
A

**Corrigé des
Exercices**

Corrigé de l'Exercice 1

1. Initialisez une session SQL*Plus avec votre ID et le mot de passe que votre instructeur vous a remis.
2. Les commandes SQL*Plus accèdent aux bases de données.
Faux
3. L'ordre SELECT suivant sera convenablement exécuté.
Vrai

```
SQL> SELECT rownum, ename, job, sal Salary
2 FROM emp;
```

4. L'ordre SELECT suivant sera convenablement exécuté.
Vrai

```
SQL> SELECT *
2 FROM salgrade;
```

5. Cet ordre comporte trois erreurs de code : pouvez-vous les trouver ?

```
SQL> SELECT empno, ename
2 salary x 12 ANNUAL SALARY
3 FROM emp;
```

- La table EMP ne contient pas de colonne "Salary", mais une colonne nommée "Sal".
 - L'opérateur de multiplication est *, et non pas x, comme indiqué à la ligne 2.
 - L'alias ANNUAL SALARY ne doit pas contenir d'espace. Vous devez écrire ANNUAL_SALARY ou l'inclure entre guillemets.
6. Affichez la structure de la table DEPT. Sélectionnez toutes les données de la table DEPT.

```
SQL> DESCRIBE dept
SQL> SELECT *
2 FROM dept;
```

7. Affichez la structure de la table EMP. Créez une requête pour afficher le nom (name), le poste (job), la date d'embauche (hire date) et le matricule (empno) de chaque employé, en plaçant le matricule en premier. Enregistrez votre ordre SQL dans un fichier nommé *plq7.sql*.

```
SQL> DESCRIBE emp
```

Corrigé de l'Exercice 6 (suite)

5. Affichez le numéro des départements dans lesquels on ne trouve pas de poste ANALYST.

```
SQL>SELECT deptno
  2 FROM dept
  3 MINUS
  4 SELECT deptno
  5 FROM emp
  6 WHERE job = 'ANALYST';
```

6. Affichez tous les postes des départements 10 et 20 qui n'existent que dans l'un ou l'autre de ces départements.

```
SQL>SELECT job
  2 FROM emp
  3 WHERE deptno in (20,10)
  4 MINUS
  5 (SELECT job
  6 FROM emp
  7 WHERE deptno = 10
  8 INTERSECT
  9 SELECT job
 10 FROM emp
 11 WHERE deptno = 20);
```

Corrigé de l'Exercice 6 (suite)

4. Affichez la liste des postes dans les départements 10, 30 et 20, en conservant cet ordre. Affichez le poste et le numéro du département.

```
col dummy noprint

SELECT job,deptno,'x' dummy
FROM emp
WHERE deptno = 10
UNION
SELECT job,deptno,'y'
FROM emp
WHERE deptno = 30
UNION
SELECT job,deptno,'z'
FROM emp
WHERE deptno = 20
ORDER BY 3
/
```

Corrigé de l'Exercice 6 (suite)

3. Ecrivez une requête composée pour afficher la liste des produits indiquant les pourcentages de remise, les identifiants des produits, ainsi que les prix réels nouveaux et anciens. Les produits dont le prix est inférieur à \$10 sont réduits de 10%, ceux dont le prix est compris entre \$10 et \$30 sont réduits de 15%, ceux dont le prix est supérieur à \$30 sont réduits de 20%, et ceux dont le prix est supérieur à \$40 ne sont pas réduits.

```
COL discount format a8

SELECT '10% off' discount,prodid, stdprice,stdprice *.9
      actprice
FROM price
WHERE stdprice < 10

UNION

SELECT '15% off',prodid, stdprice,stdprice * .85
FROM price
WHERE stdprice between 10 and 30

UNION

SELECT '20% off' ,prodid, stdprice,stdprice * .8
FROM price
WHERE stdprice > 30 and stdprice <= 40

UNION

SELECT 'no disc', prodid, stdprice,stdprice
FROM price
WHERE stdprice > 40

/
```

Corrigé de l'Exercice 6

1. Affichez le département qui ne comprend aucun employé.

```
SQL> SELECT deptno,dname
2   FROM dept
3  MINUS
4   SELECT e.deptno,d.dname
5   FROM emp e,dept d
6  WHERE e.deptno = d.deptno;
```

2. Retrouvez le poste qui était occupé dans la première moitié des années 1981 et 1982.

```
SQL> SELECT job
2   FROM emp
3  WHERE hiredate BETWEEN '01-JUN-81' AND '30-DEC-81'
4  INTERSECT
5   SELECT job
6   FROM emp
7  WHERE hiredate BETWEEN '01-JUN-82' AND '30-DEC-82';
```

Corrigé de l'Exercice 5 (suite)

Si vous souhaitez aller plus loin dans la difficulté, faites les exercices suivants :

11. Créez une requête pour afficher le nombre total d'employés puis, parmi ces employés, ceux qui ont été embauchés en 1980, 1981, 1982 et 1983. Nommez les colonnes de façon appropriée.

```
SQL> SELECT COUNT(*) total,  
2          SUM(DECODE(TO_CHAR(hiredate, 'YYYY'),  
3                      1980,1,0)) "1980",  
4          SUM(DECODE(TO_CHAR(hiredate, 'YYYY'),  
5                      1981,1,0)) "1981",  
6          SUM(DECODE(TO_CHAR(hiredate, 'YYYY'),  
7                      1982,1,0)) "1982",  
8          SUM(DECODE(TO_CHAR(hiredate, 'YYYY'),  
9                      1983,1,0)) "1983"  
10 FROM emp;
```

