

CSCI 2020 Database Fundamentals

Lab3

Submit as a text document called *LastNameFirstNameLab3.sql* to the drop box.
Print a hard copy, staple this sheet to the front and submit to your instructor before leaving the lab.

Notes:

- **Be sure all style guidelines are complied with.** (SQL style guidelines are available in the Additional Material section of Content in D2L).
- Your file should include a **header block of comments** as shown in the SQL Style Guidelines.
- **Be sure to add a comment above the solution to each task.** The comment should include the task number and the task itself.
Ex. 1) List all contents of the Instructor table.

The code should look like:

```
/* 1 – list all contents of the Instructor table */
select * from INSTRUCTOR;
```

- 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

- 1) For all instructors whose family names begin with A-C, list their name and department they work for labeled as shown below. If they do not have a department, list "No Department".

INSTRUCTOR_ASSIGNMENTS
Antilles works for No Department
Chewbacca works for MATH
Binks works for CSCI
Calrissian works for PHYS
Amidala works for No Department
C3-P0 works for ENGL

- 2) List the id's and year of hire for all department chairs. Do not show the entire hire date, only the year. Label the year as YEAR_OF_HIRE.

ID	YEAR_OF_HIRE
239	1975
202	1987
4008	1951

- 3) For all Professors, list their id, salary and difference between their salary and the average salary for a Professor (the hypothetical average salary for a Professor is \$100,000). The difference column should be labeled DIFFERENCE_FROM_AVERAGE.

ID	SALARY	DIFFERENCE_FROM_AVERAGE
239	107000	7000
202	78000	-22000
4008	115000	15000
3519	98000	-2000
4200	101500	1500
2726	132000	32000

- 4) For each student from a school that begins with an 'S', list their id, family name and the name of the department that houses their major. Be sure to include students who have yet to declare a major. If they do not have a major, the Department should be listed as UDEC for Undeclared. The column listing the department should be labeled DEPARTMENT.

ID	FAMILY_NAME	DEPARTMENT
10172736	Chakotay	CSCI
14193418	Crusher	CSCI
11204763	Data	CSCI
14270233	O'Brien	CSCI
12282934	Pulaski	CSCI
11187308	Picard	CSCI
10196427	Chekov	CSCI
11734781	Paris	CSCI
10190257	Kirk	UDEC

- 5) For all students who have taken the course CSCI1250, list their ID and grade. If a student has taken the course multiple times, display all grades. Display the grades as a letter grade instead of a numeric grade. The grade equivalents are: 1.7 = C-, 2 = C, 2.3 = C+, 2.7 = B-, 3 = B, 4 = A, etc. The output should be labeled as shown below. (8 rows returned)

STUDENTS_THAT_TOOK_CSCI1250
Student 10190257 passed the class with a grade of C
Student 10198947 passed the class with a grade of A
Student 13128601 passed the class with a grade of B
Student 11204763 passed the class with a grade of C
Student 12282934 passed the class with a grade of D

- 6) For all students who have taken a Physics course, list their ID, last name, first name, course id and grade. If a student has taken the course multiple times, display all grades. If there is no given name, display a blank. Order the results by last name, first name and then course id.

ID	FAMILY_NAME	GIVEN_NAME	COURSE	GRADE
12151048	Bashir	Julian	PHYS2010	4
12151048	Bashir	Julian	PHYS2020	3
10172736	Chakotay		PHYS2010	2
10172736	Chakotay		PHYS2020	2
12186064	Chapel	Christine	PHYS2010	3
12186064	Chapel	Christine	PHYS2020	3
10196427	Chekov	Pavel	PHYS2020	1
14193418	Crusher	Wesley	PHYS2010	4
14193418	Crusher	Wesley	PHYS2020	4
11279384	Janeway	Kathryn	PHYS2010	2
11279384	Janeway	Kathryn	PHYS2010	3
92273843	Kim	Harry	PHYS2010	4
10190257	Kirk	James	PHYS2010	4
10165270	La Forge	Geordi	PHYS2010	3
10165270	La Forge	Geordi	PHYS2020	2
11734781	Paris	Tom	PHYS2010	2
11187308	Picard	Jean-Luc	PHYS2010	4
13128601	Riker	William	PHYS2010	3
13191810	Spock		PHYS2010	4
13191810	Spock		PHYS2020	4
13181795	Sulu	Hikaru	PHYS2010	2
13181795	Sulu	Hikaru	PHYS2020	3
12103183	Tuvok		PHYS2010	3
12103183	Tuvok		PHYS2020	3

- 7) Print a course listing for the "083" term. It should include the course's id, name, meeting days(MW, TR, MWF, etc.), meeting times (in the format "8:00AM – 10:00AM"), instructor's name, and the maximum number of seats for the course section.

