

Laboratoire 1

Date à laquelle le laboratoire doit être terminé : le 05 septembre

Objectifs :

1. Prise en mains de MS SQL Server Management Studio.
2. Bref retour sur le SQL
3. Introduction à Transact-SQL

Important !

- Ce laboratoire est formatif. (Pas de points) ce qui vous enlève du stress et vous permet de le faire par vous-même.
- Une bonne révision du SQL est primordiale pour la suite du cours.
- Un quiz suivra ce laboratoire pour garantir votre compréhension de celui-ci
- L'ensemble des contraintes CHECK, PK, FK doivent avoir des noms significatifs.
C'est une obligation.
- La contrainte de FK doit être définie **sur la TABLE** et non sur la colonne même si CE SGBD vous le permet.

Consignes et bonnes pratiques :

1. Vos requêtes doivent être enregistrées dans un fichier de nom Laboratoire1.sql. Toutes les requêtes doivent être parfaitement identifiées par leur numéro.
2. Pour la question 3 (1 à 3) vous répondrez directement dans votre fichier Laboratoire1.sql en mettant vos réponses en commentaires.
3. Une boîte de remise sera ouverte pour l'ensemble des laboratoires

Question 1 : Création des tables et insertion

1. Connectez-vous à votre SGBD avec votre connexion (Authentification SQL Server)
2. Créez la base de données StockClg
3. Utilisez la base de données StockClg pour créer les tables suivantes :

Table Clients

Colonnes	Types, contraintes
idClient	SMALLINT. IDENTITY, doit commencer à 1 et s'incrémente de 1. idClient est une clé primaire de la table clients
nom	VARCHAR(20) non nul
Prenom	VARCHAR(20) non nul
typeClient	CHAR(1). Default 'r' et prend ses valeurs dans (r, o et v)

Table Articles

Colonnes	Types, contraintes
idArticle	SMALLINT IDENTITY, doit commencer à 1 et s'incrémente de 1. idArticle est une clé primaire de la table Articles
descriptions	VARCHAR(30) non nul
prix	MONEY not null. Le prix doit être plus grand que 0
quantiteStock	SMALLINT not nul. quantiteStock doit être plus grand ou égal à 1

Table Commandes

Colonnes	Types, contraintes
Idcommande	INT, IDENTITY. Doit commencer à 10 et s'incrément de 1. Idcommande est la clé primaire de la table commandes.
dateCommande	DATE. Non nul
idClient	SMALLINT Non nul. Est une clé étrangère faisant référence à idClient de la table Clients

Table LigneCommande

Colonnes	Types, contraintes
Idcommande	INT, non nul. Constitue une clé étrangère faisant référence à Idcommande de la table commandes
idArticle	SMALLINT, non nul Constitue une clé étrangère faisant référence à idArticle de la table Articles.
quantite	INT non null

4. Pour la table LigneCommande, ajouter la clé primaire. La clé primaire de cette table est la composition des deux attributs (idCommande, idArticle)
5. Pour la table Clients, ajouter la colonne Adresse de type VARCHAR (50)
6. Pour la table LigneCommande, renommer la colonne quantite par quantiteCommande.

```
EXEC sp_rename 'emp.nom', 'nomEmp', 'COLUMN';
```

emp désigne le nom de la table
 nom le nom actuelle de la colonne
 nomEmp le nouveau nom.

7. Pour la table LigneCommande, ajouter la contrainte : quantiteCommande doit être supérieur ou égale à 1
8. Dans chacune des tables, insérer les données suivantes en exécutant le script suivant : [Insertionstockclq.sql](#)

Table Articles :

idArticle	descriptions	prix	quantiteStock
1	Imprimantes Laser HP 8000	700,00	20
2	Écrans tactiles 19p	550,00	45
3	Routeurs sans fils Azus	200,00	75
4	Disques durs SSD 500 Go	175,00	45
5	Chaises de bureau	400,00	20
6	Cet Article	400,00	5
7	Mon article	100,00	10

Table Clients

idClient	nom	Prenom	typeClient	adresse
1	Patoche	Alain	v	14 rue Jupiter, Montréal
2	LeRoy	Singe	v	11 avenue de la Lune, Laval
3	LeRigolo	Coluche	r	177 rue de Venus, Montréal
4	Lefou	Ducoin	o	12 rue de Saturne Laval
5	Le magnifique	Simba	r	789 rue des Chats, Laval
6	Ce client	prenom	r	789 rue des clients, Montréal

