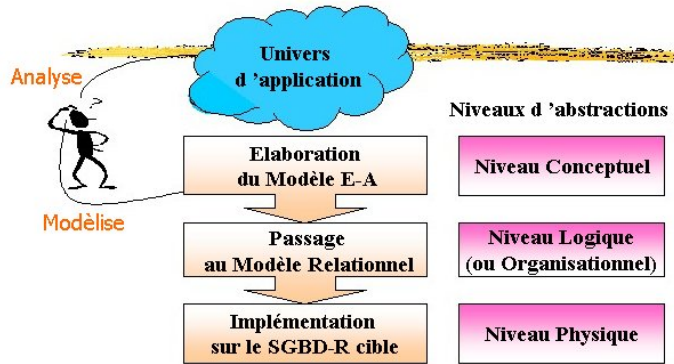


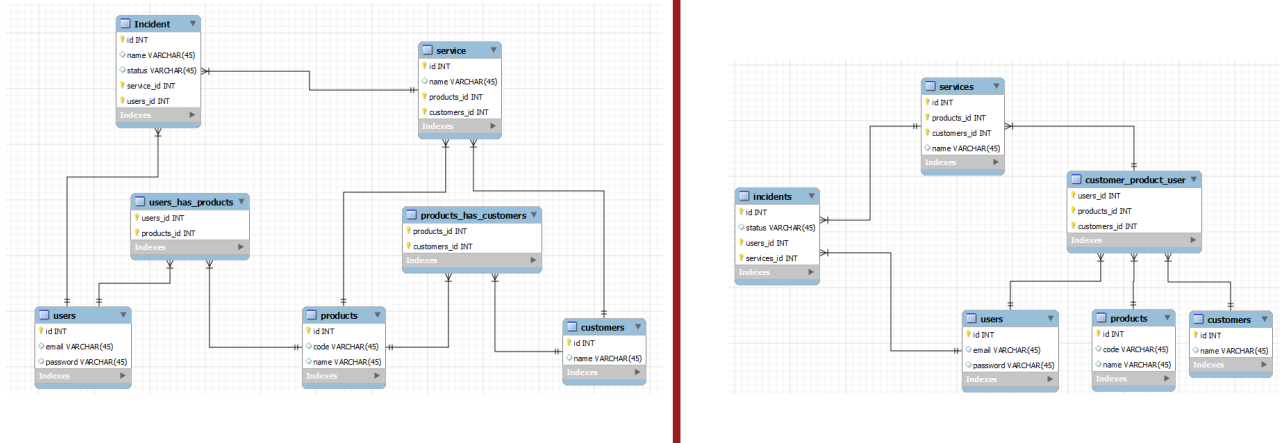
# Les Données Structurales en Base de Données

## La démarche de Conception



Les données structurales en base de données se réfèrent à la structure qui organise et définit la manière dont les données sont stockées, accédées et manipulées. Ces structures incluent des éléments comme les tables, les colonnes, les index, les clés, les vues, et les schémas. Voici un aperçu des principaux composants structuraux en base de données relationnelle :

### 1. Tables



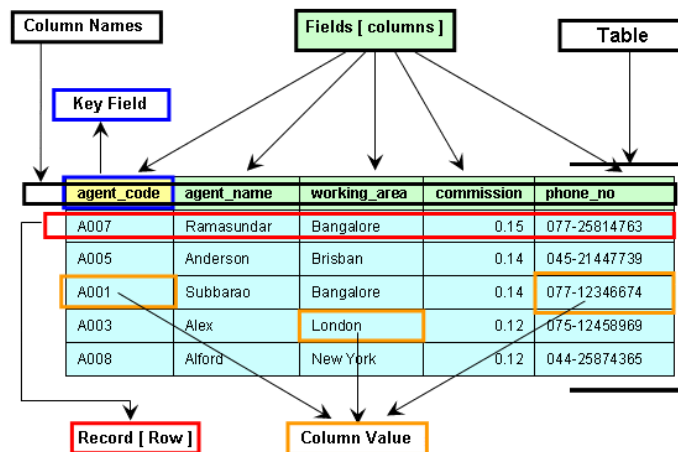
Les tables sont les éléments de base dans une base de données relationnelle. Elles stockent les données sous forme de lignes et de colonnes.

