|  |
| --- |
|  |

|  |
| --- |
| **Projet JAVA/ORACLE** |
| PROF BD |
|  |
|  |
|  |
| **Houssem Rekik, Aziz Bouhafs TP 321** |
| **18/01/2013** |
|  |

Contenu

[**I/ INTRODUCTION :** 2](#_Toc346312727)

[**II/ PARTIE COMMUNE :** 3](#_Toc346312728)

[-Schéma relationnel : 3](#_Toc346312729)

[-Analyse détaillée : 3](#_Toc346312730)

[-Diagramme des classes en Java : 4](#_Toc346312731)

[**Classe Affiche :** 4](#_Toc346312732)

[**Class Controleur :** 5](#_Toc346312733)

[**Class Modèle:** 5](#_Toc346312734)

[**Class Vue :** 6](#_Toc346312735)

[-Exemple commenté de test concernant une classe interfaçant la base de données : 8](#_Toc346312736)

[-Copies d’écran dans un manuel utilisateur : 9](#_Toc346312737)

[**III/ Script SQL créant toutes les tables :** 11](#_Toc346312738)

[**Création de la table ingrédient :** 11](#_Toc346312739)

[**Création de la table pizza\_ingrédient :** 14](#_Toc346312740)

[**Création de la table Tarification:** 23](#_Toc346312741)

[**Création de la table Pizza :** 23](#_Toc346312742)

[**Création de la table Client :** 26](#_Toc346312743)

[**Création de la table Commande :** 28](#_Toc346312744)

[**III/ Scripts PL/SQL :** 31](#_Toc346312745)

[Les triggers : 31](#_Toc346312746)

[Code source: Modele\*.java : 34](#_Toc346312747)

[modeleGestion.java : 34](#_Toc346312748)

[modeleLivreur.java : 37](#_Toc346312749)

[ModelePizzeria.java 39](#_Toc346312750)

[**IV) Conclusion:** 42](#_Toc346312751)

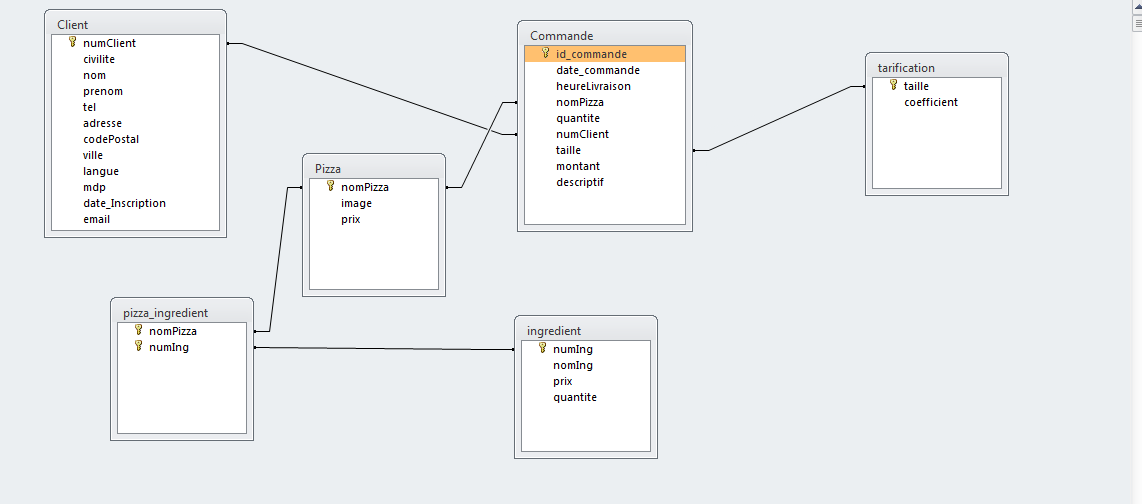
**I/ INTRODUCTION :**

Nous avons tout d’abord crée un site Web permettant la livraison de pizzas à domicile à une heure précise demandée par le client qui devait s’inscrire et s’identifier afin de passer ses commandes. Notre site proposait aux clients différentes sortes de pizzas, de menus, de boissons et de desserts. Les ingrédients d’une pizza pouvaient être ajouté, modifié ou même supprimé afin de respecter idéalement les demandes du client. De plus, un système de paiement sécurisé a été établi permettant de garantir la confiance au client.

Nous avons ensuite programmé avant quelques interfaces permettant aux différents salariés de la pizzéria de prévoir les commandes et de les préparer suivant l’heure de livraison afin qu’elle puisse être prête et après être transportée jusqu’ à destination. Puis une autre interface permettra au livreur de valider la livraison puis nous avons mis en place une interface permettant la gestion des ingrédients disponibles afin d’éviter un épuisement de stock. Une alerte est aussi activée en cas de surplus de commandes à livrer et en cas de stock d’ingrédients limité. De plus, les pizzaiolos peuvent vérifier leur chiffre d’affaire.

**II/ PARTIE COMMUNE :**

-Schéma relationnel :



-Analyse détaillée :

Pour réaliser les pizzas à livrer en fonction de la date de commande, la partie principale qui s’occupera d’afficher ces pizzas sera Java qui cherchera dans la base de données d’Oracle les pizzas par ordre de priorité à réaliser.

Pour permettre aux livreurs de réaliser leurs livraisons efficacement, Java s’occupera de présenter l’adresse et l’heure de livraison de chacune des pizzas à livrer en faisant une requête dans la base de données d’Oracle.

Pour valider la livraison, nous pourrons indiquer par son numéro que la pizza a bien été livrée sur une interface Java et ainsi modifier la valeur du booléen « livrée » qui passera de faux à vrai dans la base de données Oracle.

Pour le gestionnaire de la pizzéria, un historique des ingrédients consommés est établi par une interface Java afin de pouvoir anticiper un manque d’ingrédients.

Puis, nous avons créé un trigger afin d’anticiper un surplus de commandes et un épuisement de stocks des ingrédients.

De plus, nous avons créé une fonction sous oracle pour permettre au pizzaiolo de savoir son chiffre d’affaire.

Ainsi, java va plutôt gérer les interactions entre l’interface et le livreur ou entre l’interface et le gestionnaire ou entre l’interface et le pizzaiolo alors qu’Oracle va plutôt gérer les interactions entre java et la base de données. Si l’on désire contrôler le stock d’ingrédient on va plutôt se tourner vers Oracle mais si l’on désire modifier ou afficher la base de donnée lorsque le livreur saisie un numéro de commande.

