



MEX&GO® : MISE EN PLACE D'UNE BASE DE DONNÉES CLIENTS-TRAVAILLEURS- PRODUITS

Domaine d'application et réalisation

Résumé

Ce document a pour but d'informer le lecteur des différentes étapes ayant mené à la mise en place d'une base de données reprenant les travailleurs, les clients ainsi que les produits du restaurant Mex&Go®.

UNamur INFO B212 2021-22

Groupe 10:

Anoushka LUNA-BARRAGAN

Nicolas BALTOFSKI

Dzenetan ALITI

Georges HOEBRECHTS

Table des matières

Introduction	2
Retranscription du discours du client :	3
Contexte :	3
Fonctionnement actuel :	4
Ce que veut le client :	5
Nos contraintes :	5
Autres :	7
Schéma conceptuel relatif au domaine	8
Schéma logique équivalent	9
Problèmes liés à la conversion en DDL	10
Ajout de triggers	10
EQU	10
Ajout d'une boucle de vérification.....	11
Déploiement en java	13
Implémentation des DTO	13
Mise en place de la console	13
Conclusion et Devis	14
Bibliographie	15

Introduction

Pour ce travail, nous avons choisi de travailler avec la société Mex&Go SRL qui possède 2 restaurants. Elle désire implémenter une base de données reprenant les informations de ses travailleurs ainsi que celles de ses clients dits « ambassadeurs » bénéficiant de promotions spécifiques à leur niveau de fidélité. Ces derniers s'inscriront via un formulaire en ligne et pourront anonymiser leurs données personnelles à tout moment. La société désire aussi connaître toutes les infos relatives aux différents passages de ses clients (produits consommés, prix, promotions attribuées...).

Retranscription du discours du client :

Interview pour : Mex&Go SRL

15 rue Basse Marcelle 5000 NAMUR

BE0633965571

Contexte :

Nous avons actuellement un système de fidélité où le client reçoit une carte nominative ou l'on met des cachets à chaque passage du client. Tous les 4 passages, le client décide de prendre une réduction et recommencer à 0 ou de continuer vers jusqu'au bout de la carte afin de devenir ambassadeur de la marque.

Une fois ambassadeur, le client bénéficie d'une réduction spécifique à chaque visite sous présentation d'une carte en PVC avec code-barre comme détaillée ci-dessous.

Carte à cachets (pliée)

extérieur :



intérieur :



Carte en PVC :

BACKGROUND: MATT BLACK

SELECTIVE VARNISH RED



SELECTIVE VARNISH BLACK

EMBOSSED WHITE

BACKGROUND: MATT BLACK

MATT WHITE



SELECTIVE VARNISH: I

BARCODE (LINKED WITH NUMBER FRONT)

Fonctionnement actuel :

Une fois le client devenu ambassadeur, nous lui remettons un formulaire comme celui-ci :



BIENVENUE !

Merci de bien vouloir remplir les champs suivants
EN LETTRES CAPITALES SVP

MR/MME/MELLE

NOM

RUE/AV

N°

CODE POSTAL

PAYS

DATE DE NAISSANCE (JOUR/MOIS/ANNEE)

EMAIL

N° DE GSM/TEL

PRENOM

BTE

LOCALITE

RESERVE AU PERSONNEL

CARTE N°

VOUS DESIREZ RECEVOIR DES OFFRES PROMOTIONNELLES

Biffez la mention inutile

PAR COURRIER	OUI	NON
PAR E-MAIL	OUI	NON
PAR SMS	OUI	NON

Un travailleur encode ensuite ces données dans notre base de données.

Lorsque l'ambassadeur passe en caisse les fois suivantes, nous **scannons le code barre de la carte, ce qui affiche le nom et prénom de la personne** et nous permet de contrôler l'identité si besoin. Le travailleur sélectionne ensuite la promotion à appliquer. Les promotions varient chaque mois (ex. : boisson offerte, guacamole à 50%,...) avec la possibilité de ne pas prendre la réduction pour le client et de bénéficier de 10% sur le prix de son menu.

Aucune statistique n'est enregistrée et les mails/sms ne sont pas pris en compte car aucune promo n'est faite via ces canaux.

La base de données actuelle est exportable sous fichier Excel mais est intégrée à HORECATOUCH, logiciel privé non accessible.

