

Introduction | Création BD | Suppression BD et commentaires | Création table | Principaux types | Contraintes d'intégrités

LA DIRECTIVE AUTO_INCREMENT

- ♦ La directive **AUTO_INCREMENT** permet de spécifier qu'une colonne numérique souvent clé primaire (PRIMARY KEY) sera incrémentée automatiquement à chaque insertion d'une ligne;
- ♦ Pour appliquer l'incrémentation automatique des valeurs d'une colonne utiliser le mot clé **AUTO_INCREMENT** pendant la définition de la colonne;

Syntaxe:

```
nom_col Integer PRIMARY KEY AUTO_INCREMENT;
```

189

Introduction | Création BD | Suppression BD et commentaires | Création table | Principaux types | Contraintes d'intégrités

LA DIRECTIVE AUTO_INCREMENT

Exemple:

```
CREATE TABLE Etudiant
(
  NumInscrEt Integer PRIMARY KEY AUTO_INCREMENT,
  CinEt Char(7) UNIQUE NOT NULL,
  Nationalite VARCHAR (30) NOT NULL DEFAULT 'Marocaine' ,
  NomEt VarChar(50) NOT NULL,
  PrenomEt VarChar(50) NOT NULL,
  AdrEt VarChar(80),
  TeleEt Char(10) UNIQUE
);
```

190

Langage de Définition de Données LDD (Suite)

Modification des structures des tables

- ✓ Ajouter une colonne
- ✓ Modifier une colonne
- ✓ Supprimer une colonne
- ✓ Renommer une colonne
- ✓ Ajouter une contrainte
- ✓ Vider, renommer ou supprimer une table

Ajouter colonne | Modifier colonne | Supprimer colonne | Renommer colonne | Ajouter contrainte | Vider, renommer ou supprimer table

MODIFICATION DES STRUCTURES DES TABLES

- ♦ La définition de la structure d'une table n'est qu'une première étape dans la vie d'une base de données;
- ♦ On est toujours amené par la suite à **créer de nouvelles tables**, à **ajouter des colonnes** ou à **en modifier leurs définition**;
- ♦ La forme générale de la requête permettant de modifier la structure d'une table est :


```
ALTER TABLE nom_table ACTION description_Action;
```
- ♦ où **ACTION** peut être principalement:
 - ✓ ADD;
 - ✓ MODIFY;
 - ✓ CHANGE;
 - ✓ DROP;
- ♦ **description_Action** est la commande de modification associée à ACTION.

192

Ajouter colonne | Modifier colonne | Supprimer colonne | Renommer colonne | Ajouter contrainte | Vider, renommer ou supprimer table

AJOUTER UNE COLONNE

♦ Pour ajouter une colonne exécuter une requête contenant la commande **ALTER** conjointement avec l'action **ADD**.

Syntaxe:

```
ALTER TABLE nom_table ADD nom_col type_col ;
```

Exemple:

```
ALTER TABLE Etudiant ADD Ville_Naiss VarChar(50);
```

193

Ajouter colonne | **Modifier colonne** | Supprimer colonne | Renommer colonne | Ajouter contrainte | Vider, renommer ou supprimer table

MODIFIER UNE COLONNE

♦ Pour modifier une colonne exécuter une requête contenant la commande **ALTER** conjointement avec l'action **MODIFY**;

Syntaxe: `ALTER TABLE nom_table MODIFY nom_col Modification;`

♦ Les modifications possibles sont :

- ✓ Dans tous les cas il est possible d'**augmenter la taille d'une colonne** ;
- ✓ Il est possible de **diminuer la taille**, ou même de **changer le type** si la colonne ne contient que des valeurs nulles (sinon risque d'une perte des données pour les données existantes);
- ✓ On peut spécifier **NOT NULL** si la colonne ne contient aucune valeur NULL (sinon les NULL seront remplacées par des blancs);
- ✓ On peut dans tous les cas spécifier **NULL** pour autoriser les valeurs NULL.

