

ENTREPÔT DE DONNÉES POUR IVOIRE SHOP

Projet d'Intelligence d'Affaires

SOMMAIRE

1. Introduction
2. Analyse de la Problématique
 - Contexte et Défis Actuels
 - Structure des Systèmes Sources
 - Impacts du Cloisonnement des Données
3. Modélisation de l'Entrepôt de Données
 - Approche de Modélisation Dimensionnelle
 - Identification des Besoins Analytiques
 - Conception du Schéma en Étoile
 - Tables de Dimensions
 - Table de Faits
 - Gestion des Clés et Zone de Staging
4. Processus ETL
 - Architecture du Processus
 - Procédures ETL pour les Dimensions
 - Procédures ETL pour la Table de Faits
 - Orchestration et Automatisation

INTRODUCTION

Ce document présente la conception et la mise en œuvre d'une solution d'intelligence d'affaires pour IVOIRE SHOP, une entreprise de vente au détail opérant en Côte d'Ivoire. Face aux défis posés par la fragmentation des données entre plusieurs systèmes sources, nous avons développé un entrepôt de données intégré permettant d'analyser efficacement les performances commerciales selon différentes dimensions.

Cette solution vise à transformer les données opérationnelles en informations stratégiques exploitables, offrant ainsi à IVOIRE SHOP les moyens de prendre des décisions éclairées basées sur une vision complète et cohérente de son activité commerciale.

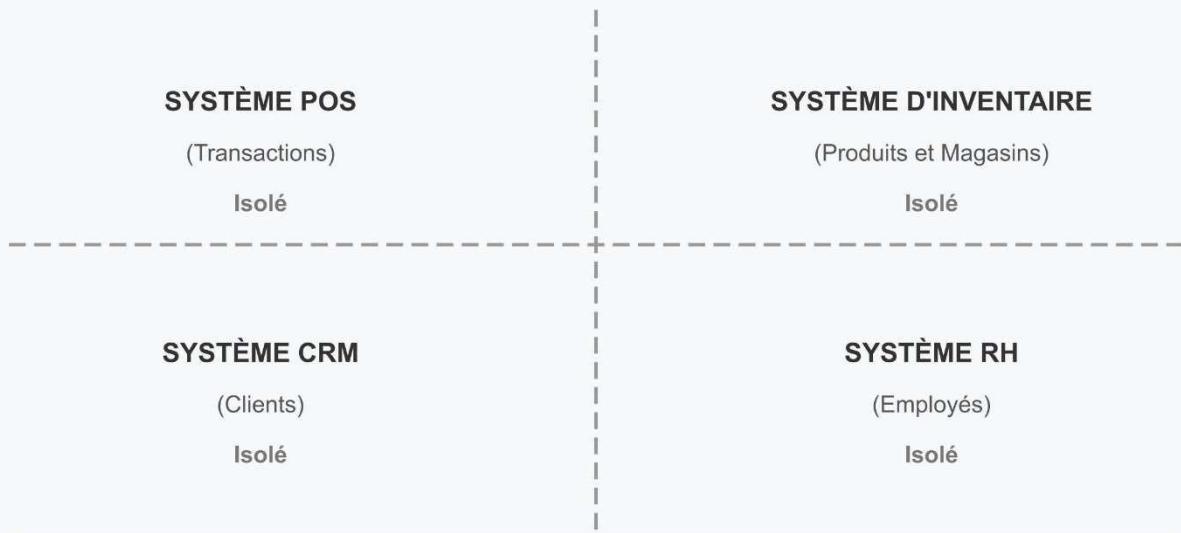
1. ANALYSE DE LA PROBLÉMATIQUE

Contexte et Défis Actuels

IVOIRE SHOP fait face à un défi majeur dans sa gestion de l'information commerciale. L'entreprise opère avec plusieurs systèmes sources cloisonnés qui ne communiquent pas efficacement entre eux.

Analyse de la Problématique - IVOIRE SHOP

Systèmes Sources Cloisonnés



Ces systèmes isolés empêchent IVOIRE SHOP d'avoir une vision intégrée de ses activités commerciales et limitent sa capacité à réaliser des analyses multidimensionnelles. L'absence d'une plateforme d'intelligence d'affaires unifiée freine considérablement le processus décisionnel et l'optimisation des performances.

Structure des Systèmes Sources

Les données de production sont actuellement réparties dans quatre schémas principaux :

1. Schéma POS (Point of Sale)

Table	Contenu
transactions	Enregistre les transactions de vente avec détails (date, produit, magasin, client, employé, quantité, prix, remises)

2. Schéma Inventory (Inventaire)

Table	Contenu
products	Informations sur les produits (code, nom, description, marque, catégorie, prix d'achat, prix de vente)
stores	Données des magasins (code, nom, adresse, ville, type, superficie)

3. Schéma CRM (Customer Relationship Management)

Table	Contenu
customers	Informations clients (code, nom, prénom, coordonnées, segment)

4. Schéma HR (Human Resources)

Table	Contenu
employees	Données des employés (code, nom, prénom, date d'embauche, fonction, salaire)

Impacts du Cloisonnement des Données

Cette fragmentation des données entre différents systèmes opérationnels génère plusieurs contraintes pour IVOIRE SHOP :

Sourcesystem

Object Explorer

Connect ▾

(local) (Microsoft Analysis Server 16.0.42.2

- Databases
- Assemblies
- Management

DESKTOP-A9RMIDD (SQL Server 16.0.1135

- Databases
- Security
- Server Objects
- Replication
- Always On High Availability
- Management
- Integration Services Catalogs
- SQL Server Agent
- XEvent Profiler

SQLQuery36.sql - P-A9RMIDD\HP (54)* MDXQuery16.mdx - not connected* MDXQuery15.mdx - not connected* MDXQuery14.mdx - not connected*

```
-- Vérifier l'insertion des données test dans les tables sources
USE Sourcesystem;

-- Vérifier les produits
SELECT COUNT(*) AS NombreProduits FROM inventory.products;

-- Vérifier les magasins
SELECT COUNT(*) AS NombreMagasins FROM inventory.stores;

-- Vérifier les clients
SELECT COUNT(*) AS NombreClients FROM crm.customers;

-- Vérifier les employés
SELECT COUNT(*) AS NombreEmployes FROM hr.employees;
```

110 %

Results Messages

NombreProduits	50
NombreMagasins	25
NombreClients	50
NombreEmployes	50
NombreTransactions	160

Query executed successfully.

