Le langage SQL DML

A. EXERCICES ACCOMPAGNÉS DES DONNÉES RÉSULTANT DE L'EXÉCUTION DE LA SOLUTION

1 Énoncés de type 1

1.1 Afficher les caractéristiques des produits (c'est-à-dire, pour chaque produit, afficher ses caractéristiques).

NPRO	LIBELLE	PRIX	QSTOCK
CS262	CHEV. SAPIN 200x6x2	75	45
CS264	CHEV. SAPIN 200x6x4	120	2690
CS464	CHEV. SAPIN 400x6x4	220	450
PA45	POINTE ACIER 45 (1K)	105	580
PA60	POINTE ACIER 60 (1K)	95	134
PH222	PL. HETRE 200x20x2	230	782
PS222	PL. SAPIN 200x20x2	185	1220

1.2 Afficher la liste des localités dans lesquelles il existe au moins un client.

LOCALITE
Bruxelles
Geneve
Lille
Namur
Paris
Poitiers
Toulouse

1.3 Afficher le numéro, le nom et la localité des clients de catégorie C1 n'habitant pas à Toulouse.

NCLI	NOM	LOCALITE
B112	HANSENNE	Poitiers
C123	MERCIER	Namur
F010	TOUSSAINT	Poitiers
S127	VANDERKA	Namur
L422	FRANCK	Namur

1.4 Afficher les caractéristiques des produits en acier.

NPRO	LIBELLE	PRIX	QSTOCK
PA45	POINTE ACIER 45 (1K)	105	580
PA60	POINTE ACIER 60 (1K)	95	134

1.5 Donner le numéro, le nom et le compte des clients de Poitiers et de Bruxelles dont le compte est positif.

NCLI	NOM	COMPTE
B112	HANSENNE	1250.00
C400	FERARD	350.00

2 Énoncés de type 2

1.7 Quelles catégories de clients trouve-t-on à Toulouse ?

CAT
B1
B2

1.8 Afficher le numéro, le nom et la localité des clients dont le nom précède alphabétiquement la localité où ils résident.

NCLI	NOM	LOCALITE
B112	HANSENNE	Poitiers
C123	MERCIER	Namur
B512	GILLET	Toulouse
B062	GOFFIN	Namur
C400	FERARD	Poitiers
C003	AVRON	Toulouse
K729	NEUMAN	Toulouse
F011	PONCELET	Toulouse
L422	FRANCK	Namur
S712	GUILLAUME	Paris
D063	MERCIER	Toulouse

1.9 Afficher le total, le minimum, la moyenne et le maximum des comptes des clients (compte non tenu des commandes actuelles).

SOMME	MOYENNE	MINIMUM	MAXIMUM
-20410.00	-1275.63	-8700.00	1250.00

1.10 Afficher les numéros des clients qui commandent le produit de numéro `'CS464'`.

```
NCLI
=======
B512
C400
F011
K111
```

1.11 Afficher les localités des clients qui commandent le produit de numéro `'CS464'`.

```
LOCALITE
==================
Lille
Poitiers
Toulouse
```

1.12 Donner le numéro et le nom des clients de Namur qui n'ont pas passé de commandes.

```
NCLI    NOM
======= ================
B062    GOFFIN
C123    MERCIER
L422    FRANCK
```

1.13 Quels sont les produits en sapin qui font l'objet d'une commande ?

```
NPRO
=========
CS262
CS464
PS222
```

1.14 Rechercher les clients qui, s'ils ont un compte négatif, ont passé au moins une commande

Suggestion. Il s'agit d'une relation d'implication.

```
NCLI
=======
B112
B332
F010
```

```
K111
S127
B512
C400
K729
F011
L422
S712
F400
```

1.15 On considère comme ci-dessus les clients qui, s'ils ont un compte négatif, ont passé au moins une commande. Rechercher les autres clients
Suggestion. Il s'agit de la *négation* d'une relation d'implication.

```
NCLI
=======
C123
B062
C003
D063
```

1.16 Rechercher les commandes qui, si elles référencent des produits en sapin, référencent aussi des pointes en acier.
Suggestion. Il s'agit d'une relation d'implication.
C1 = "référence au moins un produit en SAPIN"
C2 = "référence au moins un produit de POINTES EN ACIER"
$C1 \Rightarrow C2 = (\neg C1) \vee C2$

1.17 Ecrire les requêtes SQL qui recherchent les clients (on simplifiera si nécessaire) :

- habitant à Lille ou à Namur.

```
NCLI     LOCALITE
======= ==================
C123     Namur
K111     Lille
S127     Namur
B062     Namur
L422     Namur
```

- qui à la fois habitent à Lille et n'habitent pas à Namur.
- qui habitent à Lille ou n'habitent pas à Namur.
- qui n'habitent ni à Lille ni à Namur.

```
NCLI     LOCALITE
```

```
======= ==================
B112    Poitiers
B332    Geneve
F010    Poitiers
B512    Toulouse
C400    Poitiers
C003    Toulouse
K729    Toulouse
F011    Toulouse
S712    Paris
D063    Toulouse
F400    Bruxelles
```

- qui n'habitent pas à Lille ou qui n'habitent pas à Namur.

- de catégorie C1 habitant à Namur.

```
NCLI    LOCALITE            CAT
======= ================== ======
C123    Namur               C1
S127    Namur               C1
L422    Namur               C1
```

- de catégorie C1 ou habitant à Namur.

```
NCLI    LOCALITE            CAT
======= ================== ======
B112    Poitiers            C1
C123    Namur               C1
F010    Poitiers            C1
S127    Namur               C1
B062    Namur               B2
L422    Namur               C1
```

- de catégorie C1 n'habitant pas à Namur.

```
NCLI    LOCALITE            CAT
======= ================== ======
B112    Poitiers            C1
F010    Poitiers            C1
```

- qui n'ont pas été sélectionnés dans la question précédente.

```
NCLI    LOCALITE            CAT
======= ================== ======
C123    Namur               C1
B332    Geneve              B2
K111    Lille               B1
S127    Namur               C1
B512    Toulouse            B1
B062    Namur               B2
C400    Poitiers            B2
C003    Toulouse            B1
```

```
K729     Toulouse            <null>
F011     Toulouse            B2
L422     Namur               C1
S712     Paris               B1
D063     Toulouse            <null>
F400     Bruxelles           C2
```

- qui soit sont de catégorie B1 ou C1, soit habitent à Lille ou à Namur (ou les deux conditions).

```
NCLI     LOCALITE            CAT
======= ================== ======
B112     Poitiers            C1
C123     Namur               C1
F010     Poitiers            C1
K111     Lille               B1
S127     Namur               C1
B512     Toulouse            B1
B062     Namur               B2
C003     Toulouse            B1
L422     Namur               C1
S712     Paris               B1
```

- qui soit sont de catégorie B1 ou C1, soit habitent à Lille ou à Namur (mais pas les deux conditions).

```
NCLI     LOCALITE            CAT
======= ================== ======
B112     Poitiers            C1
F010     Poitiers            C1
B512     Toulouse            B1
B062     Namur               B2
C003     Toulouse            B1
S712     Paris               B1
```

- qui sont de catégorie B1 ou C1, et qui habitent à Lille ou à Namur.

```
NCLI     LOCALITE            CAT
======= ================== ======
C123     Namur               C1
K111     Lille               B1
S127     Namur               C1
L422     Namur               C1
```

- qui n'ont pas été sélectionnés dans la question précédente.

```
NCLI     LOCALITE            CAT
======= ================== ======
B112     Poitiers            C1
B332     Geneve              B2
F010     Poitiers            C1
B512     Toulouse            B1
B062     Namur               B2
```

```
C400     Poitiers            B2
C003     Toulouse            B1
K729     Toulouse            <null>
F011     Toulouse            B2
S712     Paris               B1
D063     Toulouse            <null>
F400     Bruxelles           C2
```

3 Énoncés de type 3

1.18 Calculer le montant de chaque détail de commande.

```
   MONTANT
==========
      4500
      5500
     26400
     57200
     39600
      2100
       315
      2310
      1900
      2850
      1425
      6650
     21160
    111000
```

1.19 Afficher la valeur totale des stocks (compte non tenu des commandes actuelles).

```
     TOTAL
==========
    904365
```

1.20 Calculer le montant commandé des produits en sapin.

```
   MONTANT
==========
    244200
```

1.21 Afficher le total et la moyenne des comptes des clients, ainsi que le nombre de clients, selon chacune des classifications suivantes :