194

Ajouter colonne | **Modifier colonne** | Supprimer colonne | Renommer colonne | Ajouter contrainte | Vider, renommer ou supprimer table

EXEMPLES

```
ALTER TABLE Etudiant MODIFY NumInscrEt Integer(7); #diminution de la taille
```

```
ALTER TABLE Etudiant MODIFY Ville_Naiss Char(30); /* Changement du type */
```

```
ALTER TABLE Etudiant MODIFY CinEt Integer(11) NOT NULL; -- Null devient Not Null
```

```
ALTER TABLE Etudiant MODIFY AdrEt VarChar(50) NULL; #Not Null devient Null
```

195

Ajouter colonne | Modifier colonne | **Supprimer colonne** | Renommer colonne | Ajouter contrainte | Vider, renommer ou supprimer table

SUPPRIMER UNE COLONNE

♦ Pour supprimer une colonne exécuter une requête contenant la commande **ALTER** conjointement avec l'action **DROP**;

Syntaxe:

```
ALTER TABLE nom_table DROP COLUMN nom_col;
```

Exemple:

```
ALTER TABLE Etudiant DROP COLUMN Ville_Naiss;
```

196

Ajouter colonne | Modifier colonne | Supprimer colonne | **Renommer colonne** | Ajouter contrainte | Vider, renommer ou supprimer table

RENOMMER UNE COLONNE

♦ Pour renommer une colonne exécuter une requête contenant la commande **ALTER** conjointement avec l'action **CHANGE**;

Syntaxe:

```
ALTER TABLE nom_table CHANGE ancien_nom_col nouveau_nom_col type_col;
```

Exemple:

```
ALTER TABLE Etudiant CHANGE AdrEt Adresse VarChar(50) ;
```

197

Ajouter colonne | Modifier colonne | Supprimer colonne | Renommer colonne | **Ajouter contrainte** | Vider, renommer ou supprimer table

AJOUTER UNE CONTRAINT

♦ Pour ajouter une contrainte exécuter une requête contenant la commande **ALTER** conjointement avec l'action **ADD CONSTRAINT**;

Syntaxe:

```
ALTER TABLE nom_table ADD CONSTRAINT nom_contrainte description_contrainte;
```

Exemples:

```
/* Ajout d'une contrainte clé primaire */
ALTER TABLE Etudiant ADD CONSTRAINT pk_etudiant PRIMARY KEY (NumInscrEt);
```

ALTER TABLE Etudiant ADD PRIMARY KEY (NumInscrEt); #Ajouter une clé primaire

```
/* Ajout d'une contrainte clé étrangère */
ALTER TABLE Note ADD CONSTRAINT fk_note_etudiant
FOREIGN KEY (NumInscrEt) REFERENCES Etudiant(NumInscrEt);
```

198

Ajouter colonne | Modifier colonne | Supprimer colonne | Renommer colonne | **Ajouter contrainte** | Vider, renommer ou supprimer table

AJOUTER UNE CONTRAINT VAEUR PAR DÉFAUT

♦ Pour ajouter (ou modifier) une **contrainte valeur par défaut** exécuter une requête contenant la commande **ALTER** conjointement avec l'action **SET DEFAULT**;

Syntaxe:

```
ALTER TABLE nom_table ALTER nom_col SET DEFAULT 'Valeur par défaut';
```

Exemple:

```
ALTER TABLE Etudiant ALTER AdrEt SET DEFAULT 'Safi' ;
```

199

Ajouter colonne | Modifier colonne | Supprimer colonne | Renommer colonne | **Ajouter contrainte** | **Vider, renommer ou supprimer table**

VIDER, RENOMMER OU SUPPRIMER UNE TABLE

♦ Pour Vider une table exécuter une requête contenant la commande **TRUNCATE**;

Syntaxe:

```
TRUNCATE nom_table;
```

Exemple:

```
TRUNCATE Etudiant ;
#Attention la clé de la table ne doit pas être clé étrangère dans une autre table
```

