

4.1. Affichez l'ensemble de la table factures

Requête - ERIC\MAITRISERSQL.comptoir.ERIC\Eric Brasseur - Sans titre1*

```
select * from factures;
```

	Destinataire	Adresse livraison	Vill.
1	Vins et alcools Chevalier	59 rue de l'Abbaye	Reim:
2	Vins et alcools Chevalier	59 rue de l'Abbaye	Reim:
3	Vins et alcools Chevalier	59 rue de l'Abbaye	Reim:
4	Toms Spezialitäten	Luisenstr. 48	Müns'
5	Toms Spezialitäten	Luisenstr. 48	Müns'
6	Hanari Carnes	Rua do Paço, 67	Rio

Grilles Messages

Lot de ERIC\MAITRISERSQL (8.0) ERIC\Eric Brasseur (51) comptoir 0:00:00 2155 lignes Lg 1, Col 24

4.2. Affichez le contenu des champs prénom, nom , date embauche de la table employés

```
select [prénom],[nom],[date embauche] from employés;
```

4.3. Affichez le contenu des champs qui parlent des dates de commande, des codes postaux et des villes de livraison de la table des commandes

Requête - ERIC\MAITRISERSQL.comptoir.ERIC\Eric Brasseur - Sans titre1*

```
select [date commande],[code postal livraison],[ville livraison]
from commandes;
```

	date commande	code postal livraison	ville livraison
1	1996-07-04 00:00:00	51100	Reims
2	1996-07-05 00:00:00	44087	Münster
3	1996-07-08 00:00:00	05454-876	Rio de Janeiro
4	1996-07-08 00:00:00	69004	Lyon
5	1996-07-09 00:00:00	B-6000	Charleroi
6	1996-07-10 00:00:00	05454-876	Rio de Janeiro
7	1996-07-11 00:00:00	3012	Bern

Grilles Messages

Lot de ERIC\MAITRISERSQL (8.0) ERIC\Eric Brasseur (51) comptoir 0:00:00 830 lignes Lg 2, Col 15

4.4. : Affichez le contenu des champs qui parlent des références, des noms, des quantités en stock et des prix unitaires des produits.

```
select [réf produit],[nom du produit],[unités en stock],[prix unitaire] from produits;
```

4.5. Nous voulons le nom, le prénom et la date d'embauche des employés engagés après le 04/03/94

Requête - ERIC\MAITRISERSQL.comptoir.ERIC\Eric Brasseur - Sans titre1*

```
SELECT [nom], [prénom], [date embauche]
FROM [employés]
WHERE [date embauche] > '04/3/94';
```

	nom	prénom	date embauche
1	Callahan	Laura	1994-03-05 00:00:00
2	Dodsworth	Anne	1994-11-15 00:00:00

Grilles Messages

Lot de requête: ERIC\MAITRISERSQL (8.0) ERIC\Eric Brasseur (51) comptoir 0:00:00 2 lignes Lg 3, Col 29

Remarquez les formats des dates et l'usage des apostrophes.
Attention : sur ORACLE, il faudrait écrire > '4-MAR-1994'

4.6. Nous voulons les noms des sociétés clientes dont nous ne connaissons pas la région

Requête - ERIC\MAITRISERSQL.comptoir.ERIC\Eric Brasseur - Sans titre1*

```
SELECT [société]
FROM [clients]
WHERE région IS NULL;
```

Toujours utiliser IS et pas =

	société
1	Alfreds Futterkiste
2	Ana Trujillo Emparedados y ...
3	Antonio Moreno Taquería
4	Around the Horn

Grilles Messages

Lot de requête: ERIC\MAITRISERSQL (8.0) ERIC\Eric Brasseur (51) comptoir 0:00:00 60 lignes Lg 1, Col 17

4.7. Nous voulons le code, le nom et la quantité des produits quand il en reste moins de 5.

```
SELECT [réf produit], [nom du produit],[unités en stock]
FROM [produits]
WHERE [unités en stock] < 5;
```

4.8. Nous voulons le code, le nom et la quantité des produits quand il en reste moins que le minimum vital.

```
SELECT [réf produit], [nom du produit],[unités en stock]
FROM [produits]
WHERE [unités en stock] < [niveau de réapprovisionnement];
```

4.9. Nous voulons le code client, l'adresse de livraison, le nom du vendeur, la remise et le numéro de facture lors d'une facture avec remise supérieure ou égale à 4 %

The screenshot shows a SQL query window with the following query:

```
SELECT [code client], [adresse livraison],[vendeur],[N° commande],
[remise (%)]
FROM [factures]
WHERE [remise (%)] >= 0.04;
```

The results are displayed in a grid with the following data:

	code client	adresse livraison	vendeur	N° commande	rem
1	HANAR	Rua do Paço, 67	Margaret Peacock	10250	0.:
2	HANAR	Rua do Paço, 67	Margaret Peacock	10250	0.:
3	VICTE	2, rue du Commerce	Janet Leverling	10251	5.0
4	VICTE	2, rue du Commerce	Janet Leverling	10251	5.0

The status bar at the bottom indicates: Lot de requêt ERIC\MAITRISERSQL (8.0) ERIC\Eric Brasseur (51) comptoir 0:00:00 832 lignes Lg 5, Col 1

4.10 Testez ceci avec l'analyseur de requêtes : IF '' = '' print 'TRUE' else print 'FALSE'

The screenshot shows a SQL query window with the following query:

```
IF '' = '' print 'TRUE' else print 'FALSE'
```

The result displayed is:

```
TRUE
```

The status bar at the bottom indicates: ERIC\MAITRISERSQL (8.0) ERIC\Eric Brasseur (51) comptoir 0:00:00 0 lignes Lg 1, Col 1

4.11 Testez ceci avec l'analyseur de requêtes : IF NULL = NULL print 'TRUE' else print 'FALSE'

The screenshot shows a SQL query window with the following query:

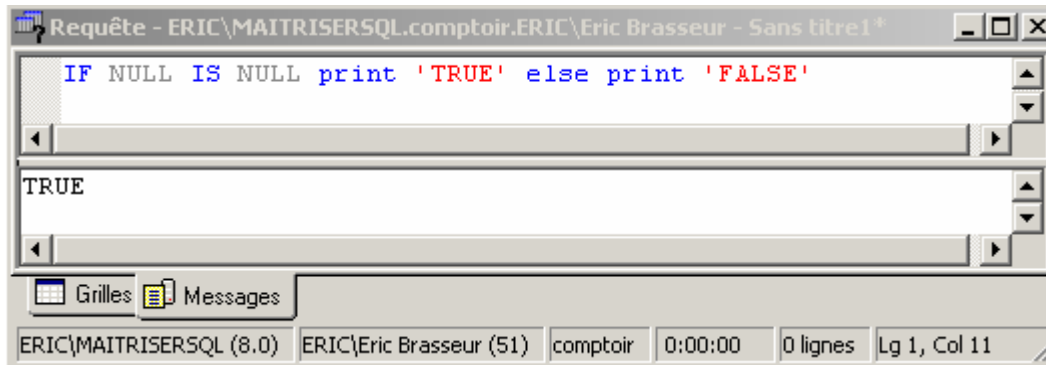
```
IF NULL = NULL print 'TRUE' else print 'FALSE'
```

The result displayed is:

```
FALSE
```

The status bar at the bottom indicates: ERIC\MAITRISERSQL (8.0) ERIC\Eric Brasseur (51) comptoir 0:00:00 0 lignes Lg 1, Col 10

4.12 Testez ceci avec l'analyseur de requêtes : IF NULL IS NULL print 'TRUE' else print 'FALSE'

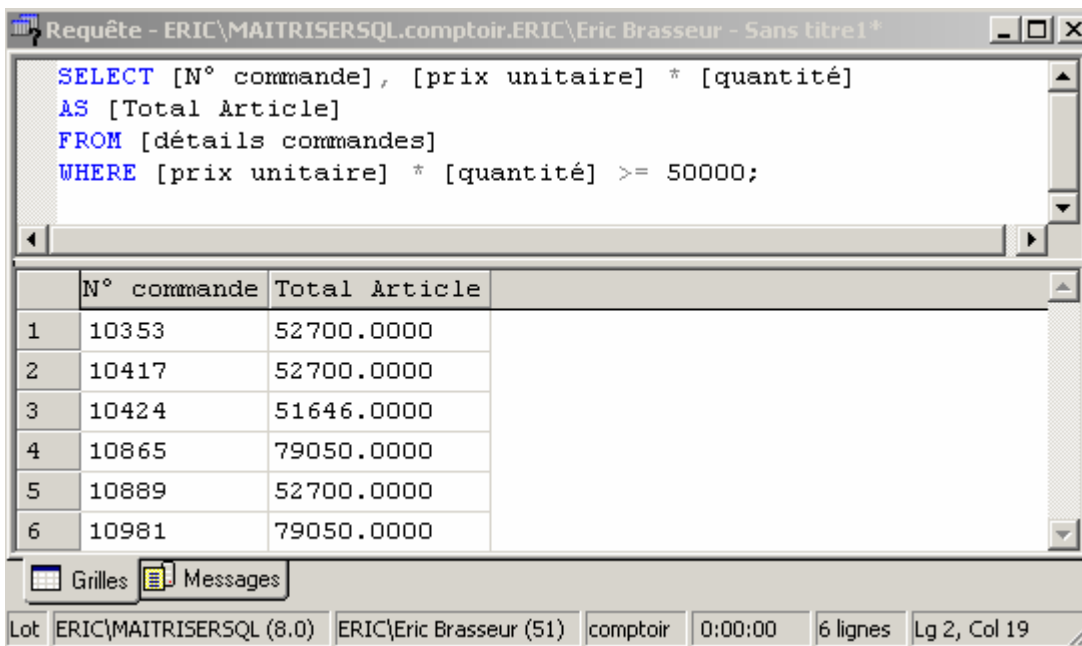


Remarquez l'importance du IS

4.13. Nous voudrions connaître les commandes dans lesquelles un client a dépensé au moins 50 000 francs pour un seul article (voir détails commandes).

```
SELECT [N° commande], [prix unitaire] * [quantité]
FROM [détails commandes]
WHERE [prix unitaire] * [quantité] >= 50000;
```

