

Table des matières

INTRODUCTION	2
PARTIE 1 : PRESENTATION DU PROJET	2
PARTIE 2 : CONCEPTION DE LA BASE DE DONNEES	3
Modèle conceptuel de données	3
Modèle Relationnel de données.....	4
#	5
PARTIE 3 : IMPLEMENTATION ET ALIMENTATION DE LA BASE DE DONNEES	6
Création base de données.....	6
Quelques requêtes de création des tables et de remplissage :.....	6
Quelques contraintes	9
PARTIE 4 : QUELQUES REQUETES, IMPLEMENTATION ET TESTS.....	9
Conclusion	15
Références Bibliographique	15

INTRODUCTION

De nos jours, avec l'apparition de l'informatique moderne, la croissance économique et la survie d'une entreprise sont en étroite relation avec la façon dont les ressources humaines et matérielles sont gérées. Ainsi, aménager une base de données répondant aux besoins réels de l'entreprise permet une croissance et une meilleure concurrence de cette entreprise face aux autres exerçant dans le même domaine. Notre travail est de concevoir une base de données permettant d'assurer la gestion des ventes dans un restaurant. Cette tâche s'articule autour de plusieurs points à savoir :

- Description du thème (vente dans un restaurant), intérêts pour la société, bénéficiaires, et problèmes socio-économiques résolus ;
- Expression des besoins clients que la base de données doit résoudre ;
- Conception de la base de données et son implémentation
- Insertion des données dans la base de données, Formulation et implémentation des requêtes sur ces données.

PARTIE 1 : PRESENTATION DU PROJET

Notre thème choisi est la gestion des ventes dans un restaurant ;

Un restaurant est un établissement commercial où l'on sert des repas contre paiement.

Dans un restaurant, les repas sont cuisinés par une équipe de cuisiniers et servis par une équipe de serveurs et une ou plusieurs femmes de ménages. Ainsi la gestion d'un restaurant, qu'il soit moderne, traditionnel ou un fast Food inclut un certain nombre d'éléments : le gérant, les employés, les clients, les stocks de produits, les salaires... etc.

La bonne gestion d'un restaurant à travers la base de données permet d'éviter les situations comme : la rupture des stocks, la mauvaise gestion du personnel, la non satisfaction des clients...

De plus cette gestion permet une bonne paye des employés en fonction des heures de travail, d'absences et des heures supplémentaires, la garantie de retour des clients suite à leur satisfaction ;

Le profit de cette gestion revient d'abord aux propriétaires des restaurants à travers l'enjeu économique offert et ensuite aux potentiels clients par le traitement rapide des factures et calculs.

PARTIE 2 : CONCEPTION DE LA BASE DE DONNEES

-La création de la base de données de gestion d'un restaurant remplir les besoins clients suivants :

- *Gestions des employés par poste, heures de travail, absences...

- *Consultation et gestion des stocks de produits utilisés pour les cuissons et services.

- *Etude des statistiques de ventes.

