

L3 Informatique - Module Base de données
Projet



Gestion de production en entreprise



Rapport du Projet de Base de données

Maréchal Anthony et Benoit

Table des matières

1	Introduction	3
2	Modélisation du système de données	3
2.1	Diagramme Use Case	3
2.1.1	Introduction	3
2.1.2	Caractéristiques du diagramme	3
2.1.3	Discussion du diagramme proposé	4
2.1.4	Les Scénarii	7
2.1.5	Bilan	9
2.2	Diagramme de Classe	9
2.2.1	Relation entre diagramme Use Case et diagramme de Classe	9
2.2.2	Définition des classes et de leurs relations	9
2.2.3	Discussion	11
2.2.4	Bilan	12
2.3	Schéma relationnel	12
2.3.1	Méthodologie	12
2.3.2	Explications liées au passage au schéma relationnel	14
2.4	Modèle physique des données	14
2.5	Choix d'implantation	15
3	Réalisation SQL	15
3.1	Introduction	15
3.2	Contraintes d'intégrité	15
3.3	Les triggers	15
3.4	Procédure (Programme PL/SQL)	17
3.5	Problèmes rencontrés	17
4	Conclusion	18
5	Bibliographie	19

1 Introduction

Ce projet de bases de données est issu du projet commun aux unités d'enseignement de modélisation¹ et de bases de données², et à pour but de décrire le système d'information d'une entreprise fonctionnant selon une stratégie de gestion de production particulière. La gestion de production choisit est la méthode dite ABC qui a été sélectionnée suite à une étude effectuée sur les différentes stratégies de gestion de production. A l'issue de cette étude et de la modélisation du système d'information de l'entreprise en UML³, nous devons traduire notre représentation en un modèle en base de donnée relationnel puis implanter celui-ci dans le SGBD⁴ d'Oracle. Le projet comporte deux parties, la première ayant pour but de passer du modèle conceptuel au modèle relationnel et la deuxième étant d'implanter ce modèle dans le SGBD d'Oracle. Dans cette première partie nous retrouverons les diagrammes Use Case et de classe issues de notre modélisation UML, ainsi que l'étude du passage au schéma relationnel à partir du diagramme de classe et le modèle physique des données. Nous terminerons cette partie concernant la modélisation du système d'information par la justification de nos différents choix d'implantation du schéma relationnel. La deuxième partie, quant à elle, sera constituée d'explications sur les différents moyens mis en place pour implanter le modèle établi dans la première partie. Nous détaillerons donc les différentes contraintes d'intégrités, triggers et programmes PL/SQL nécessaire à la réalisation SQL⁵ sous Oracle du projet. Pour l'implantation nous avons utilisé le SGBD d'Oracle 10g, dix tables et quatorze programmes PL/SQL ou trigger comme le sujet nous l'imposait.

2 Modélisation du système de données

Afin d'avoir une vision de haut niveau du système et pour en préciser les frontières nous avons développé le diagramme Use Case ce qui nous permettra de cerner les cas d'utilisations en correspondance avec les différents acteurs de ce système d'information. Ensuite nous donnerons le diagramme de classe et ses explications ce qui nous permettra d'étudier les différences existantes avec le schéma relationnel établi à partir de celui-ci. Enfin nous présenterons le modèle physique des données qui découle du schéma, et nous verrons finalement les choix effectués pour l'implantation de telle partie ou fonctionnalités du modèle relationnel.

2.1 Diagramme Use Case

2.1.1 Introduction

Pour créer les différents diagrammes nous avons dû effectuer des recherches sur le fonctionnement de la gestion de la production en entreprise, en particulier sur la mise en œuvre de la méthode ABC et plus particulièrement pour le Use Case. Pour cela nous avons utilisé de nouveaux livres ainsi que nos connaissances personnels⁶ sur le sujet ; sans oublier les diverses indications de nos professeurs lors de la validation de nos diagrammes.

2.1.2 Caractéristiques du diagramme

Pour construire le diagramme Use Case nous avons synthétisé l'ensemble des informations suivantes :

¹Unités d'enseignement du premier semestre de licence 3 d'informatique, module nommé i5tc4 ou MO1

²Unités d'enseignement de licence 3 d'informatique, module nommé i5tc2 ou BD

³Unified Modeling Language

⁴Système de Gestion de Base de Données

⁵Structured Query Language

⁶Basées sur une expérience de 6 mois en entreprise comme assistant magasinier pour Benoit et en travail en équipe (4*8) pour Anthony, tout deux à Kodak Industrie

- La gestion de la production d’une entreprise peut être caractérisée par trois grands ensembles de fonctionnalités. Elle doit gérer les matières premières, la production, ainsi que les produits finis.
- La gestion des matières premières concerne différents acteurs avec tout d’abord le responsable des commandes qui lance une commande chez un fournisseur, le comptable qui vérifie que la commande passée respecte le budget, et enfin à l’arrivée de la commande le magasinier la contrôle, la stock et confirme sa réception en la saisissant dans le système d’information.
- La gestion de la production concerne quant à elle un responsable de production⁷, qui va encadrer la production et notamment les équipes de productions qui réalise le "travail effectif" de production⁸ en vérifiant également le bon respect de la politique de production.
- La politique de production, consiste à définir les différentes valeurs telles que les périodicités, les quantités à produire, les seuils de réapprovisionnement des matières premières, les seuils de production limite, afin d’assurer un stock de sécurité et les durées de production entre les groupes de produits. Cette politique est définie par le gestionnaire, appliquée par les équipes de production et vérifiée par le responsable de production.
- La gestion des produits finis, consiste à stocker (grâce au magasinier) et vendre les produits aux clients par l’intermédiaire du responsable des expéditions qui assure le suivi des livraisons, elle est encadrée par l’agent comptable qui vérifie les transactions.
- A ces trois fonctionnalités principales vient s’ajouter une fonctionnalité transversale commune à toutes car devant se retrouver à tous les niveaux de la gestion de la production : la méthode ABC.
- La méthode ABC étant une fonctionnalité complexe et capitale puisqu’elle intervient dans les trois fonctionnalités principales, nous avons choisi de faire un Use Case spécialement pour la détailler.
- Cette méthode peut se décomposer en quatre sous parties, avec dans un premier temps la distinction des produits en groupe, il s’agit donc d’analyser les ventes puis de classer les produits en groupes ce qui est, respectivement, le travail de l’analyste et du gestionnaire, puis dans un deuxième temps de définir les différentes politiques d’approvisionnement et de production précisant les différences entre ces groupes, ce qui est également le travail du gestionnaire ou d’une équipe de gestion.

