Algèbre relationnelle & Langage SQL

Bases de données

Chérif Bachir DEME Enseignant chercheur en Cryptologie à l'UADB

Chapitre 2 : Langage SQL

Objectifs spécifiques : A la suite de ce chapitre, l'étudiant doit être capable de:

1. Comprendre le langage SQL
2. Identifier les différents sous langages de SQL
3. Créer la structure d'une table
4. Modifier la structure d'une table
5. Supprimer une tables
6. Définir les différentes contraintes d'intégrités
7. Créer les contraintes d'intégrités
8. Modifier les contraintes d'intégrités
9. Définir une requête
10. Effectuer des requêtes simples ou complexes
11. Ajouter des colonnes

Le langage SQL

Qu'est-ce que le langage SQL ?

- Le langage SQL (Structured Query Language) est un langage d'interrogation d'une base de données.
- Il permet la définition, la manipulation et le contrôle des bases de données relationnelles.
- Il est composé de cinq sous-langages :

- Langage de Définition de Données (LDD) : création, modification et suppression des définitions des tables.
- Langage de Manipulation de Données (LMD) : ajout, suppression, modification et interrogation des données.
- Langage d'Interrogation de Données (LID) : L'exploitation des données.
- Langage de Contrôle de Données (LCD) : gestion des accès multiutilisateur aux données. (Traité dans le chapitre suivant)
- Langage de Contrôle de Données (LCT) : gestion des transactions (validation et annulation des transactions). (Traité dans le chapitre suivant)

Le langage SQL

Langage de Définition de Données (LDD)

- Le langage de définition de données permet de définir les objets de la base de données.
- Il repose essentiellement sur trois mots clés :

  - CREATE pour créer les objets ;
  - DROP pour supprimer les objets ;
  - ALTER pour modifier les objets .

Le langage SQL

Les tables

- **Création**

```
CREATE TABLE nom_table (
                                attribut1   type1,
                                attribut1   type1,
                                attribut1   type1,
                                Contrainte nom_contrainte_primaire,
                                Contrainte nom_contrainte_etrangere,
                                Contrainte nom_contrainte_domaine
  );
```

Le langage SQL

Les tables

- **Création**

Chaque attribut (ou colonne) appartient à un type et on en rencontre plusieurs parmi lesquels:

Types de données	Description
INT(n)	Entier à n chiffres
NUMERIC(n, m)	Réel à n chiffres
VARCHAR(n)	Chaîne de n caractères
DATE	JJ-MM-AAAA ou AAAA-MM-JJ
BLOB	Les objets binaire de grande taille (son, vidéo, image)

Le langage SQL

Les contraintes d'intégrité

- **Clé primaire:** est l'attribut permet d'identifier chaque enregistrement d'une table de manière unique.

Syntaxe : CONSTRAINT nom_contrainte PRIMARY KEY
(attribut_clé1, attribut_clé2,….)

- **Clé étrangère :** est une clé primaire provenant d'une autre table.

Syntaxe : CONSTRAINT nom_contrainte FOREIGN KEY
(attribut_clé) REFERENCES table(attribut)

- **Contrainte de domaine** : est une contrainte de recherche

Syntaxe : CONSTRAINT nom_contrainte CHECK (condition)

Le langage SQL

Exemple

```
CREATE TABLE Client (
                        numClient int,
                        nom  varchar(15),
                        prenom varchar(20),
                        adresse varchar(30),
                        Constraint Client_pk PRIMARY KEY(numClient),
);
:
```

Le langage SQL

Les tables

- **Modifications de la structure d'une table**

Elle peut se faire comme suit:

- **Par l"ajout d'un attribut**

ALTER TABLE nom_table ADD (lattribut_concerné TYPE);

Exemple : ALTER TABLE Client ADD (email varchar(30));

- **Par modification du nom ou type d'un attribut**

ALTER TABLE nom_table MODIFY(attribut_concerné TYPE);

Exemple : ALTER TABLE Client MODIFY (adresse varchar(30));

- **Par suppresion d'un attribut**

ALTER TABLE nom_table DROP COLUMN attribut_concerné;

Exemple : ALTER TABLE Client DROP COLUMN adresse;

Le langage SQL

Les tables

- **Modifications de la structure d'une table**

Elle peut se faire comme suit:

- **Par l"ajout d'un d'une contrainte**

ALTER TABLE nom_table ADD CONSTRAINT nom_contraine

TYPE_de_contrainte(nom_attribut);

Exemple : ALTER TABLE Client ADD CONSTRAINT Client_pk PRIMARY KEY(numClient);

- **Par suppresion d'une contrainte**

ALTER TABLE nom_table DROP CONSTRAINT nom _contrainte;

Exemple : ALTER TABLE Client DROP CONSTRAINT numClient;

Les tables

- **Copie et suppression (ou destruction) d'une table**

- Copie d'une table

CREATE TALE nom_table_copie AS requête;

Exemple : CREATE TALE ClientBIS AS SELECT * FROM Client;

- Suppression d'une table

DROP TABLE nom_table ;

Exemple : DROP TABLE Client;

Le langage SQL

Langage de Manipulation de Données (LMD)

- Le langage de manipulation de données permet de manipuler les données de la base de données. Il repose essentiellement sur trois mots clés :
  - INSERT pour insérer les données ;
  - UPDATE pour mettre à jour les données ;
  - DELETE pour supprimer les données ;

Langage d'Interrogation de Données (LID)

- Le langage d'interrogation de données permet d'interroger les données à l'aide du mot clés:
  - SELECT pour interroger les données.

