

Examens Base de donnée : SQL

Schoreels Jonathan et Poncelet Jeremy : 2ème année de bachelier en Informatique.

28/12/10

Références des énoncés.

R/A	00	01	02	03	04	05	07	08	09	10
Rep1	Q1	/	/	/	Q1	/	/	/	Q1	/
Rep2	Q3	/	/	/	Q2	/	Q2	/	Q3	/
Rep3	Q5	/	/	/	Q3	/	/	/	/	/
Rep4	Q7	/	/	/	Q4	/	/	/	/	/
Rep5	/	/	/	/	Q5	/	/	Q1	/	Q1
Rep6	/	/	Q2	/	Q6	/	/	/	/	/
Rep7	/	Q1,2	/	/	Q7	/	/	Q4	/	Q4
Rep8	/	Q6	Q4	/	Q8	/	/	/	/	/
Rep9	/	/	Q3	/	/	/	/	Q2	/	Q2
Rep10	/	/	/	Q1	/	/	/	Q3	/	Q3
Rep11	/	/	/	/	/	Q1	/	Q5	/	Q5
Rep12	/	/	/	/	/	Q2	/	Q6	/	Q6
Rep13	/	/	/	/	/	Q7	/	Q7	/	Q7
Rep14	/	/	/	/	/	/	Q1	/	/	/
Rep15	/	Q4	/	/	/	/	Q3	/	/	/
Rep16	/	Q5	/	/	/	/	Q4	/	/	/
Rep17	Q4	/	/	Q6	/	/	Q5	/	/	/
Rep18	/	/	/	/	/	/	Q6	/	/	/
Rep19	/	/	/	/	/	/	Q7	/	/	/
Rep20	/	Q3	/	/	/	/	Q8	/	/	
Rep21	Q2	/	/	/	/	/	/	/	Q2	
Rep22	Q6	/	/	/	/	/	/	/	/	/
Rep23	Q8	/	/	/	/	/	/	/	/	/
Rep24	/	/	Q1	Q3	/	/	/	/	/	/
Rep25	/	/	Q5	/	/	/	/	/	/	/
Rep26	/	/	Q6	/	/	Q4	/	/	/	/
Rep27	/	/	/	Q2	/	/	/	/	/	/
Rep28	/	/	/	Q4	/	/	/	/	Q5	/
Rep29	/	/	/	Q5	/	Q6	/	/	/	/
Rep30	/	/	/	Q7	/	/	/	/	/	/
Rep31	/	/	/	/	/	Q3	/	/	/	/
Rep32	/	/	/	/	/	Q5	/	/	Q4	/

Questions non présentes :

Énoncés

1. Afficher les noms des employés du département 30, avec la première lettre en majuscule, ainsi que leur commission. Pour ceux qui ne touchent pas de commission, indiquer "pas de commission".
2. Afficher le nombre d'employés total, ainsi que le nombre de ceux engagés en 2000, en 2001, en 2002 et en 2003. Donner aux colonnes correspondantes les titres appropriés.
3. Modifier la table salgrade pour que le grade 5 corresponde au salaire maximum de 4000. Seule la partie a) ici présente. La partie b) concerne un script en Oracle !
4. Créer une table "staff".

```
SQL > Describe staff
```

Name	Type
DEPT_ID	INTEGER(2)
DEPT_NAME	VARCHAR(13)
STAFF_LIST	VARCHAR(100)

Introduire dans cette table la liste des numéros et noms des départements avec en regard, la liste des noms des employés qui en font partie, sur une ligne, classés par salaire décroissant (entre parenthèses).

5. Afficher les noms des départements, le lieu où ils sont situés, le nombre d'employés qui y travaillent et le salaire moyen du département (arrondi à deux décimales).
6. Afficher la liste des emplois pour les départements 10, 30 et 20 dans cet ordre (afficher l'emploi et le numéro de département).
7. Créer une vue ("view") représentant les noms, jobs, noms de département, salaires et grades des employés qui touchent une commission.

Dans la vue emp_v , modifier le job de Ward de 'SALESMAN' en 'CLERK' et son département de "SALES" en "RESEARCH".
8. Créer une vue donnant les noms et dates d'embauche des 5 derniers employés engagés.
9. Afficher les noms des employés qui gagnent moins que la moitié du salaire moyen de leur département.
10. Afficher les noms des employés dont les noms font exactement 4 lettres, avec la première lettre en majuscule, ainsi que leur manager. Pour celui qui n'a pas de manager, indiquer "Aucun".
11. Afficher les noms des employés qui ont un ou plusieurs collègues dans leur département qui ont été engagés après eux mais qui ont un salaire supérieur.

