SPORTIFY

Database Systems CS F212

Submitted on 17/04/2024 by:

Granth Jain (2022A7PS0172P)
Akshay Shukla (2022A7PS0087P)
Sriram Hebbale (2022A7PS0147P)
Gobind Singh (2022A7PS0083P)
Siddhartha Gotur (2022A7PS0070P)
Birla Institute Of Technology And Sciences(Pilani, Rajasthan)

Table Of Contents

Problem Description

The purpose of this project - **Sportify** is for users to be able to get a better experience when dealing with statistics relating to their favorite sports. With the immense amount of sports stats available to viewers, it is very easy to get overwhelmed and not be able to make sense of any of the numbers presented to you. This is where **Sportify** comes in handy. With an easy to use GUI, users can select their favorite sport and then choose queries that would return their desired subset of data from the database pertaining to that specific sport in an easy to read format. The functionality provided by us is focused on data from the NBA (2014-2015 season) and Formula1 (1950-2023 data) but can be expanded to include any number of sports. Further, we have also provided a login and sign up functionality where new users can create an account and login to access the portal. The frontend has been done using Java Swing while the backend has been done using MySQL with JDBC being used to connect the two.

Features

The features of our project are:

- User friendly login page with an option to sign up for a new user. The users can set a password of their choice.
- On logging in the user is given two buttons - one for dealing with NBA data and other for dealing with Formula1 data
- On selecting the Formula1 button the user is given an option to choose from the following queries -
    - Top 10 teams of all time and points
    - Total number of first position finishes for all the drivers
    - Fastest lap times of all circuits and drivers
    - Total number of points won by drivers until the old points system
    - Total number of points won by drivers with the new points system
    - Drivers and the different constructors that they have paired with
    - Drivers, circuits and the number of wins there
    - Driver with most number of wins on a particular circuit
- On selecting the NBA button the user is given an option to choose from the following queries -
    - Get any stat of any player
    - Which game happened on a particular date
    - List all Players with __ plus Points in a game and __ plus Assists
    - Arrange all the players in descending order on the basis of ___ stat
    - List Most-Valuable Players(MVP) from all Teams in the Season

Requirements

- MySQL - to store the entries of the database
- Java Swing - to create a GUI for the user to interact with (Frontend)
- Java Database Connectivity (JDBC) - to link the backend with the frontend

Anti Plagiarism Statement

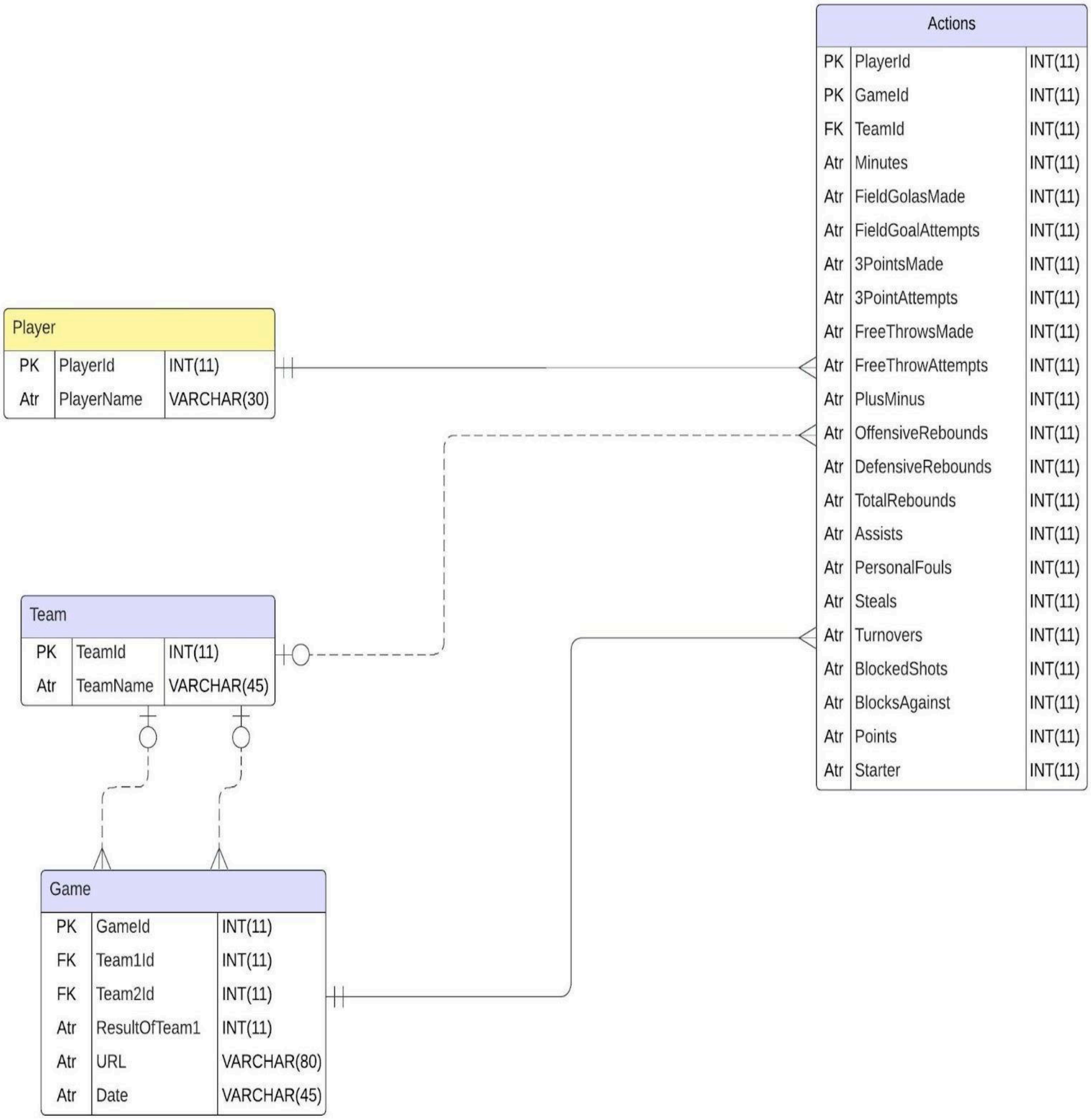
I declare in my honor that what has been written in this work has been written by me and that, with the exception of quotations, no part has been copied from scientific publications, the internet or from research works already presented in the academic field by me or by other students. In the case of parts taken from scientific publications, from the internet or from other documents, I have explicitly acknowledged the source. I also declare that I have taken note of the sanctions provided for in the case of plagiarism by the current regulations.

Name	**ID**	**Sign**
Akshay Shukla	2022A7PS0087P	AKShukla
Siddhartha Gotur	2022A7PS0070P	
Granth Jain	2022A7PS0172P	Granth
Gobind Singh	2022A7PS0083P	
Sriram Sudheer Hebbale	2022A7PS0147P	

Entity Relationship (ER) Model

Our project primarily consists of two databases - NBA and Formula1

ER Model for NBA -



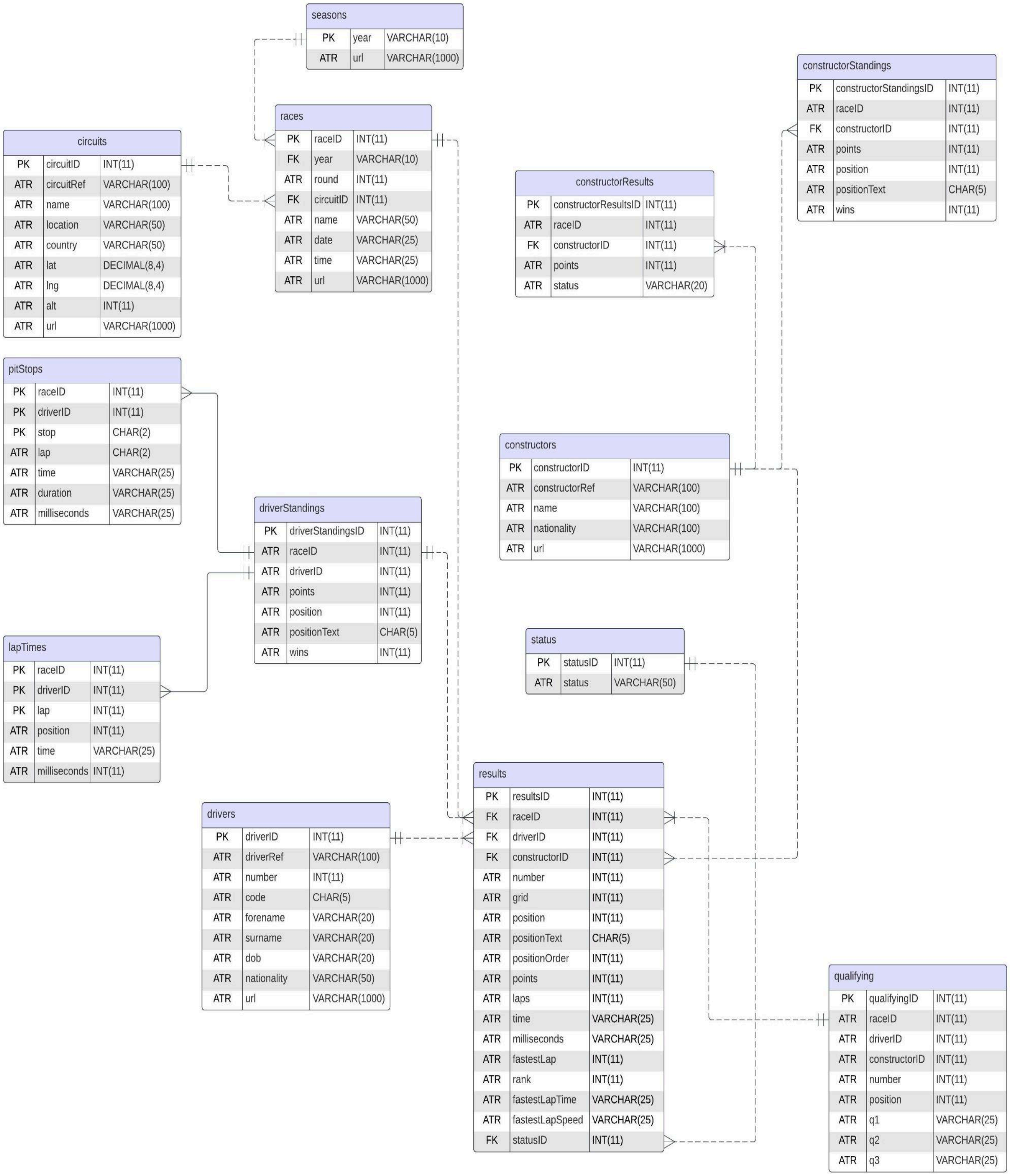
Actions		
PK	PlayerId	INT(11)
PK	GameId	INT(11)
FK	TeamId	INT(11)
Atr	Minutes	INT(11)
Atr	FieldGolasMade	INT(11)
Atr	FieldGoalAttempts	INT(11)
Atr	3PointsMade	INT(11)
Atr	3PointAttempts	INT(11)
Atr	FreeThrowsMade	INT(11)
Atr	FreeThrowAttempts	INT(11)
Atr	PlusMinus	INT(11)
Atr	OffensiveRebounds	INT(11)
Atr	DefensiveRebounds	INT(11)
Atr	TotalRebounds	INT(11)
Atr	Assists	INT(11)
Atr	PersonalFouls	INT(11)
Atr	Steals	INT(11)
Atr	Turnovers	INT(11)
Atr	BlockedShots	INT(11)
Atr	BlocksAgainst	INT(11)
Atr	Points	INT(11)
Atr	Starter	INT(11)

Player		
PK	PlayerId	INT(11)
Atr	PlayerName	VARCHAR(30)

Team		
PK	TeamId	INT(11)
Atr	TeamName	VARCHAR(45)

Game		
PK	GameId	INT(11)
FK	Team1Id	INT(11)
FK	Team2Id	INT(11)
Atr	ResultOfTeam1	INT(11)
Atr	URL	VARCHAR(80)
Atr	Date	VARCHAR(45)

ER Model for Formula1 -



seasons

PK	year	VARCHAR(10)
ATR	url	VARCHAR(1000)

constructorStandings

PK	constructorStandingsID	INT(11)
ATR	raceID	INT(11)
FK	constructorID	INT(11)
ATR	points	INT(11)
ATR	position	INT(11)
ATR	positionText	CHAR(5)
ATR	wins	INT(11)

races

PK	raceID	INT(11)
FK	year	VARCHAR(10)
ATR	round	INT(11)
FK	circuitID	INT(11)
ATR	name	VARCHAR(50)
ATR	date	VARCHAR(25)
ATR	time	VARCHAR(25)
ATR	url	VARCHAR(1000)

circuits

PK	circuitID	INT(11)
ATR	circuitRef	VARCHAR(100)
ATR	name	VARCHAR(100)
ATR	location	VARCHAR(50)
ATR	country	VARCHAR(50)
ATR	lat	DECIMAL(8,4)
ATR	lng	DECIMAL(8,4)
ATR	alt	INT(11)
ATR	url	VARCHAR(1000)

constructorResults

PK	constructorResultsID	INT(11)
ATR	raceID	INT(11)
FK	constructorID	INT(11)
ATR	points	INT(11)
ATR	status	VARCHAR(20)

pitStops

PK	raceID	INT(11)
PK	driverID	INT(11)
PK	stop	CHAR(2)
ATR	lap	CHAR(2)
ATR	time	VARCHAR(25)
ATR	duration	VARCHAR(25)
ATR	milliseconds	VARCHAR(25)

constructors

PK	constructorID	INT(11)
ATR	constructorRef	VARCHAR(100)
ATR	name	VARCHAR(100)
ATR	nationality	VARCHAR(100)
ATR	url	VARCHAR(1000)

driverStandings

PK	driverStandingsID	INT(11)
ATR	raceID	INT(11)
ATR	driverID	INT(11)
ATR	points	INT(11)
ATR	position	INT(11)
ATR	positionText	CHAR(5)
ATR	wins	INT(11)

status

PK	statusID	INT(11)
ATR	status	VARCHAR(50)

lapTimes

PK	raceID	INT(11)
PK	driverID	INT(11)
PK	lap	INT(11)
ATR	position	INT(11)
ATR	time	VARCHAR(25)
ATR	milliseconds	INT(11)

results

PK	resultsID	INT(11)
FK	raceID	INT(11)
FK	driverID	INT(11)
FK	constructorID	INT(11)
ATR	number	INT(11)
ATR	grid	INT(11)
ATR	position	INT(11)
ATR	positionText	CHAR(5)
ATR	positionOrder	INT(11)
ATR	points	INT(11)
ATR	laps	INT(11)
ATR	time	VARCHAR(25)
ATR	milliseconds	VARCHAR(25)
ATR	fastestLap	INT(11)
ATR	rank	INT(11)
ATR	fastestLapTime	VARCHAR(25)
ATR	fastestLapSpeed	VARCHAR(25)
FK	statusID	INT(11)

drivers

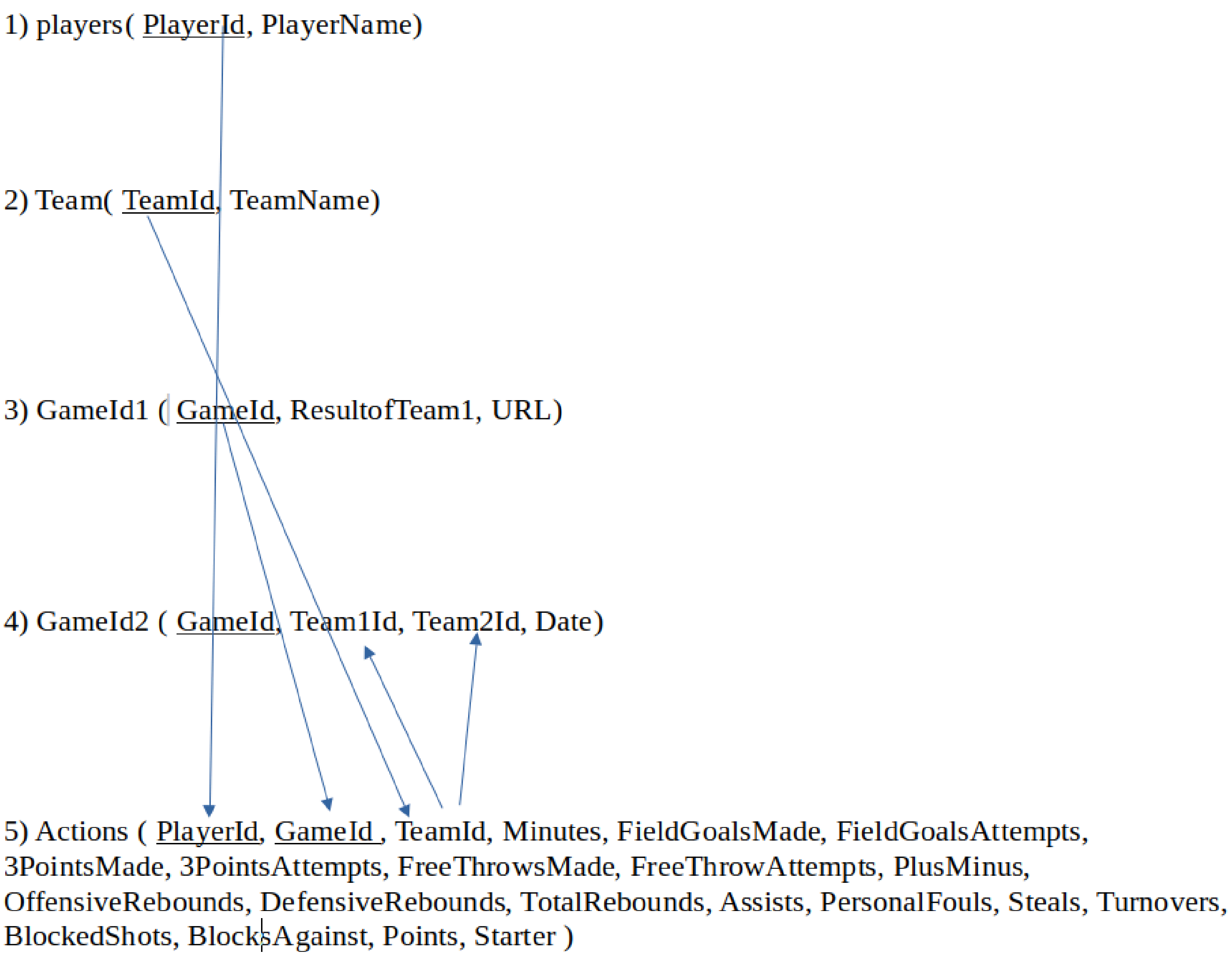
PK	driverID	INT(11)
ATR	driverRef	VARCHAR(100)
ATR	number	INT(11)
ATR	code	CHAR(5)
ATR	forename	VARCHAR(20)
ATR	surname	VARCHAR(20)
ATR	dob	VARCHAR(20)
ATR	nationality	VARCHAR(50)
ATR	url	VARCHAR(1000)

qualifying

PK	qualifyingID	INT(11)
ATR	raceID	INT(11)
ATR	driverID	INT(11)
ATR	constructorID	INT(11)
ATR	number	INT(11)
ATR	position	INT(11)
ATR	q1	VARCHAR(25)
ATR	q2	VARCHAR(25)
ATR	q3	VARCHAR(25)

The normalization of the tables has been demonstrated in - https://docs.google.com/document/d/1iCU0ygFO8ys9Ty7mnXQ01BYvAtjAJ5i7h4GxRiFJP-4/edit?usp=sharing

Relational Schema Model For NBA:



1) players(PlayerId, PlayerName)

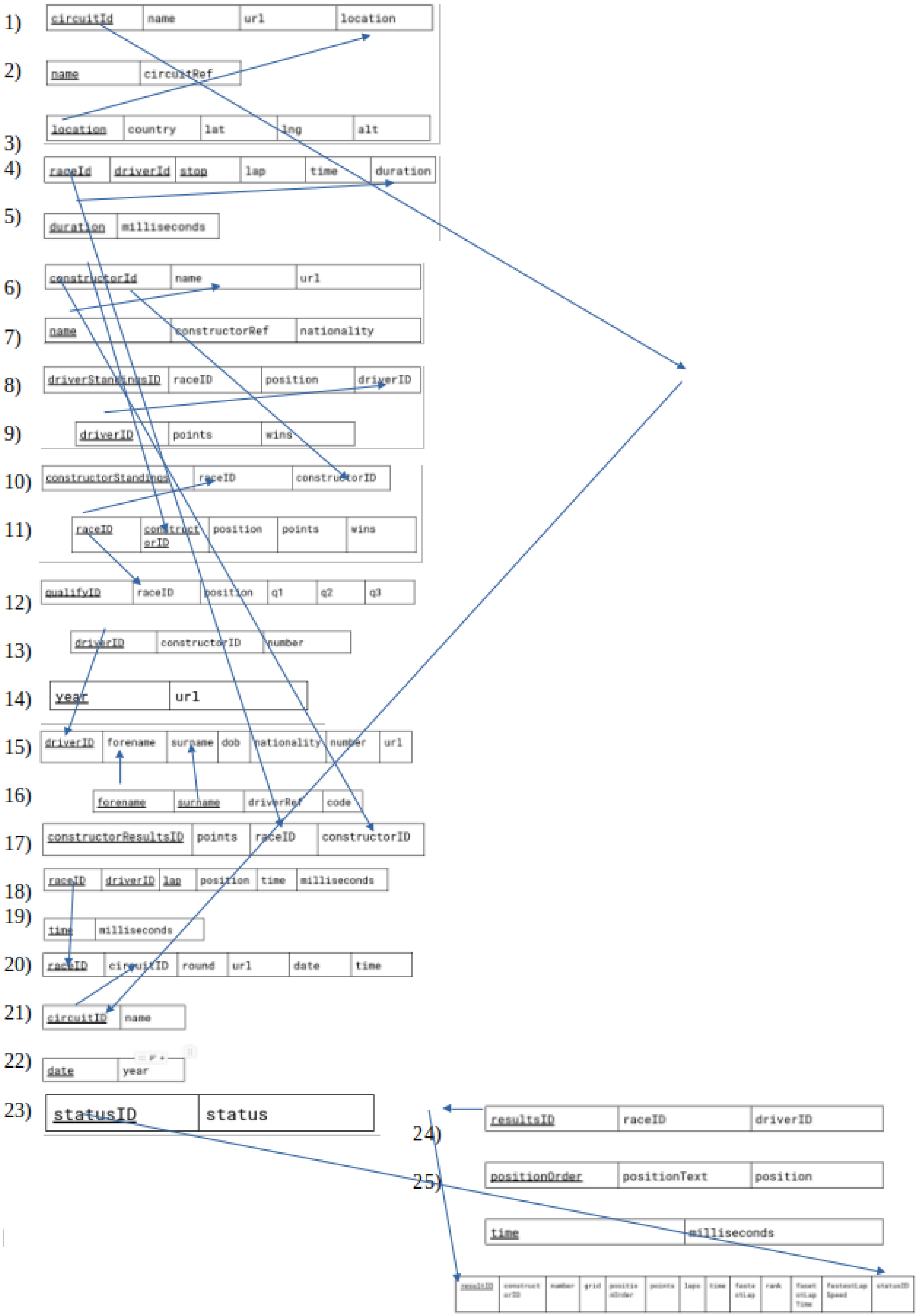
2) Team(TeamId, TeamName)

