TP1 – BDA

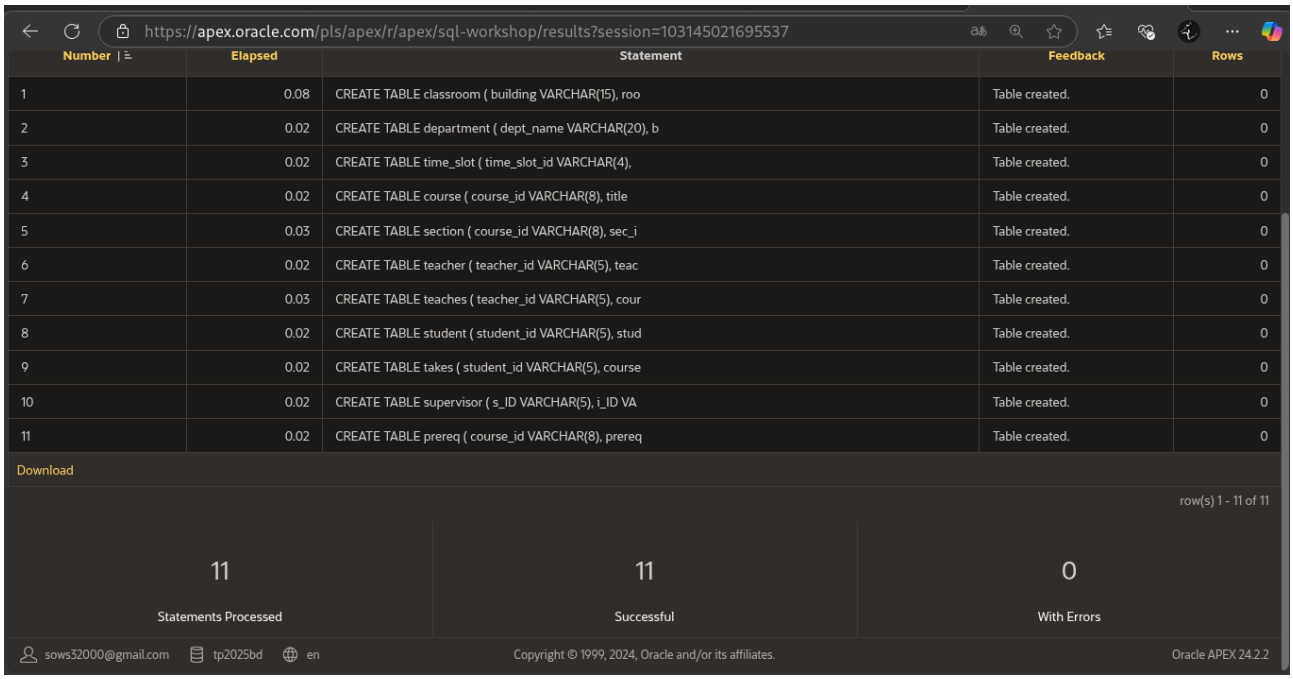
Baba SOW **ING – INFO2**

2024/2025

Exercic1 :

Script de création et de peuplement => voir repository.

Création :



https://apex.oracle.com/pls/apex/r/apex/sql-workshop/results?session=103145021695537

Number	Elapsed	Statement	Feedback	Rows
1	0.08	CREATE TABLE classroom (building VARCHAR(15), roo	Table created.	0
2	0.02	CREATE TABLE department (dept_name VARCHAR(20), b	Table created.	0
3	0.02	CREATE TABLE time_slot (time_slot_id VARCHAR(4),	Table created.	0
4	0.02	CREATE TABLE course (course_id VARCHAR(8), title	Table created.	0
5	0.03	CREATE TABLE section (course_id VARCHAR(8), sec_i	Table created.	0
6	0.02	CREATE TABLE teacher (teacher_id VARCHAR(5), teac	Table created.	0
7	0.03	CREATE TABLE teaches (teacher_id VARCHAR(5), cour	Table created.	0
8	0.02	CREATE TABLE student (student_id VARCHAR(5), stud	Table created.	0
9	0.02	CREATE TABLE takes (student_id VARCHAR(5), course	Table created.	0
10	0.02	CREATE TABLE supervisor (s_ID VARCHAR(5), i_ID VA	Table created.	0
11	0.02	CREATE TABLE prereq (course_id VARCHAR(8), prereq	Table created.	0

