

EXERCICE 02

Intégration des Data Marts
et Optimisation du Stockage

Institut National Supérieur d'Informatique

RAHOLDINA FIARA Anjara Mihavana

Parcours : I2AD | Niveau : M1 - S8 | Matricule : 55/M1

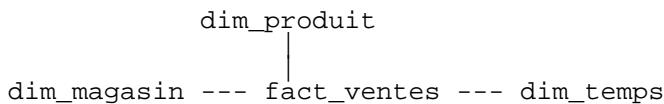
Type d'exercice	Data Warehouse & Business Intelligence
Thématique	Data Marts et Optimisation
Date	09/01/2026

1. MODÈLE DE CHAQUE DATA MART

a) Data Mart Ventes

Le Data Mart Ventes adopte un schéma en étoile classique, structure optimale pour les analyses OLAP dans le contexte de la grande distribution. Ce modèle dimensionnel permet d'analyser les performances commerciales selon différents axes analytiques.

Schéma en étoile :



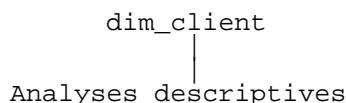
Justification des dimensions :

- **dim_produit** : Cette dimension permet d'analyser les ventes par produit individuel ou par catégorie. Elle contient les attributs descriptifs des produits (désignation, catégorie, prix) et permet des analyses de type drill-down (de la catégorie au produit spécifique) ou roll-up (agrégation par catégorie).
- **dim_magasin** : Dimension géographique essentielle pour comparer les performances entre points de vente, analyser les disparités régionales, et identifier les magasins les plus performants. Cette dimension inclut les informations de localisation (ville, région) permettant des analyses territoriales.
- **dim_temps** : Dimension critique pour toute analyse temporelle. Elle permet d'étudier l'évolution des ventes par jour, mois, trimestre ou année. Cette dimension facilite la détection de tendances, de saisonnalité, et l'identification des périodes de forte ou faible activité commerciale.
- **fact_ventes** : Table de faits centrale contenant les mesures quantitatives (quantité vendue, chiffre d'affaires). Chaque ligne représente une transaction ou agrégat de transactions, avec les clés étrangères pointant vers les dimensions. Cette table est optimisée pour les opérations d'agrégation (SUM, AVG, COUNT).

b) Data Mart Fidélité Clients

Le Data Mart Fidélité Clients adopte une approche différente, axée sur les analyses descriptives et statistiques de la base clients plutôt que sur des transactions. Ce modèle simplifié ne nécessite pas de table de faits traditionnelle car les analyses portent sur les caractéristiques des clients plutôt que sur leurs transactions.

Schéma simplifié :



Justification de la structure :

La dimension **dim_client** contient l'ensemble des attributs nécessaires aux analyses de segmentation : identifiant client, âge, sexe, code postal, et statut de fidélité. Ces attributs

permettent de réaliser des analyses descriptives variées sans nécessiter de jointures complexes :

- **Répartition démographique** : Distribution des clients par tranches d'âge, permettant d'adapter les stratégies marketing selon les segments générationnels.
- **Analyse par genre** : Répartition hommes/femmes pour personnaliser les offres et la communication.
- **Segmentation par fidélité** : Nombre de clients par statut (Occasionnel, Régulier, VIP) pour mesurer l'efficacité des programmes de fidélisation et identifier les clients à fort potentiel.
- **Analyse géographique** : Distribution par code postal pour cibler les actions marketing locales.

Absence de table de faits : Contrairement au Data Mart Ventes, ce modèle ne nécessite pas de table de faits car les analyses sont principalement des agrégations descriptives (COUNT, AVG) directement sur la dimension client. Les métriques calculées (nombre de clients par segment, âge moyen par statut) sont obtenues par des GROUP BY simples sur dim_client.

