REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITE MOHAMED BOUDIAF - M'SILA

FACULTE : Des mathématiques et De l'informatique
DEPARTEMENT D'informatique
N° :……..

DOMAINE : Mathématiques et informatique

FILIERE : Informatique

OPTION : Génie logiciel

Mémoire présenté pour l'obtention
Du diplôme de Master Académique

Par: Lebidi Abdessamad

Intitulé

La réalisation d'un outil graphique pour la transformation automatique d'un MCD vers une base de données relationnelle

Soutenu devant le jury composé de :

……	Université de M'sila	Président
Benazi Makhlouf	Université de M'sila	Rapporteur
……	Université de M'sila	Examinateur

Année universitaire : 2016 /2017

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITE MOHAMED BOUDIAF - M'SILA

FACULTE : Des mathématiques et De l'informatique
DEPARTEMENT D'informatique
N° :……..

DOMAINE : Mathématiques et informatique

FILIERE : Informatique

OPTION : Génie logiciel

Mémoire présenté pour l'obtention
Du diplôme de Master Académique

Par: Lebidi Abdessamad

Intitulé

La réalisation d'un outil graphique pour la transformation automatique d'un MCD vers une base de données relationnelle

Soutenu devant le jury composé de :

……	Université de M'sila	Président
Benazi Makhlouf	Université de M'sila	Rapporteur
……	Université de M'sila	Examinateur

Année universitaire : 2016 /2017

TABLE DES MATIERES

INTRODUCTION GENERALE

Dans un contexte d'accroissement exponentiel de la complexité des systèmes informatiques, la modélisation de ces systèmes est devenue un enjeu majeur dans la réussite des projets : une bonne prise en compte du besoin fonctionnel, la réduction des délais et des couts par la réutilisation des conceptions et des liens avec un code et enfin une souplesse nécessaire pour l'adaptation des applications aux différentes technologies actuelles ou futures. Et pour faire un modèle conceptuel d'un système complexe, le Modèle entité-association (E/A) représenté une bonne solution.

Le problème de génération automatique d'une base de données à partir d'un diagramme d'E/A a déjà été largement étudié et enseigné sur le plan théorique .souvent on se contente d'étudier les différents cas figures qui peuvent se présenter en termes de multiplicités des associations entre les entités, et de déterminer dans chaque cas le schéma relationnel adapté pour implanter ces associations.

La modélisation d'un problème, c'est-à-dire le passage du monde réel à sa représentation informatique, se définit en plusieurs étapes pour parvenir à son intégration dans un SGBD-R (Système de Gestion de Base de Données-Relationnel) et permettre la manipulation des données par le langage SQL (Structured Query Language).

Classiquement, le processus de modélisation des données passe par deux phases :

- Réalisation d'un modèle conceptuel.
- Traduction en un modèle relationnel.

Le premier niveau de modélisation, dit conceptuel, consiste en une phase d'analyse du problème réel. Cette phase est assez délicate et permet de définir les données à utiliser, leur mode d'évolution dans le temps et les relations entre elles. C'est le moment où l'on se pose les questions essentielles comme celle de savoir à quel usage on destine le modèle informatique que l'on est en train de constituer. Ce travail est réalisé par des spécialistes de l'analyse. Il s'exprime dans un formalisme de type entité-association. Il existe d'autres types de formalismes comme le formalisme UML ou Merise.

Le second niveau de modélisation, dit relationnel, conduit à élaborer l'ensemble des objets manipulables par un SGBD-R .Ce travail est souvent réalisé par l'architecte de données, ou un administrateur de SGBD. Il peut être découpé en deux étapes :

- La conception de modèle logique (représentation en tables indépendantes du SGBD)

- La traduction en un modèle physique (propre à un SGBD spécifique). Tous les SGBD n'ont pas les mêmes caractéristiques du langage SQL.

Nous proposerons des règles de traduction de chaque type des composants (Entité, Association) en structures de bases de données.

L'objectif principal de notre travail est de développer un outil graphique pour la création d'un modèle conceptuel de données et la création d'une base de données dans un SGBD prédéfini.

Cette méthode va faciliter la tâche et minimiser le temps de création, et ainsi l'élimination des erreurs.

Nous proposerons une procédure simplifiée qui conviendra pour traiter les problèmes abordés dans le mémoire. En particulier, nous ne nous préoccuperons pas de critères de performance, que nous laisserons aux professionnels.

Donc Qu'est-ce qu'une base de données? Que peut-on attendre de résultat de la conversion à partir d'E/A vers une base de données? C'est à ces questions, entre autres, que ce mémoire essaie d'apporter des réponses.

Pour pouvoir réaliser ce projet, nous avons répartis le travail en chapitres pour mettre en évidence les notions, les méthodes et les outils qui nous permettront d'aboutir à une application qui répond à nos spécifications, ainsi les chapitres sont : Généralité sur les bases des données, Analyse et Conception, Implémentation.

CHAPITRE 1
GENERALITE SUR LES BASES DES DONNEES

1 Introduction:

Le développement des techniques informatiques depuis ces dernières années a permis d'appliquer les outils informatiques dans l'organisation des entreprises. Vu, l'immense volume de données maniées par ces dernières.

Ce chapitre présente la première étape du processus de modélisation du monde réel, qui consiste à recueillir les informations puis à les transcrire sous une forme conduisant à un passage aisé au modèle relationnel. On utilise à cette fin le modèle entité-association, dont les concepts et la mise en œuvre sont présentés dans ce chapitre.

2 Méthode de Modélisation :

Une méthode de conception est définie comme une démarche d'organisation qui a pour objectif de résoudre un problème spécifique. La méthode de conception utilise un formalisme ou un langage pour exprimer le résultat.

2.1 Ingénierie Dirigée par les Modèles (IDM) :

L'ingénierie système est une démarche de réalisation des systèmes. Son objectif est de dénombrer et d'organiser les différentes activités du cycle de développement, afin de gérer la conception du système ainsi que sa complexité. Ses tâches s'occupent principalement de la mise en œuvre d'un système jusqu'à sa réalisation en passant par sa modélisation. [8]

2.2 Transformations de modèles :

L'IDM considère les opérations de transformations de modèles comme le moteur de développement, que ce soit pour l'analyse, l'optimisation ou la génération de code.

Cette procédure consiste à générer, à partir d'un ou de plusieurs modèles sources d'un système, un ou plusieurs modèles cibles du même système. Les modèles sont dans tous les cas conformes à leurs méta-modèles respectifs. Lorsque la transformation se fait dans un même formalisme elle est dite endogène. Dans l'autre cas, quand les deux méta-modèles source et cible sont différents, la transformation est dite *exogène*.

Ces transformations sont gouvernées par un ensemble de règles utilisées par le moteur de transformation. Ce moteur prend en entrée le modèle source, exécute les règles de transformation et génère le modèle cible.

2.3 Types de transformations de modèles :

Les transformations de modèles se partagent également en deux grandes classes : les transformations « *Modèle vers Code* », et les transformations « *Modèle vers Modèle* » largement étudiées dans l'approche ADM(Analyse et le Développement des Marchés). [17]

2.3.1 Transformation Modèle vers Modèle :

Ces transformations ont beaucoup évolué depuis l'apparition de l'ADM. Ce type de transformation permet la génération de plusieurs modèles intermédiaires avant d'atteindre le modèle de code, afin d'étudier les différentes vues du système, son optimisation, la vérification de ses propriétés et sa validation.

2.3.2 Transformation Modèle vers Code :

Il existe deux types de transformations « Modèle vers Code »: la première est basée sur le principe du visiteur (*visitor-based*) où l'on se sert du modèle et ses éléments qui rapprochent sa sémantique du langage de programmation pour obtenir le code ;la deuxième est basée sur le principe des patrons (*Template-based*) avec laquelle on accède aux informations du modèle source en utilisant les fragments de méta-code dans le code cible.

Il existe d'autres classifications basées sur les caractéristiques des langages de transformation des modèles utilisés.

2.4 Propriétés d'une transformation :

Une transformation est généralement caractérisée par l'ensemble de ses règles de transformation, leur ordonnancement, leur organisation, la relation entre les deux modèles source et cible, la traçabilité et la direction. [17]

a. Les règles de transformation.
b. Les spécifications.
c. La relation entre le modèle source et le modèle cible.
d. L'organisation des règles.
e. L'ordonnancement des règles.

3 Modélisation des Systèmes d'Information:

De nombreuses méthodes de développement ou d'analyse de logiciel ont été développées . Ces méthodes apportent à chaque fois des progrès considérables et des solutions à des problèmes ou des contraintes différentes. [6]

3.1 Méthode d'Etude et de Réalisation Informatique pour les Systèmes d'Entreprise(MERISE) :

Merise est un acronyme signifiant Méthode d'Étude et de Réalisation Informatique par les Sous-ensembles ou pour les Systèmes d'Entreprise.

La méthode Merise a comme objectif d'aider, de guider les SI (Système d'information), dans leurs phases d'analyses, de conception et le développement de l'applicatif.

La méthode Merise présente comme avantage indéniable de permettre une définition claire et précise de l'ensemble du Système d'Information et d'en définir correctement le périmètre.

3.2 Présentation générale de la méthode Merise :

La méthode Merise se caractérise par :

a. Une approche systémique en ayant une vue de l'entreprise en terme de systèmes.
b. Une séparation des données (le côté statique) et des traitements (le côté dynamique).
c. Une approche par niveaux.

3.3 Une approche par niveaux :

Pour la conception d'un SI, il est nécessaire de considérer quatre niveaux d'étude :

3.3.1 Le niveau conceptuel :

Le niveau conceptuel consiste à concevoir le SI en faisant abstraction de toutes les contraintes techniques ou organisationnelles et cela tant au niveau des données que des traitements.

Le niveau conceptuel répond à la question Quoi ? (le quoi faire, avec quelles données).

Le formalisme Merise employé sera :

a. Le Modèle Conceptuel des Données (MCD).
b. Le Modèle Conceptuel des Traitements (MCT).

3.3.2. Le niveau organisationnel :

Le niveau organisationnel a comme mission d'intégrer dans l 'analyse les critères liés à l'organisation étudiée. Le niveau organisationnel fera préciser les notions de

temporalité, de chronologie des opérations, d'unité de lieu, définira les postes de travail, l'accès aux bases de données…

Le formalisme Merise employé sera :

a. Le Modèle Organisationnel des Données (MOD).
b. Le Modèle Organisationnel des Traitements (MOT).

3.3.3 Le niveau logique :

Le niveau logique est indépendant du matériel informatique, des langages de programmation ou de gestion des données. C'est la réponse à la question Avec quoi ?

Le formalisme sera :

a. Le Modèle Logique des Données (MLD).
b. Le Modèle Logique des Traitements (MLT).

3.3.4 Le niveau physique :

Le niveau physique permet de définir l'organisation réelle (physique) des données. Il apporte les solutions techniques, par exemple sur les méthodes de stockage et d'accès à l'information. C'est la réponse au Comment ?

Le formalisme employé sera :

a. Le Modèle Physique des Données (MPD).
b. Le Modèle Opérationnel et physique des Traitements (MOPT).

Le tableau suivant résume la méthode Merise :

Niveaux	Données	Traitements
Conceptuel	Modèle Conceptuel des Données	Modèle Conceptuel des Traitements
Organisationnel	Modèle Organisationnel des Données	Modèle Organisationnel des Traitements
Logique	Modèle Logique des Données	Modèle Logique des traitements
Physique	Modèle Physique des Données	Modèle Opérationnel et Physique des Traitements

Table 1.1 : Tableau récapitulatif de la méthode MERISE.

