

University of Central Florida

CGS 2545

Database Concepts

DEPARTMENT OF ELECTRICAL ENGINEERING & COMPUTER SCIENCE
COMPUTER SCIENCE DIVISION

Sub Queries

- A Subquery or Inner query or a Nested query is a query within another SQL query and embedded within the WHERE clause.
- A subquery is used to return data that will be used in the main query as a condition to further restrict the data to be retrieved.

Sub Queries

- Subqueries can be used with
 - SELECT statements
 - INSERT statements
 - UPDATE statements
 - DELETE statements
 - operators
 - =
 - <
 - >
 - >=
 - <=
 - IN
 - BETWEEN
 - Etc...

Sub Queries

- There are a few rules that subqueries must follow
 - Subqueries must be enclosed within parentheses.
 - A subquery can have only one column in the SELECT clause, unless multiple columns are in the main query for the subquery to compare its selected columns.
 - An ORDER BY command cannot be used in a subquery, although the main query can use an ORDER BY. The GROUP BY command can be used to perform the same function as the ORDER BY in a subquery.

Sub Queries

- There are a few rules that subqueries must follow
 - Subqueries that return more than one row can only be used with multiple value operators such as the IN operator.
 - The SELECT list cannot include any references to values that evaluate to a BLOB, ARRAY, CLOB, or NCLOB.
 - A subquery cannot be immediately enclosed in a set function.
 - The BETWEEN operator cannot be used with a subquery. However, the BETWEEN operator can be used within the subquery.

Sub Queries

- Subqueries with the SELECT Statement
 - Subqueries are most frequently used with the SELECT statement.
 - The basic syntax is as follows

```
SELECT column_name [, column_name ]  
FROM   table1 [, table2 ]  
WHERE  column_name OPERATOR  
       (SELECT column_name [, column_name ]  
        FROM table1 [, table2 ]  
        [WHERE])
```

Sub Queries

- Example
 - Consider the CUSTOMERS table having the following records

ID	NAME	AGE	ADDRESS	SALARY
1	Ramesh	35	Ahmedabad	2000.00
2	Khilan	25	Delhi	1500.00
3	kaushik	23	Kota	2000.00
4	Chaitali	25	Mumbai	6500.00
5	Hardik	27	Bhopal	8500.00
6	Komal	22	MP	4500.00
7	Muffy	24	Indore	10000.00

Sub Queries

- Example
 - Now, let us check the following subquery with a SELECT statement.

```
SQL> SELECT *  
      FROM CUSTOMERS  
      WHERE ID IN (SELECT ID  
                   FROM CUSTOMERS  
                   WHERE SALARY > 4500) ;
```

+	-----	+	-----	+	-----	+
	ID		NAME		AGE	ADDRESS SALARY
+	-----	+	-----	+	-----	+
	4		Chaitali		25	Mumbai 6500.00
	5		Hardik		27	Bhopal 8500.00
	7		Muffy		24	Indore 10000.00
+	-----	+	-----	+	-----	+

Sub Queries

- Subqueries with the INSERT Statement
 - Subqueries also can be used with INSERT statements.
 - The INSERT statement uses the data returned from the subquery to insert into another table.
 - The selected data in the subquery can be modified with any of the character, date or number functions.

Sub Queries

- Subqueries with the INSERT Statement
 - The basic syntax is as follows.

```
INSERT INTO table_name [ (column1 [, column2 ]) ]  
    SELECT [ *|column1 [, column2 ]  
    FROM table1 [, table2 ]  
    [ WHERE VALUE OPERATOR ]
```

Sub Queries

- Example
 - Consider a table CUSTOMERS_BKP with similar structure as CUSTOMERS table.
 - Now to copy the complete CUSTOMERS table into the CUSTOMERS_BKP table, you can use the following syntax.

```
SQL> INSERT INTO CUSTOMERS_BKP  
      SELECT * FROM CUSTOMERS  
      WHERE ID IN (SELECT ID  
                  FROM CUSTOMERS) ;
```

Sub Queries

- Subqueries with the UPDATE Statement
 - The subquery can be used in conjunction with the UPDATE statement.
 - Either single or multiple columns in a table can be updated when using a subquery with the UPDATE statement.

Sub Queries

- Subqueries with the UPDATE Statement
 - The basic syntax is as follows.

```
UPDATE table
SET column_name = new_value
[ WHERE OPERATOR [ VALUE ]
  (SELECT COLUMN_NAME
   FROM TABLE_NAME)
[ WHERE ) ]
```

Sub Queries

- Example
 - Assuming, we have CUSTOMERS_BKP table available which is backup of CUSTOMERS table.
 - The following example updates SALARY by 0.25 times in the CUSTOMERS table for all the customers whose AGE is greater than or equal to 27.

```
SQL> UPDATE CUSTOMERS
      SET SALARY = SALARY * 0.25
      WHERE AGE IN (SELECT AGE FROM CUSTOMERS_BKP
                    WHERE AGE >= 27 );
```

Sub Queries

- Example
 - This would impact two rows and finally CUSTOMERS table would have the following records.

ID	NAME	AGE	ADDRESS	SALARY
1	Ramesh	35	Ahmedabad	125.00
2	Khilan	25	Delhi	1500.00
3	kaushik	23	Kota	2000.00
4	Chaitali	25	Mumbai	6500.00
5	Hardik	27	Bhopal	2125.00
6	Komal	22	MP	4500.00
7	Muffy	24	Indore	10000.00

Sub Queries

- Subqueries with the DELETE Statement
 - The subquery can be used in conjunction with the DELETE statement like with any other statements mentioned above.
 - The basic syntax is as follows.

```
DELETE FROM TABLE_NAME  
[ WHERE OPERATOR [ VALUE ]  
  (SELECT COLUMN_NAME  
   FROM TABLE_NAME)  
  [ WHERE ) ]
```


Sub Queries

- Example
 - Assuming, we have a CUSTOMERS_BKP table available which is a backup of the CUSTOMERS table.
 - The following example deletes the records from the CUSTOMERS table for all the customers whose AGE is greater than or equal to 27

```
SQL> DELETE FROM CUSTOMERS  
      WHERE AGE IN (SELECT AGE FROM CUSTOMERS_BKP  
                    WHERE AGE >= 27 );
```

Sub Queries

- Example
 - This would impact two rows and finally the CUSTOMERS table would have the following records.

ID	NAME	AGE	ADDRESS	SALARY
2	Khilan	25	Delhi	1500.00
3	kaushik	23	Kota	2000.00
4	Chaitali	25	Mumbai	6500.00
6	Komal	22	MP	4500.00
7	Muffy	24	Indore	10000.00