Assignment-7

**Submitted by: Tanmay Vig**

**Roll Number: 19BCS061**

**Batch: 2nd year CSE**

**Tables:**

mysql> CREATE TABLE student(

-> snum INT,

-> sname CHAR(10),

-> major CHAR(10),

-> level CHAR(2),

-> age INT);

Query OK, 0 rows affected (0.13 sec)

mysql> INSERT INTO student

-> VALUES

-> (101,'Jhon','CS','SR',19),

-> (102,'Smith','CS','JR',20),

-> (103,'Jacob','ECE','SR',20),

-> (104,'Tom','CS','JR',20),

-> (105,'Sid','CS','JR',20),

-> (106,'Harry','History','SR',21),

-> (107,'Hellen','CS','JR',21),

-> (108,'Bob','English','SR',22),

-> (109,'Andy','ECE','JR',21),

-> (110,'Charles','History','SR',23);

Query OK, 10 rows affected (0.01 sec)

Records: 10 Duplicates: 0 Warnings: 0

mysql> SELECT \* FROM student;

+------+---------+---------+-------+------+

| snum | sname | major | level | age |

+------+---------+---------+-------+------+

| 101 | Jhon | CS | SR | 19 |

| 102 | Smith | CS | JR | 20 |

| 103 | Jacob | ECE | SR | 20 |

| 104 | Tom | CS | JR | 20 |

| 105 | Sid | CS | JR | 20 |

| 106 | Harry | History | SR | 21 |

| 107 | Hellen | CS | JR | 21 |

| 108 | Bob | English | SR | 22 |

| 109 | Andy | ECE | JR | 21 |

| 110 | Charles | History | SR | 23 |

+------+---------+---------+-------+------+

10 rows in set (0.01 sec)

mysql> CREATE TABLE class(

-> cname VARCHAR(6),

-> meets\_at CHAR(10),

-> room VARCHAR(4),

-> fid INT);

Query OK, 0 rows affected (0.05 sec)

mysql> INSERT INTO class

-> VALUES

-> ('CSC342','Morning','R128',201),

-> ('CSC343','Noon','R128',203),

-> ('CSC345','Night','R154',204),

-> ('ECE300','Morning','R111',202),

-> ('ECE301','Noon','R111',203),

-> ('ENG366','Morning','R154',203),

-> ('ENG367','Evening','R111',205),

-> ('HIS320','Evening','R128',205);

Query OK, 8 rows affected (0.01 sec)

Records: 8 Duplicates: 0 Warnings: 0

mysql> SELECT \* FROM class;

+--------+----------+------+------+

| cname | meets\_at | room | fid |

+--------+----------+------+------+

| CSC342 | Morning | R128 | 201 |

| CSC343 | Noon | R128 | 203 |

| CSC345 | Night | R154 | 204 |

| ECE300 | Morning | R111 | 202 |

| ECE301 | Noon | R111 | 203 |

| ENG366 | Morning | R154 | 203 |

| ENG367 | Evening | R111 | 205 |

| HIS320 | Evening | R128 | 205 |

+--------+----------+------+------+

8 rows in set (0.00 sec)

mysql> CREATE TABLE enrolled(

-> snum INT,

-> cname VARCHAR(6));

Query OK, 0 rows affected (0.04 sec)

mysql> INSERT INTO enrolled

-> VALUES

-> (101,'CSC342'),

-> (101,'CSC343'),

-> (101,'CSC345'),

-> (101,'ECE300'),

-> (101,'ENG366'),

-> (102,'CSC343'),

-> (102,'CSC345'),

-> (102,'ECE301'),

-> (103,'ECE300'),

-> (103,'ECE301'),

-> (104,'CSC342'),

-> (104,'ECE301'),

-> (105,'CSC345'),

-> (105,'ECE300'),

-> (106,'ENG366'),

-> (106,'HIS320'),

-> (107,'CSC342'),

-> (107,'ENG366'),

-> (108,'ENG367'),

-> (108,'HIS320'),

-> (109,'ECE300'),

-> (109,'ECE301'),

-> (110,'ENG366'),

-> (110,'HIS320');

Query OK, 24 rows affected (0.01 sec)

Records: 24 Duplicates: 0 Warnings: 0

mysql> SELECT \* FROM enrolled;

+------+--------+

| snum | cname |

+------+--------+

| 101 | CSC342 |

| 101 | CSC343 |

| 101 | CSC345 |

| 101 | ECE300 |

| 101 | ENG366 |

| 102 | CSC343 |

| 102 | CSC345 |

| 102 | ECE301 |

| 103 | ECE300 |

| 103 | ECE301 |

| 104 | CSC342 |

| 104 | ECE301 |

| 105 | CSC345 |

| 105 | ECE300 |

| 106 | ENG366 |

| 106 | HIS320 |

| 107 | CSC342 |

| 107 | ENG366 |

| 108 | ENG367 |

| 108 | HIS320 |

| 109 | ECE300 |

| 109 | ECE301 |

| 110 | ENG366 |

| 110 | HIS320 |

+------+--------+

24 rows in set (0.00 sec)

mysql> CREATE TABLE faculty(

-> fid INT,

-> fname CHAR(15),

-> deptid INT);

Query OK, 0 rows affected (0.04 sec)

mysql> INSERT INTO faculty

-> VALUES

-> (201,'S. Jackson',301),

-> (202,'M. Shanks',302),

-> (203,'I. Teach',302),

-> (204,'A. Zobrah',303),

-> (205,'M. Jensen',303);

Query OK, 5 rows affected (0.02 sec)

Records: 5 Duplicates: 0 Warnings: 0

mysql> SELECT \* FROM faculty;

