**Nested Queries in SQL**

In nested queries, a query is written inside a query. The result of inner query is used in execution of outer query. We will use **STUDENT, COURSE, STUDENT\_COURSE** tables for understanding nested queries.

**STUDENT**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S\_ID** | **S\_NAME** | **S\_ADDRESS** | **S\_PHONE** | **S\_AGE** |
| S1 | RAM | DELHI | 9455123451 | 18 |
| S2 | RAMESH | GURGAON | 9652431543 | 18 |
| S3 | SUJIT | ROHTAK | 9156253131 | 20 |
| S4 | SURESH | DELHI | 9156768971 | 18 |

**COURSE**

|  |  |
| --- | --- |
| **C\_ID** | **C\_NAME** |
| C1 | DSA |
| C2 | Programming |
| C3 | DBMS |

**STUDENT\_COURSE**

|  |  |
| --- | --- |
| **S\_ID** | **C\_ID** |
| S1 | C1 |
| S1 | C3 |
| S2 | C1 |
| S3 | C2 |
| S4 | C2 |
| S4 | C3 |

There are mainly two types of nested queries:

* **Independent Nested Queries:** In independent nested queries, query execution starts from innermost query to outermost queries. The execution of inner query is independent of outer query, but the result of inner query is used in execution of outer query. Various operators like IN, NOT IN, ANY, ALL etc are used in writing independent nested queries
* **IN:** If we want to find out **S\_ID** who are enrolled in **C\_NAME** ‘DSA’ or ‘DBMS’, we can write it with the help of independent nested query and IN operator. From **COURSE** table, we can find out **C\_ID**for **C\_NAME** ‘DSA’ or DBMS’ and we can use these **C\_ID**s for finding **S\_ID**s from **STUDENT\_COURSE** TABLE.

**STEP 1:** Finding **C\_ID** for **C\_NAME** =’DSA’ or ‘DBMS’

Select **C\_ID** from **COURSE** where **C\_NAME** = ‘DSA’ or **C\_NAME** = ‘DBMS’

**STEP 2:** Using **C\_ID** of step 1 for finding **S\_ID**

Select **S\_ID** from **STUDENT\_COURSE** where **C\_ID** IN

(SELECT **C\_ID** from **COURSE** where **C\_NAME** = ‘DSA’ or **C\_NAME**=’DBMS’);

The inner query will return a set with members C1 and C3 and outer query will return those **S\_ID**s for which **C\_ID** is equal to any member of set (C1 and C3 in this case). So, it will return S1, S2 and S4.

**Note:** If we want to find out names of **STUDENT**s who have either enrolled in ‘DSA’ or ‘DBMS’, it can be done as:

Select S\_NAME from **STUDENT** where **S\_ID** IN

(Select **S\_ID** from **STUDENT\_COURSE** where **C\_ID** IN

(SELECT **C\_ID** from **COURSE** where **C\_NAME**=’DSA’ or **C\_NAME**=’DBMS’));

**NOT IN:**If we want to find out **S\_ID**s of **STUDENT**s who have neither enrolled in ‘DSA’ nor in ‘DBMS’, it can be done as:

Select **S\_ID** from **STUDENT** where **S\_ID** NOT IN

(Select **S\_ID** from **STUDENT\_COURSE** where **C\_ID** IN

(SELECT **C\_ID** from **COURSE** where **C\_NAME**=’DSA’ or **C\_NAME**=’DBMS’));

The innermost query will return a set with members C1 and C3. Second inner query will return those **S\_ID**s for which **C\_ID** is equal to any member of set (C1 and C3 in this case) which are S1, S2 and S4. The outermost query will return those **S\_ID**s where **S\_ID** is not a member of set (S1, S2 and S4). So it will return S3.

* **Co-related Nested Queries:** In co-related nested queries, the output of inner query depends on the row which is being currently executed in outer query. e.g.; If we want to find out **S\_NAME** of **STUDENT**s who are enrolled in **C\_ID** ‘C1’, it can be done with the help of co-related nested query as:

Select S\_NAME from **STUDENT** S where EXISTS

( select \* from **STUDENT\_COURSE** SC where S.**S\_ID**=SC.**S\_ID** and SC.**C\_ID**=’C1’);

For each row of **STUDENT** S, it will find the rows from **STUDENT\_COURSE** where S.**S\_ID** = SC.**S\_ID** and SC.**C\_ID**=’C1’. If for a **S\_ID** from **STUDENT** S, atleast a row exists in **STUDENT\_COURSE** SC with **C\_ID**=’C1’, then inner query will return true and corresponding **S\_ID** will be returned as output.