Université d'Alger1

Faculté des Sciences

Département : Informatique

Filière : Informatique

Spécialité : Master1 de l'Ingénierie des Systèmes Informatique Intelligents

Module : Entrepôt de données

Entrepôt de données Climatiques

Réalisé par :
Ouadahi Katia
Bouchareb Maroua
Labtani Daouya Sara

2023/2024

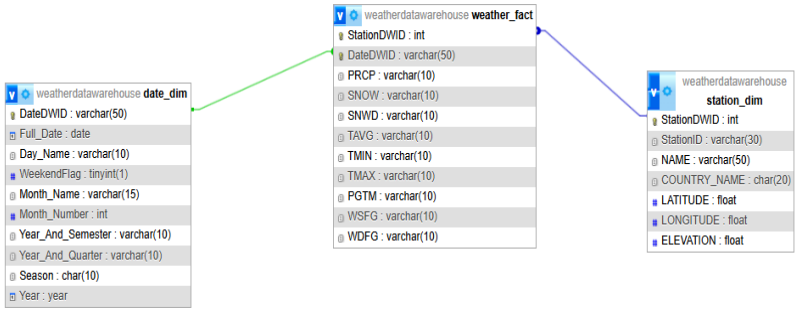
Table de Contenu:

Introduction

Ce projet vise à construire un entrepôt de données (Data Warehouse) pour centraliser et analyser les données météorologiques de diverses stations situées dans troi pays du Maghreb (Algérie , Maroc et Tunisie). Le projet utilise Python pour l'extraction, la transformation et le chargement (ETL) des données provenant de fichiers CSV dans une base de données MySQL. Les données centralisées permettent de créer des rapports et des tableaux de bord pour une analyse approfondie.

1. Le schéma en étoile de l'entrepôt de données

Le schéma en étoile est un modèle de base de données utilisé dans les entrepôts de données pour structurer les informations de manière optimisée et assure un haut niveau de performance des requêtes surtout sur la grande masse des données. Dans notre projet, Il se compose d'une table des faits centrale entourée de plusieurs tables de dimensions comme indiquer dans le schéma suivant:



1. **<u>Tables de Dimension</u>:** Les tables de dimensions fournissent des contextes descriptifs autour des données contenues dans la table des faits. Pour notre entrepôt de données climatiques, nous avons les deux tables de dimensions suivantes :
   - ***station_dim***: Contient des informations sur les stations, et sa description est :

Field	Type	Null	Key	Default	Extra
StationDWID	int	NO	PRI	NULL	auto_increment
StationID	varchar(30)	YES		NULL	
NAME	varchar(50)	YES		NULL	
COUNTRY_NAME	char(20)	YES		NULL	
LATITUDE	float	YES		NULL	
LONGITUDE	float	YES		NULL	
ELEVATION	float	YES		NULL	

La difference entre StationDWID et StationID:

-<u>StationDWID</u>: est le Clé primaire (PK) de la table "*station_dim*", garantissant l'unicité de chaque enregistrement dans la table.

-<u>StationID</u>: est un attribut de la table "*station_dim*", représente le code réel d'une station météorologique, qui ne peut pas être unique puisqu'une station peut enregistrer des statistiques sur plusieurs jours.

Voici un exemple de quelques enregistrements dans la table"*station_dim*" :

StationDWID	StationID	NAME	COUNTRY_NAME	LATITUDE	LONGITUDE	ELEVATION
1	AGE00147709	FORT NATIONAL, AG	Algerie	36.63	4.2	942
2	AGE00147709	FORT NATIONAL, AG	Algerie	36.63	4.2	942
3	AGE00147709	FORT NATIONAL, AG	Algerie	36.63	4.2	942
4	AGE00147709	FORT NATIONAL, AG	Algerie	36.63	4.2	942
5	AGE00147709	FORT NATIONAL, AG	Algerie	36.63	4.2	942

- ***date_dim*** : Fournit des détails temporels, et sa description est :

