

NAME – DUSHYANT SINGH RATHORE

ROLL NO – 2232133

COURSE – B SC MATHEMATICS (HONOURS)

GE - DATABASE MANAGEMENT SYSTEMS

ASSIGNMENT -2

SUBMITTED TO-MAAM SHRUTI SHIKHA

Ques2) Create and use the following student-course database schema for a college to answer the given queries using the standalone SQL editor.

Here, Roll No (ADMISSION) and CID (ADMISSION) are foreign keys. Note that course type may have two values viz. Fulltime and Parttime and a student may enrol in any number of courses.

ANSWER

```
mysql> SELECT*FROM STUDENT;
```

Roll_No	Student_Name	Course_ID	DOB
S001	Jahanvi	C001	1998-05-15
S002	Lakshita	C002	1999-02-28
S003	Arhaan	C003	2000-08-10
S004	Bilal	C004	1997-11-20
S005	Esha	C004	1996-04-05
S006	PRIYANSHI	C001	1999-09-12
S007	Shrishti	C004	1998-07-18
S008	Mansi	C006	2001-01-22
S009	Karan	C005	1997-03-08
S010	Dushyant	C009	2002-06-30
S011	Garvit	C008	1995-10-14
S012	Jatin	C005	2003-04-25
S013	Chandrika	C006	1998-12-03
S014	Elvis	C007	1996-08-28
S015	Arhaan	C001	2000-11-17
S016	Liza	C008	1999-01-07
S017	Mia	C002	1997-06-19
S018	Neil	C005	2001-09-09
S019	Ismail	C002	1994-12-11
S020	Alex	C001	1995-03-26

```
20 rows in set (0.00 sec)
```

```
mysql> select * from admission;
```

ROLL_NO	CID	DATE_OF_ADMISSION
S001	C009	2022-01-15
S001	C006	2022-01-15
S001	C002	2022-01-15
S002	C002	2022-02-28
S002	C010	2022-02-28
S002	C007	2022-02-28
S002	C001	2022-02-28
S003	C005	2022-03-10
S004	C003	2022-04-20
S004	C009	2022-04-20
S004	C010	2022-04-20
S004	C007	2022-04-20
S004	C002	2022-04-20
S005	C006	2022-09-08
S005	C009	2022-09-08
S006	C006	2022-01-17
S007	C002	2022-04-15
S008	C002	2022-09-28
S008	C010	2022-09-28
S008	C007	2022-09-28
S009	C001	2022-02-20
S010	C005	2022-03-20
S011	C009	2022-11-15
S011	C006	2022-11-15
S011	C002	2022-11-15
S012	C002	2022-09-28
S012	C010	2022-09-28
S012	C007	2022-09-28
S012	C001	2022-09-28
S013	C005	2022-07-15
S014	C003	2022-04-20
S014	C009	2022-05-20
S014	C010	2022-05-20
S014	C007	2022-05-20
S014	C002	2022-05-20
S015	C006	2022-09-08
S015	C009	2022-09-08
S016	C006	2022-08-17
S017	C002	2022-07-15
S018	C002	2022-01-28
S018	C010	2022-01-28
S018	C007	2022-01-28
S019	C001	2022-01-20
S020	C005	2022-07-20
S020	C003	2022-07-20

```
45 rows in set (0.00 sec)
```

```
mysql> select * from course;
```

CID	COURSE_NAME	course_type	TEACHER_IN_CHARGE	TOTAL_SEATS	DURATION
C001	Philosophy	Part-Time	Pradeep	24	45
C002	Physics	Full-Time	Shridhar	125	50
C003	History	Part-Time	Preeti	35	55
C004	Computer Science	Part-Time	Ashmit	40	75
C005	English	Part-Time	Usmaan	35	40
C006	Mathematics	Full-Time	Roger	150	60
C007	Physics	Full-Time	Harish	126	45
C008	Geography	Full-Time	Farhaan	18	35
C009	Science	Full-Time	Adhiraj	68	25
C010	English	Part-Time	Sarah	135	90

```
10 rows in set (0.00 sec)
```

Questions

1. Retrieve names of students enrolled in any course.

```
mysql> select Student_Name from student  
      -> where course_id is not null;
```

Student_Name
Jahanvi
Lakshita
Arhaan
Bilal
Esha
Danish
Shrishti
Mansi
Karan
Dushyant
Garvit
Jatin
Chandrika
Elvis
Arhaan
Liza
Mia
Neil
Ismail
Alex

```
20 rows in set (0.00 sec)
```

