PRACTICAL ASSIGNMENT - 5

- 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');
```

```sql
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;
```

localmysql: SELECT * FROM st...

sID	sName	GPA	sizeHS
Filter...	Filter...	Filter...	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

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

```sql
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
Filter...	Filter...	Filter...
Berkeley	CA	36000
Cornell	NY	21000
Harvard	MA	50040
MIT	MA	10000
Stanford	CA	15000

```sql
CREATE TABLE IF NOT EXISTS Applied(

    sID INT NOT NULL,
    cName VARCHAR(50) NOT NULL,
    major VARCHAR(50) NOT NULL,
    decision VARCHAR(1) NOT NULL
);
```

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

```sql
INSERT INTO Applied(sID, cName, major, decision) VALUES('765', 'Stanford',
'history', 'Y');

INSERT INTO applied(sID, cName, major, decision) VALUES('765', 'Stanford',
'history', 'N');

INSERT INTO applied(sID, cName, major, decision) VALUES('765', 'Cornell',
'history', 'N');

INSERT INTO applied(sID, cName, major, decision) VALUES('765', 'Cornell',
'psychology', 'Y');

INSERT INTO applied(sID, cName, major, decision) VALUES('543', 'MIT', 'CS', 'N');

SELECT * FROM applied;
```

localmysql: SELECT * FROM ap...

sID	cName	major	decision
Filter...	Filter...	Filter...	Filter...
123	Stanford	CS	Y
123	Stanford	EE	N
123	Berkeley	CS	Y
123	Cornell	EE	Y
234	Berkeley	biology	N
345	MIT	bioengineering	Y
345	Cornell	bioengineering	N
345	Cornell	CS	Y
345	Cornell	EE	N
678	Stanford	history	Y
987	Stanford	CS	Y
987	Berkeley	CS	Y
876	Stanford	CS	N
876	MIT	biology	Y
876	MIT	marine biology	N
765	Stanford	history	Y
765	Stanford	history	N
765	Cornell	history	N
765	Cornell	psychology	Y

Write SQL queries for the following:

Q1. IDs and names of students who have applied to major in CS at some college.

```
SELECT DISTINCT s.sID, s.sName
FROM Student s
JOIN Applied a ON s.sID = a.sID
WHERE a.major = 'CS';
```

Q2. Find ID and name of student having same high school size as Jay.

```
SELECT sID, sName
FROM Student
WHERE sizeHS = (SELECT sizeHS FROM Student WHERE sID = '765') AND sID != '765';
```

Q3. Find ID and name of student having same high school size as Jay but result should not include Jay.

```
SELECT sID, sName
FROM Student
WHERE sizeHS = (SELECT sizeHS FROM Student WHERE sID = '765') AND sID != '765';
```

Q4. Find the name of student with their GPA and Sid whose GPA not equal to GPA of Irene?

```
SELECT sID, sName, GPA
FROM Student
WHERE GPA != (SELECT GPA FROM Student WHERE sName = 'Irene');
```

Q5. Find college where any student having their name started from J have applied?

```
SELECT DISTINCT cName
FROM Applied
WHERE sID IN (SELECT sID FROM Student WHERE sName LIKE 'J%');
```

Q6. Find all different major where Irene has applied?

```
SELECT DISTINCT major
FROM Applied
WHERE sID = (SELECT sID FROM Student WHERE sName = 'Irene');
```

Q7. Find IDs of student and major who applied in any of major Irene had applied to?

```
SELECT DISTINCT a.sID, a.major
FROM Applied a
WHERE a.major IN (SELECT DISTINCT major FROM Applied WHERE sID = (SELECT sID FROM
Student WHERE sName = 'Irene'));
```

Q8. Find IDs of student and major who applied in any of major Irene had applied to? But this time exclude Irene sID from the list.

```
SELECT DISTINCT a.sID, a.major
FROM Applied a
WHERE a.major IN (SELECT DISTINCT major FROM Applied WHERE sID = (SELECT sID FROM
Student WHERE sName = 'Irene'))
AND a.sID != (SELECT sID FROM Student WHERE sName = 'Irene');
```

Q9. Give the number of colleges Jay applied to? (Remember count each college once no matter if he applied to same college twice with different major)

```
SELECT COUNT(DISTINCT cName) AS NumColleges
FROM Applied
WHERE sID = '765';
```

Q10. Find sID of student who applied to more or same number of college where Jay has applied?

```sql
SELECT sID
FROM Applied
GROUP BY sID
HAVING COUNT(DISTINCT cName) >= (SELECT COUNT(DISTINCT cName) FROM Applied WHERE
sID = '765');
```

