



Thème :

Dirigée par :

Dr **El Bouhissi**

Présentée par :

Mr **Djaballi** Reda Nassim

Mr **Brahmi** Mokrane

Mr **Saidene** Samy

Mr **Kharfallah** Gibril

Table des matières

Table des matières	i
Table des figures	ii
1 Introduction Générale	1
2 Généralités sur le Data Warehouse et les Systèmes d'Information Décisionnels	2
2.1 Introduction aux Data Warehouses	2
2.2 Architecture d'un Data Warehouse	2
2.3 Systèmes d'Information Décisionnels (SID)	2
2.4 Importance des Data Warehouses dans les Systèmes d'Information Décisionnels	2
2.5 Exemple d'Utilisation : Commerce Électronique	3
2.6 Environnement de travail	3
2.7 Langages utilisés	3
2.8 Conclusion	4
3 Analyse du cas d'étude	4
3.1 Analyse du Cas Considéré	4
3.1.1 Objectifs de l'entreprise	4
3.1.2 Position sur le marché	4
3.1.3 Services ou produits proposés	4
3.1.4 Formes de revenu	4
3.1.5 Informations utiles pour la prise de décision	4
3.2 Actions / Opérations à Tracer	4
3.2.1 Traitements Possibles (Requêtes Analytiques)	5
3.2.2 Ordre d'importance des actions	5
4 Conception des Data-Marts	5
4.1 Dimensions Nécessaires aux Modèles	5
4.1.1 Dimensions pour les ventes	5
4.2 Diagramme de conception	5
4.2.1 Ventes (modélisation détaillée)	5
4.3 Réponses au traitements avec le modèle	6
4.4 Estimation taille des tables	7
5 Implementaton	7
5.1 Implémentation du modèle	7
5.2 Requête analytique	12
5.3 Création et Utilisation des Vues	16
5.3.1 définition des vues	16
5.3.2 Caractéristique des vues	17
5.3.3 Vues généré	17
6 Perspectives d'avenir	18
7 Conclusion Générale	18

Table des figures

1	Diagramme de Conception	6
2	Création de la table de dimension DimProduct	7
3	Création de la table de dimension DimSupplier	8
4	Création de la table de dimension DimCustomer	8
5	Création de la table de dimension DimDate	9
6	Création de la table de dimension DimPaymentMethod	9
7	Création de la table de dimension DimPromotion	10
8	Création de la table de dimension DimShipping	10
9	Création de la table FactSales	10
10	Insertion dans la table FactSales	11
11	Requête analytique (total des ventes par trimestre)	12
12	Résultats Requête (Total des ventes par trimestre)	12
13	Requête analytique(Les produits les plus vendus)	12
14	Résultats requete(Les produits les plus vendus)	13
15	Requête analytique(Montant totale des ventes par client)	13
16	Résultats requete (Montant totale des ventes par client)	13
17	Requête analytique (Ventes par méthode de paiement)	14
18	Résultats requete (Ventes par méthode de paiement)	14
19	Requête analytique (Détails des ventes pour une promotion spécifique)	14
20	Résultats requete (Détails des ventes pour une promotion spécifique)	14
21	Requête analytique (vente par catégories de produits)	15
22	Résultats requete (vente par catégories de produits)	15
23	Requête analytique(Montant total des ventes par mois)	15
24	Résultats requete(Montant total des ventes par mois)	15
25	Requête analytique (Détails de ventes effectué par un client spécifique)	16
26	Résultats requete (Détails de ventes effectué par un client spécifique)	16
27	Requête analytique (Ventes avec un coût d'expédition élevé)	16
28	Résultats requete (Ventes avec un coût d'expédition élevé)	16
29	Requête analytique (Nombre de clients par catégorie de produit)	17
30	Résultats requete (Nombre de clients par catégorie de produit)	17
31	Vue de Ventes par Client	18
32	Résultats de Ventes par Client	18
33	Vue de Ventes par Produit	18
34	Résultats de Ventes par Produit	19
35	Vue de Ventes par Promotion	19
36	Résultats de Ventes par Promotion	19

1 Introduction Générale

L'essor du commerce électronique a révolutionné la manière dont les entreprises opèrent, offrant des opportunités sans précédent pour atteindre une clientèle mondiale. Amazon, en tant que pionnier et leader de ce secteur, illustre parfaitement l'impact significatif que peut avoir une plateforme de commerce en ligne bien conçue et optimisée. Les entreprises de commerce électronique collectent une quantité massive de données à chaque interaction client, chaque transaction et chaque mouvement logistique. Transformer ces vastes volumes de données en informations exploitables est crucial pour améliorer les performances commerciales et prendre des décisions stratégiques éclairées. L'objectif de ce projet est de concevoir, implémenter et interroger un entrepôt de données (Data Warehouse) pour une entreprise de commerce électronique. Cet entrepôt de données servira de fondation pour des analyses avancées, permettant de répondre aux besoins décisionnels de l'entreprise. En nous basant sur un cas pratique inspiré du fonctionnement d'Amazon, nous allons explorer les différentes étapes nécessaires pour créer un Data Warehouse performant et utile.

Le projet s'articule autour des étapes suivantes :

1. **Analyse du cas considéré :** Comprendre les objectifs de l'entreprise, sa position sur le marché, les services ou produits qu'elle propose, ses formes de revenu, et les informations critiques pour la prise de décision.
2. **Identification des actions et opérations :** Déterminer les opérations clés à tracer pour récupérer les informations nécessaires.
3. **Définition des traitements analytiques :** Proposer des requêtes analytiques permettant d'aider à la prise de décision.
4. **Conception de l'entrepôt de données :** Créer un modèle en étoile (star schema) basé sur les traitements identifiés.
5. **Implémentation et requêtage :** Planter les tables de faits et de dimensions dans une base de données Oracle, et créer des requêtes analytiques pour tester le modèle.

En suivant ces étapes, nous allons démontrer comment un Data Warehouse peut être utilisé pour améliorer la gestion des ventes, optimiser les opérations logistiques, et in fine, renforcer la position concurrentielle de l'entreprise sur le marché du commerce électronique.

