

PRACTICAL ASSIGNMENT – 4

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

```
CREATE DATABASE p4;
USE p4;

CREATE TABLE IF NOT EXISTS Student (
    sID INT PRIMARY KEY,
    sName VARCHAR(50),
    GPA FLOAT,
    sizeHS INT NOT NULL
);

INSERT INTO student(sID, sName, GPA, sizeHS) VALUES ('123', 'Amy', '3.9',
'1000');

INSERT INTO student(sID, sName, GPA, sizeHS) VALUES ('234', 'Bob', '3.6',
'1500');

INSERT INTO student(sID, sName, GPA, sizeHS) VALUES ('345', 'Craig', '3.5',
'500');

INSERT INTO student(sID, sName, GPA, sizeHS) VALUES ('456', 'Doris', '3.9',
'1000');

INSERT INTO student(sID, sName, GPA, sizeHS) VALUES ('567', 'Edward', '2.9',
'2000');

INSERT INTO student(sID, sName, GPA, sizeHS) VALUES ('678', 'Fay', '3.8', '200');

INSERT INTO student(sID, sName, GPA, sizeHS) VALUES ('789', 'Gary', '3.4',
'800');
```

```

INSERT INTO student(sID, sName, GPA, sizeHS) VALUES ('987', 'Helen', '3.7',
'800');

INSERT INTO student(sID, sName, GPA, sizeHS) VALUES ('876', 'Irene', '3.9',
'400');

INSERT INTO student(sID, sName, GPA, sizeHS) VALUES ('765', 'Jay', '2.9',
'1500');

INSERT INTO student (sID, sName, GPA, sizeHS) VALUES ('654', 'Amy', '3.9',
'1000');

INSERT INTO student (sID, sName, GPA, sizeHS) VALUES ('543', 'Craig', '3.4',
'2000');

SELECT * FROM student;

```

sID	sName	GPA	sizeHS
abc Filter...	abc Filter...	abc Filter...	abc Filter...
123	Amy	3.9	1000
234	Bob	3.6	1500
345	Craig	3.5	500
456	Doris	3.9	1000
543	Craig	3.4	2000
567	Edward	2.9	2000
654	Amy	3.9	1000
678	Fay	3.8	200
765	Jay	2.9	1500
789	Gary	3.4	800
876	Irene	3.9	400
987	Helen	3.7	800

```

CREATE TABLE IF NOT EXISTS College(
    cName VARCHAR(50) PRIMARY KEY,
    State VARCHAR(50),
    enrollment INT NOT NULL
);

INSERT INTO college(cName, State, enrollment) VALUES('Stanford', 'CA', '15000');

INSERT INTO college(cName, State, enrollment) VALUES('Berkeley', 'CA', '36000');

INSERT INTO college(cName, State, enrollment) VALUES('MIT', 'MA', '10000');

INSERT INTO college(cName, State, enrollment) VALUES('Cornell', 'NY', '21000');

INSERT INTO college(cName, State, enrollment) VALUES('Harvard', 'MA', '50040');

SELECT * FROM college;

```

localmysql: SELECT * FROM co... ✕

cName	State	enrollment
abc Filter...	abc Filter...	abc Filter...
Berkeley	CA	36000
Cornell	NY	21000
Harvard	MA	50040
MIT	MA	10000
Stanford	CA	15000

```

CREATE TABLE IF NOT EXISTS Apply(
    sID INT PRIMARY KEY,
    cName VARCHAR(50) NOT NULL,
    major VARCHAR(50) NOT NULL,
    decision VARCHAR(1) NOT NULL
);