4.14. Même exercice mais en donnant un titre "Total Article" à la deuxième colonne



4.15. On voudrait connaître le nombre d'articles qui restent avant d'atteindre la quantité limite

```
SELECT [nom du produit], [unités en stock] - [niveau de réapprovisionnement]
AS [Avant alerte]
FROM [produits] ;
```

4.16. On voudrait connaître tous les employés qui sont nés avant le 10/06/1952 et qui ont été embauchés après le 01/01/1991.

```
SELECT [nom], [prénom],[date de naissance],[date embauche]
FROM [employés]
WHERE [date de naissance] < '10/06/1952'
AND [date embauche] > '01/01/1991';
```

4.17. On voudrait connaître tous les numéros de commandes et le code client pour les commandes livrées à São Paulo après le 01/01/1998 ainsi que celles livrées hors de São Paulo avant le 01/01/1997.

```
SELECT [n° commande],[code client],[date envoi],[ville livraison]
FROM [commandes]
WHERE ([date envoi] > '01/01/1998' AND [ville livraison] = 'São Paulo')
      OR ([date envoi] < '01/01/1997' AND [ville livraison] <> 'São Paulo');
```

4.18. Lister tous les produits (nom, code, prix) dont le prix est compris entre 50.00 et 100.00

```
SELECT [réf produit],[nom du produit],[prix unitaire]
FROM [produits]
WHERE [prix unitaire] BETWEEN 50.00 AND 100.00;
```

4.19. Lister tous les clients dont le nom de société commence par B ou C.

```
SELECT [code client],[société]
FROM [clients]
WHERE [société] BETWEEN 'B' AND 'D';
```

Si une société porte le nom D, on aura une erreur. Nous verrons plus loin les autres possibilités

4.20. Lister les n° de commandes et la référence du produit qui en une seule commande a généré un chiffre d'affaires compris entre 10000 et 11000.

```
SELECT [N° commande],[réf produit],[Prix Unitaire]*[Quantité]
AS [TOTAL PRODUIT]
FROM [détails commandes]
WHERE [Prix Unitaire]*[Quantité] BETWEEN 10000 AND 11000;
```

4.21. Rechercher tous les produits dont la deuxième lettre est un e.

```
SELECT [Nom du produit]
FROM [produits]
WHERE [Nom du produit] LIKE '_e%';
```

4.22. Rechercher tous les clients dont le nom de société a m n ou o pour troisième lettre.

The screenshot shows a SQL query window titled "Requête - ERIC\MAITRISERSQL.comptoir.ERIC\Eric Brasseur - Sans titre1*". The query is:

```
SELECT société
FROM [clients]
WHERE [société] LIKE '__[mno]%';
```

The results are displayed in a table with the following data:

	société
1	Around the Horn
2	Blondel père et fils
3	Bon app'
4	Centro comercial Moctezuma
5	Chop-suey Chinese
6	Comércio Mineiro

The window also shows a status bar at the bottom with the text: "Lot de requêtes terminé. ERIC\MAITRISERSQL (8.0) ERIC\Eric Brasseur (51) comptoir 0:00:00 22 lignes Lg 3, Col 31".

4.23. Lister les n° de commandes et la référence du produit qui en une seule commande a généré un chiffre d'affaires compris entre 10000 et 11000. Le tout trié sur la quantité * le prix unitaire par ordre décroissant.

Requête - ERIC\MAITRISERSQL.comptoir.ERIC\Eric Brasseur - Sans titre1*

```

SELECT [N° commande], [réf produit],[Prix Unitaire] * [Quantité]
AS [TOTAL PRODUIT]
FROM [détails commandes]
WHERE [Prix Unitaire] * [Quantité] BETWEEN 10000 AND 11000
ORDER BY [Prix Unitaire] * [Quantité] DESC;

```

	N° commande	réf produit	TOTAL PRODUIT
1	10567	59	11000.0000
2	10823	59	11000.0000
3	10892	59	11000.0000
4	10598	27	10975.0000
5	11053	18	10937.5000
6	10339	17	10920.0000

Grilles Messages

Lot de requêtes terminé. ERIC\MAITRISERSQL (8.0) ERIC\Eric Brasseur (51) comptoir 0:00:00 19 lignes Lg 5, Col 41

4.24. Afficher les noms, prénoms et fonctions des employés, triés sur le nom puis le prénom par ordre croissant

```

SELECT [Nom], [prénom],[fonction]
FROM [employés]
ORDER BY [Nom], [prénom];

```

4.25. Afficher le destinataire, l'adresse de livraison[3], la ville de livraison [2], le pays de livraison [1] et le montant total de la facture quand le montant total est supérieur à 3000. Triez dans l'ordre indiqué dans les crochets

Requête - ERIC\MAITRISERSQL.comptoir.ERIC\Eric Brasseur - Sans titre1*

```

SELECT [destinataire], [adresse livraison],[ville livraison],
[pays livraison],[prixtotal]
FROM [factures]
WHERE [prixtotal] > 3000
ORDER BY [pays livraison], [ville livraison],[adresse livraison];

```

	destinataire	adresse livraison	ville livraison	pays livraison	prixtotal
1	Drachenblut Delikatessen	Walserweg 21	Aachen	Allemagne	3280.0000
2	Drachenblut Delikatessen	Walserweg 21	Aachen	Allemagne	8250.0000
3	Alfreds Futterkiste	Obere Str. 57	Berlin	Allemagne	4125.0000
4	Königlich Essen	Maubelstr. 90	Brandenburg	Allemagne	4692.0000
5	Königlich Essen	Maubelstr. 90	Brandenburg	Allemagne	5238.1200
6	Königlich Essen	Maubelstr. 90	Brandenburg	Allemagne	39525.0000
7	Königlich Essen	Maubelstr. 90	Brandenburg	Allemagne	5309.1000
8	Königlich Essen	Maubelstr. 90	Brandenburg	Allemagne	6875.0000
9	Königlich Essen	Maubelstr. 90	Brandenburg	Allemagne	6000.0000
10	Königlich Essen	Maubelstr. 90	Brandenburg	Allemagne	14854.8000
11	Königlich Essen	Maubelstr. 90	Brandenburg	Allemagne	4530.7500
12	Königlich Essen	Maubelstr. 90	Brandenburg	Allemagne	3150.0000
13	Königlich Essen	Maubelstr. 90	Brandenburg	Allemagne	6840.0000

Grilles Messages

Lot de requêtes terminé. ERIC\MAITRISERSQL (8.0) ERIC\Eric Brasseur (51) comptoir 0:00:00 597 lignes Lg 4, Col 25

4.26 Affichez les 10 plus petites commandes qui ont été passées.

```
SELECT TOP 10
    [N° commande], [SousTotal]
FROM [Sous-totaux commandes]
ORDER BY [SousTotal] ;
```

4.27 Quelles étaient les 10 plus importantes ventes de 1997 ?

```
SELECT TOP 10
    [N° commande], [PrixTotal], [Date envoi]
FROM [factures]
WHERE [date envoi] BETWEEN '1/1/1997' AND '31/12/1997'
ORDER BY [PrixTotal] DESC;
```

4.28 Affichez les 10 plus importantes commandes passées.

```
SELECT TOP 10
    [N° commande], [SousTotal]
FROM [sous-totaux commandes]
ORDER BY [SousTotal] DESC;
```

4.29 Quels étaient les dix premiers pour-cent de tous les produits vendus en 1997 ?

```
SELECT TOP 10 PERCENT
    [Nom de catégorie], [nom du produit],[ventesproduit]
FROM [Ventes pour 1997]
ORDER BY [Ventesproduit] DESC;
```

4.30 Affichez les 10 premiers clients par nom de société.

The screenshot shows a window titled "Requête - ERIC\MAITRISERSQL.comptoir.ERIC\Eric Brasseur - Sans titre1". The query window contains the following SQL code:

```
SELECT TOP 10
    [code client], [Société],[adresse]
FROM [clients]
ORDER BY [société];
```

Below the query window, a table displays the results of the query. The table has four columns: "code client", "Société", and "adresse". The results are as follows:

	code client	Société	adresse
2	ANATR	Ana Trujillo Emparedados y ...	Avda. de la Constitución 2
3	ANTON	Antonio Moreno Taquería	Mataderos 2312
4	AROUT	Around the Horn	120 Hanover Sq.
5	BERGS	Berglunds snabbköp	Berguvsvägen 8
6	BLAUS	Blauer See Delikatessen	Forsterstr. 57
7	BLONP	Blondel père et fils	24, place Kléber
8	BOLID	Bólido Comidas preparadas	C/ Araquil, 67
9	BONAP	Bon app'	12, rue des Bouchers
10	BOTTM	Bottom-Dollar Markets	23 Tsawassen Blvd.

