TP n° 1 : Quelques fonctions SQL et requˆetes

Pr´eparation de l'environnement de travail

1. Assurez-vous que votre fichier .bashrc contient bien l'environnement n´ecessaire pour l'ex´ecution d'Oracle. Vous pouvez l'ouvrir (`a manipuler avec pr´ecaution) avec votre ´editeur pr´ef´er´e, par exemple vim .bashrc et enlever le commentaire # de la ligne : source export/home/users/COMMUN/.oraclrc
2. Lancer le client d'oracle sqlplus. vous pouvez aussi utiliser le client graphique Sqldevloper, mais je vous conseille d'utiliser, au moins dans un premier temps, sqlplus.
3. Identifiez-vous avec votre login et mot de passe (fournis lors du TP) : Une fois sqlplus lanc´e, introduire votre login et puis votre mot de passe, ou en introduisant directement cette ligne sqlplus monLogin/monMotDePasse
4. La commande spool fichier.log permet de sauvegarder vos requˆetes et leurs r´esultats dans le fichier fichier.log.
5. Ci-dessous quelques commandes permettant de changer le format d'affichage :

```
set colsep '|'
set linesize 200
set pagesize 20
column column_name format a30

```

Vous d´eposez le travail r´ealis´e sur l'ENT.

Quelques fonctions SQL/Oracle

La table DUAL est une table d'une seule ligne et d'une seule colonne pr´esente par d´efaut dans les installations du SGBD Oracle. La table contient une seule colonne de type VARCHAR2(1) appel´ee DUMMY qui a pour unique valeur 'X'. Elle permet d'envoyer des ordres SQL de type SELECT sans utiliser de table particuli`ere pour tester des fonctions ou r´ecup´erer des informations ind´ependamment des donn´ees mais li´ees `a la base ou `a Oracle (r´ecup´erer la date syst`eme, par exemple). L'objectif dans ce qui suit est de tester certaines fonctions. La liste n'est pas exhaustive. Vous trouverez `a l'adresse suivante plus de fonctions https://docs.oracle.com/database/121/SQLRF/functions.htm#SQLRF006

Tester chacune des fonctions suivantes sur un exemple et expliquer bri`evement son fonctionnement.

- Exemple de r´eponse :
  SYSDATE permet d'afficher la date syst`eme. Exemple
  d'utilisation :

```
SELECT SYSDATE FROM DUAL
```

- TO CHAR ; RPAD ; LPAD ; SUBSTR ; LENGTH ; ROUND ; TRUNC ; TO DATE ; EXTRACT.

Exercice n° 1

Soit l'exemple de la gestion des cours dont le sch´ema relationnel correspondant est donn´e ci-apr`es. Le script pour cr´eer le sch´ema de la base est donn´ee `a l'adresse suivante https://moodlelms.univ-paris13.fr/mod/resource/view.php?id=173183 et pour la peupler, vous pouvez utiliser le script qui se trouve `a l'adresse suivante https://moodlelms.univ-paris13.fr/mod/resource/view.php?id=173184

Remarque A´ ce stade, vous ne pouvez pas utiliser ces scripts, il faut lire la suite et les compl´eter.

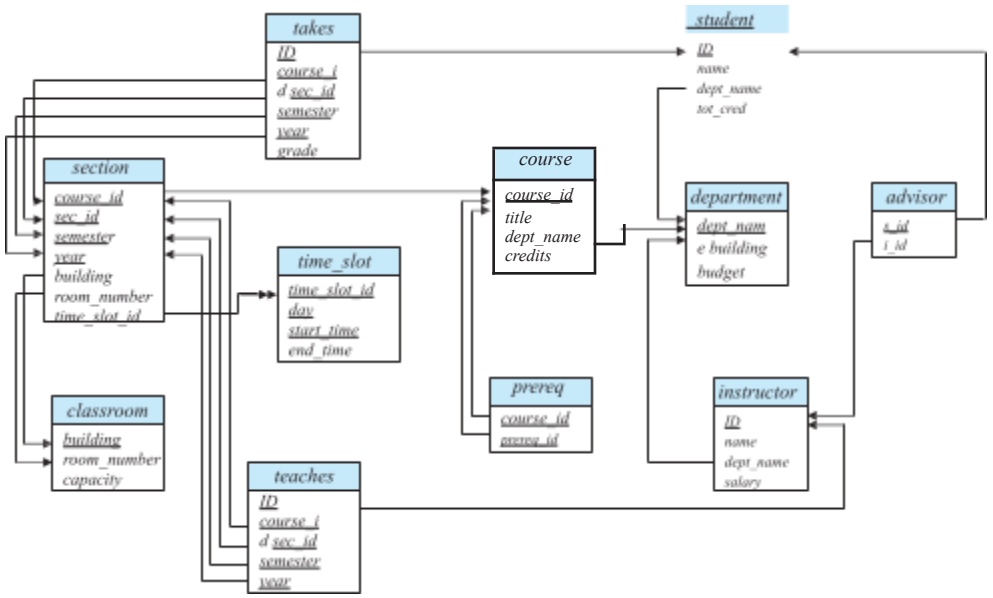

Figure 1: Sch´ema relationnel de la base de donn´ees de gestion des cours

1. D´efinir une nouvelle table **section** dont le sch´ema est
   section(**course id, sec id, semester, year**, *building, room number, time slot id*). Tous les attributs sont de type cha ̂ınes de caract`eres sauf year qui est de type num´erique.
   Ajouter la contrainte sur l'attribut semestre dont les valeurs appartienent `a l'ensemble {'Fall', 'Winter', 'Spring', 'Summer'}. Bien placer cette d´efinition dans le pr´ec´edent script.

