

I. Conception & Implémentation de la base de donnée comptoir.

1. Création de la base comptoir
2. Structuration de la base
3. Alimentation de la base par le fichier comptoire.sql

II. Création d'un entrepôt de donnée pour Suivre les Ventes.

1. Extraction des donnée à partir des sources de données (Tables)
2. Transformation
3. Chargement au niveau de la table Ventes
4. Agrégation des donnée de la table ventre selon un critère
5. Ordonnancer le résultat et afficher le sur le console
6. Gestion des erreurs

I. Conception & Implémentation de la base de donnée comptoir.

1. La Création & l'octroi des privilèges nécessaires d'un nouvel utilisateur appelé comptoir

Commencer par créer un nouvel utilisateur

Create user comptoir identified by comptoir;

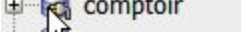
Attribuer à comptoir les droit d'un dba pour administrer le schéma comptoir

Grant dba to comptoir

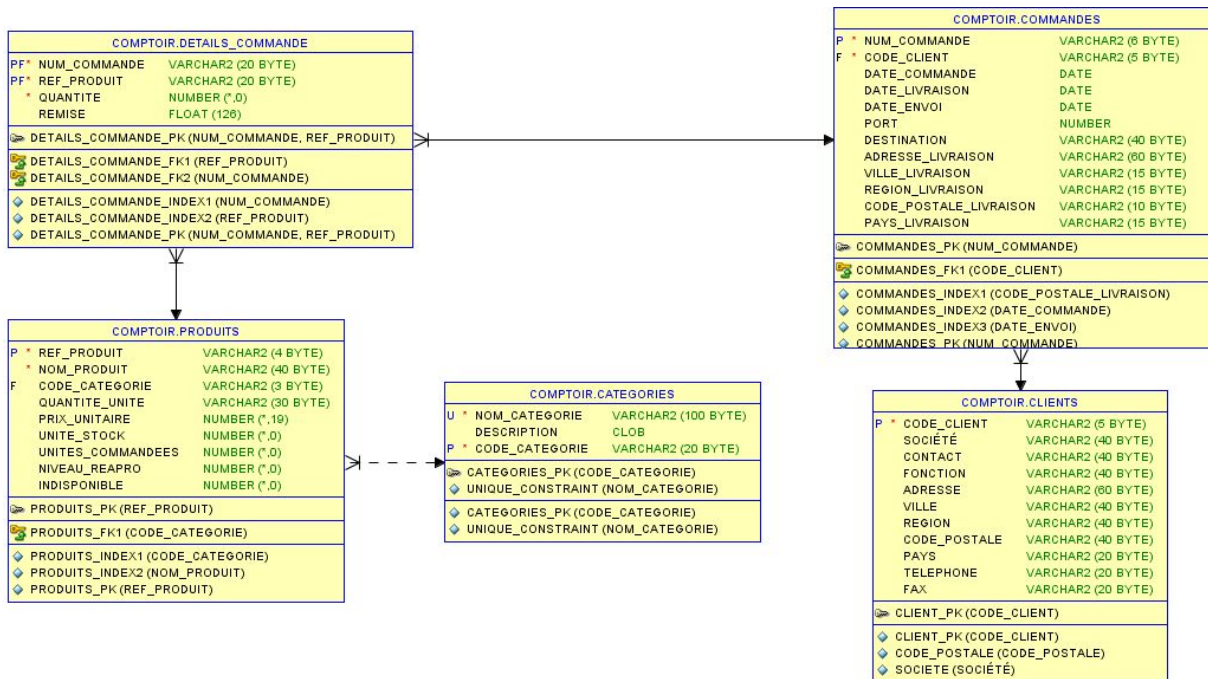
Connectez vous en tant que comptoir pour test

