

2021

Chocolaterie Bourguignonne



Nom de l'entreprise :

Nom du projet :

ChocoCommerce

Travail effectué par M.
Jaufret
Bourguet

Adresse :

Bd du docteur PetitJean
Dijon



Table des matières

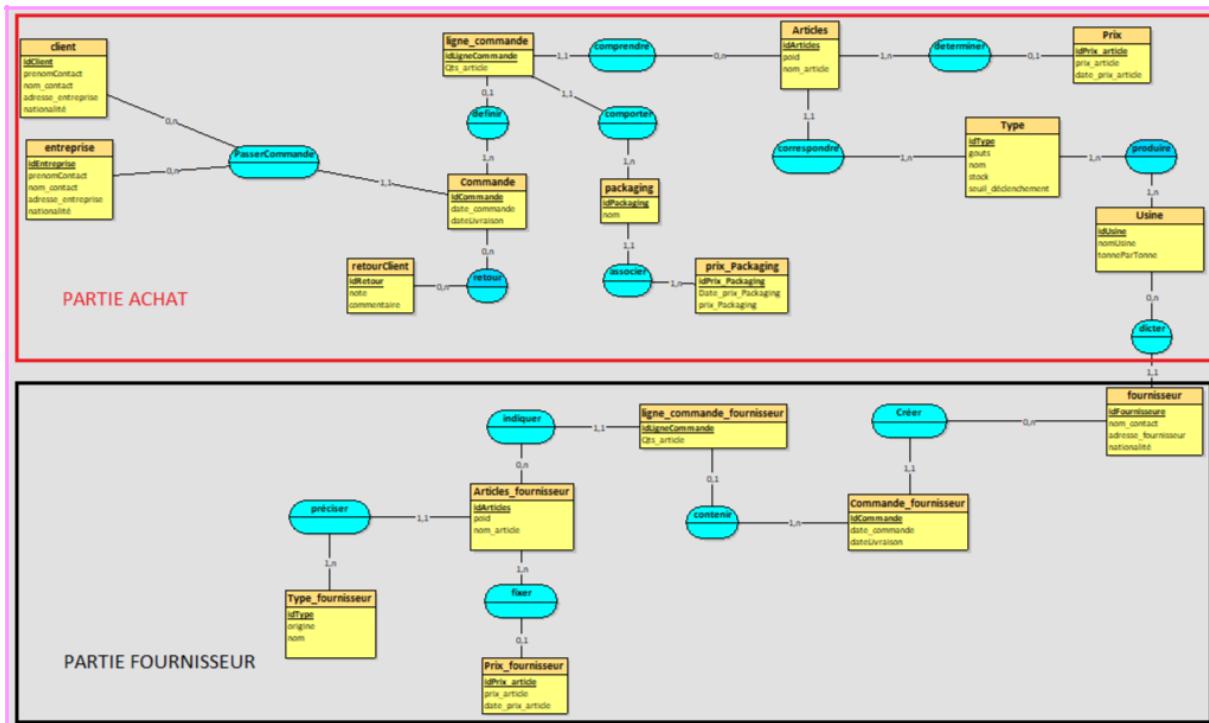
A.	Introduction du MCD (Modèle conceptuel des données)	2
I.	EXPLICATION DU MCD.....	2
B.	INTRODUCTION DU MLD (Modèles logique des données)	4
II.	REPRESENTATION TEXTUELLE DU MLD PARTIE ACHAT	4
III.	REPRESENTATION TEXTUELLE DU MLD PARTIE FOURNISSEUR	5
C.	SCRIPT PARTIE ACHAT	5
D.	SCRIPT PARTIE FOURNISSEUR.....	8
E.	LES REQUETES	9



A. Introduction du MCD (Modèle conceptuel des données)

Nous avons réalisé notre mcd à partir du logiciel looping qui nous a permis d'intégrer directement nos cardinalités.

Notre MCD est décomposé en deux parties, une partie achat en rouge et une partie sur les fournisseurs en noir.



I. EXPLICATION DU MCD

La partie achat du mcd représente la commande du client. C'est-à-dire tout ce qui va être traité par l'entreprise, quantité de produits commandés en passant par le prix de chaque article.

La table CLIENT ainsi que la table ENTREPRISE nous permettent de recueillir les informations sur les entreprises qui vont commander des articles (CLIENT) et ceux qui vont produire (ENTREPRISE). Il ne faut pas oublier que ce sont des entreprises qui communiquent entre elle et que dans les deux tables des informations enregistrées concerteront des entreprises et non des particuliers.

