Introduction aux Bases de Données

Ulysse COUTAUD
coutaudu@gmail.com

Modifier des données

Insérer un enregistrement

Clause INSERT INTO

INSERT INTO table (nom_colonne_1, nom_colonne_2, ...)
VALUES ('valeur 1', 'valeur 2', ...)

BDD "fabrique" :

Ajouter un nouvel employé dans le service "Production"
Ajouter un service "Production Après-Midi" dont les horaires seront 13h-20h.
Ajouter les horaires 16h-5h pour le service "Astreinte"
Ajouter un nouvel employé "Abdoul" au service "Production Après-Midi".

Modifier des enregistrements

Clause UPDATE

UPDATE table SET
colonne_1 = 'valeur 1',
colonne_2 = 'valeur 2', ...
WHERE conditions;

BDD "fabrique" :

Augmenter le salaire des employes de "Astreinte" de 30%
L'employé Tom change de service et passe à l'après midi

Supprimer des enregistrements

BDD "atelier":
employes(prenom,salaire,adresse,service) horaires(service,debut,fin)

Clause DELETE FROM

DELETE FROM table WHERE conditions;

BDD "fabrique" :

Supprimer le service de production après-midi.

Modifier le schéma

Créer une table

CREATE TABLE

CREATE TABLE *ma_table* (
colonne1 TEXT,
colonne2 INTEGER
);

BDD "fabrique" :

Ajouter une table "ordres_fabrications" avec 4 colonnes: id (le numéro d'OF), employe (l'employé qui va produire l'OF), reference_piece (la référence de pièce à produire)

Supprimer une table

DROP TABLE

DROP TABLE *ma_table*;

BDD "fabrique" :

Supprimer la table horaire.

Modifier une table

ALTER TABLE

ALTER TABLE ma_table mon_action ;

Avec mon_action:

- ADD ma_colonne INTEGER/TEXT/...
- DROP ma_colonne

BDD "fabrique" :

Ajouter la colonne quantite (le nombre de pièces à produire) à la table ordre_fabrication.

Les contraintes

Les contraintes

- Règles qui restreignent les données dans la BDD.
- Chaque contrainte est associée à une table.
- Bloque tout ajout/modification qui viole les contraintes.
- Garantissent **certains** aspects de la cohérence de la BDD.

Clé Primaire

PRIMARY KEY

CONSTRAINT *ma_clé_primaire* PRIMARY KEY (*colonne1*)
Unique et non nulle.
Identifie l'enregistrement (cad la ligne).

BDD "fabrique" :
Modifier le nom de Peter en John. Que ce passe il ?

Remarque: Si la clé primaire n'est pas définie à la création de la table ...

Clé Primaire

PRIMARY KEY

CONSTRAINT *ma_clé_primaire* PRIMARY KEY (*colonne1*)
Unique et non nulle.
Identifie l'enregistrement (cad la ligne).

BDD "fabrique" :
Modifier le nom de Peter en John. Que ce passe il ?

Remarque: Si la clé primaire n'est pas définie à la création de la table ... c'est l'ensemble des colonnes, cad la ligne entière.

Clé Secondaire

UNIQUE

CONSTRAINT *ma_clé_secondaire* UNIQUE (*colonne1*)
Unique mais peut être nulle.

En analysant le script de création de la BDD fabrique, trouver une clé secondaire de la table employes et expliquer sont sens.

Clé étrangère

FOREIGN KEY

CONSTRAINT *ma_clé_étrangère* FOREIGN KEY (colonne1)
REFERENCES table_etrangère(colonneX))

Assure l'*intégrité référentielle*.

Existe nécessairement dans la table de référence.

Et si la clé de référence disparait ?

Trois options:

- ON DELETE RESTRICT
- ON DELETE CASCADE
- ON DELETE SET NULL

ATTENTION: Pointe nécessairement vers une clé primaire.

Check

CHECK

CONSTRAINT *mon_check* CHECK (condition)

Remarque: Même genre de conditions que dans un WHERE. *En*

lisant le fichier fabriqueBDD.sql: quelle contrainte CHECK existe sur la table employes ?
Quelle contrainte devrait on ajouter sur la colonne salaire ?

Les transactions

Le problème

```
UPDATE accounts SET balance = balance - 100.00
    WHERE name = 'Alice';

UPDATE branches SET balance = balance - 100.00
    WHERE name = (SELECT branch_name
                  FROM accounts
                  WHERE name = 'Alice');

UPDATE accounts SET balance = balance + 100.00
    WHERE name = 'Bob';

UPDATE branches SET balance = balance + 100.00
    WHERE name = (SELECT branch_name
                  FROM accounts
                  WHERE name = 'Bob');
```

La solution

Je valide une transaction avec COMMIT

```
BEGIN;
UPDATE accounts SET balance = balance - 100.00
    WHERE name = 'Alice';
-- etc etc
COMMIT;
```

J'annule une transaction avec COMMIT

```
BEGIN;
UPDATE accounts SET balance = balance - 100.00
    WHERE name = 'Alice';
-- etc etc
ROLLBACK;
```

La solution

Le SGBDR garantit des transactions ACID:

- **A**tomicité
- **C**ohérence
- **I**solation
- **D**urabilité