Ce que veut le client :

Le client veut modifier la deuxième partie de son programme de fidélité (carte PVC) dans leurs 2 restaurants et gérer au mieux son personnel.

Une personne s'inscrit sur la base de données. Elle peut être un client ou un travailleur (et pas les deux).

Un client s'inscrit dans la base de données. Lorsqu'il passe au restaurant, il effectue des achats de produits. Sur ces produits peuvent être appliquées des promotions qui varient en fonction du statut du client.

Un travailleur ne bénéficie pas du statut ambassadeur car il mange déjà gratuitement au restaurant.

Nos contraintes :

Personnel et restaurant : Toute personne inscrite en tant que membre du personnel possède les mêmes attributs qu'un ambassadeur (obligatoires) mais ne peut être ambassadeur en même temps. A cette personne est assigné un badge unique pour ouvrir la porte d'entrée ou une clef pour désactiver la sécurité le matin. Chaque membre du personnel travaille dans un restaurant dont les attributs sont les éléments de l'adresse de celui-ci (rue, numéro, ...).

Chaque membre du personnel de chaque restaurant est identifié par son NISS et signe un contrat lors de son engagement dont la date de début et de fin sont mentionnées. Cela impose qu'un contrat ne peut se chevaucher avec un autre.

Ambassadeur : Un client ayant accumulé suffisamment de cachets sur sa carte de fidélité sans les consommer peut prétendre au titre d'Ambassadeur. En échange d'une carte pleine, il peut se voir créer un compte dans la base de données et se voit attribué un identifiant (celui de la carte qu'il reçoit, seul élément nécessaire pour s'identifier à la caisse) ainsi qu'un palier (voir ci-dessous) de niveau Bronze. Un ambassadeur n'est pas obligé de rentrer ses informations personnelles mais ne peut demander des offres promotionnelles via un canal de communication sans introduire préalablement les informations nécessaires (N° GSM, Adresse Mail, Adresse). Un ambassadeur peut modifier ses informations ainsi que supprimer toutes ses données. Dans ce dernier cas, seul son id et son palier (les préférences marketing étant mises à faux) seront conservés afin de pouvoir faire le lien avec ses achats à des fins comptables. En réalité, une suppression n'est qu'une anonymisation du compte car ce dernier reste valide, puisqu'aucune information personnelle n'est nécessaire pour avoir un compte opérationnel. Un ambassadeur peut aussi passer commande d'un ou plusieurs produits dans le restaurant de son choix. A noter que si un nom est spécifié, le prénom est requis, et inversement.

Produits et menus : Les produits sont l'ensemble des consommables disponibles dans le restaurant. Ils ont un identifiant (représenté par un code barre), un prix et un nom. Certains d'entre eux peuvent être composés d'autres produits, ce sont des menus. Il n'est pas possible d'introduire un menu se composant lui-même de façon directe ou non. Donc un menu peut être composé entre autres de menus (mais pas de lui-même).

Tickets et détails : Les tickets reprennent les informations d'une commande, chaque ticket est identifié de manière unique via un numéro, il référence la date, le restaurant et si le commanditaire est un ambassadeur ou non. De plus, on peut y trouver les différents produits commandés.

Promos et paliers : chaque produit (+ menus) peut avoir une ou plusieurs promotions d'applicables. Une promotion est soit une réduction fixe (ex : -2€), soit un pourcentage sur le prix du produit (ex : -25%). Une promotion ne s'applique que sur certains produits. Certaines promotions spéciales (appelées "Offres") sont liées à des paliers qui sont des rangs atteints ou non par les ambassadeurs. Une offre n'est donc applicable que si le produit est acheté par un ambassadeur qui a atteint ce palier. Des paliers plus difficilement atteignables proposent plus ou de meilleures offres, mais l'ambassadeur ne cumule pas les offres, il n'a accès qu'à celles de son palier. Un produit acheté ne peut avoir qu'une seule promotion d'appliqué dessus, mais des promotions peuvent être appliquées sur différents achats d'un même produit (ex : pour un achat de deux boissons, on peut appliquer une première promotion sur la première boisson, et une seconde sur la deuxième). Certaines promotions ont des conditions spécifiques (ex : promotion étudiant), mais l'évaluation de celle-ci est laissée à l'appréciation du caissier.

