TP n° 1 : Quelques fonctions SQL et requêtes

Exercice 1 :

Requêtes de Création de tables de la base de données de gestion des cours :

Language SQL Rows 10 Clear Command Find Tables

```sql
create table classroom
    (building       varchar(15),
     room_number        varchar(7),
     capacity       numeric(4,0),
     primary key (building, room_number)
    );

create table department
    (dept_name      varchar(20),
     building       varchar(15),
     budget             numeric(12,2),
     primary key (dept_name)
    );
```

```sql
create table course
    (course_id      varchar(8),
     title          varchar(50),
     dept_name      varchar(20),
     credits        numeric(2,0),
     primary key (course_id),
     foreign key (dept_name) references department
    );
create table teacher
    (ID         varchar(5),
     name           varchar(20),
     dept_name      varchar(20),
     salary         numeric(8,2),
     primary key (ID),
     foreign key (dept_name) references department
    );
```

```sql
create table section
    (course_id      varchar(8),
        sec_id          varchar(8),
     semester       varchar(6)
        check (semester in ('Fall', 'Winter', 'Spring', 'Summer')),
     year           numeric(4,0),
     building       varchar(15),
     room_number        varchar(7),
     time_slot_id       varchar(4),
     primary key (course_id, sec_id, semester, year),
     foreign key (course_id) references course,
     foreign key (building, room_number) references classroom
    );

```

```sql
create table teaches
    (ID         varchar(5),
     course_id      varchar(8),
     sec_id         varchar(8),
     semester       varchar(6),
     year           numeric(4,0),
     primary key (ID, course_id, sec_id, semester, year),
     foreign key (course_id,sec_id, semester, year) references section,
     foreign key (ID) references teacher
    );

```

```sql
create table student
    (ID         varchar(5),
     name           varchar(20),
     dept_name      varchar(20),
     tot_cred       numeric(3,0),
     primary key (ID),
     foreign key (dept_name) references department
    );

```

```sql
create table takes
(ID          varchar(5),
 course_id       varchar(8),
 sec_id          varchar(8),
 semester        varchar(6),
 year            numeric(4,0),
 grade               varchar(2),
 primary key (ID, course_id, sec_id, semester, year),
 foreign key (course_id,sec_id, semester, year) references section,
 foreign key (ID) references student
);

```

```sql
create table advisor
    (s_ID            varchar(5),
     i_ID            varchar(5),
     primary key (s_ID),
     foreign key (i_ID) references teacher (ID),
     foreign key (s_ID) references student (ID)
    );

```

```sql
create table time_slot
    (time_slot_id        varchar(4),
     day             varchar(1),
     start_hr        numeric(2),
     start_min       numeric(2),
     end_hr          numeric(2),
     end_min         numeric(2),
     primary key (time_slot_id, day, start_hr, start_min)
    );

```

```sql
create table prereq
    (course_id       varchar(8),
     prereq_id       varchar(8),
     primary key (course_id, prereq_id),
     foreign key (course_id) references course,
     foreign key (prereq_id) references course
    );
```

Résultat :

Recently Created Tables	
ADVISOR	6 minutes ago
TIME_SLOT	6 minutes ago
TEACHES	6 minutes ago
TEACHER	6 minutes ago
TAKES	6 minutes ago
STUDENT	6 minutes ago
SECTION	6 minutes ago
PREREQ	6 minutes ago
CLASSROOM	6 minutes ago
COURSE	6 minutes ago
DEPARTMENT	6 minutes ago
HTMLDB_PLAN_TABLE	17 minutes ago

2) Insérer un nouveau cours dont l'identifiant est BIO-101, intitulé Intro. to Biology, assur´e par le département Biology et son crédit est de 4

INSERT INTO course VALUES ('BIO-101', 'Intro. to Biology', 'Biology', 4);

Vérification :

```
SELECT * FROM course WHERE course_id = 'BIO-101';
```

Results | Explain | Describe | Saved SQL | History

COURSE_ID	TITLE	DEPT_NAME	CREDITS
BIO-101	Intro. to Biology	Biology	4

Exercice 2 :