12. Créez une requête pour afficher les postes, le salaire de ces postes par numéro de département et le salaire total de ces postes incluant tous les départements. Nommez les colonnes de façon appropriée.

```
SQL> SELECT      job "Job",  
2          SUM(DECODE(deptno, 10, sal)) "Dept 10",  
3          SUM(DECODE(deptno, 20, sal)) "Dept 20",  
4          SUM(DECODE(deptno, 30, sal)) "Dept 30",  
5          SUM(sal) "Total"  
6 FROM emp  
7 GROUP BY job;
```

Corrigé de l'Exercice 5 (suite)

6. Ecrivez une requête pour afficher le nombre de personnes qui occupent le même poste.

```
SQL> SELECT  job, COUNT(*)
2  FROM      emp
3  GROUP BY  job;
```

7. Déterminez le nombre de managers sans en donner la liste. Nommez la colonne Number of Managers.

```
SQL> SELECT  COUNT(DISTINCT mgr) "Number of Managers"
2  FROM      emp;
```

8. Ecrivez une requête pour afficher la différence existant entre le salaire maximum et le salaire minimum. Nommez la colonne DIFFERENCE.

```
SQL> SELECT  MAX(sal) - MIN(sal) DIFFERENCE
2  FROM      emp;
```

Si vous avez le temps, faites les exercices suivants :

9. Affichez le matricule des différents managers et le niveau de salaire le plus bas de leurs employés. Excluez toute ligne où le manager n'est pas identifié. Excluez tout groupe dans lequel le salaire minimum est inférieur à \$1000. Triez les résultats par ordre décroissant des salaires.

```
SQL> SELECT  mgr, MIN(sal)
2  FROM      emp
3  WHERE      mgr IS NOT NULL
4  GROUP BY  mgr
5  HAVING     MIN(sal) > 1000
6  ORDER BY  MIN(sal) DESC;
```

10. Ecrivez une requête pour afficher le nom du département, la localisation, le nombre d'employés et le salaire moyen pour tous les employés de ce département. Nommez les colonnes dname, loc, Number of People et Salary, respectivement.

```
SQL> SELECT  d.dname, d.loc, COUNT(*) "Number of People",
2           ROUND(AVG(sal),2) "Salary"
3  FROM      emp e, dept d
4  WHERE      e.deptno = d.deptno
5  GROUP BY  d.dname, d.loc;
```


Corrigé de l'Exercice 5

Déterminez si les affirmations suivantes sont vraies ou fausses et entourez la réponse correspondante.

1. Les fonctions de groupe agissent sur plusieurs lignes pour produire un seul résultat.
Vrai
2. Les fonctions de groupe intègrent les valeurs NULL dans leurs calculs.
Faux. Les fonctions de groupe ignorent les valeurs NULL. Si vous voulez inclure des valeurs NULL, utilisez la fonction NVL.
3. La clause WHERE restreint les lignes avant qu'elles soient incluses dans un calcul de groupe.
Vrai
4. Affichez le salaire maximum, le salaire minimum, la somme des salaires et le salaire moyen de tous les employés. Nommez respectivement les colonnes Maximum, Minimum, Sum et Average.
Arrondissez les résultats à zéro décimale. Enregistrez votre ordre SQL dans un fichier nommé

p5q4.sql.

```
SQL> SELECT      ROUND(MAX(sal),0) "Maximum",  
2              ROUND(MIN(sal),0) "Minimum",  
3              ROUND(SUM(sal),0) "Sum",  
4              ROUND(AVG(sal),0) "Average"  
5 FROM          emp;
```

```
SQL> SAVE p5q4.sql
```

```
Created file p5q4.sql;
```

5. Modifiez le fichier *p5q4.sql* pour afficher le salaire minimum, le salaire maximum, la somme des salaires et le salaire moyen pour chaque type de poste. Enregistrez votre fichier sous *p5q5.sql*.

Exécutez à nouveau votre requête.

```
SQL> EDIT p5q6.sql  
SELECT      job, ROUND(MAX(sal),0) "Maximum",  
            ROUND(MIN(sal),0) "Minimum",  
            ROUND(SUM(sal),0) "Sum",  
            ROUND(AVG(sal),0) "Average"  
FROM          emp  
GROUP BY      job  
/
```

```
SQL> START p5q5.sql;
```

Corrigé de l'Exercice 4 (suite)

9. Affichez la structure de la table SALGRADE. Créez une requête pour afficher le nom, le poste, le département, le salaire et l'échelon de tous les employés.

```
SQL> DESCRIBE salgrade  
  
SQL> SELECT e.ename, e.job, d.dname, e.sal, s.grade  
2 FROM emp e, dept d, salgrade s  
3 WHERE e.deptno = d.deptno  
4 AND e.sal BETWEEN s.losal AND s.hisal;
```

Si vous souhaitez aller plus loin dans la difficulté, faites les exercices suivants :

10. Créez une requête pour afficher le nom et la date d'embauche de tous les employés arrivés après l'employé Blake.

```
SQL> SELECT emp.ename, emp.hiredate  
2 FROM emp, emp blake  
3 WHERE blake.ename = 'BLAKE'  
4 AND blake.hiredate < emp.hiredate;
```

11. Affichez les noms et date d'embauche des employés et de leur manager, pour tous les employés ayant été embauchés avant leur manager. Nommez les colonnes Employee, Emp Hiredate, Manager et Mgr Hiredate, respectivement.

```
SQL> SELECT e.ename "Employee", e.hiredate "Emp Hiredate",  
2 m.ename "Manager", m.hiredate "Mgr Hiredate"  
3 FROM emp e, emp m  
4 WHERE e.mgr = m.empno  
5 AND e.hiredate < m.hiredate;
```

