

Tips para mejorar las consultas ABAP

En ocasiones, cuando desarrollamos algún reporte en ABAP, nos encontramos con que el mismo presenta bastante lentitud en su ejecución y por ende, mucho consumo de recursos del servidor. Para mejorar esto, veamos algunos consejos sobre cómo optimizar las consultas SQL en nuestros desarrollos.

1)- Intervalo de los operadores en la cláusula *WHERE*

Vamos a seleccionar vuelos de la tabla SFLIGHT filtrando por la compañía aérea.

```
SELECT * FROM sflight INTO wa WHERE carrid = _____
```

Las siguientes opciones serían óptimas:

= 'AA'.

IN ('AA', 'LH', 'UA').

BETWEEN 'AA' AND 'LH'.

LIKE 'A%'.

En cambio, la siguiente no sería una alternativa recomendada:

NOT carrid IN ('AA', 'LH', 'UA').

2)- El uso de negaciones en la cláusula *WHERE*.

No recomendado: negación en el WHERE:

```
SELECT * FROM sflight INTO wa  
WHERE carrid = 'AA' AND NOT connid IN ( '0017', '0019' ).
```

Opción recomendada: Usar formulaciones positivas en la medida de lo posible:

```
SELECT * FROM sflight INTO wa  
WHERE carrid = 'AA' AND connid IN ( '0020', '0021' ).
```

3)- El uso de la condición *OR* en la cláusula *WHERE*.

No recomendado: uso de la condición OR en el WHERE:

```
SELECT * FROM sflight INTO wa  
WHERE carrid = 'AA' AND ( connid = '0017' OR connid = '0019' ).
```

Opción recomendada: utilizar la cláusula IN en el WHERE:

```
SELECT * FROM sflight INTO wa  
WHERE carrid = 'AA' AND connid IN ( '0017', '0019' ).
```

4)- El uso de la condición ORDER BY en la cláusula WHERE

No recomendado: pretender ordenar por un campo en particular (no clave primaria).

```
SELECT * FROM sflight INTO wa  
WHERE carrid = 'AA' AND connid = '0017'  
ORDER BY seatsocc.  
ENDSELECT.
```

Opción recomendada: ordenar los resultados por su clave primaria.

```
SELECT * FROM sflight INTO wa  
WHERE carrid = 'AA' AND connid = '0017'  
ORDER BY PRIMARY KEY.  
ENDSELECT.
```

5)- No usar la forma SELECT-ENDSELECT.

Mi mayor recomendación, es NO USAR bajo ningún concepto la cláusula SELECT-ENDSELECT; si bien usar esta forma de consulta, permite procesar los resultados al momento de su obtención, como por ejemplo, ordenar los resultados al instante como en el ejemplo anterior, esto consume muchos recursos en memoria, ya que se realizan tantos accesos a la base de datos como resultados haya.

Recomiendo ampliamente guardar los resultados de la consulta en una tabla interna, hacer un ÚNICO acceso a la base de datos, y luego procesar esos resultados tratando directamente con la tabla interna, o lo que es lo mismo, con los resultados en memoria interna del programa.

En lugar de utilizar la consulta que se muestra en el punto 4, se recomienda plantearla de la siguiente forma:

```
SELECT * INTO TABLE itab FROM sflight  
WHERE carrid = 'AA' AND connid = '0017'.  
SORT itab BY carrid seatsocc.
```

En resumen, utiliza estos pequeños tips a la hora de desarrollar tus reportes y programas ABAP para mejorar el rendimiento de tus consultas al Diccionario de Datos, y verás que tus trabajos quedarán optimizados.