

COVER PAGE:

Name: Nuha Imran

ID:20696366

Documentation:

a) Introduction

The Formula One racing championship is held every year, with a series of races determining the winner in terms of constructors and drivers. This is a season 19 database that has all the information that a person would need to learn about the championship, including all the drivers, statistics, victories, race dates, and so on. It also includes some extra information for people who are fans of specific drivers. Grand Prix races take place on different tracks in different nations. This comprehensive database provides all necessary information in the simplest style imaginable for anyone who is beginning to get interested in formula one racing. To use it on SQL, you may use the detailed User Guide attached with this document.

B(i)Explanation of why the selected entities were chosen:

Teams: Teams are an important component of Formula One racing; they have their constructor championships, and each championship involves a total of ten teams. This aspect of the sport cannot be overlooked, thus I included it as one of the entities. Furthermore, teams have characteristics like as their names, points, and engine model, all of which influence how the races are run and how well the drivers perform. It is also important to note that a driver in the Formula One championship is always assigned to a team.

Drivers: Drivers, like teams, are an important element of the championship. Individually, these drivers have a diverse audience that watches the championship just to support them, thus I thought it was vital to feature them as an entity. These drivers have attributes such as their names, the team they are a part of, their points, and the number of races they have competed in, all of which are important to their supporters.

Races: Races have distinct names based on where they take place. If this database is to be used in the future, this is an important factor to consider, as this is where the racetracks have been established and where the race will take place. However, it only contains the attribute Grand Prix, which is the race's name, because I believe the name also indicates the location.

Tracks: The race takes place on tracks. Because these tracks have circuits, and those circuits have lengths, and they also take place in a specific city, I assumed it was something that should be stated. Those who are interested should be aware of the circuit names as well as the locations in which they are placed.

Winners: Every race has a winner, and the winner should be mentioned because their constructors get points and they as individuals get points, which leads to the championship result. Attributes include the driver's number, which is unique to identify who the driver is, and the track on which they won.

Participates in: Is one of the linked entities with race and time properties. Because drivers compete in races on specific dates and times, this information is critical if the database is displayed afterward to view the timings in real-time.

->Attached are the pictures of data types and their descriptions:

Table:Driver

Attribute Name	type	size	null	description
DNumber	INT	3	no	Number of driver, unique
DName	VARCHAR	60	no	Name of driver
Points	INT	6		Point the driver has gained in his formula 1 career
Races Entered	INT	3	no	Amount of races driver has entered
TName	VARCHAR	50	no	Team Name of driver

Table: Team

Attribute Name	type	size	null	description
TName	VARCHAR	50	no	Name of the team, unique
constructor	VARCHAR	30	no	Constructor of team engine
chassis	CHAR	25	no	Model of engine
points	INT	5	no	Points of team

Table: Races

Attribute Name	type	size	null	description
GrandPrix	VARCHAR	60	no	Name of race, unique
DriverNo	INT	3	no	Number of drivers who participates and wins

Table:Winner

Attribute Name	type	size	null	description
DriverNo	INT	3	no	Number of driver who won ,unique
Team	VARCHAR	50	no	Team of driver who won
Track	VARCHAR	25	no	Name of track driver won on

Table: Participates in

Entity Name	datatype	size	null	description
DNumber	INT	3	no	The number of driver
GrandPrix	VARCHAR	60	no	The race driver participates in
time	TIME		no	Time of race in gst
date	DATE		no	Date of race

Table:Tracks

Entity	datatype	size	null	description
circuitName	VARCHAR	50	no	Name of circuit
Circuitlength	DECIMAL	4,3	no	Length of circuit
city	VARCHAR	25	no	City where track is

The entity, relationship sets, and cardinality constraints:

Entity set	key	Other attributes
1)Driver	DNumber (primary key)	DName, Points, Races entered
2)Team	Name (primary key)	Chassis, Points, constructor
3)Races	GrandPrix(primary key)	-
4)Tracks(weak entity set)	Circuit Name (Partial key)	city, circuit length
5)Winners	Driver No (Primary key)	Team, track

Relationship set	Between entity sets	Attributes
Participates in	Races, Drivers	Date, time
Taking place on (supporting relationship)	Race, tracks	
Has	Driver, team	
Have	Races, Winners	

Relationship set	Cardinality	Participation/ other constraints
Participates in	M: N (a driver can participate in many races; races can have many drivers participating)	Drivers-total Race-total (a driver cannot exist without participating in a race, a race cannot exist without any drivers participating in it)
Taking place on	1:1(a race can take place on one track and track can be giving place to one race at a time)	Track-partial Race-total (a race cannot take place without a track; however, a track can exist without any race taking part on it)
Has	1: N (a team has two drivers, but a driver has only one team)	Team-total Driver-total (a team cannot exist without drivers and drivers must be part of a team to exist)
Have	1: N (a race has only one winner, but one winner can win multiple races)	Race-total Winner-total (a winner cannot exist without having being part of a race, a race has to have a winner)