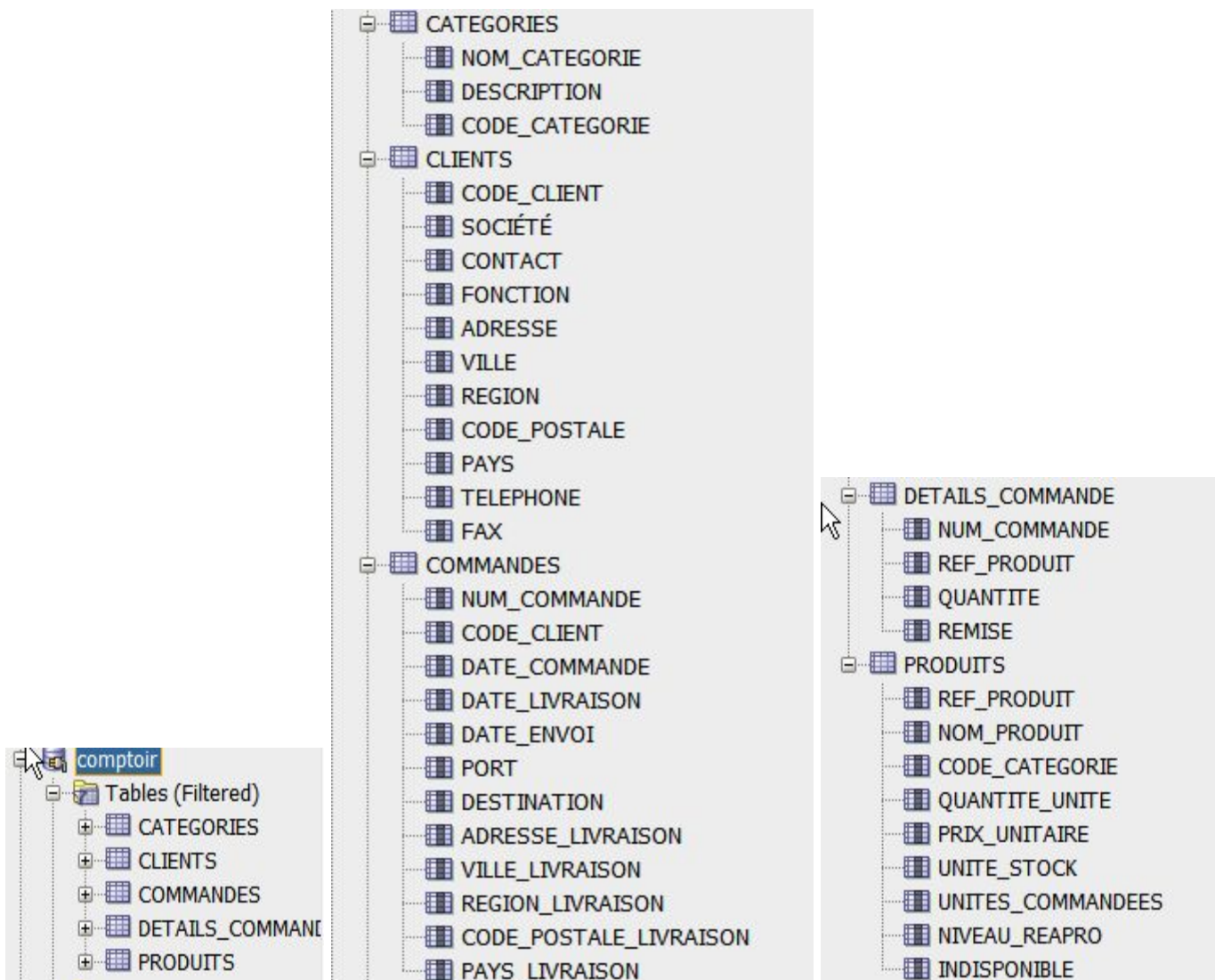
Connect comptoir/comptoir

2. Implémenter le modèle conceptuel de la base de donnée comptoir

- Connectez vous sur sql developer en tant que comptoir .
- Implémenter le schéma suivant :(Tables>>new Table)

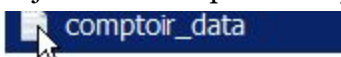
Respecter exactement la structure de la base de donnée, les noms des champs leurs types, .. afin que l'alimentation automatique ne présentera aucun problème.