Vous trouverez ci-après les deux diagrammes Use Case représentant la synthèse précédente en UML 2.0, le premier représente la vue générale (Fig 1), tandis que le second présente le détail de la méthode ABC (Fig 2).

2.1.3 Discussion du diagramme proposé

Afin que notre modèle puisse s’appliquer à un maximum d’entreprise ayant choisi la méthode ABC comme stratégie, nous avons choisi de garder un niveau d’abstraction assez élevé. C’est pourquoi nous n’avons pas voulu multiplier les acteurs, en préférant conserver les acteurs dont les rôles sont parfaitement définis et communs à toutes les productions. Malgré la présence d’un responsable des commandes et d’un responsable des expéditions, nous avons choisi de représenter un Comptable. Ayant généralement une position hiérarchique plus élevée que ces

⁷Il est appelé également "encadrant"

⁸les productions des entreprises fonctionnent généralement 24h/24h 7j/7j, les équipes de production sont alors en 4*8 le poste d’encadrant est alors réalisé par au moins 4 personnes

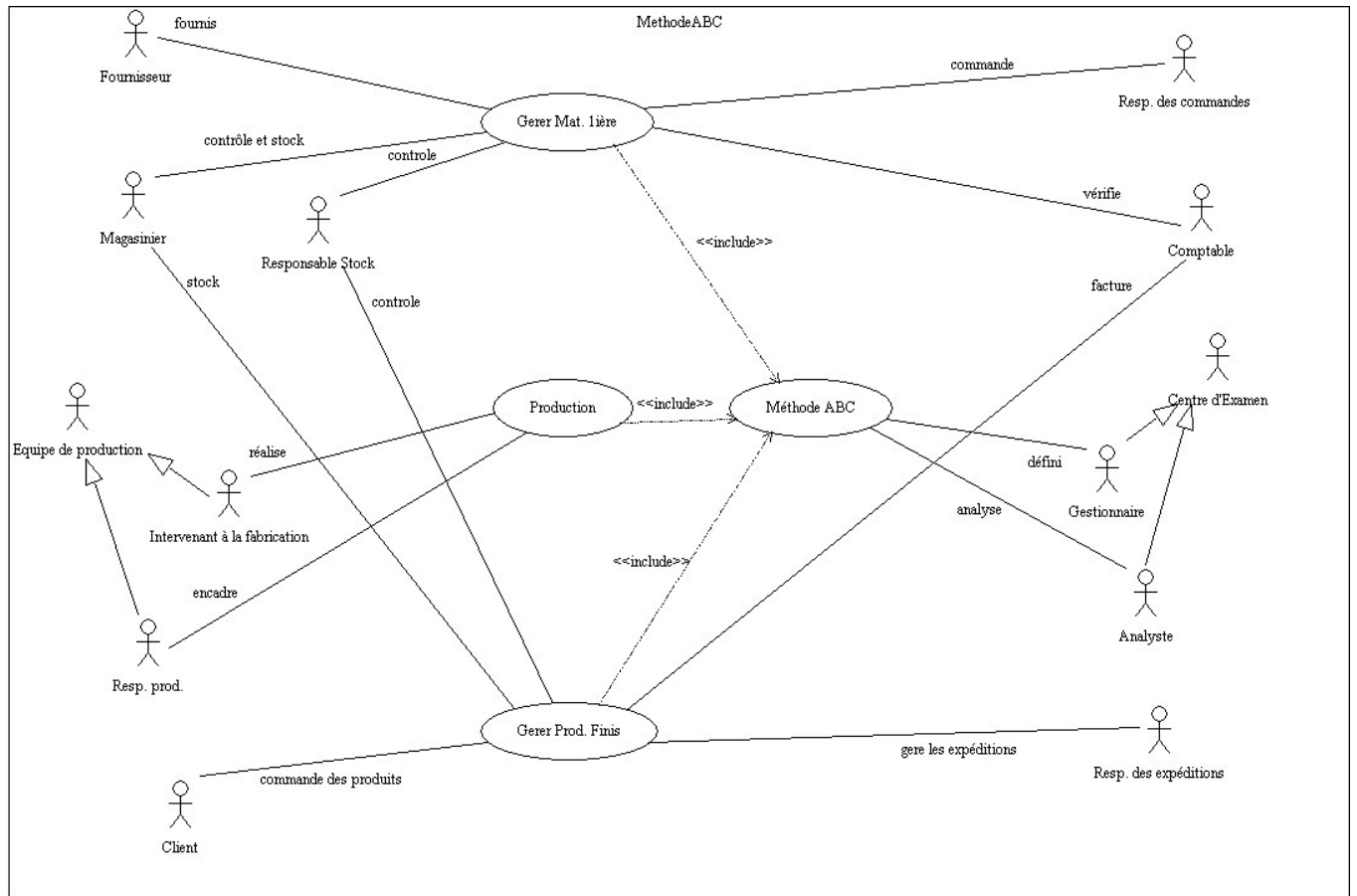


FIG. 1 – Diagramme Use case générale

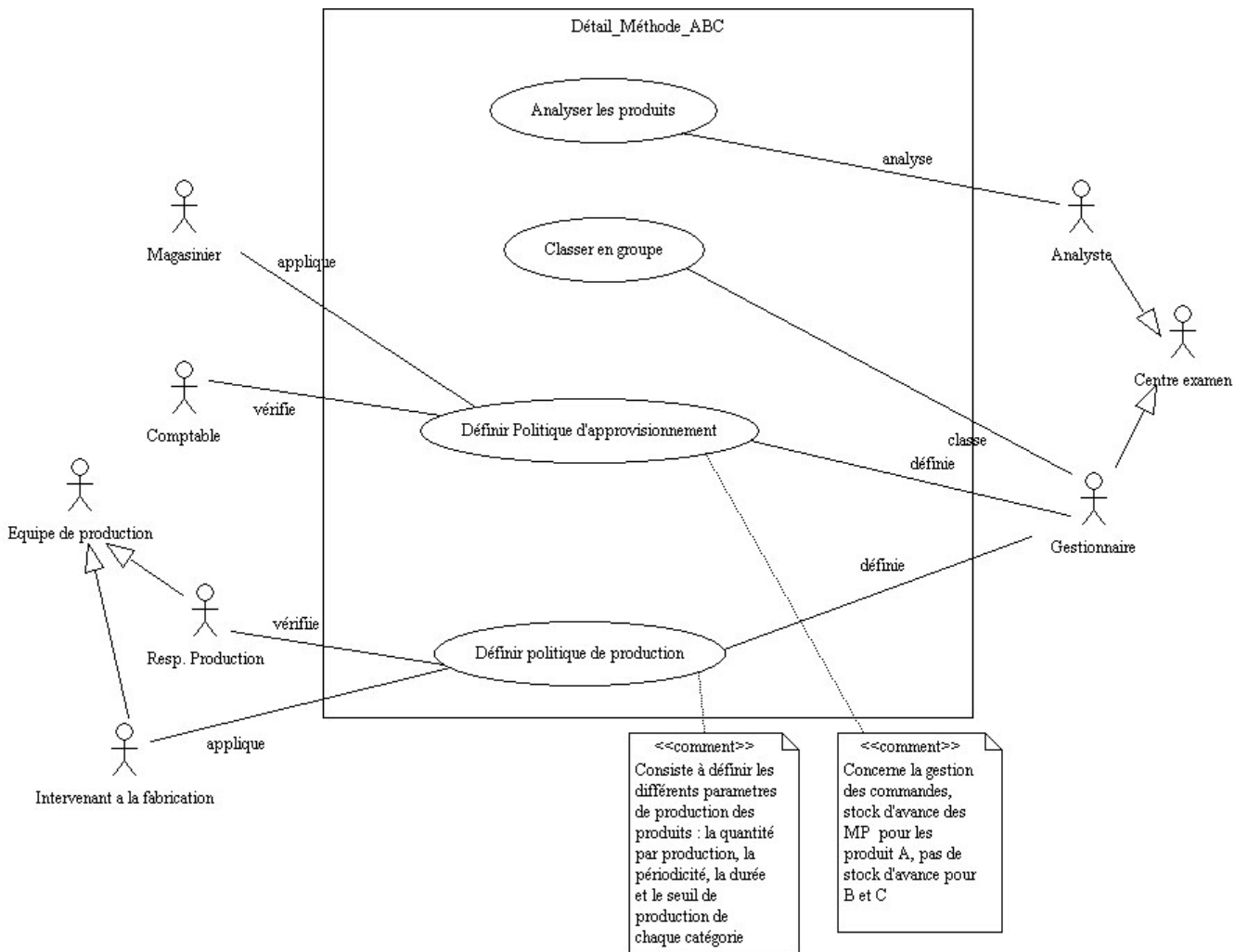


FIG. 2 – Diagramme Use Case

derniers, le comptable à deux rôles essentiels : le premier est d'assurer un lien logique entre les commandes et les expéditions pour le maintien du budget, le second est d'assurer la transparence des transactions des responsables des commandes et des expéditions. Le responsable du stock travail conjointement avec le comptable⁹, son rôle essentiel est la vérification du stock notamment sur les quantités effectives des produits (bien souvent différentes des quantités théoriques).

2.1.4 Les Scénarii

Voyons à présent les scénarii des fonctionnalités présentes dans le diagramme Use Case du "détail de la méthode ABC" (Fig 2).

Sommaire d'identification¹⁰ :

Titre : Analyser les produits

Résumé : Ce cas d'utilisation permet d'obtenir les différentes valeurs des ventes et des quantités de produits stockés nécessaires au classement par groupe.