4 Le modèle conceptuel des données :

Un modèle est une représentation simplifiée d'une réalité. Un modèle de données est une représentation abstraite des données d'un système d'information. Cette représentation est généralement exprimée à l'aide d'un langage graphique appelé Formalisme (Data model).
Un modèle conceptuel de données est une représentation des besoins en matière de données pour un système d'information. Il met en évidence les entités, leurs attributs, les associations et contraintes entre ces entités pour un domaine donné. Cette représentation, de nature sémantique, ne comporte aucune indication la structure de mémorisation des données associées aux entités. Le modèle conceptuel est généralement représenté à l'aide du Formalisme entité association (Conceptual model).[5]

4.1 Les composants de représentation graphique le Modèle conceptuel de données :

4.1.1 Entité **:** Objet concret ou abstrait du monde réel au sujet du quel une organisation est Susceptible de conserver des données (Entity).

Figure 1.1 : Entité

4.1.2 Attribut **:** Donnée élémentaire qui sert à caractériser une propriété des entités et des associations dans un modèle conceptuel de données (Attribute).
Identifiant : Attribut ou groupe d'attributs permettant d'identifier chaque occurrence d'une entité(Identifier).

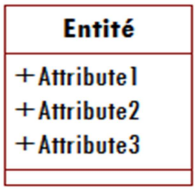

Figure 1.2 : Entité et Attribut

4.1.3 Association : Lien sémantique qui existe entre deux entités ou plus. Elle représente souvent la mémoire d'un événement qui a permis d'établir un lien logique entre ces entités (Relationship).

Figure 1.3 : Association entre deux entités.

4.3.4 Multiplicité : Contrainte inscrite à chaque extrémité d'une association binaire comportant un couple de valeurs (minimum−maximum) qui établit, pour chaque entité de l'association, les nombres minimum et maximum d'occurrences de l'autre entité qui peuvent lui être associées(Multiplicity).

Figure 1.4 : Multiplicité (1..n et 0..1) entre deux entités.

4.3.5 Contraintes entre les associations :

Les contraintes formulées dans un modèle conceptuel ne concernent pas seulement un attribut (contrainte sur le domaine) ou une association (contrainte de multiplicité) Dans certains cas exceptionnels une contrainte peut impliquer des associations On parle ici de contraintes inter-associations qui s'appliquent entre des occurrences de plusieurs associations partageant des entités.

4.2 Règles de normalisation :

Un bon schéma entités-associations doit répondre à 6 règles de normalisation, Les bonnes manières dans un schéma entités-associations **:**

- **Normalisation des entités (importante) :** toutes les entités qui sont remplaçables par une association doivent être remplacées.
- **Normalisation des noms :** le nom d'une entité**,** d'une association ou d'un attribut doit être unique.
- **Normalisation des identifiants :** chaque entité doit posséder un identifiant.
- **Normalisation des attributs (importante) :** remplacer les attributs en plusieurs exemplaires en une association supplémentaire de cardinalités maximales n et ne pas ajouter d'attribut calculable `à partir d'autres attributs.
- **Normalisation des attributs des associations (importante) :** les attributs d'une association doivent dépendre directement des identifiants de toutes les entités en association.
- **Normalisation des cardinalités :** une cardinalité minimale est toujours 0 ou 1 (et pas 2, 3 ou n) et une cardinalité maximale est toujours 1 ou n (et pas 2, 3, ...). [16]

5 Le modèle logique de données :

Un modèle de données découlant d'un modèle conceptuel mais qui le raffine pour tenir compte des caractéristiques du type de SGBD utilisé pour la réalisation de la BD (Logical Data model) [5].

5.1 Les composants le modèle logique de données :

- Relation : Une table comportant des lignes et des colonnes (Relation).
- Attribut : Une colonne nommée de la table représentant la relation (Attribute).
- Domaine : Ensemble des valeurs admises pour un attribut. Il établit les valeurs acceptables dans une colonne (Domain).
- Tuple : Une ligne dans une table (n-uplet).
- Clé primaire : Ensemble minimal de colonnes qui permet d'identifier de manière unique chaque tuple dans une table (Primary key).

- Clé étrangère : Une ou plusieurs colonnes dans une table qui a pour but d'assurer une liaison entre deux tables. On y arrive en dupliquant la clé primaire de la deuxième table dans la première. On l'appelle aussi clé externe (Foreign key)

5.2 Une représentation graphique du schéma de la BD :

Le concepteur d'une BD relationnelle doit élaborer ce qu'il est convenu d'appeler le schéma relationnel de la base de données Cette activité consiste à définir toutes les relations normalisées de la BD et les domaines de leurs attributs Théoriquement cela consisterait à décrire par intention chaque relation, définir les domaines de tous les attributs et pour chaque attribut d'une relation, établir à quel domaine il appartient Les relations du schéma doivent toutes posséder les propriétés suivantes :

- Une relation a un nom distinct de toutes les autres du même schéma.
- Chaque attribut d'une relation ne peut recevoir qu'une seule valeur atomique (type de données simple).
- Chaque attribut a un nom distinct.
- Les valeurs d'un attribut font toutes partie du même domaine : même type de données et même contraintes d'intégrité.
- Chaque tuple de la relation est distinct ; pas de tuple en double.
- L'ordre des tuples n'a pas d'importance.

6 Traduction d'un MCD vers un MLDR :

Pour traduire un MCD en un MLDR, il suffit d'appliquer cinq règles.

Notations : on dit qu'une association binaire (entre deux entités ou réflexive) est de type :

- 1 : 1 (un à un) si aucune des deux cardinalités maximales n'est n ;
- 1 : n (un à plusieurs) si une des deux cardinalités maximales est n ;
- n : m (plusieurs à plusieurs) si les deux cardinalités maximales sont n.

En fait, un schéma relationnel ne peut faire la différence entre 0,n et 1,n. Par contre, il peut la faire entre 0,1 et 1,1 (règles 2 et 4).

6.1 Règle 1 : toute entité devient une table dans laquelle les attributs deviennent les colonnes. L'identifiant de l'entité constitue alors la clé primaire de la table.

6.2 Règle 2 : une association binaire de type 1 : n disparaît, au profit d'une clé étrangère dans la table côté 0,1 ou 1,1 qui référence la clé primaire de l'autre table. Cette clé étrangère ne peut pas recevoir la valeur vide si la cardinalité est 1,1.

6.3 Règle 3 : une association binaire de type n : m devient une table supplémentaire (*parfois appelée table de jonction, table de jointure ou table d'association*) dont la clé primaire est composée de deux clés étrangères (*qui référencent les deux clés primaires des deux tables en association*). Les attributs de l'association deviennent des colonnes de cette nouvelle table.

6.4 Règle 4 : une association binaire de type 1 : 1 est traduite comme une association binaire de type 1 : n sauf que la clé étrangère se voit imposer une contrainte d'unicité en plus d'une éventuelle contrainte de non vacuité (*cette contrainte d'unicité impose à la colonne correspondante de ne prendre que des valeurs distinctes*).

6.5 Règle 5 : une association non binaire est traduite par une table supplémentaire dont la clé primaire est composée d'autant de clés étrangères que d'entités en association. Les attributs de l'association deviennent des colonnes de cette nouvelle table.

7 Les bases des données :

7.1 Notion de base de données :

Une base de données est un outil permettant de stocker et de retrouver l'intégralité de données brutes ou d'informations en rapport avec un thème ou une activité ; celles-ci peuvent être de natures différentes et plus ou moins reliées entre elles. Dans la très grande majorité des cas, ces informations sont très structurées, et la base est localisée dans un même lieu et sur un même support. Ce dernier est généralement informatisé.

La base de données est au centre des dispositifs informatiques de collecte, mise en forme, stockage, et utilisation d'informations. Le dispositif comporte un système de gestion de base de données (abréviation : SGBD) : un logiciel moteur qui manipule la base de données et dirige l'accès à son contenu. De tels dispositifs — souvent appelés base de données — comportent également des logiciels applicatifs, et un ensemble de règles relatives à l'accès et l'utilisation des informations. [2]

7.2 Utilisation d'une base de données :

La création d'une base de données recèle un but précis : elle doit permettre de retrouver de l'information par son contenu en se fondant sur des critères de recherche. La grande différence avec un programme écrit dans un langage de programmation est qu'une base de données doit pouvoir répondre à des questions pour lesquelles elle n'a pas forcément été prévue à la conception. [10]

7.3 Qualité d'une base de données :

L'un des objectifs de création d'une base de données est de pouvoir retrouver les données par leur contenu. Dans cette optique, il faut s'assurer que les données contenues dans la base sont de « bonne qualité ». Comment définir la qualité des données ?

De nombreux critères peuvent être pris en compte, on peut citer parmi les principaux :

a. La cohérence des données contenues dans la base.
b. L'absence de redondance.

7.4 Étapes de la conception des bases de données :

On peut décomposer le processus de conception d'une base de données en plusieurs étapes :

a. L'analyse du système du monde réel à modéliser.
b. La mise en forme du modèle pour l'intégrer dans un SGBD.
c. La création effective dans le SGBD des structures et leur remplissage.

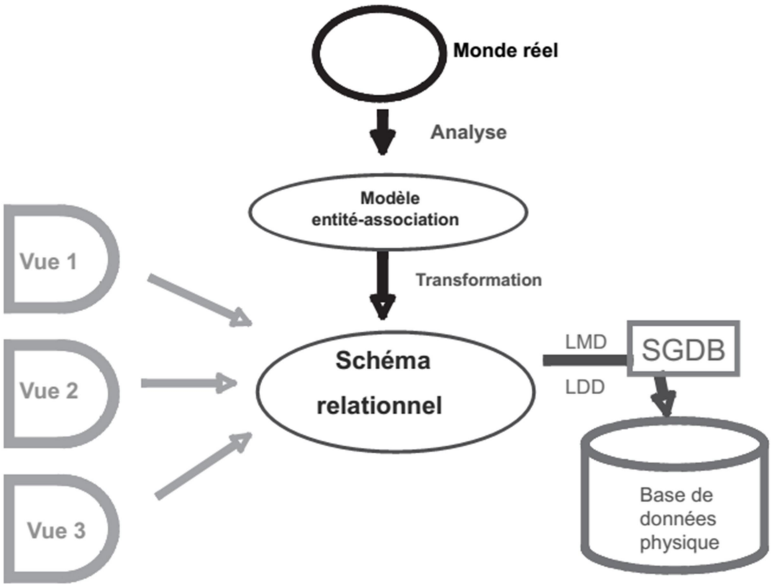

Figure 1.5 le processus de conception d'une base de données.

8 Langage SQL:

8.1 Définition :

SQL (sigle de Structured Query Language, en français langage de requête structurée) est un langage informatique normalisé servant à exploiter des bases de données relationnelles. La partie langage de manipulation des données de SQL permet de rechercher, d'ajouter, de modifier ou de supprimer des données dans les bases de données relationnelles. Outre le langage de manipulation des données, la partie langage de définition des données permet de créer et de modifier l'organisation des données dans la base de données, la partie langage de contrôle de transaction permet de commencer et de terminer des transactions, et la partie langage de contrôle des données permet d'autoriser ou d'interdire l'accès à certaines données à certaines personnes. Créé en 1974, normalisé depuis 1986, le langage est reconnu par la grande majorité des systèmes de gestion de bases de données relationnelles (abrégé SGBDR) du marché. [30]

8.2 Structure du langage:

SQL est un langage compose deux parties bien distinctes et, dans ces deux parties, de diverses subdivisions. La première de SQL est constituée de la partie déclarative du langage, c'est à dire d'ordres SQL que le SGBD-R doit exécuter .en d'autres termes, on spécifie ce que l'on veut obtenir ou faire, et c'est la machine qui décide comment elle doit l'exécuter.

