Edited with the trial version of
Foxit Advanced PDF Editor
To remove this notice, visit:
www.foxitsoftware.com/shopping

www.dirasats.com

هذا الغلاف لا يعبر عن حقوق الملكية او فحوى الكتاب ,فهو مجرد واجهة للموقع المحمل منه

شكرا لك على ثقتك بنا وعلى اختيار موقعنا

www.dirasats.com

من اجل تواصل معنا المرجو زيارة الموقع ستجد جميع المعلومات

www.dirasats.com

6

Opérateurs Ensemblistes

Objectifs

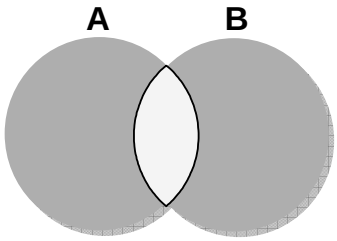
A la fin de ce chapitre, vous saurez :

- **Décrire les opérateurs ensemblistes**
- **Utiliser un opérateur ensembliste pour combiner plusieurs requêtes en une seule**
- **Vérifier l'ordre des lignes ramenées**

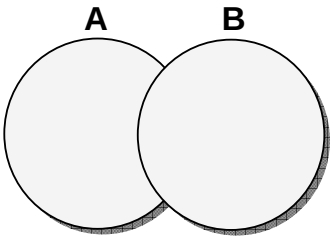
Objectifs

Au cours de ce chapitre, vous allez apprendre à écrire des requêtes avec des opérateurs ensemblistes.

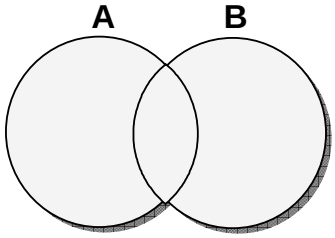
Opérateurs Ensemblistes

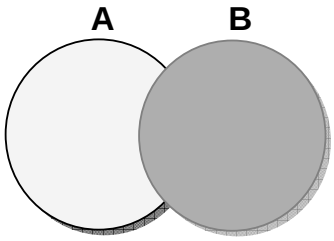


Intersect



Union / Union All





Minus

6-3

Présentation

Les opérateurs ensemblistes combinent les résultats de deux ou plusieurs requêtes en un seul résultat. Une requête composée est une requête contenant des opérateurs ensemblistes.

L'opérateur	**Ramène**
INTERSECT	Toutes les lignes communes aux deux requêtes. INTERSECT combine les deux requêtes et ramène les lignes du premier ordre SELECT identiques aux lignes du second ordre SELECT.
UNION	Toutes les lignes distinctes ramenées par les deux requêtes.
UNION ALL	Toutes les lignes sélectionnées par les deux requêtes, y compris les doublons.
MINUS	Toutes les lignes sélectionnées par le premier ordre SELECT moins les lignes sélectionnées dans le second ordre SELECT.

Tous les opérateurs ensemblistes ont la même priorité. Si un ordre SQL en contient plusieurs, la base de données les évalue de gauche à droite ou de haut en bas, si aucune parenthèse n'indique explicitement un autre ordre. Pour être conforme aux nouvelles normes SQL, la prochaine version de la base de données accordera une priorité plus importante à l'opérateur INTERSECT qu'aux autres opérateurs ensemblistes. Vous devez donc indiquer explicitement par des parenthèses l'ordre d'exécution des requêtes contenant l'opérateur INTERSECT et d'autres opérateurs ensemblistes.

