

Gustavo Queries

1. **This query was used to give us understanding of how many races were hosted in each country**

```
select c.country as 'Country', count(*) as 'Races Hosted'
from circuits as c
join races as r
using (circuitid)
GROUP BY c.country
ORDER BY count(*) DESC ;
```

2. **This query was written to inform us on how many races took place in each city**

```
SELECT c.location as 'City', count(r.circuitid) as 'Races Hosted'
from circuits as c
join races as r
using (circuitid)
group by c.location
order by count(r.circuitid) DESC;
```

3. **This query was an intricate one that informed us of the fastest lap in milliseconds at each specific circuit**

```
select c.location as "City", r.year as "Year of race", min(l.milliseconds) as "Lap Time in ms",
concat(forname,' ',surname) as 'Driver Name'
from lap_times as l
JOIN drivers
USING (driverid)
join races as r
USING (raceid)
JOIN circuits AS c
using (circuitid)
GROUP BY c.location
order by c.location;
```

4. **This query will provide us with the information to the question of how many pole positions did each driver have between 1999 and 2010**

```
select dr.surname, count(qu.position) pole_position
from qualifying as qu join
races as r
using(raceid)
```

```

join drivers dr
USING(driverId)
where r.year>1999 and r.year<2010 and qu.position=1
group by dr.surname
order by pole_position desc;

```

- 5. This query answered the question of which drivers performed best during their first season in the formula 1. In other words, how many times did a driver win a race during its debut season?**

```

select dr.surname, count(*) num_race_wins, sq.debut_season

from results rs
  join
    races r on rs.raceId=r.raceId
  join
    drivers dr on rs.driverId=dr.driverId
  join
    (select rs.driverId, min(r.year) debut_season from results rs join
    races r on rs.raceId=r.raceId
    group by rs.driverId) sq
  on
    rs.driverId=sq.driverId and r.year=sq.debut_season
where rs.position=1
group by dr.surname
order by num_race_wins desc;

```

Simon Query

1. If a fan wants to be a fan of a team with a rich history, Williams vs Ferrari constructor points over time shows two of the oldest teams compared to each other. Shows how well they competed with each other in storied rivalries for teams that still exist today.

```

select c.name, sum(points), r.year
from races as r
inner join constructor_results
on r.raceId = constructor_results.raceId
inner join constructors as c on
constructor_results.constructorId = c.constructorId
where c.name in ( 'Ferrari', 'Williams')
group by c.name, year

```

2. Want to be a fan of a driver from your country? Where should F1 target merchandise?
Counting how many drivers each country has produced.

```
select nationality, count(nationality)
from drivers
group by nationality
```

3. Want to advertise to countries that have the most races? Want to convince a fan to go to their home race? Most common races in history.

```
select name, count(distinct raceID)
from races
group by name
order by count(distinct raceID) DESC;
```

4. Wanting to find a favorite driver for a new fan? Why not pick a winning one? Total times points by drivers can be seen as a way to codify who is a reliable driver over time. This would make it so constructors have an ability to see who may not be great, but can deliver points every week.

```
select concat(forname,surname) as fullname, sum(points)
from drivers as d left join driver_standings as ds
on d.driverid = ds.driverid
group by fullname;
```

5. Qualifying is the benchmark of having a fast car and a fast driver. Qualifying is a game of elimination, which means that only the top 10 fastest move on. Making Q3 is a good metric of the relationship between car and driver. Want to be a fan of the fastest team? Where to put advertising dollars? Behind a big team.

```
select concat(forname,surname), count(q3), dense_rank() over(order by count(q3))
from drivers as d join qualifying as q
on d.driverid = q.driverid
group by surname
```

Jared queries

#1

Question: In what race does Ferrari have the highest average fastest lap speed?

Details: Visualization 1 query: This query gives the same output as the fastest speed of Ferrari per race graph. Here the average of the fastest speed from the results is taken

and joined with the races table through multiple other tables in order to group by the name of each race.

```
select r.name, avg(res.fastestlapspeed)
FROM results as res
INNER JOIN qualifying as q
ON res.driverid = q.driverid
INNER JOIN constructors as c
ON q.constructorid = c.constructorsid
INNER JOIN races as r
ON q.raceid = r.raceid
where c.name = 'Ferrari'
group by r.name
order by avg(res.fastestlapspeed) desc;
```

#2

Question: As far back as the data goes, what is the average position the Red Bull racing team finished in every year?

Details: Visualization 2 query: This query relates to the Red Bull yearly position graph. Here the average of the position results are taken and joined with the constructors table through the races table in order to group by the name of the racing team, Red Bull, while specifying each year.

```
select r.year, avg(res.position) as 'average'
from races as r
inner join results as res
using(raceid)
inner join constructors as c
on c.constructorsid = res.constructorid
group by r.year, c.name
having c.name='Red Bull';
```

#3

Question: Milliseconds are hard to work with and visualize accurately, is there a way to convert them to seconds efficiently?

Details: Here a function is created to convert milliseconds to seconds in whatever table this function is called in.

```
delimiter !!
create function sec_convert(
```

```

milliseconds int)
returns decimal
deterministic
begin
declare convert_sec decimal;
set convert_sec = milliseconds/0.001;
return (convert_sec);
end !!

