

RÉPUBLIQUE DÉMOCRATIQUE ET POPULAIRE D'ALGÉRIE
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE
SCIENTIFIQUE

UNIVERSITÉ DES SCIENCES ET DE LA TECHNOLOGIE HOUARI
BOUMEDIENE

FACULTÉ D'INFORMATIQUE
DÉPARTEMENT DE SYSTÈME INFORMATIQUE
SPÉCIALITÉ : BIG DATA ANALYTICS
GRADE : MASTER 2



Rapport TP2 du Module BI

CRÉATION DE L'ENTREPÔT DE DONNÉES
DATAWHCOMMANDE

Réalisé par :

YAKOUBI Sara

Matricule : 212131055030

Module : Business intelligence (BI)

Groupe : 2

Travail présenté à :

Mme Fatma Zohra MEKAHLIA

Année académique : 2025/2026

Table des matières

Introduction	2
Création de l'Entrepôt de Données DataWHCommande	2
Création de la Base de Données DataWHCommande	2
Script SQL de Création de l'Entrepôt de Données	2
Visualisation des Tables dans l'Explorateur d'Objets	3
Diagramme Relationnel de l'Entrepôt DataWHCommande	4
Étapes de création du diagramme dans SSMS	4
Structure des Tables	5
Contraintes et Relations	6
Conclusion	6

Introduction

Ce rapport présente les étapes de création de l'entrepôt de données **DataWHCom-mande**. L'objectif est de centraliser et structurer les informations relatives aux com-mandes, clients et employés afin de faciliter l'analyse décisionnelle. La conception repose sur le modèle **dimensionnel** avec des tables de dimensions et une table de faits.

Création de l'Entrepôt de Données DataWHCommande

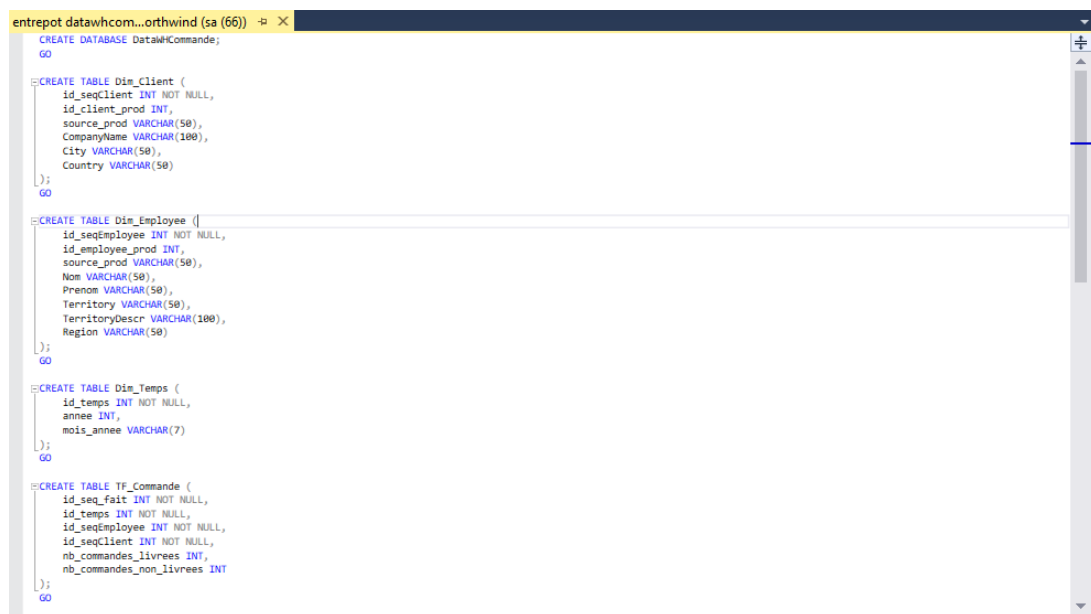
Création de la Base de Données DataWHCommande

La première étape de l'implémentation de l'entrepôt consiste à créer la base de données dans SQL Server 2022. Le script SQL suivant est exécuté dans SQL Server Management Studio (SSMS) :

```
CREATE DATABASE DataWHCommande;  
GO
```

Après exécution de cette commande, une nouvelle base de données nommée **DataWH-Commande** est créée et apparaît dans l'**Explorateur d'objets** de SSMS. Cette base servira de conteneur pour toutes les tables de dimensions et la table de faits de l'entrepôt.