La seconde partie est constituée d'un langage plus classique de type procédural dans lequel on retrouve les notions de fonction, méthodes, etc. [3]

8.2.1 SQL déclaratif :

La partie déclarative de SQL est elle-même subdivisée en quatre parties :

- le **DDL (***Data Definition Language*), c'est à dire ordres SQL permettant de créer (CREATE), modifier(ALTER) ou supprimer (DROP)les objets de la base
- le **DML** (*Data Manipulation Language*), c'est à dire ordres SQL permettant d'ajouter (INSERT), de modifier(UPDATE), de supprimer (DELETE), ou d'extraire des données (SELECT).
- Le **DCL** (*Data Control Language*), c'est à dire ordres SQL permettant de définir les privilèges afférents aux utilisateurs (GRANT, REVOKE).
- Enfin le **TCL** (*Transation Control Language*) permettant de gérer des transactions englobant des ordres des trois premières subdivisions.

8.2.2 SQL procédural :

- **PSM** (*Persistent Stored Module*) : concerne les fonctions, méthodes et déclencheurs (triggers) en tant qu'objets de la base (donc stockes dans la base et par conséquent persistants).
- **CLI** (*Call Level Interface*) :en fait des API destinées a piloter des objets encapsulant des données dans des langages hôtes par l'intermédiaire ,la plupart du temps d'un middleware(BDE , dB Express de Borland ,OBDC , ADO ,OLE DB de Microsoft, JDBC de java ,etc.).
- **Embedded SQL** : le lancement d'ordres SQL depuis un langage hôte et la récupération des données dans un programme via l'utilisateur de CURSOR

8.3 Le langage SQL DDL:

8.3. Création d'une base de données :

La création d'une base de données en SQL est possible en ligne de commande. Même si les systèmes de gestion de base de données (SGBD) sont souvent utilisés pour créer une base, il convient de connaître la commande à utiliser, qui est très simple. [27]

- Syntaxe :

Pour créer une base de données qui sera appelé « Ma_base_de_donnees » il suffit d'utiliser la requête suivante qui est très simple:

CREATE DATABASE *ma_base_de_donnees ;*

10.4 Création d'une table :

La commande CREATE TABLE permet de créer une table en SQL. Un tableau est une entité qui est contenu dans une base de données pour stocker des données ordonnées dans des colonnes. La création d'une table sert à définir les colonnes et le type de données qui seront contenus dans chacun des colonne (entier, chaîne de caractères, date, valeur binaire …)[27].

- Syntaxe :

La syntaxe générale pour créer une table est la suivante :

CREATE TABLE *nom_de_la_table*

(***colonne1*** *type_donnees*, ***colonne2*** *type_donnees*) ;

Dans cette requête, 2 colonnes ont été définies. Le mot-clé « *type_donnees* » sera à remplacer par un mot-clé pour définir le type de données (INT, DATE, TEXT …). Pour chaque colonne, il est également possible de définir des options telles que (liste non-exhaustive):

- **NOT NULL :** empêche d'enregistrer une valeur nulle pour une colonne.
- **DEFAULT :** attribuer une valeur par défaut si aucune donnée n'est indiquée pour cette colonne lors de l'ajout d'une ligne dans la table.
- **PRIMARY KEY :** indiquer si cette colonne est considérée comme clé primaire pour un index.
- **DOMAIN :** ces types de base abstraits et peu informatifs, il est possible de définir des domaines de valeurs qui rendront le schéma plus lisible et plus facile à modifier.
- **UNIQUE :** indiquer si cette colonne est considérée comme clé Expression d'un identifiant secondaire.
- **FOREIGN KEY :** On déclarera la clé étrangère et la table référencée , Syntaxe: **FOREIGN KEY** (*clé_nom_table*) **REFERENCES** *nom_table_référencée*

8.5 Les opérations sur Table :

8.5.1 Suppression d'une table: Toute table peut être supprimée. Elle est désormais inconnue et son contenu est perdu. [15]

- Syntaxe :

DROP TABLE *nom_de_la_table ;*

8.5.2 Ajout, retrait et modification d'une colonne : La commande suivante ajoute la colonne *colonne_1* à la table *table_1* :

- Syntaxe :

ALTER TABLE *table_1*

ADD COLUMN *colonne_1* **SMALLINT**

L'élimination d'une colonne d'une table s'effectue à l'aide d'une commande similaire :

- Syntaxe :

ALTER TABLE *table_1*

DROP COLUMN *colonne_1*

Il est également possible de modifier ou de supprimer un domaine (on modifie ici la valeur par défaut) :

- Syntaxe :

ALTER TABLE *table_1*

ALTER COLUMN *colonne_1* **set '***00***'**

Il est également possible de modifier ou supprimer un domaine :

- Syntaxe :

DROP DOMAIN *nom_domaine*

8.5.3 Ajout et retrait d'une contrainte : La demande d'ajout d'un identifiant sera refusée si les données que la table contient déjà violent cette propriété :

- Syntaxe :

ALTER TABLE *nom_table*

ADD PRIMARY KEY *(attribut_define_primaire*)

Il en va de même pour les identifiants secondaires déclarés via une clause unique:

- Syntaxe :

ALTER TABLE *table_1*

ADD UNIQUE (*colonne_1, colonne_2, colonne_3*)

Une colonne obligatoire peut être redéclarée facultative et inversement (si les données le permettent) :

- Syntaxe :

ALTER TABLE *table_1*

MODIFY *colonne_1* **NOT NULL**

ALTER TABLE *table_1*

MODIFY *colonne_2* **NULL**

8.5.4 Les structures physiques :

Les structures physiques être définies par des requêtes spécifiques. Nous décrirons brièvement le traitement des index et des espaces de stockage. Un index est créé par la commande **CREATE INDEX**. On spécifie son nom, la table à laquelle il est associé, les colonnes qui le composent, et l'ordre de rangement des valeurs dans l'index (**ASC**= ascendant, **DESC**= descendant). L'ordre par défaut est ascendant[15].

Syntaxe :

CREATE INDEX *nom_index*

ON *table_1* (*colonne_choix*)

8.6 Les Drivers qui supportent les requêtes SQL :

- **ECPG :** pré-compilateur permettant d'inclure des commandes SQL dans un source C
- **OLE DB :** API développée par Microsoft permettant l'accès aux données
- **Oracle Activity Report :** logiciel informatique utilitaire servant à mesurer les performances d'une base de données Oracle

- **Oracle Application Express :** environnement de développement intégré (EDI) permettant de créer des applications web en développement rapide et dont le but est d'exploiter des bases de données Oracle
- **Pro*C :** outil permettant d'inclure des commandes SQL dans un programme C
- **XA :** interface de communication entre un gestionnaire de ressource (par exemple une base SQL) et un gestionnaire de transaction
- **Xcalia Intermediation Core (XIC) :** plate-forme d'intermédiation permettant à une entreprise d'accéder à l'ensemble de ses données, de déployer des Applications Métier, à partir de briques composites

9 Notions fondamentales en matière de gestion de données :

9.1 Système de gestion de bases de données (SGBD) :

Un SGBD est un logiciel complexe qui permet de gérer et d'utiliser les données que l'on stocke en utilisant les modèles cités précédemment , le plus souvent produit par un éditeur commercial, qui gère et contrôle l'accès à une base de données, assurant ainsi une interface normalisée entre les applications et les bases de données(Data base management system)[10].
Les systèmes de gestion de base de données sont des logiciels universels, indépendants de l'usage qui est fait des bases de données. [14]

9.2 Caractéristiques des systèmes de gestion de bases de données (SGBD) :

On s'entend généralement pour dire qu'un logiciel de gestion de données pour qu'il puisse porter le nom de système de gestion de bases de données, doit posséder un certain nombre de caractéristiques fondamentales Nous les regroupons ici en six catégories:

a. Indépendance entre les données et les applications.
b. Contrôle centralisé des données pour éviter toute redondance.
c. Partage des données et accès concurrents.
d. Gestion de la cohérence et de l'intégrité des données.
e. Description des données stockées sous forme de métadonnées.
f. Gestion de la sécurité. [10]

9.3 Quelque systèmes de gestion de base de données :

Nous avons essayé de faire une vision sur quelques Systèmes de gestion de base de données les plus utilisées au monde, tout est classé sur le tableau ci-dessous [32] :

Nom SGBD	Année de création	éditeurs	Type logiciel	SQL
Firebird	1981	Firebird Foundation	serveur	✓
PostgreSQL	1985	Michael Stonebraker	serveur	✓
MS SQL Server	1989	Microsoft	serveur	✓
Oracle	1979	Oracle Corporation	serveur	✓
SQLite	2000	D. Richard Hipp	composant logiciel	✓
MS Access	1992	Microsoft	LGP4	✓
Sybase	1984	Bob Epstein	serveur	✓
MongoDB	2009	MongoDB	serveur	

Table 1.2 : les SGBD La plus utilisée à ce jour**.**

Le site DB-Engines, spécialisé dans le classement des moteurs de base de données., le DB-Engines Ranking utilise un certain nombre de paramètres. Il s'agit notamment de [20] :

- Le nombre de fois que le système est mentionné sur les sites web. Les requêtes doivent inclure le mot-clé « DataBase » ainsi que le nom du SGBD ;
- La fréquence des recherches dans Google Trends ;
- La fréquence des discussions techniques sur le système.
- Le nombre d'offres d'emploi dans lesquelles le système est mentionné sur Indeed et Simply Hired ;

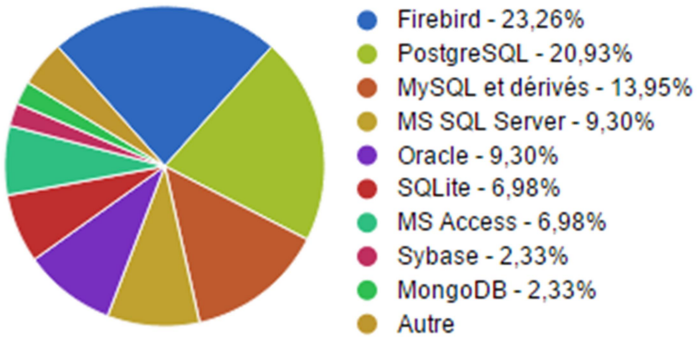

Figure 1.6 Cercle par rapport à différentes bases de données.

À partir d'une formule interne combinant ces paramètres, le BD-Engines Ranking a pu attribuer des scores aux différents systèmes de gestion de base de données. À l'issue du classement, Oracle se positionne comme le système le plus populaire de l'année 2017. Il est suivi par MySQL (2e), Microsoft SQL Server (3e), PostgreSQL (4e) et MongoDB (5e) [21].

322 systems in ranking, March 2017

Rank			DBMS	Database Model	Score		
Mar 2017	Feb 2017	Mar 2016			Mar 2017	Feb 2017	Mar 2016
1.	1.	1.	Oracle	Relational DBMS	1399.50	-4.33	-72.51
2.	2.	2.	MySQL	Relational DBMS	1376.07	-4.23	+28.36
3.	3.	3.	Microsoft SQL Server	Relational DBMS	1207.49	+4.04	+71.00
4.	4.	↑ 5.	PostgreSQL	Relational DBMS	357.64	+3.96	+58.01
5.	5.	↓ 4.	MongoDB	Document store	326.93	-8.57	+21.60
6.	6.	6.	DB2	Relational DBMS	184.91	-2.99	-3.02
7.	↑ 8.	7.	Microsoft Access	Relational DBMS	132.94	-0.45	-2.09
8.	↓ 7.	8.	Cassandra	Wide column store	129.19	-5.19	-1.14
9.	9.	↑ 10.	SQLite	Relational DBMS	116.19	+0.88	+10.42
10.	10.	↓ 9.	Redis	Key-value store	113.01	-1.03	+6.79

Figure 1.7 Top 10 des SGBD les plus populaires d'après DB-Engines.

9.4 Les types SGBD:

Nous divisons le SGBD en plusieurs groupes sont les suivants [32] :

Relationnel propriétaire	4D • DB2 •InterBase • MaxDB • Oracle Database • SQL Server • Sybase
Relationnel libre	Derby • Firebird • HSQLDB • Ingres • MariaDB • MySQL • PostgreSQL
Orienté objet	ZODB • db4o
Embarqué	BerkeleDB (Relationnel propriétaire) • SQLite (Relationnel libre)
NoSQL	HBase • LevelDB • MongoDB • Redis • RethinkDB • SimpleDB • Riak
SGBD intégrant un EDI	Access • FileMaker • HFSQL • OpenOffice.org Base • Paradox • Neo4j
Orienté séries temporelles	OpenTSDB • KairosDB

Table 1.3 : Les divisions des groupes de SGBD.

