

Rendu du projet de BDD

Quentin GRUCHET 21605636
Mickaël LE DENMAT 21804355

Décembre 2020

Sommaire

1	PARTIE 1	3
1.1	CHOIX DU PROJET	3
1.2	INFORMATIONS GÉNÉRALES	4
1.3	QUELQUES PRÉCISIONS	6
1.4	SCHÉMA ENTITÉS/ASSOCIATIONS	7
1.5	SCHÉMA RELATIONNEL	8
2	PARTIE 2	8
2.1	RAPIDE EXPLICATION DU FICHIER .SQL	8
2.2	PROBLÈMES RENCONTRÉS	9
2.3	INFORMATIONS PRATIQUES	10

1 PARTIE 1

1.1 CHOIX DU PROJET

Nous avons choisi de modéliser le fonctionnement d'une chaîne de magasin. Nous nous sommes inspirés de magasins alimentaires comme *Auchan*, *Carrefour*, *Leclerc*...

Ce choix nous paraissait évident car l'un de nous travail actuellement au sein de l'une des ces grandes enseignes. Nous avons donc une constatation direct du fonctionnement de ce genre de modèle.

Aussi, nous avons pu poser des questions et recueillir beaucoup d'informations auprès d'acteurs majeurs. Nous avons pu donc produire un schéma Entités/Associations le plus complet possible pour plus de vraisemblance.

1.2 INFORMATIONS GÉNÉRALES

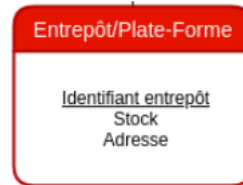
Voici les informations que l'on souhaite considérées et respectées dans notre base de données:

- Un magasin est composé d'une localisation (*il s'agit de la zone géographique : Yvelines, région Parisienne, région Nord...*), d'une taille (*proximité, supermarché ou hypermarché*), d'un identifiant magasin et d'une adresse postale.
- Chaque magasin dispose d'employés dont on souhaite connaître le matricule, le nom, le prénom, l'âge, le sexe et les horaires de chaque employé.
- Tous les employés signent un contrat de travail avec un magasin. Ce contrat dispose d'un numéro de contrat, d'un type de contrat (*CDI, CDD, Intérimaire...*) et d'une durée.
- Un magasin accueille bien sûr des clients avec comme informations une adresse mail, un nom, un prénom, un âge, une adresse postale et un sexe.
- Un client peut avoir une carte de fidélité contenant une cagnotte (*points*), un code à 4 chiffres pour utiliser cette cagnotte et d'un code barres.
- Cette carte de fidélité est gérée par des magasins qui peuvent en modifier les informations.
- Pour se faire approvisionner, le magasin passe commande auprès de la centrale d'achat. Chaque commande a un numéro, une date, des produits et une quantité demandée.
- La centrale, unique pour l'ensemble des magasins de France Métropolitaine, est identifiée par un numéro et une adresse postale.
- Celle-ci passe commande auprès d'un fournisseur pour obtenir les produits.

- Le fournisseur possède des produits caractérisés par un type, un EAN (*code barre a 12 chiffres de chaque produit*), et une marque.
- Ce fournisseur livre ses produits dans un entrepôt de stockage. Bien sur chaque livraison dispose d'une date de départ, d'une date d'arrivée, d'une quantité envoyées, et d'un numéro.
- Enfin, le magasin est livré par l'entrepôt.

1.3 QUELQUES PRÉCISIONS

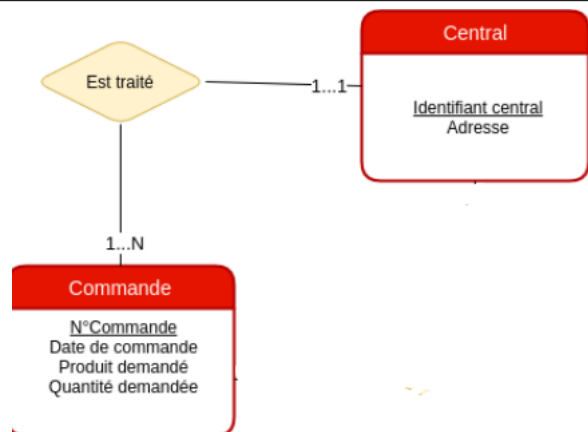
Le diagramme ci-contre représente une entité avec ces attributs. L'attribut souligné correspond à la clé primaire de l'entité.



Le diagramme ci-contre représente une association entre deux entités. Chaque association est explicite pour comprendre la relation entre deux entités.

