

# TP1 – BDA

Ex 1 :

```
44
45 CREATE TABLE section
46 (course_id VARCHAR(10),
47  sec_id VARCHAR(10),
48  semester VARCHAR(10),
49  Section_year NUMERIC(4,0),
50  building VARCHAR(15),
51  room_number VARCHAR(10),
52  time_slot_id VARCHAR(10),
53  CHECK (semester IN ('Fall', 'Winter', 'Spring', 'Summer')),
54  PRIMARY KEY (course_id, sec_id, semester, Section_year),
55  FOREIGN KEY (course_id) REFERENCES course ON DELETE CASCADE,
56  FOREIGN KEY (building, room_number) REFERENCES classroom ON DELETE CASCADE
57 );
58
```

Résultats:

Script

TP1\_BDA

Status

Complete

View

Detail

Summary

Rows

50

Go

Create App

Edit Script

Number %	Elapsed	Statement	Feedback	Rows
1	0.91	DROP TABLE classroom CASCADE CONSTRAINT	Table dropped.	0
2	0.88	DROP TABLE department CASCADE CONSTRAINT	Table dropped.	0
3	0.87	DROP TABLE course CASCADE CONSTRAINT	Table dropped.	0
4	0.85	DROP TABLE teacher CASCADE CONSTRAINT	Table dropped.	0
5	0.88	DROP TABLE section CASCADE CONSTRAINT	Table dropped.	0
6	0.85	DROP TABLE teaches CASCADE CONSTRAINT	Table dropped.	0
7	0.84	DROP TABLE student CASCADE CONSTRAINT	Table dropped.	0
8	0.84	DROP TABLE takes CASCADE CONSTRAINT	Table dropped.	0
9	0.86	DROP TABLE supervisor CASCADE CONSTRAINT	Table dropped.	0
10	0.85	DROP TABLE time_slot CASCADE CONSTRAINT	Table dropped.	0
11	0.84	DROP TABLE prereq CASCADE CONSTRAINT	Table dropped.	0
12	0.05	CREATE TABLE classroom (building VARCHAR(15), room	Table created.	0
13	0.05	CREATE TABLE department (dept_name VARCHAR(20), b	Table created.	0
14	0.05	CREATE TABLE course (course_id VARCHAR(10), title	Table created.	0
15	0.05	CREATE TABLE teacher (teacher_id VARCHAR(10), first	Table created.	0
16	0.05	CREATE TABLE section (course_id VARCHAR(10), sec_id	Table created.	0
17	0.05	CREATE TABLE teaches (teacher_id VARCHAR(10), sec_id	Table created.	0
18	0.05	CREATE TABLE student (student_id VARCHAR(10), last	Table created.	0
19	0.05	CREATE TABLE takes (takes_id VARCHAR(10), sec_id	Table created.	0
20	0.05	CREATE TABLE supervisor (supervisor_id VARCHAR(10),	Table created.	0
21	0.02	CREATE TABLE time_slot (time_slot_id VARCHAR(10),	Table created.	0
22	0.05	CREATE TABLE prereq (course_id VARCHAR(10), prereq	Table created.	0

Download

Row(s) 1 - 22 of 22

Ex 3 :

SQL Scripts \ Results

Script: TPL\_BDA\_Data

Status: Complete

Create App

Edit Script

View: Detail Summary

Rows: 15

Go

Number ↑	Elapsed	Statement	Feedback	Rows
1	0.04	delete from prereq	0 row(s) deleted.	0
2	0.02	delete from time_slot	0 row(s) deleted.	0
3	0.02	delete from supervisor	0 row(s) deleted.	0
4	0.02	delete from takes	0 row(s) deleted.	0
5	0.02	delete from student	0 row(s) deleted.	0
6	0.02	delete from teaches	0 row(s) deleted.	0
7	0.02	delete from section	0 row(s) deleted.	0
8	0.01	delete from teacher	0 row(s) deleted.	0
9	0.01	delete from course	0 row(s) deleted.	0
10	0.01	delete from department	0 row(s) deleted.	0
11	0.01	delete from classroom	0 row(s) deleted.	0
12	6.90	insert into classroom values ('Packard', '101', '500')	1 row(s) inserted.	1
13	0.01	insert into classroom values ('Painter', '514', '10')	1 row(s) inserted.	1
14	0.00	insert into classroom values ('Taylor', '3128', '70')	1 row(s) inserted.	1
15	0.01	insert into classroom values ('Watson', '100', '30')	1 row(s) inserted.	1

