

```
/
*****
***

## Devoir 4
## *****
*****

***

## Infos cours
*****

## Cours : Introduction aux bases de données
## Session : Hiver 2025
## Enseignant : Kola Couthon

## Infos étudiant
*****

## Nom : Lussier
## Prénom : Nathan

## Note : 28/30
## *****

## Bonus :
## Pénalité :

Commentaire
*****

# Consignes
# =====
# Pour bien écrire des requêtes, suivez les étapes ci-dessous :
# Consultez la base de données active
# Consultez la structure des tables (le nom des colonnes, le type des colonnes)
# Consultez les valeurs des colonnes
# Identifiez visuellement les enregistrements qui satisfont aux conditions de l'énoncé
# Interprétez bien la question posée et traduisez la en une expression conditionnelle compréhensible par SQL
# Les notions apprises en programmation sont transférables en SQL notamment les expressions conditionnelles
# Séparez les colonnes par des virgules (,) sauf la dernière
# Séparez chaque requête par un point-virgule (;)
# Incluez toujours la ou les colonnes des conditions pour vous assurer que le résultat obtenu est le résultat attendu. Vous pourrez les supprimer par la suite si l'énoncé ne demande pas de les afficher
# Mettez-vous à la place de votre client (celui qui pose la question) afin de comprendre ce qu'il veut
```

```
# Apprenez à trouver vos erreurs tout seul. Verifiez que la syntaxe est correcte.
# Pour les requêtes multitable, utilisez la démarche systématique vue en classe, puis procédez graduellement, deux tables à la fois.
# Pour les requêtes multi-critères, évaluez toutes les conditions / critères de l'énoncé
# N'oubliez pas les tris lorsque c'est demandé.
```

```
## Important
## *****
```

```
# RESPECTEZ LES SPÉCIFICATIONS DE L'ÉNONCÉ
# Structurez et indentez votre code SQL comme ceci :
```

```
SELECT
    Col1
    ,Col2
    ,Col3
FROM
    table1,
    table2
WHERE
    Cond1 AND
    Cond2 AND
    Cond3
GROUP BY
    Col1
HAVING
    Condition_groupe
ORDER BY
    Col1,
    Col2
```

```
# Écrivez votre code vis à vis de la question. Par exemple : Le code de la requête 1 DOIT se trouver en dessous de requête 1 et non requête 2. La question obtiendra la note 0 si le faites.
# Si la requête ne produit pas le résultat attendu, la question obtiendra la note 0 (critère)
# Si la requête produit le résultat attendu et qu'il manque une spécification, vous obtiendrez une partie des points. Exemple : Vous oubliez un tri...
# Dans TOUTES les requêtes, utilisez des noms d'alias de votre choix mais SIGNIFICATIFS pour désigner les colonnes lors de l'affichage.
```

```
## Base de données à utiliser : devoir4
```

```
# *****
```

```
# Vous devez créer manuellement la base de données afin de pouvoir réaliser le devoir.
```

```
****/
```

```
/****** Les objets doivent être droppés avant leur création
*****/
```

```

...ussier_2484435_Devoir_4_Remis_le_2025-05-07_17h09m04s.sql 3
/***** Le script devra s'exécuter SANS ERREUR *****/
/***** N'oubliez pas la commande GO pour valider vos transactions *****/

-- Utilisation de la base de données

USE devoir4;
GO

-- Étape préliminaire
--
-----
-----
-- Installer la base de données Xtreme.
-- Effectuer une copie des tables suivantes situées dans la base de données Xtreme
vers la nouvelle base de données "devoir4".
-- Le nom des copies sera précédé de tbl. Toutes les requêtes DOIVENT être
effectuées sur les copies dans la base de données "devoir4"
--
-----
-----
/*
*****
*****
employés --> tblEmployes,
commandes --> tblCommandes,
*****
*****/

-- Création de la table tblCommandes
DROP TABLE IF EXISTS tblCommandes;
SELECT *
INTO tblCommandes
FROM [xtremefr].[dbo].[Commandes];

-- Création de la table tblEmployes
DROP TABLE IF EXISTS tblEmployes;
SELECT *
INTO tblEmployes
FROM [xtremefr].[dbo].[Employés];

/
*****
*****/

```

# Question 1 : 15/15pts

\*\*\*\*\*

Commentaire :

Très bien. Très propre.

\*\*\*\*\*

\*\*

Énoncé:

Une entreprise RecycleVAT inc, soucieuse de l'environnement, se lance dans le recyclage des objets en fin de vie, des "SertPlusARien".

Elle reçoit les objets de la part d'associations à but non lucratif qui les ramassent chez les particuliers pour ensuite les déposer dans des centres pour y être recyclés.

RecycleVAT travaille actuellement avec un fichier Excel "RecycleVAT.xlsx" pour faire la gestion de son activité.

Elle vous donne le mandat de créer une base de données qui lui permettra de gérer plus efficacement la collecte et le dépôt des objets dans les centres.

Modélisez la base de données.

Écrire le code SQL qui permet de créer les tables ainsi que les propriétés des colonnes nécessaires.