Script SQL de Création de l'Entrepôt de Données



```
entrepot datawhcom...orthwind (sa (66)) X  
CREATE DATABASE DataWHCommande;  
GO  
  
--CREATE TABLE Dim_Client (  
id_seqClient INT NOT NULL,  
id_client_prod INT,  
source_prod VARCHAR(50),  
CompanyName VARCHAR(100),  
City VARCHAR(50),  
Country VARCHAR(50)  
);  
GO  
  
--CREATE TABLE Dim_Employee (  
id_seqEmployee INT NOT NULL,  
id_employee_prod INT,  
source_prod VARCHAR(50),  
Nom VARCHAR(50),  
Prenom VARCHAR(50),  
Territory VARCHAR(50),  
TerritoryDescr VARCHAR(100),  
Region VARCHAR(50)  
);  
GO  
  
--CREATE TABLE Dim_Temps (  
id_temps INT NOT NULL,  
annee INT,  
mois_annee VARCHAR(7)  
);  
GO  
  
--CREATE TABLE TF_Commande (  
id_seq_fait INT NOT NULL,  
id_temps INT NOT NULL,  
id_seqEmployee INT NOT NULL,  
id_seqClient INT NOT NULL,  
nb_commandes_livrees INT,  
nb_commandes_non_livrees INT  
);  
GO
```

FIGURE 1 – Script SQL de création des tables de l'entrepôt DataWHCommande.

```

entrepot datawhcom...orthwind (sa (66))
--
CREATE TABLE TF_Commande (
    id_seq_fait INT NOT NULL,
    id_temps INT NOT NULL,
    id_seqEmployee INT NOT NULL,
    id_seqClient INT NOT NULL,
    nb_commandes_livrees INT,
    nb_commandes_non_livrees INT
);
GO

ALTER TABLE Dim_Client
ADD CONSTRAINT pk_Dim_Client PRIMARY KEY (id_seqClient);
GO

ALTER TABLE Dim_Employee
ADD CONSTRAINT pk_Dim_Employee PRIMARY KEY (id_seqEmployee);
GO

ALTER TABLE Dim_Temps
ADD CONSTRAINT pk_Dim_Temps PRIMARY KEY (id_temps);
GO

ALTER TABLE TF_Commande
ADD
CONSTRAINT fk_TF_Commande_Dim_Client FOREIGN KEY (id_seqClient) REFERENCES Dim_Client(id_seqClient),
CONSTRAINT fk_TF_Commande_Dim_Employee FOREIGN KEY (id_seqEmployee) REFERENCES Dim_Employee(id_seqEmployee),
CONSTRAINT fk_TF_Commande_Dim_Temps FOREIGN KEY (id_temps) REFERENCES Dim_Temps(id_temps),
CONSTRAINT pk_TF_Commande PRIMARY KEY (id_seq_fait, id_seqClient, id_seqEmployee, id_temps);
GO

```

FIGURE 2 – Script SQL de création des contraintes de l’entrepôt DataWHCommande.