1. Afficher la structure de la relation section et son contenu (cours proposés).

Pour voir les colonnes et leurs types dans la table **Section** :
DESCRIBE section;

```
1  DESCRIBE section;
2
3
```

Results | Explain | **Describe** | Saved SQL | History

SECTION	COURSE_ID	VARCHAR2	8	-	-	1	-	-	-
	SEC_ID	VARCHAR2	8	-	-	2	-	-	-
	SEMESTER	VARCHAR2	6	-	-	3	-	-	-
	YEAR	NUMBER	-	4	0	4	-	-	-
	BUILDING	VARCHAR2	15	-	-	-	✓	-	-
	ROOM_NUMBER	VARCHAR2	7	-	-	-	✓	-	-
	TIME_SLOT_ID	VARCHAR2	4	-	-	-	✓	-	-

Afficher le contenu de la table **Section** :

SELECT * FROM section;

```
1  SELECT * FROM section;
2
```

Results | Explain | Describe | Saved SQL | History

COURSE_ID	SEC_ID	SEMESTER	YEAR	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

2) Afficher tous les renseignements sur les cours que l'on peut programmer (relation course)

SELECT * FROM course;

```
1   SELECT * FROM course;
2
```

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

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

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

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

Results | Explain | Describe | Saved SQL | History

TITLE	DEPT_NAME
Intro. to Biology	Biology
Genetics	Biology
Computational Biology	Biology
Intro. to Computer Science	Comp. Sci.

4) Afficher les noms des départements ainsi que leur budget

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

```
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

5) Afficher tous les noms des enseignants et leur département

```
SELECT name , dept_name
FROM teacher;
```

```
SELECT name , dept_name
FROM teacher;
```

Results | Explain | Describe | Saved SQL | History

NAME	DEPT_NAME
Srinivasan	Comp. Sci.
Wu	Finance
Mozart	Music
Einstein	Physics

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

```
SELECT name
FROM teacher
WHERE salary > 65000;
```

```
SELECT name
FROM teacher
WHERE salary > 65000;
```

Results | Explain | Describe | Saved SQL | History

NAME
Wu
Einstein
Gold
Katz

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