12. Sans clause LIMIT, afficher les quatre premiers employés par ordre alphabétique parmi ceux qui ne touchent pas de commission.
13. Afficher sous la forme de matrice, la somme consacrée sous forme de salaires par département et par job, ainsi que le total des départements pour chaque job (on considère savoir que seuls 3 départements sont concernés).
14. Afficher les numéros, noms et salaires de tous les employés qui gagnent plus que le salaire moyen et travaillent dans un département où il y a un employé qui a un 'T' dans son nom.
15. Afficher les jobs qui existent dans les départements 10 ou 20 mais pas dans les deux.
16. Créer une table \commlist".

```
MGR_ID INTEGER(4)
MGR_ENAME VARCHAR(14)
COMM_LIST VARCHAR(120)
```

Introduire dans cette table la liste des numeros et noms des managers avec, en regard, la liste des noms, des employés qui sont sous leurs ordres, sur ligne, avec leur commission entre parenthèses (indiquer "None" pour ceux qui n'en reçoivent pas)

17. Créer une vue ("view") qui présente les numéros et noms des employés, ainsi que leur lieu de travail, en se limitant à ceux qui travaillent dans le même département que le président ou reçoivent une commission.
18. Créer une vue donnant les noms et dates d'embauche des 3 derniers employés engages dont le grade est 4. (salaire en 2001 et 3000)
19. Afficher la liste des employés dont le nom commence par 'C' ou finit par 'S' pour les départements 30, 10 et 20 dans cet ordre (afficher le nom de l'employé et le numéro de département).
20. Afficher les noms, jobs et salaires des employés dont le salaire est supérieur à celui de tous les clerks, en ordre de salaire décroissant.
21. Afficher les noms et dates d'engagement des employés, en regard de celle de leur manager, pour les employés engagés avant leur manager.
22. Trouver le travail pour lequel quelqu'un a été engagé dans la seconde moitié de 2001 et pour lequel quelqu'un a également été engagé dans la seconde moitié de 2002.
23. Dans la table "dept", modifier les noms des départements de manière à faire apparaître avec seulement une majuscule initiale et non tout en majuscule.
24. Imaginez la suite d'instructions SQL suivante. Quel est le résultat des 2 dernières (nombre de rangées dans chaque table) ?

```
SQL > select count(*) from emp ;
COUNT(*) = 14
```

```
SQL >select count(*) from salgrade;
COUNT(*) = 5
```

```
SQL >savepoint ex;
```

```

SQL >delete from emp;
SQL >truncate table salgrade;
SQL >rollback to ex;
SQL >select count(*) from emp;
SQL >select count(*) from salgrade;

```

25. Afficher les noms des employés du département 10, avec la première lettre en majuscule, ainsi que leur manager. Pour celui qui n'a pas de manager, indiquer "Aucun".
26. Créer une table \staff"

```

GRADE INTEGER(2)
STAFF_LIST VARCHAR(100)

```

Introduire dans cette table la liste des grades avec, en regard, la liste des noms des employés qui ont un salaire correspondant à ce grade, sur une ligne, classés par salaire décroissant

27. Afficher la liste des employés qui ont un \I" dans leur nom pour les départements 10, 30 et 20 dans cet ordre (afficher le nom de l'employé et le numéro de département).
 28. Créer une vue (SQL standard, sans clause LIMIT) donnant les noms et dates d'embauche des 4 derniers employés engagés à Chicago.
 29. Créer une vue (\view") qui présente les noms et jobs des employés qui ont un salaire qui diffère de plus de 20% du salaire moyen pour leur job.
 30. Créer une table \commlist".
- ```

MGR_ID INTEGER(4)
MGR_ENAME VARCHAR(14)
NUM_EMP INTEGER(2)
STAFF_LIST VARCHAR(120)

```
- Introduire, dans cette table, pour chaque "manager", son numéro et son nom, le nombre d'employés sous ses ordres (i.e dont il est manager) et, en regard, sur une ligne, la liste de leurs noms (par ordre alphabétique) et de leurs numéros (entre parenthèses).
31. Classer les employés par salaire de leur manager. Pour un même salaire, les classer par ordre alphabétique.
  32. Créer une vue ("view") qui présente les numéros et noms des employés, ainsi que leur lieu de travail, en se limitant à ceux qui travaillent dans le même département que Turner ou ont un salaire supérieur à celui de Blake.

Modifier le lieu de travail (et seulement le sien) de Scott en "New York".

