Practical Assignment – 3

- Name: Hardik Arora
- Branch : Btech - CS
- Program: AIML
- University Roll No. : 2215500071
- Section: 2AC
- Class Roll No. : 28

Imagine you are managing a comprehensive database system for an academic institution that tracks essential information about colleges, students, and their application records. Create Student, Apply and College tables using script.

Run the following Script:

```
Create DATABASE p3;

USE p3;

CREATE TABLE College (
    CollegeID INT PRIMARY KEY,
    Name VARCHAR(255) NOT NULL,
    Location VARCHAR(255)
);

CREATE TABLE Student (
    StudentID INT PRIMARY KEY,
    Name VARCHAR(255) NOT NULL,
    Age INT,
    GPA FLOAT
);

CREATE TABLE Apply (
    StudentID INT,
```

```sql
    CollegeID INT,
    Decision VARCHAR(50),
    CONSTRAINT PK_Apply PRIMARY KEY (StudentID, CollegeID),
    CONSTRAINT FK_StudentID FOREIGN KEY (StudentID) REFERENCES Student (StudentID),
    CONSTRAINT FK_CollegeID FOREIGN KEY (CollegeID) REFERENCES College (CollegeID)
);

DROP TABLE IF EXISTS Apply;
DROP TABLE IF EXISTS Student;
DROP TABLE IF EXISTS College;

CREATE TABLE College(
    collegeName VARCHAR(10) PRIMARY KEY,
    state VARCHAR(10),
    enrollment INT
);

CREATE TABLE Student(
    sID INT PRIMARY KEY,
    sName VARCHAR(10),
    GPA DECIMAL(3, 1),
    sizeHS INT
);

CREATE TABLE Apply(
    sID INT,
    cName VARCHAR(10),
    major VARCHAR(20),
    decision CHAR(1),
    PRIMARY KEY(sID, major, cName),
    FOREIGN KEY(sID) REFERENCES Student(sID),
    FOREIGN KEY(cName) REFERENCES College(collegeName)
);


INSERT INTO College VALUES ('Stanford', 'CA', 15000);
INSERT INTO College VALUES ('Berkeley', 'CA', 36000);
INSERT INTO College VALUES ('MIT', 'MA', 10000);
INSERT INTO College VALUES ('Cornell', 'NY', 21000);
INSERT INTO College VALUES ('Harvard', 'MA', 50040);

INSERT INTO Student VALUES (123, 'Amy', 3.9, 1000);
INSERT INTO Student VALUES (234, 'Bob', 3.6, 1500);
INSERT INTO Student VALUES (345, 'Craig', 3.5, 500);
INSERT INTO Student VALUES (456, 'Doris', 3.9, 1000);
INSERT INTO Student VALUES (567, 'Edward', 2.9, 2000);
INSERT INTO Student VALUES (678, 'Fay', 3.8, 200);
INSERT INTO Student VALUES (789, 'Gary', 3.4, 800);
INSERT INTO Student VALUES (987, 'Helen', 3.7, 800);
INSERT INTO Student VALUES (876, 'Irene', 3.9, 400);
INSERT INTO Student VALUES (765, 'Jay', 2.9, 1500);
INSERT INTO Student VALUES (654, 'Amy', 3.9, 1000);
INSERT INTO Student VALUES (543, 'Craig', 3.4, 2000);

INSERT INTO Apply VALUES (123, 'Stanford', 'CS', 'Y');
INSERT INTO Apply VALUES (123, 'Stanford', 'EE', 'N');
INSERT INTO Apply VALUES (123, 'Berkeley', 'CS', 'Y');
INSERT INTO Apply VALUES (123, 'Cornell', 'EE', 'Y');
INSERT INTO Apply VALUES (234, 'Berkeley', 'biology', 'N');
INSERT INTO Apply VALUES (345, 'MIT', 'bioengineering', 'Y');
INSERT INTO Apply VALUES (345, 'Cornell', 'bioengineering', 'N');
INSERT INTO Apply VALUES (345, 'Cornell', 'CS', 'Y');
INSERT INTO Apply VALUES (345, 'Cornell', 'EE', 'N');
INSERT INTO Apply VALUES (678, 'Stanford', 'history', 'Y');
INSERT INTO Apply VALUES (987, 'Stanford', 'CS', 'Y');
INSERT INTO Apply VALUES (987, 'Berkeley', 'CS', 'Y');
INSERT INTO Apply VALUES (876, 'Stanford', 'CS', 'N');
INSERT INTO Apply VALUES (876, 'MIT', 'biology', 'Y');
INSERT INTO Apply VALUES (876, 'MIT', 'marine biology', 'N');
INSERT INTO Apply VALUES (765, 'Stanford', 'history', 'Y');
INSERT INTO Apply VALUES (765, 'Cornell', 'history', 'N');
INSERT INTO Apply VALUES (765, 'Cornell', 'psychology', 'Y');
INSERT INTO Apply VALUES (543, 'MIT', 'CS', 'N');

SELECT * FROM apply;
```