3. Alimentation de la base de donnée

Toujours sur sql developer, alimenter la base de donnée en exécutant le comptoir_data.sql



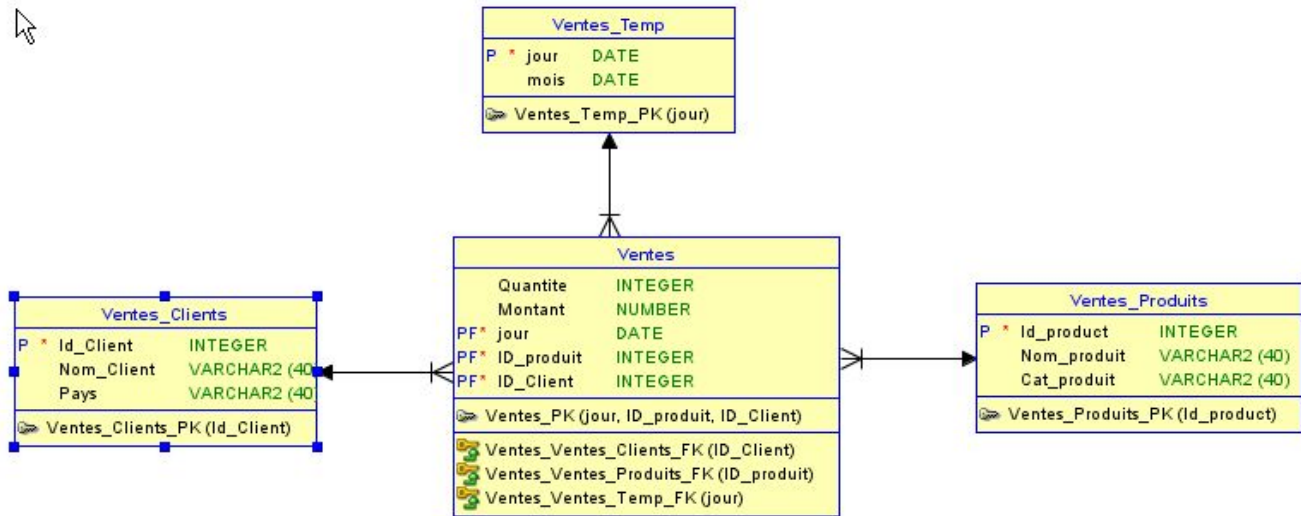
Il est préférable d'alimenter les table de la base une par une pour réagir efficacement s'il y a des problèmes.


II. Création d'un entrepôt de donnée pour analyser les Ventes.

Vous pouvez omettre cette phase si vous utilisez Talend comme ETL parce que ce dernier crée automatiquement à travers le job les tables de destination.

On crée un entrepôt de donnée, pour analyser les ventes de cette société, l'entrepôt ici représente la structure qui sera alimenté par des jobs talend.

Reproduire via Oracle sql modeler le modèle suivant:



En utilisant le ruban horizontal de oracle data modeler, créer une table , renseigner les # champs de chaque table, via le menu vertical de création des table préciser les clés primaires et celles étrangères.