Download

row(s) 1 - 15 of 149

Next

149

Statements Processed

149

Successful

0

With Errors

# Exercice 2 :

## Q.1) Description de la table Section

### DESC section

1 DESC section

Results	Explain	Describe	Saved SQL	History					
Object Type		TABLE ?	Object SECTION ?						
Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
SECTION	COURSE_ID	VARCHAR2	10	-	-	1	-	-	-
	SEC_ID	VARCHAR2	10	-	-	2	-	-	-
	SEMESTER	VARCHAR2	10	-	-	3	-	-	-
	SECTION_YEAR	NUMBER	-	4	0	4	-	-	-
	BUILDING	VARCHAR2	15	-	-	-	✓	-	-
	ROOM_NUMBER	VARCHAR2	10	-	-	-	✓	-	-
	TIME_SLOT_ID	VARCHAR2	10	-	-	-	✓	-	-

### Éléments de la table Section

## Q.2) SELECT \* FROM course;

1

2

SELECT \* FROM course;

Results

Explain

Describe

Saved SQL

History

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

More than 10 rows available. Increase rows selector to view more rows.

10 rows returned in 0.00 seconds

Download

**Q.3)**  
**SELECT TITLE,DEPT\_NAME**  
**FROM COURSE**

TITLE	DEPT_NAME
Intro. to Biology	Biology
Genetics	Biology
Computational Biology	Biology
Intro. to Computer Science	Comp. Sci.
Game Design	Comp. Sci.
Robotics	Comp. Sci.
Image Processing	Comp. Sci.
Database System Concepts	Comp. Sci.
Intro. to Digital Systems	Elec. Eng.
Investment Banking	Finance
World History	History
Music Video Production	Music
Physical Principles	Physics

**Q.4) SELECT DEPT\_NAME,BUDGET**  
**FROM DEPARTMENT**

DEPT_NAME	BUDGET
Biology	90000
Comp. Sci.	100000
Elec. Eng.	85000
Finance	120000
History	50000
Music	80000
Physics	70000

**Q.5) SELECT TEACHER\_NAME,DEPT\_NAME**  
**FROM TEACHER**

TEACHER_NAME	DEPT_NAME
Srinivasan	Comp. Sci.
Wu	Finance
Mozart	Music
Einstein	Physics
El Said	History
Gold	Physics
Katz	Comp. Sci.
Califieri	History
Singh	Finance
Crick	Biology
Brandt	Comp. Sci.
Kim	Elec. Eng.

**Q.6)**  
**SELECT TEACHER\_NAME**  
**FROM TEACHER**  
**WHERE SALARY > 65000**

TEACHER_NAME
Wu
Einstein
Gold
Katz
Singh
Crick
Brandt
Kim

**Q.7) SELECT TEACHER\_NAME  
FROM TEACHER  
WHERE SALARY BETWEEN 55000 AND 85000**

TEACHER_NAME
Srinivasan
El Said
Katz
Califieri
Singh
Crick
Kim

**Q.8) SELECT DISTINCT DEPT\_NAME  
FROM TEACHER**

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

**Q.9)  
SELECT TEACHER\_NAME  
FROM TEACHER  
WHERE DEPT\_NAME = 'Comp. Sci.' AND SALARY > 65000**

Prof Riche
Katz
Brandt

**Q10) SELECT \***  
**FROM SECTION**  
**WHERE SECTION\_YEAR = '2010' AND SEMESTER = 'Spring'**

COURSE_ID	SEC_ID	SEMESTER	SECTION_YEAR	BUILDING	ROOM_NUMBER	TIME_SLOT_ID
CS-101	1	Spring	2010	Packard	101	F
CS-315	1	Spring	2010	Watson	120	D
CS-319	1	Spring	2010	Watson	100	B
CS-319	2	Spring	2010	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

