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 (
    RecordID INT PRIMARY KEY,
    StudentID INT,
    RecordDetails TEXT
);
```

```
CREATE TABLE TemporaryStudentData (
    TempID INT PRIMARY KEY,
    TempDetails TEXT
);
```

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

```
CREATE TABLE Classes (
    ClassID INT PRIMARY KEY,
    ClassName VARCHAR(50)
);
```

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

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

Alter Table

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

Drop Table

```sql
DROP TABLE OldStudentRecords;
```

Truncate Table

```sql
TRUNCATE TABLE TemporaryStudentData;
```

2. DML (Data Manipulation Language)

Insert Data

```sql
INSERT INTO Students (StudentID, FirstName, LastName, DateOfBirth, Address)
VALUES (1, 'Alice', 'Johnson', '2005-09-01', '123 Elm St');
```

```sql
INSERT INTO Courses (CourseID, CourseName, Credits)
VALUES (1, 'Mathematics', 3);
```

```sql
INSERT INTO Enrollments (EnrollmentID, StudentID, CourseID)
VALUES (1, 1, 1);
```

Update Data

```sql
UPDATE Students
SET Address = '456 Maple St'
WHERE StudentID = 1;
```

Delete Data

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

3. DQL (Data Query Language)

Select Data

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

4. Constraints

CREATE TABLE Courses with NOT NULL

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

CREATE TABLE Students with UNIQUE Email

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

CREATE TABLE Classes with PRIMARY KEY

```
CREATE TABLE Classes (
    ClassID INT PRIMARY KEY,
    ClassName VARCHAR(50)
);
```

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)
);
```

5. Clauses

WHERE Clause

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

ORDER BY Clause

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

GROUP BY Clause

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

LIMIT Clause

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

6. Operators

AND Operator

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

OR Operator

```sql
SELECT FirstName, LastName
FROM Students
WHERE Address LIKE '%Elm%' OR Address LIKE '%Maple%';
```

LIKE Operator

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

BETWEEN Operator

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

7. Aggregate Functions

COUNT()

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

AVG()

```sql
SELECT AVG(Grade)
FROM Grades;
```

SUM()

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

MIN()

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

MAX()

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

8. Joins

INNER JOIN

```sql
SELECT Students.FirstName, Students.LastName, Courses.CourseName
FROM Students
```

```sql
INNER JOIN Enrollments ON Students.StudentID = Enrollments.StudentID
INNER JOIN Courses ON Enrollments.CourseID = Courses.CourseID;
```

LEFT JOIN

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

RIGHT JOIN

```sql
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

```sql
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

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