

20. Subqueries

Subquery is a query inside a large query. Subqueries are also used in IN, ANY(SOME), ALL, and EXISTS statements.

Operator	Description
IN	IN operator is used to check a value available in a given list of values resulting from subquery or an array of values.
ALL	ALL operator is used to check a value equal to each value of a given list resulting from a subquery.
SOME = ANY	SOME or ANY operator is used to check a value equal to one of a given list of values resulting from subquery.
EXISTS	EXISTS operator is used to checking the existence of a subquery is true.

Use the below SQL script to create tables and data for examples.

Data Script:

```
-- Create UserTypes Table
CREATE TABLE UserTypes(Id INT NOT NULL, [Description] VARCHAR(20) NOT NULL,
PRIMARY KEY (Id));

-- Create Users Table
CREATE TABLE Users(
    Id INT NOT NULL,
    Code VARCHAR(4) NOT NULL,
    UserId INT NULL,
    IsActive BIT NOT NULL,
    PRIMARY KEY (Id)
);

-- Insert User Types
INSERT INTO UserTypes(Id,Description)
VALUES (1, 'Super'), (2, 'Admin'), (3, 'Normal'), (4, 'Guest');

--Insert Users
INSERT INTO Users(Id, Code, UserId, IsActive)
VALUES (1, '0001', 1, 1), (2, '0002', 2, 1), (3, '0003', 2, 1),
```

```
(4, '0004', NULL, 1), (5, '0005', 3, 1), (6, '0006', NULL, 1);
```

Sub Queries

Syntax:

```
SELECT Column1, Column2, ... ,(SELECT Column3 FROM Table2 WHERE Condition2)
FROM Table1 WHERE Condition1
```

Example 1:

Below query shows how to get the user type using a subquery.

```
SELECT *, (SELECT UserTypes.Description FROM UserTypes WHERE
Users.UserId = UserTypes.Id) AS UserType FROM Users;
```

Example 2:

The same output can be taken using the below query.

```
SELECT *, UserTypes.Description
FROM Users LEFT JOIN UserTypes
ON Users.UserId = UserTypes.Id;
```

SOME or ANY Operator

SOME or ANY operator is used to check a value equal to one of a given list of values resulting from subquery. SOME operator is available in both **SQL Server** and **MySQL**.

Syntax:

Statement1

```
WHERE Column1 Operator1 = ANY(SELECT Column1 FROM Table1 WHERE Condition1);
```

Example:

The below script will return users where user type is not equal to SUPER.

```
SELECT * FROM Users WHERE UserId = SOME(SELECT Id FROM UserTypes
WHERE [Description] <>
'SUPER');
```

IN Operator

IN operator is used to check a value available in a given list of values resulting from subquery or in an array of values.

Syntax:

```
Statement1 WHERE Column1 Operator1 IN(SELECT Column1 FROM Table1 WHERE Condition1);
```

Example 1:

The below script will return users where user type is not equal to SUPER.

```
SELECT * FROM Users WHERE UserId IN(SELECT Id FROM UserTypes  
WHERE [Description] <> 'SUPER');
```

Example 2:

The below script will return users where user type is equal to 2, 3 or 4.

```
SELECT * FROM Users WHERE UserId IN(2, 3, 4);
```

ALL Operator

ALL operator is used to check a value equal to each value of a given list resulting from a subquery.

Syntax:

Statement1

```
WHERE Column1 Operator1 ALL(SELECT Column1 FROM Table1 WHERE Condition1);
```

Example 1:

The below script will return users where user type is equal to SUPER.

```
SELECT * FROM Users  
WHERE UserId = ALL(SELECT Id FROM UserTypes WHERE  
[Description]='SUPER');
```

Example 2:

Below script will return users where user types are not equal to SUPER and ADMIN.

```
SELECT * FROM Users
WHERE ISNULL(UserTypeId, 0) <> ALL(SELECT Id FROM UserTypes WHERE
                                   [Description] = 'SUPER' OR [Description] = 'ADMIN');
```

EXISTS in SQL

EXISTS operator is used to check the existence of a subquery is true.

EXISTS statements can be written in three steps.

1. Write the Main Query. This query may be a Select or any other statement.
2. Write **WHERE EXISTS** keywords.
3. Write the subquery within brackets. It is a select statement.

```
SELECT Column(s) FROM Table WHERE EXISTS (Subquery)
1           2           3
```

NOT EXISTS operator checks the non-existence of a subquery.

Example 1

The below example returns a list of users if user type is SUPER or ADMIN.

```
SELECT * FROM Users
WHERE EXISTS(SELECT Id FROM UserTypes
              WHERE Users.UserTypeId = UserTypes.Id
              AND ([Description] = 'SUPER' OR [Description] = 'ADMIN'));
```

Example 2:

The below script shows how to handle an IF ELSE statement using EXISTS.

```
DECLARE @UserId INT = 4; DECLARE @UserCode VARCHAR(4) = '004';

-- If the User is available
IF EXISTS(SELECT Id FROM Users WHERE Id = @UserId)
    -- Update the User
    BEGIN
        UPDATE Users SET Code = @UserCode WHERE Id = @UserCode;
    END
```

```
-- If the User is not available  
ELSE  
    -- Insert User  
    BEGIN  
        INSERT INTO Users(Id, Code) VALUES (@UserId, @UserCode);  
    END
```