

# Systemes de gestion de bases de donnees

## Exercices sur les fonctions standard et proprietaires (serie 4- Solutions)

Cours de Jacques THOORENS

Suite à une erreur dans le script fourni pour générer la table des étudiants, les dates de naissance antérieures à 1940 ont été interprétées comme situées entre 2000 et 2039. La commande suivante permet de les corriger. L'écriture de la nouvelle date aurait pu faire l'objet d'un exercice.

---

```
UPDATE Etudiants
SET Datenaiss = to_date(1900+to_char(Datenaiss,'YY')||
to_char(Datenaiss,'/MM/DD'),'YYYY/MM/DD');
```

---

1. Afficher la liste des étudiants en donnant l'initiale de leur prénom suivie d'un point et leur nom en minuscules (sauf la première lettre).

---

```
SELECT substr(prenom,1,1)||'._'||Initcap(Nom) AS etudiant
FROM etudiants;
```

---

2. Afficher la liste des années pour lesquelles nous avons des étudiants, par ordre croissant.

---

```
SELECT DISTINCT to_char(datenaiss,'YYYY')
FROM etudiants ORDER BY 1;
```

---

3. Afficher le nom des étudiants et leur date de naissance, en langage naturel (mercredi 7 janvier 1958).

---

```
SELECT * FROM
(SELECT nom, to_char(datenaiss,'day_d_month_YYYY')
FROM etudiants
order by 1)
WHERE ROWNUM<26;
```

---

4. Afficher la liste des étudiants d'anglais qui sont nés au cours du 4me trimestre de l'année.

---

```
SELECT nom, section, datenaiss
FROM etudiants
WHERE to_char(datenaiss,'Q')=4 AND section LIKE 'Anglais%';
```

---

5. Compter le nombre d'étudiants qui sont nés chaque jour de la semaine.

---

```

SELECT to_char(datenaiss, 'day'), Count (*)
FROM Etudiants
GROUP BY to_char(datenaiss, 'day');

```

---

Si on veut les jours dans l'ordre, on utilisera plutôt le format 'd'.

6. Afficher le nom et la moyenne des points obtenus pour les trois épreuves, ne gardant que deux décimales (avec la fonction `round()` et sans elle). La Solutions doit donner une valeur arrondie (susceptible d'être reprise dans des calculs ultérieurs) mais les valeurs entières peuvent ne pas avoir de décimales.

---

```

SELECT Nom,
round((Interro1+Interro2+Examen)/3,2),
floor((Interro1+Interro2+Examen)/3*100+0.5)/100
FROM Etudiants

```

---

7. Afficher le nom et la moyenne des points obtenus pour les trois épreuves, en donnant deux décimales. Donnez une Solutions d'affichage et une solution mathématiquement exacte (reprise de l'exercice précédent).

---

```

SELECT Nom,
to_char(round((Interro1+Interro2+Examen)/3,2), '0.00'),
to_char(floor((Interro1+Interro2+Examen)/3*100+0.5)/100, '0.00')
FROM Etudiants

```

---

8. Comptez le nombre d'étudiants dont le nom commence par chaque lettre de l'alphabet.

---

```

SELECT substr(Nom,1,1), count (Nom)
FROM etudiants
GROUP BY substr (Nom,1,1)
ORDER BY substr (Nom,1,1);

```

---

9. Afficher le nom et l'âge de chaque étudiant. Cette question apparemment simple est en fait assez difficile. Ce que j'attends ici est un âge entier et légal (exemple, si un règlement s'applique aux personnes d'âge minimum 18 ans, il ne s'applique pas pour quelqu'un la veille de son anniversaire). Le nombre de jours par année variant selon les années bissextiles, il n'est pas facile de calculer l'âge de quelqu'un au jour près.

Soit une requête exécutée le 20 janvier 2011 :

	NOM	DN	AGE
1	CHELKHA	27-01-1958	53
2	VERCHEVAL	19-03-1937	74
3	REZER	16-08-1955	56
4	GERON	06-01-1944	67
5	...	...	...

Le calcul sur l'année donne un résultat correct pour GERON (né le 6 janvier, il a déjà 67 ans) et faux pour tous les autres. Par exemple, CHELKA aura 53 ans le 27 janvier seulement.

En soustrayant l'année de naissance de l'année en cours, on obtient le résultat proposé dans l'énoncé. Il faut rajeunir d'un an ceux qui n'ont pas encore eu leur anniversaire cette année. Le format 'MMDD' donne un moyen simple de vérifier la condition.

---

```

SELECT nom,to_char(datenaiss,'DD-MM-YYYY') DN,
to_char(sysdate,'YYYY')-to_char(datenaiss,'YYYY')
- CASE WHEN to_char(sysdate,'MMDD')< to_char(datenaiss,'MMDD')
THEN 1
ELSE 0
END AS Age
FROM Etudiants;

```

---

## Remarque

On peut également utiliser une date autre que SYSDATE pour vérifier l'âge d'une personne à une date donnée (ce sera le cas pour certaines décisions légales).

**10.** Par tranche d'âges, comptez le nombre d'étudiant.

Comme nous devons faire figurer l'âge dans la clause SELECT et dans la clause GROUP BY, l'utilisation d'une requête imbriquée est une solution raisonnable.

