

## ASSIGNMENT 1:

Research:

1. Difference between delete and truncate?

**DELETE:** Removes specific rows based on a condition or all rows from a table and creates different transaction log for each row deletion.

**TRUNCATE:** Empties the entire table, removing all rows but creates single transaction log of clearing of the table.

2. Execution order for query processing?

In SQL, the execution order of query processing happens in the order of:

1. **FROM:** selects tables or views from which the data will be retrieved.
2. **WHERE:** condition in this clause is used to select rows from the tables identified in the **FROM** clause.
3. **GROUP BY:** Then this clause is used to group the rows by columns. Functions like SUM, AVG etc are often used with this clause.
4. **HAVING:** Similar to where clause but for specifying filter condition for group by clause
5. **SELECT:** Allows to choose which columns to display
6. **ORDER BY:** Used to sort rows based on certain columns.
7. **LIMIT / OFFSET:** Limit limits no of rows returned and offset skips first n rows returned.

3. Difference between union and union all ?

Union removes duplicate rows returned but union all does not.

## Exercises

```
-- Create two tables
-- Products: columns (product_id, product_name, category and price)
create table
    Products (product_id Int primary key, product_name varchar(69), category
varchar(69), price Decimal(12, 2))
-- Orders : columns( order_id, customer_name, product_id, quantity,
order_date)
create table
    Orders (order_id Int, customer_name varchar(69), product_id INT, quantity
INT, order_date TIMESTAMP Default CURRENT_TIMESTAMP, foreign key (product_id)
references Products (product_id));

-- QUESTIONS:
-- perform CRUD
insert into
    Products (product_id, product_name, category, price)
values
    (1, 'Seedlings - Mix, Organic', 'Prefabricated Aluminum Metal Canopies',
7861),
    (2, 'Nantucket - Pomegranate Pear', 'Site Furnishings', 6624),
    (3, 'Soup - Knorr, Classic Can. Chili', 'Fire Protection', 4944),
```

```
(4, 'Wine - Red, Colio Cabernet', 'Waterproofing & Caulking', 8598),
(5, 'Radish', 'Plumbing & Medical Gas', 4781),
(6, 'Bread Country Roll', 'Exterior Signage', 6980),
(7, 'Radish', 'Landscaping & Irrigation', 851),
(8, 'Beef - Rouladin, Sliced', 'Doors, Frames & Hardware', 5310),
(9, 'Muffin Batt - Ban Dream Zero', 'Structural and Misc Steel
(Fabrication)', 1365),
(10, 'Wine - Fume Blanc Fetzer', 'Prefabricated Aluminum Metal Canopies',
7286),
(11, 'Chivas Regal - 12 Year Old', 'EIFS', 6769),
(12, 'Yogurt - Assorted Pack', 'Fire Protection', 5185),
(13, 'Cups 10oz Trans', 'Hard Tile & Stone', 7176),
(14, 'Cleaner - Comet', 'Roofing (Asphalt)', 3716),
(15, 'Curry Powder', 'Granite Surfaces', 2353);
```

insert into

Orders (order\_id, customer\_name, product\_id, quantity, order\_date)

values

```
(1, 'Codi Casaroli', 1, 13, '1/9/2024'),
(2, 'Murdoch Rodinger', 2, 16, '7/6/2023'),
(3, 'Hardy Eckery', 3, 5, '4/24/2024'),
(4, 'Isadore Murney', 4, 19, '1/4/2024'),
(5, 'Normand Sibbson', 5, 25, '12/24/2023'),
(6, 'Winonah Lindermann', 6, 13, '5/26/2024'),
(7, 'Dorthy Lammie', 7, 19, '9/14/2023'),
(8, 'Vince Kensit', 8, 3, '3/26/2024'),
(9, 'Breena Danne', 3, 25, '11/28/2023'),
```

```
(10, 'Lauralee Guinness', 1, 22, '3/11/2024'),  
(11, 'Vasili Kelsell', 1, 24, '11/22/2023'),  
(12, 'Saw Kinsley', 2, 15, '3/16/2024'),  
(13, 'Celisse Ronca', 3, 20, '8/17/2023'),  
(14, 'Emlynne Plak', 4, 13, '2/16/2024'),  
(15, 'Hobey Mance', 1, 15, '1/6/2024');
```

```
-- View all products in the products table.
```

```
select
```

```
    *
```

```
from
```

```
    Products;
```

```
-- Discount : Update
```

```
update products
```

```
set
```

```
    price = price - 0.1 * price;
```

```
-- Delete some stuff
```

```
DELETE FROM Products
```

```
WHERE
```

```
    product_id NOT IN (
```

```
        SELECT DISTINCT
```

```
            product_id
```

```
        FROM
```

```
            Orders
```

```
    );
```

```
-- Calculate the total quantity ordered for each product category in the orders table.
```

```
SELECT
    category,
    (
        SELECT
            SUM(quantity)
        FROM
            Orders
        WHERE
            product_id IN (
                SELECT
                    product_id
                FROM
                    Products P
                WHERE
                    P.category = OuterP.category
            )
    ) AS total_quantity
FROM
    Products OuterP
GROUP BY
    category;
```

```
-- Find categories where the total number of products ordered is greater than 5.
```

```
SELECT
```

```
    P.category,  
    SUM(O.quantity) AS total_quantity  
FROM  
    Orders O  
    JOIN Products P ON O.product_id = P.product_id  
GROUP BY  
    P.category  
having  
    SUM(O.quantity) > 5;
```