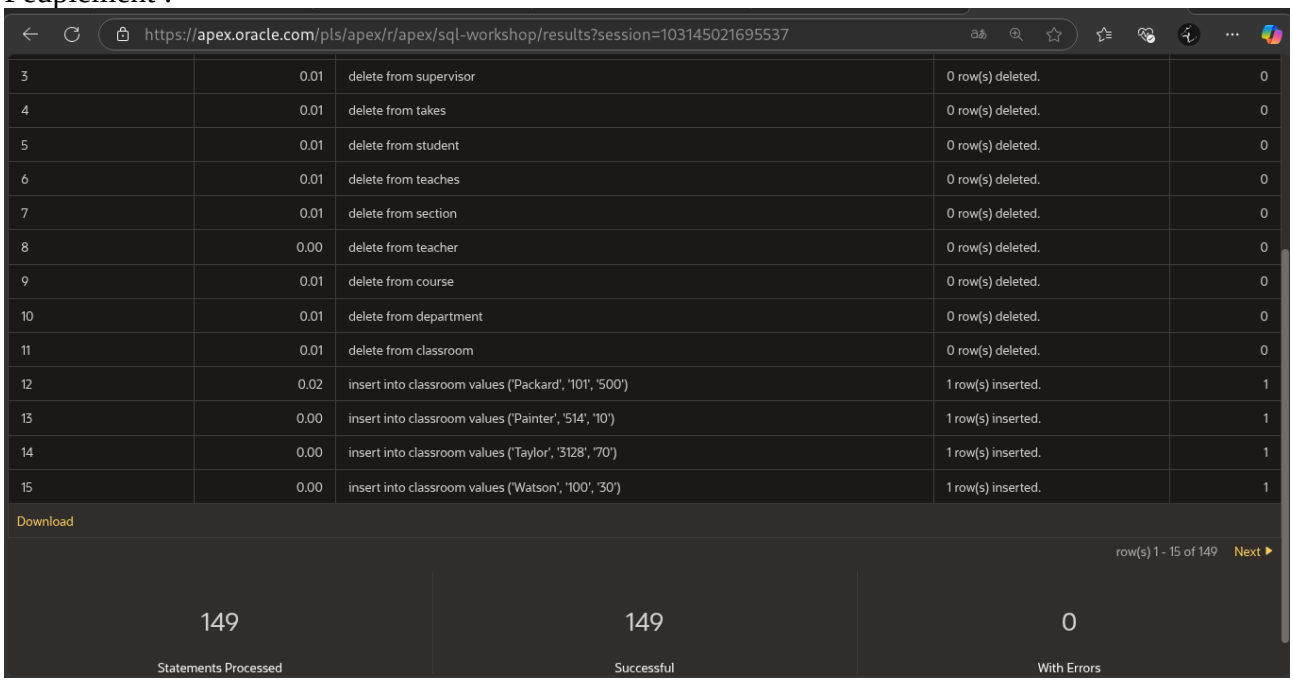
Download

row(s) 1 - 11 of 11

11	11	0
Statements Processed	Successful	With Errors

sows32000@gmail.com tp2025bd en Copyright © 1999, 2024, Oracle and/or its affiliates. Oracle APEX 24.2.2

Peuplement :



https://apex.oracle.com/pls/apex/r/apex/sql-workshop/results?session=103145021695537

