

PROJET GM1 - DOSSIER

Modélisation des données

Élèves :

BURIE Aurélien
LAFAGE Adrien
LEROUX Louis-Clément
PARAU Emmanuel

Professeur :

Renaud VÉRIN



4 novembre 2018

1 Introduction

L'objectif de ce rapport est de vous présenter la façon dont nous avons modlisé les données fournies par le client VDSA. En effet, on nous a fourni un tableau contenant l'ensemble des données concernant les clients de l'entreprise. Nous avons donc créé pour cela plusieurs entités dont les relations vous seront explicités par le biais d'un modèle conceptuel et d'un modèle logique de nos données. Finalement, vous trouverez le code SQL correspondant.

Table des matières

1	Introduction	1
2	Modèle conceptuel	3
3	Modèle logique	5
4	Code source SQL	6

2 Modèle conceptuel

Les données fournies sont de la forme suivante :

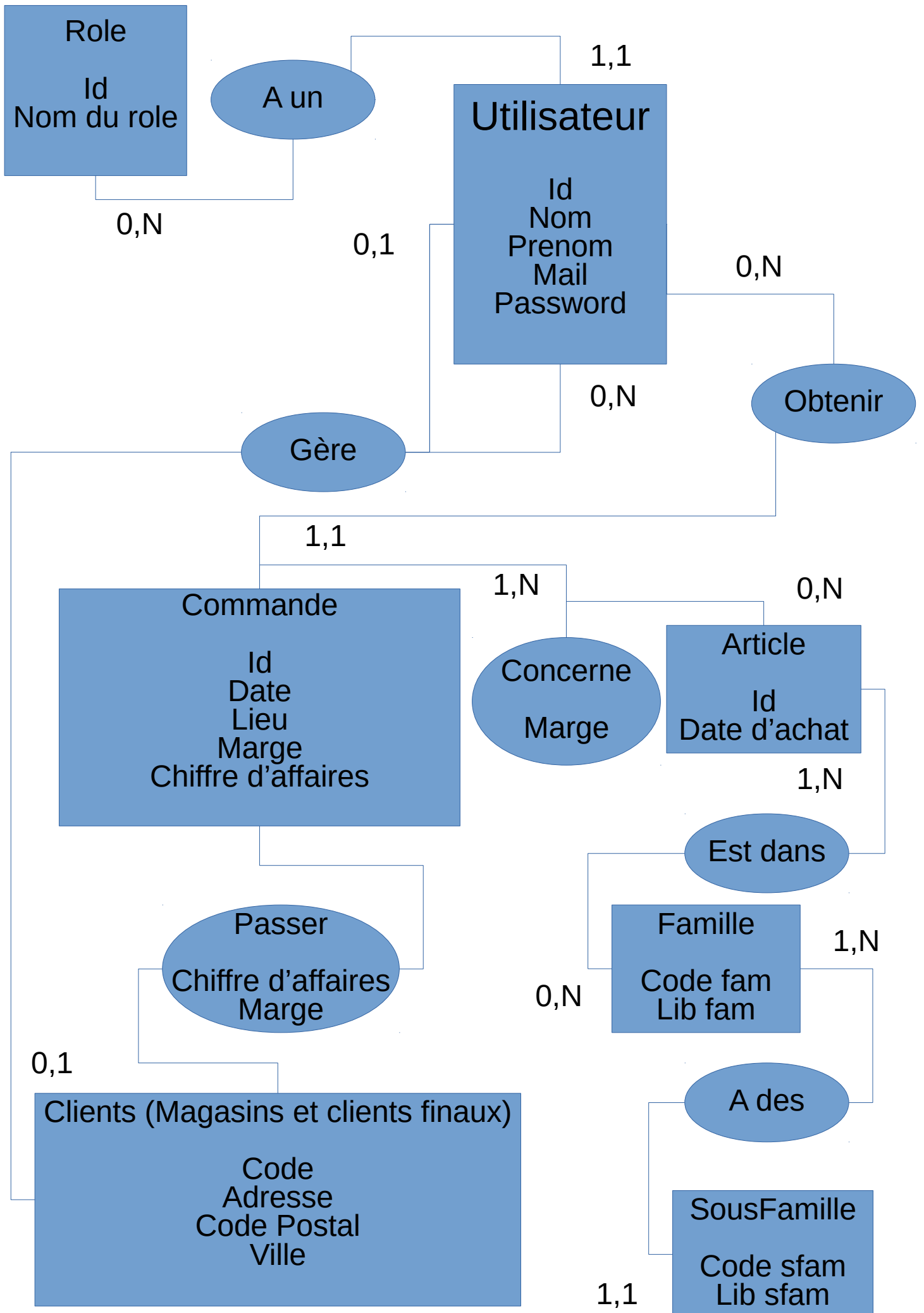
A	B	C	D	E	F	G	H	I
Code Rep	Code Clt	lib famille	lib sfam	cp cli liv	ville cli liv	date doc	Vte Facturées	Marge %(BA)
5	V00666	EQUIPEMENT ATELIER	CONSOMABLE			18/05/2011	54683.14	4.76