```
SELECT name
FROM teacher
WHERE salary between 55000 and 85000;
```

Language SQL Rows 10

Clear Command | Find Tables

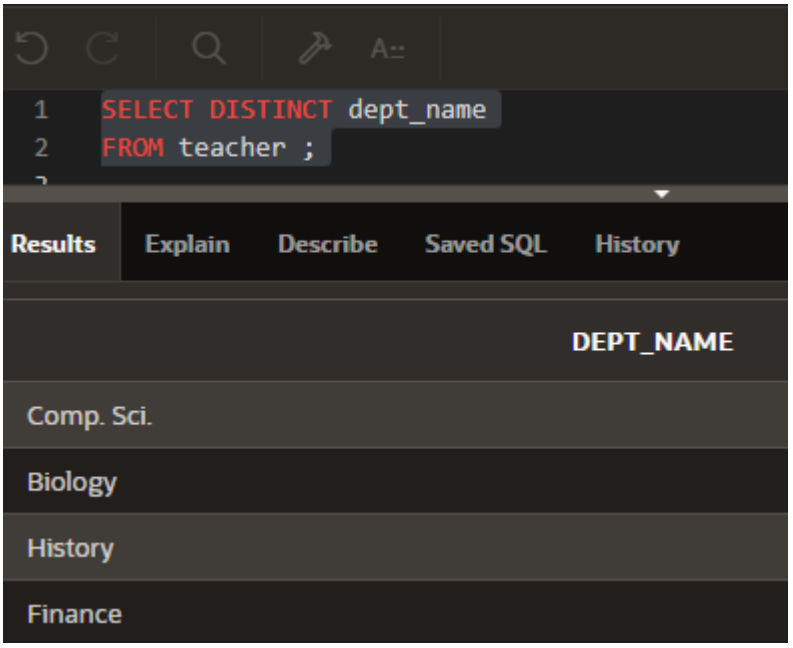
```
SELECT name
FROM teacher
WHERE salary between 55000 and 85000;
```

Results | Explain | Describe | Saved SQL | History

NAME
Srinivasan
El Said
Katz

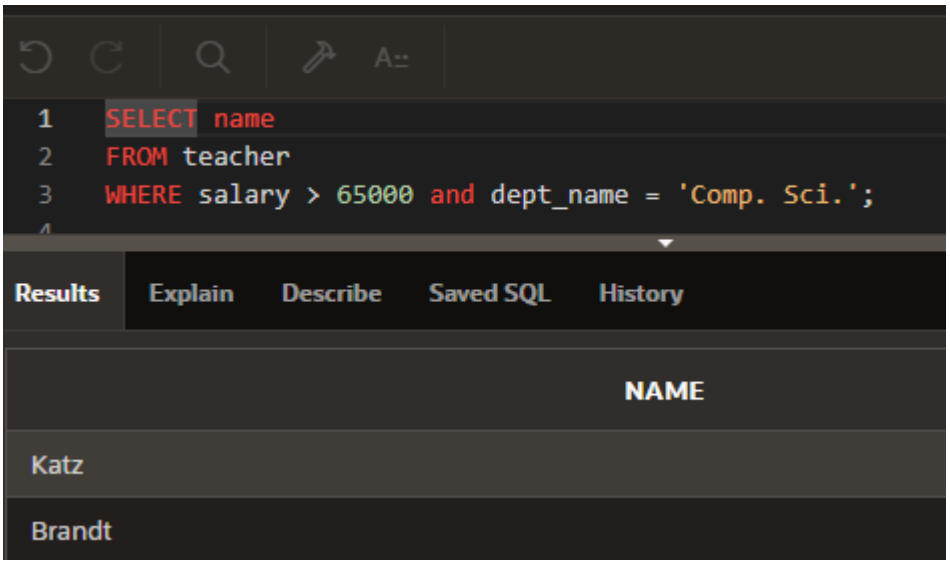
8) Afficher les noms des départements, en utilisant la relation teacher et éliminer les doublons.

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



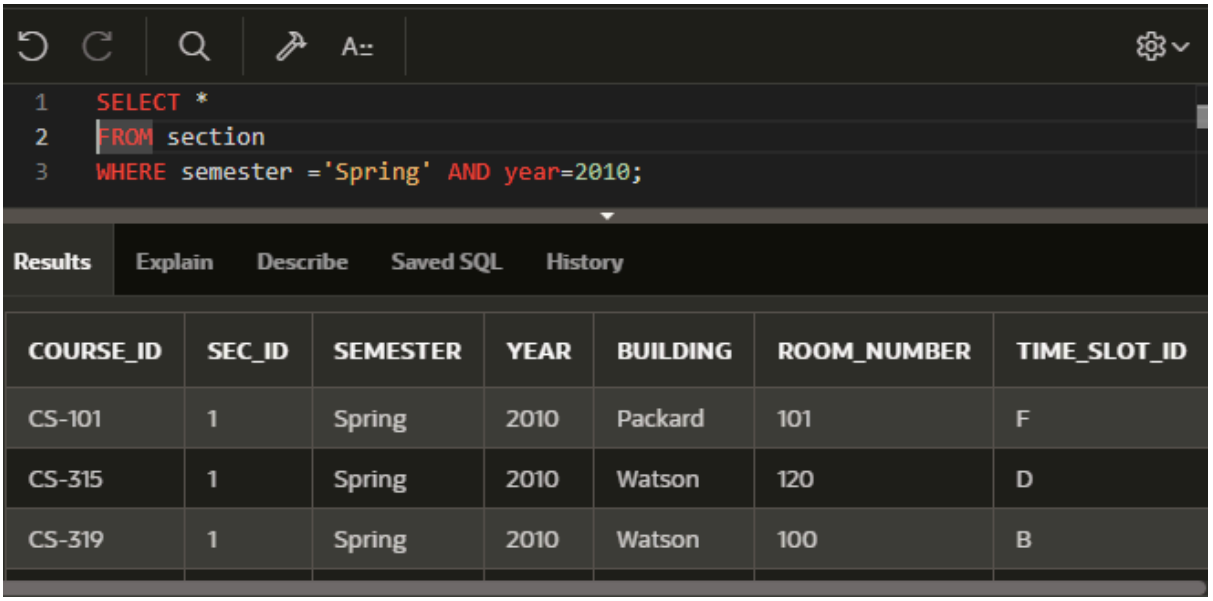
9) Afficher tous les noms des enseignants du département informatique ayant un salaire supérieur strictement à 65.000 $.

