18','30','Italy','mitchell','new york 11'),

('98','32','Russia','dale','phoenix lions'),

('53','35','Portugal','finch','phoenix lions');


insert into ProjectInitial.Bowler

values

('16','SPIN','8.3','16','280','20'),

('89','FAST','3.5','89','100','39'),

('34','FAST','5.2','34','132','36'),

('19','FAST','4.0','19','130','24'),

('56','SPIN','3.6','56','136','28'),

('11','SPIN','7.4','11','380','30'),

('66','FAST','3.6','66','200','29'),

('88','SPIN','5.4','88','142','16'),

('18','FAST','3.0','18','160','14'),

('98','FAST','2.6','98','236','48');


insert into ProjectInitial.Batsman

values

('10','50.3','10','1','188.2','100'),

('7','43.2','7','3','198.2','54'),

('5','40.8','5','4','138.2','36'),

('8','54.9','8','2','200.2','64'),

('99','42.5','99','10','168.2','21'),
('71','43.3','71','7','288.2','110'),
('55','49.2','55','11','98.2','14'),
('3','23.8','3','5','38.2','86'),
('35','64.9','35','8','170.2','4'),
('53','32.5','53','9','160.2','34');

insert into ProjectInitial.Stadium
values
('Eden Gardens','Kolkata'),
('Wankhede','Mumbai'),
('Narendra Modi Stadium','Ahmedabad'),
('DY Patil','Navi Mumbai'),
('Chinnaswamy','Chennai'),
('Chidambaram','Bangalore');

insert into ProjectInitial.Match
values
('20','Eden Gardens'),
('25','Wankhede'),
('28','Narendra Modi Stadium'),
('29','DY Patil'),
('30','Chinnaswamy'),
('34','Chidambaram ');

insert into ProjectInitial.TeamPlaysMatchOnDate
values
('20','Mumbai Indians','2021-03-08'),
('25','Chennai Super Kings','2021-03-09'),
('28','Rajasthan Royals','2021-03-07'),
('29','Kolkata Knight Riders','2021-03-10'),
('30','Royal Challengers Bangalore','2021-03-08'),
('34','Mumbai Indians','2021-03-28');

insert into ProjectInitial.StadiumIsAvailableOnDatesAvailable
values
('Eden Gardens','2021-04-08'),
('Eden Gardens','2021-03-28'),
('Eden Gardens','2021-03-18'),
('Wankhede','2021-04-23'),
('Wankhede','2021-05-01'),
('Wankhede','2021-04-01'),
('Wankhede','2021-04-28'),
('Wankhede','2021-03-31'),
('Narendra Modi Stadium','2021-03-23'),
('Narendra Modi Stadium','2021-03-18'),
('Chinnaswamy','2021-04-22'),
('Chinnaswamy','2021-04-15'),
('Chinnaswamy','2021-03-18'),
('DY Patil','2021-04-17'),
('DY Patil','2021-04-23'),
('Chidambaram','2021-03-07'),
('Chidambaram','2021-04-07'),
('Chidambaram','2021-04-27');
```

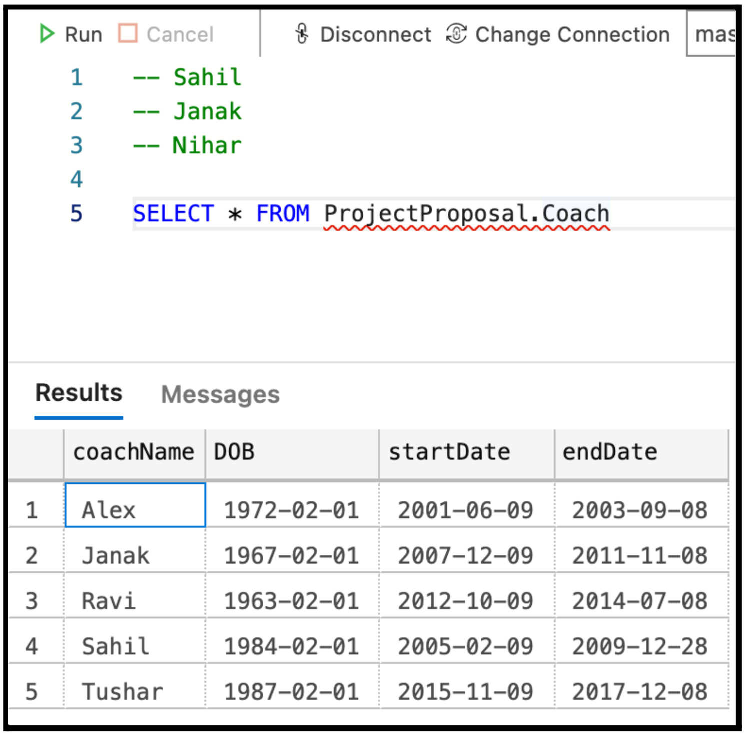
SCREENSHOTS OF OUTPUT



```sql
-- Sahil
-- Janak
-- Nihar


SELECT * FROM ProjectProposal.Team;
```

	teamName	coachName
1	Chennai Super Kings	Ravi
2	Kolkata Knight Riders	Sahil
3	Mumbai Indians	Tushar
4	Rajasthan Royals	Alex
5	Royal Challengers Bangalo…	Janak

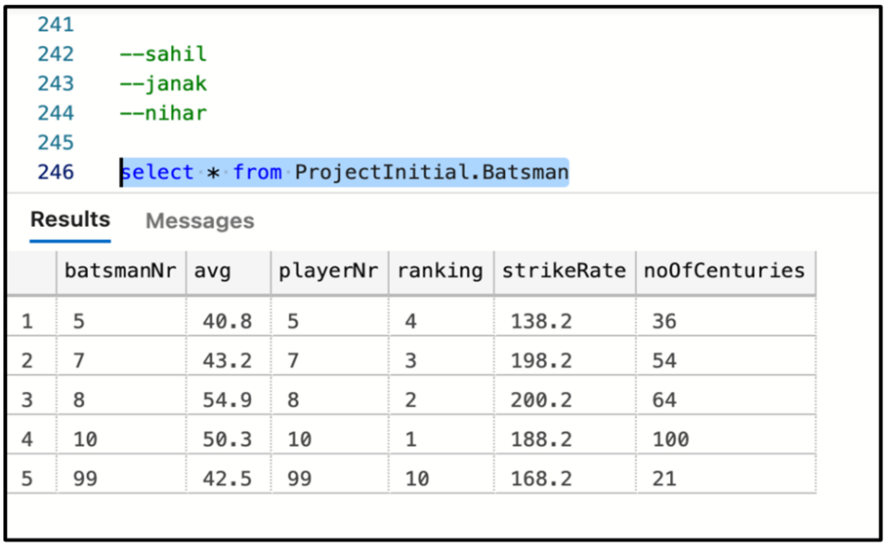
Values in Table Team



```sql
-- Sahil
-- Janak
-- Nihar

SELECT * FROM ProjectProposal.Coach
```

	coachName	DOB	startDate	endDate
1	Alex	1972-02-01	2001-06-09	2003-09-08
2	Janak	1967-02-01	2007-12-09	2011-11-08
3	Ravi	1963-02-01	2012-10-09	2014-07-08
4	Sahil	1984-02-01	2005-02-09	2009-12-28
5	Tushar	1987-02-01	2015-11-09	2017-12-08

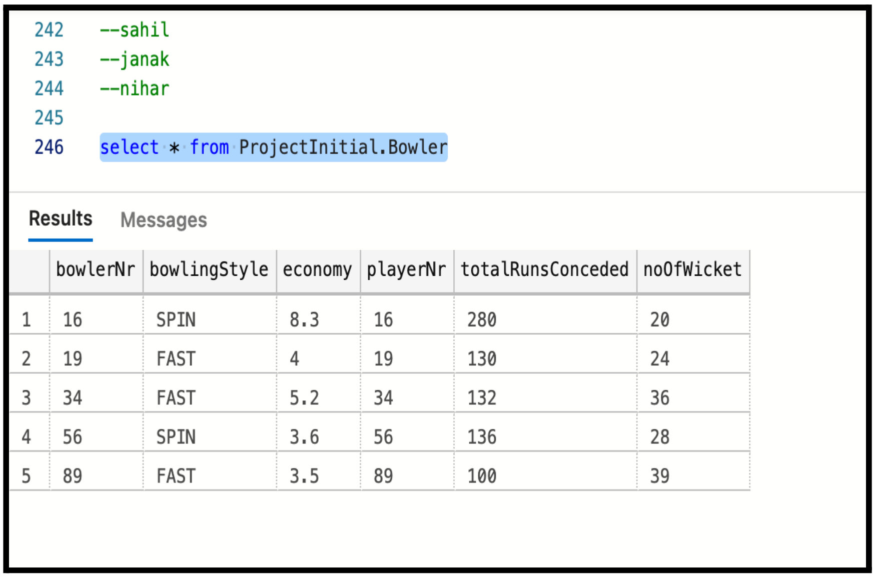
Values in Table Coach