Tables Utilisées dans ce Chapitre

EMP

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7839	KING	PRESIDENT		17-NOV-81	5000		10
7698	BLAKE	MANAGER	7839	01-MAY-81	2850		30
7782	CLARK	MANAGER	7839	09-JUN-81	1500		10
7566	JONES	MANAGER	7839	02-APR-81	2975		20
7654	MARTIN	SALESMAN					30
7499	ALLEN	SALESMAN					30
7844	TURNER	SALESMAN					30
7900	JAMES	CLERK					30
7521	WARD	SALESMAN					30
7902	FORD	ANALYST					20
7369	SMITH	CLERK					20

EMP_HISTORY

EMPID	NAME	TITLE	DATE_OUT	DEPTID
6087	SPENCER	OPERATOR	27-NOV-81	20
6185	VANDYKE	MANAGER	17-JAN-81	10
6235	BALFORD	CLERK	22-FEB-80	20
7788	SCOTT	ANALYST	05-MAY-81	20
7902	FORD	ANALYST	10-JUN-81	30

6-4

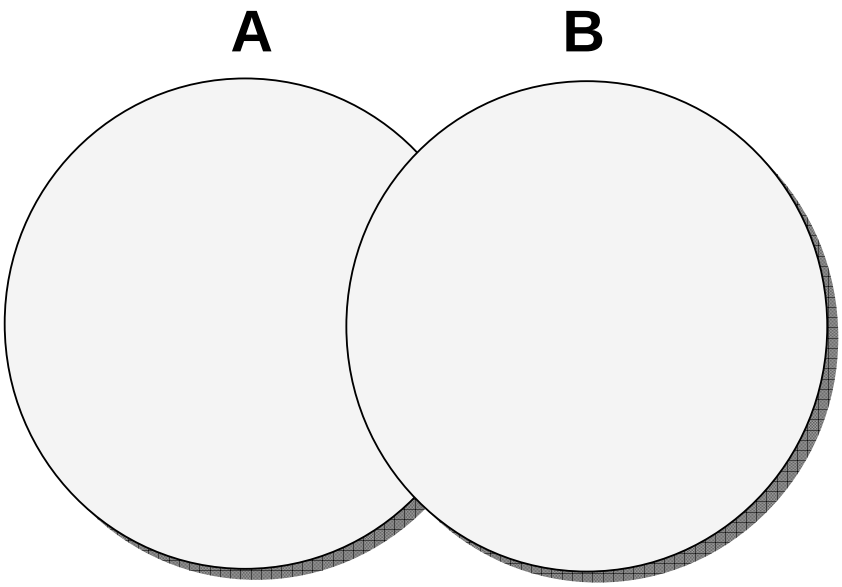
Tables utilisées dans ce chapitre

Deux tables sont utilisées dans ce cours :

- La table EMP donne des informations sur tous les employés
- EMP_HISTORY donne des informations sur les employés ayant quitté la société

Le script nécessaire à la création de la table EMP_HISTORY est indiqué dans l'exercice à la fin de ce chapitre.

UNION



L'Opérateur UNION

Cet opérateur combine le résultat de deux requêtes. Il permet de ramener toutes les lignes issues de plusieurs requêtes et d'éliminer les doublons.

Instructions

- Le nombre de colonnes et les types de données des colonnes doivent être identiques dans les deux ordres SELECT. En revanche, les noms de colonnes peuvent être différents.
- L'opérateur UNION intervient sur toutes les colonnes sélectionnées. Par exemple, si vous modifiez la requête de la page suivante pour sélectionner uniquement les noms des employés et leur poste, ALLEN n'apparaîtra qu'une seule fois dans les résultats.
- Les colonnes NULL sont ignorées lors du contrôle des doublons. Par exemple, si la colonne DEPTNO correspondant à ALLEN contenait une valeur NULL dans le premier ordre SELECT (si DEPTNO n'était pas une colonne NOT NULL) à la place de la valeur 30 comme dans le second ordre SELECT, ALLEN n'apparaîtrait qu'une seule fois dans les résultats.
- L'opérateur IN a une priorité plus élevée que l'opérateur UNION.
- Les requêtes incluant l'opérateur UNION dans la clause WHERE doivent comprendre le même nombre de colonnes et des colonnes du même type que celles de la clause SELECT.
- Par défaut, les données sont affichées par ordre ascendant.

