

CSCI 2020 Database Fundamentals Lab7

Submit as a text document called *LastnameFirstnameLab7.sql* to the drop box.

Print out a hard copy, staple this sheet to the front, and submit to your instructor before leaving the lab.

Notes:

- o **Be sure all style guidelines are complied with.** (SQL style guidelines are available in the Additional Material section of Content in D2L).
- o Your file should include a **header block of comments** as shown in the SQL Style Guidelines.
- o **Be sure to add a comment above the solution to each task.** The comment should include the task number and the task itself, and should also explain in common language how your solution accomplishes the task.

Ex. 1) List all contents of the Instructor table.

The code should look like:

```
/* 1 - list all contents of the Instructor table --
   The * operator in the select clause will
   select all columns from the specified table.
*/
select * from INSTRUCTOR;
```

- o To make your code more readable, I would prefer that you **format your select statements as follows:** (note the from, where and order by are all on separate lines indented under the select)

```
select COLUMNS_HERE
from TABLE_HERE
where CONDITION1_HERE
      or CONDITION2_HERE
order by COLUMN_HERE;
```

You are submitting a file that has the sql code in it, not the results. I must be able to run your .sql file in its entirety. Make sure that comments are added properly.

TASKS

WARNING: Remember that you have students and instructors who share family names

1. List the course ID's in which at least one student has made a 4.0. No duplicates. (25 rows returned)
2. List the names of the students who have received a 4.0 in any course. No duplicates. (11 rows returned)
3. List the instructors who have given a student a 4.0 in any course. No duplicates. (17 rows returned)

4. List the course ID's, the course names, and the names of the students who received a 4.0, along with the semester in which they took the course. Order your results by the student's family name and the semester code *(57 rows returned)*
5. List the course ID's, the course section numbers, the course name, the semester code, the student's first and last name in a single column named STUDENT (use 'M.' if the student has no first name) and the first and last name of the instructor in a single column named TEACHER (use 'M.' if the instructor has no first name) for the classes in which a student received a 4.0. Order your results by the STUDENT column and by semester code. *(57 rows returned)*
6. Which students have received the most 4.0's? *(2 rows returned)*
7. Which instructors have given the most 4.0's *(3 rows returned)*

BONUS

We want to know which classes are the "hard" ones in the Fall '08 (semester code 083) semester. We can assume that a class is hard if the average grade for the class is less than a 2.5. Write a query that lists the all of the information about the classes from the 083 semester along with instructor information, and whether it is a hard or easy class. If an instructor has a PhD, we wish to refer to them as 'Dr.', otherwise refer to them as 'Mr./Ms.' Format your output to match the following...

It looks like ENGL1020-002(Argumentative Writing) with Mr./Ms. C3-P0 is a hard class.

It looks like PHYS2010-901(General Physics I - NonCalculus) with Mr./Ms. Calrissian is an easy class.

It looks like CSCI2150-001(Computer Organization) with Dr. Kenobi is a hard class.

It looks like CSCI1100-001(Using Information Technology) with Mr./Ms. Solo is a hard class.

It looks like ARTA2071-002(Basic Weaving) with Dr. Windu is an easy class.