La table COMMANDE nous permet d'obtenir la date de commande ainsi que la date de livraison qui permettra de rédiger la facture, et de pouvoir s'occuper de retour éventuel.

Différents types de packaging sont disponibles et sont distingués à l'aide de la table PACKAGING qui nous permet de connaître le type de packaging choisi. Nous avons choisi de mettre en place une table PRIX_PACKAGING qui nous permet tout simplement d'enregistrer les différents prix mais aussi les changements de prix éventuels qui peuvent être opérés au cours de l'année.

Notre table ARTICLES permet de référencer nos différents produits disponibles à l'aide d'un nom spécifique et un poids. Chacun de nos articles est déterminé par une table PRIX, le choix de séparer le prix de la table ARTICLES nous permet de changer facilement nos prix en cours d'années et de pouvoir appliquer des promotions en ayant un historique de nos différents prix enregistrés.

La table TYPE permet de classer nos produits pour simplifier la gestion des stocks en fonction de chaque article.

La liaison entre notre partie achat se fait à l'aide de la table USINE qui est caractérisé par un nom d'usine ainsi que sa capacité de production.

Ensuite la partie fournisseur du MCD représente les données entre l'entreprise productrice et le fournisseur de matières premières, tels que le cacao. L'entreprise fera appel au fournisseur quand ses stocks de matières premières seront trop faibles par rapport à la demande du client.

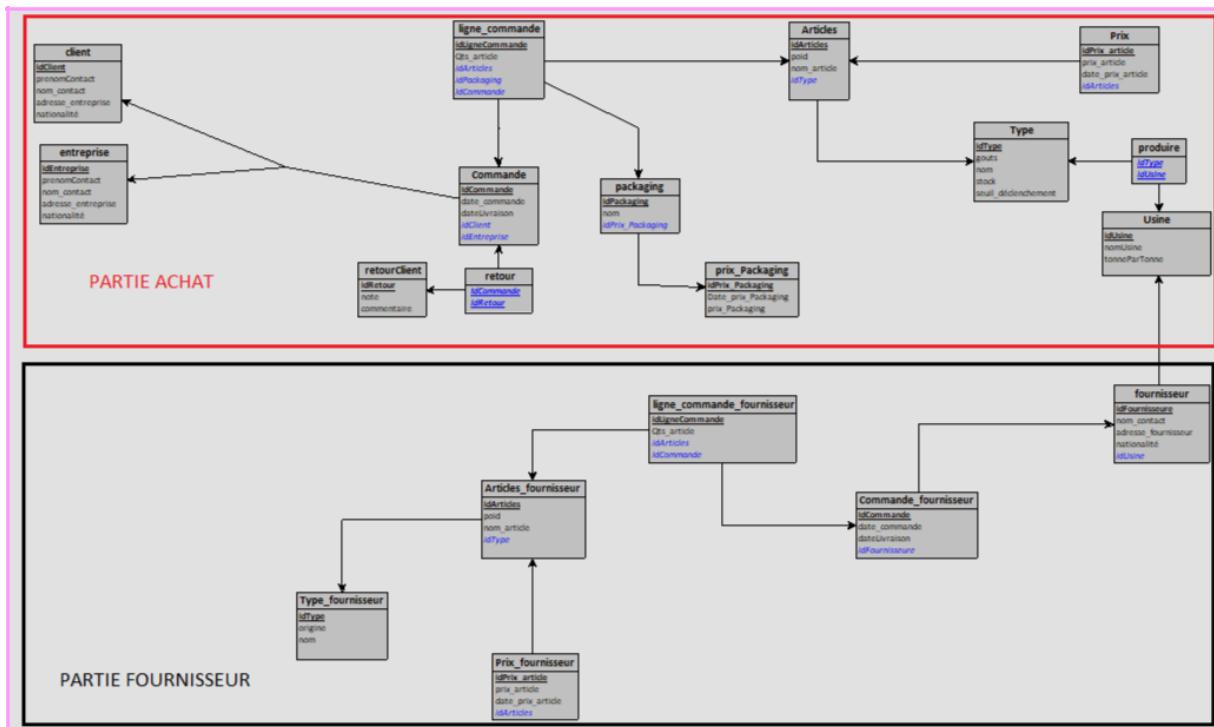
Cette partie est composée d'une table FOURNISSEUR qui représente l'entreprise qui va nous livrer les matières premières nécessaires.