♦ Pour renommer une table exécuter une requête contenant la commande **ALTER** conjointement avec l'action **RENAME TO**;

Syntaxe:

```
ALTER TABLE ancien_nom RENAME TO nouveau_nom_table;
```

Exemple:

```
ALTER TABLE Etudiant RENAME TO Et;
```

♦ Pour supprimer une table exécuter une requête contenant la commande **DROP**;

Syntaxe:

```
DROP TABLE nom_table;
```

Exemple:

```
DROP TABLE Note;
/* Attention la clé de la table ne doit pas être clé étrangère dans une autre table */
```

200

Langage de Manipulation de Données LMD

Introduction

La commande INSERT

La commande UPDATE

La commande DELETE

Introduction

La commande INSERT

La commande UPDATE

La commande DELETE

INTRODUCTION

- ♦ Le Langage de Manipulation de Données LMD (en anglais Data Manipulation Language DML) permet de mettre à jour les données d'une BD (Ajout, Modification, Suppression);
- ♦ On distingue les trois types de commandes LMD suivants:
 - ✓ INSERT: pour l'insertion des données;
 - ✓ UPDATE: pour la modification des données;
 - ✓ DELETE: pour la suppression des données.

202

Introduction

La commande INSERT

La commande UPDATE

La commande DELETE

LA COMMANDE INSERT

- ♦ Pour insérer des données dans une table utiliser la commande **INSERT**, cette commande permet d'insérer plusieurs lignes dans une table;

Syntaxe:

```
INSERT INTO nom_table [(colonne1, colonne2,...)]
VALUES (valeur1,1, valeur1,2,...), (valeur2,1, valeur2,2,...),..., (valeurn,1, valeurn,2,...);
```

- ♦ La liste des **colonnes entre crochets []** est **optionnelle** (n'est pas obligatoire), par défaut toutes les colonnes sont classées dans l'ordre donné lors de la création de la table;
- ♦ Si la commande comporte une liste de colonnes, **les colonnes qui ne sont pas mentionnées dans la liste auront la valeur par défaut ou la valeur NULL.**

203

Introduction

La commande INSERT

La commande UPDATE

La commande DELETE

LA COMMANDE INSERT

Exemples:

```
INSERT INTO Etudiant (NumInscrEt, CinEt, Nationalite, NomEt, PrenomEt, AdrEt, TeleEt)
VALUES ('1', 'C0011', 'Marocaine', 'BAKKALI', 'Rachid', 'Safi', '0611123111');
```

```
INSERT INTO Etudiant (NumInscrEt, CinEt, Nationalite, NomEt, PrenomEt, AdrEt, TeleEt)
VALUES ('2', 'C0625', 'Marocaine', 'BENALI', 'Rahma', 'Rabat', '0611451111'),
('3', 'C0086', 'Marocaine', 'MOURADI', 'Yahya', 'Casa', '061119811'),
('4', 'C0213', 'Marocaine', 'MAZOUZ', 'Sanae', 'Marrakech', '0617631111');
```

204

Introduction | La commande INSERT | La commande UPDATE | La commande DELETE

LA COMMANDE INSERT

Exemples:

#SI le NumInscrEt est auto-increment on peut lui attribuer une valeur NULL

```
INSERT INTO Etudiant
VALUES (NULL, 'C1011', 'Marocaine', 'ERRADI', 'Hicham', 'Safi', '0614253111');
```

```
INSERT INTO Etudiant
VALUES (NULL, 'C2019', 'Marocaine', 'CHTIOUI', 'Loubna', 'Fes', '0611498011'),
      (NULL, 'C1223', 'Marocaine', 'MITKAL', 'Imad', 'Tanger', '061141011'),
      (NULL, 'C1982', 'Marocaine', 'MANZOUL', 'Siham', 'Safi', '0617634521');
```

