

# Rapport de Projet : Conception et Implémentation d'un Entrepôt de Données pour MacroBus



**El Hadji Ablaye Galoup DIOP**

# Contexte

**MacroBus**, une entreprise spécialisée dans la vente de véhicules sur plusieurs marchés, souhaite optimiser ses performances commerciales et faciliter la prise de décision à travers un système dédié.

Pour ce faire nous devons mettre en place un entrepôt de données pour servir de source aux tableaux de bord décisionnels afin de piloter efficacement l'activité commerciale et analyser les commandes.

Parties Prenantes : PDG, Gestionnaire de base de données, Data Engineering, Data Analyst.

**Objectif Général** : Conception et implémentation d'un entrepôt de données permettant de répondre aux besoins décisionnels et stratégiques.

# TÂCHES CLÉS À EXÉCUTER

**01**

Restauration  
de la source  
de données

**02**

Modélisation  
en étoile

**03**

Création de  
l'entrepôt de  
données

**04**

Chargement  
des données

**05**

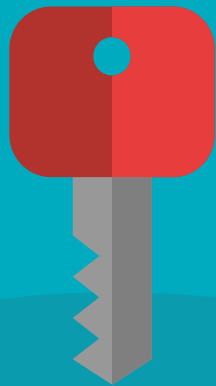
Validation  
fonctionnelle

**06**

Conception  
de tableaux  
de bord

01

# Restauration de la source de données



# Restauration de la source de données

## Objectif:

Restauration de la base de données fournie par le gestionnaire, nettoyage et préparation des données

Remarque: logiciel de gestion et d'administration de bases de données utilisé MYSQL WORKBENCH



# Les étapes à suivre pour restaurer la base de donnée source

orderNumber	productCode	quantityOrdered	priceEach	orderLineNumber
10100	S18_1749	30	136	3
10100	S18_2248	50	55	2
10100	S18_4409	22	75	4
10100	S24_3969	49	35	1
10101	S18_2325	25	108	4
10101	S18_2795	26	167	1
10101	S24_1937	45	33	3
10101	S24_2022	46	44	2
10102	S18_1342	39	96	2
10102	S18_1367	41	43	1
10103	S10_1949	26	214	11
10103	S10_4962	42	120	4
10103	S12_1666	27	122	8
10103	S18_1097	35	95	10
10103	S18_2432	22	58	2
10103	S18_2949	27	92	12
10103	S18_2957	35	62	14
10103	S18_3136	25	87	13
10103	S18_3320	46	86	16

The screenshot shows the SQL Workbench interface. On the left, the 'SCHEMAS' pane displays a tree view of the database structure, including 'Stored Procedures', 'Functions', and 'Tables'. The 'Tables' section is expanded, showing 'customers', 'employees', 'orderdetails', 'orders', 'products', 'productsline', and 'Views'. The 'orderdetails' table is selected. Below the tree, the 'Table: orderdetails' is listed with its columns: 'orderNumber' (int), 'productCode' (text), 'quantityOrdered' (int), 'priceEach' (int), 'orderLineNumber' (int), and 'Montant' (int).

The main editor displays the SQL code for creating the 'orderdetails' table:

```
CREATE TABLE `orderdetails` (
  `orderNumber` int DEFAULT NULL,
  `productCode` text,
  `quantityOrdered` int DEFAULT NULL,
  `priceEach` int DEFAULT NULL,
  `orderLineNumber` int DEFAULT NULL,
  `Montant` int DEFAULT NULL
)
```

Below the editor, the 'Result Grid' shows the data loaded into the table. The grid has columns for 'orderNumber', 'productCode', 'quantityOrdered', 'priceEach', and 'orderLineNumber'. The data is as follows:

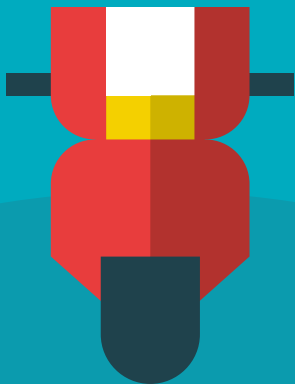
orderNumber	productCode	quantityOrdered	priceEach	orderLineNumber
10100	S18_1749	30	136	3
10100	S18_2248	50	55	2
10100	S18_4409	22	75	4
10100	S24_3969	49	35	1
10101	S18_2325	25	108	4
10101	S18_2795	26	167	1
10101	S24_1937	45	33	3
10101	S24_2022	46	44	2
10102	S18_1342	39	96	2
10102	S18_1367	41	43	1
10103	S10_1949	26	214	11
10103	S10_4962	42	120	4
10103	S12_1666	27	122	8
10103	S18_1097	35	95	10

Télécharger la base de donnée source, la transformer en fichier C.S.V pour faciliter le chargement

