**UNIVERSITE DE L’ASSOMPTION DU CONGO DE BENI**

**UAC/BENI**

**REPUBLIQUE DEMOCRATIQUE DU CONGO**

**FACULTE D’INFORMATIQUE APPLIQUEE A LA GESTION DES ENTREPRISES**

****

**TRAVAIL PRATIQUE DE BASES DE DONNEES**

**Par MUYISA VAGHENI EXAUCE**

**2024-2025**

**A.**

1. **FOREIN KEY (Clé étrangère)** : Supposons qu’on a deux tables, **commandes** et **clients**, où **commandes** contient des colonnes telles que id\_commande, id\_client, date\_commande, etc., et **clients** contient des colonnes comme id\_client, nom\_client, email, etc. Pour garantir que chaque id\_client dans la table **commandes** corresponde à un id\_clientexistant dans la table **clients**, vous définiriez une contrainte de clé étrangère sur la colonne id\_client de la table **commandes**, qui fait référence à la colonne id\_client de la table **clients**.

Exemple de la requête :

* **CREATE TABLE** commandes ( id\_commande **INT(50) PRIMARY KEY**, id\_client **INT(20)**, date\_commande **DATE**, **CONSTRAINT** fk\_id\_client **FOREIGN KEY** (id\_client) **REFERENCES** clients(id\_client) );

**2. NOT NULL (non nul)** : Supposons que vous ayez une table **employes** avec des colonnes telles que id\_employe, nom, prenom , email, etc., et vous souhaitez vous assurer que le email de chaque employé est obligatoire (non nul).

Exemple de la requête :

* **CREATE TABLE** employes ( id\_employe **INT(11) PRIMARY KEY**, nom **VARCHAR(255)** **NOT NULL**, postnom **VARCHAR(255)** **NOT NULL**, email **VARCHAR(255) NOT NULL** );

**3. UNIQUE (unique)** : Supposons que vous ayez une table **etudiants** avec des colonnes telles que id\_etudiant, nom, postnom, etc., et vous voulez vous assurer qu'aucun nom d'étudiant n'est répété.

Exemple de la requête :

* **CREATE TABLE** etudiants ( id\_etudiant **INT(30) PRIMARY KEY**, nom **VARCHAR(255) NOT NULL**, last\_name **VARCHAR(255) NOT NULL,** **UNIQUE(**nom, postnom**)** );

**4. CHECK (Vérifier)** : Supposons que vous ayez une table **employes** avec une colonne salaire où vous voulez vous assurer que le salaire de chaque employé est supérieur à 20000.

* **CREATE TABLE** employes ( id\_employe **INT(30) PRIMARY KEY**, nom **VARCHAR(255) NOT NULL,** postnom **VARCHAR(255) NOT NULL**, salaire **DECIMAL(10, 2)** **CHECK (**salary > 20000) **)**;

1. Liste des instructions :

* Pour créer un synonyme :
* **CREATE SYNONYM** nom\_synonyme **FOR** nom\_objet;
* Pour créer un indexe:
* **CREATE INDEX** nom\_index **ON** nom\_table (nom\_colonne);
* Pour créer une sequence:
* **CREATE SEQUENCE** nom\_sequence **START WITH** valeur\_depart **INCREMENT BY** valeur\_incrementation;

1. Quelques fonctions de groupes de SQL :
2. **COUNT() :** Cette fonction compte le nombre de lignes dans un groupe.

Exemple : Compter le nombre d'étudiants par classe.

1. **SUM() :** Cette fonction calcule la somme des valeurs d'une colonne dans un groupe.

Exemple : Calculer le total des ventes par produit.

1. **AVG() :** Cette fonction calcule la moyenne des valeurs d'une colonne dans un groupe.

Exemple : Calculer la moyenne des notes par matière.

1. **MIN() :** Cette fonction retourne la valeur minimale d'une colonne dans un groupe.

Exemple : Trouver la note minimale par étudiant.

1. **MAX() :** Cette fonction retourne la valeur maximale d'une colonne dans un groupe.

Exemple : Trouver la note maximale par étudiant.

1. **GROUP\_CONCAT() :** Cette fonction concatène les valeurs d'une colonne dans un groupe en une seule chaîne de caractères.

Exemple : Obtenir la liste des hobbies séparés par des virgules par étudiant.