Assignment 3: SQL Codes

EPPS 6354: Information Management

SQL codes to create lists

SQL code to get a list of Student IDs (#2i)

```
SELECT ID
FROM student;
```

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

SQL code to get a list of Instructors (#2ii)

```
SELECT name
FROM instructor;
```

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

SQL code to get a list of Departments (#2iii)

```
SELECT dept_name
FROM department;
```

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

SQL codes to execute queries

Find ID and name of each student who has taken at least one Comp. Sci. course (#3i)

```sql
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
00128	Zhang
12345	Shankar
45678	Levy
54321	Williams
76543	Brown
98765	Bourikas

Add grades to the list (#3ii)

```sql
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	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

Find ID and name of each student who has not taken any course offered before 2017 (#3iii)

```sql
SELECT DISTINCT s.ID, s.name
FROM student AS s
INNER JOIN takes AS t ON s.ID = t.ID
WHERE t.year >= '2017';
```

OR

```sql
SELECT DISTINCT s.ID, s.name
FROM student AS s
INNER JOIN takes AS t ON s.ID = t.ID
INNER JOIN section AS s on t.course_id = s.course_id
WHERE s.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

For each department, find the maximum salary of instructors in that department (#3iv)

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

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

Find the lowest, across all departments, of the per-department maximum salary computed by the preceding query (#3v)

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

dept_name	MAX(salary)
Music	40000

Add names to the list (#3vi)

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

dept_name	name	MAX(salary)
Music	Mozart	40000

SQL codes with conditions

SQL query to find the number of students in each section (#5)

```sql
SELECT course_id AS courseid, sec_id AS secid, year, semester, COUNT(ID) AS num
FROM takes
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

BONUS: Find instructor (with name and ID) who has never given an A grade in any course she/he has taught (#4)

Instructors:

- Califieri
- Einstein
- El Said
- Gold
- Kim
- Singh
- Wu

```
SELECT DISTINCT i.ID, i.name
FROM instructor AS i
INNER JOIN advisor AS a ON i.ID = a.i_ID
INNER JOIN takes AS t ON a.s_ID = t.ID
GROUP BY i.ID, i.name
HAVING t.grade NOT LIKE 'A%';
```

ID	name
22222	Einstein
76543	Singh
98345	Kim

→ **Difficulty in executing this query**

SQL codes used to determine which instructors should be included (#4)

```sql
SELECT DISTINCT i.ID, i.name, t.grade
FROM instructor AS i
LEFT JOIN teaches AS e ON i.ID = e.ID
LEFT JOIN section AS s ON e.course_id = s.course_id
LEFT JOIN takes AS t ON e.course_id = t.course_id

SELECT DISTINCT i.ID, i.name, t.grade
FROM instructor AS i
LEFT JOIN teaches AS e ON i.ID = e.ID
LEFT JOIN section AS s ON e.course_id = s.course_id
LEFT JOIN takes AS t ON e.course_id = t.course_id

SELECT DISTINCT i.ID, i.name, t.grade
FROM instructor AS i
INNER JOIN teaches ON i.ID = teaches.ID
INNER JOIN takes AS t ON teaches.course_id = t.course_id

SELECT DISTINCT i.ID, i.name, t.grade
FROM instructor AS i
INNER JOIN advisor AS a ON i.ID = a.i_ID
INNER JOIN takes AS t ON a.s_ID = t.ID
```

```sql
SELECT DISTINCT i.ID, i.name
FROM instructor AS i
INNER JOIN teaches AS e ON i.ID = e.ID
INNER JOIN advisor AS a ON e.ID = a.i_ID
INNER JOIN takes AS t ON a.s_ID = t.ID
GROUP BY i.ID, i.name
HAVING t.grade NOT LIKE 'A%';

SELECT DISTINCT i.ID, i.name
FROM instructor AS i
INNER JOIN advisor AS a ON i.ID = a.i_ID
INNER JOIN teaches AS e ON a.i_ID = e.ID
INNER JOIN takes AS t ON e.course_id = t.course_id
INNER JOIN section AS s ON t.course_id = e.course_id
GROUP BY i.ID, i.name
HAVING t.grade NOT LIKE 'A%';
```