Columns

- Primary Key
- Unique Constraints
- Indexes
- Table Level Constraints
- Existence Dependencies
- Foreign Keys
- Nested Columns
- Valid Time Dimensions
- Materialized Query
- Volume Properties
- Spatial Properties
- Column Groups
- Comments
- Comments in RDBMS
- Notes

Columns

Details Overview Security UDP

Columns:

Name	Data type
1 Id_product	Integer
2 Nom_produit	String (40)
3 Cat_produit	String (40)

Column Properties

Name: Id_product

Data Type: Domain Logical Distinct
 Structured Collection

Source Type: Integer Preferred

PK FK Mandatory Deprecated

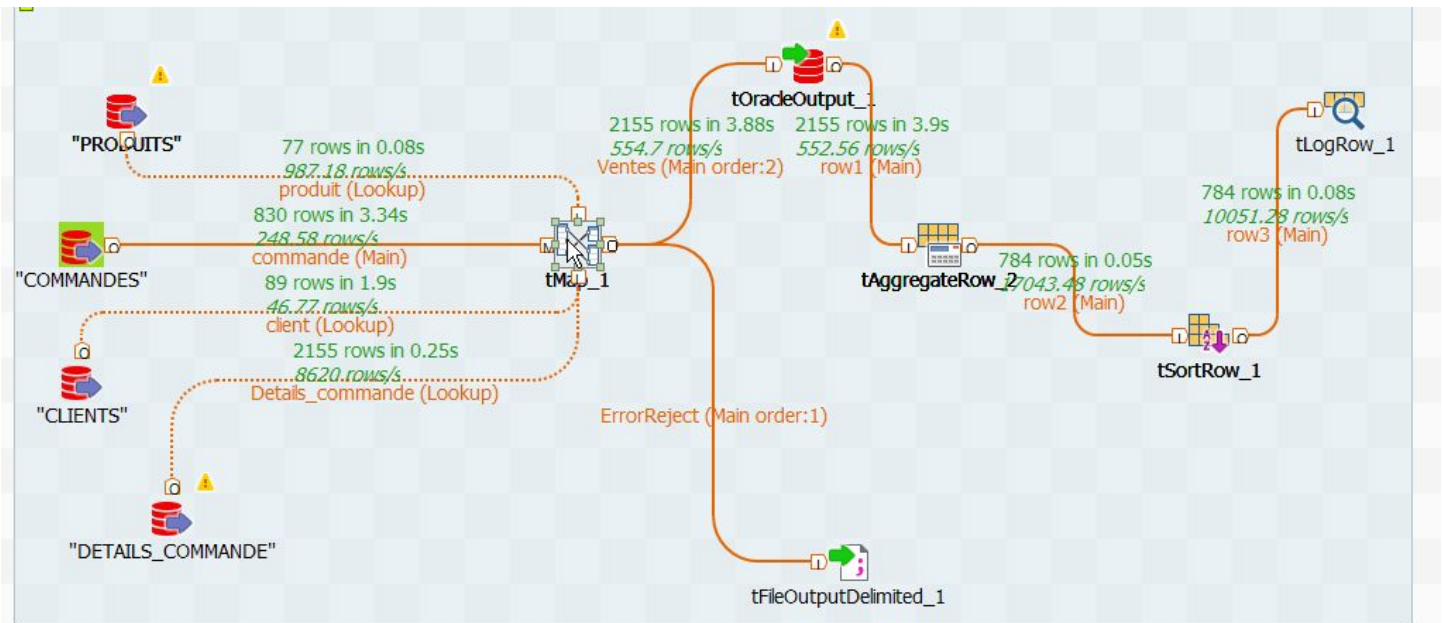
Générer le DLL correspond pour créer l'entrepôt, un aperçu est donnée ci dessous.

```

CREATE TABLE Ventes
(
    Quantite    INTEGER ,
    Montant     NUMBER ,
    jour        DATE NOT NULL ,
    ID_produit  INTEGER NOT NULL ,
    ID_Client   INTEGER NOT NULL
) ;
ALTER TABLE Ventes ADD CONSTRAINT Ventes_PK PRIMARY KEY ( jour, ID_produit, ID_Client ) ;
CREATE TABLE Ventes_Clients
  
```

Le Job Ventes

Vous réaliser le job ventes suivant :



Contenant les sources Produit, **Commande(Principal=Main)**, Client, Détails_Commande.
 Ensuite via le composant tMap on va generer un flux représentant les ventes de la structure suivante

