Bases de Données Fall, 2024

TD 4

The cut-and-paste code from this pdf file will work directly on the computer with `postgreSQL`, in case you want to try these queries.

Here are the tables we used in class:

course_id	*title*	*dept_name*	*credits*
BIO-101	Intro. to Biology	Biology	4
BIO-301	Genetics	Biology	4
BIO-399	Computational Biology	Biology	3
CS-101	Intro. to Computer Science	Comp. Sci.	4
CS-190	Game Design	Comp. Sci.	4
CS-315	Robotics	Comp. Sci.	3
CS-319	Image Processing	Comp. Sci.	3
CS-347	Database System Concepts	Comp. Sci.	3
EE-181	Intro. to Digital Systems	Elec. Eng.	3
FIN-201	Investment Banking	Finance	3
HIS-351	World History	History	3
MU-199	Music Video Production	Music	3
PHY-101	Physical Principles	Physics	4

(a) course

id	*name*	*dept_name*	*salary*
10101	Srinivasan	Comp. Sci.	65000.00
12121	Wu	Finance	90000.00
15151	Mozart	Music	40000.00
22222	Einstein	Physics	95000.00
32343	El Said	History	60000.00
33456	Gold	Physics	87000.00
45565	Katz	Comp. Sci.	75000.00
58583	Califieri	History	62000.00
76543	Singh	Finance	80000.00
76766	Crick	Biology	72000.00
83821	Brandt	Comp. Sci.	92000.00
98345	Kim	Elec. Eng.	80000.00

(b) teacher

id	*name*	*dept_name*	*tot_cred*
00128	Zhang	Comp. Sci.	102
12345	Shankar	Comp. Sci.	32
19991	Brandt	History	80
23121	Chavez	Finance	110
44553	Peltier	Physics	56
45678	Levy	Physics	46
54321	Williams	Comp. Sci.	54
55739	Sanchez	Music	38
70557	Snow	Physics	0
76543	Brown	Comp. Sci.	58
76653	Aoi	Elec. Eng.	60
98765	Bourikas	Elec. Eng.	98
98988	Tanaka	Biology	120

(c) student

course_id	*sec_id*	*semester*	*year*	*building*	*rn*	*time_id*
BIO-101	1	Summer	2009	Painter	514	B
BIO-301	1	Summer	2010	Painter	514	A
CS-101	1	Fall	2009	Packard	101	H
CS-101	1	Spring	2010	Packard	101	F
CS-190	1	Spring	2009	Taylor	3128	E
CS-190	2	Spring	2009	Taylor	3128	A
CS-315	1	Spring	2010	Watson	120	D
CS-319	1	Spring	2010	Watson	100	B
CS-319	2	Spring	2010	Taylor	3128	C
CS-347	1	Fall	2009	Taylor	3128	A
EE-181	1	Spring	2009	Taylor	3128	C
FIN-201	1	Spring	2010	Packard	101	B
HIS-351	1	Spring	2010	Painter	514	C
MU-199	1	Spring	2010	Packard	101	D
PHY-101	1	Fall	2009	Watson	100	A

(d) section

id	*course_id*	*sec_id*	*semester*	*year*
10101	CS-101	1	Fall	2009
10101	CS-315	1	Spring	2010
10101	CS-347	1	Fall	2009
12121	FIN-201	1	Spring	2010
15151	MU-199	1	Spring	2010
22222	PHY-101	1	Fall	2009
32343	HIS-351	1	Spring	2010
45565	CS-101	1	Spring	2010
45565	CS-319	1	Spring	2010
76766	BIO-101	1	Summer	2009
76766	BIO-301	1	Summer	2010
83821	CS-190	1	Spring	2009
83821	CS-190	2	Spring	2009
83821	CS-319	2	Spring	2010
98345	EE-181	1	Spring	2009

(e) teaches

id	*course_id*	*sec_id*	*semester*	*year*	*grade*
00128	CS-101	1	Fall	2009	A
00128	CS-347	1	Fall	2009	A-
12345	CS-101	1	Fall	2009	C
12345	CS-190	2	Spring	2009	A
12345	CS-315	1	Spring	2010	A
12345	CS-347	1	Fall	2009	A
19991	HIS-351	1	Spring	2010	B
23121	FIN-201	1	Spring	2010	C+
44553	PHY-101	1	Fall	2009	B-
45678	CS-101	1	Fall	2009	F
45678	CS-101	1	Spring	2010	B+
45678	CS-319	1	Spring	2010	B
54321	CS-101	1	Fall	2009	A-
54321	CS-190	2	Spring	2009	B+
55739	MU-199	1	Spring	2010	A-
76543	CS-101	1	Fall	2009	A
76543	CS-319	2	Spring	2010	A
76653	EE-181	1	Spring	2009	C
98765	CS-101	1	Fall	2009	C-
98765	CS-315	1	Spring	2010	B
98988	BIO-101	1	Summer	2009	A
98988	BIO-301	1	Summer	2010	

(f) takes

dept_name	*building*	*budget*
Biology	Watson	90000.00
Comp. Sci.	Taylor	100000.00
Elec. Eng.	Taylor	85000.00
Finance	Painter	120000.00
History	Painter	50000.00
Music	Packard	80000.00
Physics	Watson	70000.00

