

Carreras de Formula 1

Soluciones a las consultas SQL:

1.

```
SELECT DISTINCT drivers.surname, drivers.forename
FROM drivers
      INNER JOIN results ON drivers.driverId = results.driverId
      INNER JOIN races ON results.raceId = races.raceId
WHERE year = 2016
```
2.

```
SELECT name
FROM constructors
WHERE constructorId IN (SELECT constructorId
                       FROM results
                       GROUP BY constructorId
                       HAVING COUNT(DISTINCT driverId) > 50)
```
3.

```
SELECT forename, surname
FROM drivers
WHERE driverId NOT IN (SELECT driverId
                      FROM results
                      WHERE positionOrder = 1)
```
4.

```
-- Solución con HAVING COUNT
SELECT drivers.surname, drivers.forename
FROM drivers
WHERE driverId IN (SELECT driverId
                  FROM results
                  WHERE raceId IN (SELECT raceId FROM races WHERE year=2017)
                  GROUP BY results.driverId
                  HAVING COUNT(distinct results.raceId) = (SELECT COUNT(*)
                                                            FROM races
                                                            WHERE year=2017))

-- Solución con NOT EXISTS
SELECT drivers.surname, drivers.forename
FROM drivers
WHERE NOT EXISTS (SELECT *
                  FROM races
                  WHERE year = 2017
                  AND NOT EXISTS (SELECT *
                                FROM results
                                WHERE results.driverId = drivers.driverId
                                AND races.raceId = results.raceId))
```
5.

```
SELECT circuits.circuitId, circuits.name, circuits.location, circuits.country, year
FROM races
      INNER JOIN circuits ON races.circuitId = circuits.circuitId
WHERE year BETWEEN 2015 AND 2017
ORDER BY circuits.circuitId
```
6.

```
SELECT *
FROM constructors
WHERE constructors.constructorId NOT IN (SELECT qualifying.constructorId
                                       FROM qualifying)
```
7.

```
SELECT drivers.surname, drivers.forename , tabla1.Ganado
FROM (SELECT driverId , COUNT(*) Ganado
FROM `results`
WHERE positionOrder = 1
GROUP BY driverId
HAVING COUNT(*) > 30 ) tabla1 JOIN drivers ON tabla1.driverId=drivers.driverId
```