12. Créez une requête pour afficher le nom des employés et leur salaire indiqué par des astérisques. Chaque astérisque représente cent dollars. Triez les données dans l'ordre décroissant des salaires. Nommez la colonne EMPLOYEE_AND_THEIR_SALARIES.

```
SQL> SELECT rpad(ename, 8) || rpad(' ', sal/100, '*')  
2 EMPLOYEE_AND_THEIR_SALARIES  
3 FROM emp  
4 ORDER BY sal DESC;
```

Corrigé de l'Exercice 4

1. Ecrivez une requête pour afficher le nom, le numéro de département et le département de tous les employés.

```
SQL> SELECT      e.ename, e.deptno, d.dname
  2 FROM          emp e, dept d
  3 WHERE         e.deptno = d.deptno;
```

2. Créez une liste unique de tous les postes du département 30.

```
SQL> SELECT      DISTINCT e.job, d.loc
  2 FROM          emp e, dept d
  3 WHERE         e.deptno = d.deptno
  4 AND          e.deptno = 30;
```

3. Ecrivez une requête pour afficher le nom, le département et la localisation de tous les employés qui touchent une commission.

```
SQL> SELECT      e.ename, d.dname, d.loc
  2 FROM          emp e, dept d
  3 WHERE         e.deptno = d.deptno
  4 AND          e.comm IS NOT NULL;
```

4. Affichez le nom et le nom du département pour tous les employés dont le nom contient la lettre A. Enregistrez votre ordre SQL dans un fichier que vous nommerez *p4q4.sql*.

```
SQL> SELECT      e.ename, d.dname
  2 FROM          emp e, dept d
  3 WHERE         e.deptno = d.deptno
  4 AND          e.ename LIKE '%A%';
```

5. Ecrivez une requête pour afficher le nom, le poste, le numéro de département et le département de tous les employés basés à DALLAS.

```
SQL> SELECT      e.ename, e.job, e.deptno, d.dname
  2 FROM          emp e, dept d
  3 WHERE         e.deptno = d.deptno
  4 AND          d.loc = 'DALLAS';
```

Corrigé de l'Exercice 4 (suite)

6. Affichez le nom et le matricule des employés et de leur manager. Nommez les colonnes Employee, Emp#, Manager, et Mgr#, respectivement. Enregistrez votre ordre SQL dans un fichier nommé *p4q6.sql*.

```
SQL> SELECT e.ename "Employee", e.empno "Emp#",
2         m.ename "Manager", m.empno "Mgr#"
3 FROM emp e, emp m
4 WHERE e.mgr = m.empno;
SQL> SAVE p4q6.sql.
Created file p4q6.sql;
```

7. Modifiez le fichier *p4q6.sql* pour afficher tous les employés, y compris King, n'ayant pas de manager. Enregistrez à nouveau dans un fichier *p4q7.sql*. Exécutez *p4q7.sql*.

```
SQL> EDIT p4q6.sql
SELECT e.ename "Employee", e.empno "Emp#",
        m.ename "Manager", m.empno "Mgr#"
FROM emp e, emp m
WHERE e.mgr = m.empno(+)
/
SQL> START p4q7.sql;
```

Si vous avez le temps, faites les exercices suivants :

8. Créez une requête pour afficher le numéro de département et le nom de tous les employés qui travaillent dans le même département qu'un certain employé. Donnez à chaque colonne un nom approprié.

```
SQL> SELECT e.deptno department, e.ename employee,
2         c.ename colleague
3 FROM emp e, emp c
4 WHERE e.deptno = c.deptno
5 AND e.empno <> c.empno
6 ORDER BY e.deptno, e.ename, c.ename;
```

Solutions de l'Exercice 3 (suite)

10. Affichez le nom, la date d'embauche ainsi que le jour de la semaine où l'employé a débuté. Nommez la colonne DAY. Classez les résultats dans l'ordre des jours de la semaine à partir du lundi (monday).

```
SQL> SELECT  ename, hiredate,
2           TO_CHAR(hiredate, 'DAY') DAY
3 FROM      emp
4 ORDER BY TO_CHAR(hiredate - 1, 'd');
```

Si vous souhaitez aller plus loin dans la difficulté, faites les exercices suivants :

11. Créez une requête pour afficher le nom et le montant de la commission de chaque employé. Pour les employés ne touchant aucune commission, affichez "No Commission". Nommez la colonne COMM.

```
SQL> SELECT  ename,
2           NVL(TO_CHAR(comm), 'No Commission') COMM
3 FROM      emp;
```

Corrigé de l'Exercice 3 (suite)

6. Affichez le nom de chaque employé et calculez le nombre de mois travaillés depuis la date d'embauche. Nommez la colonne MONTHS_WORKED. Classez les résultats en fonction du nombre de mois d'ancienneté. Arrondissez le nombre de mois au nombre entier supérieur.

```
SQL> SELECT      ename, ROUND(MONTHS_BETWEEN
2              (SYSDATE, hiredate)) MONTHS_WORKED
3 FROM          emp
4 ORDER BY      MONTHS_BETWEEN(SYSDATE, hiredate);
```