Visualisation des Tables dans l’Explorateur d’Objets

Après l’exécution du script de création des dimensions et de la table de faits, l’entrepôt de données **DataWHCommande** apparaît dans l’Explorateur d’objets de SQL Server Management Studio. On peut y visualiser l’ensemble des tables :

- **Dim_Client** : informations sur les clients
- **Dim_Employee** : informations sur les employés
- **Dim_Temps** : informations sur la dimension temporelle
- **TF_Commande** : table de faits contenant les commandes livrées et non livrées

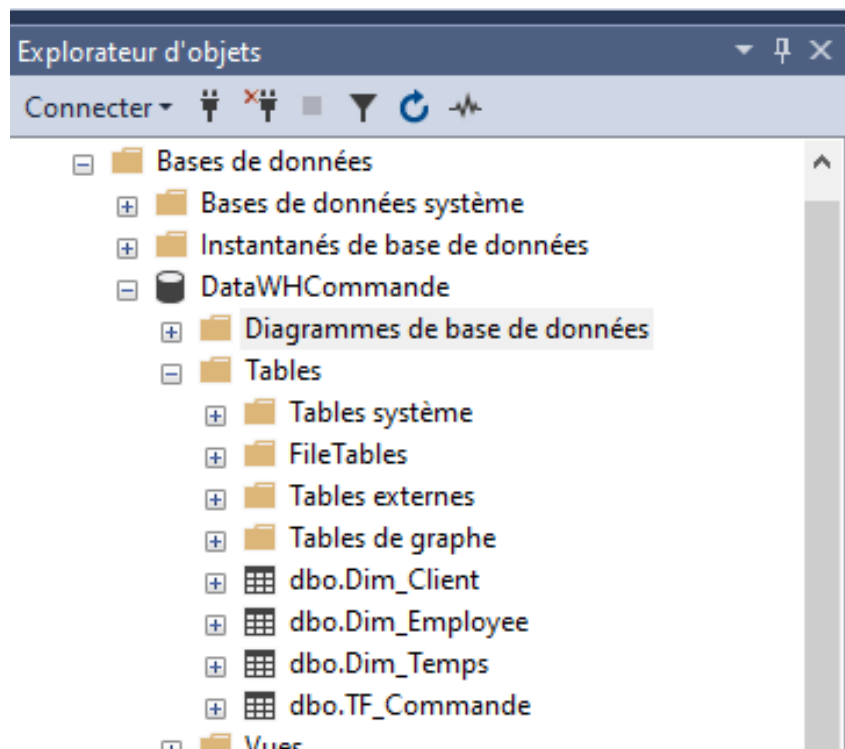


FIGURE 3 – Visualisation des tables dans l’Explorateur d’objets de SQL Server.

Diagramme Relationnel de l'Entrepôt DataWHCommande

Le diagramme relationnel permet de visualiser les relations entre la table de faits et les différentes tables de dimensions. On va créer ce diagramme afin d'obtenir une représentation graphique du modèle dimensionnel.

Étapes de création du diagramme dans SSMS

La création du diagramme se fait à travers l'**Explorateur d'objets**. Les étapes sont les suivantes :

1. Dans l'**Explorateur d'objets**, développer la base : **DataWHCommande**.
2. Faire un **clic droit** sur le dossier "**Diagrammes de base de données**" puis choisir : "*Nouveau schéma de base de données*".