2 Généralités sur le Data Warehouse et les Systèmes d'Information Décisionnels

2.1 Introduction aux Data Warehouses

Un Data Warehouse (entrepôt de données) est un système utilisé pour la collecte, le stockage et la gestion des données provenant de diverses sources. Il est conçu pour faciliter l'analyse et le reporting des données afin de prendre des décisions éclairées. Contrairement aux bases de données transactionnelles, qui sont optimisées pour les opérations courantes telles que l'insertion, la mise à jour et la suppression de données, un data warehouse est optimisé pour les requêtes et les analyses complexes.

Les caractéristiques clés d'un data warehouse incluent :

- Intégration : Agrégation de données provenant de différentes sources hétérogènes.
- Historisation : Conservation des données historiques pour l'analyse des tendances.
- Non-Volatilité : Une fois les données chargées, elles ne sont généralement pas modifiées.
- Orientation Sujet : Les données sont organisées par sujet (par exemple, ventes, clients, produits) plutôt que par application.

2.2 Architecture d'un Data Warehouse

Un data warehouse typique a plusieurs couches, chacune avec un rôle spécifique dans le processus d'intégration et de gestion des données :

- Source de Données : Comprend les systèmes transactionnels, les bases de données relationnelles, les fichiers plats, les services web, etc.
- Extraction, Transformation, Chargement (ETL) : Les processus ETL sont responsables de l'extraction des données des sources de données, de leur transformation en un format approprié, et de leur chargement dans le data warehouse.
- Entrepôt de Données Central : Le dépôt principal où les données intégrées et historisées sont stockées.
- Data Marts : Des sous-ensembles spécialisés du data warehouse, souvent créés pour répondre aux besoins spécifiques des départements ou des groupes d'utilisateurs.
- Outils d'Analyse et de Reporting : Des outils comme OLAP (Online Analytical Processing), des tableaux de bord de BI (Business Intelligence), et des outils de visualisation de données permettent aux utilisateurs finaux d'interroger et d'analyser les données.

2.3 Systèmes d'Information Décisionnels (SID)

Les Systèmes d'Information Décisionnels (SID) sont des systèmes informatiques qui supportent les processus de prise de décision dans une organisation. Ils utilisent les données stockées dans les data warehouses pour fournir des informations analytiques et stratégiques. Les SID peuvent inclure :

- Systèmes de Reporting : Produisent des rapports réguliers sur les performances opérationnelles.
- Systèmes OLAP : Permettent aux utilisateurs d'explorer et d'analyser les données multidimensionnelles.
- Systèmes de Gestion de la Performance : Surveillent et gèrent les performances organisationnelles en fonction des indicateurs clés de performance (KPI).
- Systèmes de Support à la Décision (DSS) : Aident les décideurs à prendre des décisions basées sur l'analyse des données.

2.4 Importance des Data Warehouses dans les Systèmes d'Information Décisionnels

Les data warehouses jouent un rôle crucial dans les SID pour plusieurs raisons :

- Consolidation des Données : Ils permettent de consolider les données provenant de multiples sources, offrant ainsi une vue d'ensemble cohérente et complète.
- Qualité des Données : Les processus ETL assurent que les données sont nettoyées, transformées et validées, améliorant ainsi leur qualité.
- Analyse Historique : La conservation des données historiques permet d'analyser les tendances et de prédire les futures performances.
- Prise de Décision Améliorée : Les données intégrées et analysées permettent aux décideurs d'obtenir des insights précis et exploitables, améliorant ainsi les processus décisionnels.

2.5 Exemple d'Utilisation : Commerce Électronique

Dans le contexte du commerce électronique, un data warehouse peut être utilisé pour :

- Analyse des Ventes : Suivre les performances des produits, analyser les tendances de vente, et identifier les produits les plus rentables.
- Segmentation des Clients : Catégoriser les clients en segments pour des campagnes marketing ciblées.
- Gestion des Promotions : Évaluer l'efficacité des campagnes promotionnelles et ajuster les stratégies en conséquence.
- Optimisation de la Logistique : Analyser les données de livraison pour optimiser les chaînes d'approvisionnement et réduire les coûts.

2.6 Environnement de travail

Pour mener à bien ce projet de création d'un entrepôt de données pour un commerce électronique, nous avons utilisé divers outils et technologies. Cette section décrit l'environnement de travail, les outils, et les langages de programmation utilisés tout au long du projet.

Et pour ce la on a une liste de outils utilisés:

SQL Server

- **Description :** SQL Server est un système de gestion de bases de données relationnelles développé par Microsoft. Il permet de stocker, manipuler, et gérer des données relationnelles de manière efficace.
- **Utilisation :** SQL Server a été utilisé pour héberger notre base de données, créer des tables de faits et de dimensions, et exécuter des requêtes SQL pour interroger et analyser les données.

SQL Server Management Studio (SSMS)

- **Description :** SSMS est un environnement intégré pour gérer les bases de données SQL Server. Il offre une interface graphique pour interagir avec les bases de données, écrire des scripts SQL, et exécuter des requêtes.
- **Utilisation :** SSMS a été utilisé pour administrer SQL Server, écrire et exécuter des scripts SQL, créer des vues, et analyser les résultats des requêtes.

Visual Studio Code (VS Code)

- **Description :** VS Code est un éditeur de code source développé par Microsoft. Il est léger et extensible, prenant en charge de nombreux langages de programmation.
- **Utilisation :** VS Code a été utilisé pour écrire des scripts SQL et documenter le code, grâce à ses nombreuses extensions facilitant la syntaxe SQL et la gestion de projet.

2.7 Langages utilisés

SQL (Structured Query Language)

- Description : SQL est un langage standard pour la gestion et la manipulation de bases de données relationnelles.
- Utilisation : SQL a été utilisé pour créer les tables de faits et de dimensions, insérer des données, créer des vues, et exécuter des requêtes analytiques.

Implémentation

- Création des tables de faits et de dimensions dans SQL Server.
- Insertion des données dans les tables.
- Création des vues pour simplifier les requêtes analytiques.