```

#4

Question: If milliseconds can be converted to seconds, how can there be an automatic update in a table for a conversion of units?

Details: Using the function above, this trigger has been created to automatically convert the milliseconds to seconds if a new row is inserted into the lap times table in our database.

```

create trigger sec_convert_trig
before insert
on lap_times
for each row
set new.milliseconds=sec_convert(new.milliseconds);

```

#5

Question: What drivers correspond to which constructors by their name, how many drivers does each team have, and what has been each driver's highest position?

Details: This CTE outputs the constructor and driver full name, using concatenation, along with their respective ids, joined with the max of each driver finishing position. In addition, the count of the number of drivers associated with that specific team is included with the use of a window function.

```

with cte_constructor_drivers as (
select c.constructorsid,c.name, d.driverid,concat(d.surname," ",d.forname) as 'full
name',max(ds.position), count(*) over(partition by c.name)
from constructors as c
inner join results as res
on c.constructorsid = res.constructorid
inner join drivers as d
using(driverid)
inner join driver_standings as ds
using(driverid)
group by c.constructorsid,c.name, d.driverid,concat(d.surname," ",d.forname))

```

```
select *  
from cte_constructor_drivers;
```

Justin Queries

-- First Query - Business Question: **What are the descriptive statistics for pit stop usage for each driver in the dataset?**

alter duration field in pit_stops to 3-decimal seconds value (Decimal(10,3))

```
ALTER TABLE pit_stops  
DROP COLUMN duration;  
ALTER TABLE pit_stops  
ADD COLUMN duration DECIMAL(10,3);  
UPDATE pit_stops SET duration = milliseconds / 1000;
```

```
SELECT  
    driverid,  
    SUM(duration) AS TotalStopTime,  
    COUNT(stop) AS NumberOfStopsRecorded,  
    round(AVG(duration),3) AS AvgStopDuration,  
    MAX(stop) AS MostStopsNeeded,  
    round(AVG(stop),3) AS AvgStopsNeeded  
FROM pit_stops as ps  
JOIN drivers as d  
USING(driverid)  
JOIN races as r  
USING(raceid)  
GROUP BY driverid  
ORDER BY SUM(duration) DESC;
```

-- Second Query - Business Question: **What is the number/proportion of successful race finishes by each constructor, separated by decade of data?**

```
WITH cte_FinishesByDecade AS  
(  
    SELECT  
        ra.year DIV 10 AS Decade,  
        constructorid,
```

```

        c.name,
        COUNT(statusid)
        as NumberOfSuccessfulFinishes
FROM results as r
JOIN constructors as c
ON c.constructorid = r.constructorid
JOIN races as ra
ON ra.raceid = r.raceid
WHERE statusid = 1
GROUP BY Decade, r.constructorid
ORDER BY Decade DESC, COUNT(statusid) DESC
)
SELECT Decade*10 AS race_decade, constructorid, name, NumberOfSuccessfulFinishes,
        NumberOfSuccessfulFinishes / SUM(NumberOfSuccessfulFinishes) OVER(PARTITION
BY Decade) * 100 AS PercentageOfDecadeFinishes
FROM cte_FinishesByDecade
ORDER BY Decade DESC;

```

-- Third Query - Business Question: **Historically, what have been the most common result statuses for drivers at the end of their race?**

```

SELECT statusid, status, count(statusid) as NumberOfOccurrences
FROM results as r
JOIN drivers as d
USING(driverid)
JOIN status as s
USING(statusid)
GROUP BY statusid
ORDER BY count(statusid) DESC;

```

-- Fourth Query - Business Question: **What is the career fastest lap for each driver?**

```

WITH cte_laptimes AS
(
SELECT driverid, MIN(fastestlaptime) as fastestlap
FROM results
WHERE (fastestlaptime IS NOT NULL)
GROUP BY raceid, driverid
ORDER BY raceid, driverid
)
SELECT driverid, MIN(fastestlap) as CareerFastestLap
FROM cte_laptimes
GROUP BY driverid;

```

-- Fifth Query - Business Question: On average, how many points does each constructor earn per race?

```
#DROP VIEW constructor_points;  
CREATE VIEW constructor_points AS  
SELECT constructorid as ID, points  
FROM results  
GROUP BY raceid, constructorid  
ORDER BY resultsid;
```

```
SELECT ID as constructor_id, constructors.name, AVG(points) as point_average  
FROM constructor_points  
JOIN constructors ON constructor_points.ID = constructors.constructorsid  
GROUP BY ID;
```

Jack Queries

1.What are all the circuits that have been hosted in Italy?

```
select name, country  
from circuits  
where country = 'Italy';
```

2.How many different circuits have been utilized in each host country?

```
select count(distinct(name)), country  
from circuits  
group by country  
order by count(distinct(name)) desc;
```

3. To measure changes in Ferraris effective performance over time, the following query shows number of points per year, organized by year.

```
select year, sum(points)  
from constructor_results as cr  
join constructors as c  
on cr.constructorid = c.constructorsid  
join races as r  
on cr.raceid = r.raceid  
where c.name = 'Ferrari'  
group by year
```


order by year;

