Laboratorio 8.2

Integrantes

- Pedro Domínguez
- Eduardo Arróspide

Crear la conexion a la BD

Se utilizaron los modulos: "sqlalchemy", "psycopg2" y "ipython-sql"

In [ ]:

```
import sqlalchemy as sa

# Create the connection
engine = sa.create_engine('postgresql://postgres:a@localhost:5432/dvdrentalcopy')

#Load the SQL extension
%load_ext sql
%config SqlMagic.displaycon = False

#Set up the connection:
%sql $engine.url
```

P1

In [ ]:

```
%%sql
DROP SCHEMA IF EXISTS lab82 CASCADE;

CREATE SCHEMA lab82;
SET search_path = lab82;

CREATE EXTENSION pg_trgm;

CREATE TABLE p1 (
    body text,
    body_indexed text
);

INSERT INTO p1
        SELECT
        md5(random()::text)
        FROM (
                SELECT * FROM
                        generate_series (1,100000) AS id
                ) AS x;

UPDATE p1 SET body_indexed = body;

CREATE INDEX p1_search_idx ON p1 USING gin (body_indexed gin_trgm_ops);
```

```
Done.
Done.
Done.
Done.
Done.
100000 rows affected.
100000 rows affected.
Done.
```

Out[ ]:

```
[]
```

In [ ]:

```
%%sql
EXPLAIN ANALYZE
SELECT COUNT(*) AS NO_INDEXED FROM p1 WHERE body LIKE '%abc%';
```

```
6 rows affected.
```

Out[ ]:

QUERY PLAN
Aggregate (cost=3328.00..3328.01 rows=1 width=8) (actual time=38.404..38.405 rows=1 loops=1)
-> Seq Scan on p1 (cost=0.00..3318.00 rows=4000 width=0) (actual time=8.890..38.282 rows=752 loops=1)
Filter: (body ~~ '%abc%'::text)
Rows Removed by Filter: 99248
Planning Time: 10.751 ms
Execution Time: 47.599 ms

In [ ]:

```
%%sql
EXPLAIN ANALYZE
SELECT COUNT(*) AS INDEXED FROM p1 WHERE body_indexed LIKE '%abc%';
```

```
8 rows affected.
```

Out[ ]:

QUERY PLAN
Aggregate (cost=2195.37..2195.38 rows=1 width=8) (actual time=10.780..10.781 rows=1 loops=1)
-> Bitmap Heap Scan on p1 (cost=51.00..2185.37 rows=4000 width=0) (actual time=10.213..10.749 rows=752 loops=1)
Recheck Cond: (body_indexed ~~ '%abc%'::text)
Heap Blocks: exact=580
-> Bitmap Index Scan on p1_search_idx (cost=0.00..50.00 rows=4000 width=0) (actual time=10.149..10.150 rows=752 loops=1)
Index Cond: (body_indexed ~~ '%abc%'::text)
Planning Time: 0.120 ms
Execution Time: 10.826 ms

P2

In [ ]:

```
%%sql

SET search_path = public;
DROP INDEX IF EXISTS idx_content_ts CASCADE;
```

```
ALTER TABLE film
DROP COLUMN IF EXISTS content_ts;

ALTER TABLE film ADD COLUMN content_ts tsvector;

UPDATE film
SET content_ts = x.content_ts FROM (
        SELECT film_id,
                        setweight(to_tsvector('english', title), 'A') ||
                        setweight(to_tsvector('english', description), 'B') AS content
        FROM film
        ) AS x
WHERE x.film_id = film.film_id;


CREATE INDEX idx_content_ts ON film USING gin(content_ts);
```

```
Done.
Done.
Done.
Done.
1000 rows affected.
Done.
```

Out[ ]:

```
[]
```

In [ ]:

```
%%sql
EXPLAIN ANALYZE
SELECT title, description, ts_rank_cd(content_ts, query_ts) AS score
FROM film, to_tsquery('english', 'man | woman') query_ts
WHERE query_ts @@ content_ts
ORDER BY score DESC
LIMIT 10;
```

```
11 rows affected.
```

Out[ ]:

QUERY PLAN
Limit (cost=46.14..46.16 rows=10 width=113) (actual time=1.133..1.135 rows=10 loops=1)
-> Sort (cost=46.14..46.16 rows=10 width=113) (actual time=1.131..1.132 rows=10 loops=1)
Sort Key: (ts_rank_cd(film.content_ts, '''man'' | ''woman'''::tsquery)) DESC
Sort Method: top-N heapsort Memory: 27kB
-> Bitmap Heap Scan on film (cost=12.08..45.97 rows=10 width=113) (actual time=0.070..1.069 rows=239 loops=1)
Recheck Cond: ('''man'' | ''woman'''::tsquery @@ content_ts)
Heap Blocks: exact=78
-> Bitmap Index Scan on idx_content_ts (cost=0.00..12.07 rows=10 width=0) (actual time=0.039..0.039 rows=239 loops=1)
Index Cond: (content_ts @@ '''man'' | ''woman'''::tsquery)
Planning Time: 18.904 ms
Execution Time: 1.202 ms

