Aix-Marseille School Of Economics

Managing databases: SQL and NoSQL

Léa Aumagy,

Jean-Baptiste Boris Gomez,

Dhalil Bello,

Miracle Vodoumbo.

Magistère 1 Ingénieur Economiste

May 3, 2022

Contents

1 Introduction 2

2 Model 3

2.1 The Data Dictionary 3

2.2 The Conceptual Data Model 5

2.3 The Logical Data Model 5

2.4 The creation script for MySQL 7

3 Queries 13

3.1 Basic queries 13

3.2 WHERE clause queries 14

3.3 ORDER BY queries 16

3.4 Multi-table queries 19

3.5 Queries with Numeric expressions and functions 20

3.6 GROUP BY queries 22

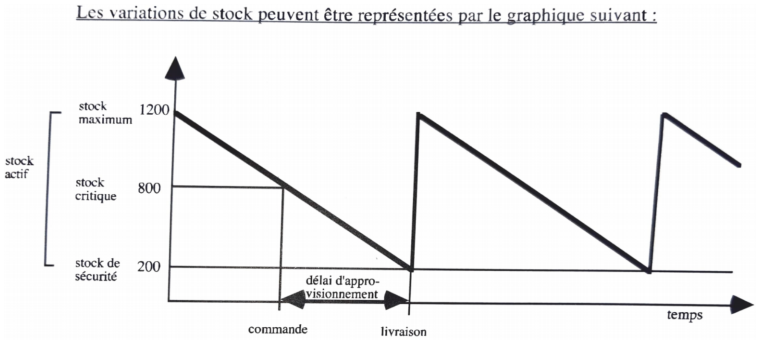
3.7 Nested queries 23

4 MongoDB 25

5 Conclusion 27

1 Introduction

The control of product inventories is one of the major difficulties encountered by companies today. Indeed, the stocks represent a considerable part of the company's capital. It is important and its good management would allow the company to avoid interruptions in the production process and to meet the customers' demands.
For this reason, it is interesting for the company manager to know the different stock thresholds (minimum stocks, maximum stocks, safety stocks, supply times).
We can visualise the turnover process of a company's stock management through a graph.



Having a very high level of stock at all times allows the company to respond quickly to customer requests. But this solution generates extremely high costs which naturally weighs on the selling prices and thus on the global competitiveness of the company.
For this reason, creating a relational database containing product records (names, location, quantity, price...) would be a reliable answer to the question asked by all companies:

How to have a better management of product stocks?

The goal of this database will be to reach a balance, i.e., to avoid both over-stocking and under-stocking of products. In this project, we were interested in the DIY (Do-it-yourself) company Leroy Merlin.
https://www.leroymerlin.fr/

Firstly, we will show how we created a database, by establishing a data dictionary, a conceptual data model that we will transform into a logical data model, by creating a script on MySQL to use this database.
In a second step, we will manipulate different relational data queries.
Then in a third step, we will manipulate different non-relational data queries.

2 Model

Lists of entities in the relational tables of the products of the company Leroy Merlin.
Leroy Merlin has several product categories such as: Terrace and garden, Decoration and lighting, Bathroom, Tile and flooring, Carpentry, building materials, Heating and plumbing, Kitchen, Storage and dressing room...

2.1 The Data Dictionary

We can establish the Data Dictionary in the form of a table (the list of all attributes with their description).

Client	
Attributs	**Description**
idClient	Numéro d'identification du client
nom_client	Nom du client
prenom_client	Prénom du client
date_naissance	Date de naissance du client
civilité	Monsieur ou Madame suivant le sexe du client
adresse_tel	Téléphone
ville	Ville de résidence
mail	Adresse mail

Commande_client	
Attributs	**Description**
idcmd_clt	Numéro d'identification de la commande
date_cmd	Date de la commande
date_livraison	Date de la livraison

Livraison_client	
Attributs	**Description**
id_bon_livraison	Numéro d'identification du bon de livraison
date_BL_client	Date d'émission du bon de livraison
Total_BL_client	Montant total sur le bon de livraison

Stock	
Attributs	**Description**
id_stock	Numéro d'identification du stock
stock_min	La quantité minimale de produit dans le stock
stock_max	La quantité maximale de produit dans le stock

Sous_Categorie	
Attributs	**Description**
id_sous_categorie	Identifiant de la sous catégorie
nom_sous_categorie	Nom de la sous catégorie

Categorie	
Attributs	**Description**
id_categorie	Identifiant de la catégorie
nom_categorie	Nom de la catégorie
type_categorie	Type de produits associés à la catégorie

Commande_frs	
Attributs	**Description**
id_cmd_frs	Identifiant de la commande chez un fournisseur
Date_cmd_frs	Date de la commande

Modele	
Attributs	**Description**
num_modele	Numéro du modèle
stock_reserve	La quantité en stock
taille	La taille du modèle

Inventaire	
Attributs	**Description**
Date_inventaire	Date de l'inventaire
quantité_inventorier	Quantité de produit inventorier

Produit	
Attributs	**Description**
id_produit	Numéro d'identification du produit
nom_produit	Nom du produit
prix_produit	Prix du produit
largeur_produit	Largeur du produit
hauteur_cm	Hauteur en cm du produit
Epaisseur_cadre	Epaisseur du cadre du produit
matière	Matière de fabrication du produit
type_panneau	Type de panneau du produit
couleur_produit	Couleur du produit
garantie_constructeur	Garantie du produit

Fournisseur	
Attributs	**Description**
id_frs	Numéro d'identification du fournisseur
nom_frs	Nom du fournisseur
telephone_frs	Téléphone du fournisseur
adresse_mail_frs	Adresse mail du fournisseur
ville_frs	Ville de résidence du fournisseur

Payement_frs	
Attributs	**Description**
id_payement	Identifiant du payement
date_payement	Date du paiement au founisseur
echeance_payement	Échéance du paiement
montant_payement	Montant payé
moyens_payement	Moyens de paiement (Carte, chèque, espèce)

Pieces	
Attributs	**Description**
id_pieces	Identifiant de la pièce
delai_livraison	Délai de livraison de la pièce
prix_cmd	Prix de la commande

Remise	
Attributs	**Description**
id_remise	Identifiant de la remise
pourcent_remise	Pourcentage de la remise

Facture_achat	
Attributs	**Description**
id_fact_ach	Identifiant de la facture d'achat
date_ach	Date d'achat
date_fact_ach	Date d'établissement de la facture
montant_HT_ach	Montant HT de l'achat
num_fact_ach	Numéro de la facture

Facture_vente	
Attributs	**Description**
id_fact_vte	Identifiant de la facture de vente
date_vte	Date de vente
date_fact_vte	Date d'établissement de la facture
montant_HT_vte	Montant HT de la vente
num_fact_vte	Numéro de la facture