Utilisation de l'Opérateur UNION

Affichez le nom, le poste et le département de tous les employés.

```
SQL> SELECT ename, job, deptno
  2  FROM   emp
  3  UNION
  4  SELECT name, title, deptid
  5  FROM emp_history;
```

```
ENAME      JOB           DEPTNO
---------- --------- ---------
ADAMS      CLERK             30
ALLEN      SALESMAN          30
ALLEN      SALESMAN          20
BALFORD    CLERK             20
BLAKE      MANAGER           30
...
20 rows selected.
```

6-6

L'Opérateur UNION

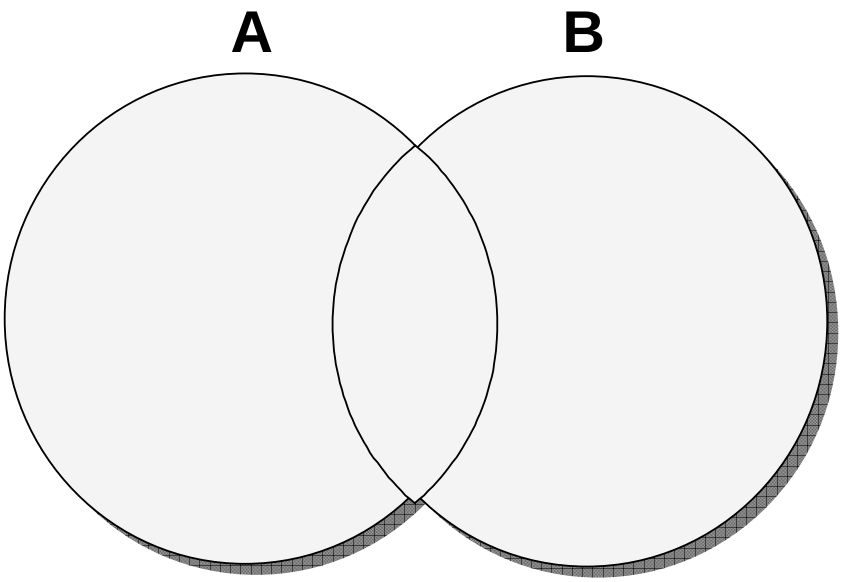
Dans la diapositive ci-dessus, 20 lignes ont été sélectionnées. Même si la combinaison des deux tables totalise plus de 20 enregistrements, seuls 20 sont ramenés, car l'opérateur UNION élimine tous les doublons.

Les tables EMP et EMP_HISTORY comprennent plusieurs colonnes en commun. Par exemple, ENAME et NAME, JOB et TITLE, EMPNO et EMPID. Que se passerait-il si vous souhaitiez afficher les noms des employés, leur poste et leur salaire à l'aide de l'opérateur UNION, sachant que leur salaire ne figure pas dans ces tables ? L'ordre suivant établit une correspondance entre les colonnes ENAME et NAME, JOB et TITLE, puis ajoute le littéral 0 dans l'ordre SELECT de la table EMP_HISTORY pour établir une correspondance avec la colonne numérique SAL dans l'ordre SELECT de la table EMP.

```
SELECT ename, job, sal FROM   emp
UNION
SELECT name, title, 0 FROM   emp_history;
```

```
ENAME      JOB              SAL
---------- --------- ---------
ADAMS      CLERK           1100
ALLEN      SALESMAN           0
ALLEN      SALESMAN        1600
BALFORD    CLERK              0
...
```

UNION ALL



L'Opérateur UNION ALL

Cet opérateur permet de ramener toutes les lignes issues de plusieurs requêtes.

Règles

- Contrairement à l'opérateur UNION, les doublons ne sont pas éliminés et le résultat n'est pas trié par défaut.
- Il n'est pas possible d'utiliser le mot-clé DISTINCT.

Remarque : Les règles relatives aux opérateurs UNION et UNION ALL sont les mêmes, excepté les deux points ci-dessus.