3) GameId1 (GameId, ResultofTeam1, URL)

4) GameId2 (GameId, Team1Id, Team2Id, Date)

5) Actions (PlayerId, GameId , TeamId, Minutes, FieldGoalsMade, FieldGoalsAttempts, 3PointsMade, 3PointsAttempts, FreeThrowsMade, FreeThrowAttempts, PlusMinus, OffensiveRebounds, DefensiveRebounds, TotalRebounds, Assists, PersonalFouls, Steals, Turnovers, BlockedShots, BlocksAgainst, Points, Starter)

Relational Schema Model For F1:



1) circuitId | name | url | location

2) name | circuitRef

3) location | country | lat | lng | alt

4) raceId | driverId | stop | lap | time | duration

5) duration | milliseconds

6) constructorId | name | url

7) name | constructorRef | nationality

8) driverStandingsID | raceID | position | driverID

9) driverID | points | wins

10) constructorStandings | raceID | constructorID

11) raceID | constructorID | position | points | wins

12) qualifyID | raceID | position | q1 | q2 | q3

13) driverID | constructorID | number

14) year | url

15) driverID | forename | surname | dob | nationality | number | url

16) forename | surname | driverRef | code

17) constructorResultsID | points | raceID | constructorID

18) raceID | driverID | lap | position | time | milliseconds

19) time | milliseconds

20) raceID | circuitID | round | url | date | time

21) circuitID | name

22) date | year

23) statusID | status

24) resultsID | raceID | driverID

25) positionOrder | positionText | position

time | milliseconds

resultID | constructorID | number | grid | positionOrder | points | laps | time | fastestLap | rank | fastestLapTime | fastestLapSpeed | statusID

SQL Queries

We now explain the important queries used

Creating the tables

For the NBA tables we directly made the normalized tables in the database while for Formula1 we initially made the 13 pre-normalization tables after which we broke these tables down into the 26 tables created post normalization.

```
CREATE DATABASE if not exists formula1;
CREATE DATABASE if not exists formula1_final;
CREATE DATABASE if not exists NBA;

USE formula1;

CREATE TABLE circuits(
circuitID INT,
circuitRef VARCHAR(100),
name VARCHAR(100),
location VARCHAR(50),
country VARCHAR(50),
lat DECIMAL(8,4),
lng DECIMAL(8,4),
alt INT,
url VARCHAR(1000),
PRIMARY KEY(circuitID)
);

LOAD DATA LOCAL INFILE
'/Users/akshayshukla/Downloads/formula1/circuits.csv'
INTO TABLE circuits
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\n'
```

```
IGNORE 1 ROWS;

CREATE TABLE constructors(
constructorID INT,
constructorRef VARCHAR(100),
name VARCHAR(100),
nationality VARCHAR(100),
url VARCHAR(1000),
PRIMARY KEY(constructorID)
);

LOAD DATA LOCAL INFILE
'/Users/akshayshukla/Downloads/formula1/constructors.csv'
INTO TABLE constructors
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\n'
IGNORE 1 ROWS;

CREATE TABLE constructorResults(
constructorResultsID INT,
raceID INT,
constructorID INT,
points INT,
status VARCHAR(20),
PRIMARY KEY(constructorResultsID),
FOREIGN KEY(constructorID) REFERENCES
constructors(constructorID)
);

LOAD DATA LOCAL INFILE
'/Users/akshayshukla/Downloads/formula1/constructorResults.csv'
INTO TABLE constructorResults
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
```

```
LINES TERMINATED BY '\n'
IGNORE 1 ROWS;

CREATE TABLE constructorStandings(
constructorStandingsID INT,
raceID INT,
constructorID INT,
points INT,
position INT,
positionText CHAR(5),
wins INT,
PRIMARY KEY(constructorStandingsID),
FOREIGN KEY(constructorID) REFERENCES
constructors(constructorID)
);

LOAD DATA LOCAL INFILE
'/Users/akshayshukla/Downloads/formula1/constructorStandings.csv
'
INTO TABLE constructorStandings
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\n'
IGNORE 1 ROWS;

CREATE TABLE drivers(
driverID INT,
driverRef VARCHAR(100),
number INT,
code CHAR(5),
forename VARCHAR(20),
surname VARCHAR(20),
dob VARCHAR(20),
nationality VARCHAR(50),
url VARCHAR(1000),
PRIMARY KEY(driverID)
);

LOAD DATA LOCAL INFILE
'/Users/akshayshukla/Downloads/formula1/drivers.csv'
INTO TABLE drivers
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\n'
IGNORE 1 ROWS;

CREATE TABLE driverStandings(
driverStandingsID INT,
raceID INT,
driverID INT,
points INT,
position INT,
positionText CHAR(5),
wins INT,
PRIMARY KEY(driverStandingsID),
UNIQUE KEY(raceID, driverID)
);

LOAD DATA LOCAL INFILE
'//Users/akshayshukla/Downloads/formula1/driverStandings.csv'
INTO TABLE driverStandings
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\n'
IGNORE 1 ROWS;

CREATE TABLE lapTimes(
raceID INT,
driverID INT,
lap INT,
position INT,
time VARCHAR(25),
milliseconds INT,
PRIMARY KEY(raceID, driverID, lap),
FOREIGN KEY(raceID, driverID) REFERENCES driverStandings(raceID,
driverID)
);

LOAD DATA LOCAL INFILE
'/Users/akshayshukla/Downloads/formula1/lapTimes.csv'
INTO TABLE lapTimes
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\n'
IGNORE 1 ROWS;

CREATE TABLE pitStops(
raceID INT,
driverID INT,
stop CHAR(2),
lap CHAR(2),
time VARCHAR(25),
duration VARCHAR(25),
milliseconds VARCHAR(25),
PRIMARY KEY(raceID, driverID, stop),
FOREIGN KEY(raceID, driverID) REFERENCES driverStandings(raceID,
driverID)
);

LOAD DATA LOCAL INFILE
'/Users/akshayshukla/Downloads/formula1/pitStops.csv'
INTO TABLE pitStops
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\n'
```

```
IGNORE 1 ROWS;

CREATE TABLE qualifying(
qualifyingID INT,
raceID INT,
driverID INT,
constructorID INT,
number INT,
position INT,
q1 VARCHAR(25),
q2 VARCHAR(25),
q3 VARCHAR(25),
PRIMARY KEY(qualifyingID),
UNIQUE KEY(raceID, driverID,constructorID)
);

LOAD DATA LOCAL INFILE
'/Users/akshayshukla/Downloads/formula1/qualifying.csv'
INTO TABLE qualifying
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\n'
IGNORE 1 ROWS;


CREATE TABLE races(
raceID INT,
year VARCHAR(10),
round INT,
circuitID INT,
name VARCHAR(50),
date VARCHAR(25),
time VARCHAR(25),
url VARCHAR(1000),
PRIMARY KEY(raceID),
FOREIGN KEY(circuitID) REFERENCES circuits(circuitID),
FOREIGN KEY(year) REFERENCES seasons(year)
);

LOAD DATA LOCAL INFILE
'/Users/akshayshukla/Downloads/formula1/races.csv'
INTO TABLE races
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\n'
IGNORE 1 ROWS;

CREATE TABLE seasons(
year VARCHAR(10),
url VARCHAR(1000),
PRIMARY KEY(year)
);

LOAD DATA LOCAL INFILE
'/Users/akshayshukla/Downloads/formula1/seasons.csv'
INTO TABLE seasons
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\n'
IGNORE 1 ROWS;

CREATE TABLE status(
statusID INT,
status VARCHAR(50),
PRIMARY KEY(statusID)
);

LOAD DATA LOCAL INFILE
'/Users/akshayshukla/Downloads/formula1/status.csv'
INTO TABLE status
FIELDS TERMINATED BY ','
```

```sql
ENCLOSED BY '"'
LINES TERMINATED BY '\n'
IGNORE 1 ROWS;

CREATE TABLE results(
resultsID INT,
raceID INT,
driverID INT,
constructorID INT,
number INT,
grid INT,
position INT,
positionText CHAR(5),
positionOrder INT,
points INT,
laps INT,
time VARCHAR(25),
milliseconds VARCHAR(25),
fastestLap INT,
rank INT,
fastestLapTime VARCHAR(25),
fastestLapSpeed VARCHAR(25),
statusID INT,
PRIMARY KEY(resultsID),
FOREIGN KEY(raceID) REFERENCES races(raceID),
FOREIGN KEY(driverID) REFERENCES drivers(driverID),
FOREIGN KEY(constructorID) REFERENCES
constructors(constructorID),
FOREIGN KEY(statusID) REFERENCES status(statusID),
FOREIGN KEY(raceID, driverID, constructorID) REFERENCES
qualifying(raceID, driverID, constructorID),
FOREIGN KEY(raceID, driverID) REFERENCES driverStandings(raceID,
driverID)
);
```

```
LOAD DATA LOCAL INFILE
'/Users/akshayshukla/Downloads/formula1/results.csv'
INTO TABLE results
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\n'
IGNORE 1 ROWS;

USE formula1_final;

CREATE TABLE circuit_info(
circuitID INT,
name VARCHAR(100),
url VARCHAR(1000),
location VARCHAR(50),
PRIMARY KEY(circuitID)
);

CREATE TABLE circuit_ref(
name VARCHAR(100),
circuitRef VARCHAR(100),
PRIMARY KEY(name)
);

CREATE TABLE locations(
location VARCHAR(50),
country VARCHAR(50),
lat DECIMAL(8,4),
lng DECIMAL(8,4),
alt INT,
PRIMARY KEY(lat,lng))
);
```

```
INSERT INTO circuit_info(circuitID,name,url,location) select
circuits.circuitID,circuits.name,circuits.url,circuits.location
from formula1.circuits;


INSERT INTO circuit_ref(name,circuitRef) select
circuits.name, circuits.circuitRef
from formula1.circuits;

INSERT INTO locations(location,country,lat,lng,alt)select
circuits.location,circuits.country,circuits.lat,circuits.lng,
circuits.alt from formula1.circuits;
```

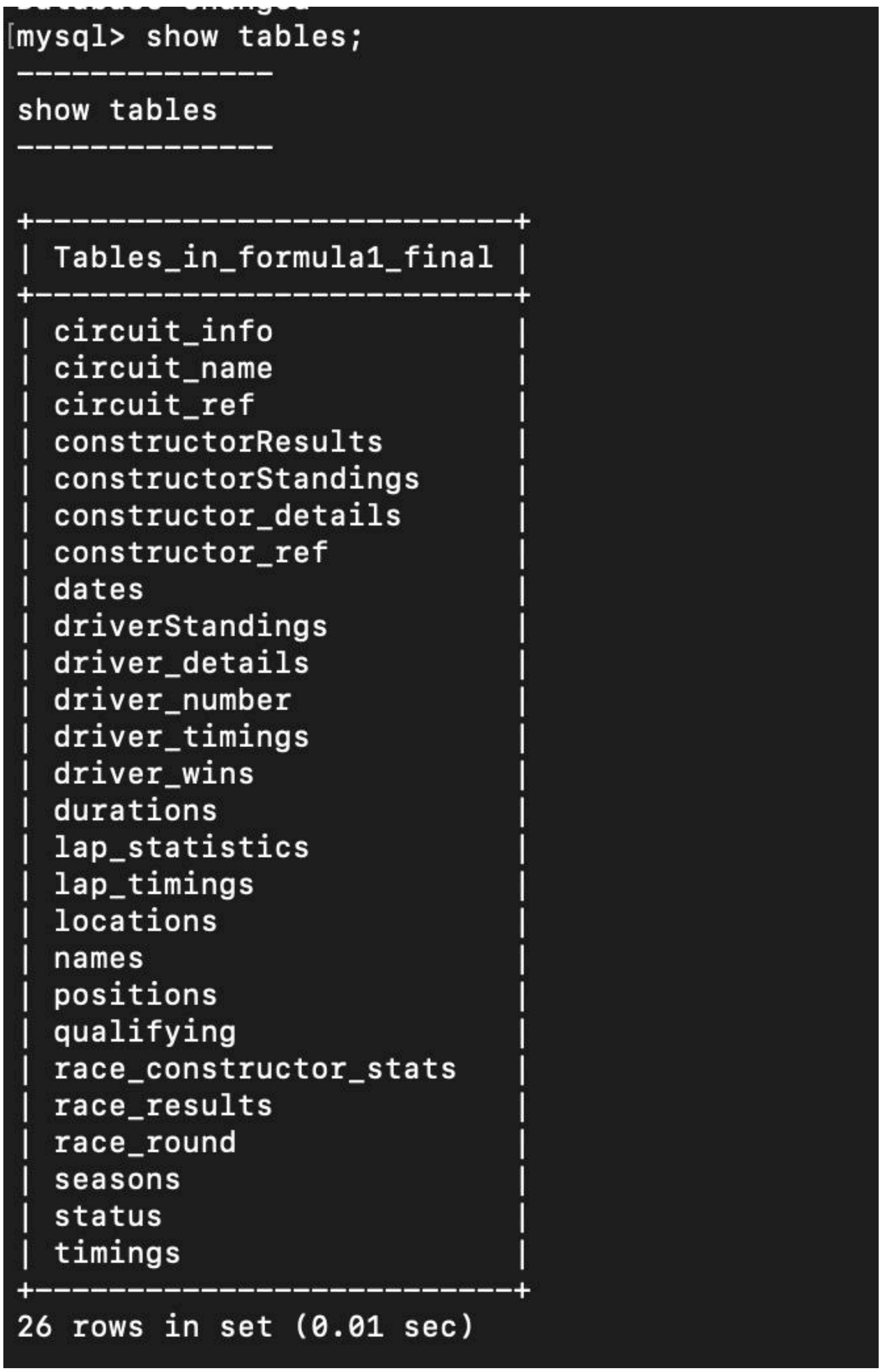
This is how we broke down the circuits table into the three child tables derived from it post normalization. We created the normalized forms separately and inserted the corresponding values of the original table using select - insertion query. A similar process was used for the other tables in the formula1_final database.

Creating of NBA tables

```
USE NBA;

create table Team(
TeamId INT,
TeamName VARCHAR(45),
PRIMARY KEY(TeamId));

create table Player(
PlayerId INT,
PlayerName VARCHAR(30),
PRIMARY KEY(PlayerId));

create table Game(
GameId INT,
Date VARCHAR(45),
Team1Id INT,
Team2Id INT,
PRIMARY KEY(GameId),
FOREIGN KEY(Team1Id) REFERENCES Team(TeamId));

create table GameStatus(
GameId INT,
URL VARCHAR(80),
ResultOfTeam1 INT,
PRIMARY KEY(GameId));


create table Actions(
PlayerId INT,
GameId INT,
TeamId INT,
Minutes INT,
FieldGoalsMade INT,
FieldGoalAttempts INT,
3PointsMade INT,
3PointAttempts INT,
FreeThrowsMade INT,
FreeThrowAttempts INT,
PlusMinus INT,
OffensiveRebounds INT,
DefensiveRebounds INT,
TotalRebounds INT,
Assist INT,
PersonalFouls INT,
Steals INT,
Turnovers INT,
BlockedShots INT,
BlocksAgainst INT,
Points INT,
Starter INT,
PRIMARY KEY(PlayerId,GameId),
FOREIGN KEY(TeamId) REFERENCES Team(TeamId));

LOAD DATA LOCAL INFILE
'/Users/akshayshukla/Downloads/NBA/Team.csv'
INTO TABLE Team
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\n'
IGNORE 1 ROWS;

LOAD DATA LOCAL INFILE
'/Users/akshayshukla/Downloads/NBA/Player.csv'
INTO TABLE Player
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\n'
IGNORE 1 ROWS;

LOAD DATA LOCAL INFILE
'/Users/akshayshukla/Downloads/NBA/Game.csv'
INTO TABLE Game
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\n'
IGNORE 1 ROWS;

LOAD DATA LOCAL INFILE
'/Users/akshayshukla/Downloads/NBA/GameStatus.csv'
INTO TABLE GameStatus
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\n'
IGNORE 1 ROWS;
```

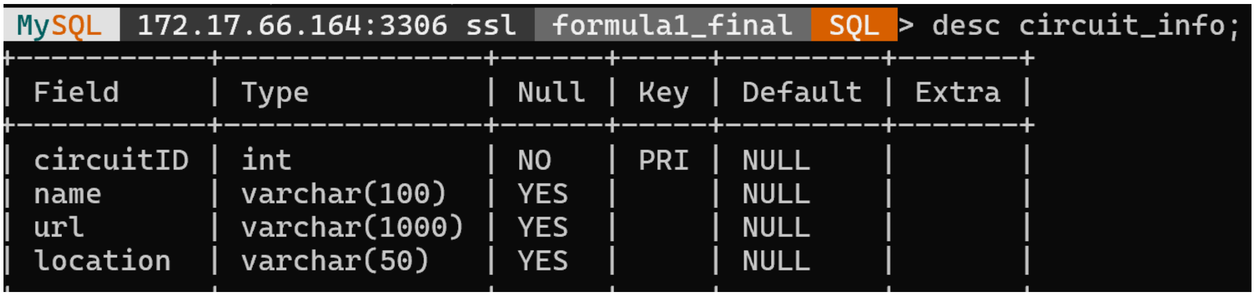
```
LOAD DATA LOCAL INFILE
'/Users/akshayshukla/Downloads/NBA/Actions.csv'
INTO TABLE Actions
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\n'
IGNORE 1 ROWS;
```

Tables Created in the Formula1 database

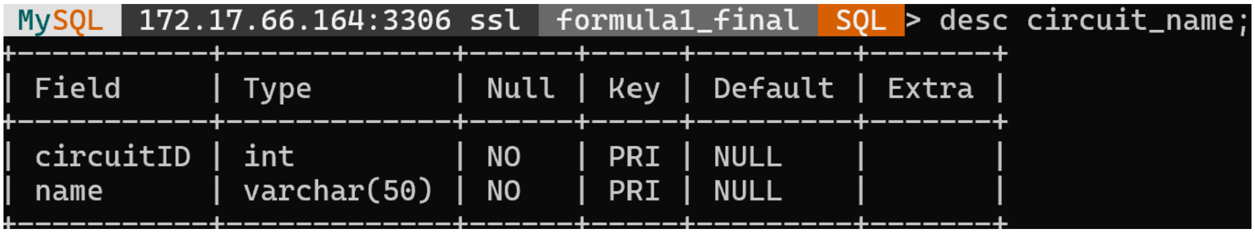


```
Database changed
[mysql> show tables;
--------------
show tables
--------------

+--------------------------+
| Tables_in_formula1_final |
+--------------------------+
| circuit_info             |
| circuit_name             |
| circuit_ref              |
| constructorResults       |
| constructorStandings     |
| constructor_details      |
| constructor_ref          |
| dates                    |
| driverStandings          |
| driver_details           |
| driver_number            |
| driver_timings           |
| driver_wins              |
| durations                |
| lap_statistics           |
| lap_timings              |
| locations                |
| names                    |
| positions                |
| qualifying               |
| race_constructor_stats   |
| race_results             |
| race_round               |
| seasons                  |
| status                   |
| timings                  |
+--------------------------+
26 rows in set (0.01 sec)
```

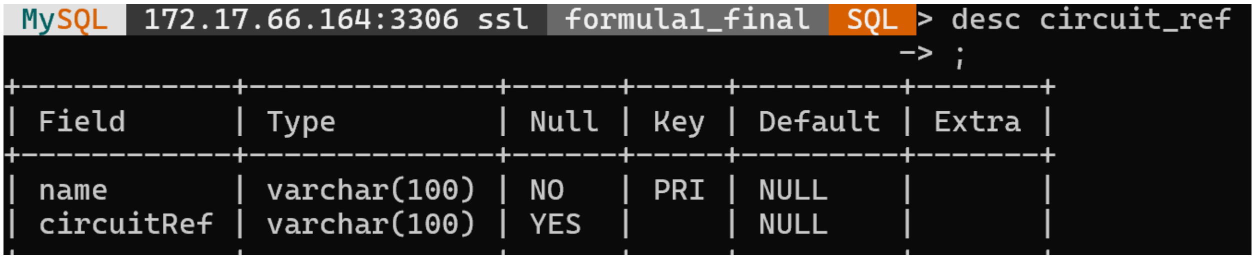
Structure of tables in F1:



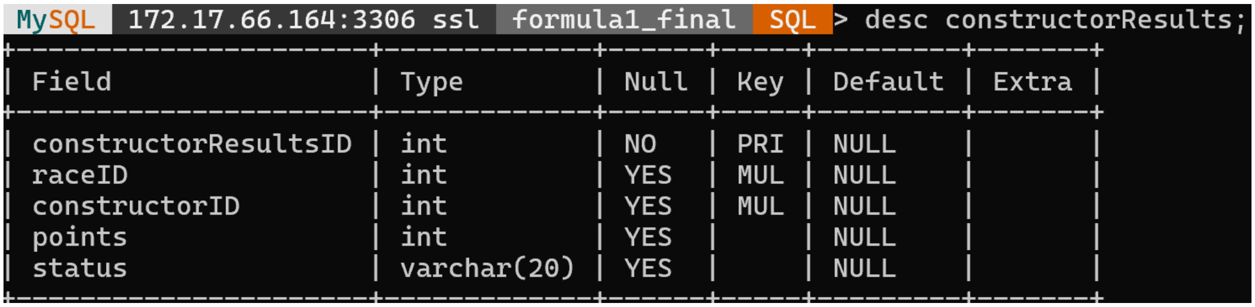
```
MySQL  172.17.66.164:3306 ssl  formula1_final  SQL > desc circuit_info;
+-----------+---------------+------+-----+---------+-------+
| Field     | Type          | Null | Key | Default | Extra |
+-----------+---------------+------+-----+---------+-------+
| circuitID | int           | NO   | PRI | NULL    |       |
| name      | varchar(100)  | YES  |     | NULL    |       |
| url       | varchar(1000) | YES  |     | NULL    |       |
| location  | varchar(50)   | YES  |     | NULL    |       |
```



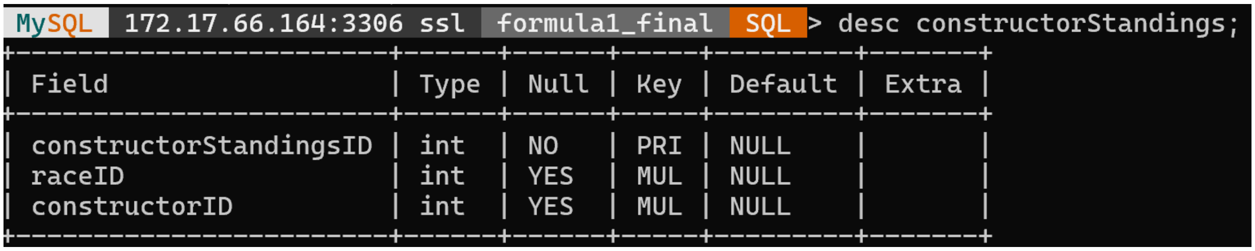
```
MySQL  172.17.66.164:3306 ssl  formula1_final  SQL > desc circuit_name;
+-----------+-------------+------+-----+---------+-------+
| Field     | Type        | Null | Key | Default | Extra |
+-----------+-------------+------+-----+---------+-------+
| circuitID | int         | NO   | PRI | NULL    |       |
| name      | varchar(50) | NO   | PRI | NULL    |       |
+-----------+-------------+------+-----+---------+-------+
```



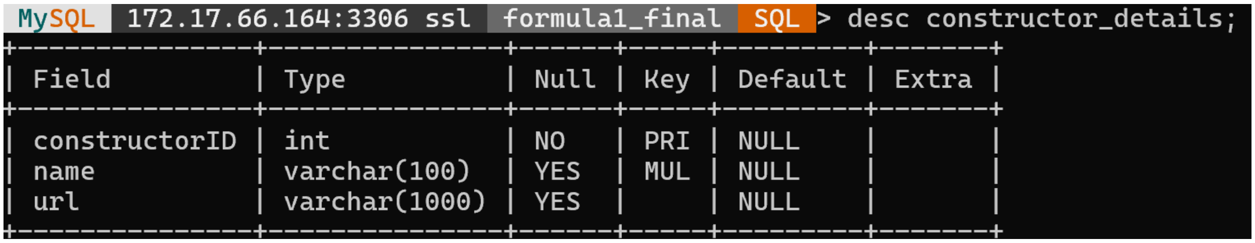
```
MySQL  172.17.66.164:3306 ssl  formula1_final  SQL > desc circuit_ref
                                                    -> ;
+------------+--------------+------+-----+---------+-------+
| Field      | Type         | Null | Key | Default | Extra |
+------------+--------------+------+-----+---------+-------+
| name       | varchar(100) | NO   | PRI | NULL    |       |
| circuitRef | varchar(100) | YES  |     | NULL    |       |
```



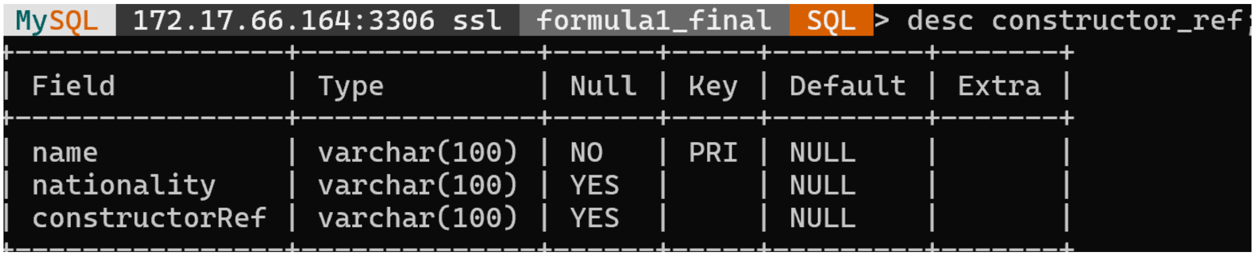
```
MySQL  172.17.66.164:3306 ssl  formula1_final  SQL > desc constructorResults;
+----------------------+-------------+------+-----+---------+-------+
| Field                | Type        | Null | Key | Default | Extra |
+----------------------+-------------+------+-----+---------+-------+
| constructorResultsID | int         | NO   | PRI | NULL    |       |
| raceID               | int         | YES  | MUL | NULL    |       |
| constructorID        | int         | YES  | MUL | NULL    |       |
| points               | int         | YES  |     | NULL    |       |
| status               | varchar(20) | YES  |     | NULL    |       |
+----------------------+-------------+------+-----+---------+-------+
```



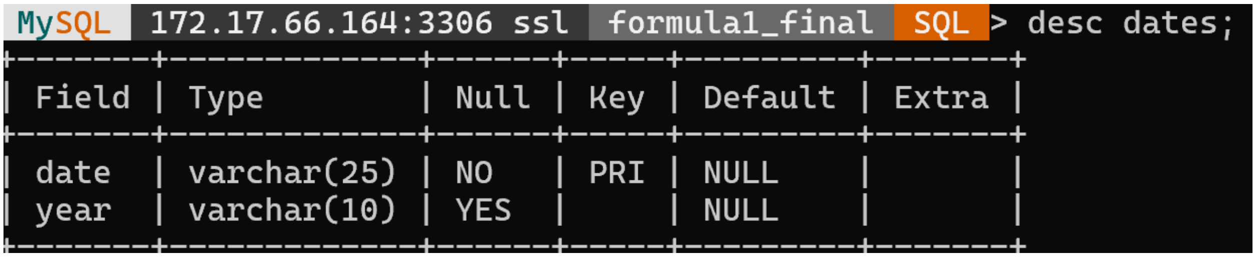
```
MySQL  172.17.66.164:3306 ssl  formula1_final  SQL > desc constructorStandings;
+------------------------+------+------+-----+---------+-------+
| Field                  | Type | Null | Key | Default | Extra |
+------------------------+------+------+-----+---------+-------+
| constructorStandingsID | int  | NO   | PRI | NULL    |       |
| raceID                 | int  | YES  | MUL | NULL    |       |
| constructorID          | int  | YES  | MUL | NULL    |       |
+------------------------+------+------+-----+---------+-------+
```



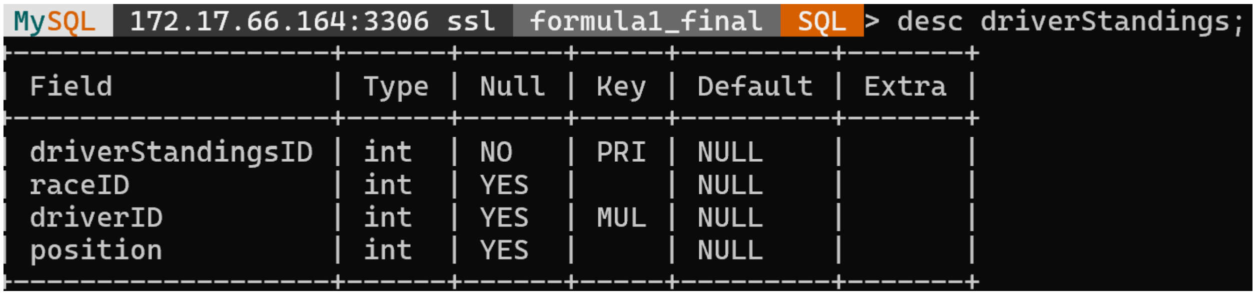
```
MySQL  172.17.66.164:3306 ssl  formula1_final  SQL > desc constructor_details;
+---------------+---------------+------+-----+---------+-------+
| Field         | Type          | Null | Key | Default | Extra |
+---------------+---------------+------+-----+---------+-------+
| constructorID | int           | NO   | PRI | NULL    |       |
| name          | varchar(100)  | YES  | MUL | NULL    |       |
| url           | varchar(1000) | YES  |     | NULL    |       |
+---------------+---------------+------+-----+---------+-------+
```



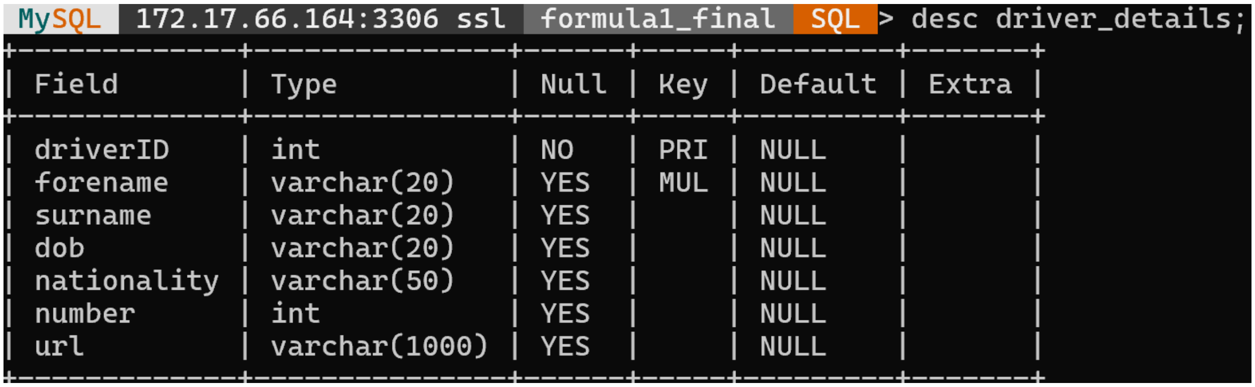
```
MySQL  172.17.66.164:3306 ssl  formula1_final  SQL > desc constructor_ref;
+----------------+--------------+------+-----+---------+-------+
| Field          | Type         | Null | Key | Default | Extra |
+----------------+--------------+------+-----+---------+-------+
| name           | varchar(100) | NO   | PRI | NULL    |       |
| nationality    | varchar(100) | YES  |     | NULL    |       |
| constructorRef | varchar(100) | YES  |     | NULL    |       |
+----------------+--------------+------+-----+---------+-------+
```



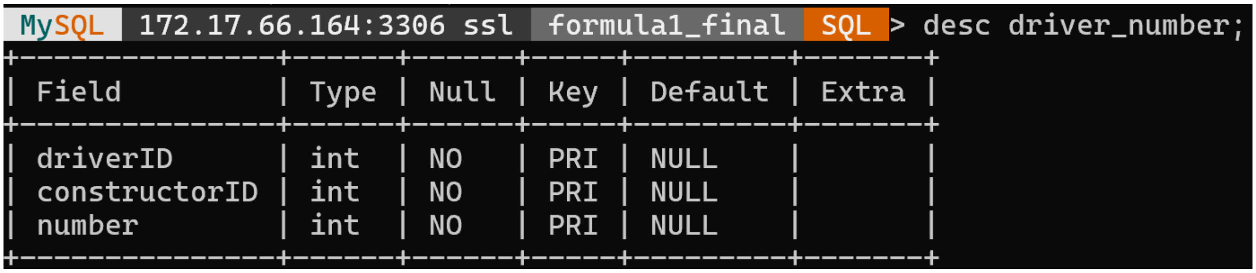
```
MySQL  172.17.66.164:3306 ssl  formula1_final  SQL > desc dates;
+-------+-------------+------+-----+---------+-------+
| Field | Type        | Null | Key | Default | Extra |
+-------+-------------+------+-----+---------+-------+
| date  | varchar(25) | NO   | PRI | NULL    |       |
| year  | varchar(10) | YES  |     | NULL    |       |
+-------+-------------+------+-----+---------+-------+
```



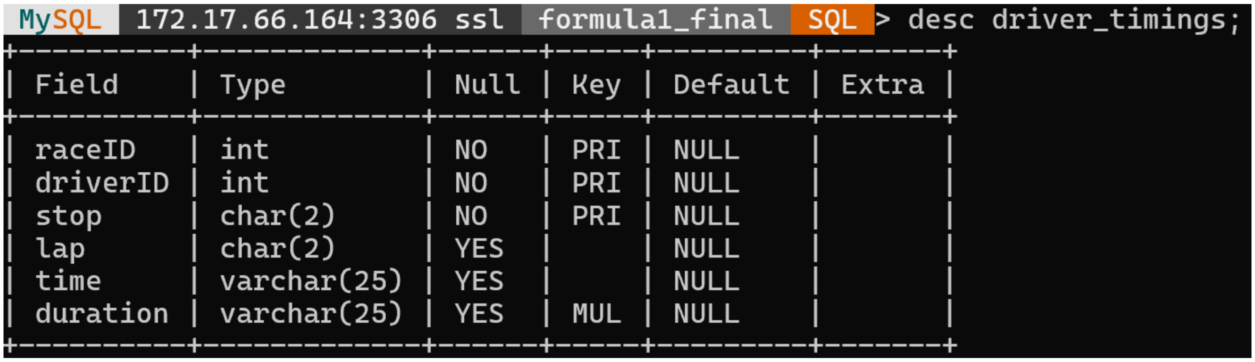
```
MySQL  172.17.66.164:3306 ssl  formula1_final  SQL > desc driverStandings;
+-------------------+------+------+-----+---------+-------+
| Field             | Type | Null | Key | Default | Extra |
+-------------------+------+------+-----+---------+-------+
| driverStandingsID | int  | NO   | PRI | NULL    |       |
| raceID            | int  | YES  |     | NULL    |       |
| driverID          | int  | YES  | MUL | NULL    |       |
| position          | int  | YES  |     | NULL    |       |
+-------------------+------+------+-----+---------+-------+
```



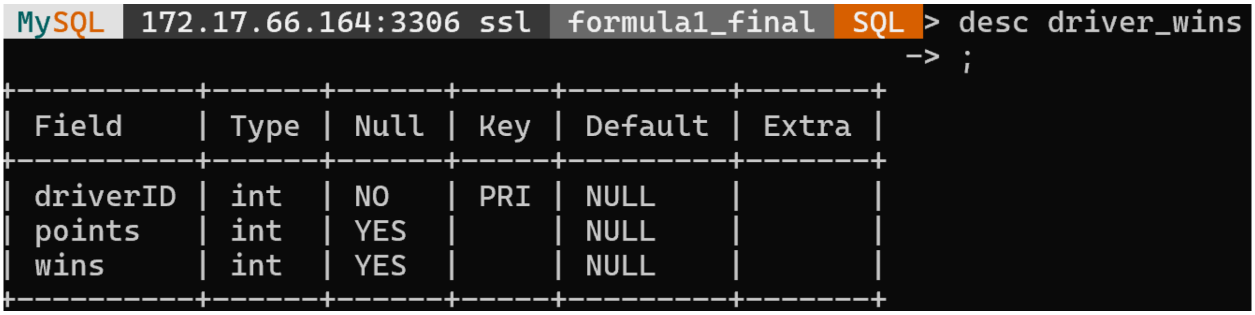
```
MySQL  172.17.66.164:3306 ssl  formula1_final  SQL > desc driver_details;
+-------------+---------------+------+-----+---------+-------+
| Field       | Type          | Null | Key | Default | Extra |
+-------------+---------------+------+-----+---------+-------+
| driverID    | int           | NO   | PRI | NULL    |       |
| forename    | varchar(20)   | YES  | MUL | NULL    |       |
| surname     | varchar(20)   | YES  |     | NULL    |       |
| dob         | varchar(20)   | YES  |     | NULL    |       |
| nationality | varchar(50)   | YES  |     | NULL    |       |
| number      | int           | YES  |     | NULL    |       |
| url         | varchar(1000) | YES  |     | NULL    |       |
+-------------+---------------+------+-----+---------+-------+
```



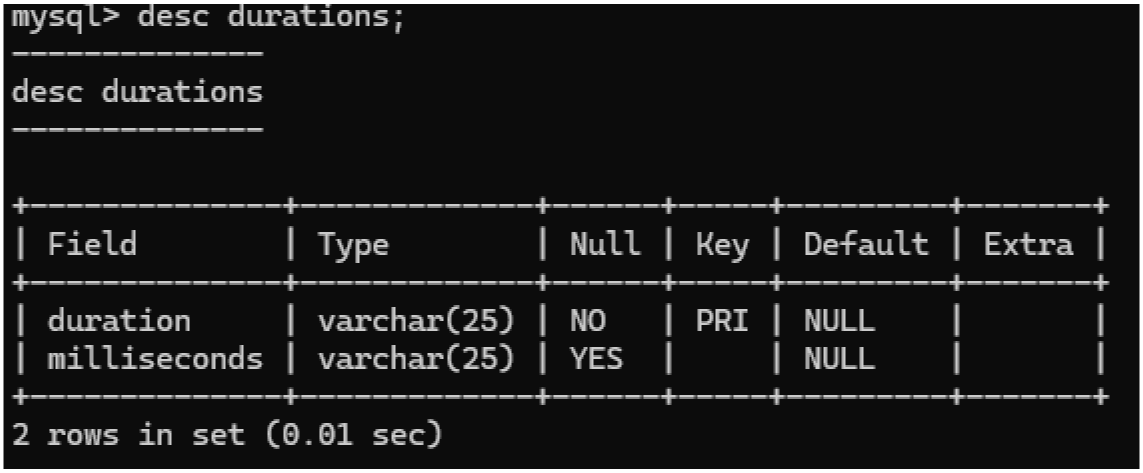
```
MySQL  172.17.66.164:3306 ssl  formula1_final  SQL > desc driver_number;
+---------------+------+------+-----+---------+-------+
| Field         | Type | Null | Key | Default | Extra |
+---------------+------+------+-----+---------+-------+
| driverID      | int  | NO   | PRI | NULL    |       |
| constructorID | int  | NO   | PRI | NULL    |       |
| number        | int  | NO   | PRI | NULL    |       |
+---------------+------+------+-----+---------+-------+
```



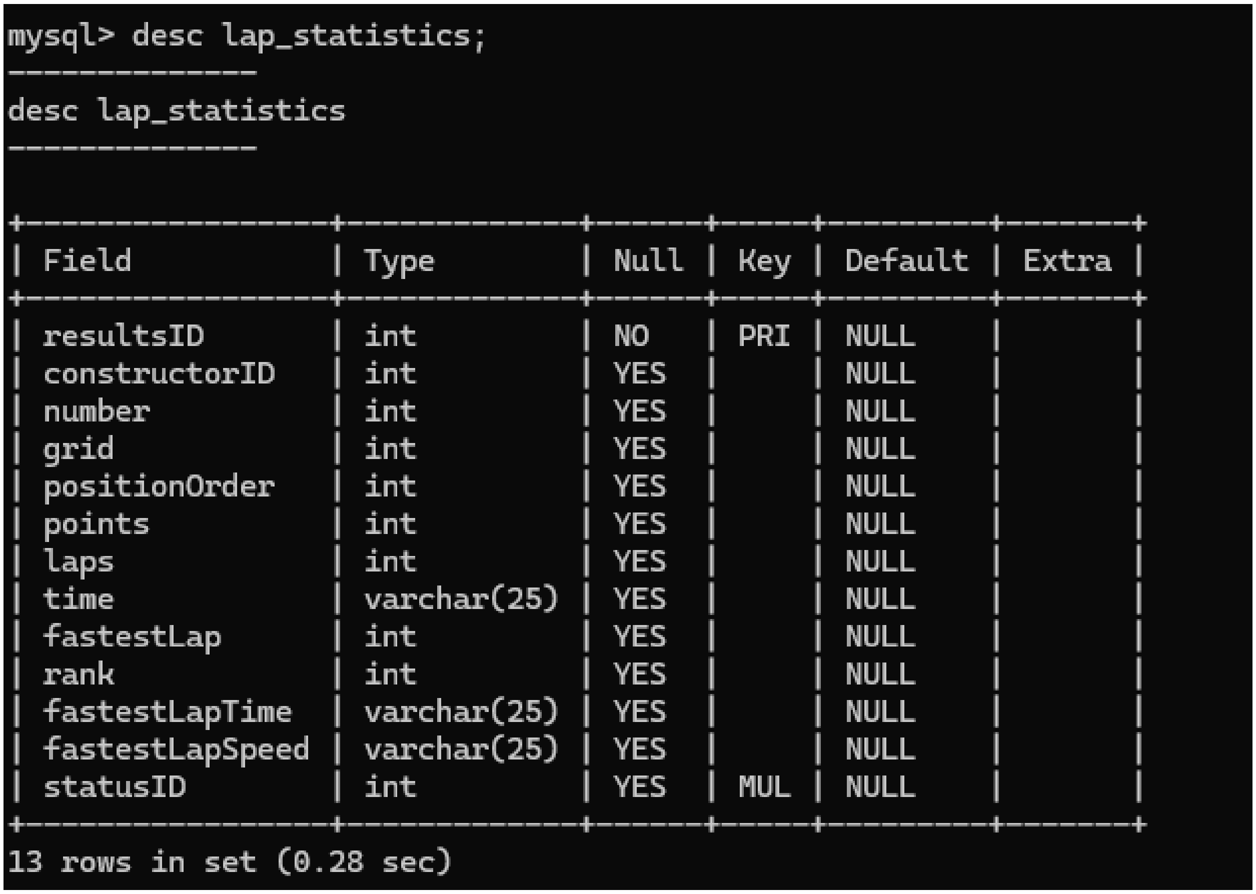
```
MySQL  172.17.66.164:3306 ssl  formula1_final  SQL > desc driver_timings;
+----------+-------------+------+-----+---------+-------+
| Field    | Type        | Null | Key | Default | Extra |
+----------+-------------+------+-----+---------+-------+
| raceID   | int         | NO   | PRI | NULL    |       |
| driverID | int         | NO   | PRI | NULL    |       |
| stop     | char(2)     | NO   | PRI | NULL    |       |
| lap      | char(2)     | YES  |     | NULL    |       |
| time     | varchar(25) | YES  |     | NULL    |       |
| duration | varchar(25) | YES  | MUL | NULL    |       |
+----------+-------------+------+-----+---------+-------+
```