# Réponses

```
Notez qu'on pourrait simplement mettre hiredate au lieu de
DATE_FORMAT .. à vous de voir, c'est selon les goûts.
En fait, le DATE_FORMAT permet d'avoir un affichage comme en
Oracle .. Donc à vous de choisir, si vous préférez l'affichage
qu'on en avait en TP, ou si vous voulez changer l'allure de la
date. Donc à vous de choisir, si vous préférez l'affichage qu'on
en avait en TP, ou si vous voulez changer l'allure de la date.
#
J'ai préféré le laisser, dans l'optique où il pourrait par
hasard demandé une forme de date plutôt similaire.

1. SELECT CONCAT(LEFT(ename,1),RIGHT(LOWER(ename),LENGTH(ename)-1)) "Nom",
 IFNULL(comm,"Pas de commission") COMM
FROM emp
WHERE deptno=30;

2. SELECT COUNT(c.empno) "total", d.empno "Emp2000", e.empno "Emp2001", f.empno "Emp2002",
 g.empno "2003"
FROM emp c,
 (SELECT COUNT(empno) empno FROM emp WHERE hiredate LIKE '2000-%') d,
 (SELECT COUNT(empno) empno FROM emp WHERE hiredate LIKE '2001-%') e,
 (SELECT COUNT(empno) empno FROM emp WHERE hiredate LIKE '2002-%') f,
 (SELECT COUNT(empno) empno FROM emp WHERE hiredate LIKE '2003-%') g
GROUP BY d.empno, e.empno, f.empno, g.empno;

3.
Partie a)

UPDATE salgrade
SET hisal = 4000
WHERE grade = 5;

Partie b) (CECI EST EN ORACLE. Ne pas l'utiliser pour un examen mysql !)

accept l_sal prompt 'Enter a low sal'
accept h_sal prompt 'Entre a high sal'
insert into salgrade
values (sal_sex.nextval, to_number(& l_sal),to_number(& h_sal))

4. (note les types NUMBER et VARCHAR2 ne font pas partie de mysql mais d'oracle ;))

#creation de la table

CREATE TABLE staff
(DEPT_ID INTEGER(2),
DEPT_NAME VARCHAR(13),
```

```

 STAFF_LIST VARCHAR(100)
) engine=innodb;

#insertion des informations demandées

INSERT INTO staff(DEPT_ID,DEPT_NAME,STAFF_LIST)
SELECT d.deptno, d.dname, GROUP_CONCAT(CONCAT(e.ename,'(',e.sal,')'))
 ORDER BY e.sal desc
 SEPARATOR ' ')

FROM dept d, emp e
WHERE d.deptno = e.deptno
GROUP BY d.deptno;

5. SELECT d.dname AS 'DNAME',d.loc AS 'LOC',COUNT(e.ename) AS 'Number of people',
 ROUND(AVG(e.sal),2) AS 'Salary'
 FROM emp e,dept d
 WHERE e.deptno = d.deptno
 GROUP by e.deptno;

6. (SELECT DISTINCT job,deptno
 FROM emp
 WHERE deptno <> 20)
 UNION (
 SELECT DISTINCT job, deptno
 FROM emp
 WHERE deptno = 20)
 ORDER BY FIELD(deptno,10,30,20), job;

7. CREATE VIEW emp_v AS
 SELECT e.ename AS ENAME ,e.job AS JOB,d.dname AS DNAME,e.sal AS SAL,s.grade AS GRADE
 FROM emp e,dept d,salgrade s
 WHERE ((e.comm IS NOT NULL)
 AND (e.deptno = d.deptno)
 AND (s.grade =(SELECT grade FROM salgrade WHERE (losal <= e.sal) and (hisal > e.sal))));

UPDATE emp
SET job = 'CLERK', deptno = 20
WHERE ename = 'WARD';

On modifie directement emp ! Si on modifiait emp_v, il n'y aurait aucun soucis
pour le job.. Mais le fait de changer le DNAME, changerait le DNAME associé
au deptno dans la table dept ! Ce qui changerait le DNAME de tous les autres
employés du département 20 !

8.
CREATE VIEW emp_v AS
SELECT e.ename AS ENAME,
DATE_FORMAT(e.hiredate,'%d-%b-%y') AS HIREDATE
FROM emp e
WHERE 5 > (SELECT COUNT(*) FROM emp d WHERE d.hiredate > e.hiredate)

9. SELECT e.ename
 FROM emp e
 WHERE e.sal < (SELECT (0.5 * AVG(b.sal)) FROM emp b WHERE b.deptno = e.deptno)

10. SELECT CONCAT(LEFT(ename,1),RIGHT(LOWER(ename),LENGTH(ename)-1)) AS Nom,
 IFNULL(mgr,'aucun') mgr
 FROM emp