- *Consultation du profil des employés

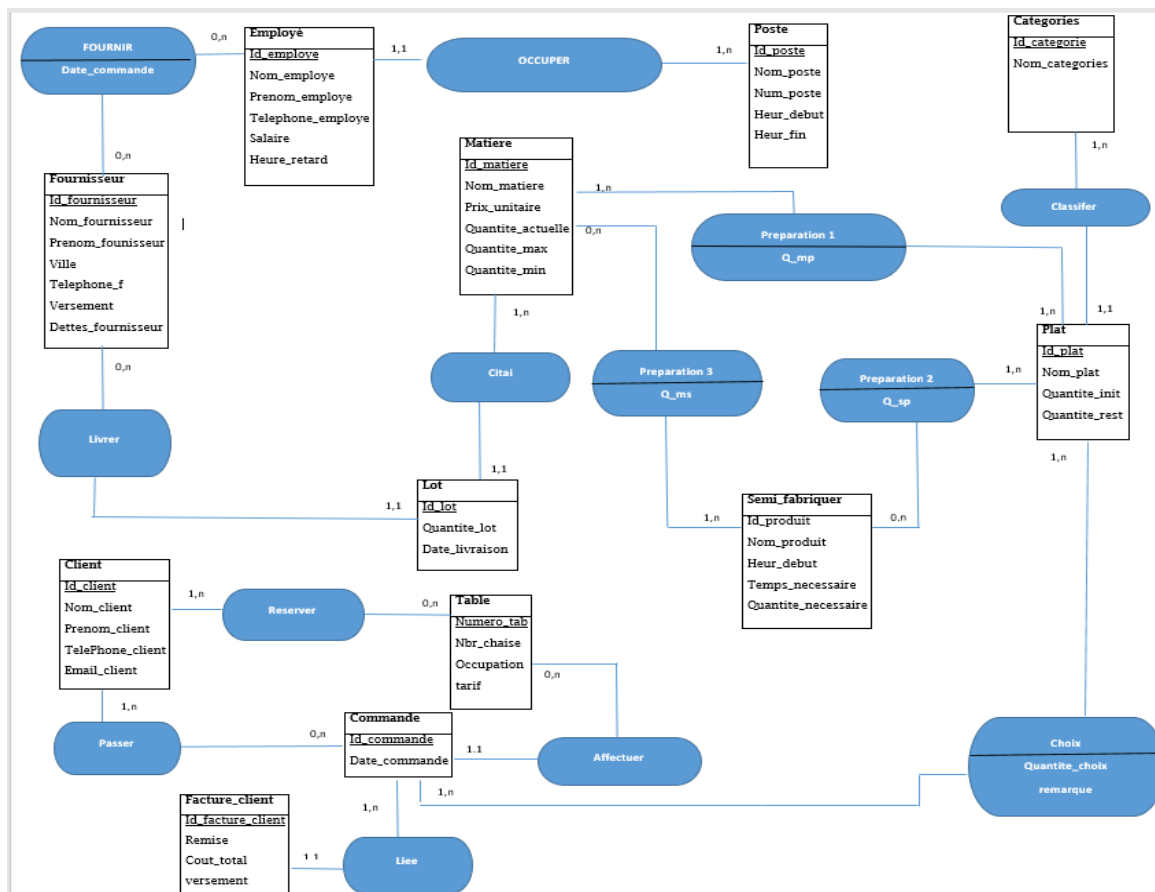
-Conception de la base de données

- * Modèle conceptuel de données : il est basé sur la méthode entité association.

Pour la gestion de notre restaurant, nous les différentes entités suivantes : les employés, les postes occupés par ces employés, les clients, factures clients, commande client, plat cuisinés, la catégorie de chaque plat, matières constituant les plats, lots de matières, fournisseur, commande fournisseur de matière, les choix, opération de cuisson ;

Modèle conceptuel de données

L'association de ces différentes entités nous donne le schéma suivant :



Modèle Relationnel de données

La transcription d'un MCD en modèle relationnel s'effectue selon quelques règles simples qui consistent d'abord à transformer toute entité en table, avec l'attribut identifiant comme clé primaire, puis à observer les valeurs prises par les cardinalités maximums de chaque association pour représenter celle-ci soit **ex** : card. Max 1-n ou 0-n) par l'ajout d'une clé étrangère dans une table existante, soit (**ex** : card. Max n-n) par la création d'une nouvelle table dont la clé primaire est obtenue par concaténation de clés étrangères correspondant aux entités liées.

Partant du modèle entité-association ci-dessus, nous obtenons le modèle relationnel suivant :

