RAPPORT FINAL

SAE 3.02

INTÉGRATION DE DONNÉES DANS UN DATAWAREHOUSE

AHMED TERIR ET ZAKARIA CHAREF

TABLE DES MATIERES

Introduction

Contexte du projet

Dans le cadre de cette Situation d'Apprentissage et d'Évaluation (SAE), nous avons travaillé sur la base de données *Chinook*, une base relationnelle représentant l'activité d'une plateforme de vente de musique. Cette base contient des informations sur les artistes, les albums, les morceaux, les clients, les factures ainsi que les ventes réalisées.

L'objectif du projet est de transformer cette base de données transactionnelle en une architecture décisionnelle permettant l'analyse des données et le calcul d'indicateurs pertinents, tels que le chiffre d'affaires. Pour cela, plusieurs étapes ont été nécessaires, allant de la modélisation des données à la mise en place d'un Data Warehouse, en passant par des phases d'intégration et de transformation des données.

Mettre en place un **Data Warehouse** à partir de la base **Chinook** est particulièrement pertinent, car cela permet de centraliser, structurer et historiser les données issues des ventes, des clients, des artistes, ou encore des employés. Contrairement à une base opérationnelle qui est conçue pour les transactions quotidiennes, le Data Warehouse est optimisé pour l'analyse et la prise de décision. Il facilite la création de tableaux de bord, d'indicateurs clés de performance (KPI) et de rapports stratégiques. Grâce à une modélisation en étoile, les requêtes deviennent plus rapides et compréhensibles pour les utilisateurs métiers. Enfin, il permet une meilleure qualité de données via l'intégration, la normalisation et l'historisation, ce qui est essentiel pour avoir une vision fiable de l'activité sur le long terme.

Objectifs de la SAE

Cette SAE a pour objectif de mettre en pratique les notions vues en cours relatives à la gestion et à l'exploitation des données décisionnelles. Plus précisément, le projet vise à :

- Analyser une base de données relationnelle existante
- Concevoir les différents modèles de données (MCD, MLD, MPD)
- Mettre en place une architecture décisionnelle composée d'un DSA, d'un ODS et d'un Data Warehouse
- Concevoir un modèle en étoile adapté aux besoins analytiques
- Mettre en œuvre les flux d'alimentation des données à l'aide d'Oracle Data Integrator (ODI)
- Vérifier la cohérence et la qualité des données intégrées
- Exploiter les données décisionnelles à travers des requêtes analytiques

Ce projet permet ainsi de comprendre l'ensemble du processus de construction d'une solution décisionnelle, depuis la modélisation initiale jusqu'à l'analyse des données finales.

---

1. Compréhension et analyse du besoin

Présentation du jeu de données Chinook

La base de données *Chinook* est une base relationnelle représentant l'activité d'une plateforme de vente de musique en ligne. Elle contient l'ensemble des informations nécessaires à la gestion des ventes, notamment les artistes, les albums, les morceaux, les clients, les employés, les factures ainsi que les lignes de facturation associées aux ventes.

Cette base est organisée autour de tables transactionnelles, telles que *Invoice* et *InvoiceLine*, qui enregistrent les opérations de vente, ainsi que de tables de référence permettant de décrire les éléments vendus et les acteurs impliqués. La structure de Chinook est typique d'une base orientée gestion, optimisée pour le traitement des transactions quotidiennes.

Analyse du besoin décisionnel

Bien que la base Chinook soit adaptée à la gestion opérationnelle des ventes, elle n'est pas conçue pour répondre efficacement à des besoins d'analyse décisionnelle. En effet, l'extraction d'indicateurs globaux ou l'analyse des ventes selon plusieurs axes nécessite des requêtes complexes et coûteuses en performance sur une base transactionnelle.

Le besoin décisionnel identifié consiste donc à permettre l'analyse des ventes de musique selon différents critères, tels que le temps, les clients, les morceaux ou les employés. L'objectif est de disposer d'une structure de données facilitant le calcul d'indicateurs clés, comme le chiffre d'affaires, les quantités vendues ou la performance commerciale, tout en garantissant la cohérence et la fiabilité des données.

Cette analyse a conduit à la mise en place d'une architecture décisionnelle, incluant des zones intermédiaires (DSA et ODS) et un Data Warehouse, afin de répondre efficacement aux besoins analytiques identifiés.

2. Modélisation conceptuelle et logique des données

Dictionnaire des données

Le dictionnaire des données a été établi afin de documenter précisément chaque attribut présent dans le modèle. Il décrit le nom des champs, leur type, leur signification ainsi que leur rôle dans le système d'information.

Cet outil permet d'assurer une compréhension commune des données et facilite la maintenance, l'évolution et l'exploitation de la base. Il constitue également une référence importante pour les étapes suivantes du projet, notamment lors de la conception du modèle décisionnel et de l'alimentation des différentes zones de données.

Voici un extrait du dictionnaire de données *:*

Dictionnaire des données : DataBase Chinook

Code	Nom de la rebrique	Type	Nature	Regle de calcul	Règle d'intégrité
EmployeeId	Identifiant de l'employé	INT	Élémentaire	Ø	Identifiant, Non nul, Unique
LastName	Nom de l'employé	TEXT	Élémentaire	Ø	Non nul
FirstName	Prénom de l'employé	TEXT	Élémentaire	Ø	Non nul
Title	Fonction de l'employé	TEXT	Élémentaire	Ø	
ReportsTo	Signalement de l'employé à l'employé	TEXT	Élémentaire	Ø	
BirthDate	Date de naissance de l'employé	DATE	Élémentaire	Ø	Non nul
HireDate	Date d'embauche de l'employé	DATE	Élémentaire	Ø	Non nul
Address	Adresse où habite l'employé	TEXT	Élémentaire	Ø	
City	Ville où habite l'employé	TEXT	Élémentaire	Ø	
State	Departement où habite l'employé	INT	Élémentaire	Ø	
Country	Pays où habite l'employé	TEXT	Élémentaire	Ø	
PostalCode	Code postal de la ville où habite l'employé	INT	Élémentaire	Ø	
Phone	Téléphone de l'employé	INT	Élémentaire	Ø	
Fax	Fax de l'employé	INT	Élémentaire	Ø	
Email	Email de l'employé	TEXT	Élémentaire	Ø	

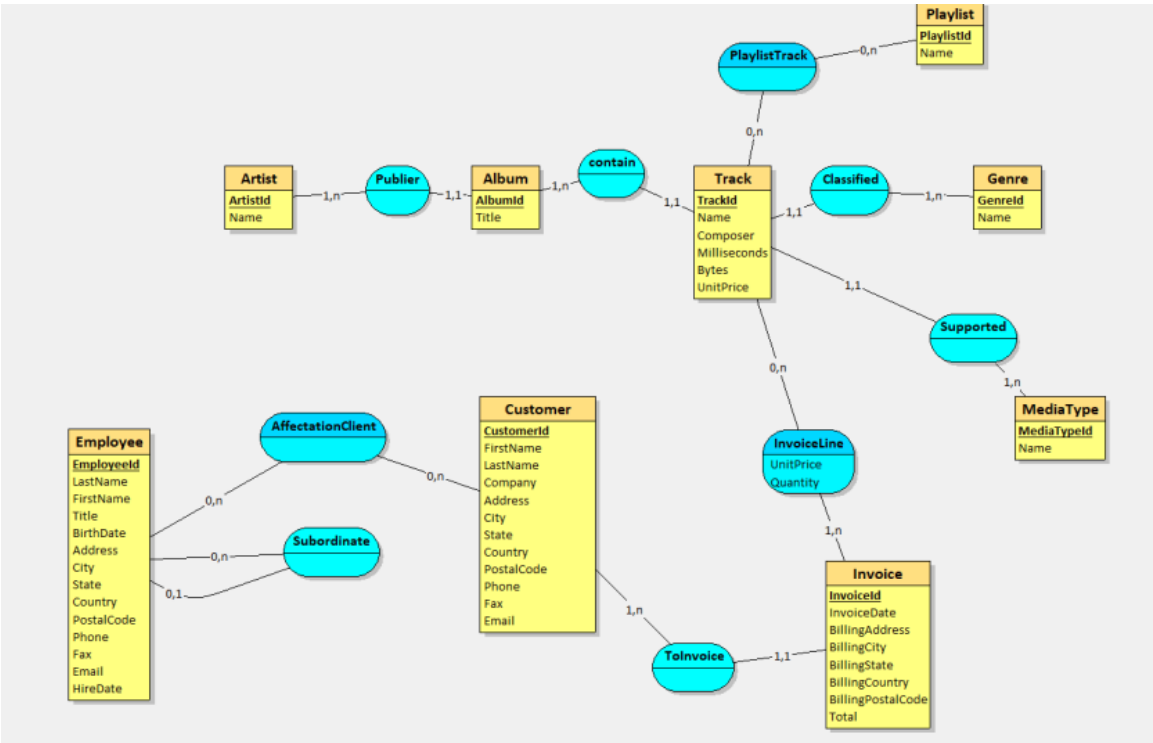
Modèle Conceptuel de Données (MCD)

La première étape de la modélisation a consisté à concevoir le Modèle Conceptuel de Données (MCD). Celui-ci permet de représenter de manière abstraite les entités principales du système d'information ainsi que les relations existant entre elles, indépendamment de toute considération technique.