- par catégorie,

```
CAT            SUM         AVG        COUNT
===== ========== ========== ==========
```

```
B1          -9680.00    -2420.00           4
B2          -2850.00     -712.50           4
C1          -5630.00    -1126.00           5
C2              0.00        0.00           1
<null>      -2250.00    -1125.00           2
```

• par localité,

```
LOCALITE                    SUM          AVG        COUNT
==================== ========== ========== ==========
Bruxelles                  0.00       0.00          1
Geneve                     0.00       0.00          1
Lille                    720.00     720.00          1
Namur                  -10080.00   -2520.00          4
Paris                      0.00       0.00          1
Poitiers                1600.00     533.33          3
Toulouse               -12650.00   -2530.00
```

• par catégorie dans chaque localité.

```
LOCALITE             CAT             SUM          AVG        COUNT
==================== ====== ========== ========== ==========
Bruxelles            C2             0.00       0.00          1
Geneve               B2             0.00       0.00          1
Lille                B1           720.00     720.00          1
Namur                B2         -3200.00   -3200.00          1
Namur                C1         -6880.00   -2293.33          3
Paris                B1             0.00       0.00          1
Poitiers             B2           350.00     350.00          1
Poitiers             C1          1250.00     625.00          2
Toulouse             B1        -10400.00   -5200.00          2
Toulouse             B2             0.00       0.00          1
Toulouse             <null>     -2250.00   -1125.00          2
```

1.22 Afficher le numéro et le libellé des produits en sapin :

• qui ne sont pas commandés,

```
NPRO       LIBELLE
========== ============================
CS264      CHEV. SAPIN 200x6x4
```

• qui sont commandés à Toulouse,

```
NPRO       LIBELLE
========== ============================
CS464      CHEV. SAPIN 400x6x4
PS222      PL. SAPIN 200x20x2
```

• qui ne sont pas commandés à Toulouse

```
NPRO       LIBELLE
```

```
========= ============================
CS262     CHEV. SAPIN 200x6x2
CS264     CHEV. SAPIN 200x6x4
```

• qui ne sont commandés qu'à Toulouse,

```
NPRO      LIBELLE
========= ============================
PS222     PL. SAPIN 200x20x2
```

• qui ne sont pas commandés qu'à Toulouse,

• qui sont commandés à Toulouse, mais aussi ailleurs.

```
NPRO      LIBELLE
========= ============================
CS464     CHEV. SAPIN 400x6x4
```

1.23 Combien y a-t-il de commandes spécifiant un (ou plusieurs) produit(s) en acier ?

```
     COUNT
==========
         6
```

1.24 Dans combien de localités trouve-t-on des clients de catégorie C1 ?

```
     COUNT
==========
         2
```

1.25 Créer une table et y ranger les données suivantes relatives aux détails de commande : numéro et date de la commande, quantité commandée, numéro et prix du produit, montant du détail.

1.26 Annuler les comptes négatifs des clients de catégorie C1.

1.27 Compléter le fragment suivant de manière à former une requête valide

```
select CAT, NPRO, sum(QCOM*PRIX)
from ...
```

```
CAT   NPRO              SUM
===== ========= ==========
B1    CS464          45100
B1    PA45            2310
```

```
B1      PA60             6650
B1      PH222           21160
B2      CS262            4500
B2      CS464           83600
B2      PA45             2415
B2      PA60             3325
B2      PS222          111000
C1      PA60             2850
```

1.28 Rechercher les clients qui ont commandé le produit PA60 ou le produit PA45, mais pas les deux.
Suggestion. Appliquer un *ou exclusif*.

```
NCLI
========
S127
F011
```

1.29 En vous basant sur le schéma 4.5, écrivez les requêtes SQL qui donnent :

- les matières premières (produits qui n'ont pas de composants);

```
NPRO
==========

p6
p7
p8
p11
p12
```

- les produits finis (qui n'entrent dans la composition d'aucun autre);

```
NPRO
==========
p1
p5
```

- les produits semi-finis (tous les autres);

```
NPRO
==========
p2
p3
p4
p9
p10
```

- le prix et poids unitaires d'un produit fini ou semi-fini dont tous les composants ont un poids et un prix unitaires.

```
NPRO              SUM         SUM
========= ========== ==========
p10               7.1       10.50
p2               74.0        4.30
p3               25.0        5.75
p9                1.0        1.80
```

4 Énoncés de type 4

1.30 Afficher le numéro et le nom des clients qui n'ont pas commandé de produits en sapin.

```
NCLI    NOM
======= ================
B112    HANSENNE
C123    MERCIER
B332    MONTI
F010    TOUSSAINT
S127    VANDERKA
B062    GOFFIN
C003    AVRON
K729    NEUMAN
L422    FRANCK
S712    GUILLAUME
D063    MERCIER
F400    JACOB
```

1.31 Rechercher les clients qui, s'ils ont commandé le produit PA60, en ont commandé plus de 500 unités au total.

Suggestion. Il s'agit d'une relation d'implication.

```
NCLI
=======
B112
C123
B332
F010
K111
B062
C003
K729
F011
L422
S712
D063
F400
```

Remarque. Curieusement, cette requête n'est pas équivalente à "*Re-*

chercher les clients qui ont commandé plus de 500 unités de PA60". En effet, les clients qui n'ont pas commandé de produit PA60 sont aussi sélectionnés.

1.32 A la question : "*rechercher les localités dans lesquelles on n'a pas commandé de produit PA60*", quatre utilisateurs proposent les requêtes suivantes. Indiquer la (ou les) requêtes correctes, et interprétez les autres.

```
select distinct LOCALITE
from   CLIENT
where  NCLI in
            (select NCLI
             from   COMMANDE
             where  NCOM in
                        (select NCOM
                         from   DETAIL
                         where  NPRO <> 'PA60'))
```

```
LOCALITE
====================
Lille
Poitiers
Toulouse
```

```
select distinct LOCALITE
from   CLIENT
where  NCLI in
            (select NCLI
             from   COMMANDE
             where  NCOM not in
                        (select NCOM
                         from   DETAIL
                         where  NPRO = 'PA60'))
```

```
LOCALITE
====================
Lille
Poitiers
```

```
select distinct LOCALITE
from   CLIENT
where  NCLI not in
            (select NCLI
             from   COMMANDE
             where  NCOM in
                        (select NCOM
                         from   DETAIL
                         where  NPRO = 'PA60'))
```

```
LOCALITE
====================
Bruxelles
```

```
Geneve
Lille
Namur
Paris
Poitiers
Toulouse
```

```
select distinct LOCALITE
from   CLIENT
where  LOCALITE not in
            (select LOCALITE
             from   CLIENT
             where  NCLI in
                     (select NCLI
                      from   COMMANDE
                      where  NCOM in
                             (select NCOM
                              from   DETAIL
                              where  NPRO = 'PA60')))
```

```
LOCALITE
==================
Bruxelles
Geneve
Lille
Paris
```

1.33 Que signifie la requête suivante ?

```
select *
from   COMMANDE
where  NCOM not in (select NCOM
                    from   DETAIL
                    where  NPRO <> 'PA60')
```

```
NCOM     NCLI         DATECOM
======= ======= ==========
30182    S127     23-DEC-2000
```

1.34 Dans quelles localités a-t-on commandé en décembre 2000 ?

```
LOCALITE
==================
Lille
Namur
Poitiers
```

1.35 Calculer le montant de chaque commande.

```
NCOM            SUM
```

```
======= ==========
30178          5500
30179          6400
30182          2850
30184         28500
30185        169625
30186           315
30188         69720
```

1.36 Calculer le montant dû par chaque client. Dans ce calcul, on ne prend en compte que le montant des commandes. Attention aux clients qui n'ont pas passé de commandes.

```
NCLI            SUM
======= ==========
B062              0
B112              0
B332              0
B512          69720
C003              0
C123              0
C400          35215
D063              0
F010              0
F011         169625
F400              0
K111           5500
K729              0
L422              0
S127           2850
S712              0
```

1.37 Calculer le montant dû par les clients de chaque localité. Dans ce calcul, on ne prend en compte que le montant des commandes. Attention aux localités dans lesquelles aucun client n'a passé de commandes.

```
LOCALITE                      SUM
================== ==========
Bruxelles                       0
Geneve                          0
Lille                        5500
Namur                        2850
Paris                           0
Poitiers                    35215
Toulouse                   239345
```

1.38 Calculez, par jour, le total des montants des commandes.

```
    DATECOM         SUM
========== ==========
21-DEC-2000        5500
```

```
22-DEC-2000        6400
23-DEC-2000       31350
 2-JAN-2001      169940
 3-JAN-2001       69720
```

1.39 On suppose qu'on n'a pas trouvé utile d'imposer un identifiant primaire sur la table PRODUIT. Il se peut donc que plusieurs lignes aient même valeur de la colonne NPRO, ce qui viole le principe d'unicité des valeurs de cette colonne.