Scénario principal :

1. Rassembler les quantités de produits stockés
2. Rassembler les chiffres d'affaires de chacun des produits
3. Calculer les différents pourcentages de chaque produit par rapport à la quantité total stockée
4. Calculer les différents pourcentages de chaque produit par rapport au chiffre d'affaire total
5. Faire un rapport des résultats avec un tableau ou un graphique reprenant l'ensemble des résultats obtenus

Sommaire d'identification

Titre : Classer en groupe

Résumé : Cette fonctionnalité permet de répartir les produits selon les trois groupes de la méthode ABC.

Précondition :

- Le rapport de l'analyste doit être rendu et juste.

Scénario principal :

1. Choisir les différents pourcentages de quantité et de chiffres d'affaires qui caractérisent les groupes A, B et C
2. Répartir les produits dans ces groupes grâce au rapport de l'analyste

⁹Noté que la plupart des entreprises actuels emploient une seule personne pour assurer ces deux rôles

¹⁰La syntaxe choisie pour représenter les scénarii est celle proposé par le livre "UML 2 par la pratique". Voir l'annexe pour connaître sa référence

Scénario secondaire, erreur 1 (E1) :

E1 : Cette erreur se produit lorsqu'un produit possède des pourcentages qui n'entrent dans aucune des catégories A, B ou C, elle démarre au point 2 du scénario principal.

3. Définir l'importance du produit dans l'entreprise.
4. Voir si une modification des pourcentages retenus au point 1 peut être apportée pour régler le problème.
5. Si le produit a une très grande importance on peut remettre en cause le choix de la méthode ABC et voir si la méthode des 80/20 pourrait s'appliquer.
6. Si au contraire le produit a une faible importance réfléchir à l'arrêt de sa production au sein de l'entreprise.

Sommaire d'identification

Titre : Définir la politique d'approvisionnement

Résumé : Cette fonctionnalité consiste à choisir les différents seuils de réapprovisionnements des matières premières.

