

## TD – Windowing Functions

### Prérequis :

Installer docker desktop : <https://www.docker.com/products/docker-desktop/>

Installer un client SQL pour mariadb <https://dev.mysql.com/downloads/workbench/>





Lancer docker desktop

Lancer le client SQL

Exécuter en powershell la commande docker compose up dans le répertoire du TD

Showing 2 items

Search

	NAME	IMAGE	STATUS	PORT(S)	STARTED	ACTIONS
<input type="checkbox"/>	 docker-mariadb 1 container	-	Running (1/1)	-		  
<input type="checkbox"/>	 mariadb a7a9aac3d51b	mariadb:latest	Running	3306	23 minutes ago	  

### 1. Se connecter avec un client SQL

Connect to Database

Stored Connection:  Select from saved connection settings

Connection Method: Standard (TCP/IP) Method to use to connect to the RDBMS

Parameters SSL Advanced

Hostname:  Port:  Name or IP address of the server host - and TCP/IP port.

Username:  Name of the user to connect with.

Password:   The user's password. Will be requested later if it's not set.

Default Schema:  The schema to use as default schema. Leave blank to select it later.

### 2. Exécuter le script : Create objects

Dans ce TD, nous allons utiliser des fonctionnalités de requêtes avancées. La base de données sera la base de ressources humaines.

### Exercice :

1. Créez une requête qui permet de trouver pour chaque département les employés avec le plus grand salaire. Pour cette question, on pourra faire une sous requête, un join ...
2. Même question mais cette fois en utilisant les fonctions de « windowing ». Mot-clé ROW\_NUMBER()
3. En utilisant une fonction de « windowing », votre requête doit renvoyer pour chaque employé l'EMPLOYEE\_ID de la personne qui a le salaire au dessus. Pour cela, on peut trier les enregistrements comment en question 2 et faire une jointure
4. Même question que précédemment mais en évitant de faire une jointure. Mot-clé LAG.
5. Renvoyez pour chaque employé, toutes les informations de la table + le salaire moyen par département. Pour cela, vous pouvez utiliser une fonction de « windowing » avec le mot clé AVG.
6. Même question précédemment mais ajouter une colonne qui dit si l'employé est au-dessus ou au-dessous du salaire moyen.
7. Pour chaque employé, renvoyez son ID, nom, prénom, salaire et la différence entre son salaire et celui de son manager.
8. Pour chaque employé, renvoyez son ID, nom, prénom, salaire et le nombre de salarié du département
9. Pour chaque employé, renvoyez la somme des salaires de tous les employés qui ont le même job\_id. Ensuite calculez un tri pour chaque somme.

Ex :    EMP1   JOB1   30000   2  
         EMP2   JOB1   30000   2  
         EMP3   JOB2   35000   3  
         EMP4   JOB3   5000    1

10. Pour chaque employé, trie-les par salaire décroissant (quel que soit le département). Pour chaque salarié, renvoyez le salarié qui a le salaire juste en dessous (soit la ligne suivante). Même question avec le salarié 2 lignes en dessous.
11. Utilisez le script *doublons.sql* et exécutez-le sur votre base de données. La table contient de nombreux doublons. Proposez un moyen de les supprimer.