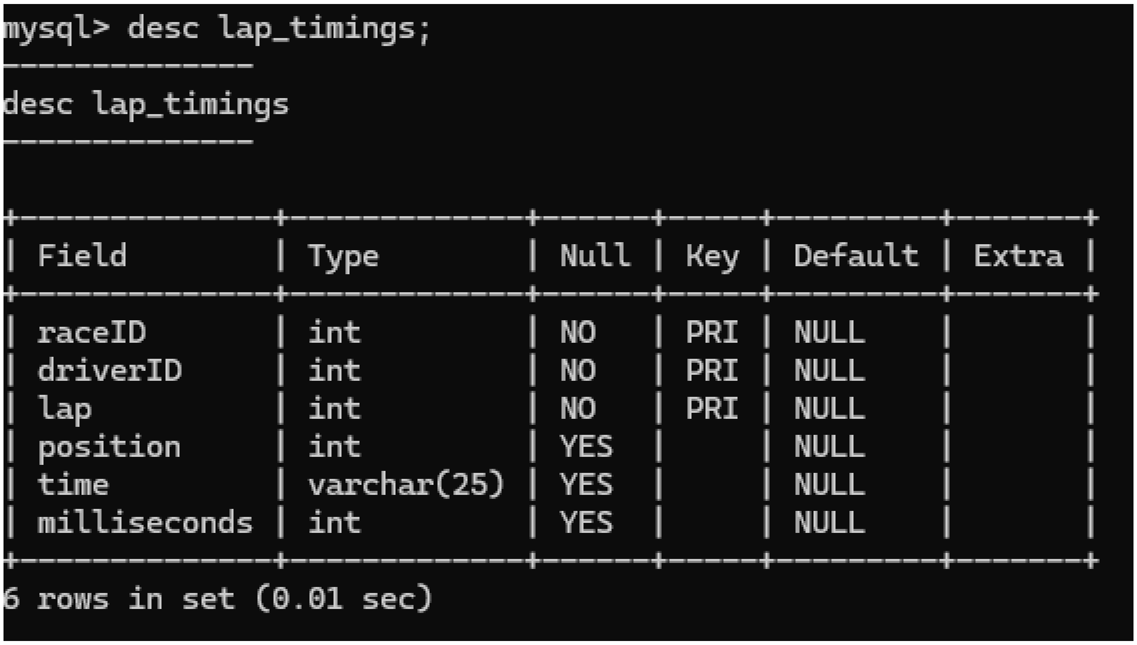
```
MySQL  172.17.66.164:3306 ssl  formula1_final  SQL > desc driver_wins
                                                    -> ;
+----------+------+------+-----+---------+-------+
| Field    | Type | Null | Key | Default | Extra |
+----------+------+------+-----+---------+-------+
| driverID | int  | NO   | PRI | NULL    |       |
| points   | int  | YES  |     | NULL    |       |
| wins     | int  | YES  |     | NULL    |       |
+----------+------+------+-----+---------+-------+
```



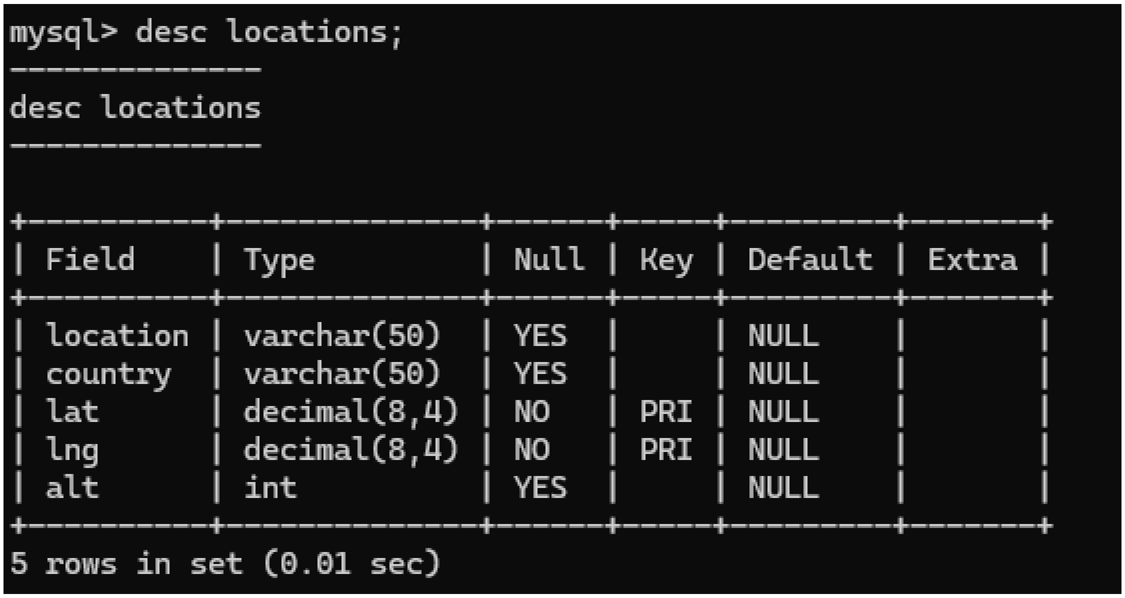
```
mysql> desc durations;
--------------
desc durations
--------------

+--------------+-------------+------+-----+---------+-------+
| Field        | Type        | Null | Key | Default | Extra |
+--------------+-------------+------+-----+---------+-------+
| duration     | varchar(25) | NO   | PRI | NULL    |       |
| milliseconds | varchar(25) | YES  |     | NULL    |       |
+--------------+-------------+------+-----+---------+-------+
2 rows in set (0.01 sec)
```



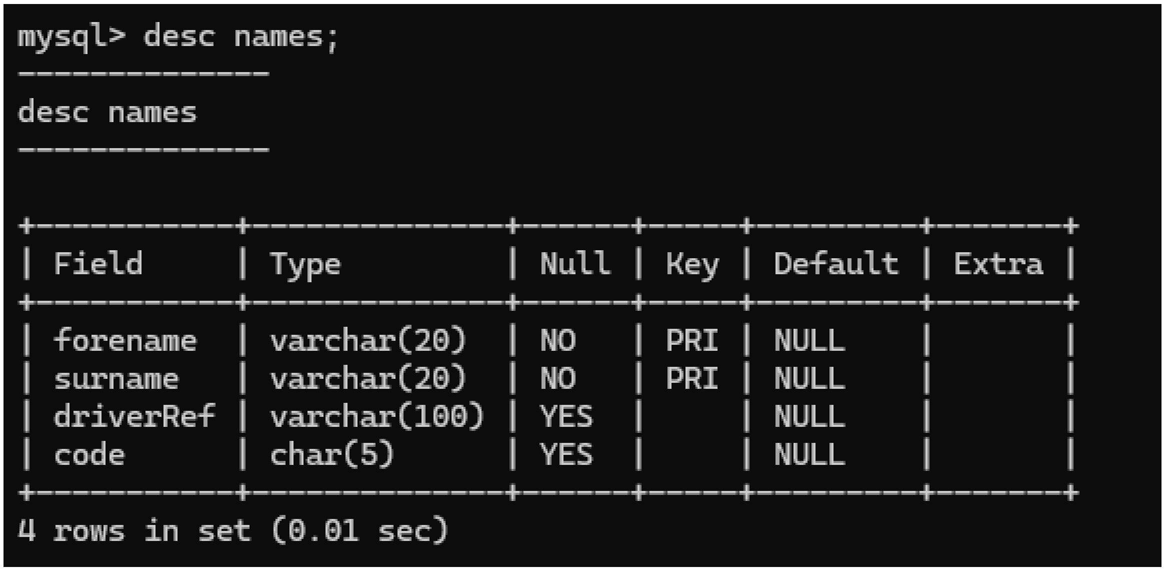
```
mysql> desc lap_statistics;
--------------
desc lap_statistics
--------------

+-----------------+-------------+------+-----+---------+-------+
| Field           | Type        | Null | Key | Default | Extra |
+-----------------+-------------+------+-----+---------+-------+
| resultsID       | int         | NO   | PRI | NULL    |       |
| constructorID   | int         | YES  |     | NULL    |       |
| number          | int         | YES  |     | NULL    |       |
| grid            | int         | YES  |     | NULL    |       |
| positionOrder   | int         | YES  |     | NULL    |       |
| points          | int         | YES  |     | NULL    |       |
| laps            | int         | YES  |     | NULL    |       |
| time            | varchar(25) | YES  |     | NULL    |       |
| fastestLap      | int         | YES  |     | NULL    |       |
| rank            | int         | YES  |     | NULL    |       |
| fastestLapTime  | varchar(25) | YES  |     | NULL    |       |
| fastestLapSpeed | varchar(25) | YES  |     | NULL    |       |
| statusID        | int         | YES  | MUL | NULL    |       |
+-----------------+-------------+------+-----+---------+-------+
13 rows in set (0.28 sec)
```



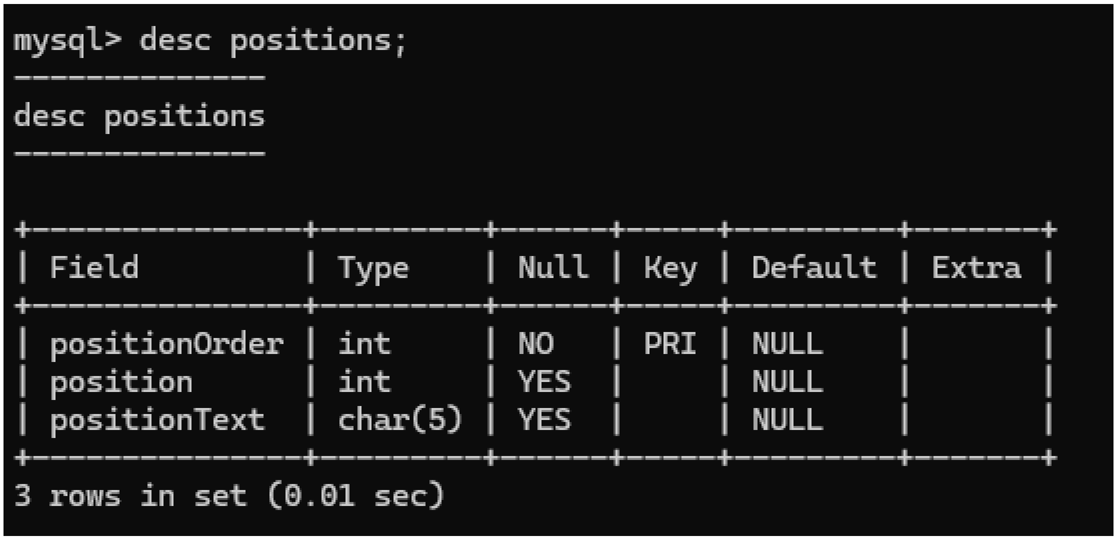
```
mysql> desc lap_timings;
--------------
desc lap_timings
--------------

+--------------+-------------+------+-----+---------+-------+
| Field        | Type        | Null | Key | Default | Extra |
+--------------+-------------+------+-----+---------+-------+
| raceID       | int         | NO   | PRI | NULL    |       |
| driverID     | int         | NO   | PRI | NULL    |       |
| lap          | int         | NO   | PRI | NULL    |       |
| position     | int         | YES  |     | NULL    |       |
| time         | varchar(25) | YES  |     | NULL    |       |
| milliseconds | int         | YES  |     | NULL    |       |
+--------------+-------------+------+-----+---------+-------+
6 rows in set (0.01 sec)
```



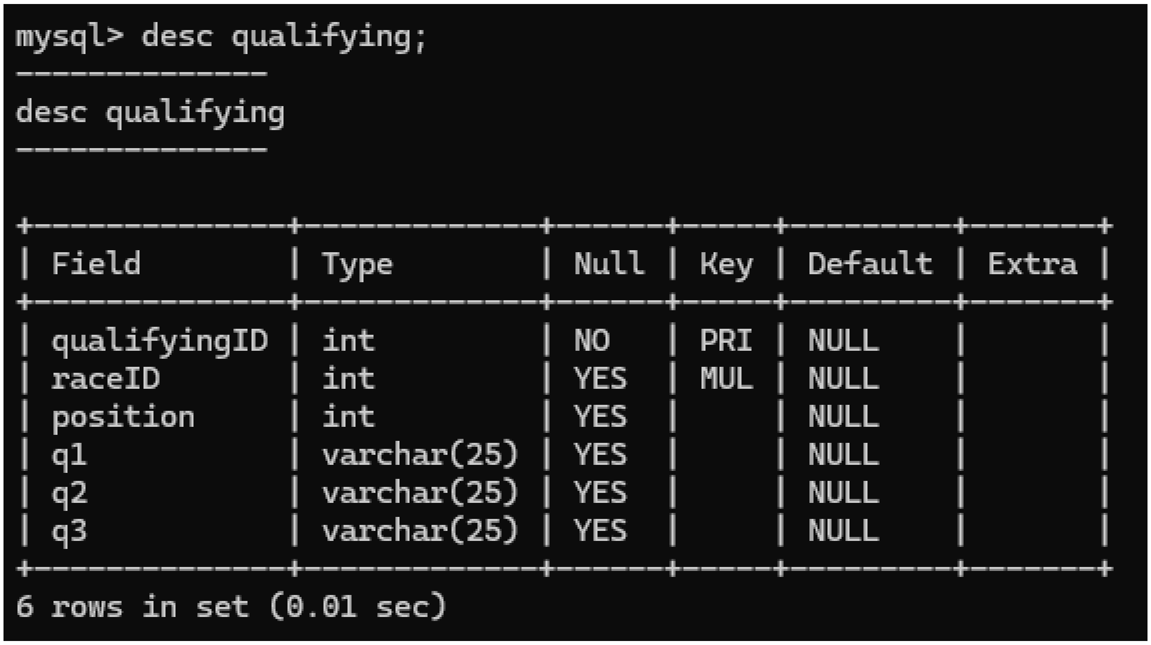
```
mysql> desc locations;
--------------
desc locations
--------------

+----------+--------------+------+-----+---------+-------+
| Field    | Type         | Null | Key | Default | Extra |
+----------+--------------+------+-----+---------+-------+
| location | varchar(50)  | YES  |     | NULL    |       |
| country  | varchar(50)  | YES  |     | NULL    |       |
| lat      | decimal(8,4) | NO   | PRI | NULL    |       |
| lng      | decimal(8,4) | NO   | PRI | NULL    |       |
| alt      | int          | YES  |     | NULL    |       |
+----------+--------------+------+-----+---------+-------+
5 rows in set (0.01 sec)
```



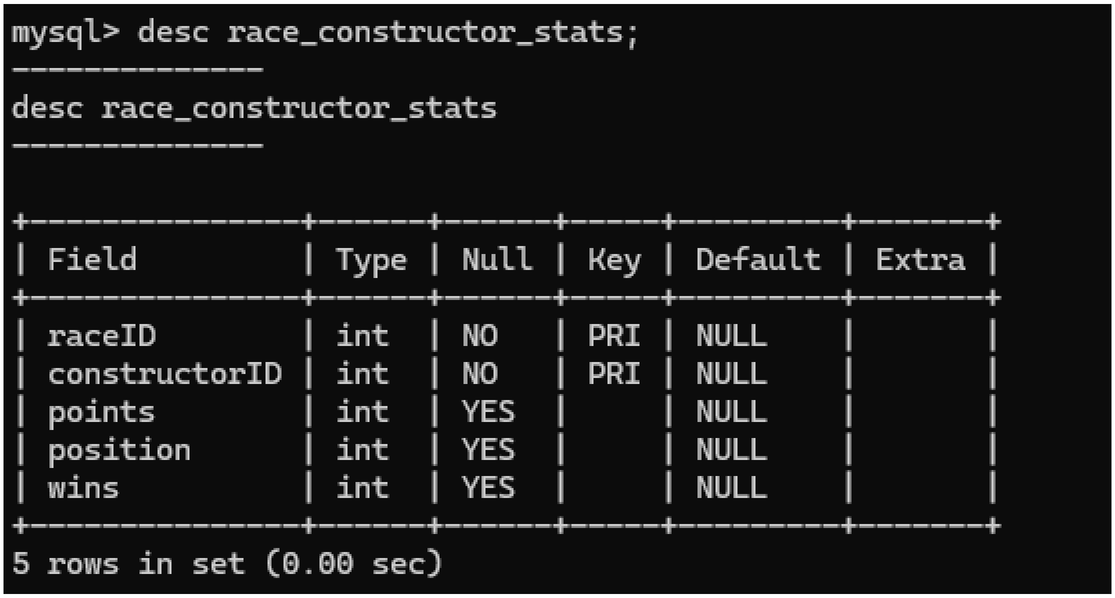
```
mysql> desc names;
--------------
desc names
--------------

+-----------+--------------+------+-----+---------+-------+
| Field     | Type         | Null | Key | Default | Extra |
+-----------+--------------+------+-----+---------+-------+
| forename  | varchar(20)  | NO   | PRI | NULL    |       |
| surname   | varchar(20)  | NO   | PRI | NULL    |       |
| driverRef | varchar(100) | YES  |     | NULL    |       |
| code      | char(5)      | YES  |     | NULL    |       |
+-----------+--------------+------+-----+---------+-------+
4 rows in set (0.01 sec)
```



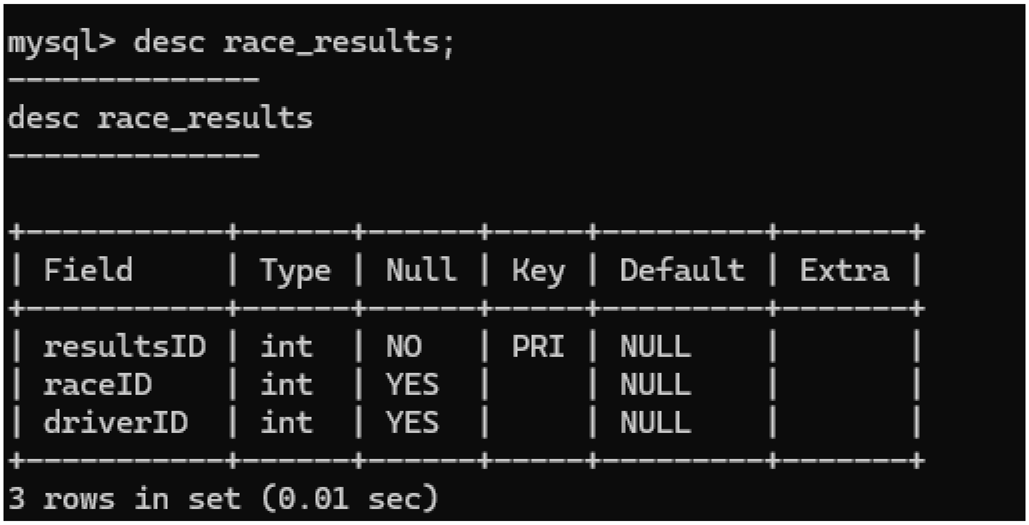
```
mysql> desc positions;
--------------
desc positions
--------------


+---------------+---------+------+-----+---------+-------+
| Field         | Type    | Null | Key | Default | Extra |
+---------------+---------+------+-----+---------+-------+
| positionOrder | int     | NO   | PRI | NULL    |       |
| position      | int     | YES  |     | NULL    |       |
| positionText  | char(5) | YES  |     | NULL    |       |
+---------------+---------+------+-----+---------+-------+
3 rows in set (0.01 sec)
```



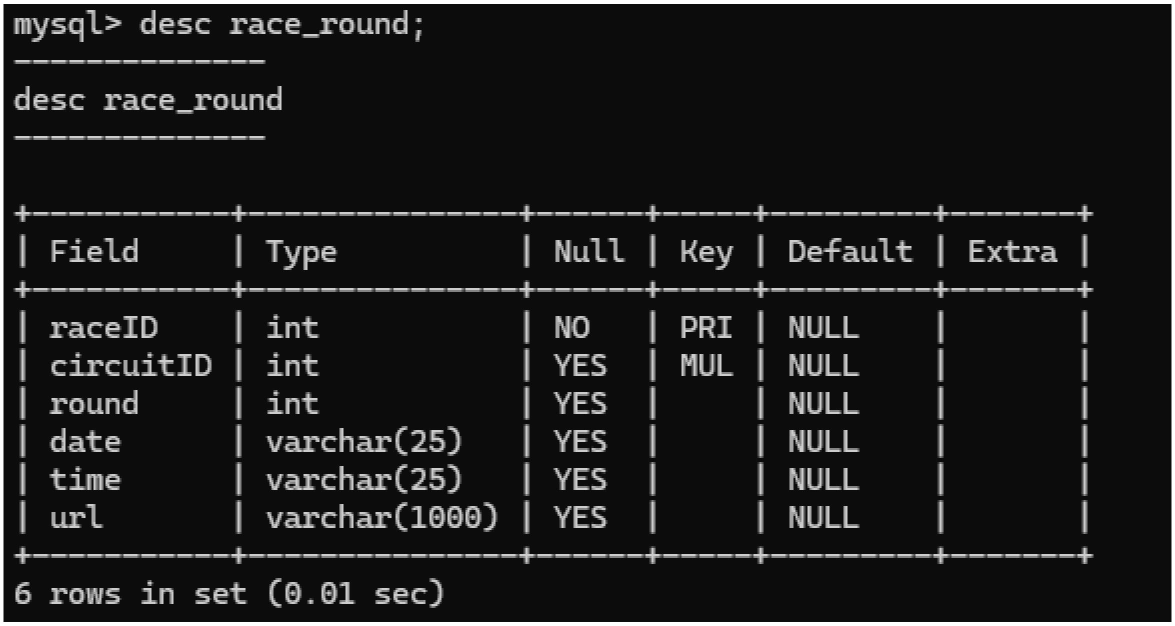
```
mysql> desc qualifying;
--------------
desc qualifying
--------------


+--------------+-------------+------+-----+---------+-------+
| Field        | Type        | Null | Key | Default | Extra |
+--------------+-------------+------+-----+---------+-------+
| qualifyingID | int         | NO   | PRI | NULL    |       |
| raceID       | int         | YES  | MUL | NULL    |       |
| position     | int         | YES  |     | NULL    |       |
| q1           | varchar(25) | YES  |     | NULL    |       |
| q2           | varchar(25) | YES  |     | NULL    |       |
| q3           | varchar(25) | YES  |     | NULL    |       |
+--------------+-------------+------+-----+---------+-------+
6 rows in set (0.01 sec)
```



