**10/16/2019**

mysql> create or replace table faculty\_more (

-> facId int not null,

-> fname varchar(20) not null,

-> lname varchar(20) not null,

-> deptCode varchar(4) not null,

-> department varchar(30) not null,

-> rank varchar(25),

-> constraint Faculty\_more\_facId\_pk primary key (facId),

-> constraint Faculty\_more\_deptCode\_fk foreign key (deptCode)

-> references Department(deptCode));

ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'table faculty\_more (

facId int not null,

fname varchar(2' at line 1

mysql>

Reason: rank becomes a keyword in MySQL 8.0.

(5) List the student names, their majors and credits for those with credits between 30 and 80 in the following format. Note the names of the result columns. The result is shown in the descending order of the number of credits.

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

| student | major | Number of credits |

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

| David Hawk | CSCI | 66 |

| Larry Johnson | ITEC | 66 |

| Tony Hawk | CSCI | 40 |

| Mary Hawk | CSCI | 35 |

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

4 rows in set (0.00 sec)

[1] Output columns:

1. Student: student.fname, ‘ ‘, student.lname
2. Major: student.major
3. Number of credits: credits

[2] Source: student

[3] condition: credits between 30 and 80

[4] Ordering: descending order of the number of credits.

Implementation:

-- F18 Q5

--

-- [1] Output columns:

-- 1. Student: student.fname, ' ', student.lname

-- 2. Major: student.major

-- 3. Number of credits: credits

-- [2] Source: student

-- [3] condition: credits between 30 and 80

-- [4] Ordering: descending order of the number of credits.

SELECT DISTINCT CONCAT(s.fname, ' ', s.lname) AS student,

s.major,

s.credits AS `Number of credits`

FROM student AS s

WHERE credits BETWEEN 30 AND 80

ORDER BY s.credits DESC;

(6) List the names of the faculty members who do not teach any CSCI course. Be mindful of the result column name.

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

| faculty not teaching CSCI classes |

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

| Sharon Mannes |

| Daniel Kim |

| Andrew Byre |

| Deborah Gump |

| Art Allister |

| Benjamin Yu |

| Katrina Bajaj |

| Jorginlo Neymar |

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

8 rows in set (0.00 sec)

[1] Output columns:

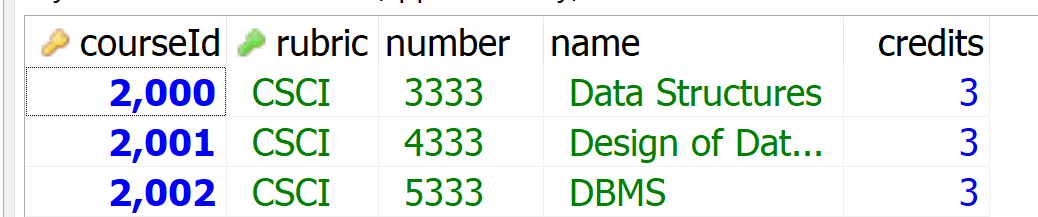
1. faculty not teaching CSCI classes: faculty.fname, ‘ ‘,faculty.lname

[2] Sources:

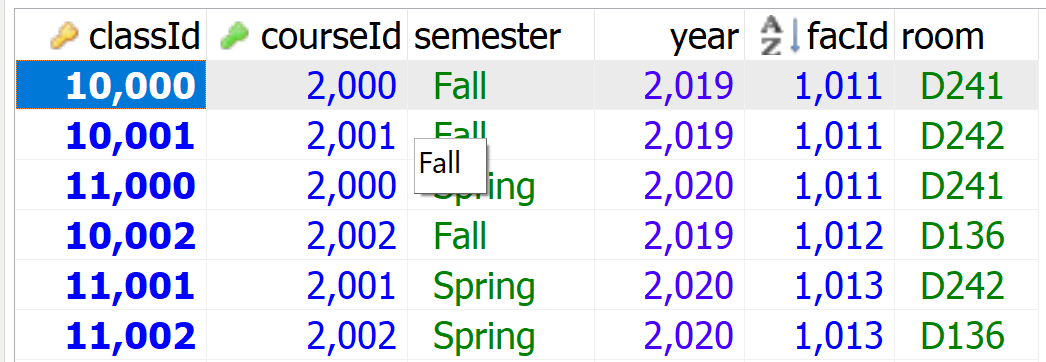
1. faculty

[3] Condition: do not teach any CSCI course.