Utilisation de l'Opérateur UNION ALL

Affichez le nom, le numéro et le poste de tous les employés.

```
SQL> SELECT ename, empno, job
  2  FROM   emp
  3  UNION ALL
  4  SELECT name, empid, title
  5  FROM   emp_history;
```

```
ENAME          EMPNO JOB
---------- --------- ---------
KING            7839 PRESIDENT
BLAKE           7698 MANAGER
CLARK           7782 MANAGER
CLARK           7782 MANAGER
MARTIN          7654 SALESMAN
...
23 rows selected.
```

6-8

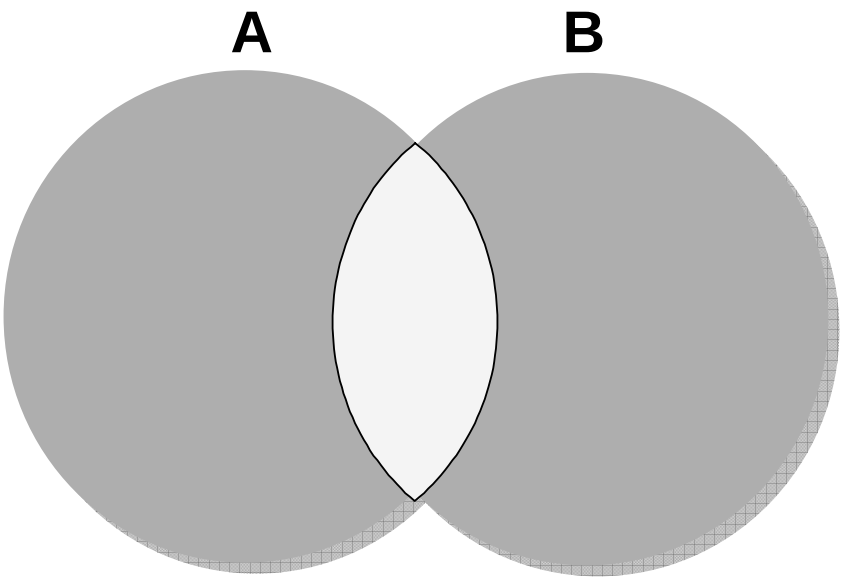
L'Opérateur UNION ALL (suite)

Dans l'exemple ci-dessus, 23 lignes ont été sélectionnées et la combinaison des deux tables totalise 23 enregistrements, ce qui prouve que l'opérateur UNION ALL n'élimine pas les doublons.

Le résultat de l'exemple ci-dessus contient trois groupes de doublons :

```
ENAME          EMPNO JOB
---------- --------- ---------
KING            7839 PRESIDENT
BLAKE           7698 MANAGER
CLARK           7782 MANAGER
CLARK           7782 MANAGER
MARTIN          7654 SALESMAN
ALLEN           7499 SALESMAN
TURNER          7844 SALESMAN
JAMES           7900 CLERK
SMITH           7369 CLERK
SCOTT           7788 ANALYST
ADAMS           7876 CLERK
MILLER          7934 CLERK
BALFORD         6235 CLERK
SCOTT           7788 ANALYST
JEWELL          7001 ANALYST
ALLEN           7499 SALESMAN
BRIGGS          7225 PAY CLERK
...
23 rows selected.
```

INTERSECT



6-9

L'Opérateur INTERSECT

Cet opérateur permet de ramener toutes les lignes communes aux deux requêtes.

- Le nombre de colonnes et les types de données des deux colonnes doivent être identiques dans les deux ordres SELECT. En revanche, les noms de colonnes peuvent être différents.
- L'inversion de l'ordre des tables interrogées ne modifie pas le résultat.
- Comme l'opérateur UNION, l'opérateur INTERSECT ignore les colonnes NULL.
- Les requêtes incluant l'opérateur INTERSECT dans la clause WHERE doivent comprendre le même nombre et des colonnes du même type que celles de la clause SELECT.

