

Bonfigli Cédric

Rapport CCP 1

# Résumé

J’ai effectué le dictionnaire de donnée, les entités, le mcd et mld et le model relationnelle en correspondance avec le cahier des charges données.

J’ai aussi effectué la base de données correspondante au mld et le logiciel c# qui sert a géré la base de données.

# Table des matières

[Résumé 1](#_Toc441015761)

[Table des matières 2](#_Toc441015762)

[Dictionnaire de donnée / entité du model 3](#_Toc441015763)

[MCD 4](#_Toc441015764)

[MLD 5](#_Toc441015765)

[MODEL RELATIONNEL 6](#_Toc441015766)

[Base De Donnée 7](#_Toc441015767)

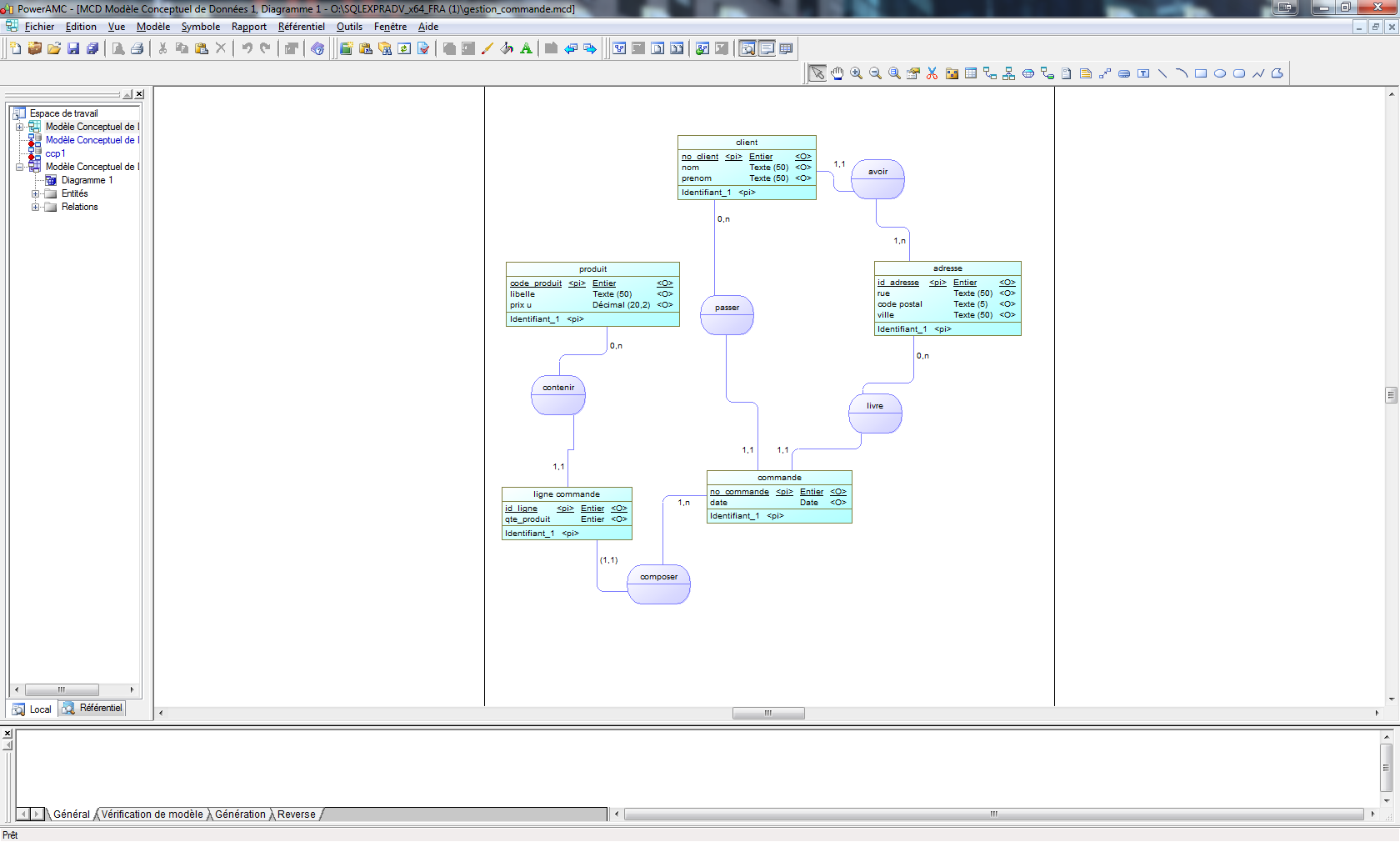
[Logiciel c# 13](#_Toc441015768)