Précondition :

- Les produits doivent être classés par groupe.

Scénario principal :

1. Choisir les différents pourcentages de quantités et de chiffres d'affaires qui caractérisent les groupes A, B et C
2. Examiner les différents prix et délais des fournisseurs
3. Choisir les fournisseurs
4. A partir des groupes de produits et de la capacité maximale du stock, définir les seuils de réapprovisionnement maximal et minimal pour chaque groupe, puis pour chaque produit

Sommaire d'identification

Titre : Définir la politique de production

Résumé : Cette fonctionnalité consiste à choisir les différentes valeurs¹¹ entrant en jeu dans la production des produits finis de l'entreprise.

Précondition :

- Les produits doivent être classés par groupe.

Scénario principal :

¹¹ les détails de ces valeurs sont donnés dans la discussion du diagramme de classe

1. Selon le groupe d'un produit le gestionnaire doit définir les différentes valeurs de la production qui caractérisent chaque catégorie.
2. Il doit ensuite définir plus spécifiquement ces valeurs pour chaque produit fini.

2.1.5 Bilan

Le travail mené pour réaliser ce diagramme repose essentiellement sur les choix des acteurs et fonctionnalités à y faire figurer. Lorsque le rôle d'un acteur n'était pas parfaitement défini ou qu'il pouvait être englobé par le rôle d'un autre acteur nous avons décidé de ne pas le faire figurer dans le schéma.

Comme nous l'avons précisé précédemment, notre diagramme repose sur un niveau d'abstraction assez élevé ce qui lui permet d'être applicable dans la plupart des entreprises s'appuyant sur la méthode ABC. Mais cet avantage, peut être perçus comme un inconvénient dans le sens où il ne représente pas plusieurs détails de la réalité. La fonctionnalité "gérer produit finis" par exemple aurait pu être plus détaillé en distinguant leur stockage de leurs expéditions, il en est de même pour la fonctionnalité "gérer matière première". Une autre remarque peut être émise sur la spécialisation du centre d'examen en deux nouveaux acteurs que sont l'"analyste" et le "gestionnaire", ces deux rôles étant généralement réalisés par une même équipe cependant nous avons voulu les distinguer afin de mettre en avant leur particularité.

2.2 Diagramme de Classe

2.2.1 Relation entre diagramme Use Case et diagramme de Classe

Nous avons élaboré le diagramme de classe (Fig 3) à partir des diagrammes Use Case qui nous ont conduit aux remarques suivantes :

- La première fonctionnalité du diagramme Use Case général (Fig 1) est "gérer les matières premières", cette fonctionnalité regroupe un ensemble d'informations sur les matières premières avec notamment leurs commandes, qui prend allure d'une transaction entre deux entreprises. L'une est un fournisseur et l'autre l'entreprise modélisée. La commande concerne un ensemble de matières premières ainsi que leurs quantités commandées. Elle rassemble également la notion de stockage des matières premières.
- La fonctionnalité "Production", concerne le processus de transformation des matières premières en produit fini. Pour cela, on utilise une nomenclature qui associe à chaque produit fini l'ensemble des matières premières ainsi que leurs quantités. Il faut aussi définir les quantités de production, les périodicités, et les durées pour chacun des produits.
- La fonctionnalité "gérer produits finis" quant à elle, regroupe la notion de stockage, ainsi que la notion de vente du produit à un client.
- Enfin la fonctionnalité "méthode ABC", avec ces sous fonctionnalités "Définir la politique d'approvisionnement" et "Définir la politique de production", peut être regroupée dans une classe qui aura pour but de définir les valeurs maximale et minimale de production pour chacun des trois groupes.

2.2.2 Définition des classes et de leurs relations

La plupart des classes du diagramme sont reliées par des relations d'associations, hormis les classes Stock et Nomenclature. Le Stock étant composé de matières premières et de produits finis, c'est pourquoi ces classes sont reliées par des relations de compositions.

En premier lieu, nous avons créé une classe Tiers¹² qui définit, le nom, le numéro de téléphone et de fax, le nom de responsable des commandes, l'adresse¹³ et l'identifiant bancaire des deux intervenants lors d'une transaction. Une transaction entre deux tiers peut prendre l'allure d'une commande ou d'une expédition, une entreprise peut faire plusieurs commandes ou expéditions vers une autre entreprise mais une transaction n'implique qu'une seule entreprise qui commande et qu'une seule entreprise qui fournit d'où les deux associations entre les classes Tiers et Transaction. Ainsi lors de l'approvisionnement, l'entreprise qui passe commande est l'entreprise modélisée et celle qui fournit est le fournisseur, tandis que lors de l'expédition de produits, l'entreprise qui passe commande est le client et l'entreprise qui fournit est l'entreprise modélisée. Dans une transaction plusieurs paramètres sont pris en compte : son type (CommandePF ou LivraisonMP), la date, la quantité totale de marchandise¹⁴ et une liste de matières premières ou de produits finis (selon son type) associée à une quantité.

Afin de connaître les quantités totales de matières premières ou de produits finis dans le stock de l'entreprise, nous avons défini une classe Stock. Cette classe possède une quantité totale, une quantité max et une liste de matières premières ou de produits finis selon son type.

Tout ce qui touche à la production ou aux produits finis est défini dans la classe ProduitFinis. Cette classe comporte le nom du produit, son code de référence et sa quantité en stock. Elle définit également les paramètres de production du produit tel que la périodicité, la durée de production, la quantité à produire par production, elle possède aussi une nomenclature qui définit une liste de matières premières associées à leurs quantités. Pour la facturation, elle possède un attribut indiquant son prix coûtant, et la marge de l'entreprise (ce qui permet de calculer le prix de vente). Chaque matière première et produit fini appartient à une classe, leurs classes respectives sont donc une spécialisation de la classe "Groupe". Cette classe définit les valeurs minimales et maximales de production et de réapprovisionnement des produits relatifs à chacun des trois groupes A, B ou C. On pourra par exemple pour un produit du groupe A, définir le seuil minimum critique enclenchant une nouvelle campagne de production du produit, il sera également intéressant de définir la périodicité et la durée maximale de ce groupe de produit, l'objectif primordial pour les produits de la catégorie A étant de ne jamais être en rupture de stock. Il en est de même pour les groupes B et C.