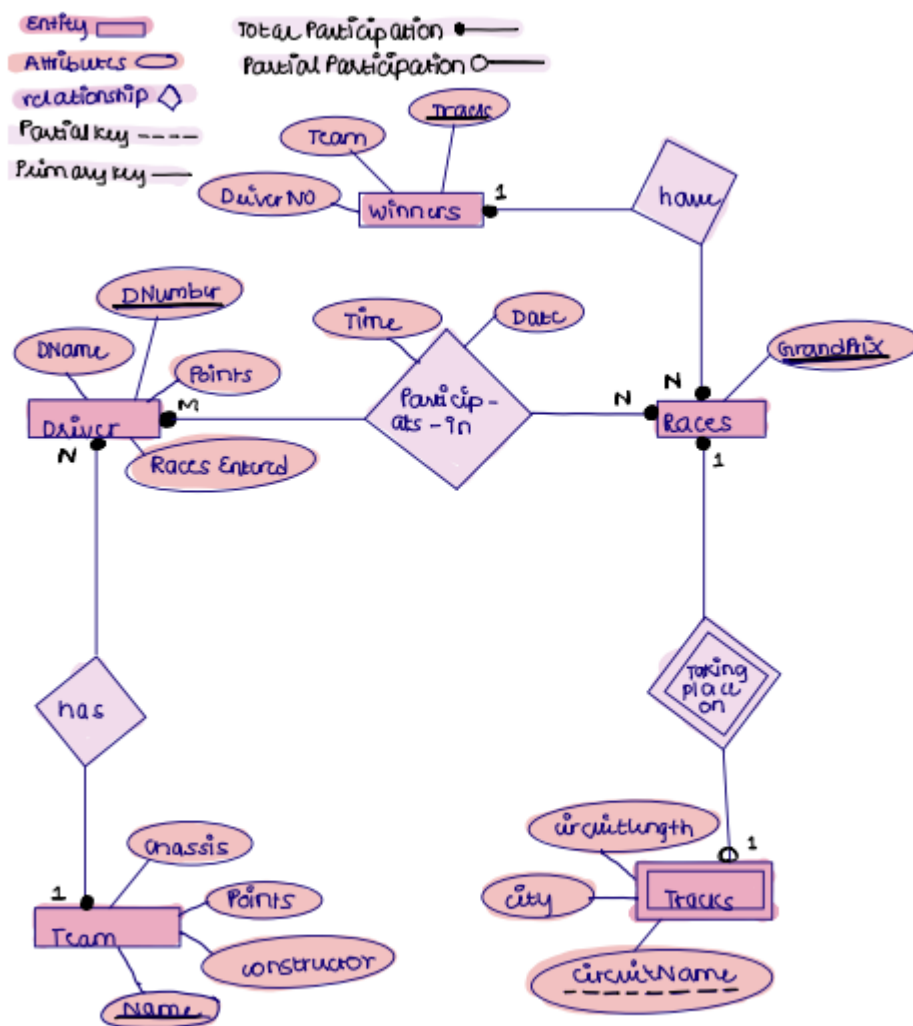
Some important assumptions:

1)Track is partial in a way that once they make the track in a certain place it will continue to exist even if no race takes part on it.

2)A driver cannot exist without a team in a way that formula 1 drivers are specific to teams if they are not in teams, they are not formula once racers.

->Below is the attached Entity-relationship diagram:

Partb(ii):



↳ Participates table :

$DNumber \rightarrow GrandPrix$, $GrandPrix \rightarrow date$, $GrandPrix \rightarrow time$

- 1) Is in 1NF since no multivalued attribute
- 2) Is in 2NF since all non-prime keys dependent on foreign keys (Primary keys of other tables)
- 3) Is in 3NF since no transitive dependency

↳ Tracks table :

$circuitName \rightarrow GrandPrix$, $circuitName \rightarrow city$, $circuitName \rightarrow length$

- 1) Is in 1NF since no multivalued attributes
- 2) Is in 2NF since all non-prime keys dependent on primary key
- 3) Is in 3NF since no transitive dependency

In-Depth Documentation

↳ Driver Table:

$DNumber \rightarrow DName, DNumber \rightarrow Points, DNumber \rightarrow RacesEntered,$
 $DNumber \rightarrow TName$

$DNumber^+ = \{DName, DNumber, Points, RacesEntered\}$

- 1) Is in 1NF since no multivalued attributes
- 2) Is in 2NF since all non-key attributes are dependent on primary key
- 3) Is in 3NF because no transitive dependency

↳ Team Table:

$TName \rightarrow constructor, TName \rightarrow chassis, TName \rightarrow Points$

$TName^+ = \{constructor, chassis, Points\}$

- 1) Is in 1NF since no multivalued attributes
- 2) Is in 2NF since all non-key attributes are dependent on primary key
- 3) Is in 3NF because no transitive dependency

↳ Races:

$GrandPrix \rightarrow location$

$GrandPrix^+ = \{location\}$

- 1) Is in 1NF since no multivalued attributes
- 2) Is in 2NF since all non-key attributes are dependent on primary key
- 3) Is in 3NF because no transitive dependency

↳ winners table:

$DriverNo \rightarrow Team, DriverNo \rightarrow Track$

$DriverNo^+ = \{Team, Track\}$

- 1) Is in 1NF since no multivalued attributes
- 2) Is in 2NF since all non-key attributes are dependent on primary key
- 3) Is in 3NF because no transitive dependency