# Dictionnaire de donnée / entité du model

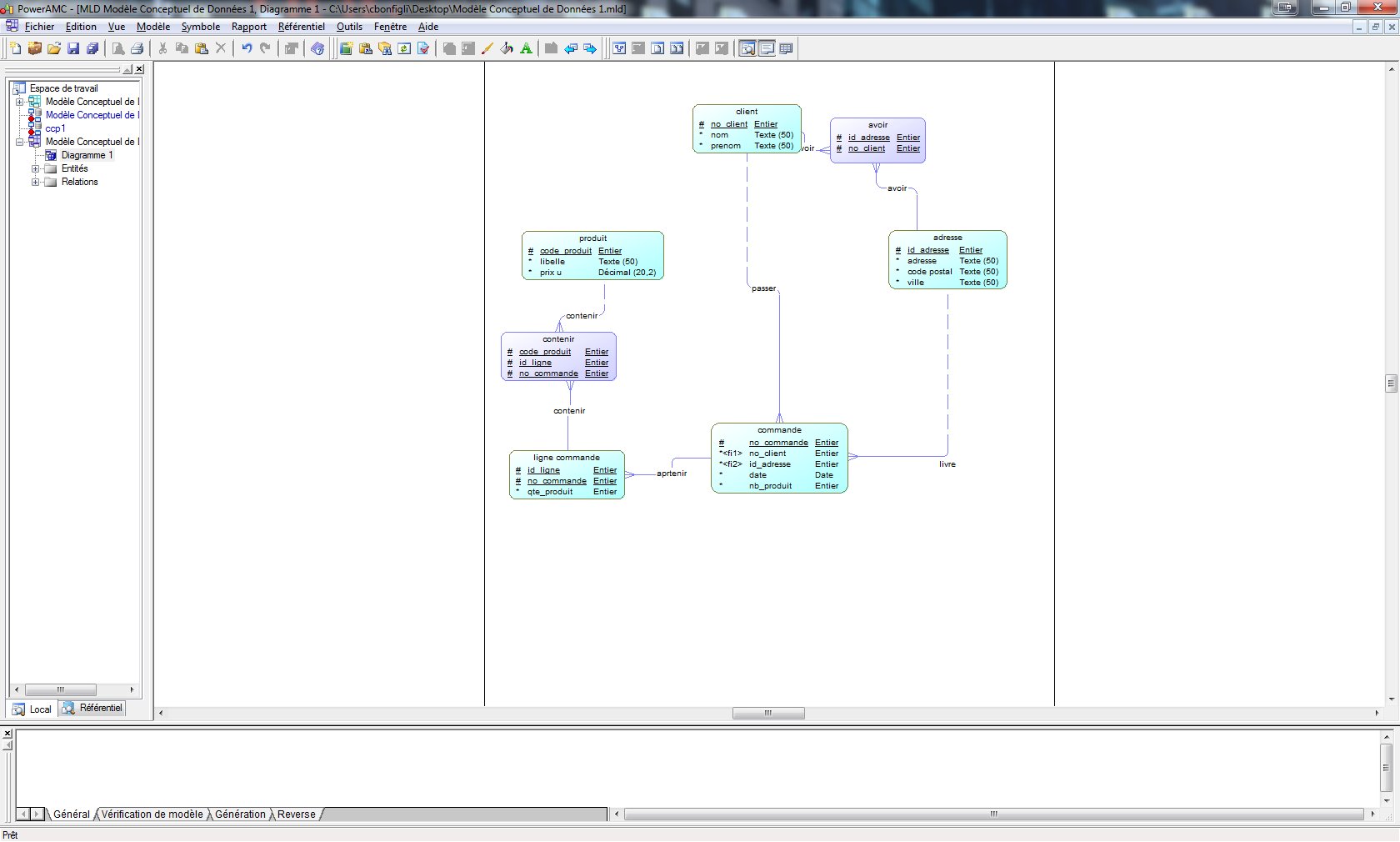
Voici le dictionnaire de donnée avec les entités correspondantes.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | type | taille | null/non null | Primaire |
| PRODUIT |  |  |  |  |
| code produit | entier |  | non null | oui |
| libellé | texte | 50 | non null |  |
| prix | float |  | non null |  |
| CLIENT |  |  |  |  |
| No client | entier |  | non null | oui |
| nom client | texte | 50 | non null |  |
| prenom | texte | 50 | non null |  |
| ID\_ADRESSE | int | 50 | non null |  |
| COMMANDE |  |  |  |  |
| No commande | entier |  | non null | oui |
| date commande | date |  | non null |  |
| ID\_ADRESSE | int | 50 | non null |  |
| LIGNE COMMANDE |  |  |  |  |
| code produit | entier |  | non null |  |
| qte commander | entier |  | non null |  |
| nb produit commander | entier |  | non null |  |
| ADRESSE |  |  |  |  |
| ID\_ADRESSE | int |  | non null | oui |
| Adresse | texte | 50 | non null |  |
| code postal | texte | 5 | non null |  |
| ville | texte | 50 | non null |  |

# MCD



# MLD



# MODEL RELATIONNEL

**CLIENT(id\_client, id\_adresse#, nom, prénom)**

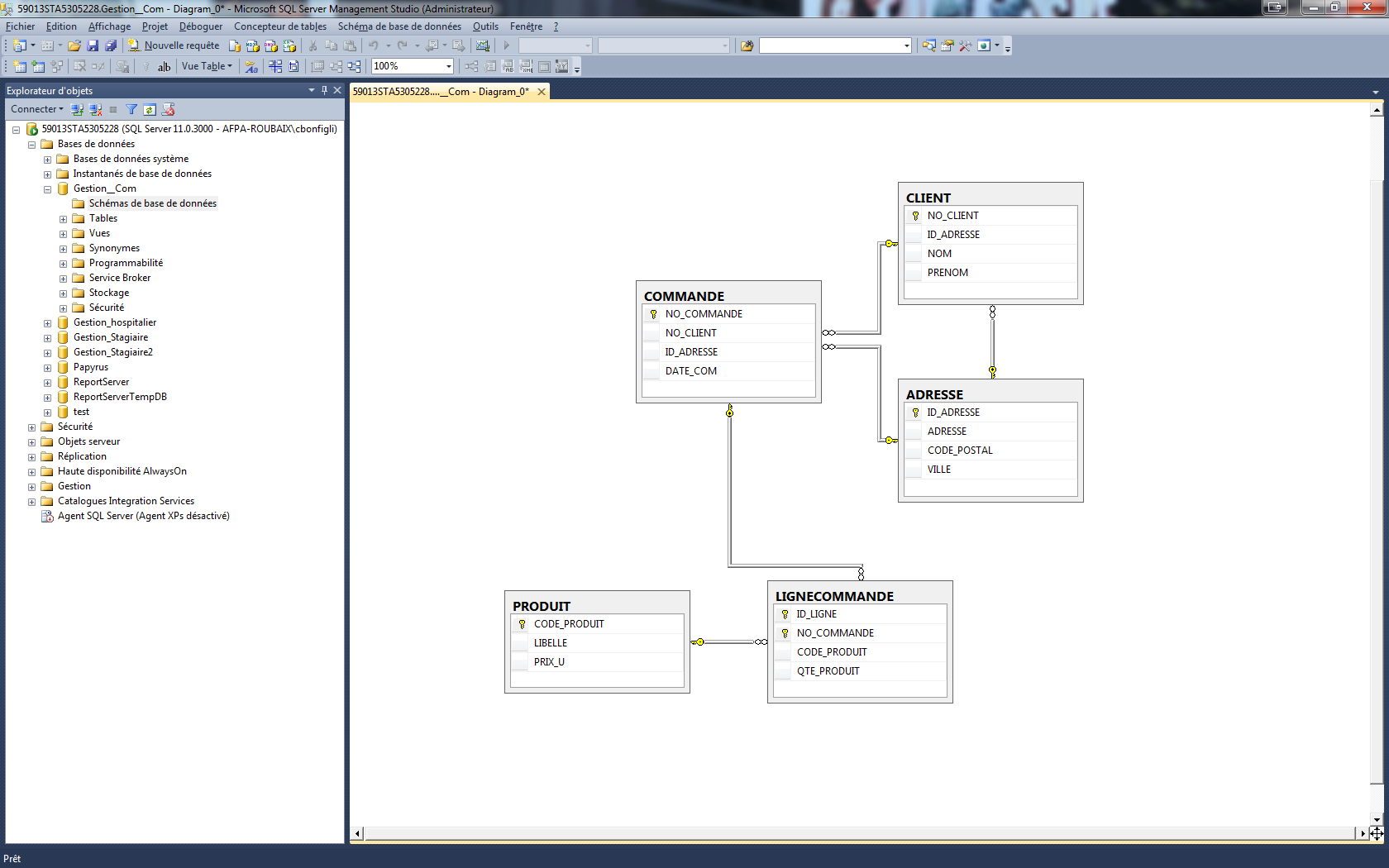
**COMMANDE(no\_commande, id\_client#, id\_adresse#, datecom)**

**ADRESSE(id\_adresse, rue, ville, codepost)**

**LIGNE\_COMMANDE(id\_ligne,no\_commande#,qte\_produit)**

**PRODUIT(code\_produit, libellé, prix\_u)**

# Base De Donnée



1. Code base de donnée

USE master;

