

1. **SELECT, DROP, DELETE, TRUNCATE** : Select Statement review :  
SELECT, FROM, WHERE, GROUP BY, HAVING, ORDER BY.

**DROP** : To drop whole table. Table won't exist anymore.

**Syntax** : DROP TABLE *Table\_Name* ;

**DELETE** : Delete a specific Record/Row. Delete some with conditions

**Syntax** : DELETE FROM *Table\_Name*

WHERE Roll = 102 ; (without where condition, all row  
will be deleted)

**TRUNCATE** : delete all Record/Row from a Table. But Table will exist.

**Syntax** : TRUNCATE TABLE *Table\_Name* ;

2. **Sub Query Basics** : is a Query. A query into another query. Nested.

Example :

SELECT \_\_\_\_ (SELECT \* FROM) → SubQuery \  
FROM \_\_\_\_ (SELECT \* FROM) → SubQuery | → whole thing is a  
Query.  
WHERE \_\_\_\_ (SELECT \* FROM) → SubQuery /

3. **Sub Query Examples** : We can also use **Multiple Sub-Query**. Also  
SubQuery into another SubQuery.

#### 4. Advanced Sub Query 1 :

Advanced Sub Query  
Thursday, March 9, 2023 6:45 PM

employee records (working in other than 'IT\_PROG' department) whose SALARY is less than at least one employee of 'IT\_PROG'

```
SELECT LAST_NAME, JOB_ID, SALARY
FROM EMPLOYEES
WHERE JOB_ID <> 'IT_PROG'
AND SALARY < ANY
(
  SELECT SALARY
  FROM EMPLOYEES
  WHERE JOB_ID = 'IT_PROG'
);
```

Handwritten notes and diagrams:

- IT
- 1 - 1000
- 2 - 2000
- 3 - 500
- 4 - 5000
- 5 - 3000
- 10000X
- Handwritten arrows and symbols indicating comparisons and logic.

#### 5. Advanced Sub Query 2 :

**Co-related Sub Query :** → Relation between main query and sub query. Both query are dependant on each other. Can't return value without other query.

In a correlated sub-query, we use row references of the main query to in the sub-query. Suppose, you need to retrieve those employees whose salary is higher than at least three other employees. To write this query using sub-query, we need to do the following:

```
SELECT *
FROM EMPLOYEES E1
WHERE 3 <=
(
  SELECT COUNT(*)
  FROM EMPLOYEES E2
  WHERE E2.SALARY < E1.SALARY
);
```

Handwritten notes and diagrams:

- main
- sub
- Handwritten arrows and symbols indicating the relationship between the main query and the sub-query.

The following query uses NOT EXISTS to find those employees whose earns the maximum salary in his/her department.

```
SELECT LAST_NAME, SALARY, DEPARTMENT_ID
FROM EMPLOYEES E1
WHERE NOT EXISTS
  (SELECT *
   FROM EMPLOYEES E2
   WHERE E2.DEPARTMENT_ID = E1.DEPARTMENT_ID AND
         E2.SALARY > E1.SALARY)
```

EXISTA — ✓  
NOT EXIST — X

*exist not*

## 6. Summary