

Conception des Systèmes d'Informations

CH3 : SQL : contraintes d'intégrité

Hela KADRI

Hela.kadri@univ-lille.fr

Création de tables

```
CREATE TABLE nom_table(  
    attribut1 type [contrainte],  
    attribut2 type [contrainte],  
    ...  
    [contraintes]);
```



Remplissage de tables

```
INSERT INTO nom_table VALUES(valeur1, valeur2, ...);
```

```
INSERT INTO nom_table (attribut2, attribut1, ...)  
VALUES(valeur2, valeur1, ...);
```

- Si la liste des attributs n'est pas précisée, l'ordre de création doit être respecté
- Si les valeurs de certains attributs ne sont pas précisées, elles sont remplies avec NULL



Différents niveaux de contraintes

- Contraintes liées à un attribut = contraintes de domaine
 - Interdiction des NULL
 - Valeur par défaut
 - Restriction d'un attribut à certaines valeurs
- Contraintes liées à une table = contraintes d'intégrité d'entité
 - Unicité (valeurs uniques pour un attribut ou un tuple dans une table)
 - Primalité (clés primaires)
- Contraintes liées à plusieurs tables = contrainte d'intégrité référentielle
 - Cohérence entre les clés étrangères et primaires



Contraintes de domaine

L'attribut est obligatoire : NOT NULL

- L'attribut est obligatoire, il a forcément une valeur

MLD

Etudiant(numEtud, nom, prenom)

Un étudiant doit avoir un nom et un prénom renseignés

SQL

```
CREATE TABLE Etudiant(  
    numEtud INTEGER,  
    nom VARCHAR(100) NOT NULL,  
    Prenom VARCHAR(100) NOT NULL);
```

```
INSERT INTO Etudiant VALUES(42000000, 'Pupin');
```

Erreur SQL :

ERREUR: une valeur NULL viole la contrainte NOT NULL de la
colonne « prenom »

DETAIL: La ligne en échec contient (42000000, Pupin, null)⁶



Une valeur par défaut : DEFAULT

- Une valeur par défaut est donnée à un attribut, si aucune valeur n'est précisée lors de l'insertion

MLD

Evaluer(numEtud#, numUE#, note, date)

SQL

```
CREATE TABLE Evaluer(  
    numEtud INTEGER,  
    numUE INTEGER,  
    note SMALLINT  
    date DATE DEFAULT current_date);
```

La date par défaut est la date du jour.

```
INSERT INTO Evaluer VALUES(42000000, 1, 18);
```

numetud	numue	note	date
42000000	1	18	2021-02-14

L'insertion a été faite le 14 février 2021



Condition d'un attribut : CHECK

- Les valeurs d'un attribut sont restreintes : `CHECK (attribut CONDITION)`

MLD

Evaluer(numEtud#, numUE#, note, date)

Une note doit être positive

SQL

```
CREATE TABLE Evaluator(  
    numEtud INTEGER,  
    numUE INTEGER,  
    note SMALLINT CHECK (note>=0),  
    date DATE DEFAULT current_date);
```

```
INSERT INTO Evaluator VALUES (42000000, 1, -1, '2021-01-25');
```

Erreur SQL :

ERREUR: la nouvelle ligne de la relation « evaluer » viole la
contrainte de vérification « evaluer_note_check »

DETAIL: La ligne en échec contient (42000000, 1, -1, 2021-01-25)



Contraintes d'intégrité liée à une table

Unicité d'un attribut : UNIQUE

- Toutes les valeurs prises par un attribut ou un tuple d'attribut sont différentes les unes des autres

MLD

UE(numUE, nom)

SQL

```
CREATE TABLE UE (  
    numUE SERIAL,  
    nom VARCHAR(200) UNIQUE);
```

Deux (ou plus) UE ne peuvent pas avoir le même nom

```
INSERT INTO UE VALUES (21, 'BDD');
```

```
INSERT INTO UE VALUES (22, 'BDD');
```

numue	nom
21	BDD

Erreur SQL :

ERREUR: la valeur d'une clé dupliquée rompt la contrainte unique « ue_nom_key »

DETAIL: La clé « (nom)=(BDD) » existe déjà.



Clé primaire

- Unique et non NULL
- Soit un attribut seul, soit un tuple d'attributs : Une seule clé primaire par table
- Identifiant de la donnée

MLD

UE(numUE, nom)

numUE est la clé primaire de UE et est incrémentée automatiquement.

```
INSERT INTO UE (nom) VALUES ( 'BDD1' );
```

```
INSERT INTO UE (nom) VALUES ( 'BDD2' );
```



numue	nom
1	BDD1
2	BDD2

SQL

```
CREATE TABLE UE (  
    numUE SERIAL PRIMARY KEY,  
    nom VARCHAR(200) UNIQUE);
```

Colonne	Type	NOT NULL	Défaut	Contraintes
numue	integer	NOT NULL	nextval('ue_numue_seq'::regclass)	
nom	character varying(200)			1