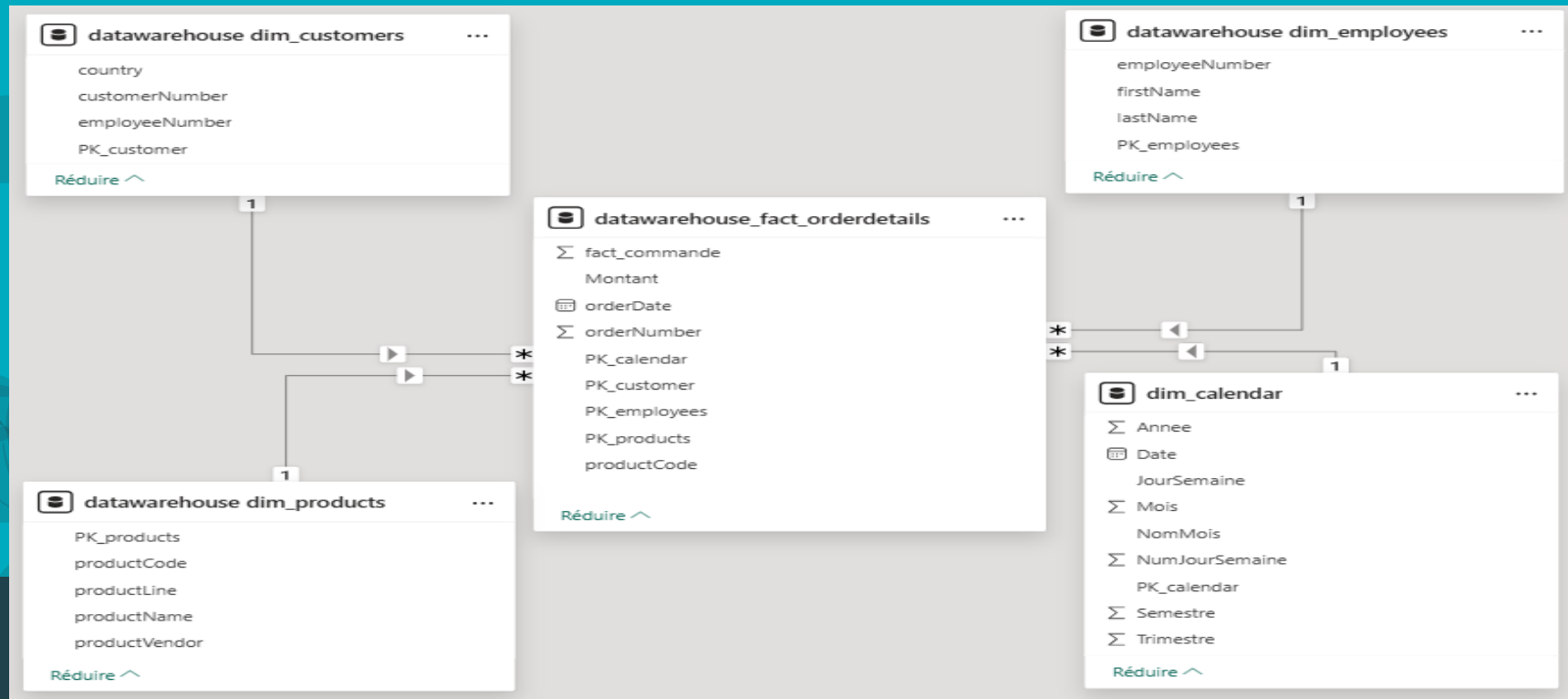
Creation et chargement de la base de donnée avec SQL workbench

02

## Modélisation en étoile pour structurer les données de manière optimale



# Modélisation en étoile : Identification des dimensions (client, produits, commerciaux, calendrier) et de la table des faits (commandes)





03

# Création de l'entrepôt de données



# Script sql de création de l'entrepôt de données

```
CREATE TABLE `dim_customers` (  
  `PK_customer` int NOT NULL,  
  `customerNumber` int NOT NULL,  
  `employeeNumber` int DEFAULT NULL,  
  `country` varchar(150) DEFAULT NULL,  
  PRIMARY KEY (`PK_customer`)  
)
```

```
CREATE TABLE `dim_employees` (  
  `PK_employees` int NOT NULL,  
  `employeeNumber` int NOT NULL,  
  `lastName` varchar(50) DEFAULT NULL,  
  `firstName` varchar(50) DEFAULT NULL,  
  PRIMARY KEY (`PK_employees`)  
)
```

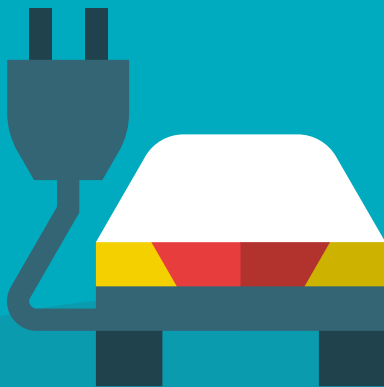
```
CREATE TABLE `dim_products` (  
  `PK_products` int NOT NULL,  
  `productCode` varchar(15) NOT NULL,  
  `productName` varchar(250) DEFAULT NULL,  
  `productline` varchar(100) DEFAULT NULL,  
  `productVendor` varchar(250) DEFAULT NULL,  
  PRIMARY KEY (`PK_products`)  
)
```

```
CREATE TABLE fact_orderdetails (  
  fact_commande INT NOT NULL,  
  PK_products INT NOT NULL,  
  PK_customers INT NOT NULL,  
  PK_employees INT NOT NULL,  
  PK_calendar INT NOT NULL,  
  Montant FLOAT(10,0) NOT NULL,  
  orderNumber INT NOT NULL,  
  orderDate DATE NOT NULL,  
  productCode VARCHAR(15) NOT NULL,  
  PRIMARY KEY (fact_commande, PK_products, PK_customers, PK_employees, PK_calendar),  
  FOREIGN KEY (PK_products) REFERENCES dim_products(PK_products),  
  FOREIGN KEY (PK_customers) REFERENCES dim_customers(PK_customer),  
  FOREIGN KEY (PK_employees) REFERENCES dim_employees(PK_employees),  
  FOREIGN KEY (PK_calendar) REFERENCES dim_calendar(PK_calendar)  
);
```

```
CREATE TABLE dim_calendar(  
  PK_calendar INT PRIMARY KEY,  
  Date DATE,  
  Annee INT NOT NULL,  
  Mois INT NOT NULL,  
  NomMois NVARCHAR(50) NOT NULL,  
  Trimestre NVARCHAR(5) NOT NULL,  
  Semestre NVARCHAR(2) NOT NULL,  
  JourSemaine NVARCHAR(50) NOT NULL,  
  NumJourSemaine INT NOT NULL  
);
```

04

# Chargement des données



# Chargement des données avec scripts SQL et Talend.



A

Chargement des données à l'aide de scripts SQL.