```
--sahil
--janak
--nihar

select * from ProjectInitial.Batsman
```

	batsmanNr	avg	playerNr	ranking	strikeRate	noOfCenturies
1	5	40.8	5	4	138.2	36
2	7	43.2	7	3	198.2	54
3	8	54.9	8	2	200.2	64
4	10	50.3	10	1	188.2	100
5	99	42.5	99	10	168.2	21

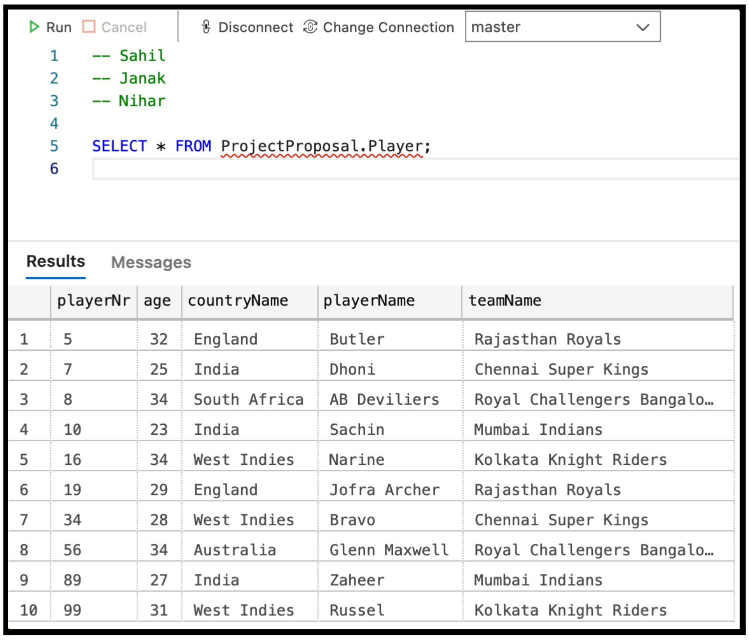
Values in Table Batsman



```
--sahil
--janak
--nihar

select * from ProjectInitial.Bowler
```

	bowlerNr	bowlingStyle	economy	playerNr	totalRunsConceded	noOfWicket
1	16	SPIN	8.3	16	280	20
2	19	FAST	4	19	130	24
3	34	FAST	5.2	34	132	36
4	56	SPIN	3.6	56	136	28
5	89	FAST	3.5	89	100	39

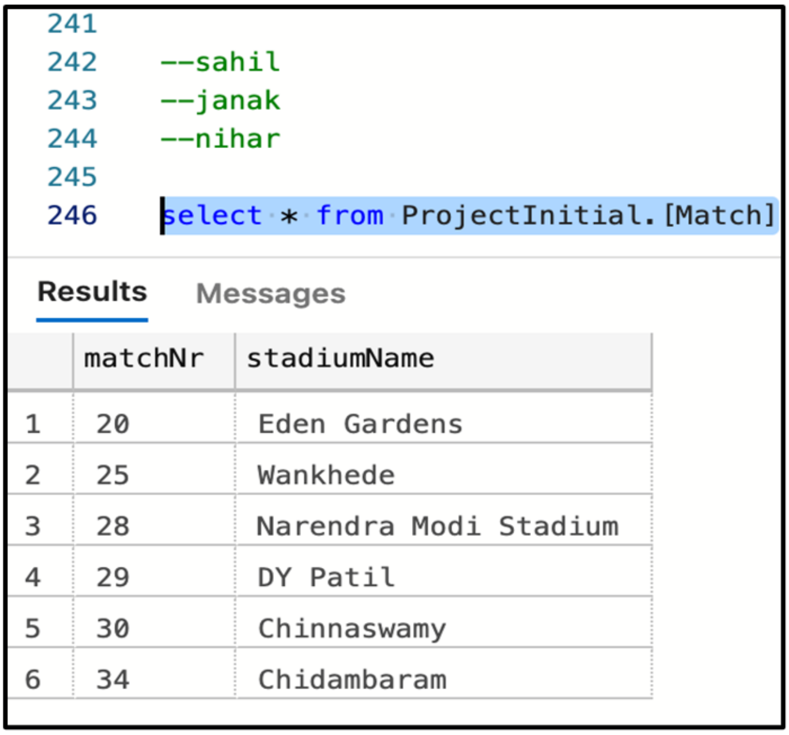
Values in Table Bowler



```sql
-- Sahil
-- Janak
-- Nihar

SELECT * FROM ProjectProposal.Player;

```

	playerNr	age	countryName	playerName	teamName
1	5	32	England	Butler	Rajasthan Royals
2	7	25	India	Dhoni	Chennai Super Kings
3	8	34	South Africa	AB Deviliers	Royal Challengers Bangalo…
4	10	23	India	Sachin	Mumbai Indians
5	16	34	West Indies	Narine	Kolkata Knight Riders
6	19	29	England	Jofra Archer	Rajasthan Royals
7	34	28	West Indies	Bravo	Chennai Super Kings
8	56	34	Australia	Glenn Maxwell	Royal Challengers Bangalo…
9	89	27	India	Zaheer	Mumbai Indians
10	99	31	West Indies	Russel	Kolkata Knight Riders

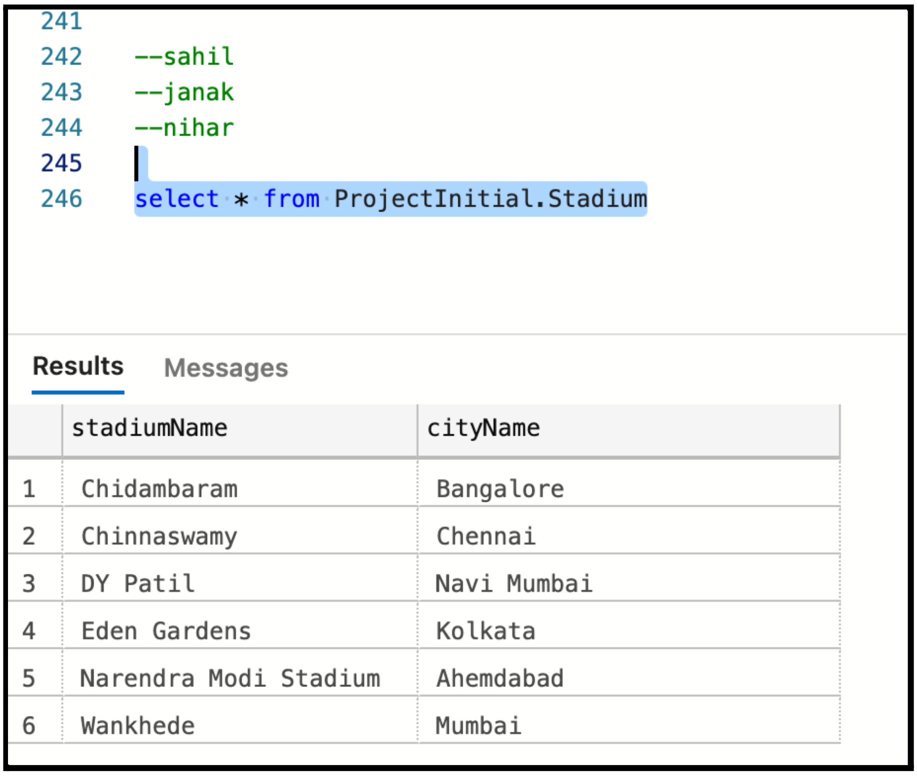
Values in Table Player



```sql

--sahil
--janak
--nihar

select * from ProjectInitial.[Match]
```

	matchNr	stadiumName
1	20	Eden Gardens
2	25	Wankhede
3	28	Narendra Modi Stadium
4	29	DY Patil
5	30	Chinnaswamy
6	34	Chidambaram

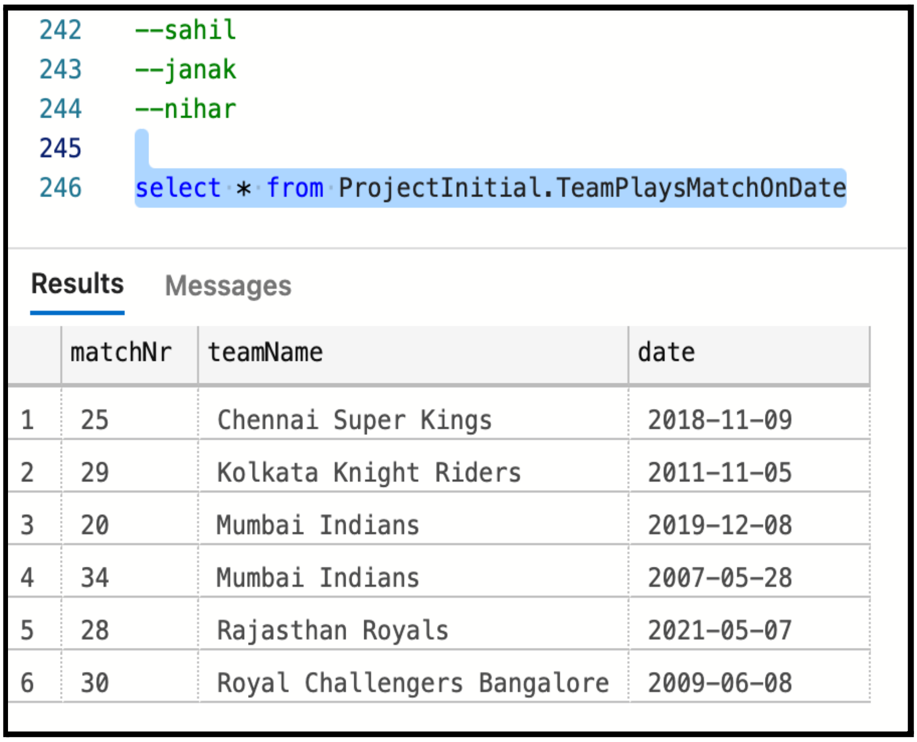
Values in Table Match