-
8.

```
SELECT drivers.forename, drivers.surname, circuits.name, races.year, results.fastestLapSpeed
FROM drivers
    INNER JOIN results ON drivers.driverId = results.driverId
    INNER JOIN races ON races.raceId = results.raceId
    INNER JOIN circuits ON circuits.circuitId = races.circuitId
WHERE fastestLapSpeed >= ALL (SELECT fastestLapSpeed
                             FROM results)
```
-
9.

```
SELECT drivers.forename, drivers.surname, results.fastestLapSpeed
FROM drivers
    INNER JOIN results ON results.driverId = drivers.driverId
    INNER JOIN races ON races.raceId = results.raceId
WHERE races.year = 2009
    AND races.name = 'Japanese Grand Prix'
    AND results.fastestLapSpeed >= ALL (SELECT fastestLapSpeed
                                       FROM results
                                       INNER JOIN races ON
                                           ↳ races.raceId =
                                           ↳ results.raceId
                                       WHERE year = 2009
                                       AND name = 'Japanese Grand Prix')
```
-
10.

```
-- Solución con HAVING COUNT
SELECT drivers.forename, drivers.surname
FROM results
    INNER JOIN races ON races.raceId = results.raceId
    INNER JOIN drivers ON drivers.driverId = results.driverId
WHERE points > 0
    AND year = 2017
GROUP BY drivers.driverId, drivers.forename, drivers.surname
HAVING COUNT(*) = (SELECT COUNT(*)
                  FROM races
                  WHERE year = 2017)

-- Solución con NOT EXISTS
SELECT drivers.surname, drivers.forename
FROM drivers
WHERE NOT EXISTS ( SELECT *
                  FROM races
                  WHERE year = 2017 AND NOT EXISTS (SELECT *
                                                    FROM results
                                                    WHERE results.points > 0
                                                       AND drivers.driverId = results.driverId
                                                       AND results.raceId = races.raceId ))
```
-
11.

```
-- Solución con GROUP BY
SELECT drivers.forename, drivers.surname, circuits.name, races.year, COUNT(*)
FROM drivers
    INNER JOIN pitstops ON pitstops.driverId = drivers.driverId
    INNER JOIN races ON races.raceId = pitstops.raceId
    INNER JOIN circuits ON circuits.circuitId = races.circuitId
GROUP BY drivers.driverId, drivers.forename, drivers.surname, races.raceId, races.year,
↳ circuits.name
HAVING COUNT(*) >= ALL(SELECT COUNT(*)
                      FROM pitstops
                      GROUP BY driverId, raceId)

UNION

SELECT drivers.forename, drivers.surname, circuits.name, races.year, COUNT(*)
FROM drivers
    INNER JOIN pitstops ON pitstops.driverId = drivers.driverId
    INNER JOIN races ON races.raceId = pitstops.raceId
    INNER JOIN circuits ON circuits.circuitId = races.circuitId
GROUP BY drivers.driverId, drivers.forename, drivers.surname, races.raceId, races.year,
↳ circuits.name
HAVING COUNT(*) <= ALL(SELECT COUNT(*)
                      FROM pitstops
                      GROUP BY driverId, raceId)

-- Solución con el atributo stop
```
-

```
SELECT d.forename, d.surname, c.name, r.year, p.stop
FROM drivers d
      INNER JOIN pitstops p ON p.driverId = d.driverId
      INNER JOIN races r ON r.raceId = p.raceId
      INNER JOIN circuits c ON c.circuitId = r.circuitId
WHERE p.stop = (SELECT MAX(stop)
                FROM pitstops)
      OR p.stop = (SELECT MIN(stop)
                  FROM pitstops)
```

```
12. SELECT drivers.forename, drivers.surname, constructors.name
FROM drivers
      INNER JOIN qualifying ON qualifying.driverId = drivers.driverId
      INNER JOIN constructors ON qualifying.constructorId = constructors.constructorId
      INNER JOIN races ON races.raceId = qualifying.raceId
WHERE races.year = 2017
      AND races.name = 'Abu Dhabi Grand Prix'
      AND constructors.constructorId IN (SELECT qualifying.constructorId
                                         FROM qualifying
                                         INNER JOIN races
                                         ↪ ON
                                         ↪ races.raceId =
                                         ↪ qualifying.raceId
WHERE races.year = 2017
      AND races.name = 'Abu
      ↪ Dhabi Grand Prix'
      AND qualifying.q1 <> ''

      AND qualifying.q2 <> ''
      AND qualifying.q3 <> ''

GROUP BY
↪ qualifying.constructorId
HAVING COUNT(DISTINCT
↪ qualifying.driverId) =
↪ 2)

ORDER BY constructors.name ASC
```

```
13. SELECT drivers.forename, drivers.surname, races.name, races.year
FROM drivers
      INNER JOIN results ON results.driverId = drivers.driverId
      INNER JOIN races ON races.raceId = results.raceId
WHERE races.raceId IN (SELECT raceId
                       FROM results
                       INNER JOIN drivers ON drivers.driverId = results.driverId
                       WHERE drivers.nationality = "Russian")

      AND races.raceId IN (SELECT raceId
                           FROM results
                           INNER JOIN drivers ON drivers.driverId = results.driverId
                           WHERE drivers.nationality = "Polish")
```

```
14. SELECT drivers.forename, drivers.surname, SUM(results.laps)
FROM drivers
      INNER JOIN results ON results.driverId = drivers.driverId
      INNER JOIN races ON races.raceId = results.raceId
WHERE races.year = 2011
GROUP BY drivers.driverId, drivers.forename, drivers.surname
HAVING SUM(results.laps) > (SELECT AVG(nLaps)
                           FROM (SELECT SUM(laps) AS nLaps
                                FROM results
                                INNER JOIN races ON
                                ↪ races.raceId =
                                ↪ results.raceId
                                WHERE year = 2010
                                GROUP BY driverId) t)
```

```
15. SELECT name, year
FROM races
WHERE raceId IN (SELECT raceId FROM qualifying)
      AND raceId NOT IN (SELECT raceId FROM pitstops)
```

- ```

16. -- Solución con NOT EXISTS
SELECT DISTINCT nationality
FROM drivers
WHERE NOT EXISTS (SELECT *
 FROM races
 WHERE name = "Australian Grand Prix"
 AND NOT EXISTS (SELECT *
 FROM results
 WHERE results.driverId = drivers.driverId
 AND results.raceId = races.raceId))

-- Solución con COUNT
SELECT drivers.nationality
FROM drivers
 INNER JOIN results ON results.driverId = drivers.driverId
 INNER JOIN races ON races.raceId = results.raceId
WHERE races.name = "Australian Grand Prix"
GROUP BY drivers.driverId, drivers.nationality
HAVING COUNT(*) = (SELECT COUNT(*)
 FROM races
 WHERE name = "Australian Grand Prix")

17. DELETE FROM qualifying
WHERE q1 = '' AND q2 = '' AND q3 = ''

18. SELECT *
FROM constructors
WHERE constructorId IN (SELECT results.constructorId
 FROM results
 INNER JOIN races ON races.raceId =
 ↳ results.raceId
 WHERE races.year BETWEEN 2003 AND 2010
 AND results.positionOrder = 1
 GROUP BY results.constructorId
 HAVING COUNT(*) > 5)
AND constructorId NOT IN (SELECT results.constructorId
 FROM results
 INNER JOIN races ON races.raceId =
 ↳ results.raceId
 WHERE races.year > 2010)

19. -- Solución con NOT EXISTS
SELECT name, year
FROM races
WHERE NOT EXISTS(SELECT *
 FROM status
 WHERE statusId BETWEEN 2 AND 7
 AND NOT EXISTS(SELECT *
 FROM results
 WHERE results.raceId = races.raceId
 AND results.statusId = status.statusId))

-- Solución con COUNT
SELECT races.raceId, races.name, races.year, COUNT(DISTINCT results.statusId)
FROM races
 INNER JOIN results ON results.raceId = races.raceId
WHERE results.statusId BETWEEN 2 AND 7
GROUP BY races.raceId, races.name, races.year
HAVING COUNT(DISTINCT results.statusId) = (SELECT COUNT(*)
 FROM status
 WHERE statusId BETWEEN 2 AND 7)

20. SELECT forename, surname, circuits.name, races.year, milliseconds
FROM drivers, pitstops, races, circuits
WHERE drivers.driverId=pitstops.driverId
 AND pitstops.raceId=races.raceId
 AND races.circuitId=circuits.circuitId
 AND drivers.nationality='Spanish'
 AND milliseconds = (SELECT MIN(milliseconds)