À partir de l'analyse de la base Chinook, les principales entités identifiées sont notamment les artistes, les albums, les morceaux, les clients, les employés, les factures et les lignes de facturation. Le MCD met en évidence les associations entre ces entités, telles que la relation entre une facture et ses lignes de vente ou encore entre un album et un artiste.

Le MCD constitue une étape essentielle car il permet de valider la compréhension globale des données et des règles de gestion avant de passer à une modélisation plus technique.

Le schéma du MCD est présenté ci-dessous :



Le catalogue est composé des entités **Artist**, **Album** et **Track**, liées de manière hiérarchique : un artiste publie des albums, et chaque album contient des morceaux. Les morceaux sont associés à un **Genre** et à un **MediaType**, ce qui permet leur classification.

Les playlists sont modélisées par une relation plusieurs-à-plusieurs entre **Playlist** et **Track**.
La partie commerciale repose sur les entités **Customer**, **Invoice** et **InvoiceLine**, permettant de représenter les achats réalisés par les clients.
Enfin, l'entité **Employee** permet de modéliser la gestion des clients ainsi que la hiérarchie interne.

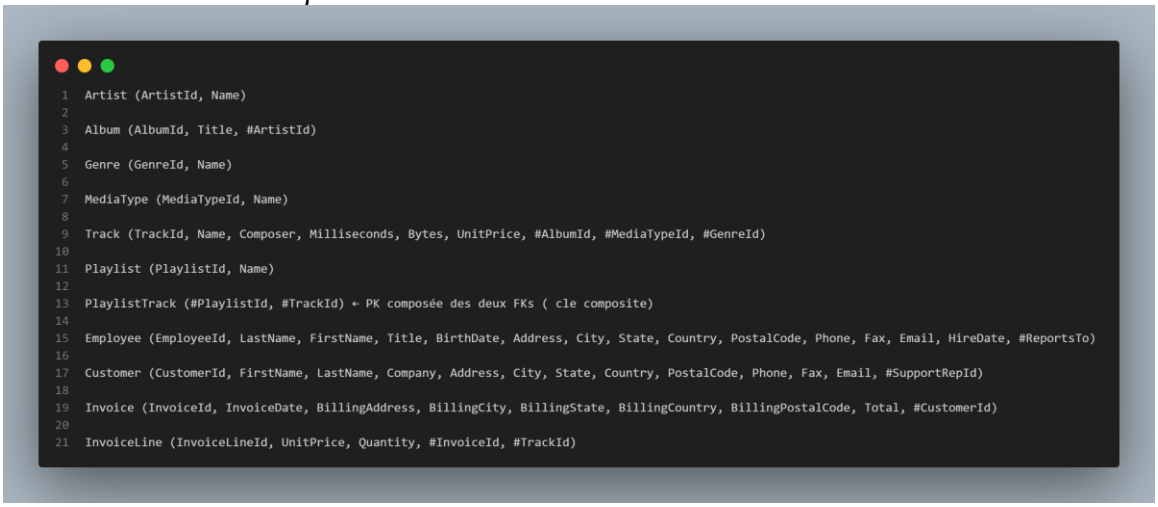
Ce MCD fournit une vision globale et cohérente des données, servant de base à la modélisation logique et décisionnelle.

Modèle Logique de Données (MLD)

À partir du Modèle Conceptuel de Données, un **Modèle Logique de Données textuel** a été élaboré.
Ce MLD traduit les entités et associations du MCD sous forme de tables relationnelles, en définissant les **clés primaires**, les **clés étrangères** et les relations entre les tables.

Le choix d'un MLD textuel permet de présenter clairement la structure relationnelle de la base de données tout en facilitant son implémentation ultérieure dans le SGBD Oracle.

Le schéma du MLD est présenté ci-dessous :



```
Artist (ArtistId, Name)

Album (AlbumId, Title, #ArtistId)

Genre (GenreId, Name)

MediaType (MediaTypeId, Name)

Track (TrackId, Name, Composer, Milliseconds, Bytes, UnitPrice, #AlbumId, #MediaTypeId, #GenreId)

Playlist (PlaylistId, Name)

PlaylistTrack (#PlaylistId, #TrackId) ← PK composée des deux FKs ( cle composite)

Employee (EmployeeId, LastName, FirstName, Title, BirthDate, Address, City, State, Country, PostalCode, Phone, Fax, Email, HireDate, #ReportsTo)

Customer (CustomerId, FirstName, LastName, Company, Address, City, State, Country, PostalCode, Phone, Fax, Email, #SupportRepId)

Invoice (InvoiceId, InvoiceDate, BillingAddress, BillingCity, BillingState, BillingCountry, BillingPostalCode, Total, #CustomerId)

InvoiceLine (InvoiceLineId, UnitPrice, Quantity, #InvoiceId, #TrackId)
```

Chaque table possède une **clé primaire** permettant d'identifier de manière unique les enregistrements, et des **clés étrangères** assurent la cohérence des relations entre les tables (par exemple entre Album et Artist, Track et Album, ou Invoice et Customer).

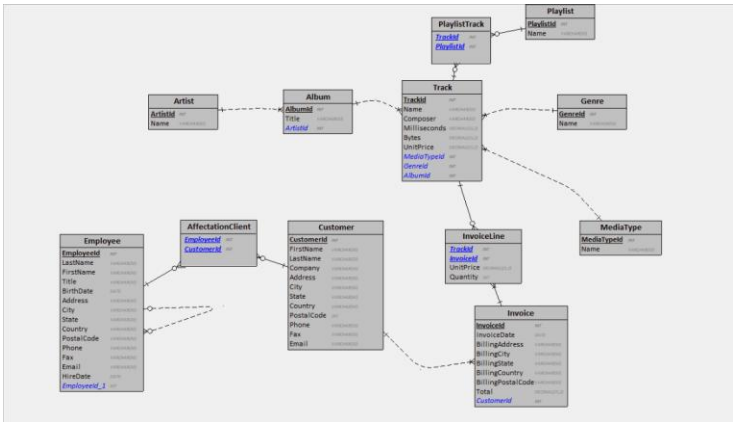
Les relations de type **plusieurs-à-plusieurs**, comme entre Playlist et Track, sont gérées à l'aide de tables de liaison avec une **clé primaire composée**.

---

3. Modélisation physique et base relationnelle

Modèle Physique de Données (MPD)

À partir du MLD, un **Modèle Physique de Données** a été défini afin de préparer l'implémentation concrète de la base dans le SGBD Oracle.
Le MPD précise les **types de données**, les **contraintes d'intégrité** (clés primaires et étrangères) ainsi que la structure exacte des tables, en respectant le modèle relationnel issu de l'analyse.



Création de la base de données relationnelle

La base de données relationnelle a ensuite été créée dans **Oracle** à l'aide de scripts SQL. Ces scripts permettent la création des tables, la définition des clés primaires et étrangères, ainsi que l'organisation des relations entre les différentes entités du modèle Chinook.

Problématique rencontrée lors de l'insertion des données

Lors de l'initialisation de la base de données relationnelle, un premier problème est apparu lors de l'insertion des données. Les scripts SQL fournis sur github initialement n'étaient pas totalement compatibles avec **Oracle SQL Developer**, notamment à cause du format de l'insertion.

Ces erreurs empêchaient l'exécution correcte des requêtes `INSERT` et bloquaient l'alimentation des tables.

Pour résoudre ce problème, un **script Python** a été développé afin de transformer automatiquement les scripts d'insertion au bon format Oracle. Ce script a permis de corriger les chaînes de caractères et de générer des requêtes SQL valides, assurant ainsi une insertion fiable et cohérente des données dans la base relationnelle.