DESKTOP-A9RMIDD (16.0 RTM) | DESKTOP-A9RMIDD\HP (54) | Sourcesystem | 00:00:01 | 5 rows

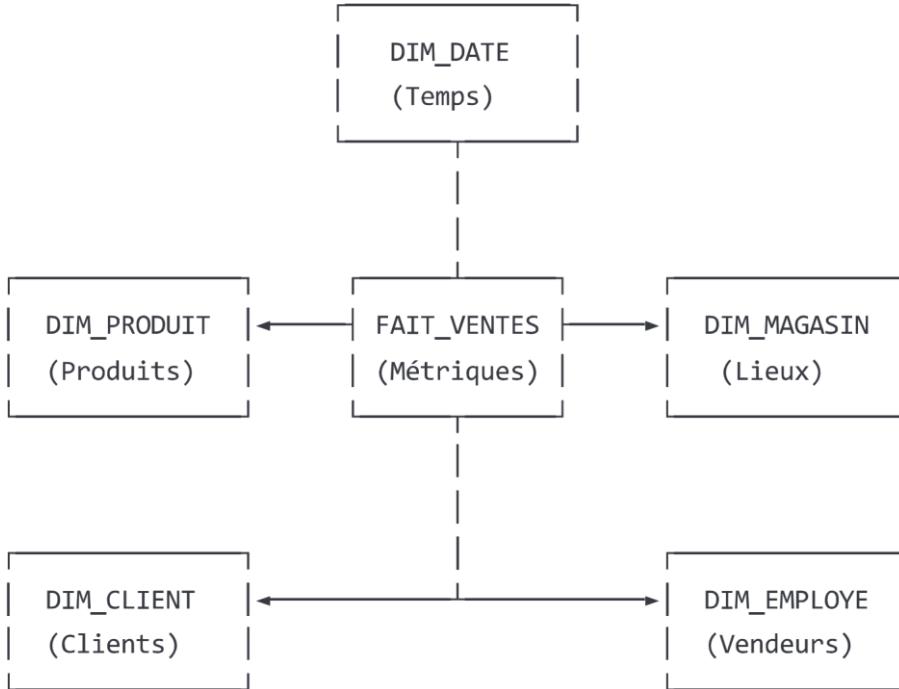
Identification des Besoins Analytiques

Avant de concevoir le modèle, nous avons identifié les principaux axes d'analyse nécessaires à IVOIRE SHOP :

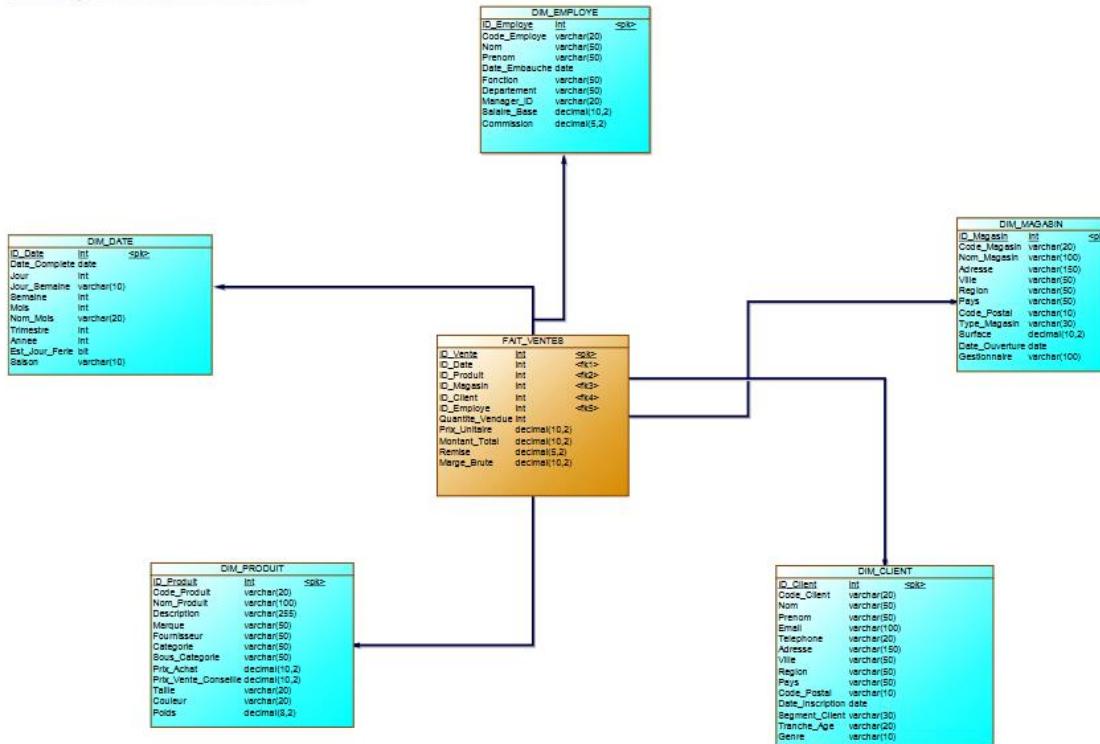
- **Analyse temporelle** des ventes (par jour, mois, trimestre, année)
- **Analyse par produit** et catégorie de produit
- **Analyse géographique** par magasin et région
- **Segmentation** par type de client
- **Évaluation des performances** des employés