At the bottom of the window, there are buttons for "Grilles" and "Messages", and a status bar showing "Lot de requêtes te ERIC\MAITRISERSQL (8.0) ERIC\Eric Brasseur (51) comptoir 0:00:00 10 lignes Lg 1, Col 15".

4.31. Quelles sont les villes des Etats-Unis et du Canada qui ont été l'objet de livraisons ?

```
SELECT DISTINCT [Ville livraison],[région livraison],[pays livraison]
FROM [commandes]
WHERE [pays livraison] = 'Etats-Unis' OR [pays livraison] = 'Canada';
```

Pourquoi ai-je ajouté la région ?

4.32. Trouvez tous les clients qui ont passé des commandes en 1997.

```
SELECT DISTINCT [code client]
FROM [commandes]
WHERE [date commande] BETWEEN '1/1/1997' AND '31/12/1997';
```

4.33. Dressez la liste de tous les salariés qui ont passé des commandes.

```
SELECT DISTINCT [N° employé]
FROM [commandes];
```

4.34. Dressez la liste des villes d'Allemagne où vit au moins un de nos clients.

The screenshot shows a Microsoft Access query window titled "Requête - ERIC\MAITRISERSQL.comptoir.ERIC\Eric Brasseur - Sans titre1*". The query text is:

```
SELECT DISTINCT [ville], [région]
FROM [clients]
WHERE [pays] = 'Allemagne';
```

The results are displayed in a table with the following data:

	ville	région
1	Aachen	NULL
2	Berlin	NULL
3	Brandenburg	NULL
4	Cunewalde	NULL
5	Frankfurt a.M.	NULL
6	Köln	NULL
7	Leipzig	NULL
8	Mannheim	NULL
9	München	NULL
10	Münster	NULL
11	Stuttgart	NULL

The status bar at the bottom indicates: Lot de requêtes te ERIC\MAITRISERSQL (8.0) ERIC\Eric Brasseur (51) comptoir 0:00:00 11 lignes Lg 3, Col 27

4.35. Listez les produits qui n'ont jamais été vendus.

```
SELECT [réf produit]
FROM [produits]
WHERE [réf produit] NOT IN
  (SELECT DISTINCT [réf produit]
   FROM [détails commandes]) ; ( ne soyez pas surpris : pas de produit invendu )
```

4.36. Listez les produits que le client ALFKI n'a jamais acheté.

```

SELECT [réf produit], [nom du produit]
FROM [produits]
WHERE [réf produit] NOT IN
  (SELECT DISTINCT [réf produit]
   FROM [détails commandes]
   WHERE [N° commande] IN
     (SELECT DISTINCT [N° commande]
      FROM [commandes]
      WHERE [code client] = 'ALFKI'
     )
  )
);

```

4.37. Quel est le montant des ventes à nos clients allemands en 1997 ?

```

SELECT [N° commande], [PrixTotal]
FROM [factures]
WHERE [code client] IN
  ( SELECT [code client]
    FROM [clients]
    WHERE [pays] = 'Allemagne'
  )
AND [date envoi] BETWEEN '01/01/1997' AND '31/12/1997' ;

```

4.38. Quels sont les transporteurs que la société ANATR n'utilise pas ?

Requête - ERIC\MAITRISERSQL.comptoir.ERIC\Eric Brasseur - Sans titre1*

```

SELECT [nom du messenger]
FROM [messagers]
WHERE [nom du messenger] NOT IN
  ( SELECT DISTINCT [Destinataire]
    FROM [commandes]
    WHERE [code client] = 'ANATR'
  );

```

	nom du messenger
1	Speedy Express
2	United Package
3	Federal Shipping

Grilles Messages

Lot de requêtes ten ERIC\MAITRISERSQL (8.0) ERIC\Eric Brasseur (51) comptoir 0:00:00 3 lignes Lg 3, Col 26

4.39. Affichage des contacts pour tout client qui nous a passé une commande.

```

SELECT [Code client],[société],[contact],[téléphone]
FROM [clients]
WHERE EXISTS
  ( SELECT DISTINCT [code client]
    FROM [commandes]
    WHERE [code client] = clients.[code client]
  )
ORDER BY [code client];

```

4.40. Même exercice mais pour les clients qui n'on rien acheté.

```

SELECT [Code client],[société],[contact],[téléphone]
FROM [clients]
WHERE NOT EXISTS
  ( SELECT DISTINCT [code client]
    FROM [commandes]
    WHERE [code client] = clients.[code client]
  )
ORDER BY [code client];

```

4.41. Liste des produits jamais vendus.

```

SELECT [réf produit]
FROM [produits]
WHERE NOT EXISTS
  (SELECT DISTINCT [réf produit]
   FROM [détails commandes]
  );

```

4.42. Liste des produits jamais achetés par le client ALFKI.

```

SELECT [réf produit], [nom du produit]
FROM [produits]
WHERE NOT EXISTS
  (SELECT DISTINCT [réf produit]
   FROM [détails commandes]
   WHERE EXISTS
     (SELECT DISTINCT [N° commande]
      FROM [commandes]
      WHERE [code client] = 'ALFKI'
            AND [Détails commandes].[N° commande]=[commandes].[N° commande]
     )
  )
AND [détails commandes].[réf produit] = [produits].[réf produit]
);

```

4.43. Quel est le montant des ventes à nos clients allemands en 1997 ?

```

SELECT [N° commande], [PrixTotal]
FROM [factures]
WHERE EXISTS
  ( SELECT [code client]
    FROM [clients]
    WHERE [pays] = 'Allemagne'
          AND [factures].[code client] = [clients].[code client]
  )
AND [date envoi] BETWEEN '01/01/1997' AND '31/12/1997' ;

```

4.44. Liste des messagers que l'entreprise ANATR n'utilise pas.

```

SELECT [nom du messenger]
FROM [Messagers]
WHERE NOT EXISTS
  ( SELECT DISTINCT [N° employé]
    FROM [commandes]
    WHERE [code client] = 'ANATR'
          AND [messagers].[nom du messenger] = [commandes].[destinataire]
  );

```

5.1. Choisissez et justifiez le type de données utilisé pour : âge, numéro de téléphone, numéro de commande, taux et pourcentages, données scientifiques et nombres importants, population d'un pays, titre, état ou province, dates, codes postaux

Age	Pour u'ne personne : TINYINT si on parle d'années Pour un fossile : DEC
Numéro de téléphone	Pour pouvoir caser des symboles (/ , espaces) il faut prendre du caractère : VARCHAR.
Numéro de commande	Si ce sont vos numéros : un entier. Si un numéro extérieur, peut-être des caractères
Taux et pourcentages	Type décimal (DEC)
Données scientifiques et valeurs importantes	Float
Population des pays	On pourrait placer un entier, mais dans quelques années la Chine nécessitera peut-être un DEC(10,0)
Titre	Champ de texte car très variable ou un VARCHAR long
Etat ou Province	Vous vous basez sur le nom d'état le plus long et vous choisissez un VARCHAR. Si un autre état avec un nom plus long apparaît, vous l'abrégez.
Dates	Avec SQL Server, utilisez DATETIME, évitez les chaînes de caractères qui ne permettent pas les fonctions de manipulation des dates.
Codes postaux	Utilisez des VAZRCHAR car vous ne devez pas effectuer de calculs sur des codes postaux. De plus, certains pays utilisent des lettres.

