



REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTRE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE IBN KHALDOUN - TIARET

MEMOIRE

Présenté à :

FACULTÉ MATHÉMATIQUES ET INFORMATIQUE
DÉPARTEMENT D'INFORMATIQUE

Pour l'obtention du diplôme de :

MASTER

Spécialité : [Réseaux & Télécommunication]

Par :

[BENASLA Maroua et BENHENNI Imene]

Sur le thème

Conception et réalisation d'un outil de suivi de l'apprenant dans les séances des travaux pratiques

Soutenu publiquement le 26 / 06 / 2018 à Tiaret devant le jury composé de :

Mr HAMDANI Abdia	MAA Université Ibn khaldoun –Tiaret	Président
Mr OUARED Abdelkader	MAA Université Ibn khaldoun –Tiaret	Encadreur
Mr LAARADJ	MAA Université Ibn khaldoun -Tiaret	Examinateur

Remerciements

Tout d'abord merci à dieu pour tous

Nous adressons nos remerciements les plus sincères à notre promoteur MR OUARED. Abdelkader, pour nous avoir proposé ce sujet et de nous avoir conseillé et dirigé tout au long de l'année.

Nous tenons également à exprimer notre gratitude à tous les enseignants de département informatique, ainsi qu'à tous nos collègues.

Merci à tous ceux qui ont contribué de près ou de loin à la réalisation de ce modeste travail, pour leurs conseils, leurs encouragements et leurs soutiens.

Enfin, nous tenons à remercier les membres du jury pour nous avoir honorés en acceptant de juger notre travail.

Dédicace

Je dédie ce travail

A la personne la plus chère à mes yeux, à ma mère qui a tout sacrifié pour ses enfants, qui a veillé à notre éducation, qui, Sans elle je ne serai pas ce que je suis.

A mon cher père, à qui je dois ma place maintenant pour ses sacrifices.

A mes très chères sœurs et frères.

A ma meilleure amie, sœur et binôme Maroua, avec qui j'ai eu un grand plaisir de partager ce modeste travail A mes amies proches et mes cousines.

A tous ceux que j'aime.

Imene

Je dédie ce travail

A la personne la plus chère à mes yeux, à ma mère qui a tout sacrifié pour ses enfants, qui a veillé à notre éducation, qui, Sans elle je ne serai pas ce que je suis.

A mon cher père, à qui je dois ma place maintenant pour ses sacrifices.

A mes très chères sœurs et frères.

A ma meilleure amie, sœur et binôme Imene, avec qui j'ai eu un grand plaisir de partager ce modeste travail A mes amies proches et mes cousines.

A tous ceux que j'aime.

Maroua

Résumé

Dans le cadre de ce projet, nous sommes intéressés au langage de requête le plus utilisé : SQL (Structured Query Language). Dans le module de base de données de 2ème année, la formulation des requêtes et le processus d'exécution de ces requêtes est apparaît difficile à assimilée pour les étudiants. Afin d'aider les enseignants/ étudiants à comprendre ces concepts relatifs au processus d'exécution des requêtes SQL et assister les enseignants durant les séances de travaux pratique « TP » (*Traçabilité de TP, Tests, Erreurs, etc.*).

Il est intéressant d'offrir aux enseignants un outil assistant de suivi de l'apprenant pour comprendre le parcours de l'apprenant, relier le comportement à la connaissance des requêtes SQL.

Notre solution permet au enseignant d'apprécier non seulement quantitativement mais surtout qualitativement le travail des étudiants et explorer la trace détaillée.

Mots clés :

Base de données, traçabilité, suivi, SQL, requête, étudiant, enseignant, erreur, Travaux pratiques.

Abstract

As part of this project, we are interested in the most used query language: SQL. In the 2nd year database module, the wording of the queries appears difficult to assimilate for the students. In order to help teachers / students to understand these concepts related to the process of executing SQL queries and to assist teachers during lab sessions (Traceability of Labs, Tests, Errors, etc.).

It is interesting to offer teachers a learner tracking assistant tool to understand the learner's journey, to relate behavior to the knowledge of SQL queries.

Our solution allows the teacher to appreciate not only quantitatively but above all qualitatively the work of the students and to explore the detailed trace.

Keyword :

SQL, tracability, errors, queries, teacher, student.

ملخص

كجزء من هذا المشروع، نحن مهتمون بلغة الاستعلام الأكثر استخداماً: SQL في وحدة قواعد بيانات السنة الثانية، يبدو أن صياغة الاستعلامات يصعب استيعابها للطلاب. من أجل مساعدة المعلمين / الطلاب على فهم هذه المفاهيم المتعلقة بعملية تنفيذ استعلامات SQL ومساعدة المعلمين أثناء جلسات الأعمال التطبيقية (إمكانية تتبع المختبرات ، والاختبارات ، والأخطاء ، وما إلى ذلك) ، فمن الممتع تقديم المعلمون أداة مساعدة تعقب المتعلم لفهم رحلة المتعلم ، لربط السلوك بمعرفة استعلامات SQL. يتيح حلنا للمعلم تقدير ليس فقط كمياً ولكن قبل كل شيء عمل الطلاب واستكشاف الأثر التفصيلي.

الكلمات المفتاحية: المعلمين ، الطلاب ، الأخطاء ، الأعمال التطبيقية، استعلامات SQL .

Table Des Matières

Contenu

Introduction générale :	6
Contexte et Motivation :.....	6
Problématique :.....	6
Objectifs :	7
Organisation de mémoire :	7
Chapitre I : Généralités sur le langage SQL	11
I.1 Introduction :	11
I.2 Traitement de requêtes dans un SGBD relationnel :.....	11
I.2.1 Analyse :.....	12
I.2.1.1 Une grammaire pour un sous-ensemble simple de SQL.....	12
I.2.1.2 Catégories syntaxiques :.....	13
I.2.2 Transformation :	15
I.2.2.1 Règles algébriques pour améliorer les plans de requête :	15
I.2.2.2 Amélioration du plan de requête logique :	18
I.2.3 Génération des plans et optimisation :	19
I.2.3.1 Algèbre physique :.....	19
I.2.3.2 Statistiques sur les données :	20
I.2.3.3 Statistiques de système :.....	21
Chapitre II:Généralités sur l'apprentissage	24
II .1 Introduction :.....	25
II.2 Concept et terminologie :.....	25
II.2.1 Pourquoi faire du e-Learning ? [9]	25
II.2.2 Définition de l'apprentissage :	25
II.2.3 Les objectifs d'apprentissage :.....	25
II .2 .4 Triangle pédagogique :	26
II. 2. 5 Les acteurs et leurs rôles :.....	27
II.3 Taxonomies d'objectifs d'apprentissage :	28
II .4 comment les étudiants apprennent: un apprentissage taxonomie ?.....	29
II.5 Taxonomie d'apprentissage en informatique :	32
II.6 Scénarisation Pédagogique :	32
II.6.1 Définitions de scénario pédagogique :.....	32
II.6.2 Autres concepts de base :.....	33

II.6.3 Principes de modélisation des activités pédagogiques en ligne :	33
II.7 Evaluation et suivi de l'apprenant :	34
II.7.1 Le suivi :	34
II.7.2 L'évaluation :	35
II.8 Positionnement de notre solution :	35
II .9 Conclusion :	35
Chapitre III : Analyse et conception de notre solution	38
III.1.Introduction :	38
III.2 Analyse de l'existant :	38
III.2.1 Méthodes d'analyse :	38
III.2.2 Justification :	39
III.2.3 Les Anomalies :	40
III.2.3.1 Anomalies informationnelles :	40
III.2.3.2 Anomalies organisationnelles :	40
III.2.3.3 Anomalies techniques :	41
III.2.4 Solutions proposées :	42
III.2.4.1 Solutions d'ordre informationnel :	42
III.2.4.2 Solutions d'ordre organisationnel :	42
III.2.4.3 Solutions d'ordre technique :	42
III.2.5 Système Projeté :	43
III.2.6 Fondement Théorique:	43
III.2.6.1 Application de taxonomie de BLOOM :	43
III.2.6.2 Classification des erreurs :	45
III.2.6.3 Directives de formulation SQL :	45
III.2.6.4 Traçabilité :	46
III.3 Etude conceptuelle:	46
III.3.1 Diagramme de contexte du système :	47
III.3 .2 Diagramme de cas d'utilisation :	47
III.3 .2 Diagramme de classe :	50
III. 3 .3Diagramme de séquence :	51
Cas N°02 : <i>Voir la traçabilité</i>	53
Cas N°03 : <i>Ajouter compte</i>	53
Cas N°04 : <i>Modifier compte</i>	54
Cas N°5 : <i>Ajouter question</i>	54

Cas N°6 : <i>Ajouter question</i>	55
Chapitre IV: Implémentation de notre outil prototype de faisabilité.....	57
IV.1 Introduction :	57
IV.2 Technologies utilisées pour le développement :	57
IV.2.1 HFSQL :	57
IV.2.2 Entreprise Architect :	58
IV.2.3 Langage de Programmation WinDev :.....	58
IV.3 WinDev et le Client/ Serveur :.....	59
IV.3.1 Mettre en place une application Hyper File Client / serveur :	60
IV.3.2 Définition de la connexion par l'éditeur d'analyses :.....	61
IV.4 Démonstrations de notre application ;.....	62
IV.4.1 Authentification :	62
IV.4.2 L'interface principale :.....	63
IV.4.3 Catalogue SQL :	63
IV.4.4 Gestion TP :	64
IV.4.5 Scénario TP :	65
IV.4.6 Traçabilité :.....	65
IV.4.7 Configuration de serveur SGBD et des utilisateurs :	66
IV.4.8 Les tables de base de données :	68
IV.4 Conclusion :	68
Conclusion Générale:.....	68
Référence bibliographique:	68
Webographie :.....	75

Introduction Générale

Introduction générale

Introduction générale :

Contexte et Motivation :

Avec les progrès récents des TIC¹ dans le domaine de l'apprentissage, il se doit, dès à présent, « penser à un apprentissage rapide et efficace », avec un minimum de problèmes d'organisation, de logistique et surtout de perte de temps.

La plateforme éducative ou l'apprentissage électronique est la solution. C'est le nom donné actuellement à une phase importante de l'introduction des TIC dans l'apprentissage.

La plateforme éducative consiste à utiliser les ressources de l'informatique et de l'Internet pour acquérir, à distance, des connaissances. Cependant, il permet également une autonomie de l'apprenant qui devient ainsi l'acteur de sa formation.

Ce besoin de formation surtout dans les instituts et universités algériennes a donné naissance au développement des modules et des simulateurs qui aident l'enseignant d'enrichir son cours avec des exemples interactifs ; ces activités pédagogiques dont l'objectif est l'apprentissage, à distance, des cours aux étudiants en informatique.

Problématique :

Depuis ces dernières années, plusieurs phénomènes renouvèlent l'intérêt pour la formation et l'apprentissage à distance, la croissance spectaculaire des TIC et d'internet surtout est l'un de ces principaux phénomènes, mais il peut aussi s'agir de phénomènes sociaux et culturels, face à l'augmentation des publics, à l'insuffisance de l'offre de formation, aux faibles capacités d'accueil et de suivi, aux disparités régionales, aux inégalités d'accès, aux besoins croissants de formation dans plusieurs domaines et spécialement le domaine informatique et technologie de l'information, le recours aux technologies éducatives serait une réponse pour relever ces défis.

L'apprentissage à distance se voit aujourd'hui comme une solution palliant aux contraintes de l'apprentissage en présentiel.

Dans le cadre d'apprentissage dans les écoles et universités algériennes, la compréhension et la manipulation des langages de requêtes (le langage SQL surtout) constituent actuellement.

Un concept de plus en plus exploré car malgré les efforts institutionnels et universitaires à tous les niveaux (évolution des programmes, plan de formation des enseignants), les difficultés de la maîtrise de ces langages de manipulation des BDD sont de plus en plus aperçues.

En effet, apprendre par cœur et de manière classique des cours en SQL, plus précisément, le processus d'optimisation des requêtes ne motive plus l'apprenant car ce dernier a besoin d'être actif et d'agir pendant les séances d'apprentissage, de contribuer à la construction de cours, d'être convaincu de ses erreurs.

¹ TIC : Technologie de l'information et de la communication.

Introduction générale

Toutefois, avec le nombre croissant d'utilisateurs de ces systèmes et la différenciation de l'espace et du temps entre les apprenants et les enseignants, le suivi et l'adaptation de la présentation du contenu dans le domaine informatique devient de plus en plus difficile.

Cependant une multitude de sites web éducatifs envahit la toile, des milliers de sites et pages, différentes par la forme et la présentation, reprennent les mêmes concepts et les notions de la même façon.

Ce travail tente de répondre à un ensemble de questions qu'on peut les résumer comme ceci :

- Comment se démarquer de la pratique d'enseignement artisanale de la conception de ressources pédagogiques afin d'éviter de les démultiplier à l'identique ?
- Comment mettre l'action sur le sens plutôt que la forme et ainsi de mettre en œuvre l'apprentissage et l'enseignement basés sur des méthodes communicatives ?
- Comment rendre la parole à l'apprenant afin de lui permettre d'être actif et de mettre sa contribution en toute liberté selon ses besoins et son parcours pédagogique ?
- Comment mettre en œuvre la relation entre les différents acteurs ?
- Qu'elles sont les méthodes et outils qui motivent et attire de plus l'apprenant et qui lui permettront de réguler son apprentissage ?

Objectifs :

Notre travail s'inscrit dans un projet de réalisation d'une plateforme éducative et interactive spécialisée en langage SQL, initiée au différents processus d'optimisations des requêtes SQL afin d'offrir l'opportunité aux apprenants de maîtriser ce processus et de mieux les armer pour leurs avenir académique et leurs vie professionnelle.