```

```

WHERE LENGTH(ename)=4;

11. SELECT e.ename AS ename
 FROM emp e
 WHERE exists (SELECT b.ename FROM emp b
 WHERE ((b.deptno = e.deptno)
 AND (b.hiredate > e.hiredate)
 AND (b.sal > e.sal)));

12. SELECT e.empno AS empno,e.ename AS ename
 FROM emp e
 WHERE ((e.comm IS NULL)
 AND 4 > (SELECT COUNT(*) FROM emp b WHERE ((b.comm IS NULL) AND b.ename < e.ename)))
 ORDER BY ename;

13. SELECT job, SUM(CASE deptno WHEN 10 THEN sal ELSE 0 END) 'Dept 10',
 SUM(CASE deptno WHEN 20 THEN sal ELSE 0 END) 'Dept 20',
 SUM(CASE deptno WHEN 30 THEN sal ELSE 0 END) 'Dept 30',
 SUM(sal) 'Total'
 FROM emp
 GROUP BY job;

14. SELECT e.empno, e.ename, e.sal
 FROM emp e
 WHERE e.sal > (SELECT AVG(sal) FROM emp)
 AND e.deptno IN (SELECT deptno FROM emp WHERE ename LIKE '%T%');

15. SELECT distinct e.job
 FROM emp e
 WHERE e.deptno IN (10, 20)
 AND NOT EXISTS (SELECT f.job
 FROM emp f
 WHERE e.job = f.job
 AND e.deptno <> f.deptno
 AND f.deptno IN (10,20));

16. # Création de la table

CREATE TABLE commlist
(MGR_ID INTEGER(4),
 MGR_NAME VARCHAR(14),
 COMM_LIST VARCHAR(120)
) engine=innodb;

Insertion des lignes

INSERT INTO commlist (MGR_ID,MGR_NAME,COMM_LIST)
SELECT e.mgr, m.ename, GROUP_CONCAT(CONCAT(LEFT(e.ename,1),
 RIGHT(LOWER(e.ename), LENGTH(e.ename)-1),
 '(',
 IFNULL(e.comm,'None'),
 ')'))
 ORDER BY e.comm desc
 SEPARATOR ' '

FROM emp m, emp e
WHERE m.empno = e.mgr
GROUP BY e.mgr;

```



```

17. # Creation de la table
CREATE view emp_v (empno, ename, loc) AS
SELECT e.empno, e.ename, d.loc
FROM emp e, dept d
WHERE e.deptno = d.deptno
 AND (e.deptno = (SELECT deptno FROM emp WHERE job = 'PRESIDENT')
 OR e.comm IS NOT NULL);

Modification du lieu de travail de Martin

UPDATE emp
SET deptno = (SELECT deptno
 FROM dept
 WHERE loc = 'NEW YORK')
WHERE ename = 'MARTIN';

18.
CREATE VIEW emp_v (ename, hiredate) AS
SELECT e.ename, DATE_FORMAT(e.hiredate,'%d-%b-%y') AS hiredate
FROM emp e
WHERE e.sal BETWEEN (SELECT losal FROM salgrade WHERE grade = 4)
 and
 (SELECT hisal FROM salgrade WHERE grade = 4)
AND (SELECT count(*)
 FROM emp f
 WHERE f.sal BETWEEN (SELECT losal FROM salgrade WHERE grade = 4)
 and
 (SELECT hisal FROM salgrade WHERE grade = 4)
 AND f.hiredate > e.hiredate) < 3;

19. SELECT ename, deptno
FROM emp
WHERE ename like 'C%' or ename like '%S'
ORDER BY FIELD(deptno,'30','10','20'), ename;

20.
SELECT ename, job, sal
FROM emp
WHERE sal > all (SELECT sal
 FROM emp
 WHERE job = 'CLERK')
ORDER BY sal DESC;

21. SELECT e.ename 'Employee',
 DATE_FORMAT(e.hiredate,'%d-%b-%y') 'Emp Hiredate',
 g.ename 'Manager',
 DATE_FORMAT(g.hiredate,'%d-%b-%y') 'Mgr Hiredate'
FROM emp e, emp g
WHERE e.mgr = g.empno
 AND e.hiredate < g.hiredate;

22. # l'énoncé a été adapté pour être valable sur nos BD actuelles (2001 et non 1981).

SELECT DISTINCT e.job
FROM emp e
WHERE EXISTS (SELECT f.ename
 FROM emp f
 WHERE f.job = e.job

