



BASES DE DATOS AVANZADAS

DOMINGO 24 DE ABRIL, 2022

Actividad N°2.

*Iván Cáceres,
Sebastián Cornejo,
Tobías Guerrero,
Marcelo Muñoz*

Profesor
Juan Ricardo Giadach

Índice

Introducción.	2
Características del entorno de trabajo	2
Eliminación de Índice	2
Recuperación de datos	3
Creación tabla veintemil.	3
Consulta para personas1.	3
Consulta para personas2.	4
Análisis de tiempos obtenidos.	5
Cálculos de parámetros de acceso	5
Cálculo tiempo de respuesta	6
Conclusión	6

Introducción

Una vez finalizada la actividad 1, la cual está relacionada a la creación de tablas en una base de datos, se procede con la siguiente; esta se centra en otra de las operaciones fundamentales, el recuperar los datos solicitados.

■ Características del entorno de trabajo

Este trabajo se realiza en un computador con las siguientes características (Software/Hardware):

Elemento	Valor
Nombre del SO	Microsoft Windows 10 Pro
Versión	10.0.19044 Compilación 19044
Descripción adicional del SO	No disponible
Fabricante del SO	Microsoft Corporation
Nombre del sistema	DESKTOP-BTF3JK5
Fabricante del sistema	HP
Modelo del sistema	HP EliteBook 830 G5
Tipo de sistema	x64-based PC
SKU del sistema	5FV92EC#ABM
Procesador	Intel(R) Core(TM) i5-8350U CPU @ 1.70GHz, 1896 Mhz, 4 procesadores princi...
Versión y fecha de BIOS	HP Q78 Ver. 01.19.00, 13-01-2022
Versión de SMBIOS	3.1
Versión de controladora integr...	4.109
Modo de BIOS	UEFI
Fabricante de la placa base	HP
Producto placa base	8383
Versión de la placa base	KBC Version 04.6D.00
Rol de plataforma	Móvil
Estado de arranque seguro	Activado
Configuración de PCR7	Se necesita elevación de privilegios para ver
Directorio de Windows	C:\WINDOWS
Directorio del sistema	C:\WINDOWS\system32
Dispositivo de arranque	\Device\HarddiskVolume1
Configuración regional	México
Capa de abstracción de hardw...	Versión = "10.0.19041.1566"
Nombre de usuario	DESKTOP-BTF3JK5\vaaaaaaaan
Zona horaria	Hora est. Sudamérica Pacifico
Memoria física instalada (RAM)	16,0 GB

Figura 1: Datos del sistema.

Elemento	Valor
Descripción	Unidad de disco
Fabricante	(Unidades de disco estándar)
Modelo	SAMSUNG MZVLB512HAJQ-000H1
Bytes/sector	512
Medio cargado	Sí
Tipo de medio	Fixed hard disk
Particiones	3
Bus SCSI	0
Unidades lógicas SCSI	0
Puerto SCSI	0
Id. de destino SCSI	0
Sectores/pista	63
Tamaño	476,94 GB (512.105.932.800 bytes)
Nº total de cilindros	62.260
Nº total de sectores	1.000.206.900
Nº total de pistas	15.876.300
Pistas/cilindro	255
Partición	Disco #0, partición #0
Tamaño de partición	260,00 MB (272.629.760 bytes)
Desplazamiento inicial de parti...	1.048.576 bytes
Partición	Disco #0, partición #1
Tamaño de partición	475,76 GB (510.847.352.832 bytes)
Desplazamiento inicial de parti...	290.455.552 bytes
Partición	Disco #0, partición #2
Tamaño de partición	922,00 MB (966.787.072 bytes)
Desplazamiento inicial de parti...	511.137.808.384 bytes

Figura 2: Especificaciones del disco.

Cabe destacar que la unidad de almacenamiento que se utiliza es del tipo ssd M.2, por lo que los resultados esperados son de un tiempo reducido en comparación a una unidad hdd.

