

ALEX & ALAIN

24.06.20

1. **Architecture d'une Base de Données Relationnelle**

### 1. Base de Données (Database)

* **Définition** : Container principal qui regroupe toutes les tables et objets liés
* **Exemple** : Une base de données "ecommerce" pour un site de vente en ligne
* **Analogie** : Comme un classeur qui contient plusieurs dossiers

### 2. Tables

* **Définition** : Structure de stockage des données organisée en lignes et colonnes
* **Caractéristiques** :
  + Chaque table représente une entité (clients, produits, commandes)
  + Structure fixe définie à la création
  + Contient des données

### 3. Colonnes (Champs/Fields)

* **Définition : Attributs qui décrivent les propriétés d'une entité**
* **Propriétés d'une colonne :**
  + **Nom : Identifiant unique dans la table**
  + **Type de données : VARCHAR, INTEGER, DATE, etc.**
  + **Contraintes : NOT NULL, UNIQUE, DEFAULT**

### 4. Lignes (Enregistrements/Records)

* **Définition : Instance unique d'une entité dans une table**
* **Caractéristiques :**
  + **Chaque ligne représente un élément distinct**
  + **Contient une valeur pour chaque colonne**
  + **Identifiée de manière unique par la clé primaire**

1. **Concepts de Clés**

### Clé Primaire (Primary Key)

* **Définition** : Colonne ou combinaison de colonnes qui identifie de manière unique chaque ligne
* **Règles strictes** :
  + **Unique** : Aucune duplication possible
  + **Non nulle** : Doit toujours avoir une valeur
  + **Immuable** : Ne change jamais une fois assignée
  + **Une seule par table** : Maximum une clé primaire par table

**Exemple :**

|  |
| --- |
| sql CREATE TABLE clients (  id SERIAL PRIMARY KEY, *-- Clé primaire auto-incrémentée*  nom VARCHAR(100) NOT NULL,  email VARCHAR(150) UNIQUE ); |

## 

## Clé Étrangère (Foreign Key)

* **Définition** : Colonne qui fait référence à la clé primaire d'une autre table
* **Rôle** : Établit et maintient les relations entre tables
* **Intégrité référentielle** : Garantit que les références sont valides

Exemple de relation :

|  |
| --- |
| sql *-- Table parente* CREATE TABLE categories (  id SERIAL PRIMARY KEY,  nom VARCHAR(100) NOT NULL );  *-- Table enfant avec clé étrangère* CREATE TABLE produits (  id SERIAL PRIMARY KEY,  nom VARCHAR(200) NOT NULL,  prix DECIMAL(10,2),  categorie\_id INTEGER REFERENCES categories(id) *-- Clé étrangère* ); |

## **Types de Relations**

### 1. Relation Un-à-Plusieurs (1:N)

* **Description** : Un enregistrement dans la table parent peut avoir plusieurs enregistrements dans la table enfant
* **Exemple** : Un client peut avoir plusieurs commandes
* **Implémentation** : Clé étrangère dans la table "plusieurs"

### 2. Relation Plusieurs-à-Plusieurs (N:N)

* **Description** : Plusieurs enregistrements d'une table peuvent être liés à plusieurs enregistrements d'une autre table
* **Exemple** : Une commande peut contenir plusieurs produits, et un produit peut être dans plusieurs commandes
* **Implémentation** : Table de liaison avec deux clés étrangères

## 

|  |
| --- |
| sql *-- Table de liaison pour relation N:N* CREATE TABLE commandes\_produits (  id SERIAL PRIMARY KEY,  commande\_id INTEGER REFERENCES commandes(id),  produit\_id INTEGER REFERENCES produits(id),  quantite INTEGER NOT NULL ); |

## 

### 3. Relation Un-à-Un (1:1)

* **Description** : Un enregistrement d'une table correspond à exactement un enregistrement d'une autre table
* **Exemple** : Un utilisateur a un seul profil détaillé
* **Implémentation** : Clé étrangère unique ou tables fusionnées

## **Normalisation des Données**

### Objectifs

* Éliminer la redondance des données
* Éviter les anomalies de mise à jour
* Optimiser l'espace de stockage
* Maintenir l'intégrité des données

### Première Forme Normale (1NF)

* Chaque cellule contient une valeur atomique (indivisible)
* Pas de groupes répétitifs
* Chaque ligne est unique

### Deuxième Forme Normale (2NF)

* Respecte 1NF
* Tous les attributs non-clés dépendent entièrement de la clé primaire
* Élimine les dépendances partielles

### Troisième Forme Normale (3NF)

* Respecte 2NF
* Aucun attribut non-clé ne dépend d'un autre attribut non-clé
* Élimine les dépendances transitives

## 

## 

## 

## 