-Diagramme des classes en Java :

Notre programme repose sur le modèle MVC.

Ainsi, nous avons créé 3 packages un package pour le Modèle, un package pour le contrôleur et un autre package pour la vue.

**Classe Affiche :**

//afficher les différentes interfaces

**public** **class** ApplicationPizzeria {

}

//connexion à la base de données

**public** **class** OutilSJDBC {

**public** **static** Connection openConnection (String url) { }

**public** **static** ResultSet exec1Requete (String requete, Connection co, **int** type)

{}

**public** **static** **void** closeConnection(Connection co){ }

}

**Class Controleur :**

//controleur pour le Gestionnaire

public class ControleurGestion extends JPanel implements ActionListener,InterfaceControleurPizzeria {

//Constructeur

public ControleurGestion(ModeleGestion modele,InterfaceVueLivreurs vue){}

public void actionPerformed(ActionEvent e) {}

}

// controleur pour pizzaiolo

public class ControleurPizzaiolo extends JPanel implements ActionListener,InterfaceControleurPizzeria{

public ControleurPizzaiolo(ModelePizzeria modele2,InterfaceVueLivreurs vue){}

// Demander a la vue de se mettre a jour

void miseAJour() { }

public void actionPerformed(ActionEvent e) {}

}

**public** **interface** InterfaceControleurPizzeria {}

**Class Modèle:**

**public** **interface** InterfaceModelePizzeria {}

//modèle pour le gestionnaire

public class ModeleGestion implements InterfaceModelePizzeria {

//Constructeur

public ModeleGestion(){}

//Envoie l'historique des ingrédients

public void historyIngr(){}

public ResultSet getRes(){}

public String getCA(){}

public Object miseAJour() {}

public void pizzaPrete(String numPizza) {}

public void infoClient(String text) {}

}

//modèle pour le livreur

public class ModeleLivreurs implements InterfaceModelePizzeria {

public ModeleLivreurs(){}

public void infoClient(String numPizza){}

public ResultSet getRes(){}

}

//modèle pour pizzaiolo

public class ModelePizzeria implements InterfaceModelePizzeria {

public ModelePizzeria(){}

public ResultSet getRes(){}

public void pizzaPrete(String numc){}

public Object miseAJour(){}

public static void main(String[] args) {}

}

**Class Vue :**

**public** **interface** InterfaceVueLivreurs {

**public** **void** miseAJourAffichage();

}

//vue pour le gestionnaire

public class VueGestionnaire extends JPanel implements InterfaceVueLivreurs {

//Constructeur

public VueGestionnaire (ModeleGestion modele){}

public void miseAJourAffichage(){}

}

//vue pour le livreur

public class VueLivreurs extends JLabel implements InterfaceVueLivreurs {

//Constructeur

public VueLivreurs(ModeleLivreurs modele2){}

public void miseAJourAffichage() {}

}

}

//vue pour le pizzaiolo

public class VuePizzeriaPizzaiolo extends JPanel implements InterfaceVueLivreurs {

//Constructeur

public VuePizzeriaPizzaiolo(ModelePizzeria modele2){}

public void miseAJourAffichage(){}

}

-Exemple commenté de test concernant une classe interfaçant la base de données :

Cette classe se connecte à la base de données afin de récupérer les différentes données du client.

package pizza.modeles;

import java.sql.Connection;

import java.sql.ResultSet;

import p.OutilSJDBC;

public class ModeleLivreurs implements InterfaceModelePizzeria {

private OutilSJDBC sql =null;

private ResultSet resultat=null;

//connection à la base de données

Connection co;

public ModeleLivreurs(){

sql=new OutilSJDBC(); co=sql.openConnection("jdbc:oracle:thin:abouhaf/azizhoussem@r2d2.etud.iut-orsay.u-psud.fr:1521:etudom");

}

public void infoClient(String numPizza){

// requite sql pour récupérer les données du client dans la base de données dont le numéro du client est identique au numéro de la pizza

String requete="SELECT nom,prenom,ville,adresse,codePostal,tel ";

requete+="FROM client WHERE numClient="+numPizza;

resultat = sql.exec1Requete(requete, co, 0);

//sql.closeConnection(co);

}

//on retourne le résultat à la classe vue

public ResultSet getRes(){

return resultat;

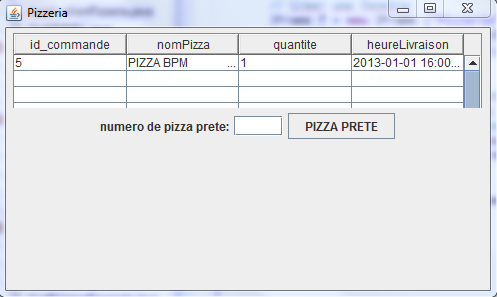
}

}

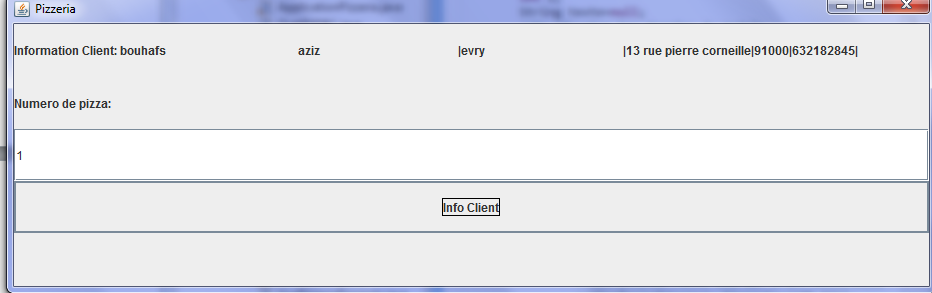
-Copies d’écran dans un manuel utilisateur :

LISTE DES PIZZAS A PREPARER PAR ORDRE D’HEURE DE LIVRAISON (order by heureLivraison)

Si la pizza est prête, on saisit son numéro dans le champ de saisie.

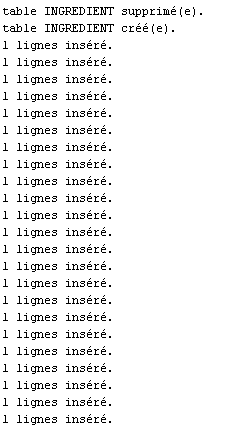


Lorsqu’on saisit un numéro de pizza on a l’adresse du client et l’heure de livraison qui s’affiche.