GO

create database Gestion\_\_Com

go

use Gestion\_\_Com

go

/\*==============================================================\*/

/\* Table : ADRESSE \*/

/\*==============================================================\*/

create table ADRESSE (

ID\_ADRESSE int identity(1,1) not null,

ADRESSE nvarchar(50) not null,

CODE\_POSTAL nvarchar(5) not null,

VILLE nvarchar(50) not null,

constraint PK\_ADRESSE PRIMARY KEY CLUSTERED (ID\_ADRESSE),

CONSTRAINT UNIQUE\_ADRESSE\_CODE\_POSTAL\_VILLE unique(ADRESSE,CODE\_POSTAL,VILLE),

)

go

/\*==============================================================\*/

/\* Table : CLIENT \*/

/\*==============================================================\*/

create table CLIENT (

NO\_CLIENT int identity(1,1) not null,

ID\_ADRESSE int not null,

NOM nvarchar(50) not null,

PRENOM nvarchar(50) not null,

constraint PK\_CLIENT PRIMARY KEY CLUSTERED (NO\_CLIENT),

)

go

/\*==============================================================\*/

/\* Table : COMMANDE \*/

/\*==============================================================\*/

create table COMMANDE (

NO\_COMMANDE int identity(1,1) not null,

NO\_CLIENT int not null,

ID\_ADRESSE int not null,

DATE\_COM datetime not null,

constraint PK\_COMMANDE PRIMARY KEY CLUSTERED (NO\_COMMANDE),

)

go

/\*==============================================================\*/

/\* Table : LIGNE COMMANDE \*/

/\*==============================================================\*/

create table LIGNECOMMANDE (

ID\_LIGNE int identity(1,1) not null,

NO\_COMMANDE int not null,

CODE\_PRODUIT int not null,

QTE\_PRODUIT int not null,

constraint PK\_LIGNECOMMANDE PRIMARY KEY CLUSTERED (ID\_LIGNE,NO\_COMMANDE),

)

go

/\*==============================================================\*/

/\* Table : PRODUIT \*/

/\*==============================================================\*/

create table PRODUIT (

CODE\_PRODUIT int identity(1,1) not null,

LIBELLE nvarchar(50) not null,

PRIX\_U decimal(20,2) not null,

constraint PK\_PRODUIT PRIMARY KEY CLUSTERED (CODE\_PRODUIT),

)

go

alter table COMMANDE WITH CHECK ADD CONSTRAINT FK\_COMMANDE\_CLIENT

FOREIGN KEY(NO\_CLIENT)

REFERENCES CLIENT (NO\_CLIENT)

GO

alter table COMMANDE WITH CHECK ADD CONSTRAINT FK\_COMMANDE\_ADRESSE

FOREIGN KEY(ID\_ADRESSE)

REFERENCES ADRESSE (ID\_ADRESSE)

GO

alter table CLIENT WITH CHECK ADD CONSTRAINT FK\_CLIENT\_ADRESSE

FOREIGN KEY(ID\_ADRESSE)

REFERENCES ADRESSE (ID\_ADRESSE)

GO

alter table LIGNECOMMANDE WITH CHECK ADD CONSTRAINT FK\_LIGNECOMMANDE\_COMMANDE

FOREIGN KEY(NO\_COMMANDE)

REFERENCES COMMANDE (NO\_COMMANDE)

GO

alter table LIGNECOMMANDE WITH CHECK ADD CONSTRAINT FK\_LIGNECOMMANDE\_PRODUIT

FOREIGN KEY(CODE\_PRODUIT)

REFERENCES PRODUIT (CODE\_PRODUIT)

GO

1. Procédure stockée

a) Cette procédure sert à insérer les commande et l’adresse et récupérer l’id adresse du client.

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE PROCEDURE [dbo].[insertion\_commande2]

@nom nvarchar(50),

@date date,

@adresse nvarchar(50),

@codepostal varchar(5),

@ville nvarchar(50)

AS

BEGIN

declare @id int;

declare @id\_cli int;

insert into ADRESSE (ADRESSE, CODE\_POSTAL, VILLE)

values(@adresse,@codepostal,@ville)

set @id =(select ID\_ADRESSE from ADRESSE where ADRESSE like '%'+@adresse+'%' and code\_postal like '%'+ @codepostal+ '%' and ville like'%'+ @ville+'%');

set @id\_cli = (select NO\_CLIENT from CLIENT where nom = @nom);

insert into commande(NO\_CLIENT, ID\_ADRESSE, DATE\_COM)

values(@id\_cli,@id,@date)

END

GO

b) Cette procédure sert à insérer les commande et récupérer l’id adresse du client.

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE PROCEDURE [dbo].[insertion\_commande]

@nom nvarchar(50),

@adresse nvarchar(50),

@codepostal varchar(5),

@ville nvarchar(50),

@date date

AS

BEGIN

declare @id int;

declare @id\_cli int;

set @id =(select ID\_ADRESSE from ADRESSE where ADRESSE like '%'+@adresse+'%' and code\_postal like '%'+ @codepostal+ '%' and ville like'%'+ @ville+'%');

set @id\_cli = (select NO\_CLIENT from CLIENT where nom = @nom);

insert into commande(NO\_CLIENT, ID\_ADRESSE, DATE\_COM)

values(@id\_cli,@id,@date)

END

GO

c) Cette procédure sert à insérer les clients et l’adresse et récupérer l’id adresse du client.

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE PROCEDURE [dbo].[insertion\_client]

@nom nvarchar(50),

@prenom nvarchar(50),

@adresse nvarchar(50),

@codepostal varchar(5),

@ville nvarchar(50)

AS

BEGIN

declare @id int;

insert into ADRESSE (ADRESSE, CODE\_POSTAL, VILLE)

values(@adresse,@codepostal,@ville)

set @id =(select ID\_ADRESSE from ADRESSE where ADRESSE like '%'+@adresse+'%' and code\_postal like '%'+ @codepostal+ '%' and ville like'%'+ @ville+'%');

insert into client (ID\_ADRESSE, NOM, PRENOM)

values (@id,@nom,@prenom)

END

GO

1. Fonction table

La fonction table sert à l’affichage des produit.

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE FUNCTION [dbo].[select\_produit]

(

)

RETURNS TABLE

AS

RETURN

(

select \* from produit

)

GO

1. Vue

La fonction vue sert à l’affichage des clients.