Conception du Schéma en Étoile

Notre modèle dimensionnel comprend une table de faits centrale reliée à cinq tables de dimensions clés :



2.2 Diagramme Modèle en Etoile



Tables de Dimensions

1. DIM_DATE

La dimension temporelle stocke toutes les dates avec leurs attributs hiérarchiques et calendaires :

DIM_DATE	
ID_DATE	Clé primaire surrogate (YYYYMMDD)
DATE_COMPLETE	Date complète au format date
JOUR	Numéro du jour dans le mois (1-31)
JOUR_SEMAINE	Nom du jour (Lundi, Mardi, etc.)
JOUR_SEMAINE_NUM	Numéro du jour (1-7)
MOIS	Nom du mois
MOIS_NUM	Numéro du mois (1-12)
TRIMESTRE	Numéro du trimestre (1-4)
ANNEE	Année (YYYY)
WEEKEND	Indicateur weekend (Oui/Non)
JOUR_FERIE	Indicateur jour férié (Oui/Non)
SAISON	Saison (Été, Automne, Hiver, etc.)

2. DIM_PRODUIT

Contient les attributs descriptifs des produits :

DIM_PRODUIT	
ID_PRODUIT	Clé primaire surrogate
CODE_PRODUIT	Code produit d'origine
NOM_PRODUIT	Nom commercial du produit
DESCRIPTION	Description détaillée
MARQUE	Nom de la marque
CATEGORIE	Catégorie principale du produit
SOUS_CATEGORIE	Sous-catégorie du produit
PRIX_ACHAT	Prix d'achat unitaire
PRIX_VENTE	Prix de vente standard
TAILLE	Taille ou dimension
COULEUR	Couleur du produit
DATE_AJOUT	Date d'ajout au catalogue

3. DIM_MAGASIN

Stocke les informations sur les points de vente :

DIM_MAGASIN	
ID_MAGASIN	Clé primaire surrogate
CODE_MAGASIN	Code magasin d'origine
NOM_MAGASIN	Nom commercial du magasin
ADRESSE	Adresse physique
VILLE	Ville d'implantation
REGION	Région administrative
PAYS	Pays (Côte d'Ivoire par défaut)
TYPE_MAGASIN	Type de point de vente
SURFACE	Surface commerciale en m ²
DATE_OUVERTURE	Date d'ouverture du magasin

4. DIM_CLIENT

Reroupe les données clients avec leur segmentation :

DIM_CLIENT	
ID_CLIENT	Clé primaire surrogat
CODE_CLIENT	Code client d'origine
NOM	Nom de famille
PRENOM	Prénom
EMAIL	Adresse email
TELEPHONE	Numéro de téléphone
ADRESSE	Adresse postale
VILLE	Ville de résidence
REGION	Région de résidence
SEGMENT	Segment client
DATE_INSCRIPTION	Date de création du compte client
SEXЕ	Genre du client
TRANCHE AGE	Tranche d'âge

5. DIM_EMPLOYE

Contient les informations sur les vendeurs :

DIM_EMPLOYE	
ID_EMPLOYE	Clé primaire surrogat
CODE_EMPLOYE	Code employé d'origine
NOM	Nom de famille
PRENOM	Prénom
DATE_EMBAUCHE	Date d'embauche
FONCTION	Poste occupé
ID_SUPERIEUR	Référence hiérarchique
ID_MAGASIN	Magasin d'affectation

```
-- Vérifier la création des tables de dimension
USE EntrepotVente;

-- Vérifier la dimension Date
SELECT COUNT(*) AS NombreDates FROM DIM_DATE;

-- Vérifier la dimension Produit
SELECT COUNT(*) AS NombreProduits FROM DIM_PRODUIT;

-- Vérifier la dimension Magasin
SELECT COUNT(*) AS NombreMagasins FROM DIM_MAGASIN;

-- Vérifier la dimension Client
SELECT COUNT(*) AS NombreClients FROM DIM_CLIENT;
```

NombreDates	1096
NombreProduits	50
NombreMagasins	25
NombreClients	50
NombreEmployes	50

Query executed successfully.

Table de Faits

FAIT_VENTES

La table de faits centrale enregistre les métriques clés de l'activité commerciale :

FAIT_VENTES

ID_DATE	Clé étrangère vers DIM_DATE
ID_PRODUIT	Clé étrangère vers DIM_PRODUIT
ID_MAGASIN	Clé étrangère vers DIM_MAGASIN
ID_CLIENT	Clé étrangère vers DIM_CLIENT
ID_EMPLOYE	Clé étrangère vers DIM_EMPLOYE
QUANTITE	Nombre d'unités vendues
PRIX_UNITAIRE	Prix unitaire appliqué
MONTANT_TOTAL	Montant total (avant remise)
REMISE	Montant de la remise appliquée
MARGE_BRUTE	Marge brute réalisée

The screenshot shows the Microsoft SQL Server Management Studio (SSMS) interface. On the left, the Object Explorer displays the database structure, including databases like 'EntrepotVente' and 'DESKTOP-A9RMIDD'. The central pane shows a T-SQL query being run against the 'EntrepotVente' database:

```
-- Vérifier les relations entre la table de faits et les dimensions
USE EntrepotVente;
SELECT
    COUNT(*) AS NombreVentesTotal,
    COUNT(DISTINCT ID_Produit) AS NombreProduitsDifferents,
    COUNT(DISTINCT ID_Magasin) AS NombreMagasinsDifferents,
    COUNT(DISTINCT ID_Client) AS NombreClientsDifferents,
    COUNT(DISTINCT ID_Employe) AS NombreEmployesDifferents,
    COUNT(DISTINCT ID_Date) AS NombreDatesDifferentes
FROM FAIT_VENTES;
```