Interrogation et analyse

- Écriture et exécution des requêtes analytiques pour répondre aux besoins de l'entreprise.
- Analyse des résultats des requêtes pour obtenir des insights sur les ventes, les clients, et les produits.

2.8 Conclusion

Les data warehouses et les systèmes d'information décisionnels sont des outils essentiels pour les entreprises modernes, leur permettant de transformer de grandes quantités de données en informations précieuses pour la prise de décision stratégique. Dans un monde où les données sont de plus en plus considérées comme un atout crucial, la capacité de gérer, analyser et exploiter efficacement ces données peut offrir un avantage concurrentiel significatif.

3 Analyse du cas d'étude

Dans cette section nous allons analyser notre cas d'étude.

3.1 Analyse du Cas Considéré

3.1.1 Objectifs de l'entreprise

L'objectif principal d'Amazon est de fournir une plateforme de commerce électronique robuste et fiable, offrant une vaste gamme de produits à des prix compétitifs. Amazon vise à augmenter ses ventes, améliorer l'expérience client, et optimiser la logistique et la gestion de l'inventaire.

3.1.2 Position sur le marché

Amazon est l'un des leaders mondiaux dans le secteur du commerce électronique, avec une présence significative sur les marchés internationaux. Il est réputé pour son large éventail de produits, son service de livraison rapide, et son service clientèle de qualité.

3.1.3 Services ou produits proposés

Amazon propose des millions de produits dans diverses catégories telles que l'électronique, les livres, les vêtements, la nourriture, et bien plus encore. Il offre également des services comme Amazon Prime, Amazon Web Services (AWS), et des plateformes de streaming comme Amazon Prime Vidéo.

3.1.4 Formes de revenu

Amazon génère des revenus principalement à travers la vente de produits, les abonnements à Amazon Prime, les frais de services logistiques, la publicité en ligne, et les services cloud via AWS.

3.1.5 Informations utiles pour la prise de décision

- Tendances de vente par catégorie de produits
- Comportement d'achat des clients
- Efficacité des promotions et des campagnes de marketing
- Performance des fournisseurs
- Gestion des stocks et optimisation des livraisons

3.2 Actions / Opérations à Tracer

1. Ventes : Tracer les ventes pour analyser les tendances et les performances des produits.
2. Livraisons : Suivre les livraisons pour optimiser les coûts et les délais de livraison.
3. Promotions : Évaluer l'efficacité des promotions pour améliorer les campagnes futures.
4. Inventaire : Gérer les niveaux de stock pour éviter les ruptures ou les surstocks.
5. Feedback des Clients : Analyser les retours des clients pour améliorer les produits et services.

3.2.1 Traitements Possibles (Requêtes Analytiques)

1. Ventes :
 - Calculer les ventes totales par produit et par période.
 - Analyser les tendances de vente par catégorie de produit.
 - Évaluer la performance des ventes par région.
2. Livraisons :
 - Calculer le coût total des livraisons par méthode de livraison.
 - Mesurer les délais de livraison et identifier les retards.
 - Analyser la satisfaction client par rapport aux délais de livraison.
3. Promotions :
 - Évaluer l'impact des promotions sur les ventes.
 - Analyser la rentabilité des promotions par catégorie de produit.
 - Comparer les ventes pendant et en dehors des périodes promotionnelles.

3.2.2 Ordre d'importance des actions

1. **Ventes** (augmentation des ventes)
2. **Promotions** (efficacité des promotions)
3. **Livraisons** (optimisation des coûts et délais)
4. **Inventaire** (gestion des stocks)
5. **Feedback des Clients** (amélioration des produits et services)

4 Conception des Data-Marts

4.1 Dimensions Nécessaires aux Modèles

La table des faits à la quelle nous allons nous intéresser sera "Ventes".

4.1.1 Dimensions pour les ventes

1. Product (ProductID, ProductName, Category, Price, Brand, SupplierID, Stock, Color, Size, Weight, Dimensions)
2. Customer (CustomerID, CustomerName, Email, Address, PhoneNumber, Gender, Age, MembershipStatus, RegistrationDate, LastPurchaseDate)
3. Date (DateKey, FullDate, Day, Month, Year, Quarter, WeekOfYear, DayOfWeek, IsHoliday, IsWeekend)
4. PaymentMethod (PaymentMethodID, PaymentMethodName, Description, Fees, Currency, Country)
5. Shipping (ShippingID, ShippingMethod, ShippingCost, DeliveryTime, Carrier, ServiceLevel, TrackingAvailable)
6. Promotion (PromotionID, PromotionName, DiscountPercentage, StartDate, EndDate, CampaignBudget, TargetAudience, MediaChannel, Goals, Metrics)

4.2 Diagramme de conception

La conception de notre entrepôt de données pour le projet de commerce électronique s'articule autour d'un modèle en étoile. Ce modèle est composé de plusieurs tables de dimension et d'une table de faits centrale. Voici un diagramme de conception représentant les principales tables de dimensions et la table de faits :

4.2.1 Ventes (modélisation détaillée)

Table des Faits : FactSales

- Mesures : QuantitySold (additive),
- TotalAmount (additive) Clés étrangères : ProductID, CustomerID, DateKey, PaymentMethodID, ShippingID

Dimensions

1. DimProduct (ProductID, ProductName, Category, Price, Brand, SupplierID)
2. DimCustomer (CustomerID, CustomerName, Email, Address, PhoneNumber)