ID	NAME	DAYS	TIMES	INSTRUCTOR	MAX_SEATS
CSCI1100	Using Information Technology	MWF	8:15 AM - 9:10 AM	Han Solo	15
CSCI1100	Using Information Technology	MW	7:00 PM - 8:20 PM	R2-D2	15
CSCI1250	Intro Computer Science I	T	5:00 PM - 8:10 PM	Anakin Skywalker	5
CSCI1260	Intro Computer Science II	TR	5:00 PM - 7:00 PM	Yoda	5
CSCI1710	WWW - Design and Creation	TR	8:15 AM - 10:15 AM	Obi-Wan Kenobi	10
CSCI2150	Computer Organization	MW	8:15 AM - 10:15 AM	Obi-Wan Kenobi	10
CSCI2160	Assembly Language	MW	8:15 AM - 10:15 AM	Count Dooku	10
CSCI2210	Data Structures	TR	8:15 AM - 10:15 AM	Han Solo	5
CSCI2910	Client / Server Side Prog	MW	8:15 AM - 10:15 AM	Yoda	7
CSCI3250	Software Engineering I	MW	5:00 PM - 7:00 PM	Yoda	7

.....etc. 34 rows returned

Bonus) Do problem 7 again, but this time translate the meeting days from characters to days of the week. For example, M would be translated to Mon or Monday.

<i>ID</i>	<i>NAME</i>	<i>DAYS</i>	<i>TIMES</i>	<i>INSTRUCTOR</i>	<i>MAX_SEATS</i>
CSCI1100	Using Information Technology	Mon Wed Fri	8:15 AM - 9:10 AM	Han Solo	15
CSCI1100	Using Information Technology	Mon Wed	7:00 PM - 8:20 PM	R2-D2	15
CSCI1250	Intro Computer Science I	Tue	5:00 PM - 8:10 PM	Anakin Skywalker	5
CSCI1260	Intro Computer Science II	Tue Thu	5:00 PM - 7:00 PM	Yoda	5
CSCI1710	WWW - Design and Creation	Tue Thu	8:15 AM - 10:15 AM	Obi-Wan Kenobi	10
CSCI2150	Computer Organization	Mon Wed	8:15 AM - 10:15 AM	Obi-Wan Kenobi	10

....etc. 34 rows returned

105

```
1  /* BuchananBrandonLab3.sql
2  ----- \
3  --
4  Programmer's Name: Brandon Buchanan
5  Course: CSCI2020                      Section Number: 002
6  Creation Date: 09/28/2011             Date of Last Modification: 09/28/201 \
7  1
8  E-mail address: zbwb8@goldmail.etsu.edu
9  ----- \
10 --
11 Purpose -
12 Lab 3
13 ----- \
14 --
15 Notes on specifications, special algorithms, and assumptions:
16 Designed to work on the database created by the schooldb.sql
17 script
18 */
19
20 /* 1. - List instructors and their department whose last names begin \
21 with
22         A, B, or C. This query displays the string's format. If a \
23 department
24         value is null, then it displays No Department, otherwise it \
25 displays
26         what it normally would. */
27
28 select INSTRUCTOR.FAMILY_NAME || ' works for ' ||
29 case
30     when INSTRUCTOR.DEPARTMENT is null then 'No Department'
31     when INSTRUCTOR.DEPARTMENT is not null then INSTRUCTOR.DEPARTMENT
32 end as INSTRUCTOR_ASSIGNMENTS
33 from INSTRUCTOR
34 where INSTRUCTOR.FAMILY_NAME like 'A%'
35     or INSTRUCTOR.FAMILY_NAME like 'B%'
36     or INSTRUCTOR.FAMILY_NAME like 'C%';
37
38 /* 2. - Display the ID and department of all chairs, displaying on t \
39 he year of
40         the hire date. This query displays the information and extr \
41 acts the
42         year information from the hire date before displaying. */
```

