Cheatography

Buenas practicas SQL Cheat Sheet

by Oficina de Datos (Miguel Ortiz) via cheatography.com/148304/cs/32343/

Creación de tablas

¡Importante! Es obligatorio guardar las tablas como Parquet y no en formato Text File en el lago de datos debido a que estos últimos ocupan más espacio de almacenamiento.

Estructura para crear esto desde Hive

```
CREATE TABLE Nombre_Tabla(Schema)
ROW FORMAT DELIMITED FIELDS
TERMINATED BY ";" STORED AS
PARQUET;
```

Sentencia SELECT

Evitar usar **SELECT** *. Es preferible listar las columnas exactas que se usarán en el análisis de forma que se optimice el almacenamiento y procesamiento de la consulta.

a. Código sin optimizar

```
-- Tabla t1 tiene 90 columnas
select *
from tabla1 t1
```

b. Código optimizado

```
select
t1.var1,
t1.var7
from tabla1 t1
```

Por otro lado, use en lo posible, la función LIMIT para limitar el número de filas visualizadas.

```
Select*
from tabla1 t1
Limit 10
```

Delimitación de periodos

Evitar realizar consultas que no contengan un **periodo delimitado**, esto permite que sea mas eficiente, optimizando el almacenamiento y procesamineto.

a. Código sin optimizar.

Delimitación de periodos (cont)

```
Select
t1.var1
t2.var2
from tabla1 t1
```

b. Código optimizado.

```
Select
t1.var1
t2.var2
from tabla1 t1
where periodo = AAAAMMDD
```

c. Código optimizado 2.

```
Select
t1.var1
t2.var2
from tabla1 t1
where periodo Between AAAAMMDD and
AAAAMMDD
```

Por otro lado, si no utiliza un delimitador de periodo debe usar la función LIMIT para limitar el número de filas visualizadas.

```
Select
from tabla1 t1
Limit 10
```

Sentencia WHERE vs JOIN

Evitar cruces entre tablas usando WHERE, preferir la definición de cruces con JOIN y todas sus variaciones (left, right, inner, etc)

a. Código sin optimizar

```
select
t1.var1,
t1.var7,
t2.var2
from tabla1 t1, tabla2 t2
where t1.var1 = t2.var5
```

b. Código optimizado

```
select
t1.var1,
t1.var7,
t2.var2 from tabla1 t1 inner join tabla2 t2
 on t1.var1 = t2.var5
```

Sentencia DISTINCT vs GROUP BY

Para eliminar duplicados preferir **DISTINCT** en vez de **GROUP BY.**

a. Código sin optimizar

```
select
t1.var1
t1.var7
from tabla1 t1
group by t1.var1, t1.var7
```

b. Código optimizado

```
select distinct
t1.var1
t1.var7
from tabla1 t1
```

Sentencias UPDATE vs CASE WHEN

Evitar sentencias de **UPDATE** posteriores a consultas y reemplazarlas en lo posible por sentencias de **CASE WHEN** para rehusar las sentencias de códigos previas.

a. Código sin optimizar

```
--- Consulta genera tabla: TemporalTabla
select
t1.var1,
t1.var7,
t2.var2
from tabla1 t1 left join tabla2 t2
 on t1.var1 = t2.var5
--- Actualización de tabla: TemporalTabla
update TemporalTabla
set var2 = “No aplica”
where var7 is null
```

b. Código optimizado

```
select
t1.var1
t1.var7
case
when var7 is null then “no aplica”
else t2.var2
end as var2
from tabla1 t1 left join tabla2 t2
on t1.var1 = t2.var5
```

By **Oficina de Datos** (Miguel Ortiz)
cheatography.com/miguel-ortiz/

Published 6th June, 2022.
Last updated 23rd September, 2022.

Sentencia IN/NOT IN vs JOINS o WITH

Evitar al máximo consultas anidadas y el uso de **IN/NOT IN.** Reemplazar por **JOINS** o por la cláusula **WITH** para garantizar un código más fácil de leer y/o depurar.

a. Código sin optimizar 1

```
select
t1.var1,
t1.var7
from tabla1 t1
where t1.var1 in (select t2.columna from
tabla2 t2)
```

b. Código optimizado 1

```
select
t1.var1,
t1.var7
from tabla1 t1 inner join tabla2 t2
on t1.var1 = t2.columna
```

c. Código sin optimizar

```
select
t1.var1,
t1.var7
from tabla1 t1 inner join
(
select
t2.columna
from tabla2 t2 left join* tabla3 t3
on t2.var1 = t3.var1
) temporal
on t1.var1 = temporal.columna
```

d. Código optimizado 2

```
with temporal as (
select
t2.columna
from tabla2 t2 left join tabla3 t3
on t2.var1 = t3.var1
)
select
t1.var1,
t1.var7
from tabla1 t1 inner join temporal
on t1.var1 = temporal.columna
```

Unificación de procesos

Unificar procesos similares. No repetir procesos.

a. Código sin optimizar

```
update
temp_comercios_nuevos_cr
set
nuevo_nombre = replace(nuevo_nombre,
'www.','')
where
nuevo_nombre like '%www.%'
update
temp_comercios_nuevos_cr
set
nuevo_nombre = replace(nuevo_nombre,
'/bill','') where
nuevo_nombre like '/bill' --etc
```

b. Código optimizado

```
select
case
when nuevo_nombre like '%/bill%' then
replace (nuevo_nombre, '/bill','')
when nuevo_nombre like '%www.%' then
replace(nuevo_nombre, 'www.','')
else nuevo_nombre
end as nuevo_nombre
from temp_comercios_nuevos_cr
```

Convenciones

Evitar el uso de alias o nombres temporales ambiguos. Se recomienda aclarar si la construcción es temporal y a que hace referencia, de forma que el proceso de negocio sea coherente y fácil de manipular.

a. Código sin optimizar

Convenciones (cont)

```
select
A.var1,
A.var7
from tabla1 A inner join
(
select
t2.columna
from tabla2 B left join tabla3 C
on B.var1 = C.var1
)D
on A.var1 = D.columna
```

b. Código optimizado

```
with estado_tx as (
select t2. columna
from tabla2 movto left join tabla3
novedades
on movto.var1 = novedades.var1 )
select
saldos.var1,
saldos.var7
from tabla1.saldos inner join estado_tx
on saldos.var1 = estado_tx.columna
```

Nombramiento tablas temporales

A manera de estándar se recomienda que las tablas temporales creadas dentro de un proceso de SQL sigan la siguiente estructura: **temp_AbreviaturaProyecto_nombreTabla.**

Nombres de variables y tablas

Como se mencionó en el punto anterior, se debe evitar la ambigüedad en las tablas o subconsultas. Sin embargo, esto aplica también en el nombramiento de las columnas en las que se recomienda usar el separador "_" para usar nombres largos. Usar abreviaciones cuando se requiera, sin que el uso de las mismas puedan confundir al lector del código.
(Ejemplo: Reemplazar NAprobaTjCredCl por nro_aprobacion_tdc_cliente)

By **Oficina de Datos** (Miguel Ortiz)
cheatography.com/miguel-ortiz/

Published 6th June, 2022.
Last updated 23rd September, 2022.