
Devoir 3

Infos cours

Cours : Introduction aux bases de données

Session : Hiver 2025

Enseignant : Kola Couthon

Infos étudiant

Nom : Lussier

Prénom : Nathan

Note : 25/35

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. Vérifiez que la syntaxe est correcte.

Pour les requêtes multitables, 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
    Col
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 : devoir3
# *****
# Vous devez créer la base de données afin de pouvoir réaliser le devoir.
*****
/
/***** Les objets doivent être droppés avant leur création
*****/
/***** Le script devra s'exécuter SANS ERREUR
*****/
/***** N'oubliez pas la commande GO pour valider vos transactions
*****/

/***** Début de script
*****/

-- Utilisation de la base de données
-- Votre code ici
```

```
USE devoir3;
GO
```

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

```
# Étape 1 : 5/5pts
```

```
*****
```

```
Commentaire :
```

```
*****
```

```
Créer une vue vEmployeInfos qui contient les informations suivantes :
```

```
-- L'ID de l'employé
-- Le nom de l'employé
-- Le prénom de l'employé
-- Le département
-- Le grade
-- Le no du poste
-- La description du poste
-- Le type du poste
-- La description du type de poste
-- Le taux horaire
-- Salaire annuel
-- Le code de la situation d'emploi
-- La description de la situation d'emploi
-- Le pourcentage de la situation d'emploi
-- Le code du statut
-- Le statut
-- Les heures travaillées
--
```

```
*****
```

```
/
-- Votre code ici
DROP VIEW IF EXISTS vEmployeInfos;
GO
```

```
CREATE VIEW vEmployeInfos
```

```
AS
```

```
SELECT
```

```
    E.[EmpID] AS 'ID Employe'
  , E.[Nom] AS Nom
  , E.[Prenom] AS Prenom
  , D.[Departement] AS Departement
  , P.[Grade] AS Grade
  , P.[NoPoste] AS 'Numero de Poste'
  , P.[DescriptionPoste] AS 'Description du Poste'
  , P.[TypePoste] AS 'Type du Poste'
  , P.[DescriptionTypePoste] AS 'Description du Type du Poste'
  , P.[TauxHoraire] AS 'Taux Horaire'
  , P.[SalaireAnnuel] AS 'Salaire Annuel'
  , P.[SituationEmploi] AS 'Code de la situation d'emploi'
  , SE.[SituationEmploi] AS 'Description de la situation d'emploi'
```

```

,SE.[PourcSituationEmploi] AS 'Pourcentage de la situation d'emploi'
,S.[CodeStatut] AS 'Code du Statut'
,[Statut] AS Statut
,EH.[HresTravaillees] AS 'Heures de Travaillees'
FROM
    [dbo].[tblEmployes] E,
    [dbo].[tblDepartement] D,
    [dbo].[tblEmployePoste] EP,
    [dbo].[tblPoste] P,
    [dbo].[tblStatutEmploi] SE,
    [dbo].[tblEmployeStatut] ES,
    [dbo].[tblStatut] S,
    [dbo].[tblEmployeHeures] EH
WHERE
    E.[CodeDept] = D.[DeptCode] AND
    E.[EmpID] = EP.[EmpID] AND
    P.[NoPoste] = EP.[NoPoste] AND
    SE.[CodeSituationEmploi] = P.[SituationEmploi] AND
    E.[EmpID] = ES.[EmpID] AND
    S.[CodeStatut] = ES.[CodeStatut] AND
    E.[EmpID] = EH.[EmpID]
GO

-- Vérifier la vue

SELECT *
FROM [dbo].[vEmployeInfos]

/*****
# Étape 2 : 2/2pts
*****/
Commentaire :

*****/
Créer une table tempEmpBoni qui contient les employés admissibles pour recevoir un  ➤
boni
à la fin de l'année. Ils doivent respecter toutes les conditions mentionnées dans  ➤
l'énoncé.

*****/ ➤
/
-- # Votre code ici

DROP TABLE IF EXISTS tempEmpBoni;
GO

SELECT
    [ID Employe]
    ,CONCAT([Prenom], ' ', [Nom]) AS 'Nom de l'employe'
    ,[Heures de Travaillees]
    ,[Taux Horaire]
    ,[Salaire Annuel]

```

```

    ,[Grade]
INTO
    tempEmpBoni
FROM
    [dbo].[vEmployeInfos]
WHERE
    [Statut] NOT LIKE('Terminé') AND
    [Type du Poste] LIKE('P')

```

-- Vérification de la création de la table

```

SELECT *
FROM tempEmpBoni

```

/*****

Étape 3 : 2/3pts

Commentaire :

Modifier la structure de la table temporaire tempEmpBoni

Ajouter la colonne GainsAdmissibles de type monétaire qui contiendra le montant qui sera pris en compte pour le calcul des bonis.

/

-- # Votre code ici

```

ALTER TABLE tempEmpBoni
    ADD GainsAdmissibles money
GO

```

-- Vérifier l'ajout de la nouvelle colonne