Livraison_clt	
Attributs	**Description**
id_bon_livraison	Numéro d'identification du bon de livraison
date_bl_clt	Date d'établissement du bon de livraison
Total_bl_clt	Montant total sur le bon de livraison

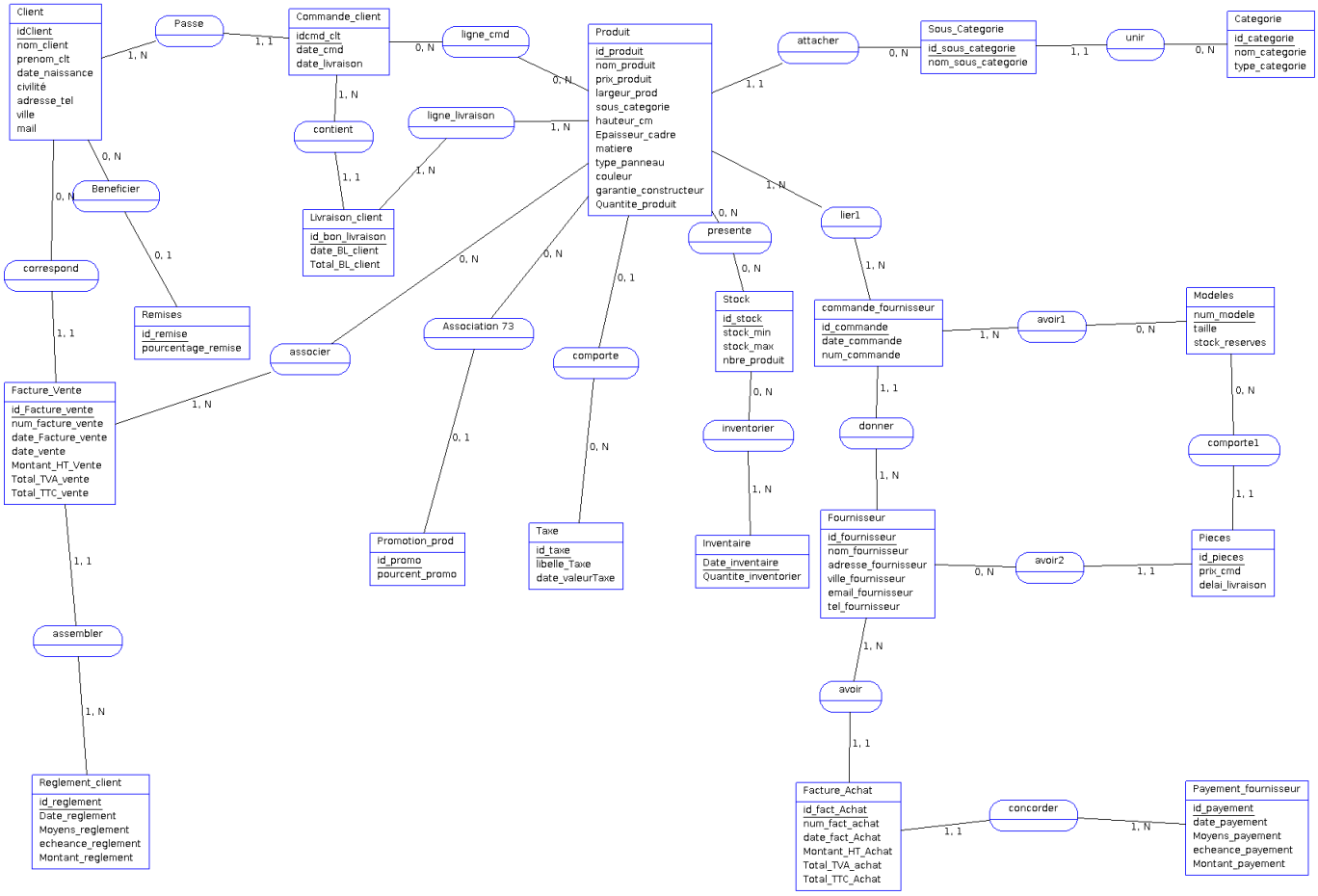
Reglement_clt	
Attributs	**Description**
id_reglement	Numéro d'identification du règlement
date_reglement	Date du règlement
echeance_reglement	Echeance du règlement
montant_reglement	Montant du règlement
moyens_reglement	Mode de paiement du règlement

Promotion_produit	
Attributs	**Description**
id_promo_prod	Identifiant de la promotion sur produit
pourcent_promo	Pourcentage de la promotion

Taxe	
Attributs	**Description**
id_tax	Identifiant de la taxe
libelle_tax	Libellé de la taxe
date_valeur_tax	Date de valeur de la taxe

2.2 The Conceptual Data Model

From this we can build the Conceptual Data Model (CDM) with AnalyseSI software.



Moreover, in the preliminary project we had an entity called "carte_de_fidelite" linked with "Client", its cardinalities were 0,1 and 1,1. We preferred to delete this entity for simplicity.
We can define the different stages of standardisation.

First normal form (1FN):
A relation is in (1NF) if all its attributes each have an atomic value (monovalent).

Second normal form (2NF): A relation is in (2NF) if and only if it is in 1NF and if any non-key field does not depend on any part of the key.

Third normal form (3NF):
A relation is in 3NF if and only if it is in 2NF and if any non-key field does not depend on another field no.

2.3 The Logical Data Model

Then we can transform the conceptual data model into a Logical Data Model. We put the primary keys in bold and foreign keys are represented by #.

