

TP1 BDA

Exercise 1 :

- 1) création de tables : course, departement, classroom, section

```
1  create table classroom
2      (building      varchar(15),
3       room_number   varchar(7),
4       capacity      numeric(4,0),
5       primary key (building, room_number)
6      );
7
8  create table department
9      (dept_name      varchar(20),
10     building        varchar(15),
11     budget          numeric(12,2),
12     primary key (dept_name)
13    );
14
15 create table course
16     (course_id      varchar(8),
17     title           varchar(50),
18     dept_name       varchar(20),
19     credits         numeric(2,0),
20     primary key (course_id),
21     foreign key (dept_name) references department
22    );
23
24 create table section
25     (course_id      varchar(8),
26     sec_id          varchar(8),
27     semester        varchar(6)
28     check (semester in ('Fall', 'Winter', 'Spring', 'Summer')),
29     year            numeric(4,0),
30     building        varchar(15),
31     room_number     varchar(7),
32     time_slot_id    varchar(4),
33     primary key (course_id, sec_id, semester, year),
34     foreign key (course_id) references course,
35     foreign key (building, room_number) references classroom
36    );
```

2) création des autres tables :

```
1 create table teacher
2   (ID          varchar(5),
3    name         varchar(20),
4    dept_name    varchar(20),
5    salary       numeric(8,2),
6    primary key (ID),
7    foreign key (dept_name) references department
8  );
```

```
1 create table teaches
2   (ID          varchar(5),
3    course_id    varchar(8),
4    sec_id       varchar(8),
5    semester     varchar(6),
6    year         numeric(4,0),
7    primary key (ID, course_id, sec_id, semester, year),
8    foreign key (course_id, sec_id, semester, year) references section,
9    foreign key (ID) references teacher
10  );
```

```
1 create table student
2   (ID          varchar(5),
3    name         varchar(20),
4    dept_name    varchar(20),
5    tot_cred     numeric(3,0),
6    primary key (ID),
7    foreign key (dept_name) references department
8  );
```

```
1 create table takes
2   (ID          varchar(5),
3    course_id    varchar(8),
4    sec_id       varchar(8),
5    semester     varchar(6),
6    year         numeric(4,0),
7    grade        varchar(2),
8    primary key (ID, course_id, sec_id, semester, year),
9    foreign key (course_id, sec_id, semester, year) references section,
10   foreign key (ID) references student
11  );
```

```
1 create table advisor
2   (s_ID         varchar(5),
3    i_ID         varchar(5),
4    primary key (s_ID),
5    foreign key (i_ID) references teacher (ID),
6    foreign key (s_ID) references student (ID)
7  );
```

```
1 create table time_slot
2   (time_slot_id varchar(4),
3    day          varchar(1),
4    start_hr     numeric(2),
5    start_min    numeric(2),
6    end_hr       numeric(2),
7    end_min      numeric(2),
8    primary key (time_slot_id, day, start_hr, start_min)
9  );
```

```
1 create table prereq
2   (course_id    varchar(8),
3    prereq_id     varchar(8),
4    primary key (course_id, prereq_id),
5    foreign key (course_id) references course,
6    foreign key (prereq_id) references course
7  );
```

3) Requêtes insert :

0.01	insert into classroom values ('Packard', '101', '500')	1 row(s) inserted.	1
0.00	insert into classroom values ('Painter', '514', '10')	1 row(s) inserted.	1
0.00	insert into classroom values ('Taylor', '302B', '70')	1 row(s) inserted.	1
0.01	insert into classroom values ('Watson', '100', '30')	1 row(s) inserted.	1
0.00	insert into classroom values ('Watson', '120', '50')	1 row(s) inserted.	1
0.00	insert into department values ('Biology', 'Watson', '90000')	1 row(s) inserted.	1
0.00	insert into department values ('Comp. Sci.', 'Taylor', '1000')	1 row(s) inserted.	1
0.00	insert into department values ('Elec. Eng.', 'Taylor', '8500')	1 row(s) inserted.	1
0.00	insert into department values ('Finance', 'Painter', '120000')	1 row(s) inserted.	1

load

row(s) 1 - 20 of 149 [Next](#)

149

Statements Processed

149

Successful

0

With Errors

[Print](#)
[Email](#)
[Share](#)

Copyright © 1999, 2024, Oracle and/or its affiliates.
 Oracle APEX 24.2.1

Exercice 2 :

1. Afficher la structure de la relation section et son contenu :

1

DESCRIBE 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	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	-	-	-	✓	-	-

Language SQL Rows 10 Clear Command Save Run

Find Tables

↶ ↷ 🔍 🔗 A:: ⚙️

```
1 select * from section;
```

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
CS-190	1	Spring	2009	Taylor	3128	E

cyrinetlili07@gmail.com
bdainfo en

Copyright © 1999, 2024, Oracle and/or its affiliates.