On passe donc de :

Script original

```
INSERT INTO Genre (GenreId, Name) VALUES
    (1, 'Rock'),
    (2, 'Jazz'),
    (3, 'Metal'),
    (4, 'Alternative '||chr(38)||' Punk'),
    (5, 'Rock And Roll'),
    (6, 'Blues'),
    (7, 'Latin'),
    (8, 'Reggae'),
    (9, 'Pop'),
    (10, 'Soundtrack'),
    (11, 'Bossa Nova'),
    (12, 'Easy Listening'),
    (13, 'Heavy Metal'),
    (14, 'R'||chr(38)||'B/Soul'),
    (15, 'Electronica/Dance'),
    (16, 'World'),
    (17, 'Hip Hop/Rap'),
    (18, 'Science Fiction'),
    (19, 'TV Shows'),
    (20, 'Sci Fi '||chr(38)||' Fantasy'),
    (21, 'Drama'),
    (22, 'Comedy'),
    (23, 'Alternative'),
    (24, 'Classical'),
    (25, 'Opera');
```

Nouveau script

```
1  INSERT INTO Genre (GenreId, Name) VALUES (1, 'Rock');
2  INSERT INTO Genre (GenreId, Name) VALUES (2, 'Jazz');
3  INSERT INTO Genre (GenreId, Name) VALUES (3, 'Metal');
4  INSERT INTO Genre (GenreId, Name) VALUES (4, 'Alternative '||chr(38)||' Pun
5  INSERT INTO Genre (GenreId, Name) VALUES (5, 'Rock And Roll');
6  INSERT INTO Genre (GenreId, Name) VALUES (6, 'Blues');
7  INSERT INTO Genre (GenreId, Name) VALUES (7, 'Latin');
8  INSERT INTO Genre (GenreId, Name) VALUES (8, 'Reggae');
9  INSERT INTO Genre (GenreId, Name) VALUES (9, 'Pop');
10 INSERT INTO Genre (GenreId, Name) VALUES (10, 'Soundtrack');
11 INSERT INTO Genre (GenreId, Name) VALUES (11, 'Bossa Nova');
12 INSERT INTO Genre (GenreId, Name) VALUES (12, 'Easy Listening');
13 INSERT INTO Genre (GenreId, Name) VALUES (13, 'Heavy Metal');
14 INSERT INTO Genre (GenreId, Name) VALUES (14, 'R'||chr(38)||'B/Soul');
15 INSERT INTO Genre (GenreId, Name) VALUES (15, 'Electronica/Dance');
16 INSERT INTO Genre (GenreId, Name) VALUES (16, 'World');
17 INSERT INTO Genre (GenreId, Name) VALUES (17, 'Hip Hop/Rap');
18 INSERT INTO Genre (GenreId, Name) VALUES (18, 'Science Fiction');
19 INSERT INTO Genre (GenreId, Name) VALUES (19, 'TV Shows');
20 INSERT INTO Genre (GenreId, Name) VALUES (20, 'Sci Fi '||chr(38)||' Fantasy
21 INSERT INTO Genre (GenreId, Name) VALUES (21, 'Drama');
22 INSERT INTO Genre (GenreId, Name) VALUES (22, 'Comedy');
23 INSERT INTO Genre (GenreId, Name) VALUES (23, 'Alternative');
24 INSERT INTO Genre (GenreId, Name) VALUES (24, 'Classical');
25 INSERT INTO Genre (GenreId, Name) VALUES (25, 'Opera');;
```

Requêtes SQL d'exploration et de validation

Des **requêtes SQL** ont été réalisées afin de vérifier la cohérence des données et le bon fonctionnement de la base relationnelle.
Ces requêtes ont notamment permis de compter le nombre d'enregistrements, de vérifier les relations entre les tables et de réaliser des premières analyses simples sur les données.

Exemple de requêtes :

'1. Quel pays a le plus grand nombre de clients ?'

```
SELECT Country,TotalClients
FROM (SELECT customer.Country,
COUNT(*) AS TotalClients
FROM Customer
GROUP BY customer.Country
ORDER BY TotalClients DESC
)
WHERE ROWNUM = 1;
```

Plan d'explication | Résultat de requête

Toutes les lignes extraites : 1 en 0,281 secondes

	COUNTRY	TOTALCLIENTS
1	USA	13

4. Quel genre de musique est le plus représenté ?

```
SELECT distinct genre.name,
count(track.genreid) as GenreNumero1
FROM TRACK JOIN GENRE ON
GENRE.Genreid = Track.Genreid
GROUP BY genre.name
FETCH FIRST 1 ROW ONLY
```

	NAME	GENRENUMERO1
1	Rock	1297

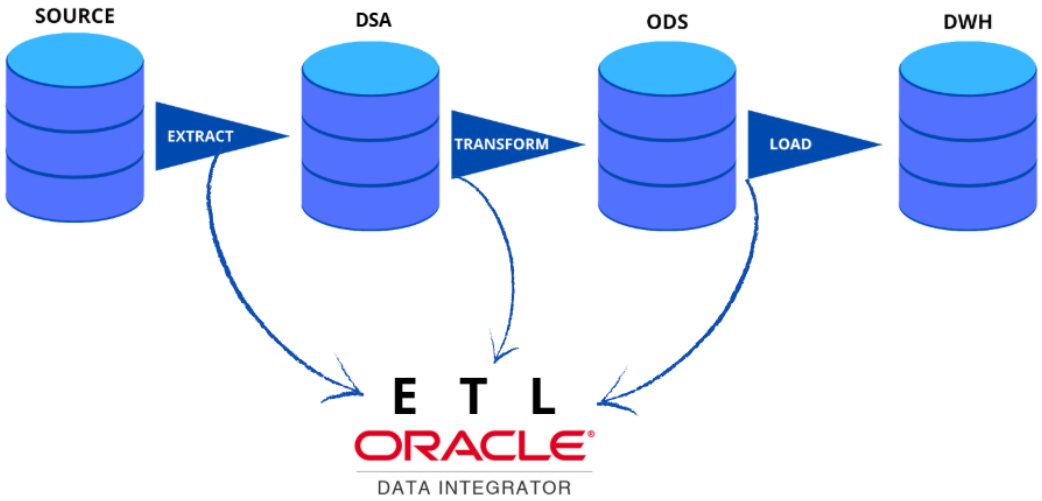
4. Mise en place de l'architecture décisionnelle

Présentation de l'architecture globale

Afin de répondre aux besoins d'analyse décisionnelle, une **architecture décisionnelle en plusieurs couches** a été mise en place.
Cette architecture permet de structurer le flux de données depuis la base opérationnelle jusqu'au Data Warehouse, en garantissant la **qualité**, la **cohérence** et la **traçabilité** des données.

L'architecture repose sur trois niveaux principaux :

- le **DSA (Data Staging Area)**,
- l'**ODS (Operational Data Store)**,
- le **Data Warehouse (DWH)**.



Rôle du DSA, de l'ODS et du Data Warehouse

DSA – Data Staging Area
Le DSA constitue la **zone de réception des données brutes** issues de la base relationnelle.
Il permet de stocker temporairement les données telles qu'elles sont extraites, sans transformation majeure, afin de sécuriser les phases de chargement et de faciliter les contrôles.

ODS – Operational Data Store
L'ODS correspond à une **zone de données nettoyées et harmonisées**.
Les données issues du DSA y sont structurées, dédupliquées et mises en cohérence, tout en restant proches du modèle opérationnel.

On y importe les données du DSA sans clé primaire et clé étrangère et sans supprimer les données précédentes, en ajoutant une colonne date d'intégration sur chacune des tables.

Data Warehouse (DWH)
Le Data Warehouse est la **base décisionnelle finale**, organisée selon un **modèle en étoile**.
Il est optimisé pour l'analyse, le calcul d'indicateurs et l'aide à la décision.
Les données y sont historisées et structurées autour de tables de faits et de dimensions.

---

5. Modélisation décisionnelle

Principe général du modèle

Le Data Warehouse a été conçu selon une **architecture en étoile**, adaptée à l'analyse décisionnelle.
Ce modèle repose sur :

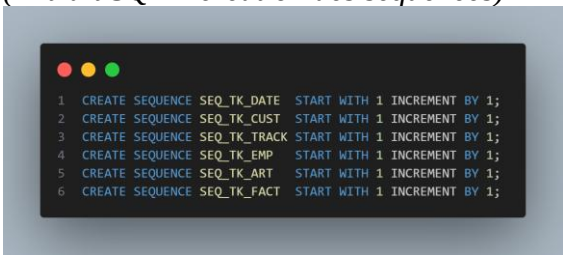
- des **tables de dimensions** décrivant les axes d'analyse,
- une **table de faits centrale** stockant les mesures quantitatives,
- l'utilisation de **clés techniques (TK)** indépendantes des clés métiers,
- la conservation des **identifiants d'origine** à des fins de traçabilité.

Ce choix garantit de **bonnes performances analytiques**, une **structure claire** et une **évolutivité** du modèle.

Génération des clés techniques

Des **séquences Oracle** ont été mises en place pour générer les clés techniques des dimensions et de la table de faits.
Ces clés techniques assurent l'unicité des enregistrements et évitent toute dépendance aux clés issues des systèmes sources.

(Extrait SQL – création des séquences)



Tables de dimensions

Dimension Date – DIM_DATE

La dimension **Date** est indispensable à toute analyse décisionnelle.
Elle permet d'analyser les ventes selon différents niveaux temporels :

- jour,
- semaine,
- mois,
- trimestre,
- année.