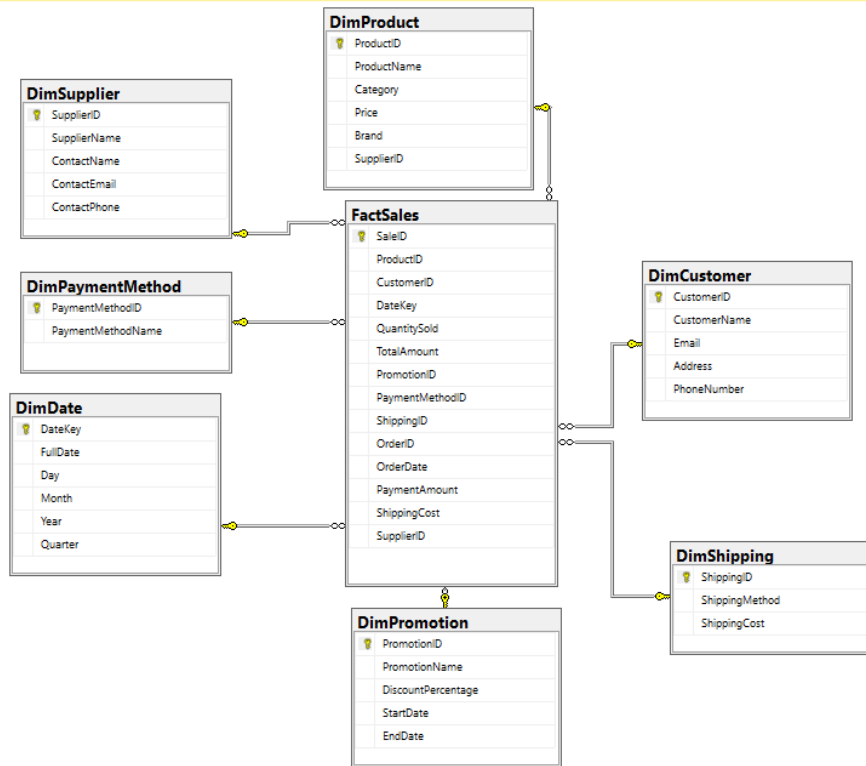


FIG. 1 – *Diagramme de Conception*

3. DimDate (DateKey, FullDate, Day, Month, Year, Quarter)
4. DimPaymentMethod (PaymentMethodID, PaymentMethodName)
5. DimShipping (ShippingID, ShippingMethod, ShippingCost) DimPromotion (PromotionID, PromotionName, DiscountPercentage)

4.3 Réponses au traitements avec le modèle

Le modèle conçu permet de répondre efficacement aux requêtes analytiques prévues grâce à la structure en étoile des data-marts. Les dimensions et les mesures définies couvrent les besoins d'analyse pour les ventes et les promotions.

▷ **Exemple :**

- DimProduct

ProdectId	ProductName	Category	Price	Brand	SupplierID
1	Laptop	Electronics	999.99	Dell	101

- FactSales

SaleId	ProductID	CustomerID	DateKey	Quantity	SoldTotalAmount
1	1	1	20240101	2	1999.98

PromotionID	PaymentMethodID	ShippingID
201	1	1

4.4 Estimation taille des tables

Pour garantir la performance et l'évolutivité de notre entrepôt de données, il est essentiel de procéder à une estimation de la taille des tables, en tenant compte du volume de données attendu pour chaque entité.

- FactSales : Estimation de 100 ventes par jour -> 36,500 lignes par an.
- DimProduct : 100 produits
- DimCustomer : 10,000 clients
- DimDate : 365 jours
- DimPromotion : 20 promotions
- DimPaymentMethod : 5 méthodes de paiement
- DimShipping : 5 méthodes de livraison

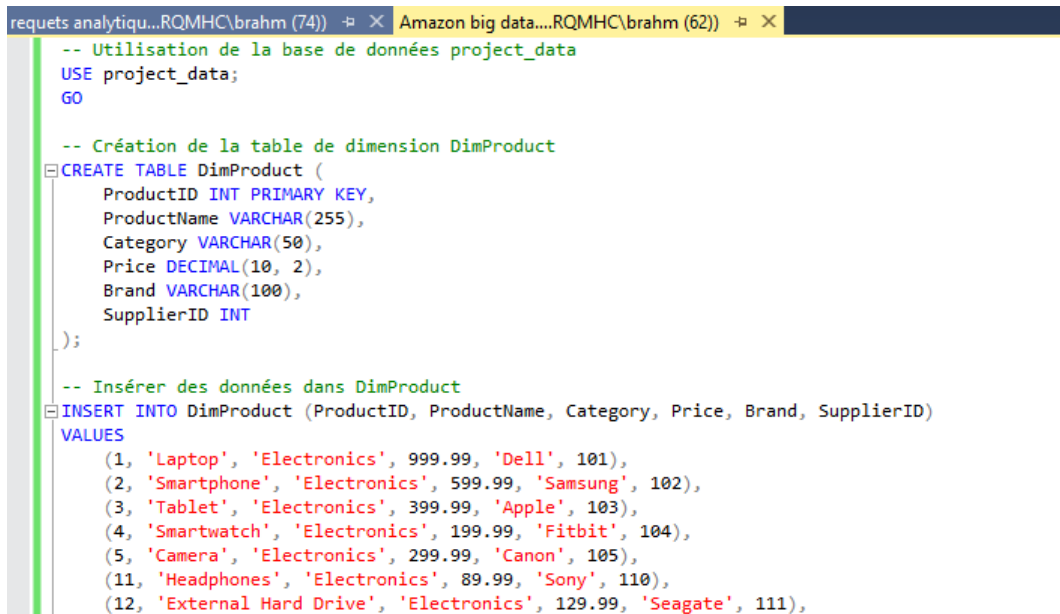
La taille estimée est raisonnable et gérable pour un entrepôt de données.

5 Implementaton

L'implémentation de l'entrepôt de données constitue une étape cruciale de notre projet. Cette phase consiste à créer les tables de faits et de dimensions selon le modèle en étoile défini, et à y charger les données nécessaires. En utilisant SQL, nous allons mettre en place la structure de l'entrepôt de données, assurer l'intégrité des données et optimiser les performances pour les requêtes analytiques.

