ANDRIANAVALONA RAKOTOFIRINGA Ando
L2 RSI

NFO_371 SQL Server

Exercice 1 :

1) Trois différents types de SQL Server et leurs rôles
   - SQL Server Express SSE : Permet de créer des applications bureautiques et de petites applications serveur pilotées par les données et disponible gratuitement.

   - SQL Server Developer : Permet de créer, testez et faire des démonstrations d'applications dans un environnement non dédié à la production ensemble de logiciel offrant l'édition comprenant l'ensemble des fonctions de SQL Server

   - SQL Server Enterprise Edition : Permet d'accéder à des fonctionnalités stratégiques pour atteindre une montée en charge et une sécurité inégalée, une haute disponibilité et des performances exceptionnelles pour la base de données, avec l'ensemble des fonctionnalités de SQL Server Standard.

2) SQL signifie : **a) Structured Query Language**
3) SQL peut être géré les accès concernant aux données :
   b. SET TRANSACTION ISOLATION SERVER
4) Leur signification et leurs rôles :
   a. T-SQL : Transact-SQL : C'est une extension propriétaire de Sybase et Microsoft au langage SQL. Le T-SQL sert à étendre les fonctionnalités du SGBDR SQL Server grâce à des procédures stockées.
   b. SSMS: SQL Server Management Studio : C'est une application logicielle lancée pour la première fois avec Microsoft SQL Server 2005 et utilisée pour la configuration, la gestion et l'administration de tous les composants de Microsoft SQL Server.
   c. SSE: SQL Server Express (Version gratuite de Microsoft SQL Server)
5) les étapes à suivre pour tracer tous les requêtes effectuées dans la base de données et dans le serveur SQL :
   a. **Démarrez SQL Server Management Studio,**
   b. **Se rendre dans le menu Outils pour démarrer SQL Server Profiler.**
   c. **Donner un nom à la trace.**
   d. **Allez dans l'onglet "sélection d'évènement"**
   e. **Cochez seulement les cases don-on a besoin, afin de minimiser la charge que la trace s'appropriera : TextData, Reads, Write.**
   f. **Exécutez**

Exercice 2 :

Creation des tables dans SSMS :

```
CREATE TABLE VILLE
(
    CodePostal INT PRIMARY KEY,
    NomVille VARCHAR(50) Not NULL
)
CREATE TABLE CINEMA
(
    NumCine INT PRIMARY KEY,
    NomCine VARCHAR(50) NOT NULL,
    Adresse VARCHAR(50) NOT NULL,
    CodePostal INT FOREIGN KEY REFERENCES VILLE(CodePostal)
)
CREATE TABLE SALLE
(
    NumSalle INT PRIMARY KEY,
    Capacite INT Not NULL,
    NumCine INT FOREIGN KEY REFERENCES CINEMA(NumCine)
)
CREATE TABLE FILM
(
    NumExploit INT PRIMARY KEY,
    titre VARCHAR(50) NOT NULL,
    Duree TIME,
)
CREATE TABLE PROJECTION
(
    NumExploit INT FOREIGN KEY REFERENCES FILM(NumExploit),
    NumSalle INT FOREIGN KEY REFERENCES SALLE(NumSalle),
    NumSemaine INT NOT NULL,
    NbEntree INT NOT NULL
```

a. Affichage du titre des films dont la durée est supérieure ou égale à deux heures

```
SELECT NumExploit , titre FROM FILM WHERE Duree >= '2:00'
```

b. La liste des noms des cinémas situés à Meknès contenant au moins une salle de plus de 100 places :

```
SELECT NomCine FROM CINEMA WHERE CodePostal=(SELECT CodePostal FROM VILLE WHERE
NomVille='Meknes') AND NumCine IN (SELECT NumCine FROM SALLE WHERE Capacite
>=100)
```

c. Affichage du nom, l'adresse et la ville des cinémas dans lesquelles on joue le film "hypnose" la semaine 18 :

```
SELECT C.NomCine , C.Adresse , V.NomVille FROM CINEMA as C, VILLE as V  WHERE
C.CodePostal=V.CodePostal AND C.NumCine IN (SELECT S.NumCine FROM SALLE as S,
FILM as F, PROJECTION as P WHERE P.NumExploit=F.NumExploit AND
P.NumSalle=S.NumSalle AND F.Titre='Hypnose' AND P.NumSemaine=18)
```

d. La liste des titres et films qui n'ont pas été projetés :

```
SELECT Titre FROM FILM WHERE NumExploit NOT IN (SELECT NumExploit FROM
PROJECTION)
```

Exercice 3

1. En utilisant le concept de la procédure stockée (PS) et de la notion T-SQL, expression SQL des requêtes :

   a. affichage de la liste des employées ayant une commission

```
CREATE PROCEDURE employeecomm
AS
BEGIN
    SELECT * FROM Employes WHERE COMM IS NOT NULL
END
GO
USE Employes
GO
SET QUOTED_IDENTIFIER ON

GO
CREATE PROCEDURE [dbo].[Selection]

AS
BEGIN
    BEGIN try
        BEGIN transaction
            SELECT * FROM Employes WHERE NOT COMM = '' OR NOT COMM = NULL
COMMIT transaction
    END try
    BEGIN catch
        ROLLBACK transaction
        print 'Erreur'
    END catch
END
```

   b. liste des noms, emplois et salaires des employés par emploi croissant et pour chaque emploi par salaire décroissant

```
CREATE PROCEDURE liste_par_emp_et_sal
AS
BEGIN
    SELECT ENOM ,PROF,SAL FROM Employes ORDER BY PROF ASC, SAL DESC
END
```

   c. affichage du salaire moyen des employés

```
CREATE PROCEDURE [dbo].[SalaireMoyen]
AS
BEGIN
    BEGIN try
        BEGIN transaction
            SELECT AVG(SAL) FROM Employes
COMMIT transaction
    END try
    BEGIN catch
        ROLLBACK transaction
        print 'Error'
    END catch
END
```