**Q.11) SELECT TITLE**  
**FROM COURSE**  
**WHERE CREDITS > 3 AND DEPT\_NAME = 'Comp. Sci.'**

TITLE
Intro. to Computer Science
Game Design

**Q.12)**

Theta-jointure : **SELECT TEACHER\_NAME,TEACHER.DEPT\_NAME,BUILDING**  
**FROM TEACHER**  
**JOIN DEPARTMENT**  
**ON TEACHER.DEPT\_NAME = DEPARTMENT.DEPT\_NAME**

Jointure normal :

**SELECT teacher.teacher\_name , teacher . dept\_name , department . building**  
**FROM teacher , department**  
**WHERE teacher . dept\_name = department . dept\_name ;**

TEACHER_NAME	DEPT_NAME	BUILDING
Srinivasan	Comp. Sci.	Taylor
Wu	Finance	Painter
Mozart	Music	Packard
Einstein	Physics	Watson
El Said	History	Painter
Gold	Physics	Watson
Katz	Comp. Sci.	Taylor
Califieri	History	Painter
Singh	Finance	Painter
Crick	Biology	Watson
Brandt	Comp. Sci.	Taylor
Kim	Elec. Eng.	Taylor

Q13)

```
SELECT DISTINCT STUDENT_NAME
FROM STUDENT,DEPARTMENT,COURSE
WHERE STUDENT.DEPT_NAME = 'Comp. Sci.' AND STUDENT.DEPT_NAME =
DEPARTMENT.DEPT_NAME AND DEPARTMENT.DEPT_NAME =
COURSE.DEPT_NAME =>Ne marche pas
```

```
SELECT DISTINCT STUDENT_NAME
FROM STUDENT, TAKES, SECTION, COURSE
WHERE
(
STUDENT.STUDENT_ID = TAKES.TAKES_ID AND TAKES.SEC_ID =
SECTION.SEC_ID AND SECTION.COURSE_ID = COURSE.COURSE_ID
AND COURSE.DEPT_NAME = 'Comp. Sci.'
) => Ne marche pas, produit cartésien redondant avec SECTION et COURSE
```

!!! Dans section, il y a plusieurs fois le même cours qui ont lieu a des moment différents (Printemps, Été ...)

```
SELECT DISTINCT STUDENT_NAME,TITLE
FROM STUDENT, TAKES, COURSE
WHERE
(
STUDENT.STUDENT_ID = TAKES.TAKES_ID -- On select l'ens des étudiant ayant au
moins chosi une matière
AND TAKES.COURSE_ID = COURSE.COURSE_ID -- À partir de l'ens précédent, on
select en plus les cours qui ont vraiment eu lieu (Certain cours n'ont pas été assuré)
AND COURSE.DEPT_NAME = 'Comp. Sci.' -- On restreint aux cours d'informatique
)
```

STUDENT_NAME	TITLE
Zhang	Intro. to Computer Science
Williams	Game Design
Shankar	Game Design
Levy	Image Processing
Zhang	Database System Concepts
Shankar	Intro. to Computer Science
Shankar	Robotics
Shankar	Database System Concepts
Brown	Intro. to Computer Science
Brown	Image Processing
Williams	Intro. to Computer Science
Bourikas	Intro. to Computer Science
Levy	Intro. to Computer Science
Bourikas	Robotics

Q14)

```
SELECT DISTINCT STUDENT_NAME FROM STUDENT,TAKES,TEACHES,TEACHER
WHERE (
    STUDENT.STUDENT_ID = TAKES.TAKES_ID → Select les étudiants
    AND TAKES.COURSE_ID = TEACHES.COURSE_ID → On select les cours qui ont été
    assuré par des profs
    AND TEACHES.TEACHES_ID = TEACHER.TEACHER_ID → On select les profs
    AND TEACHER_NAME = 'Einstein'
)
```

STUDENT_NAME
Peltier

Q15) **SELECT DISTINCT SEC\_ID,TEACHER\_NAME FROM SECTION,TEACHER**  
**WHERE SECTION.SEC\_ID = TEACHER.TEACHER\_ID => (No data found) ==> Il y a**  
**plus simple**

Accède à la table des profs (Instructor dans le schéma) de là tu va faire une jointure avec la table des cours enseigné (Teaches) →

```
SELECT DISTINCT TEACHER_NAME,TEACHES_ID FROM TEACHER,TEACHES
WHERE TEACHER_ID = TEACHES_ID
```

Remarques : Ne pas mettre **DISTINCT**, car il est normal que les profs puissent assurer plusieurs cours.

