

EXERCICI 1

Donada la següent taula denominada “MOVIMENTS” amb les dades que es mostren, resoldre les consultes SQL mitjançant l'ús d'índexs de mapa de bits segons els supòsits proporcionats en els diferents apartats.

- a. La taula “MOVIMENTS” conté índexs de mapa de bits en les columnes “CONFIRMAT” i “MONEDA”. Assumiu que les columnes “CONFIRMAT” i “MONEDA” sí que accepten nuls (NULL). La consulta a resoldre és:

```
SELECT *  
FROM MOVIMIENTOS  
WHERE FECHA >= '2014-01-06'  
AND MONTANTE < 100;
```

La query no farà servir cap bitmap index. Les columnes per les quals es cerca (FECHA i MONTANTE) no es corresponen amb les columnes definides als índexos (CONFIRMAT i MONEDA).

- b. La taula “MOVIMENTS” conté índexs de mapa de bits en les columnes “MONEDA” i “ORIGEN”. Assumiu que les columnes “MONEDA” i “ORIGEN” no permeten valors nuls (NOT NULL). La consulta a resoldre és:

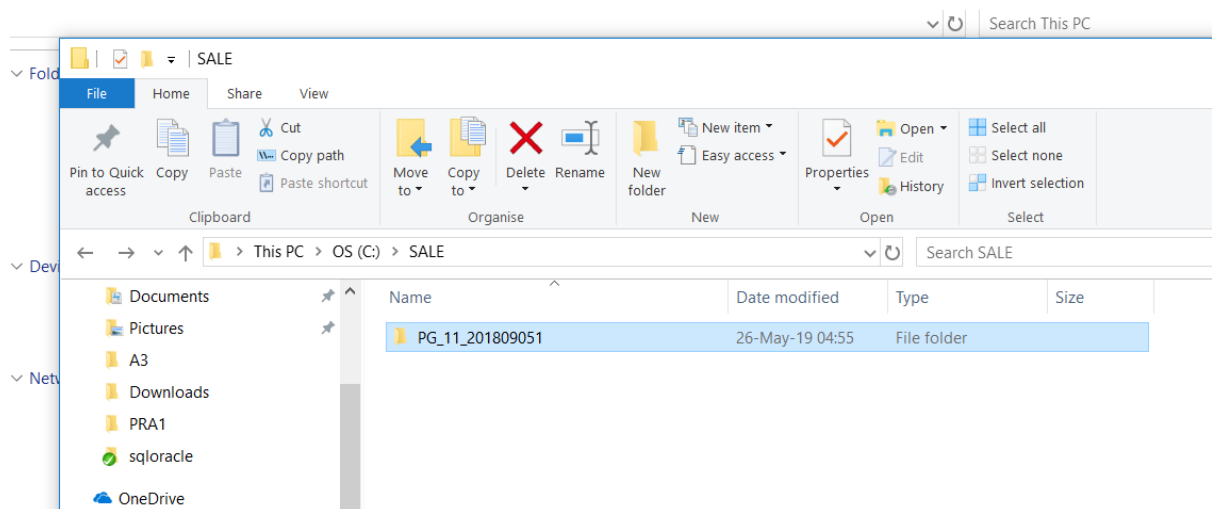
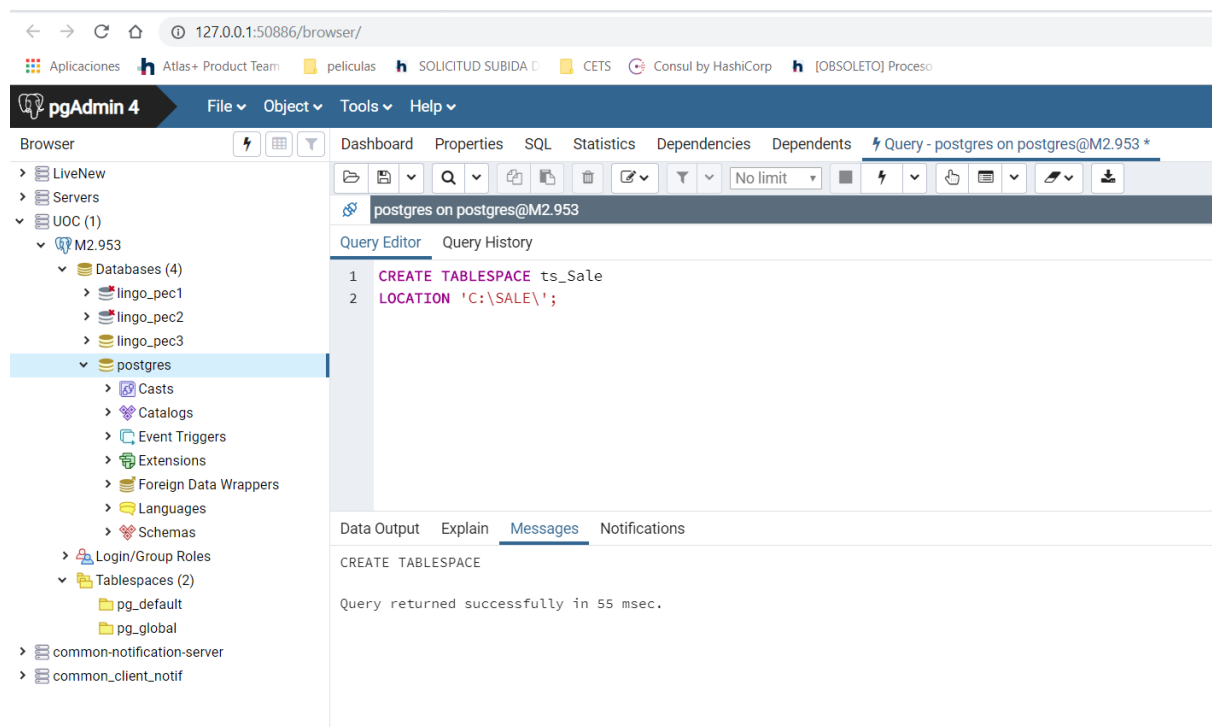
```
SELECT *  
FROM MOVIMIENTOS  
WHERE MONEDA = 'USD'  
AND ORIGEN IN ('CAJERO', 'TRN-WEB', 'CORREO')
```

