

PRESENTATION DU PROJET :

GESTION DE BOUTIQUE.

Année universitaire 2020/2021

INTRODUCTION

Afin de compléter notre formation de deux modules Conception et Gestion des Bases de Données et Structure de Données et Programmation en C en filière première année cycle d'ingénieur en génie informatique, et de mettre en pratique les connaissances acquises pendant notre formation, nous sommes amenée à réaliser un projet de fin de formation.

Dans le cadre de notre travail, nous avons décidé avec notre encadrant de générer une code source en C et créer un système d'information utilisant la méthode de merise et le logiciel SQL server qui facilitent les diverses démarches d'une boutique.

Ce rapport présente notre projet en détail, il suit le plan suivant :

- Le premier chapitre décrit le contexte général du projet aborde la phase d'analyse et de conception.
- Le deuxième chapitre traite la phase de l'étude technique en décrivant les outils et langages utilisés.

Chapitre 01 : Contexte général du projet.

I. Partie de Base de Données.

1. Présentation de projet.

Une boutique privée souhaite créer un système d'information lui permettant de gérer les factures de ses clients et son stock. Ci-dessous, la liste de règles de gestion à prendre en considération:

- ✓ Une boutique connue par un numéro, un nom, une adresse, une ville et numéro de téléphone.
- ✓ Un client peut payer une ou plusieurs factures.
- ✓ Une facture caractérisée par son numéro et une date.
- ✓ Une facture contient au moins une ligne commande.
- ✓ Un produit caractérise par un numéro, désignation, un prix unitaire et un taux de valeur ajoutée (tva).
- ✓ Un produit peut appartenir à une ligne de commandes avec une quantité.
- ✓ Lorsqu'un produit est placé sur une ligne de commande, sa quantité stockée est réduite.
- ✓ Une fois que la quantité stockée du produit est égale à zéro, sa condition de disponibilité est égale à zéro (**Etat "0" : indisponible, Etat "1" : disponible**), c'est-à-dire que le client ne peut pas commander ce produit pendant cette période.
- ✓ Une catégorie contient au moins un produit, elle caractérisée par un numéro et un nom.
- ✓ Un produit appartient à une et une seule catégorie.
- ✓ Un produit fournit au mois par un fournisseur avec une quantité donnée.
- ✓ Lorsqu'un produit est fourni par un fournisseur, sa quantité stockée est augmentée.
- ✓ Un fournisseur connaît par un numéro, un nom et numéro de téléphone, il fournit au moins un produit.
- ✓ Le client qui acheté les produits à un numéro, un nom, un prénom, une adresse, une ville, un code postal et un chiffre d'affaire.

2. Analyse de besoin.

2.1. Les entités.

Les différentes entités de notre projet sont :

- **Entité BOUTIQUE** : porte en attribut IdBoutique, NomBoutique, AdressBoutique, VilleBoutique, TelephoneBoutique.
- **Entité CLIENT** : porte en attribut IdClient, NomClient, PrenomClient, AdresseClient, CodePostaleClient, VilleClient, TelephoneClient, et ChiffreAffaireClient.
- **Entité FACTURE** : porte en attribut IdFacture, DateFature.
- **Entité LIGNE_COMMANDE** : porte en attribut Quantité.
- **Entité PRODUIT** : porte en attribut IdProduit, Designation, Prix Unitaire, Tva_Produit. Quantité_Stockée, EtatProduit.

- **Entité FOURNISSEUR:** porte en attribut IdFournisseur, NomFournisseur, PrenomFournisseur, Téléphone.
- **Entité CATEGORIE:** porte en attribut IdCategorie, NomCategorie.

2.2. Les associations.

Les associations existent entre les différentes entités de notre projet sont :

- ✓ L'association entre entité Boutique et l'entité Client (**ABONNER**).
- ✓ L'association entre entité Client et l'entité Facture (**PAYER**).
- ✓ L'association entre entité Facture et l'entité Ligne_Commande (**CONCERNER**).
- ✓ L'association entre entité Ligne_Commande et l'entité Produit (**APPARTIENT**).
- ✓ L'association entre entité Produit et l'entité Fournisseur (**FOURNIR**). Elle porte comme attribut Quantité_Fourni.
- ✓ L'association entre entité Produit et l'entité Catégorie (**CONTIENT**).