7. Ecrivez une requête affichant les informations suivantes pour chaque employé :
<employee name> earns <salary> monthly but wants <3 times salary>. Nommez la colonne "Dream Salaries".

```
SQL> SELECT  ename || ' earns '
2           || TO_CHAR(sal, 'fm$99,999.00')
3           || ' monthly but wants '
4           || TO_CHAR(sal * 3, 'fm$99,999.00')
5           || '.' "Dream Salaries"
6 FROM      emp;
```

Si vous avez le temps, faites les exercices suivants :

8. Créez une requête pour afficher le nom et le salaire de tous les employés. Le salaire sera formaté de façon à avoir 15 caractères de long, les blancs étant comblés à gauche par des \$. Nommez la colonne SALARY.

```
SQL> SELECT  ename,
2           LPAD(sal, 15, '$') SALARY
3 FROM      emp;
```

9. Ecrivez une requête pour afficher tous les noms d'employé commençant par les lettres J, A, ou M, ainsi que la longueur du nom. Le nom doit apparaître en minuscules, sauf l'initiale

qui sera en majuscules. Donnez à chaque colonne un nom approprié.

```
SQL> SELECT  INITCAP(ename) "Name",
2           LENGTH(ename) "Length"
3 FROM      emp
4 WHERE      ename LIKE 'J%'
5 OR         ename LIKE 'M%'
6 OR         ename LIKE 'A%';
```

Corrigé de l'Exercice 2 (suite)

Si vous en avez le temps, faites les exercices suivants :

10. Affichez le nom de tous les employés dont la troisième lettre du nom est un A.
Remarque : Deux caractères de soulignement (_) sont placés avant le A dans la clause WHERE.

```
SQL> SELECT      ename
2 FROM          emp
3 WHERE         ename LIKE ' _ A%';
```

11. Affichez le nom de tous les employés dont le nom contient deux L et travaillant dans le département 30 ou dont le manager est 7782.

```
SQL> SELECT      ename
2 FROM          emp
3 WHERE         ename LIKE '%L%L%'
4 AND          deptno = 30
5 OR           mgr = 7782;
```

Si vous voulez aller plus loin dans la difficulté, faites les exercices suivants :

12. Affichez le nom, le poste et le salaire de tous les 'CLERK' ou 'ANALYST' dont le salaire est différent de \$1000, \$3000 ou \$5000.

```
SQL> SELECT      ename, job, sal
2 FROM          emp
3 WHERE         job IN ('CLERK', 'ANALYST')
4 AND          sal NOT IN (1000, 3000, 5000);
```

13. Afficher le nom, le salaire et la commission de tous les employés dont le montant de commission est de plus de 10% supérieur au salaire. Enregistrez votre requête sous le nom *p2q13.sql*.

```
SQL> EDIT p2q6.sql
      SELECT      ename "Employee", sal "Monthly Salary", comm
      FROM        emp
      WHERE       comm > sal * 1.1
      /
SQL> START p2q13.sql;
```

Corrigé de l'Exercice 3

1. Ecrivez une requête pour afficher la date courante. Nommez la colonne Date.

```
SQL> SELECT sysdate "Date"
2 FROM dual;
```

2. Affichez pour chaque employé, le matricule, le nom, le salaire et le salaire augmenté de 15% sous la forme d'un nombre entier. Nommez cette colonne New Salary. Enregistrez votre ordre SQL dans un fichier appelé *p3q2.sql*.

```
SQL> SELECT empno, ename, sal,
2          ROUND(sal * 1.15, 0) "New Salary"
3 FROM emp;

SQL> SAVE p3q2.sql
Created file p3q2.sql;
```

3. Exécutez votre requête à partir du fichier *p3q2.sql*.

```
SQL> START p3q2.sql;
```

4. Modifiez votre requête *p3q2.sql* en ajoutant une colonne dans laquelle l'ancien salaire est soustrait du nouveau salaire. Nommez cette colonne Increase. Exécutez à nouveau votre requête.

```
SQL> EDIT p3q2.sql
SELECT empno, ename, sal,
        ROUND(sal * 1.15, 0) "New Salary",
        ROUND(sal * 1.15, 0) - sal "Increase"
FROM emp
/
SQL> START p3q2.sql;
```

5. Affichez le nom et la date d'embauche de chaque employé ainsi que la date de révision du salaire qui sera le premier lundi tombant après 6 mois d'activité. Nommez la colonne REVIEW. Les dates devront apparaître dans le format suivant : "Sunday, the Seventh of September, 1981".

```
SQL> SELECT ename, hiredate,
2          TO_CHAR(NEXT_DAY(ADD_MONTHS(hiredate, 6),
3          'MONDAY'),
4          'fmDay, "the" Ddspth "of" MonthYYYY') REVIEW
5          FROM emp;
```


Corrigé de l'Exercice 2 (suite)

5. Affichez le nom et le numéro de département de tous les employés des départements 10 et 30, en classant les noms par ordre alphabétique.

```
SQL> SELECT      ename, deptno
  2 FROM          emp
  3 WHERE          deptno IN (10, 30)
  4 ORDER BY      ename;
```

6. Modifiez *p2q3.sql* pour afficher la liste des noms et salaires des employés gagnant plus de \$1500 et travaillant dans le département 10 ou 30. Nommez les colonnes Employee et Monthly Salary, respectivement. Enregistrez à nouveau votre ordre SQL dans un fichier nommé *p2q6.sql*. Réexécutez votre requête.

```
SQL> EDIT p2q3.sql
      SELECT      ename "Employee", sal "Monthly Salary"
      FROM        emp
      WHERE       sal > 1500
      AND         deptno IN (10, 30)
      /
SQL> START p2q6.sql;
```