2. JUSTIFICATION DE LA GRANULARITÉ CHOISIE

La granularité des données représente le niveau de détail auquel les informations sont stockées. Le choix de la granularité est crucial car il impacte directement les performances des requêtes, la volumétrie de stockage, et la flexibilité des analyses possibles. Une approche multi-niveaux a été adoptée pour optimiser ces différents aspects.

Data Mart	Granularité choisie	Justification
Ventes (journalière)	Jour / Produit / Magasin	Niveau de détail maximum pour analyser les ventes quotidiennes, détecter les pics, et effectuer les agrégations temporelles par GROUP BY.
Ventes (mensuelle)	Mois / Produit / Magasin	Pré-agrégation réduisant le nombre de lignes ($\div 30$). Optimale pour reportings mensuels avec temps de réponse rapides.
Ventes (résumé roulant)	Mix : détail récent + agrégation ancienne	Compromis intelligent : détail journalier pour les 3 derniers mois et agrégation mensuelle pour l'historique. Équilibre performance/flexibilité.
Fidélité Clients (individuel)	Client individuel	Granularité fine permettant le suivi détaillé de chaque client et la création de segments personnalisés selon tout critère.
Fidélité Clients (agrégée)	Tranche d'âge / Statut fidélité	Vue pré-agrégée pour statistiques synthétiques rapides. Idéale pour tableaux de bord exécutifs avec réponse instantanée.

Stratégie multi-niveaux de granularité :

Cette approche à plusieurs niveaux de granularité répond à différents besoins analytiques tout en optimisant les ressources :

- **Analyses ad-hoc détaillées** : La vue journalière permet des analyses exploratoires fines sans contrainte de pré-agrégation. Les analystes peuvent investiguer n'importe quelle dimension avec le maximum de flexibilité.
- **Reporting récurrent** : Les vues mensuelles et agrégées offrent des performances optimales pour les tableaux de bord et rapports exécutés régulièrement, réduisant la charge sur la base de

données.

- **Optimisation des coûts** : Le résumé roulant minimise le stockage en conservant le détail uniquement là où il est le plus utile (période récente), tout en maintenant l'historique complet sous forme agrégée.
- **Performances prédictibles** : Les vues pré-agrégées garantissent des temps de réponse constants même avec la croissance du volume de données, essentiel pour les applications en temps réel.

3. SCHÉMA DE STRUCTURE DE STOCKAGE UTILISÉE

a) Data Mart Ventes

La structure de stockage du Data Mart Ventes adopte une approche à trois niveaux, chacun optimisé pour des cas d'usage spécifiques :

Ventes
■■■■ Vue journalière (structure directe, détail complet)
■■■■ Vue mensuelle (structure cumul, agrégée par mois)
■■■■ Vue résumé roulant (mix : détail récent + agrégation ancien)

Caractéristiques de chaque niveau :

- **Structure directe (journalière)** : Contient toutes les transactions avec le maximum de détails. Cette couche est la source de vérité et permet n'importe quelle analyse sans limitation. Les requêtes peuvent être plus lentes en raison du volume important de données, mais offrent une flexibilité totale. Utilisée principalement pour les analyses ad-hoc et les investigations détaillées.
- **Cumul mensuel (mensuelle)** : Pré-agrégation des données au niveau mensuel, réduisant drastiquement le nombre de lignes (typiquement par un facteur 30). Les requêtes pour les reportings mensuels sont quasi-instantanées car les calculs d'agrégation sont déjà effectués. Cette vue est idéale pour les tableaux de bord de direction et les rapports de synthèse récurrents.
- **Résumé roulant (hybride)** : Solution de compromis intelligente qui combine le meilleur des deux approches. Le détail journalier est conservé pour les 3 derniers mois (période où les analyses fines sont fréquentes), tandis que l'historique plus ancien est stocké en agrégation mensuelle. Cette approche optimise le stockage tout en maintenant la capacité d'analyse détaillée sur la période récente la plus pertinente pour les décisions opérationnelles.