- Client (**id_client**, nom_client, prenom_client, date_naissance, adresse_mail, civilité, adresse_tel, ville)
- Commande_client (**id_cmd_clt**, #id_client, date_cmd, date_livraison)
- Remise (**id_remise**, #id_client, pourcentage_remise)
- Reglement_client (**id_reglement**, date_reglement, moyens_reglement, echeance_reglement, montant_reglement)
- Facture_vente (**id_facture_vente**, #id_reglement, #id_client, num_facture, date_fact_vente, date_vente, montant_HT_vente)
- Livraison_client (**id_bon_livraison**, #id_cmd_clt, date_bl_clt, Total_bl_clt)
- Taxe (**id_taxe**, libelle_Taxe, date_valeurTaxe)
- Categorie (**id_categorie**, nom_categorie, type_categorie)
- Sous_categorie (**id_sous_categorie**, #id_categorie, nom_sous_categorie)
- Produit (**id_produit**, #id_sous_categorie, #id_taxe, nom_produit, prix_produit, largeur_prod, hauteur_cm, Epaisseur_cadre, matiere, type_panneau, couleur, garantie_constructeur)
- Ligne_livraison (#id_bon_livraison, #id_produit)
- Ligne_commande (#id_cmd_clt, #id_produit)
- Fact_vente_Produit (#id_facture_vente, #id_produit)
- Promotion_produit (**id_promo**, #id_produit, pourcentage_promo)
- Stock (**id_stock**, stock_min, stock_max)
- Quantite (#id_stock, #id_produit, quantite_produit)
- Inventaire (**date_inventaire**, quantite_inventorier)
- Stock_Inventaire (#id_stock, date_inventaire)
- Fournisseur (**id_fournisseur**, nom_fournisseur,ville_fournisseur,mail_fournisseur,tel_fournisseur)
- Commande_fournisseur (**id_cmd_frs**, #id_fournisseur, date_cmd_frs)
- Produit_commande_frs (#id_cmd_frs, #id_produit)
- Modele (**num_modele**, taille, stock_reserves)
- Modele_commande_frs (#num_modele, #id_cmd_frs)
- Pieces (**id_pieces**, num_modele, #id_fournisseur, prix_cmd, delai_livraison)
- Payement_frs (**id_payement**, date_payement, Moyens_payement, echeance_payement, montant_payement)
- Facture_Achat (**id_fact_Achat**, #id_fournisseur, #id_payement, num_fact_Achat, date_fact_Achat, date_Achat, Montant_HT_Achat)

2.4 The creation script for MySQL

Afterwards we were able to write the following script for mySQL, for the creation of the different tables and the allocation of primary and foreign keys. The details of the insertion of elements in each table can be found in the .txt file.

```sql
DROP DATABASE IF EXISTS leroymerlin;
CREATE DATABASE leroymerlin;
USE leroymerlin;

drop table if exists Client ;
drop table if exists Commande_clt ;
drop table if exists Remise ;
drop table if exists Reglement_clt ;
drop table if exists Facture_vente ;
drop table if exists Livraison_clt ;
drop table if exists Taxe;
drop table if exists Categorie;
drop table if exists Sous_Categorie;
drop table if exists Produit;
drop table if exists Ligne_livraison;
drop table if exists Ligne_commande;
drop table if exists Fact_Vte_Prod;
drop table if exists Promotion_produit;
drop table if exists Stock;
drop table if exists Quantite_prod;
drop table if exists Inventaire;
drop table if exists Stock_Inventaire;
drop table if exists Fournisseur;
drop table if exists Commande_frs;
drop table if exists Produit_Commande_frs;
drop table if exists Modele;
drop table if exists Modele_Commande_frs;
drop table if exists Pieces;
drop table if exists Payement_frs;
drop table if exists Facture_Achat;


CREATE TABLE Client (
  id_clt int(11) NOT NULL,
  nom_clt varchar(255) DEFAULT NULL,
  prenom_clt varchar(255) DEFAULT NULL,
  date_naissance_clt date DEFAULT NULL,
  civilite_clt varchar(255) DEFAULT NULL,
  adresse_mail_clt varchar(255) DEFAULT NULL,
  ville_clt varchar(255) DEFAULT NULL,
  telephone_clt varchar(255) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE Commande_clt (
  id_cmd_clt int(11) NOT NULL,
  id_clt int(11) NOT NULL,
  date_cmd datetime DEFAULT NULL,
  date_livraison datetime DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE Remise (
  id_remise int(11) NOT NULL,
```

```sql
  id_clt int(11) NOT NULL,
  pourcent_remise double DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE Reglement_clt (
  id_reglement int(11) NOT NULL,
  date_reglement datetime DEFAULT NULL,
  Moyens_reglement varchar(255) DEFAULT NULL,
  echeance_reglement varchar(255) DEFAULT NULL,
  montant_reglement decimal(19,3) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE Facture_vente (
  id_fact_vte int(11) NOT NULL,
  id_clt int(11) NOT NULL,
  id_reglement int(11) NOT NULL,
  num_fact_vte varchar(255) DEFAULT NULL,
  date_fact_vte datetime DEFAULT NULL,
  date_vte datetime DEFAULT NULL,
  montant_HT_vte decimal(19,3) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE Livraison_clt (
  id_bon_livraison int(11) NOT NULL,
  id_cmd_clt int(11) NOT NULL,
  date_bl_clt datetime DEFAULT NULL,
  Total_bl_clt int DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE Taxe (
  id_tax int(11) NOT NULL,
  libelle_tax varchar(255) DEFAULT NULL,
  date_valeur_tax datetime DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE Categorie (
  id_categorie int(11) NOT NULL,
  nom_categorie varchar(255) DEFAULT NULL,
  type_categorie varchar(255) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE Sous_Categorie (
  id_sous_categorie int(11) NOT NULL,
  id_categorie int(11) NOT NULL,
  nom_sous_categorie varchar(255) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE Produit (
  id_prod int(11) NOT NULL,
  id_sous_categorie int(11) NOT NULL,
  id_tax int(11) NOT NULL,
  nom_prod varchar(255) DEFAULT NULL,
  prix_prod decimal(19,3) DEFAULT NULL,
  largeur_prod double DEFAULT NULL,
  hauteur_prod int DEFAULT NULL,
  Epaisseur_cadre double DEFAULT NULL,
  matiere_prod varchar(255) DEFAULT NULL,
```

```sql
  type_panneau varchar(255) DEFAULT NULL,
  couleur_prod varchar(255) DEFAULT NULL,
  garantie_constructeur_an int(11) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE Ligne_livraison (
    id_bon_livraison int(11) NOT NULL,
    id_prod int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE Ligne_commande (
    id_cmd_clt int(11) NOT NULL,
    id_prod int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE Fact_Vte_Prod (
    id_fact_vte int(11) NOT NULL,
    id_prod int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE Promotion_produit (
    id_promo_prod int(11) NOT NULL,
    id_prod int(11) NOT NULL,
    pourcent_promo double DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE Stock (
    id_stock int(11) NOT NULL,
    stock_min int DEFAULT NULL,
    stock_max int DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE Quantite_prod (
    id_stock int(11) NOT NULL,
    id_prod int(11) NOT NULL,
    quantite int DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE Inventaire (
    Date_inventaire datetime NOT NULL,
    quantite_inventorier int(20) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE Stock_Inventaire (
    id_stock int(11) NOT NULL,
    Date_inventaire datetime NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE Fournisseur (
    id_frs int(11) NOT NULL,
    nom_frs varchar(255) DEFAULT NULL,
    adresse_mail_frs varchar(255) DEFAULT NULL,
    ville_frs varchar(255) DEFAULT NULL,
    telephone_frs varchar(255) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE Commande_frs (
```

```sql
    id_cmd_frs int(11) NOT NULL,
    id_frs int(11) NOT NULL,
    Date_cmd_frs datetime NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE Produit_Commande_frs (
    id_cmd_frs int(11) NOT NULL,
    id_prod int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE Modele (
    num_modele varchar(255) NOT NULL,
    taille double DEFAULT NULL,
    stock_reserve int DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE Modele_Commande_frs (
    num_modele varchar(255) NOT NULL,
    id_cmd_frs int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE Pieces (
    id_pieces int(11) NOT NULL,
    num_modele varchar(255) NOT NULL,
    id_frs int(11) NOT NULL,
    prix_cmd decimal(19,3) DEFAULT NULL,
    delai_livraison_jr int DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE Payement_frs (
  id_payement int(11) NOT NULL,
  date_payement datetime DEFAULT NULL,
  Moyens_payement varchar(255) DEFAULT NULL,
  echeance_payement varchar(255) DEFAULT NULL,
  montant_payement decimal(19,3) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE Facture_Achat (
  id_fact_ach int(11) NOT NULL,
  id_frs int(11) NOT NULL,
  id_payement int(11) NOT NULL,
  num_fact_ach varchar(255) DEFAULT NULL,
  date_fact_ach datetime DEFAULT NULL,
  date_ach datetime DEFAULT NULL,
  montant_HT_ach decimal(19,3) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;


ALTER TABLE Client
  ADD PRIMARY KEY (id_clt);

ALTER TABLE Commande_clt
  ADD PRIMARY KEY (id_cmd_clt),
  ADD FOREIGN KEY (id_clt) REFERENCES Client (id_clt);

ALTER TABLE Remise
  ADD PRIMARY KEY (id_remise),
```

```sql
  ADD FOREIGN KEY (id_clt) REFERENCES Client (id_clt);

ALTER TABLE Reglement_clt
  ADD PRIMARY KEY (id_reglement);

ALTER TABLE Facture_vente
  ADD PRIMARY KEY (id_fact_vte),
  ADD FOREIGN KEY (id_clt) REFERENCES Client (id_clt),
  ADD FOREIGN KEY (id_reglement) REFERENCES Reglement_clt (id_reglement);

ALTER TABLE Livraison_clt
  ADD PRIMARY KEY (id_bon_livraison),
  ADD FOREIGN KEY (id_cmd_clt) REFERENCES Commande_clt (id_cmd_clt);

ALTER TABLE Taxe
  ADD PRIMARY KEY (id_tax);

ALTER TABLE Categorie
  ADD PRIMARY KEY (id_categorie);

ALTER TABLE Sous_Categorie
  ADD PRIMARY KEY (id_sous_categorie),
  ADD FOREIGN KEY (id_categorie) REFERENCES Categorie (id_categorie);

ALTER TABLE Produit
  ADD PRIMARY KEY (id_prod),
  ADD FOREIGN KEY (id_tax) REFERENCES Taxe (id_tax),
  ADD FOREIGN KEY (id_sous_categorie) REFERENCES Sous_Categorie (id_sous_categorie);

ALTER TABLE Ligne_livraison
  ADD FOREIGN KEY (id_bon_livraison) REFERENCES Livraison_clt (id_bon_livraison),
  ADD FOREIGN KEY (id_prod) REFERENCES Produit (id_prod);

ALTER TABLE Ligne_commande
  ADD FOREIGN KEY (id_cmd_clt) REFERENCES Commande_clt (id_cmd_clt),
  ADD FOREIGN KEY (id_prod) REFERENCES Produit (id_prod);

ALTER TABLE Fact_Vte_Prod
  ADD FOREIGN KEY (id_fact_vte) REFERENCES Facture_vente (id_fact_vte),
  ADD FOREIGN KEY (id_prod) REFERENCES Produit (id_prod);

ALTER TABLE Promotion_produit
  ADD PRIMARY KEY (id_promo_prod),
  ADD FOREIGN KEY (id_prod) REFERENCES Produit (id_prod);

ALTER TABLE Stock
  ADD PRIMARY KEY (id_stock);

ALTER TABLE Quantite_prod
  ADD FOREIGN KEY (id_prod) REFERENCES Produit (id_prod),
  ADD FOREIGN KEY (id_stock) REFERENCES Stock (id_stock);

ALTER TABLE Inventaire
  ADD PRIMARY KEY (date_inventaire);