Field	Type	Null	Key	Default	Extra
DateDWID	varchar(50)	NO	PRI	NULL	
Full_Date	date	YES		NULL	
Day_Name	varchar(10)	YES		NULL	
WeekendFlag	tinyint(1)	YES		NULL	
Month_Name	varchar(15)	YES		NULL	
Month_Number	int	YES		NULL	
Year_And_Semester	varchar(10)	YES		NULL	
Year_And_Quarter	varchar(10)	YES		NULL	
Season	char(10)	YES		NULL	
Year	year	YES		NULL	

La difference entre DateDWID et Full_date:

-DateDWID: est le Clé primaire (PK) de la table "*date_dim*", générée pour garantir l'unicité de chaque enregistrement dans la table. Elle est dérivée de "Full_date" et potentiellement "Station" qui représente le code réel de la station dans les fichiers plats.

-Full_date: est un attribut qui représente la date réelle , Cette colonne n'est pas nécessairement unique car elle ne tient pas compte des différentes stations.

Voici un exemple de quelques enregistrements dans la table "date_dim" :

DateDWID	Full_Date	Day_Name	WeekendFlag	Month_Name	Month_Number	Year_And_Semester	Year_And_Quarter	Season	Year
19200101AGE00147706	1920-01-01	Thursday	0	January	1	1920-S1	1920Q1	Winter	1920
19200101AGE00147711	1920-01-01	Thursday	0	January	1	1920-S1	1920Q1	Winter	1920
19200101AGE00147712	1920-01-01	Thursday	0	January	1	1920-S1	1920Q1	Winter	1920
19200101AGE00147713	1920-01-01	Thursday	0	January	1	1920-S1	1920Q1	Winter	1920
19200101AGE00147715	1920-01-01	Thursday	0	January	1	1920-S1	1920Q1	Winter	1920

2. **Table des Faits:** La table des faits centralise les mesures quantitatives des données climatiques. Et elle comporte des clés étrangères, qui ne sont autres que les clés primaires des tables de dimension, dans notre cas, la table des faits est la table suivante:

- **weather_fact:** Contient les mesures climatiques collectées par les stations.

Field	Type	Null	Key	Default	Extra
StationDWID	int	NO	PRI	NULL	auto_increment
DateDWID	varchar(50)	NO	PRI	NULL	
PRCP	varchar(10)	YES		NULL	
SNOW	varchar(10)	YES		NULL	
SNWD	varchar(10)	YES		NULL	
TAVG	varchar(10)	YES		NULL	
TMIN	varchar(10)	YES		NULL	
TMAX	varchar(10)	YES		NULL	
PGTM	varchar(10)	YES		NULL	
WSFG	varchar(10)	YES		NULL	
WDFG	varchar(10)	YES		NULL	

"StationDWID" et "DateDWID" sont les clés étrangères référencées respectivement à la table "station_dim" et "date_dim" ainsi ils sont les clés primaires de cette table.

Les Mesures: Les mesures dans la table des faits sont les valeurs quantitatives sur lesquelles les analyses sont effectuées. Pour notre entrepôt de données climatiques, les mesures incluent :

PRCP : Précipitation	**SNOW** : Chute de neige
SNWD : Épaisseur de la neige	**TAVG** : Température moyenne
TMIN : Température minimale	**TMAX** : Température maximale
PGTM : Heure de pointe des rafales	**WSFG** : Vitesse maximale du vent en rafale
WDFG : Direction de la rafale de vent maximale	

Voici un exemple de quelques enregistrements dans la table "weather_fact" :

DateDWID	Full_Date	Day_Name	WeekendFlag	Month_Name	Month_Number	Year_And_Semester	Year_And_Quarter	Season	Year
19200101AGE00147706	1920-01-01	Thursday	0	January	1	1920-S1	1920Q1	Winter	1920
19200101AGE00147711	1920-01-01	Thursday	0	January	1	1920-S1	1920Q1	Winter	1920
19200101AGE00147712	1920-01-01	Thursday	0	January	1	1920-S1	1920Q1	Winter	1920
19200101AGE00147713	1920-01-01	Thursday	0	January	1	1920-S1	1920Q1	Winter	1920
19200101AGE00147715	1920-01-01	Thursday	0	January	1	1920-S1	1920Q1	Winter	1920

Cette schéma a été créé en utilisant le script Python "**Create_DW_Schema.py**"

2. La Granularité de la Table des Faits

La granularité de la table des faits correspond au niveau de détail des données enregistrées. Dans notre cas, la granularité est définie par le niveau journalier des mesures climatiques pour chaque station météorologique. Chaque enregistrement dans la table des faits représente une mesure quotidienne de plusieurs paramètres climatiques pour une station spécifique.

