THỰC HÀNH LAB03

Nội dung thực hành:

Truy xuất dữ liệu với:

- **DATETIME**
- **Subquery**
- **Toán tử UNION, UNION ALL**
- **EXCEPT/INTERSECT**
- **RETURNING clause**
- **UPDATE multiple records**
- **CTE**

---

1. **DATETIME**

   Chuyển từ kiểu ký tự sang kiểu date: ::, to_date()

   a. select '12-31-2023'::date;

   b. select to_date('31/12/2023','dd/mm/yyyy') ;

   -Chuyển từ kiểu date sang kiểu ký tự: to_char()

   ```
   select to_char(NOW(),'dd-mm-yyyy');
   SELECT TO_CHAR(CURRENT_TIMESTAMP,
           '"Today is "FMDay", the "DDth" day of the month of "FMMonth" of
   "YYYY');
   ```

   Trích xuất tháng hiện tại

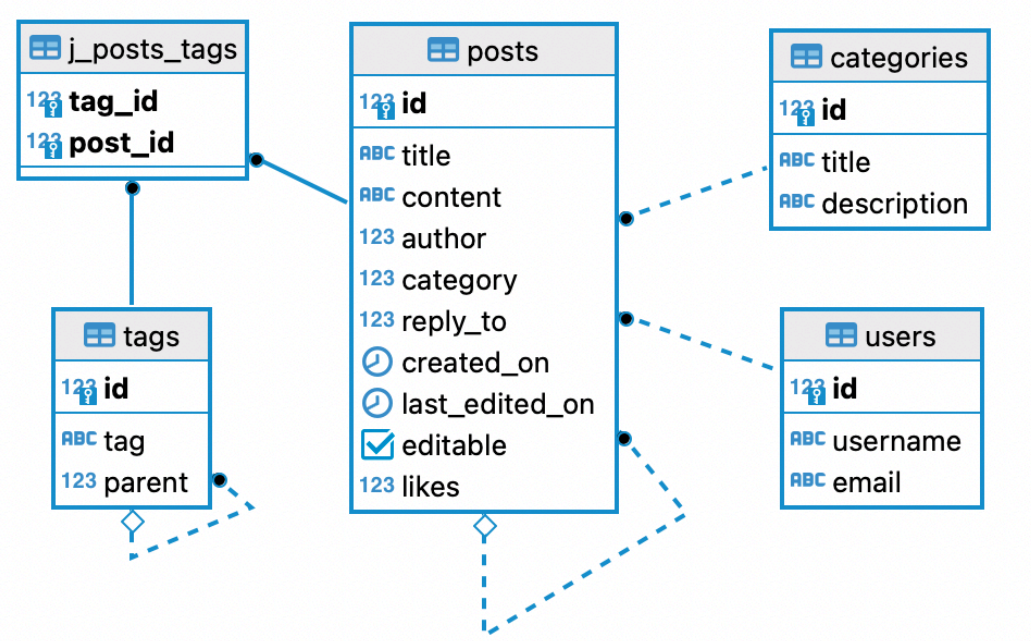
   ```
   SELECT EXTRACT(MONTH FROM NOW())
   SELECT date_part('month',NOW());
   ```

   Trích xuất giờ, phút, giây từ thời gian hiện tại

   ```
   SELECT date_part('hour',CURRENT_TIME) h,
       date_part('minute',CURRENT_TIME) m,
       date_part('second',CURRENT_TIME) s;
   ```

   Tính tuổi từ ngày sinh

```
SELECT (CURRENT_DATE - '12-31-2000') / 365 AS age
SELECT AGE(CURRENT_DATE, '12-31-2000');
SELECT AGE(timestamp '12-31-2000');

SELECT now(),
      now() - INTERVAL '1 year 3 hours 20 minutes'
          AS "3 hours 20 minutes ago of last year";
```



