

Actividad 01 - Índices

Problema 1. Consideremos la siguiente tabla que guarda información de contacto de personas.

```
Personas(
  id INT,
  nombre VARCHAR(200),
  email VARCHAR(200),
  teléfono VARCHAR(200),
  edad INT
)
```

La tabla tiene un millón de tuplas pero actualmente no está indexada, por lo que las tuplas no están ordenadas en el disco duro. Además, consideremos que el atributo *id* es candidato a llave (es decir, son valores únicos) y los valores van del 1 al 1000000. Consideremos además que cada página puede contener T tuplas.

Queremos que analices el costo I/O para un conjunto de consultas en cada uno de los siguientes casos:

- Analizar la tabla sin ningún índice.
- Usar un *B+Tree clustered* sobre el atributo *id*. El árbol es de altura h_c y supongamos que cada hoja con datos está llena (es decir, cada hoja tiene T tuplas).
- Usar un *B+Tree unclustered* sobre el atributo *id*. El árbol es de altura h_u y cada página con datos contiene P punteros a las páginas con los datos.
- Usar un *Hash Index clustered* que es dinámico y evita la creación de *overflow pages*.
- Usar un *Hash Index unclustered* que es dinámico y evita la creación de *overflow pages*.

Las consultas son:

1. Encontrar todas las tuplas de **Personas**.
2. Encontrar todas las tuplas de **Personas** donde *id* = 50.
3. Encontrar todas las tuplas de **Personas** donde *id* < 51.
4. Encontrar todas las tuplas de **Personas** donde *id* > 100 AND *id* <= 200.

Problema 2. Consideremos la misma tabla del punto anterior. Supongamos que queremos hacer consultas de igualdad y de rango para los atributos *id* y *edad*. Recordando que el atributo *id* es candidato a llave, responda las siguientes preguntas.

1. ¿Cómo esperas que sea la indexación de la tabla?
2. Considerando la indexación propuesta en el punto anterior, calcula el costo I/O de la consulta `SELECT * FROM Personas WHERE edad >= 25`. Asume que las personas con edad igual o mayor a 25 es el 70% de los registros de la tabla. Asuma los valores de altura, T y P del punto anterior.

Problema 3. Ahora considera que sumamos esta tabla al esquema.

```
Mascotas(  
  mid INT PRIMARY KEY,  
  nombre VARCHAR(200),  
  pid INT,  
  FOREIGN KEY (pid) REFERENCES Personas(id)  
)
```

Asume que las consultas de interés son:

1. Obtener los datos de una mascota en particular por su **id**.
2. Dada una persona, entrega todos los atributos de la persona junto al de sus mascotas.
3. Dada una mascota, entrega sus atributos junto a los de su persona correspondiente.

¿Cómo esperas que sea la indexación de la tabla **Mascotas**? Además, explica el plan de consulta en cada caso.