La table COMMANDE ainsi que la table SUIVI_DE_COMMANDE_FOURNISSEUR nous permettent de suivre la logistique de la commande c'est-à-dire sa date de livraison ainsi que la quantité de matières premières commandée.

La table ARTICLE_FOURNISSEUR référence les produits ainsi que leurs poids. Le choix de distinguer le prix dans une table nommée PRIX_FOURNISSEUR nous permet d'appliquer des changements de prix sans altérer notre table ARTICLE_FOURNISSEUR.



B. INTRODUCTION DU MLD (Modèles logique des données)



II. REPRESENTATION TEXTUELLE DU MLD PARTIE ACHAT

```

client = (idClient VARCHAR(50), prenomContact VARCHAR(50), nom_contact
VARCHAR(50), adresse_entreprise VARCHAR(50), nationalité VARCHAR(50));

entreprise = (idEntreprise VARCHAR(50), prenomContact VARCHAR(50), nom_contact
VARCHAR(50), adresse_entreprise VARCHAR(50), nationalité VARCHAR(50));

Commande = (IdCommande INT, date_commande DATETIME, dateLivraison DATETIME,
#idClient, #idEntreprise);

retour = (#IdCommande, #idRetour);

retourClient = (idRetour INT, note INT, commentaire VARCHAR(50));

ligne_commande = (idLigneCommande INT, Qts_article INT, #idArticles, #idPackaging,
#IdCommande);

packaging = (idPackaging INT, nom VARCHAR(50), #idPrix_Packaging);

prix_Packaging = (idPrix_Packaging INT, Date_prix_Packaging DATE, prix_Packaging
INT); Articles = (idArticles INT, poids INT, nom_article VARCHAR(50), #idType);

Prix = (idPrix_article INT, prix_article DECIMAL(15,2), date_prix_article DATE, #idArticles);

Type = (idType INT, gouts VARCHAR(50), nom VARCHAR(50), stock
INT, seuil_déclenchement INT);

produire = (#idType, #idUsine);

```



Usine = (idUsine INT, nomUsine VARCHAR(50), tonneParTonne INT);

III. REPRESENTATION TEXTUELLE DU MLD PARTIE FOURNISSEUR

fournisseur = (idFournisseur VARCHAR(50), nom_contact VARCHAR(50),
adresse_fournisseur VARCHAR(50), nationalité VARCHAR(50), #idUsine);

Commande_fournisseur = (IdCommande INT, date_commande DATETIME, dateLivraison
DATETIME, #idFournisseur);

ligne_commande_fournisseur = (idLigneCommande INT, Qts_article INT, #idArticles,
#IdCommande);

Articles_fournisseur = (idArticles INT, poid INT, nom_article VARCHAR(50), #idType);

Prix_fournisseur = (idPrix_article INT, prix_article DECIMAL(15,2), date_prix_article DATE,
#idArticles);

Type_fournisseur = (idType INT, origine VARCHAR(50), nom VARCHAR(50));

C. SCRIPT PARTIE ACHAT

```
CREATE TABLE client(
    idClient VARCHAR(50),
    prenomContact VARCHAR(50) NOT NULL,
    nom_contact VARCHAR(50) NOT NULL,
    adresse_entreprise VARCHAR(50) NOT NULL,
    nationalité VARCHAR(50) NOT NULL,
    PRIMARY KEY(idClient)
);
```

```
CREATE TABLE entreprise( idEntreprise
    VARCHAR(50), prenomContact
    VARCHAR(50) NOT NULL, nom_contact
    VARCHAR(50) NOT NULL,
    adresse_entreprise VARCHAR(50) NOT
    NULL, nationalité VARCHAR(50) NOT NULL,
    PRIMARY KEY(idEntreprise)
);
```



```

CREATE TABLE Commande(
    IdCommande INT,
    date_commande DATETIME NOT NULL,
    dateLivraison DATETIME,
    idClient VARCHAR(50) NOT NULL,
    idEntreprise VARCHAR(50) NOT NULL,
    PRIMARY KEY(IdCommande),
    FOREIGN KEY(idClient) REFERENCES client(idClient),
    FOREIGN KEY(idEntreprise) REFERENCES entreprise(idEntreprise)
);

```

```

CREATE TABLE retour(
    IdCommande INT,
    idRetour INT,
    PRIMARY KEY(IdCommande, idRetour),
    FOREIGN KEY(IdCommande) REFERENCES Commande(IdCommande),
    FOREIGN KEY(idRetour) REFERENCES retourClient(idRetour)
);