- Ecrire une requête qui recherche les valeurs de NPRO présentes en plus d'un exemplaire.

- Écrire une requête qui indique combien de valeurs de NPRO sont présentes en plus d'un exemplaire.

Suggestion. Il s'agit du calcul d'une statistique sur une autre statistique. Le plus simple est d'exprimer cette dernière sous la forme d'une vue.

```
      COUNT
==========
         0
```

- Écrire une requête qui indique combien de lignes comportent une erreur d'unicité de NPRO.

- Écrire une requête qui, pour chaque valeur de NPRO présente dans la table, indique dans combien de lignes cette valeur est présente.

- Écrire une requête qui, pour chaque valeur de NPRO qui n'est pas unique, indique dans combien de lignes cette valeur est présente.

- Écrire une suite de requêtes qui crée une table contenant les numéros de NPRO qui ne sont pas uniques.

1.40 On suppose qu'on n'a pas trouvé utile de déclarer NCOM clé étrangère dans la table DETAIL. Il est donc possible que certaines lignes de DETAIL violent la contrainte d'intégrité référentielle portant sur cette colonne. Ecrire une requête qui recherche les anomalies éventuelles.

1.41 Normalement, à toute commande doit être associé au moins un détail. Ecrivez une requête qui vérifie qu'il en bien ainsi dans la base de données.

1.42 Afficher pour chaque localité, les libellés des produits qui y sont commandés.

```
LOCALITE             LIBELLE
==================== ============================

Lille                CHEV. SAPIN 400x6x4
Namur                POINTE ACIER 60 (1K)
Poitiers             CHEV. SAPIN 200x6x2
Poitiers             CHEV. SAPIN 400x6x4
Poitiers             POINTE ACIER 45 (1K)
Poitiers             POINTE ACIER 60 (1K)
Toulouse             CHEV. SAPIN 400x6x4
Toulouse             PL. HETRE 200x20x2
Toulouse             PL. SAPIN 200x20x2
Toulouse             POINTE ACIER 45 (1K)
Toulouse             POINTE ACIER 60 (1K)
```

1.43 Affichez par localité, et pour chaque catégorie dans celle-ci, le total des montants des commandes. *Attention aux localités dans lesquelles on n'a pas commandé.*

```
LOCALITE             CAT             SUM
==================== ====== ==========
Bruxelles            C2               0
Geneve               B2               0
Lille                B1            5500
Namur                B2               0
Namur                C1            2850
Paris                B1               0
Poitiers             B2           35215
Poitiers             C1               0
Toulouse             B1           69720
Toulouse             B2          169625
Toulouse             <null>           0
```

1.44 Quels sont les produits (numéro et libellé) qui n'ont pas été commandés en 2008 ?

```
NPRO       LIBELLE
========== ============================
CS264      CHEV. SAPIN 200x6x4
PH222      PL. HETRE 200x20x2
PS222      PL. SAPIN 200x20x2
```

1.45 Indiquer, pour chaque localité, les catégories de clients qui n'y sont pas représentées

Suggestion. Construire l'ensemble de tous les couples LOCALITE x CAT possibles et en retirer ceux qui existent dans la base. Attention aux valeurs `null`, qui ne doivent pas être prises en compte.

```
LOCALITE             CAT
```

```
================== ======
Bruxelles          B1
Bruxelles          B2
Bruxelles          C1
Geneve             B1
Geneve             C1
Geneve             C2
Lille              B2
Lille              C1
Lille              C2
Namur              B1
Namur              C2
Paris              B2
Paris              C1
Paris              C2
Poitiers           B1
Poitiers           C2
Toulouse           C1
Toulouse           C2
```

1.46 Produire (à l'écran) une table de couples <X,Y> de clients tels que X et Y habitent dans la même localité. On évitera de renseigner <X,X>, mais aussi <Y,X> si <X,Y> est déjà repris.

Suggestion. Auto-jointure de CLIENT. On évitera les couples inverse en imposant un ordre sur les valeurs de NPRO (p.ex. X < Y).

```
LOCALITE           NCLI     NCLI
================== ======== ========
Namur              B062     C123
Namur              B062     L422
Namur              B062     S127
Namur              C123     L422
Namur              C123     S127
Namur              L422     S127
Poitiers           B112     C400
Poitiers           B112     F010
Poitiers           C400     F010
Toulouse           B512     C003
Toulouse           B512     D063
Toulouse           B512     F011
Toulouse           B512     K729
Toulouse           C003     D063
Toulouse           C003     F011
Toulouse           C003     K729
Toulouse           D063     F011
Toulouse           D063     K729
Toulouse           F011     K729
```

1.47 En vous basant sur le schéma 5.5, écrivez une requête de mise à jour qui complète les prix et poids unitaires des produits finis ou semi-finis. Pour simplifier la procédure, on admet que cette requête soit exécutée autant de fois que nécessaire pour que tous les produits soient complétés.

NPRO	LIBELLE	PRIX_U	POIDS_U
p1	A-200	294.0	<null>
p2	A-056	74.0	<null>
p3	B-661	25.0	<null>
p4	B-122	60.5	<null>
p5	B-326	145.5	<null>
p6	D-822	3.5	0.70
p7	D-507	8.0	0.25
p8	G-993	5.0	1.15
p9	F-016	1.0	<null>
p10	J-500	7.1	<null>
p11	J-544	0.5	0.90
p12	L-009	1.7	2.30

La mise à jour de POIDS_U se commande de manière similaire.

1.48 En considérant le schéma de la figure 5.7, calculer pour chaque ville, le prix moyen de chaque produit.

1.49 Afficher pour chaque client, le nombre de commandes, le nombre de produits commandés et le nombre de détails. On se limite aux clients qui ont passé au moins une commande.

Suggestion : il s'agit d'une requête basée sur des groupements multi-niveaux.

NCLI	COUNT	COUNT	COUNT
B512	1	4	4
C400	3	4	5
F011	1	3	3
K111	1	1	1
S127	1	1	1

1.50 Afficher, pour chaque localité et pour chaque catégorie, (1) le nombre de commandes passées par les clients de cette localité et de cette catégorie, (2) le montant total de ces commandes.

LOCALITE	CAT	COUNT	SUM
Lille	B1	1	5500
Namur	C1	1	2850
Poitiers	B2	3	35215
Toulouse	B1	1	69720
Toulouse	B2	1	169625

5 Énoncés de type 5

1.51 Calculez le nombre moyen de produits par commande. De même : le nombre moyen de commandes par client, par localité ou par catégorie.

Remarque. Il n'est pas possible de demander directement la moyenne *d'un nombre (comptage)*. *Suggestion* : construction et interrogation d'une vue ou calcul *du nombre* dans le `from`.

```
       AVG
==========
         2
```

1.52 Quel est, pour chaque localité, le nombre moyen de commandes par client.

```
LOCALITE                    AVG
==================== ==========
Lille                         1
Namur                         1
Poitiers                      3
Toulouse                      1
```

1.53 Ecrire une requête SQL qui donne, pour chaque catégorie de produit, le nombre de produits qui ont été commandés le *23-12-2008*.

```
CAT          COUNT
===== ==========
B2               2
C1               1
```

1.54 Donner pour chaque localité dans laquelle se trouve au moins un client de catégorie '*C1*' la liste des produits en sapin qu'on y a commandés.

```
LOCALITE             NPRO
==================== =========
Poitiers             CS262
Poitiers             CS464
```

1.55 Donner pour chaque produit la liste des localités dans lesquelles ce produit est commandé en plus de 500 unités (= au total pour la localité).

```
NPRO       LOCALITE                    SUM
========= ==================== ==========
PS222      Toulouse                    600
```

1.56 Afficher, pour chaque localité, les produits qu'on y commande et qui sont aussi commandés dans au moins une autre localité.

Suggestion. Un produit est intéressant si le nombre de localités dans lesquelles il est commandé est supérieur *ou* égal à 2.

```
LOCALITE            NPRO
=================== =========
Lille               CS464
Namur               PA60
Poitiers            CS464
Poitiers            PA45
Poitiers            PA60
Toulouse            CS464
Toulouse            PA45
Toulouse            PA60
```

1.57 Rechercher les clients qui ont commandé tous les produits.

Suggestion. Application du quantificateur *pour tout*. On recherche les clients tels qu'il n'existe pas de produits qui n'apparaissent pas dans les détails de leurs commandes.