- **Colonnes (Attributs) :** Définissent les types de données pour chaque champ dans une table.

- **Lignes (Enregistrements)** : Chaque ligne représente un enregistrement unique dans la table.

```
CREATE TABLE Employees (
    EmployeeID INT PRIMARY KEY,
    FirstName VARCHAR(50),
    LastName VARCHAR(50),
    BirthDate DATE
);
```

## 2. Colonnes

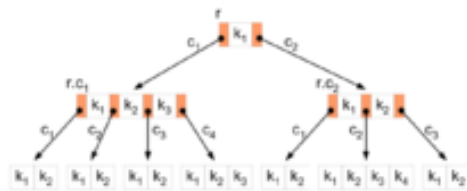


Chaque table est composée de colonnes, et chaque colonne a un type de données spécifique et peut avoir des contraintes associées.

- **Types de données** : INT, VARCHAR, DATE, etc.
- **Contraintes** : NOT NULL, UNIQUE, PRIMARY KEY, FOREIGN KEY, CHECK, DEFAULT.

```
CREATE TABLE Employees (
    EmployeeID INT PRIMARY KEY,
    FirstName VARCHAR(50) NOT NULL,
    LastName VARCHAR(50) NOT NULL,
    BirthDate DATE
);
```

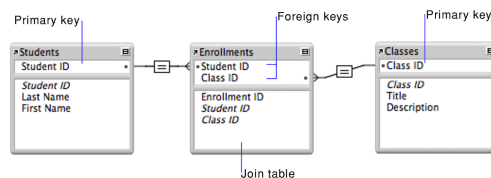
### 3. Index



Les index sont utilisés pour accélérer l'accès aux données dans une table. Ils peuvent être créés sur une ou plusieurs colonnes.

```
CREATE INDEX idx_lastname ON Employees (LastName);
```

### 4. Clés

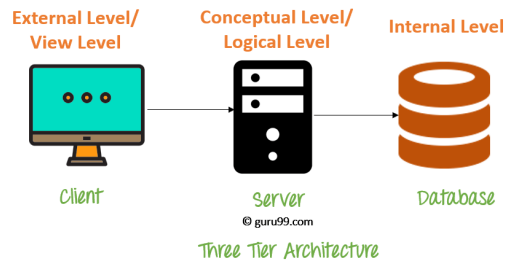


Les clés sont des colonnes ou des groupes de colonnes utilisées pour identifier de manière unique des lignes dans une table ou pour établir des relations entre les tables.

- **Clé primaire (PRIMARY KEY)** : Identifie de manière unique chaque enregistrement dans une table.
- **Clé étrangère (FOREIGN KEY)** : Établit une relation entre deux tables.
- **Clé unique (UNIQUE)** : Assure que toutes les valeurs d'une colonne sont uniques.

```
CREATE TABLE Orders (
    OrderID INT PRIMARY KEY,
    OrderDate DATE,
    EmployeeID INT,
    FOREIGN KEY (EmployeeID) REFERENCES Employees(EmployeeID)
);
```

## 5. Vues



Les vues sont des tables virtuelles basées sur le résultat d'une requête SQL. Elles permettent de simplifier les requêtes complexes et de restreindre l'accès aux données sensibles.

```
CREATE VIEW EmployeeDetails AS
SELECT FirstName, LastName, BirthDate
FROM Employees
WHERE BirthDate > '1980-01-01';
```

