upGrad ENTERPRISE

SQL Server Querying & Constraints

By
Narasimha Rao T
Microsoft.Net FSD Trainer
Professional Development Trainer
tnrao.trainer@gmail.com

1️⃣ Launching and Connecting to SQL Server

Steps to Connect via SSMS (SQL Server Management Studio)

1. Open **SSMS**.
2. In the **Connect to Server** dialog:
    - **Server type:** Database Engine
    - **Server name:** (e.g. `localhost` or `.\SQLEXPRESS`)
    - **Authentication:** Windows Authentication or SQL Server Authentication
3. Click **Connect**.
4. The Object Explorer will show all available databases.

2 Basic SQL Clauses

(a) WHERE

- Filters records based on conditions.

```
SELECT *
FROM Employees
WHERE Salary > 50000;
```

(b) ORDER BY

- Sorts query results in ascending (`ASC`) or descending (`DESC`) order.

```sql
SELECT Name, Salary
FROM Employees
ORDER BY Salary DESC;
```

(c) GROUP BY

- Groups rows sharing a property for aggregate functions.

```
SELECT DepartmentID, COUNT(*) AS EmployeeCount
FROM Employees
GROUP BY DepartmentID;
```

(d) HAVING

- Filters results after grouping (used with aggregates).

```
SELECT DepartmentID, COUNT(*) AS EmployeeCount
FROM Employees
GROUP BY DepartmentID
HAVING COUNT(*) > 5;
```

(e) Aggregate Functions

Function	Description	Example
`COUNT()`	Counts rows	`COUNT(*)`
`SUM()`	Adds numeric values	`SUM(Salary)`
`AVG()`	Calculates average	`AVG(Salary)`
`MIN()`	Finds smallest value	`MIN(Salary)`
`MAX()`	Finds largest value	`MAX(Salary)`

3 Constraints in SQL Server

Constraints enforce rules on table data.

(a) PRIMARY KEY

- Uniquely identifies each row.

```
CREATE TABLE Employees (
    EmpID INT PRIMARY KEY,
    Name VARCHAR(50)
);
```

(b) FOREIGN KEY

- Establishes a relationship between two tables.

```
CREATE TABLE Departments (
    DeptID INT PRIMARY KEY,
    DeptName VARCHAR(50)
);

CREATE TABLE Employees (
    EmpID INT PRIMARY KEY,
    Name VARCHAR(50),
    DeptID INT FOREIGN KEY REFERENCES Departments(DeptID)
);
```

(c) UNIQUE

- Ensures all values in a column are unique.

```
CREATE TABLE Users (
    UserID INT PRIMARY KEY,
    Email VARCHAR(100) UNIQUE
);
```

(d) CHECK

- Restricts values in a column based on a condition.

```
CREATE TABLE Products (
    ProductID INT PRIMARY KEY,
    Price DECIMAL(8,2) CHECK (Price > 0)
);
```

(e) DEFAULT

- Assigns a default value when no value is provided.

```
CREATE TABLE Orders (
    OrderID INT PRIMARY KEY,
    Status VARCHAR(20) DEFAULT 'Pending'
);
```

4 Column-Level vs Table-Level Constraints

Aspect	Column-Level Constraint	Table-Level Constraint
Definition	Declared within a column definition	Declared separately after all columns
Syntax	Defined inline	Defined at the end of table definition
Example	`EmpID INT PRIMARY KEY`	`PRIMARY KEY (EmpID, DeptID)`
Usage	Used for single-column constraints	Used for multi-column constraints

5 Monitoring SQL Server

(a) Create a Trace for SELECT/INSERT Operations

1. Open **SQL Server Profiler**.
2. Create a **New Trace** and connect to SQL Server.
3. Select **Events**:
    - `TSQL_SPs → RPC:Completed`
    - `TSQL → SQL:BatchCompleted`
4. Apply filters for `SELECT` and `INSERT` statements.
5. Start the trace to monitor queries.

(b) View Real-Time Query Execution

- In SSMS:
  i. Go to **Query → Include Actual Execution Plan** (or press `Ctrl+M`).
  ii. Run your query.
  iii. View the execution plan to analyze performance.

6 Example SQL Script

```sql
CREATE DATABASE CompanyDB;
USE CompanyDB;

CREATE TABLE Departments (
    DeptID INT PRIMARY KEY,
    DeptName VARCHAR(50)
);

CREATE TABLE Employees (
    EmpID INT PRIMARY KEY,
    Name VARCHAR(50),
    Salary DECIMAL(10,2),
    DeptID INT,
    FOREIGN KEY (DeptID) REFERENCES Departments(DeptID),
    CHECK (Salary > 0)
);
```

```sql
-- Insert data
INSERT INTO Departments VALUES (1, 'HR'), (2, 'IT');
INSERT INTO Employees VALUES (101, 'Alice', 60000, 2), (102, 'Bob', 45000, 1);

-- Query with aggregation
SELECT DeptID, AVG(Salary) AS AvgSalary
FROM Employees
GROUP BY DeptID
HAVING AVG(Salary) > 50000;
```


QUIZ
TIME

7 Quiz Questions

1. Write a query to display all employees earning more than 50,000 in descending order of salary.
2. Explain the difference between **HAVING** and **WHERE** clauses.
3. Create a table `Products` with `ProductID`, `Name`, and `Price` (Price should be > 0).
4. Differentiate between **Column-level** and **Table-level** constraints with examples.
5. How can you trace `SELECT` and `INSERT` queries in SQL Server Profiler?
6. What is the purpose of the `DEFAULT` constraint?

upGrad ENTERPRISE

Q & A

Narasimha Rao T
tnrao.trainer@gmail.com