```

SELECT *
FROM tempEmpBoni

```

/*****

Étape 4 : 0/2pts

Commentaire :

Les conditions de mise à jour n'ont pas été respectées

Mettre à jour la colonne GainsAdmissibles en respectant les règles d'affaires.

La colonne contiendra le montant qui sera pris en compte pour le calcul du boni.

/

-- # Votre code ici

```

UPDATE tempEmpBoni
    SET [GainsAdmissibles] = (CAST([Taux Horaire] AS INT) * [Heures de Travaillées])

```

GO

-- Vérifier la table avec la nouvelle colonne

```
SELECT *
FROM tempEmpBoni
```

```
/******
# Étape 5 : 2/3pts
Commentaire :
```

```
*****
```

Modifier la table tempEmpBoni

Rajouter les colonnes suivantes :

- PourcBoni : type entier - qui affichera le pourcentage en fonction des règles
- MontantBoni : type monétaire - Le montant du boni en fonction des règles

```
*****
```

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

```
ALTER TABLE tempEmpBoni
    ADD PourcBoni int,
    MontantBoni money
GO
```

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

Étape 6 : 4/4pts

Commentaire :

```
*****
```

Mettre à jour les colonnes PourcBoni et MontantBoni en utilisant les données de la table
tblBoniCharge

```
*****
```

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

```
UPDATE tempEmpBoni
    SET [PourcBoni] =
        CASE
            WHEN [Grade] = 'MP1' THEN 5
            WHEN [Grade] = 'MP2' THEN 10
            WHEN [Grade] = 'MP3' THEN 15
            WHEN [Grade] = 'MP4' THEN 30
            WHEN [Grade] = 'MP5' THEN 40
            WHEN [Grade] = 'MP6' THEN 100
        END
```

GO

UPDATE tempEmpBoni

```
SET [MontantBoni] = ([GainsAdmissibles] * [PourcBoni])/100
GO
```

--Vérification

```
SELECT *
FROM tempEmpBoni
```

```
/******
# Étape 7 : 2/4pts
Commentaire :
```

```
*****
```

Modifier la table tempEmpBoni

Rajouter les colonnes suivantes :

- PourcDeduction : type entier
- MontantDeduction : type monétaire
- SalaireNet : monétaire

Note : Les déductions, ce sont les charges sociales

```
*****
```

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

```
ALTER TABLE tempEmpBoni
ADD PourcDeduction int,
    MontantDeduction money,
    SalaireNet money
GO
```

```
/******
# Étape 8 : 2/6pts
Commentaire :
```

```
*****
```

Mettre à jour les trois colonnes nouvellement rajoutées

```
*****
```

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

```
UPDATE tempEmpBoni
SET PourcDeduction =
CASE
    WHEN [Grade] = 'MP1' THEN 35
    WHEN [Grade] = 'MP2' THEN 37
    WHEN [Grade] = 'MP3' THEN 40
    WHEN [Grade] = 'MP4' THEN 45
    WHEN [Grade] = 'MP5' THEN 49
    WHEN [Grade] = 'MP6' THEN 52
END
```

```
GO
UPDATE tempEmpBoni
```

```

SET MontantDeduction = ([GainsAdmissibles] * PourcDeduction)/100
GO
UPDATE tempEmpBoni
SET SalaireNet = [GainsAdmissibles] - MontantDeduction
GO

```

```

-- Vérification
SELECT *
FROM tempEmpBoni

```

```

/*****

```

```

# Étape 9 : 5/5pts

```

```

Commentaire :

```

```

*****/

```

Créer la table tblPaielement qui contient la liste finale qui sera envoyée à la comptabilité pour virement sur le compte des employés.

La structure de la table tblPaielement est la suivante:

- Le numéro de l'employé : entier, clé primaire, non nul
- Le nom complet de l'employé : chaîne de long. 50
- La description du poste : chaîne de long, :30
- le pourcentage du boni : int
- Le montant du boni : monétaire
- Le salaire net : monétaire

La table sera triée par le nom et le prénom de l'employé

```

*****/

```

```

/
-- # Votre code ici

```

```

DROP TABLE IF EXISTS tblPaielement;
GO
CREATE TABLE tblPaielement(
    noEmploye int PRIMARY KEY NOT NULL,
    nomCompletEmploye nvarchar(50),
    descriptionDuPoste nvarchar(30),
    pourcentageBoni int,
    montantBoni money,
    salaireNet money
)
GO

```

```

INSERT INTO tblPaielement
SELECT
    T.[ID Employe]
    ,T.[Nom de l'employe]
    ,V.[Description du Poste]
    ,T.[PourcBoni]
    ,T.[MontantBoni]
    ,T.[SalaireNet]
FROM

```



```
    [dbo].[tempEmpBoni] T,  
    [dbo].[vEmployeInfos] V  
WHERE  
    T.[ID Employe] = V.[ID Employe]  
ORDER BY  
    T.[Nom de l'employe]
```

```
-- Affichage de la liste finale
```

```
SELECT *  
FROM tblPaielement
```

```
/*  
# Étape 10 : 1/1pt  
Commentaire :
```

```
*****  
Nettoyer l'environnement en supprimant les objets temporaires
```

```
*****/  
-- # Votre code ici
```

```
DROP TABLE IF EXISTS tempEmpBoni;  
GO
```

```
/* ***** Fin de script ***** */
```