Les rôles et accès aux données:

Afin de limiter l'accès à certaines données et d'éviter une mauvaise utilisation de la base de données par les travailleurs, nous avons mis en place un système de rôles comme suit :

Droits accordés aux utilisateurs :

-Ouvrier : Un ouvrier peut créer un compte **ambassadeur**, voir ses informations, les mettre à jour et anonymiser le compte. Il peut créer des tickets avec les produits achetés.

-Gérant : Peut accéder et modifier l'ensemble des données tant que celles-ci restent cohérentes.

-Marketing : Le service marketing peut modifier, ajouter ou supprimer des données liées aux produits, aux offres et aux paliers. Il en va de même pour les promotions à l'exception des modifications.

Comptabilité : Le service comptable à accès aux informations liées aux chiffres d'affaires des différents restaurants. Il pourra aussi avoir les informations suivantes : Dépense moyenne mensuelle des

ambassadeurs, le coût de chaque promotion sur une année et la consommation annuelle des produits. De plus, il pourra modifier, insérer et supprimer des produits ainsi que leur composition.

GRH : Les ressources humaines peuvent ajouter, modifier et supprimer les informations des clefs/badges. Ils peuvent aussi ajouter des contrats ainsi que modifier les informations du personnel. Ils peuvent accéder à l'ensemble des informations précédemment citées.

Autres :

Chiffre d'affaires par mois ou année :

Afin de connaître la santé de l'entreprise Mex&Go®, il faut être capable d'obtenir le chiffre d'affaires par an et par mois, avec la différence de la période précédente. Si des périodes sont manquantes, il n'est pas nécessaire d'indiquer un chiffre d'affaires nul car cette information n'est pas intéressante pour les gérants.

Coût des promotions :

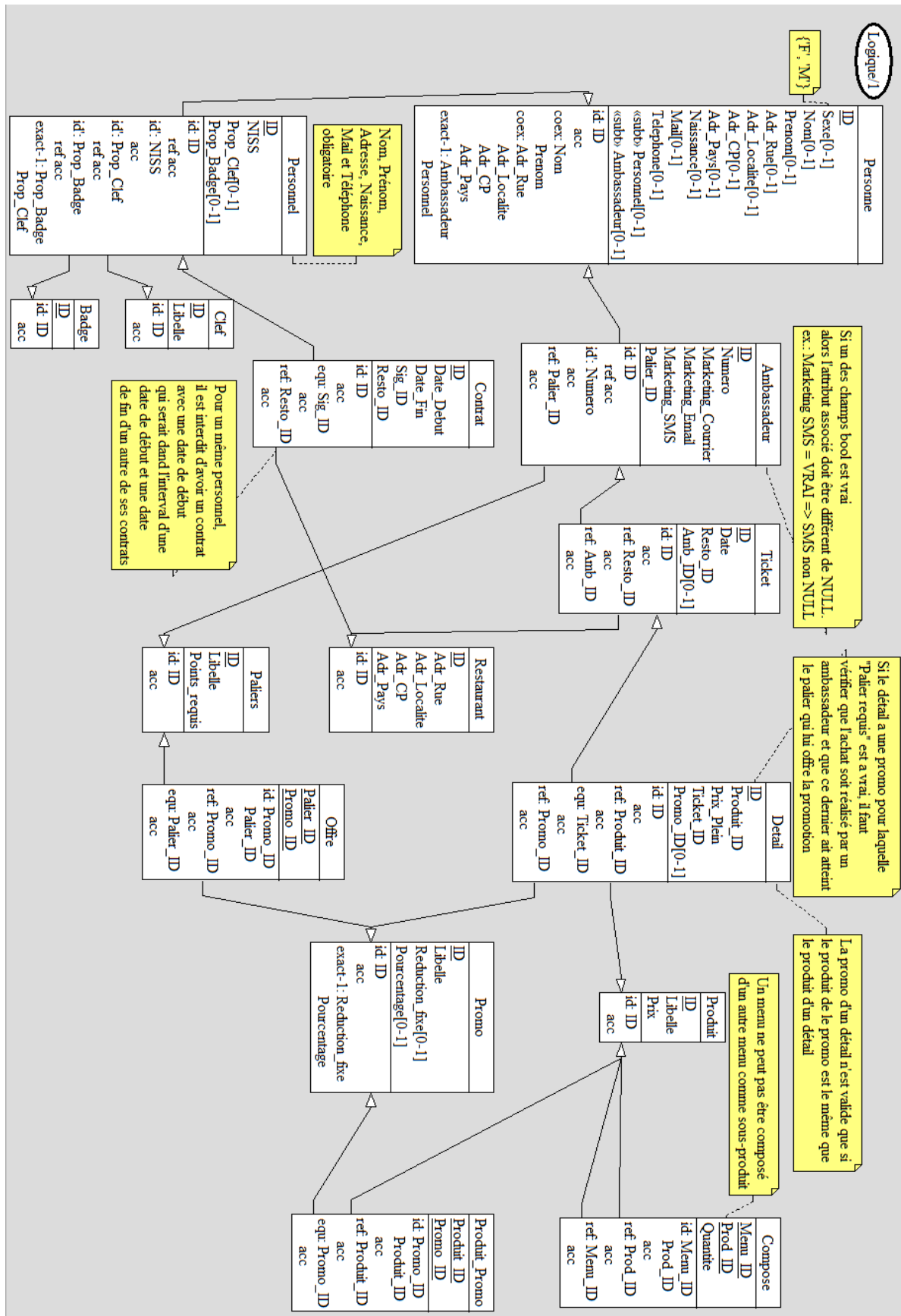
L'entreprise souhaite connaître le "coût" d'une promotion, c'est-à-dire le montant que les utilisateurs ont épargné par promotion.