2. D´eterminer le diagramme entit´e-association de cette base de donn´ees. Rappel : avant d'interroger une base de donn´ee, prendre le temps de bien comprendre sa structure et avoir `a l'esprit qu'elle ne mod´elise pas la r´ealit´e.

3. Proc´eder `a la cr´eation de la base de donn´ees et la peupler. Je vous rappelle que le lancement du script *test.sql* sous *sqlplus* se fait comme suit

```
SQL > @test
```

4. Ins´erer un nouveau cours dont l'identifiant est BIO-101, intitul´e *Intro. to Biology*, assur´e par le d´epartement *Biology* et son cr´edit est de 4.

Exercice n° 2

1. Afficher la structure de la relation section et son contenu (cours propos´es).

   desc course;

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

   select * from course

3. Afficher les titres des cours et les d´epartements qui les proposent.
   select title, dept_name from course;

4. Afficher les noms des d´epartements ainsi que leur budget.

   select dept_name, budget from department;

5. Afficher tous les noms des enseignants et leur d´epartement.

   select dept_name, name from teacher;

6. Afficher tous les noms des enseignants ayant un salaire sup´erieur strictement `a 65.000 $.

   select name from teacher where salary >65000;

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

   select name from teacher where (salary <=85000 and salary>= 55000);

8. Afficher les noms des d´epartements, en utilisant la relation teacher et ´eliminer les doublons.

   select distinct dept_name from teacher;

9. Afficher tous les noms des enseignants du d´epartement informatique ayant un salaire sup´erieur strictement `a 65.000 $.

   select dept_name, name from teacher where dept_name ='Comp. Sci.' and salary >65000;

10. Afficher tous les renseignements sur les cours propos´es au printemps 2010 (relation section).

    select * from section where semester = 'Spring' and year=2010;

11. Afficher tous les titres des cours dispens´es par le d´epartement informatique qui ont plus de trois cr´edits.

    select title from course where dept_name='Comp. Sci.' and credits>3;

---

12. Afficher tous les noms des enseignants ainsi que le nom de leur d´epartement et les noms des bˆatiments qui les h´ebergent.
    select t.name, t.dept_name, d.building from teacher t ,department d where t.dept_name= d.dept_name;

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

    select distinct s.name from student s, takes t, course co

    where

        co.dept_name = 'Comp. Sci.'

        and t.course_id= co.course_id

        and t.ID = s.ID;

14. Afficher les noms des ´etudiants ayant suivi un cours dispens´e par un enseignant nomm´e Eistein (´eliminer les doublons).

    select distinct s.name from student s

    join takes t on s.id = t.id

    join course co on t.course_id = co. course_id

    join teaches te on te.course_id = t.course_id

    join teacher ter on te.ID= ter.ID

    where ter.name='Einstein'

15. Afficher tous les identifiants des cours et les enseignants qui les ont assur´es.

```
select co.course_id, t.name
from course co
join teaches te on co.course_id = te.course_id
join teacher t on t.ID=te.ID
```

16. Afficher le nombre d'inscrits pour chaque enseignement propos´e au printemps 2010.

```
select count(s.ID)
from student s
join takes t on s.id = t.id
join section sec on t.course_id = sec.course_id
where sec.year = 2010 and sec.semester='Spring'
```

17. Afficher les noms des d´epartements et les salaires maximum de leurs enseignants.

```
select dept_name, max(salary)
from teacher
group by dept_name
```

18. Afficher le nombre d'inscrits pour chaque enseignement propos´e.

```
select sec.course_id, count(s.ID)
from student s
join takes t on s.id = t.id
join section sec on t.course_id = sec.course_id
group by sec.course_id
```

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

```
select sec.building, count(sec.course_id)
from section sec
group by sec.building
```

20. Afficher le nombre total de cours dispens´es par chaque d´epartement et qui ont eu dans le mˆeme bˆatiment qui l'abrite.

21. Afficher les titres des cours propos´es et qui ont eu lieu et les enseignants qui les ont assur´es.

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

23. Afficher pour chaque ´etudiant le nombre total de cr´edits qu'il a obtenu, en suivant des cours qui n'ont pas ´et´e propos´e par son d´epartement.

24. Pour chaque d´epartement, afficher le nombre total de cr´edits des cours qui ont eu lieu dans ce d´epartement.