The results of the query are displayed in a grid:

	NombreVentesTotal	NombreProduitsDifferents	NombreMagasinsDifferents	NombreClientsDifferents	NombreEmployesDifferents	NombreDatesDifferentes
1	160	50	25	45	9	160

At the bottom of the screen, a message bar indicates: "Query executed successfully." and shows the session details: DESKTOP-A9RMIDD (16.0 RTM) | DESKTOP-A9RMIDD\HP (54) | EntrepotVente.

```
-- Vérifier la création et le remplissage de la table de faits
USE EntrepotVente;
SELECT COUNT(*) AS NombreVentes FROM FAIT_VENTES;
```

NombreVentes
160

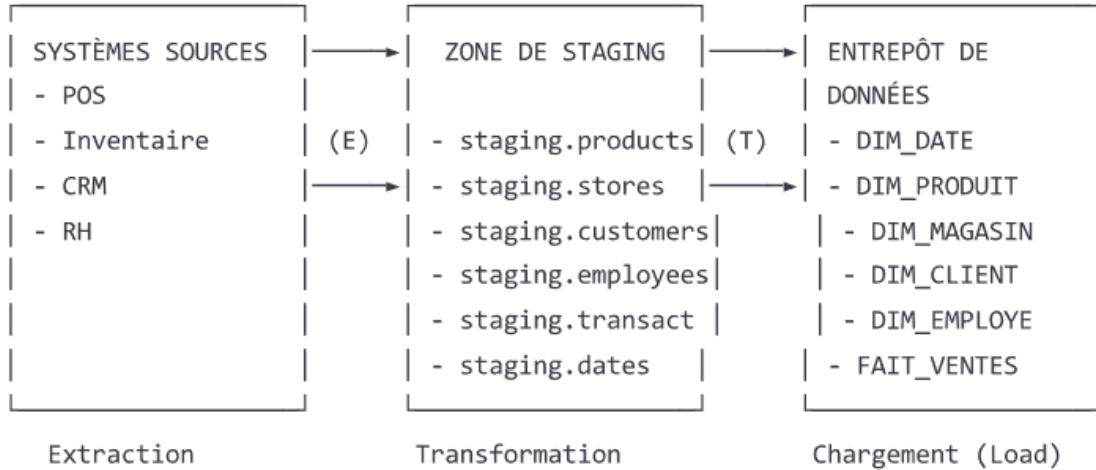
Query executed successfully.

Le grain de la table de faits est défini au niveau de la ligne de transaction individuelle, représentant la vente d'un produit spécifique, à un client particulier, par un employé donné, dans un magasin précis, à une date déterminée.

Gestion des Clés et Zone de Staging

Pour chaque dimension, nous avons créé des clés primaires surrogates (ID_Date, ID_Produit, ID_Magasin, etc.) plutôt que d'utiliser les clés métier d'origine. Cette approche facilite la gestion des changements dans les dimensions et améliore les performances des jointures.

Un schéma de staging a été prévu comme zone intermédiaire entre les systèmes sources et l'entrepôt final :



Extraction

Transformation

Changement (Load)

-- Vérifier la création des tables de staging

```
USE EntrepotVente;
SELECT TABLE_NAME
FROM INFORMATION_SCHEMA.TABLES
WHERE TABLE_SCHEMA = 'staging';
```

TABLE_NAME
transactions
products
stores
customers
employees
dates

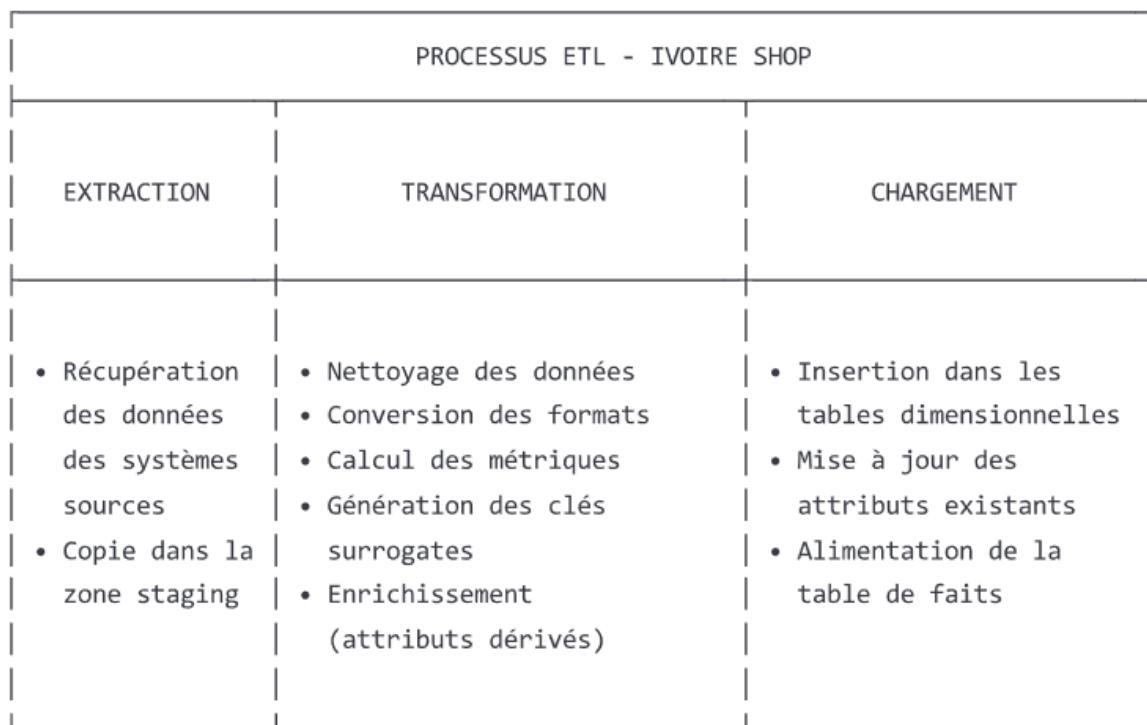
Query executed successfully.