**III/ Script SQL créant toutes les tables :**

**Création de la table ingrédient :**





CREATE TABLE ingredient (

numIng number(4) NOT NULL,

nomIng varchar(150) NOT NULL,

quantite number(4) NOT NULL,

prix float NOT NULL,

PRIMARY KEY (numIng)

);

INSERT INTO ingredient (numIng, nomIng,prix,quantite) VALUES

(1, 'Mozzarelle',2,10);

INSERT INTO ingredient (numIng, nomIng,prix,quantite) VALUES

(2, 'Jambon cru Aoste',2,10);

INSERT INTO ingredient (numIng, nomIng,prix,quantite) VALUES

(3, 'Pepperoni',2,10);

INSERT INTO ingredient (numIng, nomIng,prix,quantite) VALUES

(4, 'Jambon',2,10);

INSERT INTO ingredient (numIng, nomIng,prix,quantite) VALUES

(5, 'Merguez',2,10);

INSERT INTO ingredient (numIng, nomIng,prix,quantite) VALUES

(6, 'Boeuf',2,10);

INSERT INTO ingredient (numIng, nomIng,prix,quantite) VALUES

(7, 'Poulet',2,10);

INSERT INTO ingredient (numIng, nomIng,prix,quantite) VALUES

(8, 'Lardons',2,10);

INSERT INTO ingredient (numIng, nomIng,prix,quantite) VALUES

(9, 'Cantal',2,10);

INSERT INTO ingredient (numIng, nomIng,prix,quantite) VALUES

(10, 'St-Nectaire',2,10);

INSERT INTO ingredient (numIng, nomIng,prix,quantite) VALUES

(11, 'Poivrons verts',2,10);

INSERT INTO ingredient (numIng, nomIng,prix,quantite) VALUES

(12, 'Champignons',2,10);

INSERT INTO ingredient (numIng, nomIng,prix,quantite) VALUES

(13, 'Tomates',2,10);

INSERT INTO ingredient (numIng, nomIng,prix,quantite) VALUES

(14, 'Ananas',2,10);

INSERT INTO ingredient (numIng, nomIng,prix,quantite) VALUES

(15, 'Oignons',2,10);

INSERT INTO ingredient (numIng, nomIng,prix,quantite) VALUES

(16, 'Piments',2,10);

INSERT INTO ingredient (numIng, nomIng,prix,quantite) VALUES

(17, 'Thon',2,10);

INSERT INTO ingredient (numIng, nomIng,prix,quantite) VALUES

(18, 'Olives noires',2,10);

INSERT INTO ingredient (numIng, nomIng,prix,quantite) VALUES

(19, 'Pommes de terre',2,10);

INSERT INTO ingredient (numIng, nomIng,prix,quantite) VALUES

(20, 'Saumon fumé',2,10);

INSERT INTO ingredient (numIng, nomIng,prix,quantite) VALUES

(21, 'Emincés de poulet (Halal)',2,10);

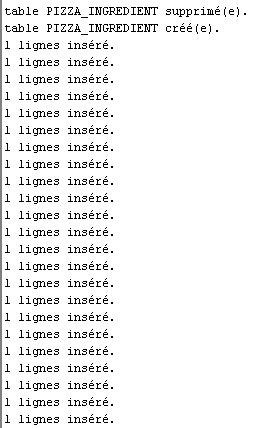
INSERT INTO ingredient (numIng, nomIng,prix,quantite) VALUES

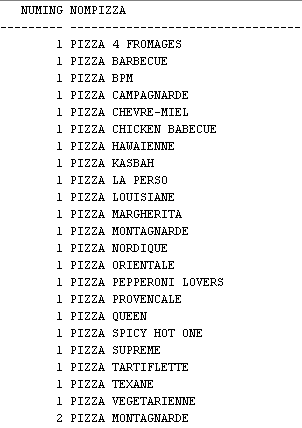
(22, 'Miel',2,10);

INSERT INTO ingredient (numIng, nomIng,prix,quantite) VALUES

(23, 'Fromage de chèvre',2,10);

**Création de la table pizza\_ingrédient :**





create table pizza\_ingredient(

numIng number(4) NOT NULL,

nomPizza char(50) NOT NULL,

PRIMARY KEY(numIng,nomPizza)

);

INSERT INTO pizza\_ingredient (numIng,nomPizza) VALUES

(1,'PIZZA 4 FROMAGES');

INSERT INTO pizza\_ingredient (numIng,nomPizza) VALUES

(9,'PIZZA 4 FROMAGES');

INSERT INTO pizza\_ingredient (numIng,nomPizza) VALUES

(13,'PIZZA 4 FROMAGES');

INSERT INTO pizza\_ingredient (numIng,nomPizza) VALUES

(23,'PIZZA 4 FROMAGES');

INSERT INTO pizza\_ingredient (numIng,nomPizza) VALUES

(24,'PIZZA 4 FROMAGES');

INSERT INTO pizza\_ingredient (numIng,nomPizza) VALUES

(1,'PIZZA BARBECUE');

INSERT INTO pizza\_ingredient (numIng,nomPizza) VALUES

(6,'PIZZA BARBECUE');

INSERT INTO pizza\_ingredient (numIng,nomPizza) VALUES

(8,'PIZZA BARBECUE');

INSERT INTO pizza\_ingredient (numIng,nomPizza) VALUES

(15,'PIZZA BARBECUE');

INSERT INTO pizza\_ingredient (numIng,nomPizza) VALUES

(1,'PIZZA BPM');

INSERT INTO pizza\_ingredient (numIng,nomPizza) VALUES

(5,'PIZZA BPM');

INSERT INTO pizza\_ingredient (numIng,nomPizza) VALUES

(6,'PIZZA BPM');

INSERT INTO pizza\_ingredient (numIng,nomPizza) VALUES

(7,'PIZZA BPM');

INSERT INTO pizza\_ingredient (numIng,nomPizza) VALUES

(1,'PIZZA CAMPAGNARDE');

INSERT INTO pizza\_ingredient (numIng,nomPizza) VALUES

(6,'PIZZA CAMPAGNARDE');

INSERT INTO pizza\_ingredient (numIng,nomPizza) VALUES

(13,'PIZZA CAMPAGNARDE');

INSERT INTO pizza\_ingredient (numIng,nomPizza) VALUES

(15,'PIZZA CAMPAGNARDE');