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

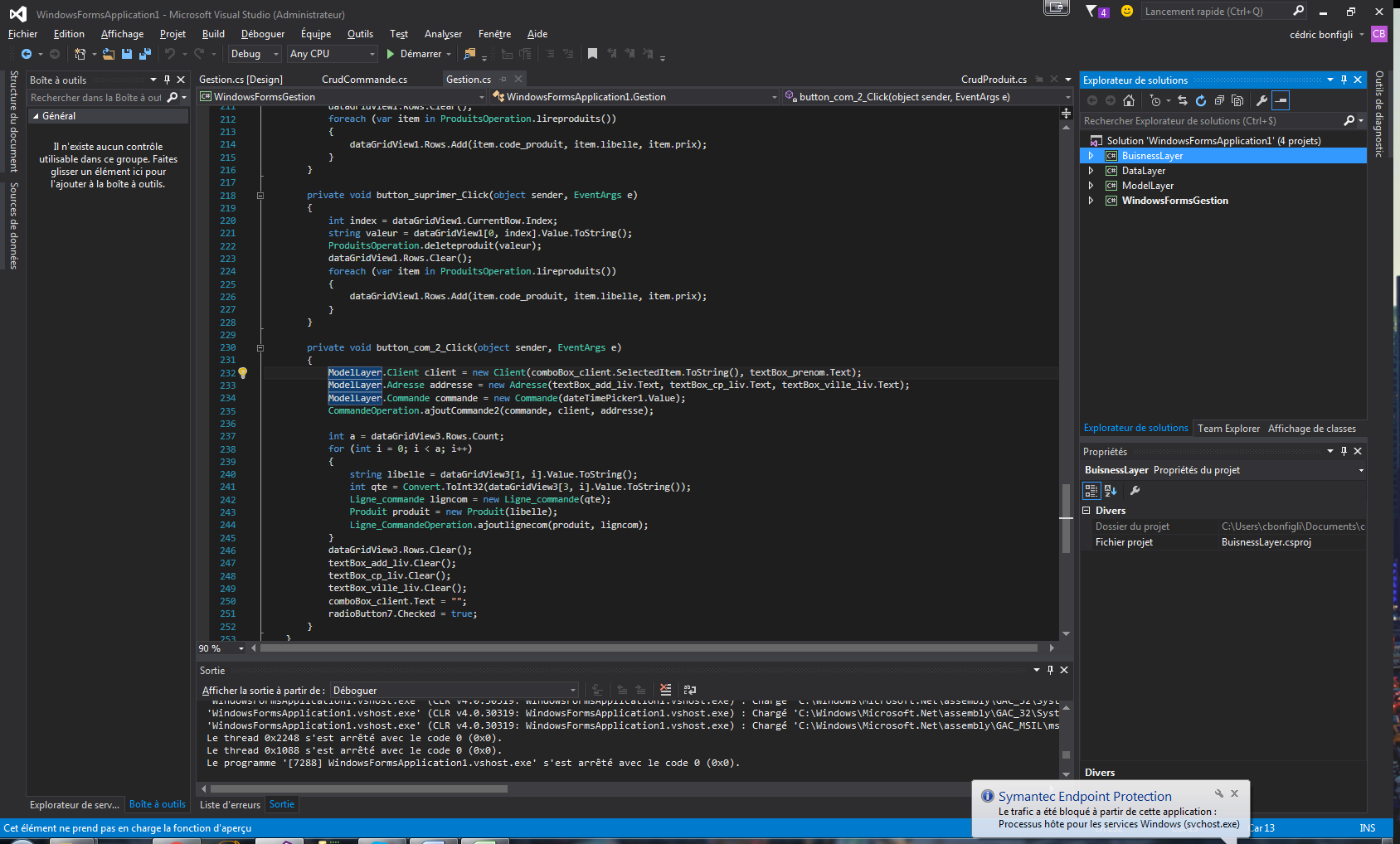
create view [dbo].[select\_client] as

select \* from client

GO

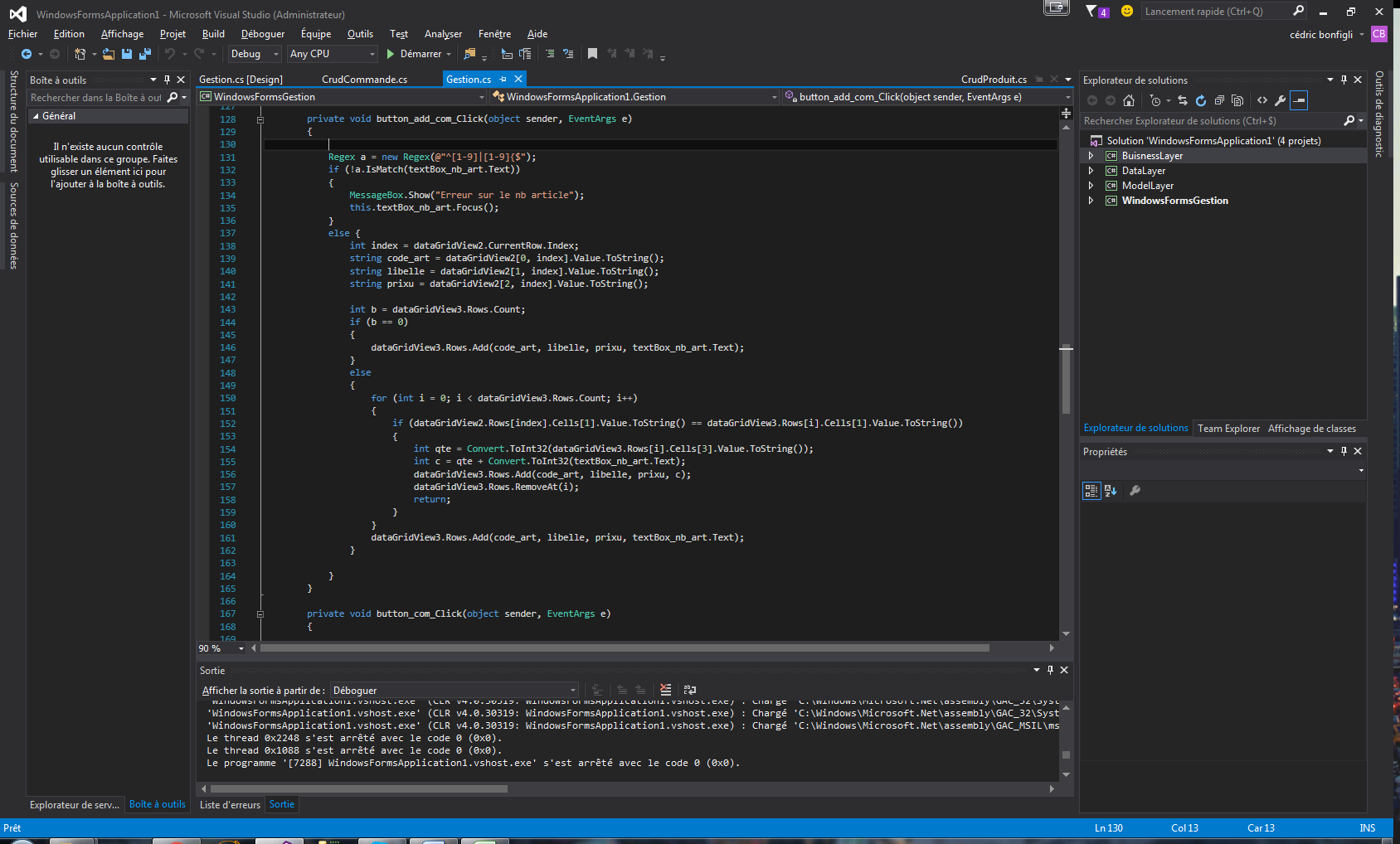
# Logiciel c#

Le logiciel a été fait en programmation 1/3 on voit bien les 3 couches la couche datalayer qui contient la connexion à la base de données et les différentes requêtes, la couche modellayer qui contient les différents objets, la couche business ou les opérations de logiciel sont faite et la couche windowsforms ou l’affichage et la gestion des boutons est gère.