->The relational schema:

Relational Schema:

Driver (DNumber, DName, Points, Races entered, TName)

FK TName REF Team (TName)

Team (TName, constructor, chassis, points)

Races (GrandPrix, track)

FK track REF Winners (track)

Winners (DriverNo, team, track)

Participates in (DNumber, Grand Prix, time, date)

FK DNumber REF Driver (DNumber)

FK GrandPrix REF Races (GrandPrix)

Tracks (circuitName, GrandPrix(foreign key), city, circuitlength)

FK GrandPrix, REF Races (GrandPrix)

C(i): Implementation of database

->Following is the implementation:

```
3 /*table for teams*/
4 DROP TABLE IF EXISTS Team;
5 CREATE TABLE Team
6   (TName VARCHAR(50) NOT NULL,
7    constructor VARCHAR(30) NOT NULL,
8    chassis CHAR(25) NOT NULL,
9    Tpoints INT(5) NOT NULL,
10   PRIMARY KEY(TName)
11  );
12
13 DROP TABLE IF EXISTS Drivers;
14 CREATE TABLE Drivers
15   (DNumber INT(3) PRIMARY KEY,
16    DName VARCHAR(60) NOT NULL,
17    points INT(6),
18    racesEntered INT(3) NOT NULL,
19    TName VARCHAR(50) NOT NULL,
20   CONSTRAINT team_Name
21   FOREIGN KEY (TName) REFERENCES Team(TName)
22  );
23
24 DROP TABLE IF EXISTS Winner;
25 CREATE TABLE Winner
26   (location VARCHAR(25) PRIMARY KEY,
27    Team VARCHAR(50) NOT NULL,
28    DriverNo INT(3) NOT NULL
29  );
30
31 DROP TABLE IF EXISTS Races;
32 CREATE TABLE Races
33   (GrandPrix VARCHAR(60) PRIMARY KEY,
34    location VARCHAR(25) NOT NULL,
35   CONSTRAINT location_l
36   FOREIGN KEY(location) REFERENCES Winner(location)
37  );
38
39 DROP TABLE IF EXISTS Tracks;
40 CREATE TABLE Tracks
41   (circuitName VARCHAR(50) PRIMARY KEY,
42    GrandPrix VARCHAR(60) NOT NULL,
43    circuitlength DECIMAL(4,3) NOT NULL,
44    city VARCHAR(25) NOT NULL,
45   CONSTRAINT track_name
46   FOREIGN KEY(GrandPrix) REFERENCES Races(GrandPrix)
47  );
48
49 DROP TABLE IF EXISTS ParticipatesIn;
50 CREATE TABLE ParticipatesIn
51   (Rdate DATE NOT NULL,
52    Rtime TIME NOT NULL,
53    DNumber INT(3) NOT NULL,
54    GrandPrix VARCHAR(60) NOT NULL,
55   FOREIGN KEY (DNumber) REFERENCES Drivers(DNumber),
56   FOREIGN KEY(GrandPrix) REFERENCES Races(GrandPrix)
57  );
58
```


