DBMS TUTORIAL - 12

NESTED SUB-QUERIES

U19CS012

1) Write the query for

(i) If user want's to fetch all records of employees whose salary is greater than 25000.

Here, we can can use Nested Query with select statment since it involves Comparision, we will use set comparator "IN"

```
SELECT * FROM EMPLOYEE
WHERE EMPLOYEE_NO IN ( SELECT EMPLOYEE_NO FROM SALARY
WHERE SALARY > 25000);
```

(ii) User has created replica of Employee table and needs the data where salary is greater than 25000.

The Employee table replica name is Employee-Bkp

Here, we can use Nested Query with Insert statement since it involves inserting data to new table .& for comparision , we will use set comparator "IN".

Outer query →

```
INSERT INTO Employee_Bkp
( SELECT * FROM EMPLOYEE
  WHERE EMPLOYEE_NO IN ( SELECT EMPLOYEE_NO FROM SALARY
  WHERE SALARY > 25000));
```

↳ Inner Query

2) Assumptions: Our database has 3 tables :

① students : [id, name, class_id, GPA]

② teachers : [id, name, subject, class_id, monthly_salary]

③ classes : [id, grade, teacher_id, number_of_students]

I)

```
SELECT subject, MAX(salary-by-subject.avg-salary) AS max_salary
FROM (
      SELECT subject, AVG(monthly-salary) AS avg-salary
      FROM    teachers
      GROUP BY subject ) salary-by-subject;
```

Alias Name of table

① Above Query = 'A Nested Query in the FROM clause'

② The inner query (□ Box) returns a table with 2 columns named subject & avg-salary with Alias table name 'salary-by-subject'. (avg-salary by subject)

like

Subject	avg-salary
Maths ~~CSE~~	₹ 51,000
DBMS	₹ 1,00,000
TOC	₹ 79,000

Inner Query returns..

Salary-by-subject TABLE

③ Outer Query : Calculates maximum avg-salary from all tuples in table from inner query.

Final result :

Subject	max-salary
DBMS	1,00,000

Therefore, above Query is used to find out subject corresponding to highest average teacher salary.

II)
```
SELECT *
FROM students
WHERE classid = (
        SELECT id
        FROM classes
        WHERE number-of-students = (
                                        ente
                                    SELECT MAX(number-of-stud
                                    FROM classes)) ;
```

① Above Query is example of " Multiple nested Queries (spc 3) in one statement"

Innermost Query □ (pencil Box) : selects the tuple with maximum no. of students and returns its count

Inner Query □ (pen Box) : ~~$~~ returns id from classes table corresponding to maximum no. of students

ANSWER: Outermost Query : Displays all the information about the (*) Students in class with maximum no. of students

3.> I)
```
SELECT * FROM EMPLOYEE
WHERE DEPT_ID = ( SELECT DEPT_ID FROM DEPARTMENTS ) ;
```
OUTPUT : ERROR

: Subquery returns more than one row.

∵ Where clause can't definately identify unique dept-id from list of dept-id returned from inner query.

Solution : Use of set comparator "IN"

When a sub-query returns multiple values, the IN keyword is used to compare if the value for checking is present in multiple values from subquery.

```
SELECT * FROM EMPLOYEE
WHERE DEPT_ID
IN ( SELECT DEPT_ID FROM DEPARTMENTS );
```

II)

```
Select * from employee where
emp-id in ( select emp-id
            from employee
            where salary > 10000 );
```

(2, 3, 4)

→ return multiple EMP_ID's with salary > 10000

Output :

EMP_ID	EMMP_NAME	AGEG	PHONE_NUM	DEPT_ID	SALARY
2	Linda	30	100234565	1	15000
3	Max	40	122222344	3	22000
4	Well	40	12323424	3	31000

above query will output all detail's of employee's whose salary > 10000

subquery returns id (2, 3, 4) & main query would check if emp-id of each row is present in values returned from sub-query. (2, 3, 4)

If present, then display all the detail.

owput same as :

```
SELECT *
FROM EMPLOYEE
WHERE SALARY > 10000;
```

4.) Consider the example of account payable system,
details of vendors & purchase orders are maintained
Goods are provided on 60 days credit.
Suppose there has been change to the credit amount to 20%
for a vendor from existing 10% ⟶ 20%

∴ We need to change credit Amount = 'UPDATE' keywords
need to be used

① Firstly, we need decide which vendor ??
Let the vendor with name "Harsh" (credit → 20%)

Get his ID,

```
SELECT   VENDOR-ID   FROM   VENDOR
WHERE   VENDOR_NAME = 'Harsh'
```

② Using this ID, we will use 'SET' keyword to change his credit amount to 20% (from existing 10%) &

~~UPDAD~~

ANS:

```
UPDATE   PURCHASE-TRANSACTION
SET   CREDIT-AMT   = AMT * 0.20          // 20% (changed)
WHERE   VENDOR_ID = ( SELECT   VENDOR-ID   FROM   VENDOR
                      WHERE   VENDOR-NAME = 'Harsh' )
```

This is excellent example of using Nested queries with Update.

5.) Write a query in SQL to find name,
city,
total sum of orders,
a salesman collects. Salesman should belong to the cities
where any customer belongs.

① Salesman should belong to cities where any of the customer belongs

```
salesman.city IN ( select city FROM customer )
```

Problems with above sub-query, ~~ct~~ city names would be repeated.

```
salesman.city IN ( select DISTINCT city
                   FROM customer );
```

② Total sum of orders a salesman collects

```
SELECT salesman_id, SUM( orders.purch_amt ) AS total_sales
FROM orders
GROUP BY salesman_id
```

① Return table of form

salesman_id	total_sales
57	27,000
64	55,000
23	71,000

Sales TABLE

Orders table has all orders of customers with there purchase amount (purch_amt)

② let's give this virtual table alias 'sales' TABLE

③ Need to find name & city of salesman,

```
SELECT salesman.name, salesman.city
FROM salesman
```

On Combining all the 3 sub-parts of the Question,

We get the below mentioned ~~sub~~ Main query

```
SELECT   salesman.name,  salesman.city,  sales.total-sales
FROM     salesman,
         ( SELECT   salesman-id, SUM (orders.purch-amt) AS total-sales
           FROM     orders
           GROUPBY  salesman-id ) sales   ←   Virtual table
WHERE    sales.salesman_id  =  salesman.salesman_id   AND
         salesman.city  IN  ( SELECT  DISTINCT  city
                              FROM  customer )
                                                  ;
```

×

SUBMITTED BY:

U19CS012

BHAGYA VINOD RANA

B-TECH (II)

CSE.