```

```sql
ALTER TABLE Stock_Inventaire
  ADD FOREIGN KEY (date_inventaire) REFERENCES Inventaire (date_inventaire),
  ADD FOREIGN KEY (id_stock) REFERENCES Stock (id_stock);

ALTER TABLE Fournisseur
  ADD PRIMARY KEY (id_frs);

ALTER TABLE Commande_frs
  ADD PRIMARY KEY (id_cmd_frs),
  ADD FOREIGN KEY (id_frs) REFERENCES Fournisseur (id_frs);

ALTER TABLE Produit_Commande_frs
  ADD FOREIGN KEY (id_prod) REFERENCES Produit (id_prod),
  ADD FOREIGN KEY (id_cmd_frs) REFERENCES Commande_frs (id_cmd_frs);

ALTER TABLE Modele
  ADD PRIMARY KEY (num_modele);

ALTER TABLE Modele_Commande_frs
  ADD FOREIGN KEY (num_modele) REFERENCES Modele (num_modele),
  ADD FOREIGN KEY (id_cmd_frs) REFERENCES Commande_frs (id_cmd_frs);

ALTER TABLE Pieces
  ADD PRIMARY KEY (id_pieces),
  ADD FOREIGN KEY (num_modele) REFERENCES Modele (num_modele),
  ADD FOREIGN KEY (id_frs) REFERENCES Fournisseur (id_frs);

ALTER TABLE Payement_frs
  ADD PRIMARY KEY (id_payement);

ALTER TABLE Facture_Achat
  ADD PRIMARY KEY (id_fact_ach),
  ADD FOREIGN KEY (id_frs) REFERENCES Fournisseur (id_frs),
  ADD FOREIGN KEY (id_payement) REFERENCES Payement_frs (id_payement);
```

3 Queries

The databse can be manipulated by creating different queries.