```

```
FROM pitstops JOIN drivers ON pitstops.driverId=drivers.driverId
WHERE drivers.nationality='Spanish');
```

- 
21. 

```
SELECT DISTINCT name
FROM constructors
 INNER JOIN results ON results.constructorId = constructors.constructorId
 INNER JOIN drivers ON drivers.driverId = results.resultId
WHERE constructors.nationality LIKE 'Italian'
 AND drivers.nationality LIKE 'Italian'
```
- 
22. 

```
SELECT drivers.forename, drivers.forename, COUNT(*)
FROM drivers
 INNER JOIN results ON results.driverId = drivers.driverId
 INNER JOIN status ON status.statusId = results.statusId
WHERE status.status LIKE 'Accident'
GROUP BY drivers.driverId, drivers.forename, drivers.surname
HAVING COUNT(*) >= ALL (SELECT COUNT(*)
 FROM results
 INNER JOIN status ON status.statusId = results.statusId
 WHERE status.status LIKE 'Accident'
 GROUP BY results.driverId)
```
- 
23. 

```
SELECT drivers.forename, drivers.surname
FROM drivers
WHERE NOT EXISTS (SELECT *
 FROM races
 WHERE year = 2015
 AND NOT EXISTS (SELECT *
 FROM qualifying
 WHERE position <= 10
 AND qualifying.driverId = drivers.driverId
 AND qualifying.raceId = races.raceId))
```
- 
24. 

```
SELECT name
FROM constructors
WHERE constructorId NOT IN (SELECT constructorId
 FROM results JOIN drivers ON results.driverId=drivers.driverId
 WHERE nationality='Italian')
AND nationality='Italian';
```
- 
25. 

```
SELECT *
FROM constructors C
WHERE NOT EXISTS (SELECT *
 FROM races R
 WHERE year=2006
 AND NOT EXISTS (SELECT *
 FROM qualifying
 WHERE qualifying.raceId=R.raceId
 AND qualifying.constructorId=C.constructorId
 AND position <= 10))
```
- 
26. 

```
SELECT forename, surname, suma
FROM (SELECT driverId, year, SUM(points) as suma
 FROM results JOIN races ON results.raceId=races.raceId
 WHERE year BETWEEN 1990 AND 2000
 GROUP BY driverId, year
 HAVING SUM(points)>0) AS puntuaciones
JOIN drivers ON puntuaciones.driverId=drivers.driverId
WHERE suma >= ALL (SELECT SUM(points)
 FROM results JOIN races ON results.raceId=races.raceId
 WHERE year BETWEEN 1990 AND 2000
 GROUP BY driverId, year
 HAVING SUM(points)>0)
```
-