Dépense par ambassadeur :

Par ambassadeur, montrer le montant moyen dépensé par mois depuis son inscription/depuis 1 mois.



Schéma logique équivalent



Problèmes liés à la conversion en DDL

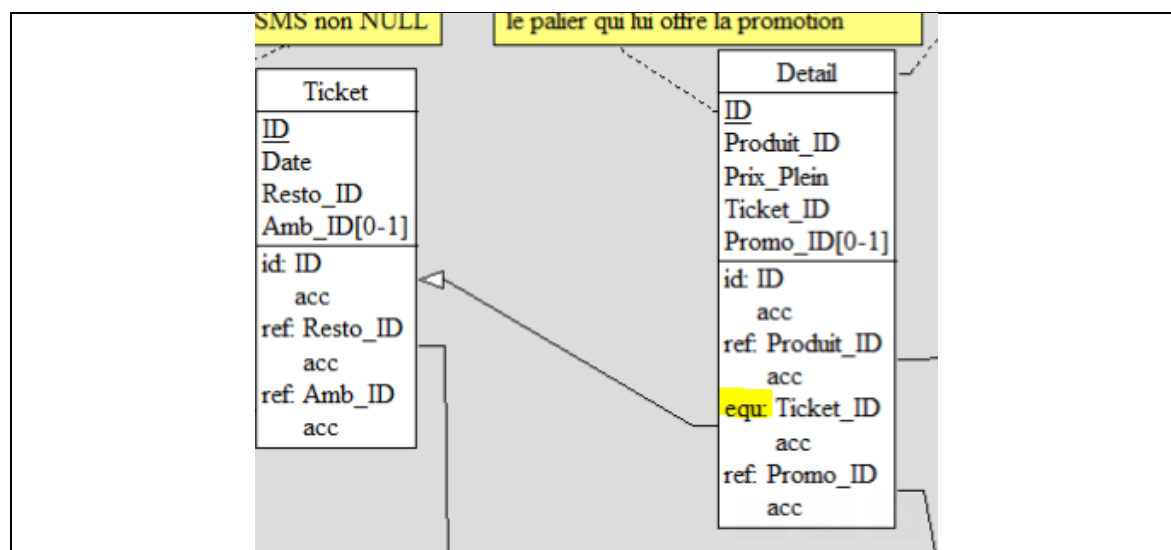
Lors de la conversion automatique de DB Main, certaines erreurs liées à l'utilisation de MySQL® nous ont contraint à faire quelques modifications dans le code DDL. Nous avons aussi cherché à améliorer le modèle automatique avec nos réflexions.