3.1 Basic queries

1. Display the types of categorie of product

```
SELECT type_categorie
FROM categorie;
```

type_categorie
Terrasse et jardin
Terrasse et jardin
Terrasse et jardin
Terrasse et jardin
Terrasse et jardin
Decoration et eclairage
Decoration et eclairage
Decoration et eclairage
Decoration et eclairage
Decoration et eclairage
Salle de bains
Salle de bains
Salle de bains
Salle de bains
Salle de bains

2. Display all properties (characteristics) of produits

```
SELECT *
FROM produit;
```

id_prod	id_sous_categorie	id_tax	nom_prod	prix_prod	largeur_prod	hauteur_prod	Epaisseur_cadre	matiere_prod	type_panneau	couleur_prod	garantie_constructeur_an
1	5	4	Fontaine de jardin en pierre	159.000	42	76	NULL	Pierre reconstituée	NULL	Gris	1
2	2	1	Salon de jardin résine	990.000	500	NULL	50	Résine tressé	NULL	Marron	5
3	6	3	Suspension pour salon	29.900	NULL	NULL	NULL	Metal	NULL	Argent	1
4	9	2	Spot encadré connecté bluetooth	22.900	NULL	NULL	NULL	NULL	NULL	Blanc	1
5	1	5	Lame de terrasse bois	12.040	14.5	205	15.4	bois	NULL	beige	NULL
6	11	8	Meuble de salle de bains en bois	189.000	99	NULL	NULL	bois	NULL	Vert	5
7	4	6	Abri de jardin Metal	319.000	500	3000	500	Metal	NULL	Argent	10
8	14	7	Mitigeur de douche	89.900	NULL	NULL	NULL	NULL	NULL	Argent	2
9	12	9	Vasque à poser céramique rectangulaire	85.000	NULL	NULL	NULL	NULL	NULL	Blanc	NULL
10	10	15	Rideau occulatant	46.900	350	300	NULL	Polyester	NULL	Jaune	1
11	15	10	Cabine de douche	5761.800	100	NULL	NULL	Résine	NULL	Blanc	10

3. Display suppliers name, city, and phone

```
SELECT nom_frs, ville_frs, telephone_frs
FROM fournisseur;
```

nom_frs	ville_frs	telephone_frs
Garcia	Marseille	0625157825
Renaud	Paris	0614587573
Martin	Roubaix	0714152395
Dupont	Lille	07648886254
Dubois	Lyon	0625147820
Pages	Bordeaux	*NULL*
Fabre	Marseille	*NULL*
Roux	Bordeaux	0756421890
Laurenzo	Paris	*NULL*
Marty	Paris	0745846520
Solez	Bordeaux	0615428471
Oliver	Lille	*NULL*
Morel	Roubaix	*NULL*
Meyer	Bordeaux	0611254870
Faure	Lyon	*NULL*

4. Display without duplicate customer payment methods

```
SELECT DISTINCT Moyens_reglement
FROM reglement_clt;
```

Moyens_reglement
cartes
espèce
cheques

5. Display without duplicate the material of the products

```
SELECT DISTINCT matiere_prod
FROM produit;
```

matiere_prod
Pierre reconstituée
Résine tressé
Metal
NULL
bois
Polyester
Résine
béton

3.2 WHERE clause queries

1. Display the numbers and labels of products whose current price is higher than 100€

```sql
SELECT nom_prod, id_prod
FROM produit
WHERE prix_prod>100;
```

nom_prod	id_prod
Fontaine de jardin en pierre	1
Salon de jardin résine	2
Meuble de salle de bains en bois	6
Abri de jardin Metal	7
Cabine de douche hamman	11
Miroir avec éclairage intégré Diam	14

2. Display the list of products whose quantity is between 20 and 30.

```sql
SELECT nom_prod
FROM produit, quantite_prod
WHERE produit.id_prod=quantite_prod.id_prod AND quantite BETWEEN 20 and 30;
```

nom_prod
Salon de jardin résine
Meuble de salle de bains en bois
Cabine de douche hamman
Ampoule E27
cloture béton
Abri de jardin Metal

3. Display the labels of products sold whose sale amount is greater than 2

```sql
SELECT nom_prod
FROM produit, fact_vte_prod, facture_vente
WHERE produit.id_prod=fact_vte_prod.id_prod AND fact_vte_prod.id_fact_vte=
facture_vente.id_fact_vte AND montant_HT_vte>2;
```

nom_prod
Lame de terrasse bois
Salon de jardin résine
Vasque à poser céramique rectangulaire
Cabine de douche hamman
Rideau occulatant
Abri de jardin Metal
Mitigeur de douche

4. Display the labels and the price of the products offered for sale from 10 to 25

```sql
SELECT nom_prod, prix_prod
FROM produit, promotion_produit
WHERE produit.id_prod=promotion_produit.id_prod AND pourcent_promo BETWEEN
0.10 and 0.25;
```

nom_prod	prix_prod
Fontaine de jardin en pierre	159.000
Salon de jardin résine	990.000
Suspension pour salon	29.900
Spot encadré connecté bluetooth	22.900
Lame de terrasse bois	12.040
Meuble de salle de bains en bois	189.000
Abri de jardin Metal	319.000
Mitigeur de douche	89.900
Vasque à poser céramique rectangulaire	85.000
Rideau occulatant	46.900
Cabine de douche hamman	5761.800
Ampoule E27	11.900
Applique Extérieur module LED	29.900
Miroir avec éclairage intégré Diam	240.000
cloture béton	29.900
Fontaine de jardin en pierre	159.000
Salon de jardin résine	990.000
Suspension pour salon	29.900
Spot encadré connecté bluetooth	22.900
Lame de terrasse bois	12.040
Meuble de salle de bains en bois	189.000
Abri de jardin Metal	319.000
Mitigeur de douche	89.900
Vasque à poser céramique rectangulaire	85.000
Rideau occulatant	46.900

5. Display model numbers with in-stock quantities less than 50

```
SELECT num_modele
FROM modele
WHERE stock_reserve<50
```

num_modele
014A
124A
145S

3.3 ORDER BY queries

1. Display prices in ascending format

```
SELECT *
FROM produit
ORDER BY prix_prod ASC;
```

id_prod	id_sous_categorie	id_tax	nom_prod	prix_prod ▲ 1	largeur_prod	hauteur_prod	Epaisseur_cadre	matiere_prod	type_panneau	couleur_prod
12	7	11	Ampoule E27	11.900	NULL	NULL	NULL	NULL	NULL	blanc
5	1	5	Lame de terrasse bois	12.040	14.5	205	15.4	bois	NULL	beige
4	9	2	Spot encadré connecté bluetooth	22.900	NULL	NULL	NULL	NULL	NULL	Blanc
3	6	3	Suspension pour salon	29.900	NULL	NULL	NULL	Metal	NULL	Argent
13	8	12	Applique Extérieur module LED	29.900	NULL	NULL	NULL	NULL	NULL	Noir
15	3	13	cloture béton	29.900	0.9	240	1.7	béton	NULL	noir
10	10	15	Rideau occulatant	46.900	350	300	NULL	Polyester	NULL	Jaune
9	12	9	Vasque à poser céramique rectangulaire	85.000	NULL	NULL	NULL	NULL	NULL	Blanc
8	14	7	Mitigeur de douche	89.900	NULL	NULL	NULL	NULL	NULL	Argent
1	5	4	Fontaine de jardin en pierre	159.000	42	76	NULL	Pierre reconstituée	NULL	Gris
6	11	8	Meuble de salle de bains en bois	189.000	99	NULL	NULL	bois	NULL	Vert
14	13	14	Miroir avec éclairage intégré Diam	240.000	128	128	NULL	NULL	Sensitif	Blanc
7	4	6	Abri de jardin Metal	319.000	500	3000	500	Metal	NULL	Argent
2	2	1	Salon de jardin résine	990.000	500	NULL	50	Résine tressé	NULL	Marron
11	15	10	Cabine de douche hamman	5761.800	100	NULL	NULL	Résine	NULL	Blanc

