

Laboratoires SQL

Par Dedeurwaerder steve

Pré requis avant la consultation du document.

Les différentes requêtes n'ayant pas été corrigées, des erreurs peuvent s'y trouver. Certaines requêtes n'ont pas été traitées, merci d'y apporter les corrections nécessaires,

Vous pouvez m'envoyer vos suggestions à l'adresse suivante :

blutch17@hotmail.com

Je mettrai également les mises à jour en ligne si ils y en a ...

Mise à jour : le 9 / 01 / 07

Laboratoire SQL n°1 : requêtes de sélection

1. Installer SQL server et le faire fonctionner correctement
2. Créer une base de données relationnelle permettant de gérer une société en utilisant SQL Server.

La table clients fait référence aux clients, elle comprend les champs :

1. ncli : numéro de référence du client. Type texte, clé primaire.
2. nom : nom du client . Type texte (20).
3. adresse : adresse client. Type texte.
4. localite : ville client. Type texte.
5. compte : solde compte client. Type monétaire.

La table commandes fait référence aux commandes du client, elle comprend les champs :

1. ncom : numéro de la commande. Type numérique, clé primaire.
2. datecom : date de la commande. Type date.

La table produits fait référence aux produits, elle comprend les champs :

1. npro : numéro du produit. Type texte, clé primaire.
2. libelle: libelle du produit. Type texte.
3. prix : prix du produit. Type monétaire.
4. qstock: quantité en stock. Type numérique.

Adapter la structure de la base de données pour qu'elle réponde aux demandes suivantes :

- un client peut faire plusieurs commandes
- un produit peut être repris dans plusieurs commandes
- une commande peut englober plusieurs produits
- Pour chaque détail de commande, mettre un champ qcom (correspondant à la quantité demandée)

3. Ajouter des enregistrements à la base de données
4. Exprimer en langage SQL les requêtes suivantes, sur papier puis sur machine, relatives à la base de données societe
 - a) Afficher toutes les caractéristiques de tous les produits
`SELECT * FROM produits`
 - b) Afficher la localité des clients (sans doublons)
Utiliser `distinct` : mais ne fonctionne pas avec du texte
`select distinct localite from clients`
 - c) Afficher le numéro, le nom et la localité des clients habitant à Namur et à Poitiers
`SELECT ncli,nom,localite FROM clients where localite='namur' or localite='poitiers'`
 - d) Afficher le numéro des clients dont le compte est compris entre 0 et 50 euros.
`SELECT ncli FROM clients where compte between 0 and 50`
 - e) Afficher les caractéristiques des produits en hêtre.
`SELECT * FROM produits where libelle like 'hetre'`

f) Afficher la valeur totale des stocks (quantité totale des produits en stock)

```
SELECT sum(qstock)somme FROM produits
```

g) Afficher le total, le minimum, la moyenne et le maximum des comptes clients

```
SELECT sum(compte),min(compte),avg(compte),max(compte) from clients
```

h) Afficher le total et la moyenne des comptes clients, ainsi que le nombre de clients :

1. par catégorie

2. par localité

```
select sum(compte)total, avg(compte)moyenne, localite,count(*) from clients group by localite
```

3. par catégorie dans chaque localité

```
select sum(compte)total, avg(compte)moyenne, localite,count(*) from clients where ncli like '_%'group by localite
```

i) Afficher le total des comptes clients par localité, s'il est supérieur à -20 euros, par ordre décroissant du total

```
select localite,compte from clients where compte > -20 order by compte desc  
select sum(compte)total,localite from clients group by localite having sum(compte) > -20 order by sum(compte) desc
```

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

```
select nom,ncli,localite from clients order by nom,localite  
select ncli,nom,localite from clients where nom < localite
```

k) Afficher 50 % de toutes les caractéristiques des produits dont la quantité en stock est supérieure à 20

```
SELECT TOP 50 PERCENT * FROM produits where qstock>20
```

l) Afficher les 3 produits dont la quantité est la plus grande

```
select top 3 qstock from produits order by qstock desc
```

Laboratoire SQL n°2 : Jointures

Exprimez en langage SQL les requêtes suivantes, sur papier puis sur machine, relatives à la base de données société

1. Afficher le produit cartésien des tables produits et commandes

```
select * from produits,commandes
```

2. Afficher le numéro et le nom des clients qui ont passé une commande (avec doublons)

```
select * from clients,commandes where clients.ncli=commandes.ncli (affiche tout)
```

```
select nom,clients.ncli from clients,commandes where clients.ncli=commandes.ncli
```

