

Chapitre 1

Le modèle relationnel

Définition :

Langage qui permet de discuter avec une base de données

Les bases de données sont gérées à l'aide de logiciels spécifiques (SGBD)

- ➔ Les plus répandus sont les relationnels
 - Conforme au modèle relationnel (comment on conçoit la structure de la base de données)
 - SGBDR

Ces logiciels proposent des interfaces utilisateurs, mais toutes les actions sont directement possible en langage SQL (Structured Query Language = Langage de requête structuré).

Traitement	Langage SQL
Définir la structure d'une base de données	Langage de définition des données (LDD) ➔ Permet de créer la base de données
Manipuler des données (Ajouter, mettre à jour, supprimer)	Langage de manipulation des données (LMD) ➔ Permet de modifier les données
Interroger des données (rechercher)	Langage d'interrogation des données (LID) ➔ Permet de rechercher des informations

Dans un modèle relationnel

- Une colonne est un champ ou un attribut
- Une ligne est une occurrence

Une clé primaire est obligatoire, c'est une info qui permet d'identifier qu'un seul élément (ex : INE élève)

Modèle relationnel de la relation ELEVE :

ELEVE (INE, NOM, PRENOM, DateDeNaissance, SEXE, STATUT, ADRESSE, MAIL)

Clé primaire : INE

Modèle relationnel entre ELEVE et CLASSE :

ELEVE (INE, NOM, PRENOM, DateDeNaissance, SEXE, STATUT, ADRESSE, MAIL, NumClasse)

Clé primaire : INE

Clé étrangère : NumClasse en référence à NumClasse de la relation CLASSE

CLASSE (NumClasse, LibelleClasse, NIVEAU)

Clé primaire : NumClasse

Dépendance entre la relation ELEVE et CLASSE

➔ Il faut donc créer la relation CLASSE avant la relation ELEVE

La contrainte d'intégrité référentiel gérée en tant que sécurité par les SGBDR, implique que :

- Une clé étrangère et la clé primaire à laquelle elle fait référence aient le même type de données.
- Toutes valeurs d'une clé étrangère préexistent en tant que valeur de la clé primaire à laquelle elle fait référence.

➔ Une clé étrangère doit d'abord être primaire

➔ La clé étrangère doit avoir le même type de données que sa clé primaire

Clé primaire : Attribut d'une table qui permet d'identifier de façon unique une occurrence dans la table

Clé étrangère : Clé primaire d'une table qui est présente dans une autre table

Le modèle relationnel nous permet d'ordonner la base de données, maintenant voyons comment la créer.

Chapitre 2

Langage de définition des données (LDD)

Définition :

En SQL, le LDD permet de réaliser 3 actions afin de créer la base de données

- Créer des tables
- Modifier les tables
- Supprimer les tables

1- Créer la base de données et les tables

1-Syntaxe :

CREATE DATABASE nomdelabase ;

CREATE TABLE nomdelatable (NomChamp1 typededonnées, NomChamp2 types de données) ;

→Lorsqu'on crée une table, on est obligé d'y mettre les attributs de la table.

Type de données	Type SQL	Description
Alphanumérique	Char(n)	Chaine de caractères de longueur fixe n (0 à 255), utilise tout l'espace indiqué même si la saisie est inférieure à n, la saisie est complétée automatiquement par des espaces
	Varchar(n)	Chaine de caractères de n caractères maximum (0 à 255) → utilise uniquement l'espace occupé par la saisie
	text	Au-delà de 255 caractères
Numérique	Intégrer (int)	Entier long (Jusque 2 147 483 647)
	Float (double)	Réel à virgule flottante (saisir les décimales après le point)
	Decimal (n,d)	N chiffre dont D décimales
Date/Heure	Date	Aaaa-mois-jj
	Time	hh :mm :ss
Booléen	Boolean	Accepte les valeurs TRUE ou FALSE 0 ou 1

Pour chaque champ créé il faut préciser le type de données.

Pour le type Varchar, il faut préciser la taille

Ex : ELEVE (INE VARCHAR(20), NOME VARCHAR(50))

2-Création de clé :

Une table n'a qu'une clé primaire (éventuellement formée de 2 champs) et peut avoir aucune ou plusieurs clés étrangères

En SQL, au moment de la création de la table, nous allons définir les champs « clé primaire » et « clé étrangère ».

- ➔ **Primary Key** (nomduchamp)
- ➔ **Foreign Key** (nomduchamp) References Tablesource (nomduchampcleprimaire)

2-Modifier des tables (Mise à jour)

Ajout d'un champ : ALTER TABLE nomdelatable

ADD nomduChamp1 type de données, nomduChamp2 type de données

Ex : ALTER TABLE Personne

ADD codePostal INTEGER ;

Suppression d'un champ : ALTER TABLE nomdelatable

DROP nomduchamp ;

3-Supprimer une table

Suppression d'une table : DROP TABLE nomdelatable ;

Chapitre 3

LID Langage d'interrogation de données

Le langage d'interrogation de la base de données permet d'envoyer des requêtes à la base sans la modifier, afin de récupérer les informations dont on a besoins.

1-Interrogation d'une table simple

- Projection :

Pour faire une projection, on utilise l'instruction **SELECT** pour recenser les champs à projeter à l'aide des tables précisées dans le **FROM**

Cette requête affiche tous les services de l'entreprise :

SELECT nom, num

FROM Service ;

Requête pour afficher les noms et fonctions de tous les employer :

SELECT nom, fonction

FROM EMPLOYER ;

Si on a besoins de **tous les éléments de la table** pour tous les employer :

SELECT *

FROM EMPLOYER ;

- La restriction :

Je veux afficher uniquement les employés qui sont commercial :

SELECT nom, fonction

FROM EMPLOYER,

WHERE fonction = « commercial » ;

Les commerciaux dont le salaire est supérieur à 2500€ :

```
SELECT nom, fonction, salaire  
FROM EMPLOYER,  
WHERE fonction = « commercial »  
AND salaire > 2500 ;
```

La requête qui permet d'afficher tous les salariés avec la date d'embauche, qui n'ont pas de commission :

```
SELECT nom, dateEmbauche, commission  
FROM EMPLOYER  
WHERE commission = 0 ;
```

Afficher les services du bâtiment C :

```
SELECT nom, lieu  
FROM SERVICE  
WHERE lieu = « Bâtiment C » ;
```

On peut également faire des restrictions à partir de calculs :

Ex : Je veux afficher les salariés dont la rémunération = (salaire + commission)

Liste des salariés dont le salaire est supérieur à 3500 :

```
SELECT *  
FROM EMPLOYER  
WHERE (salaire + commission) > 3500 ;
```

- **La jointure :**

La jointure consiste à signifier au moteur SQL les relations qui lient les tables entre elles. Les jointures se font **UNIQUEMENT** sur les clés primaires et les clés étrangères.

Ex : Je veux afficher le nom de service de tous les commerciaux :

SELECT EMPLOYER.nom, fonction, SERVICE.nom

FROM EMPLOYER.SERVICE

WHERE EMPLOYER.numServEmployer = SERVICE.num

AND fonction = « commercial » ;

(On précise dans quelle table il se trouve, étant donné qu'il y en a un dans chaque table.)

- **Tri :**

SELECT

FROM

WHERE

GROUP BY nomEleve ASC/DESC ;

- **Regroupement :**

SELECT

FROM

WHERE

GROUP BY NomVille