

# Compte rendu TP 1 bases de données

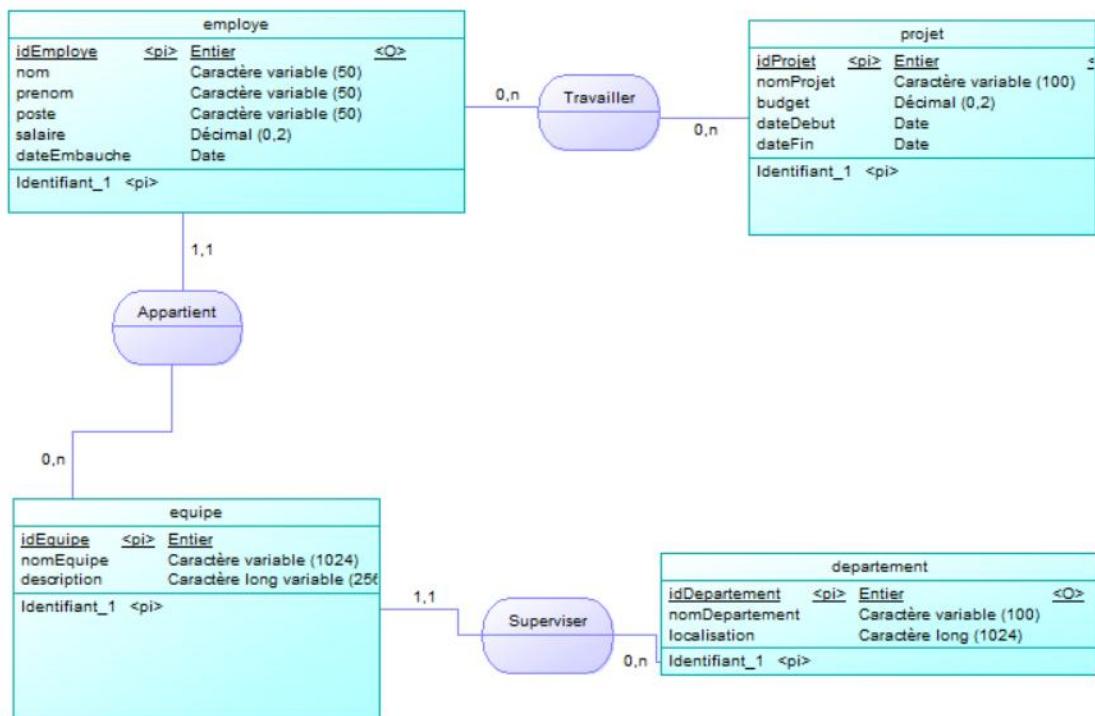
Par : ayoub zarouati - Master II-BDCC

Dans cet exercice, nous modélisons un système de gestion des employés, projets, équipes et départements d'une entreprise. Les entités principales sont : **Employé**, **Projet**, **Équipe** et **Département**, chacune avec ses attributs spécifiques ainsi que des relations entre eux,

Le MCD représente ces entités, leurs attributs et relations, servant de base au MLD.

## Modélisation :

Le mcd :



Le MLD équivalent :

Employes (idEmploye, nom, prenom, poste, salaire, dateEmbauche,#idEquipe)

Projets (idProjet, nomProjet, budget, dateDebut, dateFin)

Ligne\_intergation(#idProjet,#idEmploye)

Equipes (idEquipe, nomEquipe, description,#idDepartement )

Departements (idDepartement, nomDepartement, localisation)

## 1- Cration de la base de donnes

Cette commande cre une base de donnes nomme `DB\_ENTREPRISE` :

```
CREATE DATABASE DB_CATALOGUE;
```

Les commandes ci-dessous crent les tables, en definissant les colonnes, leurs types et les cles primaires.

## 2- Cration des tables

-- Table Departements

```
CREATE TABLE Departements (
    idDepartement INT PRIMARY KEY,
    nomDepartement VARCHAR(100) NOT NULL,
    localisation VARCHAR(100)
);
```

Executer une ou des requetes SQL sur la base de donnes « db\_entreprise »: 

```
1 CREATE TABLE Departements (
2     idDepartement INT PRIMARY KEY,
3     nomDepartement VARCHAR(100) NOT NULL,
4     localisation VARCHAR(100)
5 );
```

```
-- Table Equipes
```

```
CREATE TABLE Equipes (
    idEquipe INT PRIMARY KEY,
    nomEquipe VARCHAR(100) NOT NULL,
    description TEXT,
    idDepartement INT,
    FOREIGN KEY (idDepartement) REFERENCES Departements(idDepartement) ON
    DELETE SET NULL
);
```