Utilisation de l'Opérateur INTERSECT

Affichez les différents noms, numéros et postes des employés présents dans les tables EMP et EMP_HISTORY.

```
SQL> SELECT ename, empno, job
  2  FROM   emp
  3  INTERSECT
  4  SELECT name, empid, title
  5  FROM   emp_history;
```

```
ENAME           EMPNO JOB
---------- --------- ---------
ALLEN            7499 SALESMAN
CLARK            7782 MANAGER
SCOTT            7788 ANALYST
```

6-10

L'Opérateur INTERSECT

Dans l'exemple ci-dessus, seuls les enregistrements ayant les mêmes valeurs dans les colonnes sélectionnées des deux tables sont ramenés par la requête.

Quels seraient les résultats si on ajoutait la colonne DEPTNO dans l'ordre SELECT de la table EMP et la colonne DEPTID dans l'ordre SELECT de la table EMP_HISTORY et si on exécutait la requête ? Les résultats seraient différents en raison de l'ajout d'une autre colonne dont les valeurs peuvent être des doublons.

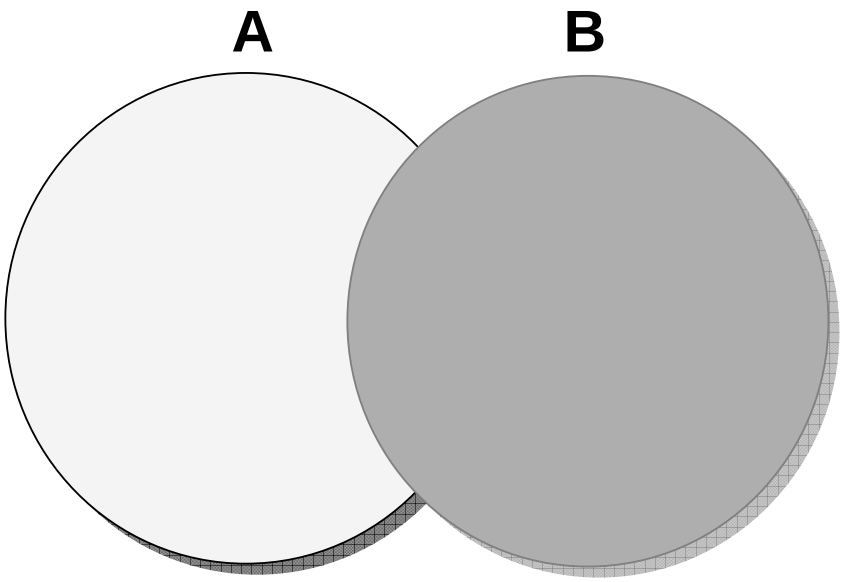
Exemple

```
SQL> SELECT ename, empno, job, deptno
  2  FROM   emp
  3  INTERSECT
  4  SELECT name, empid, title, deptid
  5  FROM   emp_history;
```

```
ENAME           EMPNO JOB           DEPTNO
---------- --------- --------- ---------
CLARK            7782 MANAGER          10
SCOTT            7788 ANALYST          20
```

L’employé ALLEN ne fait plus partie du résultat parce que la valeur de EMP.DEPTNO est différente de la valeur de EMP_HISTORY.DEPTID.

MINUS



6-11

L'opérateur MINUS

Cet opérateur ramène les lignes retournées par la première requête, qui ne le sont pas par la seconde (premier ordre SELECT moins le second).

- Le nombre de colonnes et les types de données des colonnes doivent être identiques dans les deux ordres SELECT. En revanche, les noms de colonnes peuvent être différents.
- Toutes les colonnes incluses dans la clause WHERE doivent également être incluses dans la clause SELECT pour que la requête de l'opérateur MINUS puisse être exécutée.
- Les requêtes incluant l'opérateur MINUS dans la clause WHERE doivent comprendre le même nombre de colonnes et des colonnes du même type que celles de la clause SELECT.