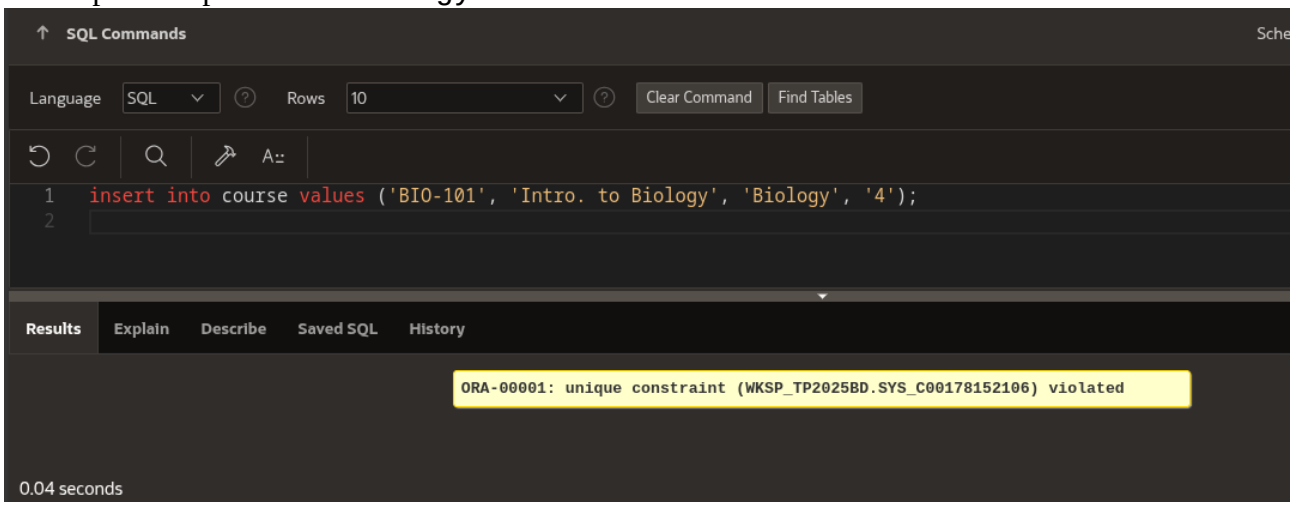
3	0.01	delete from supervisor	0 row(s) deleted.	0
4	0.01	delete from takes	0 row(s) deleted.	0
5	0.01	delete from student	0 row(s) deleted.	0
6	0.01	delete from teaches	0 row(s) deleted.	0
7	0.01	delete from section	0 row(s) deleted.	0
8	0.00	delete from teacher	0 row(s) deleted.	0
9	0.01	delete from 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	0.02	insert into classroom values ('Packard', '101', '500')	1 row(s) inserted.	1
13	0.00	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	0.00	insert into classroom values ('Watson', '100', '30')	1 row(s) inserted.	1

Download

row(s) 1 - 15 of 149 Next ▶

149	149	0
Statements Processed	Successful	With Errors

4. Insérer un nouveau cours dont l'identifiant est `BIO-101`, intitulé `Intro. to Biology`, assuré par le département `Biology` et son crédit est de 4.



C'est normale car, on a fixé comme contraintre concernant le crédit, qu'il doit être entre 0 et 2.

