

M164 – Gestion des contraintes

F. Mauron

EPAI

31 mai 2024

- 1 Introduction
- 2 Type de contraintes en SQL
- 3 Clauses ON DELETE et ON UPDATE

- 1 Introduction
- 2 Type de contraintes en SQL
- 3 Clauses ON DELETE et ON UPDATE

Les contraintes sont des règles définies sur les données dans une base de données relationnelle pour garantir leur intégrité, leur cohérence et leur validité. Elles sont utilisées pour restreindre les valeurs pouvant être stockées dans une colonne ou une table.

- 1 Introduction
- 2 Type de contraintes en SQL
- 3 Clauses ON DELETE et ON UPDATE

Contrainte de Clé Primaire (**PRIMARY KEY**) : Identifie de manière unique chaque enregistrement dans une table.

Contrainte de Clé Étrangère (**FOREIGN KEY**) : Établit une relation entre les clés de deux tables, assurant l'intégrité référentielle.

Contrainte d'unicité (**UNIQUE**) : Assure que toutes les valeurs dans une colonne ou un groupe de colonnes sont uniques.

Contrainte de Vérification (**CHECK**) : Permet de spécifier une condition pour chaque ligne dans une table.

Contrainte PRIMARY KEY

```
-- Exemple de création d'une table avec une contrainte de clé primaire
CREATE TABLE Employes (
    ID_Employe INT PRIMARY KEY,
    Nom VARCHAR(50),
    Age INT,
    Salaire DECIMAL(10, 2)
);
```

Contrainte FOREIGN KEY

-- Exemple de création d'une table avec une contrainte de clé étrangère

```
CREATE TABLE Commandes (  
    ID_Commande INT PRIMARY KEY,  
    ID_Client INT,  
    Date_Commande DATE,  
    Montant DECIMAL(10, 2),  
    FOREIGN KEY (ID_Client) REFERENCES Clients(ID_Client)  
);
```



```
-- Exemple de création d'une table avec une contrainte d'unique  
CREATE TABLE Utilisateurs (  
    ID_Utilisateur INT PRIMARY KEY,  
    Nom_Utilisateur VARCHAR(50) UNIQUE,  
    Mot_de_Passe VARCHAR(50)  
);
```

-- Exemple de création d'une table avec une contrainte de vérification

```
CREATE TABLE Employes (  
    ID_Employe INT PRIMARY KEY,  
    Nom VARCHAR(50),  
    Age INT,  
    Salaire DECIMAL(10, 2),  
    CHECK (Age >= 18)  
);
```

- Utiliser des noms de contraintes significatifs pour faciliter la maintenance et le débogage.
- Éviter d'utiliser des contraintes CHECK trop complexes qui pourraient impacter les performances.

- 1 Introduction
- 2 Type de contraintes en SQL
- 3 Clauses ON DELETE et ON UPDATE**

Les clauses **ON DELETE** et **ON UPDATE** sont **utilisées** dans les contraintes de clé étrangère (**FOREIGN KEY**) pour spécifier le comportement des données liées lorsqu'une opération de suppression ou de mise à jour est effectuée sur la table parente.

La clause ON DELETE est utilisée pour spécifier l'action à effectuer sur les enregistrements liés dans la table enfant lorsque des enregistrements correspondants sont supprimés dans la table parente.

Il existe plusieurs options pour l'action à effectuer :

- **CASCADE** : Lorsqu'un enregistrement parent est supprimé, tous les enregistrements liés dans la table enfant sont également supprimés.
- **SET NULL** : Les valeurs des colonnes étrangères dans la table enfant sont définies sur NULL lorsque l'enregistrement parent est supprimé.
- **RESTRICT** : Empêche la suppression de l'enregistrement parent si des enregistrements liés existent dans la table enfant (par défaut).
- **NO ACTION** : Identique à RESTRICT, empêche la suppression de l'enregistrement parent si des enregistrements liés existent dans la table enfant.

La clause ON UPDATE est utilisée pour spécifier l'action à effectuer sur les enregistrements liés dans la table enfant lorsque des enregistrements correspondants dans la table parente sont mis à jour. Les options sont similaires à celles de ON DELETE.

- **CASCADE** : Lorsqu'un enregistrement parent est mis à jour, les valeurs correspondantes dans la table enfant sont également mises à jour.
- **SET NULL** : Les valeurs des colonnes étrangères dans la table enfant sont définies sur NULL lorsque la valeur dans la table parente est mise à jour.
- **RESTRICT** : Empêche la mise à jour de la valeur de la clé primaire de la table parente si des enregistrements liés existent dans la table enfant (par défaut).
- **NO ACTION** : Identique à RESTRICT, empêche la mise à jour de la valeur de la clé primaire de la table parente si des enregistrements liés existent dans la table enfant.

```
CREATE TABLE Commandes (  
    ID_Commande INT PRIMARY KEY,  
    ID_Client INT,  
    Date_Commande DATE,  
    FOREIGN KEY (ID_Client)  
        REFERENCES Clients(ID_Client)  
        ON DELETE RESTRICT  
        ON UPDATE CASCADE  
);
```