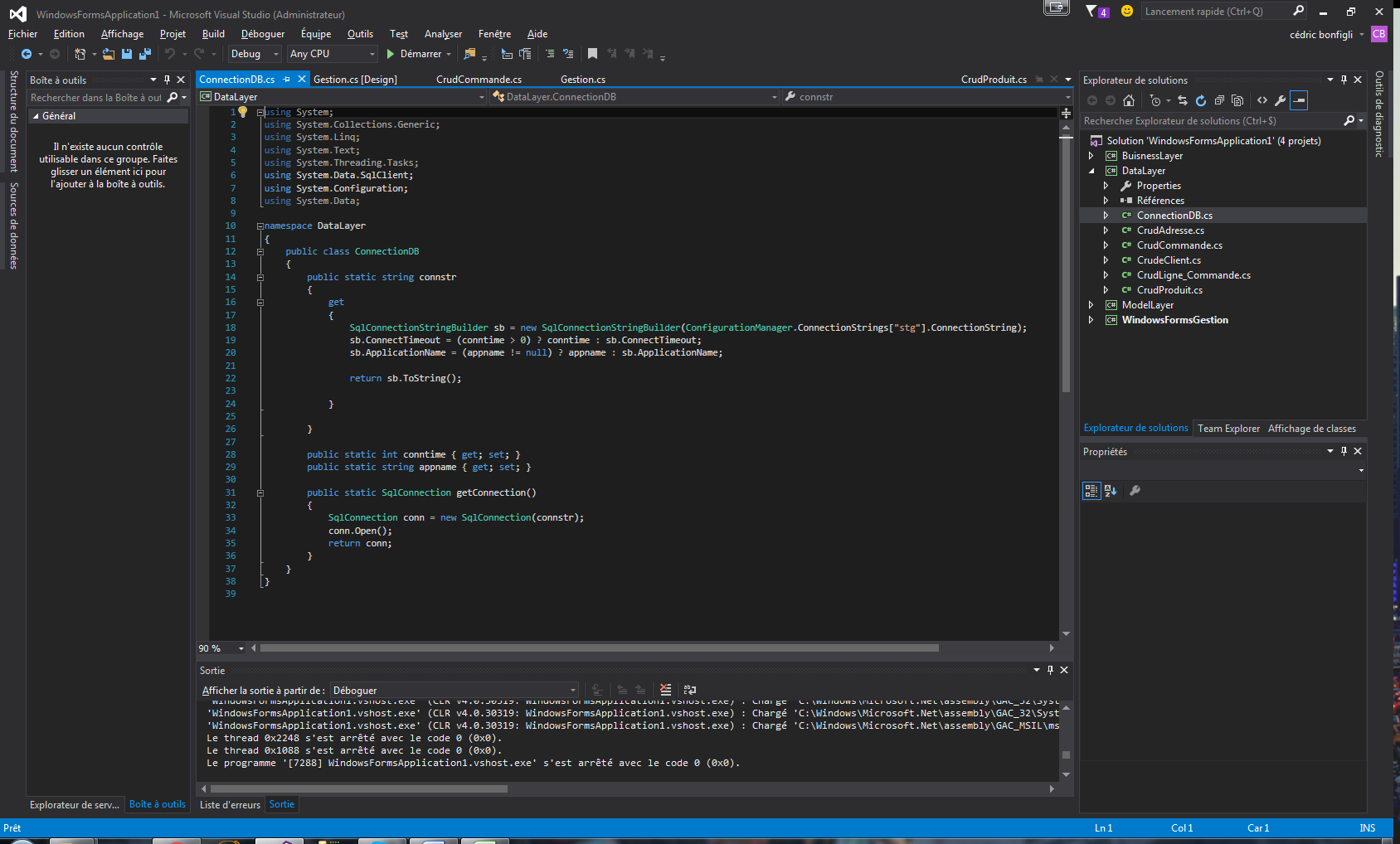


Je vais vous présenter quelque fonction du logiciel.

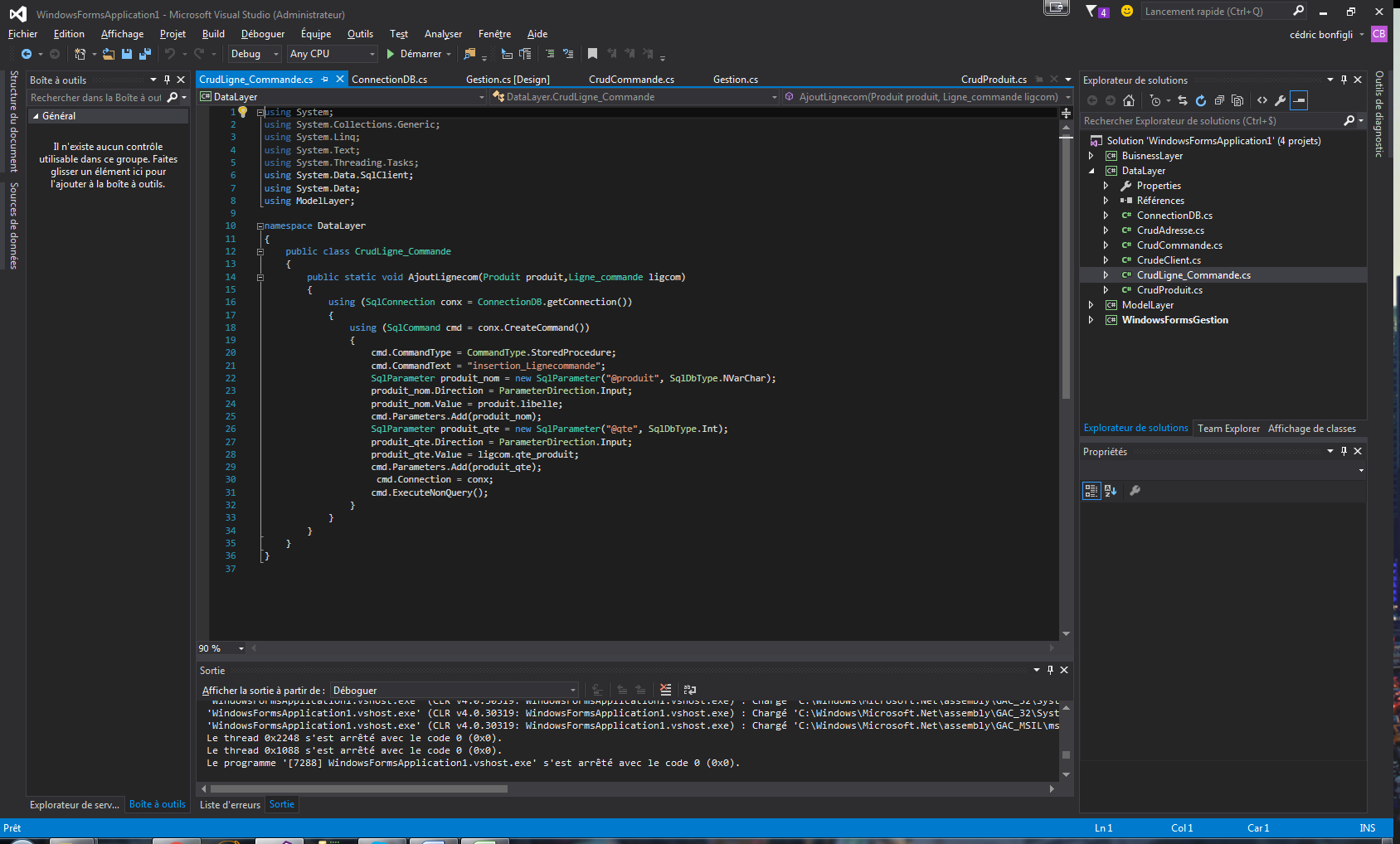
1. Je vous montre une fonction qui permet de ne pas avoir de doublon dans le datagridView de la commande