- 5 comments, but you are very close

how?

```
35
36  select INSTRUCTOR.ID, extract(year from INSTRUCTOR.HIRED) as YEAR_OF_ \
    HIRE
37  from INSTRUCTOR
38  where INSTRUCTOR.ROLE = 'Chair';
39
40
41  /* 3. - List ID, salary, and the average salary difference (assuming \
    it is
42          100,000) of all professors. This query displays the informat \
    ion and
43          subtracts 100000 from a professors salary before displaying \
    the
44          result in the DIFFERENCE_FROM_AVERAGE column. */
45
46  select INSTRUCTOR.ID, INSTRUCTOR.SALARY, (INSTRUCTOR.SALARY - 100000) \
    as DIFFERENCE_FROM_AVERAGE
47  from INSTRUCTOR
48  where INSTRUCTOR.TITLE = 'Professor';
49
50
51  /* 4. - List ID, last name, and department of all students that atten \
    ded a high
52          school that started with the letter 'S'. This query displays \
    the
53          student's department unless it is null, which would then disp \
    lay UDEC. how?
54          It also finds only students who attended a high school that s \
    tarted with how?
55          'S'. */
56
57  select STUDENT.ID, STUDENT.FAMILY_NAME,
58         case
59           when MAJOR.DEPARTMENT is null then 'UDEC'
60           when MAJOR.DEPARTMENT is not null then MAJOR.DEPARTMENT
61         end as DEPARTMENT
62  from STUDENT join MAJOR
63   on STUDENT.MAJOR = MAJOR.TITLE
64  where STUDENT.HIGH_SCHOOL like 'S%';
65
66
67  /* 5. - Display all student IDs and grades of students who took CSCI \
    1250.
```

```
68      This query displays the letter grade depending on the number \
grade
69      that he/she received.  */ how?
70
71  select 'Student ' || TAKES.STUDENT || ' passed the class with a grade \
of ' ||
72      case
73          when TAKES.GRADE = '0' then 'F'
74          when TAKES.GRADE = '0.3' then 'F+'
75          when TAKES.GRADE = '0.7' then 'D-'
76          when TAKES.GRADE = '1' then 'D'
77          when TAKES.GRADE = '1.3' then 'D+'
78          when TAKES.GRADE = '1.7' then 'C-'
79          when TAKES.GRADE = '2' then 'C'
80          when TAKES.GRADE = '2.3' then 'C+'
81          when TAKES.GRADE = '2.7' then 'B-'
82          when TAKES.GRADE = '3' then 'B'
83          when TAKES.GRADE = '3.3' then 'B+'
84          when TAKES.GRADE = '3.7' then 'A-'
85          when TAKES.GRADE = '4' then 'A'
86      end as STUDENTS_THAT_TOOK_CSCI1250
87      from TAKES
88      where TAKES.COURSE = 'CSCI1250';
89
90
91  /* 6. - Displays students' ID, names, course taken, and grade of all \
students
92      that took a physics course. This query joins two tables to r \
retrieve
93      this information and displays those students that took a cour \
se that
94      starts with 'PHYS'. how?
95
96  select STUDENT.ID, STUDENT.FAMILY_NAME,
97      case
98          when STUDENT.GIVEN_NAME is null then ' '
99          when STUDENT.GIVEN_NAME is not null then STUDENT.GIVEN_NAME
100      end as GIVEN_NAME,
101      TAKES.COURSE, TAKES.GRADE
102      from STUDENT join TAKES
103      on STUDENT.ID = TAKES.STUDENT
104      where TAKES.COURSE like 'PHYS%'
105      order by STUDENT.FAMILY_NAME, STUDENT.GIVEN_NAME, TAKES.COURSE;
```

```
106
107
108  /* 7. - Display a list of all courses taught during the '083' term; \
      also
109      displays instructor, days, times, and the maximum number of \
      seats
110      allowed in the class. This query joins three tables to retr \
      ieve the
111      information and displays only those courses taught during th \
      e 083
112      term. */
113
114  select SECTION.COURSE, COURSE.NAME, SECTION.DAYS,
115         (SECTION.START_TIME || ' - ' || SECTION.END_TIME) as TIMES,
116         (INSTRUCTOR.GIVEN_NAME || ' ' || INSTRUCTOR.FAMILY_NAME) as IN \
      TRUCTOR,
117         SECTION.MAX_ENROLLMENT as MAX_SEATS
118  from SECTION join INSTRUCTOR
119         on SECTION.INSTRUCTOR = INSTRUCTOR.ID
120  join COURSE
121         on SECTION.COURSE = COURSE.ID
122  where SECTION.TERM = '083';
123
124
125  /* BONUS - Display a list of all courses taught during the '083' ter \
      m; also
126      displays instructor, days, times, and the maximum number of \
      seats
127      allowed in the class. This query joins three tables to retr \
      ieve the
128      information and displays only those courses taught during th \
      e 083
129      term. This query also uses a case to convert the single let \
      ter day
130      to a three letter day. It should be noted that it only work \
      s for
131      the specified cases (could not figure out how to go through \
      each letter
132      of each row to convert), but displays 'Error' if this is
133      encountered. */
134
135  select SECTION.COURSE, COURSE.NAME,
136         case
```

next some replace() functions

+10

```
replace(  
replace(  
replace(  
replace(  
replace( SECTION.DAYS, 'M', 'Mon'),  
                                'T', 'Tue'),  
                                'W', 'Wed'),  
                                'R', 'Thu'),  
                                'F', 'Fri')
```

```
137         when SECTION.DAYS = 'M' then 'Mon '
138         when SECTION.DAYS = 'T' then 'Tue '
139         when SECTION.DAYS = 'W' then 'Wed '
140         when SECTION.DAYS = 'R' then 'Thu '
141         when SECTION.DAYS = 'F' then 'Fri '
142         when SECTION.DAYS = 'MW' then 'Mon Wed '
143         when SECTION.DAYS = 'TR' then 'Tue Thu '
144         when SECTION.DAYS = 'MWF' then 'Mon Wed Fri '
145         else 'Error'
146     end as DAYS,
147         (SECTION.START_TIME || ' - ' || SECTION.END_TIME) as TIMES,
148         (INSTRUCTOR.GIVEN_NAME || ' ' || INSTRUCTOR.FAMILY_NAME) as IN \
STRUCTOR,
149         SECTION.MAX_ENROLLMENT as MAX_SEATS
150 from SECTION join INSTRUCTOR
151     on SECTION.INSTRUCTOR = INSTRUCTOR.ID
152 join COURSE
153     on SECTION.COURSE = COURSE.ID
154 where SECTION.TERM = '083';
```