

# Database

---

## Queries used in Session

### 1. DDL (Data Definition Language)

#### CREATE Table

```
CREATE TABLE Students (  
    StudentID INT PRIMARY KEY,  
    FirstName VARCHAR(50),  
    LastName VARCHAR(50),  
    DateOfBirth DATE,  
    Address VARCHAR(100)  
);
```

```
CREATE TABLE OldStudentRecords (  
    StudentID INT PRIMARY KEY,  
    FirstName VARCHAR(50),  
    LastName VARCHAR(50),  
    DateOfBirth DATE,  
    Address VARCHAR(100),  
    Email VARCHAR(50)  
);
```

```
CREATE TABLE Courses (  
    CourseID INT PRIMARY KEY,  
    CourseName VARCHAR(50) NOT NULL,  
    Credits INT NOT NULL  
);
```

```
CREATE TABLE Enrollments (  
    EnrollmentID INT PRIMARY KEY,  
    StudentID INT,  
    CourseID INT,  
    Grade DECIMAL(3, 2),  
    FOREIGN KEY (StudentID) REFERENCES Students(StudentID),  
    FOREIGN KEY (CourseID) REFERENCES Courses(CourseID)  
);
```

#### Alter Table

```
ALTER TABLE Students
ADD Email VARCHAR(50);
```

### Truncate Table

```
TRUNCATE TABLE OldStudentRecords;
```

### Drop Table

```
DROP TABLE OldStudentRecords;
```

## 2. DML (Data Manipulation Language)

### Insert Data

```
INSERT INTO Students
VALUES (1, 'Ashish', 'Nikam', '2005-09-01', '123 Shivaji Nagar, Jalgaon',
'ashish.nikam@example.com');

INSERT INTO Students (StudentID, FirstName, LastName, DateOfBirth, Address, Email)
VALUES
(2, 'Vinod', 'Patil', '2006-02-10', 'Laxmi Road, Pune',
'vinod.patil@example.com'),
(3, 'Raj', 'Shinde', '2007-03-15', 'MG Road, Nagpur', 'raj.shinde@example.com'),
(4, 'Sonali', 'Deshmukh', '2005-04-20', '011 Kala Ghoda Chowk, Mumbai',
'sonali.deshmukh@example.com'),
(5, 'Priya', 'Jadhav', '2008-05-25', '303 College Road, Nashik',
'priya.jadhav@example.com');
```

```
INSERT INTO OldStudentRecords
SELECT * FROM students;
```

```
INSERT INTO Courses
VALUES (1, 'Mathematics', 3);

INSERT INTO Courses (CourseID, CourseName, Credits)
VALUES
(2, 'Science', 4),
(3, 'History', 3),
```

```
(4, 'English', 3),  
(5, 'Computer', 2);
```

```
INSERT INTO Enrollments  
VALUES (1, 1, 1, 3.5);  
  
INSERT INTO Enrollments (EnrollmentID, StudentID, CourseID, Grade)  
VALUES  
(2, 2, 2, 3.8), -- Vinod Patil enrolled in Science with Grade 3.8  
(3, 3, 3, 3.0), -- Raj Shinde enrolled in History with Grade 3.0  
(4, 4, 4, 3.7), -- Sonali Deshmukh enrolled in English with Grade 3.7  
(5, 5, 5, 4.0), -- Priya Jadhav enrolled in Physical Education with Grade 4.0  
(6, 1, 2, 3.6), -- Alice Johnson enrolled in Science with Grade 3.6  
(7, 2, 3, 3.9), -- Vinod Patil enrolled in History with Grade 3.9  
(8, 3, 4, 3.2); -- Raj Shinde enrolled in English with Grade 3.2
```

## Update Data

```
UPDATE Students  
SET Address = 'Mahesh Nagar, Jalgaon'  
WHERE StudentID = 1;
```

## Delete Data

```
DELETE FROM Students  
WHERE StudentID = 1;
```

## 3. DQL (Data Query Language)

### Select Data

```
SELECT FirstName, LastName, Address  
FROM Students  
WHERE DateOfBirth > '2005-01-01';
```

## 4. Constraints

### CREATE TABLE Courses with NOT NULL

```
CREATE TABLE Courses (  
    CourseID INT PRIMARY KEY,
```

```
CourseName VARCHAR(50) NOT NULL,  
Credits INT NOT NULL  
);
```

