

<b>SELECT COUNT(*) FROM PRUEBA WHERE</b>	<b>CLAVE = 50000</b>	<b>DISPERS O = 50000;</b>	<b>CONCEN TRADO = 5;</b>	<b>IDISPERS O = 50000;</b>	<b>ICONCEN TRADO = 5;</b>	<b>BCONCE NTRADO = 5;</b>
<b>PLAN</b>	INDEX UNIQUE SCAN  PRUEBA_ PK	TABLE ACCESS FULL NO MBRE	TABLE ACCESS FULL NO MBRE	INDEX RANGE SCAN	INDEX RANGE SCAN	BITMAP INDEX SINGLE VALUE  PBC
<b>A-Time</b>	00:00:00.0 1	00:00:00.0 7	00:00:00.0 1	00:00:00.0 1	00:00:00.0 5	00:00:00.0 1
<b>Reads</b>	3	1897	2260	3	84	8
<b>“consiste n t gets”</b>	10	2429	2452	116	287	201
<b>“physical read total bytes</b>	40960	18620416	18808832	393216	3891200	3244032

## SYSTEM

--1

```
create user usuario3 identified by usuario;
alter user usuario3 quota 1M on ts_lifefit default tablespace ts_lifefit;
grant connect to usuario3;
grant create table to usuario3;
```

grant DBA to usuario3;

## USUARIO3

--2

```
create table prueba (clave NUMBER(16, 0) primary key, disperso NUMBER(16, 0),
concentrado NUMBER(16, 0), idisperso NUMBER(16, 0),
iconcentrado NUMBER(16, 0), bconcentrado NUMBER(16, 0));
```

DECLARE

I NUMBER(16,0);

R NUMBER(16,0);

BEGIN

FOR I IN 1..500000 LOOP

R := DBMS\_RANDOM.VALUE(1,1000000000);

INSERT INTO PRUEBA VALUES(I, R, MOD(R,11), 1000000000-R, MOD(1000000000-R, 11),

```
MOD(2000000000-R, 11));  
END LOOP;  
END;  
/  
COMMIT;
```

--3

```
create index PID on PRUEBA(IDISPERSO);  
create index PIC on PRUEBA(ICONCENTRADO);  
create bitmap index PBC on PRUEBA(BCONCENTRADO);
```

```
SET AUTOTRACE ON;  
ALTER SESSION SET STATISTICS_LEVEL='ALL';
```

--OJO: antes de cada consulta estas sentencias

```
ALTER SYSTEM FLUSH SHARED_POOL;  
ALTER SYSTEM FLUSH BUFFER_CACHE;
```

--4

```
SELECT COUNT(*) FROM PRUEBA WHERE CLAVE = 50000;  
SELECT COUNT(*) FROM PRUEBA WHERE DISPERSO = 50000;  
SELECT COUNT(*) FROM PRUEBA WHERE CONCENTRADO = 5;  
SELECT COUNT(*) FROM PRUEBA WHERE IDISPERSO = 50000;  
SELECT COUNT(*) FROM PRUEBA WHERE ICONCENTRADO = 5;  
SELECT COUNT(*) FROM PRUEBA WHERE BCONCENTRADO = 5;
```

--5

--Establece cual de las consultas efectuadas es más rápida, cuál más lenta y porqué.

--La consulta más rápida es SELECT COUNT(\*) FROM PRUEBA WHERE CLAVE = 50000;  
pues está indexada al ser un atributo primary key

--La consulta más lenta es SELECT COUNT(\*) FROM PRUEBA WHERE CONCENTRADO = 5, no se hace con índices. Por lo tanto es menos eficiente.

--6

```
UPDATE PRUEBA SET DISPERSO = DISPERSO + 7;
```

--65740800 cell physical IO interconnect bytes

```
UPDATE PRUEBA SET IDISPERSO = IDISPERSO + 7;
```

--206880768 cell physical IO interconnect bytes

--La segunda operación es más costosa pues es el atributo indexado. Esto se debe a que al actualizar un atributo indexado, la base de datos necesita reorganizar el índice para reflejar los cambios, lo que implica más operaciones de escritura y un mayor consumo de recursos en comparación con la actualización de un atributo no indexado como DISPERSO.

--7

```
SELECT COUNT(*) FROM PRUEBA WHERE IDISPERSO BETWEEN 10000 AND 20000;
```

--3366912 cell physical IO interconnect bytes

--Es mucho menos costoso

--Se realiza una operación denominada INDEX RANGE SCAN. Dado que estamos realizando una búsqueda por un rango y el índice está

--ordenado se puede realizar la búsqueda del rango usando el propio índice. Esto significa que puede buscar eficientemente dentro del índice para encontrar las filas que cumplen con el rango especificado.

--8

```
SELECT COUNT(*) FROM PRUEBA WHERE IDISPERSO+ICONCENTRADO BETWEEN 10000 AND 20000;
```

--19054592 cell physical IO interconnect bytes

--Esta operación es mucho más lenta, suma y tiene que recorrer toda la tabla

--La operación que se efectúa es un TABLE ACCESS FULL. No utiliza el índice y realiza un escaneo completo de la tabla (TABLE ACCESS FULL). Esto se debe a que la expresión IDISPERSO+ICONCENTRADO no está indexada y la base de datos no puede utilizar un índice para optimizar la consulta.

--9

--Índice de una función, ahora sí se guarda la suma

```
create index FIX on PRUEBA (IDISPERSO+ICONCENTRADO);
```

--10

```
SELECT COUNT(*) FROM PRUEBA WHERE IDISPERSO+ICONCENTRADO BETWEEN 10000 AND 20000;
```

--Tarda mucho menos ahora

--458752 cell physical IO interconnect bytes

--11

--Tarda mucho menos ahora. El tiempo de ejecución disminuye y que el uso

--de recursos sea más eficiente debido al acceso más rápido a los datos mediante el índice.

--458752 cell physical IO interconnect bytes

--12

```
UPDATE PRUEBA SET IDISPERSO = IDISPERSO + 7;
```

--331268096 cell physical IO interconnect bytes

--Se produce un aumento en las estadísticas relacionadas con la escritura y la reorganización del índice

- asociado a la columna IDISPERSO. Esto se debe a que la actualización de un atributo indexado implica
- la modificación de los índices correspondientes, lo que puede resultar en un aumento del uso de recursos
- en comparación con las actualizaciones de atributos no indexados.