```

```

CREATE TABLE retourClient(
    idRetour INT,
    note INT NOT NULL,
    commentaire VARCHAR(50) NOT NULL,
    PRIMARY KEY(idRetour)
);

```

```

CREATE TABLE ligne_commande(
    idLigneCommande INT,
    Qts_article INT NOT NULL,
    idArticles INT NOT NULL,
    idPackaging INT NOT NULL,
    IdCommande INT,
    PRIMARY KEY(idLigneCommande),
    FOREIGN KEY(idArticles) REFERENCES Articles(idArticles),
    FOREIGN KEY(idPackaging) REFERENCES packaging(idPackaging),
    FOREIGN KEY(IdCommande) REFERENCES Commande(IdCommande)
);

```

```

CREATE TABLE packaging(
    idPackaging INT,
    nom VARCHAR(50) NOT NULL,
    idPrixF_Packaging INT NOT NULL,
    PRIMARY KEY(idPackaging),
    FOREIGN KEY(idPrixF_Packaging)
    REFERENCES prix_Packaging(idPrixF_Packaging)
);

```



```

CREATE TABLE prix_Packaging(
    idPrix_Packaging INT,
    Date_prix_Packaging DATE NOT NULL,
    prix_Packaging INT NOT NULL,
    PRIMARY KEY(idPrix_Packaging)
);

CREATE TABLE Articles(
    idArticles INT,
    poid INT NOT NULL,
    nom_article VARCHAR(50) NOT NULL,
    idType INT NOT NULL,
    PRIMARY KEY(idArticles),
    FOREIGN KEY(idType) REFERENCES Type(idType)
);

CREATE TABLE Prix(
    idPrix_article INT,
    prix_article DECIMAL(15,2) NOT NULL,
    date_prix_article DATE NOT NULL,
    idArticles INT,
    PRIMARY KEY(idPrix_article),
    FOREIGN KEY(idArticles) REFERENCES Articles(idArticles)
);

CREATE TABLE Type(
    idType INT,
    gouts VARCHAR(50) NOT NULL,
    nom VARCHAR(50) NOT NULL,
    stock INT,
    seuil_déclenchement INT,
    PRIMARY KEY(idType)
);

CREATE TABLE produire(
    idType INT,
    idUsine INT,
    PRIMARY KEY(idType, idUsine),
    FOREIGN KEY(idType) REFERENCES Type(idType),
    FOREIGN KEY(idUsine) REFERENCES Usine(idUsine)
);

CREATE TABLE Usine(
    idUsine INT,
    nomUsine VARCHAR(50),
    tonneParTonne INT,
    PRIMARY KEY(idUsine)
);

```



D. SCRIPT PARTIE FOURNISSEUR

```

CREATE TABLE fournisseur(
    idFournisseur VARCHAR(50),
    nom_contact VARCHAR(50) NOT NULL,
    adresse_fournisseur VARCHAR(50) NOT NULL,
    nationalité VARCHAR(50) NOT NULL,
    idUsine INT NOT NULL,
    PRIMARY KEY(idFournisseur),
    FOREIGN KEY(idUsine) REFERENCES Usine(idUsine)
);

```

```

CREATE TABLE fournisseur(
    idFournisseur VARCHAR(50),
    nom_contact VARCHAR(50) NOT NULL,
    adresse_fournisseur VARCHAR(50) NOT NULL,
    nationalité VARCHAR(50) NOT NULL,
    idUsine INT NOT NULL,
    PRIMARY KEY(idFournisseur),
    FOREIGN KEY(idUsine) REFERENCES Usine(idUsine)
);

```

```

CREATE TABLE ligne_commande_fournisseur(
    idLigneCommande INT,
    Qts_article INT NOT NULL,
    idArticles INT NOT NULL,
    IdCommande INT,
    PRIMARY KEY(idLigneCommande),
    FOREIGN KEY(idArticles) REFERENCES Articles_fournisseur(idArticles),
    FOREIGN KEY(IdCommande) REFERENCES
    Commande_fournisseur(IdCommande)
);

```

```

CREATE TABLE Articles_fournisseur(
    idArticles INT,
    poid INT NOT NULL,
    nom_article VARCHAR(50) NOT NULL,
    idType INT NOT NULL,
    PRIMARY KEY(idArticles),
    FOREIGN KEY(idType) REFERENCES Type_fournisseur(idType)
);