```

```

 AND f.hiredate BETWEEN '2001-07-01' and '2001-12-31')
 AND EXISTS (SELECT g.ename
 FROM emp g
 WHERE g.job = e.job
 AND g.hiredate BETWEEN '2002-07-01' and '2002-12-31');

23. UPDATE dept SET dname = CONCAT(LEFT(dname,1),
 RIGHT(LOWER(dname), LENGTH(dname)-1));

24. L'état attendu serait que le rollback annule les modifications. Or,
 Truncate étant une opération de plus bas niveau (elle désalloue
 les blocs mémoires relatifs aux lignes de la table), elle effectue
 automatiquement un commit après l'opération.
 Le fait de faire un truncate après le delete, va donc lancer un
 commit qui empêchera TOUT RETOUR EN ARRIERE, et ce pour toutes les
 tables de notre DB ! Le résultat final est donc que emp est vide,
 et salgrade aussi, après le rollback (Qui soit dit en passant,
 déclenchera une erreur, vu que les savepoints antérieurs au dernier
 commit sont supprimés).

25. SELECT CONCAT(LEFT(ename,1),RIGHT(LOWER(ename),LENGTH(ename)-1)) AS Nom,
 IFNULL(mgr,'aucun') mgr
 FROM emp
 WHERE deptno = '10';

26.
 # Creation de la table
 CREATE TABLE staff (
 GRADE INTEGER(2),
 STAFF_LIST VARCHAR(100)
) engine=innodb;

 # Insertion des lignes

 INSERT INTO staff (GRADE, STAFF_LIST)
 SELECT s.grade, GROUP_CONCAT(CONCAT(LEFT(e.ename,1),
 RIGHT(LOWER(e.ename), LENGTH(e.ename)-1),
 '(',
 e.sal,
 ')')
 ORDER BY e.sal desc
 SEPARATOR ' ')
 FROM emp e, salgrade s
 WHERE s.grade = (SELECT g.grade
 FROM salgrade g
 WHERE e.sal BETWEEN g.losal and g.hisal)
 GROUP BY s.grade;

27. SELECT e.ename, e.deptno
 FROM emp e
 WHERE ename LIKE '%I%'
 ORDER BY FIELD (deptno,'10','30','20');

28. CREATE VIEW emp_v AS
 SELECT e.ename, e.hiredate
 FROM emp e, dept d
 WHERE e.deptno = d.deptno and d.loc = 'CHICAGO'
 AND 4 > (SELECT COUNT(*))

```

```

 FROM emp f, dept g
 WHERE e.hiredate < f.hiredate
 AND f.deptno = g.deptno and g.loc = 'CHICAGO') ;

29. CREATE VIEW emp_v (ename, job) AS
 SELECT e.ename, e.job
 FROM emp e
 WHERE e.sal NOT BETWEEN (SELECT AVG(f.sal) FROM emp f WHERE f.job = e.job) * 0.8
 AND (SELECT avg(f.sal) FROM emp f WHERE f.job = e.job) * 1.2;

30. # Creation de la table

 CREATE TABLE staff
 (MGR_ID INTEGER(4),
 MGR_NAME VARCHAR(14),
 NUM_EMP INTEGER(2),
 STAFF_LIST VARCHAR(120)
) engine=innodb;

 # Insertion des champs

 INSERT INTO staff (MGR_ID,MGR_NAME, NUM_EMP,STAFF_LIST)
 SELECT e.mgr, m.ename,
 count(e.ename), GROUP_CONCAT(CONCAT(e.ename,
 '(',
 e.empno,
 ')')
 ORDER BY e.ename
 SEPARATOR ' ') test

 FROM emp m, emp e
 WHERE m.empno = e.mgr
 GROUP BY e.mgr;

31. SELECT e.ename 'Employee', f.ename 'Manager', f.sal 'Manager Salary'
 FROM emp e, emp f
 WHERE e.mgr = f.empno
 ORDER BY f.sal, e.ename;

32. # Creation de la vue
 CREATE VIEW emp_v (empno, ename, loc) AS
 SELECT e.empno, e.ename, d.loc
 FROM emp e, dept d
 WHERE d.deptno = e.deptno
 AND (e.deptno = (SELECT deptno FROM emp WHERE ename = 'turner')
 OR
 e.sal > (SELECT sal FROM emp WHERE ename = 'Blake'));

 # Modification du lieu de travail de Scott.

 UPDATE emp SET deptno = (SELECT deptno
 FROM dept d
 WHERE d.loc = 'NEW YORK')
 WHERE ename = 'SCOTT';

```