3. Modèle conceptuel des données

3.1. Définitions

➤ Modèle conceptuel des données

Le modèle conceptuel des données (MCD) a pour but d'écrire de façon formelle les données qui seront utilisées par le système d'information. Il s'agit donc d'une représentation des données, facilement compréhensible, permettant de décrire le système d'information à l'aide d'entités.

➤ Une entité

Une entité est la représentation d'un élément matériel ou immatériel ayant un rôle dans le système que l'on désire décrire.

➤ Une association

Une association (appelée aussi parfois relation) représente les liens sémantiques qui peuvent exister entre plusieurs entités

➤ La cardinalité

Les cardinalités permettent de caractériser le lien qui existe entre une entité et la relation à laquelle elle est reliée. La cardinalité d'une relation est composée d'un couple comportant une borne maximale et une borne minimale, intervalle dans lequel la cardinalité d'une entité peut prendre sa valeur :

La borne minimale (généralement 0 ou 1) décrit le nombre minimum de fois qu'une entité peut participer à une relation

La borne maximale (généralement 1 ou n) décrit le nombre maximum de fois qu'une entité peut participer à une relation

3.2. Le dictionnaire des données

Pendant la phase de conception, les données recueillies et spécifiées sont inscrites dans un dictionnaire. Ce dictionnaire est un outil important car il constitue la référence de toutes les études effectuées ensuite.

Cela comprend l'identification des différentes données, sachant qu'il existe trois types de données :

Données élémentaires :

Elles ne sont pas obtenues par calcul à partir d'autres données.

Données calculées.

Elles résultent d'un calcul effectué à partir d'autres données.

Données paramètres.

C'est une donnée qui ne prend qu'une unique valeur.

Le tableau suivant représente notre dictionnaire :

Données	Type	Données élémentaires
IdClient	Elémentaire	Oui
NomClient	Elémentaire	Oui
PrenomClient	Elémentaire	Oui
AdresseClient	Elémentaire	Oui
TelephoneClient	Elémentaire	Oui
VilleClient	Elémentaire	Oui
CodePostalClient	Elémentaire	Oui
ChiffreAffireClient	Calculée	Non
IdProduit	Elémentaire	Oui
DésignationProduit	Elémentaire	Oui
PrixUnitaire	Elémentaire	Oui
TvaProduit	Elémentaire	Oui
QuantiteStockee	Calculée	Non
EtatProduit	Calculée	Non
IdFacture	Elémentaire	Oui
DateFacture	Elémentaire	Oui
Quantite	Elémentaire	Oui
IdFournisseur	Elémentaire	Oui
NomFournisseur	Elémentaire	Oui
PrenomFournisseur	Elémentaire	Oui
TelephoneFournisseur	Elémentaire	Oui
QantiteFourni	Calculée	Non
IdBoutique	Paramètre	Non
NomBoutique	Paramètre	Non
VilleBoutique	Paramètre	Non
AdresseBoutique	Paramètre	Non
TelephoneBoutique	Paramètre	Non
IdCatégorie	Elémentaire	Oui
NomCategorie	Elémentaire	Oui

3.3.Les dépendances fonctionnelles

Le dictionnaire des données étant terminé, nous allons créer l'ensemble des dépendances fonctionnelles.

Le tableau suivant représente toutes les dépendances fonctionnelles.