```

```

CREATE TABLE Prix_fournisseur(
    idPrix_article INT,
    prix_article DECIMAL(15,2) NOT NULL,
    date_prix_article DATE NOT NULL,
    idArticles INT,
    PRIMARY KEY(idPrix_article),
    FOREIGN KEY(idArticles) REFERENCES Articles_fournisseur(idArticles)
);

```



```

CREATE TABLE Type_fournisseur(
    idType INT,
    origine VARCHAR(50) NOT NULL,
    nom VARCHAR(50) NOT NULL,
    PRIMARY KEY(idType)
);
    
```

E. LES REQUETES

- Affiche le nom et la somme de la quantité de l'article et le gout dans toutes les commandes effectué.

```

SELECT nom_article,SUM(Qts_article),gouts FROM ligne_commande NATURAL
JOIN articles NATURAL JOIN type GROUP BY idArticles;
    
```

Résultat :

	nom_article	SUM(Qts_article)	gouts
▶	minitablette_chocolat	25	nature
	napolitains	150	fruité
	carré	200	amende
	tablette	200	fruité
	tablette	200	noisette

- Affiche la somme du nombre de commandes distinctes et le nombre de la somme de quantité acheter par la nationalité du client.

```

SELECT count(idLigneCommande),SUM(Qts_article) FROM Client NATURAL JOIN
commande NATURAL JOIN ligne_commande GROUP by nationalité;
    
```

Résultat :

	count(idLigneCommande)	SUM(Qts_article)	nationalité
▶	6	775	Belge
	6	775	Française



- Affiche la quantité d'articles acheté avec sa date et son nom.

```
SELECT date_commande,qts_article,nom_article,gouts FROM client NATURAL
JOIN commande NATURAL JOIN ligne_commande NATURAL JOIN articles
NATURAL JOIN type ORDER BY idClient,date_commande;
```

Résultat :

	date_commande	qts_article	nom_article	gouts
▶	2021-11-30 00:00:00	50	napolitains	fruité
	2021-11-30 00:00:00	100	napolitains	fruité
	2021-11-30 00:00:00	25	minitablette_chocolat	nature
	2021-12-03 00:00:00	200	carré	amende
	2021-12-03 00:00:00	200	tablette	noisette
	2021-12-03 00:00:00	200	tablette	fruité
	2021-11-30 00:00:00	25	minitablette_chocolat	nature
	2021-11-30 00:00:00	50	napolitains	fruité
	2021-11-30 00:00:00	100	napolitains	fruité
	2021-12-03 00:00:00	200	tablette	fruité
	2021-12-03 00:00:00	200	carré	amende
	2021-12-03 00:00:00	200	tablette	noisette

- Affiche les notes et les commentaires des clients trier par leurs notes. SELECT * FROM retourclient ORDER BY note;

Résultat :

	idRetour	note	commentaire
▶	2	3	il n'y a pas souvent des réductions sur vos produi
	1	5	delai de livraison court
	3	5	je commande toujours ici c'est mes chocolats préfér
*	HULL	HULL	HULL

- Affiche le nombre d'articles vendu et le chiffre d'affaires de l'entreprise par produit.

```
SELECT nom_article,SUM(Qts_article),prix_article,SUM(Qts_article)*prix_article
FROM client NATURAL JOIN commande NATURAL JOIN ligne_commande
NATURAL JOIN articles NATURAL JOIN prix GROUP BY idarticles;
```

Résultat :

	nom_article	SUM(Qts_article)	prix_article	SUM(Qts_article)*prix_article
▶	minitablette_chocolat	100	7.50	750.00
	napolitains	600	5.40	3240.00
	carré	800	6.27	5016.00
	tablette	800	10.20	8160.00
	tablette	800	8.67	6936.00

- Affiche tous les prix par articles en fonction du gout et du nom.

```
SELECT prix_article,date_prix_article,nom_article,gouts from articles
NATURAL JOIN prix NATURAL JOIN type ORDER BY nom_article,gouts;
```

Résultat :

	prix_article	date_prix_article	nom_article	gouts
▶	6.27	2021-11-24	carré	amende
	5.27	2021-12-02	carré	amende
	7.50	2021-11-29	minitablette_chocolat	nature
	6.50	2021-12-06	minitablette_chocolat	nature
	4.40	2021-12-06	napolitains	fruité
	5.40	2021-11-02	napolitains	fruité
	9.20	2021-12-04	tablette	fruité
	10.20	2021-11-24	tablette	fruité
	8.00	2021-12-02	tablette	nature
	9.00	2021-11-24	tablette	nature
	7.67	2021-12-04	tablette	noisette
	8.67	2021-11-24	tablette	noisette