B

Chargement des données avec la plateforme d'intégration  
de données Talend

## A

## Chargement des données à l'aide de scripts SQL.

```
1 INSERT INTO dim_customers (PK_customer, customerNumber, employeeNumber, country)
2 SELECT
3     ROW_NUMBER() OVER(ORDER BY customerNumber) AS dim_customer,
4     customerNumber,
5     employeeNumber,
6     country
7 FROM customers;
```

result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

PK_customer	customerNumber	employeeNumber	country
1	103	1370	France
2	112	1166	USA
3	114	1611	Australia
4	119	1370	France
5	121	1504	Norway
6	124	1165	USA
7	125	1621	Poland

```
1 INSERT INTO dim_employees (PK_employees, employeeNumber, lastName, firstName)
2 SELECT
3     ROW_NUMBER() OVER(ORDER BY employeeNumber) AS dim_employees,
4     employeeNumber,
5     lastName,
6     firstName
7 FROM employees;
```

result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

PK_employees	employeeNumber	lastName	firstName
1	1002	Murphy	Diane
2	1056	Patterson	Mary
3	1076	Firrelli	Jeff
4	1088	Patterson	William
5	1102	Bondur	Gerard
6	1143	Bow	Anthony
7	1165	Jennings	Leslie

```

1 • INSERT INTO fact_orderdetails (fact_commande,PK_products,PK_customers,PK_employees,PK_calendar,Montant,orderNumber,orderDate,productCode)
2 SELECT
3     ROW_NUMBER() OVER (ORDER BY o.orderNumber) AS fact_commande,
4     dp.PK_products,dc.PK_customer,de.PK_employees,dcal.PK_calendar,o.Montant,o.orderNumber,dor.orderDate,o.productCode
5 FROM orderdetails o
6 JOIN dim_orders dor ON o.orderNumber = dor.orderNumber
7 JOIN dim_products dp ON o.productCode = dp.productCode
8 JOIN dim_customers dc ON dor.customerNumber = dc.customerNumber
9 JOIN dim_employees de ON o.orderNumber = de.orderNumber
10 JOIN dim_calendar dcal ON dor.orderDate = dcal.orderDate;

```

Result Grid   Filter Rows:  Export:  Wrap Cell Content: 

	fact_commande	PK_products	PK_customer	PK_employees	PK_calendar	Montant	orderNumber	orderDate	productCode
1	22	86	10	1042026	4080	10100	06/01/2003	S18_1749	
2	26	86	10	2042026	2754	10100	06/01/2003	S18_2248	
3	49	86	10	3042026	1660	10100	06/01/2003	S18_4409	
4	79	86	10	4042026	1729	10100	06/01/2003	S24_3969	
5	28	8	17	5042026	2701	10101	09/01/2003	S18_2325	
6	32	8	17	6042026	4344	10101	09/01/2003	S18_2795	
7	60	8	17	7042026	1464	10101	09/01/2003	S24_1937	
8	63	8	17	8042026	2040	10101	09/01/2003	S24_2022	
9	18	28	11	9042026	3726	10102	10/01/2003	S18_1342	
10	19	28	11	10042026	1768	10102	10/01/2003	S18_1367	
11	2	5	17	11042026	5572	10103	29/01/2003	S10_1949	
12	5	5	17	12042026	5026	10103	29/01/2003	S10_4962	
13	8	5	17	13042026	3284	10103	29/01/2003	S12_1666	

## B

## Chargement des données avec Talend

## ETAPE 1

## ETAPE 2

Connexion base de données

**Mise à jour de la connexion à la base de données - Étape 2/2**

① Vous devez cliquer sur le bouton Vérifier pour vérifier les paramètres de la base de données

Type de BdD MySQL

Version de la base de données MySQL 5

Chaîne de caractères de connexion jdbc:mysql://localhost:3306/datawarehouse?noDatetimeStringSyn

Connexion root

Mot de passe •••••

Serveur localhost

Port 3306

Base de données datawarehouse

Paramètres supplémentaires noDatetimeStringSync=true&enabledTLSProtocols=TLSv1.2,TLSv1

Tester la connexion



\*Référentiel

LOCAL: MacroBus

- Modèles SQL
- Métadonnées
  - Connexions de bases de données
    - datawarehouse\_MacroBus 0.1
      - Requêtes
      - Schémas des tables
        - dim\_customers
        - dim\_employees
        - dim\_orders
        - dim\_products
        - fact\_orderdetails
      - Schémas synonymes
      - Voir les schémas
    - Projet1 0.1
      - Requêtes
      - Schémas des tables
        - customers
        - employees
        - orderdetails
        - orders
        - products
      - Schémas synonymes
      - Voir les schémas

