13-06-2025

1) Querying Data by Using Subqueries

2) Querying Data by Using Subqueries Using the EXISTS,

3) Querying Data by Using Subqueries using ANY,

4) Querying Data by Using Subqueries using ALL Keywords

5) Querying Data by Using Subqueries using Using Nested Subqueries

6) Querying Data by Using Subqueries Using Correlated Subqueries

7) Querying Data by Using Subqueries Using UNION,

8) Querying Data by Using Subqueries using INTERSECT,

9) Querying Data by Using Subqueries using EXCEPT,

10)Querying Data by Using Subqueries using MERGE.

**Create Avengers table**

CREATE TABLE Avengers (

AvengerID INT PRIMARY KEY,

Name VARCHAR(50),

Role VARCHAR(30),

PowerLevel INT

);

CREATE TABLE Missions (

MissionID INT PRIMARY KEY,

MissionName VARCHAR(100),

DifficultyLevel INT,

AssignedTo INT,

FOREIGN KEY (AssignedTo) REFERENCES Avengers(AvengerID)

);

INSERT INTO Avengers VALUES

(1, 'Iron Man', 'Tech Genius', 95),

(2, 'Captain America', 'Leader', 90),

(3, 'Thor', 'God of Thunder', 99),

(4, 'Hulk', 'Strongest Avenger', 97),

(5, 'Black Widow', 'Spy', 88);

INSERT INTO Missions VALUES

(101, 'Battle of New York', 95, 1),

(102, 'Sokovia Incident', 97, 3),

(103, 'Hydra Base Raid', 85, 2),

(104, 'Space Threat', 99, 3),

(105, 'Infiltrate Hydra', 88, 5);

**1. Basic Subquery**

SELECT Name FROM Avengers

WHERE AvengerID IN (

SELECT AssignedTo FROM Missions WHERE DifficultyLevel > 90

);

**2. EXISTS**

SELECT Name FROM Avengers A

WHERE Name = 'Thor'

AND EXISTS (

SELECT 1 FROM Missions M WHERE M.AssignedTo = A.AvengerID

);

**3. ANY**

SELECT Name FROM Avengers

WHERE PowerLevel > ANY (

SELECT DifficultyLevel FROM Missions

);

**4. ALL**

SELECT Name FROM Avengers

WHERE PowerLevel > ALL (

SELECT DifficultyLevel FROM Missions

);

**5. NESTED SUBQUERIES**

SELECT MissionName FROM Missions

WHERE AssignedTo IN (

SELECT AvengerID FROM Avengers

WHERE PowerLevel = (

SELECT MAX(PowerLevel) FROM Avengers

)

);

-- 6. Correlated Subquery

SELECT Name FROM Avengers A

WHERE EXISTS (

SELECT 1 FROM Missions M

WHERE M.AssignedTo = A.AvengerID AND M.DifficultyLevel > A.PowerLevel

);

**7. UNION**

SELECT Name FROM Avengers WHERE PowerLevel > 95

UNION

SELECT A.Name FROM Avengers A

JOIN Missions M ON A.AvengerID = M.AssignedTo

WHERE M.DifficultyLevel > 95;

**8. INTERSECT (Note: Not supported in MySQL, use INNER JOIN and WHERE IN instead)**

SELECT Name FROM Avengers WHERE PowerLevel > 90

AND Name IN (

SELECT A.Name FROM Avengers A

JOIN Missions M ON A.AvengerID = M.AssignedTo

WHERE M.DifficultyLevel > 90

);

**9. EXCEPT (MySQL does not support EXCEPT, use LEFT JOIN and WHERE IS NULL)**

SELECT Name FROM Avengers WHERE PowerLevel > 90

AND Name NOT IN (

SELECT A.Name FROM Avengers A

JOIN Missions M ON A.AvengerID = M.AssignedTo

WHERE M.DifficultyLevel > 90

);

**10. MERGE equivalent (for MySQL: use INSERT ... ON DUPLICATE KEY UPDATE)**

CREATE TABLE AvengerStatus (

AvengerID INT PRIMARY KEY,

PowerLevel INT

);