2.2.3 Discussion

La réalisation d'un diagramme de classe nous a demandé de faire beaucoup d'hypothèses et de choix, c'est pourquoi nous allons prendre le temps de discuter des divers choix qui ont dû être pris.

Ce qui vient à l'esprit en premier lieu est d'avoir regroupé les différents acteurs qui sont le fournisseur, le client et l'entreprise modélisée au sein d'une unique classe : "Tiers". En effet les attributs de cette classe caractérisant bien ses trois acteurs, il ne nous a pas paru essentiel de faire de nouvelles classes. Cependant, si notre diagramme devait être repris par une autre personne, et que celle-ci veuille apporter de nouveaux attributs caractérisant par exemple un peu plus le client et le fournisseur et l'entreprise modélisée, elle pourrait tout à fait rajouter ses nouvelles classes en faisant une spécialisation sur notre classe Tiers, sans avoir à modifier d'avantage le diagramme. Nous pouvons émettre la même remarque pour la classe "Transaction" qui aurait pu être spécialisée en sous classe "Livraison" et "Commande" si des attributs supplémentaires avaient besoin d'être apportés.

Deuxièmement, nous avons fait le choix de ne pas représenter la production par des ateliers, en préférant inclure les attributs de production dans la classe ProduitFinis car les entreprises ont chacune leur manière de répartir la production de leurs produits dans leurs ateliers. Notre diagramme aurait alors perdu de sa généralité. Toujours en ce qui concerne la production, la méthode ABC nous permet de considérer celle-ci comme étant périodique. Ainsi un produit fini est fabriqué régulièrement dans une quantité donnée, cependant si ce produit arrive tout de même

¹²Le terme Tiers est préférable à celui d'Entreprise car cette classe peut représenter des particuliers

¹³Adresse définie par l'intermédiaire de la classe Adresse

¹⁴Quantité calculée à partir de la somme des quantités de chaque matière

au seuil de production alors une nouvelle production est enclenché d'urgence, afin de garantir un stock de sécurité. Inversement lorsqu'une nouvelle production doit être faite alors que le seuil maximum pour ce produit est atteint, la production n'est alors pas enclenchée. Une matière première peut entrer dans la fabrication d'une ou plusieurs classes il se pose alors le problème du groupe auquel la matière première doit appartenir. Nous avons donc choisi de faire appartenir ces matières premières au groupe le plus élevé en importance au produit dont elle fait partie de sa nomenclature : Exemple : Une matière première X intervient dans la fabrication d'un produit du groupe A et d'un produit du groupe B, cette matière première appartiendra alors au groupe A.

2.2.4 Bilan

La réalisation de ce diagramme de classe, nous a demandé de reprendre de manière beaucoup plus approfondi notre étude bibliographique. Nous avons dû faire beaucoup d'hypothèses et la littérature sur la gestion de la production ne se prête pas toujours à la modélisation UML dans un niveau d'abstraction donné, c'est pourquoi nous avons dû adapter nous même la manière dont les classes se relient. Aux qualités de notre diagramme nous voyons en premier lieu, sa clarté et sa cohérence générale. Clarté, car nous n'avons pas tenté de multiplier les classes pour arriver à dépassé le plus possible le nombre de tables minimum imposés par le sujet. Cohérent, parce que chacune des notions de la gestion de la production ont été définies, mises en relation et justifiées dans leur choix. Son défaut serait peut-être de n'avoir pas fait figurer une classe représentant la production proprement dite, avec les ateliers, et la répartition des produits par atelier, mais comme nous l'avons justifié dans la partie discussions, cela nous semblait descendre trop bas dans le niveau d'abstraction que nous nous étions fixé pour ce diagramme.

2.3 Schéma relationnel

Maintenant que nous avons vu nos choix qu'en à la mise en place de notre modèle conceptuel, nous allons désormais étudier le passage au schéma relationnel de notre diagramme de classe. Nous verrons d'abords la méthodologie employée, puis nous donnerons le schéma relationnel que nous accompagnerons d'explications. Ces démarches nous permettront de donner le modèle physique des données. Enfin nous discuterons des différents choix d'implantation que nous avons finalement réalisé au coeur de notre projet.

2.3.1 Méthodologie

Nous allons rappeler brièvement les différentes règles nécessaires pour construire le schéma relationnel d'une base de données à partir de son diagramme de classe :

1. Signaler les identifiants de chaque classe, ou en créer si nécessaire.
2. Les variables doivent être obligatoirement typées.
3. Les classes étant associées à une multiplicité de 1 à 1..* , doivent avoir au sein de ces attributs l'identifiant de la classe liée.
4. Les associations ayant une multiplicité de 1..* à 1..* , doivent donner lieu à la création d'une nouvelle classe possédant deux nouvelles associations qui auront une multiplicité de la forme : 1 à 1..* et 1..* à 1. Celle classe possèdera donc les identifiants de chaque classe liée (règle 3).