Ajout de triggers

EQU

Le concept de clé étrangère totale n'existe pas nativement dans MySQL. N'ayant à notre disposition que des foreign key classiques ou des checks qui ne permettent pas des sous-requêtes (par exemple des selects dans d'autres tables), nous avons implémenté le côté total par l'utilisation de triggers. Dû à une limitation de MySQL, nous avons créé des triggers pour chaque opération (insert, delete, ...). Nous avons créé des procédures ou fonctions afin de limiter la redondance de code.

Nous avons cependant eu un problème car il n'était plus possible de faire des modifications entre les triggers et les clefs étrangères comme expliqué ci-dessous.



Nous traitons ici de l'insertion d'un Ticket. Le trigger vérifie donc qu'un ticket a forcément un détail. Voici son code (plusieurs triggers ayant besoin de la même contrainte, celle-ci a été mise dans une procédure) :

```
CREATE PROCEDURE proc_equ_detail_ticket(IN Ticket_ID INT)
BEGIN
  IF NOT EXISTS(SELECT * FROM Detail WHERE Detail.Ticket_ID = Ticket_ID) THEN
    SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Un ticket doit obligatoirement avoir au moins un detail';
  END IF;
END $$
```

```
CREATE TRIGGER trigger_ticket_insert
BEFORE INSERT
ON Ticket
FOR EACH ROW
BEGIN
  CALL proc_equ_detail_ticket( Ticket_ID: NEW.ID);
END $$
```

*ancienne version

Puisqu'il existe une variable permettant de désactiver la vérification des clefs étrangères, nous avons choisi d'utiliser le même procédé dans le trigger. Nous ajoutons donc une variable globale, qui peut être mise à vrai ou faux par les procédures, et qui permettra ou non au trigger d'appeler la procédure.

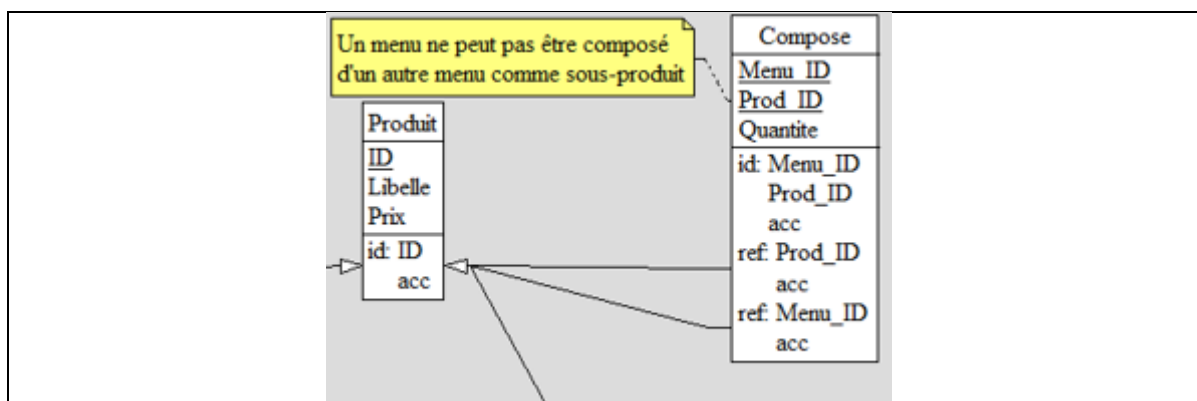
```
CREATE PROCEDURE insert_ticket()
BEGIN
    START TRANSACTION;
    SET @trigger_insert_ticket_checks = FALSE;
    INSERT INTO Ticket(Date, Resto_ID, Amb_ID) VALUES ('2000-01-01', 1, 2);
    SET @trigger_insert_ticket_checks = TRUE;
    INSERT INTO Detail(Prix_Plein, Produit_ID, Ticket_ID) VALUES (2, 1, LAST_INSERT_ID());
    COMMIT;
END;

CREATE TRIGGER trigger_ticket_insert
BEFORE INSERT
ON Ticket
FOR EACH ROW
BEGIN
    IF @trigger_insert_ticket_checks THEN
        CALL proc_equ_detail_ticket( Ticket_ID: NEW.ID);
    END IF;
END $$
```

*version corrigée

Ajout d'une boucle de vérification

Lors de nos réflexions, nous avons voulu vérifier qu'aucun produit ne se compose lui-même et nous avons pensé à ajouter une fonctionnalité non demandée par le client en cas d'évolution de son concept dans le futur.