3a. All CSCI course: courseId: 2000, 2001, 2002

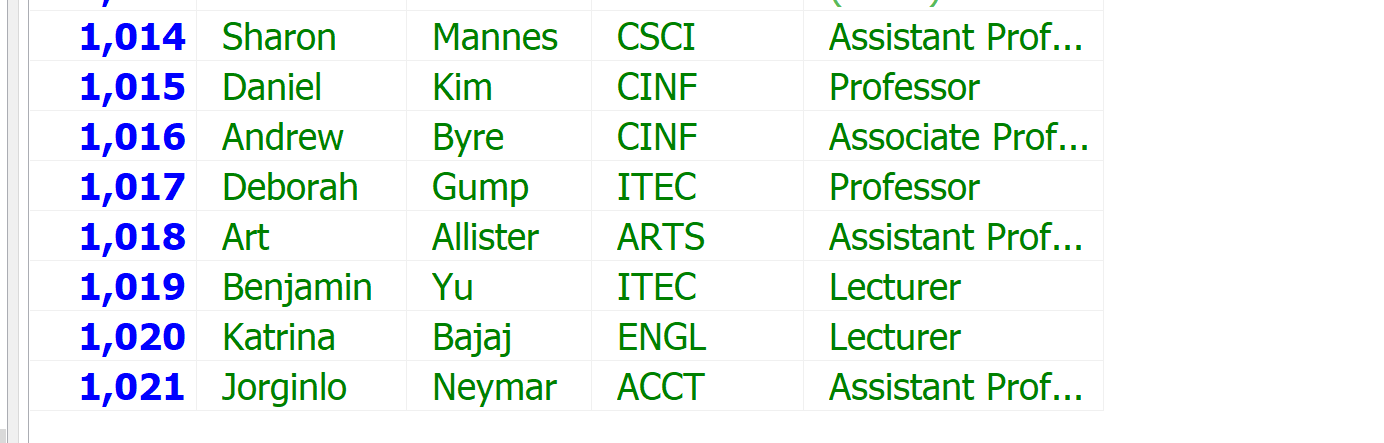


3b. All CSCI classes: 10000, 10001,11000, 10002, 11001, 11002



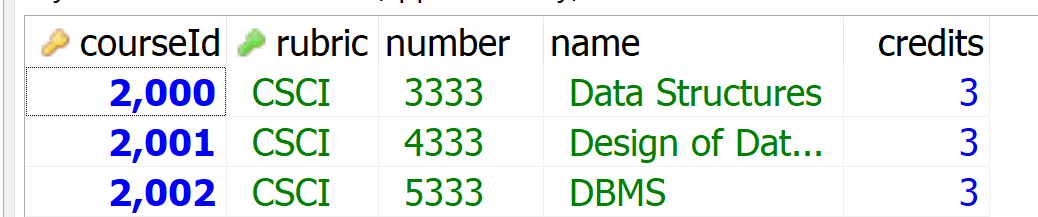
3c. faculty teaching a CSCI class: 1011, 1012, 1013

3d. faculty not teaching a CSCI class: all faculty not in 3c:

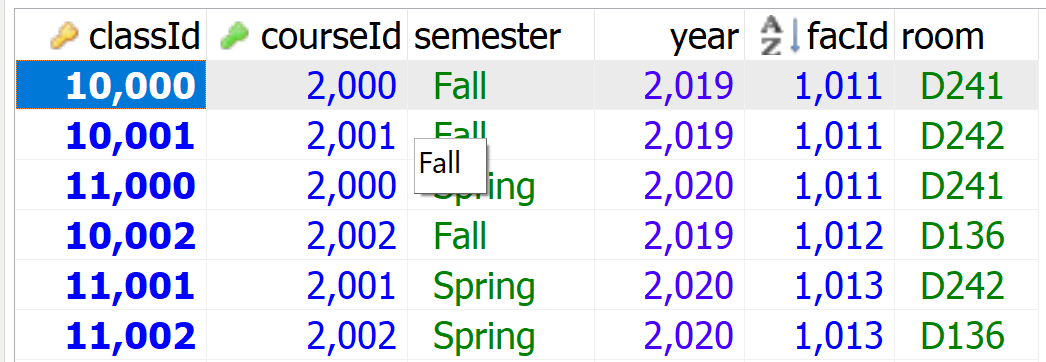


**Component problem [6b]: facId that teaches a CSCI course:**

3a. All CSCI course: courseId: 2000, 2001, 2002



3b. All CSCI classes: 10000, 10001,11000, 10002, 11001, 11002



3c. faculty teaching a CSCI class: 1011, 1012, 1013

[6b] analysis:

Ouptut: facId in faculty 1011, 1012, 1013

Source: course (CSCI), class (facId teaching the CSCI courses)

Condition:

1. Join: class.CourseId (FK in class) = course.CourseID (PK in course)
2. Problem: course.rubric = ‘CSCI’