

A decorative frame composed of two thick, light gray L-shaped lines. One L-shape is positioned in the top-left corner, and the other is in the bottom-right corner, creating an open rectangular frame around the central text.

# ASSIGNMENT 3

KIRA LOWE

## 2. Write SQL codes to get a list of:

### ■ Unique student IDs

```
1 SELECT ID
2 FROM student
```

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

### ■ Instructors

```
1 SELECT name
2 FROM instructor
```

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

### ■ Instructors

```
1 SELECT dept_name
2 FROM department
```

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

### 3. Write in SQL codes to do following queries:

- i. Students (unique ID & names) who have taken at least one CS class

```
1 SELECT DISTINCT student.ID, student.name
2 FROM student
3 INNER JOIN takes ON student.id = takes.id
4 INNER JOIN course ON takes.course_id = course.course_id
5 WHERE course.dept_name='Comp. Sci.'
```

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

### 3. Write in SQL codes to do following queries:

- Student name, ID, and grade who have taken at least one CS class

```
1 SELECT student.ID, student.name, takes.grade
2 FROM student
3 INNER JOIN takes ON student.id = takes.id
4 INNER JOIN course ON takes.course_id = course.course_id
5 WHERE course.dept_name='Comp. Sci.'
```

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

### 3. Write in SQL codes to do following queries:

- ID and name of each student who has not taken any course offered before 2017

```
1 SELECT DISTINCT student.ID, student.name
2 FROM student
3 INNER JOIN takes ON student.id = takes.id
4 INNER JOIN section ON takes.course_id = section.course_id
5 WHERE section.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

### 3. Write in SQL codes to do following queries:

- For each department, find the maximum salary of instructors in that department

```
1 SELECT dept_name, MAX(salary)
2 FROM instructor
3 GROUP BY dept_name
4 ORDER BY salary
```

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

### 3. Write in SQL codes to do following queries:

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

```
1 SELECT dept_name, MAX(salary)
2 FROM instructor
3 GROUP BY dept_name
4 ORDER BY salary
5 LIMIT 1
```

dept_name	MAX(salary)
Music	40000

- Add name

```
1 SELECT dept_name, name, MAX(salary)
2 FROM instructor
3 GROUP BY dept_name
4 ORDER BY salary
5 LIMIT 1
```

dept_name	name	MAX(salary)
Music	Mozart	40000

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”

```
1 SELECT course_id AS courseid, sec_id AS secid,  
2    year, semester, COUNT(ID) AS num  
3 FROM takes  
4 GROUP BY course_id, sec_id;
```

courseid	secid	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



4. Find instructors (with name and ID) who have never given an A grade in any course she or he has taught

```
1 SELECT DISTINCT instructor.id, name, takes.grade
2 FROM instructor
3 LEFT JOIN teaches ON instructor.id = teaches.id
4 LEFT JOIN section ON teaches.course_id = section.course_id
5 LEFT JOIN takes ON teaches.course_id = takes.course_id
6
```

- WHERE takes.grade <> '%A%'
- AND NOT takes.grade LIKE '%A%'
- FROM TABLE WHERE instructor.id NOT IN (SELECT instructor.id takes.grade LIKE '%A%')
- WHERE NOT EXISTS (SELECT 1 FROM takes takes2 WHERE takes2.instructor\_id = instructor.id AND takes2.grade LIKE '%A%')

ID	name	grade
10101	Srinivasan	A
10101	Srinivasan	A-
10101	Srinivasan	B+
10101	Srinivasan	C
10101	Srinivasan	C-
10101	Srinivasan	F
10101	Srinivasan	B
12121	Wu	C+
15151	Mozart	A-
22222	Einstein	B-
32343	El Said	B
33456	Gold	
45565	Katz	A
45565	Katz	A-
45565	Katz	B+
45565	Katz	C
45565	Katz	C-
45565	Katz	F
45565	Katz	B
58583	Califieri	
76543	Singh	
76766	Crick	A
76766	Crick	
83821	Brandt	A
83821	Brandt	B+
83821	Brandt	B
98345	Kim	C

HERE

es.grade