



## Carreras de Formula 1

La Federación Internacional de Automovilismo (FIA), como parte de su esfuerzo continuo para optimizar y mejorar la calidad de las carreras de Fórmula 1, ha tomado la decisión de recopilar y centralizar en una base de datos toda la información relevante relacionada con estos eventos de competición. Para ello, se ha diseñado una base de datos robusta y estructurada que consta de diversas tablas, las cuales tienen la función de almacenar de manera organizada toda la información referente a las carreras que se llevan a cabo durante los fines de semana en diferentes países del mundo. Estas tablas están diseñadas para capturar datos esenciales sobre cada carrera, como los equipos participantes, los pilotos, los resultados, las clasificaciones, y otros aspectos logísticos y técnicos del evento. La figura 1 ilustra el modelo relacional de esta base de datos, proporcionando una visión clara de cómo se interconectan las diferentes tablas para ofrecer una representación integral de toda la información gestionada por la FIA en el contexto de la Fórmula 1.

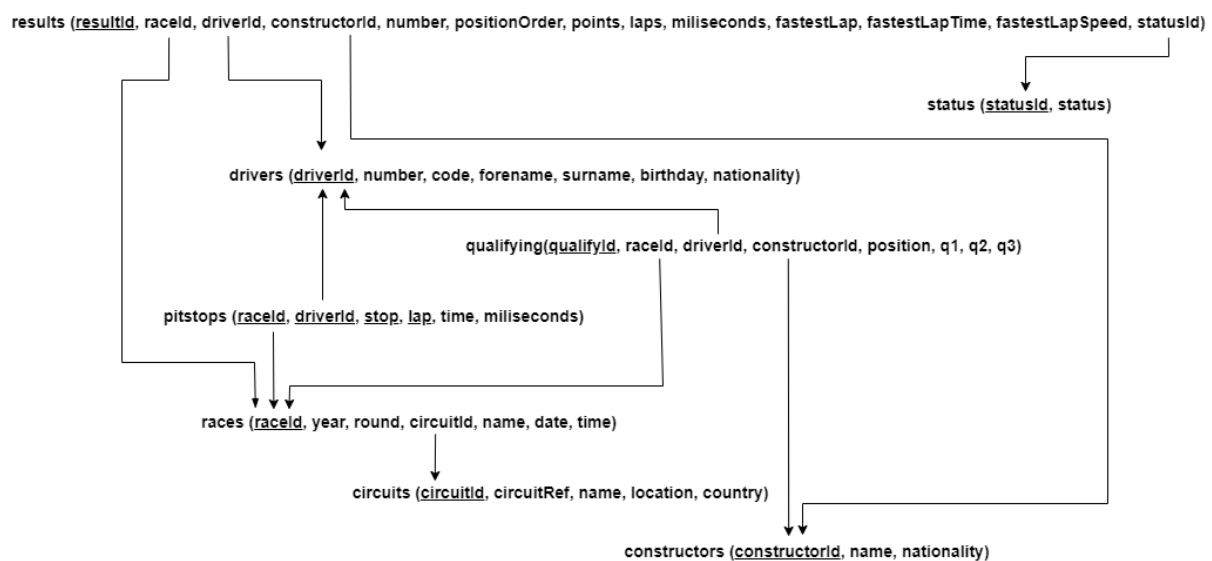


Figura 1: Modelo relacional de la Fórmula 1.

A continuación se describe el contenido de las distintas tablas de la base de datos:

- **drivers:** Esta tabla almacena información sobre los pilotos que han participado en las carreras, incluyendo detalles como su nombre (*forename*), apellidos (*surname*), nacionalidad (*nationality*), y otros datos relevantes para su identificación.
- **results:** Contiene los resultados obtenidos por cada piloto en las carreras, registrando su posición final (*positionOrder*), puntos obtenidos (*points*) y cualquier otro dato que refleje su desempeño en la competición.
- **races:** En esta tabla se almacena toda la información pertinente a cada carrera, como el nombre del Gran Premio (*name*), la fecha en que se disputó (*date*) y el circuito en el que tuvo lugar.
- **circuits:** Incluye información detallada sobre los circuitos en los que se llevan a cabo las carreras, especificando su ubicación (*location*) y nombre (*name*) entre otros.
- **constructors:** Esta tabla registra a los constructores (equipos) que participan en las carreras. Generalmente, cada equipo compite con dos pilotos en cada carrera.
- **qualifying:** Almacena los datos sobre las sesiones de calificación que determinan el orden de salida de los pilotos antes de cada carrera. Normalmente, las rondas de calificación se disputan en tres tandas (*q1*, *q2* y *q3*), en las que los pilotos más lentos van siendo eliminados sucesivamente.
- **pitstops:** Registra las paradas que realizan los pilotos durante las carreras, ya sea para cambiar neumáticos, repostar combustible o realizar ajustes en el coche, detallando el momento (*lap*) y la duración de cada una (*milliseconds*).

- *status*: Esta tabla especifica el estado final de cada piloto al término de una carrera, indicando si finalizaron en condiciones normales, si perdieron vueltas, si tuvieron problemas técnicos o si abandonaron por alguna razón.

Adicionalmente, se hace constar que este modelo tiene añadidas las siguientes restricciones:

- Todos los identificadores se presentan de forma numérica.
- Los nombres, apellidos, nacionalidades y localizaciones no podrán exceder en ningún caso los 250 caracteres.
- Los campos que indican número de vuelta, puntos y velocidades tendrán que adaptarse como valores INT, FLOAT o DOUBLE según corresponda.
- Existen tres tipos adicionales de valores con formato temporal como son *time* de tipo TIME, *date* y *birthday* de tipo DATE y *year* de tipo YEAR.



## Consultas SQL:

1. Obtener el nombre y apellidos de los pilotos españoles.
2. Obtener todos los datos de los circuitos alemanes.
3. Obtener los países en los que se disputaron carreras en el año 2010.
4. Obtener el nombre de los pilotos que han participado en al menos 1 carrera del año 2016.
5. Nombre de los constructores con los que han disputado carreras más de 50 pilotos diferentes.
6. Nombre y apellidos de los pilotos que nunca han ganado una carrera.
7. Obtener el nombre y apellidos de los pilotos que durante el año 2017 han participado en todas las carreras.
8. Obtener el nombre, localización, país y año para cada circuito de las carreras que se han disputado entre 2015 y 2017, ordenado por el id del circuito.
9. Obtener los constructores que no han participado en alguna clasificación.
10. Obtener nombres, apellidos de los pilotos que han ganado más de 30 grandes premios así como el número de grandes premios que han ganado.
11. Nombre y apellidos del piloto que obtuvo la vuelta con velocidad media más alta así como el circuito y el año en el que se obtuvo.
12. Obtener el nombre, apellidos y la velocidad media del piloto que obtuvo la vuelta con velocidad media más alta en el gran premio de Japón de 2009.
13. Obtener el nombre de los pilotos que durante el año 2017 consiguieron puntos en todas las carreras.
14. Obtener el nombre de los pilotos, el circuito y el total de paradas, para aquellos pilotos que entre todos los grandes premios disputados han realizado en alguno de ellos el mayor número de paradas y también los que han realizado el menor número de ellas.
15. De entre todos los pilotos que han participado en todas las rondas de clasificación (Q1, Q2 y Q3) del gran premio de Abu Dhabi de 2017 (`qualifying.q1 <>' '` AND `qualifying.q2 <>' '` AND `qualifying.q3 <>' '`), obtener el nombre de los pilotos y el id de los equipos, para aquellos equipos que tienen a sus dos pilotos en esa situación.
16. Obtener el nombre y apellido de los pilotos y el nombre de aquellas carreras en las que han participado pilotos rusos y polacos.
17. Obtener el nombre y apellidos de los pilotos y el número de vueltas totales recorridas en el año 2011 siempre y cuando sea mayor que la media del número de vueltas totales recorridas el año anterior por todos los pilotos.
18. Obtener el nombre y año de las carreras en las que se disputó una clasificación (`qualifying`) pero no se realizaron pitstops.
19. Obtener la nacionalidad de los pilotos que han disputado todas las ediciones del gran premio 'Australian Grand Prix'.
20. Eliminar de la tabla `qualifying` aquellas tuplas donde un piloto no haya participado en la clasificación.
21. Obtener aquellos constructores que habiendo ganado más de 5 carreras entre 2003 y 2010, no hayan participado en ninguna carrera desde el siguiente año.
22. Obtener el nombre de la carrera y el año en la que tuvo lugar todos los tipos de incidentes que se enumeran a continuación: descalificación, accidente, colisión, fallo de motor, caja de cambios y transmisión (`statusId` de 2 a 7).

