Assignment 3

Rita Savill

1. Open the Online SQL interpreter (https://www.db-book.com/db7/university-labdir/sqljs.html)

2. Write SQL codes to get a list of:

   i. Students IDs (hint: from the takes relation)
   ii. Instructors
   iii. Departments

```
SELECT ID
FROM student;

SELECT name
FROM instructor;

SELECT dept_name
FROM department;
```

ID
00128
12345
19991
23121
44553
45678
54321
55739
70557
76543
76653
98765
98988

name
Srinivasan
Wu
Mozart
Einstein
El Said
Gold
Katz
Califieri
Singh
Crick
Brandt
Kim

dept_name
Biology
Comp. Sci.
Elec. Eng.
Finance
History
Music
Physics

3. Write in SQL codes to do following queries:
i. Find the ID and name of each student who has taken at least one Comp. Sci. course; make sure there are no duplicate names in the result.

```
SELECT DISTINCT s.ID, s.name
FROM student AS s
INNER JOIN takes AS t
ON s.ID = t.ID
INNER JOIN course AS c
ON t.course_id = c.course_id
WHERE c.dept_name = 'Comp. Sci.';
```

ID	name	grade
00128	Zhang	A
00128	Zhang	A-
12345	Shankar	C
12345	Shankar	A
45678	Levy	F
45678	Levy	B+
45678	Levy	B
54321	Williams	A-
54321	Williams	B+
76543	Brown	A
98765	Bourikas	C-
98765	Bourikas	B

ii. Add grades to the list

```
# This query gives the unique grades for each student, not the grades for each CS class taken
# e.g. student Shankar took 4 CS classes, but got one C and 3 As..thus the query yields C, A
# Remove DISTINCT to see every grade for each student

SELECT DISTINCT s.ID, s.name, t.grade
FROM student AS s
INNER JOIN takes AS t
ON s.ID = t.ID
INNER JOIN course AS c
ON t.course_id = c.course_id
WHERE c.dept_name = 'Comp. Sci.';
```

ID	name
00128	Zhang
12345	Shankar
45678	Levy
54321	Williams
76543	Brown
98765	Bourikas

iii. Find the ID and name of each student who has not taken any course offered before 2017.

there are no courses taken before 2017

```
SELECT DISTINCT s.ID, s.name
FROM student AS s
INNER JOIN takes AS t
ON s.ID = t.ID
WHERE t.year >= 2017;
```

ID	name
00128	Zhang
12345	Shankar
19991	Brandt
23121	Chavez
44553	Peltier
45678	Levy
54321	Williams
55739	Sanchez
76543	Brown
76653	Aoi
98765	Bourikas
98988	Tanaka

iv. For each department, find the maximum salary of instructors in that department. You may assume that every department has at least one instructor.

```
SELECT MAX(salary), dept_name
FROM instructor
GROUP BY dept_name;
```

MAX(salary)	dept_name
72000	Biology
92000	Comp. Sci.
80000	Elec. Eng.
90000	Finance
62000	History
40000	Music
95000	Physics

v. Find the lowest, across all departments, of the per-department maximum salary computed by the preceding query.

```
SELECT MAX(salary) AS max_sal, dept_name
FROM instructor
GROUP BY dept_name
ORDER BY max_sal
LIMIT 1;
```

max_sal	dept_name
40000	Music

vi. Add names to the list

```
SELECT MAX(salary) AS max_sal, dept_name, name
FROM instructor
GROUP BY dept_name
ORDER BY max_sal
LIMIT 1;
```

max_sal	dept_name	name
40000	Music	Mozart

5. Write SQL query to find the number of students in each section. The result columns should appear in the order “courseid, secid, year, semester, num”. You do not need to output sections with 0 students.

```
SELECT course_id, sec_id, year, semester, COUNT(*) AS num
FROM takes
GROUP BY course_id, sec_id;
```

course_id	sec_id	year	semester	num
BIO-101	1	2017	Summer	1
BIO-301	1	2018	Summer	1
CS-101	1	2017	Fall	7
CS-190	2	2017	Spring	2
CS-315	1	2018	Spring	2
CS-319	1	2018	Spring	1
CS-319	2	2018	Spring	1
CS-347	1	2017	Fall	2
EE-181	1	2017	Spring	1
FIN-201	1	2018	Spring	1
HIS-351	1	2018	Spring	1
MU-199	1	2018	Spring	1
PHY-101	1	2017	Fall	1