	DICT	IdClient	IdProduit	IdFacture	IdFournisseur	IdCategorie
Client	IdClient			1		
	NomClient	1		x		
	PrenomClient	1		x		
	AdresseClient	1		x		
	TelephoneClient	1		x		
	VilleClient	1		x		
	CodePostalClient	1		x		
	ChiffreAffaireClient	1		x		
Produit	IdProduit					
	DesignationProduit		1			
	PrixUnitaire		1			
	TvaProduit		1			
	QantiteStockee		1			
	EtatProduit		1			
Facture	IdFacture					
	DateFacture			1		
LigneCommande	Quantite					
Fournisseur	IdFournisseur					
	NomFournisseur				1	
	PrenomFournisseur				1	
	TelephoneFournisseur				1	
Fournir	QantiteFourni					
Categorie	IdCategorie		1			
	NomCategorie		x			1

3.4. Modèle conceptuel des données

Le modèle conceptuel des données (modèle entité association) de notre projet est désigné sur la figure suivante :

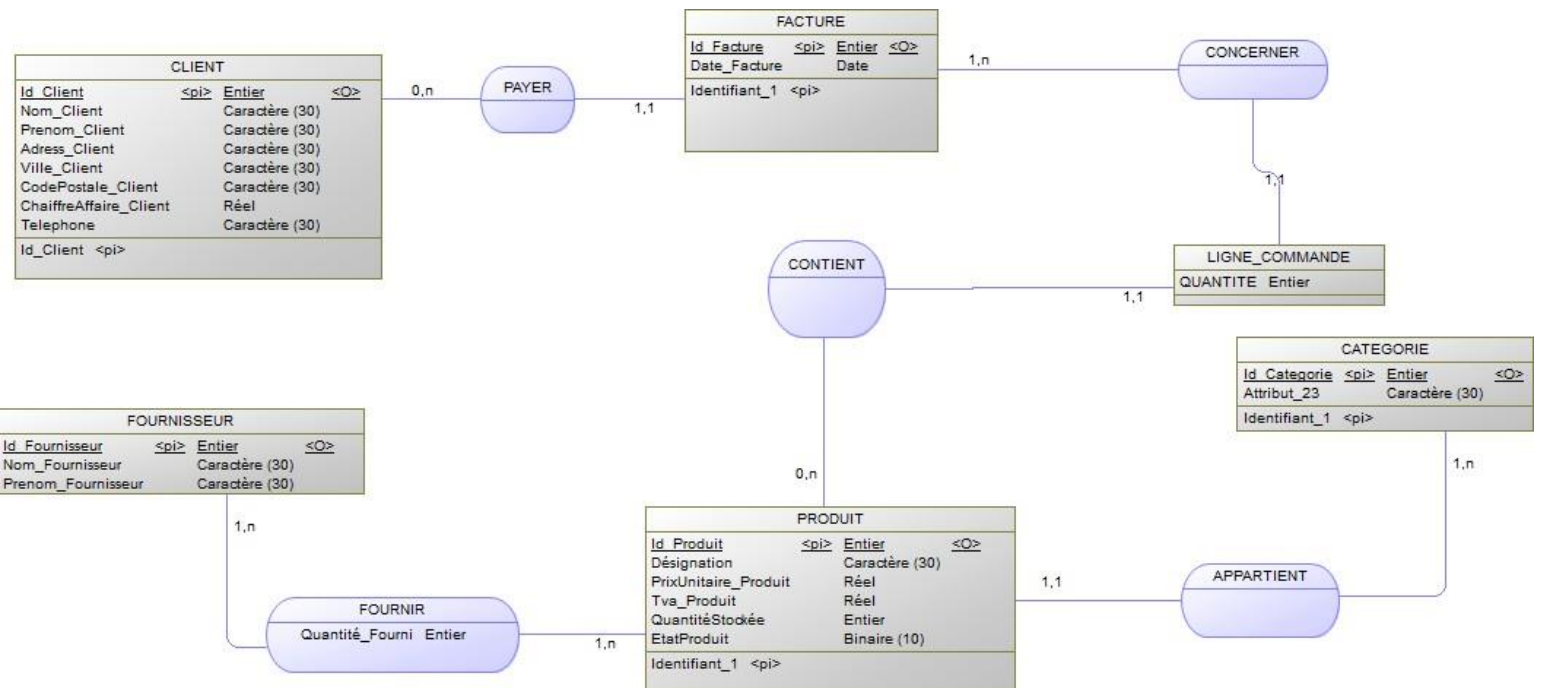


Figure 01 : Modèle conceptuel des données

3.5. Modèle physique de données.

➤ Transformation vers le modèle physique des données

- Une entité du MCD devient une relation, c'est à dire une table
- Son identifiant devient la clé primaire de la relation
- Les autres propriétés deviennent les attributs de la relation.
- Une association de type 1 : N (c'est à dire qui a les cardinalités maximales positionnées à « 1 » d'une côté de l'association et à « n » de l'autre côté) se traduit par la création d'une clé étrangère dans la relation correspondante à l'entité côté « 1 ». Cette clé étrangère référence la clé primaire de la relation correspondant à l'autre entité.
- Une association de type N : N (c'est à dire qui a les cardinalités maximales positionnées à « N » des 2 côtés de l'association) se traduit par la création d'une table dont la clé primaire est composée des clés étrangères référençant les relations correspondant aux entités liées par l'association. Les éventuelles propriétés de l'association deviennent des attributs de la relation.

➤ Modèle physique de données.

D'après le modèle conceptuel de données, il est facile de générer le modèle physique de données.

Les figures suivantes représentent notre modèle physique de données.