1	V01283	EQUIPEMENT ATELIER	MATERIEL	95420	MAGNY EN VEXIN	27/12/2011	45798	7.06
5	V00666	CARROSSERIE, PEINTURE, VITRAGE	ABRASIF, MASQUAGE			22/11/2011	34487.78	0
1	V00425	EQUIPEMENT ATELIER	MATERIEL	95310	ST OUEN L'AUMONE	23/11/2011	18558.31	36.84
1	V01219	EQUIPEMENT ATELIER	MATERIEL	95310	ST OUEN L'AUMONE	31/10/2011	12589.80	24.54
8	V01136	COMPTABILITE	DIVERS COMPTABILITE	95650	PUISEUX PONTOISE	31/12/2011	11556.73	1
1	V00208	EQUIPEMENT ATELIER	MATERIEL	78700	CONFLANS STE HONORIN	20/12/2011	10177	17.78
14	V01223	EQUIPEMENT ATELIER	MATERIEL	95330	DOMONT	25/10/2011	8780	20
8	V01156	COMPTABILITE	DIVERS COMPTABILITE	.	.	27/06/2011	8645	83.69
8	V01136	COMPTABILITE	DIVERS COMPTABILITE	95650	PUISEUX PONTOISE	27/02/2011	7434.50	0
1	V01219	EQUIPEMENT ATELIER	OUTILLAGE	95310	ST OUEN L'AUMONE	31/10/2011	6845.87	21.94
1	V01283	EQUIPEMENT ATELIER	MATERIEL	95420	MAGNY EN VEXIN	07/12/2011	6666.75	20.16
6	V00228	DIVERS	DIVERS	95300	ENNERY	14/11/2011	6120	50
6	V00979	EQUIPEMENT ATELIER	OUTILLAGE	95450	VIGNY	05/04/2011	6005.35	23.39
6	V00228	DIVERS	DIVERS	95300	ENNERY	20/01/2011	5465.60	64.29