5.2. Vous créez une table Plans d'Assurance qui contiendra les enregistrements du tableau ci-dessous et qui devront respecter ces conditions :

- assurance décès : de 500 000 à 6 000 000.
- pages du contrat : de 1 à 24 pages
- plan familial disponible : oui/non

La création de la table :

```
CREATE TABLE [Plansassur]
( [IDPanassur] TINYINT,
  [NomPlanAssur] VARCHAR(25),
  [CompagnieAssur] SMALLINT,
  [NuméroFournisseur] VARCHAR(15),
  [DateDébut] DATETIME,
  [AssurDécès] MONEY,
  [PagesContrat] TINYINT,
  [PrimeBase] SMALLMONEY,
  [PlanFamilial] BIT
);
```

Je remplis la table :

```
INSERT INTO [Plansassur]
([IDPanassur],[NomPlanAssur],[CompagnieAssur],
 [NuméroFournisseur],[DateDébut],[AssurDécès],
 [PagesContrat],[PrimeBase],[PlanFamilial])
VALUES
(1,'Plan Simplicité',1,'43N21-89','01/01/1991',
 500000,3,550,1);
INSERT INTO [Plansassur]
([IDPanassur],[NomPlanAssur],[CompagnieAssur],
 [NuméroFournisseur],[DateDébut],[AssurDécès],
 [PagesContrat],[PrimeBase],[PlanFamilial])
VALUES
(2,'Plan Bonus',1,'43N21-156','02/02/1992',
```

```

2000000,16,600,0);
INSERT INTO [PlansAssur]
  ([IDPanassur],[NomPlanAssur],[CompagnieAssur],
  [NuméroFournisseur],[DateDébut],[AssurDécès],
  [PagesContrat],[PrimeBase],[PlanFamilial])
VALUES
  (3,'Tout en un',2,'NW345','03/03/1993',
  4000000,21,1665,1);
INSERT INTO [PlansAssur]
  ([IDPanassur],[NomPlanAssur],[CompagnieAssur],
  [NuméroFournisseur],[DateDébut],[AssurDécès],
  [PagesContrat],[PrimeBase],[PlanFamilial])
VALUES
  (4,'Plan Sérénité',2,'NW002','04/04/1994',
  6000000,8,700,0);

```

Tentez un champ avec incrémentation automatique pour l'IDPanassur.

5.3. Modifiez la structure de votre table CompagniesAssurance de manière à ajouter une ville (maximum 15 caractères) et le code des noms des provinces (toujours 2 lettres). Ajoutez maintenant 2 enregistrements à la table nouvellement structurée.

```

ALTER TABLE [CompagniesAssurance]
ADD [Ville] VARCHAR(15);
ALTER TABLE [CompagniesAssurance]
ADD [Province] CHAR(2);

INSERT INTO CompagniesAssurance
  (IDCompAssur, NomCompAssur, DirecteurComptCompAssur,
  TelephDirecteurComptCompAssur, ville, province)
VALUES (3, 'Zurich', 'Pol Chose', '714-4112', 'Liège', 'LG');
INSERT INTO CompagniesAssurance
  (IDCompAssur, NomCompAssur, DirecteurComptCompAssur,
  TelephDirecteurComptCompAssur, ville, province)
VALUES (4, 'AXA', 'Marc truc', '987-6543', 'Bruxelles', 'BR');

```

	IDCompAssur	NomCompAssur	DirecteurComptCompAssur	TelephDirecteurComptCompAssur	Ville	Province
1	1	A-One	Adam Able	123-4567	NULL	NULL
2	2	Benchmark	Beth Bronson	234-5678	NULL	NULL
3	3	Zurich	Pol Chose	714-4112	Liège	LG
4	4	AXA	Marc truc	987-6543	Bruxelles	BR

5.4. Supprimez le champ des provinces en respectant le SQL pur.

```

CREATE TABLE [attente]
  ( IDCompAssur INT,
  NomCompAssur VARCHAR(25),
  DirecteurComptCompAssur CHAR(25),
  TelephDirecteurComptCompAssur CHAR(12),
  Ville VARCHAR(15)
);

INSERT INTO [Attente]
  SELECT [IDCompAssur],[NomCompAssur],[DirecteurComptCompAssur],
  [TelephDirecteurComptCompAssur],[Ville]
FROM [CompagniesAssurance] ;

```

```
DROP TABLE CompagniesAssurance;
```

```
EXEC SP_RENAME [Attente], [CompagniesAssurance];
```

The screenshot shows a SQL query window titled 'Requête - ERIC\MAITRISERSQL.comptoir.ERIC\Eric Brasseur - Sans titre1'. The query is:

```
SELECT *
FROM [CompagniesAssurance];
```

The results are displayed in a table with the following columns: IDCompAssur, NomCompAssur, DirecteurComptCompAssur, TelephDirecteurComptCompAssur, and Ville.

	IDCompAssur	NomCompAssur	DirecteurComptCompAssur	TelephDirecteurComptCompAssur	Ville
1	1	A-One	Adam Able	123-4567	NULL
2	2	Benchmark	Beth Bronson	234-5678	NULL
3	3	Zurich	Pol Chose	714-4112	Liège
4	4	AXA	Marc truc	987-6543	Bruxelles

The status bar at the bottom indicates: Lot de requêtes terminé. ERIC\MAITRISERSQL (8.0) ERIC\Eric Brasseur (51) comptoir 0:00:00 4 lignes Lg 2, Col 28

5.5. Transformez le champ IDPanAssur en SMALLINT et le champ nom de cette table en chaîne variable d'un maximum de 20 caractères.

```
ALTER TABLE [PlansAssur]
ALTER COLUMN [IDPanAssur] SMALLINT;
ALTER TABLE [PlansAssur]
ALTER COLUMN [NomPlanAssur] VARCHAR(20);
```

5.6. Construisez la base complète loclivre avec les tables et leurs contraintes pour le cas suivant :

Encodez une dizaine de clients et d'emprunts.



```
CREATE TABLE CLIENTS
(Numclient INTEGER NOT NULL PRIMARY KEY,
[Nom du Client] VARCHAR(25) NOT NULL,
[Prénom du Client] VARCHAR(25),
[Adresse du client] VARCHAR(40)
);
```

Avant d'insérer une ligne, tapez celle-ci qui vous permet de connaître le numéro de client à encoder.

```
SELECT COALESCE((SELECT MAX([Numclient]) FROM CLIENTS),0) + 1
```

```
INSERT INTO [CLIENTS]
([NumClient],[Nom du Client],[Prénom du client],[Adresse du Client])
VALUES (1,'Brasseur','Eric','Rue des Falises, 11 à 5580 Rochefort');
```

Requête - ERIC\MAITRISERSQL.LOCLIVRE.ERIC\Eric Brasseur - Sans titre1*

```
select * from clients
```

	Numclient	Nom du Client	Prénom du Client	Adresse du client
1	1	Brasseur	Eric	Rue des Falises, 11 à 5580 Rochefort
2	2	Borremans	Christiane	Rue des Falises, 11 à 5580 Rochefort
3	3	Barzin	Frédéric	Rue de La Martinette ,2 5580 Rochefort
4	4	Dupont	Pol	Rue du Bois ,12 6900 Aye
5	5	Charlier	André	Rue du Commerce ,1 à 5000 Namur
6	6	Charlier	Aline	Rue du Commerce ,1 à 5000 Namur
7	7	Marnix	Serge	Place de l'Eglise,14 à 5590 Ciney
8	8	Picard	Oscar	Place du Marché,24 à 5500 Dinant
9	9	Sénéchal	Arlette	Impasse du Carré à 1200 Bruxelles
10	10	Valentin	Henry	Bld de l'Empereur 4 à 1215 Bruxelles

Lot de requêtes terminé. ERIC\MAITRISERSQL (8.0) ERIC\Eric Brasseur (51) LOCLIVRE 0:00:00 10 lignes Lg 1, Col 22

```
CREATE TABLE EMPRUNT
([Numemprunt] INTEGER NOT NULL PRIMARY KEY,
[NumClient] INTEGER NOT NULL FOREIGN KEY
REFERENCES [CLIENTS]([NumClient]),
[LivreEmprunté] VARCHAR(25) NOT NULL,
[DateEmprunt] DATETIME NOT NULL,
[DateRetour] DATETIME
);
```

Il est peut-être intéressant de poser une contrainte supplémentaire sur la date de retour (\geq date d'emprunt) ?
On peut encore le faire :

```
ALTER TABLE EMPRUNT
ADD CONSTRAINT con_DateEmpruntInfDateRetour
CHECK ([DateEmprunt] <= [DateRetour]);
```

Ajoutons quelques emprunts :

```
SELECT COALESCE((SELECT MAX([Numemprunt]) FROM EMPRUNT),0) + 1
```

```
INSERT INTO [EMPRUNT]
([Numemprunt],[NumClient],[LivreEmprunté],[DateEmprunt],[DateRetour])
VALUES (1,5,'MAITRISER SQL','01/12/2001','06/12/01');
```