Ventes	
Expression	Column
commande.CODE_CLIENT	CODE_CLIENT
produit.REF_PRODUIT	REF_PRODUIT
client.SOCIETE	SOCIETE
Details_commande.QUANTITE	QUANTITE
produit.NOM_PRODUIT	NOM_PRODUIT
Var.Montant	Montant
Var.Mois_COMMANDE	Mois_COMMANDE
commande.DATE_COMMANDE	DATE_COMMANDE

Ce flux contient les 2 variables de transformation *var.Montant* et *Var.Mois_Commande*

Var		
Expression	Type	Variable
Details_commande...	double	<input type="checkbox"/> Montant
TalendDate.formatD...	String	<input type="checkbox"/> Mois_COM...

$Montant = Details_commande.QUANTITE * produit.PRIX_UNITAIRE$ represente le montant de la commande
 $Mois_Commande = TalendDate.formatDateInUTC("MM", commande.DATE_COMMANDE)$ extrait le mois à partir de la date des commandes.

Gestion des erreur :

S'il y a un problème lors de l'extraction des enregistrement à partir des source un fichier délimité prévu recueillie les enregistrement non extraits (nécessitant un traitement spécial).

Le fichier ErrorReject aura la structure suivante

ErrorReject	
Expression	Column
	errorMessage
	errorStackTrace
commande.NUM_COMMANDE	NUM_COMMANDE
commande.CODE_CLIENT	CODE_CLIENT
commande.DATE_COMMANDE	DATE_COMMANDE

Joindre les 4 tables de source en précisant les 2 options suivantes

Lookup Model	Load once
Match Model	All matches
Join Model	Inner Join
Store temp data	false

Deux sortie du flux sont prévues : une relative à Error reject, et l'autre relative à une création et alimentation de la table Vente dans la base comptoir.

Le composant tOracleOutput est configuré comme suit (utiliser la métadata)

Host	"localhost"	Port	"1521"
Database	"xe"	Oracle schema	"COMPTOIR"
Jusername	"comptoir"	Password	*****
Table	"Ventes"		
Action on table	Drop table if exists and create	Action on data	Insert or update
Schema	Built-In	Edit schema	Sync columns

NB : la table sera créé si elle n'existe pas (mention d'omission antérieure) sinon, si elle existe, elle sera détruite et recréer à chaque lancement de ce job.

Agrégation des ventes

Utiliser le composant Talend **tAggregateRow** pour agréger les donnée afin d'avoir des informations qui aident à la décision.

On va agréger les informations stockées au niveau de la table Ventes par rapport à la somme des montants des commandes on les groupant par mois et par produit.

Le composant Talend taggregateRowest paramétré comme suit :

Group by

Output column	Input column position
Mois_COMMANDE	Mois_COMMANDE
NOM_PRODUIT	NOM_PRODUIT



Operations

Output column	Function	Input column position	<input checked="" type="checkbox"/> Ignore null
Montant	sum	Montant	<input checked="" type="checkbox"/>

L'ordonnement.

Finalement ordonner la sortie de l'agrégation selon le mois et le produit via le composant Map **tSortRow** qui est configuré comme suit:

Schema column	sort num or alpha?	Order asc or desc?
Mois_COMMANDE	alpha	asc
NOM_PRODUIT	alpha	asc

La sortie sera redirigée vers le console Talend.

784 enregistrements sont affichés de la forme **(mois, produit, chiffre d'affaire par produit)**

```
01|Valkoinen suklaa|1215.0
01|Vegie-spread|10074.0
01|Wimmers gute Semmelknödel|16600.0
01|Zaanse koeken|1410.0
02|Alice Mutton|1560.0
02|Aniseed Syrup|6450.0
02|Boston Crab Meat|10856.0
02|Camembert Pierrot|36720.0
02|Carnarvon Tigers|10920.0
```