La table a été pré-remplie sur une plage de dates étendue afin de couvrir l'ensemble des besoins analytiques futurs.

Dimension Client – DIM_CUSTOMER

La dimension **Client** regroupe les informations permettant d'analyser les ventes par client et par zone géographique.
Elle contient une **clé technique unique** ainsi que les attributs métiers issus de la base source.

Une colonne `INTEGRATION_DATE` permet de tracer la date d'intégration des données dans le Data Warehouse.

Dimension Artiste – DIM_ARTIST

La dimension **Artiste** permet d'analyser les ventes selon les artistes.
Les attributs métiers sont conservés, et certains champs supplémentaires sont prévus afin de faciliter de futures enrichissements (nationalité, genre principal, statut).

Dimension Employé – DIM_EMPLOYEE

La dimension **Employé** permet d'analyser les ventes par employé ou responsable commercial.
Elle contient les informations professionnelles essentielles ainsi qu'une date d'intégration.

Dimension Produit – DIM_TRACK

La dimension **Track** regroupe les informations liées aux produits vendus (pistes musicales).
Elle centralise des informations provenant de plusieurs tables de la base source (album, genre, type de média), ce qui simplifie les analyses décisionnelles.

Table de faits

Table FACT_SALES

La table **FACT_SALES** constitue le cœur du Data Warehouse.
Elle enregistre les événements de vente et contient :

- les **mesures** :
  - quantité vendue,
  - prix unitaire,
  - montant total,
- les **clés étrangères** vers les dimensions,
- les identifiants d'origine (`InvoiceId`, `InvoiceLineId`) pour assurer la traçabilité,
- une date d'intégration des données.

Les relations avec les dimensions sont assurées via les **clés techniques**, garantissant la cohérence du modèle.

Conclusion sur le modèle décisionnel

Le modèle décisionnel mis en place respecte les bonnes pratiques de conception d'un Data Warehouse :

- séparation claire faits / dimensions,
- utilisation de clés techniques,
- conservation des identifiants métiers,
- intégration d'une dimension temporelle complète,
- traçabilité des données via les dates d'intégration.

Ce modèle constitue une base solide pour l'alimentation via ODI et pour la mise en œuvre d'analyses décisionnelles avancées.

6. Mise en œuvre de l'ETL avec Oracle Data Integrator

Présentation d'Oracle Data Integrator (ODI)

Oracle Data Integrator (ODI) est un outil d'intégration de données permettant de gérer les processus d'extraction, de transformation et de chargement des données au sein d'un système décisionnel. ODI repose sur une approche **ELT (Extract, Load, Transform)**, où les transformations sont exécutées directement par le moteur de la base de données cible, ce qui permet d'optimiser les performances et de tirer pleinement parti des capacités du SGBD Oracle.

Dans le cadre de cette SAE, ODI a été utilisé pour orchestrer l'ensemble des flux de données entre les différentes couches de l'architecture décisionnelle. Il permet de modéliser les traitements sous forme de **mappings**, de les organiser dans des **packages**, puis de les déployer sous forme de **scénarios** exécutables. Cette approche garantit la fiabilité, la traçabilité et la reproductibilité des chargements de données.

Configuration des référentiels et des connexions

La première étape de la mise en œuvre d'ODI a consisté à créer et configurer les **référentiels ODI**, indispensables au fonctionnement de l'outil. Un **référentiel maître** a été mis en place afin de stocker les informations techniques telles que la topologie, les connexions aux bases de données, les agents et les contextes. Un **référentiel de travail** a ensuite été associé pour gérer les objets de développement, notamment les modèles, les mappings, les packages et les scénarios.

Les différentes bases de données utilisées dans le projet (base source, DSA, ODS et Data Warehouse) ont été déclarées comme **serveurs de données Oracle** dans l'onglet *Topology*. Pour chaque serveur, les paramètres de connexion JDBC (URL, utilisateur, mot de passe) ont été définis et associés à un **contexte d'exécution**. Un **agent ODI local** a été configuré afin de permettre l'exécution des traitements et le test des connexions.

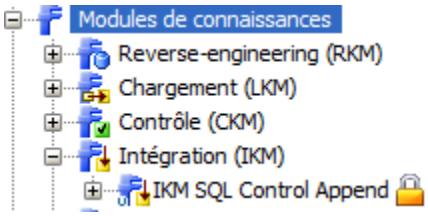
Lors de la conception des mappings, le choix de l'**IKM SQL Control Append** s'est révélé essentiel. Cet IKM permet de charger les données dans les tables cibles en

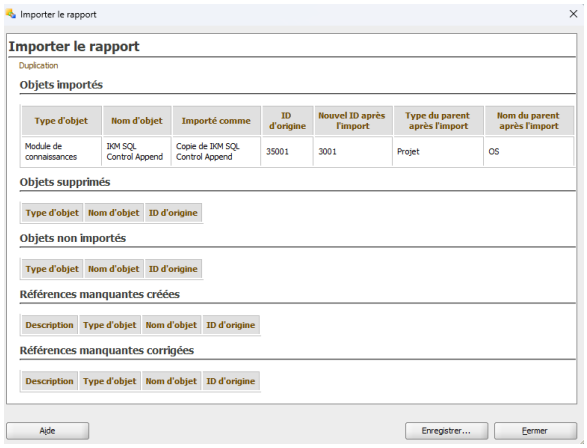
réalisant les **contrôles d'intégrité directement via des requêtes SQL**. Il assure ainsi la validation des données avant leur insertion, tout en conservant de bonnes performances grâce à l'approche ELT d'ODI.

L'utilisation de l'IKM SQL Control Append garantit :

- Le respect des règles de cohérence définies sur les tables cibles,
- La détection des éventuelles anomalies lors du chargement,
- Une meilleure traçabilité des erreurs,
- Une intégration sécurisée des données dans les différentes couches (DSA, ODS, Data Warehouse).

Cette configuration permet de fiabiliser l'ensemble des flux d'intégration et d'assurer que les données chargées sont conformes aux attentes du système décisionnel.





7. Alimentation des données

Chargement des données dans le DSA

La première étape du processus d'alimentation consiste à charger les données issues de la base relationnelle source vers le **DSA (Data Staging Area)**.
Le DSA joue le rôle de zone de transit permettant de réceptionner les données extraites sans transformation majeure, tout en sécurisant les flux et en facilitant les contrôles initiaux.

Les chargements ont été réalisés à l'aide de **mappings ODI** simples, dans lesquels chaque table source est reliée à sa table cible correspondante dans le DSA. Les attributs sont mappés de manière directe, colonne à colonne, sans logique métier complexe. Cette approche permet de conserver les données telles qu'elles sont stockées dans la base source.

Pour chaque table source, un mapping ODI a été créé :

Chaque mapping réalise les opérations suivantes :

1. Lire les données dans la base source Chinook,

2. Les transférer sans transformation vers la table correspondante dans DSA,

Afin d'accélérer les chargements et d'éviter les erreurs de CKM inutiles, nous avons désactivé :

• FLOW_CONTROL = False

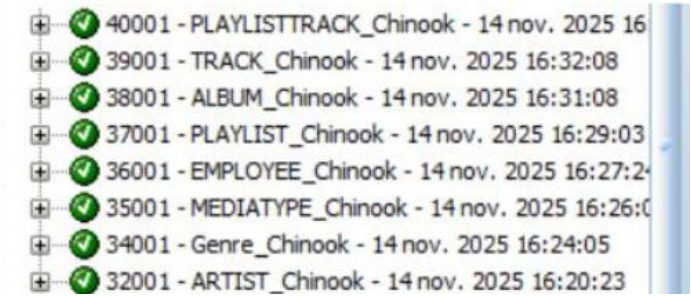
Afin d'éviter les doublons lors des rechargements, les tables du DSA peuvent être vidées avant insertion, soit par l'utilisation d'une option **TRUNCATE** dans l'IKM, soit par une commande préalable de nettoyage. Les traitements sont exécutés via l'agent ODI, garantissant une exécution contrôlée et traçable.

L'ordre d'exécution est également important pour éviter les erreurs du type « ***ORA-02291: violation de contrainte FK – clé parent introuvable*** » Ce type de problème est assez courant lors d'une intégration de données et nous avons déjà été confronté à celui-ci dans un autre contexte, sa résolution a donc été rapide et naturelle.

On suit donc cet ordre pour l'exécution :

1. ARTIST
2. ALBUM (FK → Artist)
3. MEDIATYPE
4. GENRE
5. TRACK (FK → Album, MediaType, Genre)
6. PLAYLIST
7. PLAYLISTTRACK (FK → Playlist + Track)
8. EMPLOYEE
9. CUSTOMER (FK → Employee)
10. INVOICE (FK → Customer)
11. INVOICELINE (FK → Invoice + Track)

Après configuration, chaque mapping a été exécuté depuis l'onglet Exécuter d'ODI.Les sessions ont été vérifiées dans le moniteur d'exécution, et les flèches vertes ont confirmé la réussite des chargements.