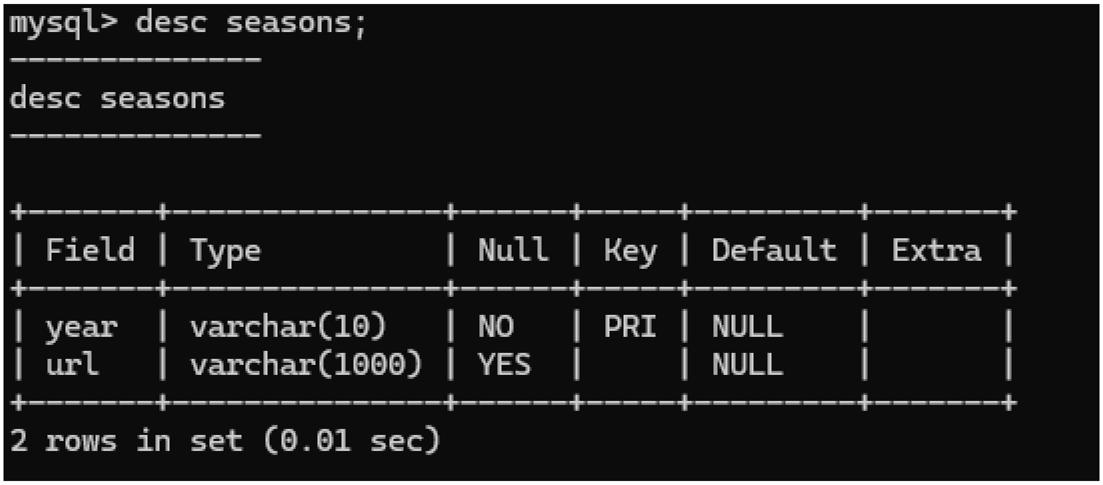
```
mysql> desc race_constructor_stats;
--------------
desc race_constructor_stats
--------------

+---------------+------+------+-----+---------+-------+
| Field         | Type | Null | Key | Default | Extra |
+---------------+------+------+-----+---------+-------+
| raceID        | int  | NO   | PRI | NULL    |       |
| constructorID | int  | NO   | PRI | NULL    |       |
| points        | int  | YES  |     | NULL    |       |
| position      | int  | YES  |     | NULL    |       |
| wins          | int  | YES  |     | NULL    |       |
+---------------+------+------+-----+---------+-------+
5 rows in set (0.00 sec)
```



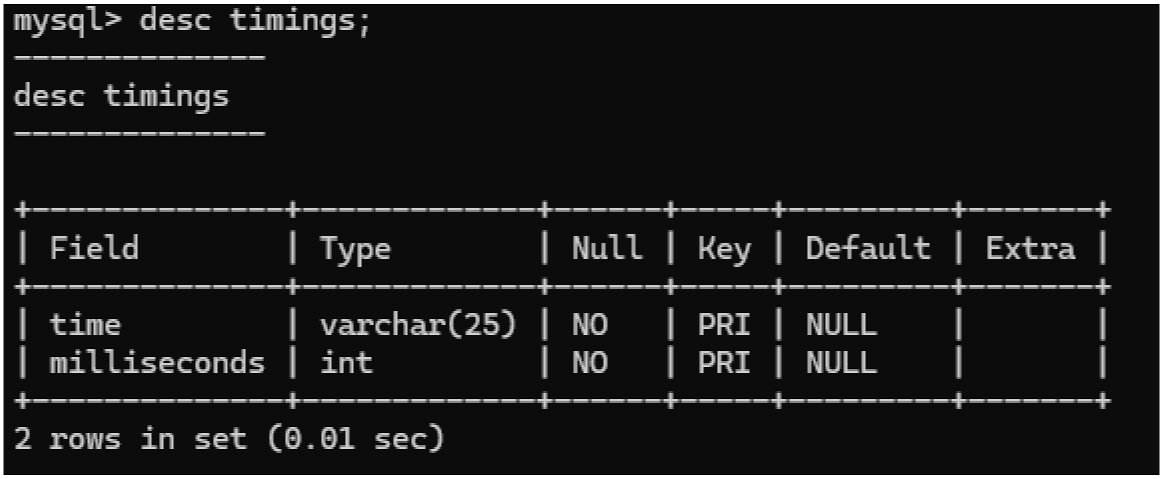
```
mysql> desc race_results;
--------------
desc race_results
--------------

+-----------+------+------+-----+---------+-------+
| Field     | Type | Null | Key | Default | Extra |
+-----------+------+------+-----+---------+-------+
| resultsID | int  | NO   | PRI | NULL    |       |
| raceID    | int  | YES  |     | NULL    |       |
| driverID  | int  | YES  |     | NULL    |       |
+-----------+------+------+-----+---------+-------+
3 rows in set (0.01 sec)
```



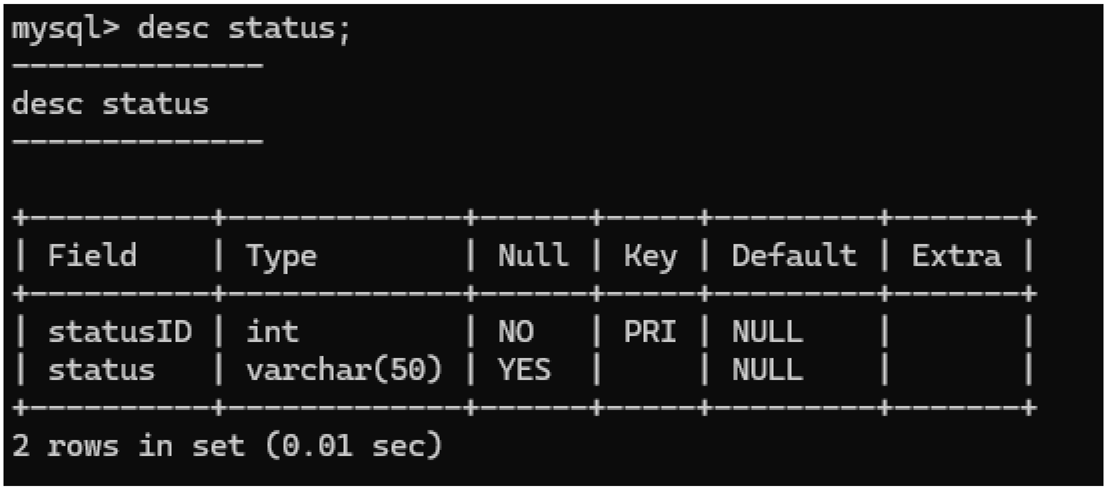
```
mysql> desc race_round;
--------------
desc race_round
--------------


+-----------+---------------+------+-----+---------+-------+
| Field     | Type          | Null | Key | Default | Extra |
+-----------+---------------+------+-----+---------+-------+
| raceID    | int           | NO   | PRI | NULL    |       |
| circuitID | int           | YES  | MUL | NULL    |       |
| round     | int           | YES  |     | NULL    |       |
| date      | varchar(25)   | YES  |     | NULL    |       |
| time      | varchar(25)   | YES  |     | NULL    |       |
| url       | varchar(1000) | YES  |     | NULL    |       |
+-----------+---------------+------+-----+---------+-------+
6 rows in set (0.01 sec)
```



```
mysql> desc seasons;
--------------
desc seasons
--------------


+-------+---------------+------+-----+---------+-------+
| Field | Type          | Null | Key | Default | Extra |
+-------+---------------+------+-----+---------+-------+
| year  | varchar(10)   | NO   | PRI | NULL    |       |
| url   | varchar(1000) | YES  |     | NULL    |       |
+-------+---------------+------+-----+---------+-------+
2 rows in set (0.01 sec)
```

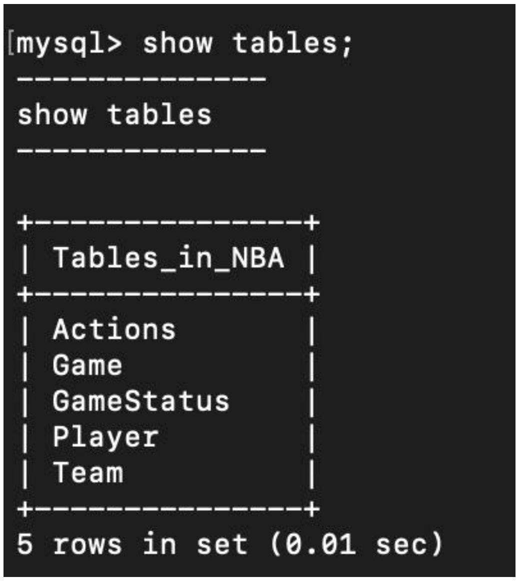


```
mysql> desc timings;
--------------
desc timings
--------------


+--------------+-------------+------+-----+---------+-------+
| Field        | Type        | Null | Key | Default | Extra |
+--------------+-------------+------+-----+---------+-------+
| time         | varchar(25) | NO   | PRI | NULL    |       |
| milliseconds | int         | NO   | PRI | NULL    |       |
+--------------+-------------+------+-----+---------+-------+
2 rows in set (0.01 sec)
```



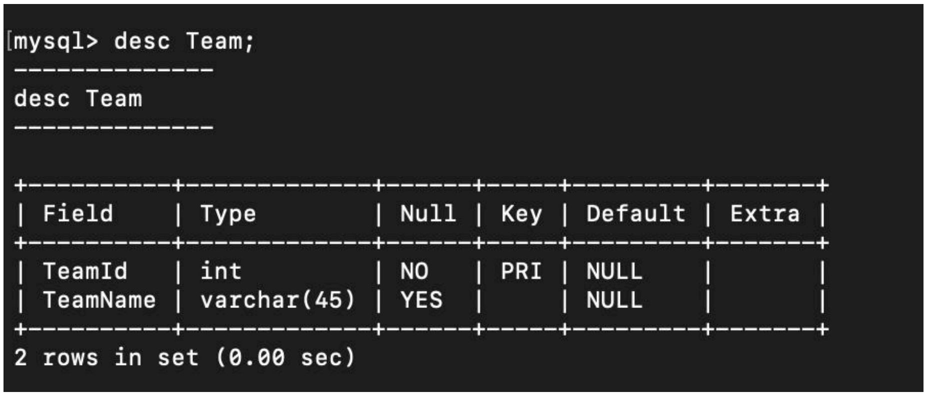
```
mysql> desc status;
--------------
desc status
--------------

+----------+-------------+------+-----+---------+-------+
| Field    | Type        | Null | Key | Default | Extra |
+----------+-------------+------+-----+---------+-------+
| statusID | int         | NO   | PRI | NULL    |       |
| status   | varchar(50) | YES  |     | NULL    |       |
+----------+-------------+------+-----+---------+-------+
2 rows in set (0.01 sec)
```

Tables Created in the NBA database

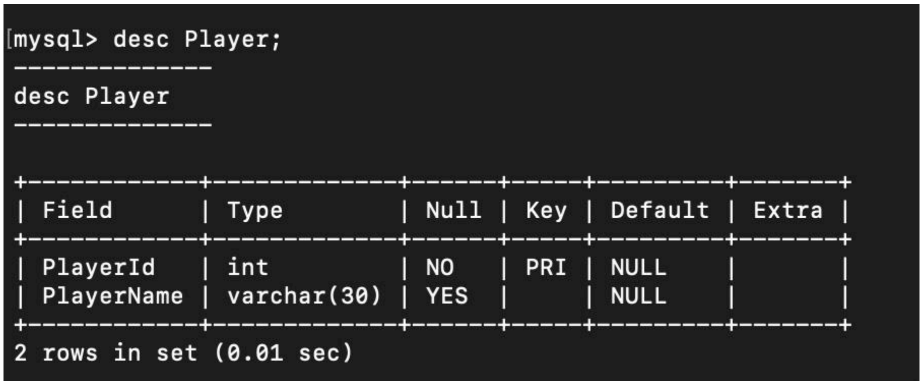


```
[mysql> show tables;
--------------
show tables
--------------

+---------------+
| Tables_in_NBA |
+---------------+
| Actions       |
| Game          |
| GameStatus    |
| Player        |
| Team          |
+---------------+
5 rows in set (0.01 sec)
```

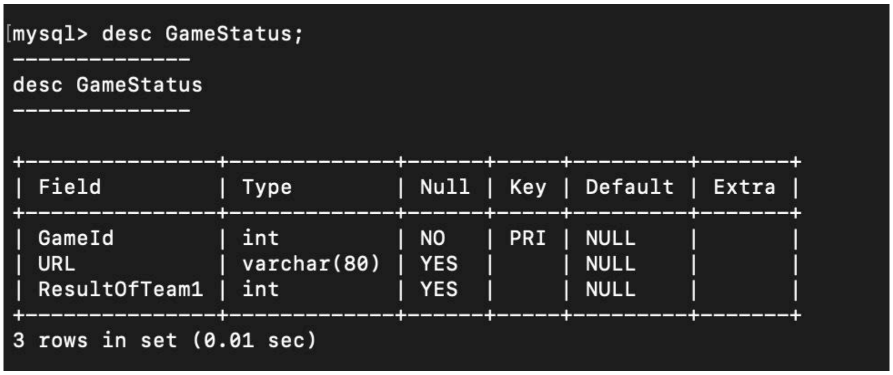
Structure of Tables in NBA



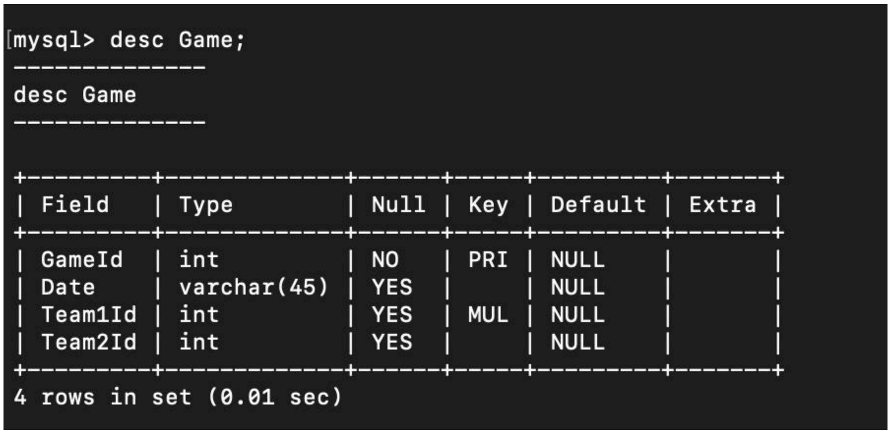
```
[mysql> desc Team;
--------------
desc Team
--------------

+----------+-------------+------+-----+---------+-------+
| Field    | Type        | Null | Key | Default | Extra |
+----------+-------------+------+-----+---------+-------+
| TeamId   | int         | NO   | PRI | NULL    |       |
| TeamName | varchar(45) | YES  |     | NULL    |       |
+----------+-------------+------+-----+---------+-------+
2 rows in set (0.00 sec)
```



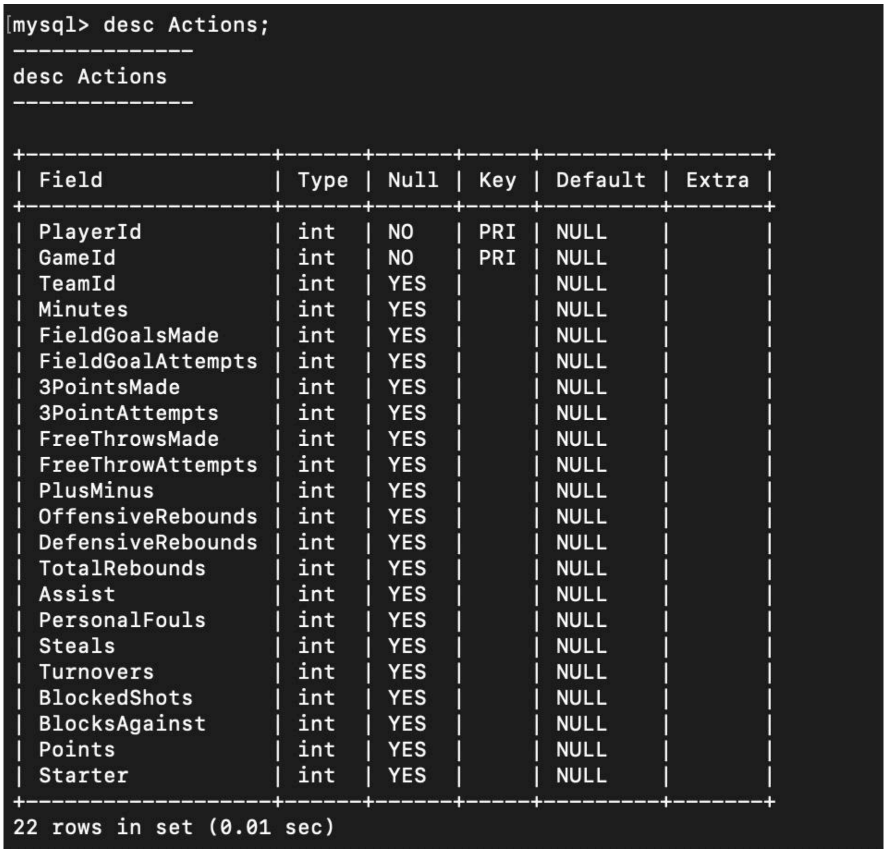
```
[mysql> desc Player;
--------------
desc Player
--------------

+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| PlayerId   | int         | NO   | PRI | NULL    |       |
| PlayerName | varchar(30) | YES  |     | NULL    |       |
+------------+-------------+------+-----+---------+-------+
2 rows in set (0.01 sec)
```



```
[mysql> desc GameStatus;
--------------
desc GameStatus
--------------

+---------------+-------------+------+-----+---------+-------+
| Field         | Type        | Null | Key | Default | Extra |
+---------------+-------------+------+-----+---------+-------+
| GameId        | int         | NO   | PRI | NULL    |       |
| URL           | varchar(80) | YES  |     | NULL    |       |
| ResultOfTeam1 | int         | YES  |     | NULL    |       |
+---------------+-------------+------+-----+---------+-------+
3 rows in set (0.01 sec)
```



```
[mysql> desc Game;
--------------
desc Game
--------------

+----------+-------------+------+-----+---------+-------+
| Field    | Type        | Null | Key | Default | Extra |
+----------+-------------+------+-----+---------+-------+
| GameId   | int         | NO   | PRI | NULL    |       |
| Date     | varchar(45) | YES  |     | NULL    |       |
| Team1Id  | int         | YES  | MUL | NULL    |       |
| Team2Id  | int         | YES  |     | NULL    |       |
+----------+-------------+------+-----+---------+-------+
4 rows in set (0.01 sec)
```

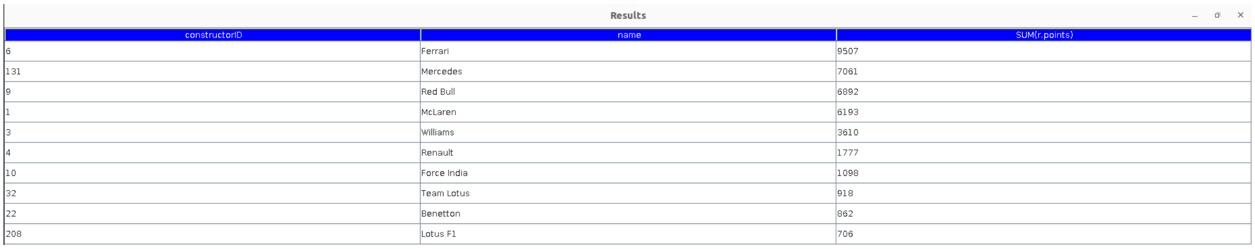


```
[mysql> desc Actions;
--------------
desc Actions
--------------

+-------------------+------+------+-----+---------+-------+
| Field             | Type | Null | Key | Default | Extra |
+-------------------+------+------+-----+---------+-------+
| PlayerId          | int  | NO   | PRI | NULL    |       |
| GameId            | int  | NO   | PRI | NULL    |       |
| TeamId            | int  | YES  |     | NULL    |       |
| Minutes           | int  | YES  |     | NULL    |       |
| FieldGoalsMade    | int  | YES  |     | NULL    |       |
| FieldGoalAttempts | int  | YES  |     | NULL    |       |
| 3PointsMade       | int  | YES  |     | NULL    |       |
| 3PointAttempts    | int  | YES  |     | NULL    |       |
| FreeThrowsMade    | int  | YES  |     | NULL    |       |
| FreeThrowAttempts | int  | YES  |     | NULL    |       |
| PlusMinus         | int  | YES  |     | NULL    |       |
| OffensiveRebounds | int  | YES  |     | NULL    |       |
| DefensiveRebounds | int  | YES  |     | NULL    |       |
| TotalRebounds     | int  | YES  |     | NULL    |       |
| Assist            | int  | YES  |     | NULL    |       |
| PersonalFouls     | int  | YES  |     | NULL    |       |
| Steals            | int  | YES  |     | NULL    |       |
| Turnovers         | int  | YES  |     | NULL    |       |
| BlockedShots      | int  | YES  |     | NULL    |       |
| BlocksAgainst     | int  | YES  |     | NULL    |       |
| Points            | int  | YES  |     | NULL    |       |
| Starter           | int  | YES  |     | NULL    |       |
+-------------------+------+------+-----+---------+-------+
22 rows in set (0.01 sec)
```