MINUS

Affichez le nom,le numéro et le poste de tous les employés ayant quitté la société.

```
SQL> SELECT name, empid, title
  2  FROM   emp_history
  3  MINUS
  4  SELECT ename, empno, job
  5  FROM   emp;
```

```
NAME           EMPID TITLE
---------- --------- ---------
BALFORD         6235 CLERK
BRIGGS          7225 PAY CLERK
JEWELL          7001 ANALYST
SPENCER         6087 OPERATOR
...
6 rows selected.
```

L'Opérateur MINUS

Dans l'exemple ci-dessus, les noms et les postes des employés de la table EMP sont soustraits de ceux de la table EMP_HISTORY. Cette table ne contient maintenant que les employés actuels et leur poste respectif.

Hanoune

Règles des Opérateurs Ensemblistes

- **Les expressions citées dans la clause SELECT doivent être égales en nombre et avoir des données du même type.**
- **Les doublons sont automatiquement éliminés, sauf avec l'opérateur UNION ALL.**
- **Les noms de colonnes apparaissant dans le résultat sont ceux de la première requête.**
- **Par défaut, le résultat est trié par ordre croissant, sauf avec l'opérateur UNION ALL.**
- **Utilisez des parenthèses pour modifier la séquence d'exécution.**

6-13

Règles des Opérateurs Ensemblistes

- Si deux requêtes sélectionnent des valeurs du type de données CHAR, les valeurs retournées sont des données du type CHAR.
- Si l'une ou l'autre requête sélectionne des valeurs du type de données VARCHAR2, les valeurs retournées sont des données du type VARCHAR2.
- La clause ORDER BY :
  - ne peut apparaître que tout à la fin de l'ordre.
  - accepte un nom de colonne, un alias ou la position de la colonne.
- Les noms de colonne ou alias, si ils sont utilisés dans la clause ORDER BY doivent provenir du premier ordre SELECT.
- Il est possible d'utiliser des opérateurs ensemblistes dans des sous-requêtes.
- Les ordres SELECT sont exécutés de gauche à droite ou de haut en bas.
- Vous pouvez modifier la priorité des opérateurs à l'aide de parenthèses.
- Les requêtes utilisant les opérateurs UNION, INTERSECT, et MINUS dans leur clause WHERE doivent avoir le même nombre et le même type de colonnes que celles sitées dans leur clause SELECT. Par exemple:

```
SQL> SELECT ename, deptno
  2  FROM   emp
  3  WHERE  (ename, deptno) IN (SELECT ename, deptno
  4                             FROM   emp) INTERSECT
  5                            (SELECT name, deptid
  6                             FROM   emp_history);
```

Correspondance des clauses SELECT

Affichez le numéro du département, le lieu et la date d'embauche de tous les employés.

```
SQL> SELECT deptno, TO_CHAR(NULL) location, hiredate
  2  FROM   emp
  3  UNION
  4  SELECT deptno, loc, TO_DATE(NULL)
  5  FROM   dept;
```

```
DEPTNO LOCATION      HIREDATE
--------- ------------- ---------
       10 NEW YORK
       10               09-JUN-81
       10               17-NOV-81
       10               23-JAN-82
       10
       20 DALLAS
       20               17-DEC-80
...
19 rows selected.
```

Correspondance des expressions dans les clauses SELECT

Comme les expressions de la clause SELECT des requêtes composées doivent être égales en nombre et contenir des données du même type, vous pouvez créer des colonnes factices et utiliser les fonctions de conversion de types de données pour respecter cette règle. Dans l'exemple ci-dessus, le nom "location" (lieu) a été attribué à l'en-tête de la colonne factice.

```
   DEPTNO LOCATION      HIREDATE
--------- ------------- ---------
       10 NEW YORK
       10               09-JUN-81
       10               17-NOV-81
       10               23-JAN-82
       10
       20 DALLAS
       20               17-DEC-80
...
       30               03-DEC-81
       40 BOSTON
19 rows selected.
```

Hanoune

Contrôler l'Ordre des Lignes

Créez une phrase anglaise à l'aide de deux opérateurs UNION.

```
SQL> COLUMN a_dummy NOPRINT
SQL> SELECT 'to sing' "My dream", 3 a_dummy
  2  FROM dual
  3  UNION
  4  SELECT 'I''d like to teach', 1
  5  FROM dual
  6  UNION
  7  SELECT 'the world', 2
  8  FROM dual
  9  ORDER BY 2;
```

```
My dream
------------------------
I'd like to teach
the world
to sing
```

6-15

Contrôler l'Ordre des Lignes

Par défaut, le résultat est trié par ordre croissant. Vous pouvez utiliser la clause ORDER BY pour modifier le tri.