Organisation de mémoire :

Notre mémoire est organisé en quatre chapitres :

- Chapitre 01 : Présente des Généralités sur le langage SQL.
- Chapitre 02 : Présente des Généralités sur l'apprentissage.
- Chapitre 03 : Est consacré à l'analyse et la conception de notre solution.
- Chapitre 04 : Présente les technologies utilisées et la démonstration de notre outil prototype de faisabilité.

La figure suivante illustre la visualisation de notre travail

Introduction générale

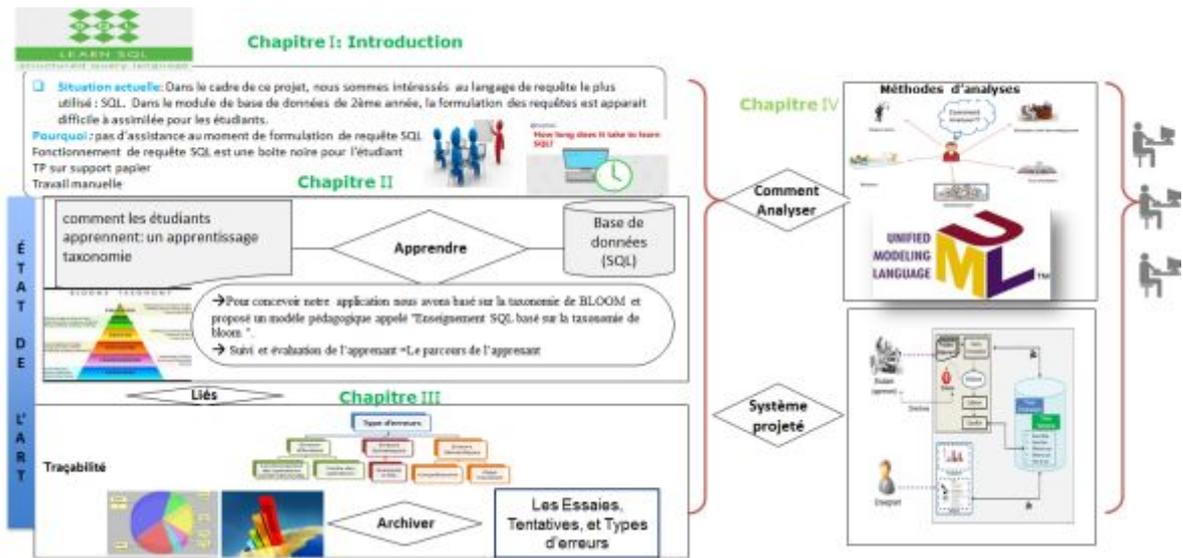


Figure 1 : Organisation de mémoire.

Chapitre I

Généralités sur le langage SQL

« The begining of knowledge is the discovery of something we do not understand »

Frank herbert

Chapitre I : Généralités sur le langage SQL

I.1 Introduction :

SQL convient pour gérer de grandes collections de données organisées en listes d'attributs. L'activité classique d'une entreprise, les transactions et les ressources sont parfaitement gérées par un tel outil.

C'est le langage le plus utilisé pour construire et utiliser les bases de données relationnelles. Le nom était au départ SEQUEL, abréviation de "*Structured English Query Language*", puis a été condensé en SQL et le nom complet "Structured Query Language", lui est attribué plus tard. Les créateurs du langage sont Donald D. Chamberlin et Raymond F. Boyce à IBM. L'implémentation System/38 a été vendue par IBM en 1979, tandis que le logiciel Oracle était créé par Relational Software la même année (cette entreprise s'est renommée Oracle plus tard) [1].

En 1982 IBM implémente SQL dans DB2. SQL devient un standard ANSI en 1986, un standard ISO¹ en 1987, mais les implémentations concrètes sont loin d'être standardisées. SQL 2003 ajoute des capacités XML² au langage[FRM, 09] .

I.2 Traitement de requêtes dans un SGBD relationnel :

Après la conception et la création de la base de données, les utilisateurs peuvent exploiter les données.

Le traitement de requêtes est le groupe de composants d'un SGBD qui transforme les requêtes des utilisateurs et les commandes de modification des données en une séquence d'opérations de base de données et exécute ces opérations. [1]

Dans ce chapitre, nous discuterons des techniques utilisées en interne par un ³SGBD pour traiter des requêtes de haut niveau. Une requête exprimée dans un langage de requête de haut niveau comme SQL doit d'abord être analysée syntaxiquement. L'analyseur identifie les mots clés de la requête (les mots clés SQL, les noms d'attributs et les noms de relations) qui apparaissent dans le texte de la requête, et vérifie la syntaxe de requête pour déterminer si elle est formulée selon les règles de syntaxe (règles de grammaire) du langage de requête. La requête doit également être validée en vérifiant que tous les noms d'attributs et de relations sont des noms valides et sémantiquement significatifs dans le schéma de la base de données. En utilisant des règles de transformation, une représentation interne de la requête est ensuite créée, habituellement sous la forme d'une structure de données arborescente appelée arbre algébrique. Le SGBD doit alors concevoir une stratégie d'exécution ou un plan de requête pour extraire les résultats de la requête à partir des fichiers de base de données. Une requête a de nombreuses stratégies d'exécution possibles, et le processus de choisir un approprié pour

¹ ISO :International Organization for Standardization.

² XML : eXtended Markup Language.

³ SGBD : Système de Gestion de Base de Données

traiter une requête est connu sous le nom d'optimisation de requête [2]. La Figure I.1 donne un aperçu du processus de traitement des requêtes par un SGBD.

I.2.1 Analyse :

Le travail de l'analyseur est de prendre du texte écrit dans une langue comme SQL et de le convertir en un arbre d'analyse, qui est un arbre dont les nœuds correspondent soit à :

(1) atomes, qui sont des éléments lexicaux tels que des mots clés (par exemple, SELECT), des noms d'attributs ou de relations, des constantes, des parenthèses, des opérateurs tels que + ou <, et d'autres éléments de schéma, ou (2) des catégories syntaxiques, qui sont des noms pour des sous-parties de requête qui ont un rôle similaire. Ils sont représentés par un nom descriptif entre les symboles < et >[1]. Par exemple, <Requête> est utilisé pour représenter une requête, et <Condition> représentera toute expression qui est une condition. Nous présenterons ces propos par un exemple dans la section suivante.

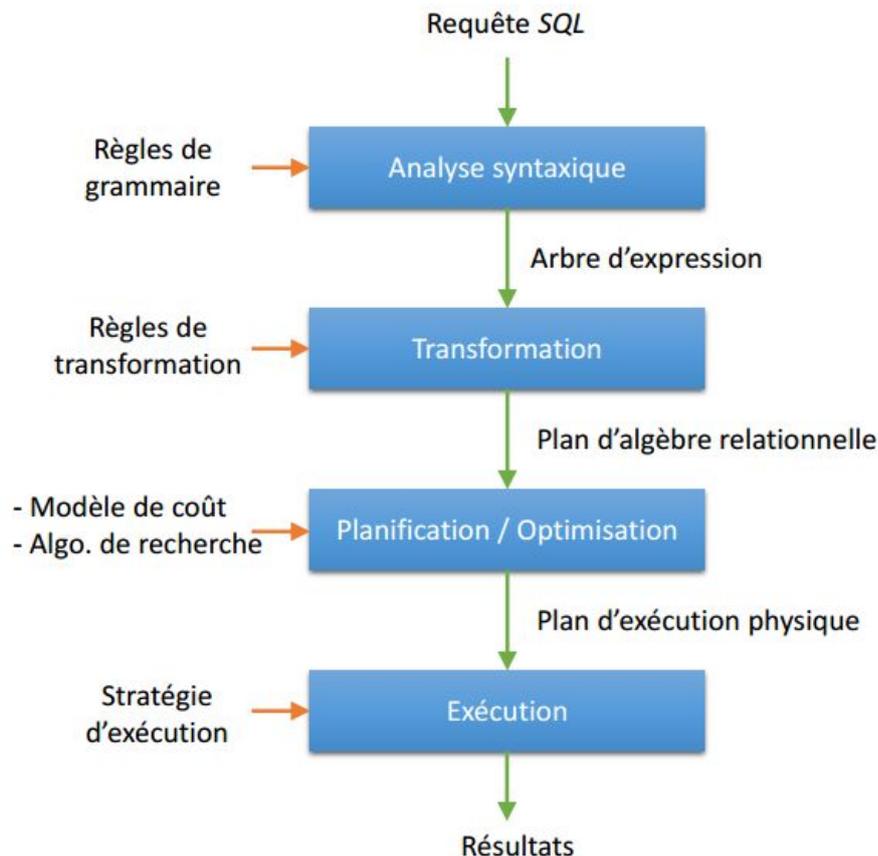


Figure I.1 : Architecture globale pour le traitement de requêtes dans un SGBD

I.2.1.1 Une grammaire pour un sous-ensemble simple de SQL

Nous allons illustrer le processus d'analyse en donnant quelques règles qui décrivent un petit sous-ensemble de requêtes SQL.

Requête

La catégorie syntaxique <Requête> est destinée à représenter certaines requêtes SQL. Nous lui donnons la règle :

<Requête> := SELECT <Sélection> FROM <Depuis> WHERE <Condition>

Le symbole := signifie « peut-être exprimer comme ». Les catégories syntaxiques <Sélection>, <Depuis> et <Condition> représentent des listes pouvant suivre SELECT, FROM et WHERE, respectivement.

Nous décrivons les formes de ces listes dans les prochaines sections.

Sélection

<Sélection> := <Attribut>, <Sélection>

<Sélection> := <Attribut>

Ces deux règles disent qu'une liste de sélection peut être une liste d'attributs séparés par des virgules : soit un attribut unique, soit un attribut, une virgule et toute liste d'un ou plusieurs attributs.

Depuis

<Depuis>:= <Relation>, <Depuis>

<Depuis>:= <Relation>

Ici, une liste de type Depuis est définie comme étant une liste de relations séparées par des virgules.

Condition

Les règles que nous utiliserons sont :

<Condition> := <Condition> AND <Condition>

<Condition> := <Attribut> IN (<Requête>)

<Condition> := <Attribut> = <Attribut>

<Condition> := <Attribut> LIKE <Motif>

I.2.1.2 Catégories syntaxiques :

Les catégories syntaxiques <Attribut>, <Relation> et <Motif> sont spéciales, puisqu'elles ne sont pas définies par des règles grammaticales, mais par des règles sur les atomes pour les quels elles peuvent être utilisées. Par exemple, dans un arbre d'analyse, l'un des fils de <Attribut> peut être n'importe quelle chaîne de caractères qui identifie un attribut du schéma de base de données. De même, <Relation> peut être remplacé par n'importe quelle

chaîne de caractères qui représente une relation dans le schéma, et <Motif> peut être remplacé par n'importe quelle chaîne entre guillemets qui est un modèle SQL légal.

Exemple 01 : Notre étude de l'analyse et de la réécriture de requêtes sera basée sur notre exemple de gestion de vente. Nous voulons répondre à la requête « trouver les noms des clients qui ont payé leurs commandes en espèces ». Nous identifions les clients qui ont payé des commandes en espèces en demandant si leur méthode de paiement est égale à « espèces », en utilisant l'opérateur =. La requête SQL utilisée pour répondre à cette question est :

```

1 SELECT Nom
2 FROM Clients, Commandes
3 WHERE Clients.ClientsID = Commandes.ClientsID AND
4     Méthode.Paiement = 'espèces';

```

L'arbre d'analyse pour notre requête, selon la grammaire que nous avons proposée, est illustré dans la **Figure I.2**. A la racine se trouve la catégorie syntaxique <Requête>. En parcourant l'arbre, nous voyons que cette requête a une forme de select-from-where, la liste de <Sélection> se compose uniquement de l'attribut Nom et la liste de <Depuis> contient plus d'une relation et deux conditions connectées par AND.

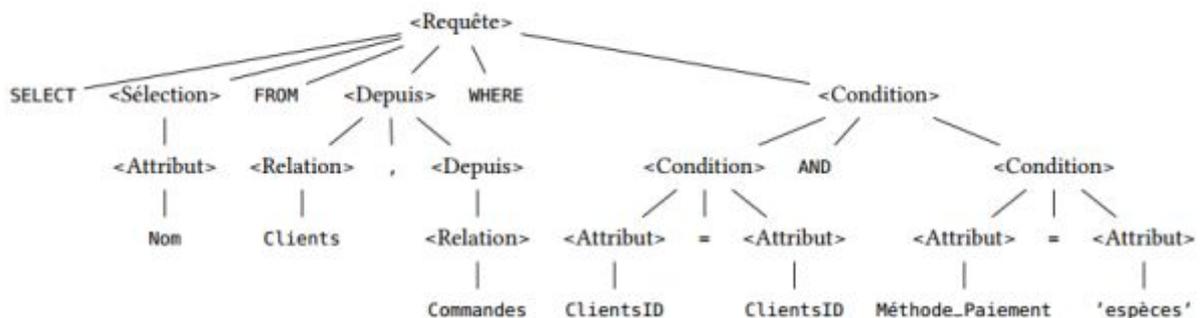


FIGURE I.2 : Exemple d'un arbre d'analyse.

L'analyseur vérifie si une relation dans la requête est en réalité une vue virtuelle, chaque utilisation de cette relation dans la requête doit être remplacée par un arbre d'analyse qui décrit la vue[1]. L'analyseur est aussi responsable de la vérification sémantique. Même si la requête est valable syntaxiquement, elle peut en fait violer une ou plusieurs règles sémantiques sur l'utilisation des noms.