Complex Queries related to the Formula1 database

1. Top 10 teams of all time based on points scored:

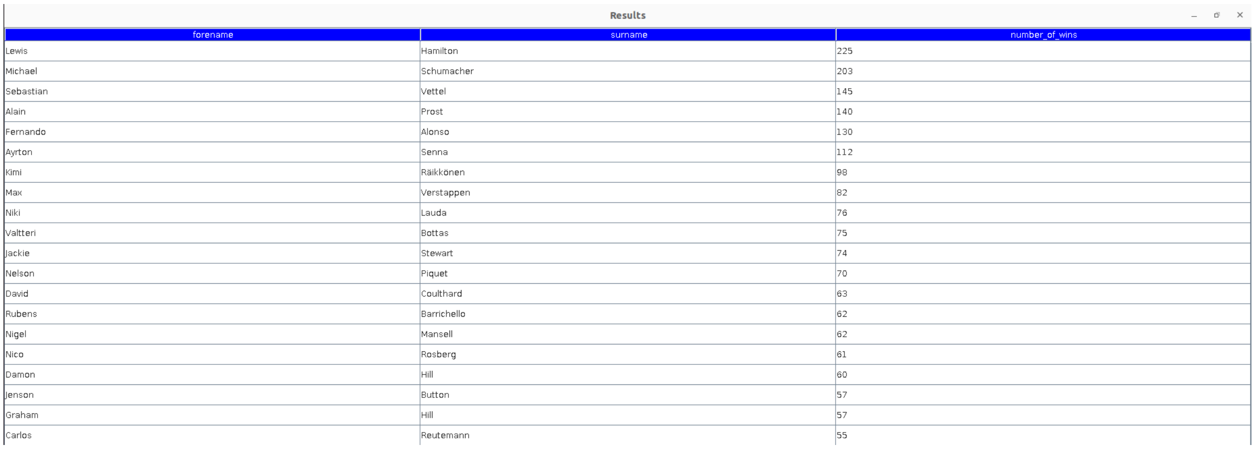
```
SELECT r.constructorID, d.name, SUM(r.points)
FROM constructorResults r, constructor_details d
WHERE r.constructorID = d.constructorID
GROUP BY r.constructorID
ORDER BY SUM(r.points) DESC
LIMIT 10;
```



constructorID	name	SUM(r.points)
6	Ferrari	9507
131	Mercedes	7061
9	Red Bull	6892
1	McLaren	6193
3	Williams	3610
4	Renault	1777
10	Force India	1098
32	Team Lotus	918
22	Benetton	862
208	Lotus F1	706

2. Total number of first position finishes for all the drivers ever (taken individually) :

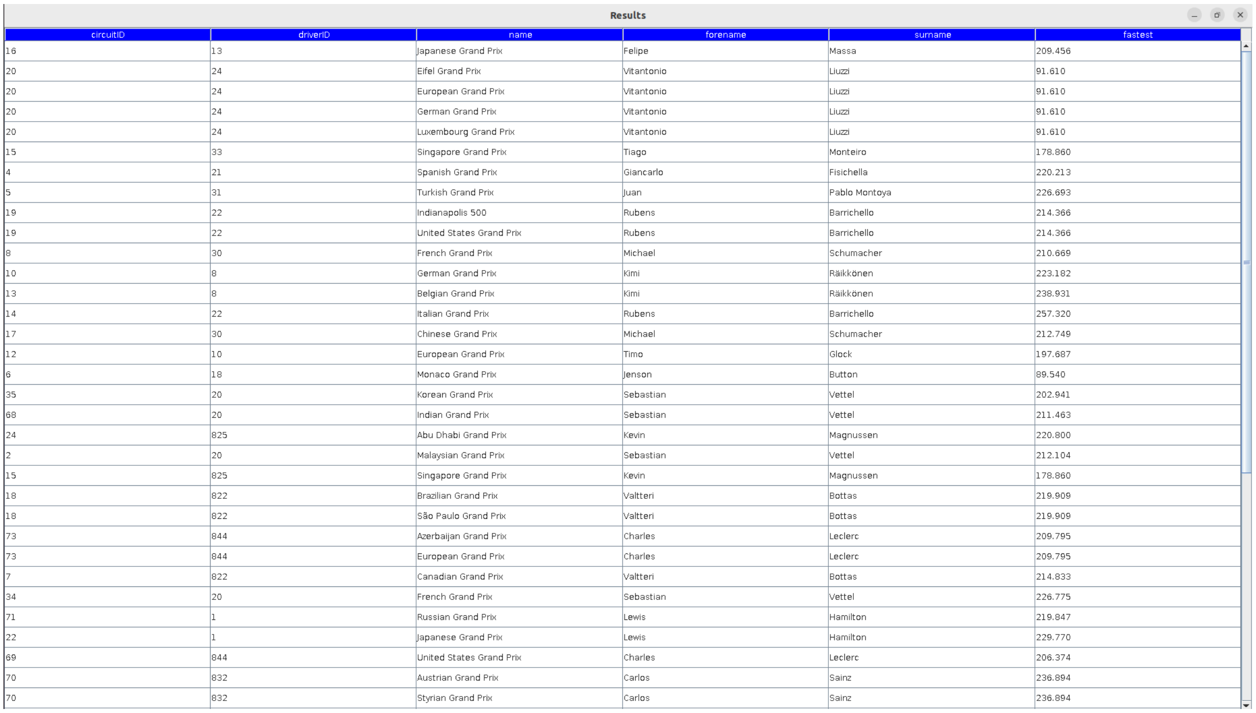
```
select d.forename, d.surname, count(*) as number_of_wins
from driver_details d, driverStandings s
where d.driverID = s.driverID and s.position in ('1', '2', '3')
group by d.forename, d.surname
order by number_of_wins desc;
```



forename	surname	number_of_wins
Lewis	Hamilton	225
Michael	Schumacher	203
Sebastian	Vettel	145
Alain	Prost	140
Fernando	Alonso	130
Ayrton	Senna	112
Kimi	Räikkönen	98
Max	Verstappen	82
Niki	Lauda	76
Valtteri	Bottas	75
Jackie	Stewart	74
Nelson	Piquet	70
David	Coulthard	63
Rubens	Barrichello	62
Nigel	Mansell	62
Nico	Rosberg	61
Damon	Hill	60
Jenson	Button	57
Graham	Hill	57
Carlos	Reutemann	55

3. Driver with the fastest lap time in a particular circuit

```
with final as
    (with fastestspeed as
      (select r.raceID, max(s.fastestLapSpeed) as FLS from
race_results r, lap_statistics s where r.resultsID = s.resultsID
group by r.raceID)
    select round.circuitID, max(f.FLS) as fastest from
race_round round, fastestspeed f  where f.raceID = round.raceID
group by round.circuitID)
select c.circuitID, res.driverID, c.name , d.forename,
d.surname, final.fastest
from circuit_name c, final, race_results res, lap_statistics s,
driver_details d
where c.circuitID = final.circuitID
and res.resultsID = s.resultsID
and s.fastestLapSpeed = final.fastest
and d.driverID = res.driverID;
```

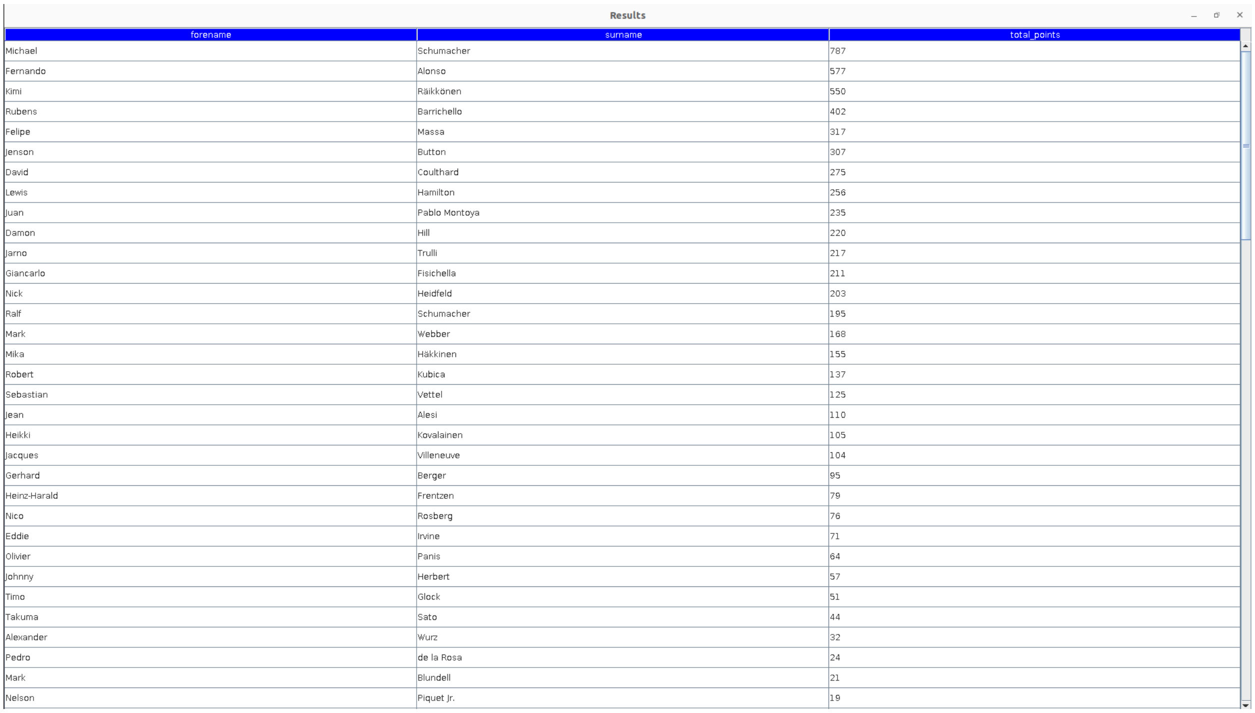


circuitID	driverID	name	forename	surname	fastest
16	13	Japanese Grand Prix	Felipe	Massa	209.456
20	24	Eifel Grand Prix	Vitantonio	Liuzzi	91.610
20	24	European Grand Prix	Vitantonio	Liuzzi	91.610
20	24	German Grand Prix	Vitantonio	Liuzzi	91.610
20	24	Luxembourg Grand Prix	Vitantonio	Liuzzi	91.610
15	33	Singapore Grand Prix	Tiago	Monteiro	178.860
4	21	Spanish Grand Prix	Giancarlo	Fisichella	220.213
5	31	Turkish Grand Prix	Juan	Pablo Montoya	226.693
19	22	Indianapolis 500	Rubens	Barrichello	214.366
19	22	United States Grand Prix	Rubens	Barrichello	214.366
8	30	French Grand Prix	Michael	Schumacher	210.669
10	8	German Grand Prix	Kimi	Räikkönen	223.182
13	8	Belgian Grand Prix	Kimi	Räikkönen	238.931
14	22	Italian Grand Prix	Rubens	Barrichello	257.320
17	30	Chinese Grand Prix	Michael	Schumacher	212.749
12	10	European Grand Prix	Timo	Glock	197.687
6	18	Monaco Grand Prix	Jenson	Button	89.540
35	20	Korean Grand Prix	Sebastian	Vettel	202.941
68	20	Indian Grand Prix	Sebastian	Vettel	211.463
24	825	Abu Dhabi Grand Prix	Kevin	Magnussen	220.800
2	20	Malaysian Grand Prix	Sebastian	Vettel	212.104
15	825	Singapore Grand Prix	Kevin	Magnussen	178.860
18	822	Brazilian Grand Prix	Valtteri	Bottas	219.909
18	822	São Paulo Grand Prix	Valtteri	Bottas	219.909
73	844	Azerbaijan Grand Prix	Charles	Leclerc	209.795
73	844	European Grand Prix	Charles	Leclerc	209.795
7	822	Canadian Grand Prix	Valtteri	Bottas	214.833
34	20	French Grand Prix	Sebastian	Vettel	226.775
71	1	Russian Grand Prix	Lewis	Hamilton	219.847
22	1	Japanese Grand Prix	Lewis	Hamilton	229.770
69	844	United States Grand Prix	Charles	Leclerc	206.374
70	832	Austrian Grand Prix	Carlos	Sainz	236.894
70	832	Styrian Grand Prix	Carlos	Sainz	236.894

(scrollable)

4.Lists the total number of points won by drivers until the old points system was implemented

```
select driver_details.forename as
Firstname,driver_details.surname as
Lastname,sum(lap_statistics.points) as total_points
from lap_statistics,race_results,dates,race_round,driver_details
where race_results.resultsID=lap_statistics.resultsID and
race_results.raceID=race_round.raceID and
race_results.driverID=driver_details.driverID and
race_round.date=dates.date and
dates.year between '1960' AND '2009'
group by driver_details.driverID
order by total_points desc;
```

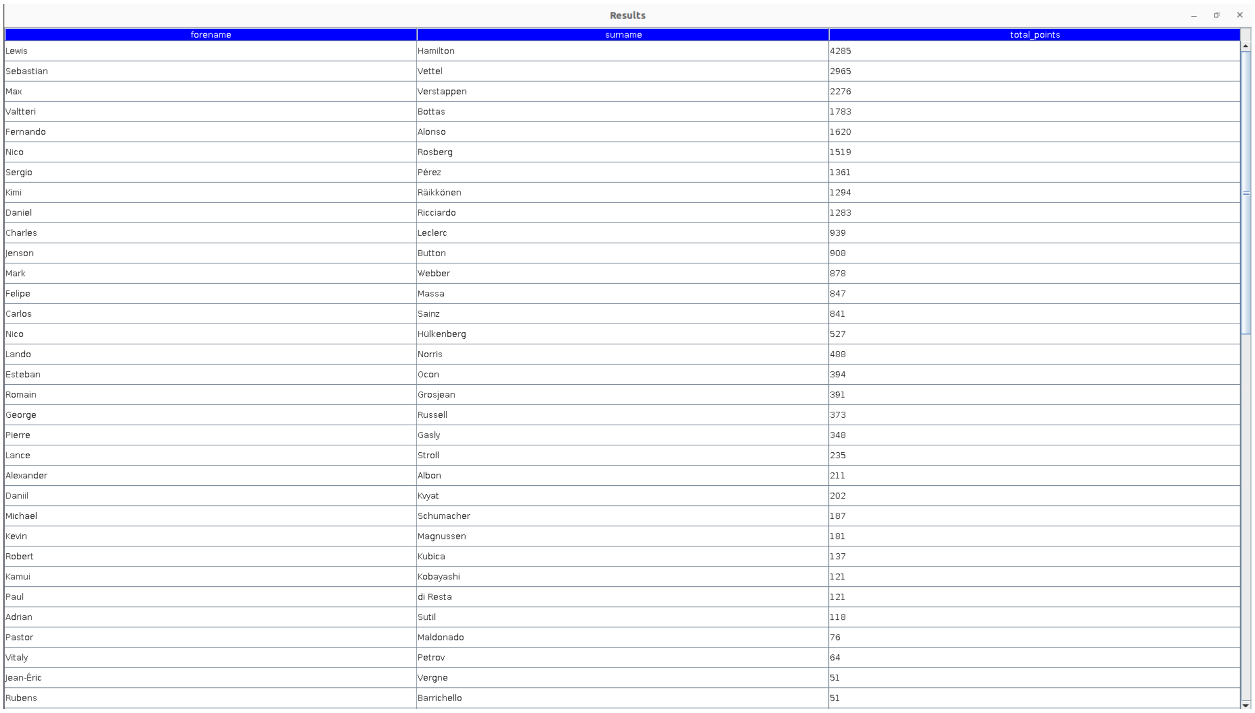


forename	surname	total_points
Michael	Schumacher	787
Fernando	Alonso	577
Kimi	Räikkönen	550
Rubens	Barrichello	402
Felipe	Massa	317
Jenson	Button	307
David	Coulthard	275
Lewis	Hamilton	256
Juan	Pablo Montoya	235
Damon	Hill	220
Jarno	Trulli	217
Giancarlo	Fisichella	211
Nick	Heidfeld	203
Ralf	Schumacher	195
Mark	Webber	168
Mika	Häkkinen	155
Robert	Kubica	137
Sebastian	Vettel	125
Jean	Alesi	110
Heikki	Kovalainen	105
Jacques	Villeneuve	104
Gerhard	Berger	95
Heinz-Harald	Frentzen	79
Nico	Rosberg	76
Eddie	Irvine	71
Olivier	Panis	64
Johnny	Herbert	57
Timo	Glock	51
Takuma	Sato	44
Alexander	Wurz	32
Pedro	de la Rosa	24
Mark	Blundell	21
Nelson	Piquet Jr.	19

(scrollable)

5. Lists the total number of points won by drivers since the new points system was implemented

```
select driver_details.forename as
Firstname,driver_details.surname as
Lastname,sum(lap_statistics.points) as total_points
from lap_statistics,race_results,dates,race_round,driver_details
where race_results.resultsID=lap_statistics.resultsID and
race_results.raceID=race_round.raceID and
race_results.driverID=driver_details.driverID and
race_round.date=dates.date and
dates.year between '2010' AND '2024'
group by driver_details.driverID
order by total_points desc;
```

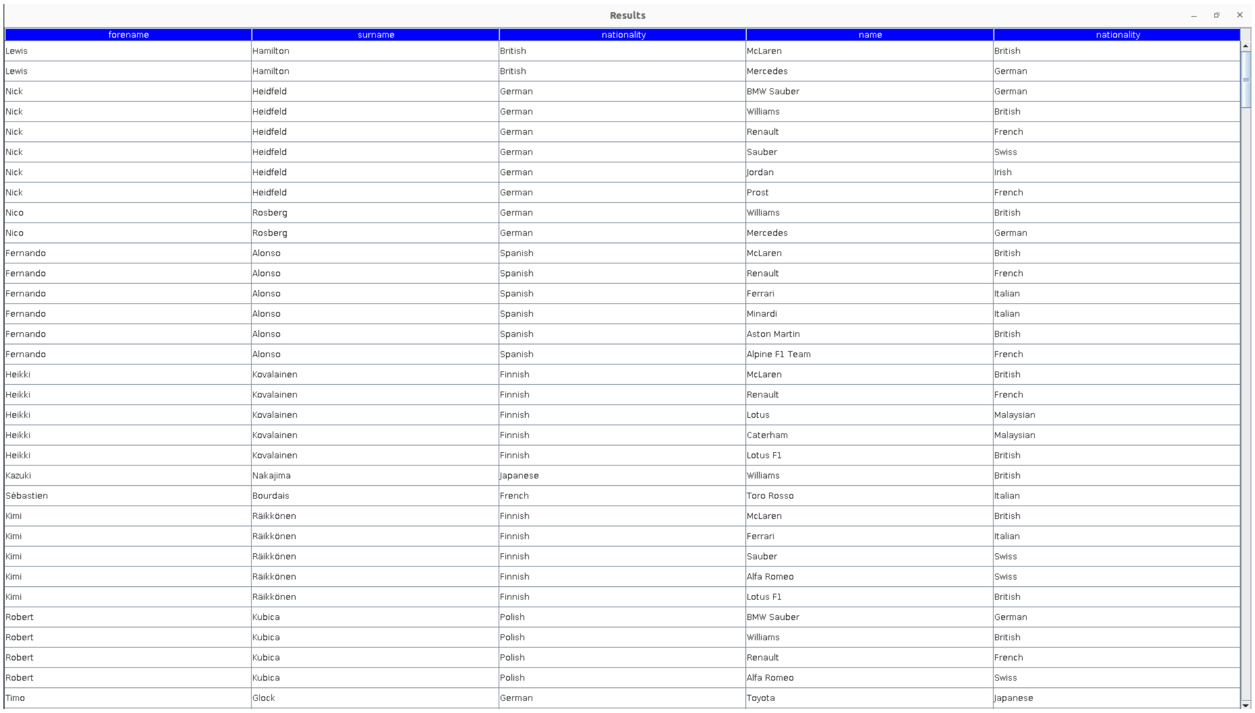


forename	surname	total_points
Lewis	Hamilton	4285
Sebastian	Vettel	2965
Max	Verstappen	2276
Valtteri	Bottas	1783
Fernando	Alonso	1620
Nico	Rosberg	1519
Sergio	Pérez	1361
Kimi	Räikkönen	1294
Daniel	Ricciardo	1283
Charles	Leclerc	939
Jenson	Button	908
Mark	Webber	878
Felipe	Massa	847
Carlos	Sainz	841
Nico	Hülkenberg	527
Lando	Norris	488
Esteban	Ocon	394
Romain	Grosjean	391
George	Russell	373
Pierre	Gasly	348
Lance	Stroll	235
Alexander	Albon	211
Daniil	Kvyat	202
Michael	Schumacher	187
Kevin	Magnussen	181
Robert	Kubica	137
Kamui	Kobayashi	121
Paul	di Resta	121
Adrian	Sutil	118
Pastor	Maldonado	76
Vitaly	Petrov	64
Jean-Éric	Vergne	51
Rubens	Barrichello	51

(scrollable)

6. List all the Drivers and the different constructors that they have paired with

```
select driver_details.forename as
Firstname,driver_details.surname as
Lastname,driver_details.nationality as DriverNationality,
constructor_details.name as
ConstructorName,constructor_ref.nationality as
ConstructorNationality from
driver_details,constructor_details,constructor_ref,lap_statistic
s,race_results
where driver_details.driverID=race_results.driverID and
race_results.resultsID=lap_statistics.resultsID and
lap_statistics.constructorID=constructor_details.constructorID
and
constructor_details.name=constructor_ref.name
group by
driver_details.driverID,constructor_details.constructorID
order by driver_details.driverID asc;
```

