

Practical no. 02

AIM:- Subquery-join operations on relational scheme

1. Find the names of students who took only four-credit courses.

```
mysql> SELECT DISTINCT s.name FROM students s WHERE s.stno IN ( SELECT DISTINCT  
g.stno FROM grade g JOIN courses c ON g.cno = c.cno WHERE c.cr = 4);
```

```
+-----+  
| name      |  
+-----+  
| Edwards P. David |  
| Mixon Leatha   |  
| Pierce Richard  |  
| Prior Lorraine  |  
| Rawlings Jerry  |  
| Lewis Jerry     |  
+-----+
```

2. Find the names of students who took no four-credit courses.

```
SELECT DISTINCT s.name FROM students s WHERE s.stno not IN ( SELECT DISTINCT g.stno  
FROM grade g JOIN courses c ON g.cno = c.cno WHERE c.cr = 4);
```

```
+-----+  
| name      |  
+-----+  
| Grogan A. Mary |  
| McLane Sandy   |  
| Novak Roland   |  
+-----+
```

3. Find the names of students who took cs210 or cs310.

```
SELECT DISTINCT s.name FROM students s WHERE s.stno IN ( SELECT DISTINCT g.stno FROM  
grade g JOIN courses c ON g.cno = c.cno WHERE c.cno = "cs210" or "cs310");
```

```
+-----+  
| name      |  
+-----+  
| Edwards P. David |  
| Mixon Leatha   |  
| Pierce Richard  |  
| Prior Lorraine  |  
| Lewis Jerry     |  
+-----+
```

4. Find the names of students who obtained the highest grade in cs210.

```
SELECT s.name FROM students s JOIN grade g ON s.stno = g.stno WHERE g.cno = 'cs210'
ORDER BY g.grade DESC LIMIT 1;
```

```
+-----+
| name      |
+-----+
| Edwards P. David |
+-----+
```

5. Find the highest grade of a student who never took cs110.

```
SELECT MAX(g.grade) AS highest_grade FROM students s JOIN grade g ON s.stno = g.stno
LEFT JOIN courses c ON g.cno = c.cno AND c.cno = 'cs110' WHERE c.cno IS NULL;
```

```
+-----+
| highest_grade |
+-----+
|          100 |
+-----+
```

6. Find the names of students who do not have an advisor.

```
SELECT s.name FROM students s LEFT JOIN advising a ON s.stno = a.stno WHERE a.stno IS
NULL;
```

Empty set (0.00 sec)

7. Find names of courses taken by students who do not live in Massachusetts (MA).

```
SELECT DISTINCT c.cname FROM courses c JOIN grade g ON c.cno = g.cno JOIN students s
ON g.stno = s.stno WHERE s.state <> 'MA';
```

```
+-----+
| cname      |
+-----+
| Computer Programming |
| Computer Architecture |
| Software Engineering |
+-----+
```

8. Find course numbers for courses that enrol at least two students; solve the same query for courses that enroll at least three students.

```
SELECT c.cno, COUNT(g.stno) AS student_count FROM courses c JOIN grade g ON c.cno =
g.cno GROUP BY c.cno HAVING student_count >= 2;
```

```
+-----+-----+
| cno | student_count |
+-----+-----+
```

cs110	6
cs210	5
cs240	6
cs310	2
cs410	2
+-----+-----+	

9. Find the names of students who have failed all their courses (failing is defined as a grade less than 60).

```
SELECT DISTINCT s.name FROM students s WHERE EXISTS ( SELECT 1 FROM grade g WHERE
s.stno = g.stno AND g.grade < 60);
```

+-----+-----+	
name	
+-----+-----+	
Edwards P. David	
Lewis Jerry	
+-----+-----+	

10. Find names of all students who have a cs210 grade higher than the highest grade given in cs310 and did not take any course with Prof. Evans.

```
SELECT DISTINCT s.name FROM students s JOIN grade g1 ON s.stno = g1.stno JOIN grade
g2 ON s.stno = g2.stno JOIN courses c1 ON g1.cno = c1.cno AND c1.cno = 'cs210' JOIN
courses c2 ON g2.cno = c2.cno AND c2.cno = 'cs310' JOIN instructors i ON g1.empno =
i.empno WHERE g1.grade > (SELECT MAX(g2.grade) FROM grade g2 WHERE g2.cno =
'cs310') AND NOT EXISTS (SELECT 1 FROM grade g3 JOIN instructors i2 ON g3.empno =
i2.empno WHERE g3.stno = s.stno AND i2.name = 'Evans Robert' );
```

Empty set (0.00 sec)