3. Le Processus Traité par l'Entrepôt de Données

1. Extraction les données :

Pour l'extraction des données climatiques, nous avons récupéré des jeux de données au format CSV provenant de différents pays. Nous avons ensuite utilisé Python et la bibliothèque pandas pour lire ces fichiers et les convertir en DataFrames. Un DataFrame est une structure de données à deux dimensions, similaire à une table de base de données ou à une feuille de calcul Excel, qui permet de manipuler et analyser facilement des données structurées. Cette conversion nous a permis de manipuler et traiter les données de manière efficace pour les étapes suivantes du processus ETL.

2. Transformation des données :

Les données que nous avons à disposition nécessitent un traitement avant d'être chargées dans l'entrepôt de données. Pour unifier le nombre d'attributs de toutes les dataframes, nous avons :

- ✓ Supprimé les attributs non pertinents tels que PRCP_ATTRIBUTES et TMIN_ATTRIBUTES.
- ✓ Ajouté,à partir de l'attribut date, les attributs Day_Name, Month_Name, Year et Semester pour pouvoir remplir la table de dimension ''date_dim''.
- ✓ Ajouté, à partir de l'attribut Name (FORT NATIONAL, AG), l'attribut Country_Name. Ce dernier prend la valeur Algeria si le deuxième élément de Name est AG, Morocco si égal à SP ou MO, et Tunisia si égal à TS.

Ensuite, nous avons rempli les valeurs manquantes en utilisant différentes stratégies, telles que le remplissage par la moyenne, la valeur précédente ou la valeur suivante. Enfin, nous avons concaténé toutes les dataframes horizontalement pour obtenir un seul fichier CSV contenant nos données nettoyées et prêtes à être utilisées.

La réalisation des deux étapes, extraction et transformation, se fait en utilisant le script Python ''**ET_climatic_data.py''**.

3. Chargement des données :

Les données transformées sont chargées dans les tables de l'entrepôt de données "weatherdatawarehouse" en utilisant le script Python “**Load_data_into_DW.py**”.

4. tableau de bord

1. Structure du Répertoire :

Le répertoire « Dashboard » contient les fichiers suivants :

dataFetch.py : Ce fichier regroupe les fonctions responsables de la récupération des données depuis la base de données. Il inclut les requêtes SQL pour obtenir les données nécessaires en fonction des sélections de l'utilisateur sur le tableau de bord.

layout.py : Ce fichier contient les fonctions de création de la mise en page du tableau de bord. Il définit la structure visuelle de l'application, y compris les différents éléments d'interface utilisateur comme les menus déroulants, les graphiques et la carte.

app.py : Ce fichier est le point d'entrée principal de l'application. Il configure et lance le serveur Dash, définit la disposition initiale du tableau de bord et les callbacks nécessaires pour l'interactivité.

2. Environnement de Développement :

Pour faire fonctionner le tableau de bord, les éléments suivants sont nécessaires :

Un IDE Python pour modifier et exécuter les fichiers Python, par exemple Visual Studio Code.

Un serveur MySQL fonctionnel.

PhpMyAdmin ou un outil similaire pour gérer la base de données et vérifier les données.