- 
27. 

```
SELECT DISTINCT d.forename, d.surname, r.name, r.year
FROM drivers d
 INNER JOIN results ON results.driverId = d.driverId
 INNER JOIN races r ON r.raceId = results.raceId
WHERE results.positionOrder = 1
 AND EXISTS (SELECT *
 FROM qualifying
 WHERE qualifying.raceId = r.raceId
 AND qualifying.driverId = d.driverId
 AND position > 10)
```
- 
28. 

```
SELECT drivers.forename, drivers.surname, COUNT(*)
FROM drivers
 INNER JOIN pitstops ON pitstops.driverId = drivers.driverId
 INNER JOIN races ON races.raceId = pitstops.raceId
WHERE races.year = 2013
GROUP BY pitstops.driverId, pitstops.raceId, drivers.forename, drivers.surname
HAVING COUNT(*) >= ALL (SELECT COUNT(*)
 FROM pitstops
 GROUP BY pitstops.driverId, pitstops.raceId)
```
- 
29. 

```
SELECT drivers.forename, drivers.surname
FROM drivers
WHERE NOT EXISTS (SELECT *
 FROM races
 WHERE year = 2017
 AND NOT EXISTS (SELECT *
 FROM results
 WHERE positionOrder <= 10
 AND results.driverId = drivers.driverId
 AND results.raceId = races.raceId))
```
- 
30. 

```
SELECT *
FROM constructors C
WHERE NOT EXISTS (SELECT *
 FROM races R
 WHERE year=2006
 AND NOT EXISTS (SELECT *
 FROM qualifying
 WHERE qualifying.raceId=R.raceId
 AND qualifying.constructorId=C.constructorId
 AND position <= 10))
```
- 
31. 

```
SELECT forename, surname, suma
FROM (SELECT driverId, year, SUM(points) as suma
 FROM results JOIN races ON results.raceId=races.raceId
 WHERE year BETWEEN 1990 AND 2000
 GROUP BY driverId, year
 HAVING SUM(points)>0) AS puntuaciones
JOIN drivers ON puntuaciones.driverId=drivers.driverId
WHERE suma >= ALL (SELECT SUM(points)
 FROM results JOIN races ON results.raceId=races.raceId
 WHERE year BETWEEN 1990 AND 2000
 GROUP BY driverId, year
 HAVING SUM(points)>0)
```
- 
32. 

```
SELECT drivers.forename, drivers.surname, COUNT(*)
FROM drivers
 INNER JOIN pitstops ON pitstops.driverId = drivers.driverId
 INNER JOIN races ON races.raceId = pitstops.raceId
WHERE races.year = 2013
GROUP BY pitstops.driverId, pitstops.raceId, drivers.forename, drivers.surname
HAVING COUNT(*) >= ALL (SELECT COUNT(*)
 FROM pitstops
 GROUP BY pitstops.driverId, pitstops.raceId)
```
-

Esta obra está bajo una licencia Creative Commons “Atribución-NoComercial-CompartirIgual 3.0 No portada”.