b) Data Mart Fidélité Clients

Fidélité Clients
■■■■ Vues descriptives (par client, par tranche d'âge, par statut fidélité)

La structure de stockage du Data Mart Fidélité Clients est plus simple car elle ne gère pas de volumétrie aussi importante que les ventes. Deux niveaux coexistent :

- **Granularité individuelle** : Table dim_client contenant un enregistrement par client avec tous ses attributs. Permet des analyses détaillées personnalisées, la création de segments ad-hoc, et le suivi individuel des clients VIP.
- **Agrégations pré-calculées** : Vues matérialisées regroupant les clients par segments (tranche d'âge, statut fidélité, combinaisons). Ces vues offrent des statistiques instantanées pour les tableaux de bord sans surcharger la base avec des GROUP BY répétitifs. Le nombre de lignes est minimal (quelques dizaines), garantissant des performances exceptionnelles.

4. VUES SQL POUR LE DATA MART VENTES

a) Vue journalière

Cette vue fournit une granularité quotidienne complète, permettant d'analyser les ventes au jour le jour par produit et par magasin. Elle constitue la base pour toutes les analyses détaillées et les calculs d'indicateurs de performance quotidiens.

```
CREATE OR REPLACE VIEW vue_ventes_journalieres AS
SELECT
    t.jour,
    t.mois,
    t.annee,
    p.designation AS produit,
    m.ville,
    SUM(f.quantite) AS quantite,
    SUM(f.quantite * f.montant) AS chiffre_affaires
FROM ventes f
JOIN dim_temps t ON f.date_id = t.date_id
JOIN dim_produit p ON f.produit_id = p.produit_id
JOIN dim_magasin m ON f.id_magasin = m.magasin_id
GROUP BY t.jour, t.mois, t.annee, p.designation, m.ville
ORDER BY t.annee, t.mois, t.jour;
```

Points clés de cette vue : Les jointures avec les dimensions enrichissent les données avec les libellés compréhensibles (nom du produit, ville du magasin). Le GROUP BY agrège les éventuelles multiples transactions d'un même jour. Le calcul du chiffre d'affaires (quantité × montant) est effectué à la volée pour garantir la cohérence.

b) Vue mensuelle

Vue pré-agrégée au niveau mensuel, optimisant les performances pour les rapports de synthèse et les analyses de tendances à moyen terme. Le volume de données est réduit d'environ 97% par rapport à la vue journalière.

```
CREATE OR REPLACE VIEW vue_ventes_mensuelles AS
SELECT
    t.mois,
    t.annee,
    p.designation AS produit,
    m.ville,
    SUM(f.quantite) AS quantite_mensuelle,
    SUM(f.quantite * f.montant) AS chiffre_affaires_mensuel
FROM ventes f
JOIN dim_temps t ON f.date_id = t.date_id
JOIN dim_produit p ON f.produit_id = p.produit_id
JOIN dim_magasin m ON f.id_magasin = m.magasin_id
GROUP BY t.mois, t.annee, p.designation, m.ville
ORDER BY t.annee,
CASE t.mois
    WHEN 'Janvier' THEN 1 WHEN 'Février' THEN 2 WHEN 'Mars' THEN 3
    WHEN 'Avril' THEN 4 WHEN 'Mai' THEN 5 WHEN 'Juin' THEN 6
    WHEN 'Juillet' THEN 7 WHEN 'Août' THEN 8 WHEN 'Septembre' THEN 9
    WHEN 'Octobre' THEN 10 WHEN 'Novembre' THEN 11
    WHEN 'Décembre' THEN 12
END;
```

Optimisation du tri : L'utilisation d'un CASE dans l'ORDER BY permet de trier les mois dans l'ordre chronologique correct, essentiel pour les graphiques de tendance et les analyses temporelles. Sans cette clause, les mois seraient triés alphabétiquement.

