

# Exercices d'application en SQL

---

## Jour 4 - Les jointures



# Présentation des 4 tables pour les exercices d'application

## 1. Employés (employees.csv) :

- **Colonnes** : id, nom, email, id\_departement, salaire, date\_embauche, age, genre.
- **Description** : Cette table contient des informations détaillées sur chaque employé, y compris son identifiant, nom, adresse e-mail, l'identifiant de son département, son salaire, sa date d'embauche, son âge et son genre.
- **Relation** : Reliée à la table des départements via id\_departement et à la table Employés\_Projets via id.

## 2. Départements (departments.csv) :

- **Colonnes** : id, nom\_departement.
- **Description** : Liste des départements avec un identifiant unique et un nom pour chaque département.
- **Relation** : Reliée à la table des employés via id (comme id\_departement dans la table des employés).

## 3. Projets (projects.csv) :

- **Colonnes** : id, nom\_projet, id\_chef, budget, date\_limite.
- **Description** : Contient des détails sur les projets, y compris un identifiant de projet, le nom du projet, l'identifiant du chef de projet (un employé), le budget du projet et sa date limite.
- **Relation** : Reliée à la table des employés via id\_chef et à la table Employés\_Projets via id.

## 4. Employés\_Projets (employees\_projects.csv) :

- **Colonnes** : id\_employe, id\_projet.
- **Description** : Une table de jointure Many-to-Many qui relie les employés aux projets. Chaque ligne représente l'association d'un employé à un projet spécifique.
- **Relation** : Reliée à la table des employés via id\_employe et à la table des projets via id\_projet.

## Exercice 1 : Association Employés-Départements

**Problématique** : Vous souhaitez obtenir une liste des employés avec le nom de leur département respectif pour mieux comprendre la répartition des employés dans l'entreprise.

**Ce qu'il faut faire** : Utilisez INNER JOIN pour combiner les tables employes et departements sur les colonnes id\_departement et id afin d'afficher le nom de chaque employé avec le nom de son département.

## Exercice 2 : Détails des projets et chefs de projets

**Problématique** : L'entreprise souhaite avoir un aperçu des projets en cours, y compris le nom du projet et le nom du chef de projet pour chaque projet.

**Ce qu'il faut faire** : Effectuez un INNER JOIN entre les tables projets et employes en utilisant id\_chef de projets et id de employes pour lister le nom de chaque projet avec le nom de son chef de projet.

### Exercice 3 : Employés sans projets

**Problématique :** Identifier les employés qui ne sont actuellement assignés à aucun projet.

**Ce qu'il faut faire :** Effectuez un LEFT JOIN entre employes et employes\_projets en utilisant id de employes et id\_employe de employes\_projets. Sélectionnez les employés où il n'y a pas de correspondance dans employes\_projets avec IS NULL.

### Exercice 4 : Tous les employés et leurs projets

**Problématique :** Obtenir une liste complète des employés avec les projets sur lesquels ils travaillent, y compris ceux qui ne travaillent sur aucun projet.

**Ce qu'il faut faire :** Effectuez un LEFT JOIN entre employes et employes\_projets sur id de employes et id\_employe de employes\_projets, puis un autre LEFT JOIN avec projets sur id\_projet pour inclure les détails du projet.

## Exercice 5 : Calcul du salaire moyen par département

**Problématique :** Pour une analyse budgétaire, l'entreprise souhaite connaître le salaire moyen dans chaque département.

**Ce qu'il faut faire :** Effectuez un LEFT JOIN entre employes et departements sur id\_departement. Utilisez la fonction d'agrégation AVG() pour calculer le salaire moyen (salaire) groupé par nom\_departement.

## Exercice 6 : Budget total des projets par chef de projet

**Problématique :** Pour une meilleure gestion financière, l'entreprise veut connaître le budget total géré par chaque chef de projet.

**Ce qu'il faut faire :** Utilisez un LEFT JOIN pour combiner employes et projets sur id de employes et id\_chef de projets. Appliquez la fonction SUM() pour calculer le budget total (budget) des projets pour chaque chef de projet, en regroupant les résultats par le nom du chef (nom de employes).

# Exercices supplémentaires : défis pour aller plus loin



## Exercice 7 : Analyse de la masse salariale par département

**Problématique** : La direction souhaite optimiser les coûts et demande une analyse de la masse salariale par département pour identifier où les ressources sont le plus investies.

**Ce qu'il faut faire** : Calculer la somme totale des salaires pour chaque département.

## Exercice 8 : Allocation des ressources aux projets

**Problématique** : Pour mieux gérer les ressources, le management veut savoir combien d'employés travaillent sur chaque projet afin d'assurer une allocation équilibrée des effectifs.

**Ce qu'il faut faire** : Trouver le nombre d'employés affectés à chaque projet et le nom du projet.

## Exercice 9 : Performance des départements

**Problématique** : Afin d'évaluer la performance des différents départements, la direction souhaite connaître le département qui a le plus grand nombre d'employés impliqués dans les projets.

**Ce qu'il faut faire** : Identifier le département avec le plus grand nombre d'employés participant à des projets.

## Exercice 10 : Projet avec le moins de ressources allouées

**Problématique** : Identifier le projet qui a le moins d'employés affectés pour potentiellement réallouer des ressources et assurer son succès.



## Exercice 11 : Optimisation des coûts des projets

**Problématique** : Dans le but d'optimiser les coûts, la direction veut identifier les projets avec un budget supérieur à la moyenne des budgets de tous les projets.

**Ce qu'il faut faire** : Sélectionner les projets dont le budget est supérieur à la moyenne des budgets de tous les projets.

Indice : il faut utiliser une sous requête SQL dans la clause WHERE.