```

--sahil
--janak
--nihar

select * from ProjectInitial.Stadium
```

	stadiumName	cityName
1	Chidambaram	Bangalore
2	Chinnaswamy	Chennai
3	DY Patil	Navi Mumbai
4	Eden Gardens	Kolkata
5	Narendra Modi Stadium	Ahemdabad
6	Wankhede	Mumbai

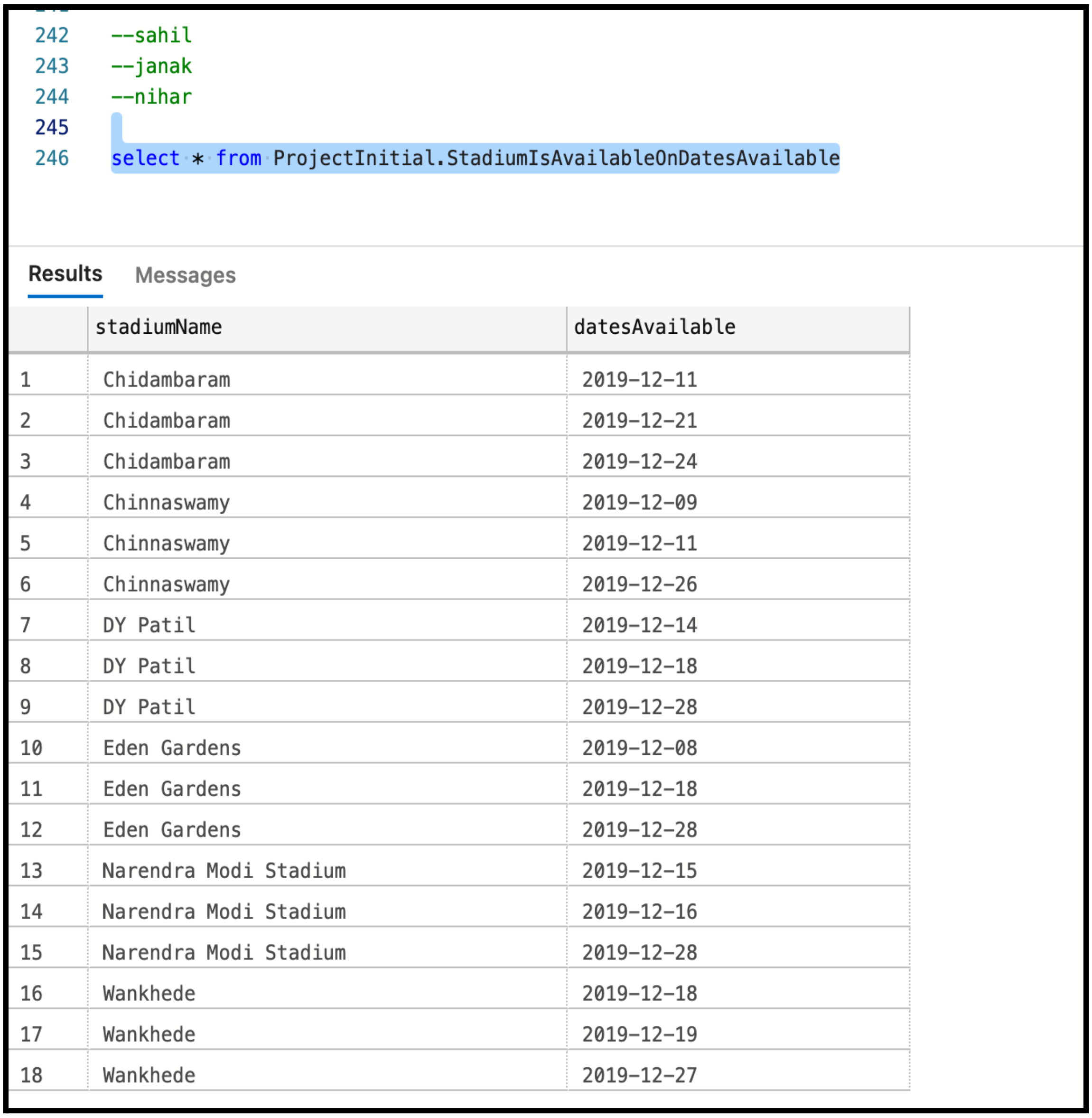
Values in Table Stadium



```
--sahil
--janak
--nihar

select * from ProjectInitial.TeamPlaysMatchOnDate
```

	matchNr	teamName	date
1	25	Chennai Super Kings	2018-11-09
2	29	Kolkata Knight Riders	2011-11-05
3	20	Mumbai Indians	2019-12-08
4	34	Mumbai Indians	2007-05-28
5	28	Rajasthan Royals	2021-05-07
6	30	Royal Challengers Bangalore	2009-06-08

Values in Table TeamPlaysMatchOnDate



```sql
--sahil
--janak
--nihar

select * from ProjectInitial.StadiumIsAvailableOnDatesAvailable
```

	stadiumName	datesAvailable
1	Chidambaram	2019-12-11
2	Chidambaram	2019-12-21
3	Chidambaram	2019-12-24
4	Chinnaswamy	2019-12-09
5	Chinnaswamy	2019-12-11
6	Chinnaswamy	2019-12-26
7	DY Patil	2019-12-14
8	DY Patil	2019-12-18
9	DY Patil	2019-12-28
10	Eden Gardens	2019-12-08
11	Eden Gardens	2019-12-18
12	Eden Gardens	2019-12-28
13	Narendra Modi Stadium	2019-12-15
14	Narendra Modi Stadium	2019-12-16
15	Narendra Modi Stadium	2019-12-28
16	Wankhede	2019-12-18
17	Wankhede	2019-12-19
18	Wankhede	2019-12-27

Values in StadiumIsAvailableOnDatesAvailable

UDF

1) We have created a user defined function in order to determine the bowler of the tournament by picking the bowler who has taken the maximum number of wickets in the tournament.

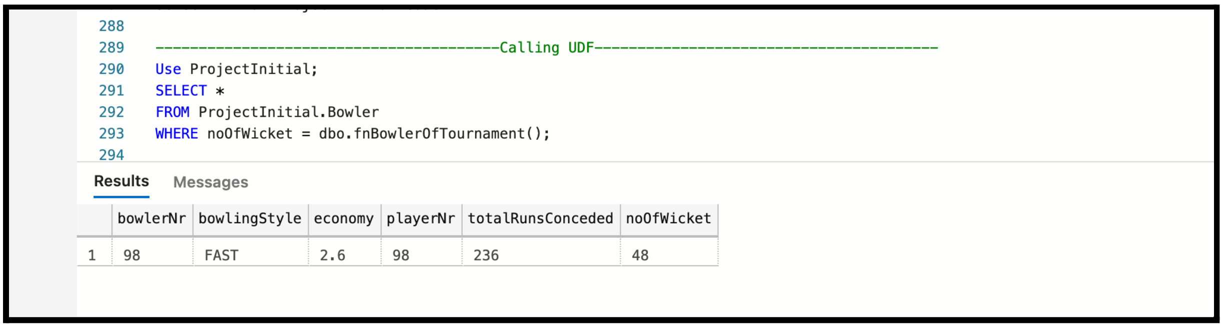
```
----------------------------------UDF 1 Initialization-------------------------

USE ProjectInitial;

GO

IF OBJECT_ID('dbo.fnBowlerOfTournament') IS NOT NULL

DROP FUNCTION fnBowlerOfTournament

GO

CREATE FUNCTION fnBowlerOfTournament()

RETURNS int

BEGIN

   RETURN

   (SELECT MAX(noOfWicket) as MaxWickets

   FROM ProjectInitial.Bowler

   )

END;
```

2) We have created a second user defined function to determine the batsman of the tournament by picking the batsman who has scored the highest number of runs in the tournament.

```
---------------------------------UDF 2 Initialization-----------------------

USE ProjectInitial;

GO

IF OBJECT_ID('dbo.fnBatsmanOfTournament') IS NOT NULL

DROP FUNCTION fnBatsmanOfTournament

GO

CREATE FUNCTION fnBatsmanOfTournament()

RETURNS int

BEGIN

   RETURN

   (SELECT MAX(noOfCenturies) as MaxAvg

   FROM ProjectInitial.Batsman

   )

END;
```

3) The third user defined function helps in checking whether a team needs a new coach. The end date is retrieved from the coach table in order to check the tenure of a coach with the team.