5.1 Implémentation du modèle

Voici le script SQL complet pour créer les tables de dimensions et la table de faits, ainsi que pour insérer les données dans ces tables.



```
requets analytiqu...RQMHC\brahm (74))  X Amazon big data...RQMHC\brahm (62))  X
-- Utilisation de la base de données project_data
USE project_data;
GO

-- Création de la table de dimension DimProduct
CREATE TABLE DimProduct (
    ProductID INT PRIMARY KEY,
    ProductName VARCHAR(255),
    Category VARCHAR(50),
    Price DECIMAL(10, 2),
    Brand VARCHAR(100),
    SupplierID INT
);

-- Insérer des données dans DimProduct
INSERT INTO DimProduct (ProductID, ProductName, Category, Price, Brand, SupplierID)
VALUES
    (1, 'Laptop', 'Electronics', 999.99, 'Dell', 101),
    (2, 'Smartphone', 'Electronics', 599.99, 'Samsung', 102),
    (3, 'Tablet', 'Electronics', 399.99, 'Apple', 103),
    (4, 'Smartwatch', 'Electronics', 199.99, 'Fitbit', 104),
    (5, 'Camera', 'Electronics', 299.99, 'Canon', 105),
    (11, 'Headphones', 'Electronics', 89.99, 'Sony', 110),
    (12, 'External Hard Drive', 'Electronics', 129.99, 'Seagate', 111),
```

FIG. 2 – Création de la table de dimension DimProduct

```

-- Création de la table de dimension DimSupplier
CREATE TABLE DimSupplier (
    SupplierID INT PRIMARY KEY,
    SupplierName VARCHAR(255),
    ContactName VARCHAR(255),
    ContactEmail VARCHAR(100),
    ContactPhone VARCHAR(20)
);

-- Insérer des données dans DimSupplier
INSERT INTO DimSupplier (SupplierID, SupplierName, ContactName, ContactEmail, ContactPhone)
VALUES
    (101, 'Dell', 'Michael Dell', 'michael.dell@dell.com', '555-1111'),
    (102, 'Samsung', 'Lee Kun-hee', 'lee.kunhee@samsung.com', '555-2222'),
    (103, 'Apple', 'Tim Cook', 'tim.cook@apple.com', '555-3333'),
    (104, 'Fitbit', 'James Park', 'james.park@fitbit.com', '555-4444'),
    (105, 'Canon', 'Fujio Mitarai', 'fujio.mitarai@canon.com', '555-5555'),
    (110, 'Sony', 'Kenichiro Yoshida', 'kenichiro.yoshida@sony.com', '555-1010'),
    (111, 'Seagate', 'Dave Mosley', 'dave.mosley@seagate.com', '555-1111'),
    (112, 'GoPro', 'Nicholas Woodman', 'nicholas.woodman@gopro.com', '555-1212'),
    (113, 'Amazon', 'Andy Jassy', 'andy.jassy@amazon.com', '555-1313');

```

FIG. 3 – Création de la table de dimension DimSupplier

```

-- Création de la table de dimension DimCustomer
CREATE TABLE DimCustomer (
    CustomerID INT PRIMARY KEY,
    CustomerName VARCHAR(255),
    Email VARCHAR(100),
    Address VARCHAR(255),
    PhoneNumber VARCHAR(20)
);

-- Insérer des données dans DimCustomer
INSERT INTO DimCustomer (CustomerID, CustomerName, Email, Address, PhoneNumber)
VALUES
    (1, 'John Doe', 'john.doe@example.com', '123 Elm Street', '555-1234'),
    (2, 'Jane Smith', 'jane.smith@example.com', '456 Oak Street', '555-5678'),
    (3, 'Jim Brown', 'jim.brown@example.com', '789 Pine Street', '555-8765'),
    (4, 'Anna White', 'anna.white@example.com', '135 Maple Street', '555-1357'),
    (5, 'Mike Green', 'mike.green@example.com', '246 Birch Street', '555-2468'),
    (6, 'Linda Black', 'linda.black@example.com', '357 Cedar Street', '555-3579'),
    (7, 'David Blue', 'david.blue@example.com', '111 Spruce Street', '555-7890'),
    (8, 'Eve Grey', 'eve.grey@example.com', '222 Cypress Street', '555-2345'),
    (9, 'Frank Yellow', 'frank.yellow@example.com', '333 Redwood Street', '555-3456');

```

FIG. 4 – Création de la table de dimension DimCustomer

```
-- Création de la table de dimension DimDate
CREATE TABLE DimDate (
    DateKey INT PRIMARY KEY,
    FullDate DATE,
    Day INT,
    Month INT,
    Year INT,
    Quarter INT
);

-- Insérer des données dans DimDate
INSERT INTO DimDate (DateKey, FullDate, Day, Month, Year, Quarter)
VALUES
    (20240101, '2024-01-01', 1, 1, 2024, 1),
    (20240102, '2024-01-02', 2, 1, 2024, 1),
    (20240103, '2024-01-03', 3, 1, 2024, 1),
    (20240201, '2024-02-01', 1, 2, 2024, 1),
    (20240301, '2024-03-01', 1, 3, 2024, 1),
    (20240402, '2024-04-02', 2, 4, 2024, 2),
    (20240501, '2024-05-01', 1, 5, 2024, 2),
    (20240502, '2024-05-02', 2, 5, 2024, 2),
    (20240601, '2024-06-01', 1, 6, 2024, 2),
    (20240602, '2024-06-02', 2, 6, 2024, 2);
```

FIG. 5 – Création de la table de dimension *DimDate*

```
-- Création de la table de dimension DimPaymentMethod
CREATE TABLE DimPaymentMethod (
    PaymentMethodID INT PRIMARY KEY,
    PaymentMethodName VARCHAR(50)
);

-- Insérer des données dans DimPaymentMethod
INSERT INTO DimPaymentMethod (PaymentMethodID, PaymentMethodName)
VALUES
    (1, 'Credit Card'),
    (2, 'PayPal'),
    (3, 'Bank Transfer'),
    (4, 'Cash on Delivery');
```

FIG. 6 – Création de la table de dimension *DimPaymentMethod*

```

-- Création de la table de dimension DimPromotion
CREATE TABLE DimPromotion (
    PromotionID INT PRIMARY KEY,
    PromotionName VARCHAR(255),
    DiscountPercentage DECIMAL(5, 2),
    StartDate DATE,
    EndDate DATE
);

-- Insérer des données dans DimPromotion
INSERT INTO DimPromotion (PromotionID, PromotionName, DiscountPercentage, StartDate, EndDate)
VALUES
    (201, 'New Year Sale', 10.00, '2024-01-01', '2024-01-31'),
    (202, 'Valentine Day Sale', 15.00, '2024-02-01', '2024-02-14'),
    (203, 'Spring Sale', 20.00, '2024-03-01', '2024-03-31'),
    (204, 'Summer Sale', 25.00, '2024-06-01', '2024-06-30'),
    (205, 'Back to School Sale', 30.00, '2024-08-01', '2024-08-31');