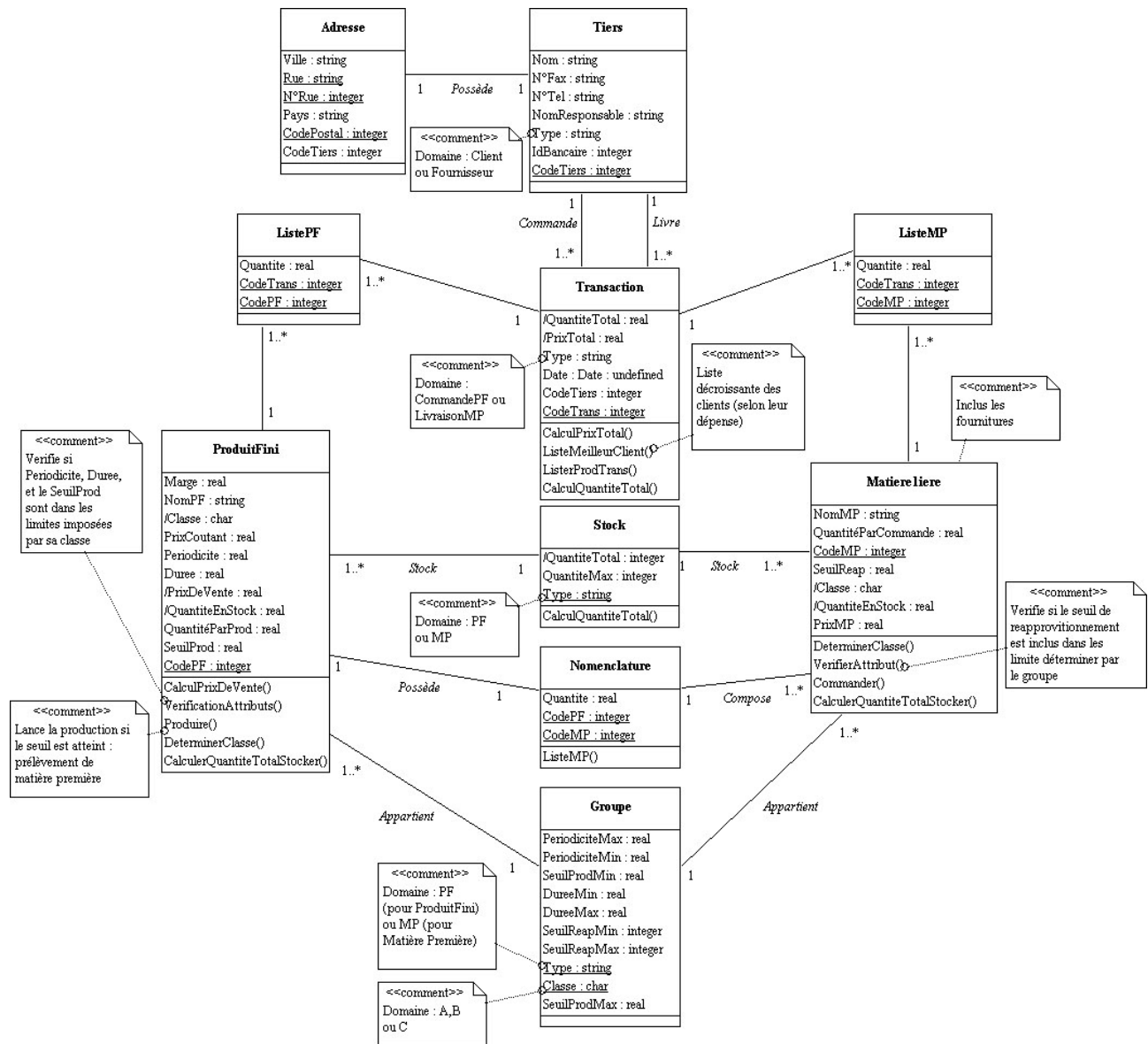


FIG. 4 – Schéma relationnel

5. Les classes d'association seront, quant à elles, reliée directement à chaque classe dont elle était l'association ; avec une multiplicité équivalente à celle donnée en règle 4 (c'est-à-dire 1 à 1..* et 1..* à 1), et possèdera logiquement les identifiants de chaque classe liée (règle 3).

2.3.2 Explications liées au passage au schéma relationnel

Grâce à la méthodologie précédente nous avons pu établir le schéma relationnel (Fig 4)

Nous allons donc maintenant dresser la liste des différences avec le diagramme de classe et donner les explications liées à ces modifications :

- L'association de composition entre ProduitFini et Transaction qui comprend une classe d'association ListePF devient, comme la règle 5 le prévoit, une association 1 à 1..* vers la table ListePF et un lien 1..* à 1 de cette table à la table Transaction.
- Il en est de même pour l'association de composition reliant la table Matiere1iere à la table Transaction et comprenant la classe d'association ListeMP.
- Les relations de compositions "Commande" et "Fournit" deviennent logiquement de simple association ayant la même multiplicité.
- Il en est de même pour les relations de composition entre les tables ProduitFini - Stock et Matiere1iere - Stock.
- La spécialisation sur la table Groupe devient en schéma relationnel deux association de multiplicité 1 à 1..* puisqu'on peut dire que les deux classes spécialisées appartiennent à la classe dont elles étaient la spécialisation.

Nous constatons donc que les règles issues de la méthodologie permettent de créer le schéma relationnel de manière machinale il en est de même pour le reste des modifications apporté aux diagrammes de classe, ce qui explique pourquoi il existe des logiciels (comme objecteering) qui proposent d'effectuer automatiquement de tels modifications.

2.4 Modèle physique des données

A partir du schéma relationnel il est aisé d'obtenir le modèle physique des données. Les attributs soulignés représente les clés primaires et les attributs en *italique* représentent les clés étrangères. Voici notre modèle :

ADRESSE (CodePostal, Rue, NRue, Ville, Pays, *CodeTiers*)

TIERS (CodeTiers, Nom, NFax, NTel, NomResponsable, Type, IdBancaire)

TRANSACTION (CodeTrans, QuantiteTotal, Date, PrixTotal, Type, *CodeTiers*)

LISTEMP (CodeTrans, *CodeMP*, Quantite)

LISTEPF (CodeTrans, *CodePF*, Quantite)

PRODUITFINI (CodePF, NomPF, SeuilProd, Periodicite, Duree, QuantiteParProd, QuantiteEnStock, Prix-Coutant, Marge, PrixDeVente, *Classe*)

MATIERE1IERE (CodeMP, NomMP, PrixMP, SeuilReap, QuantiteParCommande, QuantiteEnStock, *Classe*)

STOCK (Type, QuantiteTotal, QuantiteMax)

NOMENCLATURE (CodePF, CodeMP, Quantite)

GROUPE (Classe, Type, PeriodiciteMin, PeriodiciteMax, SeuilProdMin, SeuilProdMax, DureeMin, DureeMax, SeuilReapMin, SeuilReapMax)