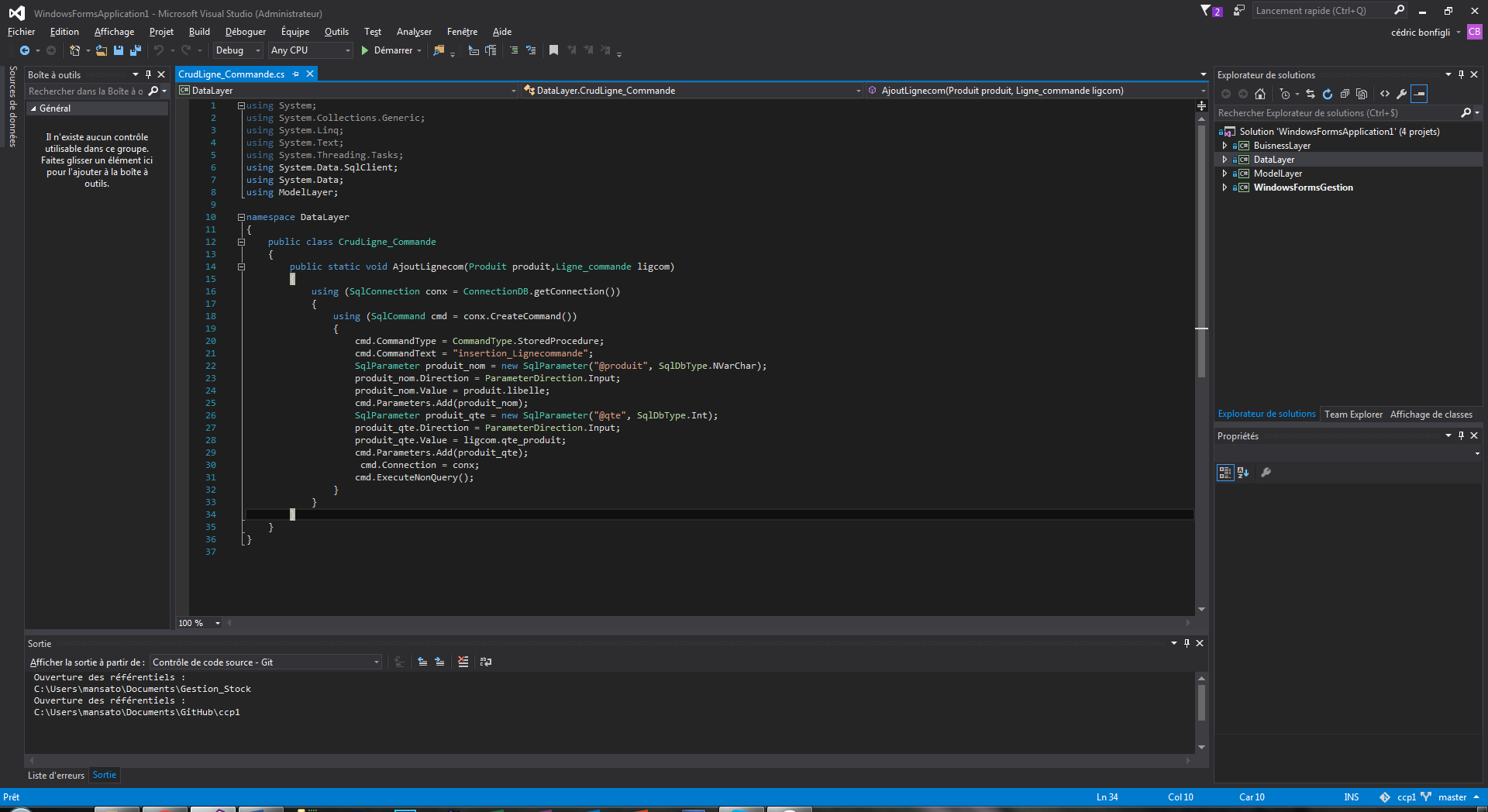
1. La classe connexion



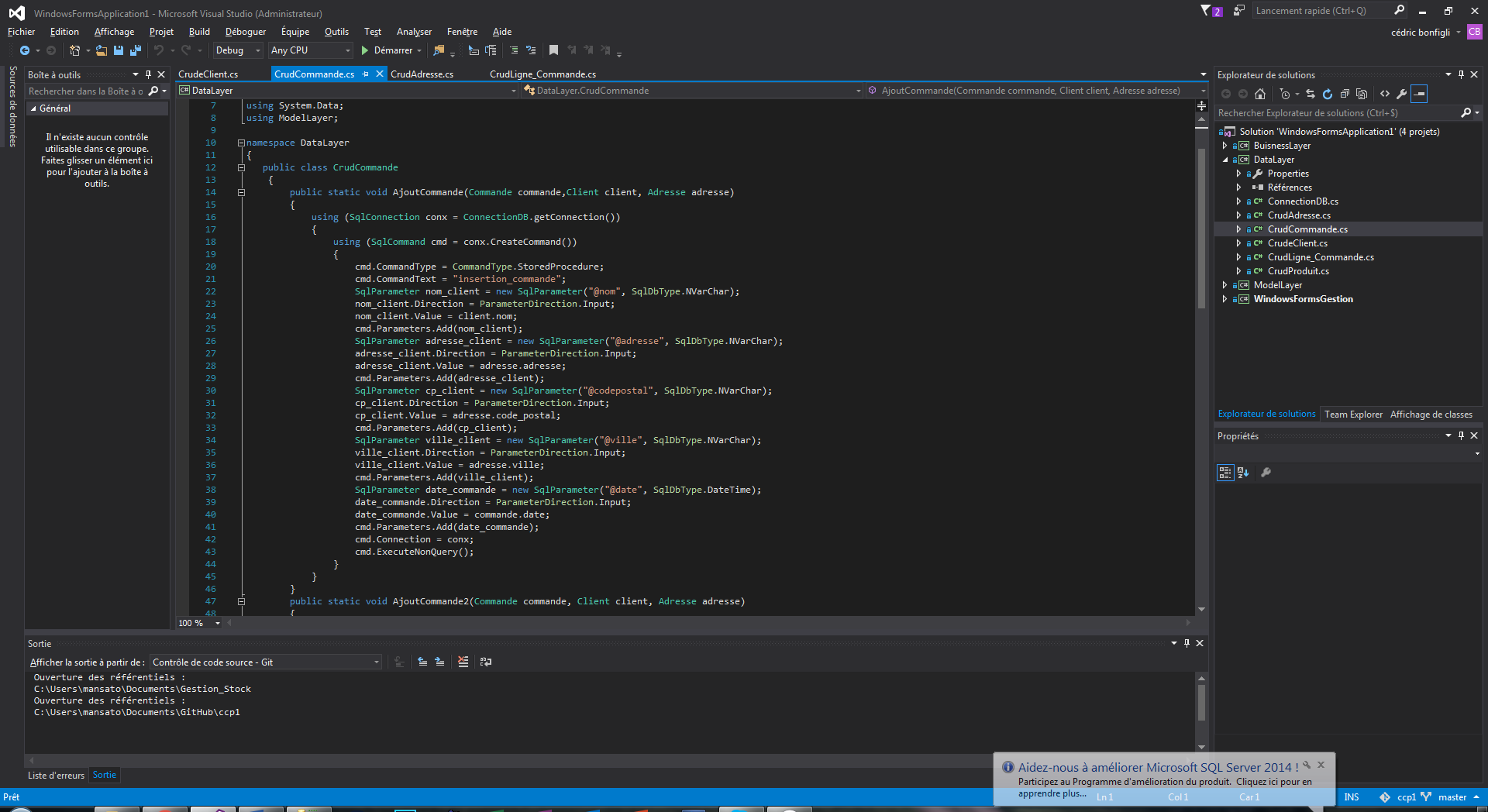
1. Une connexion à la base de données



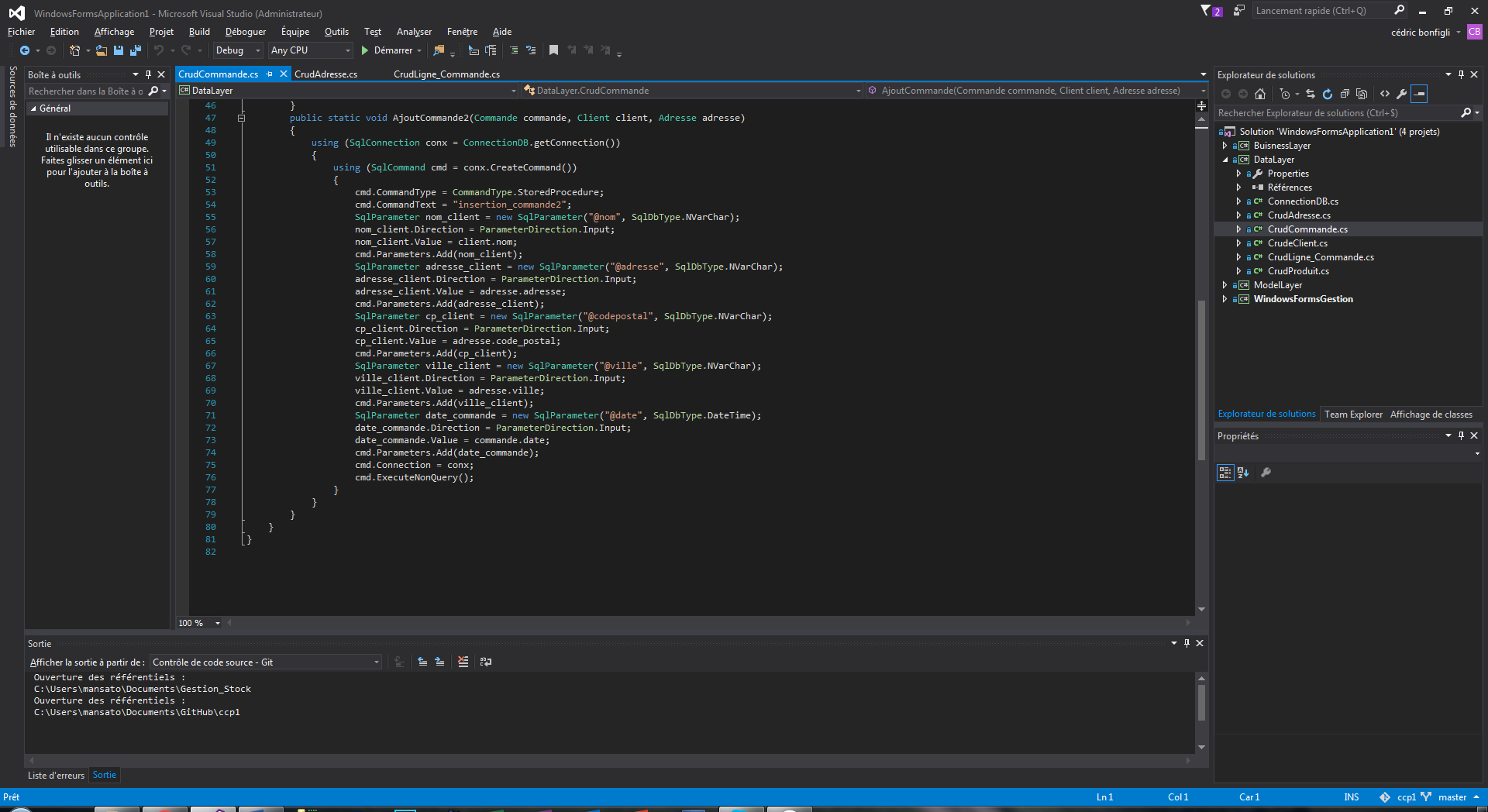
1. Explication des procédures avec le code
2. J’utilise ici une procédure pour mes lignes de commande car je récupère l’id de la table commande et les id des produits. J’ai préférais passer pour une procédure pour que mon code soit plus clair et que je ne fasse pas plusieurs appels a la base de données pour insérée des lignes de commande dans la base.