- Commande (**Id_Commande**, #num_table, Date_Commande, letat_commande, #Id_Client)
- Categorie (Id_Categ, nom_categ)
- Client(**Id_Client**, Nom_Client, Prenom_Client, Telephone, mail)
- Commandes_Fournisseurs (**Id_Cf**, # Id_Fournisseur, #Id_Matiere, Date_Cf, Date_LivrQuantite_Commande)
- Employe (**Id_Emp**, # Id_Poste, Nom_Emp, Prenom_Emp, Prenom_Emp, Telephone_Emp, Salaire, Heur_Retard)
- Facture Client (**Id_Facture2**, #Id_Commande, remise, Cout_Total_F, versement)
- Fournisseurs (**Id_Fournisseur**, Nom_F, Prenom_F, Adrrese, Email, Telephone, Versement_For, DettesFournisseur)
- Le choix (**id_choix2**, #Id_Commande, # id_plat, quantite_choix, remarque)
- Lot (**Id_lot**, #Id_Matiere, Quantite_Lot Date, _Livraison_lot)
- Matière (**Id_Matiere**, Nom_Matiere, Prix_Unitaire, Quantite_Actuelle, Quantite_Max, Quantite_Min, Date_Fin_Consomation, Nom_Magasin, Dubit_stockDate_Fabrique)
- Plat (**id_plat**, #id_categ, NomPlat, Image, PrixPlat, Qu_init, Qu_Res)
- Poste (**Id_Poste**, Nom_Poste, Num_Emp, Heure_Début, Heur_Fin)
- Sm (**id_Operation**, Nom_Produit, HEUR_Debut_O, T_Nécessaire, Quantité_Nécessaire)
- Tbl (**Numéro_table**, Num_Chaise, Occupation, Tarif_SurTable)

Les nouvelles tables sur Notre modélisation sont :

- pr1 (#id_plat, #id_Operation, q_sp)
- PR2 (#id_plat, #Id_Matiere, #q_mp)
- PR3 (#id_Operation, # Id_Matiere, Q_ms)
- Reservation (#Numéro_table, #Id_Client, jour_Reservation, heur_debut_Reservation, heur_fin_Reservation)

Les symboles utilisés sont:

ID	Les clés primaires
#	Les clés étrangères

PARTIE 3 : IMPLEMENTATION ET ALIMENTATION DE LA BASE DE DONNEES

Création base de données

```
1 CREATE DATABASE IF NOT EXISTS bd_restaurant;
```

Quelques requêtes de création des tables et de remplissage :

Tab catégorie

```
CREATE TABLE IF NOT EXISTS `categorie` (  
  `id_categorie` int NOT NULL AUTO_INCREMENT,  
  `nom_categorie` varchar(30) CHARACTER SET utf8 COLLATE utf8_general_ci NOT NULL,  
  PRIMARY KEY (`id_categorie`)  
) ENGINE=InnoDB AUTO_INCREMENT=7 DEFAULT CHARSET=utf8;
```

```
INSERT INTO `categorie` (`id_categorie`, `nom_categorie`) VALUES  
(1, 'entrée chaude'),  
(2, 'entrée froide'),  
(3, 'soupes'),  
(4, 'a base de riz'),  
(5, 'tarte'),  
(6, 'a base de feuille');
```

Tab client

```
CREATE TABLE IF NOT EXISTS `client` (  
  `id_client` int NOT NULL AUTO_INCREMENT,  
  `nom_client` varchar(30) CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci NOT NULL,  
  `prenoms_client` varchar(20) CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci NOT NULL,  
  `telephone_client` int NOT NULL,  
  `email_client` varchar(80) CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci NOT NULL,  
  PRIMARY KEY (`id_client`)  
) ENGINE=InnoDB AUTO_INCREMENT=8 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

```
INSERT INTO `client` (`id_client`, `nom_client`, `prenoms_client`, `telephone_client`, `email_client`) VALUES
(2, 'tema', 'steve', 652154874, 'steve225@gmail.com'),
(3, 'kameni', 'jordan', 685142154, ''),
(4, 'kiary', 'jule', 652124852, 'fiq5@gmail.com'),
(5, 'ndjenou', 'carelle', 699564842, 'carndj@gmail.com'),
(6, 'leuthe', 'louis', 654214750, ''),
(7, 'dedieue', 'moise', 699564212, 'moise24h@gmail.com');
```

Tab matière

