ASSIGNMENT 7

mysql> create database college\_011; Query OK, 1 row affected (0.02 sec)

mysql> use college\_011; Database changed **Creating Tables**

mysql> create table departments\_011 (

-> DepartmentID int primary key auto\_increment,

-> Deptame varchar(50) unique not null,

-> HOD varchar(50) not null

-> );

mysql> create table students\_011 (

-> StudentID int primary key auto\_increment,

-> ame varchar(50) not null,

-> Age int check (age >= 18),

-> DepartmentID int,

-> CGflA decimal(3,2) check (cgpa between 0 and 10),

-> foreign key (DepartmentID) references departments\_011(DepartmentID) on delete set null

-> );

mysql> create table courses\_011 (

-> CourseID varchar(10) primary key,

-> Courseame varchar(50) not null,

-> DepartmentID int,

-> foreign key (DepartmentID) references departments\_011(DepartmentID) on delete cascade

mysql> create table enrollments\_011 (

-> EnrollmentID int primary key auto\_increment,

-> StudentID int, mysql> create database college\_011; Query OK, 1 row affected (0.02 sec)

-> CourseID varchar(10),

-> Marks int check (marks between 0 and 100),

-> foreign key (StudentID) references students(StudentID),

-> foreign key (CourseID) references courses(CourseID),

-> unique (StudentID, CourseID)

-> );

**-> Inserted Data in Tables mysql>** select \* from courses\_011 ;

+ + + +

| CourseID | Courseame | DepartmentID |

+ + + +

| AI501 | Machine Learning | 5 |

| AI502 | Deep Learning | 5 |

| CE401 | Structural Analysis | 4 |

| CE402 | Transportation Engineering | 4 |

| CS101 | Data Structures and Algorithms | 1 |

| CS102 | Database Management Systems | 1 |

| ET301 | Digital Signal flrocessing | 3 |

| ET302 | Embedded Systems | 3 |

| ME201 | Thermodynamics | 2 |

| ME202 | Fluid Mechanics | 2 |

+ + + +

10 rows in set (0.02 sec)

mysql> select \* from departments\_011;

+ + +

-+

| DepartmentID | Deptame | HOD

|

+ + +

-+

| 1 | Computer Engineering | Dr. S. S. Limaye

|

| 2 | Mechanical Engineering | Dr. R. B. flatil

|

| 3 | Electronics fi Telecommunication | Dr. A. fl. Kulkarni

|

| 4 | Civil Engineering | Dr. M. G. Joshi

|

| 5 | Artificial Intelligence fi Data Science | Dr. . R. Deshpande

|

+ + +

-+

5 rows in set (0.02 sec)

mysql> select \* from enrollments\_011;

+ + + + +

| EnrollmentID | StudentID | CourseID | Marks |

+ + + + +

| 216 | 101 | CS101 | 85 |

| 217 | 102 | CS102 | 78 |

| 218 | 101 | CS102 | 82 |

| 219 | 102 | ME201 | 79 |

| 220 | 103 | ET301 | 88 |

| 221 | 115 | CS101 | 85 |

| 222 | 114 | CS101 | 39 |

+ + + + +

7 rows in set (0.02 sec)

mysql> select \* from students\_011;

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| + | + | + | + | + | + |
| | StudentID | | ame | | Age | | DepartmentID | | CGflA | | |
| + | + | + | + | + | + |
| | 101 | | flrasad Bange | | 20 | | 1 | | 8.20 | | |
| | 102 | | Aarush Borkar | | 19 | | 2 | | 7.50 | | |
| | 103 | | Kaiwalya Ladekar | | 21 | | 3 | | 9.10 | | |
| | 104 | | Shreya Bhagat | | 18 | | 4 | | 8.00 | | |
| | 105 | | Mihir flatil | | 22 | | 5 | | 8.90 | | |
| | 106 | | prathamesh bhangari | | 20 | | 1 | | 7.80 | | |

| 107 | Aniket Damedhar | 19 | 2 | 8.30 |

| 108 | Dhruv flujari | 21 | 3 | 9.00 |

| 109 | Anuj Ahire | 18 | 4 | 7.60 |

| 110 | flrathamesh Bhangari | 22 | 5 | 8.70 |

| 111 | Mihik Shah | 20 | 1 | 8.40 |

| 112 | flrathamesh Mane | 19 | 2 | 7.90 |

| 113 | Janvee Ghadge | 21 | 3 | 9.20 |

| 114 | Sarvesh Todkar | 18 | 4 | 7.70 |

| 115 | Atharv Desai | 22 | 5 | 8.80 |

| 116 | Vedant Banne | 20 | 1 | 7.80 |

+ + + + + +

16 rows in set (0.02 sec)

**SQL Queries:->**

# Find students who have a CGflA higher than the university average CGflA

**->** mysql> select \* from students\_011

-> where cgpa > (select avg(cgpa) from students\_011);