9.5 Logiciels de modélisation de la gestion des données :

Il existe de nombreux programmes et rappel les éléments suivants [28]:

- **AnalyseSI :** vous permet de modéliser votre base de données à l'aide de la méthode MERISE.
- **DBDesigner :** est une application Web qui vous permet de concevoir votre schéma de base de données sans écrire SQL.

- **Devaki-nextobjects :** est un projet sous license GNU/Public. C'est un IDE (Integrated Developpement Environment) permettant de réaliser les modèles de données de projets informatiques. NextObject peut générer les scripts pour créer les schémas en utilisant le dialecte SQL des principales bases de données
- **JMerise :** est un logiciel dédié à la modélisation des modèles conceptuels de donnée pour Merise
- **Mocodo :** programme qui génère des diagrammes entités-associations, ou Modèles Conceptuels de Données (MCD), ainsi que des schémas relationnels, ou Modèles Logiques de Données (MLD), et des tables SQL
- **MPD Designer :** programme de modélisation physique de données, permettant de générer les scripts SQL correspondant au MPD aussi bien pour MySQL 5.0 que pour SQL Server 2005 et Oracle 10g
- **Open Model Sphere :** outil de génie logiciel permettant la modélisation relationnelle de données, la modélisation des processus d'affaires, et la modélisation UML
- **PG Modeler :** logiciel de modélisation de base de données PostgreSQL

10 Conclusion :

On a vu dans ce chapitre les caractéristiques des bases des données on générale et de la façon sa réalisation, en suite on abordera la méthode de modélisation UML par quelque notions générales, l'on procédera à la réalisation d'une plus importante étape les diagrammes.

CHAPITRE 2
ANALYSE ET CONCEPTION

1 Introduction:

L'étape de l'analyse et la conception est la phase la plus importante pour établir un système efficace. Dans ce chapitre, Nous avons choisi UML, qui propose la conception basée sur la réalité, nous nous appuierons sur le diagramme de cas d'utilisation, puis nous nous traduirons immédiatement le diagramme de classe et de séquence, nous avons trois diagrammes pour atteindre à l'équivalent de la conception de notre travail.

2 Présentation d'UML:

Le Langage de modélisation unifié se définit comme un langage de modélisation graphique et textuel destiné à comprendre et à décrire des besoins, spécifier et documenter des systèmes, esquisser des architectures logicielles, concevoir des solutions et communiquer des points de vue. **[**11]

2.1 UML 2.0:

2.2 Les Diagrammes d'UML 2.0:

UML dans sa version 2 propose treize diagrammes qui peuvent être utilisés dans la description d'un système.

Ces diagrammes sont regroupés dans deux grands ensembles.

2.2.1. Les diagrammes structurels:

Ces diagrammes au nombre de six, ont vocation à représenter l'aspect statique d'un système :

- Diagramme de classe.
- Diagramme d'objet.
- Diagramme de composant (modifié dans UML 2).
- Diagramme de déploiement (modifié dans UML 2).
- Diagramme de paquetage (nouveau dans UML 2).
- Diagramme de structure composite (nouveau dans UML 2).

2.2.2. Les diagrammes de comportement :

Ces diagrammes représentent la partie dynamique d'un système réagissant aux événements et permettant de produire les résultats attendus par les utilisateurs. Quatre diagrammes sont proposés par UML.

- Diagramme des cas d'utilisation.
- Diagramme d'état-transition (machine d'état).
- Diagramme d'activités (modifié dans UML 2).
- Diagramme de séquence (modifié dans UML 2).

3 Modélisation d'application Orient Object :

Pour modéliser notre application on utilise UML et les diagrammes suivants:

1. Diagramme de cas d'utilisation.
2. Diagramme de classes.
3. Diagramme de séquence.

4 Spécification des besoins :

4.1Identification des cas d'utilisation :

Les cas d'utilisation sont des outils formels qui permettent de consigner et d'exprimer les interactions et les dialogues entre le système et L'utilisateur « l'acteur ».l'ensemble des use cases est une description détaillé du comportement de système aux sollicitations de ces derniers, un cas d'utilisation doit exprimer ce que le système doit faire sans préjuger de la façon dont cela sera fait.

- **Utilisateur :** L'utilisateur fait la conception d'un modèle de données afin de mettre en place une base de données.

4.2. Tableau montrant la relation entre l'acteur et les cas d'utilisation: Ce tableau indique la liaison qui existe entre l'acteur et les cas d'utilisation correspondants, Comme suit:

Acteur	Cas d'utilisation	Décomposition en actions
Utilisateur	C1- Ouvrir un projet	S01-Modifier un Projet S02-Enregistrer un Projet
	C2-Créer un graphe E/A	S03-Gérer les entités S04-Gérer les attributs
	C3-Transfèrer un graphe	S05-Choisir le type du SGBD
	C4-Gérer les associations	S06-Créer une association S07-Modifier une association S08-Supprimer une association
	C5-Gérer les entités	S09-Créer une entité S10-Modifier le nom d'une entité S11-Supprimer une entité
	C6-Insérer un lien	S12-Changer la cardinalité
	C7-Gérer les Attributs	S13- Ajouter un attribut S14-Modifier l'Attribut S15-Supprimer l'Attribut S16-Choisir un Clé Primaire

Table. 2.1: Tableau montrant la relation entre l'utilisateur et les cas d'utilisation.

5 Diagramme de cas d'utilisation générale :

Le diagramme de cas d'utilisation permet de représenter visuellement une séquence d'actions réalisées par un système, représenté par une boîte rectangulaire, appelé *acteur principal*, et ceci indépendamment de son fonctionnement interne. [18]

5.1 Représentation le diagramme de cas d'utilisation générale :

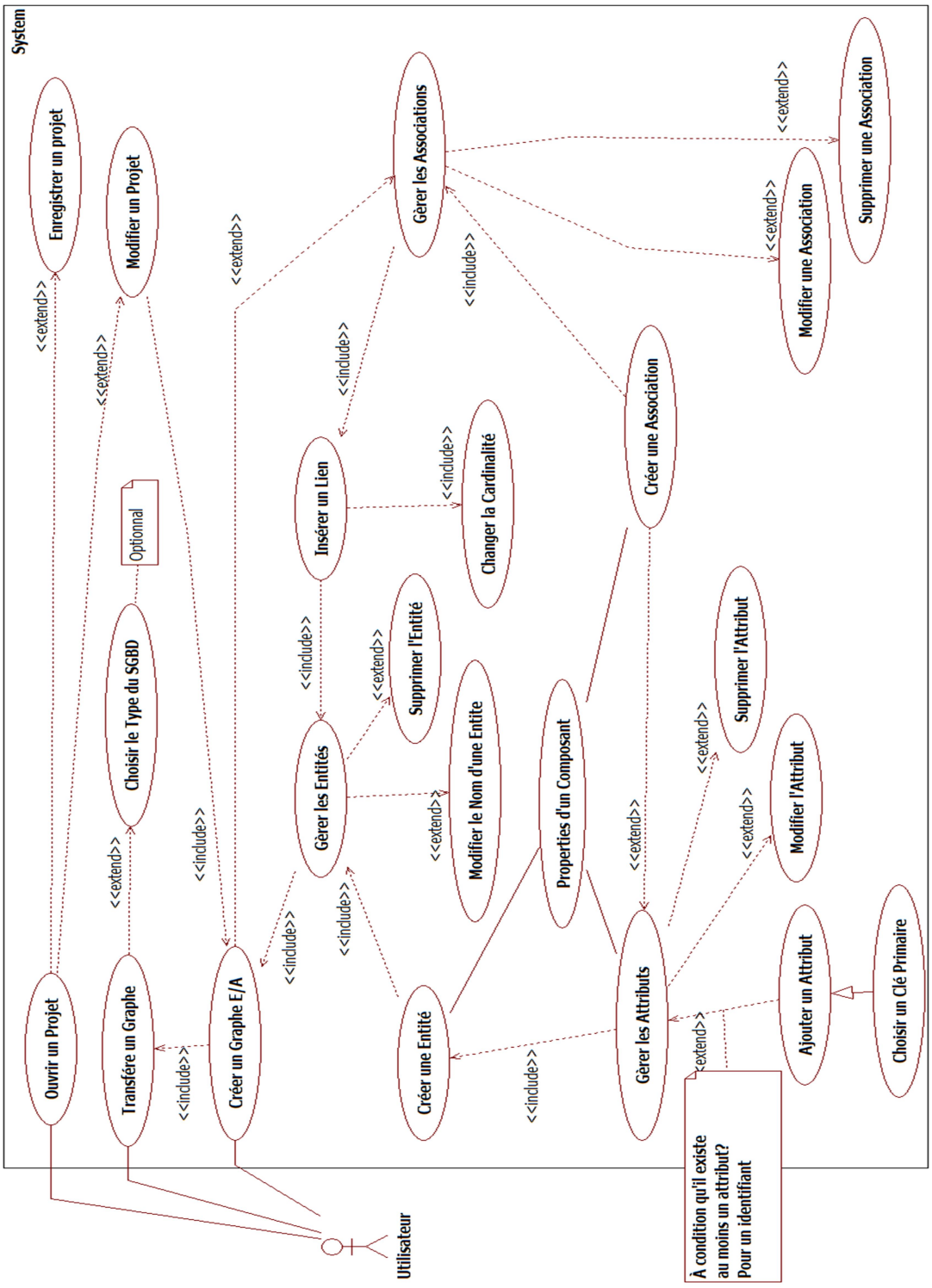

Figure.2.1: Le diagramme de cas d'utilisation générale.

5.1.1. Explication du diagramme de cas d'utilisation générale :

- Le diagramme de cas d'utilisation précédent (**Figure.2.1**), explique l'interaction entre les utilisateurs et le système.
- Notre système contient plusieurs cas comme le tableau précédant (**Table. 2.1**).
- Au début l'utilisateur ouvre un projet et choisi entre l'enregistrement ou la modification d'un autre projet ce qui résulte un nouveau projet.
- Le résultat de transformation du graphe est un fichier qui contient des scripts SQL.
- Le système nous donne aussi le choix du SGBD.
- L'ajout d'une entité dans le graphe **E/A** a des conditions : définir le nom de l'entité et la clé primaire.
- L'utilisateur peut ajouter les attributs de chaque entité.
- L'ajout de l'association nous exige la détermination de son nom.
- Après la mise de l'association l'utilisateur doit préciser la cardinalité de la relation, et les entités des extrémités.
- Finalement le système vérifie la validité du graphe.

5.1.2 Description textuelle des cas d'utilisation :

5.1.2.1. *Description textuelle de cas d'utilisation « Créer un Graphe E/A».*

Nom	Créer un Graphe E/A.
Objectif	Création d'une série des modèles, qui permettent de représenter le graphe E/A.
Pré-condition	Ouvrir un nouveau projet.
Acteur principal	Utilisateur.
Scénario nominale	1- L'utilisateur ouvre un nouveau projet. 2-Le système affiche la liste des outils et l'espace de dessin du graphe.
Post-condition	a- L'espace du dessin est prêt à utiliser.

Table.2.3: Cas d'utilisation «Créer un Graphe E/A».

5.1.2.2. *Description textuelle de cas d'utilisation « Transférer un Graphe».*

Nom	Transférer un Graphe E/A.
Objectif	Convertir un graphe E/A en Script SQL.
Pré-condition	Le graphe E/A existé est valide.
Acteur principal	Utilisateur.
Scénario nominale	1-L'utilisateur choisit le lien Transformation du Graphe à partir des options de l'outil graphique. 2-Le système affiche un message de confirmation**.** 3-L'utilisateur clique sur le choix « Choisir type SGBD ». 4-Le système vérifie la validité des informations envoyées.
Scénarios alternatifs	5-Le système renvoie un message d'erreur.
Post-condition	a- Transformation est réussite.