1.58 Dans quelles localités a-t-on commandé tous les produits en acier (tous clients confondus) ?

Exercice préparatoire : *quels sont les clients qui ont commandé tous les produits en acier.*

```
NCLI
========
B512
C400
```

Question d'origine :

```
les localités L telles
     qu'il n'existe pas
           de produits en acier qui ne soit commandé
                          par un client de la localité L
```

```
LOCALITE
====================
Poitiers
Toulouse
```

1.59 Rechercher les produits qui ont été commandés par tous les clients.

1.60 Rechercher les localités dont aucun client n'a passé de commande.

```
LOCALITE
==================
Bruxelles
Geneve
Paris
```

1.61 Rechercher les localités dont tous les clients ont passé au moins une commande.

```
LOCALITE
==================
Lille
```

1.62 Rechercher les produits qui sont commandés dans toutes les localités.

1.63 Dans quelles localités peut-on trouver au moins un client qui a commandé tous les produits en sapin, et ce, pour un montant total, pour ces produits, dépassant 10.000 ?

1.64 Calculer, pour chaque localité, le nombre de catégories distinctes.

```
LOCALITE                  COUNT
================== ==========
Bruxelles                    1
Geneve                       1
Lille                        1
Namur                        2
Paris                        1
Poitiers                     2
Toulouse                     2
```

1.65 La section 4.10.1 suggère une comparaison entre un instantané et une vue de même définition. En effet, ces deux techniques pourraient être considérées comme équivalentes pour la formulation de requêtes. Etablissez une liste de critères de comparaison et appliquez-les aux deux techniques.

1.66 Mettre à jour les comptes des clients en en déduisant le montant des commandes en cours.
Suggestion. *cfr* mise à jour des quantités en stock des produits; attention aux clients qui n'ont pas commandé.

```
NPRO            QSTOCK
========= ==========
```

```
CS262              -15
CS264             2690
CS464             -135
PA45               535
PA60                -1
PH222              690
PS222              620
```

1.67 Mettre à jour les quantités en stock des produits en en déduisant les quantités commandées par les clients de catégorie B1 et C1.

```
NPRO           QSTOCK
========== ==========
CS262              45
CS264            2690
CS464             245
PA45              558
PA60               34
PH222             690
PS222            1220
```

1.68 On suppose qu'on dispose de deux tables PRODUIT1 et PRODUIT2 décrivant des produits de deux filiales distinctes, et de même structure que PRODUIT. Il est possible qu'un produit soit présent simultanément dans les deux filiales. Le même numéro de produit est repris alors dans les deux tables. On désire intégrer les deux dépôts. En conséquence, on fusionne les deux tables initiales en une troisième selon les règles suivantes :

- si un produit n'est présent que dans une seule filiale, sa ligne est reprise telle quelle,
- si un produit est présent dans les deux filiales, on ne le décrit qu'une seule fois. Son libellé est celui de PRODUIT1, son prix est le minimum des deux valeurs et sa quantité en stock est la somme des deux valeurs.

```
NPRO       LIBELLE                          PRIX      QSTOCK
========== ============================ ========== ==========
A001       Farine                         0.80          65
A002       Sucre                          1.20          45
A003       Huile olive                    5.20          31
B001       Sel                            0.45         155
```

1.69 Soit une jointure de 2 ou plusieurs tables, dont la clause `select` spécifie certaines colonnes. On désire que le résultat ne comporte pas de lignes en double. Indiquer dans quelles conditions le modifieur `distinct` est inutile.

Suggestion. `distinct` est inutile lorsque la clause `select` comprend tous les composants de l'identifiant de la table mentionnée dans la clause `from` ou (cas d'une jointure par exemple) construite dans les clauses `from-where-group-by`.

1.70 On considère une base de données constituée des deux tables suivantes : R(A,B) et S(B,C). Evaluer l'ordre de grandeur du nombre de requêtes différentes qu'il est possible de formuler sur cette base de données.
Réponse : une infinité

6 Énoncés de type 6 (*non résolus*)

1.71 La figure 7.21 (Voyages en train) représente un schéma qui est accompagné de suggestions de questions auxquelles une base de données traduisant ce schéma permettrait de répondre. On suppose que ce schéma est traduit en structure de tables (*cf.* exercice 9.3). Exprimez chacune de ces questions sous la forme de requêtes SQL.

1.72 On désire réaliser le couplage deux à deux de certaines localités où résident des clients. A cette fin, on désire construire des listes de couples candidats. On se limite cependant aux couples de localités dans lesquelles on achète au moins un même produit.

1.73 Même question que 4.69 ci-dessus, mais on se limite aux couples de localités dans lesquelles on achète exactement les mêmes produits.

1.74 Pour chaque produit, indiquer la ville dans laquelle la quantité totale commandée est minimale, et celle dans laquelle cette quantité est maximale.

7 Énoncés de type 7 (*non résolus*)

1.75 Rédigez un script (suite de requêtes) SQL qui réalise en fin de mois les opérations de gestion régies par les règles suivantes :

- pour chaque client, on calcule le total des montants des commandes du mois écoulé (= les commandes d'un mois et d'une année déterminés);

- si ce montant est supérieur à celui du mois précédent, on applique une réduction de 5%;

- ensuite, si le client est le seul de sa localité, on applique une réduction supplémentaire de 5%;

- on range dans une table les informations nécessaires à l'édition des factures du mois;

- on met à jour le compte des clients;

- on met à jour le niveau de stock (QSTOCK) des produits commandés;

- pour chaque produit dont le niveau de stock a été mis à jour, et dont le niveau est inférieur ou égal à 0, on prépare dans une table adéquate les informations de réapprovisionnement.

B. EXERCICES ACCOMPAGNÉS D'UNE SOLUTION TYPE

1 Énoncés de type 1

1.1 Afficher les caractéristiques des produits (c'est-à-dire, pour chaque produit, afficher ses caractéristiques).

1.2 Afficher la liste des localités dans lesquelles il existe au moins un client.

```
select distinct LOCALITE
from   CLIENT
```

1.3 Afficher le numéro, le nom et la localité des clients de catégorie C1 n'habitant pas à Toulouse.

```
select NCLI, NOM, LOCALITE
from   CLIENT
where  CAT = 'C1'
and    LOCALITE <> 'Toulouse'
```

1.4 Afficher les caractéristiques des produits en acier.

```
select *
from   PRODUIT
where  LIBELLE like '%ACIER%'
```

1.5 Donner le numéro, le nom et le compte des clients de Poitiers et de Bruxelles dont le compte est positif.

```
select NCLI, NOM, COMPTE
from   CLIENT
where  LOCALITE in ('Poitiers','Bruxelles')
and    COMPTE > 0
```

1.6 Dessiner l'organigramme des personnes selon le contenu de la table 4.3. De même, dessiner le graphe de composition de la nomenclature de la figure 4.6.

2 Énoncés de type 2

1.7 Quelles catégories de clients trouve-t-on à Toulouse ?

```
select distinct CAT
from   CLIENT
where  LOCALITE = 'Toulouse'
```

```
and    CAT is not null
```

1.8 Afficher le numéro, le nom et la localité des clients dont le nom précède alphabétiquement la localité où ils résident.

```
select NCLI, NOM, LOCALITE
from   CLIENT
where  NOM < LOCALITE
```

1.9 Afficher le total, le minimum, la moyenne et le maximum des comptes des clients (compte non tenu des commandes actuelles).

```
select sum(COMPTE) as somme,
       avg(COMPTE) as moyenne,
       min(COMPTE) as minimum,
       max(COMPTE) as maximum
from   CLIENT
```

1.10 Afficher les numéros des clients qui commandent le produit de numéro `'CS464'`.

```
select distinct NCLI
from   COMMANDE
where  NCOM in (select NCOM
                from   DETAIL
                where  NPRO = 'CS464')
```

1.11 Afficher les localités des clients qui commandent le produit de numéro `'CS464'`.

```
select distinct LOCALITE
from   CLIENT
where  NCLI in (select NCLI
                from   COMMANDE
                where  NCOM in (select NCOM
                                from   DETAIL
                                where  NPRO = 'CS464'))
```

1.12 Donner le numéro et le nom des clients de Namur qui n'ont pas passé de commandes.

1.13 Quels sont les produits en sapin qui font l'objet d'une commande ?

1.14 Rechercher les clients qui, s'ils ont un compte négatif, ont passé au moins une commande

Suggestion. Il s'agit d'une relation d'implication.

```
select NCLI
from   CLIENT
where  COMPTE >= 0
or     NCLI in (select NCLI from COMMANDE)
```

1.15 On considère comme ci-dessus les clients qui, s'ils ont un compte négatif, ont passé au moins une commande. Rechercher les autres clients

Suggestion. Il s'agit de la *négation* d'une relation d'implication.

```
select NCLI
from CLIENT
where COMPTE < 0
and NCLI not in (select NCLI from COMMANDE)
```

1.16 Rechercher les commandes qui, si elles référencent des produits en sapin, référencent aussi des pointes en acier.

Suggestion. Il s'agit d'une relation d'implication.