3. Instructions d'Installation et d'Exécution :

Suivez les étapes ci-dessous pour mettre en place et exécuter le tableau de bord :

Configuration de l'Environnement :

- Assurez-vous que Python est installé sur votre machine.
- Installez les bibliothèques nécessaires en exécutant la commande suivante dans votre terminal :
- pip install pymysql pandas dash plotly dash-bootstrap-components

Préparation du data warehouse :

- ouvrez le dossier "DATAWAREHOUSE-PROJECT" dans votre environnement de développement intégré
- Connectez-vous à la base de données dans phpMyAdmin avec votre nom d'utilisateur et votre mot de passe.
- Mettez à jour votre nom d'utilisateur et votre mot de passe dans les fichiers "Create_DW_Schema.py", "Load_data_into_DW.py" et "dataFetch.py" dans la partie du connexion à la base de données.
- Si vous vous connectez avec user='root' et password='', vous n'avez pas besoin de modifier les fichiers car la connexion par défaut est déjà configurée.
- Le dossier "weather_data" contient les données sous forme de fichiers CSV.
- Exécutez le fichier "ET_climatic_data.py" pour nettoyer les données. Un fichier "climatic_dataSet.csv" sera généré dans le même répertoire.
- Exécutez le fichier "Create_DW_Schema.py" pour créer la base de données. Le schéma du datawarehouse sera créé dans phpMyAdmin.
- Exécutez le fichier "Load_climatic_data_intoDW.py" pour charger les données du fichier "climatic_dataSet.csv" dans le datawarehouse.

Altrnative : Importation Directe

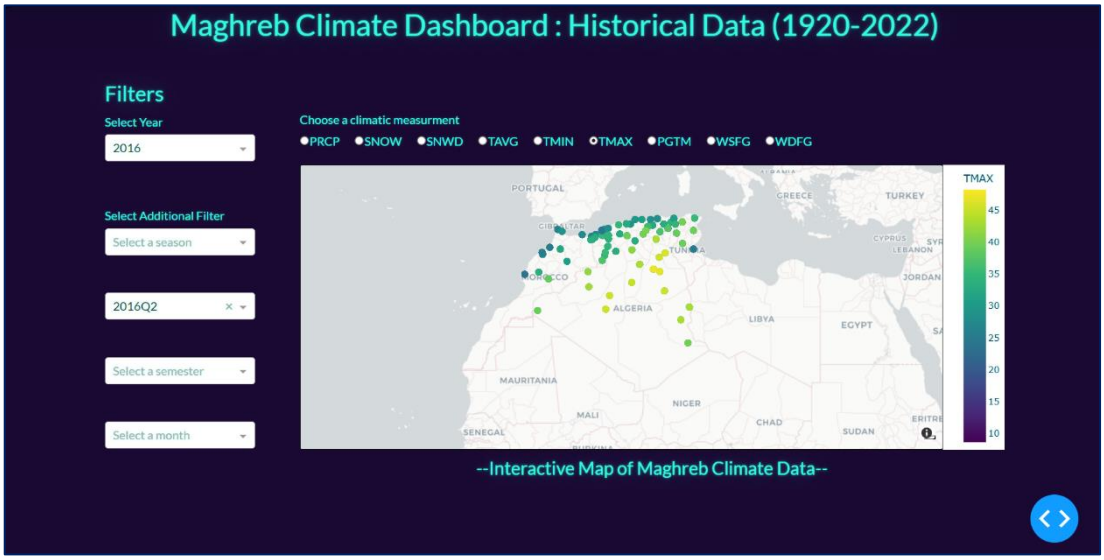
Vous pouvez également importer directement la base de données attaché weatherdatawarehouse.sql dans phpMyAdmin pour créer le datawarehouse.

Exécution du Dashboard :