Table.2.4: Cas d'utilisation «Transfère un Graphe E/A».

5.1.2.3. *Description textuelle de cas d'utilisation « Modifier un Project ».*

Nom	Modifier un Project
Objectif	Ouvrir un Projet par utilisateur pour faire une modification.
Pré-condition	Accéder à la barre menue du projet.
Acteur principal	Utilisateur.
Scénario nominale	1-L'utilisateur choisit « Modifier » à partir de la barre menue. 2-Le système affiche une boite de dialogue pour choisir le fichier. 3-L'utilisateur choisit le fichier et l'ouvrir. 4-Le système affiche le graphe E/A dans le mode création.
Scénarios alternatifs	5-Le système renvoie un message d'erreur, Scénario 3 invalide.
Post-condition	a- Ouverture du graphe dans l'espace du dessin.

Table.2.5: Cas d'utilisation «Modifier un Project ».

5.1.2.4. *Description textuelle de cas d'utilisation « Modifier l'Attribut».*

Nom	Modifier une Attribut.
Objectif	Changer le nom et le type d'attribut qui a été choisi par l'utilisateur.
Pré-condition	a-Mode création de graphe E/A. b-Cet attribut existe dans une entité bien définie.
Acteur principal	Utilisateur.
Scénario nominale	1-L'utilisateur choisit l'entité dans le graphe et choisit l'attribut à modifier. 2- Le système affiche la boite de dialogue « modifier » 3-L'utilisateur remplit les champs et les valide. 4-Le système vérifie la validité des informations envoyées.
Scénarios alternatifs	5-Le système renvoie un message d'erreur. , Scénario 3 invalide.
Post-condition	a-Modification de l'attribut est réussite. b-Le graphe est mis à jour.

Table.2.6: Cas d'utilisation «Modifier l'Attribut».

5.1.2.5. *Description textuelle de cas d'utilisation « Créer une entité».*

Nom	Créer une entité.
Objectif	Création d'une entité dans le graphe.
Pré-condition	a-Mode création de graphe E/A. b-Demande de la création de l'entité
Acteur principal	Utilisateur.
Scénario nominale	1- L'utilisateur choisit« Création entité » à partir de la liste des outils. 2- Le système affiche le message de dialogue « ajouter nom entité ». 3-L'utilisateur remplit le champ et le valide. 4-Le système affiche le composant entité et définie leur nom. 5-L'utilisateur confirme les données dans l'entité. 6- Le système vérifie la validité des informations envoyées.
Scénarios alternatifs	7-Le système renvoie un message d'erreur, Scénario 3 invalide.
Post-condition	a. L'ajout de l'entité est réussi. b. Le graphe est mis à jour et le composant « entité » est ajoutée.

Table.2.7 : Cas d'utilisation « Créer une entité».

5.1.2.6. *Description textuelle de cas d'utilisation « Supprimer une Association».*

Nom	Supprimer une Association.
Objectif	Supprimer Association dans le graphe E/A entre deux entités.
Pré-condition	a- Mode création de graphe E/A. b-Demande de suppression.
Acteur principal	Utilisateur.
Scénario nominale	1- L'utilisateur sélectionne un composant Association dans le graphe et clique sur le lien « Supprimer une Association». 2-Le système affiche un message de confirmation. 3- L'utilisateur choisi et valide l'opération. 4- Le système vérifie la validité des informations envoyées.
Post-condition	a-Suppression est réussie. b -Le graphe est mis à jour et la suppression de l'association est effectuée.

Table.2.8 : Cas d'utilisation « Supprimer une Association».

6 Diagramme de classes :

Les classes, les associations et les attributs sont les concepts UML fondamentaux pour l'analyse orientée objet. Nous Apprendrons à identifier les concepts du domaine à partir de l'expression initiale des besoins de notre étude de cas. Nous Verrons comment ajouter des attributs et des associations à ces Concepts ainsi que les représentations graphiques UML Associées. [13]

6.1 Diagramme de classes :

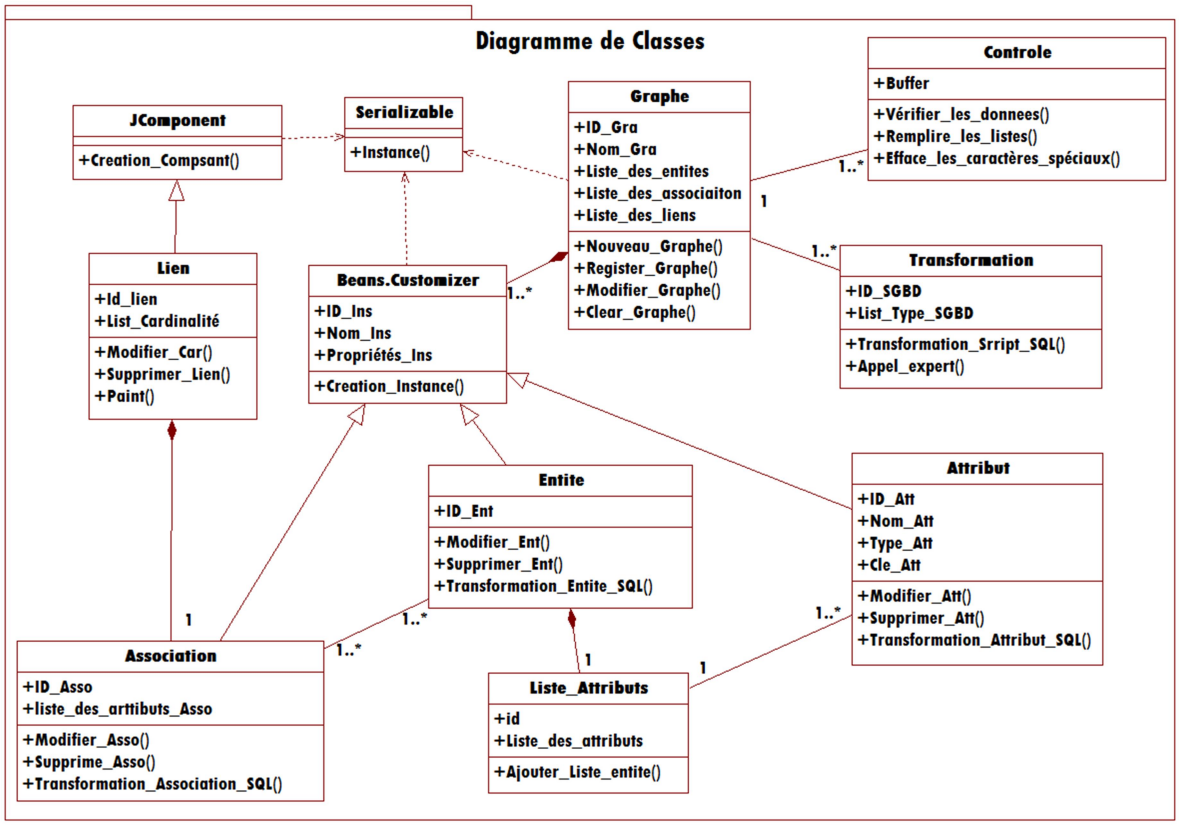

Figure 2.5 : Diagramme de classes.

6.1.1 Explication de diagramme de classes:

Ce diagramme affiche de nombreuses fonctionnalités et spécifications tout au long de notre travail. Ils Sont examinés par les besoins potentiels que nous avons identifiés dans ce chapitre et la conclusion de la relation entre les éléments du système, connus sous le nom d'une entité et une association.

Les quatre éléments de base identifiés dans le graphe E/A sont: Entité - Attribut- Association (Hérité en class **beans.customizer**) - Cardinalité (Hérité en class **jComposant** parceque il contient une fonction **Paint**()). En suite ajouter un attribut comme composant dans une entité.

Nous avons utilisé les listes afin d'enregistrer les données de chaque élément dans le graphe E/A. Pour classe de contrôler, afin d'alerter l'utilisateur, et la classe de transformation permet la conversion un graphe E/A à un Script SQL ou directement exécuté cette fichier dans le système de gestion de base de données spécifique.

7 Diagrammes de séquence système:

Les cas d'utilisation décrivent les interactions d'acteur avec application que nous voulons concevoir. Lors de ces interactions, l'acteur produit des messages qui affectent le système informatique. Nous allons isoler ces messages et les représenter graphiquement sur des diagrammes de séquence UML. [29]

7.1. Les diagrammes de séquence conception:

7.1.1. Diagramme de séquence « Créer une entité» :

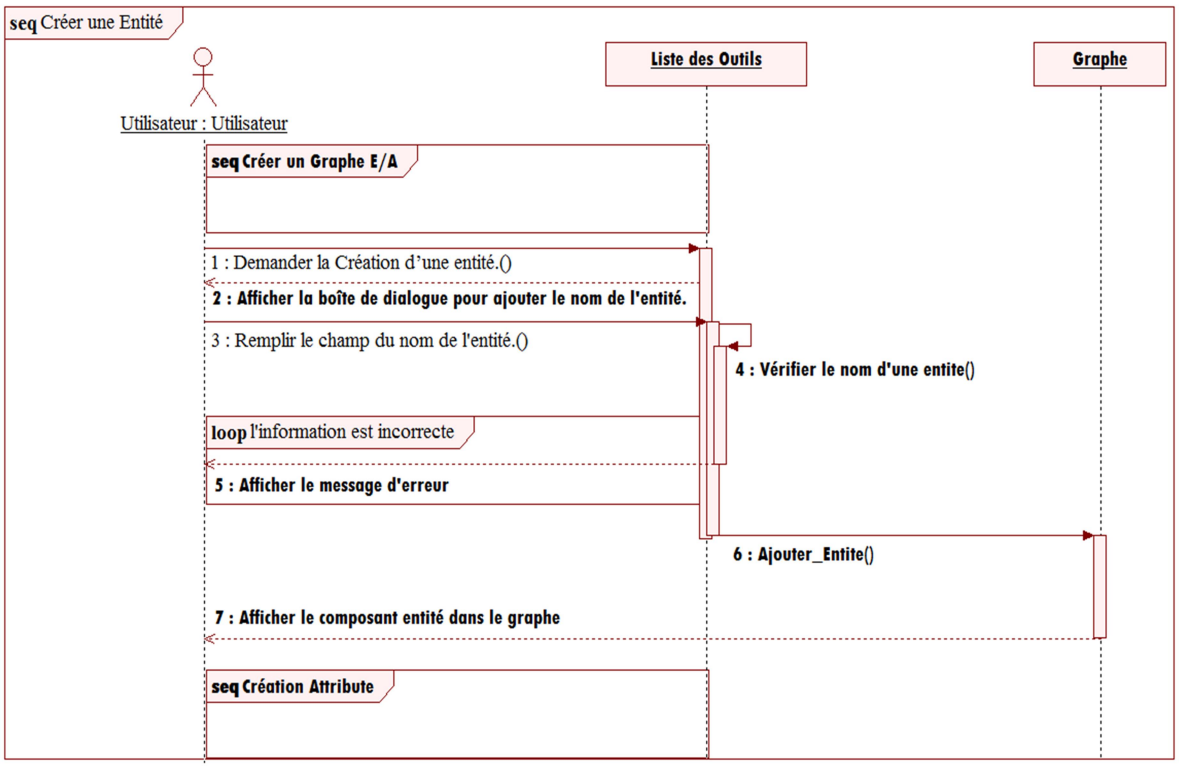

Figure.2.2: Diagramme de séquence « Créer une entité».