2.5 Choix d'implantation

Les différents choix d'implantation que nous avons effectivement modélisé sont : les dix tables du modèle physique de données, les différentes opérations notés sur le schémas relationnel ont été implémenté mais non-vérifié suite à un jeux de test inexistant. Ces triggers ont donc simplement été codés et sont compilable. Les triggers aurait du avoir pour objectifs de garantir l'intégrité de la bases de données. Malgré tout, nous avons dû effectuer des choix d'implantation comme pour la table stock qui ne permet finalement que de contenir deux tuples (la quantité totale stocké de matière première et la quantité total stockée de produit fini) alors que celle-ci était initialement prévu pour contenir les quantités de chaque produit (ou matière première). Nous avons opter pour cette implémentation car il est apparu qu'il était plus logique d'ajouter un attribut QuantitéEnStock dans la table ProduitFini afin de facilité les différentes requête.

3 Réalisation SQL

3.1 Introduction

L'implémentation dans le SGBD d'Oracle en langage SQL a été effectuée sur les machines de la salle D119. Nous allons maintenant définir quelles ont été les différentes contraintes d'intégrité, triggers, et programme PL/SQL qui ont été nécessaire pour réaliser l'implémentation de notre système d'information.

3.2 Contraintes d'intégrité

Les contraintes qu'il a fallut définir concernent plusieurs thèmes. Le premier était d'assurer l'intégrité référentielle des données. Donc les clés primaires et étrangères ont été implémentées de manière à correspondre exactement au schéma relationnel afin d'assurer cette intégrité. Le deuxième thème concerne le contrôle des données, pour cela nous avons dû définir le "champs d'action" de chaque attribut afin de connaître les différentes valeurs qu'il pouvait prendre au sein de notre représentation. Nous pouvons ainsi synthétisé les différents contrôle qu'il à été nécessaire d'insérer en deux principaux points :

- Les quantités, et les prix sont des réels et doivent être strictement positifs.
- Certains attributs doivent posséder des domaines de valeur limité à des valeurs prédéfinies ; comme les attributs "Classe" (domaine 'A', 'B' ou 'C') et "Type" (domaine 'Produit Fini' et 'Matiere Premiere') par exemple.

3.3 Les triggers

Afin d'avoir un système d'information le plus autonome possible nous avons créer des triggers qui nous permette entres autres de mettre à jours les données. Nous allons voir en détails chaque triggers que nous avons implémenté en décrivant ce pourquoi il à été créer et sont fonctionnement.

VERIFICATION_ATTRIBUTS_PF permet de vérifier si les attributs de la table *ProduitFini* correspondent bien au domaine de définition déterminé par la table *Groupe*. Ce trigger va vérifier si les attributs "Periodicite", "Duree", "SeuilProd" sont bien dans les limites définies entre les champs "PeriodiciteMin"- "PeriodiciteMax", "DureeMin"- "DureeMax" et "SeuilProdMin"- "SeuilProdMax" présent dans la table *Groupe*.

VERIFICATION_ATTRIBUT_MP permet de vérifier si l'attribut "SeuilReap" de la table *MatiereLiere* correspondent bien au domaine de définition déterminé par la table *Groupe*. Il s'agit du même fonctionnement que la vérification des attributs de la table *ProduitFini*.

CALCUL_QUANTITE_TOTAL_PF fait la somme de toutes les quantités en stock de l'ensemble des produits finis, ce qui permet de remplir l'attribut "QuantiteTotal" de la table *Stock*.

CALCUL_QUANTITE_TOTAL_MP comme pour le trigger précédent, afin de remplir la table *Stock* nous effectuons le calcul de la quantité total de matière première.

CALCUL_PRIXDEVENTE, ce trigger permet de remplir l'attribut "PrixDeVente" automatiquement. En effet, grâce aux prix coûtant et au pourcentage de marge définis dans la table *ProduitFini* il est aisément possible de calculer le prix de vente avec la formule suivante : $\text{PrixCoutant} + (\text{PrixCoutant} \times \text{Marge}) / 100$.

PRODUIRE est un trigger de la table *ProduitFini* et qui permet de lancer la production d'un produit dès que celui-ci à atteint ou est passé sous le seuil de production. Le processus de fabrication va consister à diminuer la quantité des matières première nécessaire à la création du produit. Pour cela le trigger liste les matières première dont la production à besoin pour créer le produit grâce à la table *Nomenclature*, puis à partir de cette liste, et pour chaque matière, il va soustraire les quantités nécessaire à sa fabrication.

DETERMINER_CLASSE_PF, notre gestion de la production étant gérée par la méthode ABC nous devons par conséquent définir la classe ('A', 'B' ou 'C') de chaque produit. Cette classe est déterminée par le chiffre d'affaire du produit et la quantité en production, avec les critères suivant :

- Chiffre d'affaire du produit supérieur ou égale à 80% et quantité en stock inférieur ou égale à 20%, alors il s'agira d'un produit de classe 'A'.
- Chiffre d'affaire du produit supérieur ou égale à 15% et quantité en stock inférieur ou égale à 30%, alors il s'agira d'un produit de classe 'B'.
- Chiffre d'affaire du produit supérieur ou égale à 5% et quantité en stock inférieur ou égale à 50%, alors il s'agira d'un produit de classe 'C'.

Le chiffre d'affaire du produit étant calculé à partir du chiffre d'affaire total (chiffre d'affaire total correspondant à la somme de toute les ventes) grâce à la multiplication du prix de vente par la quantité vendu et divisant ce nombre par le chiffre d'affaire total.

DETERMINER_CLASSE_MP, tout comme pour déterminer la classe du produit fini, nous avons ici la même méthode pour déterminer la classe des matières première à la différence que nous avons besoin non plus du chiffre d'affaire mais de la somme des dépenses engendré par la matière première.

CALCUL_QUANTITE_TRANS, afin de déterminer l'attribut "QuantiteTotal" de la table *Transaction* nous devons implémenté un trigger permettant de mettre à jour cette donnée. Pour cela il faut parcourir la table *ListePF* et sommer toute les quantités.