2. Retrieve names of students enrolled in at least one part time course.

```
mysql> select distinct student.student_name from student
-> join admission on student.roll_no=admission.roll_no
-> join course on admission.CID=course.CID
-> where course_type="Part-Time";
+-----+
| student_name |
+-----+
| Lakshita     |
| Arhaan       |
| Bilal        |
| Mansi        |
| Karan        |
| Dushyant     |
| Jatin        |
| Chandrika    |
| Elvis        |
| Neil         |
| Ismail       |
| Alex         |
+-----+
12 rows in set (0.00 sec)
```

3. Retrieve students' names starting with letter 'A'.

```
mysql> Select Student_Name from Student where student_name LIKE 'A%';
+-----+
| Student_Name |
+-----+
| Arhaan       |
| Arhaan       |
| Alex         |
+-----+
3 rows in set (0.35 sec)
```

4. Retrieve students' details studying in courses 'computer science' or 'chemistry'.

```
mysql> SELECT student.*
-> FROM student
-> JOIN admission ON student.Roll_No = admission.Roll_No
-> JOIN course ON admission.cid = course.cid
-> WHERE course.course_name IN ('Computer Science', 'Chemistry');
+-----+-----+-----+-----+
| Roll_No | Student_Name | Course_ID | DOB      |
+-----+-----+-----+-----+
| S016    | Liza         | C008      | 1999-01-07 |
+-----+-----+-----+-----+
1 row in set (0.01 sec)
```

5. Retrieve students' names whose roll no either starts with 'S' and ends with '9'

```
mysql> SELECT student_name
-> FROM student
-> WHERE Roll_No LIKE 'S%9';
+-----+
| student_name |
+-----+
| Karan        |
| Ismail       |
+-----+
2 rows in set (0.00 sec)
```

6. Find course details with more than N students enrolled where N is to be input by the user.

```
mysql> SET @N=4;
Query OK, 0 rows affected (0.00 sec)

mysql> SELECT course.CID, course.COURSE_NAME, COUNT(admission.Roll_No) AS No_of_students_enrolled
-> FROM course
-> JOIN admission ON course.cid = admission.cid
-> GROUP BY course.cid, course.COURSE_NAME
-> HAVING No_of_students_enrolled > @N;
+-----+-----+-----+
| CID | COURSE_NAME | No_of_students_enrolled |
+-----+-----+-----+
| C009 | Science | 6 |
| C006 | Mathematics | 5 |
| C002 | Physics | 10 |
| C010 | English | 6 |
| C007 | Physics | 6 |
+-----+-----+-----+
5 rows in set (0.00 sec)
```

7. Update student table for modifying a student name.

```
mysql> UPDATE STUDENT
-> SET STUDENT_NAME="PRIYANSHI"
-> WHERE ROLL_NO='S006';
Query OK, 1 row affected (0.52 sec)
Rows matched: 1 Changed: 1 Warnings: 0
```

8. Find course names in which more than five students have enrolled

```
mysql> SELECT course.CID, course.COURSE_NAME, COUNT(admission.Roll_No) AS No_of_students_enrolled
-> FROM course
-> JOIN admission ON course.cid = admission.cid
-> GROUP BY course.cid, course.COURSE_NAME
-> HAVING No_of_students_enrolled > 5;
```

CID	COURSE_NAME	No_of_students_enrolled
C009	Science	6
C002	Physics	10
C010	English	6
C007	Physics	6

```
4 rows in set (0.00 sec)
```

9. Find the name of youngest student enrolled in course 'English'

```
mysql> SELECT student_name FROM student
-> JOIN admission ON student.roll_no = admission.roll_no
-> JOIN course ON admission.cid = course.cid
-> WHERE course.course_name = 'English'
-> ORDER BY student.DOB DESC
-> LIMIT 1;
```

student_name
Jatin

```
1 row in set (0.00 sec)
```


10. Find the name of most popular Subject (on the basis of enrolled students).

```
mysql> SELECT course.CID, course.COURSE_NAME, COUNT(admission.Roll_No) AS No_of_students_enrolled
-> FROM course
-> JOIN admission ON course.cid = admission.cid
-> GROUP BY course.cid, course.COURSE_NAME
-> ORDER BY No_of_students_enrolled DESC
-> LIMIT 1;
+-----+-----+-----+
| CID | COURSE_NAME | No_of_students_enrolled |
+-----+-----+-----+
| C002 | Physics | 10 |
+-----+-----+-----+
1 row in set (0.00 sec)
```

