Résumé SGBD1 et SGBD2

SGBD1

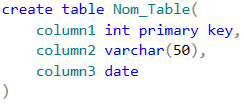
L’instruction CREATE DATABASE est utilisée pour créer une nouvelle base de données SQL :

1.PNG

L’instruction DROP DATABASE est utilisée pour supprimer une base de données SQL existante :

2.PNG

L’instruction CREATE TABLE est utilisée pour créer une nouvelle table dans une base de données :



L’instruction DROP TABLE est utilisée pour supprimer une table existante dans une base de données :

4.PNG

L’instruction TRUNCATE TABLE est utilisée pour supprimer les données dans une table, mais pas la table elle-même :

5.PNG

L’instruction ALTER TABLE est utilisée pour ajouter, supprimer ou modifier des colonnes dans une table existante.

L’instruction ALTER TABLE est également utilisée pour ajouter et supprimer diverses contraintes sur une table existante.

Pour ajouter une colonne dans une table, utilisez la syntaxe suivante :

6.PNG

Pour supprimer une colonne dans une table, utilisez la syntaxe suivante :

7.PNG

Pour modifier le type de données d’une colonne dans une table, utilisez la syntaxe suivante :

8.PNG

Les contraintes SQL sont utilisées pour spécifier des règles pour les données d’une table.

Des contraintes peuvent être spécifiées lorsque la table est crée avec l’instruction CREATE TABLE ou après la création de la table avec l’instruction ALTER TABLE.

Les contraintes suivantes sont couramment utilisées dans SQL :

NOT NULL : Garantit qu’une colonne ne peut pas avoir une valeur NULL.

UNIQUE : Vérifie que toute les valeurs d’une colonne sont différentes.

PRIMARY KEY : Une combinaison d’un NOT NULL et UNIQUE. Identifie de manière unique chaque ligne d’une table.

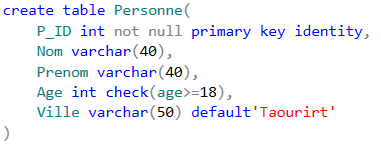
TOUCHE ETRANGERE : Identifie de manière une ligne / un enregistrement dans une table.

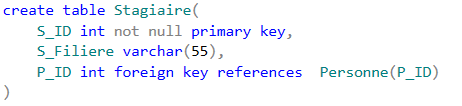
CHECK : Vérifie que toutes les valeurs d’une colonne satisfont à une condition spécifique.

DEFAULT : Définit une valeur par défaut pour une colonne lorsqu’une valeur n’est spécifique.

INDEX : Utilisé pour créer et extraire les données de la base de données très rapidement.

Exemples :





Pour créer une contrainte UNIQUE sur la colonne « P\_ID » lorsque la table et déjà créée :

11.PNG

Pour nommer une contrainte UNIQUE et pour définir une contrainte UNIQUE sur plusieurs colonnes :

12.PNG

Pour supprimer une contrainte UNIQUE :

13.PNG

Pour créer une contrainte PRIMARY KEY sur la colonne « P\_ID » lorsque la table est déjà créée :

14.PNG

Pour autoriser la dénomination d’une contrainte PRIMARY KEY et pour définir une contrainte PRIMARY KEY sur plusieurs colonnes :

15.PNG

Pour supprimer contrainte PRIMARY KEY :

16.PNG

Pour créer une contrainte FOREIGN KEY dans la colonne « P\_ID » lorsque la table stagiaire est déjà créée :

17.PNG

Pour autoriser la dénomination d’une contrainte FOREIGN KEY et pour définir une contrainte FOREIGN KEY sur plusieurs colonnes :

18.PNG

Pour supprimer une contrainte FOREIGN KEY :

19.PNG

Pour créer une contrainte CHECK dans la colonne « Age » lorsque la table est déjà créée :

20.PNG

Pour autoriser la dénomination d’une contrainte CHECK sur plusieurs colonnes :

21.PNG

Pour supprimer une contrainte CHECK :

22.PNG

Pour créer une contrainte DEFAULT dans la colonne « Ville » lorsque la table est déjà créée :

23.PNG

Pour supprimer une contrainte DEFAULT :

24.PNG

Pour créer INDEX :

25.PNG

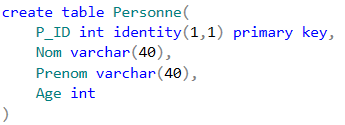
Pour créer UNIQUE INDEX :

26.PNG

L’instruction DROP INDEX est utilisée pour un index dans une table :

27.PNG

L’instruction SQL suivante définit la colonne « P\_ID » comme un champ de clé primaire auto-incrément dans la table « Personne » :



Pour insérer un nouvel enregistrement dans le tableau « Personne », nous ne devrons pas spécifier de valeur pour la colonne « P\_ID » (une valeur unique sera ajoutée automatiquement) :

29.PNG

L’instruction suivante sélectionne tous les enregistrements dans la table « Personne » :

30.PNG

Les commandes SQL les plus importantes :

SELECT : Extrait les données d’une base de données.