```
DROP TABLE IF EXISTS `matiere`;
CREATE TABLE IF NOT EXISTS `matiere` (
  `id_matiere` int NOT NULL AUTO_INCREMENT,
  `nom_matiere` varchar(20) NOT NULL,
  `prix_unitaire` double NOT NULL,
  `quantite_actuelle` int NOT NULL,
  `quantite_max` int NOT NULL,
  `quantite_min` int NOT NULL,
  `date_fin_consommation` date NOT NULL,
  `nom_magasin` varchar(20) NOT NULL,
  `debut_stockdate_fabrication` date NOT NULL,
  PRIMARY KEY (`id_matiere`)
) ENGINE=InnoDB AUTO_INCREMENT=8 DEFAULT CHARSET=utf8;
```

```
INSERT INTO `matiere` (`id_matiere`, `nom_matiere`, `prix_unitaire`, `quantite_actuelle`, `quantite_max`,
`quantite_min`, `date_fin_consommation`, `nom_magasin`, `debut_stockdate_fabrication`) VALUES
(1, 'sel', 12, 14, 200, 10, '0000-00-00', 'm1', '2021-03-03'),
(2, 'piment', 50, 20, 25, 2, '2021-03-17', 'm1', '2021-03-01'),
(3, 'tomate', 50, 200, 500, 10, '2021-03-10', 'm1', '2021-03-07'),
(4, 'riz', 500, 2000, 50000, 500, '2021-04-09', 'm2', '2019-03-21'),
(5, 'carotte', 100, 20, 25, 0, '2021-03-05', 'm1', '2021-03-01'),
(6, 'salade', 200, 10, 30, 2, '2021-03-04', 'm1', '2021-03-01'),
(7, 'condiment_verte', 50, 100, 100, 20, '2021-03-02', 'm1', '2021-02-25');
```

Tab plat

```
CREATE TABLE IF NOT EXISTS `plat` (
  `id_plat` int NOT NULL AUTO_INCREMENT,
  `id_categorie` int NOT NULL,
  `nom_plat` varchar(20) COLLATE utf8mb4_general_ci NOT NULL,
  `image` blob,
  `qu_init` int NOT NULL,
  `qu_res` int NOT NULL,
  `prix_plat` double NOT NULL,
  PRIMARY KEY (`id_plat`),
  KEY `id_categ` (`id_categorie`)
) ENGINE=InnoDB AUTO_INCREMENT=8 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

```
INSERT INTO `plat` (`id_plat`, `id_categorie`, `nom_plat`, `image`, `qu_init`, `qu_res`, `prix_plat`) VALUES
(4, 2, 'katala', '', 21, 14, 2500),
(5, 4, 'tato riz', NULL, 21, 0, 1500),
(6, 6, 'okok', NULL, 32, 2, 2000),
(7, 6, 'ndole', NULL, 100, 12, 1000);
```

Tab commande fournisseur

```
CREATE TABLE IF NOT EXISTS `commande_fournisseurs` (
  `id_cf` int NOT NULL AUTO_INCREMENT,
  `id_fournisseur` int NOT NULL,
  `id_matiere` int NOT NULL,
  `date_cf` timestamp NOT NULL ON UPDATE CURRENT_TIMESTAMP,
  `date_livrquantite_command` timestamp NOT NULL ON UPDATE CURRENT_TIMESTAMP,
  PRIMARY KEY (`id_cf`),
  KEY `id_fournisseur` (`id_fournisseur`),
  KEY `id_matiere` (`id_matiere`)
) ENGINE=InnoDB AUTO_INCREMENT=3 DEFAULT CHARSET=utf8;
```

```
INSERT INTO `commande_fournisseurs` (`id_cf`, `id_fournisseur`, `id_matiere`, `date_cf`, `date_livrquantite_command`) VALUES
(1, 1, 6, '2021-02-28 23:20:40', '2021-03-04 23:20:48'),
(2, 2, 2, '2021-03-15 23:22:51', '2021-03-25 23:22:51');
```