INSERT INTO pizza\_ingredient (numIng,nomPizza) VALUES

(16,'PIZZA CAMPAGNARDE');

INSERT INTO pizza\_ingredient (numIng,nomPizza) VALUES

(1,'PIZZA CHEVRE-MIEL');

INSERT INTO pizza\_ingredient (numIng,nomPizza) VALUES

(22,'PIZZA CHEVRE-MIEL');

INSERT INTO pizza\_ingredient (numIng,nomPizza) VALUES

(23,'PIZZA CHEVRE-MIEL');

INSERT INTO pizza\_ingredient (numIng,nomPizza) VALUES

(1,'PIZZA CHICKEN BABECUE');

INSERT INTO pizza\_ingredient (numIng,nomPizza) VALUES

(7,'PIZZA CHICKEN BABECUE');

INSERT INTO pizza\_ingredient (numIng,nomPizza) VALUES

(11,'PIZZA CHICKEN BABECUE');

INSERT INTO pizza\_ingredient (numIng,nomPizza) VALUES

(12,'PIZZA CHICKEN BABECUE');

INSERT INTO pizza\_ingredient (numIng,nomPizza) VALUES

(15,'PIZZA CHICKEN BABECUE');

INSERT INTO pizza\_ingredient (numIng,nomPizza) VALUES

(1,'PIZZA HAWAIENNE');

INSERT INTO pizza\_ingredient (numIng,nomPizza) VALUES

(4,'PIZZA HAWAIENNE');

INSERT INTO pizza\_ingredient (numIng,nomPizza) VALUES

(13,'PIZZA HAWAIENNE');

INSERT INTO pizza\_ingredient (numIng,nomPizza) VALUES

(14,'PIZZA HAWAIENNE');

INSERT INTO pizza\_ingredient (numIng,nomPizza) VALUES

(1,'PIZZA KASBAH');

INSERT INTO pizza\_ingredient (numIng,nomPizza) VALUES

(13,'PIZZA KASBAH');

INSERT INTO pizza\_ingredient (numIng,nomPizza) VALUES

(15,'PIZZA KASBAH');

INSERT INTO pizza\_ingredient (numIng,nomPizza) VALUES

(21,'PIZZA KASBAH');

INSERT INTO pizza\_ingredient (numIng,nomPizza) VALUES

(1,'PIZZA LA PERSO');

INSERT INTO pizza\_ingredient (numIng,nomPizza) VALUES

(13,'PIZZA LA PERSO');

INSERT INTO pizza\_ingredient (numIng,nomPizza) VALUES

(1,'PIZZA LOUISIANE');

INSERT INTO pizza\_ingredient (numIng,nomPizza) VALUES

(7,'PIZZA LOUISIANE');

INSERT INTO pizza\_ingredient (numIng,nomPizza) VALUES

(11,'PIZZA LOUISIANE');

INSERT INTO pizza\_ingredient (numIng,nomPizza) VALUES

(12,'PIZZA LOUISIANE');

INSERT INTO pizza\_ingredient (numIng,nomPizza) VALUES

(13,'PIZZA LOUISIANE');

INSERT INTO pizza\_ingredient (numIng,nomPizza) VALUES

(15,'PIZZA LOUISIANE');

INSERT INTO pizza\_ingredient (numIng,nomPizza) VALUES

(1,'PIZZA MARGHERITA');

INSERT INTO pizza\_ingredient (numIng,nomPizza) VALUES

(13,'PIZZA MARGHERITA');

INSERT INTO pizza\_ingredient (numIng,nomPizza) VALUES

(1,'PIZZA MONTAGNARDE');

INSERT INTO pizza\_ingredient (numIng,nomPizza) VALUES

(2,'PIZZA MONTAGNARDE');

INSERT INTO pizza\_ingredient (numIng,nomPizza) VALUES

(10,'PIZZA MONTAGNARDE');

INSERT INTO pizza\_ingredient (numIng,nomPizza) VALUES

(12,'PIZZA MONTAGNARDE');

INSERT INTO pizza\_ingredient (numIng,nomPizza) VALUES

(1,'PIZZA NORDIQUE');

INSERT INTO pizza\_ingredient (numIng,nomPizza) VALUES

(20,'PIZZA NORDIQUE');

INSERT INTO pizza\_ingredient (numIng,nomPizza) VALUES

(1,'PIZZA ORIENTALE');

INSERT INTO pizza\_ingredient (numIng,nomPizza) VALUES

(5,'PIZZA ORIENTALE');

INSERT INTO pizza\_ingredient (numIng,nomPizza) VALUES

(12,'PIZZA ORIENTALE');

INSERT INTO pizza\_ingredient (numIng,nomPizza) VALUES

(13,'PIZZA ORIENTALE');

INSERT INTO pizza\_ingredient (numIng,nomPizza) VALUES

(1,'PIZZA PEPPERONI LOVERS');

INSERT INTO pizza\_ingredient (numIng,nomPizza) VALUES

(13,'PIZZA PEPPERONI LOVERS');

INSERT INTO pizza\_ingredient (numIng,nomPizza) VALUES

(15,'PIZZA PEPPERONI LOVERS');

INSERT INTO pizza\_ingredient (numIng,nomPizza) VALUES

(1,'PIZZA PROVENCALE');

INSERT INTO pizza\_ingredient (numIng,nomPizza) VALUES

(13,'PIZZA PROVENCALE');

INSERT INTO pizza\_ingredient (numIng,nomPizza) VALUES

(15,'PIZZA PROVENCALE');

INSERT INTO pizza\_ingredient (numIng,nomPizza) VALUES

(16,'PIZZA PROVENCALE');

INSERT INTO pizza\_ingredient (numIng,nomPizza) VALUES

(17,'PIZZA PROVENCALE');

INSERT INTO pizza\_ingredient (numIng,nomPizza) VALUES

(18,'PIZZA PROVENCALE');

INSERT INTO pizza\_ingredient (numIng,nomPizza) VALUES

(1,'PIZZA QUEEN');

INSERT INTO pizza\_ingredient (numIng,nomPizza) VALUES

(4,'PIZZA QUEEN');

INSERT INTO pizza\_ingredient (numIng,nomPizza) VALUES

(12,'PIZZA QUEEN');

INSERT INTO pizza\_ingredient (numIng,nomPizza) VALUES

(13,'PIZZA QUEEN');

INSERT INTO pizza\_ingredient (numIng,nomPizza) VALUES

(1,'PIZZA SPICY HOT ONE');