7.1.1.1 Explication diagramme de séquence « Créer une entité» :

- Le séquencement des processus passe par certaines étapes, nous identifierons ici le dialogue entre l'utilisateur et le système au cours de la création d'une entité.
- L'utilisateur demande à insérer une entité, en tenant compte d'un graphe E/A dans le cas d'un changement ou d'un nouveau projet.
- Le système affiche une boîte de dialogue demandant le nom de l'entité.
- L'utilisateur ajoute le nom de l'entité.
- Le système vérifie le nom du l'entité est incorrect. Il envoie un message d'erreur pour informer l'utilisateur.

- Si le nom du l'entité est correct, Le système affiche le composant entité dans le graphe E/A et définie leur nom. Ce qui permet de créer un attribut.

7.1.1.2 Les fonctions utilisées dans ce diagramme:

- 4 : Verifier_nom_entite() ;
- 6 : Ajouter_entité() ;

7.1.1.3 Les messages affichés par le système:

- 2 : Affiche une boîte de dialogue demandant le nom de l'entité
- 3 : Afficher un message d'erreur.
- 5 : Afficher le composant entité dans le graphe.

7.1.2. Diagramme de séquence de «Créer un Graphe E/A» :

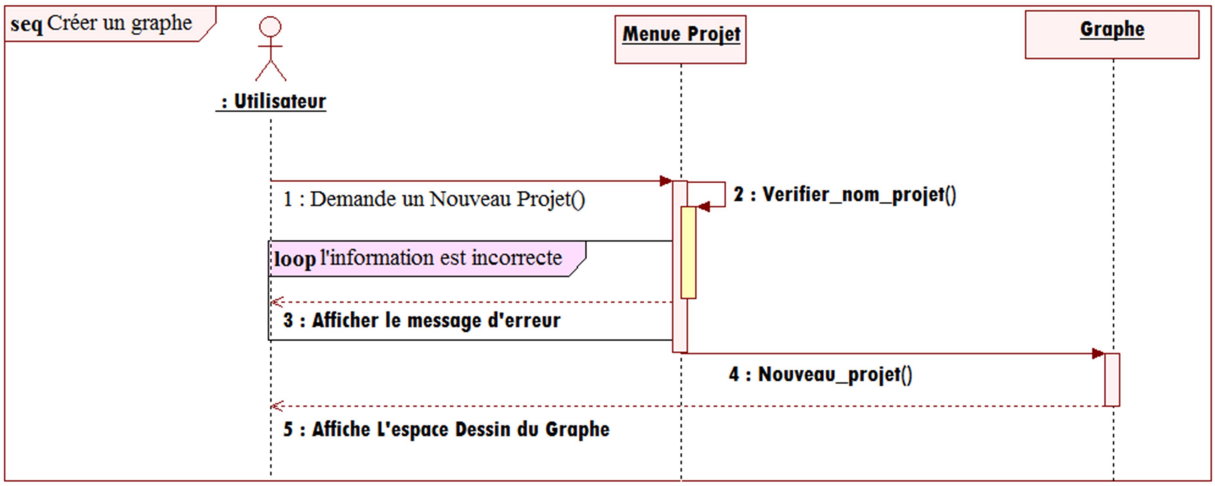

Figure.2.3: Diagramme de séquence de « Créer un Graphe E/A».

7.1.2.1 Explication diagramme de séquence «Créer un Graphe E/A» :

- L'utilisateur demande de créer un nouveau projet.
- Le système vérifie le nom du projet est incorrect. Il envoie un message d'erreur pour informer l'utilisateur.
- Si le nom du projet est correct, le système ouvre l'espace de dessin et permet à l'utilisateur d'ajouter plusieurs composants.
- La base de données prend le nom du projet.

7.1.2.2 Les fonctions utilisées dans ce diagramme:

- 2 :Verifier_nom_projet() ;
- 4 : Afficher_l'espace_dessin_du_graphe() ;

7.1.2.3 Les messages affichés par le système:

- 3 : Afficher un message d'erreur.
- 5 : Afficher l'espace de dessin.

7.1.3. Diagramme de séquence de «Transférer un Graphe E/A» :

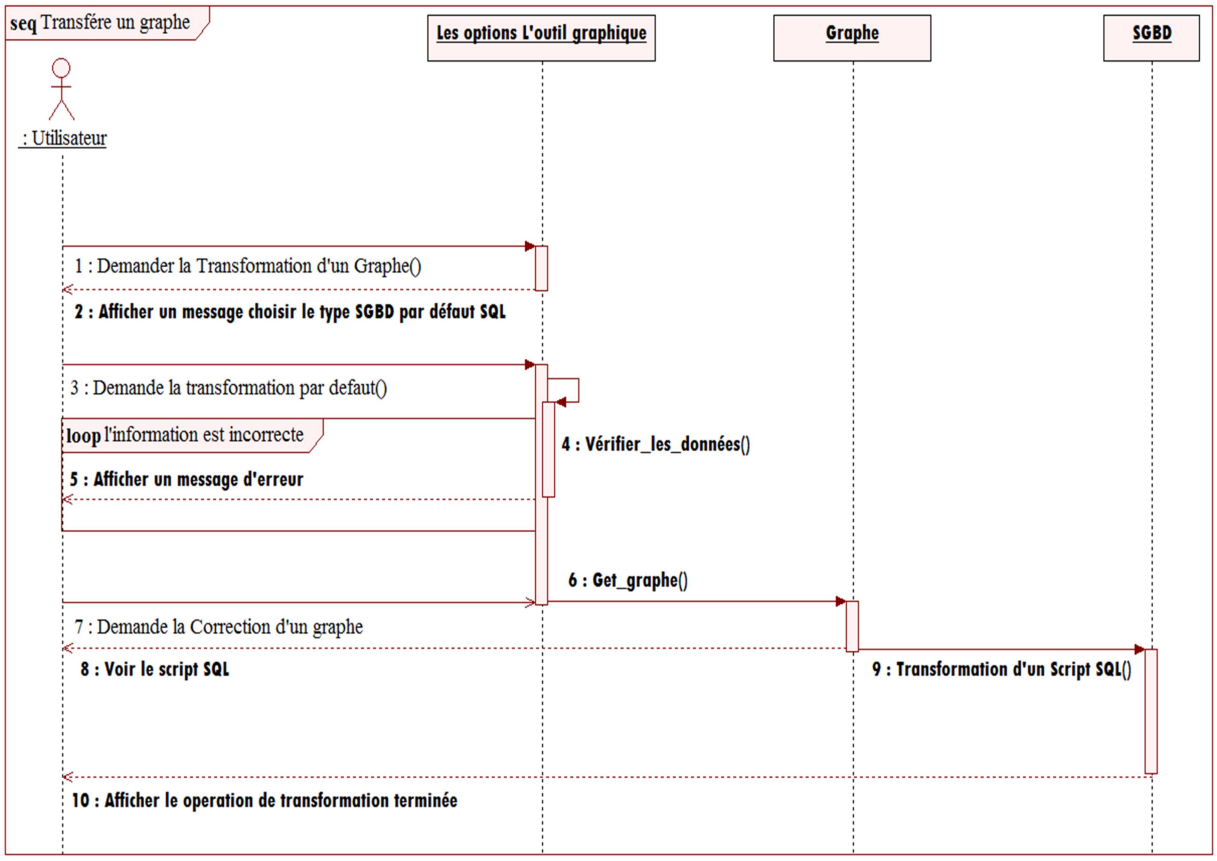

Figure.2.4: Diagramme de séquence de « Transfère un Graphe E/A».

7.1.3.1 Explication diagramme de séquence «Transférer un Graphe E/A» :

- L'utilisateur demande la conversion du graphe.
- Le système envoie un message de confirmation que le processus de conversion à la forme par défaut d'un script SQL, et lui demande si il veut choisir un autre SGBD.
- L'utilisateur peut voir si le graphe est correct ou non par la demande de vérification.

- Ensuite le système envoie une demande à l'SGBD et créer une base de données en utilisant une connexion de serveur dédié au travail et exécuter le fichier script SQL.

7.1.3.2 Les fonctions utilisées dans ce diagramme:

- 4 :Verifier_les_donnees() ;
- 6 : Get_graphe() ;
- 9 : Transformation d'un script SQL

7.1.3.3 Les messages affichés par le système:

- 3 : Afficher un message choisir le type SGBD par défaut SQL.
- 5 : Afficher un message d'erreur
- 8 : Voir le script SQL.
- 10 : Afficher l'opération de transformation terminée

8 Conclusion :

Dans ce chapitre, nous avons proposé une démarche de modélisation pour développer notre application, en basant sur la méthode UML, Nous avons commencé par la spécification des besoins et les divers cas d'utilisation, puis la conception des diagrammes de classes, et en phase de conception nous avons élaboré le diagramme de séquence.

Dans le prochain chapitre, nous défini les outils et les langages de programmation qui vont nous aider à mettre en œuvre notre application, ce qui sera l'objet du chapitre suivant.

CHAPITRE 3
IMPLEMENTATION

1. Introduction :

Ce chapitre couvre la création et la mise en œuvre des différents programmes, qui servent à la constitution de notre application et de ses fonctionnalités.

Nous décrivons l'environnement de création du système, ensuite nous présenterons quelques interfaces résultantes.

2. Présentation des outils de développement:

2.1 NetBeans:

Est un environnement de développement intégré (EDI), placé en open source par Sun en juin 2000 sous licence CDDL (Common Development and Distribution License). En plus de Java. [31].

L'Éditeur de source NetBeans n'est pas seulement un mécanisme pour écrire du texte, mais aussi un environnement complet conçu pour vous aider. Qu'il s'agisse d'abréviations pour un codage plus rapide, l'achèvement automatique du code ou les aides à la navigation et à la documentation, l'éditeur source a pour but de fournir toute la commodité possible. [1]

Figure 3.1:NetBeans.

2.2 Environnement de base :

L'environnement de base comprend les fonctions générales suivantes:

- configuration et gestion de l'interface graphique des utilisateurs.
- support de différents langages de programmation.
- traitement du code source (édition, navigation, formatage, inspection..),
- fonctions d'import/export depuis et vers d'autres IDE, tels qu'Eclipse ou JBuilder,
- accès et gestion de bases de données, serveurs Web, ressources partagées,
- gestion de tâches (à faire, suivi …),documentation intégrée. [25]

2.3 Le langage Java

Java est un langage de programmation et une plate-forme informatique créée par Sun Microsystems en 1995. Il s'agit de la technologie sous-jacente qui permet l'exécution de programmes modernes et performants, notamment dans la construction des utilitaires, des jeux et des applications professionnelles. [22]

Les caractéristiques du Java :

- Java est portable et simple.
- Java est orienté objet.
- Java est fortement typée.
- Java assure la gestion de la mémoire.
- Java est sûre et économique.
- Java est multitâche. [19]

3 Les Composants graphiques de Java (JComponent) :

3.1 Aperçu général :

- Les programmes à interfaces graphiques font usage des classes AWT (abstract windowing toolkit)et/ou swing.
- Ils sont dirigés par évènements.
- Classe de base des AWT: la classe abstraite Component.
- Classe de base des composants swing: JComponent.
- Swing offre une palette bien plus large. [9]

3.2 Contexte graphique:

L'outil de dessin est le contexte graphique, objet de la classe Graphics,Il encapsule l'information nécessaire, sous forme d'état graphique, Celui-ci comporte :

- la zone de dessin (le composant), pour les méthodes draw*() et fill*()
- une éventuelle translation d'origine
- le rectangle de découpe (clipping)
- la couleur courante et la fonte courante
- l'opération de dessin (simple ou xor)
- la couleur du xor, s'il y a lieu. [4]

4 Le Driver JDBC (Java Database Connectivity) :

est une API de niveau SQL qui permet de passer les instructions SQL dans les arguments des interfaces JDBC. Pour que cela puisse se faire de façon indépendante de la base de données, JDBC impose que les fournisseurs de base de données offrent une implémentation de ses interfaces . [23]

- Permet de dialoguer avec un DBMS (DataBase Management System).
- Au travers d'une API(Application Programming Inteface).
- Grâce au langage SQL.

