

## Carreras de Formula 1

La Federación Internacional de Automovilismo (FIA) está tratando de recopilar toda la información que han recogido en los últimos años durante las carreras de Fórmula 1. Con todos los datos que han conseguido recabar quieren realizar diversos estudios con el fin de mejorar el espectáculo. No obstante, antes de poder realizar ningún estudio se necesita crear una base de datos donde se pueda almacenar la información.

Para ello, la compañía nos ha suministrado el Modelo Relacional que se muestra en la figura 1, con el fin de facilitar las pautas necesarias a tener en cuenta para poder crear la base de datos.

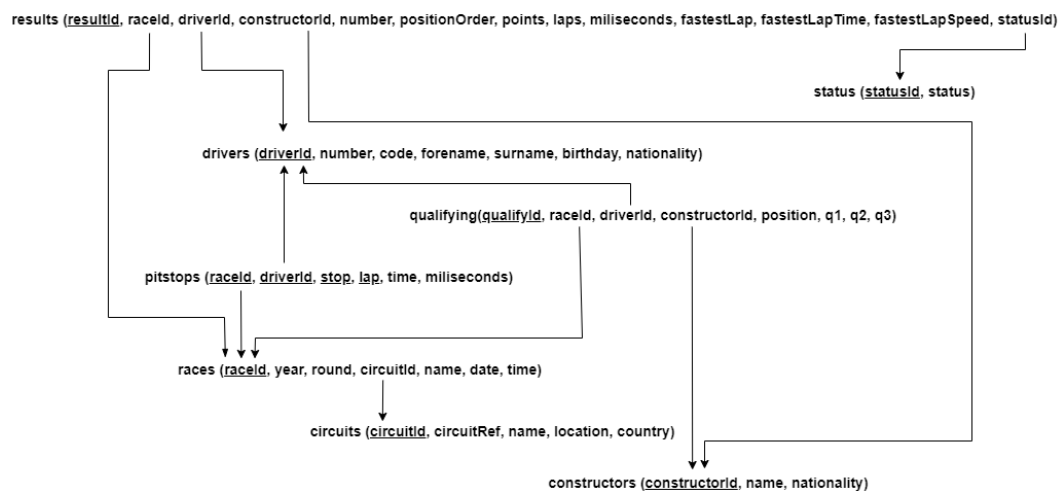


Figura 1: Modelo relacional de la Fórmula 1.

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.

Se desean implementar los siguientes procedimientos, funciones y triggers:

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
  - E54 = Error presión del aceite
  - Cualquier otro valor = Error de comando
3. Crea un procedimiento un procedimiento que reciba el nombre de un país como parámetro de entrada y realice una consulta sobre la tabla **drivers** para obtener todos los pilotos de esa nacionalidad.
  4. Escribe una función que devuelva el número de puntos que se ha conseguido el campeón de ese año en cada mundial.
  5. 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.
  6. Escribe una función que dado el **driverId** de un piloto devuelva el número total de años en activo que ha estado compitiendo.
  7. Añade una nueva columna en la tabla **drivers** que se llame años en activo. A continuación escribir un procedimiento que calcule el número total de años que ha estado un piloto compitiendo y actualice la tabla. Este procedimiento hará uso de la función creada en un ejercicio anterior.
  8. A partir de este momento la FIA no va a permitir que haya equipos con más de 2 pilotos, por ello se debe desarrollar un trigger que impida que se puedan incorporar en un mismo equipo más de 2 pilotos a la base de datos. Para ello, se debe impedir toda operación que haga que un 3 o más pilotos pasen a formar parte del mismo equipo de constructores, ya sea mediante inserción o actualización de los datos.
  9. Crea una tabla **crashes** con la siguiente estructura:

driverId	crashId	Descripción
----------	---------	-------------

Crea un trigger que cuando se inserte una fila en la tabla **results** con el parámetro **statusId** de valores 1, 2 o 3 rellene la tabla **crashes** con la información correspondiente.

10. Crea un trigger para impedir que en un mismo año un piloto esté en dos equipos de constructores diferentes.
11. 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.
12. 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.

13. 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.
14. 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 y el dinero que aporta anualmente. Luego, diseñe un trigger para la nueva tabla *sponsors*. Cada carrera puede contar con varios patrocinadores, los cuales se dividen en dos tipos, oficiales (deben aportar una cantidad igual o superior a 5M de euros anuales) y cooficiales (cantidades inferiores). Diseñe un trigger que clasifique al sponsor y rellene el atributo *type* de su correspondiente tabla. Ten en cuenta que un sponsor puede cambiar de año en año la cuantía que aporta.
15. 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.
16. Codifique un procedimiento llamado **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.

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