Cette zone comprend des tables temporaires qui permettent de préparer et transformer les données avant leur chargement dans les dimensions et la table de faits.

3. PROCESSUS ETL

Architecture du Processus

Le processus ETL (Extract, Transform, Load) développé pour IVOIRE SHOP suit une approche en trois phases distinctes :



Cette architecture minimise l'impact sur les systèmes opérationnels et optimise les performances du processus d'intégration.

Procédures ETL pour les Dimensions

Dimension Produit

Deux procédures complémentaires ont été créées :

ETL_Extract_Products
<ul style="list-style-type: none">• Purge la table staging.products• Extrait les données de inventory.products• Copie les données dans staging.products

ETL_Load_DIM_PRODUIT

- Compare staging.products avec DIM_PRODUIT
- Met à jour les attributs des produits existants
- Insère les nouveaux produits
- Conserve les clés surrogates

Dimension Magasin

De façon similaire :

ETL_Extract_Stores

- Purge la table staging.stores
- Extrait les données de inventory.stores
- Copie les données dans staging.stores

ETL_Load_DIM_MAGASIN

- Compare staging.stores avec DIM_MAGASIN
- Met à jour les attributs des magasins existants
- Insère les nouveaux magasins
- Conserve les clés surrogates

Dimension Client et Dimension Employé

Le même principe est appliqué pour les dimensions Client et Employé avec leurs procédures respectives:

ETL_Extract_Customers → ETL_Load_DIM_CLIENT

ETL_Extract_Employees → ETL_Load_DIM_EMPLOYE

Dimension Date

La dimension Date est traitée différemment car elle est générée algorithmatiquement :

Orchestration et Automatisation

Pour coordonner l'ensemble du processus ETL, une procédure maître a été développée :

ETL_Generate_Load_DIM_DATE
<ul style="list-style-type: none">• Génère un calendrier (2010 à actuel+10 ans)• Calcule tous les attributs calendaires (jour, mois, trimestre, année, saison)• Stocke d'abord dans staging.dates• Charge dans DIM_DATE avec clé au format YYYYMMDD

- Génère un calendrier (2010 à actuel+10 ans)
- Calcule tous les attributs calendaires (jour, mois, trimestre, année, saison)
- Stocke d'abord dans staging.dates
- Charge dans DIM_DATE avec clé au format YYYYMMDD

Procédures ETL pour la Table de Faits

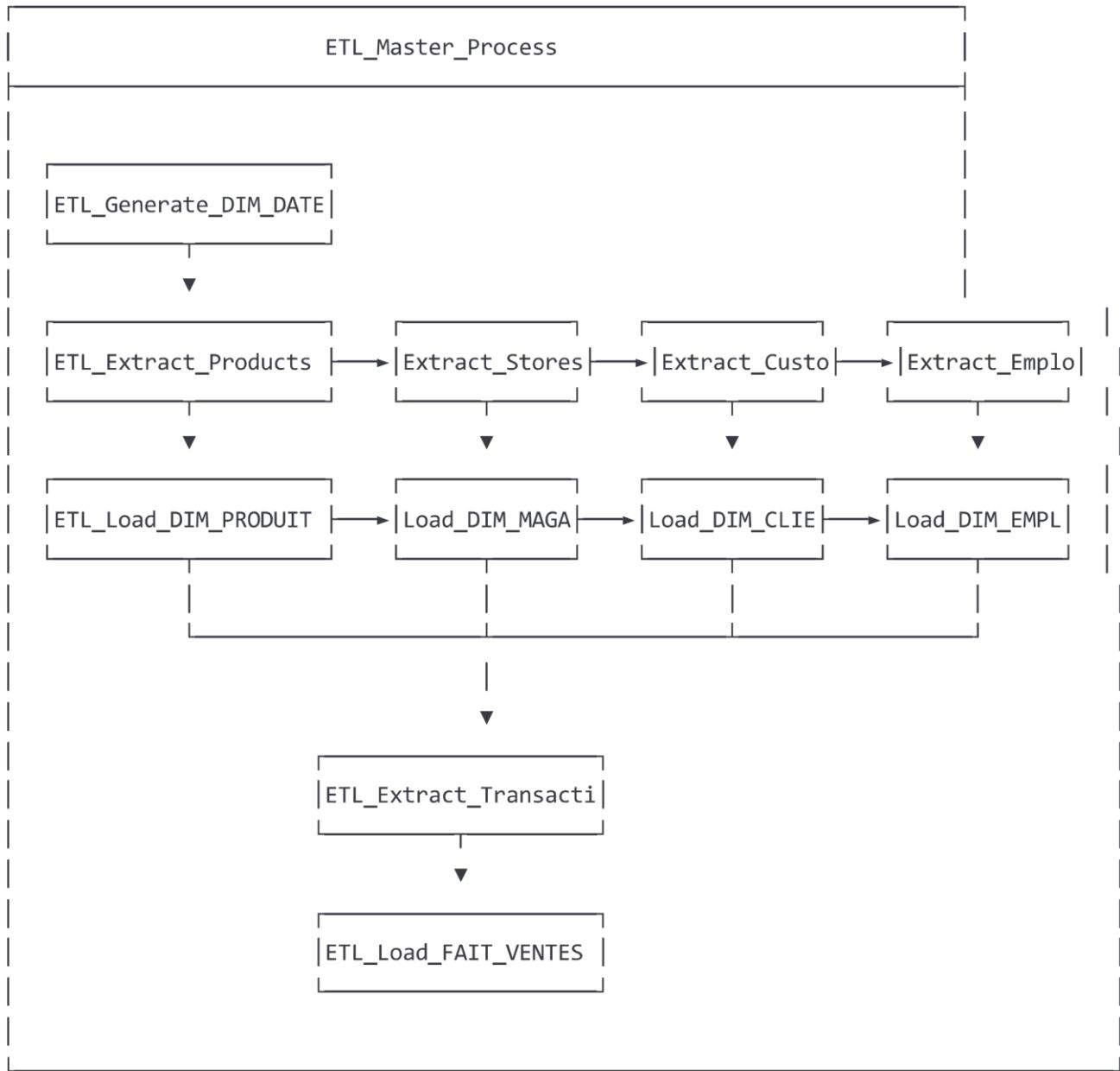
Pour alimenter la table FAIT_VENTES :