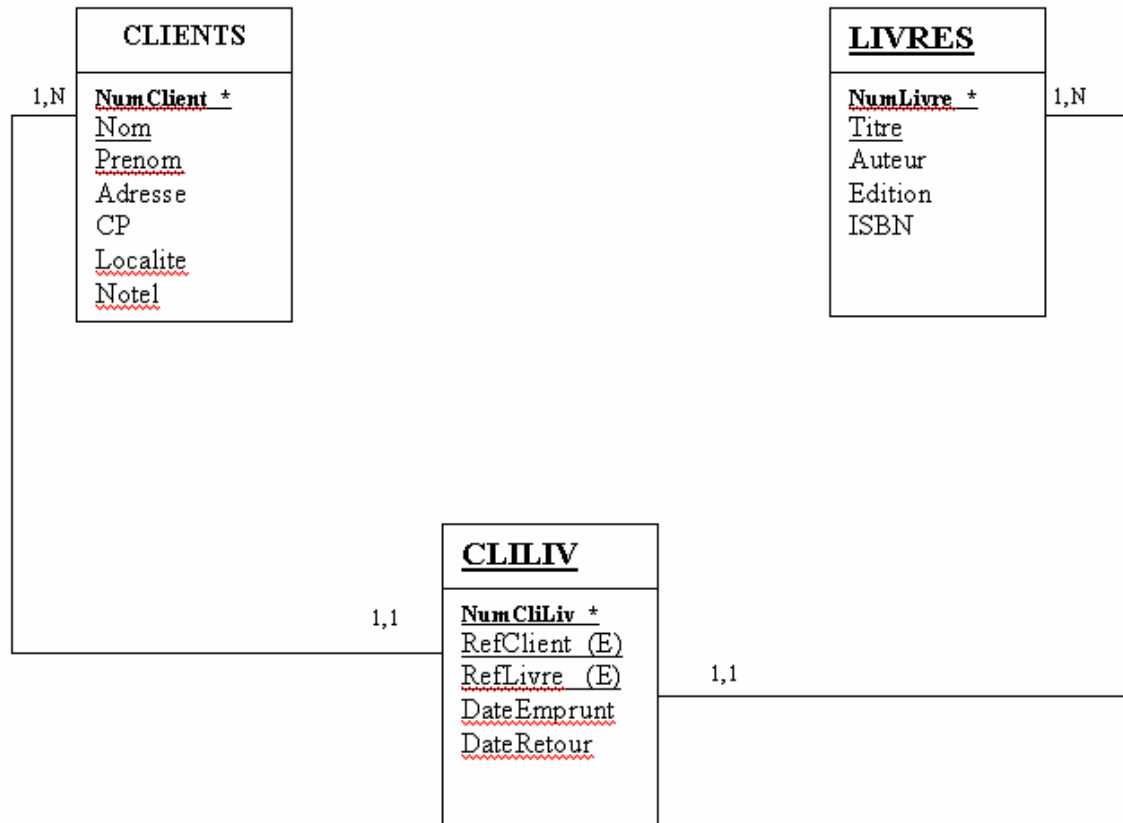
Requête - ERIC\MAITRISERSQL.LOCLIVRE.ERIC\Eric Brasseur - Sans titre1*

```
SELECT * FROM EMPRUNT
```

	Numemprunt	NumClient	LivreEmprunté	DateEmprunt	DateRetour
1	1	5	MAITRISER SQL	2001-12-01 00:00:00.000	2001-12-06 00:00:00.000
2	2	3	MySQL	2002-01-21 00:00:00.000	2002-02-05 00:00:00.000
3	3	6	Bases de Données	2002-02-15 00:00:00.000	2002-03-05 00:00:00.000

Lot de requêtes terminé. ERIC\MAITRISERSQL (8.0) ERIC\Eric Brasseur (51) LOCLIVRE 0:00:00 3 lignes Lg 1, Col 2

5.7. Même exercice mais un peu plus difficile :



```

CREATE TABLE CLIENTS
(Numclient INTEGER NOT NULL PRIMARY KEY,
[Nom] VARCHAR(25) NOT NULL,
[Prénom] VARCHAR(25),
[Adresse] VARCHAR(40),
[CP] CHAR(6),
[Localité] VARCHAR(25),
[N° Tél] VARCHAR(15),
);
  
```

Puis une dizaine de clients :

```

INSERT INTO [CLIENTS]
([NumClient],[Nom],[Prénom],[Adresse],[CP],[Localité],[N° Tél])
VALUES (1,'Brasseur','Eric','Rue des Falises, 11','5580','Rochefort','061/465699');
  
```

	Numclient	Nom	Prénom	Adresse	CP	Localité	N° Tél
1	1	Brasseur	Eric	Rue des Falises, 11	5580	Rochefort	061/465699
2	2	Borremans	Cristiane	Rue des Falises, 11	5580	Rochefort	061/465699
3	3	Barzin	Frédéric	Rue des Falises, 11	5580	Rochefort	NULL
4	4	Dupont	Pierre	Rue des Ecoles, 61	6900	On	NULL
5	5	Davreux	Pol	Place Carrée, 6	6832	Curfoz	NULL
6	6	Castoin	Yves	Place du Marais, 6a	75000	Paris	003345778921
7	7	Dubois	Marc	Boulevard du Rhum	16154	Aalst	NULL
8	8	Covillard	Henry	Allée du Renard 35	5590	Ciney	083/21 35 84
9	9	Pierret	Jacques	Rue des Estaminets 25	5000	Namur	081/22 12 14
10	10	Panis	Serge	Ruelle du char	5500	Dinant	082/32 77 78

```
CREATE TABLE LIVRES
([NumLivre] INTEGER NOT NULL PRIMARY KEY,
 [Titre] VARCHAR(35) NOT NULL,
 [Auteur] VARCHAR(45),
 [Edition] VARCHAR(20),
 [ISBN] VARCHAR(20)
);
```

```
INSERT INTO [LIVRES]
([NumLivre],[Titre],[Auteur],[Edition],[ISBN])
VALUES (1,'De Merise aux Bases de Données','Banos - Mouyssinat','Eyrolles',NULL)
```

	NumLivre	Titre	Auteur	Edition	ISBN
1	1	De Merise aux Bases de Données	Banos - Mouyssinat	Eyrolles	NULL
2	2	MySQL	Fabrice Métayer	Micro-Application	2-7429-2109-5
3	3	Grand Livre JAVA 2	Maurers, Baufels, Friedrich	Micro-Application	2-7429-1498-6
4	4	Programmation JAVA 2	Niedermair	Micro-Application	2-7429-1828-0
5	5	Maîtrise des algorithmes en C	Kyle Loudon	O'REILLY	2-84177-096-6
6	6	L'Essentiel sur Merise	Dominique Dionisi	Eyrolles	2-212-09046-3
7	7	Programmer en langage C	Claude Delannoy	Eyrolles	2-212-08985-6
8	8	Le livre du C	Claude Delannoy	Eyrolles	2-212-08838-8
9	9	Architecture de l'Ordinateur	Andrew Tanenbaum	InterEditions	2-7296-0641-6
10	10	Apprenez le Langage C en 21...	Peter Aitken	S & S Macmillan	2-7440-0351-4

```
CREATE TABLE CLILIV
([NumCliLiv] INTEGER NOT NULL PRIMARY KEY,
 [RefClient] INTEGER NOT NULL FOREIGN KEY REFERENCES [CLIENTS]([NumClient]),
 [RefLivre] INTEGER NOT NULL FOREIGN KEY REFERENCES [LIVRES] ([NumLivre]),
 [DateEmprunt] DATETIME NOT NULL,
 [DateRetour] DATETIME
);
```

Nous ajoutons deux contraintes à cette table : les emprunts ne sont effectués que depuis le 01/09/2001 et les dates de retour sont au moins postérieures ou égales aux dates d'emprunt. Ceci donne :

```
ALTER TABLE CLILIV
ADD CONSTRAINT con_DateEmpruntISupAout2001
CHECK ([DateEmprunt] > '31/08/2001');
ALTER TABLE CLILIV
ADD CONSTRAINT con_DateEmpruntInfDateRetour
CHECK ([DateEmprunt] <= [DateRetour]);
```

On peut en faire une seule

```
INSERT INTO CLILIV
([NumCliLiv],[RefClient],[RefLivre],[DateEmprunt],[DateRetour])
VALUES (4,5,8,'01/01/2002',NULL)
```

	NumCliLiv	RefClient	RefLivre	DateEmprunt	DateRetour
1	1	3	7	2001-10-01 00:00:00.000	2001-10-12 00:00:00.000
2	2	1	7	2001-10-21 00:00:00.000	2001-10-22 00:00:00.000
3	3	8	2	2001-10-21 00:00:00.000	NULL
4	4	5	8	2002-01-01 00:00:00.000	NULL
5	5	9	9	2002-01-01 12:14:28.000	NULL

Conservez ces tables. Elles nous serviront plus tard.

6.1. Depuis la dernière version de la base de données LOCLIVRE, listez

- les livres qui sont en prêt actuellement (avec les titres, ...)
- qui a emprunté quoi
- qui a emprunté quoi et quand
- les clients qui n'ont jamais rien emprunté
- les livres qui n'ont jamais été prêtés
- les clients avec les livres éventuels empruntés
- les livres avec leurs emprunteurs éventuels.