C1 = "référence au moins un produit en SAPIN"

C2 = "référence au moins un produit de POINTES EN ACIER"

$C1 \Rightarrow C2 = (\neg C1) \vee C2$

```
select *
from   COMMANDE
where  not (C1)
or     C2
```

1.17 Ecrire les requêtes SQL qui recherchent les clients (on simplifiera si nécessaire) :

- habitant à Lille ou à Namur.
- qui à la fois habitent à Lille et n'habitent pas à Namur.
- = "qui habitent à Lille".
- qui habitent à Lille ou n'habitent pas à Namur.
- = "qui n'habitent pas à Namur".
- qui n'habitent ni à Lille ni à Namur.

- qui n'habitent pas à Lille ou qui n'habitent pas à Namur.
- = "tous les clients".

- de catégorie C1 habitant à Namur.

- de catégorie C1 ou habitant à Namur.

- de catégorie C1 n'habitant pas à Namur.

- qui n'ont pas été sélectionnés dans la question précédente.

```
select NCLI, LOCALITE, CAT
from   CLIENT
where  CAT <> 'C1'
or     CAT is null
or     LOCALITE = 'Namur'
```

- qui soit sont de catégorie B1 ou C1, soit habitent à Lille ou à Namur (ou les deux conditions).

```
select NCLI, LOCALITE, CAT
from   CLIENT
where  CAT in ('B1','C1')
or     LOCALITE in ('Lille','Namur')
```

- qui soit sont de catégorie B1 ou C1, soit habitent à Lille ou à Namur (mais pas les deux conditions).

```
select NCLI, LOCALITE, CAT
from   CLIENT
where  (CAT in ('B1','C1') and LOCALITE not in ('Lille','Namur'))
or     (CAT not in ('B1','C1') and LOCALITE in ('Lille','Namur'))
```

- qui sont de catégorie B1 ou C1, et qui habitent à Lille ou à Namur.

- qui n'ont pas été sélectionnés dans la question précédente.

```
select NCLI, LOCALITE, CAT
from   CLIENT
where  CAT not in ('B1','C1')
or     LOCALITE not in ('Lille','Namur')
```

3 Énoncés de type 3

1.18 Calculer le montant de chaque détail de commande.

```
select QCOM*PRIX as MONTANT
from   DETAIL D,PRODUIT P
```

```
where  D.NPRO = P.NPRO
```

1.19 Afficher la valeur totale des stocks (compte non tenu des commandes actuelles).

```
select sum(QSTOCK*PRIX) as TOTAL
from PRODUIT
```

1.20 Calculer le montant commandé des produits en sapin.

```
select sum(QCOM*PRIX) as MONTANT
from   DETAIL D,PRODUIT P
where  D.NPRO = P.NPRO
and    P.LIBELLE like '%SAPIN%'
```

1.21 Afficher le total et la moyenne des comptes des clients, ainsi que le nombre de clients, selon chacune des classifications suivantes :

- par catégorie,
- par localité,
- par catégorie dans chaque localité.

1.22 Afficher le numéro et le libellé des produits en sapin :

- qui ne sont pas commandés,

```
select NPRO, LIBELLE
from   PRODUIT
where  LIBELLE like '%SAPIN%'
and    NPRO not in (select NPRO from DETAIL)
```

- qui sont commandés à Toulouse,

```
select NPRO, LIBELLE
from   PRODUIT
where  LIBELLE like '%SAPIN%'
and    NPRO in (select NPRO
                from DETAIL
                where NCOM in
                     (select NCOM
                      from COMMANDE where NCLI in
                           (select NCLI
                            from CLIENT
                            where LOCALITE = 'Toulouse')))
```

ou

```
select distinct NPRO, LIBELLE
from   CLIENT C, COMMANDE M, DETAIL D, PRODUIT P
```

```
where  LOCALITE = 'Toulouse'
and    C.NCLI = M.NCLI
and    M.NCOM = D.NCOM
and    D.NPRO = P.NPRO
and    LIBELLE like '%SAPIN%'
```

• qui ne sont pas commandés à Toulouse

• qui ne sont commandés qu'à Toulouse,
• = "qui sont commandés à Toulouse et qui ne sont pas commandés dans une localité qui n'est pas Toulouse",

```
select NPRO, LIBELLE
from   PRODUIT
where  LIBELLE like '%SAPIN%'
and    NPRO in (select NPRO
                from DETAIL
                where NCOM in
                     (select NCOM
                      from COMMANDE where NCLI in
                           (select NCLI
                            from CLIENT
                            where LOCALITE = 'Toulouse')))
and    NPRO not in (select NPRO
                from DETAIL
                where NCOM in
                     (select NCOM
                      from COMMANDE where NCLI in
                           (select NCLI
                            from CLIENT
                            where LOCALITE <> 'Toulouse')))
```

ou, plus simplement : *qui sont commandés (quelle que soit la localité), mais qui ne sont pas commandés dans une localité qui n'est pas Toulouse*

```
select NPRO, LIBELLE

from   PRODUIT
where  LIBELLE like '%SAPIN%'
and    NPRO in (select NPRO
                from DETAIL)
and    NPRO not in (select NPRO
                from DETAIL
                where NCOM in
                     (select NCOM
                      from COMMANDE where NCLI in
                           (select NCLI
                            from CLIENT
                            where LOCALITE <> 'Toulouse')))
```

• qui ne sont pas commandés qu'à Toulouse,
• = "ceux qui ne satisfont pas la requête précédente",

• qui sont commandés à Toulouse, mais aussi ailleurs.

1.23 Combien y a-t-il de commandes spécifiant un (ou plusieurs) produit(s) en acier ?

```
select count(distinct M.NCOM)
from   COMMANDE M, DETAIL D, PRODUIT P
where  M.NCOM = D.NCOM
and    D.NPRO = P.NPRO
and    LIBELLE like '%ACIER%'
     ou
select count(*)
from   COMMANDE M
where  NCOM in (select NCOM
                from DETAIL D, PRODUIT P
                where  D.NPRO = P.NPRO
                and    LIBELLE like '%ACIER%')
```

1.24 Dans combien de localités trouve-t-on des clients de catégorie C1 ?

```
select count(distinct LOCALITE)
from   CLIENT
where  CAT = 'C1'
```

1.25 Créer une table et y ranger les données suivantes relatives aux détails de commande : numéro et date de la commande, quantité commandée, numéro et prix du produit, montant du détail.

```
insert into DETAIL_COM(NCOM, DATECOM, QCOM, NPRO, PRIX, MONTANT)
select M.NCOM, DATECOM, QCOM, P.NPRO, PRIX, QCOM*PRIX
from   COMMANDE M, DETAIL D, PRODUIT P
where  M.NCOM = D.NCOM
and    D.NPRO = P.NPRO
```

1.26 Annuler les comptes négatifs des clients de catégorie C1.

```
update CLIENT
set    COMPTE = 0
where  COMPTE < 0
and    CAT = 'C1'
```

1.27 Compléter le fragment suivant de manière à former une requête valide

```
select CAT, NPRO, sum(QCOM*PRIX)
from ...
```

```
select CAT, NPRO, sum(QCOM*PRIX)
```

```
from   CLIENT C, COMMANDE M, DETAIL D, PRODUIT P
where  C.NCLI = M.NCLI
and    M.NCOM = D.NCOM
and    D.NPRO = P.NPRO
group by CAT, P.NPRO
```

1.28 Rechercher les clients qui ont commandé le produit PA60 ou le produit PA45, mais pas les deux.
Suggestion. Appliquer un *ou exclusif*.

1.29 En vous basant sur le schéma 4.5, écrivez les requêtes SQL qui donnent :

- les matières premières (produits qui n'ont pas de composants);

```
select NPRO
from   PRODUIT
where  NPRO not in (select COMPOSE from COMPOSITION)
```

- les produits finis (qui n'entrent dans la composition d'aucun autre);

- les produits semi-finis (tous les autres);

- le prix et poids unitaires d'un produit fini ou semi-fini dont tous les composants ont un poids et un prix unitaires.