2. Display the discount percentage in a descending format

```
SELECT *
FROM remise
ORDER BY pourcent_remise DESC;
```

id_remise	id_clt	pourcent_remise ▼ 1
11	15	0.24
5	11	0.24
3	6	0.22
9	12	0.21
8	10	0.2
10	14	0.19
15	3	0.19
2	5	0.18
13	8	0.18
6	1	0.15
4	9	0.15
12	7	0.15
1	2	0.15
14	13	0.12
7	4	0.11

3. Display the promotion percentages in descending format

```
SELECT *
FROM promotion_produit
ORDER BY pourcent_promo DESC;
```

id_promo_prod	id_prod	pourcent_promo 1
15	3	0.49
2	5	0.48
7	4	0.31
12	7	0.25
1	2	0.25
11	15	0.24
5	11	0.24
3	6	0.22
9	12	0.21
8	10	0.2
10	14	0.19
13	8	0.18
4	9	0.15
6	1	0.15
14	13	0.12

4. Display settlement amounts in ascending format

```
SELECT *
FROM reglement_clt
ORDER BY montant_reglement ASC;
```

id_reglement	date_reglement	Moyens_reglement	echeance_reglement	montant_reglement 1
10	2014-03-24 00:00:00	cheques	2 mois	0.890
4	2014-06-21 00:00:00	cartes	5 mois	1.590
1	2014-01-15 00:00:00	cartes	3 mois	1.800
14	2014-03-13 00:00:00	cheques	11 mois	1.840
7	2014-06-18 00:00:00	cartes	7 mois	1.850
9	2014-03-20 00:00:00	cartes	1 mois	1.880
3	2014-03-27 00:00:00	cartes	3 mois	1.890
12	2014-06-16 00:00:00	cartes	6 mois	2.400
8	2014-08-09 00:00:00	espèce	4 mois	2.450
2	2014-03-12 00:00:00	espèce	2 mois	2.970
5	2014-03-18 00:00:00	espèce	6 mois	3.340
6	2014-08-11 00:00:00	cheques	7 mois	3.990
13	2014-08-11 00:00:00	espèce	1 mois	3.990
15	2014-08-15 00:00:00	cheques	12 mois	4.290
11	2014-04-20 00:00:00	cartes	5 mois	5.450

5. Display the inventoried quantities in ascending format

```
SELECT *
FROM inventaire
ORDER BY quantite_inventorier ASC;
```

Date_inventaire	quantite_inventorier ▲ 1
2014-08-09 00:00:00	2
2014-03-13 00:00:00	5
2014-08-11 00:00:00	10
2014-08-10 00:00:00	10
2014-06-20 00:00:00	15
2014-03-23 00:00:00	15
2014-08-19 00:00:00	17
2014-06-16 00:00:00	20
2014-03-27 00:00:00	20
2014-03-21 00:00:00	20
2014-01-15 00:00:00	20
2014-03-17 00:00:00	22
2014-06-17 00:00:00	30
2014-03-14 00:00:00	35
2014-03-12 00:00:00	45

3.4 Multi-table queries

1. Display customers who have ordered and received an invoice in alphabetical order

```sql
SELECT nom_clt, prenom_clt, date_cmd, date_livraison, date_fact_vte
FROM client c , commande_clt cc, facture_vente f
WHERE c.id_clt=cc.id_clt AND c.id_clt=f.id_clt
ORDER BY nom_clt;
```

nom_clt ▲ 1	prenom_clt	date_cmd	date_livraison	date_fact_vte
Alpha	Barry	2014-08-09 00:00:00	2014-08-19 00:00:00	2014-06-18 00:00:00
Bernard	louis	2014-03-12 00:00:00	2014-03-26 00:00:00	2014-03-27 00:00:00
Caron	chloe	2014-03-24 00:00:00	2014-04-06 00:00:00	2014-08-19 00:00:00
Diallo	marion	2014-08-11 00:00:00	2014-08-16 00:00:00	2014-08-11 00:00:00
Egon	deyaert	2014-01-15 00:00:00	2014-01-24 00:00:00	2014-08-15 00:00:00
Fleury	adam	2014-08-15 00:00:00	2014-08-20 00:00:00	2014-03-24 00:00:00
Gautier	jade	2014-08-11 00:00:00	2014-08-17 00:00:00	2014-03-12 00:00:00
Girard	rose	2014-06-21 00:00:00	2014-06-28 00:00:00	2014-08-09 00:00:00
Jacquet	mael	2014-04-20 00:00:00	2014-04-27 00:00:00	2014-03-13 00:00:00
kall	matt	2014-03-18 00:00:00	2014-04-02 00:00:00	2014-01-16 00:00:00
Laurent	leo	2014-03-27 00:00:00	2014-04-06 00:00:00	2014-06-21 00:00:00
Morel	hugo	2014-06-18 00:00:00	2014-06-26 00:00:00	2014-03-21 00:00:00
Petit	alice	2014-06-16 00:00:00	2014-06-30 00:00:00	2014-06-16 00:00:00
Renaud	ambre	2014-03-20 00:00:00	2014-03-27 00:00:00	2014-03-18 00:00:00
Roux	mia	2014-03-13 00:00:00	2014-03-21 00:00:00	2014-08-11 00:00:00

2. Display orders and invoices placed with suppliers that are under 2.000

```sql
SELECT *
FROM commande_frs cf, facture_achat fa
WHERE cf.id_frs=fa.id_frs AND fa.montant_HT_ach<2.000;
```

id_cmd_frs	id_frs	Date_cmd_frs	id_fact_ach	id_frs	id_payement	num_fact_ach	date_fact_ach	date_ach	montant_HT_ach
2	2	2014-03-12 00:00:00	1	2	11	0012	2014-01-16 00:00:00	2014-01-15 00:00:00	1.604
3	4	2014-03-27 00:00:00	7	4	1	0107	2014-06-18 00:00:00	2014-06-17 00:00:00	1.840
8	3	2014-08-09 00:00:00	15	3	8	0001	2014-08-15 00:00:00	2014-03-14 00:00:00	0.560
9	7	2014-08-19 00:00:00	12	7	14	0156	2014-06-16 00:00:00	2014-06-16 00:00:00	1.330
10	15	2014-03-23 00:00:00	11	15	12	0109	2014-03-21 00:00:00	2014-03-21 00:00:00	1.850
12	14	2014-06-16 00:00:00	10	14	15	0108	2014-03-24 00:00:00	2014-03-23 00:00:00	1.450