->To insert values in the database:

```
114 INSERT INTO Tracks(circuitName,GrandPrix,circuitlength,city)
115
116 VALUES
117 ('Albert Park','Australian GP',5.303,'Melbourne'),
118 ('Bahrain International','Bahrain GP',5.412,'Sakhir'),
119 ('Shanghai International','Chinese GP',5.412,'Shanghai'),
120 ('Baku City','Azerbaijan GP',6.003,'Baku'),
121 ('Circuit de Barcelona-Catalunya','Spanish GP',4.655,'Spain'),
122 ('Circuit de Monaco','Monaco GP',3.337,'Monte-Carlo'),
123 ('Circuit Gilles-Villeneuve','Canadian GP',4.361,'Montreal'),
124 ('Circuit Paul Ricard','French GP',5.842,'La Castellet'),
125 ('Red Bull Ring','Austrian GP',5.842,'Spielberg'),
126 ('Silverstone circuit','British GP',5.891,'Silverstone'),
127 ('Hockenheimring','German GP',4.574,'Hockenheim'),
128 ('Hungaroring','Hungarian GP',4.381,'Budapest'),
129 ('Circuit de Spa-Francorchamps','Belgian GP',7.004,'Budapest'),
130 ('Autodromo Nazionale Monza','Italian GP',5.793,'Monza'),
131 ('Marina Bay Street','Singapore GP',5.063,'Singapore'),
132 ('Sochi Autodrom','Russian GP',5.848,'Sochi'),
133 ('Suzuka International Racing Course','Japanese GP',5.807,'Suzuka'),
134 ('Autodromo Hermanos Rodriguez','Mexican GP',4.304,'Mexico city'),
135 ('Circuit of the Americas','United States GP',5.513,'Austin'),
136 ('Autodromo Jose Carlos Pace','Brazilian GP',4.309,'Sab Paulo'),
137 ('Yas Marina','Abu Dhabi GP',5.554,'Yas Island');
```

```
19 INSERT INTO Drivers VALUES
20 (44,'Lewis Hamilton',3431,250,'Mercedes');
21 INSERT INTO Drivers VALUES
22 (77,'Valtteri Bottas',1289,140,'Mercedes');
23 INSERT INTO Drivers VALUES
24 ('16','Charles Leclerc',303,42,'Ferrari');
25 INSERT INTO Drivers VALUES
26 (33,'Max Verstappen',948,102,'Red Bull Racing');
27 INSERT INTO Drivers VALUES
28 (16,'Charles Leclerc',303,42,'Ferrari');
29 INSERT INTO Drivers VALUES
30 (5,'Sebastian Vettel',2985,241,'Ferrari');
31 INSERT INTO Drivers VALUES
32 (55,'Carlos Sainz',267,102,'McLaren');
33 INSERT INTO Drivers VALUES
34 (10,'Pierre Gasly',124,47,'Scuderia Toro Rosso');
35 INSERT INTO Drivers VALUES
36 (23,'Alexander Albon',92,21,'Red Bull Racing');
37 INSERT INTO Drivers VALUES
38 (3,'Daniel Ricciardo',1040,171,'Renault');
39 INSERT INTO Drivers VALUES
40 (11,'Sergio Perez',581,178,'Racing Point');
41 INSERT INTO Drivers VALUES
42 (4,'Lando Norris',49,21,'McLaren');
43 INSERT INTO Drivers VALUES
44 (7,'Kimi Raikkonen',1859,315,'Alfa Romeo Racing');
45 INSERT INTO Drivers VALUES
46 (26,'Daniil Kvyat',170,95,'Scuderia Toro Rosso');
```

```

44 (7,'Kimi Raikkonen',1859,315,'Alfa Romeo Racing');
45 INSERT INTO Drivers VALUES
46 (26,'Daniil Kvyat',170,95,'Scuderia Toro Rosso');
47 INSERT INTO Drivers VALUES
48 (27,'Nico Hulkenberg',511,179,'Renault');
49 INSERT INTO Drivers VALUES
50 (18,'Lance Stroll',67,62,'Racing Point');
51 INSERT INTO Drivers VALUES
52 (20,'Kevin Magnussen',157,103,'Haas');
53 INSERT INTO Drivers VALUES
54 (99,'Antonio Giovinazzi',14,23,'Alfa Romeo Racing');
55 INSERT INTO Drivers VALUES
56 (8,'Romain Grosjean',389,166,'Haas');
57 INSERT INTO Drivers VALUES
58 (88,'Robert Kubica',274,97,'Williams');
59 INSERT INTO Drivers VALUES
60 (63,'George Russell',0,21,'Williams');

```

```

139 INSERT INTO ParticipatesIn(Rdate,Rtime,DNumber,GrandPrix)

```

```

140 VALUES
141 ('2019/03/17','05:10:00',77,'Australian GP'),
142 ('2019/03/31','04:10:00',44,'Bahrain GP'),
143 ('2019/04/14','07:10:00',44,'Chinese GP'),
144 ('2019/04/28','01:10:00',77,'Azerbaijan GP'),
145 ('2019/05/12','02:10:00',44,'Spanish GP'),
146 ('2019/05/26','02:10:00',44,'Monaco GP'),
147 ('2019/06/09','07:10:00',44,'Canadian GP'),
148 ('2019/06/23','02:10:00',44,'French GP'),
149 ('2019/06/30','02:10:00',33,'Austrian GP'),
150 ('2019/07/14','02:10:00',44,'British GP'),
151 ('2019/07/28','02:10:00',33,'German GP'),
152 ('2019/08/04','02:10:00',44,'Hungarian GP'),
153 ('2019/09/01','02:10:00',16,'Belgian GP'),
154 ('2019/09/08','02:10:00',16,'Italian GP'),
155 ('2019/09/22','01:10:00',5,'Singapore GP'),
156 ('2019/09/29','12:10:00',44,'Russian GP'),
157 ('2019/10/13','06:10:00',77,'Japanese GP'),
158 ('2019/10/27','07:10:00',44,'Mexican GP'),
159 ('2019/11/3','07:10:00',77,'United States GP'),
160 ('2019/11/17','05:10:00',33,'Brazilian GP'),
161 ('2019/12/01','01:10:00',44,'Abu Dhabi GP');
162