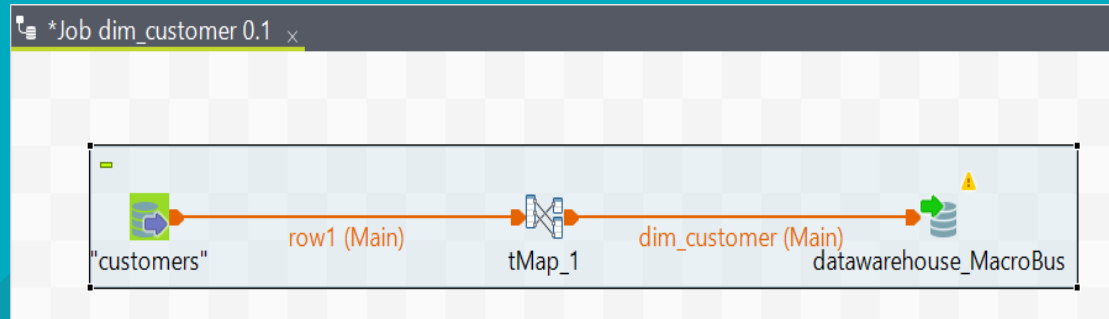
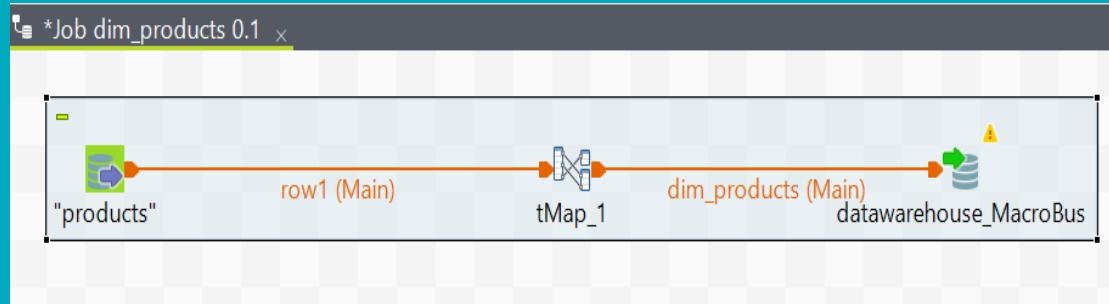
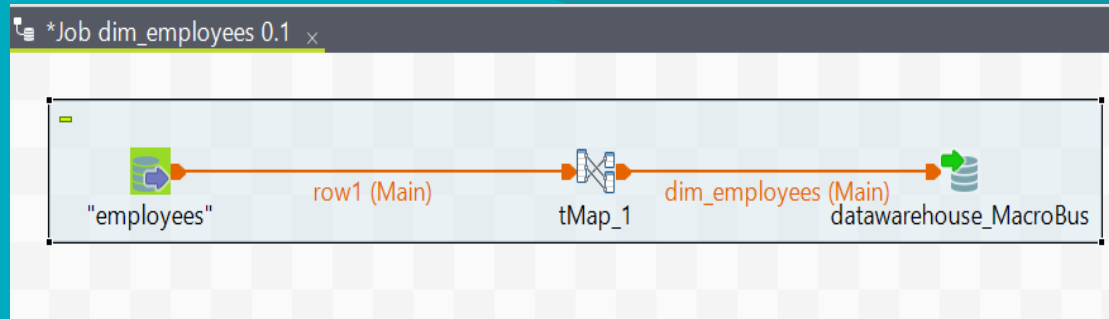
Paramètre de connexion a la base de donnée

Connexions aux bases de données existantes



### ETAPE 3

concevoir différents jobs pour charger plusieurs tables à partir d'une table source. À l'intérieur de ces jobs, le composant **tMap** est utilisé pour transformer, filtrer et répartir les données vers les différentes tables cibles





# Création de la table de dimension (dim\_customer)

## ETAPE 4

Talend Open Studio for Data Integration - tMap - tMap\_1

Find :

Auto map !

**row1**

Column
customerNumber
customerName
contactLastName
contactFirstName
phone
addressLine1
addressLine2
city
state
postalCode
country
employeeNumber
creditLimit

**Var**

Expression	Type	Variable
Numeric.sequence("s1...	Integer	<input checked="" type="checkbox"/> var1

**dim\_customer**

Expression	Column
Var.var1	dim_customer
row1.customerNumber	customerNum...
row1.employeeNumber	employeeNum...
row1.country	country

Éditeur de schéma Éditeur d'expression

row1

Colonne	C...	Type	<input checked="" type="checkbox"/> N.	Modèle ...	Lon...	Préci...	Par...	Com...
customerNu...	<input checked="" type="checkbox"/>	int	<input type="checkbox"/>		10	0		
customerName	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>		50	0		
contactLastNa...	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>		50	0		
contactFirstN...	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>		50	0		

dim\_customer

Colonne	C...	Type	<input checked="" type="checkbox"/> N.	Modèle d...	Long...	Préci...	Par...	Com...
dim_customer	<input checked="" type="checkbox"/>	int	<input type="checkbox"/>					
customerNum...	<input type="checkbox"/>	int	<input type="checkbox"/>		10	0		
employeeNum...	<input type="checkbox"/>	Inte...	<input checked="" type="checkbox"/>		10	0		
country	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>		150	0		