Par exemple, l'analyseur doit :

- **Vérifiez les relations :** Chaque relation mentionnée dans une clause FROM doit être une relation ou une vue dans le schéma de la base de données courante.
- **Vérifiez et résoudre les attributs :** Chaque attribut mentionné dans la clause SELECT ou WHERE doit être un attribut d'une certaine relation dans la requête. Par exemple, l'attribut Nom est un attribut de Clients, donc l'analyseur valide cette utilisation. Il

vérifie également l'ambiguïté, signalant une erreur si un attribut est dans la portée de deux ou plusieurs relations.

- **Vérifiez les types :** Tous les attributs doivent être d'un type approprié à leurs utilisations. Par exemple, Méthode_Paiement est utilisé dans une comparaison =, ce qui nécessite que cet attribut soit une chaîne ou un type qui peut être contraint à une chaîne. De même, les opérateurs sont vérifiés pour voir qu'ils s'appliquent aux valeurs des types appropriés et compatibles.

I.2.2 Transformation :

Dans cette étape, l'arbre d'analyse résultant de l'étape précédente est transformé en une expression de l'algèbre relationnelle étendue. L'objectif de la transformation des requêtes est (1) la construction d'un arbre standardisé pour l'optimisation des requêtes (standardisation), (2) l'élimination de la redondance (simplification), et(3) la construction d'une expression améliorée (amélioration)[3].

Nous verrons dans cette section comment appliquer des heuristiques qui améliorent l'expression algébrique de la requête, en utilisant certaines des nombreuses règles algébriques issues de l'algèbre relationnelle [2]. À titre préliminaire, cette section répertorie les règles algébriques qui transforment un arbre d'expression en un arbre d'expression équivalent qui peut avoir un plan de requête physique plus efficace. Le résultat de l'application de ces transformations algébriques est le plan de requête logique qui est la sortie de cette phase.

I.2.2.1 Règles algébriques pour améliorer les plans de requête :

Une règle est dite commutative sur un opérateur si le résultat reste invariable si l'on intervertit les arguments de l'opérateur. Par exemple, + et * sont des opérateurs commutatifs d'arithmétique. Plus précisément, $x + y = y + x$ et $x * y = y * x$ pour tout nombre x et y . D'autre part, - n'est pas un opérateur arithmétique commutatif : $x - y \neq y - x$.

Une règle est dite associative sur un opérateur si nous pouvons regrouper deux utilisations de l'opérateur soit de la gauche ou de la droite. Par exemple, + et * sont des opérateurs arithmétiques associatifs, ce qui signifie que $(x + y) + z = x + (y + z)$ et $(x * y) * z = x *(y* z)$. D'autre part, - n'est pas associatif : $(x - y) - z \neq x - (y - z)$.

Nous allons présenter quelques règles de transformation de certains opérateurs SQL[4,2,1]:

1. **Décomposition de σ .** Une condition de sélection conjonctive peut être décomposée en une cascade (c'est-à-dire une séquence) d'opérations σ individuelles :

$$\sigma_{c_1 \text{ AND } c_2 \text{ AND } \dots \text{ AND } c_n}(R) \equiv \sigma_{c_1}(\sigma_{c_2}(\dots(\sigma_{c_n}(R))\dots))$$
2. **Commutativité de σ .** L'opération de sélection σ est commutative :

$$\sigma_{c_1}(\sigma_{c_2}(R)) \equiv \sigma_{c_2}(\sigma_{c_1}(R))$$
3. **Décomposition de π .** Dans une décomposition (séquence) d'opérations de projection π , tous sauf le dernier peuvent être ignorés :

$$\pi_{Liste_1}(\pi_{Liste_2}(\dots(\pi_{Liste_n}(R))\dots)) \equiv \pi_{Liste_1}(R)$$
4. **Commuter σ avec π .** Si la condition de sélection c ne concerne que les attributs A_1, \dots, A_n dans la liste de projection, les deux opérations peuvent être commutées :

$$\pi_{A_1, A_2, \dots, A_n}(\sigma_c(R)) \equiv \sigma_c(\pi_{A_1, A_2, \dots, A_n}(R))$$
5. **Commutativité de \bowtie (et \times).** L'opération de jointure est commutative, de même que l'opération de produit cartésien \times :

$$R \bowtie_c S \equiv S \bowtie_c R$$

$$R \times S \equiv S \times R$$
6. **Commuter σ avec \bowtie (ou \times).** Si tous les attributs de la condition de sélection c ne comportent que les attributs d'une des relations jointes, par exemple R , les deux opérations peuvent être commutées comme suit :

$$\sigma_c(R \bowtie S) \equiv (\sigma_c(R)) \bowtie S$$
7. **Commuter π avec \bowtie (ou \times).** Supposons que la liste de projection est $L = A_1, \dots, A_n, B_1, \dots, B_m$, où A_1, \dots, A_n sont des attributs de R et B_1, \dots, B_m sont des attributs de S . Si la condition de jointure c implique seulement les attributs en L , les deux opérations peuvent être commutées comme suit :

$$\pi_L(R \bowtie_c S) \equiv (\pi_{A_1, \dots, A_n}(R)) \bowtie_c (\pi_{B_1, \dots, B_m}(S))$$
8. **Commutativité des opérations ensemblistes.** Les opérations ensemblistes \cup et \cap sont commutatives, mais $-$ ne l'est pas.
9. **Associativité de \bowtie, \times, \cup et \cap .** Ces quatre opérations sont individuellement associatives ; C'est-à-dire si les occurrences de θ représentent la même opération qui est l'une de ces quatre opérations, on a :

$$(R \theta S) \theta T \equiv R \theta (S \theta T)$$

10. **Commuter σ avec des opérations ensemblistes.** L'opération σ commute avec \cup , \cap et $-$. Si θ représente l'une de ces trois opérations dans une expression, on a :

$$\sigma_c(R \theta S) \equiv (\sigma_c(R)) \theta (\sigma_c(S))$$

11. **L'opération π commute avec \cup .**

$$\pi_L(R \cup S) \equiv (\pi_L(R)) \cup (\pi_L(S))$$

12. **Conversion d'une séquence (σ, \times) en \bowtie .** Si la condition c d'une σ qui suit un \times correspond à une condition de jointure, convertissez la séquence (σ, \times) en une \bowtie comme suit :

$$(\sigma_c(R \times S)) \equiv (R \bowtie_c S)$$

13. **Descendre σ en bas en conjonction avec la différence ensembliste.**

$$\sigma_c(R - S) = \sigma_c(R) - \sigma_c(S)$$

14. **Descendre σ à un seul argument dans \cap .** Si dans la condition σ_c tous les attributs sont de la relation R , alors :

$$\sigma_c(R \cap S) = \sigma_c(R) \cap \sigma_c(S)$$

15. **Élimination des doubles.** L'opérateur δ élimine les doublons d'un ensemble. Si θ représente l'une de ces trois opérations : \bowtie , \bowtie_c ou \times , dans une expression, alors :

$$\delta(R \theta S) \equiv \delta(R) \theta \delta(S)$$

$$\delta(\sigma_c(R)) \equiv \sigma_c(\delta(R)).$$

16. **Règles de groupement et d'agrégation.** L'opérateur γ a plusieurs règles, par exemple, il absorbe δ . Nous pouvons projeter des attributs inutiles avant d'appliquer l'opération δ . Nous pouvons également supprimer les doublons dans le cas d'agrégation de type MIN et MAX :

$$\delta(\gamma_L(R)) \equiv \gamma_L(R)$$

$$\gamma_L(R) \equiv \gamma_L(\pi_M(R)); \text{ Si } M \text{ est une liste contenant tous les attributs mentionnés dans } L$$

$$\gamma_L(R) \equiv \gamma_L(\delta(R)); L \text{ est MIN et/ou MAX}$$

17. **Quelques transformations triviales.**

$$\text{Si } S \text{ est vide, alors } R \cup S = R$$

$$\text{Si la condition } c \text{ dans } \sigma_c \text{ est vraie pour tout } R, \text{ alors } \sigma_c(R) = R$$

Exemple 02 : Sur notre exemple de gestion de vente, supposons que nous voulons savoir pour chaque client la date de commande la plus récente. Nous pouvons exprimer cette requête comme :

```

1 SELECT Nom, MAX(Date_Com)
2 FROM Clients, Commandes
3 WHERE Clients.ClientsID = Commandes.ClientsID
4 GROUP BY Nom;

```

Un plan de requête logique initial construit directement à partir de la requête est représenté dans la **Figure I.3** La liste FROM est exprimée par un produit cartésien et la clause WHERE par une sélection. Le groupement et l'agrégation sont exprimés par l'opérateur. Nous pouvons appliquer les transformations suivantes :

Combiner la sélection et le produit dans une jointure (règle 12).

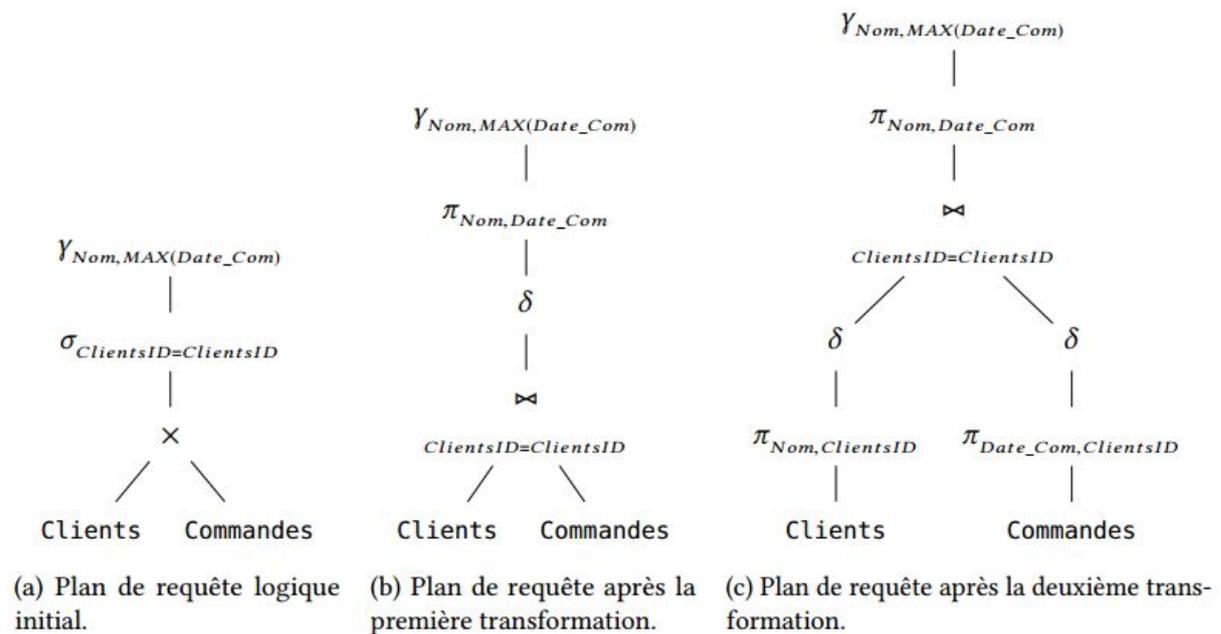


Figure I. 3: Étape de transformation du plan de requête logique pour l'Exemple 02.

2. Générer un δ en dessous du σ (règle 16).
3. Générer un π entre le σ et le δ généré pour projeter sur Nom et Date_Com (règle 16).

Le plan résultant est représenté à la **Figure I.3** Nous pouvons maintenant pousser le δ sous la \Join et introduire π (règle 15).

I.2.2.2 Amélioration du plan de requête logique :

Il existe un certain nombre de règles algébriques qui améliorent les plans de requêtes logiques. Les règles suivantes sont les plus couramment utilisées par les optimiseurs [1,2] :

- Les sélections peuvent être poussées vers le bas dans l'arborescence d'expression dans la mesure du possible. Si une condition de sélection est composée de AND de plusieurs conditions, nous pouvons diviser la condition et pousser chaque partie vers le bas de l'arbre séparément. Cette stratégie est probablement la technique d'amélioration la plus efficace.
- De même, les projections peuvent être poussées vers le bas de l'arbre, ou de nouvelles projections peuvent être ajoutées.
- Les opérations d'éliminations des doubles peuvent parfois être supprimées ou déplacées vers une position plus pratique dans l'arborescence.
- Certaines sélections peuvent être combinées avec un produit ci-dessous pour transformer la paire d'opérations en une jointure, ce qui est généralement beaucoup plus efficace à évaluer que les deux opérations séparément.