UPDATE : Met à jour les données.

DELETE : Supprime les données.

INSERT INTO : Insère de nouvelles données.

CREATE DATABASE : Crée nouvelle base de données.

ALTER DATABASE : Modifie une base de données.

CREATE TABLE : Crée une nouvelle table.

ALTER TABLE : Modifie une table.

DROP TABLE : Supprime une table.

CREATE INDEX : Crée un index (clé de recherche).

DROP INDEX : Supprime un index.

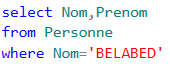
SELECT :

31.PNG

SELECT DISTINCT : est utilisée pour renvoyer uniquement de valeurs distinctes (différentes).

32.PNG

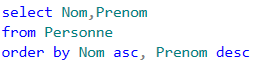
WHERE : est utilisée pour extraire uniquement les enregistrements qui remplissent une condition spécifiée.



Les opérateurs AND et OR sont utilisées pour filtrer les enregistrements en fonctions de plusieurs conditions.

L’opérateur NOT affiche un enregistrement si la ou les conditions sont fausses.

ORDER BY : trie les enregistrements par ordre croissant par défaut. Pour trier les enregistrements dans l’ordre décroissant, utilisez le mot clé DESC.



INSERT INTO : est utilisée pour insérer de nouveaux enregistrements.

29.PNG

Un champ avec une valeur NULL est un champ sans valeur.

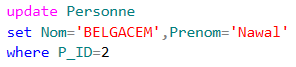
IS NULL :

35.PNG

IS NOT NULL :

36.PNG

UPDATE : est utilisée pour modifier les enregistrements existants dans une table.



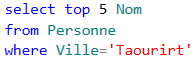
DELETE : est utilisée pour supprimer les enregistrements existants dans une table.

38.PNG

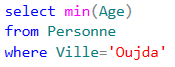
Pour supprimer tous les enregistrements :

39.PNG

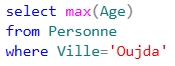
SELECT TOP : est utilisée pour spécifier le nombre d’enregistrements renvoyé.



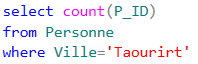
La fonction MIN ( ) renvoie la plus petite valeur de la colonne sélectionnée.



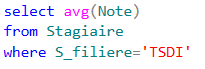
La fonction MAX ( ) renvoie la plus grande valeur de la colonne sélectionnée.



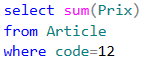
La fonction COUNT ( ) renvoie le nombre de lignes correspondant à un critère spécifié.



La fonction AVG ( ) renvoie la moyenne d’une colonne numérique.



La fonction SUM ( ) renvoie la somme totale d’une colonne numérique.

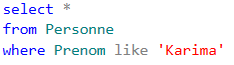


L’opérateur LIKE est utilisé dans une clause WHERE pour rechercher un modèle spécifié dans une colonne.

Deux jokers sont utilisés conjointement avec l’opérateur LIKE :

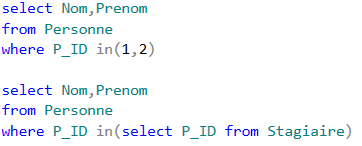
% Le signe pourcentage représente zéro, un ou plusieurs caractères.

\_ Le trait de soulignement représente un seul caractère.

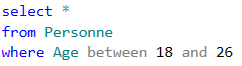


L’opérateur IN vous permet de spécifier plusieurs valeurs dans une clause WHERE.

L’opérateur IN est un raccourci pour plusieurs conditions OR.



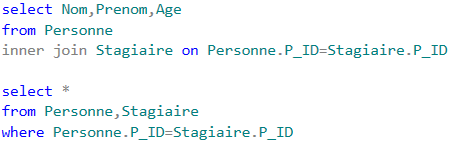
L’opérateur BETWEEN sélectionne les valeurs dans une plage donnée. Les valeurs peuvent être des nombres, du texte, ou des dates.



Alias SQL sont souvent utilisés pour donner une table ou une colonne, un nom temporaire.

49.PNG

INNER JOIN : sélectionne des enregistrements qui ont des valeurs correspondantes dans les deux tables.

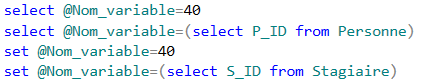


SGBD2 (Transact SQL)

Déclarer une variable :

51.PNG

Affecter une valeur à une variable :

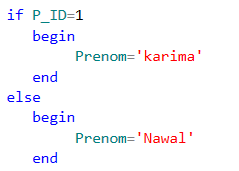


Afficher les informations :

53.PNG

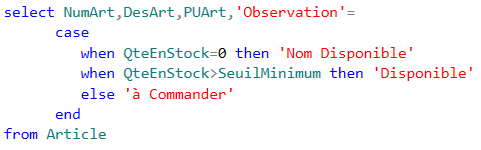
La fonction de conversion : CONVERT (type de conversion, valeur à convertir)

La structure alternative : IF…ELSE



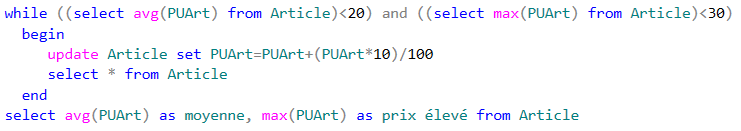
L’instruction CASE : permet d’effectuer, selon une condition, une valeur à un champ dans une requête SELECT.