```
-----------------------------UDF 3 Initialization------------------------

USE ProjectInitial;

GO

IF OBJECT_ID('dbo.fnTeamNeedsNewCoach') IS NOT NULL

DROP FUNCTION fnTeamNeedsNewCoach

GO

CREATE FUNCTION fnTeamNeedsNewCoach()

RETURNS date

BEGIN

 RETURN

   (SELECT min(endDate) as minDate

   FROM ProjectInitial.coach

   )

END;


--------------------------------Calling UDF 1----------------------------

Use ProjectInitial;

SELECT *

FROM ProjectInitial.Bowler

WHERE noOfWicket = dbo.fnBowlerOfTournament();
```

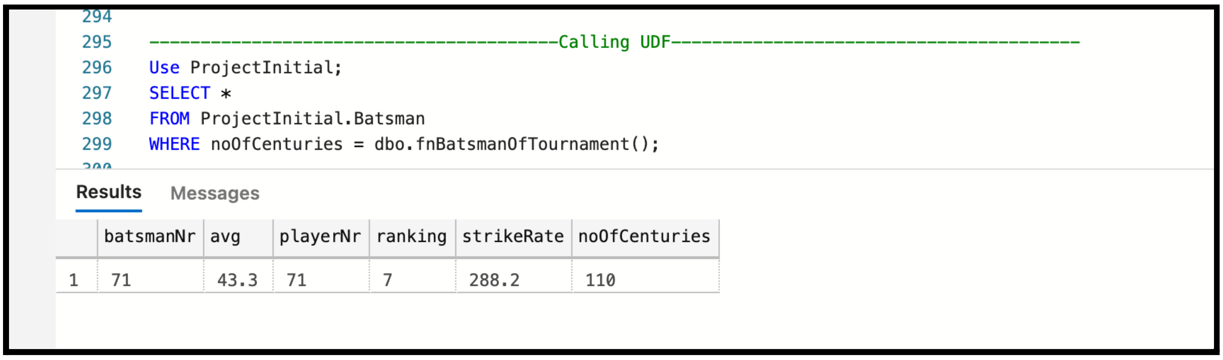


```
----------------------------------------Calling UDF----------------------------------------
Use ProjectInitial;
SELECT *
FROM ProjectInitial.Bowler
WHERE noOfWicket = dbo.fnBowlerOfTournament();
```

	bowlerNr	bowlingStyle	economy	playerNr	totalRunsConceded	noOfWicket
1	98	FAST	2.6	98	236	48

UDF 1 (Bowler of Tournament) Output

```
--------------------------------Calling UDF 2----------------------------

Use ProjectInitial;

SELECT *

FROM ProjectInitial.Batsman

WHERE noOfCenturies = dbo.fnBatsmanOfTournament();
```

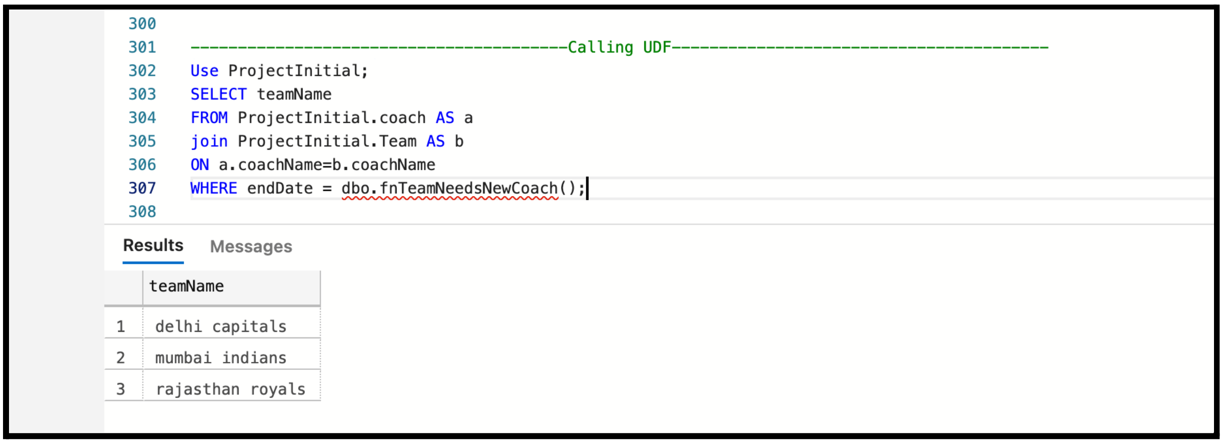


```
---------------------------------------Calling UDF---------------------------------------
Use ProjectInitial;
SELECT *
FROM ProjectInitial.Batsman
WHERE noOfCenturies = dbo.fnBatsmanOfTournament();
```

	batsmanNr	avg	playerNr	ranking	strikeRate	noOfCenturies
1	71	43.3	71	7	288.2	110

UDF 2 (Batsman of Tournament) Output

```
--------------------------------Calling UDF 3----------------------------

Use ProjectInitial;

SELECT teamName

FROM ProjectInitial.coach AS a

join ProjectInitial.Team AS b

ON a.coachName=b.coachName

WHERE endDate = dbo.fnTeamNeedsNewCoach();
```



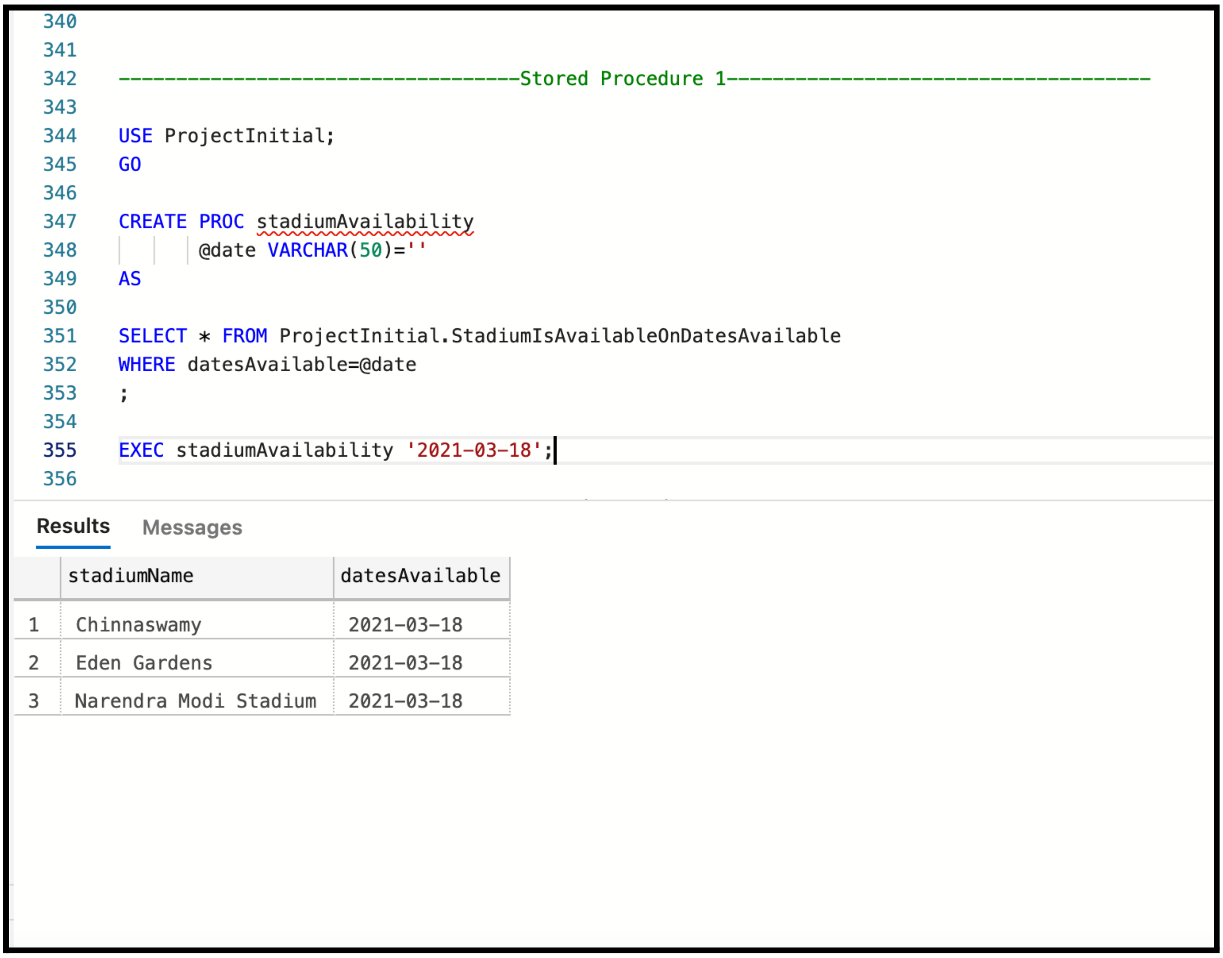
```
---------------------------------------Calling UDF---------------------------------------
Use ProjectInitial;
SELECT teamName
FROM ProjectInitial.coach AS a
join ProjectInitial.Team AS b
ON a.coachName=b.coachName
WHERE endDate = dbo.fnTeamNeedsNewCoach();
```

	teamName
1	delhi capitals
2	mumbai indians
3	rajasthan royals

UDF 3 (Team needs new coach) Output

STORED PROCEDURE

1. We have a stored procedure for checking the availability of the stadium. We check the available dates column in the stadium table. The available stadium names are displayed.

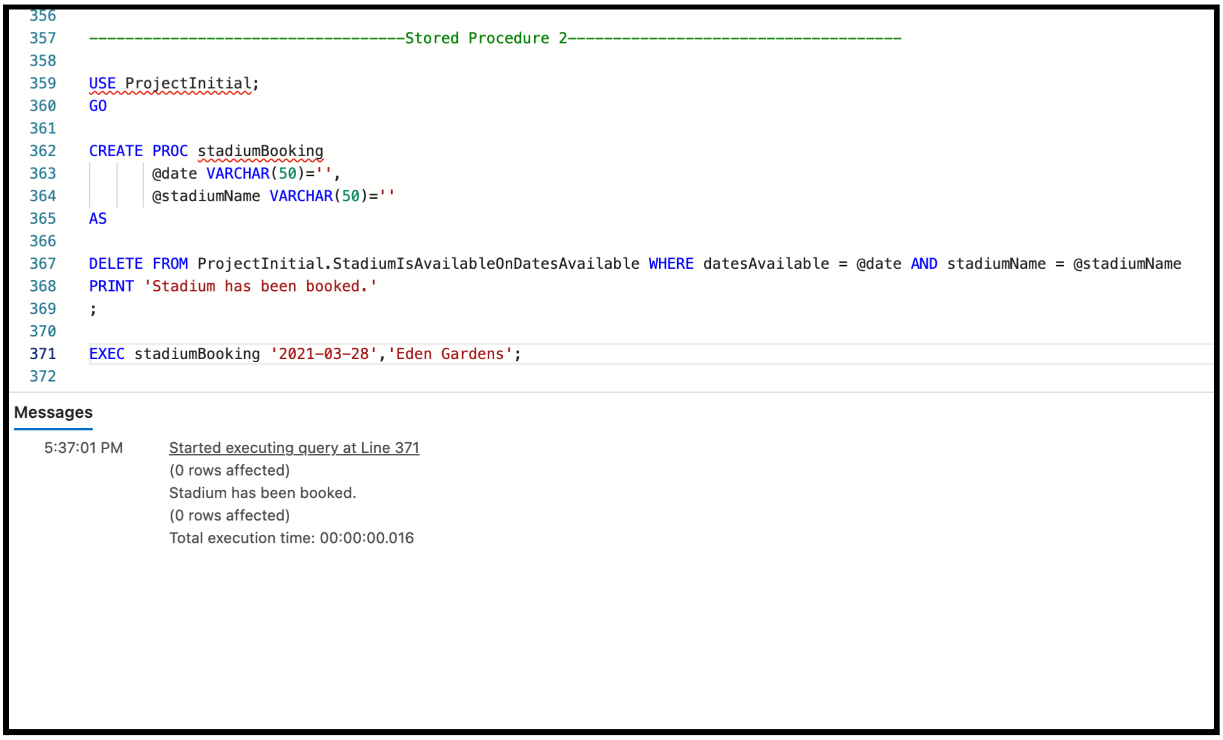