A noter qu'un package aurait également pu être réalisé pour regrouper les interfaces et les exécuter en une seule fois

---

Alimentation de l'ODS

Une fois les données chargées dans le DSA, la deuxième étape consiste à alimenter l'**ODS (Operational Data Store)**.
L'ODS constitue une zone dans la quel on importe les données du DSA sans clés primaire et étrangères et sans supprimer les données précédentes, On ajoute egalement une colonne date d'intégration sur chaque une des tables.

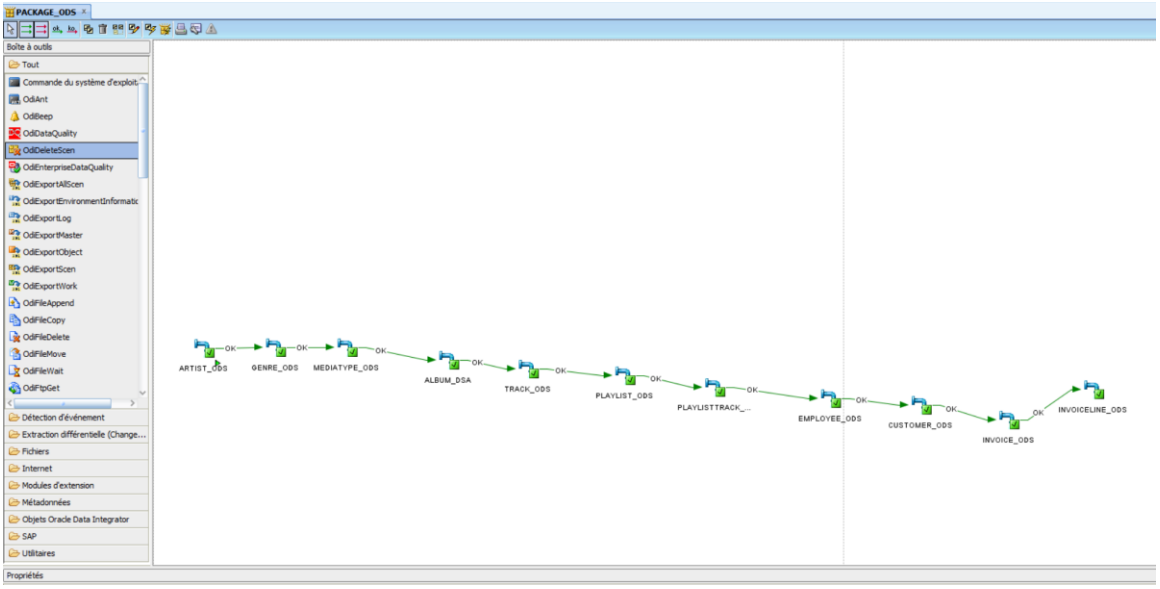
```
INTEGRATION_DATE TIMESTAMP DEFAULT SYSTIMESTAMP
```

Cette colonne enregistre **la date et l'heure d'arrivée de chaque ligne**, ce qui permet d'historiser les chargements.

Les flux d'alimentation du DSA vers l'ODS sont également réalisés à l'aide de mappings ODI. Contrairement au DSA, cette étape peut inclure des contrôles supplémentaires, tels que la gestion des doublons, la vérification des clés ou la normalisation de certains attributs. L'utilisation de l'**IKM SQL Control Append** permet de valider les règles d'intégrité avant l'insertion des données dans les tables de l'ODS.

Dans Oracle Data Integrator, nous avons :

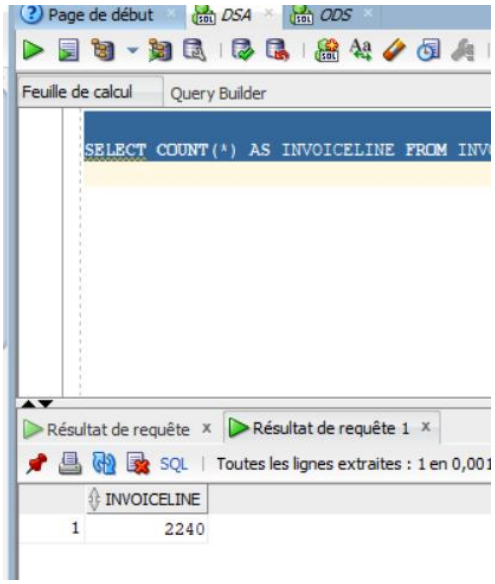
- Déclaré un nouveau serveur de données pointant vers le schéma `C##ODS`,
- Importé les tables via Reverse Engineering,
- Créé un mapping par table : **source = DSA**, **cible = ODS**.
- Crée un package qui exécute le tout



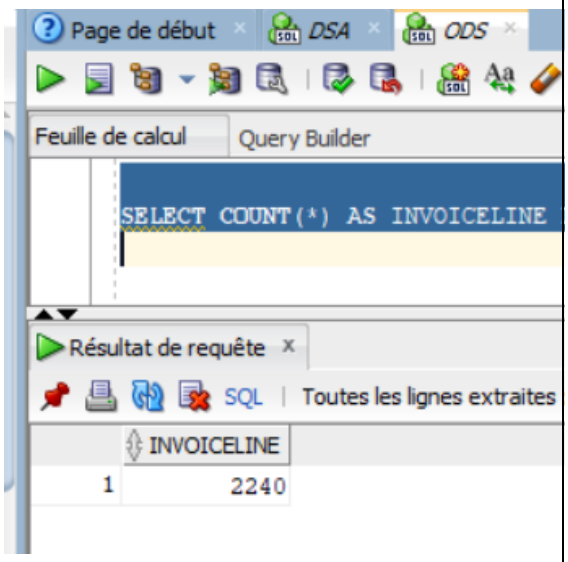
Après la bonne exécution du package, une vérification via des requetes SQL pour s'assurer de la conformité des données a été réalisée

Le nombre de lignes est identique entre DSA et ODS pour chaque table.

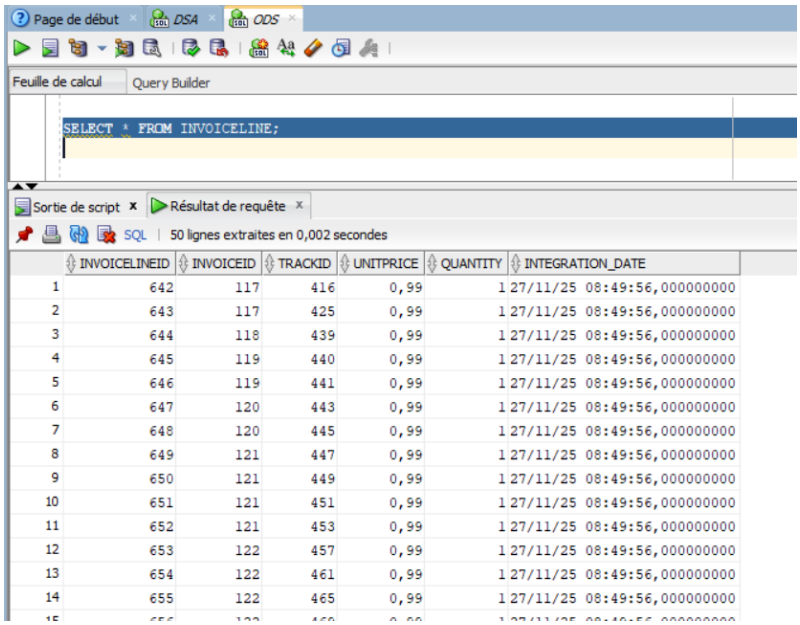
DSA



ODS



La colonne INTEGRATION_DATE contient bien des données de temps distincts pour chaque chargement. Sur chaque table :



Alimentation du Data Warehouse

L'étape finale de notre chaîne décisionnelle consiste à alimenter le Data Warehouse (DWH) à partir des données nettoyées et consolidées de l'ODS. Contrairement aux étapes précédentes (DSA et ODS) qui reproduisaient une structure proche de la source, cette étape implique une restructuration majeure des données pour respecter le modèle en étoile défini lors de la conception.

Cette alimentation est toujours orchestrée via Oracle Data Integrator (ODI) et repose sur une stratégie stricte de gestion des clés et d'ordonnancement des traitements.

1. Stratégie d'Alimentation et Gestion des Clés Techniques

Pour garantir l'indépendance du Data Warehouse vis-à-vis des systèmes sources, nous avons opté pour l'utilisation de clés de substitution (Surrogate Keys). Ces clés techniques (TK), de type numérique, sont générées automatiquement lors de l'insertion grâce aux séquences Oracle créées précédemment (ex : `SEQ_TK_CUST.NEXTVAL`).