In [ ]:

```
%%sql
```

```
EXPLAIN ANALYZE
SELECT title, description, ts_rank_cd(fulltext, query_ts) AS score
FROM film, to_tsquery('english', 'man | woman') query_ts
WHERE query_ts @@ fulltext
ORDER BY score DESC
LIMIT 10;
```

```
9 rows affected.
```

Out[ ]:

QUERY PLAN
Limit (cost=248.24..248.26 rows=10 width=113) (actual time=1.417..1.419 rows=10 loops=1)
-> Sort (cost=248.24..248.83 rows=238 width=113) (actual time=1.416..1.417 rows=10 loops=1)
Sort Key: (ts_rank_cd(film.fulltext, '''man'' | ''woman'''::tsquery)) DESC
Sort Method: top-N heapsort Memory: 27kB
-> Seq Scan on film (cost=0.00..243.09 rows=238 width=113) (actual time=0.028..1.346 rows=239 loops=1)
Filter: ('''man'' | ''woman'''::tsquery @@ fulltext)
Rows Removed by Filter: 761
Planning Time: 0.302 ms
Execution Time: 1.444 ms

P3

In [ ]:

```
%%sql

SET search_path = public;
DROP TABLE IF EXISTS news CASCADE;
DROP INDEX IF EXISTS idx_content_ts CASCADE;

CREATE TABLE news(
    num integer,
    id integer,
    title text,
    publication text,
    author text,
    date text,
    year float,
    month float,
    url text,
    content text
);


COPY news FROM 'C:\Francisco\OneDrive - UNIVERSIDAD DE INGENIERIA Y TECNOLOGIA\UTEC\UT

ALTER TABLE news ADD COLUMN content_ts tsvector;

UPDATE news
SET content_ts = x.content_ts FROM (
        SELECT id,
                        setweight(to_tsvector('english', title), 'A') ||
                        setweight(to_tsvector('english', content), 'B') AS content_ts
        FROM news
        ) AS x
WHERE x.id = news.id;

ALTER TABLE news ADD COLUMN content_ts_no_index tsvector;

UPDATE news
SET content_ts_no_index = x.content_ts FROM (
        SELECT id,
                        setweight(to_tsvector('english', title), 'A') ||
                        setweight(to_tsvector('english', content), 'B') AS content_ts
        FROM news
        ) AS x
WHERE x.id = news.id;

CREATE INDEX idx_content_ts ON news USING gin(content_ts);
```

```
Done.
Done.
Done.
Done.
50000 rows affected.
Done.
50000 rows affected.
Done.
50000 rows affected.
Done.
```

Out[ ]:

```
[]
```

In [ ]:

```
%%sql
EXPLAIN ANALYZE
SELECT title, content, ts_rank_cd(content_ts, query_ts) AS score
FROM news, to_tsquery('english', 'trump | president') query_ts
WHERE query_ts @@ content_ts
ORDER BY score DESC;
```

```
13 rows affected.
```

Out[ ]:

```
QUERY PLAN
Gather Merge (cost=20912.85..24109.28 rows=27396 width=599) (actual time=595.522..701.099 rows=27596 loops=1)
Workers Planned: 2
Workers Launched: 2
-> Sort (cost=19912.83..19947.07 rows=13698 width=599) (actual time=522.907..526.314 rows=9199 loops=3)
Sort Key: (ts_rank_cd(news.content_ts, '''trump'' | ''presid'''::tsquery)) DESC
Sort Method: external merge Disk: 5720kB
Worker 0: Sort Method: external merge Disk: 5024kB
Worker 1: Sort Method: external merge Disk: 4600kB
-> Parallel Seq Scan on news (cost=0.00..15317.66 rows=13698 width=599) (actual time=1.827..497.390 rows=9199 loops=3)
Filter: ('''trump'' | ''presid'''::tsquery @@ content_ts)
Rows Removed by Filter: 7468
Planning Time: 0.209 ms
Execution Time: 793.653 ms
```

In [ ]:

```
%%sql
EXPLAIN ANALYZE
SELECT title, content, ts_rank_cd(content_ts_no_index, query_ts) AS score
FROM news, to_tsquery('english', 'trump | president') query_ts
WHERE query_ts @@ content_ts_no_index
ORDER BY score DESC;
```

```
13 rows affected.
```

Out[ ]:

```
QUERY PLAN
Gather Merge (cost=20912.85..24109.28 rows=27396 width=599) (actual time=490.696..613.661 rows=27596 loops=1)
Workers Planned: 2
Workers Launched: 2
-> Sort (cost=19912.83..19947.07 rows=13698 width=599) (actual time=422.897..425.947 rows=9199 loops=3)
Sort Key: (ts_rank_cd(news.content_ts_no_index, '''trump'' | ''presid'''::tsquery)) DESC
Sort Method: external merge Disk: 5648kB
Worker 0: Sort Method: external merge Disk: 4904kB
Worker 1: Sort Method: external merge Disk: 4792kB
-> Parallel Seq Scan on news (cost=0.00..15317.66 rows=13698 width=599) (actual time=1.457..400.010 rows=9199 loops=3)
Filter: ('''trump'' | ''presid'''::tsquery @@ content_ts_no_index)
Rows Removed by Filter: 7468
Planning Time: 0.261 ms
Execution Time: 934.272 ms
```