## 6. Schémas

COMPARISON BASIS	DATABASE	SCHEMA
<b>DEFINITION</b>	The database is an application that stores the organized collection of interrelated data	The schema is a logical representation of a database
<b>STATEMENT</b>	A DML is used to generate and modify the records in the database	A DDL statement is used to define the schema for a database structure
<b>INCLUDE</b>	A database is an organized collection of structured information or data	A schema describes how the metadata is set up, and usually addresses standards for common components of metadata like dates, names, and places
<b>MEMORY</b>	It uses memory to store data	It does not use memory to store data

Un schéma est une collection logique d'objets de base de données comme les tables, les vues, les procédures stockées, etc. Il aide à organiser et à structurer les objets de base de données.

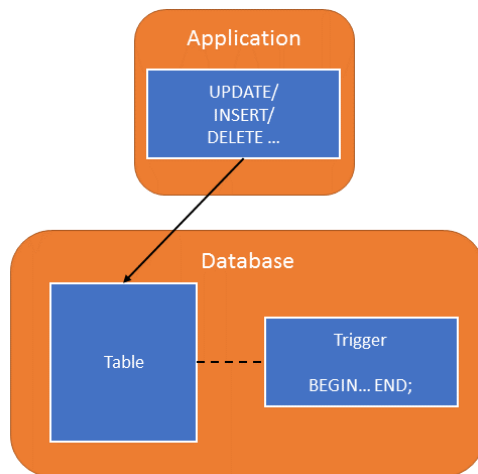
```
CREATE SCHEMA hr;
CREATE TABLE hr.Employees (
    EmployeeID INT PRIMARY KEY,
    FirstName VARCHAR(50),
    LastName VARCHAR(50),
```

```

        BirthDate DATE
    );

```

## 7. Triggers



Les triggers sont des procédures qui s'exécutent automatiquement en réponse à certains événements sur une table ou une vue, comme une insertion, une mise à jour ou une suppression.

```

CREATE TRIGGER trg_AfterInsert
ON Employees
AFTER INSERT
AS
BEGIN
    PRINT 'A new employee has been added.'
END;

```

## 8. Procédures et Fonctions Stockées

The screenshot shows a MySQL Command Line Client window with the following text:

```

mysql> use hr;
Database changed
mysql> CREATE PROCEDURE TEST()
-> SELECT * FROM USER;
Query OK, 0 rows affected (0.00 sec)

mysql> CALL TEST();
+-----+-----+-----+
| userId | password | name |
+-----+-----+-----+
| scott123 | 1238sco | Scott |
| ferp0734 | d1eeu063 | Palash |
| diana094 | ku5j023 | Diana |
+-----+-----+-----+
3 rows in set (0.00 sec)

Query OK, 0 rows affected (0.01 sec)

mysql>

```

Les procédures stockées et les fonctions sont des blocs de code SQL réutil-

isables stockés dans la base de données. Elles peuvent être utilisées pour encapsuler des opérations complexes.

- **Procédures stockées :**

```
CREATE PROCEDURE GetEmployeeDetails
    @EmployeeID INT
AS
BEGIN
    SELECT FirstName, LastName, BirthDate
    FROM Employees
    WHERE EmployeeID = @EmployeeID;
END;
```

- **Fonctions :**

```
CREATE FUNCTION GetEmployeeAge(@EmployeeID INT)
RETURNS INT
AS
BEGIN
    DECLARE @Age INT;
    SELECT @Age = DATEDIFF(YEAR, BirthDate, GETDATE())
    FROM Employees
    WHERE EmployeeID = @EmployeeID;
    RETURN @Age;
END;
```

## Conclusion

Les données structurales en base de données constituent l'ossature qui organise la manière dont les données sont stockées, récupérées et manipulées. Elles incluent des tables, des colonnes, des index, des clés, des vues, des schémas, des triggers, des procédures stockées et des fonctions. La bonne conception de ces structures est cruciale pour assurer la performance, l'intégrité et la sécurité des bases de données.