3. Display orders to suppliers with their information, and if the supplier comes from Marseille

```
SELECT *
FROM commande_frs cf, fournisseur f
WHERE cf.id_frs=f.id_frs AND f.ville_frs='Marseille';
```

id_cmd_frs	id_frs	Date_cmd_frs	id_frs	nom_frs	adresse_mail_frs	ville_frs	telephone_frs
7	1	2014-06-17 00:00:00	1	Garcia	garcia@gmail.com	Marseille	0625157825
9	7	2014-08-19 00:00:00	7	Fabre	fabre@gmail.com	Marseille	*NULL*

4. Display where product quantities are below the minimum stock

```
SELECT *
FROM quantite_prod q, stock s, stock_inventaire i
WHERE q.id_stock=s.id_stock AND q.id_stock=i.id_stock AND q.quantite<s.
stock_min;
```

id_stock	id_prod	quantite	id_stock	stock_min	stock_max	id_stock	Date_inventaire
2	5	48	2	50	4800	2	2014-03-12 00:00:00
3	6	22	3	60	2200	3	2014-08-09 00:00:00
4	9	15	4	90	1500	4	2014-03-27 00:00:00
5	11	24	5	110	2400	5	2014-03-17 00:00:00
6	1	15	6	100	1500	6	2014-08-10 00:00:00
7	4	31	7	40	3100	7	2014-08-19 00:00:00
8	10	2	8	100	2000	8	2014-03-14 00:00:00
9	12	21	9	120	20010	9	2014-03-13 00:00:00
10	14	19	10	140	1900	10	2014-06-20 00:00:00
11	15	24	11	150	2400	11	2014-01-15 00:00:00
12	7	25	12	70	2500	12	2014-03-21 00:00:00
13	8	18	13	80	1800	13	2014-08-11 00:00:00
14	13	12	14	130	1200	14	2014-06-16 00:00:00
15	3	49	15	151	4900	15	2014-03-23 00:00:00

5. Display products with a percentage of promotions between 25% and 50%.

```
SELECT nom_prod, prix_prod, pourcent_promo
FROM promotion_produit pp, produit pd
WHERE pp.id_prod = pd.id_prod AND pourcent_promo BETWEEN 0.25 AND 0.50;
```

nom_prod	prix_prod	pourcent_promo
Salon de jardin résine	990.000	0.25
Lame de terrasse bois	12.040	0.48
Spot encadré connecté bluetooth	22.900	0.31
Abri de jardin Metal	319.000	0.25
Suspension pour salon	29.900	0.49

3.5 Queries with Numeric expressions and functions

1. Display the number of products offered for sale

```
SELECT COUNT(*)
FROM produit;
```

COUNT(*)
15

2. Display for each product the label, the current price and the current price reduced by 10

```sql
SELECT nom_prod, prix_prod, prix_prod*0.9 as prix_prod_r duit
FROM produit;
```

nom_prod	prix_prod	prix_prod_réduit
Fontaine de jardin en pierre	159.000	143.1000
Salon de jardin résine	990.000	891.0000
Suspension pour salon	29.900	26.9100
Spot encadré connecté bluetooth	22.900	20.6100
Lame de terrasse bois	12.040	10.8360
Meuble de salle de bains en bois	189.000	170.1000
Abri de jardin Metal	319.000	287.1000
Mitigeur de douche	89.900	80.9100
Vasque à poser céramique rectangulaire	85.000	76.5000
Rideau occulatant	46.900	42.2100
Cabine de douche hamman	5761.800	5185.6200
Ampoule E27	11.900	10.7100
Applique Extérieur module LED	29.900	26.9100
Miroir avec éclairage intégré Diam	240.000	216.0000
cloture béton	29.900	26.9100

3. Display the total amount of product sales

```sql
SELECT SUM(montant_HT_vte)
FROM facture_vente;
```

sum(montant_HT_vte)
35.197

4. Displays the date of purchase and the name of the supplier of the products ordered by amount HT in descending order

```sql
SELECT date_fact_ach, montant_HT_ach, nom_frs
FROM facture_achat, fournisseur
WHERE facture_achat.id_frs=fournisseur.id_frs
ORDER BY montant_HT_ach DESC;
```

date_fact_ach	montant_HT_ach ▼ 1	nom_frs
2014-08-11 00:00:00	6.456	Garcia
2014-03-27 00:00:00	5.502	Pages
2014-08-19 00:00:00	3.995	Oliver
2014-03-18 00:00:00	3.840	Solez
2014-03-12 00:00:00	3.415	Dubois
2014-08-09 00:00:00	2.880	Marty
2014-03-13 00:00:00	2.684	Morel
2014-08-11 00:00:00	2.560	Roux
2014-06-21 00:00:00	2.250	Laurenzo
2014-03-21 00:00:00	1.850	Faure
2014-06-18 00:00:00	1.840	Dupont
2014-01-16 00:00:00	1.604	Renaud
2014-03-24 00:00:00	1.450	Meyer
2014-06-16 00:00:00	1.330	Fabre
2014-08-15 00:00:00	0.560	Martin

5. Display the average stock quantity of products

```sql
SELECT AVG(quantite) as quantite_moyenne
FROM quantite_prod;
```

quantite_moyenne
23.3333

3.6 GROUP BY queries

1. Display the number of customers in each city

```
SELECT  ville_clt , COUNT(*)
FROM client
GROUP BY ville_clt;
```

ville_clt	COUNT(*)
Bordeaux	4
Lille	2
Lyon	2
Marseille	2
Paris	3
Roubaix	2

2. Display the average price of products from the same production material

```
SELECT matiere_prod, AVG(prix_prod)
FROM produit
GROUP BY matiere_prod;
```

matiere_prod	AVG(prix_prod)
NULL	79.9333333
béton	29.9000000
bois	100.5200000
Metal	174.4500000
Pierre reconstituée	159.0000000
Polyester	46.9000000
Résine	5761.8000000
Résine tressé	990.0000000

3. Display the total sum of payments made by means of payment

```
SELECT Moyens_payement, SUM(montant_payement)
FROM payement_frs
GROUP BY Moyens_payement;
```

