**TRAVAUX PRATIQUES**

**Introduction au langage Transact SQL (T-SQL)**

Microsoft SQL Server

**Tableau récapitulatif des syntaxes :**

|  |  |
| --- | --- |
| **Action** | **Syntaxe** |
| Déclaration de variables | **DECLARE @nom\_variable type [ = valeur\_initiale ];** |
| Affectation de variables | **SET @nom\_variable = expression;** |
| **SELECT @nom\_variable = expression;** |
| Affichage de variables | **PRINT @nom\_variable;** (nécessite une conversion si **non-string**) |
| **SELECT @nom\_variable [AS alias];** |
| Fonction de conversion CAST | **CAST(expression AS type);** ou **CAST(@nom\_variable AS type);** |
| Fonction de conversion CONVERT | **CONVERT(type, expression);** ou **CONVERT(type, @nom\_variable);** |

**Fonctions temporelles :**

|  |  |  |  |
| --- | --- | --- | --- |
| **Fonction** | **Description** | **Syntaxe** | **Exemple** |
| GETDATE() | Retourne la date et l'heure actuelle du système en tenant compte des paramètres régionaux. | **GETDATE()** | **DECLARE @DateActuelle DATETIME = GETDATE();**  **SELECT GETDATE() AS CurrentDateTime;** |
| DATEDIFF() | Calcule la différence entre deux dates en termes d'une unité de temps, telle que :  year , month , day, week, hour , minute et second | **DATEDIFF(unit, start\_date, end\_date)** | **DECLARE @NbJour INT;**  **SET @NbJour = DATEDIFF(day, '2000-01-01', '2020-12-31');**    **SELECT DATEDIFF(YEAR, '2000-01-01', '2020-12-31');** |

**Remarque :**

En T-SQL, il est recommandé d'utiliser le format de date standard ISO 8601 "**YYYY-MM-DD**" pour éviter toute confusion ou ambigüité dans le traitement des dates.

En T-SQL, **les noms des variables ne sont pas sensibles à la casse**. Cela signifie que vous pouvez utiliser des noms de variables en **majuscules**, en **minuscules** ou en combinant les deux, **sans affecter le comportement ou la fonctionnalité de votre code**.

**Schéma de la base de données EMSIBD :**

**Eleve(**id\_eleve, nom, prenom, date\_naissance, **#**id\_classe**)**

**Enseignant(**id\_enseignant, nom, prenom, email**)**

**Cours(**id\_cours, nom, **#**id\_enseignant**)**

**Classe(**id\_classe, nom**)**

**Note(#**id\_eleve, **#**id\_cours, note**)**

1. Manipulation 1 : Chargement et exécution de la base de données EMSIBD

Dans cette première étape, vous allez charger et exécuter le script SQL EMSIBD\_Script.sql afin de créer la base de données EMSIBD dans SQL Server Management Studio (SSMS).

Étapes à suivre :

1. **Ouvrir SQL Server Management Studio (SSMS)**
   * Lancez l'application SSMS et connectez-vous à votre serveur SQL.
2. **Charger le script SQL**
   * Dans la barre de menu, cliquez sur **File** *(ou* ***Fichier*** *en français)*.
   * Sélectionnez **Open** *(ou* ***Ouvrir*** *en français)*, puis **File...** *(Fichier...)*.
   * Parcourez votre ordinateur pour trouver le fichier **EMSIBD\_Script.sql** et ouvrez-le.
3. **Exécuter le script**
   * Une fois le script chargé dans l'éditeur SQL, cliquez sur le bouton **Execute** (ou **Exécuter** en français) en haut de l’interface, ou appuyez sur la touche **F5** de votre clavier.
4. **Vérifier la création de la base de données**
   * Dans l'Explorateur d'Objets (Object Explorer), rafraîchissez la liste des bases de données en effectuant un clic droit sur **Databases** (Bases de données) et en sélectionnant **Refresh** (Actualiser).
   * Assurez-vous que la base de données **EMSIBD** apparaît dans la liste.
5. Manipulation 2 : Déclaration et affectation de variables

#### ****Question 1 : Déclarez trois variables en T-SQL :****

* Une variable **@NomEleve** de type VARCHAR(50)
* Une variable **@PrenomEleve** de type VARCHAR(50)
* Une variable **@IdEleve** de type INT
* Assignez les valeurs suivantes aux variables à l’aide de **SET** :
  + 17 pour **@IdEleve**
  + 'Amrani' pour **@NomEleve**
  + 'Hamza' pour **@PrenomEleve**
* Affichez les valeurs des variables en utilisant **PRINT**.