```
-----------------------------------Stored Procedure 1-------------------------------------

USE ProjectInitial;
GO

CREATE PROC stadiumAvailability
            @date VARCHAR(50)=''
AS

SELECT * FROM ProjectInitial.StadiumIsAvailableOnDatesAvailable
WHERE datesAvailable=@date
;

EXEC stadiumAvailability '2021-03-18';

```

	stadiumName	datesAvailable
1	Chinnaswamy	2021-03-18
2	Eden Gardens	2021-03-18
3	Narendra Modi Stadium	2021-03-18

Stored Procedure 1 (Stadium Availability)

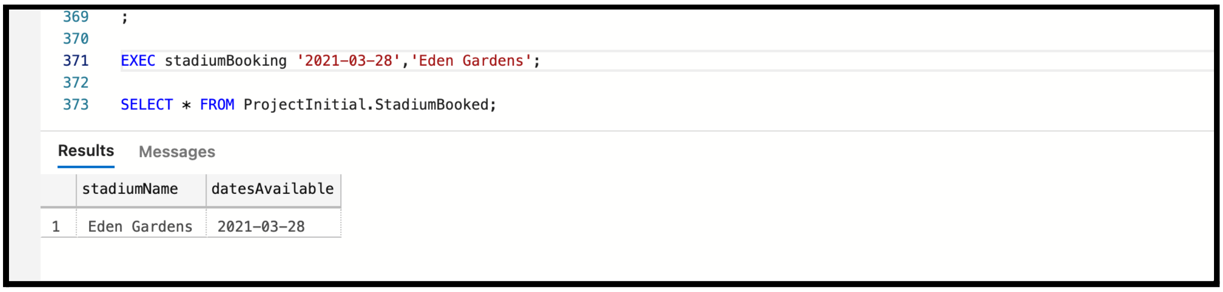
2. We have a second stored procedure in order to book the available stadium. When there is a stadium available on the required date, the system automatically books the stadium for the user. The stadium table is updated once a stadium has been booked.



```sql
------------------------------------Stored Procedure 2-------------------------------------

USE ProjectInitial;
GO

CREATE PROC stadiumBooking
        @date VARCHAR(50)='',
        @stadiumName VARCHAR(50)=''
AS

DELETE FROM ProjectInitial.StadiumIsAvailableOnDatesAvailable WHERE datesAvailable = @date AND stadiumName = @stadiumName
PRINT 'Stadium has been booked.'
;

EXEC stadiumBooking '2021-03-28','Eden Gardens';
```

Messages

```
5:37:01 PM     Started executing query at Line 371
               (0 rows affected)
               Stadium has been booked.
               (0 rows affected)
               Total execution time: 00:00:00.016
```



```sql
;

EXEC stadiumBooking '2021-03-28','Eden Gardens';

SELECT * FROM ProjectInitial.StadiumBooked;
```

Results   Messages

	stadiumName	datesAvailable
1	Eden Gardens	2021-03-28

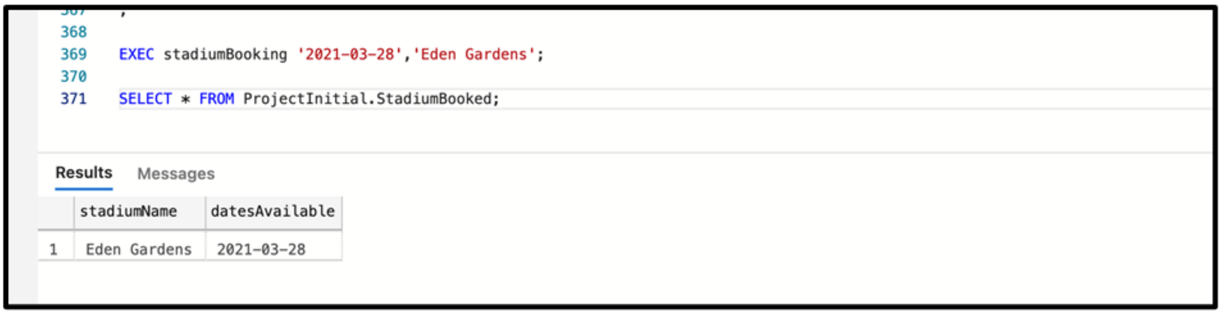
Stored Procedure 2 (Stadium Booking)

TRIGGER

We have created a trigger on delete. When we run the stored procedure to book a stadium a trigger is fired on delete and the data is added to the availability table.

```sql
----------------------------------------Trigger-----------------------------

IF EXISTS (SELECT DB_ID('StadiumBookings'))
DROP TRIGGER ProjectInitial.StadiumBookings;
GO
CREATE TRIGGER StadiumBookings ON ProjectInitial.StadiumIsAvailableOnDatesAvailable
AFTER DELETE
AS
INSERT INTO StadiumBooked
(stadiumName, datesAvailable)
SELECT stadiumName, datesAvailable FROM Deleted
PRINT 'Stadium has been booked.'


------------------------Query for Trigger Table---------------------------

DROP TABLE ProjectInitial.StadiumBooked;
CREATE TABLE ProjectInitial.StadiumBooked
(
   stadiumName nvarchar(30) NOT NULL,
   datesAvailable date NOT NULL,
)
GO
```

OUTPUT



```sql
;

EXEC stadiumBooking '2021-03-28','Eden Gardens';

