

# Introduction aux Bases de données

## Cours 9 : Modification des données.

UFR 919 – 2<sup>e</sup> année Licence Informatique

# Objectifs

Cours précédent

Commandes SQL pour :

1. Création de schémas relationnels
2. Définition de contraintes d'intégrité

Ce cours

1. Mise à jour de données

# Modification de Données

Trois types de modifications

- Insertion de n-uplets
- Suppression de n-uplets
- Mises à jour de n-uplets

Chaque modification s'applique sur une base cohérente vis à vis des CI et préserve cette cohérence

# Insertion de n-uplets : Valeurs fournies

Peuplement d'une BD initialement vide

**insert into table values** ('v<sub>1</sub>', 'v<sub>2</sub>', ..., 'v<sub>n</sub>');  
l'ordre des v<sub>i</sub> respecte le schéma de *table*

**insert into table**(a<sub>1</sub>, a<sub>2</sub>, ..., a<sub>m</sub>)  
**values** ('v<sub>1</sub>', 'v<sub>2</sub>', ..., 'v<sub>m</sub>');  
l'ordre des v<sub>j</sub> respecte (a<sub>1</sub>, a<sub>2</sub>, ..., a<sub>m</sub>)  
les attributs non considérés mis à Null

**Rq** : les ' et ' facultatifs pour les types numériques

# Exemples d'insertions de valeurs fournies

```
create table Etudiant(  
    eid numeric(8),  
    nom varchar(10),  
    moy numeric(4,2)  
    primary key (eid)  
);
```

```
create table Module(  
    code varchar(10) ,  
    intitule varchar(20),  
    niveau char(2),  
    suit varchar(10),  
    primary key(code)  
);
```

```
insert into Etudiant values (1243, 'joe',12.5);  
insert into Etudiant values (1244, 'doe',null);  
insert into Module values ('L234','Techno','L2',null );  
insert into Module(code, intitule) values ('L101','logique');
```

# Insertion de n-uplets : valeurs calculées

Insérer le résultat d'une requête

**insert into** *table*

select att1, ...

from autreTable, ...

where Condition ;

# Exemples d'insertions de valeurs calculées

```
create table Etudiant(  
    eid numeric(8),  
    nom varchar(10),  
    moy numeric(4,2)  
    primary key (eid)  
);
```

```
create table MajorsPromo(  
    eid numeric(8),  
    promo char(2),  
    moy numeric(4,2),  
    primary key(eid, promo)  
    foreign key (eid) references  
        Etudiant);
```

- Insérer dans MajorsPromo le major du L2

```
insert into MajorsPromo  
    select eid, 'L2', moy  
    from Etudiant  
    where moy = (select max (moy) from Etudiant);
```

# Mise à jour de n-uplets

Utile pour modifier les valeurs d'un sous-ensemble d'attributs

**update** *table*

**set** attr = *Expression* | *valeur*

**where** *Condition* ;

où

- *Expression* peut combiner attr avec d'autres attributs de *table* ou avec des valeurs
- *Condition* est comme celle de la clause where



# Exemples de mise à jour

```
create table Etudiant(  
    eid numeric(8),  
    nom varchar(10),  
    moy numeric(4,2)  
    primary key (eid));
```

```
create table Eval(  
    eid numeric(8),  
    mid char(3),  
    note numeric(4,2),  
    primary key(eid, mid));
```

- Remonter la moyenne de tous les étudiants de 10%  
update Etudiant set moy=moy\*1.1;
- Racheter les étudiants ayant obtenu 9.5  
update Etudiant set moy=10 where moy between 9.5 and 10;
- Affecter aux étudiants ayant obtenu la plus petite moy. null  
update Etudiant set moy=null  
where moy < 5 and moy=(select min(moy) from Etudiant);

# Mise à jour de n-uplets

Utile pour modifier les valeurs d'un sous-ensemble d'attributs

**update** *table*

**set** attr = sous-requête ;

- sous-requête doit retourner un singleton et un seul attribut

# Exemples de mises à jour

```
create table Etudiant(  
    eid numeric(8),  
    nom varchar(10),  
    moy numeric(4,2)  
    primary key (eid));
```

```
create table Eval(  
    eid numeric(8),  
    mid char(3),  
    note numeric(4,2),  
    primary key(eid, mid));
```

- Renseigner la moyenne de chaque étudiant

```
update Etudiant  
set moy = (select avg(note)  
           from Eval v, Etudiant e where v.eid=e.eid);
```

Que produit cette instruction?

# Exemples de mises à jour

```
create table Etudiant(  
    eid numeric(8),  
    nom varchar(10),  
    moy numeric(4,2)  
    primary key (eid));
```

```
create table Eval(  
    eid numeric(8),  
    mid char(3),  
    note numeric(4,2),  
    primary key(eid, mid));
```

○ Renseigner la moyenne de chaque étudiant

```
update Etudiant E  
set moy = (select avg(note)  
           from Eval  
           where E.eid=Eval.eid);
```

Pourquoi utiliser une variable de corrélation?