INSERT INTO pizza\_ingredient (numIng,nomPizza) VALUES

(6,'PIZZA SPICY HOT ONE');

INSERT INTO pizza\_ingredient (numIng,nomPizza) VALUES

(13,'PIZZA SPICY HOT ONE');

INSERT INTO pizza\_ingredient (numIng,nomPizza) VALUES

(15,'PIZZA SPICY HOT ONE');

INSERT INTO pizza\_ingredient (numIng,nomPizza) VALUES

(16,'PIZZA SPICY HOT ONE');

INSERT INTO pizza\_ingredient (numIng,nomPizza) VALUES

(1,'PIZZA SUPREME');

INSERT INTO pizza\_ingredient (numIng,nomPizza) VALUES

(3,'PIZZA SUPREME');

INSERT INTO pizza\_ingredient (numIng,nomPizza) VALUES

(6,'PIZZA SUPREME');

INSERT INTO pizza\_ingredient (numIng,nomPizza) VALUES

(11,'PIZZA SUPREME');

INSERT INTO pizza\_ingredient (numIng,nomPizza) VALUES

(12,'PIZZA SUPREME');

INSERT INTO pizza\_ingredient (numIng,nomPizza) VALUES

(13,'PIZZA SUPREME');

INSERT INTO pizza\_ingredient (numIng,nomPizza) VALUES

(15,'PIZZA SUPREME');

INSERT INTO pizza\_ingredient (numIng,nomPizza) VALUES

(1,'PIZZA TARTIFLETTE');

INSERT INTO pizza\_ingredient (numIng,nomPizza) VALUES

(8,'PIZZA TARTIFLETTE');

INSERT INTO pizza\_ingredient (numIng,nomPizza) VALUES

(10,'PIZZA TARTIFLETTE');

INSERT INTO pizza\_ingredient (numIng,nomPizza) VALUES

(19,'PIZZA TARTIFLETTE');

INSERT INTO pizza\_ingredient (numIng,nomPizza) VALUES

(1,'PIZZA TEXANE');

INSERT INTO pizza\_ingredient (numIng,nomPizza) VALUES

(4,'PIZZA TEXANE');

INSERT INTO pizza\_ingredient (numIng,nomPizza) VALUES

(6,'PIZZA TEXANE');

INSERT INTO pizza\_ingredient (numIng,nomPizza) VALUES

(8,'PIZZA TEXANE');

INSERT INTO pizza\_ingredient (numIng,nomPizza) VALUES

(12,'PIZZA TEXANE');

INSERT INTO pizza\_ingredient (numIng,nomPizza) VALUES

(13,'PIZZA TEXANE');

INSERT INTO pizza\_ingredient (numIng,nomPizza) VALUES

(15,'PIZZA TEXANE');

INSERT INTO pizza\_ingredient (numIng,nomPizza) VALUES

(1,'PIZZA VEGETARIENNE');

INSERT INTO pizza\_ingredient (numIng,nomPizza) VALUES

(12,'PIZZA VEGETARIENNE');

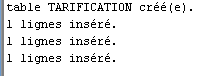
INSERT INTO pizza\_ingredient (numIng,nomPizza) VALUES

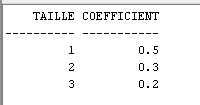
(13,'PIZZA VEGETARIENNE');

INSERT INTO pizza\_ingredient (numIng,nomPizza) VALUES

(15,'PIZZA VEGETARIENNE');

**Création de la table Tarification:**





create table tarification(

taille number(2) NOT NULL,

coefficient float NOT NULL,

PRIMARY KEY (taille)

);

INSERT INTO tarification (taille,coefficient) VALUES

(1,0.5);

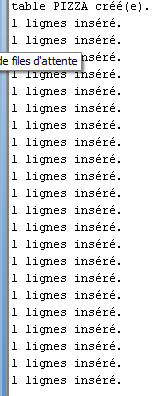
INSERT INTO tarification (taille,coefficient) VALUES

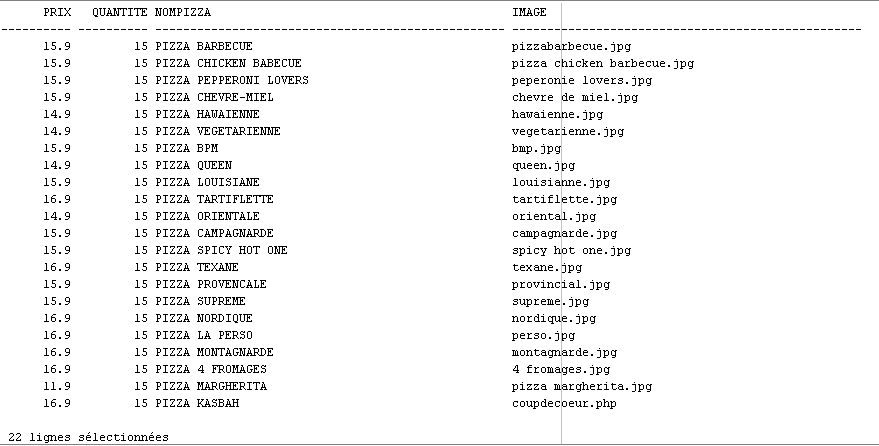
(2,0.3);

INSERT INTO tarification (taille,coefficient) VALUES

(3,0.2);

**Création de la table Pizza :**





CREATE TABLE pizza (

prix float NOT NULL,

quantite number(4) NOT NULL,

nomPizza char(50) NOT NULL,

image char(50) NOT NULL,

PRIMARY KEY (nomPizza)

);

INSERT INTO pizza (prix, quantite, nomPizza, image) VALUES

(15.9, 15, 'PIZZA BARBECUE', 'pizzabarbecue.jpg');

INSERT INTO pizza (prix, quantite, nomPizza, image) VALUES

(15.9, 15, 'PIZZA CHICKEN BABECUE', 'pizza chicken barbecue.jpg');

INSERT INTO pizza (prix, quantite, nomPizza, image) VALUES

(15.9, 15, 'PIZZA PEPPERONI LOVERS', 'peperonie lovers.jpg');

INSERT INTO pizza (prix, quantite, nomPizza, image) VALUES

( 15.9, 15, 'PIZZA CHEVRE-MIEL', 'chevre de miel.jpg');

INSERT INTO pizza ( prix, quantite, nomPizza, image) VALUES

( 14.9, 15, 'PIZZA HAWAIENNE', 'hawaienne.jpg');