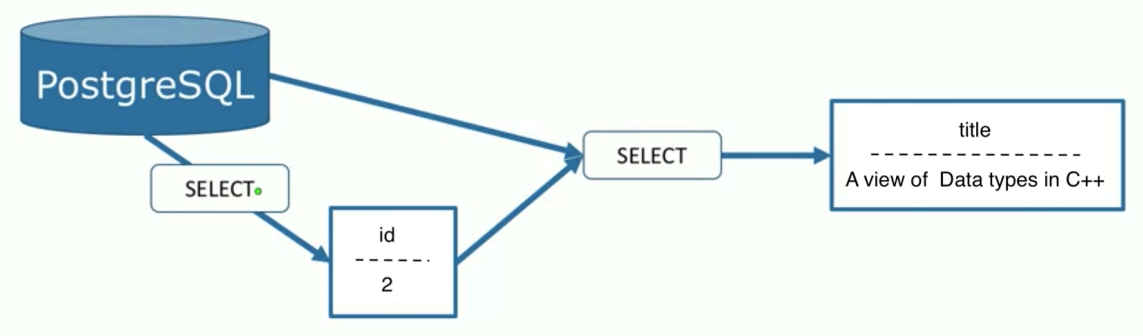
2. **Subquery**

   SELECT id FROM users WHERE email='enrico@pgtraiing.com '; -> id = 2

   SELECT title FROM posts WHERE author=2;

   => Kết hợp 2 câu SELECT trên thành một câu query:

```
SELECT title FROM posts
        WHERE author = ( SELECT id FROM users
                                WHERE email='enrico@pgtraiing.com ') ;
```



SELECT select_list FROM table1
WHERE columnA operator (SELECT columnB FROM table2
WHERE condition);

a. **Subqueries và điều kiện IN/NOT IN**

- **Toán tử IN/ NOT IN**
  - Tìm tất cả dữ liệu trong bảng categories có id = 1 hoặc id = 2
    SELECT * FROM categories WHERE id = 1 OR id = 2;
    SELECT * FROM categories WHERE id IN (1,2);
  - NOT IN: ngược lại của IN
    SELECT * FROM categories WHERE id NOT IN (1,2);
- Hiển thị tất cả các post thuộc về catagory 'Database'
  SELECT * FROM posts WHERE category IN
  (SELECT id FROM categories WHERE title LIKE '%Database%');
- Tìm tất cả các post mà catagory không phải là 'Database'

b. **Subqueries và điều kiện EXIST/NOT EXIST**
SELECT select_list FROM table1 WHERE EXISTS
(SELECT 1 FROM table2 WHERE condition);

- **Toán tử EXIST/ NOT EXIST:** is a boolean operator that checks the existence of rows in a subquery.
- Hiển thị tất cả các post thuộc về catagory 'Database'
  SELECT * FROM posts WHERE EXISTS
  (SELECT 1 FROM categories WHERE title ='Database' AND posts.category=id);

c. **Correlated Subquery**

Hiển thị tất cả các post có lượt likes lớn hơn lượt like trung bình

```
SELECT * FROM posts
        WHERE likes > ( SELECT AVG(likes)FROM posts);
```

d. **ALl**

```
SELECT  *FROM  employees
WHERE  salary > ALL (SELECT salary FROM managers);
```

e. **ANY**

```
SELECT  *FROM  employees
WHERE  salary = ANY (SELECT salary FROM managers);
```

3. **Toán tử UNION**: kết hợp hai hoặc nhiều câu lệnh SELECT lại với nhau và loại bỏ đi các giá trị trùng nhau. UNION ALL : tương tự như UNION nhưng không loại bỏ duplicates
   - Số lượng các cột trong mỗi câu lệnh SELECT phải bằng nhau
   - Data types tương tự nhau
   - Trật tự của các cột phải giống nhau

```
SELECT tag AS datalist FROM tags
UNION
SELECT title AS datalist FROM categories;
```

4. **EXCEPT/INTERSECT**
   a. EXCEPT: so sánh kết quả 2 câu SELECT và trả về các dòng kết quả của câu lệnh thứ nhất mà kết quả này không có trong kết quả của câu lệnh thứ 2.

```
SELECT title AS datalist FROM categories
EXCEPT
SELECT tag AS datalist FROM tags
ORDER BY 1;
```

   b. INTERSET: trả về kết quả chung của hai câu SELECT

```
SELECT title AS datalist FROM categories
        INTERSECT
SELECT tag AS datalist FROM tags;
```