Les tables seront créées dans la base de données "devoir4"

\*\*\*\*\*

\*\*\*/  
-- Votre code ici

```
DROP TABLE IF EXISTS tblDepot;
```

```
GO
```

```
DROP TABLE IF EXISTS tblCentre;
```

```
GO
```

```
DROP TABLE IF EXISTS tblAssociation;
```

```
GO
```

```
CREATE TABLE tblAssociation(
```

```
    NoAssociation int PRIMARY KEY not null,
```

```
    NomAssociation nvarchar(15),
```

```
    TéléphoneAssociation nvarchar(8),
```

```
    PrénomResponsable nvarchar(15),
```

```
    NomResponsable nvarchar(15)
```

```
);
```

```
GO
```

```
CREATE TABLE tblCentre(
```

```
    NoCentre int PRIMARY KEY not null,
```

```
    NomCentre nvarchar(15),
```

```
    TéléphoneCentre nvarchar(8),
```

```
    PrénomContact nvarchar(15),
```

```
    NomContact nvarchar(15)
```

```
);
```

```
GO
```

```
CREATE TABLE tblDepot(
    NoDépôt int PRIMARY KEY not null,
    Dangereux bit not null,
    NoCentre int FOREIGN KEY REFERENCES tblCentre(NoCentre),
    NoAssociation int FOREIGN KEY REFERENCES tblAssociation(NoAssociation),
    DateDépôt date,
    PoidsDépôt smallint
```

```
);
```

```
GO
```

```
--
-----
--
```

```
--
*****
**
```

```
-- Toutes les requêtes de mise à jour doivent suivre les trois étapes :
-- 1- Identification des enregistrements à mettre à jour
-- 2- Mise à jour à proprement parler
-- 3- Vérification de la mise à jour.(une instruction avec la condition appropriée
    pour vérifier que la mise à jour a bien fonctionné)
```

```
/
*****
****
```

```
# Question 2 : 5/5pts
```

```
*****
```

```
Commentaire :
```

```
*****
**
```

```
Énoncé:
```

```
Écrire une requête qui augmente de 5% le salaire de tous les employés embauchés
après 1996.
```

```
*****
***/
```

```
-- # Votre code ici
```

```
-- Identification - 2pts
```

```
SELECT
```

```
    [ID employé]
    ,[Nom]
    ,[Date d'embauche]
    ,[Salaire]
```

```
FROM
```

```

[dbo].[tblEmployes]
WHERE
    YEAR([Date d'embauche]) > 1996

-- Mise à jour - 2pts

UPDATE [dbo].[tblEmployes]
    SET [Salaire] = CASE
        WHEN YEAR([Date d'embauche]) > 1996 THEN [Salaire] * 1.05
        ELSE [Salaire]
    END
GO

```

```
-- Vérification de la mise à jour - 1pt
```

```

SELECT
    [ID employé]
    , [Nom]
    , [Date d'embauche]
    , [Salaire]
FROM
    [dbo].[tblEmployes]
WHERE
    YEAR([Date d'embauche]) > 1996

```

```
/
```

```

*****
****

```

```
# Question 3 : 5/5pts
```

```
*****
```

```
Commentaire :
```

```

*****
**

```

```
Énoncé:
```

```
Modifiez la table tblEmployes en rajoutant la colonne "StatutEmploi" de type texte
de longueur 15.
```

```
Écrire une requête qui met à jour la colonne "StatutEmploi" avec "Cadre" pour les
employés ayant un salaire supérieur à 60 000.
```

```

*****
***/

```

```
-- # Votre code ici
```

```
-- Vérification - 1pt
```

```

ALTER TABLE [dbo].[tblEmployes]
    DROP COLUMN IF EXISTS StatutEmploi
GO

```

-- Ajout de la colonne StatutEmploi - 1pt

```
ALTER TABLE [dbo].[tblEmployes]
    ADD StatutEmploi nvarchar(15)
GO
```

-- Vérification de l'ajout

```
SELECT
    [ID employé]
    ,StatutEmploi
FROM
    [dbo].[tblEmployes]
```

-- Identification des enregistrements à mettre à jour - 1pt

```
SELECT
    [ID employé]
    , [Nom]
    , [Prénom]
    , [Salaire]
    , [StatutEmploi]
FROM
    [dbo].[tblEmployes]
WHERE
    [Salaire] > 60000
```

-- 2- Mise à jour à proprement parler - 1pt

```
UPDATE [dbo].[tblEmployes]
    SET [StatutEmploi] =
        CASE
            WHEN [Salaire] > 60000 THEN 'Cadre'
            ELSE [StatutEmploi]
        END
GO
```

-- 3- Vérification de la mise à jour. - 1pt

```
SELECT
    [ID employé]
    , [Nom]
    , [Prénom]
    , [Salaire]
    , [StatutEmploi]
FROM
    [dbo].[tblEmployes]
WHERE
    [Salaire] > 60000
```

```
/
*****
****
# question 4 : 3/5pts
*****
Commentaire :
Vous avez oublié la condition.
*****
**
Énoncé:
Créer une vue vMeilleursVendeurs qui affiche les employés ayant réalisé les
meilleures ventes en 2005.
On affichera :
-- Le nom complet de l'employé
-- Les ventes réalisées (C'est le montant des commandes puisque c'est l'employé qui
a traité les commandes)

Afficher la liste en ordre décroissant des ventes.
*****
***/
-- # Votre code ici

-- Verification vue - 1pt

DROP VIEW IF EXISTS vMeilleursVendeurs
GO

-- Création vue - 1/3pts

CREATE VIEW vMeilleursVendeurs
AS
SELECT
    Concat([Prénom], ' ', [Nom]) AS 'Nom Complet Employer'
    ,SUM([Montant commande]) AS 'Total des Ventes'
FROM
    [dbo].[tblEmployes] E
    ,[dbo].[tblCommandes] C
WHERE
    E.[ID employé] = C.[ID employé]
GROUP BY
    Concat([Prénom], ' ', [Nom])
GO

-- Affichage de la liste en ordre décroissant des ventes - 1pt

SELECT
    *
FROM
```



vMeilleursVendeurs

ORDER BY

[Total des Ventes] DESC

/\* \*\*\*\*\* Fin de l'examen  
\*\*\*\*\* \*/