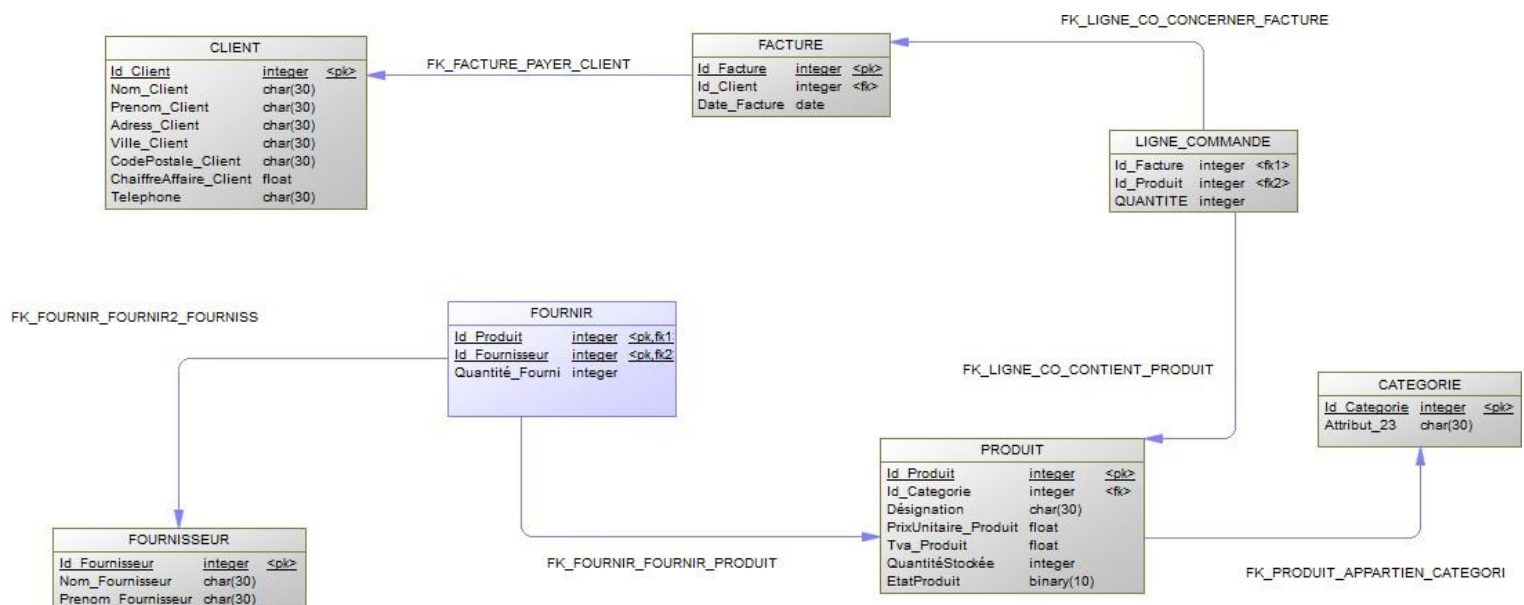


Figure 02 : Modèle physique des données avec PowerDesigner

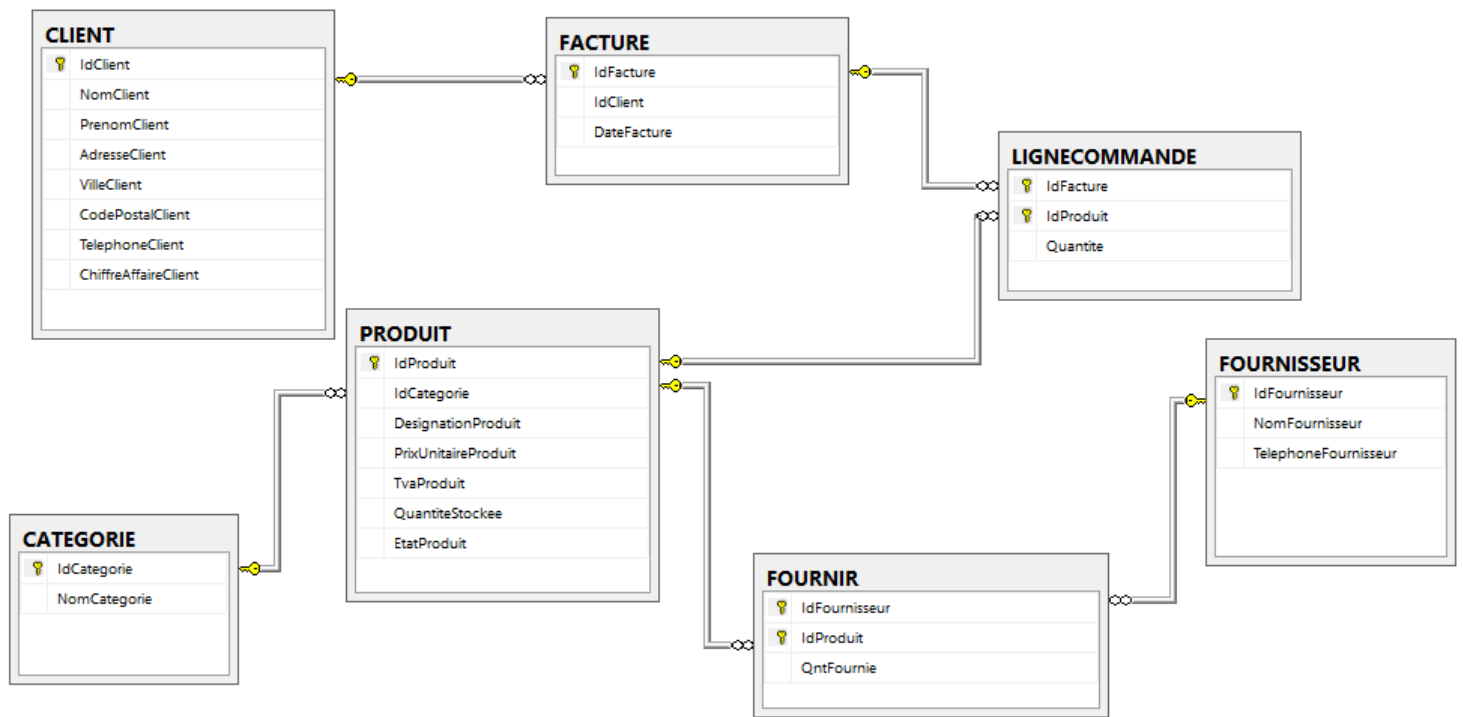


Figure 03 : Modèle physique des données avec SQL server

4. Travail à faire.

Après avoir obtenu le modèle physique, nous allons maintenant créer des procédures, des vues, des fonctions et des curseurs pour répondre à diverses demandes liées à la gestion de la boutique utilisant SQL server 2014.

