

```
SET STATISTICS TIME ON;
SELECT
    pes.nombre_piloto AS Nombre,
    pes.apellido1_piloto AS Apellido,
    pes.superlicencia AS Superlicencia,
    esc.nombre_escuderia AS Escuderia,
    cir.nombre_circuito AS Circuito,
    gp.fecha_carrera AS Fecha_Carrera
FROM
    tb_piloto_escuderia_circuito_participa_gran_premio pecp
JOIN
    tb_pilotos pes ON pecp.superlicencia = pes.superlicencia
JOIN
    tb_escuderias esc ON pecp.id_escuderia = esc.id_escuderia
JOIN
    tb_circuitos cir ON pecp.id_circuito = cir.id_circuito
JOIN
    tb_gran_premio gp ON pecp.id_gp = gp.id_gp
WHERE
    pes.num_victorias > 0 and esc.id_escuderia like 'E007'
```

89 %

Results Messages

Tiempo de análisis y compilación de SQL Server:
Tiempo de CPU = 6 ms, tiempo transcurrido = 6 ms.

(1 row affected)

Tiempos de ejecución de SQL Server:
Tiempo de CPU = 0 ms, tiempo transcurrido = 0 ms.

Completion time: 2024-04-22T13:25:13.3831246+02:00

```
SET STATISTICS TIME ON;
```

Con este comando se muestran las estadísticas de tiempo de una instrucción, en este caso queremos saber el tiempo que tarda una consulta en ejecutarse, por ello el TIME.

Al realizar la consulta, el tiempo que tarda al ejecutarla es de 6ms

Al crear un índice que abarque estas cuatro columnas, podemos optimizar las consultas que buscan información sobre la participación de pilotos en diferentes Grandes Premios, en circuitos específicos y con escuderías particulares.

```
SELECT
    pes.nombre_piloto AS Nombre,
    pes.apellido_piloto AS Apellido,
    pes.superlicencia AS Superlicencia,
    esc.nombre_escuderia AS Escuderia,
    cir.nombre_circuito AS Circuito,
    gp.fecha_carrera AS Fecha_Carrera
FROM
    tb_piloto_escuderia_circuito_participa_gran_premio pecp
JOIN
    tb_pilotos pes ON pecp.superlicencia = pes.superlicencia
JOIN
    tb_escuderias esc ON pecp.id_escuderia = esc.id_escuderia
JOIN
    tb_circuitos cir ON pecp.id_circuito = cir.id_circuito
JOIN
    tb_gran_premio gp ON pecp.id_gp = gp.id_gp
WHERE
    pes.num_victorias > 0 and esc.id_escuderia like 'E007'

CREATE INDEX idx_tbpilesccirpartgp_composite ON tb_piloto_escuderia_circuito_participa_gran_premio(superlicencia, id_gp, id_circuito, id_escuderia);
```

89 %

Results Messages

Tiempo de análisis y compilación de SQL Server:
Tiempo de CPU = 0 ms, tiempo transcurrido = 0 ms.
Tiempo de análisis y compilación de SQL Server:
Tiempo de CPU = 14 ms, tiempo transcurrido = 14 ms.

(1 row affected)

Tiempos de ejecución de SQL Server:
Tiempo de CPU = 0 ms, tiempo transcurrido = 0 ms.

Completion time: 2024-04-22T13:26:28.0924915+02:00

Al crear este índice, como podemos comprobar el tiempo de respuesta es mayor, por lo que no sirve.

```
DROP INDEX idx_tbpilesccirpartgp_composite ON tb_piloto_escuderia_circuito_participa_gran_premio;
```

Como lo comprobado no nos sirve, decidimos borrar el index

```
SELECT
    m.id_motor,
    m.marca AS Marca_Motorista,
    e.id_escuderia,
    e.nombre_escuderia AS Nombre_Escuderia,
    e.nombre_jefe AS Jefe_Escuderia
FROM
    tb_motoristas m
INNER JOIN
    tb_escuderias e ON m.id_motor = e.id_motor
WHERE
    m.marca = 'MERCEDES';
```

89 %

Results Messages

Tiempo de análisis y compilación de SQL Server:
Tiempo de CPU = 0 ms, tiempo transcurrido = 7 ms.

(4 rows affected)

Tiempos de ejecución de SQL Server:
Tiempo de CPU = 0 ms, tiempo transcurrido = 0 ms.

Completion time: 2024-04-22T13:32:56.9023095+02:00

Ejecutamos una consulta sobre las tablas motoristas y escudería, observamos un tiempo de 7ms

```
CREATE INDEX idx_motorista_escuderia ON tb_motoristas (marca);
```

Decidimos crear un índice sobre tabla motorista en el campo de marca, ya que consideramos que muchas de las consultas van a ser sobre la marca del coche a

```
SELECT
    m.id_motor,
    m.marca AS Marca_Motorista,
    e.id_escuderia,
    e.nombre_escuderia AS Nombre_Escuderia,
    e.nombre_jefe AS Jefe_Escuderia
FROM
    tb_motoristas m
INNER JOIN
    tb_escuderias e ON m.id_motor = e.id_motor
WHERE
    m.marca = 'MERCEDES';
```

%

Results Messages

Tiempo de análisis y compilación de SQL Server:
Tiempo de CPU = 0 ms, tiempo transcurrido = 0 ms.
Tiempo de análisis y compilación de SQL Server:
Tiempo de CPU = 5 ms, tiempo transcurrido = 5 ms.