ETL_Extract_Transactions
<ul style="list-style-type: none">• Extrait les transactions du système POS• Calcule la marge brute en utilisant le prix d'achat des produits• Stocke dans staging.transactions

- Extrait les transactions du système POS
- Calcule la marge brute en utilisant le prix d'achat des produits
- Stocke dans staging.transactions

ETL_Load_FAIT_VENTES
<ul style="list-style-type: none">• Convertit les dates au format ID_DATE• Récupère les clés surrogates des dimensions• Vérifie l'absence de doublons• Insère les nouvelles transactions• Conserve les métriques calculées

- Convertit les dates au format ID_DATE
- Récupère les clés surrogates des dimensions
- Vérifie l'absence de doublons
- Insère les nouvelles transactions
- Conserve les métriques calculées



Cette procédure maître exécute séquentiellement toutes les procédures ETL dans l'ordre approprié :

1. Génération et chargement de la dimension Date (en premier pour garantir l'existence de toutes les clés temporelles)
2. Extraction des données pour toutes les dimensions à partir des systèmes sources
3. Chargement des dimensions dans l'ordre approprié
4. Extraction des transactions
5. Chargement de la table de faits

La procédure maître peut être exécutée manuellement ou programmée pour s'exécuter à intervalles réguliers selon les besoins de fraîcheur des données.

The screenshot shows two separate sessions in SQL Server Management Studio (SSMS) illustrating the results of an Extract, Transform, Load (ETL) process.

Session 1:

```
-- Vérifier la création des procédures stockées ETL
USE EntrepotVente;
SELECT name
FROM sys.procedures
WHERE name LIKE 'ETL%';
```

Session 2:

```
-- Vérifier les résultats après l'ETL
SELECT COUNT(*) FROM FAIT_VENTES;
```

Results from Session 1:

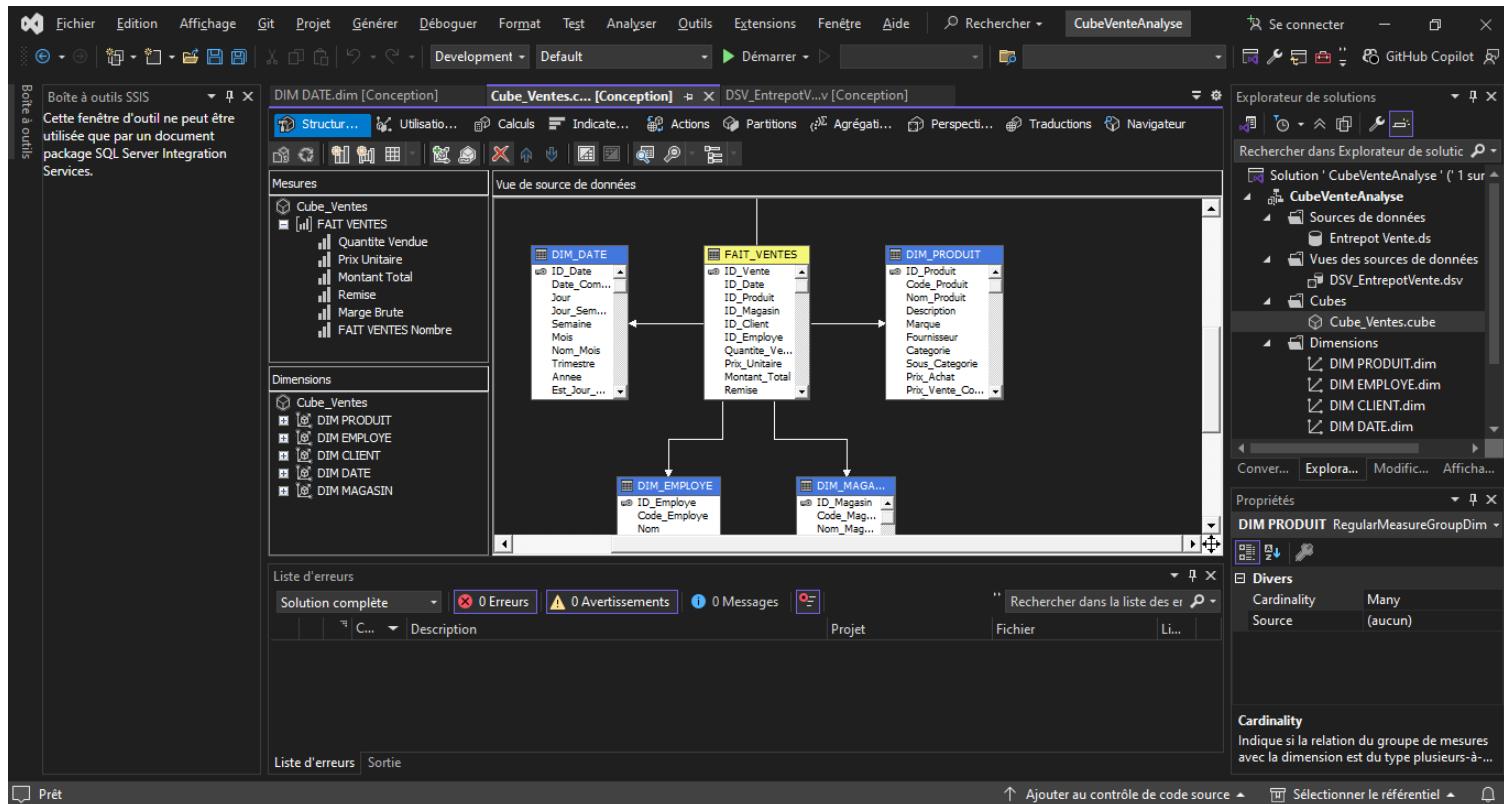
name
1 ETL_Extract_Customers
2 ETL_Extract_Employees
3 ETL_Extract_Products
4 ETL_Extract_Stores
5 ETL_Extract_Transactions
6 ETL_Generate_Load_DIM_DATE
7 ETL_Load_DIM_CLIENT
8 ETL_Load_DIM_EMPLOYEE
9 ETL_Load_DIM_MAGASIN
10 ETL_Load_DIM_PRODUCT
11 ETL_Load_FAIT_VENTES
12 ETL_Merge_Procedures