```

```

89 INSERT INTO Races(GrandPrix,location)
90
91 VALUES
92 ('Australian GP','Australia'),
93 ('Bahrain GP','Bahrain'),
94 ('Chinese GP','China'),
95 ('Azerbaijan GP','Azerbaijan'),
96 ('Spanish GP','Spain'),
97 ('Monaco GP','Monaco'),
98 ('Canadian GP','Canada'),
99 ('French GP','France'),
100 ('Austrian GP','Austria'),
101 ('German GP','Germany'),
102 ('British GP','Great Britain'),
103 ('Hungarian GP','Hungary'),
104 ('Belgian GP','Belgium'),
105 ('Italian GP','Italy'),
106 ('Singapore GP','Singapore'),
107 ('Russian GP','Russia'),
108 ('Japanese GP','Japan'),
109 ('Mexican GP','Mexico'),
110 ('United States GP','United States'),
111 ('Brazilian GP','Brazil'),
112 ('Abu Dhabi GP','Abu Dhabi');

4 INSERT INTO
5   Team(TName,constructor,chassis,Tpoints)
6
7 VALUES
8 ('Ferrari','Ferrari','SF90',739),
9 ('Mercedes','Mercedes','F1 W10 EQ Power+',504),
10 ('Red Bull Racing','Honda','RB15',417),
11 ('McLaren','Renault','MCL34',145),
12 ('Renault','Renault','R.S.19',91),
13 ('Alfa Romeo Racing','Ferrari','C38',57),
14 ('Scuderia Toro Rosso','Honda','STR14',85),
15 ('Racing Point','BWT Mercedes','RP19',73),
16 ('Haas','Ferrari','VF-19',28),
17 ('Williams','Mercedes','FW42',1);

62 INSERT INTO Winner(DriverNo,Team,location)
63
64 VALUES
65 (77,'Mercedes','Australia'),
66 (44,'Mercedes','Bahrain'),
67 (44,'Mercedes','China'),
68 (77,'Mercedes','Azerbaijan'),
69 (44,'Mercedes','Spain'),
70 (44,'Mercedes','Canada'),
71 (44,'Mercedes','France'),
72 (33,'Red Bull Racing','Austria'),
73 (44,'Mercedes','Great Britain'),
74 (33,'Red Bull Racing','Germany'),
75 (44,'Mercedes','Hungary'),
76 (16,'Ferrari','Belgium'),
77 (16,'Ferrari','Italy'),
78 (5,'Ferrari','Singapore'),
79 (44,'Mercedes','Russia'),
80 (77,'Mercedes','Japan'),
81 (44,'Mercedes','Mexico'),
82 (77,'Mercedes','United States'),
83 (33,'Red Bull Racing','Brazil'),
84 (44,'Mercedes','Abu Dhabi'),
85 (44,'Mercedes','Monaco');

```

d)Design of queries and sample outputs:

Query1:Use and sample output

```
mysql> SELECT DNumber,DName,TName,points
-> FROM Drivers
-> WHERE racesEntered > 50;
```

DNumber	DName	TName	points
3	Daniel Ricciardo	Renault	1040
5	Sebastian Vettel	Ferrari	2985
7	Kimi Raikkonen	Alfa Romeo Racing	1859
8	Romain Grosjean	Haas	389
11	Sergio Perez	Racing Point	581
18	Lance Stroll	Racing Point	67
20	Kevin Magnussen	Haas	157
26	Daniil Kvyat	Scuderia Toro Rosso	170
27	Nico Hulkenberg	Renault	511
33	Max Verstappen	Red Bull Racing	948
44	Lewis Hamilton	Mercedes	3431
55	Carlos Sainz	McLaren	267
77	Valtteri Bottas	Mercedes	1289
88	Robert Kubica	Williams	274

14 rows in set (0.00 sec)

This query reveals the points of drivers who have competed in more than 50 races during their careers. Using this query, it is simpler to identify that drivers with 1250 points have won multiple times. Can assist users in locating the driver of their choice.

Query2: Sample output

```
mysql> SELECT w.DriverNo, COUNT(w.DriverNo),d.DName,d.TName
-> From Winner w INNER JOIN Drivers d
-> ON d.DNumber = w.DriverNo
-> GROUP BY DriverNo
-> HAVING COUNT(DriverNo) > 1;
```

DriverNo	COUNT(w.DriverNo)	DName	TName
44	11	Lewis Hamilton	Mercedes
77	4	Valtteri Bottas	Mercedes
33	3	Max Verstappen	Red Bull Racing
16	2	Charles Leclerc	Ferrari

4 rows in set (0.00 sec)

This query will assist the user in determining the driver who would have won the championships, as such drivers would have the most wins, as well as those who finished second and third. It will also help in figuring out who would win the constructor championship.

Query3:Sample output

```
mysql> SELECT CONCAT(Rdate,' ',Rtime) AS Race_Schedule, GrandPrix
-> FROM ParticipatesIn;
```

Race_Schedule	GrandPrix
2019-03-17 05:10:00	Australian GP
2019-03-31 04:10:00	Bahrain GP
2019-04-14 07:10:00	Chinese GP
2019-04-28 01:10:00	Azerbaijan GP
2019-05-12 02:10:00	Spanish GP
2019-05-26 02:10:00	Monaco GP
2019-06-09 07:10:00	Canadian GP
2019-06-23 02:10:00	French GP
2019-06-30 02:10:00	Austrian GP
2019-07-14 02:10:00	British GP
2019-07-28 02:10:00	German GP
2019-08-04 02:10:00	Hungarian GP
2019-09-01 02:10:00	Belgian GP
2019-09-08 02:10:00	Italian GP
2019-09-22 01:10:00	Singapore GP
2019-09-29 12:10:00	Russian GP
2019-10-13 06:10:00	Japanese GP
2019-10-27 07:10:00	Mexican GP
2019-11-03 07:10:00	United States GP
2019-11-17 05:10:00	Brazilian GP
2019-12-01 01:10:00	Abu Dhabi GP

21 rows in set (0.00 sec)

This query will return the entire schedule for the Grand Prix, including the race time and date. It may also be used to help users identify the date and time of a live stream, and if the championship is ongoing, it can be used to help supporters buy tickets in advance.