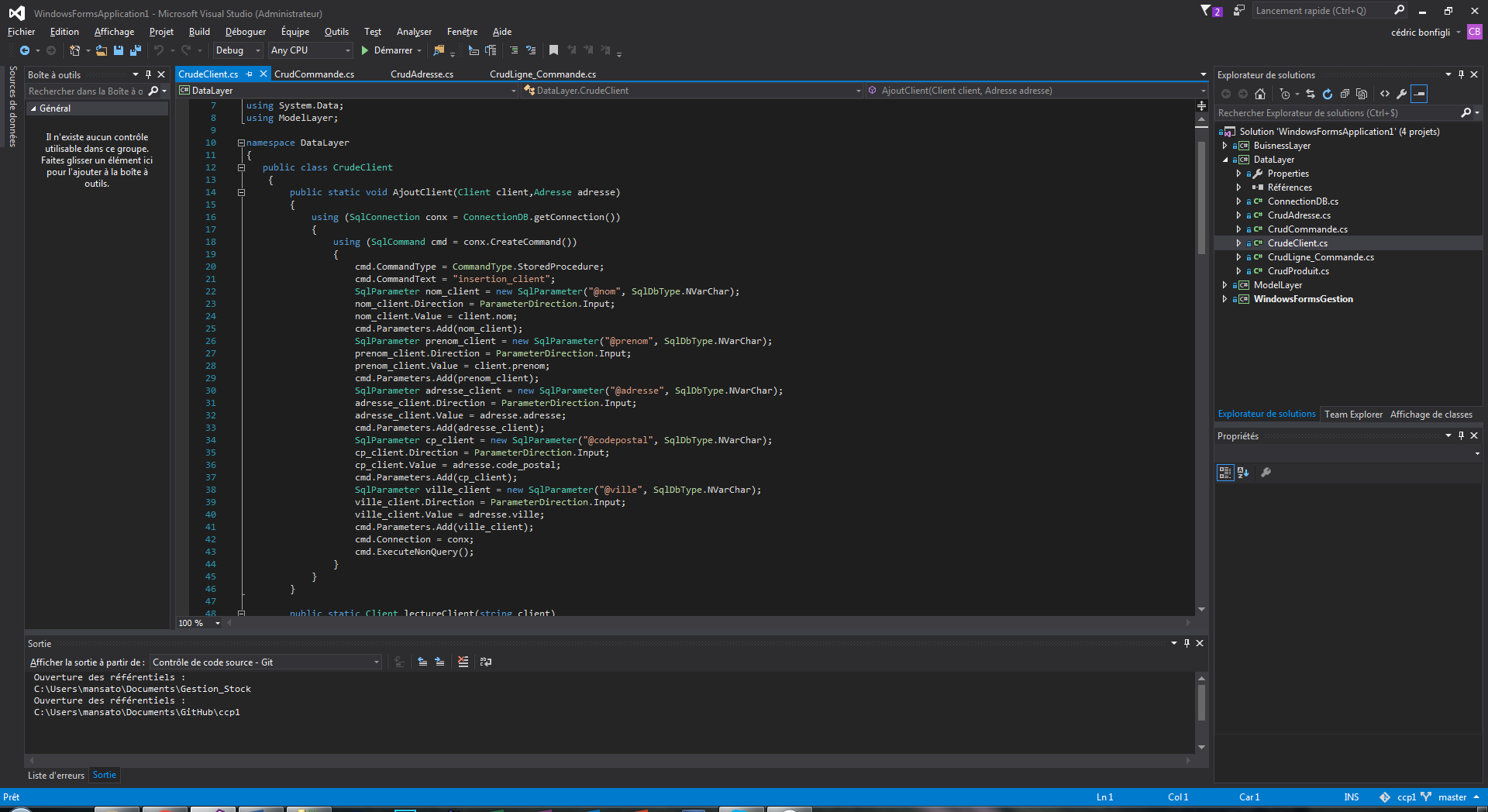
1. J’utilise ici une procédure pour mes commandes car je récupère l’id de la table adresse et les id clients.



1. J’utilise ici une procédure pour mes commandes car je récupère l’id de la table adresse et les id clients. La particularité de cette procédure c’est qu’elle va crée une nouvelle adresse dans la table adresse et récupéré l’id cette procédure sert pour le cas ou le client commande avec une adresse de livraison différente.



1. J’utilise ici une procédure pour mes insertions clients car je récupère l’id de la table adresse pour pouvoir la mettre dans la table client au moment de l’insert.

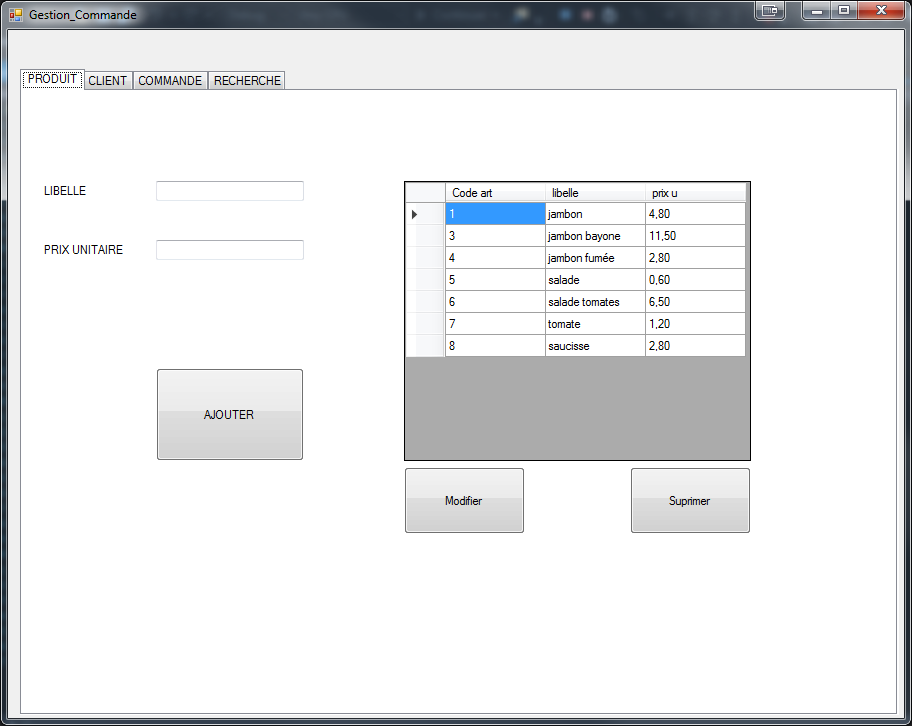


1. Explication sur le logiciel

J’ai voulu faire un logiciel simple avec de fonction simple qui comprend, une création d’article, de client et une prise de commande.

1. Cette page permet d’ajouter les produits.

On peut choisir un produit dans le datagridView il s’affiche dans les textbox sur la gauche il nous suffit de modifier sont libelle ou sont prix ou les 2 et de cliqué sur modifier, on peut aussi le supprimer directement.



1. La fenêtre commande qui nous sert à passer les commandes on tape les premières lettres d’un produit ou le libelle complet et en cliquant sur rechercher il nous sort tous les produits qui commence par la ou les mêmes lettres à ce moment-là on peut choisir un article et lui donner ça quantité et l’ajouter dans le datagridview de la commande.

Pour toute commande on doit choisir un client on peut lui garder son adresse ou la changer et on choisit le jour de la commande souhaiter. De plus si on s’est trompé sur l’ajout d’un produit on peut toujours supprimer une ligne dans le datagridview de la commande avent l’envoie de celle-ci.