7. Affichez le nom et la date d'embauche de chaque employé entré en 1982.

```
SQL> SELECT      ename, hiredate
  2 FROM          emp
  3 WHERE          hiredate LIKE '%82';
```

8. Affichez le nom et le poste de tous les employés n'ayant pas de manager.

```
SQL> SELECT      ename, job
  2 FROM          emp
  3 WHERE          mgr IS NULL;
```

9. Affichez le nom, le salaire et la commission de tous les employés qui perçoivent des commissions. Triez les données dans l'ordre croissant des salaires et des commissions.

```
SQL> SELECT      ename, sal, comm
  2 FROM          emp
  3 WHERE          comm IS NOT NULL
  4 ORDER BY      sal DESC, comm DESC;
```

Corrigé de l'Exercice 2

1. Créez une requête destinée à afficher le nom et le salaire des employés gagnant plus de \$2850. Enregistrez l'ordre SQL créé dans un fichier nommé *p2q1.sql*. Exécutez votre requête.

```
SQL> SELECT      ename, sal
  2 FROM          emp
  3 WHERE         sal > 2850;
```

```
SQL> SAVE p2q1.sql
Created file p2q1.sql;
```

2. Créez une requête destinée à afficher le nom et le numéro de département de l'employé dont le matricule est 7566.

```
SQL> SELECT      ename, deptno
  2 FROM          emp
  3 WHERE         empno = 7566;
```

3. Modifiez *p2q1.sql* de manière à afficher le nom et le salaire de tous les employés dont le salaire n'est pas compris entre \$1500 et \$2850. Enregistrez ce nouvel ordre SQL dans un fichier nommé *p2q3.sql*. Exécutez cette requête.

```
SQL> EDIT p2q1.sql

      SELECT      ename, sal
      FROM        emp
      WHERE       sal NOT BETWEEN 1500 AND 2850
      /

SQL> START p2q3.sql;
```

4. Affichez le nom, le poste et la date d'entrée (hire date) des employés embauchés entre le 20 février 1981 et le 1 mai 1981. Classez le résultat par date d'embauche croissante.

```
SQL> SELECT      ename, job, hiredate
  2 FROM          emp
  3 WHERE         hiredate BETWEEN '20-Feb-81' AND '01-May-81'
  4 ORDER BY     hiredate;
```

Corrigé de l'Exercice 1 (suite)

Si vous souhaitez aller plus loin dans la difficulté, faites l'exercice suivant :

12. Créez une requête pour afficher toutes les données de la table EMP dans une seule colonne d'affichage. Séparez chaque colonne par une virgule. Nommez la colonne d'affichage THE_OUTPUT.

```
SQL> SELECT empno || ',' || ename || ',' || job || ',' ||  
2 mgr || ',' || hiredate || ',' || sal || ',' ||  
3 comm || ',' || deptno THE_OUTPUT  
4 FROM emp;
```

Corrigé de l'Exercice 1 (suite)

```
SQL> SELECT empno, ename, job, hiredate
2 FROM emp;
SQL> SAVE plq7.sql
Wrote file plq7.sql
```

- 8 Exécutez la requête que vous avez placée dans le fichier *plq7.sql*.

```
SQL> START plq7.sql
```

- 9 Créez une requête pour afficher chaque type de poste (job) différent existant dans la table EMP.

```
SQL> SELECT DISTINCT job
2 FROM emp;
```

Si vous en avez le temps, faites les exercices suivants :

10. Editez *plq7.sql*. Donnez respectivement les noms suivants aux en-têtes de colonne : Emp #, Employee, Job, et Hire Date. Exécutez à nouveau votre requête.

```
SQL> GET plq7.sql
1 SELECT empno, ename, job, hiredate
2* FROM emp
SQL> 1 SELECT empno "Emp #", ename "Employee",
SQL> 2 job "Job", hiredate "Hire Date"
3i
SQL> SAVE plq7.sql REPLACE
Wrote file plq7.sql
SQL> START plq7.sql;
```

11. Affichez le nom concaténé avec le poste en les séparant par une virgule suivie d'un espace, puis nommez la colonne "Employee and Title".

```
SQL> SELECT ename||', '||job "Employee and Title"
2 FROM emp;
```

Corrigé de l'Exercice 7

1. Créez une requête pour afficher le nom et la date d'embauche de tous les employés travaillant dans le même département que Blake, à l'exclusion de Blake.

```
SQL> SELECT      ename, hiredate
  2 FROM          emp
  3 WHERE         deptno IN (SELECT      deptno
  4                                FROM          emp
  5                                WHERE         ename = 'BLAKE')
  6 AND           ename != 'BLAKE';
```

2. Créez une requête pour afficher le matricule et le nom de tous les employés qui gagnent plus que le salaire moyen. Triez les résultats par ordre décroissant des salaires.

```
SQL> SELECT      empno, ename
  2 FROM          emp
  3 WHERE         sal > (SELECT AVG(sal)
  4                                FROM emp)
  5 ORDER BY sal DESC;
```

3. Ecrivez une requête pour afficher le matricule et le nom de tous les employés qui travaillent dans le même département que tout employé dont le nom contient un T. Enregistrez votre ordre SQL dans un fichier nommé *p6q3.sql*.

```
SQL> SELECT      empno, ename
  2 FROM          emp
  3 WHERE         deptno IN (SELECT      deptno
  4                                FROM          emp
  5                                WHERE         ename LIKE '%T%');
SQL> SAVE p6q3.sql
Created file p6q3.sql
```

