FAIZAN CHOUDHARY

20BCS021

DBMS LAB

4th April 2022

Creation:

- ```
  mysql> use 20bcs021_faizan;
  ```
- ```
  mysql> CREATE TABLE Student2
      -> (
      -> snum     INT     NOT NULL,
      -> sname    VARCHAR(15),
      -> major    VARCHAR(10),
      -> level    VARCHAR(3),
      -> age      INT,
      -> PRIMARY KEY (snum)
      -> );
  ```
- ```
  mysql> INSERT INTO Student2 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);
  ```
- ```
  mysql> SELECT * FROM Student2;
  ```

```
mysql> SELECT * FROM Student2;
+------+---------+---------+-------+------+
| 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 (0.00 sec)
```

- ```
  mysql> CREATE TABLE Faculty
      -> (
      -> fid       INT     NOT NULL,
      -> fname     VARCHAR(15),
      -> deptid    INT,
      -> PRIMARY KEY (fid)
      -> );
  ```
- ```
  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);
  ```
- ```
  mysql> SELECT * FROM Faculty;
  ```

```
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)
```

- ```
  mysql> CREATE TABLE Class
      -> (
      -> cname     VARCHAR(10)      NOT NULL,
      -> meets_at  VARCHAR(10),
      -> room      VARCHAR(5),
      -> fid       INT     NOT NULL,
      -> PRIMARY KEY (cname),
      -> FOREIGN KEY (fid) REFERENCES Faculty(fid)
      -> );
  ```
- ```
  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);
```

- mysql> SELECT * FROM Class;

```
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 Enrolled2
    -> (
    -> snum     INT,
    -> cname    VARCHAR(10),
    -> FOREIGN KEY (snum) REFERENCES Student2(snum),
    -> FOREIGN KEY (cname) REFERENCES Class (cname)
    -> );
```

-
```
mysql> INSERT INTO Enrolled2 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');
```

- `mysql> SELECT * FROM Enrolled2;`

```
mysql> SELECT * FROM Enrolled2;
+------+--------+
| 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)
```

Queries:

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

```
mysql> SELECT sname, cname
    -> FROM Student2 S NATURAL JOIN Enrolled2 E NATURAL JOIN Class
C NATURAL JOIN Faculty F
    -> WHERE S.level='JR' AND F.fname='I. Teach';
```

OUTPUT:

```
+--------+--------+
| 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.**

```
mysql> SELECT S.sname, S.age
    -> FROM Student2 S NATURAL JOIN Enrolled2 E NATURAL JOIN Class
C NATURAL JOIN Faculty F
    -> WHERE major='History' AND fname='I. Teach' AND S.age =
(SELECT MAX(age) FROM Student2 WHERE major='History' OR fname='I.
Teach');
```

OUTPUT:

```
+---------+------+
| sname   | age  |
+---------+------+
| Charles |   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.**

```
mysql> SELECT C.cname, COUNT(*) AS count
    -> FROM Student2 S NATURAL JOIN Enrolled2 E NATURAL JOIN Class
C NATURAL JOIN Faculty F
    -> WHERE room='R128' OR C.cname IN (SELECT cname FROM Enrolled2
GROUP BY cname HAVING COUNT(*) >= 5) GROUP BY C.cname;
```

OUTPUT:

```
+--------+-------+
| cname  | count |
+--------+-------+
| CSC342 |     3 |
| CSC343 |     2 |
| HIS320 |     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.**

```
mysql> SELECT sname
    -> FROM Student2 S NATURAL JOIN Enrolled2 E NATURAL JOIN Class
C NATURAL JOIN Faculty F
    -> GROUP BY sname, meets_at
    -> HAVING COUNT(*) >= 2;
```

OUTPUT:

```
+--------+
| sname  |
+--------+
| Jhon   |
| Hellen |
| Smith  |
| 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.**

```
mysql> SELECT fname
    -> FROM Class C NATURAL JOIN Faculty F
    -> GROUP BY fname
    -> HAVING COUNT(*) = (SELECT COUNT(DISTINCT room) FROM Class);
```

OUTPUT:

```
+----------+
| 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.**

```
mysql> SELECT fname, COUNT(*) AS count
    -> FROM Class C NATURAL JOIN Enrolled2 E NATURAL JOIN Faculty F
    -> GROUP BY fname
    -> HAVING COUNT(*) < 5;
```

OUTPUT:

```
+------------+-------+
| fname      | count |
+------------+-------+
| S. Jackson |     3 |
| M. Shanks  |     4 |
| A. Zobrah  |     3 |
| M. Jensen  |     4 |
+------------+-------+
4 rows in set (0.00 sec)
```

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

```
mysql> SELECT level, AVG(age) as avg_age
    -> FROM Student2
    -> GROUP BY level;
```

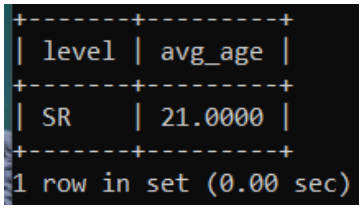
OUTPUT:

```
+-------+---------+
| level | avg_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.**

```
mysql> SELECT level, AVG(age) as avg_age
    -> FROM Student2
    -> WHERE level <> 'JR'
    -> GROUP BY level;
```

OUTPUT:

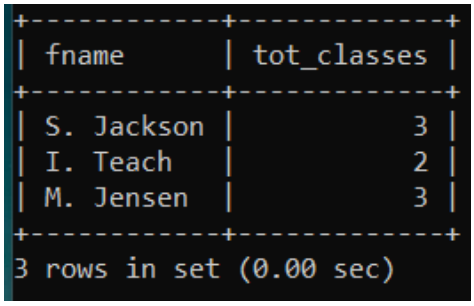


```
+-------+---------+
| level | avg_age |
+-------+---------+
| SR    | 21.0000 |
+-------+---------+
1 row in set (0.00 sec)
```

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.**

```
mysql> SELECT fname, COUNT(*) as tot_classes
    -> FROM Faculty NATURAL JOIN Class NATURAL JOIN Enrolled2
    -> WHERE room = 'R128'
    -> GROUP BY fname;
```

OUTPUT:

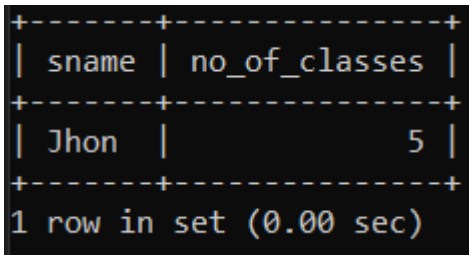


```
+------------+-------------+
| fname      | tot_classes |
+------------+-------------+
| S. Jackson |           3 |
| I. Teach   |           2 |
| M. Jensen  |           3 |
+------------+-------------+
3 rows in set (0.00 sec)
```

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

```
mysql> SELECT DISTINCT S.sname, COUNT(snum) AS no_of_classes
    -> FROM Student2 S NATURAL JOIN Enrolled2
    -> WHERE S.snum IN (SELECT E.snum FROM Enrolled2 E GROUP BY
E.snum HAVING COUNT(*) >= ALL (SELECT COUNT(*) FROM Enrolled2 E2
GROUP BY E2.snum));
```

OUTPUT:



```
+-------+---------------+
| sname | no_of_classes |
+-------+---------------+
| Jhon  |             5 |
+-------+---------------+
1 row in set (0.00 sec)
```