```
SELECT name
FROM teacher
WHERE salary > 65000 and dept_name = 'Comp. Sci.';
```



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

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



COURSE_ID	SEC_ID	SEMESTER	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

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;
```

```
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

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

```
SELECT teacher.name , teacher.dept_name , department.building
FROM teacher , department
WHERE teacher.dept_name = department.dept_name ;
```

```
SELECT teacher.name , teacher.dept_name , department.building
FROM teacher , department
WHERE teacher.dept_name = department.dept_name ;
```

Results | Explain | Describe | Saved SQL | History

NAME	DEPT_NAME	BUILDING
Srinivasan	Comp. Sci.	Taylor
Wu	Finance	Painter
Mozart	Music	Packard

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

```
SELECT distinct student.name
FROM student , takes , course
WHERE student.id = takes.id and takes.course_id = course .
course_id and course.dept_name = 'Comp. Sci.';
```

```
SELECT distinct student.name
FROM student , takes , course
WHERE student.id = takes.id and takes.course_id = course .
course_id and course.dept_name = 'Comp. Sci.';
```

Results | Explain | Describe | Saved SQL | History

NAME
Bourikas
Shankar
Williams

6 rows returned in 0.04 seconds Download

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

```
SELECT DISTINCT student.name
FROM student, teacher, takes, teaches
WHERE student.id = takes.id
  AND takes.course_id = teaches.course_id
  AND takes.sec_id = teaches.sec_id
  AND takes.semester = teaches.semester
  AND takes.year = teaches.year
  AND teaches.id = teacher.id
  AND teacher.name = 'Einstein';
```

```
SELECT DISTINCT student.name
FROM student, teacher, takes, teaches
WHERE student.id = takes.id
  AND takes.course_id = teaches.course_id
  AND takes.sec_id = teaches.sec_id
  AND takes.semester = teaches.semester
  AND takes.year = teaches.year
  AND teaches.id = teacher.id
```

Results | Explain | Describe | Saved SQL | History

NAME
Peltier

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

```
SELECT teacher.name, teaches.course_id
FROM teacher, teaches
WHERE teacher.id = teaches.id;
```

```
SELECT teacher.name, teaches.course_id
FROM teacher, teaches
WHERE teacher.id = teaches.id;
```

Results | Explain | Describe | Saved SQL | History

NAME	COURSE_ID
Srinivasan	CS-101
Srinivasan	CS-315
Srinivasan	CS-347

16) Afficher le nombre d’inscrits pour chaque enseignement proposé au printemps 2010

```
SELECT takes.course_id, takes.sec_id, takes.semester, takes.year, COUNT(*)
FROM takes
WHERE takes.semester = 'Spring' AND takes.year = 2010
GROUP BY takes.course_id, takes.sec_id, takes.semester, takes.year;
```

```sql
SELECT takes.course_id, takes.sec_id, takes.semester, takes.year, COUNT(*)
FROM takes
WHERE takes.semester = 'Spring' AND takes.year = 2010
GROUP BY takes.course_id, takes.sec_id, takes.semester, takes.year;
```

Results | Explain | Describe | Saved SQL | History

COURSE_ID	SEC_ID	SEMESTER	YEAR	COUNT(*)
CS-315	1	Spring	2010	2
CS-101	1	Spring	2010	1
CS-319	2	Spring	2010	1

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

SELECT dept_name , max (salary) FROM teacher 2 GROUP BY dept_name ;

```sql
SELECT dept_name , max ( salary ) FROM teacher
GROUP BY dept_name ;
```

Results | Explain | Describe | Saved SQL | History

DEPT_NAME	MAX(SALARY)
Comp. Sci.	92000
Biology	72000
History	62000
Finance	90000

18) Afficher le nombre d'inscrits pour chaque enseignement proposés

SELECT takes.course_id, takes.sec_id, takes.semester, takes.year, COUNT(*)
FROM takes
GROUP BY takes.course_id, takes.sec_id, takes.semester, takes.year;

SQL Commands

Schema WKSP_DBATPS

Language SQL Rows 10

Clear Command Find Tables Save Run