Tab Table


```
CREATE TABLE IF NOT EXISTS `tab` (
  `numero-tab` int NOT NULL AUTO_INCREMENT,
  `num_chaise` int NOT NULL,
  `occupation` varchar(20) CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci DEFAULT NULL,
  `tarif_sur_tab` double NOT NULL,
  PRIMARY KEY (`numero-tab`)
) ENGINE=InnoDB AUTO_INCREMENT=6 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

```
INSERT INTO `tab` (`numero-tab`, `num_chaise`, `occupation`, `tarif_sur_tab`) VALUES
(1, 6, NULL, 2500),
(2, 10, NULL, 2500),
(3, 2, NULL, 100),
(4, 5, NULL, 1500),
(5, 6, NULL, 2000);
```

Quelques contraintes

```
--
-- Contraintes pour la table `choix`
--
ALTER TABLE `choix`
  ADD CONSTRAINT `choix_ibfk_1` FOREIGN KEY (`id_plat`) REFERENCES `plat` (`id_plat`) ON DELETE RESTRICT ON UPDATE RESTRICT,
  ADD CONSTRAINT `choix_ibfk_2` FOREIGN KEY (`id_commande`) REFERENCES `commande` (`id_commande`) ON DELETE RESTRICT ON UPDATE RESTRICT;
--
-- Contraintes pour la table `commande`
--
ALTER TABLE `commande`
  ADD CONSTRAINT `commande_ibfk_1` FOREIGN KEY (`numero_table`) REFERENCES `tab` (`numero-tab`) ON DELETE RESTRICT ON UPDATE RESTRICT,
  ADD CONSTRAINT `commande_ibfk_2` FOREIGN KEY (`id_client`) REFERENCES `client` (`id_client`) ON DELETE RESTRICT ON UPDATE RESTRICT;
--
```

PARTIE 4 : QUELQUES REQUETES, IMPLEMENTATION ET TESTS

1) Nom et le salaire des cuisiniers

```
SELECT nom_employe, salaire_employe
FROM employe, poste
WHERE id_employe = poste.id_poste
```

✓ Affichage des lignes 0 - 3 (total de 4, traitement en 0,0013 seconde(s).)

```
SELECT nom_employe,salaire_employe FROM employe,poste WHERE id_employe = poste.id_poste
```

☐ Tout afficher | Nombre de lignes : 25 ▼ Filtre les lignes:

+ Options

nom_employe	salaire_employe
onana	50000
kameni	75000
djimboum	37000
zomo	50000

2) afficher la catégorie à laquelle appartient l'okok

```
SELECT nom_categorie
FROM categorie, plat
WHERE id_categorie = plat.id_plat and plat.nom_plat = "okok";
```

✓ Affichage des lignes 0 - 0 (total de 1, traitement en 0,0009 seconde(s).)

```
SELECT nom_categorie FROM categorie,plat WHERE id_categorie = plat.id_plat and plat.nom_plat = "okok"
```

☐ Profilage

☐ Tout afficher | Nombre de lignes : 25 ▼ Filtre les lignes:

+ Options

nom_categorie
a base de feuille

3) nombre de client

```
select(count(id_client))
      from (client);
```

La requête SQL a été exécutée avec succès.

```
SELECT COUNT(id_client) FROM `client`
```

+ Options

COUNT(id_client)

6

4) liste des fournisseurs

```
select *
      From (fournisseurs) ;
```

✓ Affichage des lignes 0 - 3 (total de 4, traitement en 0,0014 seconde(s).)

```
SELECT * FROM `fournisseur`
```

☐ Profilage [Éditer en ligne] [Éditer] [Expliquer SQL] [Créer le code source PHP]

☐ Tout afficher | Nombre de lignes : 25 ▼ | Filtrer les lignes: Chercher dans cette table | Trier par clé : Aucun(e) ▼

+ Options

	id_fournisseur	nom_f	prenom_f	email_f	ville_f	telephone_f	versement_f	dettes_fournisseur
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	1	ngounou	jean	jend@gmail.com	manjo	685214578	140000	0
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	2	hugue	loic	cuefq@ticto.com	douala	684541214	25000	15000
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	3	didan	luck	die@gmail.com	yaounde	652145121	140000	0
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	4	tiomene	serge	sergiok52@mrt.cm	abidjan	675421240	20000	0

5) Nom des matières livrer le 4 mars 2021

```
select (nom_matière)
      from (matière like m, lot like l)
      Where (l.id_matière = m.id_matière) and (date de livraison_lot ='2021-03-04') ;
```

✓ Affichage des lignes 0 - 0 (total de 1, traitement en 0,0013 seconde(s).)