Utilisation de la Clause ORDER BY pour Trier des Lignes

Dans une requête composée, la clause ORDER BY ne peut être utilisée qu'une seule fois et elle doit être placée à la fin de la requête. Cette clause accepte le nom de la colonne, un alias ou la position de la colonne.

Remarque : La clause ORDER BY, lorsuq’elle est utilisée dans une requête composée avec l’opérateur UNION (utilisé plus d’une fois), peut seulement utiliser les positions, pas les expressions.

Résumé

- **L'opérateur UNION ramène toutes les lignes distinctes.**
- **L'opérateur UNION ALL ramène toutes les lignes, y compris les doublons.**
- **L'opérateur INTERSECT ramène toutes les lignes partagées par deux requêtes.**
- **L'opérateur MINUS ramène toutes les lignes distinctes sélectionnées par la première requête, et non par la seconde.**
- **La clause ORDER BY doit être placée à la fin de l'ordre.**

6-16

Résumé

La clause ORDER BY ne peut être placée qu'à la fin d'une requête composée.

Les expressions correspondantes des listes SELECT doivent être égales en nombre et comprendre des données du même type.

Présentation des Exercices

Dans ces exercices, vous allez écrire des requêtes en incluant des opérateurs ensemblistes.

- **Utilisation d'autres méthodes de jointure**
- **Ecriture de requêtes composées sous forme d'ordres IF**

Remarque : Pour créer la table EMP_HISTORY, exécuter le script *emphis.sql*.

Exercice 6

1. Affichez le département qui ne comprend aucun employé.

```
   DEPTNO DNAME
--------- --------------
       40 OPERATIONS
```

2. Retrouvez le poste qui a été attribué dans le deuxième semestre des années 1981 et 1982.

```
JOB
---------
ANALYST
```

3. Ecrivez une requête composée pour afficher la liste des produits (utiliser la table PRICE) en indiquant le pourcentage de remise, l'identifiant des produits (PRODID), ainsi que le nouveau prix (avec remise) et l'ancien prix (STDPRICE):
   - 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%
   - ceux dont le prix est supérieur à $40 ne sont pas réduits.

```
DISCOUNT    PRODID  STDPRICE  ACTPRICE
-------- --------- --------- ---------
10% off     100870       2.4      2.16
10% off     100870       2.8      2.52
10% off     100871       4.8      4.32
10% off     100871       5.6      5.04
10% off     102130       3.4      3.06
10% off     200376       2.4      2.16
10% off     200380         4       3.6
15% off     100860        30      25.5
15% off     101860        24      20.4
15% off     101863      12.5    10.625
20% off     100860        32      25.6
20% off     100860        35        28
20% off     100861        39      31.2
no disc     100861        42        42
no disc     100861        45        45
no disc     100890        54        54
no disc     100890        58        58
```

Exercice 6

Si vous avez le temps, faites les exercices suivants :

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.

   Note : La commande SQL*Plus suivante `permet de ne pas afficher la colonne DUMMY(DUMMY est le nom d'une colonne ramenée par le SELECT) : COL dummy NOPRINT`

```
JOB         DEPTNO
--------- --------
CLERK           10
MANAGER         10
PRESIDENT       10
CLERK           30
MANAGER         30
SALESMAN        30
ANALYST         20
CLERK           20
MANAGER         20
```

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

```
DEPTNO
--------
    10
    30
    40
```

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

```
JOB
--------
ANALYST
PRESIDENT
```