SELECT * FROM ProjectInitial.StadiumBooked;
```

	stadiumName	datesAvailable
1	Eden Gardens	2021-03-28

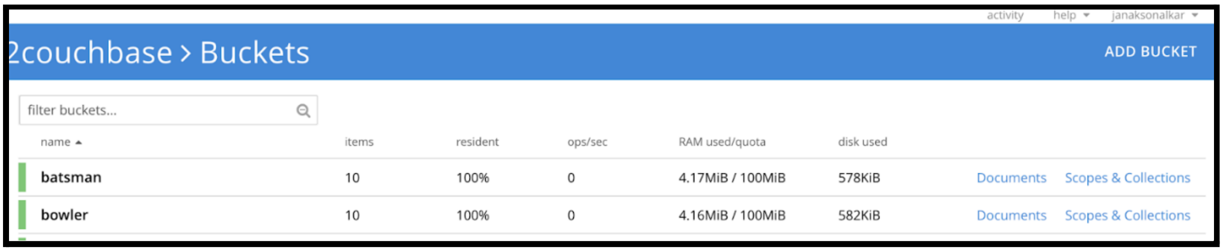
Stadium Booked Table Updated (Using Trigger Execution)

NoSQL

We are creating two buckets:

1) Batsman - The batsman bucket would store information such as batman name, batsman number, age, team name and country. The batsman stat would be nested in the batsman JSON.
2) Bowler - The bowler bucket would store information such as bowler name, bowler number, age, team name and country. The bowler stat would be nested in the bowler JSON.

Creating Two buckets:



name ▲	items	resident	ops/sec	RAM used/quota	disk used		
batsman	10	100%	0	4.17MiB / 100MiB	578KiB	Documents	Scopes & Collections
bowler	10	100%	0	4.16MiB / 100MiB	582KiB	Documents	Scopes & Collections

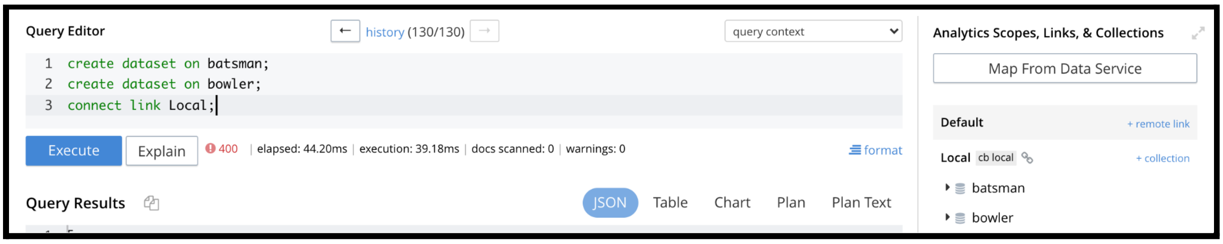
Bucket 1 Insert Query: Bowler (Nested Bowler Statistics)

```
insert into bowler (key, value)
values ( "1001",
  { "player_name": "Rabada",
    "jersey_no": 32,
    "country": "West Indies",
    "team": "delhi capitals",
    "player_stats": [ { "economy": 6.8,
                        "bowlingstyle": "Fast",
                        "numberOfWickets": 6,
                        "totalRunsConceded": 102
                      }
                    ]
  }

), ( "1002",

  { "player_name": "Mitchell",
    "jersey_no": 100,
    "country": "Italy",
    "team": "new york 11",
    "player_stats": [ { "economy": 4,
                        "bowlingstyle": "Spin",
                        "numberOfWickets": 36,
                        "totalRunsConceded": 345
                      }
                    ]
  }


), ( "1003",

  { "player_name": "Ashwin",
    "jersey_no": 161,
    "country": "India",
    "team": "royal challengers bangalore",
    "player_stats": [ { "economy": 9.11,
                        "bowlingstyle": "Pacer",
                        "numberOfWickets": 34,
                        "totalRunsConceded": 45
                      }
                    ]
  }

), ( "1004",

  { "player_name": "Butler",
    "jersey_no": 63,
    "country": "England",
    "team": "rajasthan royals",
    "player_stats": [ { "economy": 4.56,
                        "bowlingstyle": "Fast",
                        "numberOfWickets": 1,
                        "totalRunsConceded": 128
                      }
                    ]
  }

), ( "1005",

  { "player_name": "Russel",
    "jersey_no": 42,
    "country": "Australia",
    "team": "kolkata knight riders",
    "player_stats": [ { "economy": 3,
                        "bowlingstyle": "Spin",
                        "numberOfWickets": 12,
                        "totalRunsConceded": 48
                      }
                    ]
  }

), ( "1006",

  { "player_name": "Narine",
    "jersey_no": 81,
    "country": "Afghanistan",
    "team": "punjab kings",
    "player_stats": [ { "economy": 23,
                        "bowlingstyle": "Pacer",
                        "numberOfWickets": 64,
                        "totalRunsConceded": 87
                      }
                    ]
  }

), ( "1007",

  { "player_name": "Yuzi",
    "jersey_no": 91,
    "country": "South Africa",
    "team": "pune warriors",
    "player_stats": [ { "economy": 12.5,
                        "bowlingstyle": "Spin",
                        "numberOfWickets": 21,
                        "totalRunsConceded": 819
                      }
                    ]
  }

), ( "1008",

  { "player_name": "Harbhajan",
    "jersey_no": 88,
    "country": "South Africa",
    "team": "royal challengers bangalore",
    "player_stats": [ { "economy": 21,
                        "bowlingstyle": "Fast",
                        "numberOfWickets": 54,
                        "totalRunsConceded": 169
                      }
                    ]
  }

), ( "1009",

  { "player_name": "Kaif",
    "jersey_no": 666,
    "country": "India",
    "team": "delhi capital",
    "player_stats": [ { "economy": 4.67,
                        "bowlingstyle": "Spin",
                        "numberOfWickets": 13,
                        "totalRunsConceded": 230
                      }
                    ]
  }

), ( "1010",

  { "player_name": "Lee",
    "jersey_no": 69,
    "country": "Australia",
    "team": "kings punjab",
    "player_stats": [ { "economy": 8.9,
                        "bowlingstyle": "Fast",
                        "numberOfWickets": 77,
                        "totalRunsConceded": 408
                      }
                    ]
  }
);
```

Bucket 2 Insert Query: Batsman (Nested Batsman Statistics)

```
insert into batsman (key, value)

values ( "1001",

  { "player_name": "Dhoni",
    "jersey_no": 7,
    "country": "India",
    "team": "chennai super kings",
    "player_stats": [ { "strike_rate": 135,
                        "ranking": 3,
                        "numberOfCenturies": 62,
                        "average": 87.98
                      }
                    ]
  }

), ( "1002",

  { "player_name": "Sachin",
    "jersey_no": 10,
    "country": "India",
    "team": "mumbai indians",
    "player_stats": [ { "strike_rate": 124,
                        "ranking": 1,
                        "numberOfCenturies": 100,
                        "average": 97.43
                      }
                    ]
  }


), ( "1003",

  { "player_name": "Virat",
    "jersey_no": 11,
    "country": "India",
    "team": "royal challengers bangalore",
    "player_stats": [ { "strike_rate": 136,
                        "ranking": 2,
                        "numberOfCenturies": 103,
                        "average": 107.11
                      }
                    ]
  }

), ( "1004",

  { "player_name": "Dale",
    "jersey_no": 63,
    "country": "South Africa",
    "team": "sunrisers hyderabad",
    "player_stats": [ { "strike_rate": 94,
                        "ranking": 21,
                        "numberOfCenturies": 10,
                        "average": 67.98
                      }
                    ]
  }

), ( "1005",

  { "player_name": "Maxwell",
    "jersey_no": 10,
    "country": "Australia",
    "team": "royal challengers bangalore",
    "player_stats": [ { "strike_rate": 31,
                        "ranking": 56,
                        "numberOfCenturies": 16,
                        "average": 91.61
                      }
                    ]
  }

), ( "1006",

  { "player_name": "KL Rahul",
    "jersey_no": 43,
    "country": "Afghanistan",
    "team": "punjab kings",
    "player_stats": [ { "strike_rate": 79.56,
                        "ranking": 7,
                        "numberOfCenturies": 23,
                        "average": 134.57
                      }
                    ]
  }

), ( "1007",

  { "player_name": "Ganguly",
    "jersey_no": 31,
    "country": "South Africa",
    "team": "pune warriors",
    "player_stats": [ { "strike_rate": 62,
                        "ranking": 9,
                        "numberOfCenturies": 60,
                        "average": 87.11
                      }
                    ]
  }

), ( "1008",

  { "player_name": "AB De Villiers",
    "jersey_no": 18,
    "country": "South Africa",
    "team": "royal challengers bangalore",
    "player_stats": [ { "strike_rate": 34,
                        "ranking": 4,
                        "numberOfCenturies": 80,
                        "average": 165
                      }
                    ]
  }

), ( "1009",

  { "player_name": "Rishab",
    "jersey_no": 333,
    "country": "India",
    "team": "delhi capital",
    "player_stats": [ { "strike_rate": 59,
                        "ranking": 13,
                        "numberOfCenturies": 20,
                        "average": 147.08
                      }
                    ]
  }

), ( "1010",
   { "player_name": "Gilchrist",
    "jersey_no": 169,
    "country": "Australia",
    "team": "kings punjab",
    "player_stats": [ { "strike_rate": 156,
                        "ranking": 6,
                        "numberOfCenturies": 89,
                        "average": 189.34
                      }
                    ]
  });
```

Creating Analytics Service on buckets