Exemple :



La structure répétitive : WHILE condition BEGIN instruction END

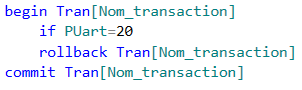
Exemple :



L’instruction IF UPDATE renvoie une valeur TRUE ou FALSE pour déterminer si une colonne à été modifié : IF UPDATE (nom colonne) BEGIN…END

L’instruction GOTO renvoie l’exécution du programme vers un point spécifique repéré par une étiquette : GOTO étiquette.

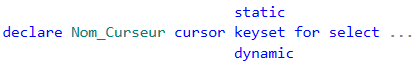
La transaction :



L’instruction RAISERROR affiche un message d’erreur système :

RAISERROR (numéro message | texte message, gravité, état [paramètre 1, paramètre 2, …])

Pour déclarer un curseur :



Pour ouvrir un curseur :

59.PNG

Pour lire un enregistrement à partir d’un curseur :

* Atteindre le premier enregistrement du curseur :

60.PNG

* Atteindre l’enregistrement du curseur suivant celui en cours :

61.PNG

Ou

62.PNG

* Atteindre l’enregistrement du curseur précédent celui en cours :

63.PNG

* Atteindre le dernier enregistrement du curseur :

64.PNG

* Atteindre l’enregistrement se trouvant à la position n dans le curseur :

65.PNG

* Atteindre l’enregistrement se trouvant à la position n de la ligne en cours :

66.PNG

La variable système @@FETCH\_STATUS est utilisée pour détecter la fin du curseur.

Pour fermer un curseur :

67.PNG

Libérer les ressources utilisées par un curseur :

68.PNG

Procédure stockée sans paramètre :

CREATE PROCEDURE nom procédure AS instructions

Exécuter : EXEC nom procédure

Procédure stockée avec des paramètres en entrée :

CREATE PROCEDURE nom procédure

Nom paramètre 1 d’entrée type donnée = valeur par défaut,

Nom paramètre 2 d’entrée type donnée = valeur par défaut,…

AS

Instructions

Exécuter : EXEC nom procédure valeur paramètre 1, valeur paramètre 2 …

Procédure stockée avec des paramètres en sortie :

CREATE PROCEDURE nom procédure

Nom paramètre 1 d’entrée type donnée = valeur par défaut,

Nom paramètre 2 d’entrée type donnée = valeur par défaut,…

Nom paramètre 1 de sortie type donnée OUTPUT,

Nom paramètre 2 de sortie type donnée OUTPUT, …

AS

Instructions

Exécuter : DECLARE variable paramètre 1 de sortie type paramètre de sortie

DECLARE variable paramètre 2 de sortie type paramètre de sortie

EXEC nom procédure valeur paramètre 1 d’entrée, valeur paramètre 2 d’entrée …

Nom paramètre 1 de sortie type donnée OUTPUT,

Nom paramètre 2 de sortie type donnée OUTPUT, …

Procédure stockée avec valeur de retour :

CREATE PROCEDURE nom procédure

AS instructions

RETURN valeur de sortie

Exécuter : DECLARE variable de retour Type de variable de retour

EXEC variable de retour=nom procédure

Cryptage d’une procédure stockée :

CREATE PROCEDURE …………WITH ENCRYPRTION AS instructions

Pour supprimer une procédure :

DROP PROCEDURE nom procédure

Pour modifier une procédure :

ALTER PROCEDURE nom procédure AS nouvelles instructions

Fonction qui retourne une valeur scalaire :

CREATE FUNCTION nom fonction (nom paramètre 1 type donnée,..)

RETURNS type de retour

AS instructions

RETURN valeur

Fonction qui retourne une table :

CREATE FUNCTION nom fonction (nom paramètre 1 type donnée,..)

RETURNS nom table Table (champ 1 type 1, …)

AS select …

RETURN

Pour créer un déclencheur (trigger) :

CREATE TRIGGER nom trigger

ON nom table

INSTEAD OF / FOR opération 1, opération 2, …

AS instructions

Pour supprimer un trigger :

DROP TRIGGER nom trigger

Pour modifier un trigger :

ALTER TRIGGER nom trigger

ON nom table

FOR opération 1, opération 2, …

AS nouvelles instructions

\_\_ Bon courage \_\_

\_\_ Karima BELABED \_\_