

Sub queries

- A subquery is a nested query inserted inside a large query
- Generally occurs with select, from , where
- Also known as inner query or inner select,
- Result of the inner query is passed to the main query

Single row subquery

- It returns zero or single row of data to the outer main query.

Eg.

Select the details of the student with a roll number whose name is 'mohita'

```
SELECT roll, name, section, age
```

```
FROM student
```

```
WHERE roll =
```

```
(SELECT roll
```

```
FROM student
```

```
WHERE name = 'mohita');
```

- You can use operators like =, <, > in single row subqueries
- Select the employee id of the person having minimum salary

Multiple row subquery

- You can use IN, ANY, ALL operators to retrieve the multiple rows from subqueries

Eg. `select * from student`

Where roll in (`select roll from student where age>18`);

1. Write a query to find the salary of employees whose salary is greater than the salary of employee whose id is 100.
2. Write a query to find the employees who all are earning the highest salary

Correlated sub query

They depend upon the data provided by the outer query. They generally include sub queries that include 'EXISTS' clause.

Eg.

Select employee details from employee table if data exists in incentive table ?

```
select * from EMPLOYEE where exists (select *  
from INCENTIVES)
```

Relational algebra operations

- Union
- Intersection
- Cartesian product/cross join
- Set difference
- Selection
- projection

union

- In order to union two table there are a couple of requirements:
- The number of columns must be the same for both select statements.
- The columns, in order, must be of the same data type.

```
select * from table1
```

```
UNION
```

```
Select * from table2
```

Eg.

```
select * from course
```

```
union
```

```
select * from course2
```

Union all

- The UNION ALL set operator returns all rows selected by either query. That means any duplicates will remain in the final result set

Eg.

```
Select * from course
```

```
Union all
```

```
Select * from course2
```

intersection

It returns all the distinct rows from both the queries

```
Select * from table1
```

```
Intersect
```

```
Select * from table2
```

Eg.

```
select * from course
```

```
intersect
```

```
select * from course2
```

Difference (minus)

Select * from table1

Minus

Select * from table2

Eg. Select * from course

Minus

Select * from course2

- Selection – to select rows from the table
- Projection – to select columns from the table

- **Select TOP 2 salary from employee table**

```
select * from (select * from employee order by SALARY desc) where rownum <3
```

- **Select First_Name, LAST_NAME from employee table as separate rows**

```
select FIRST_NAME from EMPLOYEE union select LAST_NAME from EMPLOYEE
```

- **Get Employee ID's of those employees who didn't receive incentives without using sub query ?**

```
select EMPLOYEE_ID from EMPLOYEE
```

MINUS

```
select EMPLOYEE_REF_ID from INCENTIVES
```

- What is the difference between joins and subqueries
- What is the difference between set operators and joins

Join combine columns and set operators combine rows.

Both joins and unions can be used to combine data from one or more tables into a single results. They both go about this in different ways. Whereas a join is used to combine columns from different tables, the union is used to combine rows

Adding zero or more attributes to **candidate key** generates super key. A **candidate key** is a super key but vice versa is not true.

- Decomposition in DBMS removes redundancy, anomalies and inconsistencies from a database by dividing the table into multiple tables.

Lossless Decomposition

- Decomposition is lossless if it is feasible to reconstruct relation R from decomposed tables using Joins. This is the preferred choice. The information will not lose from the relation when decomposed. The join would result in the same original relation.

- **Lossy Decomposition**

As the name suggests, when a relation is decomposed into two or more relational schemas, the loss of information is unavoidable when the original relation is retrieved.