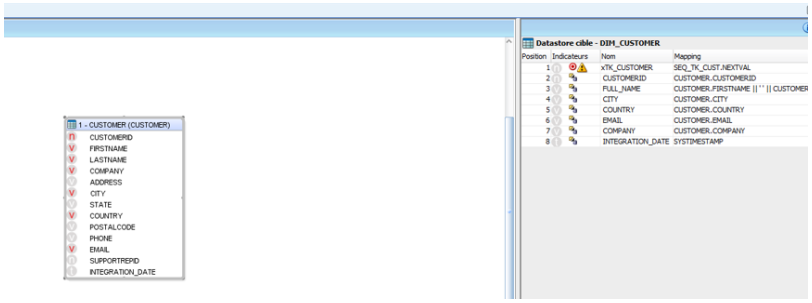
2. Alimentation des Dimensions

Les dimensions constituant les axes d'analyse, elles doivent impérativement être alimentées **avant** la table de faits. Chaque dimension fait l'objet d'un mapping dédié prenant en source les tables de l'ODS.

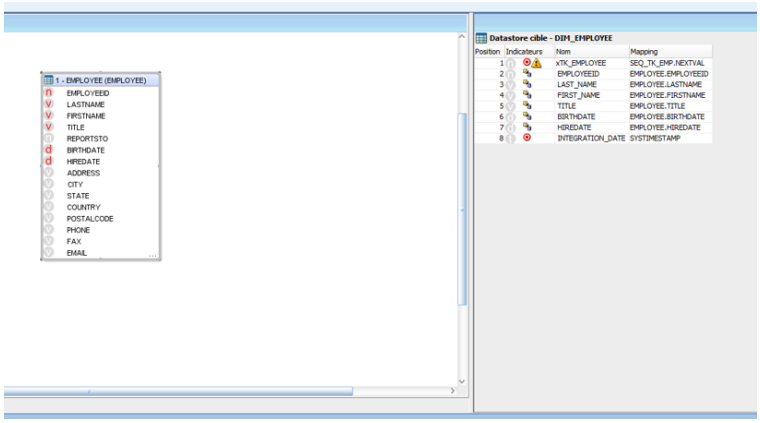
- **Dénormalisation :** Certaines dimensions, comme `DIM_TRACK`, résultent de la fusion de plusieurs tables sources (`Track`, `Album`, `Genre`, `MediaType`). Le mapping ODI réalise donc les jointures nécessaires pour regrouper tous les attributs descriptifs (Nom de l'album, Libellé du genre) au sein d'une entité unique.
- **Historisation :** Une colonne `INTEGRATION_DATE` est alimentée via la fonction `SYSTIMESTAMP` pour assurer la traçabilité des chargements.

Voici les différents mapping qui de chaque dimension :

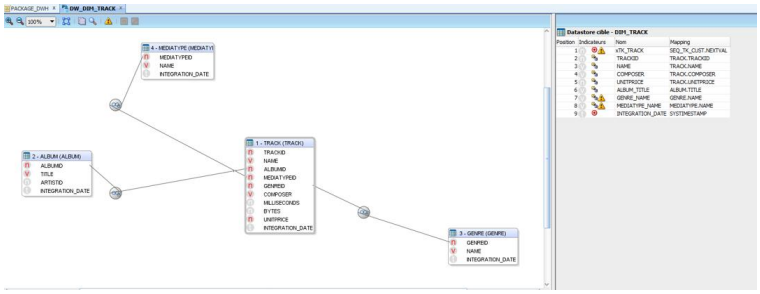
Customer :



Employee :



Track :



La table date, elle a été réalisée via le script SQL de création des tables et non via ODI

Voici le code permettant l'alimentation de celle-ci :

```sql
DECLARE
    v_start_date DATE := TO_DATE('01/01/2009', 'DD/MM/YYYY');
    v_end_date   DATE := TO_DATE('31/12/2030', 'DD/MM/YYYY');
    v_curr_date  DATE;
BEGIN
    v_curr_date := v_start_date;
    WHILE v_curr_date <= v_end_date LOOP
        INSERT INTO DIM_DATE (
            TK_DATE, DATE_REELLE, ANNEE, TRIMESTRE, MOIS_NOM, JOUR_NOM, SEMAINE_ANNEE
        ) VALUES (
            SEQ_TK_DATE.NEXTVAL,
            v_curr_date,
            TO_NUMBER(TO_CHAR(v_curr_date, 'YYYY')),
            TO_NUMBER(TO_CHAR(v_curr_date, 'Q')),
            TO_CHAR(v_curr_date, 'Month', 'NLS_DATE_LANGUAGE=FRENCH'),
            TO_CHAR(v_curr_date, 'Day', 'NLS_DATE_LANGUAGE=FRENCH'),
            TO_NUMBER(TO_CHAR(v_curr_date, 'IW'))
        );
        v_curr_date := v_curr_date + 1;
    END LOOP;
    COMMIT;
END;
```

3. Alimentation de la Table de Faits (FACT_SALES)

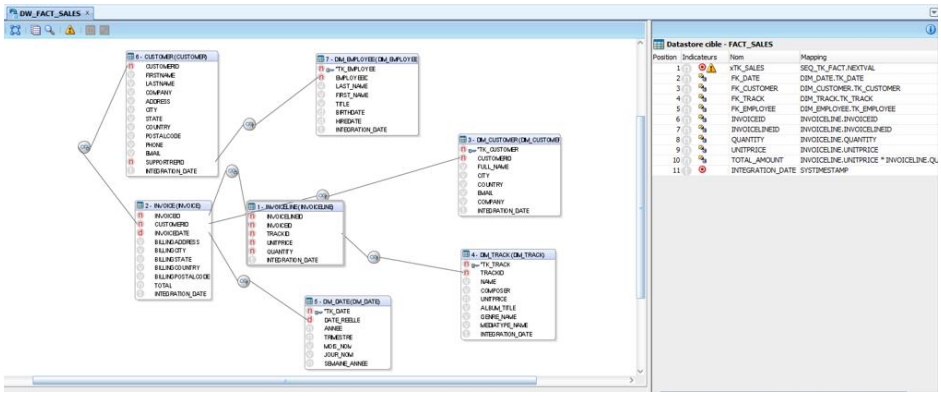
L'alimentation de la table de faits `FACT_SALES` est l'étape la plus complexe car elle nécessite de croiser les données transactionnelles (Ventes) avec les dimensions fraîchement chargées pour récupérer les clés techniques (TK).

Dans le mapping, nous avons mis en place des mécanismes de recherche (Lookups) ou de jointure pour remplacer les identifiants fonctionnels par les `TK_` correspondants :

- `FK_CUSTOMER` est récupéré en joignant la vente avec `DIM_CUSTOMER`.
- `FK_TRACK` est récupéré via `DIM_TRACK`.

Focus Technique : La jointure complexe Vendeur Un défi technique a été relevé concernant l'attribution des ventes aux employés. Dans la source, la facture n'est pas directement liée au vendeur. Pour peupler correctement la colonne `FK_EMPLOYEE`, nous avons dû reconstruire une jointure transitive dans le mapping : `ODS_INVOICE` → `ODS_CUSTOMER` → `DIM_EMPLOYEE`. Cette opération garantit qu'aucune vente n'est orpheline de vendeur (valeur `NULL`).

Enfin, les identifiants métiers d'origine (`InvoiceId`, `InvoiceLineId`) ont été conservés dans la table de faits pour permettre, si besoin, de remonter à la transaction source précise.



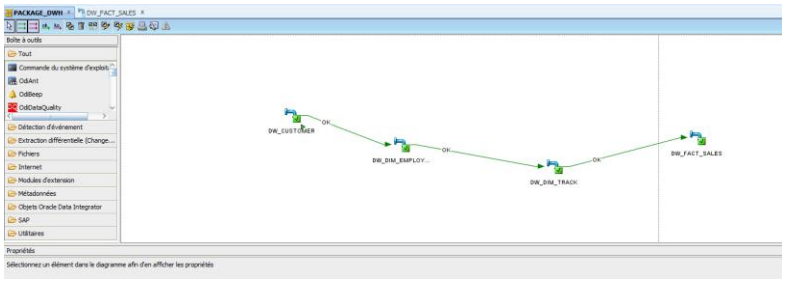
4. Orchestration et Automatisation

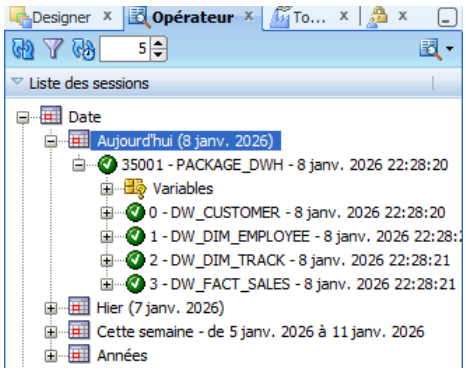
Afin d'automatiser le chargement complet et de respecter les contraintes d'intégrité référentielle (Foreign Keys), nous avons regroupé l'ensemble des interfaces dans un **Package ODI**.