■ Eliminación de Índice

En este punto se pide deshacer el índice creado para la tabla personas1 en la actividad anterior, para esto se realiza la instrucción “*DROP INDEX indice;*” donde ‘indice’ es el nombre otorgado a este en la primera actividad.

```
bddavz=# drop index indice;
DROP INDEX
```

Figura 3: Eliminación Índice

■ Recuperación de datos

→ Creación tabla veintemil

Primero para crear la tabla se usa “*CREATE TABLE veintemil(rut numeric(10));*”

```
bddavz=# CREATE TABLE veintemil(rut numeric(10));
CREATE TABLE
```

Figura 4: Creación tabla veintemil

A continuación para realizar la inserción se ocupa el siguiente comando: “*COPY veintemil(rut) from 'C:/Varios/U/5TO SEMESTRE/BDDAVZ/veintemil';*”

```
bddavz=# COPY veintemil(rut) FROM 'C:/Varios/U/5TO SEMESTRE/BDDAVZ/veintemil';
COPY 20000
Duración: 84,536 ms
```

Figura 5: Inserción de datos a tabla veintemil

→ Consulta para personas1

Para obtener el nombre y dirección de las 20.000 personas de la tabla veintemil que se buscan en personas1, se realiza la siguiente instrucción, la cual no entrega los datos en la consola, sino que los copia directamente en un archivo.csv.

Comando: “*COPY (select p.nombre, p.direccion from personas1 as p, veintemil as v where p.rut=v.rut) to 'C:/Varios/U/5TO SEMESTRE/BDDAVZ/resultado.csv' (format CSV);*”

Este proceso se repite 2 veces más para tener mayor consistencia en los datos obtenidos, cabe destacar que entre cada intento se reinicia el dispositivo para liberar la memoria de manera que esta no influya en los resultados.

Prueba 1

```
bddavz=# COPY (select p.nombre, p.direccion from personas1 as p, veintemil as v where p.rut=v.rut) to 'C:/Varios/U/5TO SEMESTRE/BDDAVZ/resultado.csv' (format CSV);
COPY 20000
Duración: 9810,274 ms (00:09,810)
```

Figura 6: Comando primera prueba

Prueba 2

```
bddavz=# COPY (select p.nombre, p.direccion from personas1 as p, veintemil as v where p.rut=v.rut) to 'C:/Varios/U/5TO SEMESTRE/BDDAVZ/resultado.csv' (format CSV);
COPY 20000
Duración: 12392,611 ms (00:12,393)
```

Figura 7: Comando segunda prueba

Prueba 3

```
bddavz=# COPY (select p.nombre, p.direccion from personas1 as p, veintemil as v where p.rut=v.rut) to 'C:/Varios/U/5TO SEMESTRE/BDDAVZ/resultado.csv' (format CSV);
COPY 20000
Duración: 8756,715 ms (00:08,757)
```

Figura 8: Comando tercera prueba

Archivo .csv con las 20.000 filas entregadas por la consulta anterior.



Figura 9: Filas archivo .csv en un txt

→ Consulta para personas2

Se realiza casi la misma consulta cambiando la tabla utilizada por personas2: ***COPY (select p.nombre, p.direccion from personas2 as p, veintemil as v where p.rut=v.rut) to 'C:/Varios/U/5TO SEMESTRE/BDDAVZ/resultado2.csv' (format CSV);***.

Este proceso se repite 2 veces más, para tener mayor consistencia en los datos obtenidos, cabe destacar que entre cada intento se reinicia el dispositivo para liberar la memoria, de manera que esta no influya en los resultados.

Obteniendo:

Prueba 1

```
bddavz=# COPY (select p.nombre, p.direccion from personas2 as p, veintemil as v where p.rut=v.rut) to 'C:/Varios/U/5TO SEMESTRE/BDDAVZ/resultado2.csv' (format CSV);
COPY 20000
Duración: 2980,684 ms (00:02,981)
```

Figura 10: Comando primera prueba

Prueba 2

```
bddavz=# COPY (select p.nombre, p.direccion from personas2 as p, veintemil as v where p.rut=v.rut) to 'C:/Varios/U/5TO SEMESTRE/BDDAVZ/resultado2.csv' (format CSV);
COPY 20000
Duración: 5126,833 ms (00:05,127)
```

Figura 11: Comando segunda prueba

Prueba 3

```
bddavz=# COPY (select p.nombre, p.direccion from personas2 as p, veintemil as v where p.rut=v.rut) to 'C:/Varios/U/5TO SEMESTRE/BDDAVZ/resultado2.csv' (format CSV);
COPY 20000
Duración: 4362,492 ms (00:04,362)
```

Figura 12: Comando tercera prueba

Archivo .csv con las 20.000 filas entregadas por la consulta anterior.