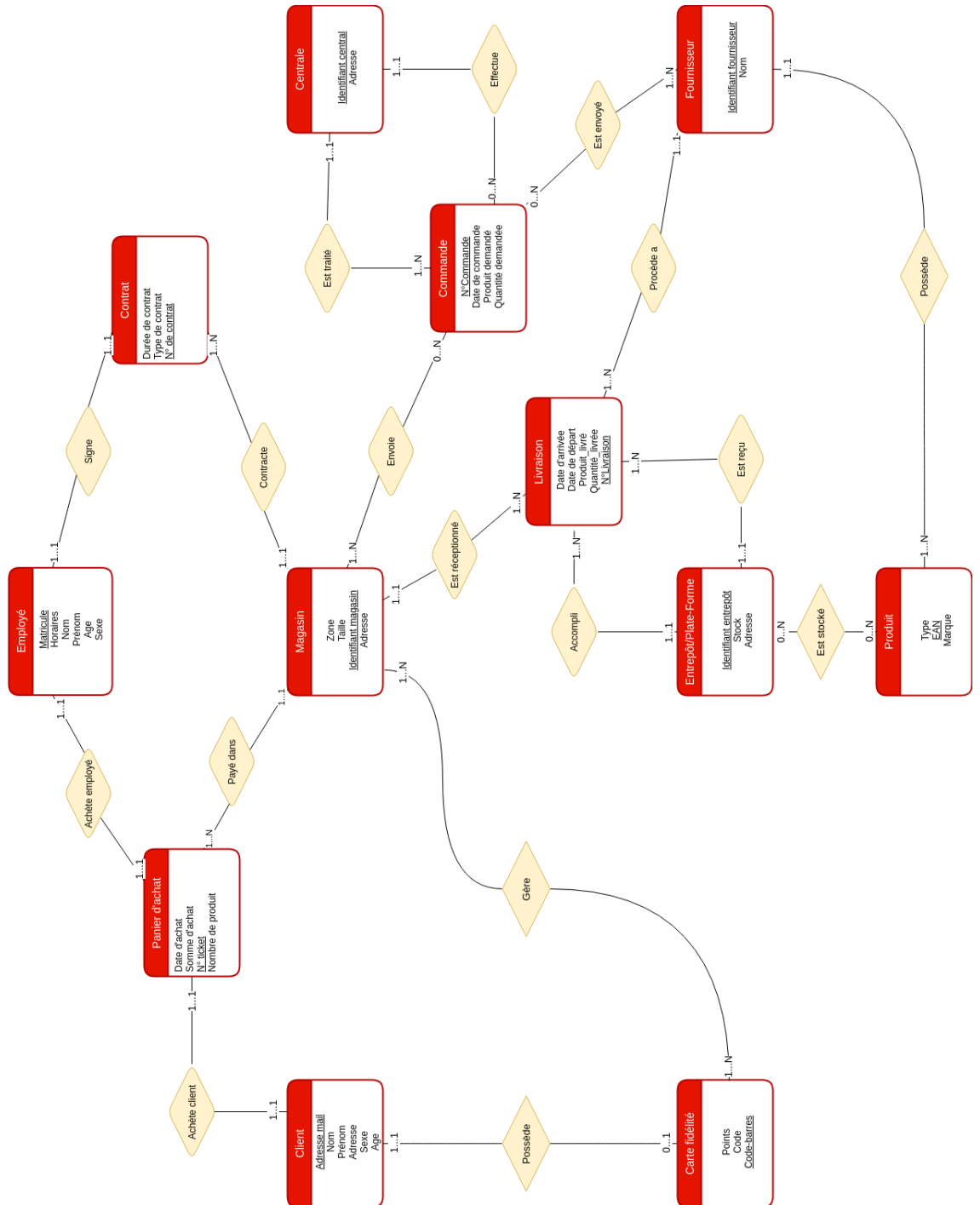


Pour la cardinalité, nous la lisons à l'opposé. Dans l'exemple ci-contre, la commande est traitée par une seule centrale et la centrale traite une ou plusieurs commandes.



Dans notre schéma relationnel, un attribut en vert représente une clé primaire, tandis qu'un attribut en violet représente une clé étrangère.