```
create dataset on batsman;
create dataset on bowler;
connect link Local;
```

400 | elapsed: 44.20ms | execution: 39.18ms | docs scanned: 0 | warnings: 0

We have executed 3 queries on the database created in couchbase using Analytics service.

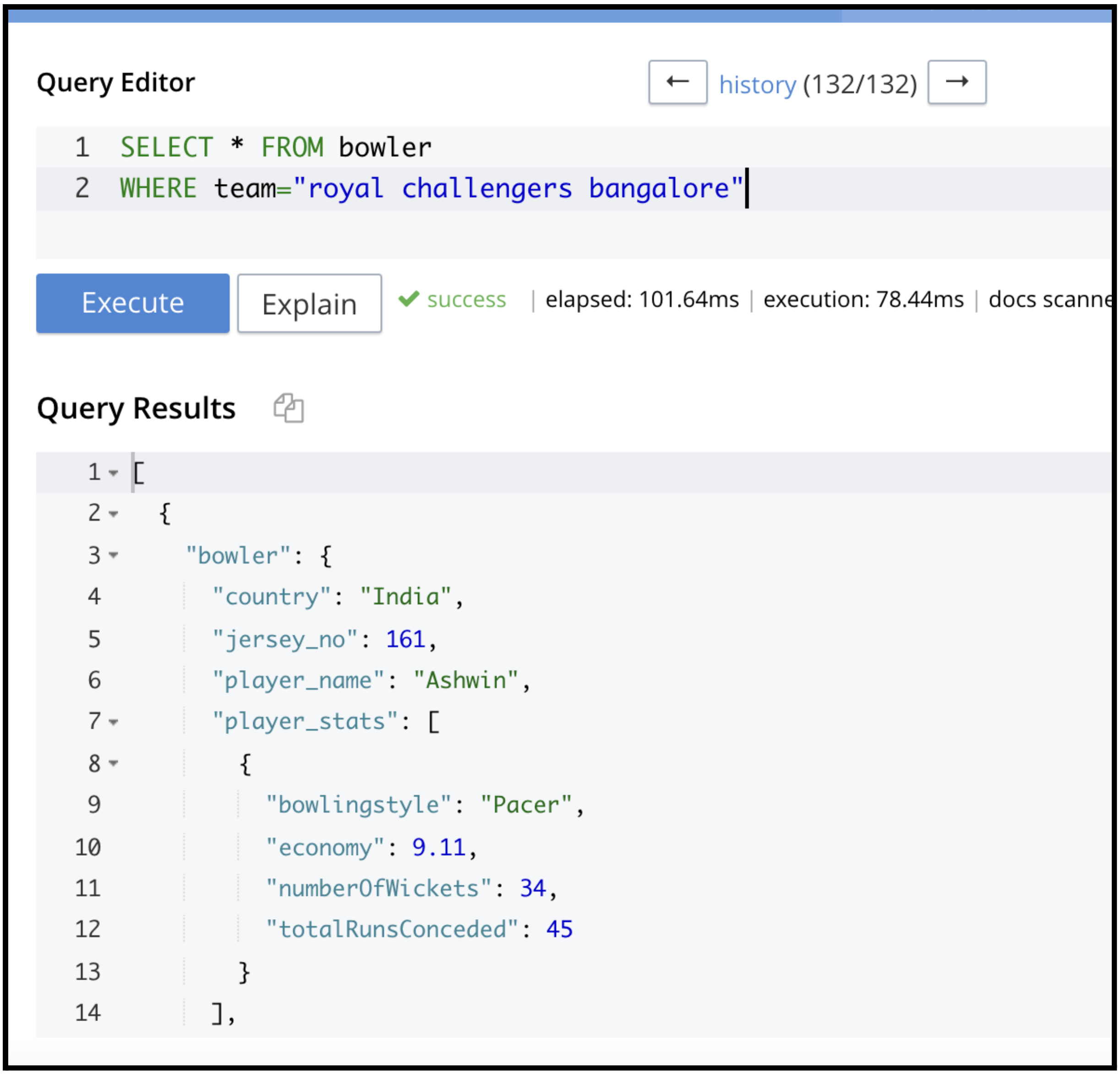
SELECT Query 1:

SELECT *

FROM bowler

WHERE team="royal challengers Bangalore";

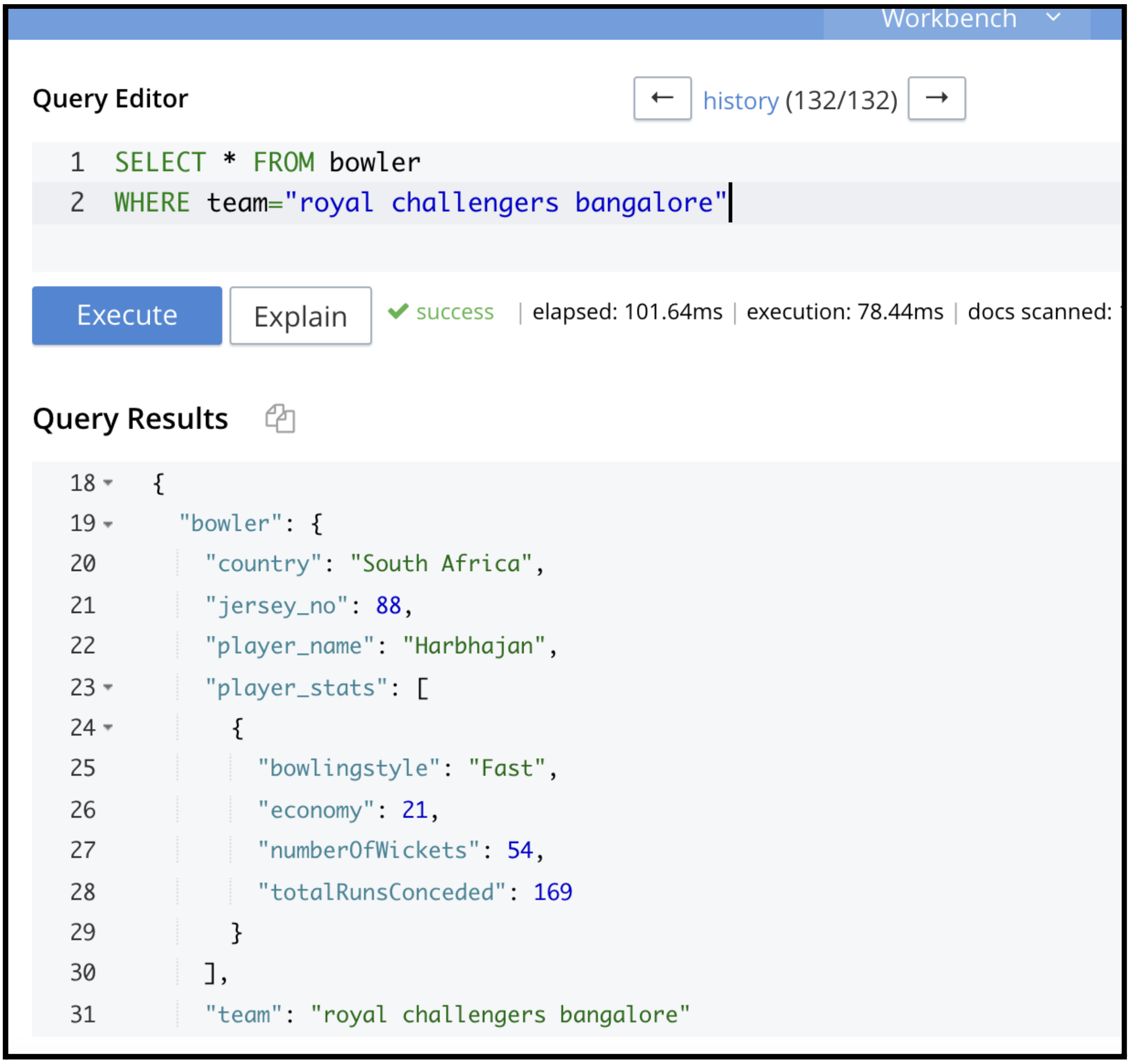
OUTPUT:



```
SELECT * FROM bowler
WHERE team="royal challengers bangalore"
```

✓ success | elapsed: 101.64ms | execution: 78.44ms | docs scanne

```
[
  {
    "bowler": {
      "country": "India",
      "jersey_no": 161,
      "player_name": "Ashwin",
      "player_stats": [
        {
          "bowlingstyle": "Pacer",
          "economy": 9.11,
          "numberOfWickets": 34,
          "totalRunsConceded": 45
        }
      ],
```



```
SELECT * FROM bowler
WHERE team="royal challengers bangalore"
```

✓ success | elapsed: 101.64ms | execution: 78.44ms | docs scanned:

```
{
  "bowler": {
    "country": "South Africa",
    "jersey_no": 88,
    "player_name": "Harbhajan",
    "player_stats": [
      {
        "bowlingstyle": "Fast",
        "economy": 21,
        "numberOfWickets": 54,
        "totalRunsConceded": 169
      }
    ],
    "team": "royal challengers bangalore"
```

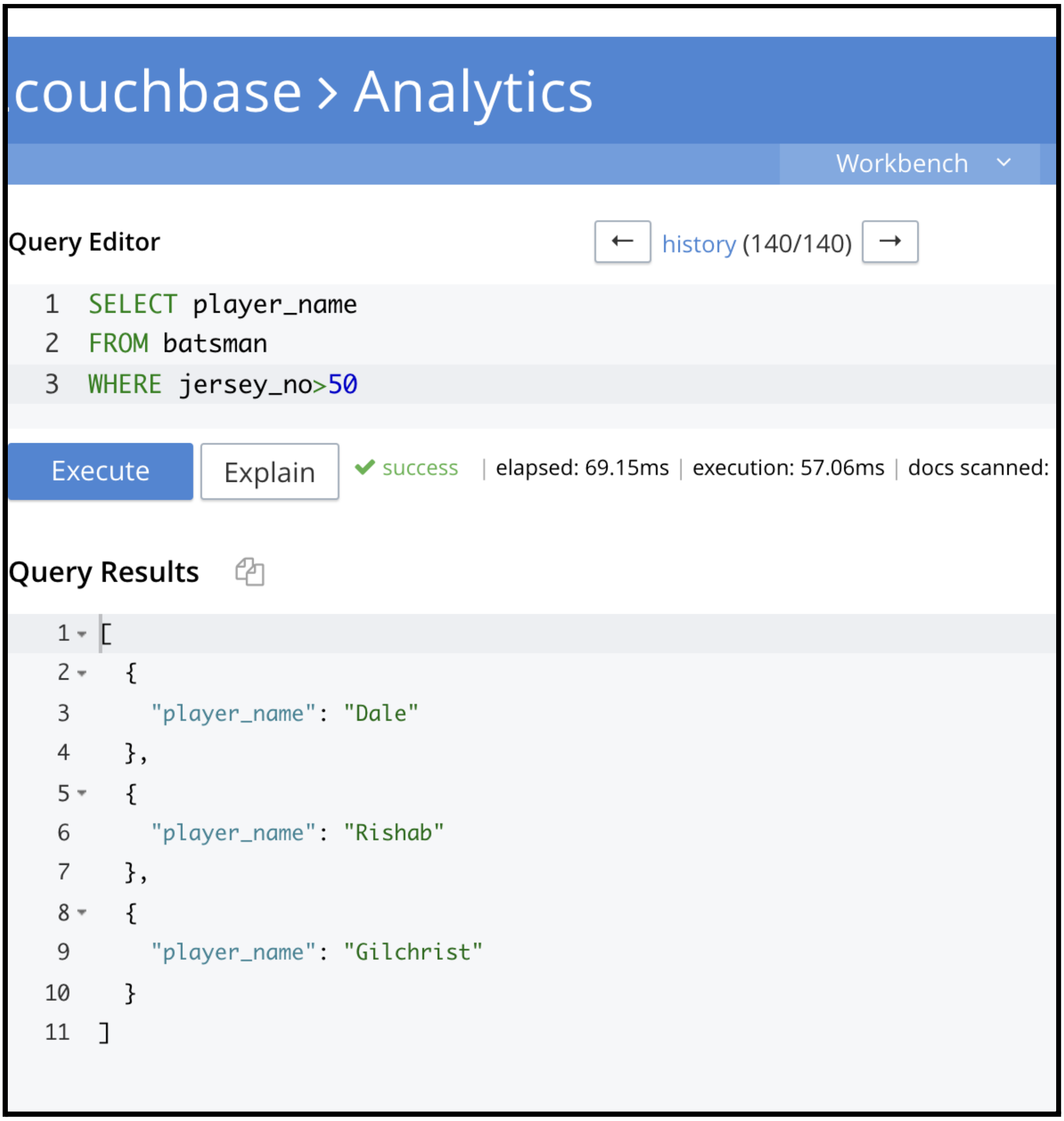
SELECT Query 2:

SELECT player_name

FROM batsman

WHERE jersey_no>50;

OUTPUT:



```
SELECT player_name
FROM batsman
WHERE jersey_no>50
```

✓ success | elapsed: 69.15ms | execution: 57.06ms | docs scanned:

```
[
  {
    "player_name": "Dale"
  },
  {
    "player_name": "Rishab"
  },
  {
    "player_name": "Gilchrist"
  }
]
```

JOIN Query:

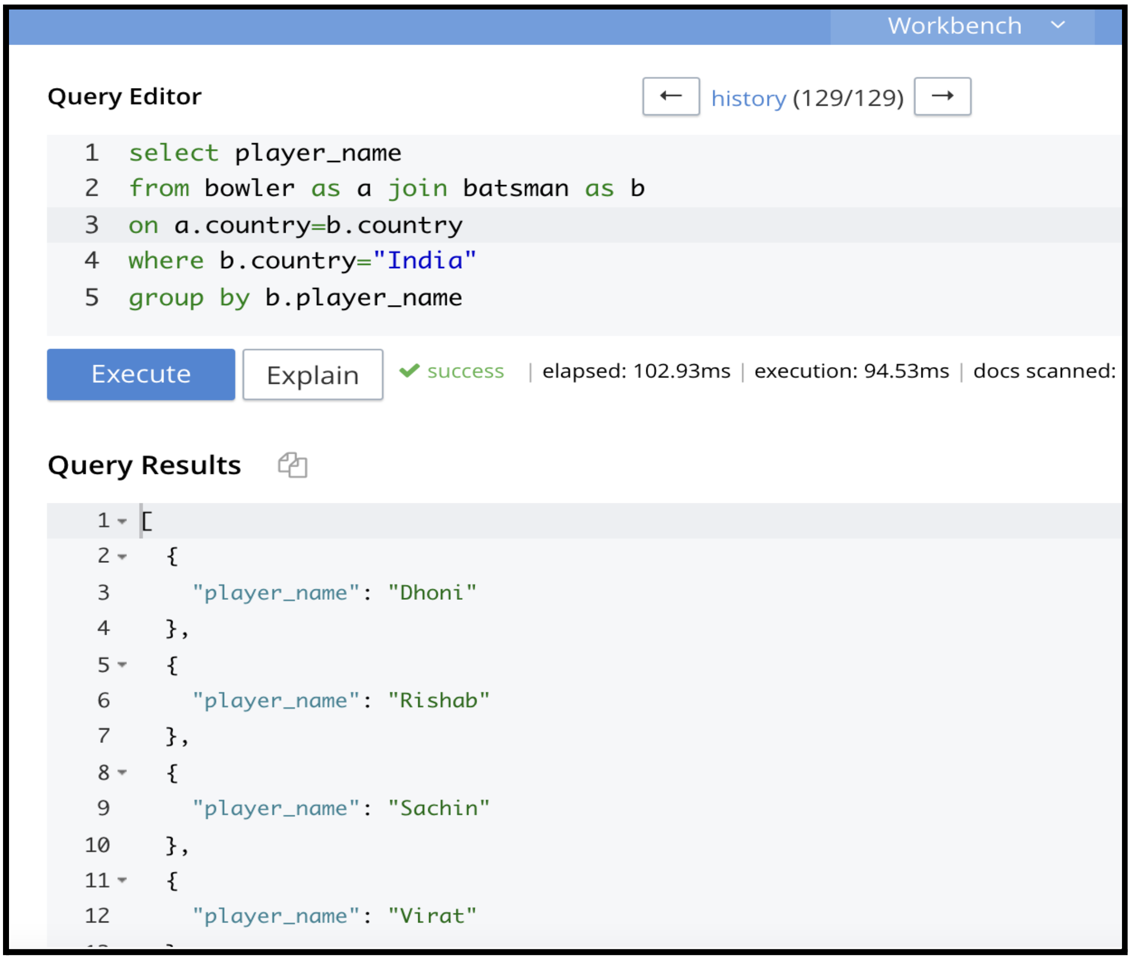
SELECT player_name

FROM bowler as a join batsman as b

ON a.country=b.country

WHERE b.country=”India”

GROUP BY b.player_name;



```
select player_name
from bowler as a join batsman as b
on a.country=b.country
where b.country="India"
group by b.player_name
```

✔ success | elapsed: 102.93ms | execution: 94.53ms | docs scanned:

```
[
  {
    "player_name": "Dhoni"
  },
  {
    "player_name": "Rishab"
  },
  {
    "player_name": "Sachin"
  },
  {
    "player_name": "Virat"
```

Summary

We came up with this idea after the ongoing IPL season was cancelled and was shifted to Dubai from India due to covid restrictions in our country. We figured it was quite a challenging task for the organizers to shift the entire tournament to a different country altogether. In Dubai, there are far fewer stadiums than there are in India. As a result, the entire schedule of the tournament had to be altered. A solution to this problem would be to create a system that would maintain records of all matches played in these stadiums on any given date during the duration of the competition. Even as a selector it becomes quite difficult in order to select a team on a short notice. The system displays the stats of each player as well, which helps in the selection process. The organizers can determine who the best player of the tournament is based on the stats of the tournament. We have used various functionalities in order to achieve this such as UDF to determine the player of the tournament. We have used a stored procedure to check the availability of the stadiums and a trigger is fired when a stadium is booked on a particular date.

CONCLUSION

After developing this project, we have realised the importance of object role modelling and how it can be an extremely effective tool to create databases that can provide tremendous amounts of ease to various applications in our day-to-day lives. Through our approach, we aim to decrease data redundancy and increase database connectivity. We have integrated various concepts taught in class such as triggers, stored procedures, and user defined functions. We have added various functionalities to our model using the concepts of buckets, JSON structures, Couchbase.

The project is still in its early stages, which means there are many more features to be added to enhance existing features. It is possible to use data prediction to help selectors pick teams without having to examine each player's stats individually. The system currently lacks a user interface, so users would need to have a working knowledge of SQL and databases. A graphical user interface should be developed along with a web application. We can create a dashboard to show which players have been selected for the upcoming matches, display teams that need a new coach and so on. Although the current state of the project is functional on a basic level, Enhancing this project's features would make it shine and set it apart from the rest.

REFERENCES

Textbooks:

- Terry Halpin’s Object-Role Modeling Fundamentals: A Practical Guide to Data Modeling with ORM
- Murach's SQL for SQL Server

Articles:

- Halpin, T. 2010, ‘Object-Role Modeling: Principles and Benefits’, International Journal of Information Systems Modeling and Design,

Websites:

- https://cricheroes.in/cricket-tournament-organiser-handbook
- https://www.researchgate.net/figure/Database-Tables-of-the-Central-Cricket-Database_tbl1_216361330
- https://data.world/datasets/cricket