Results from Session 2:

(No column name)
1 160

4- CONCEPTION DU CUBE OLAP

Ci-dessous l'interface de représentation de notre cube « Cube_Ventes » sous Visual Studio 2022



5- 7 requêtes MDX sur notre cube « Cube_Ventes »

Requête 1 :

The screenshot shows the SSAS Management Studio interface with multiple tabs open. The active tab is "MDXQuery19.mdx...KTOP-A9RMIDD\HP)". The query results show the top 5 stores by total sales:

Boutique	Montant Total_FCFA
Boutique Treichville	1 816 700 FCFA
Centre Commercial San Pedro	1 762 000 FCFA
Boutique Cocody	1 630 700 FCFA
Centre Commercial Riviera	1 586 500 FCFA
Boutique Yamoussoukro	1 489 400 FCFA

The status bar at the bottom indicates "Query executed successfully." and "00:00:01".

Requête 2 :

MDXQuery20.mdx...KTOP-A9RMIDD\HP* MDXQuery19.mdx...KTOP-A9RMIDD\HP* MDXQuery17.mdx...KTOP-A9RMIDD\HP* SQLQuery36.sql - not connected*

Cube: Cube_Ventes

Measure Group: <All>

```
-- 2. Comparaison du chiffre d'affaires et de la marge brute
WITH
MEMBER [Measures].[CA_FCFA] AS
[Measures].[Montant Total],
FORMAT_STRING = "#,#0.00 FCFA"
MEMBER [Measures].[Marge FCFA] AS
[Measures].[Marge Brute],
FORMAT_STRING = "#,#0.00 FCFA"
SELECT
{[Measures].[CA_FCFA], [Measures].[Marge FCFA]} ON COLUMNS,
NON EMPTY
[DIM MAGASIN].[Type Magasin].Members *
[DIM MAGASIN].[Region].Members
ON ROWS
FROM [Cube_Ventes]
```

100 %

		CA_FCFA	Marge FCFA
All	Cocody	1 914 400,00 FCFA	852 900,00 FCFA
All	Gôh	108 000,00 FCFA	64 000,00 FCFA
All	Grands-Ponts	98 200,00 FCFA	51 200,00 FCFA
All	Guémon	54 800,00 FCFA	29 000,00 FCFA
All	Haut-Sassandra	464 500,00 FCFA	180 900,00 FCFA
All	Indénié-Djuablin	1 166 400,00 FCFA	496 400,00 FCFA
All	Kabadougou	748 000,00 FCFA	334 000,00 FCFA
All	Lôh-Djiboua	547 500,00 FCFA	226 000,00 FCFA
All	Marcory	956 000,00 FCFA	438 600,00 FCFA
All	Nord	665 500,00 FCFA	286 500,00 FCFA
All	Plateau	988 000,00 FCFA	488 000,00 FCFA

Query executed successfully.

(local) DESKTOP-A9RMIDD\HP CubeVenteAnalyse 00:00:01

Requête 3 :

MDXQuery20.mdx...KTOP-A9RMIDD\HP* MDXQuery19.mdx...KTOP-A9RMIDD\HP* MDXQuery17.mdx...KTOP-A9RMIDD\HP* SQLQuery36.sql - not connected*

Cube: Cube_Ventes

Measure Group: <All>

```
--Requête 3 : Ventes trimestrielles par segment client (en FCFA)
WITH MEMBER [Measures].[Montant Total_FCFA] AS
[Measures].[Montant Total],
FORMAT_STRING = "#,#0 FCFA"
SELECT
NON EMPTY [DIM CLIENT].[Segmentation].[Segment Client].MEMBERS ON ROWS,
NON EMPTY [DIM DATE].[Calendrier].[Trimestre].MEMBERS ON COLUMNS
FROM [Cube_Ventes]
WHERE [Measures].[Montant Total_FCFA]
```

100 %

	1	2	3	4	1
Premium	2 305 000 FCFA	1 188 700 FCFA	662 900 FCFA	1 304 500 FCFA	684 500 FCFA
Standard	3 979 500 FCFA	1 577 500 FCFA	2 842 000 FCFA	1 824 300 FCFA	1 362 500 FCFA

Query executed successfully.