4. Because of changes in race formatting, points over time may not be a great indicator of performance trends, so the following query tracks number of race wins over time.

```
select year, count(position)
from constructor_standings as cs
join constructors as c
on c.constructorid = cs.constructorid
join races as r
on cs.raceid = r.raceid
where position = 1 and c.name = 'Ferrari'
group by year
order by year;
```

5. One historic racing constructor is Williams, who despite their recent struggles have a high number of total race wins. Show the constructors that have more race wins than Williams and their number of wins

```
select c.name, sum(position)
from constructor_standings as cs
join constructors as c
on c.constructorid = cs.constructorid
where cs.position = 1
group by c.name
having sum(position) > (select sum(position)
                        from constructor_standings as cs
                        join constructors as c
                        on c.constructorid = cs.constructorid
                        where position = 1 and c.name = 'Williams');
```

GUSTAVO:

- Country viz - Like the select statement informed us of the count distribution of races by country. This was a simple form of bar graph.
- Cityviz - Like the select statement informed us of the count distribution of races by city. For this it used a bar graph.(location).

- Presence Country - This visualization allows us to have a geographical understanding of the presence of f1 races worldwide. Using longitude and latitude (generated by Tableau) and introducing countries as marks and a light blue color to identify those countries.
- Danger - This visualization is based upon a filter on status selecting only for extremely dangerous circuits. Those being the one with statuses of collision, fatal accident and regular accidents. A count of statuses grouped by location in a bar graph fashion
- Successful constructors - is a graph that sums the points by constructors in a fashion illustrated by points and their corresponding values.

SIMON:

- Drivers by Nationality is a bar graph that shows how many drivers each country produced. Made by linking the database in mySQL to Tableau. It counts nationality and driver IDs.
- Most Wins in a Season by a Driver is another bar graph that shows the record of the most dominant seasons by a driver. It was made by linking the two services and goes to show which drivers can be the best regardless of time.
- Most Points by a Driver in a career shows recent dominance of drivers, as the scoring change went into effect at the turn of the century. Shows Lewis Hamilton being the all-time points leader.
- Most Common Races is a display of how many times certain races have been hosted and a representation of which races have pedigree. More races in a location, more likely it is a legacy race that will always be around.
- Williams vs Ferrari is a line graph overlaid with the two oldest teams, Ferrari and Williams. These teams have the most pedigree and have competed for decades.

Jared:

- Ferrari speed: This heat table shows the average fastest speed of Ferrari per race. Here the average of the fastest speed from the results is taken and joined with the races table through multiple other tables in order to group by the name of each race. The colors show the gradient of numbers from greatest to least.
- Red Bull position: This line graph relates to the Red Bull yearly position. Here the average of the position results are taken and joined with the constructors table through the races table in order to group by the name of the racing team, Red Bull, while specifying each year. A line is then used to visualize the trend of the data over time.
- Map: This map plots the precise location, by longitude and latitude. The color range also specifies the altitude of the tracks which can have a significant impact on each team's racing strategies.
- Hamilton lap times: This box plot chart shows Lewis Hamilton's lap times at the Indianapolis Motor Speedway. His median time is 75441 milliseconds or around 75 seconds.
- Longest track: This bar graph shows the average milliseconds per track, ordered from greatest to least, to give us the longest track. The longest track is the Autodromo Internazionale del Mugello in Tuscany, Italy.

Jack:

- The first visualization is a simple table corresponding to my first select statement. It shows all the race locations that have taken place in Italy, along with some additional info about the races. I simply filtered by country, and put the additional info I wanted under rows.
- This visualization corresponds to the second select statement I wrote. It shows in descending order the number of different race tracks that have been utilized in each country. For this, I put country as the columns, distinct circuit id as the rows, put the countries as colors, and then ordered the columns by descending count.
- This visualization matches with another select statement I wrote tracking Ferrari's success over time in formula 1. I inner joined four tables, put sum points as the rows and year as the columns, changed it to a line graph, and switched the color scheme to red for thematic purposes.
- This bubble chart compares constructors on a couple different metrics of success. First I joined 5 tables to link constructors to specific results. I put the constructors as color and label, then made cumulative points and average position the axes, illustrating both concentrated success and longevity in the sport.
- For this tree map, I joined the same tables as before and allocated size by percentage of total races won. I then placed total race wins as an additional label to make the map more interpretable. Finally I colored the map according to total points, highlighting teams whose overall success is not represented in their total race wins.

JUSTIN:

- First one is a graph of the number of status occurrences. These can range anywhere from being lapped to having an engine blow up. Shows how bad things can go.
- Second is the distribution of pitstop times. Having a low time is vital because every second a car isn't out on the track is very detrimental to the team.
- Third is the races by year. Very self explanatory, just shows it in graph form to digest easier.
- Fourth is a histogram of the average points per conductor. Can show dominance over time in a sport that has been around for 61 years.
- Fifth is the average position every lap. Those who race faster should mostly be towards the front. This could be affected by car unreliability and speed.