Exemple 03 : Considérons l'arbre d'analyse de la Figure II.02. L'expression d'algèbre relationnelle résultante après la transformation de cet arbre est illustrée dans la Figure II.04a. Maintenant, nous voulons appliquer les nouvelles règles de transformation. Tout d'abord, nous pouvons diviser les deux parties de la sélection en $Clients\ ID = ClientsID\ \text{et}\ Methode_Paiement = 'espèces'$. Ce dernier peut être poussé vers le bas de l'arbre, puisque le seul attribut impliqué, $Methode_Paiement$, est de la relation $Commandes$. Ensuite, la sélection et le produit peuvent être combinés dans une seule jointure. L'effet de ces transformations est démontré sur la Figure I.2

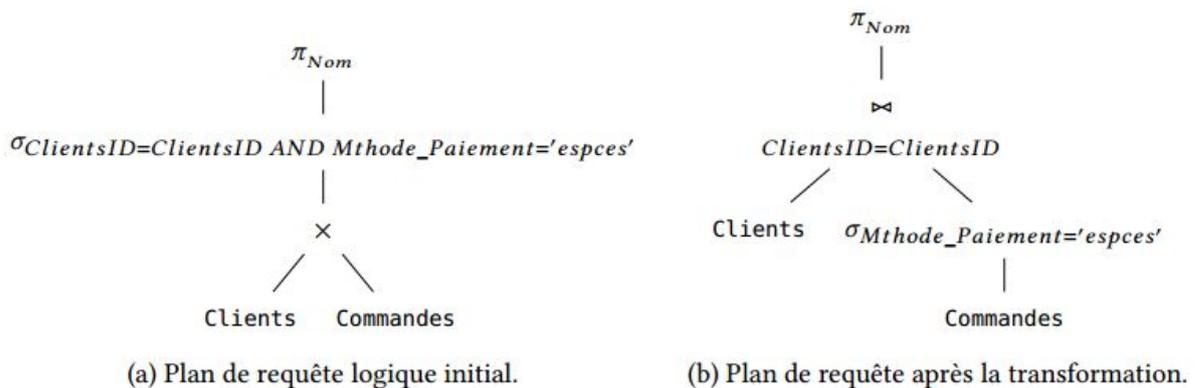


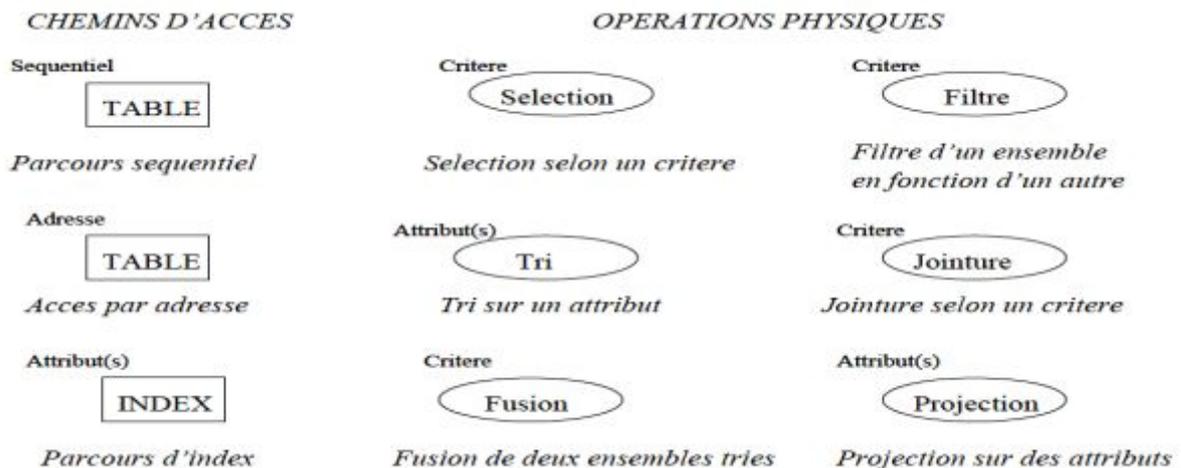
Figure I.4: Transformation du plan de requête logique pour l'Exemple 01.

I.2.3 Génération des plans et optimisation :

Après avoir analysé et transformé une requête en un plan de requête logique, la prochaine étape consiste à transformer le plan logique en un plan physique. Cela se fait normalement en considérant plusieurs plans physiques qui sont dérivés du plan logique, et en évaluant ou en estimant le coût de chacun. Après cette évaluation, souvent appelée génération de plans basée sur le modèle coût[5], le plan de requête physique avec le coût le moins estimé est choisi. Ce plan est passé au moteur d'exécution de requêtes[1]. La **Figure I.4** donne un aperçu du processus d'optimisation des requêtes.

I.2.3.1 Algèbre physique :

- Chemins d'accès aux données
- Opérations physiques



Le premier problème est de savoir comment estimer les coûts des plans avec précision. Ces estimations sont basées sur :

(1) des statistiques sur données de la base, et (2) des statistiques de système. Compte tenu des valeurs de ces paramètres, nous pouvons faire un certain nombre d'estimations raisonnables de la taille des relations qui peuvent être utilisés pour prédire le coût d'un plan physique complet.

I.2.3.2 Statistiques sur les données :

Étant donné un arbre d'opérateur d'une requête, il est d'une importance fondamentale d'être en mesure d'évaluer avec précision et efficacité son coût. L'estimation des coûts doit être précise parce que l'optimisation est seulement aussi bonne que ses estimations de coûts. L'estimation des coûts doit être efficace car elle est dans la boucle interne de l'optimisation des requêtes et est invoquée à plusieurs reprises[6]. Un SGBD moderne permet généralement à l'utilisateur ou à l'administrateur de demander explicitement la collecte de statistiques telles que $T(R)$, le nombre de tuples dans une relation R , et $V(R, a)$, le nombre de valeurs différentes dans la colonne de relation R pour l'attribut a . Ces statistiques sont ensuite utilisées dans l'optimisation des requêtes, inchangées jusqu'à la prochaine commande de collecte de statistiques[1]. Généralement, les statistiques sont sauvegardées dans la base de données comme relations appelées catalogue ou dictionnaire de données[2].

En balayant entièrement la relation R , il est facile de compter le nombre de tuples $T(R)$ et aussi de découvrir le nombre de valeurs $V(R; a)$ différentes pour chaque attribut a . De plus, un SGBD peut calculer un histogramme de distribution des valeurs d'un attribut donné. Si $V(R;A)$ n'est pas trop grand, l'histogramme peut être constitué du nombre (ou fraction) des tuples ayant chacune des valeurs de l'attribut a . Les histogrammes les plus courants sont :

- Largeur égale.
- Hauteur égale.
- Valeurs les plus fréquentes.

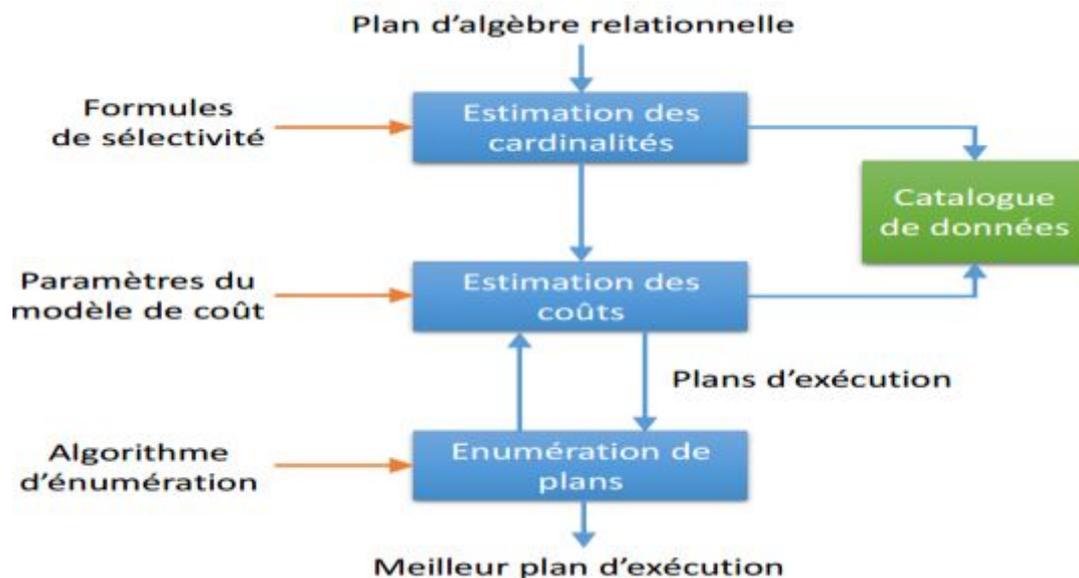


Figure I.5 : Une architecture globale pour l'étape d'optimisation de requêtes.

Les SGBD modernes comme IBM DB2, Informix, Microsoft SQL Server, Oracle et Sybase ASE utilisent la technique d'histogramme pour générer les statistiques sur les données[7].

I.2.3.3 Statistiques de système :

Le coût de l'évaluation des requêtes peut être mesuré en fonction d'un certain nombre de ressources système différentes. Les ressources ou paramètres les plus importants sont les suivants :

— Coût d'accès au stockage secondaire. C'est le coût du transfert (lecture et écriture) des blocs de données entre le stockage de disque secondaire et les tampons de mémoire principale. Cela est également connu sous le nom de coût d'E/S (entrée/sortie) ou CEI S. Le coût de la recherche d'enregistrements dans un fichier disque dépend du type de structures d'accès sur ce fichier et la nature d'allocation des blocs de fichiers.

— Coût de stockage sur disque. C'est le coût de stockage sur disque de tous les fichiers intermédiaires générés par une stratégie d'exécution de la requête.

— Coût de calcul. C'est le coût d'exécution des opérations sur les enregistrements en mémoire tampon pendant l'exécution de la requête. Ces opérations comprennent la recherche, le tri, la fusion d'enregistrements pour une opération de jointure ou de tri et l'exécution de calculs sur les valeurs. Cela est également connu comme le coût de ¹CPU (unité centrale de traitement) ou CCPU.

¹:Central Processing Unit.

— Coût d'utilisation de la mémoire. C'est le coût correspondant au nombre de tampons mémoire nécessaires pendant l'exécution de la requête.

— Coût de la communication. Il s'agit du coût d'expédition de la requête et de ses résultats à partir du site de base de données vers le site ou le terminal d'où provient la requête. Dans les bases de données distribuées, il inclut également le coût du transfert des tables et des résultats entre différents ordinateurs au cours de l'évaluation des requêtes. Le coût de communication est abrégé CCOM.

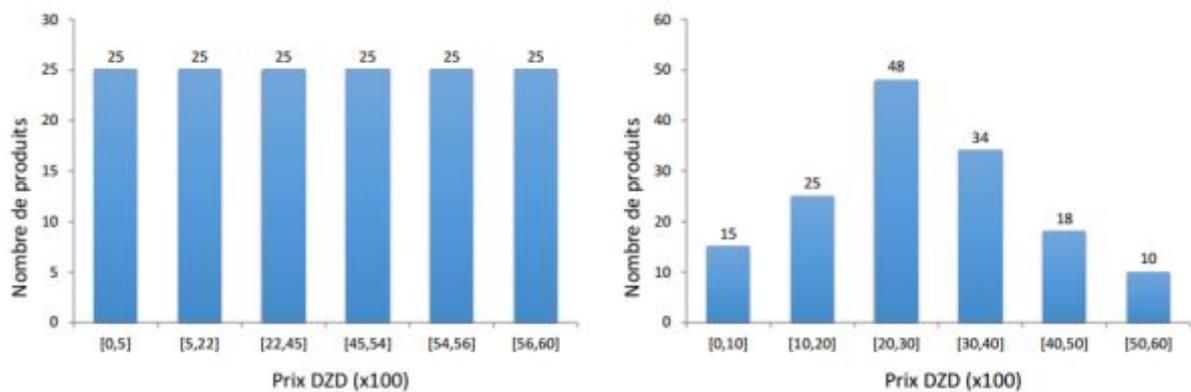


Figure I.6: Exemples des histogrammes de distribution des valeurs d'attribut Prix de la table Produit.

Conclusion :

Dans ce chapitre nous avons examiné le processus de traitement de requêtes dans un SGBD. Nous discuterons des techniques utilisées en interne par un SGBD pour traiter des requêtes de haut niveau. Une requête exprimée dans un langage de requête de haut niveau comme SQL doit d'abord être analysée syntaxiquement. L'analyseur identifie les mots clés de la requête qui apparaît dans le texte de la requête, et vérifie la syntaxe de requête pour déterminer si elle est formulée selon les règles de syntaxe du langage de requête. La requête doit également être validée en vérifiant que tous les noms d'attributs et de relations sont des noms valides et sémantiquement significatifs dans le schéma de la base de données. En utilisant des règles de transformation, une représentation interne de la requête est ensuite créée, habituellement sous la forme d'une structure de données arborescente appelée arbre algébrique.

Un plan de requête logique initial construit directement à partir de l'arbre d'analyse que nous pouvons transformer plusieurs fois en respectant les règles de transformation, après l'étape de transformation une autre phase sera suivie afin d'améliorer le plan de requête logique par l'utilisation de certains nombres de règles algébriques. La prochaine étape consiste à transformer le plan logique en un plan physique et évaluer le coût de chaque phase de

transformation, après cette évaluation le plan de requête physique avec le cout le moins estimé est choisi et puis ce plan passé au moteur d'exécution de requêtes.

La dernière étape du processus est l'exécution de la requête ou toutes les opérations indiquées au plan physique sont exécutées. Avec la fin d'exécution de la requête, nous sommes arrivés à la fin de ce chapitre et aussi la clôture de la partie état de l'art, la partie suivante consiste à déterminer et d'expliquer la méthodologie de conception d'un module d'apprentissage de requêtes en langage SQL.

Chapitre II

Généralités sur l'Apprentissage

« Le savoir est la seule matière qui s'accroît quand on la partage » **Socrate**

Chapitre II : Généralités sur l'apprentissage.

II.1 Introduction :

Le E-Learning est un mode de formation issu des nouvelles technologies de l'information et de la communication dont nous retiendrons la définition suivante : "Utilisation des nouvelles technologies multimédia et de l'Internet pour améliorer la qualité de l'apprentissage" [EDU,08] .

Son objectif est de créer un environnement plus **attractif** et plus **interactif**, avec lequel il est possible d'enseigner et de s'engager plus profond en pédagogie , L'évaluation c'est un acte important, c'est un moyen de suivre les progrès : une formation a des objectifs à atteindre, en termes de transmission de *savoir*(connaissances), *savoir-faire* (pratiques), *savoir-être* (attitudes), l'évaluation permet donc de situer les apprenants vis-à-vis de ces objectifs. Cela permet de motiver les apprenants, de leur faire prendre conscience qu'ils ont besoin de fournir un effort, de leur montrer qu'ils se sont améliorés, mais cela permet aussi au Tuteur de se remettre en question, d'adapter la formation (forme et contenu).

L'enseignant est intéressé par l'évaluation pour trois raisons :

- Savoir si les objectifs sont atteints à la fin d'une session
- Utiliser l'évaluation continue en cours de session comme moyen de régulation
- Tirer des conclusions pour améliorer sa pratique.