Q. Exercise:-

Write SQL queries for each of the following:

Q1. Produce a combine table in which each student is combine with every other application.

```
SELECT
    s.sID AS student_id,
    s.sName AS student_name,
    a.cName AS college_name,
    a.major AS major,
    a.decision AS decision
FROM
    Student s
CROSS JOIN
    Apply a
ORDER BY
    s.sID,
    a.cName;
```

Q2. Give Student ID, name, GPA and name of college and major each student applied to.

```
SELECT
    s.sID AS student_id,
    s.sName AS student_name,
    s.GPA,
    a.cName AS college_name,
    a.major
FROM
    Student s
JOIN
    Apply a ON s.sID = a.sID
ORDER BY
    s.sID,
    a.cName;
```

Q3. Find detail of applications who applied to California State.

```
SELECT
    s.sID AS student_id,
    s.sName AS student_name,
    s.GPA,
    c.collegeName AS college_name,
    a.major
FROM
    Apply a
JOIN
    Student s ON a.sID = s.sID
```

```sql
JOIN
    College c ON a.cName = c.collegeName
WHERE
    c.state = 'CA';
```

Q4. IDs, name, GPA of students and name of college with GPA > 3.7 applying to Stanford

```sql
SELECT
    s.sID AS student_id,
    s.sName AS student_name,
    s.GPA,
    c.collegeName AS college_name
FROM
    Student s
JOIN
    Apply a ON s.sID = a.sID
JOIN
    College c ON a.cName = c.collegeName
WHERE
    s.GPA > 3.7
    AND c.collegeName = 'Stanford';
```

Q5. Find detail of Student who apply to CS major and their application are rejected

```sql
SELECT
    s.sID AS student_id,
    s.sName AS student_name,
    s.GPA,
    c.collegeName AS college_name,
    a.major,
    a.decision
FROM
    Student s
JOIN
    Apply a ON s.sID = a.sID
JOIN
    College c ON a.cName = c.collegeName
WHERE
    a.major = 'CS'
    AND a.decision = 'N';
```

Q6. Find detail of student and application who applied to colleges at New York.

```sql
SELECT
    s.sID AS student_id,
    s.sName AS student_name,
    s.GPA,
    c.collegeName AS college_name,
    a.major,
    a.decision
FROM
    Student s
JOIN
    Apply a ON s.sID = a.sID
JOIN
    College c ON a.cName = c.collegeName
WHERE
    c.state = 'NY';
```

Q7. Find detail of student who have not applied to any of college.

```sql
SELECT
    s.sID AS student_id,
    s.sName AS student_name,
    s.GPA
FROM
    Student s
LEFT JOIN
    Apply a ON s.sID = a.sID
WHERE
    a.sID IS NULL
```

Q8. Find college where no student have applied

```sql
SELECT
    c.collegeName AS college_name
FROM
    College c
LEFT JOIN
    Apply a ON c.collegeName = a.cName
WHERE
    a.cName IS NULL;
```

Q9. Find sID who have only one application.

```sql
SELECT
    sID
FROM
    Apply
GROUP BY
    sID
HAVING
    COUNT(*) = 1;
```

Q10. Find name and GPA of applicants who apply to any college whose enrollment is not morethan 25000.

```sql
SELECT
    s.sName AS student_name,
    s.GPA
FROM
    Student s
JOIN
    Apply a ON s.sID = a.sID
JOIN
    College c ON a.cName = c.collegeName
WHERE
    c.enrollment <= 25000;
```

Q11. Find pair of students (sID) having same GPA. (*each pair should occur just once in result*)

```sql
SELECT
    DISTINCT s1.sID AS student1_id,
    s2.sID AS student2_id
FROM
    Student s1
JOIN
    Student s2 ON s1.sID < s2.sID AND s1.GPA = s2.GPA
```

Exercise:

For each of the following you need to write three queries i.e. three version first using :CROSS Join Second using: Natural Join And third using: Inner Join

Q12. Find student and major he / she applied to.

```sql
-- Query using CROSS JOIN
SELECT s.sID, s.sName, a.major
FROM Student s
CROSS JOIN Apply a;

-- Query using NATURAL JOIN
SELECT s.sID, s.sName, a.major
FROM Student s
NATURAL JOIN Apply a;

-- Query using INNER JOIN
SELECT s.sID, s.sName, a.major
FROM Student s
INNER JOIN Apply a ON s.sID = a.sID;
```