Je vous conseille de d'abord encoder une série de nouveaux enregistrements dans les trois tables.

les livres qui sont en prêt actuellement (avec les titres, ...)

```

SELECT l.[titre], l.[auteur], c.[nom],c.[prénom], cl.DateEmprunt
FROM CLILIV AS cl INNER JOIN CLIENTS AS c
ON c.numclient = cl.refclient
INNER JOIN LIVRES AS L
ON l.numlivre = cl.reflivre
WHERE cl.dateretour IS NULL;

```

	titre	auteur	nom	prénom	DateEmprunt
1	MySQL	Fabrice Métayer	Covillard	Henry	2001-10-21 ...
2	Le livre du C	Claude Delannoy	Davreux	Pol	2002-01-01 ...
3	Architecture de l'Ordinateur	Andrew Tanenbaum	Pierret	Jacques	2002-01-01 ...

Lot de requêtes terminé. ERIC\MAITRISERSQL (8.0) ERIC\Eric Brasseur (51) LOCLIVRE 0:00:00 3 lignes Lg 1, Col 1

qui a emprunté quoi

```

SELECT l.[titre], l.[auteur], c.[nom],c.[prénom]
FROM CLILIV AS cl INNER JOIN CLIENTS AS c
ON c.numclient = cl.refclient
INNER JOIN LIVRES AS L
ON l.numlivre = cl.reflivre ;

```

	titre	auteur	nom	prénom
1	Programmer en langage C	Claude Delannoy	Barzin	Frédéric
2	Programmer en langage C	Claude Delannoy	Brasseur	Eric
3	MySQL	Fabrice Métayer	Covillard	Henry
4	Le livre du C	Claude Delannoy	Davreux	Pol
5	Architecture de l'Ordinateur	Andrew Tanenbaum	Pierret	Jacques

Lot de req ERIC\MAITRISERSQL (8.0) ERIC\Eric Brasseur (51) LOCLIVRE 0:00:00 5 lignes Lg 1, Col 49

qui a emprunté quoi et quand

Requête - ERIC\MAITRISERSQL.LOCLIVRE.ERIC\Eric Brasseur - Sans titre1*

```

SELECT l.[titre], l.[auteur], c.[nom],c.[prénom], cl.dateemprunt, cl.dateretour
FROM CLILIV AS cl INNER JOIN CLIENTS AS c
ON c.numclient = cl.refclient
INNER JOIN LIVRES AS L
ON l.numlivre = cl.reflivre ;

```

	titre	auteur	nom	prénom	dateemprunt	dateretour
1	Programmer en langage C	Claude Delannoy	Barzin	Frédéric	2001-10-01 ...	2001-10-12 ...
2	Programmer en langage C	Claude Delannoy	Brasseur	Eric	2001-10-21 ...	2001-10-22 ...
3	MySQL	Fabrice Métayer	Covillard	Henry	2001-10-21 ...	NULL
4	Le livre du C	Claude Delannoy	Davreux	Pol	2002-01-01 ...	NULL
5	Architecture de l'Ordinateur	Andrew Tanenbaum	Pierret	Jacques	2002-01-01 ...	NULL

Lot de requêtes terminé. ERIC\MAITRISERSQL (8.0) ERIC\Eric Brasseur (51) LOCLIVRE 0:00:00 5 lignes Lg 1, Col 74

les clients qui n'ont jamais rien emprunté

Requête - ERIC\MAITRISERSQL.LOCLIVRE.ERIC\Eric Br...

```

SELECT c.[nom],c.[prénom]
FROM CLIENTS AS c LEFT JOIN CLILIV AS cl
ON c.numclient = cl.refclient
WHERE cl.[refclient] IS NULL;

```

	nom	prénom
1	Borremans	Cristiane
2	Dupont	Pierre
3	Castoin	Yves
4	Dubois	Marc
5	Panis	Serge

ERIC\Eric Brasseur (51) LOCLIVRE 0:00:00 5 lignes Lg 1, Col 26

les livres qui n'ont jamais été prêtés

Requête - ERIC\MAITRISERSQL.LOCLIVRE.ERIC\Eric Brasseur - Sans titre1*

```

SELECT l.[titre],l.[auteur]
FROM LIVRES AS l LEFT JOIN CLILIV AS cl
ON l.numlivre = cl.reflivre
WHERE cl.[reflivre] IS NULL;

```

	titre	auteur
1	De Merise aux Bases de Données	Banos - Mouyssinat
2	Grand Livre JAVA 2	Maurers, Baufels, Friedrich
3	Programmation JAVA 2	Niedermair
4	Maîtrise des algorithmes en C	Kyle Loudon
5	L'Essentiel sur Merise	Dominique Dionisi
6	Apprenez le Langage C en 21...	Peter Aitken

Lot de | ERIC\MAITRISERSQL (8.0) ERIC\Eric Brasseur (51) LOCLIVRE 0:00:00 6 lignes Lg 4, Col 1

les clients avec les livres éventuels empruntés

Requête - ERIC\MAITRISERSQL.LOCLIVRE.ERIC\Eric Brasseur - Sans titre1*

```

SELECT c.[nom],c.[prénom],l.[titre],l.[auteur]
FROM LIVRES AS l RIGHT JOIN CLILIV AS cl
ON l.numlivre = cl.reflivre
RIGHT JOIN CLIENTS AS C
ON cl.[refclient] = c.[numclient];

```

	nom	prénom	titre	auteur
1	Brasseur	Eric	Programmer en langage C	Claude Delannoy
2	Borremans	Cristiane	NULL	NULL
3	Barzin	Frédéric	Programmer en langage C	Claude Delannoy
4	Dupont	Pierre	NULL	NULL
5	Davreux	Pol	Le livre du C	Claude Delannoy
6	Castoin	Yves	NULL	NULL
7	Dubois	Marc	NULL	NULL
8	Covillard	Henry	MySQL	Fabrice Métayer
9	Pierret	Jacques	Architecture de l'Ordinateur	Andrew Tanenbaum
10	Panis	Serge	NULL	NULL

Grilles Messages

Lot de requêtes terminé. ERIC\MAITRISERSQL (8.0) ERIC\Eric Brasseur (51) LOCLIVRE 0:00:00 10 lignes Lg 1, Col 8

les livres avec leurs emprunteurs éventuels.

Requête - ERIC\MAITRISERSQL.LOCLIVRE.ERIC\Eric Brasseur - Sans titre1*

```

SELECT l.[titre],l.[auteur], c.[nom],c.[prénom]
FROM LIVRES AS l LEFT JOIN CLILIV AS cl
ON l.numlivre = cl.reflivre
LEFT JOIN CLIENTS AS C
ON cl.[refclient] = c.[numclient];

```

	titre	auteur	nom	prénom
1	De Merise aux Bases de Données	Banos - Mouyssinat	NULL	NULL
2	MySQL	Fabrice Métayer	Covillard	Henry
3	Grand Livre JAVA 2	Maurers, Baufels, Friedrich	NULL	NULL
4	Programmation JAVA 2	Niedermair	NULL	NULL
5	Maîtrise des algorithmes en C	Kyle Loudon	NULL	NULL
6	L'Essentiel sur Merise	Dominique Dionisi	NULL	NULL
7	Programmer en langage C	Claude Delannoy	Barzin	Frédéric
8	Programmer en langage C	Claude Delannoy	Brasseur	Eric
9	Le livre du C	Claude Delannoy	Davreux	Pol
10	Architecture de l'Ordinateur	Andrew Tanenbaum	Pierret	Jacques
11	Apprenez le Langage C en 21...	Peter Aitken	NULL	NULL

Grilles Messages

Lot de requêtes terminé. ERIC\MAITRISERSQL (8.0) ERIC\Eric Brasseur (51) LOCLIVRE 0:00:00 11 lignes Lg 4, Col 10

9.1. Affichez le décompte des commandes mensuelles. (en utilisant si possible le nom du mois et pas son numéro).

Requête - ERIC\MAITRISERSQL.comptoir.ERIC\Eric Brasseur - Sans titre1*

```
SELECT DATENAME(MONTH,[Date Commande]) AS [Commande du mois],
COUNT([N° Commande]) AS [Nombre de Commandes]
FROM Commandes GROUP BY DATENAME(MONTH,[Date Commande])
ORDER BY MAX([Date Commande]);
```

	Commande du mois	Nombre de Commandes
1	juin	30
2	juillet	55
3	août	58
4	septembre	60
5	octobre	64
6	novembre	59
7	décembre	79
8	janvier	88
9	février	83
10	mars	103
11	avril	105
12	mai	46

Grilles Messages

Lot de ERIC\MAITRISERSQL (8.0) ERIC\Eric Brasseur (51) comptoir 0:00:00 12 lignes Lg 4, Col 30

9.2. Même résultat mais avec les années en plus.

Requête - ERIC\MAITRISERSQL.comptoir.ERIC\Eric Brasseur - Sans titre1*

```
SELECT DATENAME(MONTH,[Date Commande]) AS [Commande du mois],
DATENAME(YEAR,[Date Commande]) AS [Année du mois],
COUNT([N° Commande]) AS [Nombre de Commandes]
FROM Commandes
GROUP BY DATENAME(YEAR,[Date Commande]), DATENAME(MONTH,[Date Commande])
ORDER BY MAX([Date Commande]);
```

	Commande du mois	Année du mois	Nombre de Commandes
1	juillet	1996	22
2	août	1996	25
3	septembre	1996	23
4	octobre	1996	26
5	novembre	1996	25
6	décembre	1996	31
7	janvier	1997	33
8	février	1997	29
9	mars	1997	30

Grilles Messages

Lot de requêtes terminé. ERIC\MAITRISERSQL (8.0) ERIC\Eric Brasseur (51) comptoir 0:00:00 23 lignes Lg 6, Col 28

9.3. Nous voulons un état détaillé du nombre de chacun des produits vendus par chaque employé chaque mois, indiquant le montant total des ventes pour chaque produit mensuellement.