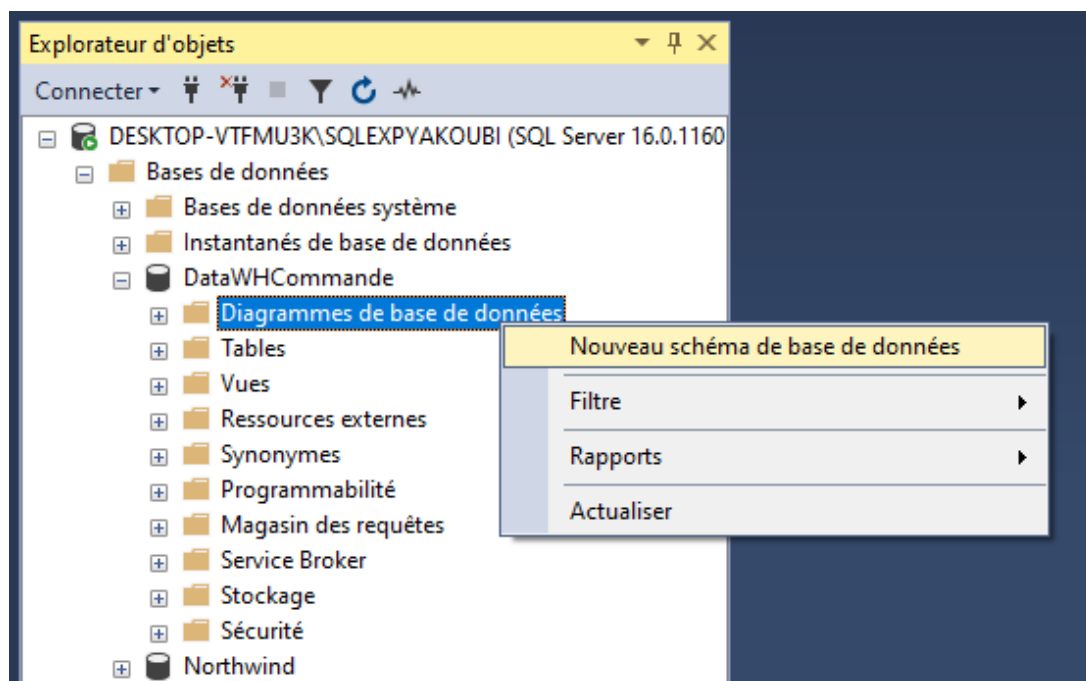


FIGURE 4 – Création d'un nouveau diagramme via un clic droit sur "Diagrammes de base de données".

4. Une fenêtre apparaît pour sélectionner les tables à inclure dans le diagramme. Sélectionner :
 - Dim_Client
 - Dim_Employee
 - Dim_Temps
 - TF_Commandepuis cliquer sur "*Ajouter*".

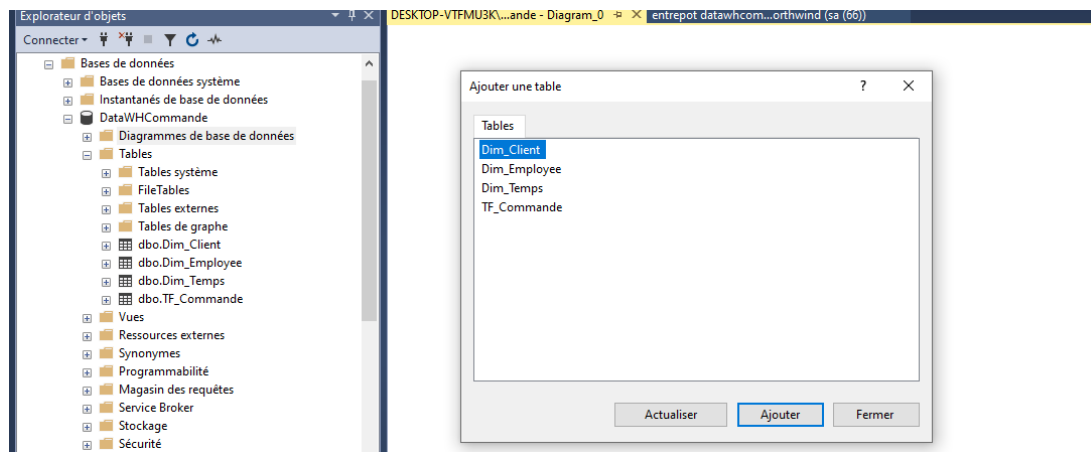


FIGURE 5 – Fenêtre d'ajout des tables au diagramme.

5. SSMS génère automatiquement le diagramme relationnel en affichant les liens entre les clés primaires et étrangères.
6. Le diagramme est ensuite enregistré dans le dossier **Diagrammes de base de données**.