---

```

Select Age, count (nom)
FROM (
  SELECT nom,to_char(datenaiss,'DD-MM-YYYY') DN,
to_char(sysdate,'YYYY')-to_char(datenaiss,'YYYY')
- CASE WHEN to_char(sysdate,'MMDD')< to_char(datenaiss,'MMDD')
THEN 1
ELSE 0
END AS Age
FROM Etudiants)
GROUP BY Age
ORDER BY 1;

```

---

**11.** Afficher la hiérarchie des collègues de Scott de manière à obtenir l'organigramme suivant (conseil ; créez d'abord une relation qui associe les chefs et les employés) :

	NC	NE
1	KING	dirige JONES
2	JONES	dirige SCOTT
3	SCOTT	dirige ADAMS
4	JONES	dirige FORD
5	FORD	dirige SMITH
6	KING	dirige BLAKE
7	BLAKE	dirige ALLEN
8	BLAKE	dirige WARD
9	BLAKE	dirige MARTIN
10	BLAKE	dirige TURNER
11	BLAKE	dirige JAMES
12	KING	dirige CLARK
13	CLARK	dirige MILLER

---

```

SELECT lpad('  ',4*(level-1),'  ')||nomchef nc,
'dirige_'||nomemp ne
FROM (
SELECT chef.ename as nomchef, e.ename nomemp
FROM emp chef
INNER JOIN emp e

```

```
ON e.mgr= chef.empno)
START with nomchef='KING'
CONNECT BY PRIOR nomemp=nomchef;
```

---

12. Afficher les notes de la première interrogation en langage naturel (un, deux, trois...).

---

```
SELECT Nom, Interro1, case Interro1
  WHEN 0 THEN 'zéro'
  WHEN 1 THEN 'un'
  WHEN 2 THEN 'deux'
  WHEN 3 THEN '_trois'
  WHEN 4 THEN 'quatre'
  WHEN 5 THEN 'cinq'
  WHEN 6 THEN 'six'
  WHEN 7 THEN 'sept'
  WHEN 8 THEN 'huit'
  WHEN 9 THEN 'neuf'
  WHEN 10 THEN 'dix' END AS note
FROM etudiants;
```

---

13. Afficher les points obtenus à l'examen par les étudiants en italien, en indiquant 'absent' pour ceux qui n'ont pas fait l'examen. Petit problème, la chaîne 'absent' est d'un type incompatible avec celui des notes.

---

```
SELECT section,nom, coalesce(to_char(examen),'absent')
FROM etudiants
WHERE section LIKE 'Italien%'
ORDER BY 1,2,3;
```

---

On notera l'astuce de `to_char` qui permet de mélanger les nombres des points de l'examen avec la chaîne 'absent'.

14. Calculez les points obtenus au total pour les étudiants en italien. Pour ceux qui n'ont pas fait l'examen, vous calculerez la moyenne sur base des deux interrogations.

---

```
SELECT nom, interro1, interro2, examen,
coalesce(interro1+interro2+examen, (interro1+interro2)/2*3)
FROM etudiants
WHERE section LIKE 'Ita%';
```

---

15. Affichez les étudiants en précisant leur prénom et leur premier prénom (on considère qu'il y a un second prénom lorsqu'il est séparé du premier par une espace, s'il n'y a pas d'espace, le premier prénom se confond avec le prénom entier).

---

```
SELECT prenom,
coalesce(substr(prenom,1,instr(prenom,'_',1)),prenom)
FROM etudiants;
```

---

16. Recherchez le second prénom des étudiants (placez un tiret pour ceux qui n'en ont pas).

---

```

SELECT prenom, coalesce(substr(prenom,1,instr(prenom,'_',1)),prenom),
CASE
WHEN instr(prenom,'_',1)>0 THEN substr(prenom,1+instr(prenom,'_',1))
ELSE '-' END
FROM etudiants;

```

---

**Remarque :** cette requête et la suivante ne fonctionnent pas dans l'interface de SQL Developer. Les totaux ne s'affichent pas. Il faut les lancer dans un navigateur Web.

SECTION	EXAMEN	NOMBRE
Allemand 1	4	11
	5	8
	6	6
	7	5
	8	7
*****		-----
Nombre Total		37
Allemand 2	4	4
	5	8
	6	4
	7	7
	8	8
*****		-----
Nombre Total		31
Allemand 3	3	2

TITRESERIE	THEME	INVITES
Bof Story	Bach m'horripile	1
	J'aime pas les maths	2
	Je déteste Maman	2
	Les filles m'ennuient	1
*****		-----
Nombre invités		6
C'est pas votre choix	Aller en vacances à Paris	2
	Critiquer son prof	2
	Dormir tout nu	1
	Manger du chocolat	2
*****		-----
Nombre invités		7
Fichue soirée	L'amour	2
	Le goût	2
	Le sommeil	2
	Les enfants	2
*****		-----
Nombre invités		8

**17.** Comptez les étudiants de chaque section, en les groupant par points à l'examen et en détaillant le nombre d'étudiants par groupe.

---

```

Break on section;
COMPUTE Sum label 'Nombre_Total' of Nombre on Section
SELECT Section, Examen, Count(*) as Nombre
FROM Etudiants
group by section,examen
Order BY 1,2

```

---

**18.** Comptez le nombre d'invités par émission et par série.

---

```

Break on TitreSerie
compute sum label 'Nombre_invités' of invites on TitreSerie
select TitreSerie,Theme,count(NomInv) invites
from emissions
inner join series using(idserie)
inner join participer using(idemission)
inner join invites using(idinvite)
group by TitreSerie, Theme
order by 1,2,3

```

---