TEACHER_NAME	COURSE_ID
Srinivasan	CS-101
Srinivasan	CS-315
Srinivasan	CS-347
Wu	FIN-201
Mozart	MU-199
Einstein	PHY-101
El Said	HIS-351
Katz	CS-101
Katz	CS-319
Crick	BIO-101
Crick	BIO-301
Brandt	CS-190
Brandt	CS-190
Brandt	CS-319
Kim	EE-181



**Q16) SELECT  
TAKES.COURSE\_ID,TAKES.SEC\_ID,TAKES.SEMESTER,TAKES.TAKES\_YEAR,count  
(\*) AS Nb\_inscrit FROM takes  
WHERE TAKES.SEMESTER = 'Spring' and TAKES.TAKES\_YEAR = 2010  
GROUP BY  
TAKES.COURSE\_ID,TAKES.SEC\_ID,TAKES.SEMESTER,TAKES.TAKES\_YEAR;**

COURSE_ID	SEC_ID	SEMESTER	TAKES_YEAR	NB_INSCRIT
CS-315	1	Spring	2010	2
CS-101	1	Spring	2010	1
CS-319	2	Spring	2010	1
CS-319	1	Spring	2010	1
FIN-201	1	Spring	2010	1
HIS-351	1	Spring	2010	1
MU-199	1	Spring	2010	1

**Q17) SELECT DEPT\_NAME,MAX(SALARY) AS MAX\_SALARY FROM TEACHER  
GROUP BY DEPT\_NAME**

DEPT_NAME	MAX_SALARY
Comp. Sci.	92000
Biology	72000
History	62000
Finance	90000
Elec. Eng.	80000
Music	40000
Physics	95000

7 rows returned in 0.03 seconds [Download](#)

Rmq: Arrête systématiquement de faire une jointure quand une table possède les attributs recherché

**Q18) SELECT  
TAKES.COURSE\_ID,TAKES.SEC\_ID,TAKES.SEMESTER,TAKES.TAKES\_YEAR,count  
(\*) AS Nb\_inscrit FROM takes  
GROUP BY  
TAKES.COURSE\_ID,TAKES.SEC\_ID,TAKES.SEMESTER,TAKES.TAKES\_YEAR;**

COURSE_ID	SEC_ID	SEMESTER	TAKES_YEAR	NB_INSCRIT
CS-190	2	Spring	2009	2
CS-319	1	Spring	2010	1
CS-347	1	Fall	2009	2
MU-199	1	Spring	2010	1
CS-101	1	Fall	2009	6
CS-101	1	Spring	2010	1
FIN-201	1	Spring	2010	1
BIO-101	1	Summer	2009	1
CS-315	1	Spring	2010	2
HIS-351	1	Spring	2010	1
PHY-101	1	Fall	2009	1
EE-181	1	Spring	2009	1
BIO-301	1	Summer	2010	1
CS-319	2	Spring	2010	1

**Q19) SELECT BUILDING,count (\*) AS Courses\_number FROM SECTION  
WHERE (SEMESTER,SECTION\_YEAR) IN ( ( 'Fall' , 2009) , ( 'Spring' , 2010) )  
GROUP BY BUILDING;**

BUILDING	COURSES_NUMBER
Watson	3
Packard	4
Taylor	2
Painter	1

**Q20)**

**Q21)**

**Q22) SELECT SECTION.SEMESTER,count(\*) AS Courses\_number FROM SECTION GROUP BY SECTION.SEMESTER;**

SEMESTER	COURSES_NUMBER
Fall	3
Summer	2
Spring	10

3 rows returned in 0.01 seconds [Download](#)

**Q23) SELECT student.student\_name,sum( credits )  
FROM student , course , takes  
WHERE student.student\_id = takes.takes\_id and takes. course\_id = course .  
course\_id and student.dept\_name != course.dept\_name  
GROUP BY student.student\_name;**

STUDENT_NAME	SUM(CREDITS)
Levy	11
Bourikas	7

**Q23)**

**SELECT section.building , sum ( course.credits ) FROM section , course  
WHERE section.course\_id = course.course\_id GROUP BY section.building ;**

BUILDING	SUM(COURSE.CREDITS)
Watson	10
Packard	14
Taylor	17
Painter	11