205

Introduction | La commande INSERT | La commande UPDATE | La commande DELETE

LA COMMANDE INSERT

Exemples:

#On peut écrire plusieurs requêtes à la fois mais chaque requête se termine par « ; »

```
INSERT INTO Etudiant
VALUES (NULL, 'C5638', 'Marocaine', 'MABROUK', 'Nabila', 'Safi', '0614253980');
```

```
INSERT INTO Etudiant
VALUES (NULL, 'C1891', 'Marocaine', 'ALLA', 'Jaouad', 'Rabat', '0614253394');
```

```
INSERT INTO Etudiant
VALUES (NULL, 'C1381', 'Marocaine', 'ETTAKI', 'Adil', 'Tanger', '0614253098');
```

206

Introduction | La commande INSERT | La commande UPDATE | La commande DELETE

LA COMMANDE INSERT

Exemples:

/*
Si la valeur d'une colonne n'est pas renseignée, elle va prendre une valeur:
- par défaut: si une contrainte valeur par défaut est appliquée, cas de la colonne Nationalite;
- incrémentation de la valeur précédente: si la colonne est auto-increment cas de la colonne NumInscrEt;
- NULL: pour les colonnes qui peuvent être NULL.
*/

```
INSERT INTO Etudiant (CinEt, NomEt, PrenomEt, TeleEt)
VALUES ('C3459', 'ZAKI', 'Saloua', '0616598011'),
      ('C8123', 'BENIS', 'Chawki', '0613410191'),
      ('C2382', 'ANIS', 'Mouna', '0636634521');
```

207

Introduction | La commande INSERT | La commande UPDATE | La commande DELETE

LA COMMANDE UPDATE

- ◆ Pour modifier des données dans une table utiliser la commande **UPDATE**, cette commande permet de **modifier plusieurs lignes dans une table**;

Syntaxe:

```
UPDATE nom_table
SET col1 = exp1, col2 = exp2, ..., coln = expn
[ WHERE Condition ] ;
```

- ◆ L'expression peut être une valeur ou une opération;
- ◆ La clause **WHERE** est optionnelle (n'est pas obligatoire);
- ◆ Si la condition de modification n'est pas exprimée (pas de clause **WHERE**), toutes les lignes de la table seront modifiées (pour les colonnes clés la modification sera annulée).

208

Introduction | La commande INSERT | **La commande UPDATE** | La commande DELETE

LA COMMANDE UPDATE

Exemples:

```
UPDATE Etudiant
SET AdrEt='Tétouan', TeleEt='0666767689'
WHERE NumInscrEt=1;
```

```
UPDATE Note
SET Note=Note + 2
WHERE Note <= 6 AND CodeMat=4;
```

```
UPDATE Note
SET Note=Note + 2;
```

209

Introduction | La commande INSERT | La commande UPDATE | **La commande DELETE**

LA COMMANDE DELETE

♦ Pour supprimer des données à partir d'une table utiliser la commande **DELETE**, cette commande permet de **supprimer plusieurs lignes à partir d'une table**;

Syntaxe:

```
DELETE FROM nom_table
[ WHERE Condition ] ;
```

♦ La clause **WHERE** est **optionnelle** (n'est pas obligatoire);

♦ Si la condition de suppression n'est pas exprimée (pas de clause WHERE), **toutes les lignes de la table seront supprimées**.

210

Introduction | La commande INSERT | La commande UPDATE | **La commande DELETE**

LA COMMANDE DELETE

Exemples:

```
DELETE FROM Etudiant
WHERE NumInscrEt=14;
```

```
DELETE FROM Note
WHERE Note <= 8 AND CodeMat=4;
```

211

Introduction | La commande INSERT | La commande UPDATE | **La commande DELETE**

LA COMMANDE DELETE

Exemples:

```
DELETE FROM Note
WHERE Note <= 8 AND (CodeMat=4 OR CodeMat=1) ;
```

```
DELETE FROM Note; # permet de vider la table Note
```

212