4 Énoncés de type 4

1.30 Afficher le numéro et le nom des clients qui n'ont pas commandé de produits en sapin.

```
select NCLI, NOM
from CLIENT
where NCLI not in (select NCLI from COMMANDE
                   where NCOM in (select NCOM from DETAIL
                                  where NPRO in (select NPRO
                                                 from PRODUIT
                                        where LIBELLE like '%SAPIN%')));
```

1.31 Rechercher les clients qui, s'ils ont commandé le produit PA60, en ont commandé plus de 500 unités au total.
Suggestion. Il s'agit d'une relation d'implication.

```
select NCLI
from   CLIENT C
where  NCLI not in (select NCLI
```

```
                    from COMMANDE
                    where  NCOM in (select NCOM
                                    from DETAIL
                                    where  NPRO = 'PA60'))
or      (select sum(QCOM)
         from DETAIL
         where NCOM in (select NCOM
                        from COMMANDE
                        where NCLI = C.NCLI)) > 500;
```

Remarque. Curieusement, cette requête n'est pas équivalente à "*Rechercher les clients qui ont commandé plus de 500 unités de PA60*". En effet, les clients qui n'ont pas commandé de produit PA60 sont aussi sélectionnés.

1.32 A la question : "*rechercher les localités dans lesquelles on n'a pas commandé de produit PA60*", quatre utilisateurs proposent les requêtes suivantes. Indiquer la (ou les) requêtes correctes, et interprétez les autres.

```
select distinct LOCALITE
from   CLIENT
where  NCLI in
            (select NCLI
             from   COMMANDE
             where  NCOM in
                        (select NCOM
                         from   DETAIL
                         where  NPRO <> 'PA60'))

select distinct LOCALITE
from   CLIENT
where  NCLI in
            (select NCLI
             from   COMMANDE
             where  NCOM not in
                        (select NCOM
                         from   DETAIL
                         where  NPRO = 'PA60'))

select distinct LOCALITE
from   CLIENT
where  NCLI not in
            (select NCLI
             from   COMMANDE
             where  NCOM in
                        (select NCOM
                         from   DETAIL
                         where  NPRO = 'PA60'))

select distinct LOCALITE
from   CLIENT
where  LOCALITE not in
```

```
                (select LOCALITE
                 from   CLIENT
                 where  NCLI in
                         (select NCLI
                          from   COMMANDE
                          where  NCOM in
                                 (select NCOM
                                  from   DETAIL
                                  where  NPRO = 'PA60')))
```

expression correcte

1.33 Que signifie la requête suivante ?

```
select *
from   COMMANDE
where  NCOM not in (select NCOM
                    from   DETAIL
                    where  NPRO <> 'PA60')
```

Interprétation : les commandes qui ne spécifient que produit 'PA60' (= "qui n'apparaissent pas parmi les commandes qui spécifient, notamment, un produit autre que PA60")

1.34 Dans quelles localités a-t-on commandé en décembre 2008 ?

1.35 Calculer le montant de chaque commande.

```
select NCOM, sum(QCOM*PRIX)
from   COMMANDE M, DETAIL D, PRODUIT P
where  M.NCOM = D.NCOM
and    D.NPRO = P.NPRO
group by NCOM;
```

1.36 Calculer le montant dû par chaque client. Dans ce calcul, on ne prend en compte que le montant des commandes. Attention aux clients qui n'ont pas passé de commandes.

```
select NCLI, sum(QCOM*PRIX)
from   COMMANDE M, DETAIL D, PRODUIT P
where  M.NCOM = D.NCOM
and    D.NPRO = P.NPRO
group by NCLI
union
select NCLI, 0
from   CLIENT C
where  not exists (select * from COMMANDE where NCLI = C.NCLI);
```

1.37 Calculer le montant dû par les clients de chaque localité. Dans ce calcul, on ne prend en compte que le montant des commandes. Attention aux localités dans lesquelles aucun client n'a passé de commandes.

1.38 Calculez, par jour, le total des montants des commandes.

```
select DATECOM, sum(QCOM*PRIX)
from   COMMANDE M, DETAIL D, PRODUIT P
where  M.NCOM = D.NCOM
and    D.NPRO = P.NPRO
group by DATECOM
```

1.39 On suppose qu'on n'a pas trouvé utile d'imposer un identifiant primaire sur la table PRODUIT. Il se peut donc que plusieurs lignes aient même valeur de la colonne NPRO, ce qui viole le principe d'unicité des valeurs de cette colonne.

- Ecrire une requête qui recherche les valeurs de NPRO présentes en plus d'un exemplaire.

```
select NPRO, count(*)
from   PRODUIT
group by NPRO
having count(*) > 1

réponse vide, évidemment !
```

- Écrire une requête qui indique combien de valeurs de NPRO sont présentes en plus d'un exemplaire.

Suggestion. Il s'agit du calcul d'une statistique sur une autre statistique. Le plus simple est d'exprimer cette dernière sous la forme d'une vue.

```
create view OCCURRENCES(NPRO,NOMBRE) as
   select NPRO, count(*)
   from   PRODUIT
   group by NPRO;

select count(*)
from   OCCURRENCES
where  NOMBRE > 1;
```

- Écrire une requête qui indique combien de lignes comportent une erreur d'unicité de NPRO.

```
select sum(NOMBRE) - count(*) as lignes
from   OCCURRENCES;
```

- Écrire une requête qui, pour chaque valeur de NPRO présente dans la table, indique dans combien de lignes cette valeur est présente.

```
select * from OCCURRENCES;
```

- Écrire une requête qui, pour chaque valeur de NPRO qui n'est pas unique, indique dans combien de lignes cette valeur est présente.

```
select * from OCCURRENCES;
where  NOMBRE > 1;
```

- Écrire une suite de requêtes qui crée une table contenant les numéros de NPRO qui ne sont pas uniques.

1.40 On suppose qu'on n'a pas trouvé utile de déclarer NCOM clé étrangère dans la table DETAIL. Il est donc possible que certaines lignes de DETAIL violent la contrainte d'intégrité référentielle portant sur cette colonne. Ecrire une requête qui recherche les anomalies éventuelles.

```
select NCOM, NPRO
from   DETAIL D
where  not exists (select * from COMMANDE where NCOM = D.NCOM);
```

1.41 Normalement, à toute commande doit être associé au moins un détail. Ecrivez une requête qui vérifie qu'il en bien ainsi dans la base de données.

```
select NCOM
from   COMMANDE M
where  not exists (select * from DETAIL where NCOM = M.NCOM);
```

1.42 Afficher pour chaque localité, les libellés des produits qui y sont commandés.

```
select LOCALITE, LIBELLE
from   CLIENT C, COMMANDE M, DETAIL D, PRODUIT P
where  C.NCLI = M.NCLI
and    M.NCOM = D.NCOM
and    D.NPRO = P.NPRO
group by LOCALITE, LIBELLE
order by LOCALITE, LIBELLE
```

1.43 Affichez par localité, et pour chaque catégorie dans celle-ci, le total des montants des commandes. *Attention aux localités dans lesquelles on n'a pas commandé.*

```
select LOCALITE, CAT, sum(QCOM*PRIX)
from   CLIENT C, COMMANDE M, DETAIL D, PRODUIT P
where  C.NCLI = M.NCLI
```

```
and    M.NCOM = D.NCOM
and    D.NPRO = P.NPRO
group by LOCALITE, CAT
   union
select LOCALITE, CAT, 0
from   CLIENT C
group by LOCALITE, CAT
having not exists (select *
                   from COMMANDE
                   where  NCLI in (select NCLI
                                   from   CLIENT
                                   where  LOCALITE = C.LOCALITE
                                   and    CAT = C.CAT))
```

1.44 Quels sont les produits (numéro et libellé) qui n'ont pas été commandés en 2008 ?

```
select NPRO, LIBELLE
from   PRODUIT
where  NPRO not in (select NPRO
                    from   DETAIL D, COMMANDE M
                    where  D.NCOM = M.NCOM
                    and    M.DATECOM like '%2005%');
```

1.45 Indiquer, pour chaque localité, les catégories de clients qui n'y sont pas représentées

Suggestion. Construire l'ensemble de tous les couples LOCALITE x CAT possibles et en retirer ceux qui existent dans la base. Attention aux valeurs `null`, qui ne doivent pas être prises en compte.

```
select distinct L.LOCALITE, C.CAT
from   CLIENT L, CLIENT C
where  CAT is not null
and not exists (select *
                from CLIENT
                where LOCALITE = L.LOCALITE and CAT = C.CAT)
```

1.46 Produire (à l'écran) une table de couples <X,Y> de clients tels que X et Y habitent dans la même localité. On évitera de renseigner <X,X>, mais aussi <Y,X> si <X,Y> est déjà repris.