Query4:Sample output

```
mysql> SELECT circuitlength,city,GrandPrix
-> FROM Tracks
-> WHERE circuitlength = (SELECT MAX(circuitlength)
-> FROM Tracks);
```

circuitlength	city	GrandPrix
7.004	Budapest	Belgian GP

1 row in set (0.01 sec)

This query will assist the user in determining the longest circuit length and its name. With this information, the user may determine the circuit's winner as well as other pertinent information through the table.

Query5: Sample Output

```
mysql> SELECT TName, constructor
-> FROM Team
-> WHERE constructor IN (SELECT constructor
-> FROM Team
-> GROUP BY constructor
-> HAVING COUNT(*) > 1);
```

TName	constructor
Alfa Romeo Racing	Ferrari
Ferrari	Ferrari
Haas	Ferrari
McLaren	Renault
Mercedes	Mercedes
Red Bull Racing	Honda
Renault	Renault
Scuderia Toro Rosso	Honda
Williams	Mercedes

9 rows in set (0.00 sec)

This query will show users which teams have the same engine constructors, allowing them to determine which chassis to utilize if the competition is fair.

Query6: Sample output

```
mysql> SELECT DName,DNumber
-> FROM Drivers
-> WHERE TName LIKE '%Ferrari%';
```

DName	DNumber
Sebastian Vettel	5
Charles Leclerc	16

2 rows in set (0.01 sec)

This query will tell you which teams have drivers, so you can do point distribution accordingly. You can also use this query to discover drivers for each team by changing the team's name in the query. Then there's no need to compare the teams by going through the entire driver's table.

Query7:Sample output


```
mysql> SELECT points,DName
-> FROM Drivers
-> ORDER BY points DESC;
```

points	DName
3431	Lewis Hamilton
2985	Sebastian Vettel
1859	Kimi Raikkonen
1289	Valtteri Bottas
1040	Daniel Ricciardo
948	Max Verstappen
581	Sergio Perez
511	Nico Hulkenberg
389	Romain Grosjean
303	Charles Leclerc
274	Robert Kubica
267	Carlos Sainz
170	Daniil Kvyat
157	Kevin Magnussen
124	Pierre Gasly
92	Alexander Albon
67	Lance Stroll
49	Lando Norris
14	Antonio Giovinazzi
0	George Russell

```
20 rows in set (0.00 sec)
```

This query will show you which driver has consistently had the best results in the championships. It might be useful for a user that is trying to find their favorite drivers position overall

Query8:Sample output

```
mysql> SELECT t.circuitlength,t.city,p.DNumber
-> FROM Tracks t INNER JOIN ParticipatesIn p
-> ON t.GrandPrix = p.GrandPrix
-> ORDER BY t.circuitlength DESC;
```

circuitlength	city	DNumber
7.004	Budapest	16
6.003	Baku	77
5.891	Silverstone	44
5.848	Sochi	44
5.842	Spielberg	33
5.842	La Castellet	44
5.807	Suzuka	77
5.793	Monza	16
5.554	Yas Island	44
5.513	Austin	77
5.412	Shanghai	44
5.412	Sakhir	44
5.303	Melbourne	77
5.063	Singapore	5
4.655	Spain	44
4.574	Hockenheim	33
4.381	Budapest	44
4.361	Montreal	44
4.309	Sab Paulo	33
4.304	Mexico city	44
3.337	Monte-Carlo	44

```
21 rows in set (0.00 sec)
```

This query will help you determine all circuit lengths, the cities where the circuits were held, and the driver who won on them. You can also determine the circuit's complexity by looking at its turns and other factors, and then see which driver did the best to achieve such results.

Query9:Sample output

```
mysql> SELECT COUNT(GrandPrix) AS totalRaces
-> FROM Races;
+-----+
| totalRaces |
+-----+
|          21 |
+-----+
1 row in set (0.14 sec)
```

This query will simply assist the user in finding the total races for each season directly

Query10:sample output

```
mysql> SELECT d.DName,d.TName
-> FROM Drivers d INNER JOIN Winner w
-> ON w.DriverNo = d.DNumber
-> GROUP BY d.DNumber;
+-----+-----+
| DName          | TName          |
+-----+-----+
| Lewis Hamilton | Mercedes       |
| Valtteri Bottas | Mercedes       |
| Max Verstappen | Red Bull Racing |
| Charles Leclerc | Ferrari        |
| Sebastian Vettel | Ferrari        |
+-----+-----+
5 rows in set (0.00 sec)
```

This query will assist in determining that the driver which had won belonged to which team so who will get the constructor points

Query11: Sample output

```
mysql> SELECT TName,DName
-> FROM Drivers
-> WHERE TName = 'Racing Point';
+-----+-----+
| TName          | DName          |
+-----+-----+
| Racing Point   | Sergio Perez   |
| Racing Point   | Lance Stroll   |
+-----+-----+
2 rows in set (0.00 sec)
```

The query will help finding drivers belonging to each team without looking through the tables

Query12:Sample output

```
mysql> SELECT DATEDIFF("2019/03/31","2019/03/17") AS daysapart;
+-----+
| daysapart |
+-----+
|         14 |
+-----+
1 row in set (0.01 sec)
```

When new races take place, this will help the user in determining the date of the date race or the time gap that will be given between each date.