Vous trouvez le code source (les requêtes) dans les fichiers [Boutique_log.ldf](#) et [Boutique.mdf](#).

II. Partie de langage C.

Dans cette partie, nous examinerons les tableaux de base de données en tant que structures et stockerons les valeurs de tableaux dans des fichiers.

Ensuite, nous créerons des fonctions et des procédures pour gérer et exécuter des opérations sur des structures afin de répondre aux différentes demandes liées à la boutique

Vous trouvez le code source dans le dossier [Gestion_Boutique](#) .

Chapitre 02 : Les outils

1. Langages utilisés :

Au cours de la réalisation de notre projet, nous avons choisi des langages qui sont liés à ses deux modules, ces langages sont

Langage C :

C est un langage de programmation impératif généraliste, de bas niveau. Inventé au début des années 1970 pour réécrire UNIX, **C** est devenu un des langages les plus utilisés, encore de nos jours.

C offre au développeur une marge de contrôle importante sur la machine (notamment sur la gestion de la mémoire) et est de ce fait utilisé pour réaliser les « fondations » (compilateurs, interpréteurs...) de ces langages plus modernes.



Figure 04 : Logo de Langage C.

Transact-SQL (T-SQL) :

Le **Transact-SQL (T-SQL)** est une extension propriétaire de Sybase et Microsoft au langage SQL1. Transact-SQL a été développé à l'origine par la société Sybase, dès les premières versions de son moteur de base de données du même nom.



Figure 05 : Logo de Transact-SQL (T-SQL).

2. Logiciels utilisés :

Dev-C++ :

Dev-C++ est un environnement de développement intégré (IDE) permettant de programmer en C et en C++ pour les systèmes d'exploitation Windows. Il a été créé et développé par Colin Laplace et sa première version rendue publique en 1998. Longtemps à l'abandon, le projet a été repris par un autre développeur en 2011 et est régulièrement mis à jour.



Figure 06: Logo de Dev-C++.

Code::Blocks :

Code::Blocks est un environnement de développement intégré libre et multiplate-forme. Il est écrit en C++ et utilise la bibliothèque wxWidgets. Code::Blocks est orienté C et C++, mais il supporte d'autres langages comme FORTRAN ou le D.

Code::Blocks existe pour Linux, Windows et Mac OS X. Des utilisateurs indiquent avoir réussi à compiler le code source sous FreeBSD[réf. nécessaire]1.

Il obtient une note de satisfaction de 4.7 (sur un maximum de 5) de la part de ceux qui l'ont téléchargé sur SourceForge.



Figure 07 : Logo de Code::Blocks .

Eclipse :

Eclipse est un projet, décliné et organisé en un ensemble de sous-projets de développements logiciels, de la fondation Eclipse visant à développer un environnement de production de logiciels libre qui soit extensible.

Compilateur : MacOS gcc



Figure 08 : Logo d'Eclipse.

Visual Studio Code : est un éditeur de code open-source développé par Microsoft supportant un très grand nombre de langages grâce à des extensions.



Figure 09 : Logo de Visual Studio Code.

Microsoft SQL Server (2014) :

est un système de gestion de base de données (SGBD) en langage SQL incorporant entre autres un SGBDR (SGBD relationnel ») développé et commercialisé par la société Microsoft. Il fonctionne sous les OS Windows et Linux (depuis mars 2016), mais il est possible de le lancer sur Mac OS via Docker, car il en existe une version en téléchargement sur le site de Microsoft.



Figure 10 : Logo de SQL Server 2014.

PowerDesigner (anciennement PowerAMC) :

est un logiciel de conception créé par la société SAP, qui permet de modéliser les traitements informatiques et leurs bases de données associées.

Il a été créé par SDP sous le nom AMC*Designor, racheté par Powersoft qui lui-même a été racheté par Sybase en 1995. Depuis 2010 Sybase appartient à l'éditeur allemand SAP.



Figure 11 : Logo de PowerDesigner.

Conclusion :

Dans ce chapitre on a cité les différents langages et logiciels utilisés en justifiant leur choix,