Appliquer Ok Annuler

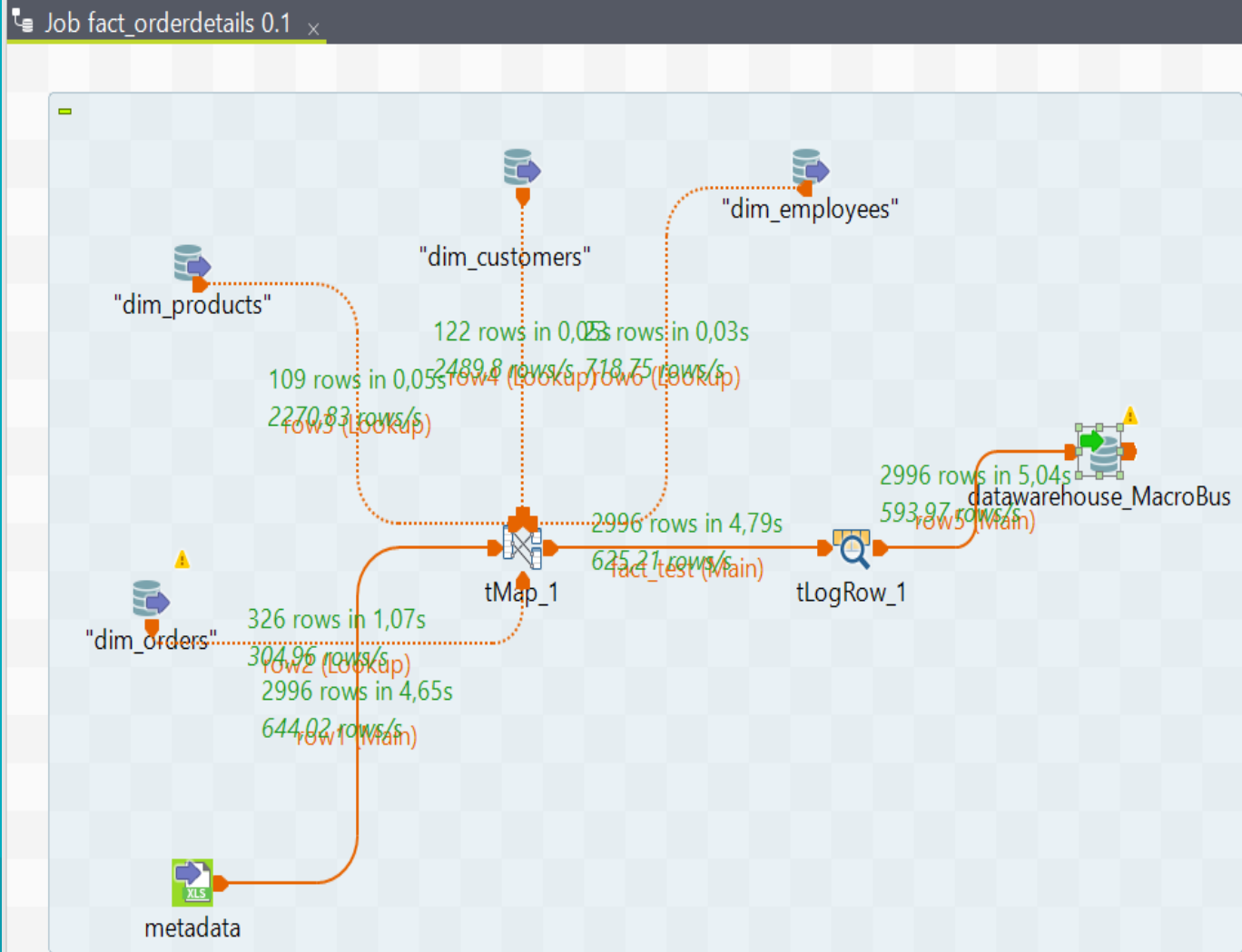
## ETAPE 5

# Chargement de la table de dimension dans entrepôt de donnée

The screenshot displays the Talend Studio interface. At the top, a workflow diagram shows a sequence of components: a green icon labeled "customers", a connector labeled "row1 (Main)", a tMap component labeled "tMap\_1", another connector labeled "dim\_customer (Main)", and finally the "datawarehouse\_MacroBus" component. Below the diagram, the "Designer" tab is active, showing the configuration for the "datawarehouse\_MacroBus(tDBOutput\_1)(MySQL)" component. The configuration is organized into sections: "Paramètres simples", "Paramètres avancés", "Paramètres dynamiques", "Vue", and "Documentation". The "Paramètres simples" section is expanded, showing the following settings: Database: MySQL (with an "Appliquer" button); Type de propriété: Référentiel (with a dropdown arrow); Bases de données (MYSQL): d...; Version de la base de données: MySQL 5 (with a dropdown arrow); ☐ Utiliser une connexion existante; Hôte: "localhost" (with a dropdown arrow); Port: "3306" (with a dropdown arrow); Base de données: "datawarehouse" (with a dropdown arrow); Utilisateur: "root" (with a dropdown arrow); Mot de passe: "\*\*\*\*\*" (with a dropdown arrow); Table: "dim\_customers" (with a dropdown arrow and a "..." button); Action sur la table: Créer la table si elle n'existe pas (with a dropdown arrow); Action sur les données: Insert Ignor...; Schéma: Built-in (with a dropdown arrow); Modifier le schéma: Sync colonnes (with a button); Source de données: This option only applies when deploying and running in the Talend Runtime; ☐ Spécifier un alias de source de données; ☐ Arrêter en cas d'erreur.