```

FIG. 7 – Création de la table de dimension *DimPromotion*

```

-- Création de la table de dimension DimShipping
CREATE TABLE DimShipping (
    ShippingID INT PRIMARY KEY,
    ShippingMethod VARCHAR(50),
    ShippingCost DECIMAL(10, 2)
);

-- Insérer des données dans DimShipping
INSERT INTO DimShipping (ShippingID, ShippingMethod, ShippingCost)
VALUES
    (1, 'Standard Shipping', 5.99),
    (2, 'Express Shipping', 9.99),
    (3, 'Overnight Shipping', 19.99),
    (4, 'International Shipping', 29.99);

```

FIG. 8 – Création de la table de dimension *DimShipping*

```

-- Création de la table FactSales
CREATE TABLE FactSales (
    SaleID INT PRIMARY KEY IDENTITY(1,1),
    ProductID INT,
    CustomerID INT,
    DateKey INT,
    QuantitySold INT,
    TotalAmount DECIMAL(10, 2),
    PromotionID INT,
    PaymentMethodID INT,
    ShippingID INT,
    OrderID INT,
    OrderDate DATE,
    PaymentAmount DECIMAL(10, 2),
    ShippingCost DECIMAL(10, 2),
    FOREIGN KEY (ProductID) REFERENCES DimProduct(ProductID),
    FOREIGN KEY (CustomerID) REFERENCES DimCustomer(CustomerID),
    FOREIGN KEY (DateKey) REFERENCES DimDate(DateKey),
    FOREIGN KEY (PromotionID) REFERENCES DimPromotion(PromotionID),
    FOREIGN KEY (PaymentMethodID) REFERENCES DimPaymentMethod(PaymentMethodID),
    FOREIGN KEY (ShippingID) REFERENCES DimShipping(ShippingID)
);

```

FIG. 9 – Création de la table *FactSales*

```

-- Insérer des données dans FactSales
INSERT INTO FactSales (ProductID, CustomerID, DateKey, QuantitySold, TotalAmount, PromotionID, PaymentMethodID, ShippingID, OrderID, OrderDate, PaymentAmount, Shipping
VALUES
(1, 1, 20240101, 2, 1999.98, 201, 1, 1, 1001, '2024-01-01', 1999.98, 5.99),
(2, 2, 20240102, 1, 599.99, 202, 2, 2, 1002, '2024-01-02', 599.99, 9.99),
(3, 3, 20240103, 3, 1199.97, 203, 3, 1, 1003, '2024-01-03', 1199.97, 5.99),
(4, 4, 20240201, 1, 199.99, NULL, 4, 2, 1004, '2024-02-01', 199.99, 9.99),
(5, 5, 20240301, 2, 599.98, 203, 1, 1, 1005, '2024-03-01', 599.98, 5.99),
(11, 6, 20240402, 1, 89.99, NULL, 2, 1, 1006, '2024-04-02', 89.99, 5.99),
(12, 7, 20240501, 2, 259.98, 204, 3, 2, 1007, '2024-05-01', 259.98, 9.99),
(13, 8, 20240502, 1, 399.99, 205, 1, 3, 1008, '2024-05-02', 399.99, 19.99),
(14, 9, 20240601, 3, 899.97, NULL, 2, 4, 1009, '2024-06-01', 899.97, 29.99),
(15, 1, 20240602, 1, 149.99, NULL, 1, 1, 1010, '2024-06-02', 149.99, 5.99);

```

FIG. 10 – Insertion dans la table *FactSales*

5.2 Requête analytique

Pour exploiter pleinement les données stockées dans notre entrepôt de données, nous avons défini un ensemble de requêtes analytiques. Ces requêtes visent à extraire des informations précieuses sur divers aspects des opérations commerciales, permettant ainsi une prise de décision éclairée. Voici quelques-unes des requêtes analytiques les plus significatives, accompagnées de leurs résultats attendus :

Total des ventes par trimestre.

```
--1. Ventes totales par trimestre
SELECT
    Year,
    Quarter,
    SUM(TotalAmount) AS TotalSalesAmount
FROM FactSales fs
JOIN DimDate dd ON fs.DateKey = dd.DateKey
GROUP BY Year, Quarter
ORDER BY Year, Quarter;
```

FIG. 11 – Requête analytique (total des ventes par trimestre)

Résultats:

	Year	Quarter	TotalSalesAmount
1	2024	1	4599.91
2	2024	2	1799.92

FIG. 12 – Résultats Requête (Total des ventes par trimestre)

Les produits les plus vendues.

```
--2. Produits les plus vendus (par quantité)
SELECT
    p.ProductName,
    SUM(fs.QuantitySold) AS TotalQuantitySold
FROM FactSales fs
JOIN DimProduct p ON fs.ProductID = p.ProductID
GROUP BY p.ProductName
ORDER BY TotalQuantitySold DESC;
```

FIG. 13 – Requête analytique(Les produits les plus vendus)

Résultats:

	ProductName	TotalQuantitySold
1	Action Camera	3
2	Tablet	3
3	Camera	2
4	External Hard Drive	2
5	Laptop	2
6	Smart Speaker	1
7	Smartphone	1
8	Smartwatch	1

FIG. 14 – Résultats requête (Les produits les plus vendus)

Montant totale des ventes par client

```
--3. Montant total des ventes par client
SELECT
    c.CustomerName,
    SUM(fs.TotalAmount) AS TotalSalesAmount
FROM FactSales fs
JOIN DimCustomer c ON fs.CustomerID = c.CustomerID
GROUP BY c.CustomerName
ORDER BY TotalSalesAmount DESC;
```

FIG. 15 – Requête analytique (Montant totale des ventes par client)

Résultats:

	CustomerName	TotalSalesAmount
1	John Doe	2149.97
2	Jim Brown	1199.97
3	Frank Yellow	899.97
4	Jane Smith	599.99
5	Mike Green	599.98
6	Eve Grey	399.99
7	David Blue	259.98
8	Anna White	199.99