5. **UPDATE multiple records**

Tạo 1 bảng tạm t_categories có cấu trúc giống bảng categories

```
CREATE temp TABLE t_categories AS select * from categories LIMIT 0;
```

Insert dữ liệu vào bảng t_categories

```
INSERT INTO t_categories (id, title, description) VALUES
        (4,'Machine Learning','Machine Learning discussions'),
        (5,'Software engineering','Software engineering discussions');
```

Update dữ liệu từ bảng t_categories vào bảng categories

```
UPDATE categories c SET title=t.title, description=t.description
FROM t_categories t
WHERE c.id=t.id;
```

6. **MERGE**

```
CREATE temp TABLE new_data AS select * from categories limit 0;
INSERT INTO new_data (id,title,description) values
        (1,'Database Discussions','Database discussions'),
        (2,'Unix/Linux discussion','Unix and Linux discussions');

MERGE INTO categories c
USING new_data n ON c.id=n.id
WHEN matched THEN
        UPDATE SET title=n.title, description=n.description
WHEN not matched THEN
        INSERT (id, title, description)
        OVERRIDING SYSTEM VALUE values (n.id, n.title, n.description);
```

7. **RETURNING**

```
INSERT INTO j_posts_tags (tag_id,post_id) values(1,3) RETURNING *;
UPDATE posts SET title = 'A view of  Data types in C++' WHERE id = 3
RETURNING *;
DELETE FROM t_categories WHERE id=4 returning id, title, description;
```

Truy xuất id, title được viết bởi tác giả enrico_pirozzi

```
SELECT id,title FROM
    (SELECT p.* FROM posts p
        INNER JOIN users u ON p.author=u.id
        WHERE u.username='enrico_pirozzi') posts_author_1;
```

8. **CTE** (common table expression): kết quả tạm thời của một câu lệnh SQL

```
WITH cte_name (column list) AS(
        CTE_query_definition
)
statement;
WITH posts_author_1 AS (
        SELECT p.* FROM posts p
         INNER JOIN users u ON p.author=u.id
         WHERE username='enrico_pirozzi')
SELECT id, title FROM posts_author_1;
```

Thực hành CTE:

a. Tạo 1 bảng tạm t_posts có cấu trúc và dữ liệu như bảng posts
b. Tạo bảng delete_posts có cấu trúc như bảng posts (không có dữ liệu)
c. Xóa những dòng dữ liệu trong bảng t_posts có category tên là Database, đồng thời các dòng bị xóa đó được insert vào bảng delete_posts. (CTE)
d. Tạo bảng tạm t_posts2 có cấu trúc và dữ liệu như bảng posts
e. Tạo bảng inserted_posts có cấu trúc như bảng posts (không có dữ liệu)
f. Di chuyển (move) tất cả dữ liệu trong bảng t_posts2 vào bảng inserted_posts. (CTE)