3. Afficher le numéro et le nom des clients qui ont passé une commande (sans doublons)

```
select distinct nom,clients.ncli from clients,commandes where clients.ncli=commandes.ncli
```

autre façon :

```
select distinct nom,clients.ncli from clients inner join commandes on clients.ncli=commandes.ncli
```

4. Afficher le numéro des clients qui commandent le produit de numéro PH222

```
select clients.ncli from clients inner join commandes on commandes.ncli=clients.ncli inner join detail on commandes.ncom=detail.ncom where detail.npro='ph222'
```

5. Afficher le montant dû pour chaque commande

```
select detail.ncom,(detail.qcom*produits.prix)total_commande from detail inner join produits on detail.npro=produits.npro
```

6. Afficher un tableau qui donne pour chaque ligne de commande la date de la commande et le libelle du produit

```
select commandes.ncom,commandes.datecom,produits.libelle from detail inner join commandes on detail.ncom=commandes.ncom inner join produits on detail.npro=produits.npro
```

7. Afficher un tableau qui donne pour chaque commande le nom, le numéro du client, le numéro, la date et le montant total de la commande

```
Select commandes.ncom,nom ,commandes.ncli, commandes.datecom, (detail.qcom*produits.prix) total_commande from clients inner join commandes on commandes.ncli=clients.ncli inner join detail on commandes.ncom=detail.ncom inner join produits on produits.npro=detail.npro
```

8. Afficher tous les clients et leurs commandes même si le client n'a pas fait de commande (donner 2 façons d'exprimer la requête)

```
select nom,ncom from commandes inner join clients on commandes.ncli=clients.ncli order by nom
```

~~ou~~

```
select nom,ncom from commandes,clients where commandes.ncli=clients.ncli order by nom
```

```
select nom,ncom from commandes right join clients on commandes.ncli=clients.ncli
```

```
ou  
select nom,ncom from clients left join commandes on commandes.ncli=clients.ncli
```

9. Afficher les commandes dont le nom de produit contient hêtre

```
select detail.ncom,libelle from produits inner join detail on produits.npro=detail.npro where libelle like '%hêtre%'
```

10. Afficher le nom du produit et la quantité commandée

```
select libelle,sum (detail.qcom)quantité_commandé from produits inner join detail on produits.npro=detail.npro group by libelle
```

Laboratoire SQL N°3 : Sous-requêtes

1. Afficher le numéro et le libellé des produits qui ne sont pas commandés.

```
select npro,libelle from produits where npro not in(select npro from detail)
```

2. Afficher le numéro et le libellé des produits qui ont été commandés.

```
select npro,libelle from produits where npro in(select npro from detail)
```

3. Afficher le nom et le numéro des clients qui n'ont pas effectué de commande.

```
select ncli,nom from clients where ncli not in(select ncli from commandes)
```

4. Afficher le nom et le numéro des clients qui ont effectué de commande.

```
select ncli,nom from clients where ncli in(select ncli from commandes)
```

5. Afficher tous les clients qui ont effectué plus d'une commande.

```
select * from clients where (select count(*)from commandes where commandes.ncli=clients.ncli)>1
```

6. Afficher tous les produits dont le numéro de produit n'est repris dans aucune commande. (de 2 façons différentes)

```
select npro from produits where npro not in (select npro from detail)
```

ou

7. Afficher toutes les commandes où la date de commande est supérieure à toutes les commandes dont le numéro de client commence par B.

```
select * from commandes where datecom >all(select datecom from commandes inner join clients on commandes.ncli=clients.ncli where clients.ncli like'b%')
```

8. Afficher toutes les commandes où la date de commande est supérieure à n'importe laquelle des commandes dont le numéro de client commence par B.

```
select * from commandes where datecom >any(select datecom from commandes inner join clients on commandes.ncli=clients.ncli where clients.ncli like'b%')
```

9. Afficher par date de commande la somme des quantités commandées qui sont < à la moyenne des quantités commandées.

```
select datecom,sum(qcom)from commandes inner join detail on commandes.ncom=detail.ncom where qcom <(select avg(qcom)from detail)group by datecom
```

Laboratoire SQL n°4: requêtes de mise à jour des données

Créer une base de données vide societeb et exporter les données de societe vers societeb.

[Clicdroit/ nouvelle tache / export base de donnée](#)

Travailler sur la base de données societeb.