Implementation and explanation of advanced features

- 1) Procedures 1 and 3 assist in the integration of new teams into existing teams. I'm thinking that this database will be used for a new season, which will make data entry much easier because, while the teams remain the same, 1-2 teams are added in certain years, which can assist shorten the labor. The third query adds a new winner every year to a database that can be created and populated with new winners using this technique, making the user's job easier. The third procedure just provides you the number of races for each year; all you have to do is call it and figure out how many years there are. Here's how they're created, as well as how they'll be called and how they'll be utilized.

```
2 CREATE PROCEDURE insNewTeam(
3   TeamName VARCHAR(50),
4   EngineMaker VARCHAR(30),
5   EngineModel CHAR(25),
6   TeamPoints INT(5)
7 )
8 COMMENT 'Inserts another team in team table'
9 INSERT INTO Team(TName, constructor, chassis, Tpoints)
10 VALUES(TeamName,EngineMaker,EngineModel,TeamPoints);
11
12 /*USAGE:CALL insNewTeam('Range Rover','Range Rover','C34',0);*/
13
14 /*procedure to insert new values into table winner*/
15
16 CREATE PROCEDURE insNewWinner(
17   place VARCHAR(25),
18   TeamName VARCHAR(50),
19   DriverNumber INT(3)
20 )
21 INSERT INTO Winner(location, Team, DriverNo)
22 VALUES(place,TeamName,DriverNumber);
23
24 /*USAGE--CALL insNewWinner('Morocco','merced',53);*/
25
26
```

```

/*USAGE:CALL insNewTeam('Range Rover','Range Rover','C34',0);*/

CREATE PROCEDURE findTotalRaces(
OUT tRace INT
)
COMMENT 'Count the number of races in each season.'
SELECT COUNT(GrandPrix) FROM Races;
/*USAGE CALL findTotalRaces(@result);*/

```

The first trigger is a simple check that the points of the drivers being updated are not lower than they were before because this table does not need to be changed for each season, but the points and races entered can be increased but not decreased, so this trigger simply addresses that issue.

```

1 /*trigger for error checking when the points of drivers are updated they cannot be lesser than the old one*/
2 DELIMITER //
3
4 CREATE TRIGGER beforeUpdatingPoints
5 BEFORE UPDATE
6 ON Drivers FOR EACH ROW
7 BEGIN
8 DECLARE errorMessage VARCHAR(100);
9 SET errorMessage = CONCAT('The new points', ' ', NEW.points, ' ',
10 'cannot be lesser than', ' ', OLD.points);
11 IF NEW.points < OLD.points THEN
12 SIGNAL SQLSTATE '45000'
13 SET MESSAGE_TEXT = errorMessage;
14 END IF;
15 END//
16 DELIMITER ;

```

```

mysql> UPDATE Drivers SET points = 32 WHERE DName = 'Lewis Hamilton';
ERROR 1644 (45000): The new points 32 cannot be lesser than 3431

```

The second trigger basically deleted the team drivers when the team is deleted because when a team is gone its drivers can no longer participate, it simply makes the work of the user easier.

```

18 /*deletes from drivers once team deleted*/
19
20 DELIMITER //
21 CREATE TRIGGER afterTeamDeleted
22 BEFORE DELETE ON Team
23 FOR EACH ROW
24 BEGIN
25 IF OLD.TName IS NOT NULL THEN
26 DELETE FROM Drivers
27 WHERE TName = OLD.TName;
28 END IF;
29 END //
30 DELIMITER ;

```

```

mysql> DELETE FROM Team WHERE TName = 'Williams';
Query OK, 1 row affected (0.01 sec)

```

3) These are two straightforward views. The first indicates the best drivers if the user is interested in learning who they are, and the second indicates drivers who have won more than one race, putting them in a higher position than others. For someone who is interested, this information may be crucial in determining who is most likely to win the championship while it is still ongoing.

```

1 /*this view shows the best drivers from all championships*/
2 CREATE VIEW best_drivers AS
3 SELECT DNumber,DName,points,racesEntered
4 FROM Drivers
5 WHERE racesEntered > 50 AND points > 2000;
6
7 /*USAGE--SELECT * FROM best_drivers;*/
8
9 /*view is of drivers who have more than one win*/
10 CREATE VIEW winningStreak AS
11 SELECT w.DriverNo, COUNT(w.DriverNo),d.DName,d.TName
12 From Winner w INNER JOIN Drivers d
13 ON d.DNumber = w.DriverNo
14 GROUP BY DriverNo
15 HAVING COUNT(DriverNo) > 1;
16
17

```

->Python implementation:

I established a python connection and connected it to my database now it is easier for the user to view everything, unlike SQL they won't have to type individual commands.

e) Challenges:

->The data entry was a very tough and time-consuming task, there may be better ways to do this than copying data from a table

->The choice was very wide and being someone who is not interested in the championship it was very hard to choose which data would be essential to include hence I would conclude this as a very basic database

->This is my first-time using python hence the functionality of the database would be very less, it would have been more detailed if it was a programming language, I was aware of.