4. Affichez le nom, le numéro de département et le poste de tous les employés dont le département est situé à Dallas.

```
SQL> SELECT      ename, deptno, job
  2 FROM          emp
  3 WHERE         deptno IN (SELECT      deptno
  4                                FROM          dept
  5                                WHERE         loc = 'DALLAS');
```

Corrigé de l'Exercice 7 (suite)

5. Affichez le nom et le salaire de tous les employés dont le manager est King.

```
SQL> SELECT  ename, sal
2  FROM      emp
3  WHERE mgr IN (SELECT  empno
4                  FROM    emp
5                  WHERE   ename = 'KING');
```

6. Affichez le numéro de département, le nom et le poste de tous les employés travaillant dans le département des ventes (Sales).

```
SQL> SELECT  deptno, ename, job
2  FROM      emp
3  WHERE deptno IN (SELECT deptno
4                  FROM    dept
5                  WHERE   dname = 'SALES');
```

Si vous avez le temps, faites les exercices suivants.

7. Modifiez *p6q3.sql* afin d'afficher le matricule, le nom et le salaire de tous les employés qui gagnent plus que le salaire moyen et qui travaillent dans un département avec tout employé dont le nom contient un T. Enregistrez à nouveau votre requête sous le nom *p6q7.sql*, puis réexécutez-la.

```
SQL> EDIT p6q3.sql
SELECT empno, ename, sal
FROM emp
WHERE sal > (SELECT AVG(sal)
              FROM emp)
AND deptno IN (SELECT deptno
               FROM emp
               WHERE ename LIKE '%T%')
/
SQL> START p6q7.sql;
```

Corrigé de l'Exercice 8

1. Ecrivez une requête pour afficher le nom, le numéro de département et le salaire de tout employé dont le numéro de département et le salaire sont tous les deux équivalents au numéro de département et au salaire de n'importe quel employé touchant une commission.

```
SQL> SELECT      ename, deptno, sal
  2 FROM          emp
  3 WHERE          (sal, deptno) IN
  4                (SELECT      sal, deptno
  5                  FROM        emp
  6                  WHERE       comm IS NOT NULL);
```

X

Affichez le nom, le numéro de département et le salaire de tout employé dont le ^{comm}numéro de département et le salaire sont tous les deux équivalents au salaire et à la commission de n'importe quel employé basé à Dallas.

```
SQL> SELECT      ename, dname, sal
  2 FROM          emp e, dept d
  3 WHERE          e.deptno = d.deptno
  4 AND            (sal, NVL(comm,0)) IN
  5                (SELECT      sal, NVL(comm,0)
  6                  FROM        emp e, dept d
  7                  WHERE       e.deptno = d.deptno
  8                  AND         d.loc = 'DALLAS');
```

3. Créez une requête pour afficher le nom, la date d'embauche et le salaire pour tous les employés touchant le même salaire et la même commission que Scott.

```
SQL> SELECT      ename, hiredate, sal
  2 FROM          emp
  3 WHERE          (sal, NVL(comm,0)) IN
  4                (SELECT      sal, NVL(comm,0)
  5                  FROM        emp
  6                  WHERE       ename = 'SCOTT')
  7 AND            ename != 'SCOTT';
```

4. Créez une requête pour afficher les employés qui perçoivent un salaire supérieur à tout employé dont le poste est CLERK. Triez le résultat dans l'ordre décroissant des salaires.

```
SQL> SELECT      ename, job, sal
  2 FROM          emp
  3 WHERE          sal > ALL (SELECT sal
  4 FROM          emp
  5 WHERE          job = 'CLERK')
  6 ORDER BY      sal DESC;
```

Corrigé de l'Exercice 9

1. Écrivez une requête pour afficher les trois meilleurs salaires dans la table EMP. Affichez les noms des employés et leur salaire.

```
SQL>SELECT ename
2  FROM emp e
3  WHERE 3 > (SELECT COUNT(*)
4             FROM emp
5             WHERE e.sal > sal);
```

2. Recherchez tous les employés qui ne sont pas des responsables.
 - a. Utilisez d'abord l'opérateur EXISTS.

```
SQL>SELECT outer.ename
2  FROM emp outer
3  WHERE EXISTS (SELECT empno
4                 FROM emp inner
5                 WHERE inner.mgr = outer.empno);
```

- b. Pouvez-vous effectuer cette opération à l'aide de l'opérateur IN ? Pourquoi ?

```
SQL>SELECT outer.ename
2  FROM emp outer
3  WHERE outer.empno NOT IN (SELECT inner.mgr
                             FROM emp inner);
```

Cette solution est incorrecte. Comme la sous-requête sélectionne une valeur NULL, la requête entière ne ramènera aucune ligne. Toutes les conditions comparant une valeur NULL retournent un résultat NULL. Ainsi, chaque fois que des valeurs NULL risquent d'être intégrées à un ensemble de valeurs, *n'utilisez pas* NOT IN à la place de l'opérateur NOT EXISTS.