Les tables à utiliser sont :

- Employés
- Commandes
- Détails Commandes
- Produits.

Commençons par la jointure permettant de relier les tables :

The screenshot shows a SQL query window with the following query:

```
SELECT Employés.[N° Employé], Employés.nom, Employés.Prénom,
Produits.[Réf produit], produits.[Nom du produit]
FROM Produits INNER JOIN(
(Employés INNER JOIN Commandes ON Employés.[N° Employé] = Commandes.[N° Employé])
INNER JOIN [Détails Commandes] ON Commandes.[N° Commande] = [Détails Commandes].[N° Commande]) ON
Produits.[Réf Produit] = [Détails Commandes].[Réf Produit];
```

The resulting table is as follows:

	N° Employé	nom	Prénom	Réf produit	Nom du produit
1	5	Buchanan	Steven	11	Queso Cabrales
2	5	Buchanan	Steven	42	Singaporean Hokkien Fried Mee
3	5	Buchanan	Steven	72	Mozzarella di Giovanni
4	6	Suyama	Michael	14	Tofu
5	6	Suyama	Michael	51	Manjimup Dried Apples
6	4	Peacock	Margaret	41	Jack's New England Clam Chowder
7	4	Peacock	Margaret	51	Manjimup Dried Apples
8	4	Peacock	Margaret	65	Louisiana Fiery Hot Pepper ...
9	3	Leverling	Janet	22	Gustaf's Knäckebröd
10	3	Leverling	Janet	57	Ravioli Angelo
11	3	Leverling	Janet	65	Louisiana Fiery Hot Pepper ...
12	4	Peacock	Margaret	20	Sir Rodney's Marmalade
13	4	Peacock	Margaret	33	Geitost
14	4	Peacock	Margaret	60	Camembert Pierrot
15	3	Leverling	Janet	31	Gorgonzola Telino
16	3	Leverling	Janet	39	Chartreuse verte

La partie ombragée ne devra pas être changée. La partie supérieure nous permet juste de contrôler si nous avons bien construit notre jointure.

Il faut maintenant déterminer les champs qui devront être affichés et les champs qui vont en dériver d'autres.

- le nom et le prénom de chaque employé (concaténés pour gagner de la place) + son numéro
- le nom du produit et sa référence
- les mois de commandes (avec les années concaténés)
- le nombre de ventes
- le montant de ces ventes

Comment concaténer les dates mois et années ? (en affichant le nom du mois et pas son numéro)

DATENAME(MONTH, Commandes.[Date Commande]) + ' / ' + DATENAME(YEAR, Commandes.[Date Commande]) que nous grouperons en premier lieu

Le n° employé sera groupé en deuxième position

La concaténation des noms et prénoms :

MAX(Employés.Prénom + ' ' + Employés.Nom) (pourquoi MAX ?)

La référence du produit formera le troisième groupement

Mon du produit

MAX(Produits.[Nom du Produit])

Nombres de ventes :

COUNT([Détails Commandes].[Réf Produit])

Montant :

SUM([Détails Commandes].[Prix unitaire] * [détails commandes].Quantité)

```

SELECT DATENAME(MONTH, Commandes.[Date Commande]) + ' / ' +
DATENAME(YEAR, Commandes.[Date Commande]) AS [Com. Mois],
Employés.[N° Employé] AS [n°emp],
MAX(Employés.Prénom + ' ' + Employés.Nom) AS [Nom employé],
Produits.[Réf Produit] AS [N°Prod],
MAX(Produits.[Nom du Produit]) AS [Nom du Produit],
COUNT([Détails Commandes].[Réf Produit]) AS nbventes,
SUM([Détails Commandes].[Prix unitaire] * [détails commandes].Quantité) AS Montant
FROM Produits INNER JOIN (
(Employés INNER JOIN Commandes ON Employés.[N° Employé] = Commandes.[N° Employé])
INNER JOIN [Détails Commandes] ON Commandes.[N° Commande] = [Détails Commandes].[N° Commande]) ON
Produits.[Réf Produit] = [Détails Commandes].[Réf Produit]
GROUP BY Commandes.[Date Commande], Employés.[N° Employé], Produits.[Réf Produit];

```

	Com. Mois	n°emp	Nom employé	N°Prod	Nom du Produit	nbventes	Montant
1	juillet / 1996	5	Steven Buchanan	11	Queso Cabrales	1	840.0000
2	juillet / 1996	5	Steven Buchanan	42	Singaporean Hokkien Fried Mee	1	490.0000
3	juillet / 1996	5	Steven Buchanan	72	Mozzarella di Giovanni	1	870.0000
4	juillet / 1996	6	Michael Suyama	14	Tofu	1	837.0000
5	juillet / 1996	6	Michael Suyama	51	Manjimup Dried Apples	1	8480.0000
6	juillet / 1996	3	Janet Leverling	22	Gustaf's Knäckebröd	1	504.0000
7	juillet / 1996	3	Janet Leverling	57	Ravioli Angelo	1	1170.0000
8	juillet / 1996	3	Janet Leverling	65	Louisiana Fiery Hot Pepper ...	1	1680.0000
9	juillet / 1996	4	Margaret Peacock	41	Jack's New England Clam Chowder	1	385.0000
10	juillet / 1996	4	Margaret Peacock	51	Manjimup Dried Apples	1	7420.0000
11	juillet / 1996	4	Margaret Peacock	65	Louisiana Fiery Hot Pepper ...	1	1260.0000
12	juillet / 1996	4	Margaret Peacock	20	Sir Rodney's Marmalade	1	12960.0000
13	juillet / 1996	4	Margaret Peacock	33	Geitost	1	250.0000
14	juillet / 1996	4	Margaret Peacock	60	Camembert Pierrot	1	5440.0000
15	juillet / 1996	3	Janet Leverling	31	Corsica's Tofino	1	1000.0000

10.1. Créons une procédure stockée et testons la : elle doit permettre l'affichage du code client, le nom de la société, la région à partir de la table clients. (sp_ListeMondialeClient)

```

CREATE PROCEDURE sp_ListeMondialeClient
AS
SELECT [Code Client], Société, Région
FROM Clients;

```

	Code Client	Société	Région
1	ALFKI	Alfreds Futterkiste	NULL
2	ANATR	Ana Trujillo Emparedados y ...	NULL
3	ANTON	Antonio Moreno Taquería	NULL
4	AROUT	Around the Horn	NULL
5	BERGS	Berglunds snabbköp	NULL
6	BLAUS	Blauer See Delikatessen	NULL

10.2. Même exercice, mais nous voulons que cette procédure nous permette d'entrer la région et uniquement pour les USA et le Canada.

```
CREATE PROCEDURE sp_ListeMondialeClientNA
@Région nvarchar(15)
AS
SELECT [Code Client], Société, Région
FROM Clients
WHERE Pays IN ('États-Unis','Canada') AND Région = @Région;
```

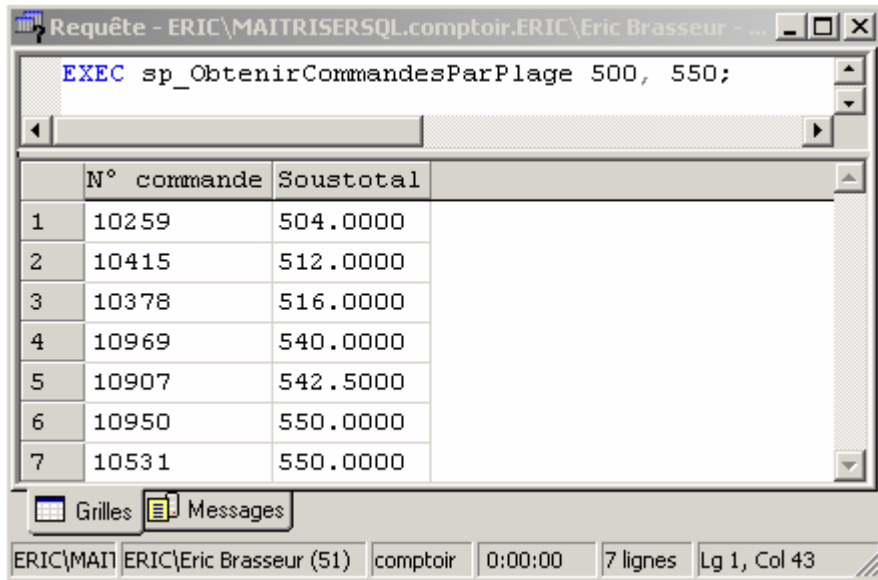
	Code Client	Société	Région
1	LAZYK	Lazy K Kountry Store	WA
2	TRAIH	Trail's Head Gourmet Provis...	WA
3	WHITC	White Clover Markets	WA