Moyens_payement	SUM(montant_payement)
cartes	21.860
cheques	10.010
espèce	10.750

4. Display the different types of taxes applied with their number

```
SELECT libelle_tax, COUNT(*)
FROM taxe
GROUP BY libelle_tax;
```

libelle_tax	COUNT(*)
impôt sur le Revenu	2
impôt sur le Revenu (IR)	1
impôt sur les Sociétés	2
La Flat Tax	1
La taxe foncière	3
La taxe habitation	4
TVA	2

5. Display the total sum of payments made by means of payment

```
SELECT Moyens_reglement, SUM(montant_reglement)
FROM reglement_clt
GROUP BY Moyens_reglement;
```

Moyens_reglement	SUM(montant_reglement)
cartes	16.860
cheques	11.010
espèce	12.750

3.7 Nested queries

1. Display the most expensive product

```
SELECT nom_prod, prix_prod
FROM produit
WHERE prix_prod=(SELECT MAX(prix_prod)FROM produit);
```

nom_prod	prix_prod
Cabine de douche hamman	5761.800

2. Display products that are below the minimum stock

```
SELECT *
FROM quantite_prod q, stock_inventaire i
WHERE q.id_stock=i.id_stock AND q.quantite<(SELECT MIN(stock_min) FROM stock)
;
```

id_stock	id_prod	quantite	id_stock	Date_inventaire
4	9	15	4	2014-03-27 00:00:00
6	1	15	6	2014-08-10 00:00:00
8	10	2	8	2014-03-14 00:00:00
10	14	19	10	2014-06-20 00:00:00
13	8	18	13	2014-08-11 00:00:00
14	13	12	14	2014-06-16 00:00:00

3. Display orders with a delivery time per day greater than the average delivery time per day

```
SELECT delai_livraison_jr, id_frs, id_pieces
FROM pieces
WHERE delai_livraison_jr >(SELECT AVG(delai_livraison_jr) FROM pieces);
```

delai_livraison_jr	id_frs	id_pieces
21	1	1
45	10	10
23	15	15

4. Display product names that are priced lower than the concrete fence

```
SELECT nom_prod, prix_prod
FROM produit
WHERE prix_prod<(SELECT prix_prod FROM produit WHERE nom_prod='cloture beton'
);
```

nom_prod	prix_prod
Spot encadré connecté bluetooth	22.900
Lame de terrasse bois	12.040
Ampoule E27	11.900

5. Display product names with a price above 100.000

```
SELECT nom_prod, prix_prod
FROM produit
WHERE nom_prod IN (SELECT DISTINCT nom_prod FROM produit WHERE prix_prod
>100.000);
```

nom_prod	prix_prod
Fontaine de jardin en pierre	159.000
Salon de jardin résine	990.000
Meuble de salle de bains en bois	189.000
Abri de jardin Metal	319.000
Cabine de douche hamman	5761.800
Miroir avec éclairage intégré Diam	240.000

4 MongoDB

We can now manipulate various non-relational data queries, using MongoDB.

For this, we have chosen the following three entities:

- Client
- Reglement_Clt
- Stock

Consequently, we obtained the JSON files of the 3 respective entities via an export on phpMyAdmin.

In order to carry out the script of insertions of the three Collections/Entities we have adopted the following approach:

```
sudo systemctl status mongod  #pour le demarrage du serveur mongo
mongo   # pour rentrer dans la console mongo
show dbs
use LeroyMerlin   #cree la base de donn es mongod nommee LeroyMerlin
exit # pour sortir de la console mongo

# Execution du script d'insertion pour importer les trois fichiers json des trois collections respectivement

# Script d'insertion

# -d pour dire database, -c pour le nom de la Collection, -- type pour le type du fichier    importer, --file suivi du chemin d'acces au fichier (ici dans notre cas le fichier est dans le dossier Documents)

mongoimport -d LeroyMerlin -c Client --type json --file ~/Documents/client.json

mongoimport -d LeroyMerlin -c Reglement_clt --type json --file ~/Documents/reglement_clt.json

mongoimport -d LeroyMerlin -c Stock --type json --file ~/Documents/client.json

#verification de la creation de la base de donnees LeroyMerlin et l'importation des 3 collections

mongo  # pour rentrer dans la console mongo
show dbs  # pour montrer tous les base de donn es

use LeroyMerlin  # utiliser la base de donnees LeroyMerlin

show collections # afficher les collections contenues dans la base de donn es LeroyMerlin

exit # sortir du console mongo
```

```python
# Installation de python3, pip et pymongo

from pymongo import MongoClient

client = MongoClient()

db = client.LeroyMerlin

# Display all records of the collection Reglement_clt
cursor = db.Reglement_clt
for document in cursor:
    print(document)

#################################### Requete 1 ####################################
# Display all records of the collection Reglement_clt whose payment method is "cards".
cursor = db.Reglement_clt.find({"Moyens_reglement": "cartes"})
for document in cursor:
    print(document)

#################################### Requete 2 ####################################
# Display all records of the collection Reglement_clt whose payment amount is greater than 2.000
cursor = db.Reglement_clt.find({"montant_reglement": {"$gt": "2.000"}})
for document in cursor:
    print(document)
#################################### Requete 3 ####################################
# Display all records of the collection Reglement_clt whose payment amount is less than 3.000
cursor = db.Reglement_clt.find({"montant_reglement": {"$lt": "3.000"}})
for document in cursor:
    print(document)
#################################### Requete 4 ####################################
# Change the payment method of the record whose id is 1 to "Cash".
result = db.Reglement_clt.update_one(
    {"id_reglement": "1"}, {"$set": {"Moyens_reglement": "esp ce"}}
)
result.matched_count
#################################### Requete 5 ####################################
# Display the grouping by means of payment and theirs respectives counts
cursor = db.Reglement_clt.aggregate(
    [{"$group": {"_id": "$Moyens_reglement", "count": {"$sum": 1}}}]
)
for document in cursor:
    print(document)
```

5 Conclusion

Thanks to various SQL and NoSQL queries we were able to analyse the database and manage the different products and stocks of the company LeroyMerlin.

However, these databases have some limitations on stock management. These databases do not take into account the budget of a company, for example the working capital requirement (BFR in french), which corresponds to the money that the company needs permanently to finance its operations. But also the Global Net Working Capital (FRNG in french), which represents the amount of money the company has available to pay its suppliers, employees and all its operating expenses, while waiting to be paid by its customers. These two measures allow the proper functioning of a company's cash flow by taking into account inventory management.

Moreover, managing databases in SQL or NoSQL requires having relational or non-relational databases in a company that is not too large. Since the larger these databases are, the slower the process will be to execute, which can waste a company's time.