Lorsque l'on conçoit une évaluation, il est donc primordial de savoir quels sont les objectifs de la formation, afin de pouvoir déterminer lesquels seront évalués.[8]

L'objectif de ce chapitre est de définir la notion de **l'apprentissage, suivi et évaluation.**

II.2 Concept et terminologie :

II.2.1 Pourquoi faire du e-Learning ? [9]

Le but principal est bien d'améliorer la qualité de l'apprentissage et non de se substituer aux modes d'apprentissages traditionnels.

II.2.2 Définition de l'apprentissage :

Le E-Learning le nom donné actuellement à une phase importante de l'introduction des Technologies de l'information et de la Communication dans la formation. Il s'agit d'une évolution rapide des Technologies pour l'apprentissage, rendue possible par le développement très rapide et planétaire de l'Internet [10] .

II.2.3 Les objectifs d'apprentissage[ELE 08] :

La formulation des objectifs d'apprentissage permet de déterminer les connaissances à acquérir et les compétences à développer par les étudiants au terme d'une activité d'apprentissage. Elle intervient dès le début du processus de planification d'un cours. Par

souci de cohérence, le choix des évaluations, des activités d'enseignement-apprentissage et l'organisation du contenu du cours découlent des objectifs d'apprentissage.

Les objectifs d'apprentissage facilitent la communication entre l'enseignant et les étudiants, puisqu'ils précisent *les apprentissages à faire, les actions à accomplir* ou *les performances à atteindre*. L'étudiant sait précisément ce qui est attendu de lui[8,10].

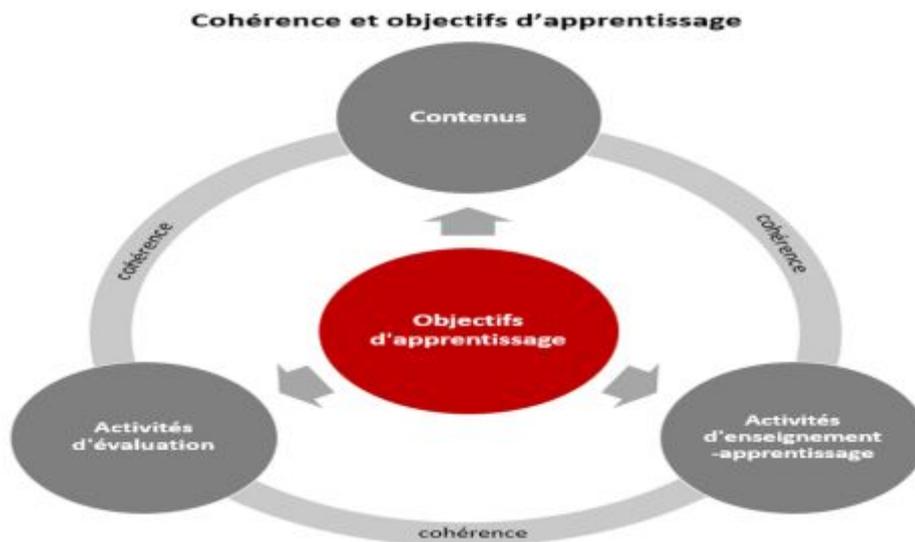


Figure II.1 : Cohérence et objectifs d'apprentissage.

II .2 .4 Triangle pédagogique :

Le triangle pédagogique, également appelé triangle didactique, une représentation graphique d'un modèle qui fonde les relations entre *apprenant, enseignant et savoir*

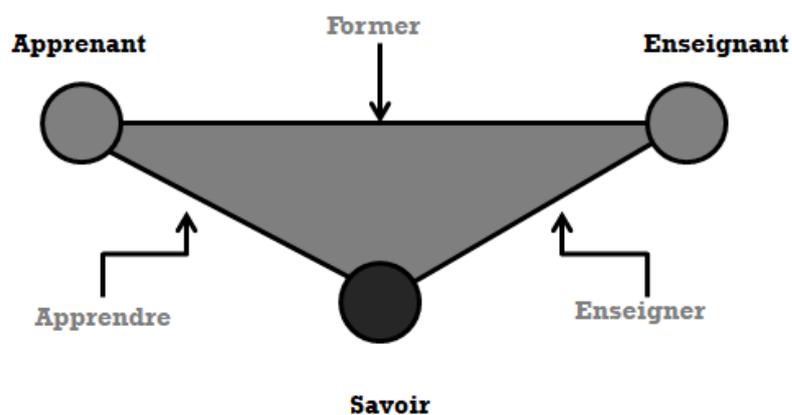


Figure II.2: Triangle pédagogique.

Ce modèle explique les situations pédagogiques du modèle présentiel où les apprenant et enseignant se trouvent en présence l'un de l'autre, face à face, dans un même lieu et au même moment (*amphithéâtre, laboratoire de travaux pratiques, salle de classe...etc.*)[11].

Le processus «apprendre» : (relations apprenant - savoir) : Décrit la phase réflexive

Le processus « enseigner » : (relations Enseignant - savoir) : Pédagogie traditionnelle, centrée sur la transmission des connaissances. L'enseignant transpose les savoirs dans l'intention de les expliquer et de les mettre le mieux possible à la portée des apprenants.

Le processus « former » : (relations enseignant-apprenant) : Avantage la relation duale enseignant-apprenant. Il donne une place prédominante aux composantes affectives et psychologiques. La relation humaine y est au premier plan. de l'apprenant face aux connaissances. Il intervient à un moment ou à un autre dans toute situation pédagogique lorsque les connaissances doivent être intériorisées.

II. 2. 5 Les acteurs et leurs rôles :

Dans une plateforme d'apprentissage à distance, on distingue trois types d'acteurs Principaux : **l'apprenant, l'enseignant, et l'administrateur.**

L'enseignant :

Crée les séances ainsi que des quiz, consulte les cours. Il est à la disposition des apprenants afin de répondre aux questions de ces derniers [12 ,13]. Son rôle est :

- crée des parcours pédagogiques types et individualisés de son enseignement.
- incorpore des ressources pédagogiques multimédias.
- suit les activités des apprenants.

L'apprenant :

C'est l'axe de l'apprentissage [12], son rôle est:

- consulte en ligne ou télécharge les contenus pédagogiques qui lui sont recommandés.
- organise et a une vue de l'évolution de son travail.
- effectue des exercices, s'auto évalue et transmet des travaux à corriger

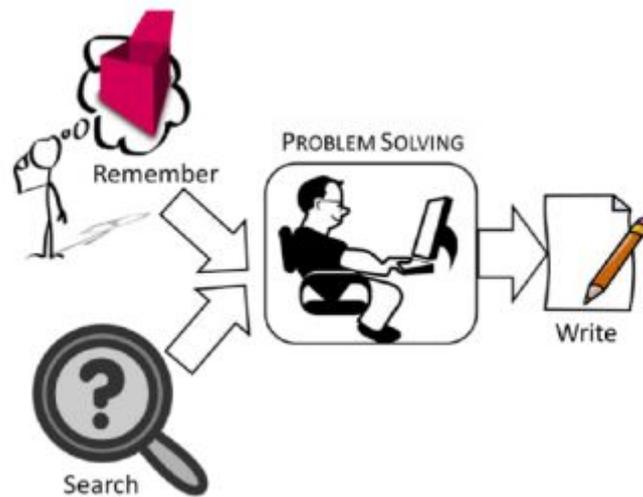


Figure II.3 : illustration : L'espace de l'apprenant.

Formateurs et apprenants communiquent individuellement ou en groupe, créent des thèmes de discussion et peuvent dans certains cas collaborer à des documents communs.

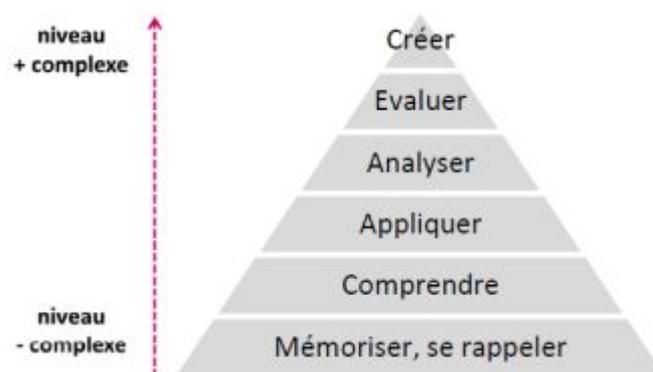
L'administrateur :

L'administrateur, de son côté, assure l'installation et la maintenance du système, gère les droits d'accès, crée des liens vers d'autres systèmes et ressources externes. Ainsi, une plateforme peut comporter des fonctionnalités relatives à la gestion des compétences, à la gestion des ressources pédagogiques, à la gestion de la qualité de la formation, etc.

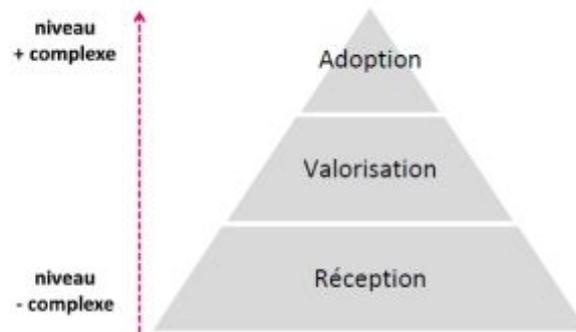
II.3 Taxonomies d'objectifs d'apprentissage :

On présente trois taxonomie de trois types différent :

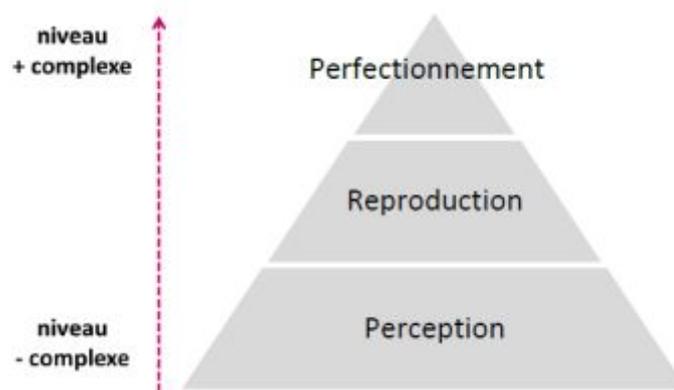
1. [La taxonomie de Bloom](#) (1956) révisée par Krathwohl (2002) peut être utile pour cibler un niveau d'apprentissage de **type cognitif**. [15]



2. Pour les apprentissages de **type affectif**, on peut utiliser [la taxonomie proposée par Berthiaumeet Daele](#)[16].



3. [Berthiaume et Daele](#) proposent aussi une taxonomie pour les apprentissages de **type psychomoteur**[17].



Celle qui s'adapte mieux à notre étude est **la taxonomie de bloom**

II .4 comment les étudiants apprennent: un apprentissage taxonomie ?

L'apprentissage des taxonomies présente un modèle d'un ensemble de niveaux d'engagement cognitif avec le matériel appris. Ces taxonomies sont utilisées dans la conception et l'évaluation du cours pour s'assurer que l'enseignement et l'évaluation trouvent le bon équilibre entre les connaissances (contenu du cours) et les compétences (telles que les synthèses et l'évaluation).

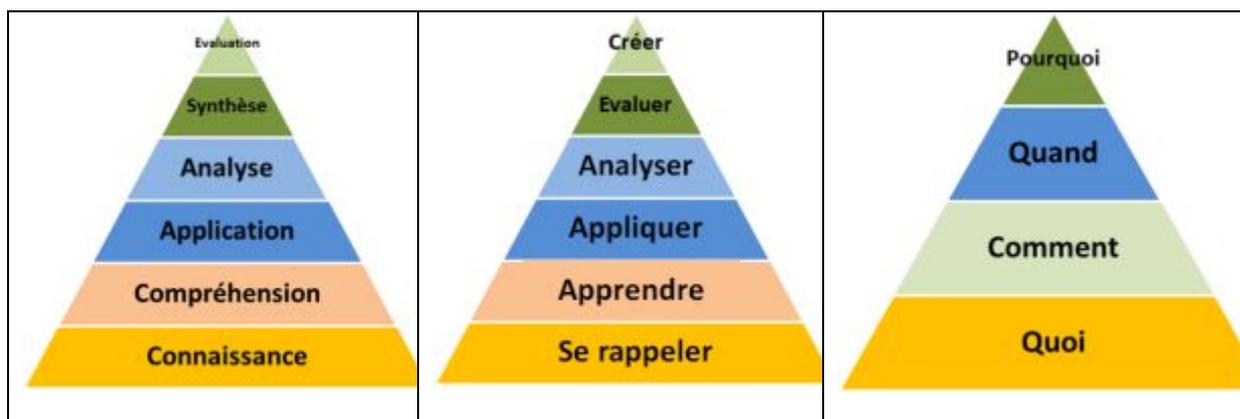


Figure II.4a: Taxonomie de <u>Bloom</u>	Figure II.4b : Taxonomie de <u>Krathwohl</u>	Figure II.4c: Taxonomie de <u>Gorman</u>
Figure II.4 : Les taxonomies d'apprentissage		

Bloom [18] a proposé une taxonomie qui classe les formes d'apprentissage. Il a identifié six niveaux d'apprentissage et soutenu que les niveaux supérieurs ne devraient pas être tentés avant que les niveaux inférieurs aient été maîtrisés. Sa taxonomie est illustrée à la Figure II.4 a

Voici des exemples **de verbes d'action** de **Taxonomie de Krathwohl** qui est illustré dans le tableau

<p>Créer</p> <p>Assembler des éléments pour former un tout nouveau et cohérent, ou faire une production originale</p>	<p>adapter, agencer, anticiper, arranger, assembler, combiner, commenter, composer, concevoir, connecter, construire, créer, développer, écrire,</p>
<p>Evaluer</p> <p>Porter un jugement sur la base de critères et de normes</p>	<p>apprécier, argumenter, attaquer, choisir, conclure, critiquer, défendre, déterminer, estimer, évaluer, juger, justifier, soutenir...</p>
<p>Analyser</p> <p>Décomposer les parties constitutives d'un tout et déterminer les liens qui unissent ces parties entre elles et à une structure ou une finalité d'ensemble</p>	<p>analyser, cibler, comparer, contraster, critiquer, découper, déduire, délimiter, différencier, discriminer, disséquer, distinguer, examiner</p>

<p>Appliquer</p> <p>Exécuter ou utiliser une procédure dans une situation donnée</p>	<p>administrer, appliquer, assembler, calculer, catégoriser, colliger, construire, contrôler, découvrir, démontrer, dessiner, déterminer, employer, établir</p>
<p>Comprendre</p> <p>Construire la signification d'informations reçues (orales, écrites et graphiques)</p>	<p>classer, comparer, convertir, démontrer, différencier, dire dans ses mots, illustrer (à l'aide d'exemples), expliquer, exprimer, faire une analogie</p>
<p>Se rappeler</p> <p>Extraire les connaissances significatives issues de sa mémoire à long-terme</p>	<p>associer, citer, décrire, définir, dupliquer, enregistrer, énumérer, étiqueter, identifier, indiquer, lister, localiser, mémoriser, nommer, ordonner</p>

Tableau II.1 : Verbes d'action de Taxonomie de Krathwohl [18,19].