En aquest cas, la query sí farà servir els dos índexos. Les columnes indexades coincideixen amb les columnes de cerca. Adjunt la resolució de l'índex al fitxer Resolucio_1_b.xlsx. Les files que compleixen les condicions estan marcades en blau.

Exercici 2

Crear un espai de taules mitjançant una sentència SQL. Aquest *tablespace* ha de dir-se ts_Sale i s'ha d'emmagatzemar en la ruta "C:\SALE\"

```
CREATE TABLESPACE ts_Sale LOCATION 'C:\SALE\';
```



Expliqueu si heu hagut de realitzar alguna operació extra en el sistema operatiu

Sí, s'ha hagut de crear el directori SALE

Calcular les estadístiques a les taules de l'esquema sale per a proporcionar un accés eficient a aquestes

Faig també vacuum per tal d'eliminar físicament les files que hagin pogut estat eliminades

```
VACUUM sale.tb_category;
```

```
ANALYZE sale.tb_category;
```

```
VACUUM sale.tb_city;
```

```
ANALYZE sale.tb_city;
```

```
VACUUM sale.tb_client;
```

```
ANALYZE sale.tb_client;
```

```
VACUUM sale.tb_order;
```

```
ANALYZE sale.tb_order;
```

```
VACUUM sale.tb_order_line;
```

```
ANALYZE sale.tb_order_line;
```

```
VACUUM sale.tb_product;
```

```
ANALYZE sale.tb_product;
```

```
VACUUM sale.tb_subcategory;
```

```
ANALYZE sale.tb_subcategory;
```

Proporcionar la sentència SQL per a crear els següents índexs. Aquests índexs s'han de generar en el *tablespace* ts_Sale.