Suggestion. Auto-jointure de CLIENT. On évitera les couples inverse en imposant un ordre sur les valeurs de NPRO (p.ex. X < Y).

```
select C1.LOCALITE, C1.NCLI, C2.NCLI
from   CLIENT C1, CLIENT C2
where  C1.NCLI < C2.NCLI
and    C1.LOCALITE = C2.LOCALITE
order by C1.LOCALITE, C1.NCLI, C2.NCLI
```

1.47 En vous basant sur le schéma 5.5, écrivez une requête de mise à jour qui complète les prix et poids unitaires des produits finis ou semi-finis. Pour simplifier la procédure, on admet que cette requête soit exécutée autant de fois que nécessaire pour que tous les produits soient complétés.

```
update PRODUIT PH
set PRIX_U = (select  sum(QTE*PB.PRIX_U)
              from    COMPOSITION C, PRODUIT PB
              where   PH.NPRO = C.COMPOSE
              and     C.COMPOSANT = PB.NPRO
              group by PH.NPRO)
where  PRIX_U is null
and    not exists (select * from  COMPOSITION CC, PRODUIT BB
                   where CC.COMPOSANT = BB.NPRO
                   and   CC.COMPOSE = PH.NPRO
                   and   BB.PRIX_U is null)
```

après 3 exécutions, on obtient :

NPRO	LIBELLE	PRIX_U	POIDS_U
p1	A-200	294.0	<null>
p2	A-056	74.0	<null>
p3	B-661	25.0	<null>
p4	B-122	60.5	<null>
p5	B-326	145.5	<null>
p6	D-822	3.5	0.70
p7	D-507	8.0	0.25
p8	G-993	5.0	1.15
p9	F-016	1.0	<null>
p10	J-500	7.1	<null>
p11	J-544	0.5	0.90
p12	L-009	1.7	2.30

La mise à jour de POIDS_U se commande de manière similaire.

1.48 En considérant le schéma de la figure 5.7, calculer pour chaque ville, le prix moyen de chaque produit.

```
select VILLE, PRODUIT, avg(PRIX)
from   VENTE V, LOCALISATION L
where  V.CHAINE = L.CHAINE
group by VILLE, PRODUIT
```

1.49 Afficher pour chaque client, le nombre de commandes, le nombre de produits commandés et le nombre de détails. On se limite aux clients qui ont passé au moins une commande.
Suggestion : il s'agit d'une requête basée sur des groupements multi-niveaux.

```
select  C.NCLI, count(distinct NCOM), count(distinct NPRO), count(NPRO)
from    CLIENT C, COMMANDE M, DETAIL D
```

```
where   C.NCLI = M.NCLI and M.NCOM = D.NCOM
group by C.NCLI

select  C.NCLI, count(distinct NCOM), count(distinct NPRO), count(NPRO)
from    CLIENT C, COMMANDE M, DETAIL D
where   C.NCLI = M.NCLI and M.NCOM = D.NCOM
group by C.NCLI
```

1.50 Afficher, pour chaque localité et pour chaque catégorie, (1) le nombre de commandes passées par les clients de cette localité et de cette catégorie, (2) le montant total de ces commandes.

```
select LOCALITE, CAT, count(distinct NCOM), sum(QCOM*PRIX)
from   CLIENT C, COMMANDE M, DETAIL D, PRODUIT P
where  C.NCLI = M.NCLI
and    M.NCOM = D.NCOM
and    D.NPRO = P.NPRO
group by LOCALITE, CAT
```

5 Énoncés de type 5

1.51 Calculez le nombre moyen de produits par commande. De même : le nombre moyen de commandes par client, par localité ou par catégorie.

Remarque. Il n'est pas possible de demander directement la moyenne *d'un nombre (comptage)*. *Suggestion* : construction et interrogation d'une vue ou calcul *du nombre* dans le `from`.

```
create view NOMBRE(NCOM,NBRE) as
select NCOM, count(*)
from   DETAIL
group by NCOM;

select avg(NBRE)
from NOMBRE;
```

1.52 Quel est, pour chaque localité, le nombre moyen de commandes par client.

```
create view NOMBRE(NCLI,NBRE) as
select C.NCLI, count(*)
from   CLIENT C, COMMANDE M
where  C.NCLI = M.NCLI
group by C.NCLI;

select LOCALITE, avg(NBRE)
from   CLIENT C, NOMBRE N
where  C.NCLI = N.NCLI
group by LOCALITE;
```

1.53 Ecrire une requête SQL qui donne, pour chaque catégorie de produit, le nombre de produits qui ont été commandés le *23-12-2008*.

```
select CAT, count(distinct NPRO)
from   CLIENT C, COMMANDE M, DETAIL D
where  C.NCLI = M.NCLI
and    M.NCOM = D.NCOM
AND    DATECOM = '23-DEC-2000'
group by CAT
```

1.54 Donner pour chaque localité dans laquelle se trouve au moins un client de catégorie '*C1*' la liste des produits en sapin qu'on y a commandés.

```
select LOCALITE, D.NPRO
from   CLIENT C, COMMANDE M, DETAIL D, PRODUIT P
where  C.NCLI = M.NCLI
and    M.NCOM = D.NCOM
and    D.NPRO = P.NPRO
and    LOCALITE in (select LOCALITE from CLIENT where CAT = 'C1')
and LIBELLE like '%SAPIN%'
group by LOCALITE, D.NPRO
```

1.55 Donner pour chaque produit la liste des localités dans lesquelles ce produit est commandé en plus de 500 unités (= au total pour la localité).

```
select D.NPRO, LOCALITE, sum(QCOM)
from   CLIENT C, COMMANDE M, DETAIL D
where  C.NCLI = M.NCLI
and    M.NCOM = D.NCOM
group by D.NPRO, LOCALITE
having sum(QCOM) > 500
```

1.56 Afficher, pour chaque localité, les produits qu'on y commande et qui sont aussi commandés dans au moins une autre localité.

Suggestion. Un produit est intéressant si le nombre de localités dans lesquelles il est commandé est supérieur *ou* égal à 2.

```
select LOCALITE, NPRO
from   CLIENT C, COMMANDE M, DETAIL D
where  C.NCLI = M.NCLI
and    M.NCOM = D.NCOM
and    D.NPRO in (select NPRO
                  from   CLIENT C, COMMANDE M, DETAIL D
                  where  C.NCLI = M.NCLI
                  and    M.NCOM = D.NCOM
                  group by NPRO
                  having count(distinct LOCALITE) > 1)
group by LOCALITE, D.NPRO
```

1.57 Rechercher les clients qui ont commandé tous les produits.

Suggestion. Application du quantificateur *pour tout*. On recherche les clients tels qu'il n'existe pas de produits qui n'apparaissent pas dans les détails de leurs commandes.

```
select NCLI
from   CLIENT C
where not exists (select *
                  from PRODUIT
                  where NPRO not in (select NPRO
                                     from   DETAIL D, COMMANDE M
                                     where  D.NCOM = M.NCOM
                                     and    M.NCLI = C.NCLI))
```

1.58 Dans quelles localités a-t-on commandé tous les produits en acier (tous clients confondus) ?

Exercice préparatoire : *quels sont les clients qui ont commandé tous les produits en acier.*

```
select NCLI
from   CLIENT C
where not exists (select *
                  from   PRODUIT
                  where  LIBELLE like '%ACIER%'
                  and    NPRO not in (select NPRO
                                      from   DETAIL D, COMMANDE M
                                      where  D.NCOM = M.NCOM
                                      and    M.NCLI = C.NCLI))
```

Question d'origine :

```
les localités L telles
     qu'il n'existe pas
           de produits en acier qui ne soit commandé
                         par un client de la localité L

select distinct LOCALITE
from   CLIENT C
where not exists (select *
                  from   PRODUIT
                  where  LIBELLE like '%ACIER%'
                  and    NPRO not in
                            (select NPRO
                             from   DETAIL D, COMMANDE M, CLIENT CC
                             where  D.NCOM = M.NCOM
                             and    M.NCLI = C.NCLI
                             and    CC.LOCALITE = C.LOCALITE))
```

1.59 Rechercher les produits qui ont été commandés par tous les clients.

```
select NPRO
from   PRODUIT P
where  not exists (select *
                   from   CLIENT
                   where  NCLI not in (select NCLI
                                       from   COMMANDE M, DETAIL D
                                       where  M.NCOM = D.NCOM
                                       and    D.NPRO = P.NPRO))
```

Résultat vide (comme attendu).