(4 rows affected)

Tiempos de ejecución de SQL Server:
Tiempo de CPU = 0 ms, tiempo transcurrido = 0 ms.

Completion time: 2024-04-22T13:33:58.6551691+02:00

Una vez creado el índice podemos ver que el tiempo de la consulta ha disminuido a 5ms

```
SELECT
    nombre_piloto,
    apellido1_piloto,
    apellido2_piloto,
    num_victorias,
    (SELECT MAX(num_victorias) FROM tb_pilotos) - num_victorias AS diferencia_victorias
FROM
    tb_pilotos;
```

.00 %

Results Messages

Tiempo de análisis y compilación de SQL Server:
Tiempo de CPU = 0 ms, tiempo transcurrido = 1 ms.

(20 rows affected)

Tiempos de ejecución de SQL Server:
Tiempo de CPU = 0 ms, tiempo transcurrido = 0 ms.

Completion time: 2024-05-05T11:06:59.7058746+02:00

Podemos ver que al momento de ejecutar la consulta son 1ms

```
SELECT
    nombre_piloto,
    apellido1_piloto,
    apellido2_piloto,
    num_victorias,
    (SELECT MAX(num_victorias) FROM tb_pilotos) - num_victorias AS diferencia_victorias
FROM
    tb_pilotos;

CREATE INDEX idx_num_victorias ON tb_pilotos (num_victorias);
```

100 %

Results Messages

Tiempo de análisis y compilación de SQL Server:
Tiempo de CPU = 0 ms, tiempo transcurrido = 0 ms.

(20 rows affected)

Tiempos de ejecución de SQL Server:
Tiempo de CPU = 0 ms, tiempo transcurrido = 0 ms.

Completion time: 2024-05-05T11:08:59.9424088+02:00

Y una vez que hemos creado el índice sobre la columna num_victorias el tiempo disminuye

```
SELECT
    nombre_piloto,
    apellido1_piloto,
    apellido2_piloto,
    num_victorias,
    num_carreras,
    num_victorias - num_carreras AS diferencia_victorias_carreras
FROM
    tb_pilotos
ORDER BY
    diferencia_victorias_carreras DESC;
```

100 %

Messages

Tiempo de análisis y compilación de SQL Server:
Tiempo de CPU = 0 ms, tiempo transcurrido = 0 ms.

Tiempos de ejecución de SQL Server:
Tiempo de CPU = 0 ms, tiempo transcurrido = 5 ms.

Completion time: 2024-05-05T11:13:51.6447508+02:00

El tiempo en esta consulta son de 5ms, también utiliza el índice creado anteriormente pero como contiene otros campos el tiempo aumenta

```
SELECT
    nombre_piloto,
    apellido1_piloto,
    apellido2_piloto,
    num_victorias,
    num_carreras,
    num_victorias - num_carreras AS diferencia_victorias_carreras
FROM
    tb_pilotos
ORDER BY
    diferencia_victorias_carreras DESC;

CREATE INDEX idx_num_carreras ON tb_pilotos (num_carreras);
```

100 %

Results Messages

Tiempo de análisis y compilación de SQL Server:
Tiempo de CPU = 0 ms, tiempo transcurrido = 0 ms.
Tiempo de análisis y compilación de SQL Server:
Tiempo de CPU = 0 ms, tiempo transcurrido = 0 ms.

(20 rows affected)

Tiempos de ejecución de SQL Server:
Tiempo de CPU = 0 ms, tiempo transcurrido = 0 ms.

Completion time: 2024-05-05T11:15:50.7290504+02:00

Creamos un índice sobre el num:_carreras y podemos ver que el tiempo disminuye a 0ms

```
SELECT
    nombre_circuito,
    km_circuitos,
    vuelta_rapida
FROM
    tb_circuitos
WHERE
    km_circuitos > 6.0
ORDER BY
    km_circuitos DESC;
```

100 %

Results Messages

Tiempo de análisis y compilación de SQL Server:
Tiempo de CPU = 14 ms, tiempo transcurrido = 14 ms.
Tiempo de análisis y compilación de SQL Server:
Tiempo de CPU = 0 ms, tiempo transcurrido = 0 ms.

(10 rows affected)

Tiempos de ejecución de SQL Server:
Tiempo de CPU = 0 ms, tiempo transcurrido = 1 ms.

Completion time: 2024-05-05T11:29:51.9332814+02:00

Vemos que el tiempo es de 14ms

```
SELECT
    nombre_circuito,
    km_circuitos,
    vuelta_rapida
FROM
    tb_circuitos
WHERE
    km_circuitos > 6.0
ORDER BY
    km_circuitos DESC;

CREATE INDEX idx_km_circuitos ON tb_circuitos (km_circuitos);
exec sp_helpindex tb_circuitos
--DROP INDEX idx_km_circuitos ON tb_circuitos
```

100 %

Results Messages

Tiempo de análisis y compilación de SQL Server:
Tiempo de CPU = 1 ms, tiempo transcurrido = 1 ms.

Tiempo de análisis y compilación de SQL Server:
Tiempo de CPU = 0 ms, tiempo transcurrido = 0 ms.

(2 rows affected)

Tiempos de ejecución de SQL Server:
Tiempo de CPU = 0 ms, tiempo transcurrido = 0 ms.

Completion time: 2024-05-05T11:31:59.1755605+02:00

El índice idx_km_circuitos se crea para mejorar el rendimiento de consultas que filtran circuitos por su longitud, permitiendo una búsqueda más eficiente de los circuitos cuya longitud sea superior a 6.0 kilómetros en la tabla