## ETAPE 6

# Création du job pour concevoir et charger la table de fait



## ETAPE 7

Sélection des différentes colonnes à charger dans la table de fait en fonction des jointures appropriées

**row1**

Column
orderNumber
productCode
quantityOrdered
priceEach
orderLineNumber

**row2**

Property	Value
Lookup Model	Charger une fois
Match Model	Correspondance unique
Join Model	Inner Join
Store temp data	false

Clé d'expr.	Column
	dim_orders
row1.orderNumber	orderNumber
	orderDate
	customerNumber

**Var**

Expression	Type	Variable
Numeric.sequence("s1",1,1)	Integer	<input checked="" type="checkbox"/> var1

**fact test**

Expression	Column
Var.var1	fact_commande
row7.PK_calendar	PK_calendar
row3.dim_products	PK_products
row4.dim_customer	PK_customer
row6.dim_employees	PK_employees
row1.quantityOrdered * row1.priceEach	Montant
row1.orderNumber	orderNumber
row2.orderDate	orderDate
row3.productCode	productCode

Éditeur de schéma

Éditeur d'expression

row2




Colonne	C...	Type	<input checked="" type="checkbox"/> N.	Modèle de date...	Longueur	Précision	Par dé...	Comme...
dim_orders	<input checked="" type="checkbox"/>	int	<input type="checkbox"/>		10	0		
orderNumber	<input type="checkbox"/>	Integer	<input checked="" type="checkbox"/>		10	0		
orderDate	<input type="checkbox"/>	Date	<input type="checkbox"/>	"yyyy-MM-dd"	10	0		
customerNumber	<input type="checkbox"/>	int	<input type="checkbox"/>		65535	0		















fact\_test

Colonne	C...	Type	<input checked="" type="checkbox"/> N.	Modèle de date...	Longueur	Précision	Par d...	Comme...
fact_commande	<input checked="" type="checkbox"/>	int	<input type="checkbox"/>		10			
PK_calendar	<input checked="" type="checkbox"/>	Date	<input checked="" type="checkbox"/>	"dd-MM-yyyy"	0			
PK_products	<input checked="" type="checkbox"/>	int	<input type="checkbox"/>		10	0		
PK_customer	<input checked="" type="checkbox"/>	int	<input type="checkbox"/>		10	0		

## ETAPE 8

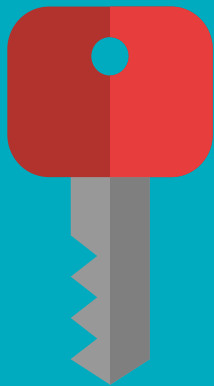
# Chargement de la table de fait dans entrepôt de donnée

 datawarehouse\_MacroBus(tDBOutput\_1)(MySQL)  

Paramètres simples	Database <span>MySQL</span>  <span>Appliquer</span>
Paramètres avancés	Type de propriété <span>Référentiel</span>  <span>Bases de données (MySQL):d</span> 
Paramètres dynamiques	Version de la base de données <span>MySQL 5</span> 
Vue	<input type="checkbox"/> Utiliser une connexion existante
Documentation	Hôte <span>"localhost"</span>  Port <span>"3306"</span> 
	Base de données <span>"datawarehouse"</span> 
	Utilisateur <span>"root"</span>  Mot de passe <span>*****</span> 
	Table <span>"fact_orderdetails"</span>  
	Action sur la table <span>Créer la table si elle n'existe pas</span>  Action sur les données <span>Insert Ignore</span>
	Schéma <span>Built-in</span>  <span>Modifier le schéma</span>  <span>Sync colonnes</span>
	Source de données This option only applies when deploying and running in the Talend Runtime <input type="checkbox"/> Spécifier un alias de source de données <input type="checkbox"/> Arrêter en cas d'erreur

05

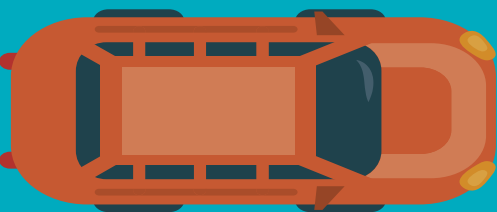
Validation fonctionnel de l'entrepôt



La **validation fonctionnelle** est une étape cruciale dans la mise en place d'un **entrepôt de données (data warehouse)**. Elle permet de s'assurer que le système répond bien aux **besoins métiers** et que les **données sont fiables, cohérentes et exploitables**.

Pour y parvenir, nous allons vérifier que nous obtenons les mêmes résultats quelque soit l'approche de chargement de la base de données :

- ❖ **Sur la base de donnée source**
- ❖ **Sur la datawarehouse 1 (script sql)**
- ❖ **Sur la datawarehouse 2 (Talend avec Power BI)**



Sur la datawarehouse 1 (script sql)

```
4 • SELECT COUNT(*) AS nombretotaldelignes
5 FROM orderdetails;
```

Result Grid	Filter Rows:	Export:
nombretotaldelignes		
2996		

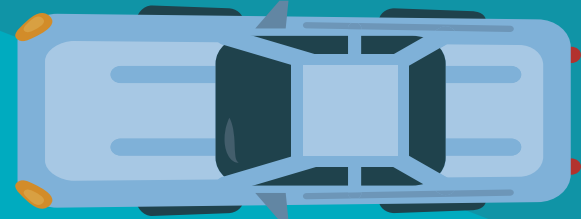


Sur la base de donnée source

Nombre de lignes  
chargées

```
4 • SELECT COUNT(*) AS NombreTotalDeLignes
5 FROM fact_orderdetails;
```

Result Grid	Filter Rows:	Export:
NombreTotalDeLignes		
2996		



Sur la datawarehouse 2  
(Talend avec Power BI)

Nombre de ligne charger

2996

Montant total des  
commandes

```
7 • SELECT SUM(Montant) AS somme_totale
8 FROM orderdetails;
```