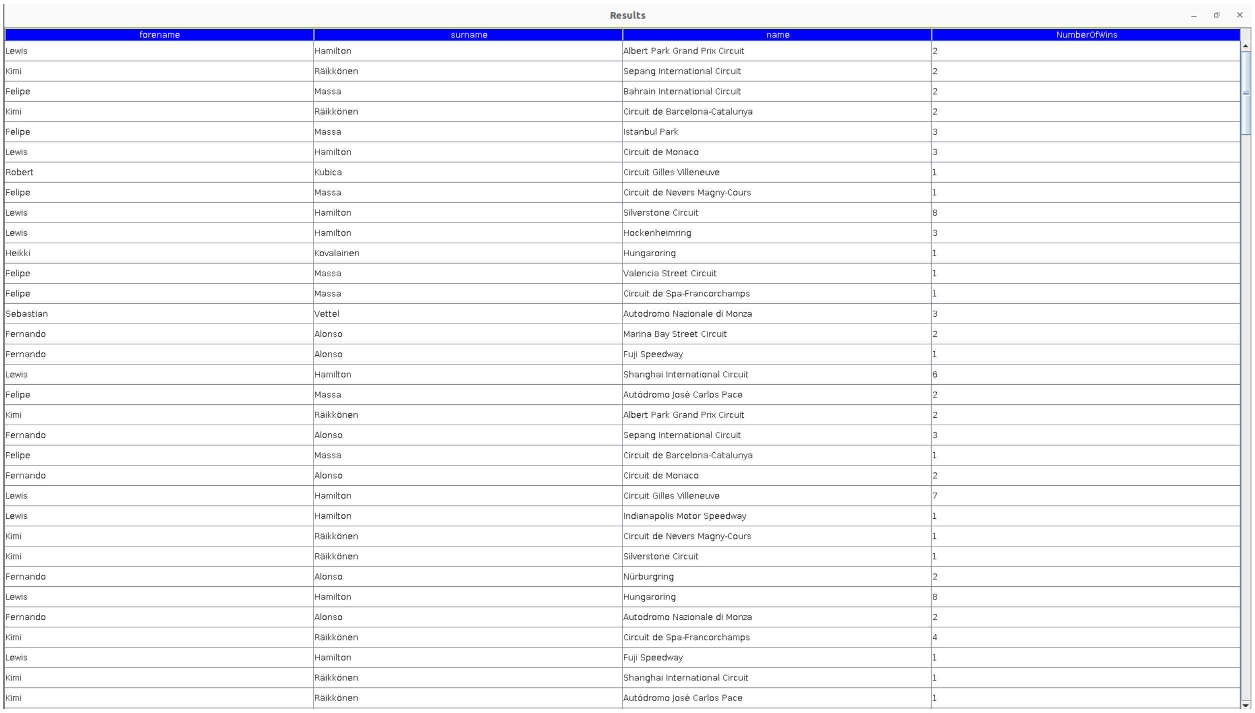


forename	surname	nationality	name	nationality
Lewis	Hamilton	British	McLaren	British
Lewis	Hamilton	British	Mercedes	German
Nick	Heidfeld	German	BMW Sauber	German
Nick	Heidfeld	German	Williams	British
Nick	Heidfeld	German	Renault	French
Nick	Heidfeld	German	Sauber	Swiss
Nick	Heidfeld	German	Jordan	Irish
Nick	Heidfeld	German	Prost	French
Nico	Rosberg	German	Williams	British
Nico	Rosberg	German	Mercedes	German
Fernando	Alonso	Spanish	McLaren	British
Fernando	Alonso	Spanish	Renault	French
Fernando	Alonso	Spanish	Ferrari	Italian
Fernando	Alonso	Spanish	Minardi	Italian
Fernando	Alonso	Spanish	Aston Martin	British
Fernando	Alonso	Spanish	Alpine F1 Team	French
Heikki	Kovalainen	Finnish	McLaren	British
Heikki	Kovalainen	Finnish	Renault	French
Heikki	Kovalainen	Finnish	Lotus	Malaysian
Heikki	Kovalainen	Finnish	Caterham	Malaysian
Heikki	Kovalainen	Finnish	Lotus F1	British
Kazuki	Nakajima	Japanese	Williams	British
Sébastien	Bourdais	French	Toro Rosso	Italian
Kimi	Räikkönen	Finnish	McLaren	British
Kimi	Räikkönen	Finnish	Ferrari	Italian
Kimi	Räikkönen	Finnish	Sauber	Swiss
Kimi	Räikkönen	Finnish	Alfa Romeo	Swiss
Kimi	Räikkönen	Finnish	Lotus F1	British
Robert	Kubica	Polish	BMW Sauber	German
Robert	Kubica	Polish	Williams	British
Robert	Kubica	Polish	Renault	French
Robert	Kubica	Polish	Alfa Romeo	Swiss
Timo	Glock	German	Toyota	Japanese

(scrollable)

7. Breakdown of the number of wins each driver has at each circuit

```
select driver_details.forename as
FirstName,driver_details.surname as LastName,circuit_info.name
as CircuitName,count(*) as NumberOfWins
from
driver_details,circuit_info,lap_statistics,race_results,race_rou
nd
where driver_details.driverID=race_results.driverID and
race_results.resultsID=lap_statistics.resultsID and
race_round.raceID=race_results.raceID and
race_round.circuitID=circuit_info.circuitID and
lap_statistics.positionOrder=1
group by circuit_info.circuitID,driver_details.driverID;
```

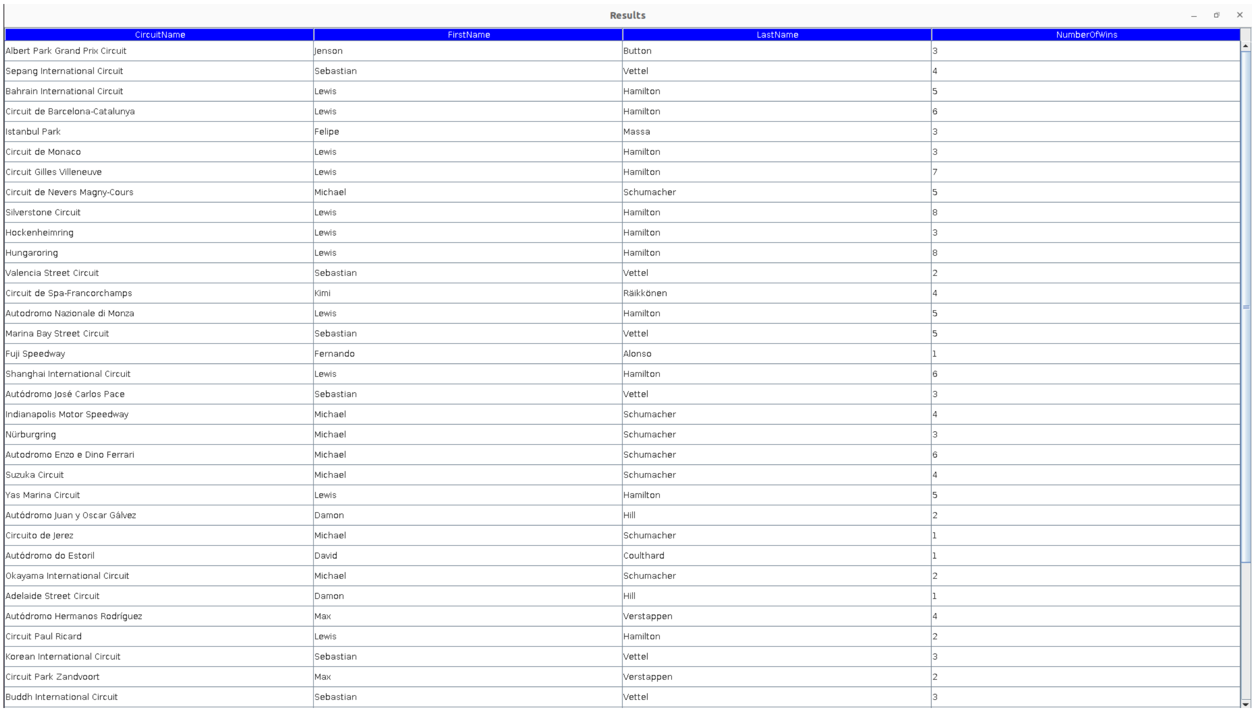


forename	surname	name	NumberOfWins
Lewis	Hamilton	Albert Park Grand Prix Circuit	2
Kimi	Räikkönen	Sepang International Circuit	2
Felipe	Massa	Bahrain International Circuit	2
Kimi	Räikkönen	Circuit de Barcelona-Catalunya	2
Felipe	Massa	Istanbul Park	3
Lewis	Hamilton	Circuit de Monaco	3
Robert	Kubica	Circuit Gilles Villeneuve	1
Felipe	Massa	Circuit de Nevers Magny-Cours	1
Lewis	Hamilton	Silverstone Circuit	8
Lewis	Hamilton	Hockenheimring	3
Heikki	Kovalainen	Hungaroring	1
Felipe	Massa	Valencia Street Circuit	1
Felipe	Massa	Circuit de Spa-Francorchamps	1
Sebastian	Vettel	Autodromo Nazionale di Monza	3
Fernando	Alonso	Marina Bay Street Circuit	2
Fernando	Alonso	Fuji Speedway	1
Lewis	Hamilton	Shanghai International Circuit	6
Felipe	Massa	Autódromo José Carlos Pace	2
Kimi	Räikkönen	Albert Park Grand Prix Circuit	2
Fernando	Alonso	Sepang International Circuit	3
Felipe	Massa	Circuit de Barcelona-Catalunya	1
Fernando	Alonso	Circuit de Monaco	2
Lewis	Hamilton	Circuit Gilles Villeneuve	7
Lewis	Hamilton	Indianapolis Motor Speedway	1
Kimi	Räikkönen	Circuit de Nevers Magny-Cours	1
Kimi	Räikkönen	Silverstone Circuit	1
Fernando	Alonso	Nürburgring	2
Lewis	Hamilton	Hungaroring	8
Fernando	Alonso	Autodromo Nazionale di Monza	2
Kimi	Räikkönen	Circuit de Spa-Francorchamps	4
Lewis	Hamilton	Fuji Speedway	1
Kimi	Räikkönen	Shanghai International Circuit	1
Kimi	Räikkönen	Autódromo José Carlos Pace	1

(scrollable)

8. Driver with the most number of wins at a particular circuit

```
WITH temp AS (
     SELECT
      circuit_info.name AS CircuitName,
     driver_details.forename AS FirstName,
     driver_details.surname AS LastName,
     COUNT(*) AS NumberOfWins,
     ROW_NUMBER() OVER (PARTITION BY circuit_info.circuitID
ORDER BY COUNT(*) DESC) AS rank_final
     FROM
     driver_details, circuit_info, lap_statistics, race_results,
race_round
     WHERE
     driver_details.driverID = race_results.driverID
     AND race_results.resultsID = lap_statistics.resultsID
     AND race_round.raceID = race_results.raceID
     AND race_round.circuitID = circuit_info.circuitID
     AND lap_statistics.positionOrder = 1
     GROUP BY
     circuit_info.circuitID, driver_details.driverID
)
SELECT
    CircuitName,
     FirstName,
     LastName,
     NumberOfWins
FROM
     temp
WHERE
     rank_final = 1;
```



CircuitName	FirstName	LastName	NumberOfWins
Albert Park Grand Prix Circuit	Jenson	Button	3
Sepang International Circuit	Sebastian	Vettel	4
Bahrain International Circuit	Lewis	Hamilton	5
Circuit de Barcelona-Catalunya	Lewis	Hamilton	6
Istanbul Park	Felipe	Massa	3
Circuit de Monaco	Lewis	Hamilton	3
Circuit Gilles Villeneuve	Lewis	Hamilton	7
Circuit de Nevers Magny-Cours	Michael	Schumacher	5
Silverstone Circuit	Lewis	Hamilton	8
Hockenheimring	Lewis	Hamilton	3
Hungaroring	Lewis	Hamilton	8
Valencia Street Circuit	Sebastian	Vettel	2
Circuit de Spa-Francorchamps	Kimi	Räikkönen	4
Autodromo Nazionale di Monza	Lewis	Hamilton	5
Marina Bay Street Circuit	Sebastian	Vettel	5
Fuji Speedway	Fernando	Alonso	1
Shanghai International Circuit	Lewis	Hamilton	6
Autódromo José Carlos Pace	Sebastian	Vettel	3
Indianapolis Motor Speedway	Michael	Schumacher	4
Nürburgring	Michael	Schumacher	3
Autodromo Enzo e Dino Ferrari	Michael	Schumacher	6
Suzuka Circuit	Michael	Schumacher	4
Yas Marina Circuit	Lewis	Hamilton	5
Autódromo Juan y Oscar Gálvez	Damon	Hill	2
Circuito de Jerez	Michael	Schumacher	1
Autódromo do Estoril	David	Coulthard	1
Okayama International Circuit	Michael	Schumacher	2
Adelaide Street Circuit	Damon	Hill	1
Autódromo Hermanos Rodríguez	Max	Verstappen	4
Circuit Paul Ricard	Lewis	Hamilton	2
Korean International Circuit	Sebastian	Vettel	3
Circuit Park Zandvoort	Max	Verstappen	2
Buddh International Circuit	Sebastian	Vettel	3

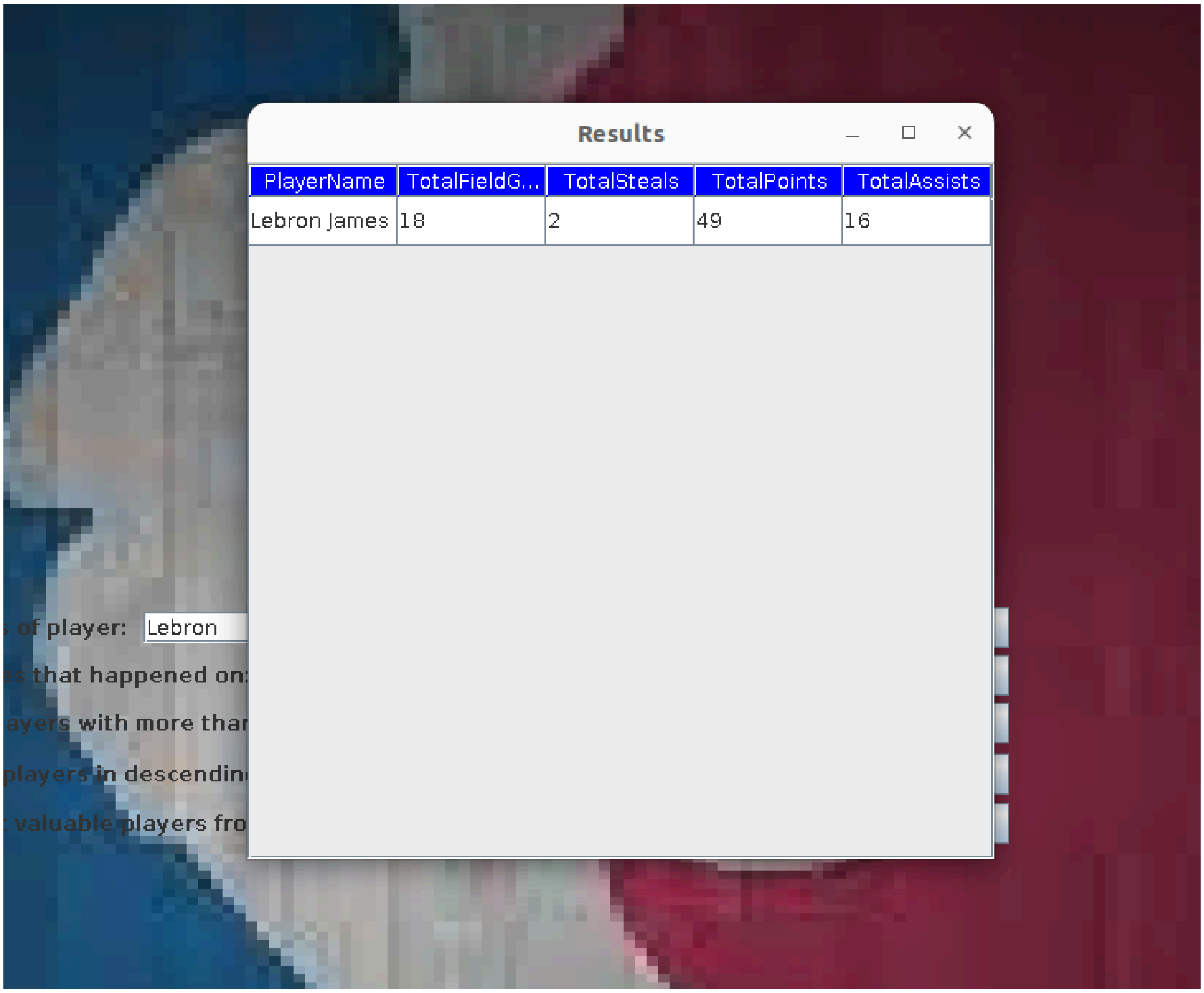
(scrollable)

Complex Queries related to the NBA database

1. Get Statistics of any given Player.

```
SELECT
    Player.PlayerName,
    SUM(Actions.FieldGoalsMade) AS TotalFieldGoalsMade,
    SUM(Actions.Steals) AS TotalSteals,
    SUM(Actions.Points) AS TotalPoints,
    SUM(Actions.Assist) AS TotalAssists
FROM
    Player
INNER JOIN
    Actions ON Player.PlayerId = Actions.PlayerId
WHERE
    Player.PlayerName like ?
```

```
GROUP BY
    Player.PlayerName;
```

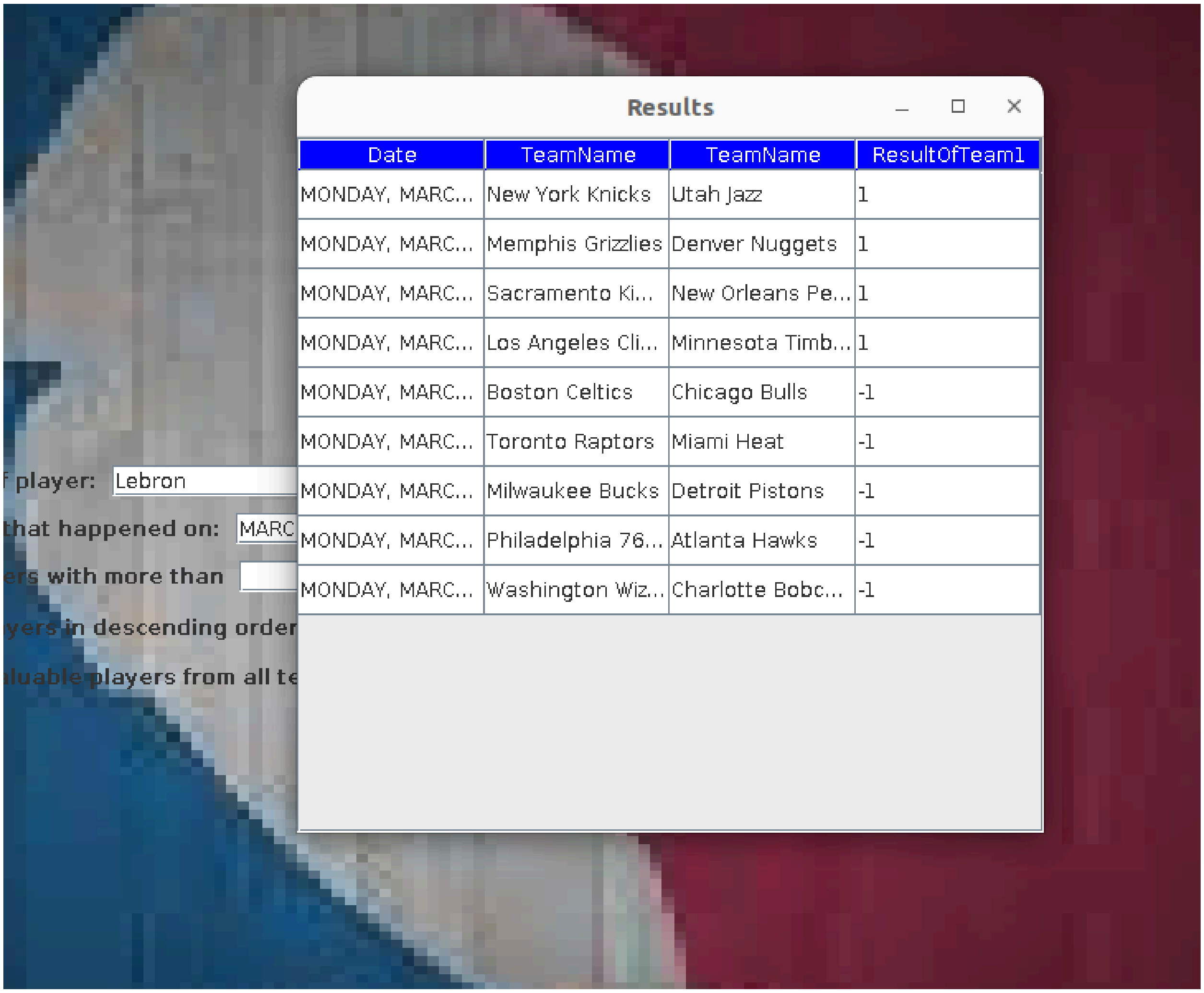


PlayerName	TotalFieldG...	TotalSteals	TotalPoints	TotalAssists
Lebron James	18	2	49	16

2. Which game happened on a particular date?

```
SELECT
    g.Date,
    t1.TeamName AS Team1Name,
    t2.TeamName AS Team2Name,
    gs.ResultOfTeam1
FROM
    Game AS g
INNER JOIN
```

```
    Team AS t1 ON g.Team1Id = t1.TeamId
INNER JOIN
    Team AS t2 ON g.Team2Id = t2.TeamId
LEFT JOIN
    GameStatus AS gs ON g.GameId = gs.GameId
WHERE
    g.Date = ?;
```