INSERT INTO pizza ( prix, quantite, nomPizza, image) VALUES

( 14.9, 15, 'PIZZA VEGETARIENNE', 'vegetarienne.jpg');

INSERT INTO pizza (prix, quantite, nomPizza, image) VALUES

( 15.9, 15, 'PIZZA BPM', 'bmp.jpg');

INSERT INTO pizza (prix, quantite, nomPizza, image) VALUES

( 14.9, 15, 'PIZZA QUEEN', 'queen.jpg');

INSERT INTO pizza (prix, quantite, nomPizza, image) VALUES

( 15.9, 15, 'PIZZA LOUISIANE', 'louisianne.jpg');

INSERT INTO pizza (prix, quantite, nomPizza, image) VALUES

( 16.9, 15, 'PIZZA TARTIFLETTE', 'tartiflette.jpg');

INSERT INTO pizza (prix, quantite, nomPizza, image) VALUES

( 14.9, 15, 'PIZZA ORIENTALE', 'oriental.jpg');

INSERT INTO pizza ( prix, quantite, nomPizza, image) VALUES

( 15.9, 15, 'PIZZA CAMPAGNARDE', 'campagnarde.jpg');

INSERT INTO pizza (prix, quantite, nomPizza, image) VALUES

( 15.9, 15, 'PIZZA SPICY HOT ONE', 'spicy hot one.jpg');

INSERT INTO pizza (prix, quantite, nomPizza, image) VALUES

( 16.9, 15, 'PIZZA TEXANE ', 'texane.jpg');

INSERT INTO pizza (prix, quantite, nomPizza, image) VALUES

( 15.9, 15, 'PIZZA PROVENCALE', 'provincial.jpg');

INSERT INTO pizza (prix, quantite, nomPizza, image) VALUES

( 15.9, 15, 'PIZZA SUPREME', 'supreme.jpg');

INSERT INTO pizza (prix, quantite, nomPizza, image) VALUES

( 16.9, 15, 'PIZZA NORDIQUE', 'nordique.jpg');

INSERT INTO pizza (prix, quantite, nomPizza, image) VALUES

( 16.9, 15, 'PIZZA LA PERSO', 'perso.jpg');

INSERT INTO pizza (prix, quantite, nomPizza, image) VALUES

( 16.9, 15, 'PIZZA MONTAGNARDE', 'montagnarde.jpg');

INSERT INTO pizza (prix, quantite, nomPizza, image) VALUES

( 16.9, 15, 'PIZZA 4 FROMAGES', '4 fromages.jpg');

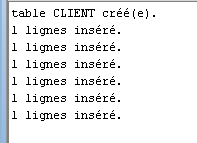
INSERT INTO pizza (prix, quantite, nomPizza, image) VALUES

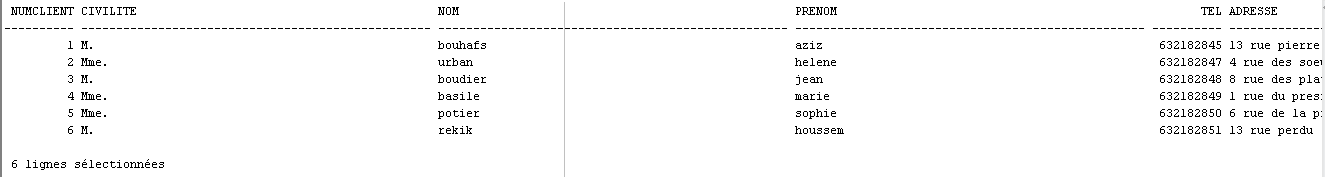
( 11.9, 15, 'PIZZA MARGHERITA', 'pizza margherita.jpg');

INSERT INTO pizza (prix, quantite, nomPizza, image) VALUES

( 16.9, 15, 'PIZZA KASBAH', 'coupdecoeur.php');

**Création de la table Client :**





CREATE TABLE client(

numClient number(4) NOT NULL,

civilite char(50) NOT NULL,

nom char(50) NOT NULL,

prenom char(50) NOT NULL,

tel number(15) NOT NULL,

adresse varchar2(250) NOT NULL,

codePostal number(6) NOT NULL,

ville char(50) NOT NULL,

langue char(50) NOT NULL,

mdp varchar2(255) NOT NULL,

date\_inscription date NOT NULL,

email varchar2(250) NOT NULL,

PRIMARY KEY (numClient));

INSERT INTO client (numClient, civilite, nom, prenom, tel, adresse, codePostal, ville, langue, mdp, date\_inscription, email) VALUES

(1, 'M.', 'bouhafs', 'aziz', 0632182845, '13 rue pierre corneille', 91000, 'evry', 'francais','123',TO\_DATE('16/01/2013','DD/MM/YYYY'),'aziz.bouhafs@hotmail.fr');

INSERT INTO client (numClient, civilite, nom, prenom, tel, adresse, codePostal, ville, langue, mdp, date\_inscription, email) VALUES

(2, 'Mme.', 'urban', 'helene', 0632182847, '4 rue des soeurs', 91440, 'bures-sur-yvette', 'francais','123456',TO\_DATE('01/01/2013','DD/MM/YYYY'),'helene@hotmail.fr');

INSERT INTO client (numClient, civilite, nom, prenom, tel, adresse, codePostal, ville, langue, mdp, date\_inscription, email) VALUES

(3, 'M.', 'boudier', 'jean', 0632182848, '8 rue des platines', 91000, 'evry', 'francais','bob93',TO\_DATE('06/01/2013','DD/MM/YYYY'),'jean@hotmail.fr');

INSERT INTO client (numClient, civilite, nom, prenom, tel, adresse, codePostal, ville, langue, mdp, date\_inscription, email) VALUES

(4, 'Mme.', 'basile', 'marie', 0632182849, '1 rue du president kenedy', 91000, 'evry', 'francais','ss22',TO\_DATE('06/01/2013','DD/MM/YYYY'),'marie@hotmail.fr');

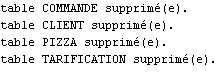
INSERT INTO client (numClient, civilite, nom, prenom, tel, adresse, codePostal, ville, langue, mdp, date\_inscription, email) VALUES

