

```
SQL> START requetes.sql
SQL> -- 1. Les numéros des clients (sans répétition) qui ont placé au moins une commande
SQL> SELECT DISTINCT noClient
2 FROM Commande;
```

```
NOCLIENT
-----
30
40
20
10
```

```
SQL>
SQL>
SQL> -- 2. Le numéro et la description des articles dont le numéro est entre 20 et 80 (inclusivement) et
le prix est 10.99 ou 25.99
SQL> SELECT noArticle, description
2 FROM Article
3 WHERE noArticle BETWEEN 20 AND 80 AND ( prixUnitaire = 10.99 OR prixUnitaire=25.99);
```

```
NOARTICLE DESCRIPTION
-----
40 epinette bleue
70 Herbe ? puce
```

```
SQL>
SQL>
SQL> -- 3. Le numéro et la description des articles dont la description débute par la lettre C ou contient
la chaîne 'bl'
SQL> SELECT noArticle, description
2 FROM Article
3 WHERE (description LIKE 'C%') OR (description LIKE '%bl%');
```

```
NOARTICLE DESCRIPTION
-----
10 C?dre en boule
40 epinette bleue
50 Chene
60 erable argente
81 Catalpa
```

```
SQL>
SQL>
SQL> -- 4. Le numéro et le nom des clients qui ont placé une commande le 9 juillet 2000
SQL> SELECT noClient, nomclient
2 FROM Client NATURAL JOIN Commande
3 WHERE datecommande = '09/07/2000';
```

```
NOCLIENT NOMCLIENT
-----
30 Lin B?
20 Dollard Tremblay
```

```
SQL>
SQL>
SQL> -- 5. Les noms des clients, numéros de commande, date de commande et noArticle pour
SQL> -- les articles livrés le 4 juin 2000 dont la quantité livrée est supérieure à 1
SQL> SELECT nomClient, noCommande, dateCommande, noArticle
2 FROM Client
3 NATURAL JOIN Commande
4 NATURAL JOIN LigneCommande
5 NATURAL JOIN DetailLivraison
6 NATURAL JOIN Livraison
7 WHERE dateLivraison = '04/06/2000' AND quantiteLivree >1;
```

```
NOMCLIENT          NOCOMMANDE DATECOMMAN NOARTICLE
-----
Luc Sansom          1 01/06/2000      10
Dollard Tremblay    2 02/06/2000      40
```

```
SQL>
SQL> -- 6. La liste des dates du mois de juin 2000 pour lesquelles il y a au moins une livraison ou une commande.
```

```
SQL> --Les résultats sont produits en une colonne nommée DateÉvénement.
```

```
SQL> SELECT DISTINCT *
  2 FROM
  3 (SELECT datecommande AS DateEvenement
  4 FROM Commande
  5 WHERE datecommande LIKE '%06/2000%')
  6 union
  7 (SELECT datelivraison AS DateEvenement
  8 FROM Livraison
  9 WHERE datelivraison LIKE '%06/2000%');
```

```
DATEEVENEM
```

```
-----
01/06/2000
02/06/2000
03/06/2000
04/06/2000
05/06/2000
```

```
SQL>
```

```
SQL> -- 7. Les noArticle et la quantité totale livrée de l'article incluant les articles
```

```
SQL> --dont la quantité totale livrée est égale à 0.
```

```
SQL> SELECT noArticle, SUM(COALESCE(quantiteLivree,0)) AS "QUANTITE LIVREE"
  2 FROM DetailLivraison NATURAL RIGHT OUTER JOIN Article
  3 GROUP BY noArticle;
```

```
NOARTICLE QUANTITE LIVREE
```

```
-----
10 10
20 1
40 3
50 0
60 0
70 7
80 0
81 0
90 1
95 1
```

```
10 rows selected.
```

```
SQL>
```

```
SQL> -- 8. Les noArticle et la quantité totale livrée de l'article pour les articles dont le
```

```
SQL> --prix est inférieur à $20 et dont la quantité totale livrée est inférieure à 5
```

```
SQL> SELECT noArticle, QUANTITELIVREE AS "QUANTITE LIVREE"
  2 FROM (SELECT noArticle, QUANTITELIVREE
  3 FROM (SELECT noArticle, SUM(COALESCE(quantiteLivree,0)) AS QUANTITELIVREE
  4 FROM DetailLivraison NATURAL RIGHT OUTER JOIN Article
  5 GROUP BY noArticle)
  6 WHERE QUANTITELIVREE<5) NATURAL JOIN Article
  7 WHERE prixUnitaire<20
  8 ORDER BY noArticle;
```

```
NOARTICLE QUANTITE LIVREE
```

```
-----
20 1
60 0
95 1
```

```

SQL>
SQL> -- 9. Le noLivraison, noCommande, noArticle, la date de la commande, la quantité commandée,
SQL> --la date de la livraison, la quantité livrée et le nombre de jours écoulés entre la commande et la
SQL> --livraison dans le cas où ce nombre a dépassé 2 jours et le nombre de jours écoulés depuis la commande
SQL> --jusqu'à aujourd'hui est supérieur à 100
SQL> SELECT noLivraison, noCommande, noArticle, datecommande, quantite, datelivraison, quantiteLivree,
(dateLivraison-dateCommande) AS "NOMBRE JOURS ECOULES"
2 FROM Commande
3 NATURAL JOIN Lignecommande
4 NATURAL JOIN DetailLivraison
5 NATURAL JOIN Livraison
6 WHERE (dateLivraison-dateCommande > 2)
7 AND (SYSDATE-datecommande>100);