FIG. 16 – Résultats requête (Montant totale des ventes par client)

Ventes par méthode de paiement

Résultats:


```
--4. Ventes par méthode de paiement
SELECT
    pm.PaymentMethodName,
    COUNT(*) AS TotalSalesCount
FROM FactSales fs
JOIN DimPaymentMethod pm ON fs.PaymentMethodID = pm.PaymentMethodID
GROUP BY pm.PaymentMethodName
ORDER BY TotalSalesCount DESC;
```

FIG. 17 – Requête analytique (Ventes par méthode de paiement)

	PaymentMethodName	TotalSalesCount
1	Credit Card	4
2	PayPal	3
3	Bank Transfer	2
4	Cash on Delivery	1

FIG. 18 – Résultats requête (Ventes par méthode de paiement)

Détails des ventes pour une promotion spécifique

```
--5. Détails des ventes pour une promotion spécifique
SELECT
    fs.OrderID,
    p.ProductName,
    fs.QuantitySold,
    fs.TotalAmount,
    dp.PromotionName
FROM FactSales fs
JOIN DimProduct p ON fs.ProductID = p.ProductID
JOIN DimPromotion dp ON fs.PromotionID = dp.PromotionID
WHERE dp.PromotionName = 'New Year Sale';
```

FIG. 19 – Requête analytique (Détails des ventes pour une promotion spécifique)

Résultats:

	OrderID	ProductName	QuantitySold	TotalAmount	PromotionName
1	1001	Laptop	2	1999.98	New Year Sale

FIG. 20 – Résultats requête (Détails des ventes pour une promotion spécifique)

vente par catégories de produits

Résultats:

Montant total des ventes par mois

Résultats:

```
--6. Ventes par catégorie de produit
SELECT
    p.Category,
    SUM(fs.QuantitySold) AS TotalQuantitySold,
    SUM(fs.TotalAmount) AS TotalSalesAmount
FROM FactSales fs
JOIN DimProduct p ON fs.ProductID = p.ProductID
GROUP BY p.Category
ORDER BY TotalSalesAmount DESC;
```

FIG. 21 – Requête analytique (vente par catégories de produits)

Results Messages			
	Category	TotalQuantitySold	TotalSalesAmount
1	Electronics	17	6399.83

FIG. 22 – Résultats requete (vente par catégories de produits)

```
--7. Montant total des ventes par mois
SELECT
    Year,
    Month,
    SUM(TotalAmount) AS TotalSalesAmount
FROM FactSales fs
JOIN DimDate dd ON fs.DateKey = dd.DateKey
GROUP BY Year, Month
ORDER BY Year, Month;
```

FIG. 23 – Requête analytique(Montant total des ventes par mois)

	Year	Month	TotalSalesAmount
1	2024	1	3799.94
2	2024	2	199.99
3	2024	3	599.98
4	2024	4	89.99
5	2024	5	659.97
6	2024	6	1049.96

FIG. 24 – Résultats requete(Montant total des ventes par mois)

Détails de ventes effectué par un client spécifique

Résultats:

Ventes avec un coût d'expédition élevé

Résultats:

```
--8. Détails des ventes effectuées par un client spécifique
SELECT
    fs.OrderID,
    p.ProductName,
    fs.QuantitySold,
    fs.TotalAmount,
    fs.OrderDate
FROM FactSales fs
JOIN DimProduct p ON fs.ProductID = p.ProductID
WHERE fs.CustomerID = 1;
```

FIG. 25 – Requête analytique (Détails de ventes effectué par un client spécifique)

	OrderID	ProductName	QuantitySold	TotalAmount	OrderDate
1	1001	Laptop	2	1999.98	2024-01-01
2	1010	Smart Speaker	1	149.99	2024-06-02

FIG. 26 – Résultats requete (Détails de ventes effectué par un client spécifique)

```
--9. Ventes avec coûts d'expédition élevés
SELECT
    fs.OrderID,
    p.ProductName,
    fs.QuantitySold,
    fs.TotalAmount,
    ds.ShippingMethod,
    ds.ShippingCost
FROM FactSales fs
JOIN DimProduct p ON fs.ProductID = p.ProductID
JOIN DimShipping ds ON fs.ShippingID = ds.ShippingID
WHERE ds.ShippingCost > 10.00;
```

FIG. 27 – Requête analytique (Ventes avec un coût d'expédition élevé)

	OrderID	ProductName	QuantitySold	TotalAmount	ShippingMethod	ShippingCost
1	1008	Gaming Console	1	399.99	Overnight Shipping	19.99
2	1009	Action Camera	3	899.97	International Shipping	29.99

FIG. 28 – Résultats requete (Ventes avec un coût d'expédition élevé)

Nombre de clients par catégorie de produit

Résultats:

5.3 Création et Utilisation des Vues

5.3.1 définition des vues

Une vue en SQL est une table virtuelle basée sur le résultat d'une requête SQL. Contrairement à une table standard, une vue ne contient pas de données elle-même, mais affiche les données stockées dans d'autres tables. Les vues peuvent être utilisées pour simplifier des requêtes complexes, restreindre l'accès aux données sensibles, ou

```
--10. Nombre de ventes par jour pour une période spécifique
SELECT
    dd.FullDate,
    COUNT(*) AS TotalSalesCount
FROM FactSales fs
JOIN DimDate dd ON fs.DateKey = dd.DateKey
WHERE dd.FullDate BETWEEN '2024-01-01' AND '2024-01-31'
GROUP BY dd.FullDate
ORDER BY dd.FullDate;
```

FIG. 29 – Requête analytique (Nombre de clients par catégorie de produit)

	FullDate	TotalSalesCount
1	2024-01-01	1
2	2024-01-02	1
3	2024-01-03	1

FIG. 30 – Résultats requête (Nombre de clients par catégorie de produit)

présenter les données de manière spécifique.