```

```
INSERT INTO apply(sID, cName, major, decision) VALUES('123', 'Stanford', 'CS', 'Y');

INSERT INTO apply(sID, cName, major, decision) VALUES('123', 'Stanford', 'EE', 'N');

INSERT INTO apply(sID, cName, major, decision) VALUES('123', 'Berkeley', 'CS', 'Y');

INSERT INTO apply(sID, cName, major, decision) VALUES('123', 'Cornell', 'EE', 'Y');

INSERT INTO apply(sID, cName, major, decision) VALUES('234', 'Berkeley', 'biology', 'N');

INSERT INTO apply(sID, cName, major, decision) VALUES('345', 'MIT', 'bioengineering', 'Y');

INSERT INTO apply(sID, cName, major, decision) VALUES('345', 'Cornell', 'bioengineering', 'N');

INSERT INTO apply(sID, cName, major, decision) VALUES('345', 'Cornell', 'CS', 'Y');

INSERT INTO apply(sID, cName, major, decision) VALUES('345', 'Cornell', 'EE', 'N');

INSERT INTO apply(sID, cName, major, decision) VALUES('678', 'Stanford', 'history', 'Y');

INSERT INTO apply(sID, cName, major, decision) VALUES('987', 'Stanford', 'CS', 'Y');

INSERT INTO apply(sID, cName, major, decision) VALUES('987', 'Berkeley', 'CS', 'Y');

INSERT INTO apply(sID, cName, major, decision) VALUES('876', 'Stanford', 'CS', 'N');

INSERT INTO apply(sID, cName, major, decision) VALUES('876', 'MIT', 'biology', 'Y');

INSERT INTO apply(sID, cName, major, decision) VALUES('876', 'MIT', 'marine biology', 'N');
```

```
INSERT INTO apply(sID, cName, major, decision) VALUES('765', 'Stanford',  
'history', 'Y');  
  
INSERT INTO apply(sID, cName, major, decision) VALUES('765', 'Stanford',  
'history', 'N');  
  
INSERT INTO apply(sID, cName, major, decision) VALUES('765', 'Cornell',  
'history', 'N');  
  
INSERT INTO apply(sID, cName, major, decision) VALUES('765', 'Cornell',  
'psychology', 'Y');  
  
INSERT INTO apply(sID, cName, major, decision) VALUES('543', 'MIT', 'CS', 'N');
```

Q. Solve the following :

Q1. Count the total number of Students.

```
SELECT COUNT(*) AS total_students FROM Student;
```

Q2. Calculate the average GPA of all Student.

```
SELECT AVG(GPA) AS average_gpa FROM Student;
```

Q3. Determine the minimum and maximum GPA. Rename the titles as 'max_GPA' and 'min_GPA' respectively.

```
SELECT MAX(GPA) AS max_GPA, MIN(GPA) AS min_GPA FROM Student;
```

Q4. Count the number of students having GPA greater than or equal to 3.7.

```
SELECT COUNT(*) AS students_above_3_7 FROM Student WHERE GPA >= 3.7;
```

Q5. Find Maximum, Average, Minimum, total GPA of all student.

```
SELECT MAX(GPA) AS max_gpa, AVG(GPA) AS avg_gpa, MIN(GPA) AS min_gpa, SUM(GPA) AS  
total_gpa FROM Student;
```

Q6. Find total number of colleges in our Application Database.

```
SELECT COUNT(*) AS total_colleges FROM College;
```

Q7. Find how many different majors student had applied in.

```
SELECT COUNT(DISTINCT major) AS different_majors FROM Apply;
```

Q8. Find total no. of Applications in our Application System's Database.

```
SELECT COUNT(*) AS total_applications FROM Apply;
```

Q9. Find average of all distinct GPA.

```
SELECT AVG(DISTINCT GPA) AS avg_distinct_gpa FROM Student;
```

Q10. Display the total number of application accepted.

```
SELECT COUNT(*) AS applications_accepted FROM Apply WHERE decision = 'Y';
```

Q11. Find number of students having GPA>3.4 and coming from high school having size>1000.

```
SELECT COUNT(*) AS students_criteria FROM Student WHERE GPA > 3.4 AND sizeHS > 1000;
```

Q12. Find how many student applied to 'marine biology'.

```
SELECT COUNT(DISTINCT sID) AS students_marine_biology FROM Apply WHERE major = 'marine biology';
```