Gorman a proposé une taxonomie simplifiée, comme le montre la [Figure II.4c](#). Avec seulement quatre niveaux.

Quand on examine ces trois taxonomies, un certain nombre de similitudes émergent.

- Le «*quoi*» de Gorman s'aligne sur les «connaissances» de Bloom et les niveaux «souvenir» d'Andersen et Krathwhol. C'est la connaissance déclarative qui fait référence à la mémoire pour les faits ou les événements.
- Le "*Comment*" de Gorman s'aligne avec la "Compréhension" des autres taxonomies. Ce sont les connaissances procédurales qui sont codées d'abord en termes déclaratifs, puis traduites en procédures[20].
- Le "*Quand*" ou le Jugement de Gorman peut impliquer "Application", et "Analyse".
- "L'évaluation du jugement de tâche implique de reconnaître qu'un problème a des caractéristiques similaires à celles dont le chemin de solution est connu et de savoir quand appliquer une procédure particulière[21].
- Le niveau supérieur de Gorman est "*Pourquoi*", qui aligne les niveaux "Création" d'Anderson et Krathwhol. Selon Gorman[21], «la sagesse est la capacité à réfléchir sur ce que quelqu'un fait et, si nécessaire, à proposer une nouvelle ligne de conduite».
- Cette partie fournit une source abstraite pour les autres parties de la recherche en explorant les différents types de connaissances que les apprenants SQL doivent avoir. Ces différents types d'activités cognitives ou de type de connaissances sont discutés et mentionnés dans différentes parties de cette thèse. **[PRE 01]**

II.5 Taxonomie d'apprentissage en informatique :

Les enseignants en informatique ont appliqué ces taxonomies de la même manière que les autres domaines. Selon Cutts et al[22].

«Les taxonomies d'apprentissage sont importantes pour le calcul de l'éducation parce qu'elles donnent à la communauté un vocabulaire à utiliser pour discuter de la compréhension et de l'apprentissage des apprenants- et du curriculum pour la soutenir»

L'applicabilité de diverses taxonomies d'apprentissage à l'informatique (CS) a été explorée par les chercheurs[23,24] Lahtinen[24] , en particulier, a étudié si une taxonomie spécifique à un sujet serait plus utile aux instructeurs CS que les taxinomiques génériques existants. Il a signalé que les activités cognitives de Bloom étaient effectivement applicables à l'informatique en général.

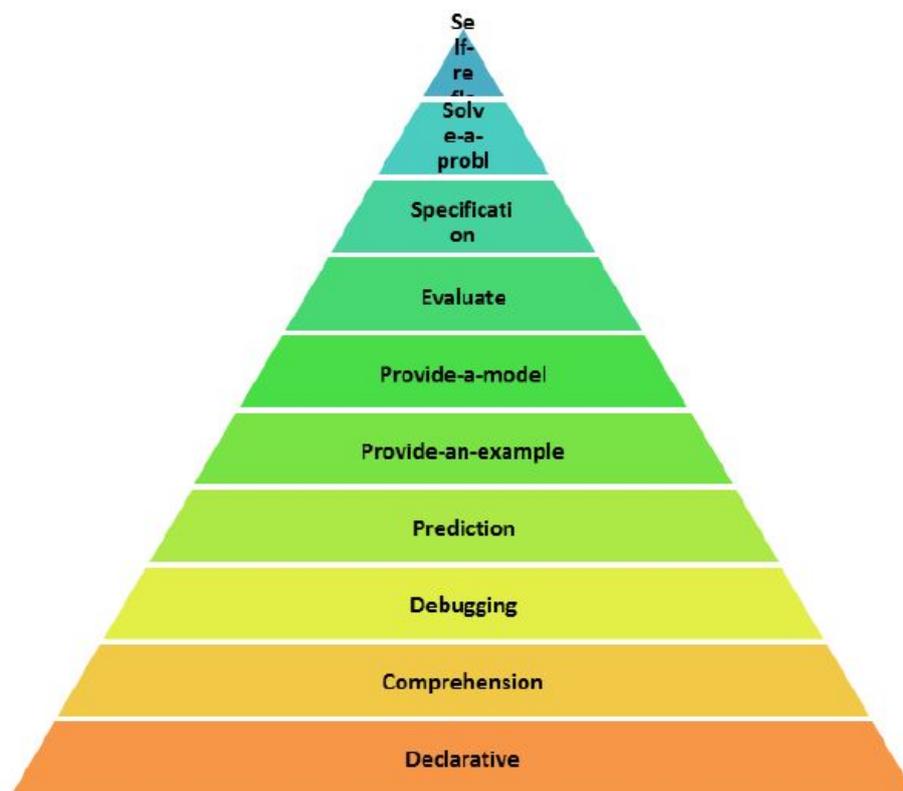


Figure II.5 : Taxonomie des types de tâches dans l'informatique[25]

II.6 Scénarisation Pédagogique :

II.6.1 Définitions de scénario pédagogique :

Charlier et Daele définissent le scénario pédagogique comme : « le résultat du processus de conception d'une activité d'apprentissage, processus s'inscrivant dans un temps donné et aboutissant à la mise en œuvre du scénario. Dans un scénario, on trouve donc *des objectifs, une planification des activités d'apprentissage, un horaire, une description des tâches des*

étudiants, des modalités d'évaluation qui sont définis, agencés et organisés au cours d'un processus de design. »[26].

II.6.2 Autres concepts de base :

II.6.2.1 Définition d'une activité d'apprentissage[WIK, 08] :

Les activités d'apprentissage visent un objectif d'apprentissage précis. Elles sont de courte durée. Leur organisation et leur planification sont sous la responsabilité de l'enseignant. Souvent, tu apprendras la nature des activités d'apprentissage la période même.

Cela peut être des exposés, des mises en situation pour te sensibiliser à une difficulté, des exercices ou des expérimentations. L'utilisation des activités d'apprentissage pour la préparation d'une mission ou un projet. Dans le cas des activités d'apprentissage, il faut définir les règles du jeu en fonction des exigences du travail en groupe [27].

II.6.3 Principes de modélisation des activités pédagogiques en ligne :

Il existe deux principes essentiels qui sont[CUR 08]:

Principe n°1 : Concevoir une activité pédagogique en ligne consiste à scénariser l'utilisation de contenus et de services déjà existants à travers la spécification d'un dispositif interactionnel. Ce principe découle de l'analyse de l'usage que les enseignants font des ENT lorsqu'ils essaient d'adapter les contenus et les services à leurs besoins, d'organiser les échanges, ou de disposer d'un feedback efficace sur les résultats des élèves.

Principe n°2 : Toute activité pédagogique repose sur 4 piliers fondamentaux qui sont :

- ✓ L'organisation : préparation, mise en place des moyens nécessaires et mobilisation des acteurs impliqués dans l'activité;
- ✓ L'apprentissage_ : ensemble d'activités à finalité didactique, expérimentale, productive ou d'entraînement qui déterminent l'objet, la stratégie et le niveau des apprentissages.
- ✓ L'observation_ : activité qui, de façon concomitante à l'apprentissage, permet à l'enseignant d'en apprécier le déroulement, de vérifier la bonne exécution des travaux, de détecter les difficultés éventuelles rencontrées par les élèves;
- ✓ L'évaluation : activité dans laquelle les effets des apprentissages sont constatés, ou dans laquelle les besoins des apprenants sont diagnostiqués [28].

II.6.4 Les trois niveaux de scénarisation :

Il définit les trois niveaux de scénarisation suivants :

- Scénariser la structuration pédagogique : organiser les unités de structuration pédagogique telles que les cours, les modules, les leçons, les cursus, les crédits.
- Scénariser l'enchaînement d'activités : organiser les activités d'un apprenant ou d'un groupe d'apprenants au sein d'une séquence pédagogique, d'une séance, etc.

- Scénariser la réalisation de tâche : décrire le déroulement d'une tâche précise (comme par exemple parcourir un hyper-document, résoudre un exercice, manipuler une simulation).[29]

II.7 Evaluation et suivi de l'apprenant :

Le **suivi de l'apprenant** ainsi que son *évaluation* sont des éléments importants dans l'apprentissage pour juger sa progression en direction des objectifs fixés et des résultats attendus. Ils se font en cours et à la fin de la session d'apprentissage. En effet, le suivi pédagogique permet aux enseignants de connaître toutes les activités effectuées par l'apprenant durant son apprentissage et de récupérer tous les résultats sur ses activités afin de les analyser. Pour un meilleur suivi de l'apprenant, il est intéressant de l'évaluer d'une manière systématique et continue qui est l'objectif de notre travail[30].

Cela permet *d'analyser* ses progrès dans le temps. Donc le système d'apprentissage doit disposer des données nécessaires pour effectuer une telle tâche. Ces données sont collectées à partir de ses évaluations à travers des tests, proposés par les enseignants permettant à ces derniers de tester et d'évaluer la compréhension et **les connaissances** des apprenants afin de savoir si l'objectif de la formation est atteint.

Dans notre étude on va s'intéresser au suivi d'activité collaborative et les principales tâches des évaluateurs qui sont : *La correction des évaluations, l'attribution des notes, l'analyse des résultats des apprenants* et la fourniture des résultats aux autres acteurs de la formation, en particulier pour les apprenants et leurs tuteurs[31].

II.7.1 Le suivi :

Le suivi individuel d'un apprenant engagé dans une activité collective doit mettre en évidence l'importance de prendre en compte la dimension individuelle dans une activité de groupe, d'une part pour animer l'activité collective, et d'autre part, pour évaluer individuellement chaque apprenant. Le tuteur a besoin de percevoir l'implication de l'apprenant dans la vie du groupe, en s'assurant de l'adhésion de chacun au groupe. Aussi, doit être en mesure de détecter l'état de motivation de chaque apprenant, en se basant sur sa participation mais aussi sur sa non-participation aux discussions. On distingue deux types de suivi : l'une concerne les apprenants et l'autre le déroulement de l'activité d'apprentissage[XPER, 08] .

Le suivi individuel concerne la situation dans laquelle chaque apprenant suivi a un parcours de formation individualisé et adapté à ses besoins de formation et à sa disponibilité. Il permet de définir un cadre pour mieux orienter les apprenants vers les activités à réaliser (Voir figure II.6).

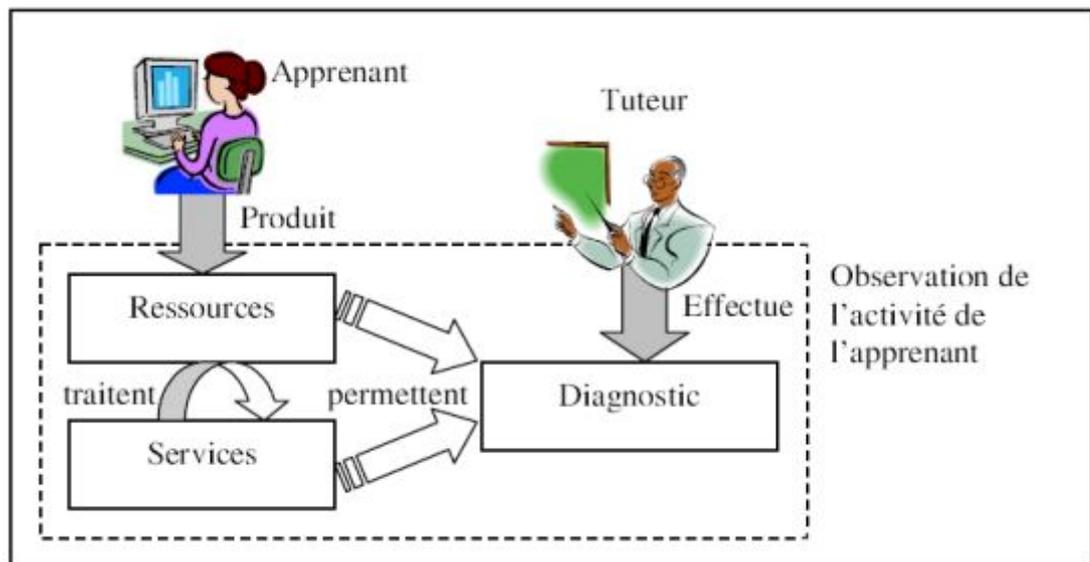


Figure II.6: Suivi d'un apprenant.