+ + + + + +

| StudentID | ame | Age | DepartmentID | CGflA |

+ + + + + +

| 103 | Kaiwalya Ladekar | 21 | 3 | 9.10 |

| 105 | Mihir flatil | 22 | 5 | 8.90 |

| 108 | Dhruv flujari | 21 | 3 | 9.00 |

| 110 | flrathamesh Bhangari | 22 | 5 | 8.70 |

| 111 | Mihik Shah | 20 | 1 | 8.40 |

| 113 | Janvee Ghadge | 21 | 3 | 9.20 |

| 115 | Atharv Desai | 22 | 5 | 8.80 |

+ + + + + +

7 rows in set (0.00 sec)

# Find students who have never enrolled in any course

**->**mysql> select \* from students\_011

-> where studentid not in (select distinct studentid from enrollments\_011);

+ + + + + +

| StudentID | ame | Age | DepartmentID | CGflA |

+ + + + + +

| 104 | Shreya Bhagat | 18 | 4 | 8.00 |

| 105 | Mihir flatil | 22 | 5 | 8.90 |

| 106 | prathamesh bhangari | 20 | 1 | 7.80 |

| 107 | Aniket Damedhar | 19 | 2 | 8.30 |

| 108 | Dhruv flujari | 21 | 3 | 9.00 |

| 109 | Anuj Ahire | 18 | 4 | 7.60 |

| 110 | flrathamesh Bhangari | 22 | 5 | 8.70 |

| 111 | Mihik Shah | 20 | 1 | 8.40 |

| 112 | flrathamesh Mane | 19 | 2 | 7.90 |

| 113 | Janvee Ghadge | 21 | 3 | 9.20 |

| 116 | Vedant Banne | 20 | 1 | 7.80 |

+ + + + + +

11 rows in set (0.00 sec)

# Get courses that have no students enrolled

**->**mysql> select \* from courses\_011

-> where courseid not in (select courseid from enrollments\_011);

+ + + +

| CourseID | Courseame | DepartmentID |

+ + + +

| AI501 | Machine Learning | 5 |

| AI502 | Deep Learning | 5 |

| CE401 | Structural Analysis | 4 |

| CE402 | Transportation Engineering | 4 |

| ET302 | Embedded Systems | 3 |

| ME202 | Fluid Mechanics | 2 |

+ + + +

6 rows in set (0.00 sec)

# Get students enrolled in courses offered by the 'Computer Sci' department

**->** mysql> select s.\* from students\_011 s

-> join enrollments\_011 e on s.studentid = e.studentid

-> join courses\_011 c on e.courseid = c.courseid

-> join departments\_011 d on c.departmentid = d.departmentid

-> where d.deptname like 'Computer%';

+ + + + + +

| StudentID | ame | Age | DepartmentID | CGflA |

+ + + + + +

| 101 | flrasad Bange | 20 | 1 | 8.20 |

| 115 | Atharv Desai | 22 | 5 | 8.80 |

| 114 | Sarvesh Todkar | 18 | 4 | 7.70 |

| 102 | Aarush Borkar | 19 | 2 | 7.50 |

| 101 | flrasad Bange | 20 | 1 | 8.20 |

+ + + + + +

5 rows in set (0.01 sec)

# Find students who have the same age as the oldest student

**->**mysql> select \* from students\_011 where age = (select max(age) from students\_011);

+ + + + + +

| StudentID | ame | Age | DepartmentID | CGflA |

+ + + + + +

| 105 | Mihir flatil | 22 | 5 | 8.90 |

| 110 | flrathamesh Bhangari | 22 | 5 | 8.70 |

| 115 | Atharv Desai | 22 | 5 | 8.80 |

+ + + + + +

3 rows in set (0.00 sec)

# Retrieve courses where at least one student scored more than 85 marks

**->**mysql> select distinct c.\* from courses\_011 c

-> join enrollments\_011 e on c.courseid = e.courseid

-> where e.marks >= 85;

+ + + +

| CourseID | Courseame | DepartmentID |

+ + + +

| CS101 | Data Structures and Algorithms | 1 |

| ET301 | Digital Signal flrocessing | 3 |

+ + + +

2 rows in set (0.00 sec)

# Find the highest CGflA in each department

**->**mysql> select departmentid, max(cgpa) as highest\_cgpa from students\_011 group by departmentid;

+ + +

| departmentid | highest\_cgpa |

+ + +

| 1 | 8.40 |

| 2 | 8.30 |

| 3 | 9.20 |

| 4 | 8.00 |

| 5 | 8.90 |

+ + +

5 rows in set (0.01 sec)

# Get students who have enrolled in more than one course

**->**mysql> select studentid from enrollments\_011 group by studentid having count(courseid) > 1;

+ +

| studentid |

+ +

| 101 |

| 102 |

+ +

2 rows in set (0.00 sec)

# Get the department with the most students enrolled

**->**mysql> select departmentid, count(\*) as student\_count

-> from students\_011 group by departmentid order by student\_count desc limit

1;

+ + +

| departmentid | student\_count |

+ + +

| 1 | 4 |

+ + +

1 row in set (0.00 sec)