11. Find the name of two popular part time courses (on the basis of enrolled students)

```
mysql> SELECT course.CID, course.COURSE_NAME, COUNT(admission.Roll_No) AS No_of_students_enrolled
-> FROM course
-> JOIN admission ON course.cid = admission.cid
-> WHERE course.course_type = 'part-time'
-> GROUP BY course.cid, course.COURSE_NAME
-> ORDER BY No_of_students_enrolled DESC
-> LIMIT 1;
+-----+-----+-----+
| CID | COURSE_NAME | No_of_students_enrolled |
+-----+-----+-----+
| C010 | English | 6 |
+-----+-----+-----+
1 row in set (0.00 sec)
```

12. Find the student names who are admitted to full time courses only.

```
mysql> SELECT distinct(student.student_name)
-> FROM student
-> JOIN admission ON student.Roll_No = admission.Roll_No
-> JOIN course ON admission.cid = course.cid
-> WHERE course.course_type = 'full-time';
+-----+
| student_name |
+-----+
| Jahanvi      |
| Lakshita     |
| Bilal        |
| Esha         |
| PRIYANSHI    |
| Shrishti     |
| Mansi        |
| Garvit       |
| Jatin        |
| Elvis        |
| Arhaan       |
| Mia          |
| Neil         |
+-----+
13 rows in set (0.00 sec)
```

13. Find course names in which total seats are more than 30.

```
mysql> select course_name from course where total_seats>30;
+-----+
| course_name |
+-----+
| Physics     |
| History     |
| Computer Science |
| English     |
| Mathematics |
| Physics     |
| Science     |
| English     |
+-----+
8 rows in set (0.00 sec)
```

15. Find course names where teacher-in-charge has 'Roger' as name and the course is full time.

```
mysql> select course_name from course where teacher_in_charge='Roger' and course_type='full-time';
+-----+
| course_name |
+-----+
| Mathematics |
+-----+
1 row in set (0.00 sec)
```

17. Display the vacant seats for each course

```
mysql> SELECT course.cid,course.COURSE_NAME,course.total_seats - COUNT(distinct(admission.roll_no)) AS vacant_seats
-> FROM course JOIN
-> admission ON course.cid = admission.cid
-> GROUP BY
-> course.cid, course.COURSE_NAME, course.total_seats;
```

cid	COURSE_NAME	vacant_seats
C001	Philosophy	25
C002	Physics	140
C003	History	39
C004	Computer Science	47
C005	English	38
C006	Mathematics	175
C007	Physics	145
C009	Science	76
C010	English	156

```
9 rows in set (0.03 sec)
```

18. Increment Total Seats of each course by 20%

```
mysql> Update Course
-> Set Total_Seats = Total_Seats + ROUND(0.2*Total_Seats);
Query OK, 10 rows affected (0.52 sec)
Rows matched: 10  Changed: 10  Warnings: 0

mysql> select * from course;
```

CID	COURSE_NAME	course_type	TEACHER_IN_CHARGE	TOTAL_SEATS	DURATION
C001	Philosophy	Part-Time	Pradeep	29	45
C002	Physics	Full-Time	Shridhar	150	50
C003	History	Part-Time	Preeti	42	55
C004	Computer Science	Part-Time	Ashmit	48	75
C005	English	Part-Time	Usmaan	42	40
C006	Mathematics	Full-Time	Roger	180	60
C007	Physics	Full-Time	Harish	151	45
C008	Geography	Full-Time	Farhaan	22	35
C009	Science	Full-Time	Adhiraj	82	25
C010	English	Part-Time	Sarah	162	90

```
10 rows in set (0.00 sec)
```

19. Add enrolment fees paid ('yes'/'No') field in the admission table.

```
mysql> ALTER TABLE admission
-> ADD COLUMN fees_paid VARCHAR(3);
Query OK, 0 rows affected (0.79 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> UPDATE Admission
-> SET fees_paid = 'yes';
Query OK, 45 rows affected (0.46 sec)
Rows matched: 45  Changed: 45  Warnings: 0
```

20. Update the date of admission for all the courses by 1 year.

```
mysql> UPDATE admission
      -> SET date_of_admission = DATE_ADD(date_of_admission, INTERVAL 1 YEAR);
Query OK, 45 rows affected (0.44 sec)
Rows matched: 45  Changed: 45  Warnings: 0
```