Oracle APEX 24.2.1

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

Language SQL Rows 10 Clear Command Save Run

Find Tables

↶ ↷ 🔍 🔗 A:: ⚙️

```
1 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

cyrinetlili07@gmail.com
bdainfo en

Copyright © 1999, 2024, Oracle and/or its affiliates.

Oracle APEX 24.2.1

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

Language SQL ? Rows 10 ? Clear Command Save Run

Find Tables

↶ ↷ 🔍 🔗 A:: ⚙️

```
1 select title , dept_name 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.
Game Design	Comp. Sci.

cyrinetlili07@gmail.com bdainfo en Copyright © 1999, 2024, Oracle and/or its affiliates. Oracle APEX 24.2.1

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

Language SQL ? Rows 10 ? Clear Command Save Run

Find Tables

↶ ↷ 🔍 🔗 A:: ⚙️

```
1 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

cyrinetlili07@gmail.com bdainfo en Copyright © 1999, 2024, Oracle and/or its affiliates. Oracle APEX 24.2.1

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

Language SQL Rows 10 Clear Command Save Run



Find Tables

↶ ↷ 🔍 🔗 A:: ⚙️

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

Results Explain Describe Saved SQL History

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

 cyrinetlili07@gmail.com
bdainfo  en

Copyright © 1999, 2024, Oracle and/or its affiliates. Oracle APEX 24.2.1

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

Language SQL Rows 10 Clear Command Save Run



Find Tables

↶ ↷ 🔍 🔗 A:: ⚙️

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

Results Explain Describe Saved SQL History

NAME
Wu
Einstein
Gold
Katz
Singh

 cyrinetlili07@gmail.com
bdainfo  en

Copyright © 1999, 2024, Oracle and/or its affiliates. Oracle APEX 24.2.1

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

The screenshot shows the Oracle APEX SQL Editor interface. At the top, the 'Language' is set to 'SQL' and 'Rows' is set to '10'. The 'Run' button is highlighted in green. The SQL command entered is: `1 SELECT name FROM teacher WHERE salary between 55000 and 85000;`. Below the command, the 'Results' tab is selected, displaying a table with one column, 'NAME', and five rows of teacher names: Srinivasan, El Said, Katz, Califieri, and Singh. The footer shows the user 'cyrinetlili07@gmail.com', the date 'Copyright © 1999, 2024, Oracle and/or its affiliates.', and the version 'Oracle APEX 24.2.1'.

NAME
Srinivasan
El Said
Katz
Califieri
Singh

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

The screenshot shows the Oracle APEX SQL Editor interface. At the top, the 'Language' is set to 'SQL' and 'Rows' is set to '10'. The 'Run' button is highlighted in green. The SQL command entered is: `1 SELECT DISTINCT dept_name FROM teacher ;`. Below the command, the 'Results' tab is selected, displaying a table with one column, 'DEPT_NAME', and five rows of department names: Comp. Sci., Biology, History, Finance, and Elec. Eng. The footer shows the user 'cyrinetlili07@gmail.com', the date 'Copyright © 1999, 2024, Oracle and/or its affiliates.', and the version 'Oracle APEX 24.2.1'.

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

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

Language SQL Rows 10 Clear Command Save Run

Find Tables

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

Results Explain Describe Saved SQL History

NAME
Katz
Brandt

2 rows returned in 0.00 seconds [Download](#)

cyrinetlili07@gmail.com
bdainfo en Copyright © 1999, 2024, Oracle and/or its affiliates. Oracle APEX 24.2.1

10. Afficher tous les renseignements sur les cours proposés au printemps 2010 :

Language SQL Rows 10 Clear Command Save Run

Find Tables

1 `SELECT * FROM section where semester = 'Spring' and year = '2010';`

Results Explain Describe Saved SQL History

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
CS-319	2	Spring	2010	Taylor	3128	C
FIN-201	1	Spring	2010	Packard	101	B

cyrinetlili07@gmail.com
bdainfo en Copyright © 1999, 2024, Oracle and/or its affiliates. Oracle APEX 24.2.1

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

Language SQL Rows 10 Clear Command Save Run

Find Tables

1 `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 0.00 seconds [Download](#)

cyrinetlilii07@gmail.com
bdainfo en Copyright © 1999, 2024, Oracle and/or its affiliates. Oracle APEX 24.2.1

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

Language SQL Rows 10 Clear Command Save Run

Find Tables

1 `SELECT t.name , t.dept_name , d.building FROM teacher t, department d`
2 `WHERE t.dept_name = d.dept_name ;`

Results Explain Describe Saved SQL History

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

cyrinetlilii07@gmail.com
bdainfo en Copyright © 1999, 2024, Oracle and/or its affiliates. Oracle APEX 24.2.1

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

The screenshot shows a SQL query editor interface. At the top, there's a toolbar with 'Language' set to 'SQL', 'Rows' set to '10', and buttons for 'Clear Command', 'Save', and 'Run'. Below the toolbar is a 'Find Tables' button. The main area contains a SQL query:

```
1 SELECT distinct s.name FROM student s , takes t , course c
2 WHERE s.id = t.id and t.course_id = c.course_id and c.dept_name = 'Comp. Sci.';
```

Below the query, there are tabs for 'Results', 'Explain', 'Describe', 'Saved SQL', and 'History'. The 'Results' tab is active, showing a table with one column named 'NAME'. The table contains six rows of student names: Brown, Zhang, Levy, Bourikas, Shankar, and Williams. At the bottom, it states '6 rows returned in 0.02 seconds' and provides a 'Download' link.

NAME
Brown
Zhang
Levy
Bourikas
Shankar
Williams

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

The screenshot shows a SQL query editor interface. At the top, there's a toolbar with 'Language' set to 'SQL', 'Rows' set to '10', and buttons for 'Clear Command', 'Save', and 'Run'. Below the toolbar is a 'Find Tables' button. The main area contains a SQL query:

```
1 SELECT distinct s.name FROM student s, teacher t, takes ta, teaches te
2 WHERE s.id = ta.id and ta.course_id = te.course_id
3 and ta.sec_id = te.sec_id and ta.semester = te.semester
4 and ta.year = te.year and te.id = t.id and t.name = 'Einstein' ;
```

Below the query, there are tabs for 'Results', 'Explain', 'Describe', 'Saved SQL', and 'History'. The 'Results' tab is active, showing a table with one column named 'NAME'. The table contains one row with the name 'Peltier'. At the bottom, it states '1 rows returned in 0.05 seconds' and provides a 'Download' link.

NAME
Peltier

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

Language SQL Rows 10 Clear Command Save Run

Find Tables

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

Results Explain Describe Saved SQL History

NAME	COURSE_ID
Einstein	PHY-101
El Said	HIS-351
Katz	CS-101
Katz	CS-319
Crick	BIO-101

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

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

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

Language SQL Rows 10 Clear Command Save Run

Find Tables

1 `SELECT takes.course_id , takes.sec_id , takes.semester , takes.year , count (*)`
 2 `FROM takes`
 3 `WHERE takes.semester = 'Spring' and takes . year = '2010'`
 4 `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-319	1	Spring	2010	1
CS-315	1	Spring	2010	2
CS-101	1	Spring	2010	1
MU-199	1	Spring	2010	1

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

Language SQL Rows 10 Clear Command Save Run

Find Tables

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

Results Explain Describe Saved SQL History

DEPT_NAME	MAX(SALARY)
Comp. Sci.	92000
Biology	72000
History	62000
Finance	90000
Elec. Eng.	80000
Music	40000

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

Language SQL Rows 10 Clear Command Save Run

Find Tables

1 SELECT takes.course_id , takes.sec_id , takes.semester , takes.year , count (*)
2 FROM takes
3 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

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

Language SQL Rows 10 Clear Command Save Run

Find Tables

```

1 SELECT building , count (*)
2 FROM section
3 WHERE ( semester , year ) IN ( ( 'Fall' , 2009 ) , ( 'Spring' , 2010 ) )
4 GROUP BY building ;

```

Results Explain Describe Saved SQL History

BUILDING	COUNT(*)
Watson	3
Packard	4
Taylor	2
Painter	1

4 rows returned in 0.00 seconds Download

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

Language SQL Rows 10 Clear Command Save Run

Find Tables

```

1 SELECT department . dept_name , count (*)
2 FROM section , department , teacher , teaches
3 WHERE
4 section.course_id = teaches.course_id
5 and section.sec_id = teaches.sec_id
6 and section.semester = teaches.semester
7 and section.year = teaches.year
8 and teaches . id = teacher . id
9 and teacher . dept_name = department.dept_name
10 and department . building = section.building
11 GROUP BY department.dept_name ;

```

Results Explain Describe Saved SQL History

DEPT_NAME	COUNT(*)
History	1
Elec. Eng.	1
Music	1

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

Language: SQL Rows: 10 Clear Command Save Run

Find Tables

```

1 SELECT course . title , teacher . name
2 FROM section , teacher , teaches , course
3 WHERE section . course_id = teaches . course_id
4 and section . sec_id = teaches . sec_id
5 and section . semester = teaches . semester
6 and section . year = teaches . year
7 and teaches . id = teacher . id
8 and section . course_id = course . course_id
9 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.

Language: SQL Rows: 10 Clear Command Save Run

Find Tables

```

1 SELECT section . semester , count (*) FROM section
2 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. 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.

Language SQL Rows 10 Clear Command Save Run

Find Tables

```

1 SELECT student . name , sum (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 ;

```

Results Explain Describe Saved SQL History

NAME	SUM(CREDITS)
Levy	11
Bourikas	7

2 rows returned in 0.04 seconds Download

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

Language SQL Rows 10 Clear Command Save Run

Find Tables

```

1 select section.building , sum (course.credits)
2 from section , course
3 WHERE section.course_id = course.course_id
4 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 0.03 seconds Download