Si le restaurant décidait de mettre un petit menu qui lui-même, accompagné d'un autre produit, compose un plus grand menu, notre base de données réagirait favorablement. En effet, un trigger vérifie lors de l'introduction d'un menu qu'il n'est pas composé de lui-même. Ainsi, la récursion va créer une table éphémère reprenant l'ensemble des produits et menu nécessaires pour produire le menu introduit et vérifier qu'aucun n'apparaisse en amont de la composition du menu.

```

CREATE FUNCTION fct_is_menu_self_composed(ID_M INT, ID_P INT) RETURNS BOOLEAN
READS SQL DATA
BEGIN
    DECLARE is_comp BOOLEAN;
    #VERIFICATION QUE LE MENU INSERE EXISTE DEJA COMME COMPOSANT D'UN AUTRE MENU
    IF EXISTS(SELECT * FROM Compose WHERE ID_M = Prod_ID)
    THEN
        #VERIFICATION QUE DANS LES MENU COMPOSE DE CE MENU INSERE, IL N'Y AI PAS LE PRODUIT INSERE
        #SINON IL SE COMPOSE LUI MEME INDIRECTEMENT
        IF EXISTS(
            WITH RECURSIVE COMPOSITION(NIVEAU, COMPOSE, COMPOSANT)
            #INITIALISATION AVEC LES MENU COMPOSE DU MENU INSERE COMME PRODUIT
            AS (
                SELECT 1, Menu_ID, Prod_ID
                FROM Compose
                WHERE Prod_ID = ID_M
                UNION ALL
                #RECURENCE CONSTRUISANT LE CHEMIN MONTANT ENTRE LES MENU ET CE QU'ILS COMPOSENT
                SELECT CION.NIVEAU + 1, C.Menu_ID, C.Prod_ID
                FROM COMPOSITION CION,
                Compose C
                WHERE CION.COMPOSE = C.Prod_ID
            )
            #SELECTION DES LIGNES OU LE PRODUIT INSERE EST COMPOSANT DE LUI MEME
            SELECT NIVEAU, COMPOSE, COMPOSANT
            FROM COMPOSITION
            WHERE ID_P = COMPOSE
        )
        #SI LE PRODUIT INSERE EST COMPOSANT DE LUI MEME CETTE VALLEUR EST VRAIE
    THEN
        SET is_comp = TRUE;
    ELSE
        #SINON FAUSSE
        SET is_comp = FALSE;
    END IF;
ELSE
    #SI LE MENU INSERE N'EST COMPOSANT D'AUCUN MENU, ALORS LE PROD NE PEUT PAS ETRE AUTO-COMPOSE
    #ET DONC VALEUR MISE A FAUX
    SET is_comp = FALSE;
END IF;
RETURN is_comp;
END $$

```

Déploiement en java

Implémentation des DTO

- AmbassadeurDTO : classe contenant les attributs des ambassadeurs (ID, genre prénom, nom, rue, ville, code postal, pays, date de naissance, adresse e-mail, numéro de téléphone, number, option marketing par courrier postal, option marketing par e-mail, option marketing par SMS)
- TicketDTO : classe contenant les attributs des tickets de caisse (ID ticket, date, ID restaurant, ambassadeur)
- PersonneDTO : classe contenant les attributs des personnes (ID, genre, prénom, nom, rue, ville, code postal, pays, date de naissance, adresse e-mail, numéro de téléphone)
- InfoMarketingDTO : classe contenant les infos marketing (ID restaurant, année, mois, chiffre d'affaires, différence entre le chiffre d'affaires courant et celui de l'année précédente)
- DetailDTO : classe contenant les attributs des détails des commandes (ID ticket, menu commandé, prix sans réduction, prix avec réduction, réduction fixe, pourcentage de réduction)

Mise en place de la console

Le projet est codé en java et permet une utilisation via un terminal. Les différentes commandes sont affichées lors du lancement et sont :