21. Create a view to keep track of course names with total number of students enrolled in it.

```
mysql> CREATE VIEW CourseEnrollmentView AS
      -> SELECT
      ->     course.cid,
      ->     course.COURSE_NAME,
      ->     COUNT(admission.roll_no) AS Total_Students_Enrolled
      -> FROM
      ->     course
      -> JOIN
      ->     admission ON course.cid = admission.cid
      -> GROUP BY
      ->     course.cid, course.COURSE_NAME;
Query OK, 0 rows affected (0.47 sec)
```

```
mysql> select * from courseenrollmentview;
+-----+-----+-----+
| cid | COURSE_NAME | Total_Students_Enrolled |
+-----+-----+-----+
| C009 | Science | 6 |
| C006 | Mathematics | 5 |
| C002 | Physics | 10 |
| C010 | English | 6 |
| C007 | Physics | 6 |
| C001 | Philosophy | 4 |
| C005 | English | 4 |
| C003 | History | 3 |
| C004 | Computer Science | 1 |
+-----+-----+-----+
9 rows in set (0.12 sec)
```

22. Count the number of courses with more than 5 students enrolled for each type of course.

```
mysql> SELECT course.course_type, COUNT(distinct(course.course_name)) AS Courses_Count
-> FROM course
-> JOIN admission ON course.cid = admission.cid
-> GROUP BY course.course_type
-> HAVING COUNT(admission.roll_no) > 5;
```

course_type	Courses_Count
Full-Time	3
Part-Time	4

2 rows in set (0.37 sec)

23. Add column Mobile number in student table with default value '9999999999'

```
mysql> ALTER TABLE student
-> ADD COLUMN Mobile_number VARCHAR(10) DEFAULT '9999999999';
Query OK, 0 rows affected (1.28 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> select * from student;
```

Roll_No	Student_Name	Course_ID	DOB	Mobile_number
S001	Jahanvi	C001	1998-05-15	9999999999
S002	Lakshita	C002	1999-02-28	9999999999
S003	Arhaan	C003	2000-08-10	9999999999
S004	Bilal	C004	1997-11-20	9999999999
S005	Esha	C004	1996-04-05	9999999999
S006	PRIYANSHI	C001	1999-09-12	9999999999
S007	Shrishti	C004	1998-07-18	9999999999
S008	Mansi	C006	2001-01-22	9999999999
S009	Karan	C005	1997-03-08	9999999999
S010	Dushyant	C009	2002-06-30	9999999999
S011	Garvit	C008	1995-10-14	9999999999
S012	Jatin	C005	2003-04-25	9999999999
S013	Chandrika	C006	1998-12-03	9999999999
S014	Elvis	C007	1996-08-28	9999999999
S015	Arhaan	C001	2000-11-17	9999999999
S016	Liza	C008	1999-01-07	9999999999
S017	Mia	C002	1997-06-19	9999999999
S018	Neil	C005	2001-09-09	9999999999
S019	Ismail	C002	1994-12-11	9999999999
S020	Alex	C001	1995-03-26	9999999999

20 rows in set (0.05 sec)

24. Find the total number of students whose age is > 18 years.

```
mysql> SELECT COUNT(Roll_No) AS Total_Students_Over_18
-> FROM student
-> WHERE TIMESTAMPDIFF(YEAR,DOB, CURDATE()) > 18;
+-----+
| Total_Students_Over_18 |
+-----+
|                20      |
+-----+
1 row in set (0.19 sec)
```

25. Find names of students who are born in 2001 and are admitted to at least one part time course.

```
mysql> SELECT student_name
-> FROM student
-> JOIN admission ON student.Roll_No = admission.Roll_No
-> JOIN course ON admission.cid = course.cid
-> WHERE YEAR(student.DOB) = 2001
-> AND course.course_type = 'part-time';
+-----+
| student_name |
+-----+
| Mansi        |
| Neil         |
+-----+
2 rows in set (0.42 sec)
```

26. Count all courses having 'science' in the name and starting with the word 'Computer'.

```
mysql> SELECT COUNT(*) AS Total_Courses
-> FROM course
-> WHERE course_name LIKE 'Computer%' AND course_name LIKE '%Science%';
+-----+
| Total_Courses |
+-----+
|              1 |
+-----+
1 row in set (0.38 sec)
```