Ce package orchestre l'exécution séquentielle des mappings :

1. Chargement des Dimensions (`DIM_CUSTOMER`, `DIM_EMPLOYEE`, `DIM_TRACK`...).
2. Chargement de la Table de Faits (`FACT_SALES`) en dernier.

Cette structure permet de lancer l'alimentation complète du DWH en une seule opération via un Scénario, facilitant ainsi la planification et la maintenance.





8. Contrôles et validation des données

Une fois l'alimentation du Data Warehouse terminée, une phase de recette technique a été réalisée pour garantir la fiabilité du système décisionnel. Cette étape est cruciale pour certifier que les données mises à disposition des utilisateurs correspondent fidèlement à la réalité opérationnelle.

Vérification de l'intégrité des données

L'intégrité référentielle est assurée par le modèle en étoile et les contraintes de clés étrangères définies dans le SGBD cible. Un point d'attention particulier a été porté sur la complétude des informations de vente. Lors des premiers tests, nous avons identifié que le lien entre une vente et un vendeur n'était pas direct dans la source. Grâce à la mise en place d'une jointure transitive (`Invoice` → `Customer` → `Employee`) dans l'ETL, nous avons pu valider que chaque ligne de la table de faits `FACT_SALES` est désormais correctement rattachée à un employé via la clé technique `FK_EMPLOYEE`.

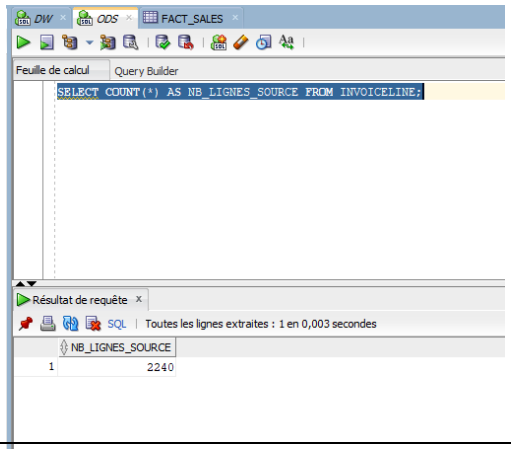
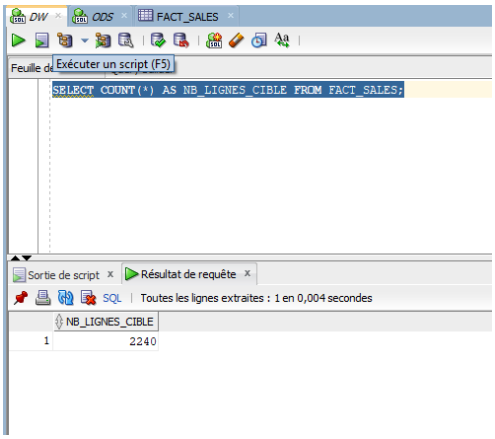
La requête de contrôle ci-dessous confirme qu'aucune vente n'est orpheline de vendeur (absence de valeurs `NULL`) :

	TK_SALES	FK_DATE	FK_CUSTOMER	FK_TRACK	FK_EMPLOYEE	INVOICEID	INVOICELINEID	QUANTITY	UNITPRICE	TOTAL_AMOUNT	INTEGRATION_DATE
1	282	5130	1	1114	4	171	921	1	0,99	0,99	08/01/26 22:28:21,632000000
2	283	5130	1	1118	4	171	922	1	0,99	0,99	08/01/26 22:28:21,632000000
3	284	5130	1	1122	4	171	923	1	0,99	0,99	08/01/26 22:28:21,632000000
4	285	5130	1	1126	4	171	924	1	0,99	0,99	08/01/26 22:28:21,632000000
5	286	5133	7	1132	5	172	925	1	0,99	0,99	08/01/26 22:28:21,632000000
6	287	5133	7	1138	5	172	926	1	0,99	0,99	08/01/26 22:28:21,632000000
7	288	5133	7	1646	5	172	927	1	0,99	0,99	08/01/26 22:28:21,632000000
8	289	5133	7	1652	5	172	928	1	0,99	0,99	08/01/26 22:28:21,632000000
9	290	5133	7	2082	5	172	929	1	0,99	0,99	08/01/26 22:28:21,632000000
10	291	5133	7	2088	5	172	930	1	0,99	0,99	08/01/26 22:28:21,632000000
11	292	5133	7	2094	5	172	931	1	0,99	0,99	08/01/26 22:28:21,632000000
12	293	5133	7	2100	5	172	932	1	0,99	0,99	08/01/26 22:28:21,632000000
13	294	5133	7	2106	5	172	933	1	0,99	0,99	08/01/26 22:28:21,632000000
14	295	5138	16	2115	5	173	934	1	0,99	0,99	08/01/26 22:28:21,632000000

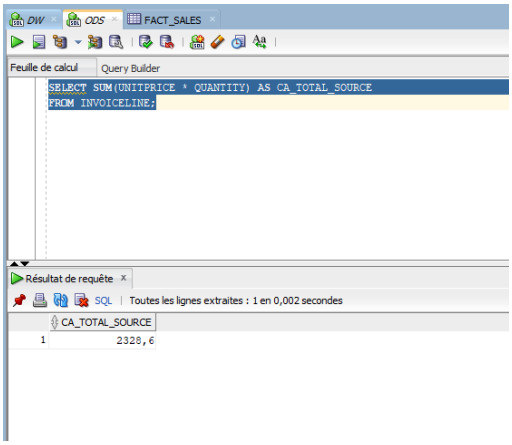
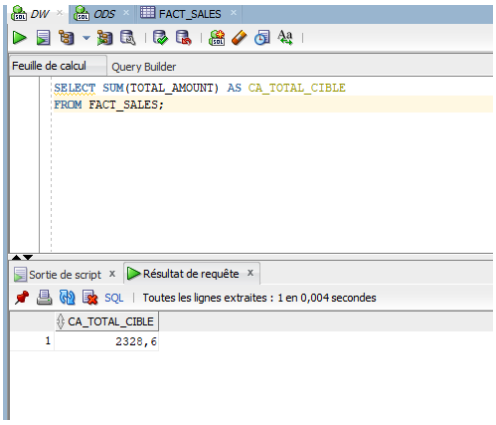
Contrôles de volumes et de cohérence

Pour valider l'exhaustivité des chargements, nous avons procédé à une "réconciliation des données" entre la source (ODS) et la cible (DWH). Deux indicateurs majeurs ont été comparés :

1. **Volumétrie (Nombre de lignes) :** Nous avons vérifié que le nombre d'enregistrements dans la table de faits correspond exactement au nombre de lignes de facturation dans l'ODS.
    - Requête ODS : `SELECT COUNT(*) FROM INVOICELINE_ODS;`
    - Requête DWH : `SELECT COUNT(*) FROM FACT_SALES;`

SOURCE	DATAWAREHOUSE
	

2. **Cohérence Financière (Montants) :** Le montant total des ventes a été recalculé des deux côtés pour s'assurer qu'aucune altération de valeur (problème d'arrondi ou de type de données) n'a eu lieu durant l'ETL. Les sommes sont strictement identiques.

SOURCE	DATAWAREHOUSE
	

9. Exploitation des données décisionnelles

L'objectif final de la SAE était de rendre les données "intelligibles" et facilement exploitables pour la prise de décision. Le modèle en étoile simplifie considérablement l'écriture des requêtes par rapport au modèle relationnel source très normalisé.