+------+------------+--------+

| fid | fname | deptid |

+------+------------+--------+

| 201 | S. Jackson | 301 |

| 202 | M. Shanks | 302 |

| 203 | I. Teach | 302 |

| 204 | A. Zobrah | 303 |

| 205 | M. Jensen | 303 |

+------+------------+--------+

5 rows in set (0.00 sec)

**1. Find the names of all Juniors(Level = JR) who are enrolled in a class taught by I. Teach.**

**Answer:**

mysql> SELECT s.sname, e.cname

-> FROM student s,class c,enrolled e, faculty f

-> WHERE e.snum = s.snum

-> AND c.cname = e.cname

-> AND f.fid = c.fid

-> AND s.level = 'JR'

-> AND f.fname = 'I. Teach';

+--------+--------+

| sname | cname |

+--------+--------+

| Smith | CSC343 |

| Smith | ECE301 |

| Tom | ECE301 |

| Hellen | ENG366 |

| Andy | ECE301 |

+--------+--------+

5 rows in set (0.00 sec)

**2. Find the age of the oldest student who is either a History major or enrolled in a course taught by I. Teach.**

**Answer:**

mysql> SELECT MAX(s.age)

-> FROM student s, class c, enrolled e, faculty f

-> WHERE s.snum = e.snum

-> AND e.cname = c.cname

-> AND c.fid = f.fid

-> AND ( s.major = 'History' AND f.fname = 'I. Teach');

+------------+

| MAX(s.age) |

+------------+

| 23 |

+------------+

1 row in set (0.00 sec)

**3. Find the names of all classes that either meet in room R128 or have five or more students enrolled.**

**Answer:**

mysql> SELECT c.cname, c.room , COUNT(e.snum)

-> FROM student s, class c, enrolled e, faculty f

-> WHERE s.snum = e.snum

-> AND e.cname = c.cname

-> AND c.fid = f.fid

-> GROUP BY e.cname

-> HAVING (c.room = 'R128' OR COUNT(e.snum) >= 5);

+--------+------+---------------+

| cname | room | COUNT(e.snum) |

+--------+------+---------------+

| CSC343 | R128 | 2 |

| CSC342 | R128 | 3 |

| HIS320 | R128 | 3 |

+--------+------+---------------+

3 rows in set (0.00 sec)

**4. Find the names of all students who are enrolled in two class that meet at the same time.**

**Answer:**

mysql> SELECT DISTINCT s.sname

-> FROM class c1, enrolled as e1, class c2, enrolled e2, student s

-> WHERE s.snum = e1.snum

-> AND s.snum = e2.snum

-> AND c1.cname = e1.cname

-> AND c2.cname = e2.cname

-> AND e1.snum = e2.snum

-> AND e1.cname <> e2.cname AND c1.meets\_at = c2.meets\_at;

+--------+

| sname |

+--------+

| Jhon |

| Smith |

| Hellen |

| Bob |

+--------+

4 rows in set (0.00 sec)

**5. Find the names of faculty members who teach in every room in which some class is taught.**

**Answer:**

mysql> SELECT f.fname

-> FROM class c, faculty f

-> WHERE f.fid = c.fid

-> GROUP BY f.fid

-> HAVING COUNT(DISTINCT c.room) = (SELECT COUNT(DISTINCT room)

-> FROM class);

+----------+

| fname |

+----------+

| I. Teach |

+----------+

1 row in set (0.00 sec)

**6. Find the names of faculty members for whom the combined enrollment of the course that they teach is less than five.**

**Answer:**

mysql> SELECT f.fname

-> FROM enrolled e, class c, faculty f

-> WHERE c.cname = e.cname

-> AND f.fid = c.fid

-> GROUP BY c.fid

-> HAVING COUNT(e.snum)<5;

+------------+

| fname |

+------------+

| S. Jackson |

| A. Zobrah |

| M. Shanks |

| M. Jensen |

+------------+

4 rows in set (0.00 sec)

**7. For each level, print the level and the average age of students for that level.**

**Answer:**

mysql> SELECT s.level, AVG(s.age)

-> FROM student s

-> GROUP BY s.level;

+-------+------------+

| level | AVG(s.age) |

+-------+------------+

| SR | 21.0000 |

| JR | 20.4000 |

+-------+------------+

2 rows in set (0.00 sec)

**8. For all levels except JR, print the level and the average age of students for that level.**

**Answer:**

mysql> SELECT s.level, AVG(s.age)

-> FROM student s

-> GROUP BY s.level

-> HAVING s.level <> 'JR';

+-------+------------+

| level | AVG(s.age) |

+-------+------------+

| SR | 21.0000 |

+-------+------------+

1 row in set (0.00 sec)

**9. For each faculty member that has taught class only in room R128 print the faculty member's name and the total number of classes he or she has taught.**

**Answer:**

mysql> SELECT f.fname, count

-> FROM class c, faculty f, (SELECT c.fid t, COUNT(c.cname) count

-> FROM class c

-> GROUP BY c.fid) t1

-> WHERE t = c.fid AND c.fid = f.fid

-> AND c.room = 'R128';

+------------+-------+

| fname | count |

+------------+-------+

| S. Jackson | 1 |

| I. Teach | 3 |

| M. Jensen | 2 |

+------------+-------+

3 rows in set (0.00 sec)

**10. Find the names of students enrolled in the maximum number of classes.**

**Answer:**

mysql> SELECT s.sname

-> FROM enrolled as e, student as s

-> WHERE s.snum=e.snum

-> GROUP BY e.snum

-> ORDER BY COUNT(e.cname) DESC

-> LIMIT 1;

+-------+

| sname |

+-------+

| Jhon |

+-------+

1 row in set (0.00 sec)