4.1 Fonctionnement de JDBC :

a. Établir une connexion avec la base de données (Driver_Manager).
b. Construire et exécuter une requête (Statement ou Prepared_Statement).
c. Traiter le résultat (Result_Set). [23]

Connexion à la base de données - URL et Connexion - :

- On réalise la connexion au travers d'une URL : **jdbc**:nom du protocole:**URL BD**
- Des exceptions sont levées en cas de problème : SQLException, ClassNotFound

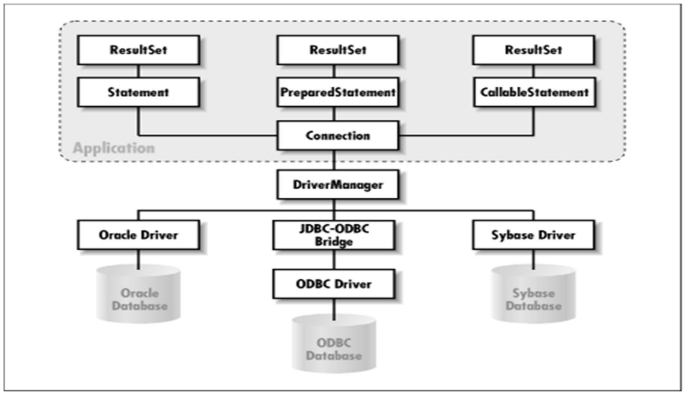

Figure 3.2 : Architecture d'un JDBC.

4.2 Cycle JDBC:

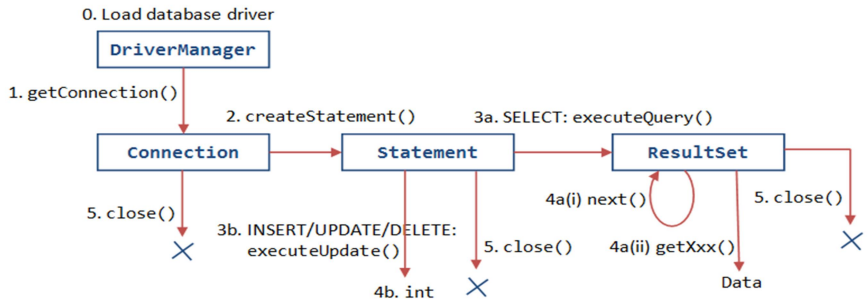

Figure 3.3 : Cycle JDBC. [15]

5. La présentation de certaines interfaces de notre outil graphique:

5.1. Mode de création d'un graphe E / A:

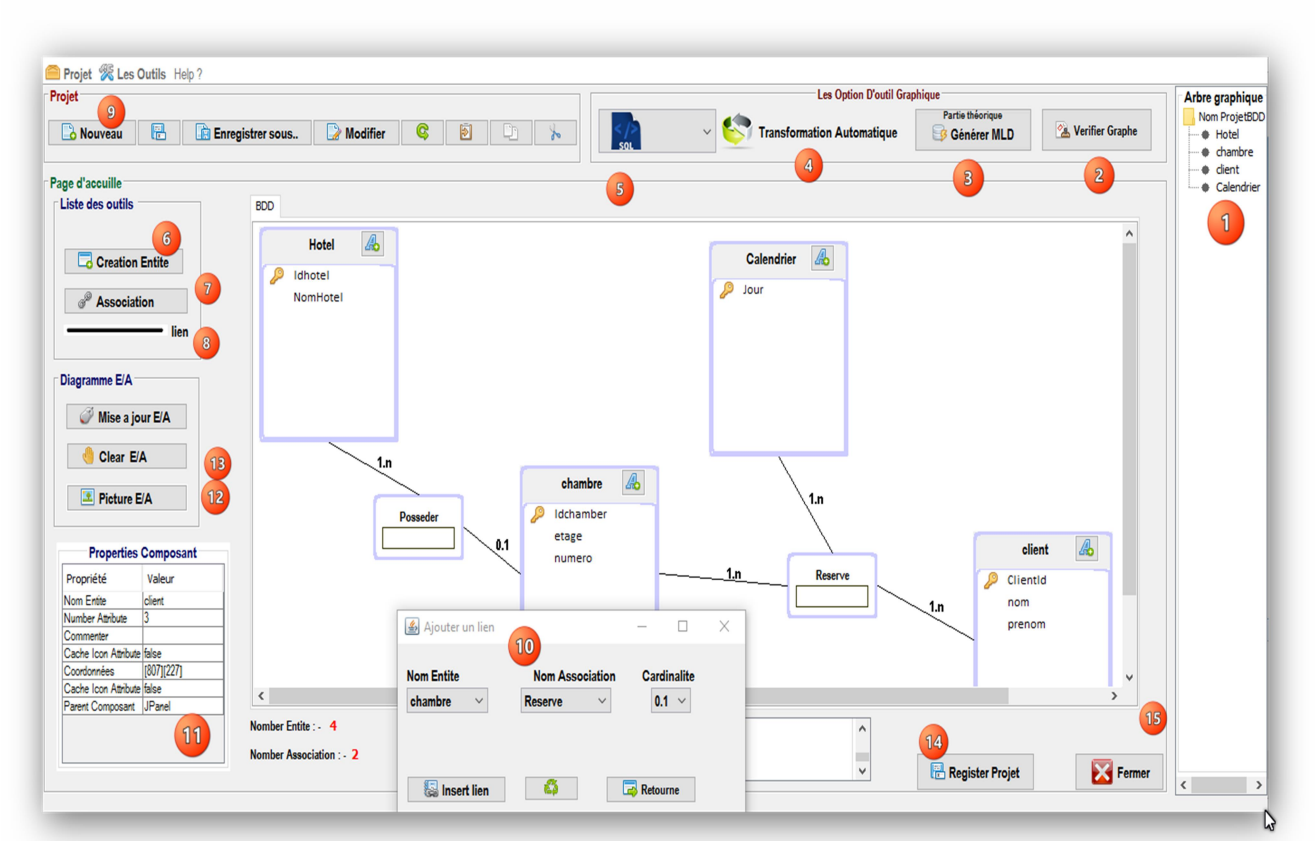

Figure 3.4 : Mode de création d'un graphe E / A.

➢ *Explication du Mode de création d'un graphe E/A :*

1- L'arbre du modèle est une liste qui répertorie l'ensemble des entités.

2- Vérification de graphe E/A et définition du problème comme la figure (3.5).

3- Générer le modèle logique de données

4- La transformation automatique de graphe E/A en script SQL

5-Choisir le type de SGBD de la transformation du graphe.

6-Ajouter une entité dans le graphe E/A

7-Ajouter une association dans le graphe

8- Ouvrir la fenêtre du dialogue numéro 10 pour créer la liaison.

9- Cette barre permettra l'ajout d'un nouveau projet, l'enregistrement ou la modification.

10- Boîte de dialogue qui permet la création et l'affichage des liens.

11- Affichage des propriétés de chaque composant du graphe avec la possibilité de modification.

12-Export le graphe E/A sous forme d'image.

13- Réinitialise le graphe en supprimant tous les composants visuels.

14- Enregistrer le projet (E/A) sur tout on choisissant le répertoire de destination.

15- Fermer l'outil graphique avec l'enregistrement du projet ou annuler l'enregistrement.

5.2. Vérification d'un graphe E/A:

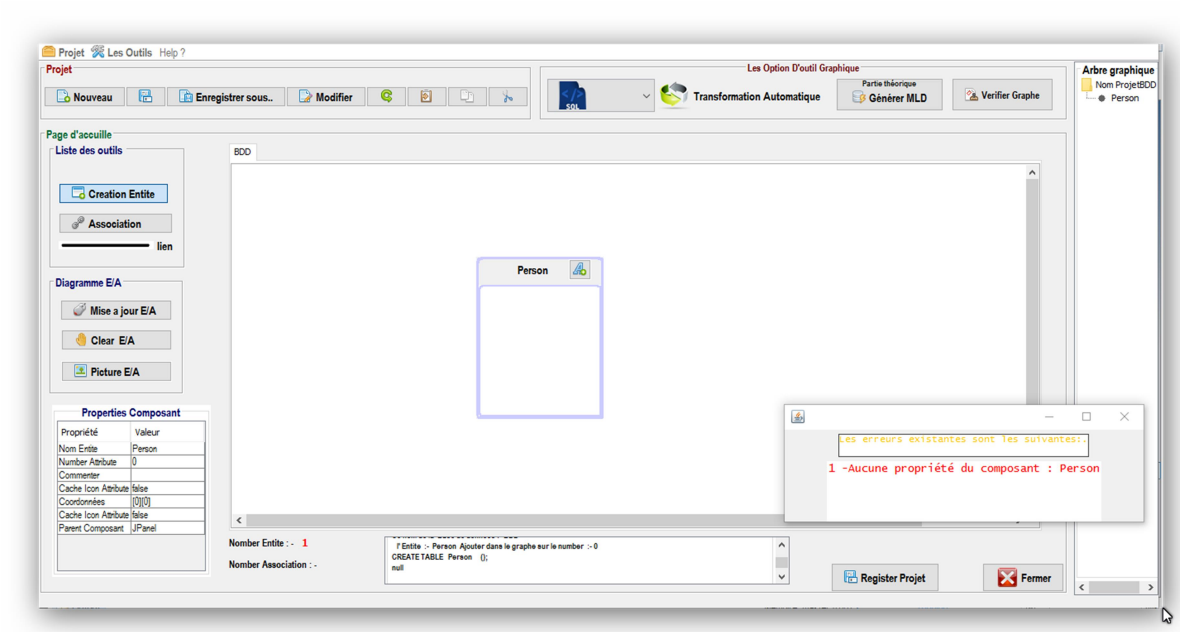

Figure 3.5 : Vérification d'un graphe E/A.

➢ *Explication du Vérification d'un graphe E/A:*

Dans ce cas l'outil graphique fait la vérification du graphe et détecte les erreurs.

5.3. Transformation automatique vers le fichier de script SQL:

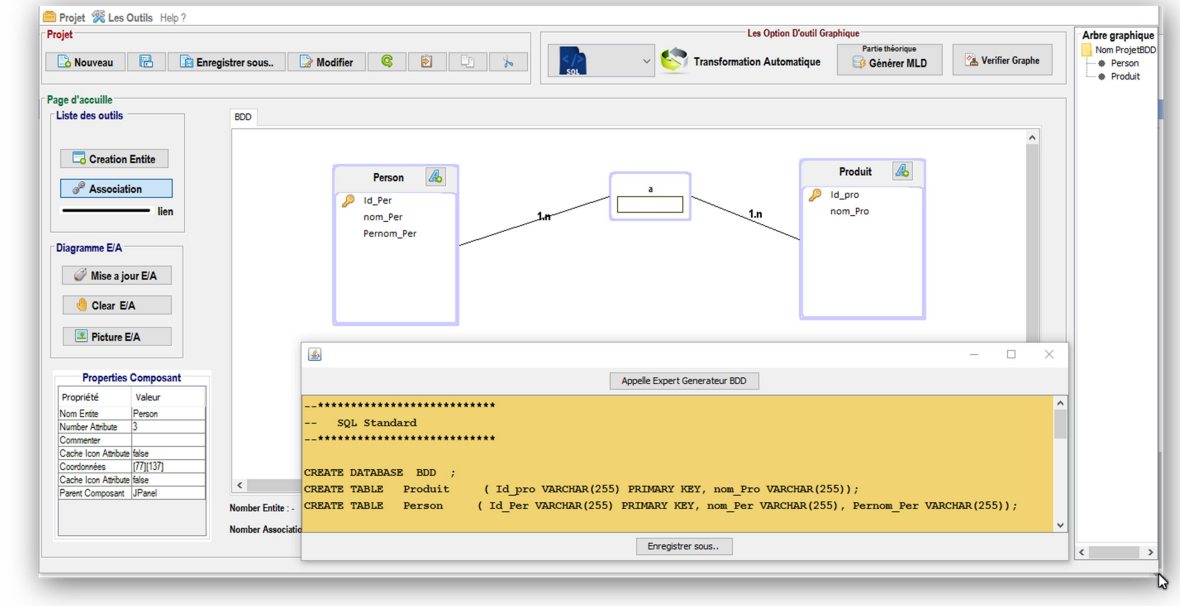

Figure 3.6 : Transformation automatique vers le fichier de script SQL.