9. **Recursive CTEs (đệ quy)**

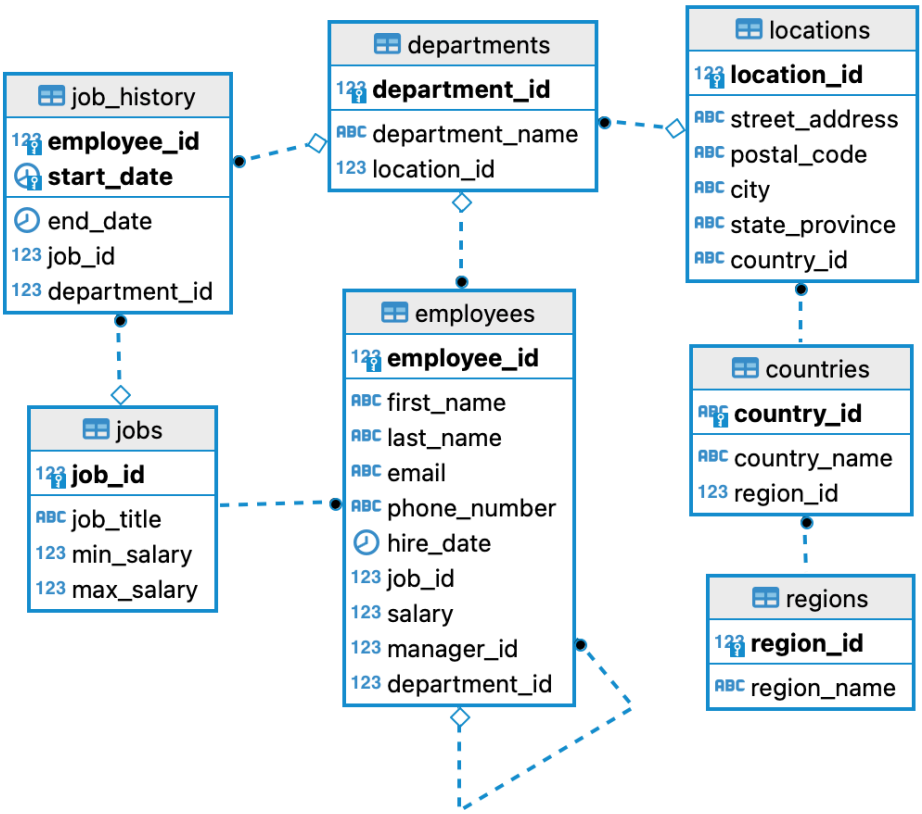
```
WITH RECURSIVE cte_name (column list) AS(
        -- câu lệnh không đệ quy
        SELECT select_list FROM table1 WHERE condition

        UNION [ALL]
        -- câu lệnh đệ quy
```

```
    SELECT select_list FROM cte_name WHERE
recursive_condition
)
SELECT * FROM cte_name;
```

Ví dụ:
```
WITH RECURSIVE tens (n) AS (
        SELECT 10
        UNION ALL
        SELECT n+10 FROM tens WHERE n+10<= 100
    )
    SELECT n FROM tens;
```

```
WITH RECURSIVE tags_tree(tag, id, level) AS (
     -- non recursive statement
    SELECT tag, id, 1
    FROM tags WHERE parent IS NULL
    UNION
    -- recursive statement
    SELECT tt.tag|| ' -> ' || ct.tag, ct.id, tt.level + 1
    FROM tags ct
    JOIN tags_tree tt ON tt.id = ct.parent
)
SELECT level,tag FROM tags_tree ORDER BY level;
```

Bài tập (tiếp theo bài tập tuần 2)



a. Hiển thị danh sách các phòng ban (department_name, city) kèm theo số lượng nhân viên, mức lương thấp nhất, cao nhất, trung bình và tổng lương của phòng ban tương ứng, sắp xếp theo id.
b. Hiển thị danh sách các phòng ban (department_name, city) chỉ thuộc khu vực Americas kèm theo số lượng nhân viên, tổng lương của phòng ban tương ứng, sắp xếp theo tổng lương từ cao đến thấp và chỉ hiển thị danh sách có tổng lương > 30000.
c. Hiển thị danh sách các nhân viên được tuyển dụng vào tháng 6 nhưng loại trừ những nhân viên ở London.
d. Hiển thị danh sách các manager (id, first_name, salary, job_title) có mức lương thuộc vào top 5 mức lương cao nhất.
e. Hiển thị first_name, last_name, salary, manager_id của những nhân viên chịu sự quản lý của các manager làm việc ở 'United States of America' mà có mức lương lớn hơn mức lương trung bình của các thành viên trong nhóm tương ứng.

f. Dùng CTE đệ quy phân chia cây như sau: level 0 là người đứng đầu công ty (employee có manager_id là NULL), level 1 là manager chịu sự quản lý của người ở level 0, level 2 là manager chịu sự quản lý của người ở level 1,...