# Suppression de n-uplets

Opération inverse de l'insertion

**delete from** *table*

**where** Condition ;

La syntaxe ressemble à celle des requêtes,

La clause where est facultative

# Exemples de suppressions

```
create table Etudiant(  
    eid numeric(8),  
    nom varchar(10),  
    moy numeric(4,2)  
    primary key (eid));
```

```
create table Eval(  
    eid numeric(8),  
    mid char(3),  
    note numeric(4,2) not null,  
    primary key(eid, mid));
```

Supprimer toutes les notes

```
delete from Eval;
```

Supprimer l'étudiant n'ayant passé aucun examen

```
delete from Etudiant
```

```
where eid not in (select eid from Eval);
```

Supprimer les lignes de Eval dont aucune note ne dépasse 7

```
delete from Eval where mid in
```

```
(select mid from Eval group by mid having max (note)<7);
```

# Mises à jour et violations CI

- Rappel : les CI doivent être préservées suite aux mises à jour
- En cas de violation d'(au moins) une CI, la mise à jour est **annulée**
- **Rq** : notions de transactions hors programme (cours de BD en L3 !)

# Insertions et violations CI

- Violation clé primaire/candidate :
  - Tentative d'insertion d'un nuplet avec la même clé (primaire ou candidate) qu'un autre nuplet
- Violation clé étrangère :
  - Tentative d'insertion d'une référence à un nuplet inexistant



# Modifications et violations CI

- Violation clé primaire/candidate :
  - Tentative de changement des valeurs d'attributs clés avec valeurs existantes
- Violation clé étrangère :
  - Tentative de changement des valeurs d'attributs de la clé étrangère avec références inexistantes
  - Tentative de changement de clés (primaire ou candidate) référencées par d'autres nuplets

# Suppressions et violations CI

- Violation clé primaire/candidate :
  - Pas possible
- Violation clé étrangère :
  - Tentative de suppression de nuplets référencés par d'autres nuplets

# Contraintes de clés étrangères et mises à jour

- Par défaut, si une MAJ tente de supprimer/modifier des nuplets référencés par d'autres elle est **annulée**
- Possibilité de spécifier un traitement alternatif
  - Propagation de la suppression/modification
  - Affectations de valeurs par défaut

# Traitement alternatif pour les clés étrangères

Syntaxe complète

**foreign key**( $a_1, \dots, a_n$ ) **references** *table* ( $b_1, \dots, b_n$ )

	<b>delete</b>	<b>restrict</b>
<b>on</b>		<b>cascade</b>
	<b>update</b>	<b>set default   null</b>

- **restrict** annule la mise à jour
- **set** permet de mettre la valeur par défaut ou null
- **cascade** propage son effet

Suppression nuplets référencés → suppression nuplets dépendants  
modification nuplets référencés → modification nuplets dépendants

# Le mode en cascade : exemple

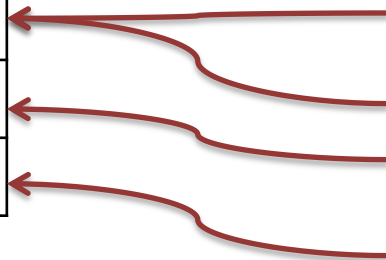
```
create table Etudiant(  
  eid numeric(8),  
  nom varchar(10),  
  moy numeric(4,2)  
  primary key (eid)  
);
```

```
create table Eval(  
  eid numeric(8),  
  mid char(3),  
  note numeric(4,2) not null,  
  primary key(eid, mid),  
  foreign key (eid) references  
  Etudiant
```

**on delete cascade,**  
**on update cascade);**

eid	nom	moy
12345	Joe	15.25
2471	Doe	13.5
1024	Lars	16

eid	mid	note
12345	L21	12
12345	L17	10
2471	L21	18
1024	L21	14



# Modification du schéma d'une BD

Le langage de définition des données permet

- Création des tables (avec contraintes d'intégrité)

- Suppression de table

  - drop table** nom [**cascade constraints**] ;

- Modification des tables

  - alter table** nom <instr\_modif\_table>

  - où instr\_modif\_table peut être

    - **rename to** autre\_nom
    - **drop** attrib
    - **add** attrib domaine

# Exemple

```
create table Etudiant(  
    eid numeric(8),  
    nom varchar(10),  
    moy numeric(4,2)  
    primary key (eid));
```

```
create table Eval(  
    eid numeric(8),  
    mid char(3),  
    note numeric(4,2),  
    primary key(eid, mid));
```

Supprimer la table Eval

```
drop table eval;
```

Renommer la table Etudiant en Etudiants

```
alter table Etudiant Rename to Etudiants;
```

Enlever l'attribut moy de la table Etudiant

```
alter table Etudiant drop moy;
```

Rajouter à un la table Etudiant attribut date naissance de type date devant être renseigné

```
alter table Etudiant add dateNaiss Date Not Null;
```