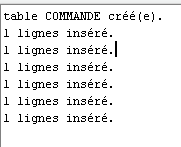
(5, 'Mme.', 'potier', 'sophie', 0632182850, '6 rue de la piscine', 91000, 'evry', 'francais','91000',TO\_DATE('10/01/2013','DD/MM/YYYY'),'sophie@hotmail.fr');

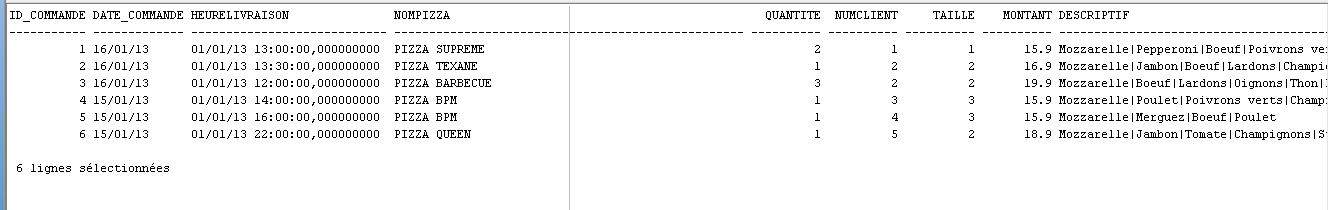
INSERT INTO client (numClient, civilite, nom, prenom, tel, adresse, codePostal, ville, langue, mdp, date\_inscription, email) VALUES

(6, 'M.', 'rekik', 'houssem', 0632182851, '13 rue perdu', 91000, 'evry', 'francais','evry',TO\_DATE('12/01/2013','DD/MM/YYYY'),'houssem@hotmail.fr');

**Création de la table Commande :**







CREATE TABLE commande(

id\_commande number(4) NOT NULL,

date\_commande date NOT NULL,

heurelivraison timestamp NOT NULL,

nomPizza char(50) NOT NULL,

quantite number(4) NOT NULL,

numClient number(4) NOT NULL,

taille number(4) NOT NULL,

montant float NOT NULL,

descriptif varchar2(255) NOT NULL,

PRIMARY KEY (id\_commande),

constraint fk1\_t2\_t1 foreign key (numClient) references client (numClient),

constraint fk2\_t2\_t1 foreign key (nomPizza) references pizza (nomPizza),

constraint fk3\_t2\_t1 foreign key (taille) references tarification (taille)

);

INSERT INTO commande (id\_commande,date\_commande,heurelivraison,nomPizza,quantite,numClient,taille,montant,descriptif) VALUES

(1,TO\_DATE('16/01/2013','DD/MM/YYYY'),TO\_DATE('13:00:00','HH24:MI:SS'),'PIZZA SUPREME',2,1,1,15.9,'Mozzarelle|Pepperoni|Boeuf|Poivrons verts|Champignons|Tomates|Oignons');

INSERT INTO commande (id\_commande,date\_commande,heurelivraison,nomPizza,quantite,numClient,taille,montant,descriptif) VALUES

(2,TO\_DATE('16/01/2013','DD/MM/YYYY'),TO\_DATE('13:30:00','HH24:MI:SS'),'PIZZA TEXANE',1,2,2,16.9,'Mozzarelle|Jambon|Boeuf|Lardons|Champignons|Tomates|Oignons');

INSERT INTO commande (id\_commande,date\_commande,heurelivraison,nomPizza,quantite,numClient,taille,montant,descriptif) VALUES

(3,TO\_DATE('16/01/2013','DD/MM/YYYY'),TO\_DATE('12:00:00','HH24:MI:SS'),'PIZZA BARBECUE',3,2,2,19.9,'Mozzarelle|Boeuf|Lardons|Oignons|Thon|Piments');

INSERT INTO commande (id\_commande,date\_commande,heurelivraison,nomPizza,quantite,numClient,taille,montant,descriptif) VALUES

(4,TO\_DATE('15/01/2013','DD/MM/YYYY'),TO\_DATE('14:00:00','HH24:MI:SS'),'PIZZA BPM',1,3,3,15.9,'Mozzarelle|Poulet|Poivrons verts|Champignons');

INSERT INTO commande (id\_commande,date\_commande,heurelivraison,nomPizza,quantite,numClient,taille,montant,descriptif) VALUES

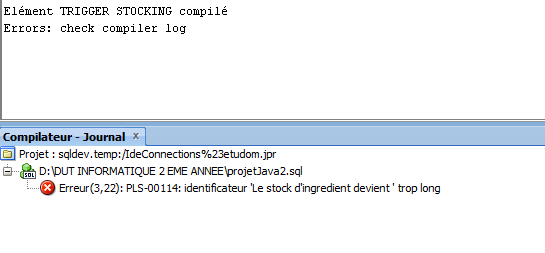
(5,TO\_DATE('15/01/2013','DD/MM/YYYY'),TO\_DATE('16:00:00','HH24:MI:SS'),'PIZZA BPM',1,4,3,15.9,'Mozzarelle|Merguez|Boeuf|Poulet');

INSERT INTO commande (id\_commande,date\_commande,heurelivraison,nomPizza,quantite,numClient,taille,montant,descriptif) VALUES

(6,TO\_DATE('15/01/2013','DD/MM/YYYY'),TO\_DATE('22:00:00','HH24:MI:SS'),'PIZZA QUEEN',1,5,2,18.9,'Mozzarelle|Jambon|Tomate|Champignons|St-Nectaire|Thon');

**III/ Scripts PL/SQL :**

Les triggers :



create trigger stockIng

AFTER UPDATE ON ingredient

for each row

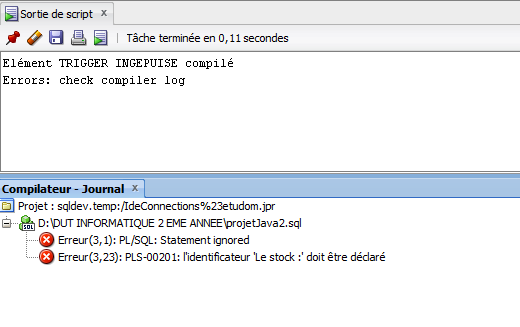
BEGIN

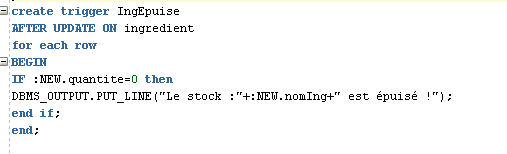
IF :NEW.quantite < 2 then

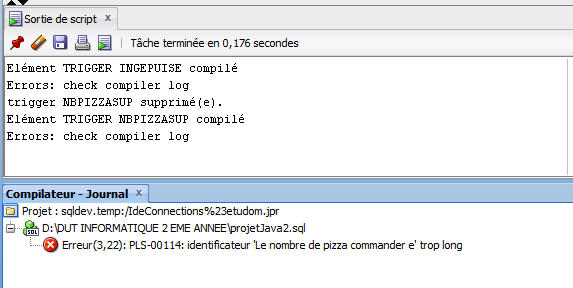
DBMS\_OUTPUT.PUT\_LINE("Le stock d'ingredient devient critique !");

