

Bases de Données Relationnelles 2



**APPROFONDISSEMENT SQL LDD:
CONSTRAINTES ET NORMALISATION**

L2 MIASH 2022-2023

**UNIVERSITÉ PARIS NANTERRE
RAFAEL ANGARITA**

Rappel SQL LDD: création d'une table

2

```
CREATE TABLE IF NOT EXISTS Clients(  
  ID_Client INT PRIMARY KEY AUTO_INCREMENT,  
  Nom TEXT(50) NOT NULL,  
  Prenom TEXT(50) NOT NULL,  
  CP Int,  
  Ville Text(50)  
);
```

- Est-ce que cette table est normalisée ?
- Comment vous feriez pour implémenter votre solution on SQL ?

Rappel SQL LDD

3

SQL LDD

Créer une table
**CREATE
TABLE**

Supprimer une
table
DROP TABLE

Modifier une
table **ALTER
TABLE**

**ADD
COLUMN /
CONSTRAINT**

**DROP
COLUMN /
CONSTRAINT**

**ALTER
COLUMN**

Rappel SQL LDD: création d'une table

4

CREATE TABLE *nomtable* (

champ₁ *type* [(*taille*)] [**NOT NULL**],

champ₂ *type* [(*taille*)] [**NOT NULL**],

.....

CONSTRAINT *nomc₁* **PRIMARY KEY** (*champ_i* [, *champ_j* , ...]) |

CONSTRAINT *nomc₂* **FOREIGN KEY** (*champ_m* [, *champ_n* , ...])
REFERENCES *table_E* (*champ_E*)

.....

)

Rappel SQL LDD: modification d'une table

5

ALTER TABLE *table*

ADD COLUMN *champ* *type*[(*taille*)] [**NOT NULL**] |

ALTER COLUMN *champ* *type*[(*size*)] |

DROP COLUMN *champ* |

ADD CONSTRAINT ... |

DROP CONSTRAINT *nom*

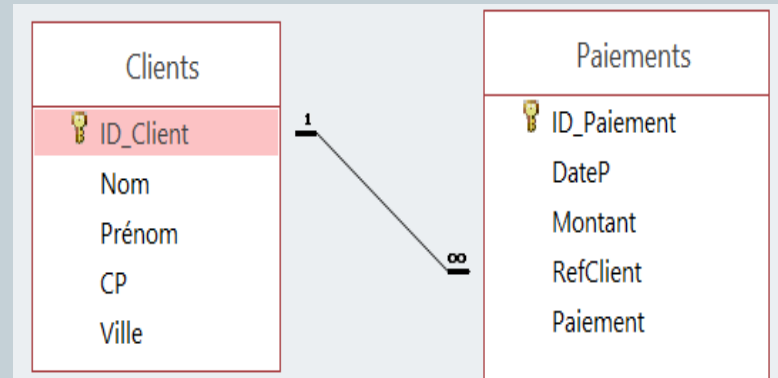
Vérifier l'existence d'une table

6

- Il est conseillé de précéder à une vérification de l'existence d'une table avant l'exécution d'une commande LDD:

CREATE TABLE IF NOT EXISTS
DROP TABLE IF EXISTS

```
CREATE TABLE IF NOT EXISTS
clients (
ID_Client INT PRIMARY KEY ,
Nom VARCHAR,
Prenom VARCHAR(50),
CP INT(11),
Ville VARCHAR(50)
)
```



Créer une table à partir d'une autre

7

- Il est possible de créer une table à partir du schéma complet ou une partie du schéma d'une table existante:

```
CREATE TABLE nouvelle_table AS SELECT * FROM table_existante
```

➔ Copie aussi les données dans la table d'origine.

```
CREATE TABLE IF NOT EXISTS clients_copie AS  
SELECT * FROM clients;
```

clients

#	Nom	Type de données
1	ID_Client	INT
2	Nom	VARCHAR
3	Prenom	VARCHAR
4	CP	INT
5	Ville	VARCHAR



Clients_copie

#	Nom	Type de données
1	ID_Client	INT
2	Nom	VARCHAR
3	Prenom	VARCHAR
4	CP	INT
5	Ville	VARCHAR

Créer une table à partir d'une autre

8

- Il est possible de copier seulement une partie de la table d'origine

```
CREATE TABLE nouvelle_table AS SELECT <champ1, champ2, ..>  
FROM table_existante [WHERE ..]
```

Exemple: Créer une table des villes des clients

```
CREATE TABLE IF NOT EXISTS villes AS  
SELECT DISTINCT CP, Ville FROM clients;
```

Nom	Type de données
CP	INT
Ville	VARCHAR



CP	Ville
3 720	IVRY
68 000	COLMAR
13 000	MARSEILLE
84 002	AVIGNON
30 126	LIRAC
5 000	GAP
69 005	LYON
34 003	MONTPELLIER
74 800	CHAMONIX
51 000	REIMS

Les Contraintes sur les champs

9

Type	Objectif de la contrainte	Syntaxe
Contrainte sur attribut	La valeur d'un attribut doit être unique au sein de la table	UNIQUE
	Un attribut doit toujours avoir une valeur	NOT NULL
	Toute règle s'appliquant à la valeur d'un attribut	CHECK
Contrainte d'intégrité	Un attribut (ou un ensemble d'attributs) constitue(nt) la clé de la table	PRIMARY KEY
	Un attribut dans une table est liée à la clé primaire d'une autre table (<i>intégrité référentielle</i>).	FOREIGN KEYREFERENCES