Corrigé de l'Exercice 9 (suite)

3. Ecrivez une requête pour rechercher tous les employés dont le salaire est supérieur au salaire moyen de leur département. Affichez le numéro de chaque employé, son salaire, son numéro de département et le salaire moyen du département.

```
SQL>SELECT e.ename ename,e.sal salary,
  2 e.deptno deptno,AVG(a.sal) dept avg
  3 FROM emp e, emp a
  4 WHERE e.deptno = a.deptno
  5 AND e.sal > (SELECT AVG(sal)
  6               FROM emp
  7               WHERE deptno = e.deptno)
  8 GROUP BY e.ename,e.sal,e.deptno
  9 ORDER BY AVG(a.sal);
```

4. Ecrivez une requête pour afficher les employés dont le salaire est inférieur à la moitié du salaire moyen de leur département.

```
SQL>SELECT ename
  2 FROM emp outer
  3 WHERE outer.sal < (SELECT avg(inner.sal/2)
  4                     FROM emp inner
  5                     WHERE inner.deptno = outer.deptno);
```

5. Ecrivez une requête pour afficher les employés ayant un ou plusieurs collègues de leur département dont les dates d'embauche sont postérieures aux leurs et dont les salaires sont plus élevés que les leurs.

```
SQL>SELECT ename
  2 FROM emp outer
  3 WHERE EXISTS (SELECT inner.empno
  4               FROM emp inner
  5               WHERE inner.deptno = outer.deptno
  6               AND inner.hiredate > outer.hiredate
  7               AND inner.sal > outer.sal);
```

Corrigé de l'Exercice 10

Indiquez si les affirmations ci-dessous sont vraies ou fausses :

1. Une variable de substitution à simple "et commercial" (&) n'affiche un message d'invite qu'une seule fois.

Vrai

Cependant, si la variable est définie, la variable de substitution à simple perluète n'affichera pas d'invite. Elle prend la valeur de la variable prédéfinie.

2. La commande ACCEPT est une commande SQL.

Faux

La commande ACCEPT est une commande SQL*Plus qui est émise à la suite du prompt SQL.

3. Ecrivez un fichier script pour afficher le nom, le poste et la date d'embauche de tous employés ayant commencé entre deux dates spécifiées. Concaténez le nom et le poste en séparant par une virgule et un espace, et nommez la colonne "Employees". Demandez à l'utilisateur d'entrer l'intervalle de dates au moyen de la commande ACCEPT. Utilisez le format MM/DD/YY. Enregistrez le fichier sous *p8q3.sql*.

```
SET ECHO OFF
SET VERIFY OFF
ACCEPT low_date DATE FORMAT 'MM/DD/YY' -
PROMPT 'Please enter the low date range (''MM/DD/YY'') : '
ACCEPT high_date DATE FORMAT 'MM/DD/YY' -
PROMPT 'Please enter the high date range (''MM/DD/YY'') : '
COLUMN EMPLOYEES FORMAT A25
SELECT      ename || ', ' || job EMPLOYEES, hiredate
FROM        emp
WHERE       hiredate BETWEEN
            TO_DATE('&low_date', 'MM/DD/YY')
            AND TO_DATE('&high_date', 'MM/DD/YY')

/
UNDEFINE low_date
UNDEFINE high_date
COLUMN EMPLOYEES CLEAR
SET VERIFY ON
SET ECHO ON
SQL> START p8q3.sql;
```

Corrigé de l'Exercice 10 (suite)

4. Écrivez un script pour afficher le nom, le poste et le nom du département des employés travaillant dans un département déterminé. La recherche doit pouvoir s'effectuer indifféremment sur des majuscules ou des minuscules. Enregistrez le fichier script sous *p8q4.sql*.

```
SET ECHO OFF
SET VERIFY OFF
ACCEPT p_location PROMPT 'Please enter the location name: '
COLUMN ename HEADING "EMPLOYEE NAME" FORMAT A15
COLUMN dname HEADING "DEPARTMENT NAME" FORMAT A15
SELECT e.ename, e.job, d.dname
FROM   emp e, dept d
WHERE  e.deptno = d.deptno
AND    LOWER(d.loc) LIKE LOWER('%&p_location%')
/
UNDEFINE p_location
COLUMN ename CLEAR
COLUMN dname CLEAR
SET VERIFY ON
SET ECHO ON
SQL> START p8q4.sql;
```

Corrigé de l'Exercice 10 (suite)

5. Modifiez *p8q4.sql* pour créer un état contenant le nom du département, le nom, la date d'embauche, le salaire et le salaire annuel de tous les employés d'une même localisation. Demandez à l'utilisateur d'entrer cette localisation. Nommez les colonnes DEPARTMENT NAME, EMPLOYEE NAME, START DATE, SALARY et ANNUAL SALARY, en répartissant les en-têtes sur plusieurs lignes. Enregistrez de nouveau le script sous le nom *p8q5.sql*.

```
SET ECHO OFF
SET FEEDBACK OFF
SET VERIFY OFF
BREAK ON dname
ACCEPT p_location PROMPT 'Please enter the location name:

COLUMN dname HEADING "DEPARTMENT|NAME" FORMAT A15
COLUMN ename HEADING "EMPLOYEE|NAME" FORMAT A15
COLUMN hiredate HEADING "START|DATE" FORMAT A15
COLUMN sal HEADING "SALARY" FORMAT $99,990.00
COLUMN asal HEADING "ANNUAL|SALARY" FORMAT $99,990.00
SELECT      d.dname, e.ename, e.hiredate,
            e.sal, e.sal * 12 asal
FROM        emp e, dept d
WHERE       e.deptno = d.deptno
AND         LOWER(d.loc) LIKE LOWER('%&p_location%')
ORDER BY   dname
/
UNDEFINE p_location
COLUMN dname CLEAR
COLUMN ename CLEAR
COLUMN hiredate CLEAR
COLUMN sal CLEAR
COLUMN asal CLEAR
CLEAR BREAK
SET VERIFY ON
SET FEEDBACK ON
SET ECHO ON
SQL> START p8q5.sql;
```

Corrigé de l'Exercice 10 (suite)

6. Les tâches de la colonne de gauche ne correspondent pas aux commandes de la colonne de droite. Tracez des lignes pour indiquer la commande appropriée à chaque tâche.