Create ambassadeur	Insère un nouvel ambassadeur
Connect id mdp	Connecte l'employé
Get ambassadeur all	Affiche tous les ambassadeurs
Get ambassadeur int	Affiche les infos de l'ambassadeur à l'id int
Update ambassadeur int	Accède à un panel pour modifier les infos d'un ambassadeur
Delete ambassadeur int	Anonymise les informations d'un ambassadeur
Get turnover year all	Affiche le chiffre d'affaires de chaque année pour tous les restaurants, avec la différence entre l'année courante et l'année précédente
Get turnover year int	Affiche le chiffre d'affaires pour une année spécifique
Get turnover month all	Affiche le chiffre d'affaires de chaque mois pour tous les restaurants, avec la différence entre le mois courant et le précédent
Get turnover month int	Affiche le chiffre d'affaires de chaque mois pour tous les restaurants pour une année spécifique, avec la différence entre le mois courant et le précédent
buy	Permet de commander des produits
help	Affiche les commandes
exit	Quitte la base de données

Conclusion et Devis

Lors de ce travail que nous avons réalisé en groupe, nous avons mis en place les techniques vues au cours de nos deux années de cours de base de données. Nous avons pu nous familiariser avec DBMain ainsi qu'avec dataGrip, Docker, JetBrains et JDBC. Ces logiciels permettant la création, l'implémentation, le test et la visualisation de notre base de données. Notre groupe étant bien organisé, il ne nous a pas été compliqué de garder le timing pour les deadlines malgré quelques petits soucis liés à l'utilisation de MySql tel que la gestion de triggers simultanés (cfr. Triggers EQU). Ce qui nous a permis de nous dépasser dans notre réflexion afin de contourner ces problèmes.

Nous avons estimé nos heures de travail et joint un devis fictif à ce document.

FACTURE									
Facture n°		DEVIS							
Date		10-05-22							
<div style="display: flex; justify-content: space-between;"> <div style="width: 45%;"> <p>Unamur Jenny Rue Grandgagnage 21</p> <p>5000 Namur gotyournumberonthewall@jenny.com</p> <p>US 8675309</p> </div> <div style="width: 45%;"> <p>Devis pour Nom Mex&Go SRL Adresse 15 rue Basse Marcelle</p> <p>Code postal - Ville 5000 NAMUR Adresse e-mail mexngo@gmail.com</p> <p>No TVA BE0633965571</p> </div> </div>									
Validité du devis 10-06-22									
Description	commentaire	Réf	Qté (h)	% TVA	Prix unitaire HT	Prix unitaire TTC	Total HT	Total TVA	Total TTC
premier rendez-vous		GRATUIT	RDVG	2	21,0%	0,00 €	0,00 €	0,00 €	0,00 €
Analyse demande client		GRATUIT		4	21,0%	0,00 €	0,00 €	0,00 €	0,00 €
Retour client + corrections				2	21,0%	15,00 €	18,15 €	30,00 €	36,30 €
Domaine d'application final				4	21,0%	15,00 €	18,15 €	60,00 €	72,60 €
Schéma conceptuel				10	21,0%	30,00 €	36,30 €	300,00 €	363,00 €
Schéma logique				2	21,0%	30,00 €	36,30 €	60,00 €	72,60 €
code DDL	conversion			1	21,0%	40,00 €	48,40 €	40,00 €	48,40 €
	corrections EOL			15	21,0%	40,00 €	48,40 €	600,00 €	726,00 €
	boucles			12	21,0%	40,00 €	48,40 €	480,00 €	580,80 €
	rôles			15	21,0%	40,00 €	48,40 €	600,00 €	726,00 €
	relecture			5	21,0%	40,00 €	48,40 €	200,00 €	242,00 €
Déploiement Java	console			15	21,0%	40,00 €	48,40 €	600,00 €	726,00 €
	scenario test			2	21,0%	40,00 €	48,40 €	80,00 €	96,80 €
Total HT 3.050,00 € TVA 640,50 € Total TTC 3.690,50 €								Réduction (travail par des étudiants) 30% TOTAL DEVIS 2583,35	

TVA:8675309 - BELFIUS BE61 0689 0293 3217

Bibliographie

- HAINAUT, J-L., *Bases de données*, Dunod, 2018
- <https://dev.mysql.com/doc/>
- <https://www.geeksforgeeks.org>
- <https://docs.docker.com/>
- <https://www.w3schools.com/>
- <https://www.baeldung.com>
- <https://maven.apache.org/guides/>
- <https://projectlombok.org/features/all>