1.60 Rechercher les localités dont aucun client n'a passé de commande.

```
select distinct LOCALITE
from   CLIENT
where  LOCALITE not in (select LOCALITE
                        from   CLIENT C
                        where exists (select *
                                      from   COMMANDE
                                      where  NCLI = C.NCLI))
```

1.61 Rechercher les localités dont tous les clients ont passé au moins une commande.

```
select distinct LOCALITE
from   CLIENT C
where  not exists (select * from CLIENT
                   where  LOCALITE = C.LOCALITE
                   and    NCLI not in (select NCLI from COMMANDE))
```

1.62 Rechercher les produits qui sont commandés dans toutes les localités.

```
select NCLI
from   CLIENT C
where not exists (select *
                  from PRODUIT
                  where NPRO not in (select NPRO
                                     from   DETAIL D, COMMANDE M
                                     where  D.NCOM = M.NCOM
                                     and    M.NCLI = C.NCLI))

select NPRO
from   PRODUIT P
where not exists
           (select *
            from   CLIENT
            where  LOCALITE not in
                      (select LOCALITE
                       from   CLIENT C, COMMANDE M, DETAIL D
                       where  M.NCLI = C.NCLI
                       and    D.NCOM = M.NCOM
```

```
                        and    D.NPRO = P.NPRO))
```

```
Résultat vide (comme attendu).
```

1.63 Dans quelles localités peut-on trouver au moins un client qui a commandé tous les produits en sapin, et ce, pour un montant total, pour ces produits, dépassant 10.000 ?

1.64 Calculer, pour chaque localité, le nombre de catégories distinctes.

```
select LOCALITE, count(distinct CAT)
from   CLIENT
where  CAT is not null
group by LOCALITE
```

1.65 La section 4.10.1 suggère une comparaison entre un instantané et une vue de même définition. En effet, ces deux techniques pourraient être considérées comme équivalentes pour la formulation de requêtes. Etablissez une liste de critères de comparaison et appliquez-les aux deux techniques.

1.66 Mettre à jour les comptes des clients en en déduisant le montant des commandes en cours.

Suggestion. cfr mise à jour des quantités en stock des produits; attention aux clients qui n'ont pas commandé.

```
update PRODUIT P
set QSTOCK = QSTOCK - (select sum(QCOM)
                       from   DETAIL
                       where  NPRO = P.NPRO)
where exists (select *
              from   DETAIL
              where  NPRO = P.NPRO)

select NPRO, QSTOCK
from   PRODUIT
```

1.67 Mettre à jour les quantités en stock des produits en en déduisant les quantités commandées par les clients de catégorie B1 et C1.

```
update PRODUIT P
set QSTOCK = QSTOCK - (select sum(QCOM)
                       from   DETAIL D, COMMANDE M, CLIENT C
                       where  D.NPRO = P.NPRO
```

```
                        and    D.NCOM = M.NCOM
                        and    M.NCLI = C.NCLI
                        and    CAT in ('B1','C1'))
where exists (select *
              from   DETAIL D, COMMANDE M, CLIENT C
              where  D.NPRO = P.NPRO
              and    D.NCOM = M.NCOM
              and    M.NCLI = C.NCLI
              and    CAT in ('B1','C1'))

select NPRO, QSTOCK from PRODUIT
```

1.68 On suppose qu'on dispose de deux tables PRODUIT1 et PRODUIT2 décrivant des produits de deux filiales distinctes, et de même structure que PRODUIT. Il est possible qu'un produit soit présent simultanément dans les deux filiales. Le même numéro de produit est repris alors dans les deux tables. On désire intégrer les deux dépôts. En conséquence, on fusionne les deux tables initiales en une troisième selon les règles suivantes :

- si un produit n'est présent que dans une seule filiale, sa ligne est reprise telle quelle,
- si un produit est présent dans les deux filiales, on ne le décrit qu'une seule fois. Son libellé est celui de PRODUIT1, son prix est le minimum des deux valeurs et sa quantité en stock est la somme des deux valeurs.

```
create table PRODUIT1
(NPRO char(10) not null primary key,
 LIBELLE char(30) not null,
 PRIX decimal(5,2) not null,
 QSTOCK decimal(6) not null);

insert into PRODUIT1 values ('A001','Farine', 0.8, 65);
insert into PRODUIT1 values ('B001','Sel', 0.5, 120);
insert into PRODUIT1 values ('A003','Huile olive', 5.2, 9);

create table PRODUIT2
(NPRO char(10) not null primary key,
 LIBELLE char(30) not null,
 PRIX decimal(5,2) not null,
 QSTOCK decimal(6) not null);

insert into PRODUIT2 values ('A002','Sucre', 1.2, 45);
insert into PRODUIT2 values ('B001','Sel iodé', 0.45, 35);
insert into PRODUIT2 values ('A003','Huile', 5.6, 22);

create table PRODUIT3
(NPRO char(10) not null primary key,
 LIBELLE char(30) not null,
 PRIX decimal(5,2) not null,
 QSTOCK decimal(6) not null);

insert into PRODUIT3
```

```
select *
from PRODUIT1 P1
where not exists (select * from PRODUIT2 where NPRO = P1.NPRO);

insert into PRODUIT3
select *
from PRODUIT2 P2
where not exists (select * from PRODUIT1 where NPRO = P2.NPRO);

insert into PRODUIT3
select P1.NPRO, P1.LIBELLE, P1.PRIX, P1.QSTOCK + P2.QSTOCK
from   PRODUIT1 P1, PRODUIT2 P2
where  P1.NPRO = P2.NPRO
and    P1.PRIX <= P2.PRIX;

insert into PRODUIT3
select P1.NPRO, P1.LIBELLE, P2.PRIX, P1.QSTOCK + P2.QSTOCK
from   PRODUIT1 P1, PRODUIT2 P2
where  P1.NPRO = P2.NPRO
and    P1.PRIX > P2.PRIX;
```

1.69 Soit une jointure de 2 ou plusieurs tables, dont la clause `select` spécifie certaines colonnes. On désire que le résultat ne comporte pas de lignes en double. Indiquer dans quelles conditions le modifieur `distinct` est inutile.

Suggestion. `distinct` est inutile lorsque la clause `select` comprend tous les composants de l'identifiant de la table mentionnée dans la clause `from` ou (cas d'une jointure par exemple) construite dans les clauses `from-where-group-by`.

1.70 On considère une base de données constituée des deux tables suivantes : R(A,B) et S(B,C). Evaluer l'ordre de grandeur du nombre de requêtes différentes qu'il est possible de formuler sur cette base de données.

Réponse : une infinité. A démontrer.

6 Énoncés de type 6 (*non résolus*)

1.71 La figure 7.21 (Voyages en train) représente un schéma qui est accompagné de suggestions de questions auxquelles une base de données traduisant ce schéma permettrait de répondre. On suppose que ce schéma est traduit en structure de tables (*cf.* exercice 9.3). Exprimez chacune de ces questions sous la forme de requêtes SQL.

1.72 On désire réaliser le couplage deux à deux de certaines localités où résident des clients. A cette fin, on désire construire des listes de couples candidats. On se limite cependant aux couples de localités dans lesquelles on achète au moins un même produit.

1.73 Même question que 4.69 ci-dessus, mais on se limite aux couples de localités dans lesquelles on achète exactement les mêmes produits.

1.74 Pour chaque produit, indiquer la ville dans laquelle la quantité totale commandée est minimale, et celle dans laquelle cette quantité est maximale.

7 Énoncés de type 7 (*non résolus*)

1.75 Rédigez un script (suite de requêtes) SQL qui réalise en fin de mois les opérations de gestion régies par les règles suivantes :

- pour chaque client, on calcule le total des montants des commandes du mois écoulé (= les commandes d'un mois et d'une année déterminés);
- si ce montant est supérieur à celui du mois précédent, on applique une réduction de 5%;
- ensuite, si le client est le seul de sa localité, on applique une réduction supplémentaire de 5%;
- on range dans une table les informations nécessaires à l'édition des factures du mois;
- on met à jour le compte des clients;
- on met à jour le niveau de stock (QSTOCK) des produits commandés;

- pour chaque produit dont le niveau de stock a été mis à jour, et dont le niveau est inférieur ou égal à 0, on prépare dans une table adéquate les informations de réapprovisionnement.