5.3.2 Caractéristique des vues

- **Virtualité** : Les vues n'existent pas physiquement dans la base de données, sauf si elles sont matérialisées.
- **Simplicité** : Les vues peuvent simplifier des requêtes complexes en les encapsulant dans une vue.
- **Sécurité** : Les vues peuvent restreindre l'accès à certaines colonnes ou lignes de données.
- **Modifiabilité** : Les vues peuvent parfois être utilisées pour mettre à jour les données sous-jacentes.

5.3.3 Vues généré

Voici trois exemples de vues. Ces vues sont basées sur les données de la table FactSales et d'autres tables de dimension.

Vue1: Ventes par Client

Cette vue regroupe les informations sur les ventes par client, permettant d'analyser les achats de chaque client, la somme totale dépensée et le nombre de produits achetés.

Résultats:

Vue2: Ventes par Produit

Cette vue fournit un résumé des ventes par produit, incluant le nombre total de ventes, la quantité vendue et le revenu total généré par chaque produit.

Résultats:

Vue3: Ventes par Promotion

Résultats:

Ces vues peuvent être utilisées pour des analyses spécifiques, aidant les décideurs à obtenir des insights sur les performances des clients, des produits et des périodes de temps spécifiques. Elles simplifient également les requêtes en encapsulant des agrégations complexes dans des vues facilement interrogeables.

```

CREATE VIEW SalesByCustomer AS
SELECT
    c.CustomerID,
    c.CustomerName,
    SUM(f.TotalAmount) AS TotalSpent,
    COUNT(f.SaleID) AS TotalPurchases,
    SUM(f.QuantitySold) AS TotalQuantity
FROM
    FactSales f
JOIN
    DimCustomer c ON f.CustomerID = c.CustomerID
GROUP BY
    c.CustomerID, c.CustomerName;

```

FIG. 31 – Vue de Ventes par Client

select * from SalesByCustomer;

100 %

Results Messages

	CustomerID	CustomerName	TotalSpent	TotalPurchases	TotalQuantity
1	1	John Doe	2149.97	2	3
2	2	Jane Smith	599.99	1	1
3	3	Jim Brown	1199.97	1	3
4	4	Anna White	199.99	1	1
5	5	Mike Green	599.98	1	2
6	6	Linda Black	89.99	1	1
7	7	David Blue	259.98	1	2
8	8	Eve Grey	399.99	1	1
9	9	Frank Yellow	899.97	1	3

FIG. 32 – Résultats de Ventes par Client

```

CREATE VIEW SalesByProduct AS
SELECT
    p.ProductID,
    p.ProductName,
    p.Category,
    p.Brand,
    SUM(f.QuantitySold) AS TotalQuantitySold,
    SUM(f.TotalAmount) AS TotalRevenue
FROM
    FactSales f
JOIN
    DimProduct p ON f.ProductID = p.ProductID
GROUP BY
    p.ProductID, p.ProductName, p.Category, p.Brand;

```

FIG. 33 – Vue de Ventes par Produit

6 Perspectives d'avenir

7 Conclusion Générale

En conclusion, ce projet a permis de démontrer l'importance et la puissance des systèmes décisionnels dans le secteur du commerce électronique. À travers la conception, l'implémentation et l'interrogation d'un entrepôt de données, nous avons illustré comment une entreprise peut transformer des volumes massifs de données en informa-

```
select * from SalesByProduct;
```

	ProductID	ProductName	Category	Brand	TotalQuantitySold	TotalRevenue
1	1	Laptop	Electronics	Dell	2	1999.98
2	2	Smartphone	Electronics	Samsung	1	599.99
3	3	Tablet	Electronics	Apple	3	1199.97
4	4	Smartwatch	Electronics	Fitbit	1	199.99
5	5	Camera	Electronics	Canon	2	599.98
6	11	Headphones	Electronics	Sony	1	89.99
7	12	External Hard Drive	Electronics	Seagate	2	259.98
8	13	Gaming Console	Electronics	Sony	1	399.99
9	14	Action Camera	Electronics	GoPro	3	899.97
10	15	Smart Speaker	Electronics	Amazon	1	149.99

FIG. 34 – Résultats de Ventes par Produit

```
CREATE VIEW SalesByPromotion AS
SELECT
    pr.PromotionID,
    pr.PromotionName,
    pr.DiscountPercentage,
    SUM(f.QuantitySold) AS TotalQuantitySold,
    SUM(f.TotalAmount) AS TotalRevenue
FROM
    FactSales f
LEFT JOIN
    DimPromotion pr ON f.PromotionID = pr.PromotionID
GROUP BY
    pr.PromotionID, pr.PromotionName, pr.DiscountPercentage;
```

FIG. 35 – Vue de Ventes par Promotion

```
select * from SalesByPromotion;
```

	PromotionID	PromotionName	DiscountPercentage	TotalQuantitySold	TotalRevenue
1	NULL	NULL	NULL	6	1339.94
2	201	New Year Sale	10.00	2	1999.98
3	202	Valentine Day Sale	15.00	1	599.99
4	203	Spring Sale	20.00	5	1799.95
5	204	Summer Sale	25.00	2	259.98
6	205	Back to School Sale	30.00	1	399.99

FIG. 36 – Résultats de Ventes par Promotion

tions exploitables pour améliorer ses performances commerciales. En nous inspirant du modèle de succès d'Amazon, nous avons montré comment un Data Warehouse bien structuré peut aider à optimiser les ventes, la logistique, la gestion des stocks et les campagnes marketing.

L'analyse approfondie des différentes dimensions et des requêtes analytiques proposées permet de répondre efficacement aux besoins décisionnels de l'entreprise. De plus, l'utilisation d'outils tels que les systèmes OLAP et les tableaux de bord de BI facilite l'exploration des données et la prise de décisions stratégiques.

Enfin, les perspectives et le potentiel d'amélioration pour le commerce électronique sont vastes. En continuant à affiner et à enrichir notre entrepôt de données, et en intégrant des technologies avancées telles que l'intelligence artificielle et l'apprentissage automatique, l'entreprise pourra non seulement maintenir mais également renforcer sa position concurrentielle sur le marché global. Les données sont indéniablement un atout précieux, et leur gestion efficace est la clé du succès dans l'environnement commercial dynamique d'aujourd'hui.