Q11. Find details of Students who applied to major CS but not applied to major EE? (sID 987, 876, 543 should only be include in result)

```sql
SELECT sID, sName, GPA, sizeHS
FROM Student
WHERE sID IN (
    SELECT DISTINCT a.sID
    FROM Applied a
    WHERE a.major = 'CS'
)
AND sID NOT IN (
    SELECT DISTINCT a.sID
    FROM Applied a
    WHERE a.major = 'EE'
)
AND sID IN ('987', '876', '543');
```

Q12. All colleges such that some other college is in same state. (Cornell should not be part of result as no other college in New York Hint: use exists)

```sql
SELECT cName, State
FROM College c1
WHERE EXISTS (
    SELECT 1
    FROM College c2
    WHERE c1.State = c2.State
    AND c1.cName != c2.cName
)
AND cName != 'Cornell';
```

Q13. Find the college with highest enrollment.

```sql
SELECT cName, enrollment
FROM College
WHERE enrollment = (SELECT MAX(enrollment) FROM College);
```

Q14. Find name of student having lowest GPA.

```sql
SELECT sName
FROM Student
WHERE GPA = (SELECT MIN(GPA) FROM Student);
```

Q15. Find the most popular major.

```sql
SELECT major, COUNT(*) AS popularity
```

```
FROM Applied
GROUP BY major
ORDER BY popularity DESC
LIMIT 1;
```

Q16. Find sID, sName, sizeHS of all students NOT from smallest HS

```
SELECT sID, sName, sizeHS
FROM Student
WHERE sizeHS != (SELECT MIN(sizeHS) FROM Student);
```

Q17. Find the name of student who applies to all the colleges where sID 987 has applied? (Hint: see Query Find IDs of student applied to all colleges)

```
insert into Applied
select s1.sID, 'Berkeley', 'CSE', 'Y'
from Student s1
where s1.sID IN (select s.sID from Student s
MINUS
select a.sID from Applied a where a.cName = 'Berkeley');
```

Q18. . Find sid of student who have not applied to Stanford.

```
SELECT sID
FROM Student
EXCEPT
SELECT sID
FROM Applied
WHERE cName = 'Stanford';
```

Q19. Find sid of Student that applied to both Stanford and Berkeley.

```
SELECT sID
FROM Applied
WHERE cName = 'Stanford'
INTERSECT
SELECT sID
FROM Applied
WHERE cName = 'Berkeley';
```

Q20. Give list of all names including all names of colleges and students.

```
SELECT sName AS Name
FROM Student
UNION
SELECT cName AS Name
FROM College;
```

Q21. Create a table ApplicationInfo having columns sID: int, sName: varchar2(10) and number_of_applications: number(2) they filed? Populate this table with appropriate data using insert command. (Remember to include details of ALL students that have applied or not applied)

```sql
-- Create the ApplicationInfo table
CREATE TABLE ApplicationInfo (
    sID INT,
    sName VARCHAR(10),
    number_of_applications INT
);

-- Populate the ApplicationInfo table with data
INSERT INTO ApplicationInfo (sID, sName, number_of_applications)
SELECT s.sID, s.sName, COUNT(a.sID) AS number_of_applications
FROM Student s
LEFT JOIN Applied a ON s.sID = a.sID
GROUP BY s.sID, s.sName;

-- Insert data for students who have not applied
INSERT INTO ApplicationInfo (sID, sName, number_of_applications)
SELECT sID, sName, 0 AS number_of_applications
FROM Student
WHERE sID NOT IN (SELECT sID FROM Applied);
```

Q22. Create table ApplicationData and load with ID, name of college where they applied with state of college on runtime using single query.

```sql
-- Create the ApplicationData table
CREATE TABLE ApplicationData (
    sID INT,
    cName VARCHAR(50),
    State VARCHAR(50)
);

-- Populate the ApplicationData table with data
INSERT INTO ApplicationData (sID, cName, State)
SELECT a.sID, a.cName, c.State
FROM Applied a
JOIN College c ON a.cName = c.cName;
```

Q23. Stanford decide not to take any student who have also applied to its rival Berkeley turn their application decision to N. (Hint: Turn decision to N only if any student applied to both Stanford and Berkeley, Update decision of only Stanford Application)

```sql
UPDATE Applied
SET decision = 'N'
WHERE cName = 'Stanford'
AND sID IN (
    SELECT sID
    FROM Applied
    WHERE cName = 'Stanford'
```

```
    INTERSECT
    SELECT sID
    FROM Applied
    WHERE cName = 'Berkeley'
);
```

Q24. Delete applications that are filed Colleges situated at city ‘New York’.

```
DELETE FROM Applied
WHERE cName IN (
    SELECT cName
    FROM College
    WHERE City = 'New York'
);
```

- Name: Hardik Arora
- Branch : Btech - CS
- Program: AIML
- University Roll No. : 2215500071
- Section: 2AC
- Class Roll No. : 28
- Submitted to: Ayushi Mam
- Submissipn date: 05/05/2024