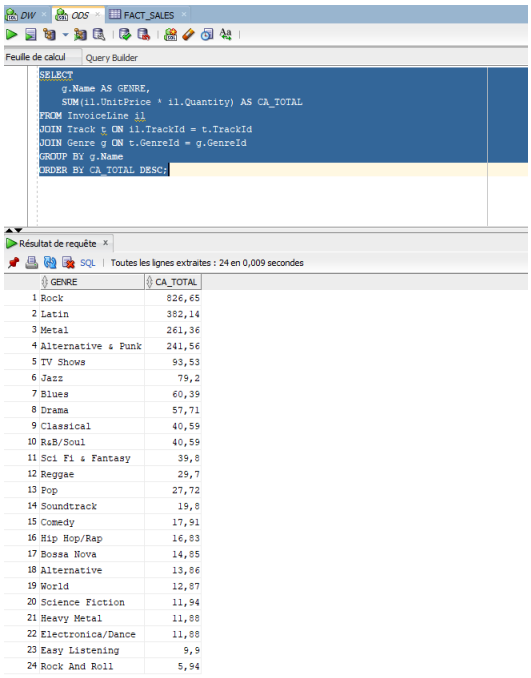
Requêtes analytiques

Grâce aux dimensions consolidées (comme `DIM_TRACK` qui regroupe Album, Genre et Média), les analyses croisées deviennent plus performantes. Nous avons pu réaliser des requêtes répondant à des questions métiers précises, telles que :

- *Quels sont les genres musicaux les plus vendus ?*
- *Quelle est la répartition des ventes par pays client ?*
- *Qui sont les meilleurs vendeurs ?*

L'exemple ci-dessous illustre l'analyse des ventes par genre musical, prouvant le bon fonctionnement de la dimension Produit :

SOURCE

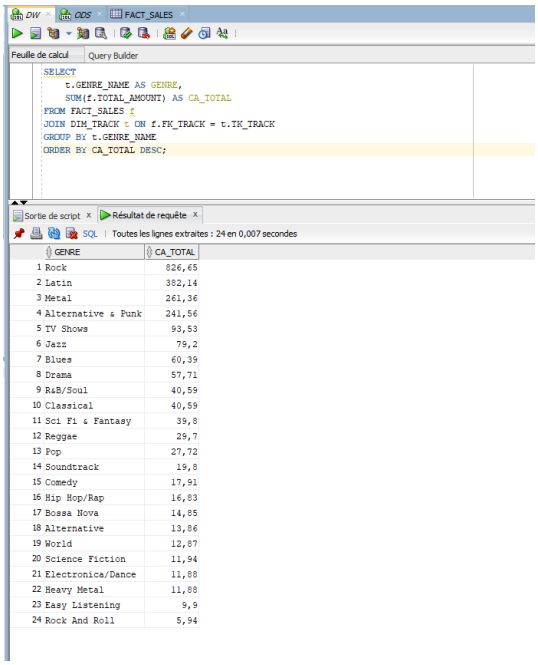


```
SELECT
    g.Name AS GENRE,
    SUM(il.UnitPrice * il.Quantity) AS CA_TOTAL
FROM InvoiceLine il
JOIN Track t ON il.TrackId = t.TrackId
JOIN Genre g ON t.GenreId = g.GenreId
GROUP BY g.Name
ORDER BY CA_TOTAL DESC;
```

Toutes les lignes extraites : 24 en 0,009 secondes

	GENRE	CA_TOTAL
1	Rock	826,65
2	Latin	382,14
3	Metal	261,36
4	Alternative & Punk	241,56
5	TV Shows	93,53
6	Jazz	79,2
7	Blues	60,39
8	Drama	57,71
9	Classical	40,59
10	R&B/Soul	40,59
11	Sci Fi & Fantasy	39,8
12	Reggae	29,7
13	Pop	27,72
14	Soundtrack	19,8
15	Comedy	17,91
16	Hip Hop/Rap	16,83
17	Bossa Nova	14,85
18	Alternative	13,86
19	World	12,87
20	Science Fiction	11,94
21	Heavy Metal	11,88
22	Electronica/Dance	11,88
23	Easy Listening	9,9
24	Rock And Roll	5,94

DATAWAREHOUSE



```
SELECT
    t.GENRE_NAME AS GENRE,
    SUM(f.TOTAL_AMOUNT) AS CA_TOTAL
FROM FACT_SALES f
JOIN DIM_TRACK t ON f.FK_TRACK = t.TK_TRACK
GROUP BY t.GENRE_NAME
ORDER BY CA_TOTAL DESC;
```

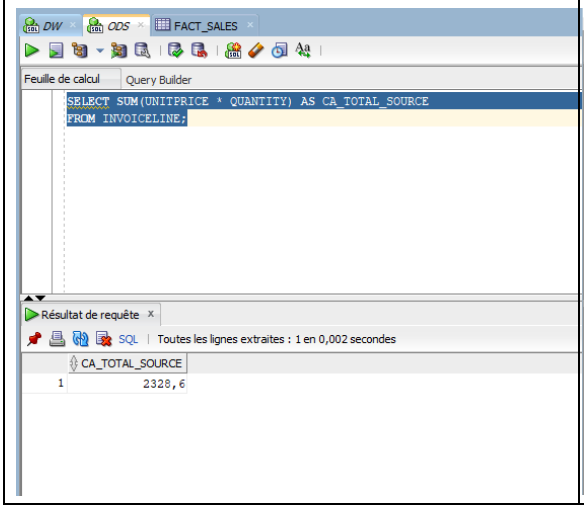
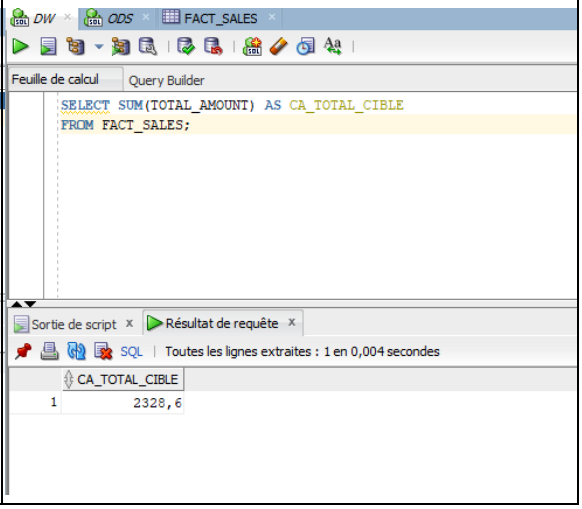
Toutes les lignes extraites : 24 en 0,007 secondes

	GENRE	CA_TOTAL
1	Rock	826,65
2	Latin	382,14
3	Metal	261,36
4	Alternative & Punk	241,56
5	TV Shows	93,53
6	Jazz	79,2
7	Blues	60,39
8	Drama	57,71
9	R&B/Soul	40,59
10	Classical	40,59
11	Sci Fi & Fantasy	39,8
12	Reggae	29,7
13	Pop	27,72
14	Soundtrack	19,8
15	Comedy	17,91
16	Hip Hop/Rap	16,83
17	Bossa Nova	14,85
18	Alternative	13,86
19	World	12,87
20	Science Fiction	11,94
21	Electronica/Dance	11,88
22	Heavy Metal	11,88
23	Easy Listening	9,9
24	Rock And Roll	5,94

Calcul des indicateurs (chiffre d'affaires)

L'indicateur clé de performance (KPI) principal de l'entreprise est le Chiffre d'Affaires (CA). Dans la table de faits `FACT_SALES`, le montant de chaque ligne est pré-calculé (`Quantity * UnitPrice`). L'analyse temporelle du CA est facilitée par la dimension `DIM_DATE`.

Nous sommes désormais capables de restituer l'évolution du chiffre d'affaires par année ou par trimestre via une simple requête d'agrégation, ce qui était complexe à réaliser sur la base transactionnelle d'origine.

SOURCE	DATAWAREHOUSE
	

Conclusion

Bilan du projet

Ce projet nous a permis de parcourir l'intégralité de la chaîne de valeur de la Business Intelligence. Partant d'une base de données relationnelle classique (Chinook), nous avons conçu et implémenté une architecture décisionnelle robuste comprenant un DSA, un ODS et un Data Warehouse. L'utilisation d'Oracle Data Integrator (ODI) a été centrale pour automatiser les flux de données. Nous avons réussi à surmonter plusieurs défis techniques, notamment la gestion des séquences Oracle pour les clés techniques et la

reconstruction de liens complexes entre les entités (Ventes/Employés). Le système final est fonctionnel : les données sont propres, historisées et structurées pour l'analyse.

Apports et limites

Apports :

- **Centralisation :** Toutes les données analytiques sont regroupées dans un modèle unique optimisé pour la lecture (Schéma en étoile).
- **Performance :** Les requêtes d'analyse sont simplifiées (moins de jointures) et plus rapides.
- **Indépendance :** Grâce aux clés de substitution (TK), le décisionnel est indépendant des changements techniques de la base de production.

Limites et perspectives :

- **Visualisation :** Le projet s'arrête à la base de données. L'étape suivante consisterait à connecter un outil de Dataviz (comme Power BI ou Tableau) pour visualiser les indicateurs graphiquement.
- **Automatisation temporelle :** Bien que le package ODI soit fonctionnel, sa planification automatique (via un ordonnanceur) n'a pas été mise en place.
- **Gestion des mises à jour :** Nous avons opté pour une insertion simple. Une gestion plus fine des changements de dimension permettrait de mieux gérer l'historique des changements d'adresse des clients ou de poste des employés.

Remerciement

Nous tenons à adresser nos sincères remerciements à **Monsieur Le Véler** pour son encadrement et sa disponibilité tout au long de cette SAE.

Ses conseils avisés, tant sur la méthodologie de gestion de projet que sur les aspects techniques liés à Oracle Data Integrator et à la modélisation décisionnelle, nous ont été d'une aide précieuse.

Son expertise et sa pédagogie nous ont permis de surmonter les difficultés rencontrées et de mener à bien la construction de ce Data Warehouse.