```

NOLIVRAISON	NOCOMMANDE	NOARTICLE	DATECOMMAN	QUANTITE	DATELIVRAI
103	1	90	01/06/2000	1	05/06/2000
	1	4			
101	1	10	01/06/2000	10	04/06/2000
	3	3			

```

SQL>
SQL> -- 10. La liste des Articles triée en ordre décroissant de prix et pour chacun des prix en
SQL> --ordre croissant de numéro
SQL> SELECT noArticle,description,prixUnitaire ,quantiteEnStock
2 FROM Article
3 ORDER BY prixUnitaire DESC, noArticle ASC;

```

NOARTICLE	DESCRIPTION	PRIXUNITAIRE	QUANTITEENSTOCK
80	Poirier	26,99	10
40	epinette bleue	25,99	10
81	Catalpa	25,99	10
90	Pommier	25,99	10
50	Chene	22,99	10
60	erable argente	15,99	10
95	Genevrier	15,99	10
20	Sapin	12,99	10
10	C?dre en boule	10,99	10
70	Herbe ? puce	10,99	10

10 rows selected.

```

SQL>
SQL> -- 11. Le nombre d'articles dont le prix est supérieur à 25 et le nombre d'articles
SQL> --dont le prix est inférieur à 15 (en deux colonnes)
SQL> SELECT *
2 FROM (SELECT Count(prixUnitaire) AS "NOMBREPLUSCHERQUE25"
3 FROM Article
4 WHERE prixUnitaire>25
5 ),( SELECT Count(prixUnitaire) AS "NOMBREMOINSCHERQUE15"
6 FROM Article
7 WHERE prixUnitaire<15 );

```

NOMBREPLUSCHERQUE25	NOMBREMOINSCHERQUE15
4	3

```

SQL>
SQL> -- 12. Les noCommande des commandes qui n'ont aucune livraison correspondante
SQL> SELECT noCommande
2 FROM Commande NATURAL LEFT OUTER JOIN DetailLivraison
3 WHERE noLivraison IS NULL;

```

NOCOMMANDE

```
-----
      8
      6
      7
```

```
SQL>
SQL> -- 13. En deux colonnes, les paires de numéros de commandes (différentes) qui sont faites à la même
date ainsi que la date de commande. Il faut éviter de répéter deux fois la même paire.
```

```
SQL> SELECT DISTINCT c1.noCommande, c2.noCommande, c2.dateCommande
      2 FROM Commande c1 , Commande c2
      3 WHERE c1.dateCommande = c2.dateCommande AND c1.noCommande != c2.noCommande AND c1.noCommande <
c2.noCommande;
```

NOCOMMANDE	NOCOMMANDE	DATECOMMAN
2	3	02/06/2000
5	6	09/07/2000
7	8	15/07/2000

```
SQL>
SQL> -- 14. Le montant total commandé pour chaque paire (dateCommande, noArticle) dans les cas où le
montant total dépasse 50$.
```

```
SQL> SELECT t1.datecommande, t1.noArticle, SUM(t1.MONTANTTOTALCOMMANDE) as "MONTANT TOTAL COMMANDE"
      2 FROM(SELECT dateCommande, noArticle,
      3 (quantite * prixUnitaire) AS MONTANTTOTALCOMMANDE
      4 FROM Commande
      5 NATURAL JOIN LigneCommande
      6 NATURAL JOIN Article
      7 WHERE (quantite*prixUnitaire)>50) t1
      8 GROUP BY t1.noArticle, t1.datecommande;
```

DATECOMMAN	NOARTICLE	MONTANT TOTAL COMMANDE
01/06/2000	10	109,9
09/07/2000	10	109,9
02/06/2000	40	51,98
01/06/2000	70	54,95
09/07/2000	20	64,95

```
SQL>
SQL> -- 15. Les noArticle des articles commandés dans toutes et chacune des commandes du client 20
```

```
SQL> SELECT noArticle
      2 FROM Article
      3 WHERE NOT EXISTS
      4 (SELECT noCommande
      5 FROM Commande
      6 WHERE noClient = 20 AND NOT EXISTS
      7 (SELECT *
      8 FROM LigneCommande
      9 WHERE noArticle = Article.noArticle
     10 AND noCommande = Commande.noCommande));
```

```
NOARTICLE
-----
      40
```

```
SQL>
SQL>
SQL>
SQL>
SQL> SP00L OFF
```