Exercice 2

1. Afficher la structure de la table `section` :

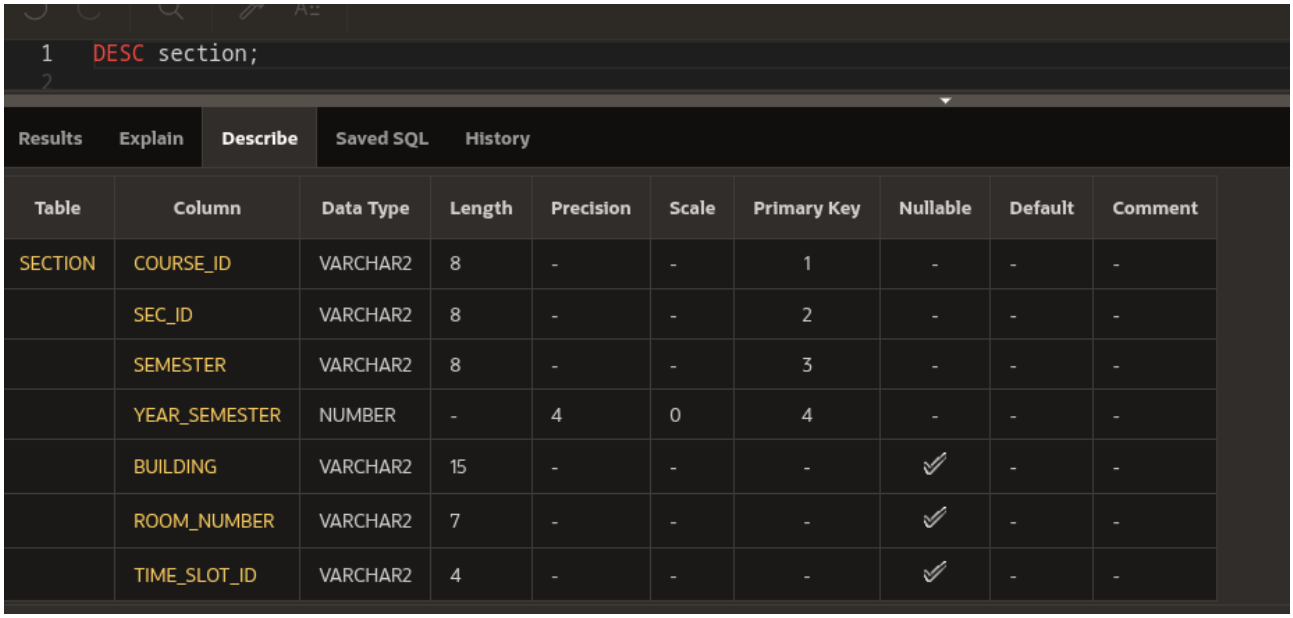


Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
SECTION	COURSE_ID	VARCHAR2	8	-	-	1	-	-	-
	SEC_ID	VARCHAR2	8	-	-	2	-	-	-
	SEMESTER	VARCHAR2	8	-	-	3	-	-	-
	YEAR_SEMESTER	NUMBER	-	4	0	4	-	-	-
	BUILDING	VARCHAR2	15	-	-	-	✓	-	-
	ROOM_NUMBER	VARCHAR2	7	-	-	-	✓	-	-
	TIME_SLOT_ID	VARCHAR2	4	-	-	-	✓	-	-

2. Afficher le contenu de la table `section` :

```
SELECT * FROM section;
```

Results | Explain | Describe | Saved SQL | History

COURSE_ID	SEC_ID	SEMESTER	YEAR_SEMESTER	BUILDING	ROOM_NUMBER	TIME_SLOT_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

3. Afficher les titres des cours et les départements qui les proposent.

```
SELECT title, dept_name
FROM course;
```

Results | Explain | Describe | Saved SQL | History

TITLE	DEPT_NAME
Image Processing	Comp. Sci.
Database System Concepts	Comp. Sci.
Intro. to Digital Systems	Elec. Eng.
Investment Banking	Finance
More than 10 rows available. Increase rows selector to view more rows.	

10 rows returned in 0.00 seconds Download

4. Afficher les noms des départements ainsi que leur budget.

```
SELECT dept_name, budget
FROM department;
```

Results | Explain | Describe | Saved SQL | History

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

5. Afficher tous les noms des enseignants et leur département.

```
1  SELECT teacher_name, dept_name
2  FROM teacher;
3
```

Results | Explain | Describe | Saved SQL | History

TEACHER_NAME	DEPT_NAME
Srinivasan	Comp. Sci.
Wu	Finance
Mozart	Music
Einstein	Physics
El Said	History
Gold	Physics

sows32000@gmail.com | tp2025bd | en |

6. Afficher tous les noms des enseignants ayant un salaire supérieur strictement à 65.000 $.

```
1  SELECT teacher_name
2  FROM teacher
3  WHERE salary > 65000;
```

Results | Explain | Describe | Saved SQL | History

TEACHER_NAME
Einstein
Gold
Katz
Singh
Crick
Brandt
Kim

sows32000@gmail.com | tp2025bd | en |

7. Afficher les noms des enseignants ayant un salaire compris entre 55.000 $ et 85.000 $.

```
1  SELECT teacher_name
2  FROM teacher
3  WHERE salary BETWEEN 55000 AND 85000;
```

Results | Explain | Describe | Saved SQL | History

TEACHER_NAME
Srinivasan
El Said
Katz
Califieri
Singh
Crick
Kim

sows32000@gmail.com | tp2025bd | en |

8. Afficher les noms des départements, en utilisant la relation teacher et éliminer les doublons.

```
SELECT DISTINCT dept_name
FROM teacher;
```

Results | Explain | Describe | Saved SQL | History

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

sows32000@gmail.com | tp2025bd | en | Copyright © 1999, 2024, Oracle and/or its affiliates.

9. Afficher tous les noms des enseignants du département informatique ayant un salaire supérieur strictement à 65.000 $.

```
SELECT teacher_name
FROM teacher
WHERE dept_name = 'Comp. Sci.'
  AND salary > 65000;
```

Results | Explain | Describe | Saved SQL | History

TEACHER_NAME
Katz
Brandt

2 rows returned in 0.01 seconds Download

sows32000@gmail.com | tp2025bd | en | Copyright © 1999, 2024, Oracle and/or its affiliates.