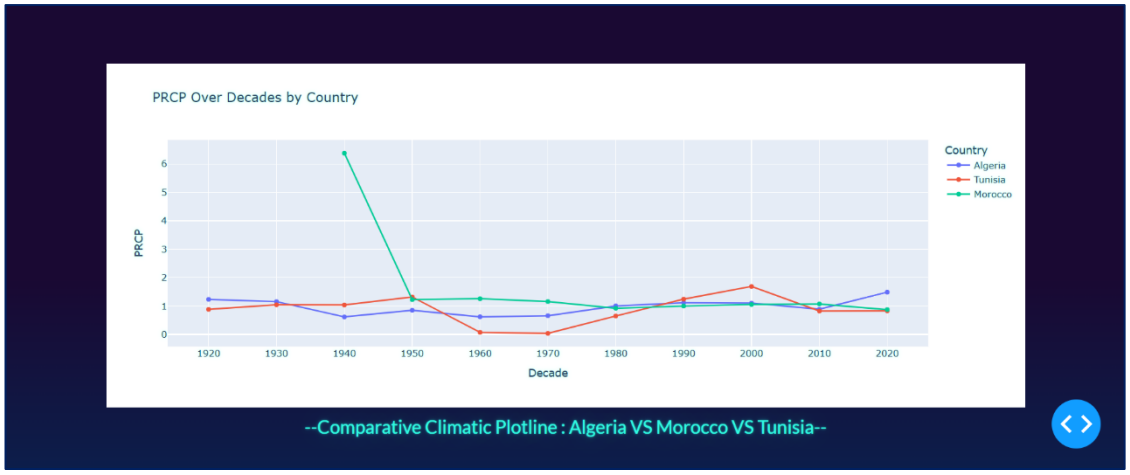
- Exécutez le fichier "app.py" en utilisant votre environnement Python. Cela démarrera le serveur Dash pour l'application.
- Accédez à l'application via votre navigateur en ouvrant l'adresse http://127.0.0.1:8050/.
- Lors du téléchargement de données ou de mises à jour suite à une interaction, la fenêtre affichera le nom "Updating". Une fois le dashboard mis à jour, la fenêtre affichera le nom "Dash".

Contenu du Dashboard :

- Le dashboard visualise les données du data warehouse.
- Il comprend une Carte interactive des données climatiques du Maghreb et un Graphique climatique comparatif pour l'Algérie, le Maroc et la Tunisie.
- Des filtres sont disponibles pour assurer une interaction dynamique



Capture d'écran du tableau de bord affichant la carte interactive des données climatiques du Maghreb



Capture d'écran du tableau de bord présentant le graphique climatique comparatif pour l'Algérie, le Maroc et la Tunisie

Manipulation du tableau de bord :

Une fois le tableau de bord lancé, vous pouvez l'utiliser comme suit :

- **Sélection de l'Année :** Utilisez le menu déroulant pour sélectionner une année spécifique. Par défaut, l'année 1920 est sélectionnée.

- **Sélection d'un Filtre Additionnel :** Vous pouvez affiner votre recherche en sélectionnant un filtre seul supplémentaire (saison, semestre, trimestre, mois) pour voir les données climatiques spécifiques à cette période.
- **Choix de la Mesure Climatique :** Utilisez les boutons radio pour choisir la mesure climatique que vous souhaitez visualiser (par exemple, PRCP, TAVG, TMIN, TMAX, etc.). Le graphique et la carte seront mis à jour en conséquence.
- **Interactivité du Tableau de Bord :** La carte affiche les données climatiques sous forme de points colorés représentant les différentes stations météorologiques. Vous pouvez survoler les points pour voir les détails de chaque station.
- Le graphique linéaire montre l'évolution de la mesure climatique choisie sur une période décennale pour chaque pays du Maghreb (Maroc, Algérie, Tunisie).

Conclusion:

Ce projet a permis de concevoir et mettre en œuvre un entrepôt de données pour l'analyse des données météorologiques du Maghreb, ainsi qu'un tableau de bord interactif pour visualiser ces données. Les points clés à retenir sont :

- Processus ETL : Extraction, transformation et chargement des données réalisés avec succès.
- Conception de l'Entrepôt de Données : Utilisation efficace du modèle en étoile pour structurer les données.
- Tableau de Bord Interactif : Création d'un tableau de bord interactif avec Dash et Plotly pour une analyse visuelle approfondie.

Cependant, il reste des opportunités d'amélioration. Nous aurions aimé ajouter davantage de fonctionnalités, telles que des filtres supplémentaires comme le week-end, mais le temps ne nous a pas permis de le faire dans le cadre de ce projet. Ces perspectives d'amélioration pourraient être explorées dans des travaux futurs pour enrichir davantage l'expérience utilisateur et l'analyse des données météorologiques.