Contraintes sur les attributs – Exemple 1

10

Supprimer la table Pays si elle existe

```
DROP TABLE IF EXISTS Pays;  
CREATE TABLE Pays (codePays varchar(4) NOT  
NULL,  
nom varchar(30) default 'Inconnu' NOT NULL,  
langue varchar(30) NOT NULL,  
CONSTRAINT pk PRIMARY KEY(codePays));
```

La valeur doit être renseignée

Valeur par défaut si non renseignée

Contrainte de clé primaire

```
insert into Pays values ('FR', 'France', 'Français');
```

Contraintes sur les attributs – Exemple 2

11

Créer la table Film si elle n'existe pas

```
CREATE TABLE IF NOT EXISTS Film (  
  idFilm integer NOT NULL,  
  titre varchar (50) NOT NULL,  
  année integer CHECK (année between 1890 and  
  2021),  
  genre varchar (20) CHECK (genre in  
  ('Histoire', 'Western', 'Drame')),  
  résumé varchar (255),  
  codePays varchar (4),  
  CONSTRAINT pk2 PRIMARY KEY (idFilm),  
  CONSTRAINT fk FOREIGN KEY (codePays) REFERENCES  
  Pays (odePays) );
```

Toute valeur
entrée pour
l'année doit être
entre 1890 et
2021

Toute valeur
entrée pour
genre doit être
dans cet
ensemble


Contrainte d'intégrité référentielle vers la
table Pays

Normalisation d'une base par SQL LDD

Exemple

12

Soit la table ZOO suivante

	1	Num_Animal	INT
	2	Nom_Animal	VARCHAR
	3	Annee_naissa...	INT
	4	Espece	VARCHAR
	5	Prenom_Gardi...	VARCHAR
	6	Nom_Gardien	VARCHAR
	7	Salaire	DECIMAL
	8	Classe	VARCHAR
	9	Origine	VARCHAR
	10	Num_empl	INT
	11	Surface	INT
	12	Type_empl	INT
	13	Libelle_empl	VARCHAR

- Chaque ligne correspond à un animal auquel on attribue **un nom propre**, **une année de naissance** et **une espèce** (Ours, Lion, Boa, etc.). Cet animal est pris en charge par un **gardien** (avec **prénom et salaire**) et occupe un emplacement dans le zoo (**numéro d'emplacement**, **surface**, **type emplacement** et **libellé emplacement** : savane, désert, forêt, etc.). Enfin chaque espèce appartient à **une classe** (les mammifères, poissons, reptiles, batraciens ou oiseaux) et on considère pour simplifier qu'elle provient d'une **origine unique** (Afrique, Europe, etc.).

Normalisation d'une base par SQL LDD

Exemple

13

- Cette table ne respecte pas la 2^{ème} et la 3^{ème} forme normale
- ➔ Plusieurs concepts ont été regroupés dans la même table
- ➔ Pour Normaliser la table, il faut créer une table différente pour chaque concept, déplacer les données et créer une référence par clé externe
- ➔ Les concepts concernés sont : le *gardien*, l'*emplacement* et la *classe*.

Normalisation d'une base par SQL LDD

Exemple

14

Création de la table Emplacement à partir de la table Zoo

```
CREATE TABLE Emplacement
AS SELECT DISTINCT Num_empl, Surface, Type_empl, Libelle_empl
FROM zoo;
```

Modifier la table Emplacement pour ajouter une clé primaire

```
ALTER TABLE emplacement
ADD CONSTRAINT PRIMARY KEY (Num_empl);
```

Supprimer les données de l'emplacement dans la table Zoo, mais garder le numéro qui deviendra une clé externe

```
ALTER TABLE zoo DROP COLUMN Surface;
ALTER TABLE zoo DROP COLUMN Type_empl;
ALTER TABLE zoo DROP COLUMN Libelle_empl;
```

Normalisation d'une base par SQL LDD

Exemple

15

Créer la contrainte d'intégrité référentielle

```
ALTER TABLE zoo  
ADD CONSTRAINT FOREIGN KEY (Num_empl)  
REFERENCES emplacement (Num_empl);
```

#	Nom	Type de données
1	Num_Animal	INT
2	Nom_Animal	VARCHAR
3	Annee_naissa...	INT
4	Espece	VARCHAR
5	Prenom_Gardi...	VARCHAR
6	Nom_Gardien	VARCHAR
7	Salaire	DECIMAL
8	Classe	VARCHAR
9	Origine	VARCHAR
10	Num_empl	INT

#	Nom	Type de données
1	Num_empl	INT
2	Surface	INT
3	Type_empl	INT
4	Libelle_empl	VARCHAR

Normalisation d'une base par SQL LDD

Exemple

16

Les mêmes étapes sont à faire avec les concepts « Classe » et « Gardien »

```
#créer table Classe
CREATE TABLE Classe
AS SELECT DISTINCT classe, Origine
FROM zoo;
#Ajouter une clé primaire
ALTER TABLE Classe
ADD CONSTRAINT PRIMARY KEY (classe);
#Ajouter une contrainte de référence
ALTER TABLE zoo
ADD CONSTRAINT FOREIGN KEY (classe) REFERENCES
Classe(classe);
#Supprimer les données dans Zoo
ALTER TABLE zoo DROP COLUMN Origine;
.....
```