Kết quả như sau:

level	path	manager_name	employee_name	manager_id	employee_id
0	Steven		Steven		100
1	Steven->Neena	Steven	Neena	100	101
1	Steven->Lex	Steven	Lex	100	102
1	Steven->Den	Steven	Den	100	114
1	Steven->Matthew	Steven	Matthew	100	120
1	Steven->Adam	Steven	Adam	100	121
1	Steven->Payam	Steven	Payam	100	122
1	Steven->Shanta	Steven	Shanta	100	123
1	Steven->John	Steven	John	100	145
1	Steven->Karen	Steven	Karen	100	146
1	Steven->Jonathon	Steven	Jonathon	100	176
1	Steven->Jack	Steven	Jack	100	177
1	Steven->Kimberely	Steven	Kimberely	100	178
1	Steven->Charles	Steven	Charles	100	179
1	Steven->Michael	Steven	Michael	100	201
2	Steven->Lex->Alexander	Lex	Alexander	102	103
2	Steven->Neena->Nancy	Neena	Nancy	101	108
2	Steven->Den->Alexander	Den	Alexander	114	115
2	Steven->Den->Shelli	Den	Shelli	114	116
2	Steven->Den->Sigal	Den	Sigal	114	117
2	Steven->Den->Guy	Den	Guy	114	118
2	Steven->Den->Karen	Den	Karen	114	119
2	Steven->Matthew->Irene	Matthew	Irene	120	126
2	Steven->Shanta->Sarah	Shanta	Sarah	123	192
2	Steven->Shanta->Britney	Shanta	Britney	123	193
2	Steven->Neena->Jennifer	Neena	Jennifer	101	200
2	Steven->Michael->Pat	Michael	Pat	201	202
2	Steven->Neena->Susan	Neena	Susan	101	203
2	Steven->Neena->Hermann	Neena	Hermann	101	204
2	Steven->Neena->Shelley	Neena	Shelley	101	205
3	Steven->Lex->Alexander->Bruce	Alexander	Bruce	103	104
3	Steven->Lex->Alexander->David	Alexander	David	103	105
3	Steven->Lex->Alexander->Valli	Alexander	Valli	103	106
3	Steven->Lex->Alexander->Diana	Alexander	Diana	103	107
3	Steven->Neena->Nancy->Daniel	Nancy	Daniel	108	109
3	Steven->Neena->Nancy->John	Nancy	John	108	110
3	Steven->Neena->Nancy->Ismael	Nancy	Ismael	108	111
3	Steven->Neena->Nancy->Jose Manuel	Nancy	Jose Manuel	108	112
3	Steven->Neena->Nancy->Luis	Nancy	Luis	108	113
3	Steven->Neena->Shelley->William	Shelley	William	205	206

Không bắt buộc:

Dùng CTE đệ quy phân chia cây như sau: Mức 0 là region, mức 1 là country thuộc region đó, mức 2 là city thuộc region-country đó.

Kết quả như sau:

level	region	id
0	Europe	1
0	Americas	2
0	Asia	3
0	Middle East and Africa	4
1	Europe -> Denmark	DK
1	Europe -> Belgium	BE
1	Europe -> France	FR
1	Europe -> Netherlands	NL
1	Europe -> Switzerland	CH
1	Europe -> United Kingdom	UK
1	Europe -> Italy	IT
1	Europe -> Germany	DE
1	Americas -> Canada	CA
1	Americas -> United States of America	US
1	Americas -> Argentina	AR
1	Americas -> Brazil	BR
1	Americas -> Mexico	MX
1	Asia -> India	IN
1	Asia -> China	CN
1	Asia -> Japan	JP
1	Asia -> Singapore	SG
1	Asia -> HongKong	HK
1	Asia -> Australia	AU
1	Middle East and Africa -> Zimbabwe	ZW
1	Middle East and Africa -> Egypt	EG
1	Middle East and Africa -> Israel	IL
1	Middle East and Africa -> Kuwait	KW
1	Middle East and Africa -> Nigeria	NG
1	Middle East and Africa -> Zambia	ZM
2	Europe -> United Kingdom -> Oxford	2500
2	Europe -> United Kingdom -> London	2400
2	Europe -> Germany -> Munich	2700
2	Americas -> Canada -> Toronto	1800
2	Americas -> United States of America -> Seattle	1700
2	Americas -> United States of America -> South San Francisco	1500
2	Americas -> United States of America -> Southlake	1400