23. Obtener nombre y apellidos del piloto, el nombre del circuito, el año de la carrera donde un piloto español obtuvo el tiempo de parada más pequeño (atributo `milliseconds`). Incluya este atributo en la salida de la consulta.
24. Codifique una consulta que obtenga el nombre de aquellos constructores que sean italianos (`nationality = 'Italian'`) con los que hayan disputado carreras al menos un piloto italiano.
25. Codifique una consulta que obtenga el nombre y apellidos del piloto que más accidentes (`status.status = 'Accident'`) ha tenido. Mostrar también el número de accidentes.
26. Codifique una consulta que obtenga el nombre y apellidos de los pilotos que hayan calificado entre los 10 primeros puestos (`position <= 10`) de todas las carreras del año 2015.
27. Obtener una lista con los nombres de aquellos constructores italianos ('Italian' en inglés) que nunca han competido con pilotos italianos.
28. Obtener toda la información de los constructores que en todas las carreras del año 2006, consiguieron que alguno de sus pilotos quedara entre los diez primeros de la clasificación.
29. Obtener nombre y apellidos del piloto que acabó la competición con más puntos entre los años 1990 y 2000, así como dicha suma de puntos.
30. Codifique una consulta que obtenga el nombre y apellidos de los pilotos que ganaron una carrera (`results.positionOrder = 1`) sin haber estado clasificados entre los 10 primeros pilotos (`qualifying.position >10`). Mostrar además el nombre de la carrera y el año en la que lo consiguieron.
31. Codifique una consulta que obtenga el nombre y apellidos del piloto que realizó más *pitstops* en una carrera del año 2013. Mostrar también el número de *pitstops*.
32. Codifique una consulta que obtenga el nombre y apellidos de los pilotos que hayan quedado entre los 10 primeros puestos (`positionOrder <= 10`) de todas las carreras del año 2017.
33. Obtener toda la información de los constructores que en todas las carreras del año 2006, consiguieron que alguno de sus pilotos quedara entre los diez primeros de la clasificación.
34. Obtener nombre y apellidos del piloto que acabó la competición con más puntos entre los años 1990 y 2000, así como dicha suma de puntos.
35. Nombre y apellidos del piloto que realizó más *pitstops* en una carrera del año 2013. Mostrar también el número de *pitstops*.



