

BDA TP2

Ex1

- 1) select dept_name from department where budget in (select max(budget) from department)
- 2) select name, salary from teacher t where salary > (select avg(salary) from teacher)
- 3)

```
SELECT t.ID AS teacher_id,
       t.name AS teacher_name,
       s.ID AS student_id,
       s.name AS student_name,
       COUNT(takes.course_id) AS total_courses_taken
FROM teacher t
JOIN teaches ON t.ID = teaches.ID
JOIN takes ON teaches.course_id = takes.course_id
            AND teaches.semester = takes.semester
            AND teaches.year = takes.year
JOIN student s ON takes.ID = s.ID
GROUP BY t.ID, t.name, s.ID, s.name
HAVING COUNT(takes.course_id) > 1
ORDER BY t.ID, s.ID;
```
- 4)

```
WITH course_counts AS (
  SELECT t.ID AS teacher_id,
         t.name AS teacher_name,
         s.ID AS student_id,
         s.name AS student_name,
         COUNT(takes.course_id) AS total_courses_taken
  FROM teacher t
  JOIN teaches ON t.ID = teaches.ID
  JOIN takes ON teaches.course_id = takes.course_id
              AND teaches.sec_id = takes.sec_id
              AND teaches.semester = takes.semester
              AND teaches.year = takes.year
  JOIN student s ON takes.ID = s.ID
  GROUP BY t.ID, t.name, s.ID, s.name
)
SELECT *
FROM course_counts
WHERE total_courses_taken > 2
ORDER BY teacher_id, student_id;
```
- 5)

```
select distinct s.id, s.name
```

from student s join takes t on s.id= t.id
 where t.id in (select id from takes where year <2010) and t.id not in (
 select id from takes where year >2010)

6) select name from teacher where name like 'E%'

7)

```
SELECT name, salary
FROM (
  SELECT name, salary, DENSE_RANK() OVER (ORDER BY salary DESC) AS rnk
  FROM teacher
) ranked
WHERE rnk = 4;
```

8)

```
SELECT name, salary
FROM (
  SELECT name, salary, DENSE_RANK() OVER (ORDER BY salary) AS rnk
  FROM teacher
) ranked
WHERE rnk <=3
ORDER BY salary DESC
```

9) select distinct id, name from student where id in (select id from takes where year = 2009 and semester = 'Fall')

10) select distinct id, name from student where id = some (select id from takes where year = 2009 and semester = 'Fall')

11) select distinct id, name from student s natural inner join takes t where t.year = 2009 and t.semester = 'Fall'

12) select distinct id, name from student where exists (select id from takes where year = 2009 and semester = 'Fall')

13)

```
select distinct t1.id, t2.id from takes t1 join takes t2 on
t1.course_id=t2.course_id
and t1.semester=t2.semester
and t1.year=t2.year
where not t1.id = t2.id
order by t1.id
```

14)

```
select T.name, count(ta.id) x from teacher T join teaches te on T.id = te.id
join takes ta on
  te.course_id = ta.course_id
  and te.semester=ta.semester
  and te.year=ta.year
join student s on s.id = ta.id
group by t.name
order by x desc
```

15)

```
select T.name, coalesce(count(ta.id),0) x from teacher T
left join teaches te on T.id = te.id
left join takes ta on
    te.course_id = ta.course_id
    and te.semester=ta.semester
    and te.year=ta.year
left join student s on s.id = ta.id
group by t.name
order by x desc
```

16)

```
select t.name, count(ta.grade) x
from teacher t join teaches te on t.id = te.id
join takes ta on te.course_id = ta.course_id
where grade = 'A'
group by t.name
```

17)

```
select t.name, s.name, count(ta.id) from teacher t join teaches te on t.id =te.id
join takes ta on te.course_id = ta.course_id
join student s on ta.id = s.id
group by t.name, s.name
order by t.name
```

18)

```
select t.name, s.name, count(ta.id) from teacher t join teaches te on t.id =te.id
join takes ta on te.course_id = ta.course_id
join student s on ta.id = s.id
group by ta.id, t.name, s.name
having count(ta.id)>=2
order by t.name
```

Ex2

- 1) 2NF
- 2) BCNF
- 3) 1NF

Ex3

- 1) 4 par l'énoncé, $A \rightarrow BC$ et $E \rightarrow A \Leftrightarrow E \rightarrow BC$,