Q13. Find detail of student who came from high school have size less than 20000 and applied to CSat Stanford.

```sql
-- Query using CROSS JOIN
SELECT s.sID, s.sName, s.GPA, s.sizeHS, a.cName, a.major
FROM Student s
CROSS JOIN Apply a
JOIN College c ON a.cName = c.collegeName
WHERE s.sizeHS < 20000 AND s.sizeHS IS NOT NULL
AND a.major = 'CS' AND c.collegeName = 'Stanford';

-- Query using NATURAL JOIN
SELECT s.sID, s.sName, s.GPA, s.sizeHS, a.cName, a.major
FROM Student s
NATURAL JOIN Apply a
JOIN College c ON a.cName = c.collegeName
WHERE s.sizeHS < 20000 AND s.sizeHS IS NOT NULL
AND a.major = 'CS' AND c.collegeName = 'Stanford';

-- Query using INNER JOIN
SELECT s.sID, s.sName, s.GPA, s.sizeHS, a.cName, a.major
FROM Student s
INNER JOIN Apply a ON s.sID = a.sID
JOIN College c ON a.cName = c.collegeName
WHERE s.sizeHS < 20000 AND s.sizeHS IS NOT NULL
AND a.major = 'CS' AND c.collegeName = 'Stanford';
```

Q14. Provide complete detail of each student where they applied what major they applied to whatwas the decision and complete detail of college they applied.

```sql
-- Query using CROSS JOIN
SELECT s.sID, s.sName, s.GPA, s.sizeHS, a.cName, a.major
FROM Student s
CROSS JOIN Apply a
JOIN College c ON a.cName = c.collegeName
WHERE s.sizeHS < 20000 AND s.sizeHS IS NOT NULL
AND a.major = 'CS' AND c.collegeName = 'Stanford';

-- Query using NATURAL JOIN
SELECT s.sID, s.sName, s.GPA, s.sizeHS, a.cName, a.major
FROM Student s
NATURAL JOIN Apply a
JOIN College c ON a.cName = c.collegeName
WHERE s.sizeHS < 20000 AND s.sizeHS IS NOT NULL
AND a.major = 'CS' AND c.collegeName = 'Stanford';

-- Query using INNER JOIN
SELECT s.sID, s.sName, s.GPA, s.sizeHS, a.cName, a.major
FROM Student s
INNER JOIN Apply a ON s.sID = a.sID
JOIN College c ON a.cName = c.collegeName
WHERE s.sizeHS < 20000 AND s.sizeHS IS NOT NULL
AND a.major = 'CS' AND c.collegeName = 'Stanford';
```

Q15. Names and GPAs of students with HS>1000 who applied to CS and were rejected

```sql
-- Query using CROSS JOIN
SELECT s.sName, s.GPA
FROM Student s
CROSS JOIN Apply a
JOIN College c ON a.cName = c.collegeName
WHERE s.sizeHS > 1000 AND s.sizeHS IS NOT NULL
AND a.major = 'CS' AND a.decision = 'N';

-- Query using NATURAL JOIN
SELECT s.sName, s.GPA
FROM Student s
NATURAL JOIN Apply a
JOIN College c ON a.cName = c.collegeName
WHERE s.sizeHS > 1000 AND s.sizeHS IS NOT NULL
AND a.major = 'CS' AND a.decision = 'N';
```

```sql
-- Query using INNER JOIN
SELECT s.sName, s.GPA
FROM Student s
INNER JOIN Apply a ON s.sID = a.sID
JOIN College c ON a.cName = c.collegeName
WHERE s.sizeHS > 1000 AND s.sizeHS IS NOT NULL
AND a.major = 'CS' AND a.decision = 'N';
```

Q16. Names and GPAs of students with HS>1000 who applied to CS at college with enr>20,000 and were rejected.

```sql
-- Query using CROSS JOIN
SELECT s.sName, s.GPA
FROM Student s
CROSS JOIN Apply a
JOIN College c ON a.cName = c.collegeName
WHERE s.sizeHS > 1000 AND s.sizeHS IS NOT NULL
AND a.major = 'CS' AND a.decision = 'N'
AND c.enrollment > 20000;

-- Query using NATURAL JOIN
SELECT s.sName, s.GPA
FROM Student s
NATURAL JOIN Apply a
JOIN College c ON a.cName = c.collegeName
WHERE s.sizeHS > 1000 AND s.sizeHS IS NOT NULL
AND a.major = 'CS' AND a.decision = 'N'
AND c.enrollment > 20000;

-- Query using INNER JOIN
SELECT s.sName, s.GPA
FROM Student s
INNER JOIN Apply a ON s.sID = a.sID
JOIN College c ON a.cName = c.collegeName
WHERE s.sizeHS > 1000 AND s.sizeHS IS NOT NULL
AND a.major = 'CS' AND a.decision = 'N'
AND c.enrollment > 20000;
```

- Suubmitted by:: Hardik Arora
- Branch : Btech - CS

- Program: AIML
- University Roll No. :   2215500071
- Section: 2AC
- Class Roll No. :   28
- Submitted to: Ayushi Mam