10.3. Nous voulons afficher le numéro de commande et le montant à partir de [Sous-totaux commandes] en filtrant les informations : commandes entre @petitmontant et @grandmontant. Ces paramètres sont optionnels (leur valeur par défaut est NULL). Si NULL est envoyé à la procédure, la procédure remplace la valeur de @grandmontant par la plus grande de la série et @petitmontant par la plus petite de la série. Nous voulons un affichage ordonné par sous-total.

```
CREATE PROCEDURE sp_ObtenerCommandesParPlage
@petitmontant decimal(19,4) = NULL,
@grandmontant decimal(19,4) = null
AS
IF @petitmontant IS NULL
BEGIN
SELECT @petitmontant = MIN(SousTotal) FROM [Sous-totaux commandes]
END
IF @grandmontant IS NULL
```

```
BEGIN
  SELECT @grandmontant = MAX(SousTotal) FROM [Sous-totaux commandes]
END
```

```
SELECT [N° commande], Soustotal FROM [Sous-Totaux commandes]
WHERE SousTotal BETWEEN @petitmontant AND @grandmontant
ORDER BY soustotal;
```

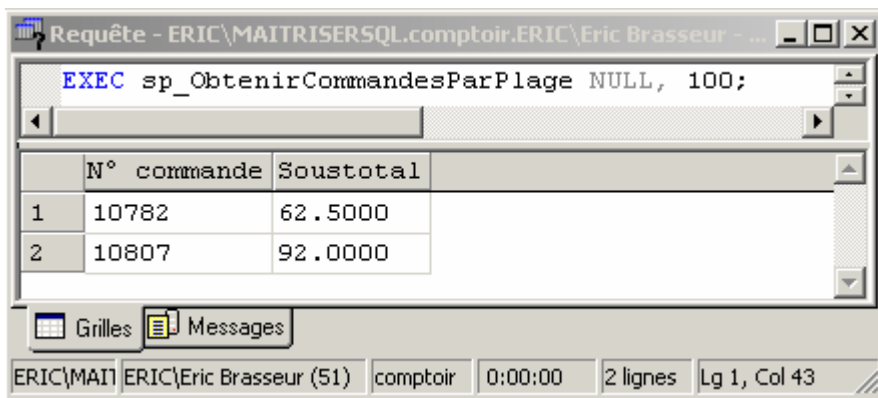


Requête - ERIC\MAITRISERSQL.comptoir.ERIC\Eric Brasseur - ...

```
EXEC sp_ObttenirCommandesParPlage 500, 550;
```

	N° commande	Soustotal
1	10259	504.0000
2	10415	512.0000
3	10378	516.0000
4	10969	540.0000
5	10907	542.5000
6	10950	550.0000
7	10531	550.0000

ERIC\MAI ERIC\Eric Brasseur (51) comptoir 0:00:00 7 lignes Lg 1, Col 43



Requête - ERIC\MAITRISERSQL.comptoir.ERIC\Eric Brasseur - ...

```
EXEC sp_ObttenirCommandesParPlage NULL, 100;
```

	N° commande	Soustotal
1	10782	62.5000
2	10807	92.0000

ERIC\MAI ERIC\Eric Brasseur (51) comptoir 0:00:00 2 lignes Lg 1, Col 43

10.4. Ecrivons une procédure stockée qui permet de mettre à jour un client. S'il existe (voir son code), on procède au remplacement autrement on ajoute ce nouveau client. Pour simplifier, on se contentera de son code et de son nom de société.

```
CREATE PROCEDURE sp_ActualiserDonnéesClients
  @codeclient char(5),
  @Société varchar(40)
AS
IF EXISTS ( SELECT [Code Client] FROM Clients WHERE [Code Client] = @codeclient )
  BEGIN -- actualisation du client
    UPDATE clients
      SET Société = @Société FROM Clients WHERE [Code client] = @codeclient
  END
ELSE
```

```
BEGIN -- ici, le client n'existe pas
INSERT INTO Clients ([Code client], société)
VALUES (@codeclient, @société)
END
```

EXEC sp_ActualiserDonnéesClients 'TROTT', 'Trotter inc'; (ici , c'est un ajout)

EXEC sp_ActualiserDonnéesClients 'BLAUS', 'Claus & CO'; (ici , c'est une actualisation).

11.1. Créez une vue qui nous contenant la liste des produits dont le prix se situe entre 50 et 75 . Essayez d'ajouter des données (valides ou non valides) ou de modifier les données dans cette vue.

```
CREATE VIEW [vw_ListeFiltréeProduits] AS
SELECT [Réf Produit], [Nom du Produit],[Prix Unitaire]
FROM Produits
WHERE [Prix unitaire] BETWEEN 50 AND 75
WITH CHECK OPTION
```

	Réf Produit	Nom du Produit	Prix Unitaire	
1	3	Aniseed Syrup	52.5000	
2	21	Sir Rodney's Scones	52.5000	
3	25	NuNuCa Nuß-Nougat-Creme	73.5000	
4	31	Gorgonzola Telino	65.6250	
5	34	Sasquatch Ale	73.5000	
6	41	Jack's New England Clam Chowder	50.6625	
7	42	Singaporean Hokkien Fried Mee	73.5000	
8	46	Spegesild	63.0000	
9	48	Chocolade	66.9375	
10	58	Escargots de Bourgogne	69.5625	
11	67	Laughing Lumberjack Lager	73.5000	
12	68	Scottish Longbreads	65.6250	
13	74	Longlife Tofu	52.5000	
14	77	Original Frankfurter grüne Soße	68.2500	

```
UPDATE [vw_ListeFiltréeProduits]
SET [Prix Unitaire] = 100
WHERE [Réf Produit] = 41
```

```
UPDATE [vw_ListeFiltréeProduits]
SET [Prix Unitaire] = 60
WHERE [Réf Produit] = 42
```

```
INSERT INTO [vw_ListeFiltréeProduits]
VALUES (140, 'Baisers de Rochefort', 64)
```

Vous constaterez qu'il refuse de l'ajouter dans Produits . Vérifiez dans la table des produits le résultat de ces requêtes.

11.2. Créez une vue pour afficher les noms des clients, le numéro de commande et la date d'une commande donnée ainsi que l'employé responsable du suivi de la commande.

```
CREATE VIEW vw_InfoSupplCommande AS
SELECT Commandes.[N° Commande], Clients.[Société], Commandes.[Date Commande],
       [Employés].nom + ', ' + [Employés].[Prénom] AS [Employé]
FROM Commandes INNER JOIN Clients ON Commandes.[Code Client] = Clients.[Code Client]
       INNER JOIN [Employés] ON Commandes.[N° Employé] = [Employés].[N° Employé]
```

	N° Commande	Société	Date Commande	Employé
1	10248	Vins et alcools Chevalier	1996-07-04 00:00:00	Buchanan, Steven
2	10249	Toms Spezialitäten	1996-07-05 00:00:00	Suyama, Michael
3	10250	Hanari Carnes	1996-07-08 00:00:00	Peacock, Margaret
4	10251	Victuailles en stock	1996-07-08 00:00:00	Leverling, Janet
5	10252	Suprêmes délices	1996-07-09 00:00:00	Peacock, Margaret
6	10253	Hanari Carnes	1996-07-10 00:00:00	Leverling, Janet
7	10254	Chop-suey Chinese	1996-07-11 00:00:00	Buchanan, Steven
8	10255	Richter Supermarkt	1996-07-12 00:00:00	Dodsworth, Anne
9	10256	Wellington Importadora	1996-07-15 00:00:00	Leverling, Janet
10	10257	HILARIÓN-Abastos	1996-07-16 00:00:00	Peacock, Margaret
11	10258	Ernst Handel	1996-07-17 00:00:00	Davolio, Nancy
12	10259	Centro comercial Moctezuma	1996-07-18 00:00:00	Peacock, Margaret
13	10260	Ottilies Käseladen	1996-07-19 00:00:00	Peacock, Margaret
14	10261	Que Delícia	1996-07-19 00:00:00	Peacock, Margaret
15	10262	Rattlesnake Canyon Grocery	1996-07-22 00:00:00	Callahan, Laura
16	10263	Ernst Handel	1996-07-23 00:00:00	Dodsworth, Anne
17	10264	Folk och få HB	1996-07-24 00:00:00	Suyama, Michael
18	10265	Blondel père et fils	1996-07-25 00:00:00	Fuller, Andrew
19	10266	Wartian Herkku	1996-07-26 00:00:00	Leverling, Janet