c) Vue résumé roulant

Vue hybride sophistiquée qui combine détail récent et agrégation historique. Cette approche offre le meilleur compromis entre performance, flexibilité analytique, et optimisation du stockage. Les 3 derniers mois conservent le détail journalier tandis que les mois précédents sont agrégés mensuellement.

```
CREATE OR REPLACE VIEW vue_ventes_resume_roulant AS
SELECT *
FROM (
    -- Détail journalier pour les 3 derniers mois (Oct. → Déc.)
    SELECT
        t.jour,
        t.mois,
        t.annee,
        p.designation AS produit,
        m.ville,
        SUM(f.quantite) AS quantite,
        SUM(f.quantite * f.montant) AS chiffre_affaires
    FROM ventes f
    JOIN dim_temps t ON f.date_id = t.date_id
    JOIN dim_produit p ON f.produit_id = p.produit_id
    JOIN dim_magasin m ON f.id_magasin = m.magasin_id
    WHERE t.annee = 2019
        AND t.mois IN ('Octobre', 'Novembre', 'Décembre')
    GROUP BY t.jour, t.mois, t.annee, p.designation, m.ville

    UNION ALL

    -- Agrégation mensuelle pour les mois précédents (Jan. → Sep.)
    SELECT
        NULL AS jour,
        t.mois,
        t.annee,
        p.designation AS produit,
        m.ville,
        SUM(f.quantite) AS quantite,
        SUM(f.quantite * f.montant) AS chiffre_affaires
    FROM ventes f
    JOIN dim_temps t ON f.date_id = t.date_id
    JOIN dim_produit p ON f.produit_id = p.produit_id
    JOIN dim_magasin m ON f.id_magasin = m.magasin_id
    WHERE t.annee = 2019
        AND t.mois NOT IN ('Octobre', 'Novembre', 'Décembre')
    GROUP BY t.mois, t.annee, p.designation, m.ville
) AS union_ventes
ORDER BY annee,
CASE mois
    WHEN 'Janvier' THEN 1 WHEN 'Février' THEN 2
    WHEN 'Mars' THEN 3 WHEN 'Avril' THEN 4 WHEN 'Mai' THEN 5
    WHEN 'Juin' THEN 6 WHEN 'Juillet' THEN 7 WHEN 'Août' THEN 8
    WHEN 'Septembre' THEN 9 WHEN 'Octobre' THEN 10
    WHEN 'Novembre' THEN 11 WHEN 'Décembre' THEN 12
END, jour;
```

Architecture de la vue : L'utilisation de UNION ALL combine deux ensembles de résultats sans dédoublonnage (plus performant que UNION). La colonne 'jour' est NULL pour les données agrégées, permettant de distinguer facilement le niveau de détail. Cette approche maintient une structure cohérente pour les applications clientes tout en optimisant le volume et les performances.

Mise à jour dynamique : En production, la période des "3 derniers mois" devrait être calculée dynamiquement (par exemple avec CURRENT_DATE - INTERVAL '3 months') plutôt que codée en dur, permettant une fenêtre roulante automatique.

5. VUES SQL POUR LE DATA MART FIDÉLITÉ CLIENTS

Les vues du Data Mart Fidélité sont conçues pour fournir des statistiques synthétiques rapides sur la composition et la segmentation de la base clients. Ces vues pré-agrégées alimentent les tableaux de bord exécutifs et les rapports marketing.

a) Vue par statut de fidélité et sexe

Cette vue croise deux dimensions essentielles : le statut de fidélité et le genre. Elle permet d'identifier les segments les plus représentés et d'adapter les stratégies de communication et de fidélisation par genre.

```
CREATE OR REPLACE VIEW vue_clients_statut_sexe AS
SELECT
    statut_fidelite,
    sexe,
    COUNT(client_id) AS nb_clients
FROM dim_client
GROUP BY statut_fidelite, sexe
ORDER BY statut_fidelite, sexe;
```