2)limitations

->As mentioned above the database is very basic it is not in-depth doesn't consider all aspects of the formula 1 races

->The queries are simple compared to a real-life database

3)future direction

->Although this database is basic, it can be used when the championship resumes. The driver's table does not require any changes; only the points and races entered must be upgraded, and new values can be inserted if necessary.

->The race schedule can be shown on the website, and a second entity named ticketing can be created so that individuals can buy tickets based on their location by glancing at the schedule.

->The tables may be recreated for each season, with fresh values updated while the race is in progress, and another table that displays the points earned by each driver at each race can be included as well.

```

1 # Nuha Imran
2 # 20696366
3 import mysql.connector
4 username = input("Enter your username?")
5 passwordp = input("Enter your password?")
6 databased = input("Enter databasename?")
7
8 conn = mysql.connector.connect(user=username,
9                                password=passwordp,
10                               database=databased)
11 cursor = conn.cursor()
12
13 print("1)Display driver table")
14 print("2)Display Team table")
15 print("3)Display Winner table")
16 print("4)Display Participates In table")
17 print("5)Display Tracks table")
18 print("6)Drivers with multiple wins")
19 print("7)Teams that have same engine constructors")
20 print("8)Drivers who have highest points in championship so far?")
21 print("9)To insert more teams to table team")
22 print("10)To insert more winners")
23 print("11)To view best drivers")
24 print("12)To view drivers who have won more than one race")
25 userinput = input("\nChoose which table you want to display?\n")
26
27 if userinput == "1":
28     # Using the cursor as iterator
29     cursor.execute("SELECT * FROM Drivers")
30     for row in cursor:
31         print(row)
32
33     print("\n")
34
35 elif userinput == "2":
36     cursor.execute("SELECT * FROM Team")
37     for row in cursor:
38         print(row)
39     print("\n")
40
41 elif userinput == "3":
42     cursor.execute("SELECT * FROM Winner")
43     for row in cursor:
44         print(row)
45
46     print("\n")
47
48 elif userinput == "4":
49     cursor.execute("SELECT * FROM ParticipatesIn")
50     for row in cursor:
51         print("Date:", row[0], "Time:", row[1],
52               "DriverNo:", row[2], "GrandPrix:", row[3])
53     print("\n")
54
55

```

```

57 elif userInput == "5":
58     cursor.execute("SELECT * FROM Tracks")
59     for row in cursor:
60         print("CircuitName:", row[0], "GrandPrix:", row[1],
61             "circuitlength:", row[2], "city:", row[3])
62
63
64 elif userInput == "6":
65     cursor.execute("SELECT w.DriverNo, COUNT(w.DriverNo),d.DName,d.TName From Winner w INNER JOIN Drivers d ON d.DNumber = w.DriverNo
66     GROUP BY DriverNo HAVING COUNT(DriverNo) > 1")
67     for row in cursor:
68         print(row)
69
70 elif userInput == "7":
71     cursor.execute("SELECT TName, constructor FROM Team WHERE constructor IN (SELECT constructor FROM Team GROUP BY constructor
72     HAVING COUNT(*) > 1)")
73     for row in cursor:
74         print(row)
75
76     print("\n")
77
78 elif userInput == "8":
79     cursor.execute("SELECT points,DName FROM Drivers ORDER BY points DESC")
80     for row in cursor:
81         print(row)
82
83     print("\n")
84
85 elif userInput == "9":
86     TeamName = input("Write a team Name?")
87     TeamCons = input("Write constructor Name?")
88     chas = input("Write engine model?")
89     poi = input("Write team point?")
90
91     points = int(poi)
92     TVal = (TeamName,TeamCons,chas,points)
93     cursor.execute("CALL insNewTeam(%s,%s,%s,%s)",TVal)
94     conn.commit()
95
96 elif userInput == "10":
97     place = input("Enter the place where the race took place?")
98     Team_Name = input("Team the winner belonged to?")
99     DriverNo = input("Driver Number of the winner?")
100
101     DNo = int(DriverNo)
102     Val2 = (place,Team_Name,DNo)
103     cursor.execute("CALL insNewWinner(%s,%s,%s)",Val2)
104     conn.commit()
105
106 elif userInput == "11":
107     cursor.execute("SELECT DNumber,DName,points,racesEntered FROM Drivers WHERE racesEntered > 50 AND points > 2000")
108     for row in cursor:
109         print(row)
110
111
112     print("\n")
113
114 elif userInput == "12":
115     cursor.execute("SELECT w.DriverNo, COUNT(w.DriverNo),d.DName,d.TName From Winner w INNER JOIN Drivers d ON d.DNumber = w.DriverNo GROUP
116     BY DriverNo HAVING COUNT(DriverNo) > 1")
117     for row in cursor:
118         print(row)
119
120 cursor.close()
121 conn.close()
122

```