```
SELECT nom_matiere FROM matiere,lot WHERE id_lot = matiere.id_matiere and date_livraison_lot = '2021-03-04'
```

☐ Profilage [Éditer en ligne]

☐ Tout afficher | Nombre de lignes : 25 ▼ Filtrer les lignes: Chercher dans cette table

+ Options

nom_matiere
tomate

6- calcule la somme des dettes de tous les fournisseurs

SELECT SUM (dettes_fournisseur) FROM fournisseur

✓ Affichage des lignes 0 - 0 (total de 1, traitement en 0,0015 seconde(s).)

```
SELECT SUM(dettes_fournisseur) FROM fournisseur
```

☐ Tout afficher | Nombre de lignes : 25 ▼ Filtrer les lignes

+ Options

SUM(dettes_fournisseur)
15000

7) afficher un client et le montant de versement de sa commande

Select (non_client, montant_versement)

from (client, factureclient)

where (facture.id_commande = commande.id_commande) and
(commande.id_client = client.id_client);

✓ Affichage des lignes 0 - 1 (total de 2, traitement en 0,0017 seconde(s).)

```
SELECT nom_client, versement FROM facture_client, client, commande WHERE facture_client.id_commande = commande.id_commande and commande.id_commande = client.id_client
```

☐ Profilage [Éditer en ligne] [Éditer] [Expliquer SQL] [Créer le code s

☐ Tout afficher | Nombre de lignes : 25 ▼ Filtrer les lignes: Chercher dans cette table

+ Options

nom_client	versement
tema	150
kameni	15000

8) nom et numéros de téléphone du fournisseur de tomate

SELECT telephone_f, nom_f

FROM fournisseur, matiere, lot

WHERE fournisseur.id_fournisseur = lot.id_fournisseur AND
lot.id_matiere = matiere.id_matiere

GROUP by telephone_f

✓ Affichage des lignes 0 - 1 (total de 2, traitement en 0,0012 seconde(s).)

```
SELECT telephone_f, nom_f FROM fournisseur, matiere, lot WHERE fournisseur.id_fournisseur = lot.id_fournisseur AND lot.id_matiere = matiere.id_matiere GROUP by telephone_f
```

☐ Profilage [Éditer en ligne] [Éditer] [Expliquer SQL] [Créer le code sour

☐ Tout afficher | Nombre de lignes : 25 ▼ Filtrer les lignes: Chercher dans cette table

+ Options

telephone_f	nom_f
675421240	tiomene
685214578	ngounou

9) nom et téléphone des clients qui ont occupé la table numéro 2

SELECT nom_client, telephone_client FROM client, tab, commande

WHERE client.id_client = commande.id_client AND commande.numero_table = 2

GROUP by nom_client, telephone_client

✓ Affichage des lignes 0 - 0 (total de 1, traitement en 0,0021 seconde(s).)

```
SELECT nom_client, telephone_client FROM client, tab, commande WHERE client.id_client = commande.id_client AND commande.numero_table = 2 GROUP by nom_client, telephone_client
```

☐ Profilage [Éditer en ligne] [Éditer] [Expliquer SQL] [Créer le code source PH

☐ Tout afficher | Nombre de lignes : 25 ▼ Filtrer les lignes: Chercher dans cette table

+ Options

nom_client	telephone_client
kiary	652124852

10- listes des employés

✓ Affichage des lignes 0 - 6 (total de 7, traitement en 0,0014 seconde(s).)