➢ *Explication du Vérification d'un graphe E/A:*

Cette interface permet d'afficher le contexte SQL.

5.4 La connexion avec SGBD et l'utilisation du fichier de script SQL existant :

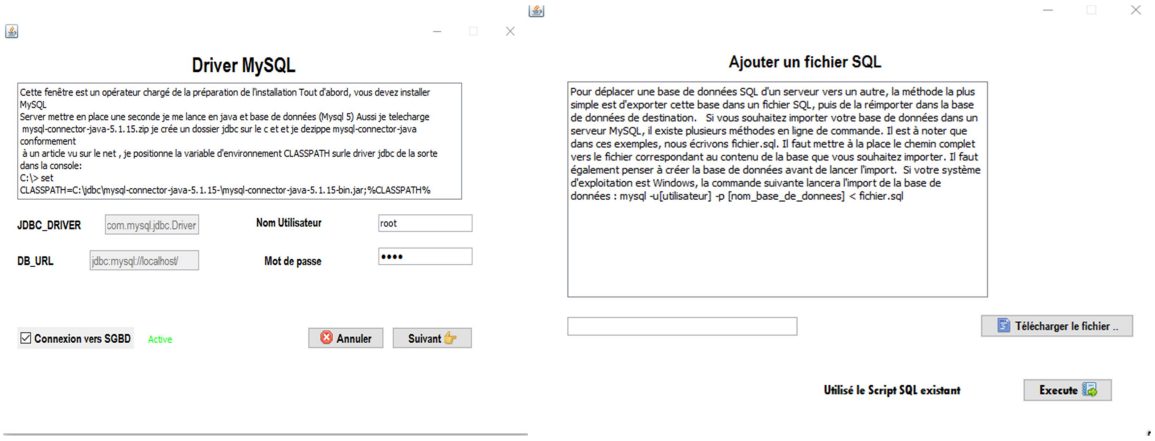

Figure 3.7 : La connexion avec SGBD et l'utilisation du fichier de script SQL existant.

➢ *Explication du Vérification d'un graphe E/A:*

À partir de ces deux fenêtres on peut accéder à l'SGBD (MySQL) et Créer la base de données du graphe.

5.5 Les fonctionnalités de l'outil graphique sont :

1. Création d'un ou plusieurs Modèles conceptuels de données MCD.

2. Vérification des MCDs et génération du MLD.

3. Génération des scripts SQL propres à différentes SGBD : MySQL, PostGres, SQL Server, Access, SQLite, Derby, Firebird, Oracle.

4. Création des bases de données : MySQL, Oracle ;

5. Exportation des dictionnaires de données et la liste des types sous format texte.

6. Exportation des MCD/MLD sous forme d'image jpg.

7. Importation ou ouverture des MCDs.

6 Conclusion:

Dans ce chapitre nous avons, au premier lieu, présenté les différents outils et langages que nous avons utilisé pour implémenter notre application. Par la suite, nous avons présenté quelque interface.

CONCLUSION GENERALE

Les SGBD relationnels sont à l'heure actuelle les systèmes les plus diffusés sur le marché. Leur succès est dû au modèle de données sur lequel ils sont basés à savoir : le modèle relationnel. Ce modèle est basé sur des fondements mathématiques (théorie des ensembles et des relations) et offre une approche méthodologique pour la conception d'une B.D. De plus, il faut noter aussi que le succès de ce modèle a été grandement favorisé par l'adoption par l'ISO du langage SQL (Structured Query Language) comme une norme pour les langages de description et de manipulation d'une base de données relationnelle.

Pour que l'Ingénierie Dirigée par les Modèles se diffuse auprès des développeurs, le savoir doit être essaimé des nouveaux outils qui doivent être développés. Pour produire ces outils qui font encore aujourd'hui défauts, plusieurs projets ont été lancés. Ces outils permettront l'automatisation des transformations ainsi que la génération automatique d'un code à partir des modèles.

Notre travail est un outil graphique permettant de convertir automatiquement une base de données relationnelle, et la conception d'un modèle de données, après nous avons procédé à la conception du l'outil en modélisant ces différentes parties à l'aide de langage de modélisation UML grâce à son extension pour l'outil graphique .Cette démarche nous a permet de réaliser une application qui répond aux spécifications et qui offre à l'utilisateur de l'application.

Le travail que nous avons réalisé nous a permet d'acquérir les notions de base sur des nouvelles connaissances des langages JAVA, SQL, et Driver JBDC, le langage de modélisation UML, et aussi d'utiliser et de maîtriser plusieurs logiciels tels que NetBeans, et le système de gestion de bases de données MySQL.

Ce travail nous a permet également de nous familiariser avec les outils d'analyse et de conception d'UML.

Comme perspective, nous avons dégagé les points suivants :

- Améliorer l'outil graphique qu'on a proposé par :
  - L'ajout des autres composants.
  - Améliorer notre outil pour suivre le cycle de vie de développement des applications.
  - Génération du Code java pour avoir un squelette d'application.

BIBLIOGRAPHIE ET CITATIONS

1. Les ouvrages:

[1]Adam Myatt,"Pro NetBeans™ IDE 6 Rich Client Platform Edition", APRESS, USA, 2008.p25.

[2]Carlos Coronel, Steven Morris et Peter Rob," Database Systems: Design, Implementation", and Management, CENGAGE LEARNING,USA ,2012.

[3] Christian Soutou, Frédéric Brouard,Rudi Bruchez, « SQL 4e édition Synthex », PEARSON EDUCATION, 2012.

[4] G. Roussel, E. Duris, "Java et Internet", VUIBERT, 2000.

[5] Gilles Roy, « Conception de bases de données avec UML », PRESSES DE L'UNIVERSITE DU QUEBEC, Canada, 2009.

[6] Jean-Luc BAPTISTE , « Merise - guide pratique - modélisation des données et des traitements , manipulations avec le langage SQL » ,2e édition, ENI Editions,France,2012.

[7] Jean-Luc Hainaut , « bases de données et modèles de calcul 4édition », DUNOD, Paris, 2005.

[8]Luciano Lavagno, Grant Martin and Bran V. Selic, "UML for real: design of embedded real-time systems", KLUWER ACADEMIC PUBLISHERS, Norwell USA, 2003.

[9] M. Robinson, P. Vorobiev, "Swing", MANNING PUBL, 2000.

[10] Nicolas Larrousse, « Création de bases de données », Pearson Education, France, 2009.

[11] Pascal. Roques. « UML 2 par la pratique. » , ÉDITIONS EYROLLES, 61, bd Saint-Germain75240 Paris Cedex 05 6e édition, 2008.P71 et P4 ,P5 .

[12] Pascal. Roques. « UML 2 par la pratique » , ÉDITIONS EYROLLES, 61, bd Saint-Germain75240 Paris Cedex 05 6e édition, 2008, P81.

[13] Pascal. Roques. « UML 2 par la pratique » , ÉDITIONS EYROLLES, 61, bd SaintGermain75240 Paris Cedex 05 6e édition, 2008, P100.

[14] S. Sumathi, S. Esakkirajan, "Fundamentals of Relational Database Management Systems», SPRINGER, Canada, 2007.

[15] White Fisher,"JDBC API Tutorial and Reference", 3rd EDS, ADDISON WESLEY, 2003.

2. Les articles :

[16] Jean-Christophe FORTON, « Conception BDD », WIKI, 2011, p. 12-16.

3. Mémoires et thèses :

[17] Mouna Bouarioua, Thèse de doctorat : « Une approche basée transformation de graphes pour la génération de Modèles », UNIVERSITÉ CONSTANTINE 2,2013.

4. Les sites Web:

[18] Commentcamarche ,www.commentcamarche.net/contents/1138-uml-cas-d-utilisationuse-cases. Consulté le : 05/04/2017

[19] Developpez ,http://jmdoudoux.developpez.com/cours/developpons/java/chappresentation.php. Consulté le : 06/04/2015

[20]Developpez,https://www.developpez.com/actu/94680/DB-Engines-Ranking-Oracle-designe-systeme-de-gestion-de-base-de-donnees-de-l-annee-2015-quel-est-votre-SGBD-prefere/. Consulté le : 21/03/17.

[21] Db-engines, http://db-engines.com/en/ranking. Consulté le : 22/03/2017

[22] Java, https://www.java.com/fr/download/faq/whatis_java.xml. Consulté le : 06/04/2015

[23]Microsoft,http://msdn.microsoft.com/fr-fr/library/ms378857(v=sql.90).aspx. Consulté le : 06/04/2015

[24]Netbeans, http://bits.netbeans.org/dev/javadoc/index.html. Consulté le : 06/04/2015

[25]Netbeans, http://netbeans.org/features/ide/index.html. Consulté le : 06/04/2015

[26]Netbeans, http://netbeans.org/kb/trails/platform.html. Consulté le : 06/04/2015

[27]Sql, https://sql.sh/cours/create-database. Consulté le : 05/03/2017

[28]Sql, https://sql.sh/logiciels.., Consulté le : 05/03/2017.

[29]Wikipedia,www.wikipedia.org/wiki/Diagramme_de_séquence, Consulté le : 04/04/2015.

[30]Wikipedia ,https://fr.wikipedia.org/wiki/Structured_Query_Language, Consulté le: 30/03/2017.

[31]Wikipedia,https://fr.wikipedia.org/wiki/netbeans, Consulté le : 06/04/2015.

[32]Wikipedia,https://fr.wikipedia.org/wiki/Systeme_de_gestion_de_base_de_donnees.Consulté le: 28/03/2017.

BIBLIOGRAPHIE ET CITATIONS

ملخص:

في أيامنا هاته، مجمع بسيط لا يكفي لإنجاز برامج ذات جودة عالية، بل يحتاج الى العديد من البرامج وذلك لإنجازه بطريقة اوتوماتيكية وتحسين الانتاجية، والمعروفة باسم: هندسة البرمجيات بمساعدة الحاسوب، يندرج عملنا في هذا المجال، حيث نقوم بإنجاز برنامج رسومي لتصميم قاعدة بيانات باستخدام نموذج الكيان / الرابطة المعروف باسم MCD وتحويله إلى قاعدة بيانات جاهزة للاستخدام.

الكلمات المفتاحية: الكيان, الرابطة , نموذج تصميم البيانات, النموذج المنطقي للبيانات العلائقي, التحويل الآلي, قاعدة بيانات, هندسة البرمجيات بمساعدة الحاسوب.

Résumé :

De nos jours, un simple compilateur ne suffit pas pour réaliser un logiciel de qualité mais il faut avoir tout une panoplie de logiciels pour automatiser au max le processus de développement et améliorer la productivité, connu sous le nom de: atelier de génie logiciel, notre travail s'inscrit dans le cadre de ce domaine, il consiste à réaliser un outil graphique permettent de modélise une base de données sous forme d'un modèle Entité / Association connu communément par MCD et de le convertir vers une base de données prête à l'exploitation.

Mots-clés : Entité, Association, MCD, MLD, Transformation automatique, BDD, Atelier de génie logiciel.

Summary:

Nowadays, a simple compiler is not enough to produce quality software, but it takes a lot of software to automate the development process and improve productivity. Known as: Computer Aided Software Engineering CASE, Our work falls within the scope of this field, it consists of realizing a graphical tool to design a database using Entity / Association model commonly known as MCD and to convert it to a database ready to use.

Keywords: Entity, Relationship, Attribute, conceptual data model, Logical Data model, Automatic Transformation, DataBase,CASE.