1.4 SCHÉMA ENTITÉS/ASSOCIATIONS



1.5 SCHÉMA RELATIONNEL

ENTITÉS :

Magasin(Identifiant magasin, Zone, Taille, Adresse)
Client(Adresse mail, Nom, Prénom, Adresse, Sexe, Age)
Employé(Matricule, Horaires, Nom, Prénom, Age, Sexe)
Central(Identifiant central, Adresse)
Fournisseur(Identifiant fournisseur, Nom)
Entrepôt(Identifiant entrepôt, Stock, Adresse)
Produit(EAN, Type, Marque, Identifiant fournisseur)
Carte fidélité(Code-barre, Points, Code)
Panier d'achat(N° ticket, Somme total, Date d'achat, Nombre de produit, Identifiant magasin)
Livraison(N°Livraison, Date d'arrivée, Date de départ, Quantité, Identifiant fournisseur, Identifiant entrepot, Identifiant magasin)
Commande(N° Commande, Date de commande, Produit demandé, Quantité demandée, Identifiant centrale)
Contrat(N°Contrat, Durée de contrat, Type de contrat, Identifiant magasin)

ASSOCIATION :

Est stocké(Identifiant entrepôt, EAN)
Est envoyé(N°commande, Identifiant fournisseur)
Envoie(Identifiant magasin, N°commande)
Signe(N°contrat, Matricule)
Achète-client(N°ticket, Adresse mail)
Achète-employe(Matricule, N°ticket)
Possède(Code-Barre, Adresse mail)
Gère(Code-Barre, Identifiant magasin)

2 PARTIE 2

2.1 RAPIDE EXPLICATION DU FICHIER .SQL

Le procédé est assez simple. Nous créons tout d'abord une table avec ses définitions d'attributs. Nous insérons ensuite toute les valeurs que nous jugeons utiles pour notre base de données. Nous répétons ces deux opérations jusqu'au moment où toute les tables soit créées. Après cela nous créons les view. Dernière et non des moindres étape, nous définissons toutes les contraintes de clés primaires et de clés étrangères sur nos tables.

2.2 PROBLÈMES RENCONTRÉS

- Un client ne peut pas consulter le contenu ou l'historique de ses paniers d'achat. Nous avons eu l'idée qu'à chaque utilisateur, nous lui attribuons un ID unique qui est une clé primaire dans la table Panier. Malheureusement nous n'avons pas trouvé la solution pour qu'un client se connecte et puisse, retrouver via son ID, le contenu de ses paniers.
- Nous avons décidé de commencer la programmation par la création de toutes les tables puis ensuite de les remplir avec des données. Cependant lors de nos déclarations nos types d'attributs n'étaient pas forcément adaptés aux données de remplissage. Nous avons donc, et ceux à de multiple reprises, modifié nos types attributs (souvent des clés étrangères, ce qui nous a pris plus de temps que prévu).
- Nous nous sommes aperçus d'une erreur de cardinalité dans notre diagramme Entité/Association au niveau de la table d'association entre entrepôt et livraison. Cela nous a posé un gros soucis lorsque l'on a essayé de créer la table.
- Nous avons quelques débats concernant les droits d'utilisateurs. Nous avons dû nous mettre à la place du client par exemple, pour déterminer quels droits étaient attribués et à qui ils l'étaient.
- Pour rendre notre base de données la plus ressemblante à la réalité possible, nous avons dû insérer beaucoup de données et cela nous paraissait long. Nous avons donc, pour plusieurs tables, utilisé un générateur de données en ligne qui nous permettait de nous donner en peu de temps. Par exemple le générateur nous donnait un ID, une date ou encore un prénom avec des intervalles, formes, choisies à l'avance. Cependant le générateur ne nous permettant pas de remplir l'entièreté de notre base de données, nous avons écrit, en langage CPP, des programmes nous permettant de lire des tables déjà remplies afin d'en remplir d'autres. Ces programmes nous ont été très utiles notamment pour les tables d'association où les répétitions sont nombreuses. Malheureusement comme les données sont générés aléatoirement, nous n'avons pas vérifié que les données étaient cohérentes les unes par rapport aux autres.
- Enfin nous nous sommes aperçus que la table Achete était dès l'origine mal conçue. En effet nous avons besoin du matricule

dans le cas où l'employé est un client, d'une adresse mail dans la cas d'un client lambda ainsi que d'un numéro du ticket. Toutes les trois étaient des clés primaires/étrangères et par conséquent ne pouvaient être nul. Nous avons donc modifié notre schéma pour séparer en deux tables distincts (Achete-client et Achete-employe).

2.3 INFORMATIONS PRATIQUES

Nous avons dans l'archive mis deux fichiers .sql car l'un de nous travaille sur un linux natif tandis que l'autre travaille sur machine virtuelle (celle de la fac). Pour un linux natif veuillez utiliser Auchan.sql. Pour la VM de la fac veuillez utiliser Auchan2.sql.