end if;

end;







create trigger nbPizzaSup

AFTER UPDATE ON commande

for each row

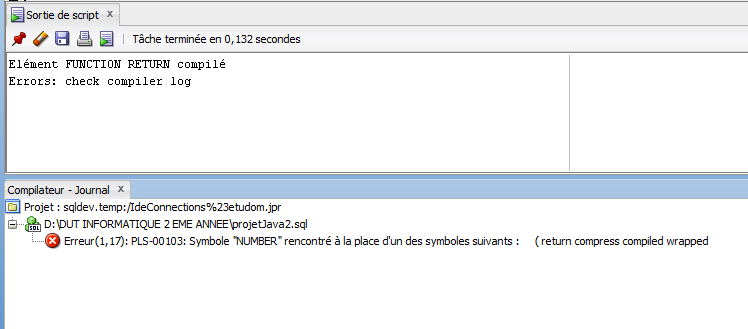
BEGIN

IF (:NEW.quantite > 20) then

DBMS\_OUTPUT.PUT\_LINE("Le nombre de pizza commander est trop élévé !");

end if;

end;



create or replace function return number is chiffreAff

begin

select sum(montant) into chiffreAff

from commande

return chiffreAff;

end;

Code source: Modele\*.java :

modeleGestion.java :

package pizza.modeles;

import java.sql.Connection;

import java.sql.ResultSet;

import java.sql.SQLException;

import p.OutilSJDBC;

public class ModeleGestion implements InterfaceModelePizzeria {

private OutilSJDBC sql =null;

private ResultSet resultat=null;

private Object tab\_ingr[][]=null;

Connection co;

private String ca=null;

private String requete;

//Constructeur

public ModeleGestion(){

sql=new OutilSJDBC();

//Connection à la base de données

co=sql.openConnection("jdbc:oracle:thin:abouhaf/azizhoussem@r2d2.etud.iut-orsay.u-psud.fr:1521:etudom");

//Requete

requete="chiffreAff() ";

//Resultat de le requete

resultat = sql.exec1Requete(requete, co, 0);

try{

while(resultat.next()){

ca=resultat.getString(1);

}

}

catch(SQLException e){

System.out.println("Pb lors du parcours");

}

}

//Envoie l'historique des ingrédients

public void historyIngr(){

//Tableau d'object contenant les lignes de la requete

tab\_ingr = new Object[100][3];

//Requete

requete="SELECT numIngredient,nomIngredient,quantite FROM ingredients";

//Resultat de le requete

resultat = sql.exec1Requete(requete, co, 0);

}

public ResultSet getRes()

{

return resultat;

}

public String getCA(){

return ca;

}

public Object miseAJour() {

// TODO Auto-generated method stub

return null;

}

public void pizzaPrete(String numPizza) {

// TODO Auto-generated method stub

}

public void infoClient(String text) {

// TODO Auto-generated method stub

}

}

modeleLivreur.java :

package pizza.modeles;

import java.sql.Connection;

import java.sql.ResultSet;

import p.OutilSJDBC;

public class ModeleLivreurs implements InterfaceModelePizzeria {

private OutilSJDBC sql =null;

private ResultSet resultat=null;

Connection co;

public ModeleLivreurs(){

sql=new OutilSJDBC();

co=sql.openConnection("jdbc:oracle:thin:abouhaf/azizhoussem@r2d2.etud.iut-orsay.u-psud.fr:1521:etudom");

}

public void infoClient(String numPizza){

String requete="SELECT nom,prenom,ville,adresse,codePostal,tel ";

requete+="FROM client WHERE numClient="+numPizza;

resultat = sql.exec1Requete(requete, co, 0);

//sql.closeConnection(co);

}

public ResultSet getRes(){

return resultat;

}

}

ModelePizzeria.java

package pizza.modeles;

import java.sql.Connection;

import java.sql.ResultSet;

import java.sql.SQLException;

import p.OutilSJDBC;

public class ModelePizzeria implements InterfaceModelePizzeria {

private OutilSJDBC sql =null;

private ResultSet resultat=null;

private Connection co;

private Object tab\_pizz[][]=null;

private String requete;

public ModelePizzeria(){

sql=new OutilSJDBC();

co=sql.openConnection("jdbc:oracle:thin:abouhaf/azizhoussem@r2d2.etud.iut-orsay.u-psud.fr:1521:etudom");

requete="SELECT id\_commande, nomPizza, quantite, heureLivraison from commande order by date\_commande";

resultat = sql.exec1Requete(requete, co, 0);

}

public ResultSet getRes(){

//sql.closeConnection(co);

return resultat;

}

public void pizzaPrete(String numc){

String requete="update commande set prete=1 where id\_commande="+numc+"";

resultat = sql.exec1Requete(requete, co, 0);

}

public Object miseAJour(){

tab\_pizz = new Object[40][2];

int i=0,j;

try{

while(resultat.next()){

for(j=0;j<=7;j++){

tab\_pizz[i][j]=resultat.getString(j);

}

i=i+1;

}

}

catch(SQLException e){

System.out.println("Pb lors du parcours");

}

//sql.closeConnection(co);

return tab\_pizz;

}

public static void main(String[] args) {

// TODO Auto-generated method stub

//ModelePizzeria m =new ModelePizzeria();

}

}

**IV) Conclusion:**

Ce projet Java/Bd nous a permis de nous familiariser avec les logiciels Oracle et éclipse ainsi fusionné ces deux logiciels.

De plus, le projet nous a permis d’accroître nos connaissances sur le développement JAVA/Oracle qui est très utile pour construire les sites Internet d’aujourd’hui car ils y apportent une exploitation de la base de données via un langage de programmation. En effet, nous avons pu intégrer des interfaces gérant les manipulations des salariés de l’entreprise qui pourront travailler efficacement pour la livraison des pizzas via le site WEB.