Date	TeamName	TeamName	ResultOfTeam1
MONDAY, MARC...	New York Knicks	Utah Jazz	1
MONDAY, MARC...	Memphis Grizzlies	Denver Nuggets	1
MONDAY, MARC...	Sacramento Ki...	New Orleans Pe...	1
MONDAY, MARC...	Los Angeles Cli...	Minnesota Timb...	1
MONDAY, MARC...	Boston Celtics	Chicago Bulls	-1
MONDAY, MARC...	Toronto Raptors	Miami Heat	-1
MONDAY, MARC...	Milwaukee Bucks	Detroit Pistons	-1
MONDAY, MARC...	Philadelphia 76...	Atlanta Hawks	-1
MONDAY, MARC...	Washington Wiz...	Charlotte Bobc...	-1

f player: Lebron

that happened on: MARC

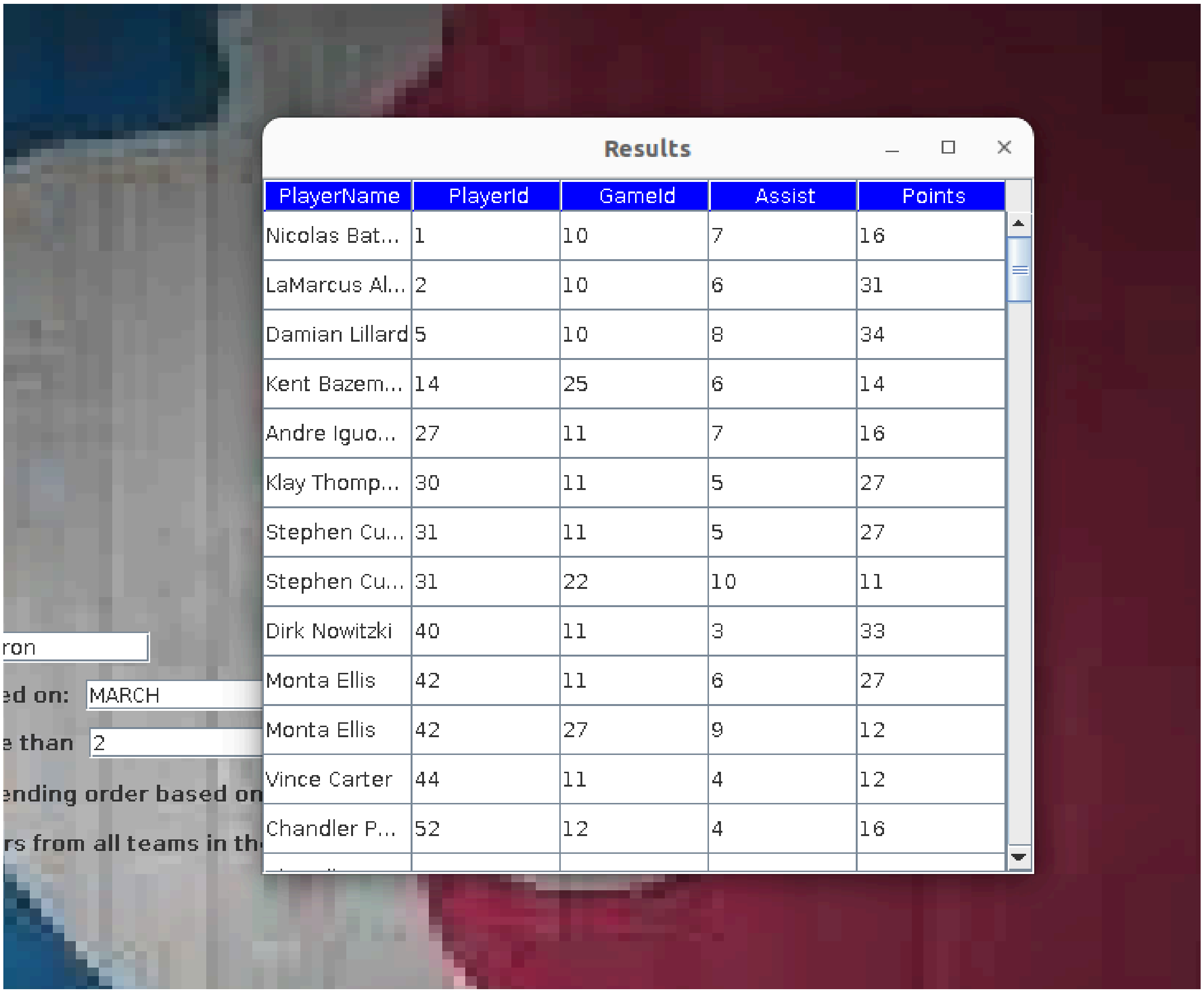
ers with more than

yers in descending order

luable players from all te

3. List all Players with __ plus Points and __ plus Assists in a Game.

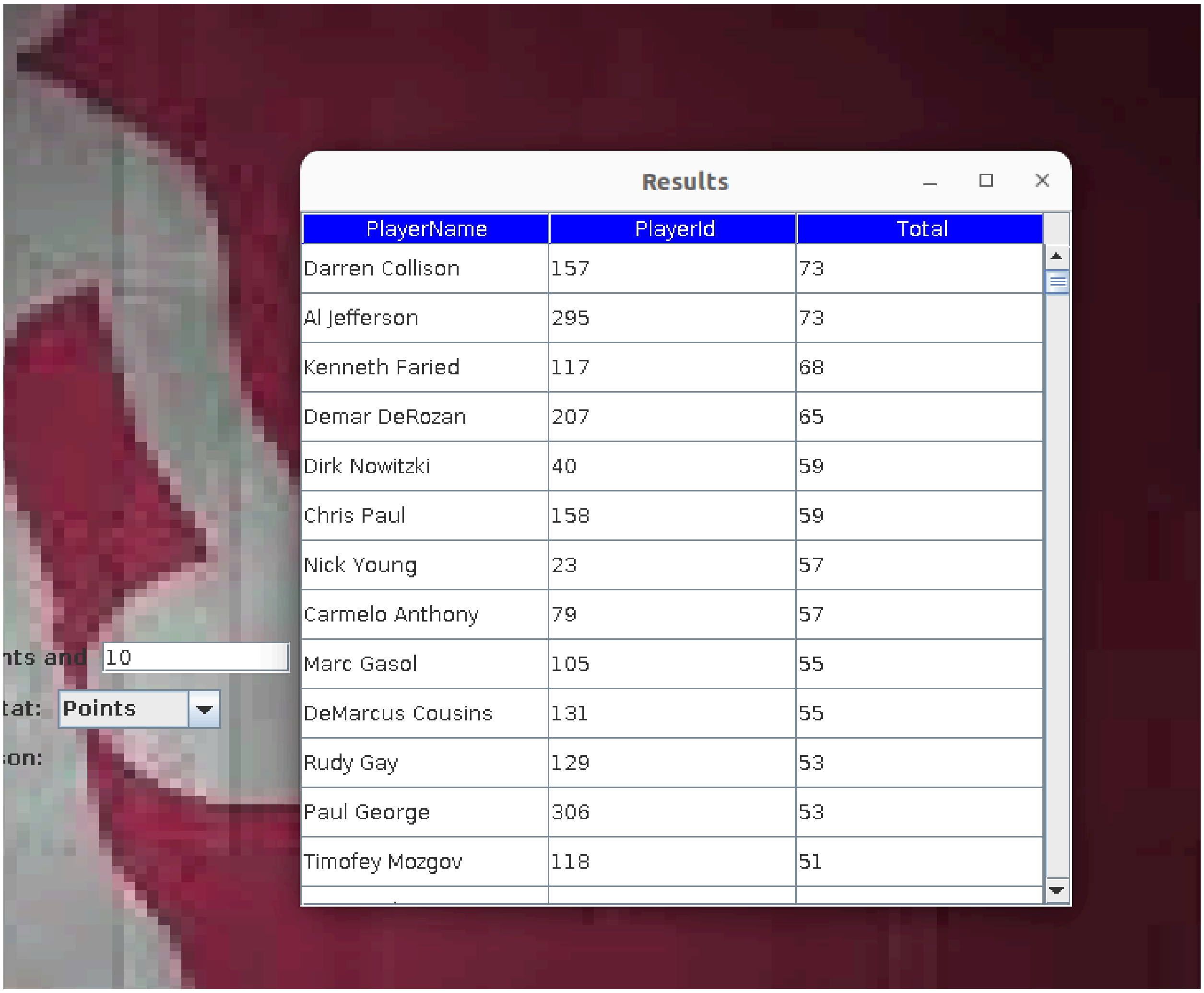
```
select Player.PlayerName,
Actions.PlayerId,Actions.GameId,Actions.Assist,Actions.Points
from Player,Actions
where Actions.PlayerId = Player.PlayerId
and Assist > ?
and Points > ?;
```



PlayerName	PlayerId	GameId	Assist	Points
Nicolas Bat...	1	10	7	16
LaMarcus Al...	2	10	6	31
Damian Lillard	5	10	8	34
Kent Bazem...	14	25	6	14
Andre Iguo...	27	11	7	16
Klay Thomp...	30	11	5	27
Stephen Cu...	31	11	5	27
Stephen Cu...	31	22	10	11
Dirk Nowitzki	40	11	3	33
Monta Ellis	42	11	6	27
Monta Ellis	42	27	9	12
Vince Carter	44	11	4	12
Chandler P...	52	12	4	16

4. Arrange all the players in descending order on the basis of ___ stat

```
select Player.PlayerName,
Actions.PlayerId, sum(Actions.?) as Totalfrom Player,Actions
where Player.PlayerId = Actions.PlayerId
Group by Actions.PlayerId
order by Total desc;
```



nts and 10

tat: Points

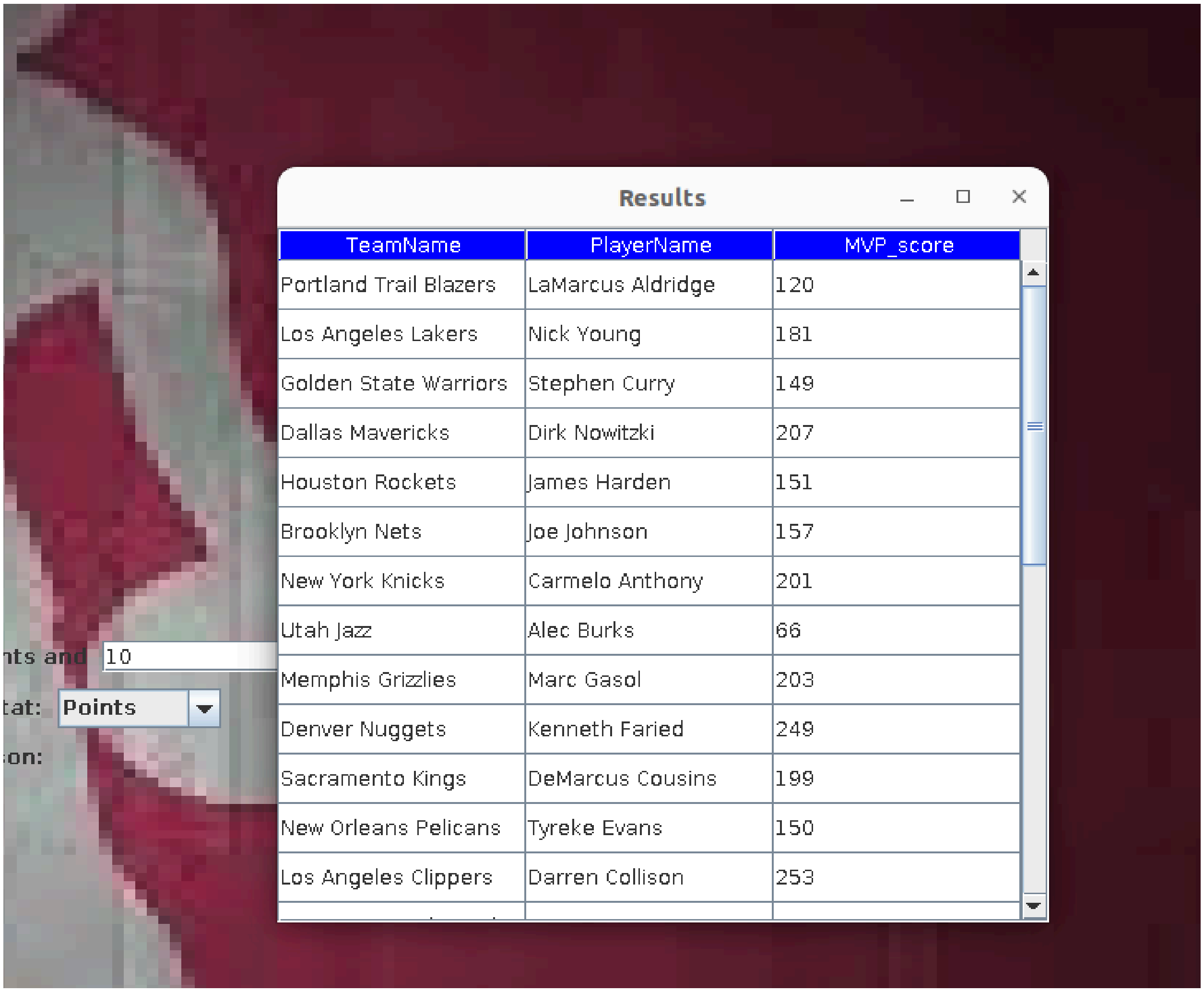
on:

Results

PlayerName	PlayerId	Total
Darren Collison	157	73
Al Jefferson	295	73
Kenneth Faried	117	68
Demar DeRozan	207	65
Dirk Nowitzki	40	59
Chris Paul	158	59
Nick Young	23	57
Carmelo Anthony	79	57
Marc Gasol	105	55
DeMarcus Cousins	131	55
Rudy Gay	129	53
Paul George	306	53
Timofey Mozgov	118	51

5. List Most-Valuable Players(MVP) from all Teams in the Season

```
WITH temp AS (
     SELECT
     Team.TeamName,
     Player.PlayerName,
     (SUM(Actions.Points) * 3) + (SUM(Actions.Assist) * 2) +
(SUM(Actions.TotalRebounds) * 1) AS MVP_score,
     ROW_NUMBER() OVER (PARTITION BY Team.TeamId ORDER BY
(SUM(Actions.Points) * 3) + (SUM(Actions.Assist) * 2) +
(SUM(Actions.TotalRebounds) * 1) DESC) AS 'rank_final'
     FROM
     Team, Player, Actions
     WHERE
     Player.PlayerId = Actions.PlayerId
     AND Team.TeamId = Actions.TeamId
     GROUP BY
     Team.TeamId, Player.PlayerId
)
SELECT
     TeamName,
     PlayerName,
     MVP_score
FROM
     temp
WHERE
     rank_final = 1
```



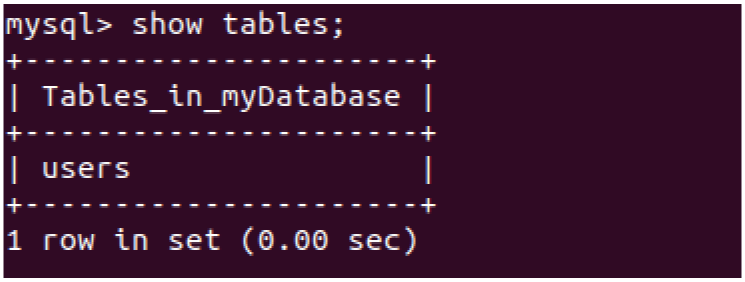
TeamName	PlayerName	MVP_score
Portland Trail Blazers	LaMarcus Aldridge	120
Los Angeles Lakers	Nick Young	181
Golden State Warriors	Stephen Curry	149
Dallas Mavericks	Dirk Nowitzki	207
Houston Rockets	James Harden	151
Brooklyn Nets	Joe Johnson	157
New York Knicks	Carmelo Anthony	201
Utah Jazz	Alec Burks	66
Memphis Grizzlies	Marc Gasol	203
Denver Nuggets	Kenneth Faried	249
Sacramento Kings	DeMarcus Cousins	199
New Orleans Pelicans	Tyreke Evans	150
Los Angeles Clippers	Darren Collison	253

User Login System:
We created a user login system for our application, which allows only signed-up users to use our application.

The details about the user such as ‘username’ and ‘password’ are stored in a database called ‘myDatabase’.

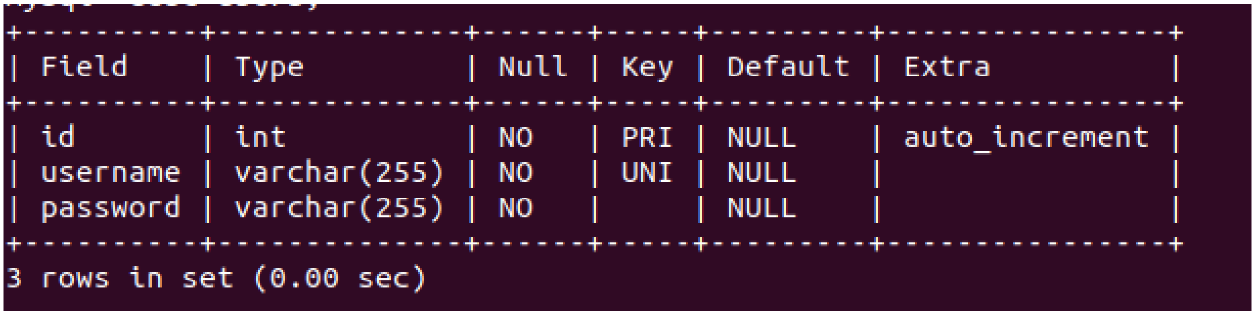
We used SHA 256 encoding to make sure the password is protected.

Any user can either login, sign in to a new account or delete his/her existing account when our application is started.



```
mysql> show tables;
+----------------------+
| Tables_in_myDatabase |
+----------------------+
| users                |
+----------------------+
1 row in set (0.00 sec)
```

users creation query



```
+----------+--------------+------+-----+---------+----------------+
| Field    | Type         | Null | Key | Default | Extra          |
+----------+--------------+------+-----+---------+----------------+
| id       | int          | NO   | PRI | NULL    | auto_increment |
| username | varchar(255) | NO   | UNI | NULL    |                |
| password | varchar(255) | NO   |     | NULL    |                |
+----------+--------------+------+-----+---------+----------------+
3 rows in set (0.00 sec)
```

create table users (
id int primary key auto_increment,
username varchar(255) unique,
password varchar(255));

Insertion query into the table is done through the JDBC connection with the help of an insert query like :

insert into users (username, password) values (?, SHA(?, 256));

over here '?' represents the corresponding text that the user enters in the Jswing pop up box.

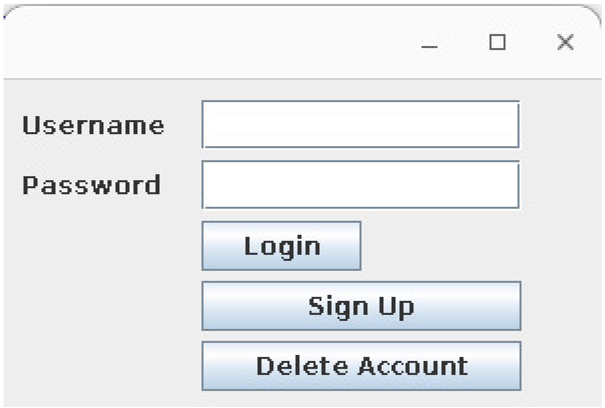
NBA PAGE STRUCTURE

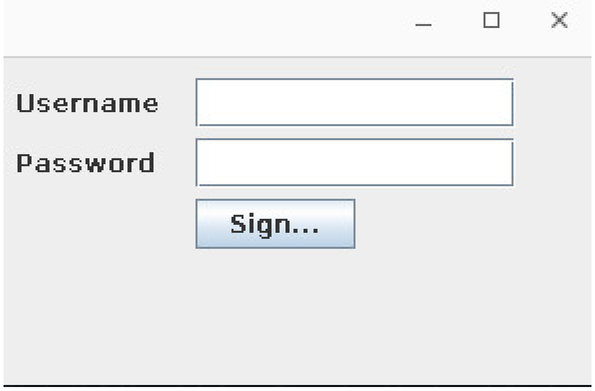
We have taken 5 SQL queries for the NBA database. Here we are using a blank box to fill it with a variable.

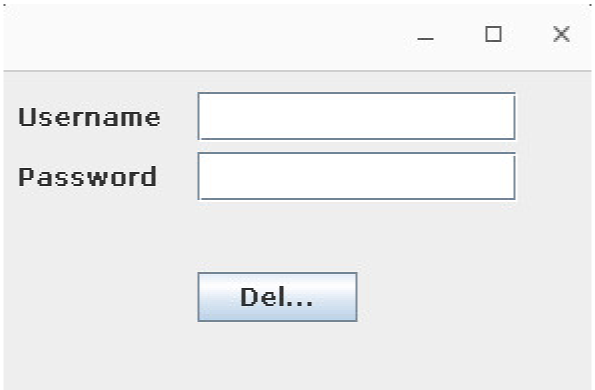
1. First query returns the stats of the player which the user gives as an input.

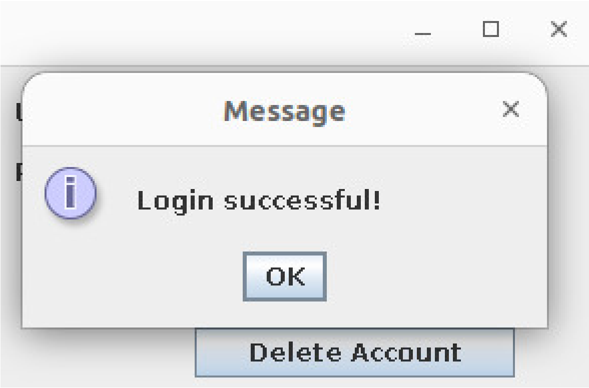
2. Second query returns all the games that happened on the particular date decided by the user.
3. Third query returns all the players with points more than “A” and assists more than “B”. A and B are taken as an input from the user on the page.
4. Fourth query returns a list which arranges players in a descending order based on the stat - “A”. “A” is decided by the user. For example: “A” can be TotalRebounds, Assists, DefensiveRebounds,etc.
5. Fifth query returns a List of the Most Valuable Player(MVP) from all teams. The MVP-score is calculated as the sum of Total points, Total assists and Total rebounds by the player with points having 3, assists having 2 and rebounds having 1 MVP-points.

WORKING OF INSERT AND DELETE QUERIES









Login successful!