(local) DESKTOP-A9RMIDD\HP CubeVenteAnalyse 00:00:01

Requête 4 :

MDXQuery20.mdx...KTOP-A9RMIDD\HP* MDXQuery19.mdx...KTOP-A9RMIDD\HP* MDXQuery17.mdx...KTOP-A9RMIDD\HP* SQLQuery36.sql - not connected*

--Requête 4 : Performance des vendeurs (employés) par département (en FCFA)

```

WITH MEMBER [Measures].[Montant Total_FCFA] AS
    [Measures].[Montant Total],
    FORMAT_STRING = "#,##0 FCFA"
SELECT
    NON EMPTY EXCEPT([DIM EMPLOYE].[Nom].MEMBERS, {[DIM EMPLOYE].[Nom].[A11]}) ON ROWS,
    {[Measures].[Montant Total_FCFA], [Measures].[Quantite Vendue]} ON COLUMNS
FROM [Cube_Ventes]
WHERE [DIM EMPLOYE].[Departement].&[Ventes]
  
```

100 %

	Montant Total_FCFA	Quantite Vendue
Adou	4 679 800 FCFA	076
Ahoure	827 900 FCFA	026
Bah	1 788 900 FCFA	076
Bakayoko	569 800 FCFA	044
Gohou	2 029 700 FCFA	026
Kalou	4 947 900 FCFA	061
Konaté	876 400 FCFA	032
Koné	1 204 000 FCFA	036
Traoré	807 000 FCFA	029

Query executed successfully.

(local) | DESKTOP-A9RMIDD\HP | CubeVenteAnalyse | 00:00:01

Requête 5 :

MDXQuery20.mdx...KTOP-A9RMIDD\HP* MDXQuery19.mdx...KTOP-A9RMIDD\HP* MDXQuery17.mdx...KTOP-A9RMIDD\HP* SQLQuery36.sql - not connected*

--Requête 5 : Top 3 des produits les plus vendus par catégorie (en FCFA)

```

WITH
    MEMBER [Measures].[Montant Total_FCFA] AS
        [Measures].[Montant Total],
        FORMAT_STRING = "#,##0 FCFA"
    SET [Top3ProduitsCat] AS
        GENERATE(
            [DIM PRODUIT].[Categorie].MEMBERS,
            TOPCOUNT(
                EXISTING [DIM PRODUIT].[Nom Produit].MEMBERS,
                3,
                [Measures].[Montant Total]
            )
        )
SELECT
  
```

100 %

	Montant Total_FCFA
All	17 731 400 FCFA
Laptop Pavilion	3 250 000 FCFA
Téléviseur Smart TV	2 500 000 FCFA
Sac à main Kita	280 000 FCFA
Perles de taille	84 000 FCFA
Poivre de Man	54 000 FCFA
Chocolat Abidjan	42 500 FCFA
Éventail en raphia	24 500 FCFA
Bracelet Akan	600 000 FCFA
Chaussures Bassam	210 000 FCFA
Sandales Cuir	72 000 FCFA

Query executed successfully.

(local) | DESKTOP-A9RMIDD\HP | CubeVenteAnalyse | 00:00:01

Requête 6 :

MDXQuery20.mdx...KTOP-A9RMIDD\HP)* MDXQuery19.mdx...KTOP-A9RMIDD\HP)* MDXQuery17.mdx...KTOP-A9RMIDD\HP)* SQLQuery36.sql - not connected*

Cube: Cube_Ventes

Measure Group: <All>

```
--Requête 6 : Performance des produits de marques
WITH MEMBER [Measures].[Montant Total_FCFA] AS
    [Measures].[Montant Total],
    FORMAT_STRING = "#,##0 FCFA"
SELECT
    NON EMPTY [DIM PRODUIT].[Marque].MEMBERS ON ROWS,
    {[Measures].[Montant Total_FCFA], [Measures].[Quantite Vendue]} ON COLUMNS
FROM [Cube_Ventes]
```

100 %

Messages Results

	Montant Total_FCFA	Quantite Vendue
Cacao Excellence	28 000 FCFA	007
Canon	230 000 FCFA	002
Choco Ivoire	42 500 FCFA	017
Coco Cl	32 000 FCFA	008
Délices Cl	25 200 FCFA	021
Délices d'Abidjan	13 600 FCFA	017
Épices de l'Ouest	54 000 FCFA	012
HP	3 250 000 FCFA	005
JBL	90 000 FCFA	002
Karite Pur	52 500 FCFA	007
Kenza	210 000 FCFA	006

Query executed successfully. | (local) | DESKTOP-A9RMIDD\HP | CubeVenteAnalyse | 00:00:01

Requête 7 :

MDXQuery20.mdx...KTOP-A9RMIDD\HP)* MDXQuery19.mdx...KTOP-A9RMIDD\HP)* MDXQuery17.mdx...KTOP-A9RMIDD\HP)* SQLQuery36.sql - not connected*

Cube: Cube_Ventes

Measure Group: <All>

```
--Requête 7 : Analyse des ventes mensuelles avec variation en pourcentage
WITH
MEMBER [Measures].[Montant Total_FCFA] AS
    [Measures].[Montant Total],
    FORMAT_STRING = "#,##0 FCFA"
MEMBER [Measures].[Variation] AS
    ([DIM DATE].[Calendrier].CurrentMember, [Measures].[Montant Total_FCFA]) /
    ([DIM DATE].[Calendrier].PrevMember, [Measures].[Montant Total_FCFA]) - 1,
    FORMAT_STRING = "Percent"
SELECT
    {[Measures].[Montant Total_FCFA], [Measures].[Variation]} ON COLUMNS,
    NON EMPTY DESCENDANTS([DIM DATE].[Calendrier].[Annee]&[2023], [DIM DATE].[Calendrier].[Mois]) ON ROWS
FROM [Cube_Ventes]
```

100 %

Messages Results

	Montant Total_FCFA	Variation
1	2 207 500 FCFA	inf
2	1 825 500 FCFA	-17,30%
3	2 251 500 FCFA	23,34%
4	980 000 FCFA	-56,47%
5	864 700 FCFA	-11,77%
6	921 500 FCFA	6,57%
7	667 000 FCFA	-27,62%
8	1 448 200 FCFA	117,12%
9	1 389 700 FCFA	-4,04%
10	917 200 FCFA	-34,00%
11	794 000 FCFA	-13,43%

Query executed successfully. | (local) | DESKTOP-A9RMIDD\HP | CubeVenteAnalyse | 00:00:01