**Question 2 :**

* Déclarer deux entiers et les initialiser à 5 et 3 respectivement.
* Calculer la somme de ces deux entiers,
* Afficher le résultat comme suit: “La somme est 8”.

#### ****Question 3 : Utilisez**** SELECT ****pour affecter les valeurs d’un élève aux variables****

* Déclarez trois variables @Nom, @Prenom et @DateNaissance.
* Récupérez le **nom**, le **prénom** et la **date de naissance** d’un élève ayant id\_eleve = 2 depuis la table **Eleve** en utilisant une requête **SELECT**.
* Affichez les valeurs obtenues avec **PRINT**.
* La date de naissance doit être affichée sous forme de texte (VARCHAR).

#### ****Question 4 : Calculez l’âge d’un élève en années****

* Déclarez une variable **@DateNaissance** et affectez-lui la date de naissance d’un élève en fonction de son id\_eleve *(par exemple id\_eleve =3)* depuis la table **Eleve**.
* Déclarez une variable **@AgeEleve** et affectez-lui l’âge de cet élève en calculant la différence entre la date de naissance et la date actuelle **(GETDATE())** en utilisant **DATEDIFF(YEAR, ...)**.
* Affichez la date de naissance, l’âge calculé et la date actuelle.

**Question 5 : Déclaration d’une variable de type table**

* Déclarez une variable **@tableEleve** de type table contenant les champs suivants: ID de type INT, nom de type NVARCHAR(50), et IDclass de type INT.
* Insérer l’enregistrement suivant dans la table **tableEleve**:

|  |  |
| --- | --- |
| **Colonne** | **Valeur** |
| ID | 10 |
| nom | Ahmed |
| IDclasse | 1 |

* Puis, afficher les enregistrements de la table **tableEleve.**

### **Question 5 : Affichez les informations d’un élève avec sa classe**

* Déclarez une variable **@IdEleve** de type INT et affectez-lui un identifiant d’élève **de votre choix** *(de 1 à 3)*.
* Déclarez une variable **@NomEleve** de type VARCHAR(50), **@PrenomEleve** de type VARCHAR(50), et **@NomClasse** de type VARCHAR(50).
* Utilisez une jointure interne pour récupérer le nom, le prénom de l’élève et le nom de sa classe en fonction de son id\_eleve.
* Affichez les informations récupérées selon le schèma (NomEleve, PrenomEleve et NomClasse) avec la commande **PRINT**.

### **Question 6 : Affichez les informations des cours avec le nom des enseignants**

* Déclarez une variable **@IdCours** de type INT et affectez-lui un identifiant de cours de votre choix *(de 1 à 3)*.
* Déclarez une variable **@NomCours** de type VARCHAR(50), **@NomEnseignant** de type VARCHAR(50), et **@PrenomEnseignant** de type VARCHAR(50).
* Utilisez une **jointure interne** entre les tables Cours et Enseignant pour récupérer le **nom du cours** et le **nom et prénom de l’enseignant** qui l’enseigne.
* Affichez ces informations avec **PRINT**.

### **Question 7 : Affichez les notes des élèves avec leurs cours et leurs enseignants**

* Déclarez une variable **@IdEleve** de type INT et affectez-lui un identifiant d’élève de votre choix *(de 1 à 3)*.
* Déclarez une variable **@NomCours** de type VARCHAR(50) et affectez-lui la valeur ‘Histoire’.
* Déclarez une variable **@NomEleve** de type VARCHAR(50), **@PrenomEleve** de type VARCHAR(50) et **@Note** de type DECIMAL(5,2).
* Utilisez une **double jointure interne** pour récupérer :
  + Le **nom et prénom de l’élève**
  + Le **nom du cours**
  + La **note de l’élève dans ce cours**
* Affichez ces informations avec PRINT.

Exemple :

*Etudiant : Amrani Ahmed*

*Cours: Histoire*

*Note: 13*