10. Afficher tous les renseignements sur les cours proposés au printemps 2010 (relation section).

```
SELECT *
FROM section
WHERE semester = 'Spring'
  AND year semester = 2010;
```

Results | Explain | Describe | Saved SQL | History

COURSE_ID	SEC_ID	SEMESTER	YEAR_SEMESTER	BUILDING	ROOM_NUMBER
CS-101	1	Spring	2010	Packard	101
CS-315	1	Spring	2010	Watson	120
CS-319	1	Spring	2010	Watson	100
CS-319	2	Spring	2010	Taylor	3128
FIN-201	1	Spring	2010	Packard	101
HIS-351	1	Spring	2010	Painter	514

sows32000@gmail.com | tp2025bd | en | Copyright © 1999, 2024, Oracle and/or its affiliates.

11. Afficher tous les titres des cours dispensés par le département informatique qui ont plus de trois crédits.

```
SELECT title
FROM course
WHERE dept_name = 'Comp. Sci.'
  AND credits > 3;
```

Results | Explain | Describe | Saved SQL | History

TITLE
Intro. to Computer Science
Game Design

2 rows returned in 0.01 seconds Download

sows32000@gmail.com tp2025bd en Copyright © 1999, 2024, Oracle and/or its affiliates.

12. Afficher tous les noms des enseignants ainsi que le nom de leur département et les noms des bâtiments qui les hébergent.

```
SELECT teacher.teacher_name, teacher.dept_name, department.building
FROM teacher
JOIN department ON teacher.dept_name = department.dept_name;
```

Results | Explain | Describe | Saved SQL | History

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

13. Afficher tous les étudiants ayant suivi au moins un cours en informatique.

```
SELECT DISTINCT student.student_name
FROM student
JOIN takes ON student.student_id = takes.student_id
JOIN section ON takes.course_id = section.course_id
            AND takes.sec_id = section.sec_id
            AND takes.semester = section.semester
            AND takes.semester_year = section.year_semester
JOIN course ON section.course_id = course.course_id
WHERE course.dept_name = 'Comp. Sci.';
```

Results | Explain | Describe | Saved SQL | History

STUDENT_NAME
Brown
Zhang
Levy
Bourikas
Shankar
Williams

6 rows returned in 0.07 seconds Download

14. Afficher les noms des étudiants ayant suivi un cours dispensé par un enseignant nommé Einstein (éliminer les doublons).

```
SELECT DISTINCT student.student_name
FROM student
JOIN takes ON student.student_id = takes.student_id
JOIN section ON takes.course_id = section.course_id
            AND takes.sec_id = section.sec_id
            AND takes.semester = section.semester
            AND takes.semester_year = section.year_semester
JOIN teaches ON section.course_id = teaches.course_id
            AND section.sec_id = teaches.sec_id
            AND section.semester = teaches.semester
            AND section.year_semester = teaches.semester_year
JOIN teacher ON teaches.teacher_id = teacher.teacher_id
WHERE teacher.teacher_name = 'Einstein';
```

Results | Explain | Describe | Saved SQL | History

STUDENT_NAME
Peltier

1 rows returned in 0.05 seconds Download

15. Afficher tous les identifiants des cours et les enseignants qui les ont assurés.

```
SELECT DISTINCT teaches.course_id, teacher.teacher_name
FROM teaches
JOIN teacher ON teaches.teacher_id = teacher.teacher_id;
```

Results | Explain | Describe | Saved SQL | History

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

16. Afficher le nombre d'inscrits pour chaque enseignement proposé au printemps 2010.

```
1  SELECT course_id, sec_id, COUNT(student_id) AS nb_inscrits
2  FROM takes
3  WHERE semester = 'Spring'
4    AND semester_year = 2010
5  GROUP BY course_id, sec_id;
```

Results | Explain | Describe | Saved SQL | History

COURSE_ID	SEC_ID	NB_INSCRITS
CS-315	1	2
CS-101	1	1
CS-319	2	1
CS-319	1	1
FIN-201	1	1
HIS-351	1	1
MU-199	1	1

7 rows returned in 0.04 seconds Download

17. Afficher les noms des départements et les salaires maximum de leurs enseignants.

```
1  SELECT dept_name, MAX(salary) AS max_salary
2  FROM teacher
3  GROUP BY dept_name;
4
5
```

Results | Explain | Describe | Saved SQL | History

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

7 rows returned in 0.01 seconds Download

18. Afficher le nombre d'inscrits pour chaque enseignement proposé.

```
1  SELECT course_id, sec_id, COUNT(student_id) AS nb_inscrits
2  FROM takes
3  GROUP BY course_id, sec_id;
```