Q13. Find how many applications were rejected and accepted by the colleges.

```
SELECT  
    SUM(CASE WHEN decision = 'Y' THEN 1 ELSE 0 END) AS accepted,  
    SUM(CASE WHEN decision = 'N' THEN 1 ELSE 0 END) AS rejected  
FROM Apply;
```

Q14. Find how many students applied to a particular major. (show count(sid) as No_of_applications).

```
SELECT major, COUNT(sID) AS No_of_applications FROM Apply GROUP BY major;
```

Q15. Find number of applications received by particular college.

```
SELECT cName, COUNT(*) AS applications_received FROM Apply GROUP BY cName;
```

Q16. Find number of applications received in a particular major at a particular college.

```
SELECT cName, major, COUNT(*) AS applications_received FROM Apply GROUP BY cName, major;
```

Q17. Give the college name and major, where number of applications received are greater than or equal to 2.

```
SELECT cName, major FROM Apply GROUP BY cName, major HAVING COUNT(*) >= 2;
```

Q18. Give the name and no of applications of all those colleges which receives applications from 3 or more students.

```
SELECT cName, COUNT(DISTINCT sID) AS no_of_applications  
FROM Apply  
GROUP BY cName  
HAVING COUNT(DISTINCT sID) >= 3;
```

Q19. Give state and number of colleges of a state that has more than 1 college.

```
SELECT state, COUNT(*) AS no_of_colleges  
FROM College  
GROUP BY state  
HAVING COUNT(*) > 1;
```

Q20. Find the name of students that are duplicate.

```
SELECT sName  
FROM Student  
GROUP BY sName  
HAVING COUNT(*) > 1;
```

Q21. Find how many applications are filed by each student. [Hint: use left join as we need information about all 12 students here. If they applied nowhere than show zero in front of them]

```
SELECT s.sID, s.sName, COALESCE(COUNT(a.sID), 0) AS no_of_applications  
FROM Student s  
LEFT JOIN Apply a ON s.sID = a.sID
```

```
GROUP BY s.sID;
```

Q22. Provide name of students that file 3 or more applications.

```
SELECT s.sName
FROM Student s
JOIN Apply a ON s.sID = a.sID
GROUP BY s.sID
HAVING COUNT(a.sID) >= 3;
```

Q23. Provide name of student who have not applied to any college.

```
SELECT sName
FROM Student
WHERE sID NOT IN (SELECT DISTINCT sID FROM Apply);
```

Q24. Find maximum GPA, Average GPA, and minimum GPA among applicants of each college. (i.e. say sID 123, 324 and 987 had applied to Berkley then compute and display max GPA among these three)

```
SELECT a.cName, MAX(s.GPA) AS max_GPA, AVG(s.GPA) AS avg_GPA, MIN(s.GPA) AS
min_GPA
FROM Apply a
JOIN Student s ON a.sID = s.sID
GROUP BY a.cName;
```

Q25. Find how many student have same GPA among all students. (provide this frequency in two column table as GPA 3.9 is 4 times, GPA 2.9 is 2 times)

```
SELECT GPA, COUNT(*) AS frequency
FROM Student
GROUP BY GPA;
```

Q26. Find how many application of each major are rejected and accepted.

```
SELECT major, decision, COUNT(*) AS no_of_applications
FROM Apply
GROUP BY major, decision;
```

Q27. Find out the acceptance rate for each college. (Acceptance Rate is percentage of number application accepted w. r. t. number of application received)


```
SELECT cName,  
       ROUND((COUNT(CASE WHEN decision = 'Y' THEN 1 END) / CAST(COUNT(*) AS  
FLOAT)) * 100, 2) AS acceptance_rate  
FROM Apply  
GROUP BY cName;
```

-
- Name: Hardik Arora
 - Branch : Btech - CS
 - Program: AIML
 - University Roll No. : 2215500071
 - Section: 2AC
 - Class Roll No. : 28
 - Submitted To : Ayushi Mam
 - Submission Date : 10th April, 2024.