Figura 13: Filas archivo .csv en un txt

■ **Análisis de tiempos obtenidos**

	personas1	personas2
Prueba 1	9810,274 ms	2980,684 ms
Prueba 2	12392,611 ms	5126,833 ms
Prueba 3	8756,715 ms	4362,492 ms

Cuadro 1: Tiempos de ambas consultas.

Consulta	personas1	personas2
Tiempo promedio (ms)	10319,867	4156,669

Cuadro 2: Tiempo promedio de ambas consultas.

El tiempo de consulta para la tabla personas1 aproximadamente duplica el tiempo de consulta para la tabla personas2, esto es debido a que la tabla personas2 tiene la llave primaria en rut, lo cual significa que se crea un B-tree que permite una búsqueda por índice, esto es significativamente más rápido que una búsqueda secuencial. Esta ultima es la que se utiliza en la consulta para la tabla personas1.

■ **Cálculos de parámetros de acceso**

Parámetros a calcular: **filas por bloque (fb)** y **orden del árbol (m)**, siendo T_1 el tiempo promedio de la consulta para la tabla personas1 y T_2 el promedio para la tabla personas2.

$$T_1 = \left(\frac{n}{fb} \right) * T_a$$
$$T_1 = \left(\frac{49875490}{fb} \right) * 1ms = 10319,867ms$$
$$\Rightarrow 49875490 = 10319,867 * fb$$
$$\Rightarrow fb = 4833$$
$$T_2 = (\log_m N + X) * VecesQueSeRealiza * T_a$$

X tiene el valor de 1 porque rut es llave primaria por lo que no hay valores repetidos y se accede directamente al dato y se realiza 20000 veces porque busca cada rut de la tabla veintemil.

$$T_2 = (\log_m 49875490) * 20000 * 1ms = 4156,669ms$$

$$\Rightarrow m = 1,09328 \rightarrow 2$$

▪ Cálculo tiempo de respuesta

A continuación se realiza el cálculo de los tiempos de respuesta teóricos para ambas consultas considerando que los parámetros filas por bloque y orden de árbol son 50 y 300 respectivamente.

$$T_1 = \left(\frac{n}{fb} \right) * T_a$$

$$T_1 = \left(\frac{49875490}{50} \right) * 1ms = 997509,8ms = 997,5098seg$$

$$T_2 = (\log_m N + X) * VecesQueSeRealiza * T_a$$

X tiene el valor de 1 porque rut es llave primaria por lo que no hay valores repetidos y se accede directamente al dato y se realiza 20000 veces porque busca cada rut de la tabla veintemil.

$$T_2 = (\log_{300} 49875490 + 1) * 20000 * 1ms = 82151,88022ms = 82,151seg$$

	personas1	personas2
Teórico	997509,8 ms	82151,88 ms
Empírico	10319,867 ms	4156,669 ms

Cuadro 3: Comparación tiempos teóricos v/s experimentales.

Es posible observar que el tiempo de respuesta teórico y el empírico se diferencian bastante, esto probablemente ocurre por que el sistema de bases de datos optimiza automáticamente como hace las consultas y esto se refleja en un mejor tiempo empírico.

Conclusión

El presente trabajo se centra en una de las operaciones fundamentales de una base de datos, la cual es la recuperación o lectura. Esto se realiza mediante los ítem pedidos en esta tarea, la cual corresponde a la número 2 de la asignatura. Al realizarse se encuentran valores que no coinciden con los resultados esperados teóricamente, el orden del árbol esperado es de 300 y las filas por bloque de 50, no obstante, los resultados arrojan 2 y 4833 respectivamente, esto probablemente debido a que como se dijo antes, el sistema de bases de datos optimiza las consultas y como el cálculo teórico no contempla estas optimizaciones los resultados son distintos. Por otra parte se observa que el tiempo de la consulta 1 es algo más que el doble que el tiempo de la consulta 2, esto se debe a que el algoritmo de resolución de la consulta 1 es una búsqueda secuencial mientras que el de la consulta 2 es una búsqueda por índices, que para esta cantidad de datos, resulta mas eficiente la segunda. Comparando estos resultados con los tiempos de inserción en cada tabla para la tarea número 2 se puede notar que la inserción de datos relativamente masivos es mucho mas rápida en una tabla sin índice y sin llave primaria, en comparación a una que si tenga esto, mientras que en la búsqueda o recuperación de datos es lo contrario.