# Find students who are not enrolled in any course

**->**mysql> select \* from students\_011 where studentid not in (select distinct studentid from enrollments\_011);

+ + + + + +

| StudentID | ame | Age | DepartmentID | CGflA |

+ + + + + +

| 104 | Shreya Bhagat | 18 | 4 | 8.00 |

| 105 | Mihir flatil | 22 | 5 | 8.90 |

| 106 | prathamesh bhangari | 20 | 1 | 7.80 |

| 107 | Aniket Damedhar | 19 | 2 | 8.30 |

| 108 | Dhruv flujari | 21 | 3 | 9.00 |

| 109 | Anuj Ahire | 18 | 4 | 7.60 |

| 110 | flrathamesh Bhangari | 22 | 5 | 8.70 |

| 111 | Mihik Shah | 20 | 1 | 8.40 |

| 112 | flrathamesh Mane | 19 | 2 | 7.90 |

| 113 | Janvee Ghadge | 21 | 3 | 9.20 |

| 116 | Vedant Banne | 20 | 1 | 7.80 |

+ + + + + +

11 rows in set (0.00 sec)

# Retrieve students who scored below the average marks in their courses

**->**mysql> select s.\* from students\_011 s

-> join enrollments\_011 e on s.studentid = e.studentid

-> where e.marks < (select avg(marks) from enrollments\_011 where courseid = e.courseid);

+ + + + + +

| StudentID | ame | Age | DepartmentID | CGflA |

+ + + + + +

| 102 | Aarush Borkar | 19 | 2 | 7.50 |

| 114 | Sarvesh Todkar | 18 | 4 | 7.70 |

+ + + + + +

2 rows in set (0.00 sec)

# Find the department where the highest CGflA student belongs

**->**mysql> select d.\* from departments\_011 d

-> join students\_011 s on d.departmentid = s.departmentid

-> where s.cgpa = (select max(cgpa) from students\_011);

+ + + +

| DepartmentID | Deptame | HOD |

+ + + +

| 3 | Electronics fi Telecommunication | Dr. A. fl. Kulkarni |

+ + + +

1 row in set (0.00 sec)

# Find the department with the highest number of students enrolled

**->**mysql> select departmentid, count(\*) as student\_count from students\_011

-> group by departmentid order by student\_count desc limit 1;

+ + +

| departmentid | student\_count |

+ + +

| 1 | 4 |

+ + +

1 row in set (0.00 sec)

# Get courses where the lowest score is higher than 70

**->** mysql> select courseid from enrollments\_011 group by courseid having min(marks) > 70;

+ +

| courseid |

+ +

| CS102 |

| ET301 |

| ME201 |

+ +

3 rows in set (0.00 sec)

# Retrieve students who have the same CGflA as 'Alice'

**->** mysql> select \* from students\_011 where cgpa = (select cgpa from students\_011 where name = 'prathmesh bhagari');

+ + + + + +

| StudentID | ame | Age | DepartmentID | CGflA |

+ + + + + +

| 106 |prathmesh bhagari | 20 | 1 | 7.80 |

| 116 | Vedant Banne | 20 | 1 | 7.80 |

+ + + + + +

2 rows in set (0.00 sec)

# Get the course where the maximum number of students have enrolled

**->** mysql> select courseid, count(studentid) as total\_enrollments\_011

-> from enrollments\_011 group by courseid order by total\_enrollments desc limit 1;

+ + +

| courseid | total\_enrollments |

+ + +

| CS101 | 3 |

# Find students who are enrolled in more than two courses

**->**mysql> select studentid from enrollments\_011 group by studentid having count(courseid) >=2;

+ +

| studentid |

+ +

| 101 |

| 102 |

+ +

2 rows in set (0.00 sec)

# Retrieve students who have scored the highest marks in any course(Hint: Use Join along with nested quesries).

**->** mysql> select s.\* from students\_011 s

-> join enrollments\_011 e on s.studentid = e.studentid

-> where (e.courseid, e.marks) in (select courseid, max(marks) from enrollments\_011 group by courseid);

+ + + + + +

| StudentID | ame | Age | DepartmentID | CGflA |

+ + + + + +

| 101 | flrasad Bange | 20 | 1 | 8.20 |

| 101 | flrasad Bange | 20 | 1 | 8.20 |

| 102 | Aarush Borkar | 19 | 2 | 7.50 |

| 103 | Kaiwalya Ladekar | 21 | 3 | 9.10 |

| 115 | Atharv Desai | 22 | 5 | 8.80 |

+ + + + + +

5 rows in set (0.00 sec)

# Get the department with the lowest average CGflA

**->**mysql> select departmentid, avg(cgpa) as avg\_cgpa from students\_011

-> group by departmentid order by avg\_cgpa limit 1;

+ + +

| departmentid | avg\_cgpa |

+ + +

| 4 | 7.766667 |

+ + +

1 row in set (0.00 sec)

# Find the course in which the maximum number of students have failed (Marks < 40)

**->**mysql> select courseid, count(\*) as fail\_count

-> from enrollments\_011 where marks < 40 group by courseid order by fail\_count desc limit 1;

+ + +

| courseid | fail\_count |

+ + +

| CS101 | 1 |

+ + +

1 row in set (0.00 sec)