Réaliser sur papier, puis sur machine les requêtes suivantes :

1. Supprimer dans la table clients tous les enregistrements dont la localité commence par T.

```
delete from clients where localite like 'T%'
```

2. Modifier dans la table produit les quantités en stock qui valent 0, leur ajouter 5.

```
update produits set qstock =qstock + 5 where qstock=0
```

3. Ajouter à la table clients le numéro de client dont le numéro de commande est compris entre 30172 et 30174.

```
insert into clients2(ncli) select clients.ncli from clients inner join commandes on clients.ncli=commandes.ncli where ncom between 30172 and 30174
```

4. Ajouter dans la table produits, l'enregistrement suivant npro=PA50, libelle='pointe acier 50', prix=20 et qstock=100.

```
insert into produits(npro,libelle,prix,qstock) VALUES ('PA50','pointe acier 50',20,100)
```

Laboratoire SQL n°5 : Création de tables

1. Ajouter une table fournisseur avec les attributs suivants :
 - a. Nfour de type nvarchar de 50 caractères , clé primaire
 - b. Nom de type chaîne de 20 caractères, null interdit
 - c. Type de type chaîne de caractères avec comme valeur par défaut bois, doit être unique
 - d. CodePostal de type numérique sur 4 chiffres, doit être compris entre 1000 et 9999
 - e. Elle a comme clé étrangère npro de la table produit

```
create table fournisseur(Nfour nvarchar (50) Primary key, Nom char(20)NOT NULL, Type char (20)default'bois' unique,CodePostal NUMERIC (4)check (CodePostal between 1000 and 9999),npro nvarchar (50) FOREIGN KEY (npro) references produits(npro))
```

2. Entrer des enregistrements dans la table fournisseur et vérifier que tout se passe comme demandé en 1.
3. Supprimer la contrainte nom_unique de la table fournisseur

```
Alter table fournisseur drop constraint nom_unique
```

4. Mettre la contrainte nom non null dans la table fournisseur

```
Alter TABLE fournisseur ADD constraint nom check (nom is NOT NULL)
```

5. Ajouter la colonne localité de type chaîne de 20 caractères dans la table fournisseur.

```
alter table fournisseur add localite char(20)
```

6. Entrer un nouvel enregistrement pour tester la modification du point 3.

```
update fournisseur set localite=('tournai')
```

7. Quelle est la commande que vous utilisez pour supprimer la table fournisseur ?

```
drop table fournisseur
```

Laboratoire SQL n°6 : Gestion de bases de données

En utilisant le langage sql, effectuer les requêtes sql suivantes sur papier puis sur machine.

1. Créer une base de données ayant les caractéristiques suivantes :

	Nom	Nom fichier	Taille	Taille Maximale	Incrément
Base de données	helho_data	Helho.mdf	2Mo	15Mo	1%
Journal	Helho_log	Helho.ldf	1Mo	5Mo	1Mo

```
CREATE DATABASE helho
ON
( NAME = 'helho_data',
  FILENAME = 'C:\Documents and Settings\STEVO\Mes
documents\Dbt\2ème\programmation\SQL\Helho.mdf',
  SIZE = 2,
  MAXSIZE = 15,
  FILEGROWTH = 1% )
LOG ON
( NAME = 'Helho_log',
  FILENAME = 'C:\Documents and Settings\STEVO\Mes
documents\Dbt\2ème\programmation\SQL\Helho.ldf',
  SIZE = 1,
  MAXSIZE = 5,
  FILEGROWTH = 1 )
```

2. Quelle est l'instruction pour regarder que la base de données est bien créée ?
3. Ajouter une table etudiant avec les attributs suivants :
 - a. Netu de type numeric de 4 chiffres , clé primaire
 - b. Nom de type chaîne de 20 caractères, null interdit
 - c. Année de type integer avec comme valeur par défaut 1

```
CREATE TABLE etudiant(Netu numeric (4) PRIMARY KEY,Nom char (20)NOT
NULL,Année integer default 1)
```

4. Entrer des enregistrements dans la table etudiant et vérifier que tout se passe comme demandé en 3.
5. Ajouter une table professeur avec les attributs suivants :
 - a. Nmatricule de type chaîne de caractères de 5 caractères , clé primaire
 - b. Nom de type chaîne de 20 caractères, null interdit, pas 2 fois le même nom
 - c. Ancienneté de type integer

```
CREATE TABLE professeur(Nmatricule char (5) PRIMARY KEY,Nom char (20)unique
NOT NULL,Anciennete integer)
```

6. Entrer des enregistrements dans la table professeur et vérifier que tout se passe comme demandé en 5.
7. Ajouter une table de jonction professeuretudiant qui a comme champ les clés étrangères netu de étudiant et nmatricule de professeur.

```
CREATE TABLE professeuretudiant(Netu numeric (4)foreign key references
etudiant(netu),nmatricule char(5)foreign key references professeur(nmatricule))
```

8. Ajouter des enregistrements dans la table créée en 7.
9. Renommer la base de données en helho2.
10. Quelle est l'instruction pour supprimer la base de données helho2 ?

```
DROP DATABASE helho2
```