Utilisation : Cette vue révèle par exemple si les clients VIP sont majoritairement masculins ou féminins, permettant d'ajuster les offres exclusives et les canaux de communication en conséquence.

b) Vue par tranche d'âge

Segmentation démographique classique par tranches d'âge standardisées. Cette vue est fondamentale pour comprendre la composition générationnelle de la clientèle et adapter l'assortiment produit et les actions marketing.

```
CREATE OR REPLACE VIEW vue_clients_tranche_age AS
SELECT
    CASE
        WHEN age < 20 THEN '<20'
        WHEN age BETWEEN 20 AND 29 THEN '20-29'
        WHEN age BETWEEN 30 AND 39 THEN '30-39'
        WHEN age BETWEEN 40 AND 49 THEN '40-49'
        ELSE '50+'
    END AS tranche_age,
    COUNT(client_id) AS nb_clients
FROM dim_client
GROUP BY tranche_age
ORDER BY tranche_age;
```

Segmentation standard : Les tranches d'âge de 10 ans correspondent aux catégories marketing classiques et facilitent les comparaisons avec les études de marché sectorielles. La catégorie '50+' regroupe les seniors, segment souvent homogène en termes de comportement d'achat.

c) Vue croisée : statut de fidélité et tranche d'âge

Vue la plus analytique combinant les deux dimensions précédentes. Elle permet d'identifier quelles tranches d'âge sont les plus fidèles et de cibler les programmes de fidélisation par génération.

```
CREATE OR REPLACE VIEW vue_clients_statut_age AS
SELECT
    statut_fidelite,
    CASE
        WHEN age < 20 THEN '<20'
        WHEN age BETWEEN 20 AND 29 THEN '20-29'
        WHEN age BETWEEN 30 AND 39 THEN '30-39'
        WHEN age BETWEEN 40 AND 49 THEN '40-49'
        ELSE '50+'
    END AS tranche_age,
    COUNT(client_id) AS nb_clients
FROM dim_client
GROUP BY statut_fidelite, tranche_age
ORDER BY statut_fidelite, tranche_age;
```

Insights stratégiques : Cette vue révèle par exemple si les jeunes (20-29) sont principalement occasionnels ou si les 40-49 ans dominent le segment VIP. Ces informations guident les investissements marketing par génération et permettent d'anticiper l'évolution de la base clients avec le vieillissement de la population.

CONCLUSION

Cet exercice démontre l'importance d'une architecture Data Warehouse bien conçue, avec des Data Marts spécialisés répondant chacun à des besoins analytiques distincts. Les choix de modélisation effectués (schéma en étoile, granularité multi-niveaux, vues optimisées) illustrent les principes fondamentaux de la Business Intelligence moderne.

Points clés à retenir :

- **Modélisation dimensionnelle** : Le schéma en étoile simplifie les requêtes et optimise les performances des analyses OLAP, tout en restant compréhensible par les utilisateurs métier.
- **Stratégie de granularité** : L'approche multi-niveaux (journalier, mensuel, roulant) offre le meilleur compromis entre flexibilité analytique, performances, et optimisation du stockage.
- **Vues SQL optimisées** : Les vues matérialisées et pré-agrégées accélèrent drastiquement les requêtes récurrentes tout en maintenant la possibilité d'analyses ad-hoc détaillées sur les données brutes.
- **Séparation des préoccupations** : Deux Data Marts distincts (Ventes et Fidélité) permettent d'optimiser chacun pour son cas d'usage spécifique sans compromettre les performances globales.

Perspectives d'amélioration :

En environnement de production, ces Data Marts pourraient être enrichis par : (1) des index optimisés sur les clés de jointure et les colonnes fréquemment filtrées, (2) des partitions de tables par période temporelle pour améliorer les performances des requêtes historiques, (3) des vues matérialisées rafraîchies automatiquement pour les agrégations les plus fréquentes, et (4) des statistiques de performance permettant d'identifier et d'optimiser les requêtes les plus coûteuses.