Tâche		Commande
Modifier les caractéristiques d'affichage d'une colonne		SET LINESIZE
Calculer et imprimer des lignes de totaux		TTITLE
Contrôler le nombre de caractères sur une page		COLUMN
Ajouter un titre à l'état		BREAK
Forcer un saut de page		SET NEWPAGE
Déterminer le nombre de lignes blanches à la fin d'une page et au début de la suivante		COMPUTE

Corrigé de l'Exercice 10 (suite)

7. Produisez un état semblable à celui ci-après. N'oubliez pas de mettre en forme les données, puis enregistrez votre travail dans un fichier nommé *p5_rep.sql*. Le fichier *p5_sel.sql* contient l'ordre SELECT sur lequel vous pouvez vous baser pour créer votre état.

```
set echo off
set feedback off
col today format a12 noprint
col dep format 9999 heading 'Dept'
col job format A9 heading 'Job'
col ename format A7 heading 'Name'
col sal format B99,999.99 heading 'Monthly|Salary'
col comm format 9,990.99 heading 'Annual|Comm'
col totsals format 999,999.99 heading 'Total'

select to_char(sysdate,'mm/dd/yy')today, d.deptno dep, job,
       ename,
       sal, comm, sal * 12 + nvl(comm,0) totsals
from emp e, dept d
where e.deptno = d.deptno
order by d.deptno,job
/

clear columns
set feedback on
set echo on
```

Corrigé de l'Exercice 10 (suite)

- 8 Améliorez la présentation de l'état que vous avez créé précédemment pour obtenir un état semblable à celui ci-après.

```
set echo off
set feedback off
set null ''
col today noprint new_value my_date
col dep format 9999 heading 'Dept'
col job format A9 heading 'Job'
col ename format A7 heading 'Name'
col sal format B99,999.99 heading 'Monthly|Salary'
col commi format 9,990.99 heading 'Annual|Comm'
col totalsal format 999,999.99 heading 'Total'

tttitle left my_date center 'Employee Report' -
right 'Page:' format 99 sql.pno skip 2
bttitle center 'Company Confidential'

break on report on dep skip 2 on job
compute sum label total of sal commi totalsal on dep report

select to_char(sysdate,'mm/dd/yy')today, d.deptno dep,
       job,
       ename,
       sal, nvl(comm,0) commi, sal * 12 + nvl(comm,0) totalsal
from emp e, dept d
where e.deptno = d.deptno
order by d.deptno,job
/

tttitle off
bttitle off
clear breaks
clear computes
clear columns
set feedback on
set echo on
```

Corrigé de l'Exercice 11

- I. Examinez les exemples suivants. Sont-ils le résultat d'une requête hiérarchique ? Expliquez pourquoi oui ou pourquoi non.

Exemple 1 : Requête non hiérarchique. L'état est trié en fonction de la colonne mgr.

EMPNO	ENAME	MGR	SAL	DEPTNO
7839	KING		5000	10
7902	FORD	7566	3000	20
7788	SCOTT	7566	3000	20
7566	JONES	7839	2975	20
7698	BLAKE	7839	2850	30
7782	CLARK	7839	2450	10
7499	ALLEN	7698	1600	30

Exemple 2 : Requête non hiérarchique impliquant deux tables.

EMPNO	MGR	DNAME	LOC
7698	7839	SALES	CHICAGO
7782	7839	ACCOUNTING	NEW YORK
7566	7839	RESEARCH	DALLAS

Exemple 3 : Oui, cette requête est hiérarchique.

RANK	ENAME	MGR
1	KING	
2	BLAKE	7839
3	MARTIN	7698
3	ALLEN	7698
3	TURNER	7698
3	JAMES	7698
3	WARD	7698

Corrigé de l'Exercice 11 (suite)

2. Créez un état représentant l'organigramme du département de Jones. Imprimez les noms des employés, leur salaire et leur numéro de département.

```
SQL>SELECT ename,sal deptno
2 FROM emp
3 CONNECT BY PRIOR empno = mgr
4 START WITH ename = 'JONES'
```

3. Créez un état dans lequel figurent les noms de tous les responsables pour lesquels travaille Adams.

```
SQL>SELECT ename
2 FROM emp
3 WHERE ename != 'ADAMS'
4 CONNECT BY PRIOR mgr = empno
5 START WITH ename = 'ADAMS';
```

4. Créez un état représentant la hiérarchie des dirigeants par une indentation. Affichez les noms des employés, le numéro de leur département ainsi que le numéro de leur responsable. Commencez par l'employé ayant le grade le plus élevé.

```
SQL>COL name FORMAT A20
SQL>SELECT LPAD(' ',3*LEVEL-3) || ename name,mgr,deptno
2 FROM emp
3 CONNECT BY PRIOR empno = mgr
4 START WITH mgr IS NULL;
```

Corrigé de l'Exercice 11 (suite)

S'il vous reste encore du temps, effectuez l'exercice suivant :

5. Créez l'organigramme d'une société représentant la hiérarchie des dirigeants. Commencez par la personne ayant le grade le plus élevé et excluez tous les employés occupant le poste ANALYST, ainsi que le département de CLARK.

```
SQL>SELECT ename,empno,mgr
2 FROM emp
3 WHERE job != 'ANALYST'
4 CONNECT BY PRIOR empno = mgr
5 AND ename != 'CLARK'
6 START WITH mgr IS NULL;
```