II.7.2 L'évaluation :

L'évaluation est un aspect fondamental et incontournable de l'acte pédagogique. Il est en effet crucial que l'enseignant, au cours de son enseignement, puisse évaluer ce que ses apprenants ont assimilé ou non ; de la même manière, il est important que l'apprenant, au cours de son apprentissage, puisse évaluer ses connaissances. Divers types ou formes d'évaluation peuvent exister, les plus utilisées sont : l'évaluation formative, et l'évaluation sommative. Le déroulement comme nous allons expliquer dans la suite peut se faire durant, pendant et à la fin de l'acte de l'apprentissage pour assurer la continuité du suivi[30,31].

II.8 Positionnement de notre solution :

- Département d'informatique .
- Travaux pratiques (base de données).
- Observation.
- Traçabilité.

II .9 Conclusion :

Dans ce chapitre nous avons constaté que il est important d'orienter et de suivre pas à pas l'apprenant, afin de lui transmettre les bonnes connaissances dont il a besoin, tous en prenant mieux en compte de ses connaissances, ses compétences, ses difficultés, ses points forts et ses faiblesses, etc. Ceci constitue le profil de l'apprenant.

Au plus il s'avère important de récolter les informations les plus fines sur le déroulement de chaque session d'apprentissage afin de vérifier l'évolution des connaissances de

l'apprenant et principalement pour mettre en œuvre les mécanismes nécessaires pour l'adaptation de son parcours pédagogique. Une telle appréciation des difficultés et des facilités de l'apprenant s'inscrit dans une démarche d'évaluation.

L'évaluation est avant tout une activité pédagogique. Elle a pour rôle d'interpréter, de juger et contrôler la production de l'apprenant durant son apprentissage. Elle est censée intervenir dans toutes les étapes du cycle de vie du processus d'apprentissage.

Chapitre III

Analyse et conception de notre
solution

Chapitre III : Analyse et conception de notre solution**III.1.Introduction :**

Le suivi de l'apprenant ainsi que son évaluation sont des éléments importants dans L'apprentissage pour juger sa progression en direction des objectifs fixés et des résultats attendus. Ils se font en cours et à la fin de la session d'apprentissage. En effet, le suivi pédagogique permet aux enseignants de connaître toutes les activités effectuées par l'apprenant durant son apprentissage et de récupérer tous les résultats sur ses activités afin de les analyser. Pour un meilleur suivi de l'apprenant, il est intéressant de l'évaluer d'une manière systématique et continue qui est l'objectif de notre travail. Cela permet d'analyser ses progrès dans le temps. Donc le système d'apprentissage doit disposer des données nécessaires pour effectuer une telle tâche. Ces données sont collectées à partir de se évaluations à travers des tests, proposés par les enseignants permettant à ces derniers de tester et d'évaluer la compréhension et les connaissances des apprenants afin de savoir si l'objectif de la formation est atteint. Dans notre étude on va s'intéresser au suivi d'activité collaborative et les principales tâches des évaluateurs qui sont : La correction des évaluations, l'attribution des notes, l'analyse des résultats des apprenants et la fourniture des résultats aux autres acteurs de la formation, en particulier pour les apprenants et leurs tuteurs.

III.2 Analyse de l'existant :

Afin de réaliser notre projet, nous avons suit une démarche ,Cette démarche est constituée de quatre phases :

- ✓ Système existant.
- ✓ Analyse de système.
- ✓ Modélisation de système Implémentation et mise ne ouvre

III.2.1 Méthodes d'analyse :

Il existe plusieurs méthodes d'analyse des besoins et d'études des données opérationnelles (Voir FigureIII.1) notamment :

• Les entretiens :

L'entretien est une technique destinée à recueillir des informations à l'aide de questions posées verbalement par l'analyste. Les entretiens sont menés soit individuellement, ou par petits groupes. C'est sans doute la technique la plus utilisée.

• Les réunions :

Cette technique consiste à planifier des réunions en groupes importants, dirigées par un animateur. Ces réunions peuvent favoriser une libre expression créative.

- **Elaboration Le questionnaire :**

Un questionnaire permet d'interroger directement des individus en définissant au préalable, par une approche qualitative, les modalités de réponses au travers des questions dites " fermées ". C'est un outil de collecte de données primaires bien adapté aux recherches quantitatives puisqu'il permet de traiter de grands échantillons et d'établir des relations statistiques ou des comparaisons chiffrées.

Les questionnaires peuvent remplacer avantageusement les entretiens. Toutefois ils ne conviennent pas à toutes les situations. Il faut les concevoir avec beaucoup de soin pour un maximum d'efficacité.

Les questionnaires peuvent être la seule méthode pratique pour entrer en contact avec un grand nombre de personnes. Bien sûr il n'est pas possible de prévoir les réactions ou les expressions des personnes.

- **L'observation :**

L'observation est un mode de collecte des données par lequel le chercheur observe de lui-même des processus ou des comportements se déroulant dans une organisation, pendant une période de temps délimité. L'observation constitue un mode de recueil alternatif de l'entretien. Mais cette méthode prend beaucoup de tem

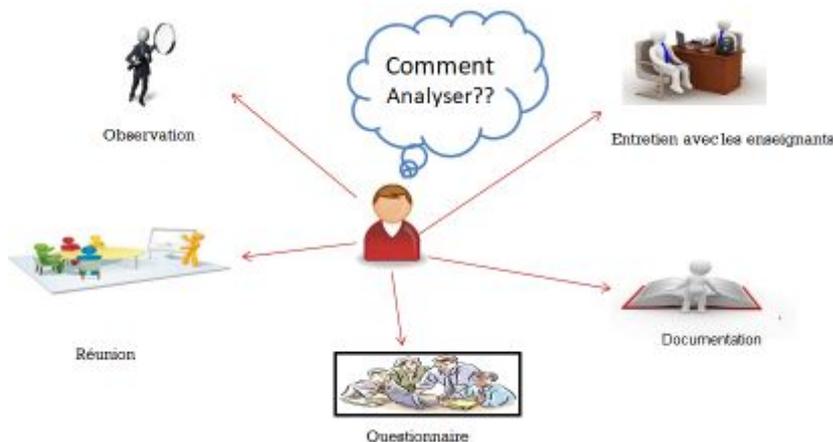


Figure III.1 : Les méthodes d'analyse.

III.2.2 Justification :

Afin de faire une étude aussi complète que possible, nous avons choisi d'adopter une démarche qui nous a permis de combiner, d'une manière assez satisfaisante, entre l'approche

Après l'étude des différentes approches de collecte d'informations, notre choix s'est porté sur l'interview, compte tenu de ses multiples avantages, néanmoins cette méthode présente certains inconvénients dont notamment la difficulté d'interviewer un grand nombre de personnes à cause de leur contrainte d'emploi du temps. Pour palier à cette contrainte nous avons eu recours au questionnaire et à l'observation.

III.2.3 Les Anomalies :

Dans cette section, nous allons présenter les anomalies liées au l'existant (situation actuelle de TP dans notre département) avec les causes et les conséquences. Nous allons essayer de citer celles que nous avons constatées durant notre étude. Nous citerons :

III.2.3.1 Anomalies informationnelles :

ANOMALIE 01 : pas d'assistance au moment de formulation de requête SQL (Travail manuel).

Causes	Conséquences
1. Manque d'un outil informatique. 2. Cours /TD/TP seulement présentiel.	1. Avoir des erreurs syntaxiques /sémantiques. 2. Problème de vérification de ces erreurs.

Tableau III.1 : Pas d'assistance au moment de formulation de requête SQL (Travail manuel).

ANOMALIE 02 : Difficulté de valorisation de l'effort de l'apprenant durant la séance de travaux pratique.

Causes	Conséquences
1. Nombre important des étudiants concerne par le TP/TD 2. Difficulté pour quantifier les efforts.	1. Evaluation incorrect difficulté d'apprécier quantitativement et qualitativement le travail d'étudiant sans savoir d'explorer la trace détaillée.

Tableau.III.2: Difficulté de valorisation de l'effort de l'apprenant durant la séance de travaux pratiques.

ANOMALIE 03 : Erreur fréquente et difficulté de comprendre comment restituer la table résultat de SQL.

Causes	Conséquences
1. Manque du support qui sert à aider l'étudiant pour améliorer cette compétence.	1. Pas d'avancement dans les études

Tableau III.3: ERREUR FREQUENTE ET DIFFICULTE DE COMPRENDRE COMMENT RESTITUER LA TABLE RESULTAT DE SQL.

III.2.3.2 Anomalies organisationnelles :

ANOMALIE 01 : Problème de traçabilité

Causes	Conséquences
1. Nombre important des étudiant. 2. La durée du TP est courte et insuffisante. 3. Absence d'outil de surveillance/journalisation. 4. Communication entre le prof et l'étudiant inefficace.	1. Difficulté de savoir qui fais quoi et à quel moment. 2. Difficulté de suivre chaqu'un individuellement.

Tableau III.4: Problèmes de traçabilité.

III.2.3.3 Anomalies techniques :

ANOMALIE 01 : Déroulement de travaux pratique(TP) de manière manuelle.

Causes	Conséquences
1. Manque de support informatique. 2. Manque de logiciels destiner pour les travaux pratiques.	1. Perte de temps. 2. Risque de perte d'informations. 3. Perte du temps et d'effort dans le contrôle et la vérification. 4. Non-utilisation de matériels.

Tableau III.5: Déroulement de travaux pratique(TP) de manière manuelle.

ANOMALIE 2 : Manque d'un espace informatique spécialise pour le TP.

Causes	Conséquences
1. Transmission de certaine information d'une manière informelle (verbalement).	1. Risque de perte d'information. 3. Risque d'erreur de compréhension de l'information transmet. 4. Beaucoup d'effort. 5. Problème dans la durée du TP (insuffisante).

Tableau III.6: Manque d'un espace informatique spécialise pour le TP.

La figure suivante illustre la gestion actuelle de l'enseignant(Voir figure III.2.3).

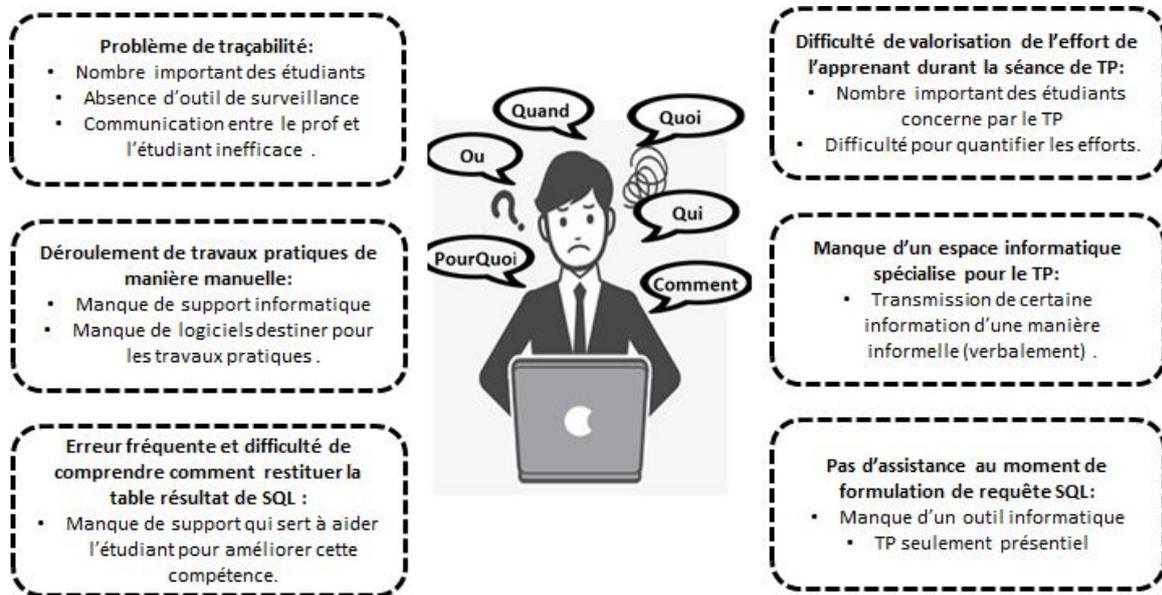


Figure III.2: Illustration de gestion actuelle de l'enseignant chargé des TP.

III.2.4 Solutions proposées :

Pour régler les problèmes cités précédemment, nous proposons les solutions suivantes :

III.2.4.1 Solutions d'ordre informationnel :

Nous pouvons suggérer les solutions suivantes :

- Créer une application destinée pour faire assister les TP.
- Automatiser les tâches manuelles.
- Vérifier au temps de saisie de requête dans application.
- Faire des statistiques pour aider à évaluer l'étudiant.

III.2.4.2 Solutions d'ordre organisationnel :

Aider l'étudiant avec des cours (définition, explication, exemple...).

Utilités des documents numériques.

III.2.4.3 Solutions d'ordre technique :

Permettre au professeur :

- D'accéder au compte de tous les étudiants.
- Créer un compte pour chaque étudiant.
- Archiver une copie de travail de l'étudiant.

III.2.5 Système Projeté :

Cette figure montre une vue globale de notre solution proposée (figure III.2.5), on a un ensemble des étudiants qui se connecte à travers une plateforme pour formuler sa requête dans un éditeur informatique, avec la possibilité de vérifier la syntaxe et la sémantique. Quand il s'agit d'une erreur on peut l'aider avec des directives pour améliorer sa requête, toutes ces informations sont stockées dans une base de données qui sera exploitée par l'enseignant pour la visualisation des données graphiques.

