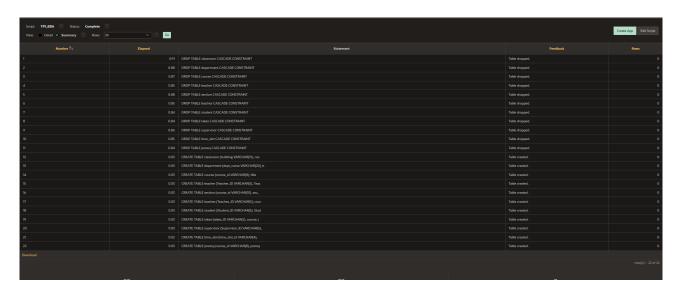
TP1 – BDA

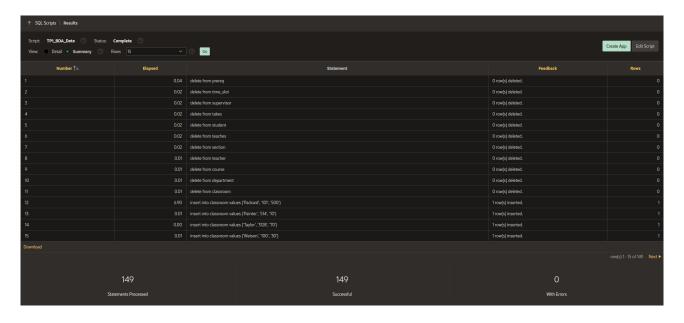
Ex 1:

```
44
45 CREATE TABLE section
46 (course_id VARCHAR(10),
47 sec_id VARCHAR(10),
48 semester VARCHAR(10),
59 building VARCHAR(15),
50 room_number VARCHAR(10),
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 );
```

Résultats:



Ex 3:



Exercice 2:

Q.1) Déscription de la table Section

DESC section

1 DE	1 DESC section									
						•				
Results	Explain Describe	e Saved SQI	. History	/						
Obje	ct Type TABLE (Object	SECTION (
Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment	
SECTION	COURSE_ID	VARCHAR2	10							
	SEC_ID	VARCHAR2	10			2				
	SEMESTER	VARCHAR2	10							
	SECTION_YEAR	NUMBER		4	0	4				
	BUILDING	VARCHAR2	15				S/			
	ROOM_NUMBER	VARCHAR2	10				 ✓			
	TIME_SLOT_ID	VARCHAR2	10				s/			

Éléments de la table Section

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

sults Explain Describe	Saved SQL History		
COURSE_ID	TITLE	DEPT_NAME	CREDITS
IO-101	Intro. to Biology	Biology	
IO-301	Genetics	Biology	4
10-399	Computational Biology	Biology	
S-101	Intro. to Computer Science	Comp. Sci.	4
S-190	Game Design	Comp. Sci.	
S-315	Robotics	Comp. Sci.	
S-319	Image Processing	Comp. Sci.	
S-347	Database System Concepts	Comp. Sci.	
E-181	Intro. to Digital Systems	Elec. Eng.	
IN-201	Investment Banking	Finance	

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	

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		Spring	2010	Packard	101	F
CS-315	1	Spring	2010	Watson	120	D
CS-319		Spring	2010	Watson	100	В
CS-319	2	Spring	2010	Taylor	3128	С
FIN-201		Spring	2010	Packard	101	В
HIS-351	1	Spring	2010	Painter	514	С
MU-199		Spring	2010	Packard	101	D

Q.11) SELECT TITLE FROM COURSE WHERE CREDITS > 3 AND DEPT_NAME = 'Comp. Sci.'



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 cartesien redondant avec SECTION et COURSE
!!! Dans section, il y a plusieurs fois le même cours qui ont leiu 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 précedent, on
select en plus les cours qui ont vraiment eu lieu (Certaint 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

)

SELECT DISTINCT STUDENT_NAME FROM STUDENT, TAKES, TEACHES, TEACHER WHERE (

STUDENT_ID = TAKES.TAKES_ID → Select les étudiants **AND TAKES.COURSE_ID = TEACHES.COURSE_ID** → On select les cours qui ont eté

assuré par des profs

AND TEACHES_ID = TEACHER.TEACHER_ID → On select les profs

AND TEACHER_NAME = 'Einstein'

Peltier STUDENT_NAME

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		Spring	2010	2
CS-101	1	Spring	2010	1
CS-319	2	Spring	2010	
CS-319	1	Spring	2010	1
FIN-201		Spring	2010	
HIS-351	1	Spring	2010	1
MU-199		Spring	2010	

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: Arrète 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		Spring	2010	
CS-347		Fall	2009	
MU-199		Spring	2010	1
CS-101		Fall	2009	
CS-101		Spring	2010	
FIN-201		Spring	2010	
BIO-101		Summer	2009	
CS-315		Spring	2010	
HIS-351		Spring	2010	
PHY-101		Fall	2009	
EE-181		Spring	2009	
BIO-301		Summer	2010	
CS-319	2	Spring	2010	

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