Le langage SQL

Insertion, mis à jour et suppression de données

- **Insertion de données dans une table : enregistrement**

INSERT INTO TABLE nom_table VALUES((Valeur_attribut1, Valeur_attribut2, …);

Exemple : INSERT INTO TABLE Client VALUES((1, ‘Diop’, ‘Awa’, ‘Pikine’);

- **Mise à jour de la valeur d’un attribut**

UPDATE nom_table SET attribut=nouvelle_valeur [WHERE condition];

Exemple : UPDATE Client SET adresse=‘Ouakam’ WHERE numClient=1;

- **Suppression d’un enregistrement**

DELETE FROM nom_table [WHERE condtion];

Exemple : DELETE FROM Client WHERE numClient =1;

Interrogation des données

- **Forme générale**

SELECT [ALL | DISTINCT] liste_attributs1 FROM liste_tables

[WHERE condition1]

[GROUP BY liste_attributs2 [HAVING condition2]

[ORDER BY liste_attributs3 [ASC | DESC]];

- **Afficher de tous les n-uples d'une table**

SELECT * FROM nom_table;

Exemple : SELECT * FROM Client;

Le langage SQL

Interrogation des données

- **Une projection**

SELECT liste_attributs1 FROM nom_table;

Exemple : SELECT nom, prenom FROM Client;

- **Une sélection**

SELECT * FROM nom_table WHERE condition;

Exemple : SELECT nom, prenom FROM Client WHERE adresse='Pikine';

- **Le renommage**

SELECT attribut AS nouveau_nom FROM nom_table;

Exemple : SELECT adresse AS 'habite à' FROM Client;

Interrogation des données

- **Union de requêtes**

```
SELECT  liste_attributs1 FROM nom_table1
UNION
SELECT  liste_attributs1 FROM nom_table2 ;
```

- **Intersection de requêtes**

```
SELECT *  FROM nom_table1 INTERSECT SELECT *  FROM nom_table2 ;
```

- **La différence**

```
SELECT *  FROM nom_table1 MINUS SELECT *  FROM nom_table 2;
```

Le langage SQL

Interrogation des données

- **Produit cartésien**

SELECT liste_attributs FROM nom_table1, nom_table2 ;

Exemple : SELECT * FROM Client, Commandes;

- **La jointure**

SELECT liste_attributs FROM nom_table1 NATURAL JOIN nom_table 2;

Exemple : SELECT * FROM Client NATURAL JOIN Commandes;

- **La jointure externe**

SELECT liste_attributs FROM nom_table1 OUTER JOIN nom_table 2 ON

nom_table1.id_table1=nom_table2.id_table1_fk;

Exemple :SELECT * FROM Client OUTER JOIN Commandes ON

Client.numClient=Commandes.numClient;

Interrogation des données

- **La jointure externe gauche**

SELECT liste_attributs FROM nom_table1 LEFT OUTER JOIN nom_table2 ON nom_table1.id_table1=nom_table2.id_table1_fk;

Exemple : SELECT * FROM Client LEFT OUTER JOIN Commandes

ON Client.numClient=Commandes.numClient;

- **La jointure externe droite**

SELECT liste_attributs FROM nom_table1 RIGHT OUTER JOIN nom_table2 ON nom_table1.id_table1=nom_table2.id_table1_fk;

Exemple : SELECT * FROM Client RIGHT OUTER JOIN Commandes

ON Client.numClient=Commandes.numClient;

Application

- Soit le schéma relationnel suivant :

Acheteur(idAcheteur, nom, prenom, tel, adresse email,#ninea)

Vendeur(numVendeur, nom, prenom, adresse, email, tel,#ninea)

Articles(numArt, designation, prix_unitaire, quantitéStock,#ninea)

Boutique(ninea, adresse, chiffreAffaire, tel,)

Travail à faire: Donner les commandes SQL correspondant aux requêtes suivantes:

1. La liste des vendeurs.
2. La liste des potentiels acheteurs
3. La liste de tous articles d'une boutique
4. La liste de toutes les boutiques
5. Les articles qui ont été achetés
6. Les articles vendus
7. Les articles achetés et vendus par la même personne c'est-à-dire l'acheteur est le vendeur

Application

8. Les vendeurs qui vendent les mêmes articles
9. Les acheteurs habitant Pikine.
10. Les vendeurs et acheteurs ayant la même adresse
11. Les articles dont le stock est inférieur à 10.
12. Les articles achetés au vendeur Abdou Faye.
13. Les articles achetés à la boutique 18 et dont le prix_unitaire est égale à 15 000.
14. Le vendeur Doudou Gueye qui a vendu l'article puce orange 4g, à l'acheteur Alimatou BA
15. Les boutiques situées dans la même zone que ses acheteurs et vendeurs.
16. Les Vendeurs portant le Diop.
17. Les achats dont le montant est compris entre 150 000 et 250 000
18. Les articles dont le nom commence par C et O.
19. Les boutiques dont le chiffre d'affaire dépasse les 1 000 000
20. Les boutiques qui ont vendus beaucoup plus d'articles.