Result Grid	Filter Rows:	Export:
somme_totale		
9604536		

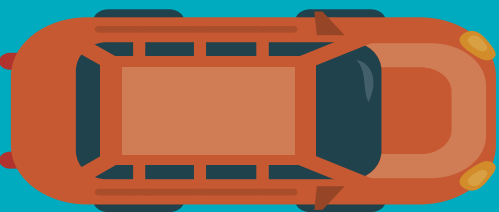
```
5 • SELECT SUM(Montant) AS SommeDesMontant
6 FROM fact_orderdetails;
```

Result Grid	Filter Rows:	Export:
SommeDesMontant		
9604149		

Montant total des commandes

9 604 149





Sur la datawarehouse 1 (script sql)

```
10 • SELECT COUNT(DISTINCT orderNumber)
11     AS nombre_de_lignes_distinctes
12     FROM orderdetails;
```

Result Grid	Filter Rows:	Exports	Wrap Cell Content:
nombre_de_lignes_distinctes			
326			



Sur la base de donnée source

Nombre total de commandes

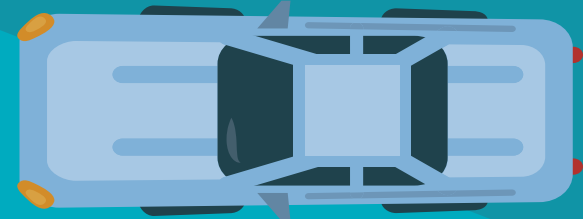
```
9 • SELECT COUNT(DISTINCT orderNumber) AS NombreElementDistinct
10 FROM fact_orderdetails;
```

Result Grid	Filter Rows:	Exports	Wrap Cell Content:
NombreElementDistinct			
326			

Montant total des commandes durant le second semestre de l'année 2003

```
14 • SELECT SUM(Montant) AS MontantTotal
15 FROM fact_orderdetails
16 WHERE orderDate BETWEEN '2003-07-01' AND '2003-12-31';
17
```

Result Grid	Filter Rows:	Exports	Wrap Cell Content:
MontantTotal			
2395971			



Sur la datawarehouse 2  
(Talend avec Power BI)

Nombre total des commandes

326

Montant total des commandes

2 395 971



06

## Conception de tableaux de bord pour l'analyse



# Question 1 : Nombre et montant total des commandes par territoire (Territory)



Montant total des commandes

9 604 149

Nombre de pays

25

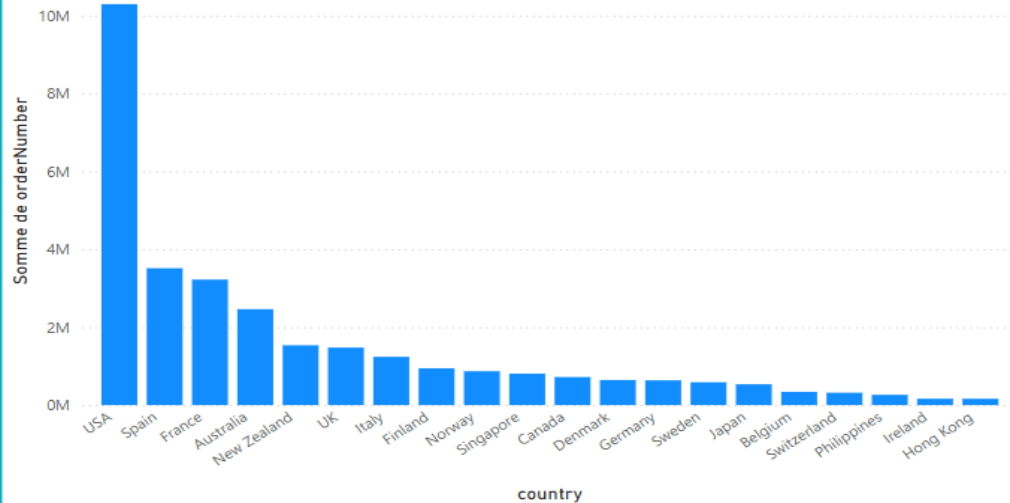
Nombre total des commandes

326

Montant total des commandes par territoire

country	Somme de Montant
Hong Kong	45 481
Ireland	49 898
Philippines	87 469
Belgium	100 067
Switzerland	108 776
Japan	167 911
Sweden	187 638
Germany	196 474
Canada	205 915
Denmark	218 994
Singapore	264 000
Norway	270 842
Finland	295 147
Italy	360 615
UK	436 952
New Zealand	476 836
Australia	751 120
France	1 007 367
Spain	1 099 381
USA	3 273 266
<b>Total</b>	<b>9 604 149</b>