Clé primaire


MLD

Etudiant(numEtud, nom, prenom)

numEtu est la clé primaire de Etudiant

SQL

```
CREATE TABLE Etudiant(  
    numEtud INTEGER PRIMARY KEY,  
    nom VARCHAR(100) NOT NULL,  
    Prenom VARCHAR(100) NOT NULL);
```

Colonne	Type	NOT NULL	Défaut	Contraintes
numetud	integer	NOT NULL		
nom	character varying(100)	NOT NULL		
prenom	character varying(100)	NOT NULL		



Clé primaire

MLD

Evaluer(numEtud#, numUE#, note, date)

Le couple (numEtu, numUE) est la clé primaire de Evaluer.

SQL

```
CREATE TABLE Evaluer(  
    numEtud INTEGER,  
    numUE INTEGER,  
    note SMALLINT CHECK (note >= 0),  
    date DATE DEFAULT current_date,  
    PRIMARY KEY (numEtud, numUE) );
```



Contraintes d'intégrité référentielle

Clés étrangères

- Une table peut contenir aucune ou plusieurs clés étrangères
 - Une clé étrangère est soit un attribut seul, soit un tuple d'attributs
- Les tables et données contenant les clés primaires doivent être créées avant les tables et données contenant les clés étrangères



Clés étrangères

MLD

Evaluer(numEtud#, numUE#, note, date)

numEtu est une clé étrangère qui référence la clé primaire de la table Etudiant

numUE est une autre clé étrangère qui référence la clé primaire de la table UE

SQL

```
CREATE TABLE Evaluer(  
    numEtud INTEGER,  
    numUE INTEGER REFERENCES UE,  
    note SMALLINT CHECK (note >= 0),  
    date DATE DEFAULT current_date,  
    PRIMARY KEY (numEtud, numUE),  
    FOREIGN KEY (numEtud)  
        REFERENCES Etudiant (numEtud) );
```



Clés étrangères

```
INSERT INTO Etudiant VALUES (42000000, 'Pupin', 'Maude');
INSERT INTO Etudiant VALUES (42000001, 'Bilasco', 'Marius');
INSERT INTO UE(nom) VALUES ('BDD1');
INSERT INTO UE(nom) VALUES ('BDD2');

INSERT INTO Evaluer VALUES (42000000, 1, 18, '2021-01-25');
INSERT INTO Evaluer VALUES (42000001, 1, 18, '2021-01-25');
INSERT INTO Evaluer VALUES (42000000, 3, 15, '2021-01-18');
```

numetud	nom	prenom	numue	nom
42000000	Pupin	Maude	1	BDD1
42000001	Bilasco	Marius	2	BDD2

Erreur SQL :

ERREUR: une instruction insert ou update
sur la table « evaluer » viole la contrainte

de clé étrangère « evaluer_numue_fkey »

DETAIL: La clé (numue)=(3) n'est pas présente dans la table «
ue ».

numetud	numue	note	date
42000000	1	18	2021-01-25
42000001	1	18	2021-01-25

Modification et suppression de données

```
UPDATE nom_table SET attributI=valI WHERE condition;
```

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

- Met à jour ou efface toutes les lignes qui répondent à la clause du `WHERE`
- Ces changements doivent respecter les contraintes d'intégrité



Violation de contraintes d'intégrité référentielle

- On ne peut pas effacer un Etudiant ou une UE si il ou elle est associé.e à une note (un enregistrement de la table Evaluer).
- Un enregistrement de la table Evaluer ne peut pas exister sans l'Etudiant ou l'UE à laquelle il se réfère.

```
DELETE FROM Etudiant WHERE numEtud=42000000;
```

Erreur SQL :

ERREUR: UPDATE ou DELETE sur la table « etudiant » viole la contrainte de clé étrangère

« evaluer_numetud_fkey » de la table « evaluer »

DETAIL: La clé (numetud)=(42000000) est toujours référencée à partir de la table « evaluer ».



numetud	nom	prenom
42000000	Pupin	Maude
42000001	Bilasco	Marius

numetud	numue	note	date
42000000	1	18	2021-01-25
42000001	1	18	2021-01-25

Préserver la cohérence entre tables

- Plusieurs comportements sont disponibles pour gérer la modification de données dans des tables associées via des clés
 - Interdiction de supprimer une ligne qui contient une clé primaire si elle est déjà liée à au moins une ligne d'une autre table via une clé étrangère
 - Comportement par défaut : **ON DELETE RESTRICT**
 - La suppression d'une ligne qui contient une clé primaire provoque la suppression des lignes des autres tables qui contiennent les clés étrangères qui y sont liées
 - **ON DELETE CASCADE**
 - La suppression d'une ligne qui contient une clé primaire provoque le changement des clés étrangères qui y sont liées pour une valeur par défaut
 - **ON DELETE SET DEFAULT**
- Le comportement est porté par la clé étrangère

