

TP SQL AVANCE “GalaxiEat”



Introduction du projet :

Nous sommes en l'an **2487**. La chaîne de restaurants **GalaxiEat™** connaît un succès fou à travers l'univers : burgers d'astéroïdes, pizzas à l'antimatière douce, smoothies de nébuleuse...

Le PDG, un alien à huit bras extrêmement stressés, a décidé de moderniser le système d'information de la chaîne. Votre équipe est mandatée pour :

Construire et manipuler la base SQL qui gère :

- ★ Les restaurants répartis dans la galaxie
 - ★ Les employés
 - ★ Les plats proposés
- ★ Les commandes des clients
 - ★ Les ingrédients utilisés

Objectifs pédagogiques

- ★ Créer et modifier des tables
- ★ Insérer / mettre à jour / supprimer des données
 - ★ Effectuer des jointures
 - ★ Produire des agrégations (SUM, AVG)
 - ★ Trier, filtrer, manipuler les NULL
- ★ Écrire des requêtes réalistes dans un vrai scénario professionnel
- ★ Structurer progressivement une base complète

Création de l'univers “GalaxiEat”

Mise en place des tables (DDL)

Créer les tables suivantes :

Restaurants	Employees	Dishes	Orders	OrderItems
IdRestaurant	IdEmployees	IdDishes	IdOrders	IdOrderItems
Name	Firstname	Name	IdRestaurant	IdOrders
Planet	Lastname	Price	Total_amount	IdDishes
Opening_year	Role	Category	Customer_name	Quantity
	IdRestaurant			

Ajouts et modifications de colonnes

- ★ Ajouter à **Employees** la colonne : hire_date.
- ★ Ajouter à **Dishes** la colonne : is_vegan (NULL possible).
- ★ Renommer la table **Orders** → **CustomerOrders**.

Insérer de la donnée (libre de choix)

Restaurants : “Milky Way Diner”, planète : Terre (exemple)

Dishes : “Burger d’Astéroïde”, 12.5, “Burgers” (exemple)

Employees : Zorglub → Chef cuisinier (exemple)

Premières requêtes

- ★ Lister tous les restaurants de la galaxie.
- ★ Lister tous les plats classés par prix décroissant.
 - ★ Lister tous les employés triés par rôle.
 - ★ Lister tous les plats végétariens
- ★ Lister les plats dont le prix est supérieur à la moyenne des prix.

Gestion des null

- ★ Trouver les plats dont le champ `is_vegan` n’a pas encore été rempli.
- ★ Trouver les employés dont la `hire_date` est NULL.

Exploitation et gestion quotidienne



Création et gestion des commandes

- ★ Insérer 3 commandes dans différents restaurants.
- ★ Ajouter des OrderItems associés.
- ★ Calculer pour chaque commande : (Nombre total d'articles / montant total)
- ★ Lister les commandes triées par montant décroissant.

Jointures

- ★ Liste des employés avec le nom de leur restaurant.
- ★ Liste des plats commandés avec : (name (Dishes) / (name of customers) / planète (Restaurant))
- ★ Lister tous les restaurants avec leur nombre d'employés.

Mise à jour intelligente

Le PDG veut appliquer une réduction exceptionnelle :

- ★ Si le plat coûte > 12 → réduction de 10%
- ★ Sinon → réduction de 5%

Suppressions

- ★ Supprimer tous les plats à prix NULL (si existant)
- ★ Supprimer les commandes de test dont le total est inférieur à 5.

Mini-analyse finale

- ★ Prix moyen des plats par catégorie.
- ★ Montant total des ventes.
- ★ Les 3 plats les plus chers.
- ★ Les employés dont le nom contient la lettre “a”

Bonus (facultatif)

- ★ Créer une table **Ingredients** + relation DishIngredients
 - ★ Générer un “menu vegan” automatique
- ★ Créer un classement des restaurants par ancienneté
- ★ Simuler une “commande la plus haute de la semaine”

Livrables attendus

Le travail demandé consiste à créer un **fichier SQL complet**. Ce fichier doit contenir : la **création des tables**, l'**insertion des données**, puis **toutes les requêtes du TP**.

En plus du fichier SQL, il faut préparer un **mini-rapport**, en PDF. Ce rapport doit expliquer simplement la **structure de la base de données**, décrire les **difficultés rencontrées**, et montrer les **résultats importants** sous forme de copies ou d'exemples.

En option, il est aussi possible d'ajouter un **schéma de la base de données**, réalisé avec un outil comme Draw.io ou DBDiagram.