FIGURE 1 – Exemple de données fournies par VDSA

De celles-ci nous en avons déduits les entités : Client, Commande, Article, Famille, Sous-Famille. Un client possède un code (Code Clt), une adresse, un code postal (cp cli liv) et une ville (ville cli liv). Une commande est composée d'un identifiant, d'une date (date doc), d'un lieu, d'une marge (Marge%(BA)) et d'un chiffre d'affaire. Un article a un identifiant et une date d'achat. Une famille possède une code pour l'identifier et d'un libellé (lib famille). Une sous-famille comprend un code pour l'identifier et d'un libellé (lib sfam).

De plus, notre site devra gérer des comptes utilisateur. Ainsi nous avons en plus créé l'entité Utilisateur. Celle-ci est constituée d'un identifiant, d'un nom, d'un prénom, d'un mail et d'un mot de passe. L'utilisateur pourra avoir un rôle qui lui est associé (exemple : administrateur). Soit le rôle fera parti de l'entité Utilisateur, soit nous créerons l'entité Rôle. Celle-ci sera composée d'un identifiant et d'un nom.

Vous trouverez sur la page suivante le modèle conceptuel de nos données.



3 Modèle logique

En utilisant le modèle conceptuel de nos données nous en déduisons le modèle logique suivant :

- Utilisateur(id, nom, prenom, mail, password, role*)
- Client(code, adresse, code postal, ville, #id_utilisateur)
- Commande(id, date, lieu, margeTotal, chiffre d'affaires, #id_utilisateur)
- Composition(#id_commande, #id_article, marge)
- Article(id, dateAchat, #codeFam_famille)
- Famille(codeFam, libFam)
- SousFamille(codeSfam, libSfam, #codeFam_famille)

** : Nous nous réservons la possibilité de supprimer ce paramètre d'Utilisateur et de créer à la place l'entité : Role(id, nomRole, #id_utilisateur)*

Notons par ailleurs que nous pourrions effectuer les modifications suivantes :

- Modifier la cardinalité de la relation entre un article et une famille telle que article(1,1)→famille.
- Décomposer l'adresse du client. En effet dans un souci de normalisation nous pourrions la diviser en un numéro et un nom de rue.

4 Code source SQL

Voici comment nous créons nos entités en SQL :

- Utilisateur :

```
DROP TABLE IF EXISTS utilisateur;  
CREATE TABLE utilisateur(  
    id INT PRIMARY KEY,  
    nom VARCHAR(30) ,  
    prenom VARCHAR(30) ,  
    email VARCHAR(50) ,  
    password VARCHAR(30) ,  
    role VARCHAR(30) ,  
    CONSTRAINT password CHECK (LENGTH(password) BETWEEN 8  
        AND 99) ,  
);
```

- Client :

```
DROP TABLE IF EXISTS client;  
CREATE TABLE client(  
    code INT PRIMARY KEY,  
    adresse VARCHAR(50) ,  
    codePostal INT,  
    ville VARCHAR(30) ,  
    idUtilisateur INT,  
    FOREIGN KEY fk_utilisateur(idUtilisateur) REFERENCES  
        utilisateur(id)  
);
```

- Commande :

```
DROP TABLE IF EXISTS commande;  
CREATE TABLE commande(  
    id INT PRIMARY KEY,  
    date VARCHAR(15) ,  
    lieu VARCHAR(30) ,  
    margeTotal INT,  
    chiffreAffaire INT,  
    idUtilisateur INT,  
    FOREIGN KEY fk_utilisateur(idUtilisateur) REFERENCES  
        utilisateur(id)  
);
```

- Famille :

```
DROP TABLE IF EXISTS famille ;
CREATE TABLE famille (
    codeFam INT PRIMARY KEY,
    libFam VARCHAR(30)
);
```

- Sous-Famille :

```
DROP TABLE IF EXISTS sousFamille;
CREATE TABLE sousFamille(
    codeSfam INT PRIMARY KEY,
    libSfam VARCHAR(30) ,
    idFam INT,
    FOREIGN KEY fk_famille(idFam) REFERENCES famille(
        codeFam)
);
```

- Article :

```
DROP TABLE IF EXISTS article ;
CREATE TABLE article (
    id INT PRIMARY KEY,
    dateAchat VARCHAR(30) ,
    idFam INT,
    FOREIGN KEY fk_famille(idFam) REFERENCES famille(
        codeFam)
);
```

- Composition :

```
DROP TABLE IF EXISTS composition;
CREATE TABLE composition(
    marge INT,
    idCommande INT,
    idArticle INT,
    FOREIGN KEY fk_commande(idCommande) REFERENCES commande
        (id) ,
    FOREIGN KEY fk_articler(idArticle) REFERENCES article(
        id)
);
```

Si nous choisissons finalement de créer une entité Role, on aura :

```
DROP TABLE IF EXISTS role ;
CREATE TABLE role (
    id INT PRIMARY KEY,
    nomRole VARCHAR(30) ,
```



```
    idUtilisateur INT,  
    FOREIGN KEY fk_utilisateur(idUtilisateur) REFERENCES  
        utilisateur(id)  
);
```