Com a nota, dir que els índexos B+ són es que es creen per defecte (no s'ha d'especificar cap clàusula específica)

- Índex B+ idx_product_code amb el camp product_code de la taula tb_product.

He creat l'índex amb unique perquè el camp és clau primària. És més eficient que un índex no únic.

```
CREATE UNIQUE INDEX idx_product_code ON sale.tb_product (product_code) TABLESPACE ts_Sale;
```

- Índex B+ idx_orderline_prodcodeprc amb els camps product_code i unit_price de la taula tb_order_line.

```
CREATE INDEX idx_orderline_prodcodeprc ON sale.tb_order_line (product_code, unit_price) TABLESPACE ts_Sale;
```

- Índex B+ idx_product_price amb el camp price de la taula tb_product.

```
CREATE INDEX idx_product_price ON sale.tb_product (price) TABLESPACE ts_Sale;
```

Proporcionar la sentència SQL necessària per a obtenir el pla d'execució de la següent consulta, una captura de pantalla de dita i explicar breument per què creieu que el optimitzador genera aquest pla

EXPLAIN

```
SELECT
    tb_client.client_code,
    tb_client.client_name,
    tb_client.country
FROM
    Sale.tb_client
WHERE
    tb_client.phone IS NOT NULL
    AND tb_client.address IS NULL
ORDER BY
    tb_client.client_code ASC;
```

8
9
10
11
12
13
14
15
16
17
18
19
20
21

```
EXPLAIN
SELECT
    tb_client.client_code,
    tb_client.client_name,
    tb_client.country
FROM
    Sale.tb_client
WHERE
    tb_client.phone IS NOT NULL
    AND tb_client.address IS NULL
ORDER BY
    tb_client.client_code ASC;
```

Data Output Explain Messages Notifications

QUERY PLAN
text

1

Sort (cost=1.26..1.26 rows=1 width=260)

2

Sort Key: client_code

3

-> Seq Scan on tb_client (cost=0.00..1.25 rows=1 width=260)

4

Filter: ((phone IS NOT NULL) AND (address IS NULL))

L'optimitzador primer determina que s'han de filtrar les dades accedint directament a la taula (no hi ha cap índex) i després ordena els resultats. Filtra abans d'ordenar perquè és molt menys costós que ordenar tota la taula i després filtrar. No aplica cap índex perquè no hi ha cap que estigui disponible.

EXERCICI 3 (30%)

Indica, per a cadascuna de les següents afirmacions sobre bases de dades relacionals, si és **veritat o falsa**, i **explicar breument** per què.

- a. En termes de cost de manteniment, és sempre indicat l'ús d'índexs per a millorar el rendiment de les consultes.
No. Moltes vegades és més òptim fer un full de la taula. Per exemple, si la taula té molts pocs registres o si el percentatge de valors filtrats és molt alt.

- b. L'ordenació física de les files en una taula no té mai impacte en la millora del temps de resposta.
Sí pot tenir impacte; per exemple, a l'hora de recuperar registres per un rang pot evitar múltiples lectures de disc. Les lectures a disc penalitzen el temps de resposta.

- c. Per a la consulta *"SELECT * FROM all_ *users WHERE USER_ANEU > 100"* els índexs de tipus dispersió no són els més recomanats.
Correcte. Un índex de dispersió només és útil per filtres d'igualtat. En aquest cas, utilitzem <, així que és aconsellable fer servir un altre tipus d'índex.

- d. Les vistes materialitzades són vistes de la base de dades que contenen els resultats d'una consulta precalculada emmagatzemant el resultat en l'emmagatzematge volàtil.
No és correcte. El resultat és emmagatzemat en memòria no volàtil.