

Actividad 01 - Índices

Para esta actividad vamos a considerar el siguiente esquema:

- `Personas(pid INT PRIMARY KEY, nombre VARCHAR(200), edad INT)`
- `Mascotas(mid INT PRIMARY KEY, pid INT, nombre VARCHAR(200), tipo VARCHAR(200))`

Donde estamos representando **Personas** y **Mascotas**. En la tabla mascota, el atributo `pid` indica el identificador del dueño de la mascota. Así, una persona puede tener muchas mascotas y una mascota puede tener solamente un dueño. Asuma que la tabla de personas tiene 100000 registros y la tabla de mascotas tiene 500000; además, considere que una página de disco duro puede almacenar 10 tuplas y que ambas **PRIMARY KEY** aumentan de 1 en 1. Finalmente, asuma que ambas llaves primarias están indexadas con un B+Tree Clustered.

Pregunta 01 [2 pts]

Ahora, considera las siguientes consultas en SQL:

- `SELECT * FROM Personas WHERE pid = i`
- `SELECT * FROM Mascotas WHERE mid >= 0 AND mid <200`

En donde `i` y `j` son números arbitrarios que existen en la base de datos.

Indique el costo de I/O para cada una de las consultas considerando los índices existentes.

Pregunta 02 [2 pts]

Considere las siguientes consultas de *join* en SQL:

- `SELECT * FROM Personas P, Mascotas M WHERE P.pid = M.pid`
- `SELECT * FROM Personas P, Mascotas M WHERE P.pid = M.pid AND M.mid = i`

Indique que nuevos índices usaría (si es que es necesario), y sobre que atributos, para que cada una de estas consultas se computen de forma eficiente. Piense las dos consultas por separado.

Pregunta 03 [2 pts]

Ahora considera las siguientes consultas de agregación y de selección para la base de datos:

- `SELECT edad, COUNT(*) FROM Personas GROUP BY edad`
- `SELECT * FROM PERSONAS WHERE edad >= 18 and edad <30`
- `SELECT pid, COUNT(*) FROM Personas P, Mascotas M WHERE P.pid = M.pid GROUP BY pid`

Indique que nuevos índices usaría (si es que es necesario), y sobre que atributos, para que cada una de estas consultas se computen de forma eficiente. Piense las tres consultas por separado.