```sql
SELECT takes.course_id, takes.sec_id, takes.semester, takes.year, COUNT(*)
FROM takes
GROUP BY takes.course_id, takes.sec_id, takes.semester, takes.year;
```

Results Explain Describe Saved SQL History

COURSE_ID	SEC_ID	SEMESTER	YEAR	COUNT(*)
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

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

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

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

Results | Explain | Describe | Saved SQL | History

COURSE_ID	SEC_ID	SEMESTER	YEAR	COUNT(*)
CS-190	2	Spring	2009	2
CS-319	1	Spring	2010	1
CS-347	1	Fall	2009	2
MU-199	1	Spring	2010	1

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

```sql
SELECT department.dept_name, COUNT(*)
FROM section, department, teacher, teaches
WHERE section.course_id = teaches.course_id
  AND section.sec_id = teaches.sec_id
  AND section.semester = teaches.semester
  AND section.year = teaches.year
  AND teaches.id = teacher.id
  AND teacher.dept_name = department.dept_name
  AND department.building = section.building
GROUP BY department.dept_name;
```

```sql
SELECT department.dept_name, COUNT(*)
FROM section, department, teacher, teaches
WHERE section.course_id = teaches.course_id
  AND section.sec_id = teaches.sec_id
  AND section.semester = teaches.semester
  AND section.year = teaches.year
  AND teaches.id = teacher.id
  AND teacher.dept_name = department.dept_name
  AND department.building = section.building
GROUP BY department.dept_name;
```

Results | Explain | Describe | Saved SQL | History

DEPT_NAME	COUNT(*)
Physics	1

5 rows returned in 0.14 seconds Download

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

```sql
SELECT course.title, teacher.name
FROM section, teacher, teaches, course
WHERE section.course_id = teaches.course_id
  AND section.sec_id = teaches.sec_id
  AND section.semester = teaches.semester
  AND section.year = teaches.year
  AND teaches.id = teacher.id
  AND section.course_id = course.course_id
ORDER BY course.title;
```

```sql
SELECT course.title, teacher.name
FROM section, teacher, teaches, course
WHERE section.course_id = teaches.course_id
  AND section.sec_id = teaches.sec_id
  AND section.semester = teaches.semester
  AND section.year = teaches.year
  AND teaches.id = teacher.id
  AND section.course_id = course.course_id
ORDER BY course.title;
```

Results | Explain | Describe | Saved SQL | History

TITLE	NAME
Database System Concepts	Srinivasan
Game Design	Brandt

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

SELECT section . semester , count (*) FROM section 2 GROUP BY section . semester

Language SQL | Rows 10 | Save | Run

Clear Command | Find Tables

```sql
SELECT section . semester , count (*) FROM section
GROUP BY section . semester
```

Results | Explain | Describe | Saved SQL | History

SEMESTER	COUNT(*)
Fall	3
Summer	2
Spring	10

3 rows returned in 0.01 seconds Download

23) Affiché 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.

```
SELECT student.name, SUM(course.credits)
FROM student, course, takes
WHERE student.id = takes.id
  AND takes.course_id = course.course_id
  AND student.dept_name != course.dept_name
GROUP BY student.name;
```

```
1  SELECT student.name, SUM(course.credits)
2  FROM student, course, takes
3  WHERE student.id = takes.id
4    AND takes.course_id = course.course_id
5    AND student.dept_name != course.dept_name
6  GROUP BY student.name;
7
```

Results | Explain | Describe | Saved SQL | History

NAME	SUM(COURSE.CREDITS)
Levy	11
Bourikas	7

2 rows returned in 0.05 seconds Download

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

SELECT section . building , sum (course . credits) FROM section , course 2 WHERE section . course_id = course . course_id GROUP BY section . building ;

Language SQL Rows 10 Save Run

Clear Command Find Tables

```sql
SELECT section . building , sum ( course . credits ) FROM section , course
WHERE section . course_id = course . course_id GROUP BY section . building ;
```

Results Explain Describe Saved SQL History

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

4 rows returned in 0.02 seconds Download