CALCUL_PRIXTOTAL_TRANS, pour déterminer l'attribut "PrixTotal" de la table *Transaction* nous devons cette fois-ci parcourir la liste de produits finis lié à la transaction (*ListePF*) et sommer le calcul "PrixDeVente*Quantité" par le nombre de produit de la transaction.

3.4 Procédure (Programme PL/SQL)

Les procédures que nous avons implémentées ont pour but d'obtenir des informations sur différentes tables. Voyons quelles sont ces procédures :

LISTE_MEILLEUR_CLIENT, est une procédure permettant de dresser la liste des dix clients ayant dépensé le plus par ordre décroissant de dépenses. Cette procédure permet de se rendre compte rapidement de quels sont les clients les plus intéressants pour l'entreprise ce qui pourrait par exemple engendrer des futures remises pour de tels clients.

LISTE_LISTE_MP, est une procédure qui permet de connaître la liste des matières premières selon le produit fini passé en paramètre.

LISTE_PROD_TRANS, nous permet d'obtenir la liste des produits d'une transaction passée avec un tiers. Cette liste permet de visualiser le contenu d'une transaction effectuée par un client, on pourra donc analyser les préférences des clients, etc.

3.5 Problèmes rencontrés

Nous avons rencontrés effectivement beaucoup de problèmes tout au long de la réalisation de ce projet. Tout d'abord la partie de modélisation nous a pris beaucoup de temps et mis à part la validation de nos diagrammes Use Case et de classe nous nous sommes sentis relativement seuls face à l'ampleur du projet d'autant plus qu'il s'agissait de modules¹⁵ totalement nouveaux et dont seul "l'expérience acquise en TD" nous a permis d'effectuer le travail de modélisation. Le deuxième plus gros problème que nous avons connu est la partie concernant l'implantation même du projet, ceci étant dû à la perte de notre connexion Internet¹⁶ durant les vacances de Noël, ce qui nous a empêché de travailler sur les machines du CRI. Nous pensions que le temps qui nous restait entre la rentrée et le jour de la présentation suffirait pour rattraper le retard mais nous nous sommes trompés. Nous n'avons donc pu mettre en place un jeu de test suffisant pour vérifier l'efficacité de nos choix d'implantation essentiellement en ce qui concerne nos triggers.

¹⁵Les modules de modélisation et de bases de données

¹⁶suite à un changement de fournisseur d'accès

4 Conclusion

Grâce à notre étude bibliographique puis notre dossier de modélisation, et enfin l'étude du passage au relationnel nous avons pu implanter notre projet de base de données. Nous voyons donc qu'il a fallu effectué beaucoup de recherches, rapport, et dossier pour pouvoir seulement commencé l'aspect technique du projet à savoir son implantation. On se rend rapidement compte que la cohérence de l'implantation dépend directement des différentes études effectuées auparavant et malheureusement seul la validation de nos diagrammes Use Case et de Classe nous permette de ne pas avancer à l'aveugle depuis notre étude bibliographique. C'est pourquoi même après avoir implanté notre projet, nous ne savons pas si nous avons suivit les bonnes directions d'implantation, s'il ne manque pas par exemple une grande notion qui aurait pu ou du être intégré afin de répondre correctement au sujet. Cette liberté dont nous pouvions nous attendre en troisième année de licence nous a malgré tout surpris, dotant plus qu'il comprenait des modules nouveaux (modélisation et base de données) et qu'il s'agissait d'implanter un système d'information qui nous était jusqu'alors inconnus puisque traitant de la gestion de production.

Malgré cet aspect nous avons cependant appris beaucoup de nouvelles choses grâce à ce projet. C'est ainsi que nous avons pu établir un passage vers le schéma relationnel à partir d'un diagramme de classe que nous avons nous même construit, puis la création du modèle physique de données à partir du schéma aura également été un exercice intéressant, nous avons ainsi pu vérifier que nous avions assimilé cette méthode.

Malheureusement pour nous, au delà de la création des tables, et des insertions, nous n'avons pas eu le temps de procédé a des tests suffisant pour garantir le bon fonctionnement de nos quatorze triggers/procédures. Nous ne pensions pas, en effet ces vérifications serait si complexe à mettre en oeuvre puisque celles-ci ne consiste plus, comme nous le fisions avec des langages de programmation, par une simple modifications du code puis une compilation, mais par des suppressions de données/de tables/tests d'insertions et enfin tests de selections pour être sûr du fonctionnement des programmes PL/SQL. Ce projet nous aura permis d'une manière générale d'abordé les bases de données non plus comme un simple langage supplémentaire mais bel et bien comme un système informatique à part entière et ayant une philosophie différentes de tous ce que nous avons pu voir jusque là.

5 Bibliographie

Ce rapport a été réalisé à partir des ouvrages [1, 2, 3, 4], des sites Internet [5, 6, 7, 8] et de notre étude bibliographique [9].

Références

- [1] UML 2 par la pratique. Pascal Roques. Editions Eyrolles. ISBN : 2-212-11480.
- [2] Vincent Giard . Gestion de la production et des flux. Collection, Gestion Production et technique qualitatives appliqués à la gestion. ISBN : 2717844988.
- [3] Pierre Zernati. Pratique de la gestion des stocks. Collection, Fonction de l'entreprise. ISBN : 2100079719.
- [4] André Boyé. Les fondamentaux de l'entreprise. Collection, Les indispensables de la gestion . ISBN : 2708131842.
- [5] Aes plus : la gestion des stocks. http://www.aesplus.net/gestion_stocks.htm.
- [6] Banque de ressources interactives SES. <http://brises.org>.
- [7] Gestion de stocks. <http://membres.lycos.fr/hconline/stocks.htm>.
- [8] Techniques de gestion. <http://www.sciences.ch/htmlfr/mathssociales/mathssgestion01.php>.
- [9] Notre étude bibliographique http://depinfo.u-bourgogne.fr/~am524741/MO/etude_biblio.pdf