Montant total des commandes par territoire



## Question 2: Top 3 commerciaux qui reçoivent le plus de commandes

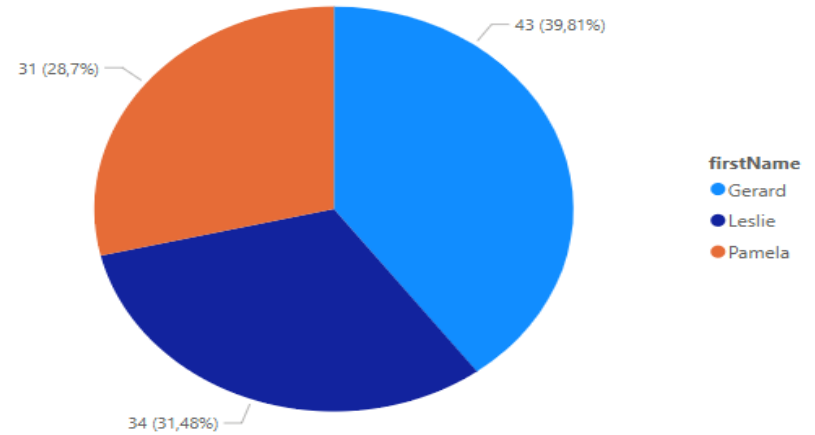


firstName

Andy Anthony Barry Diane Foon Yue George Gerard Jeff Julie Larry Leslie Loui Mami Martin Mary Pamela >

firstName	lastName	Nombre de orderNumber
Gerard	Hernandez	43
Leslie	Jennings	34
Pamela	Castillo	31
Barry	Jones	25
George	Vanauf	22
Larry	Bott	22
Loui	Bondur	20
Andy	Fixter	19
Peter	Marsh	19
Steve	Patterson	18
Foon Yue	Tseng	17
Mami	Nishi	16
Julie	Firrelli	14
Leslie	Thompson	14
Martin	Gerard	12
Total		326

TOP 3 des des meilleures commerciaux



# Question 3 : Les trois 03 produits les plus commandés durant le deuxième trimestre de l'année 2003



Année

2003

2004

2005

Nombre de commande

20

Trimestre

1

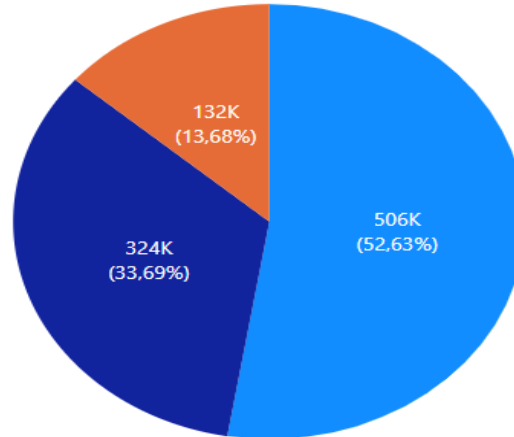
2

3

4



Les 3 produits les plus commandés durant le deuxième trimestre



**productLine**

● Classic Cars

● Vintage Cars

● Motorcycles

# Question 4 : Montant total des commandes par filiale et par catégorie de produit durant le second semestre de l'année 2003



Montant total des commandes

2 395 971

Nombre de produit

109

Nombre total des commandes

77

Trimes... ▾

☐ 1

☐ 2

☒ 3

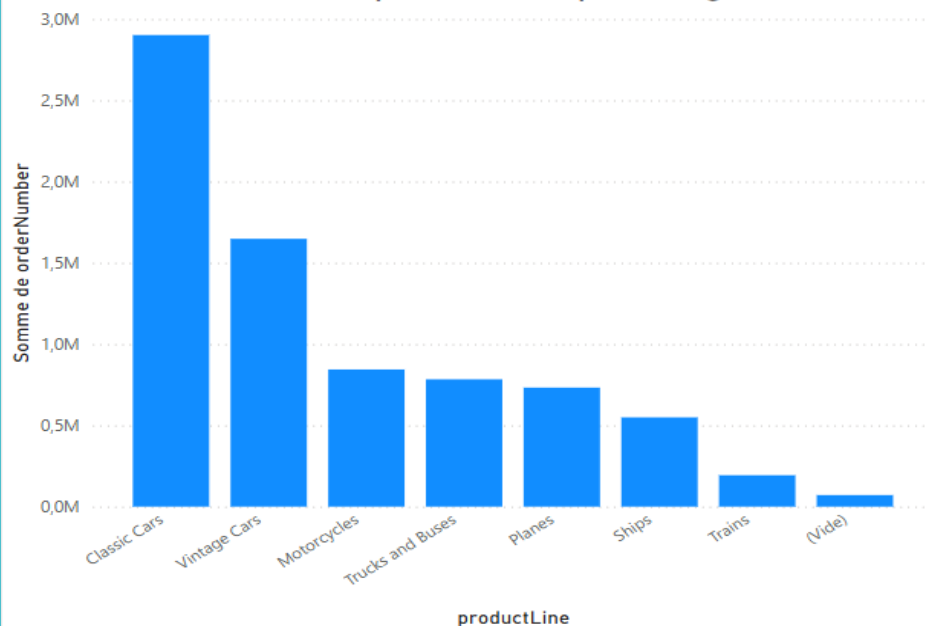
☐ 4

Année ▾

☒ 2003

☐ 2004

Montant total par filiale ou par catégorie



Montant total par filiale et par catégorie

productVendor	Somme de Montant
Welly Diecast Productions	203 796
Unimax Art Galleries	230 021
Studio M Art Models	119 285
Second Gear Diecast	175 862
Red Start Diecast	116 351
Motor City Art Classics	163 063
Min Lin Diecast	177 600
Highway 66 Mini Classics	155 688
Gearbox Collectibles	222 542
Exoto Designs	196 804
Classic Metal Creations	252 433
Carousel DieCast Legends	156 045
Autoart Studio Design	184 246
Total	2 395 971



# Conclusion et recommandation

**Conclusions** : Le projet optimise les performances commerciales de l'entreprise MacroBus et faciliter la prise de décision à travers un système dédié

**Recommandations** : Étendre la solution pour inclure des prédictions basées sur l'historique des commandes.

Exemple : Les modèles prédictifs basés sur le machine learning

# THANKS