Table Commandes

idcommande	dateCommande	idClient
10	2023-08-21	1
11	2023-08-28	1
12	2022-12-05	1
13	2022-11-17	2
14	2022-09-12	2
15	2023-08-12	3

Table LigneCommande

s Messages		
idcommande	idArticle	quantiteCommande
10	1	2
10	2	3
10	3	5
10	4	6
11	3	3
11	4	2
12	3	5
13	1	1
13	2	5
13	3	6
14	3	1
14	4	1
15	3	1

Question 2 : écrire les requêtes suivantes :

- Liste des clients de Montréal. Afficher tous les attributs.
- Afficher la description des articles commandés par le client 'Patoche' dans la commande numéro 10.
- Liste des Clients n'ayant fait aucune commande.
- Afficher le numéro des Articles, la description des articles, la quantité commandée. Cet affichage doit inclure les articles qui n'ont jamais été commandés.
- Combien de commandes sont passées par le Client : 'Patoche'?
- Afficher le nom du client ainsi que le nombre total de commande pour chaque client
- A** - Quel est le nom du client qui a passé le plus de commande. Afficher également le nombre de commandes.

B- pour cette requête vous devez afficher le nom de tous les clients ayant fait le plus de commandes. Pour vérifier votre résultat exécuter l'instruction suivantes :

```
INSERT INTO Commandes(datecommande,idClient) VALUES('2025-09-12',2);
```

C-donnez toutes les remarques que vous avez pu faire pour la question 7.

8. Pour la table LigneCommande, ajouter la colonne Montant de type Money.
Attention! La table obtenue ne sera pas normalisée
9. Mettre à jour cette colonne comme suit : le montant est égal à la quantité commandée (quantiteCommande) *le prix, pour chaque article (le prix est dans la table Articles). Ici, on vous demande d'écrire une requête par article. Une pour chaque Article. (Attention ! la solution n'est pas optimale, il existe une solution avec des curseurs, voir chapitre 9)
10. Calculer la somme des montants pour chaque commande.
11. Afficher toutes les commandes avec les noms clients des commandes qui sont passées après septembre 2022 et avant janvier 2023.
12. Créer une vue Vcommande_Client qui va contenir : le nom du client, le numéro de commande et la somme pour chaque commande.
13. Utiliser la vue Vcommande_Client pour calculer pour chaque client le grand total pour ses commandes.

Question 3

1. Dans la table Client, insérez l'enregistrement suivant :
50, 'Lenom', 'Leprenom', 'l'Adresse'). Pour cela vous devez mettre INDENTITY_INSERT à ON
2. Remettez à nouveau NDENTITY_INSERT à OFF puis insérer un autre enregistrement de votre choix. Quelle est la valeur de la clé primaire pour cet enregistrement ? Qu'est-ce que nous pouvons conclure ?
3. Dans la table LigneCommande , pensez-vous que nous aurions pu déclarer Idcommande ou idArticle avec l'option INDENTITY ? **Justifiez votre réponse**
4. Écrire le code Transact-SQL qui permet d'afficher le nom , le prénom du client ainsi que le typeClient. Lorsque le typeClient est v alors, afficher VIP, lorsque le typeClient est r afficher Régulier, lorsque le typeClient est o afficher Occasionnel.
5. Écrire le code Transact-SQL qui permet de faire les opérations suivantes :
(comme un TOUT, doit être dans un bloc BEGIN et END)

- a. On crée une nouvelle commande pour le client numéro 6. (ou un client de votre choix. (Donc on insère dans la table Commandes)
- b. On insère dans la table LigneCommande la commande que l'on vient de créer pour l'article numéro 7 (ou un article de votre choix)
- c. Si la quantité dans la table Articles n'est pas suffisante : ce qui veut dire que la quantité commandée par le client est plus grande ou égale à la quantité en stock, alors on annule la transaction.
- d. Sinon Si la quantité dans la table Articles est suffisante alors
 - i. On met à jour la table Articles par la nouvelle quantité en stock. La nouvelle quantité est égale à la quantité initiale moins la quantité commandée.
 - ii. On met à jour le montant de la commande pour cet article dans la table LigneCommande.
 - iii. On officialise la transaction
- e. Faites les tests suivants :
 - i. La première fois la quantité en stock est insuffisante
 - ii. La deuxième fois la quantité en stock est suffisante
 - iii. Vous pouvez faire les tests : le client n'existe pas. La commande n'existe pas.