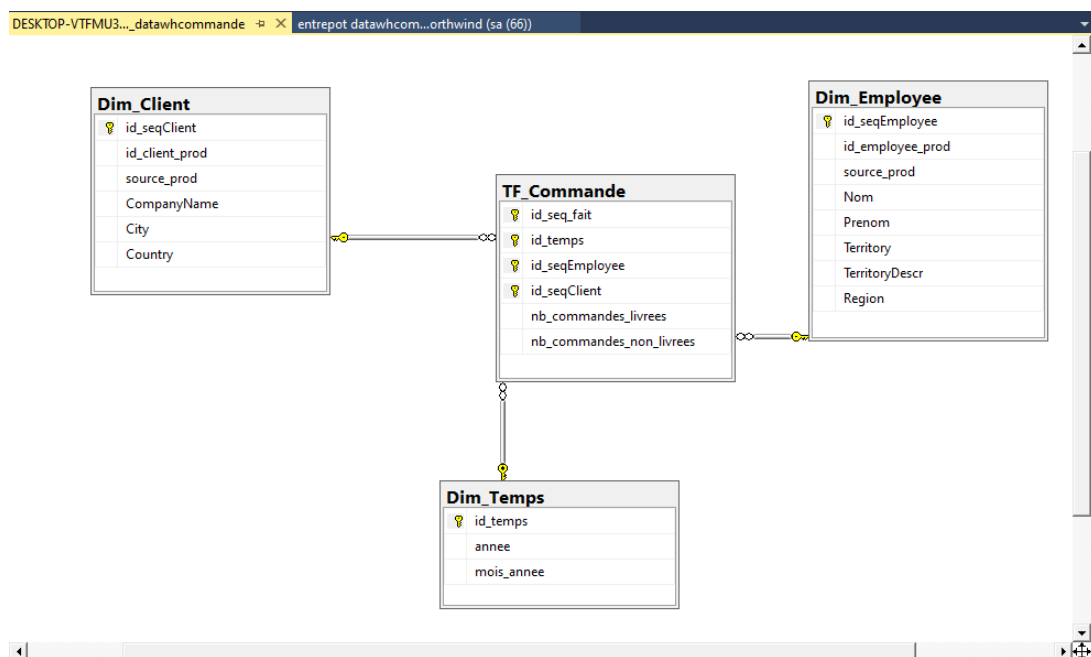


FIGURE 6 – Diagramme relationnel de l'entrepôt DataWHComande.

Structure des Tables

- **Dim_Client** : id_seqClient (PK), id_client_prod, source_prod, CompanyName, City, Country
- **Dim_Employee** : id_seqEmployee (PK), id_employee_prod, source_prod, Nom, Prenom, Territory, TerritoryDescr, Region
- **Dim_Temps** : id_temps (PK), annee, mois_annee
- **TF_Commande** : id_seq_fait (PK composite), id_temps (FK), id_seqEmployee (FK), id_seqClient (FK), nb_commandes_livrees, nb_commandes_non_livrees

Contraintes et Relations

Les relations entre la table de faits et les tables de dimensions assurent l'intégrité référentielle :

- **TF_Commande.id_seqClient** \rightarrow **Dim_Client.id_seqClient**
- **TF_Commande.id_seqEmployee** \rightarrow **Dim_Employee.id_seqEmployee**
- **TF_Commande.id_temps** \rightarrow **Dim_Temps.id_temps**

Toutes les tables de dimensions possèdent une clé primaire, et la table de faits possède une clé primaire composite sur (id_seq_fait, id_seqClient, id_seqEmployee, id_temps).

Conclusion

La création de l'entrepôt de données **DataWHCommande** permet de centraliser les informations relatives aux commandes et de suivre facilement les commandes des clients et le travail des employés.

Cette architecture dimensionnelle facilite les requêtes analytiques et la génération de rapports décisionnels.