### CREATE TABLE Students with UNIQUE Email and Primary Key

```
CREATE TABLE Students (  
    StudentID INT PRIMARY KEY,  
    FirstName VARCHAR(50),  
    LastName VARCHAR(50),  
    DateOfBirth DATE,  
    Address VARCHAR(100),  
    Email VARCHAR(50) UNIQUE  
);
```

### CREATE TABLE Enrollments with FOREIGN KEY

```
CREATE TABLE Enrollments (  
    EnrollmentID INT PRIMARY KEY,  
    StudentID INT,  
    CourseID INT,  
    FOREIGN KEY (StudentID) REFERENCES Students(StudentID),  
    FOREIGN KEY (CourseID) REFERENCES Courses(CourseID)  
);
```

### CREATE TABLE Grades with CHECK Constraint

```
CREATE TABLE Grades (  
    StudentID INT PRIMARY KEY,  
    Grade DECIMAL(3, 2),  
    CHECK (Grade BETWEEN 0 AND 4.0)  
);
```

### ALTER TABLE Enrollments with CHECK CONSTRAINT

```
ALTER TABLE Enrollments  
ADD CONSTRAINT CHECK (grade BETWEEN 0 AND 4.0);
```

## 5. Clauses

### WHERE Clause with LIKE

```
SELECT FirstName, LastName
FROM Students
WHERE Address LIKE '%road%';
```

### ORDER BY Clause

```
SELECT FirstName, LastName
FROM Students
ORDER BY LastName ASC;
```

### GROUP BY Clause

```
SELECT COUNT(*), Address
FROM Students
GROUP BY Address;
```

### LIMIT Clause

```
SELECT FirstName, LastName
FROM Students
LIMIT 10;
```

## 6. Operators

### LIKE Operator

```
SELECT FirstName, LastName
FROM Students
WHERE LastName LIKE 'J%';
```

### AND Operator

```
SELECT FirstName, LastName
FROM Students
WHERE Address LIKE '%Road%' AND DateOfBirth > '2005-01-01';
```

### OR Operator

```
SELECT FirstName, LastName
FROM Students
WHERE Address LIKE '%Pune%' OR Address LIKE '%Nashik%';
```

### **BETWEEN Operator**

```
SELECT FirstName, LastName
FROM Students
WHERE DateOfBirth BETWEEN '2000-01-01' AND '2005-12-31';
```

## 7. Aggregate Functions

### **COUNT()**

```
SELECT COUNT(*)
FROM Students;
```

### **AVG()**

```
SELECT AVG(Grade)
FROM Enrollments;
```

### **SUM()**

```
SELECT SUM(Credits)
FROM Courses;
```

### **MIN()**

```
SELECT MIN(DateOfBirth)
FROM Students;
```

### **MAX()**

```
SELECT MAX(DateOfBirth)
FROM Students;
```

## 8. Joins

### INNER JOIN

```
SELECT Students.FirstName, Students.LastName, Courses.CourseName
FROM Students
INNER JOIN Enrollments ON Students.StudentID = Enrollments.StudentID
INNER JOIN Courses ON Enrollments.CourseID = Courses.CourseID;
```

### LEFT JOIN

```
SELECT Students.FirstName, Students.LastName, Grades.Grade
FROM Students
LEFT JOIN Grades ON Students.StudentID = Grades.StudentID;
```

### RIGHT JOIN

```
SELECT Courses.CourseName, Students.FirstName, Students.LastName
FROM Courses
RIGHT JOIN Enrollments ON Courses.CourseID = Enrollments.CourseID
RIGHT JOIN Students ON Enrollments.StudentID = Students.StudentID;
```

### FULL OUTER JOIN

```
SELECT Students.FirstName, Students.LastName, Courses.CourseName
FROM Students
FULL OUTER JOIN Enrollments ON Students.StudentID = Enrollments.StudentID
FULL OUTER JOIN Courses ON Enrollments.CourseID = Courses.CourseID;
```

### SELF JOIN

```
SELECT A.FirstName AS MentorFirstName, B.FirstName AS MenteeFirstName
FROM Students A, Students B
WHERE A.StudentID = B.MentorID;
```