#### Procedimientos almacenados:

1. Crea un procedimiento almacenado de nombre `getRacesInAYear`, que obtenga como salida los nombres de las carreras y el número total de constructores que han participado en la carrera, para un año concreto que se pasará como parámetro de entrada. En el procedimiento no se definirán parámetros de salida.
2. Se quiere testear los mensajes que se le envían a los pilotos por pantalla en plena carrera. Para ello se quiere crear un procedimiento de nombre `getOnRaceMessages`, que reciba un código de mensaje, e indique en un parámetro de salida el mensaje concreto teniendo en cuenta las siguientes condiciones:
  - E01 = Error en la presión de las ruedas
  - E02 = Pinchazo
  - E03 = Temperatura alta en el motor
  - E04 = Frenos sobre-calentados
  - E05 = Error presión del aceite
  - Cualquier otro valor = Error de comando
3. Crea un procedimiento almacenado para obtener los pilotos y los circuitos que ganaron carreras de un año concreto (como argumento) con un constructor del mismo país que el piloto.
4. Escribe un procedimiento que realice la selección de aquellos pilotos que en el año que se le pase como parámetro de entrada, obtuvieron un primer, un segundo y un tercer puesto.
5. Procedimiento almacenado de nombre `getsConstructoresYPilotos`, que tomando como entrada un año concreto, obtenga como salida los nombres de todos los pilotos que participaron en el mundial ese año y el equipo de constructores con el cual compitieron. Adicionalmente, también se quiere obtener el total de puntos que obtuvo cada piloto en el mundial.
6. Codifique un procedimiento almacenado denominado `getNumberOfVictories` que disponga de un parámetro de entrada `type`. Si `type = 'nationality'`, el procedimiento listará el número de victorias que han obtenido cada una de las nacionalidades de la tabla `drivers`. Por contra, si `type` toma cualquier otro valor, el procedimiento listará el número de victorias que han obtenidos cada uno de los constructores (`constructors.name`) existentes en la base de datos. Los resultados deberá estar ordenados de mayor a menor número de victorias.
7. Añade una nueva columna en la tabla `drivers` que se llame años en activo. A continuación escribir un procedimiento que, haciendo uso de un `Cursor`, calcule el número total de años que ha estado un piloto compitiendo y actualice la tabla.
8. Crea un procedimiento que reciba una nacionalidad como parámetro de entrada y realice una consulta sobre la tabla `drivers` para obtener todos los pilotos de esa nacionalidad. Los nombres y apellidos de estos pilotos deberán devolverse en un parámetro de salida de tipo `TEXT` separados por comas (,).





## Funciones almacenadas:

1. Escribe una función que devuelva el número de puntos que se ha conseguido el campeón de ese año en cada mundial.
2. Escribe una función que devuelva el valor medio de puntos por año que ha conseguido un determinado constructor que se recibirá como parámetro de entrada. El parámetro de entrada será el nombre del constructor.
3. Escribe una función que dado el `driverId` de un piloto devuelva el número total de años en activo que ha estado compitiendo.
4. Codifique una función almacenada llamada `diffPoints` que reciba los identificadores de dos pilotos como parámetros (`driver1` y `driver2`) y devuelva la diferencia de puntos totales conseguidos por `driver1` con respecto de `driver2`. Tenga en cuenta que el valor será positivo si `driver1` ha conseguido más puntos que `driver2` y negativo en caso contrario.



## Triggers:

1. Desarrolle todos los triggers necesarios para impedir que en una misma carrera puedan participar más de 2 pilotos con un constructor. Se considera que participar en una carrera es aparecer en la tabla `results` (se ignora por tanto la tabla `qualifying`).
2. Crea una tabla `crashes` con la siguiente estructura:

<code>crashId</code>	<code>driverId</code>	<code>descripción</code>
----------------------	-----------------------	--------------------------

- Crea un trigger que cuando se inserte una fila en la tabla `results` con el parámetro `statusId` de valores 3 ó 4 rellene la tabla `crashes` con la información correspondiente.
3. Crea un trigger para impedir que en un mismo año un piloto participe en carreras con dos constructores diferentes. Se considera que un piloto habrá participado en una carrera si aparece en la tabla `results`.
  4. Cree una nueva tabla `sponsors` en la base de datos que almacene como atributos, su identificador propio, su nombre, tipo, el identificador de la carrera que patrocina (un patrocinador solo puede patrocinar una carrera) y el dinero que aporta anualmente. Existen dos tipos de patrocinadores: oficiales, deben aportar una cantidad igual o superior a 5M de euros, y cooficiales, para cantidades inferiores. Diseñe un trigger que clasifique al sponsor y rellene el atributo `type` de su correspondiente tabla.
  5. A partir de ahora, los patrocinadores van a poder patrocinar varias carreras. Realice las modificaciones que considere necesarias en el `schema` de la base de datos y actualice los triggers para garantizar que el tipo de patrocinador sea siempre correcto.

Esta obra está bajo una licencia Creative Commons “Atribución-NoComercial-CompartirIgual 4.0 Internacional”.