(g) department

1. List all pairs of students who have taken at least one class together.

   Énumérez tous les couples d'étudiants qui ont suivi au moins un cours ensemble.

```
SELECT distinct S1.name, S2.name
FROM student as S1, takes as T1,
     student as S2, takes as T2
WHERE S1.id = T1.id AND
      S2.id = T2.id AND
      T1.course_id = T2.course_id AND
      T1.sec_id = T2.sec_id AND
      T1.semester = T2.semester AND
      T1.year = T2.year AND
      S1.id <> S2.id AND
      S1.name < S2.name;
```

name	*name*
Bourikas	Brown
Bourikas	Levy
Bourikas	Shankar
Bourikas	Williams
Bourikas	Zhang
Brown	Levy
Brown	Shankar
Brown	Williams
Brown	Zhang
Levy	Shankar
Levy	Williams
Levy	Zhang
Shankar	Williams
Shankar	Zhang
Williams	Zhang

2. List all pairs of students who have taken at least two classes together.

   Énumérez tous les couples d'étudiants qui ont suivi au moins deux cours ensemble.

```
SELECT distinct S1.name, S2.name
FROM student as S1, takes as T1,
     student as S2, takes as T2
WHERE S1.id = T1.id AND
      S2.id = T2.id AND
      T1.course_id = T2.course_id AND
      T1.sec_id = T2.sec_id AND
      T1.semester = T2.semester AND
      T1.year = T2.year AND
      S1.id <> S2.id AND
      S1.name < S2.name
GROUP BY S1.id, S2.id
HAVING COUNT(*) >= 2;
```

name	*name*
Shankar	Zhang
Shankar	Williams
Bourikas	Shankar

3. Find the name and budget of all departments with the maximum budget (among all departments).

   Trouvez le nom et le budget de tous les départements avec le budget maximum (parmi toutes les départements).

```
SELECT dept_name, budget
FROM department
WHERE budget IN
        (SELECT max(budget)
         FROM department);
```

dept_name	*budget*
Finance	120000.00

4. Find the total number of courses taught in each building, from Fall 2009 to Spring 2010.
   Trouvez le nombre total de cours enseignés dans chaque bâtiment, de l'automne 2009 au printemps 2010.

```
SELECT building, count(*)
FROM section
WHERE (semester, year) IN ( ('Fall', 2009), ('Spring', 2010) )
GROUP BY building;
```

building	*count*
Packard	4
Painter	1
Taylor	2
Watson	3

5. Find the lowest, across all departments, of the per-department maximum salary.
   Trouvez le salaire maximum le plus bas par département, tous départements confondus.

```
SELECT min(maxsalary)
FROM (SELECT dept_name, max(salary) as maxsalary
      FROM teacher
      GROUP BY dept_name) as mytable;
```

min
40000.00

6. Find the names and salaries of all teachers that earn more than the average salary.
   Trouvez les noms et les salaires de tous les enseignants qui gagnent plus que le salaire moyen.

```
SELECT A.name,  A.salary
FROM teacher as A
WHERE A.salary >  (SELECT avg(B.salary)
                   FROM teacher as B);
```

name	*salary*
Wu	90000.00
Einstein	95000.00
Gold	87000.00
Katz	75000.00
Singh	80000.00
Brandt	92000.00
Kim	80000.00

7. For each teacher, list all the student names of those students who have taken at least 2 classes from that teacher. **You must write this query using HAVING.**

   Pour chaque enseignant, listez tous les noms des élèves qui ont suivi au moins 2 cours de cet enseignant. **Vous devez écrire cette requête en utilisant HAVING.**

```
SELECT teacher.name, student.name, count(*)
FROM teacher, student, takes, teaches
WHERE teacher.id = teaches.id and student.id = takes.id and (
   takes.course_id, takes.sec_id, takes.semester, takes.year) = (
   teaches.course_id, teaches.sec_id, teaches.semester, teaches.
   year)
GROUP BY teacher.name, student.name
HAVING count(*) >= 2;
```

teachername	*studentname*	*totalcount*
Crick	Tanaka	2
Katz	Levy	2
Srinivasan	Bourikas	2
Srinivasan	Shankar	3
Srinivasan	Zhang	2

8. For each teacher, list all the student names of those students who have taken at least 2 classes from that teacher. **You must write this query without HAVING**.

   Pour chaque enseignant, listez tous les noms des élèves qui ont suivi au moins 2 cours de cet enseignant. **Vous devez écrire cette requête sans HAVING**.

```
SELECT T.teachername, T.studentname, T.totalcount
FROM   (SELECT teacher.name as teachername, student.name as
   studentname, count(*) as totalcount
        FROM teacher, student, takes, teaches
        WHERE teacher.id = teaches.id and student.id = takes.id
   and (takes.course_id, takes.sec_id, takes.semester, takes.year
   ) = (teaches.course_id, teaches.sec_id, teaches.semester,
   teaches.year)
        GROUP BY teacher.name, student.name) as T
WHERE T.totalcount >= 2
ORDER BY T.teachername;
```

teachername	*studentname*	*totalcount*
Crick	Tanaka	2
Katz	Levy	2
Srinivasan	Bourikas	2
Srinivasan	Shankar	3
Srinivasan	Zhang	2