Results | Explain | Describe | Saved SQL | History

COURSE_ID	SEC_ID	NB_INSCRITS
CS-190	2	2
CS-315	1	2
CS-101	1	7
PHY-101	1	1
CS-319	2	1
CS-319	1	1
CS-347	1	2
BIO-101	1	1
FIN-201	1	1
EE-181	1	1
BIO-301	1	1
HIS-351	1	1
MU-199	1	1

13 rows returned in 0.12 seconds Download

19. Afficher le nombre total de cours qui ont eu lieu dans chaque bâtiment, pendant l'automne 2009 et le printemps 2010.

```
1  SELECT building, COUNT(*) AS nb_cours
2  FROM section
3  WHERE (semester = 'Fall' AND year_semester = 2009)
4      OR (semester = 'Spring' AND year_semester = 2010)
5  GROUP BY building;
```

Results | Explain | Describe | Saved SQL | History

BUILDING	NB_COURS
Watson	3
Packard	4
Taylor	2
Painter	1

4 rows returned in 0.02 seconds Download

20. Afficher le nombre total de cours dispensés par chaque département et qui ont eu lieu dans le même bâtiment qui l'abrite.

```sql
SELECT course.dept_name, COUNT(*) AS nb_cours
FROM section
JOIN course ON section.course_id = course.course_id
JOIN department ON course.dept_name = department.dept_name
WHERE section.building = department.building
GROUP BY course.dept_name;
```

Results | Explain | Describe | Saved SQL | History

DEPT_NAME	NB_COURS
Comp. Sci.	4
History	1
Elec. Eng.	1
Music	1
Physics	1

5 rows returned in 0.10 seconds Download

21. Afficher les titres des cours proposés et qui ont eu lieu et les enseignants qui les ont assurés.

```sql
SELECT DISTINCT course.title, teacher.teacher_name
FROM teaches
JOIN section ON teaches.course_id = section.course_id
            AND teaches.sec_id = section.sec_id
            AND teaches.semester = section.semester
            AND teaches.semester_year = section.year_semester
JOIN course ON section.course_id = course.course_id
JOIN teacher ON teaches.teacher_id = teacher.teacher_id;
```

Results | Explain | Describe | Saved SQL | History

TITLE	TEACHER_NAME
Intro. to Computer Science	Srinivasan
Intro. to Digital Systems	Kim
Investment Banking	Wu
Robotics	Srinivasan
Image Processing	Katz
Database System Concepts	Srinivasan
Image Processing	Brandt
Physical Principles	Einstein
World History	El Said
Intro. to Biology	Crick
Intro. to Computer Science	Katz

22. Afficher le nombre total de cours qui ont eu lieu pour chacune des périodes Summer, Fall et Spring.

```
1  SELECT semester, COUNT(*) AS nb_cours
2  FROM section
3  WHERE semester IN ('Summer', 'Fall', 'Spring')
4  GROUP BY semester;
5
```

Results | Explain | Describe | Saved SQL | History

SEMESTER	NB_COURS
Fall	3
Summer	2
Spring	10

3 rows returned in 0.02 seconds Download

23. Afficher pour chaque étudiant le nombre total de crédits qu'il a obtenu, en suivant des cours qui n'ont pas été proposés par son département.

```
1  SELECT student.student_name, SUM(course.credits) AS total_credits
2  FROM student
3  JOIN takes ON student.student_id = takes.student_id
4  JOIN course ON takes.course_id = course.course_id
5  WHERE student.dept_name <> course.dept_name
6  GROUP BY student.student_name;
```

Results | Explain | Describe | Saved SQL | History

STUDENT_NAME	TOTAL_CREDITS
Levy	11
Bourikas	7

2 rows returned in 0.06 seconds Download

24. Pour chaque département, afficher le nombre total de crédits des cours qui ont eu lieu dans ce département.

```
1  SELECT department.dept_name, SUM(course.credits) AS total_credits
2  FROM course
3  JOIN section ON course.course_id = section.course_id
4  JOIN department ON course.dept_name = department.dept_name
5  WHERE section.building = department.building
6  GROUP BY department.dept_name;
```

Results | Explain | Describe | Saved SQL | History

DEPT_NAME	TOTAL_CREDITS
Comp. Sci.	14
History	3
Elec. Eng.	3
Music	3
Physics	4

5 rows returned in 0.09 seconds Download