DML

Data Manipulation Language

Introduction aux requêtes DML

Les requêtes DML, ou langage de manipulation de données, sont utilisées pour manipuler les données stockées dans une base de données.

Les requêtes DML permettent d'insérer de nouvelles données, de mettre à jour des données existantes et de supprimer des données.

Les requêtes DML modifient le contenu des tables et permettent aux utilisateurs d'interagir avec les données de la base.

Insertion des données dans une table

Lorsque la table possède des colonnes auto incrémentées (GENERATED ALWAYS AS IDENTITY) ou des colonnes attachées à des séquences, il n'est pas nécessaire de fournir des valeurs pour ces colonnes.

```
CREATE TABLE names (
    id INT GENERATED ALWAYS AS IDENTITY,
    nom VARCHAR(20),
    CONSTRAINT pk_names_id PRIMARY KEY (id)
);
```

Exemples d'insertions de valeurs.

```
INSERT INTO names VALUES (DEFAULT, 'James');

INSERT INTO names (nom) VALUES ('Jane');

INSERT INTO names (id, nom) VALUES (DEFAULT, 'John');
```

Insertion des données dans une table

Considérons la table suivante

```
CREATE TABLE dictionnaire (
    cle VARCHAR(20),
    valeur VARCHAR(20)
);
```

	cle character varying (20)	valeur character varying (20)
1	Bonjour	Hello
2	[null]	?
3	Pas de valeur	[null]

Voici les requêtes d'insertion

```
INSERT INTO dictionnaire VALUES ('Bonjour', 'Hello');

INSERT INTO dictionnaire VALUES (DEFAULT, '?');

INSERT INTO dictionnaire (cle) VALUES ('Pas de valeur');
```

Insertion des données dans une table

Nous pouvons également insérer plusieurs lignes à l'aide d'une commande `INSERT INTO`.

Exemple

```
INSERT INTO employees (name, age, salary, department_id)
   VALUES ('AB De-Villiers', 30, 5000.00, 1),
  ('Steve Smith', 28, 6000.00, 1),
  ('Michael Johnson', 35, 8000.00, 2),
  ('Ollie Robinson', 32, 5500.00, 2);
```

Insertion de plusieurs lignes à partir d'un SELECT sur une autre table

```
INSERT INTO Customers (CustomerName, City, Country)
  SELECT SupplierName, City, Country FROM Suppliers;
```

Mise à jour des données d'une table

Pour mettre à jour les données d'une table, vous pouvez la clause UPDATE.

Cette clause est de la forme suivante

```
UPDATE <nom de la table à mettre à jour>
SET <les nouvelles valeurs à attribuer>
WHERE <condition>;
```

Exemple

```
UPDATE matiere
SET libelle='Français'
WHERE id=1;
```

La clause WHERE est facultative, elle permet de restreindre les enregistrements à modifier.

Mise à jour des données d'une table

Nous pouvons mettre à jour les données d'une table en référençant les données d'une autre table.

Dans cet exemple toutes les notes de François Pignon sont fixées à 20.

```
UPDATE notes AS n
SET note = 20
FROM eleves e
WHERE e.id = n.id_eleve AND e.nom='Pignon' AND e.prenom='François';
```

Supprimer des lignes d'une table

Pour supprimer des lignes d'une table, vous devez utiliser la clause DELETE FROM.

Cette clause est de la forme suivante

```
DELETE FROM nom_table
WHERE condition;
```

Exemple

```
DELETE FROM eleves
WHERE nom='Pignon' AND prenom='François';
```

Exemple de suppression avec référence à une autre table utilisant le mot clé USING de PostgreSQL.

```
DELETE FROM notes n
USING eleves e
WHERE e.id = n.id_eleve AND e.nom='Pignon' AND e.prenom='François';
```