Exécuter une ou des requêtes SQL sur la base de données « db\_entreprise »: 

```
1 CREATE TABLE Equipes (
2     idEquipe INT PRIMARY KEY,
3     nomEquipe VARCHAR(100) NOT NULL,
4     description TEXT,
5     idDepartement INT,
6     FOREIGN KEY (idDepartement) REFERENCES Departements(idDepartement) ON DELETE SET NULL
7 );
```

```
-- Table Employes
```

```
CREATE TABLE Employes (
    idEmploye INT PRIMARY KEY,
    nom VARCHAR(100) NOT NULL,
    prenom VARCHAR(100) NOT NULL,
    poste VARCHAR(100),
```

```

salaire DECIMAL(10, 2),

dateEmbauche DATE,

idEquipe INT,

FOREIGN KEY (idEquipe) REFERENCES Equipes(idEquipe) ON DELETE SET NULL

);


```

Exécuter une ou des requêtes SQL sur la base de données « db\_entreprise »: 

```

1 CREATE TABLE Employes (
2     idEmploye INT PRIMARY KEY,
3     nom VARCHAR(100) NOT NULL,
4     prenom VARCHAR(100) NOT NULL,
5     poste VARCHAR(100),
6     salaire DECIMAL(10, 2),
7     dateEmbauche DATE,
8     idEquipe INT,
9     FOREIGN KEY (idEquipe) REFERENCES Equipes(idEquipe) ON DELETE SET NULL
10 );

```

-- Table Projets

```

CREATE TABLE Projets (

    idProjet INT PRIMARY KEY,

    nomProjet VARCHAR(100) NOT NULL,

    budget DECIMAL(12, 2),

    dateDebut DATE,

    dateFin DATE

);

```

Exécuter une ou des requêtes SQL sur la base de données « db\_entreprise »: 

```
1 CREATE TABLE Projets (
2     idProjet INT PRIMARY KEY,
3     nomProjet VARCHAR(100) NOT NULL,
4     budget DECIMAL(12, 2),
5     dateDebut DATE,
6     dateFin DATE
7 );
```

-- Table Ligne\_intergation (association entre les Employes et les Projets)

```
CREATE TABLE Ligne_intergation (
    idProjet INT,
    idEmploye INT,
    PRIMARY KEY (idProjet, idEmploye),
    FOREIGN KEY (idProjet) REFERENCES Projets(idProjet) ON DELETE CASCADE,
    FOREIGN KEY (idEmploye) REFERENCES Employes(idEmploye) ON DELETE CASCADE
);
```

Exécuter une ou des requêtes SQL sur la base de données « db\_entreprise »: 

```
1 CREATE TABLE Ligne_intergation (
2     idProjet INT,
3     idEmploye INT,
4     PRIMARY KEY (idProjet, idEmploye),
5     FOREIGN KEY (idProjet) REFERENCES Projets(idProjet) ON DELETE CASCADE,
6     FOREIGN KEY (idEmploye) REFERENCES Employes(idEmploye) ON DELETE CASCADE
7 );
```

## 2. Requêtes CRUD

### a. Insertion : Ajouter un nouvel employé dans une équipe donnée

```
INSERT INTO departements (idDepartement,localisation, nomDepartement) values (1,'4eme','DOSI');
```

```
INSERT INTO `equipes`(`idEquipe`, `nomEquipe`, `description`, `idDepartement`) VALUES ('1','TMA','Maintenane et evolution',1);
```

```
INSERT INTO `employes`(`idEmploye`, `nom`, `prenom`, `poste`, `salaire`, `dateEmbauche`, `idEquipe`) VALUES (1,'zarouati','ayoub','Développeur',10000.00,'2024-01-01',1);
```

### b. Mise à jour : Changer le département auquel appartient une équipe

```
INSERT INTO `departements`(`idDepartement`, `nomDepartement`, `localisation`) VALUES (2,'Finance','3eme');
```

```
UPDATE Equipes SET idDepartement = 2 WHERE idEquipe = 1;
```

### c. Suppression : Supprimer un projet terminé

```
DELETE FROM Projets WHERE dateFin <= CURRENT_DATE;
```

### d. Consultation : Lister tous les projets d'un employé donné, avec les détails de son équipe et de son département

```
SELECT p.nomProjet, p.budget, p.dateDebut, p.dateFin, e.nomEquipe, e.description, d.nomDepartement, d.localisation FROM Projets p  
JOIN Ligne_interigation li ON p.idProjet = li.idProjet JOIN Employes emp ON li.idEmploye = emp.idEmploye JOIN Equipes e ON emp.idEquipe = e.idEquipe JOIN Departements d ON e.idDepartement = d.idDepartement WHERE emp.idEmploye = 1;
```

FIN DE DOCUMENT.