...etc *(26 rows returned)*



```
1  /* BuchananBrandonLab7.sql
2  ----- \
3  --
4  Programmer's Name: Brandon Buchanan
5  Course: CSCI2020                      Section Number: 002
6  Creation Date: 10/26/2011             Date of Last Modification: 10/26/201 \
7  1
8  E-mail address: zbwb8@goldmail.etsu.edu
9  ----- \
10 --
11 Purpose -
12   Lab 7 - Make up lab and review of material learned throughout
13           semester
14 ----- \
15 --
16 Notes on specifications, special algorithms, and assumptions:
17 Designed to work on the database created by the schooldb.sql
18 script
19 */
20
21 /* 1. - Display courses that have at least one student that made a 4 \
22        .0
23        grade. This query displays all course IDs where the student \
24        's grade
25        is 4, but doesn't display the same course twice using distin \
26        ct. */
27
28 select distinct TAKES.COURSE
29   from TAKES
30  where TAKES.GRADE = '4';
31
32
33 /* 2. - Display students who made a 4.0 in any course. This query di \
34        splays
35        the students' first and last names when that student received \
36        a 4
37        as a grade for a course, but students are not listed twice. \
38        */
39
40 select distinct STUDENT.GIVEN_NAME, STUDENT.FAMILY_NAME
41   from STUDENT
42  join TAKES
43 on STUDENT.ID = TAKES.STUDENT
```

```
34      where TAKES.GRADE = '4';
35
36  /* 3. - Display a list of instructors who gave a student a 4.0 in any \
course.
37      This query displays the instructors' first and last names and \
joins
38      the INSTRUCTOR, SECTION, and TAKES table, then displays on th \
e
39      instructors who gave a student a 4.0 grade. No instructor is \
listed
40      twice. */
41
42  select distinct INSTRUCTOR.GIVEN_NAME, INSTRUCTOR.FAMILY_NAME
43      from INSTRUCTOR
44      join SECTION
45      on INSTRUCTOR.ID = SECTION.INSTRUCTOR
46      join TAKES
47      on SECTION.CODE = TAKES.SECTION and SECTION.COURSE = TAKES.COURSE \
and
48      SECTION.TERM = TAKES.TERM
49      where TAKES.GRADE = '4';
50
51  /* 4. - Display the course ID and name, student name, and term taken \
when a
52      student received a 4.0 in any course. This query displays t \
he
53      course ID and name, students' first and last names', and the \
term
54      the course was taken in whenever the grade received was a 4. \
0. The
55      output is ordered first by the student's last name and then \
ordered
56      by the semester code it was taken in when last names are ali \
ke. */
57
58  select TAKES.COURSE, COURSE.NAME, STUDENT.GIVEN_NAME, STUDENT.FAMILY_ \
NAME,
59      TAKES.TERM
60      from TAKES
61      join COURSE
62      on TAKES.COURSE = COURSE.ID
63      join STUDENT
64      on TAKES.STUDENT = STUDENT.ID
```

```
100     where TAKES.GRADE = '4'
101     order by STUDENT, TAKES.TERM asc;
102
103  /* 6. - Display the students who have received the most 4.0 grades.  \
The inner
104     query finds the maximum number of 4.0 grades that exist, while \
the
105     outer query displays the student's first and last name when the \
total
106     number of 4.0 grades he/she received is the maximum number of \
4.0
107     grades achieved. */
108
109  select STUDENT.GIVEN_NAME, STUDENT.FAMILY_NAME
110  from STUDENT
111  join TAKES
112    on STUDENT.ID = TAKES.STUDENT
113  where TAKES.GRADE = '4'
114  group by STUDENT.GIVEN_NAME, STUDENT.FAMILY_NAME
115  having count(*) = (
116
117    select max(count(*))
118    from STUDENT
119    join TAKES
120      on STUDENT.ID = TAKES.STUDENT
121    where TAKES.GRADE = '4'
122    group by STUDENT.GIVEN_NAME, STUDENT.FAMILY_NAME
123  );
124
125
126  /* 7. - Display the instructors who have given the most 4.0 grades.  \
The inner
127     query finds the maximum number of 4.0 grades that exist, while \
the
128     outer query displays the instructor's first and last name when \
the total
129     number of 4.0 grades he/she has given is the maximum number of \
4.0
130     grades achieved. */
131
132  select INSTRUCTOR.GIVEN_NAME, INSTRUCTOR.FAMILY_NAME
133  from INSTRUCTOR
134  join SECTION
```

```
173         else 'easy class.'
174     end
175
176     from TAKES
177     join SECTION
178         on SECTION.CODE = TAKES.SECTION and SECTION.COURSE = TAKES.COURSE \
and
179         SECTION.TERM = TAKES.TERM
180     join INSTRUCTOR
181         on INSTRUCTOR.ID = SECTION.INSTRUCTOR
182     join COURSE
183         on TAKES.COURSE = COURSE.ID
184
185     where TAKES.TERM = '083';
186
```