```
SELECT * FROM `employe`
```

☐ Profilage [Éditer en ligne] [Éditer] [Expliquer SQL] [Créer le code source PHP]

☐ Tout afficher | Nombre de lignes : 25 | Filtrer les lignes: Chercher dans cette table | Trier par clé : Aucun(e)

+ Options

		id_employe	id_poste	nom_employe	prenom_employe	telephone_employe	salaire_employe	heur_retard
<input type="checkbox"/>	Éditer Copier Supprimer	1	2	onana	andré	652145213	50000	0
<input type="checkbox"/>	Éditer Copier Supprimer	2	2	kameni	brenda	675421215	75000	0
<input type="checkbox"/>	Éditer Copier Supprimer	3	4	djimboum	stephanie	672124563	37000	0
<input type="checkbox"/>	Éditer Copier Supprimer	4	1	zomo	federic	682154719	50000	0
<input type="checkbox"/>	Éditer Copier Supprimer	5	2	gimno	farelle	695462132	85000	4
<input type="checkbox"/>	Éditer Copier Supprimer	6	3	taniko	dan	695621430	105000	12
<input type="checkbox"/>	Éditer Copier Supprimer	7	3	biamkop	valery	685952130	200000	1

11- supprimer les ménagères

DELETE employe From employe,poste

WHERE employe.id_poste = poste.id_poste AND poste.nom_poste= "menagere"

✓ 1 ligne affectée. (traitement en 0,0830 seconde(s).)

```
DELETE employe FROM employe,poste WHERE employe.id_poste = poste.id_poste AND poste.nom_poste = "menagere"
```

Nouvelle liste:

✓ Affichage des lignes 0 - 5 (total de 6, traitement en 0,0019 seconde(s).)

```
SELECT * FROM `employe`
```

☐ Profilage [Éditer en ligne] [Éditer] [Expliquer SQL] [Créer le code source PHP]

☐ Tout afficher | Nombre de lignes : 25 | Filtrer les lignes: Chercher dans cette table | Trier par clé : Aucun(e)

+ Options

		id_employe	id_poste	nom_employe	prenom_employe	telephone_employe	salaire_employe	heur_retard
<input type="checkbox"/>	Éditer Copier Supprimer	1	2	onana	andré	652145213	50000	0
<input type="checkbox"/>	Éditer Copier Supprimer	2	2	kameni	brenda	675421215	75000	0
<input type="checkbox"/>	Éditer Copier Supprimer	4	1	zomo	federic	682154719	50000	0
<input type="checkbox"/>	Éditer Copier Supprimer	5	2	gimno	farelle	695462132	85000	4
<input type="checkbox"/>	Éditer Copier Supprimer	6	3	taniko	dan	695621430	105000	12
<input type="checkbox"/>	Éditer Copier Supprimer	7	3	biamkop	valery	685952130	200000	1

Conclusion

En définitive il était question pour nous dans notre travail de déterminer les différents services pour résoudre le problème de gestion d'un restaurant, de concevoir une base de données répondant à ce besoin, formuler et implémenter des requêtes qui réalisent les différents services attendus. Notre travail se partitionne ainsi en 04 de l'analyse à l'implémentation ; Dans la première partie apparaît la présentation du projet où nous avons insisté sur les acteurs dans le service de la restauration et leurs rôles nous avons également regardé en quelques point l'importance de la base de données dans ce milieu. Dans la deuxième partie nous nous sommes penchés sur la conception de la base de données où nous sommes appuyés d'abord sur le model conceptuel puis sur le model relationnel de notre base. La troisième partie qui consistait à l'implémentation s'est faite grâce au SGBD

PhpmyAdmin dont l'usage de plusieurs requêtes afin d'arriver à notre base de données. Et notre dernière partie qui implique beaucoup plus le test où nous avons effectué un certain nombre de test dans notre base donnée et en avons vu que cette dernière est opérationnelle.

Références Bibliographique

[1] - 2017. Définition : restaurant. <http://www.restaurantsquifontamanger.fr/restaurant.htm>. [En ligne] Mai 2017.

[2] - L. Baptiste, Merise Guide pratique : Modélisation des données et des traitements, manipulations avec le langage SQL, ENI éditions, 2009.