

Ejercicios Sesión 02 Fundamentos de bases de datos

Big Data Management

16 de abril de 2020

1. Índices

Dada la tabla Tienda:

producto	fecha_compra	cliente_id	precio	unidades	categoria_producto
Tablet Gstar	24/02/2020 23:14:58	00012340	45	1	Tecnología
Microondas	18/01/2020 16:01:24	00585141	100	1	Electrodomésticos
Smart TV	22/02/2020 17:12:14	00621458	560	1	Tecnología
mandoPlay X	22/02/2020 18:00:12	00621458	94	4	Gamer
cascos X	11/12/2020 15:54:52	00147852	67	2	Tecnologías
Smart Watch	21/04/2020 12:00:00	00045872	24	1	Tecnología
Smartphone	11/11/2020 23:04:45	00048752	240	1	Tecnología
Licuadaora X	12/08/2020 15:24:23	00057056	10	1	Electrodomésticos

- 1 Crear índice B-tree para el atributo precio, de orden 2 y carga 75 %
- 2 Crear un índice B+ Cluster para el atributo categoría_producto, de orden 2 y carga 75 %
- 3 Crear un índice Hash sobre el atributo categoría_producto
- 4 Considerando los índices anteriormente creados y la estructura de la tabla Tienda, justifica qué plan de acceso minimizaría el número de I/O para las siguientes consultas:

```
SELECT * FROM Tienda WHERE fecha_compra > 30/12/2020 23:59:59 AND  
fecha_compra > 31/12/2020 23:59:59
```

```
SELECT cliente_id FROM Tienda WHERE categoria_producto = 'Tecnologia'
```

```
SELECT AVG(precio) FROM Tienda WHERE categoria_producto = 'Tecnologia'
```

2. Query Optimiser

Considerando la tabla Tienda y la tabla Cliente:

cliente_id	creacion_cuenta	edad
00012340	01/02/2017 12:00:52	32
00585141	01/05/2017 12:00:58	27
00621458	24/02/2017 18:50:57	22
00147852	11/02/2017 12:20:52	35
00045872	10/08/2016 12:00:52	38
00048752	01/02/2018 12:00:52	32
00057056	01/03/2017 11:00:52	42

1 Crear un árbol sintáctico óptimo para la siguiente consulta:

```
SELECT DISTINCT(C.cliente_id),C.edad FROM Clientes C, Tienda T
WHERE T.cliente_id = C.cliente_id
AND T.categoria_producto = 'Tecnologia'
ORDER BY C.edad
```