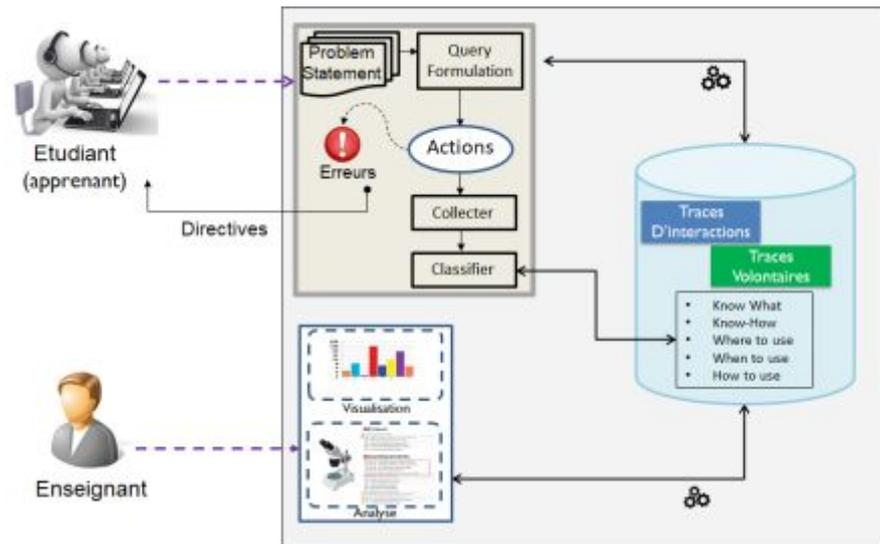


Figure III.3 : Système projeté.

III.2.6 Fondement Théorique:

III.2.6.1 Application de taxonomie de BLOOM :

Pour concevoir notre application nous avons basé sur la taxonomie de BLOOM qui a été défini dans *le Chapitre II* Nous avons proposé un modèle pédagogique appelé "Enseignement SQL basé sur la taxonomie de bloom".

Scheidemann[32] se concentrait sur les aspects liés aux facteurs humains des interactions avec les bases de données et sur la façon de faciliter l'utilisation du langage d'interrogation. Cependant, dans cette recherche, l'accent est mis sur la façon de faciliter l'apprentissage de SQL. En regardant les cinq tâches mentionnées, il était possible de les relier à la théorie de l'apprentissage et de les organiser en un niveau de taxonomie d'apprentissage (taxonomie de bloom)

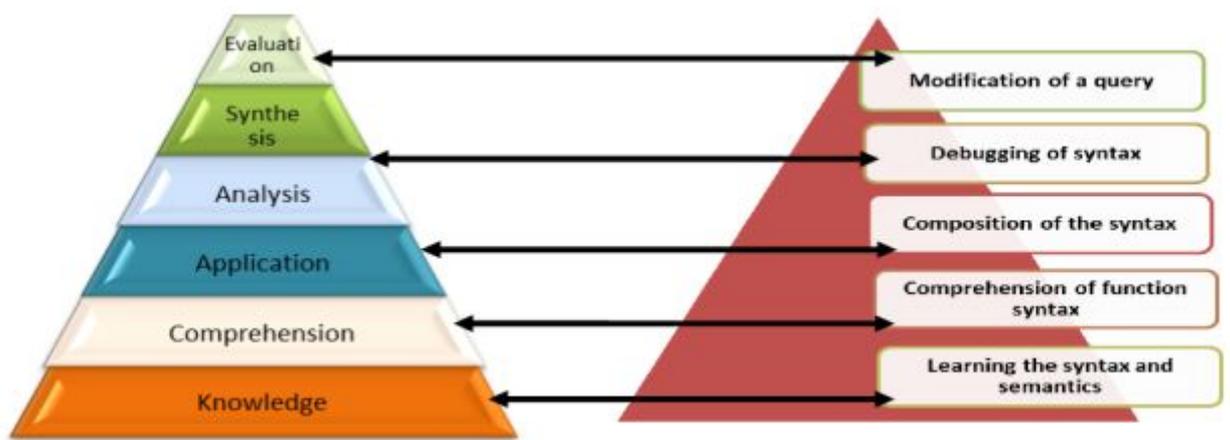


Figure III.4a : Taxonomie de BLOOM

Figure III.4b : Cinq tâches de Scheidemann.

Figure III.4 : Les cinq tâches de Scheiderman et la tâche connexe dans l'apprentissage de la taxonomie

Scheidemann[7] a mis en évidence cinq tâches que l'on peut appliquer pour fournir une requête pour récupérer des informations pour la base de données. Voici les cinq tâches:

1. **Le niveau Connaissances** correspond à *Apprendre la syntaxe et la sémantique* de la spécification de la fonction. Il soutient qu'un objectif typique à ce stade est de réduire le temps d'apprentissage.

2. **Le niveau Compréhension** correspond à *Composition de la syntaxe requise* pour exécuter la fonction requise. La composition comprend l'écriture d'une requête ou la formulation d'une requête en langage naturel.

3. **Le niveau Application** correspond à *Compréhension de la syntaxe de fonction* composée par quelqu'un d'autre. Il est souvent nécessaire de lire la syntaxe composée par d'autres pour apprendre ou autre fins. La syntaxe facilement composée peut ne pas être facile à comprendre. La compréhension est souvent une composante des autres.

4. **Le niveau Analyse et Synthèse** correspond à *Débogage de la syntaxe ou de la sémantique* écrite par d'autres ou par les utilisateurs. L'objectif principal du débogage est de corriger les erreurs. Scheidemann a déclaré que le débogage nécessite la compréhension et la capacité de composition, mais inclut d'autres compétences cognitives complexes. Il a suggéré que le débogage du langage de requête nécessitera de nouvelles stratégies de débogage. Le problème central sera de fournir aux utilisateurs un retour d'information pour les aider à déterminer si la sémantique de la fonction qu'ils invoquent correspond à leurs intentions.

5. **Le niveau Evaluation** correspond à *Modification d'une requête écrite par soi-même ou par d'autres*. Les requêtes de base de données existantes seront souvent la base de nouvelles requêtes. Cette tâche nécessite également des compétences de composition et de compréhension.

III.2.6.2 Classification des erreurs :

Nous avons classé les types d’erreurs relatives ou tentatives de formulation de requête SQL Dans la figure III.5 on a :

- **Erreurs d’analyse** : qui montre la difficulté chez l’apprenant à analyser le *fonctionnement des opérateurs* et savoir *l’ordre des opérateurs*.
- **Erreurs syntaxiques** : générer si la requête SQL est non conforme à la **grammaire SQL**.
- **Erreurs sémantiques** : qui signifie que la requête SQL est correcte syntaxiquement mais le résultat retourner par cette requête est fausse comme par exemple référencier d’un attribut ou bien une table qui n’appartient pas au dictionnaire de base de données.

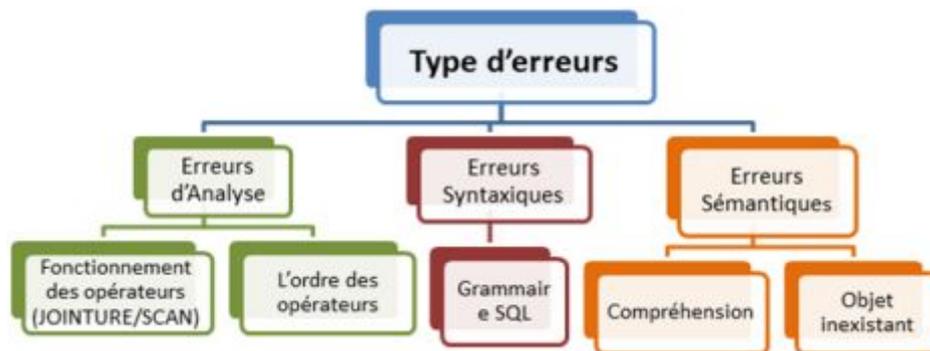


Figure III.5: Organigramme du type d’erreurs.

III.2.6.3 Directives de formulation SQL :

La catégorie des erreurs donne la possibilité de classer chaque tentative de l’apprenant dans une catégorie. Afin d’aider l’apprenant et augmenter l’interactivité avec notre système proposé, Nous avons proposé un algorithme qui affiche des directives (HINT en anglais). Ces directives sont affichés selon le type d’erreurs et le nombre de tentative.(voir tableau 07)

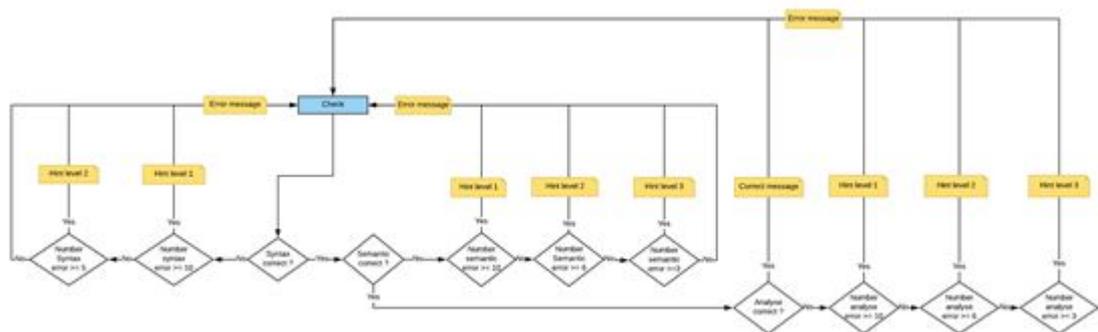


Figure III.6 : Directives de formulation des requêtes SQL.

Niveau de directives	Description
Niveau 1	Directives en langage SQL
Niveau 2	Directives en langage naturel
Niveau 3	Directives contient le type d'erreur

Tableau III.7 : Les niveaux de directives.

III.2.6.4 Traçabilité :

Rappelons que notre objectif est suivi le comportement de l'apprenant durant les séances de travaux pratiques. Pour atteindre cet objectif notre système permet d'acquérir :

- (i) Les entrées utilisateurs (par ex. texte de requête saisie)
- (ii) Les actions volontaires (par ex. tentative)
- (iii) Les actions involontaires
- (iv) Le parcours de support de cours

Ces données collectées sont utilisés pour représenter sous forme graphiques les tentatives, type d'erreurs, comportement et la progression.

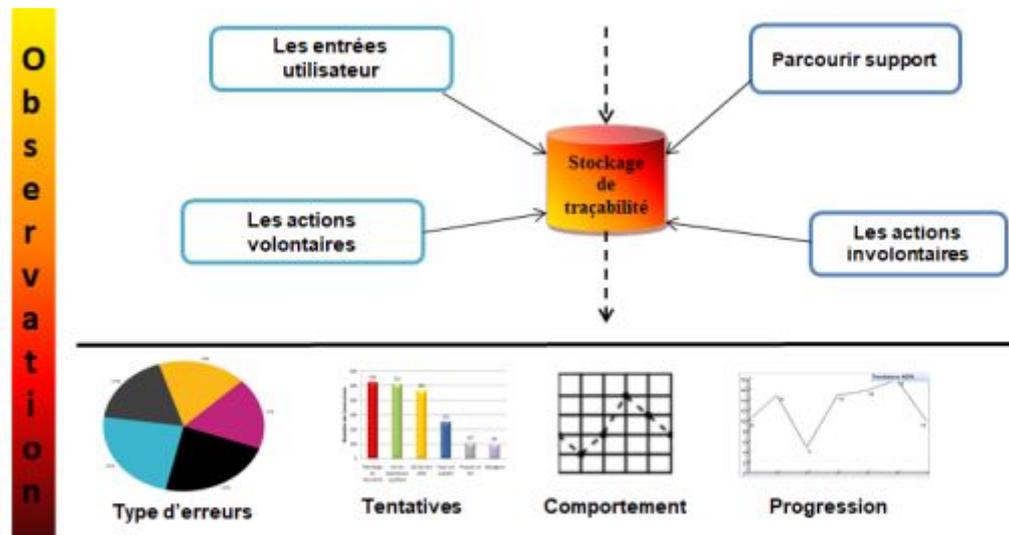


Figure III.7 : Traçabilité des utilisateurs.

III.3 Etude conceptuelle:

L'objectif de l'étude conceptuelle, est de concevoir le nouveau système, pour cela nous avons le langage UML pour la modélisation.

Le langage UML se définit comme suit :

« UML se définit comme un langage de modélisation graphique et textuel destiné à comprendre et décrire des besoins, spécifier et documenter des systèmes, esquisser des architectures logicielles, Concevoir des solutions et communiquer des points de vue. » [33]

III.3.1 Diagramme de contexte du système :

Le diagramme de contexte représente les messages et les informations échangés entre le système d'information et les acteurs de ce dernier.

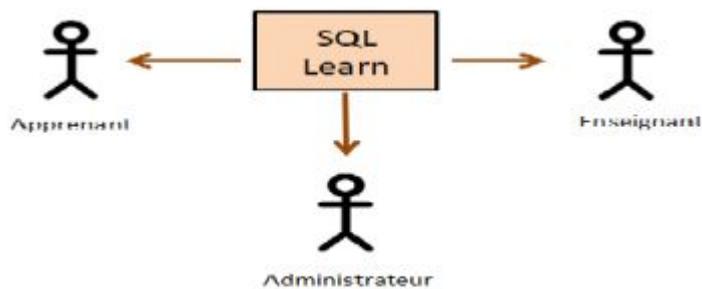


Figure III.8 : Diagramme de contexte.

III.3 .2 Diagramme de cas d'utilisation :

Les diagrammes de cas d'utilisation sont des diagrammes UML utilisés pour donner une vision globale du comportement fonctionnel d'un système logiciel.

❖ **Enseignant :**

C'est l'acteur responsable d'une très grande partie de suivi et d'évaluation,

Acteur	Cas d'utilisation
<i>Enseignant</i>	Gestion des TP
	Gestion des exercices
	Gestion des réponses
	Gestion d'évaluation
	Traçabilité

Tableau III.8 : Cas d'utilisation Enseignant

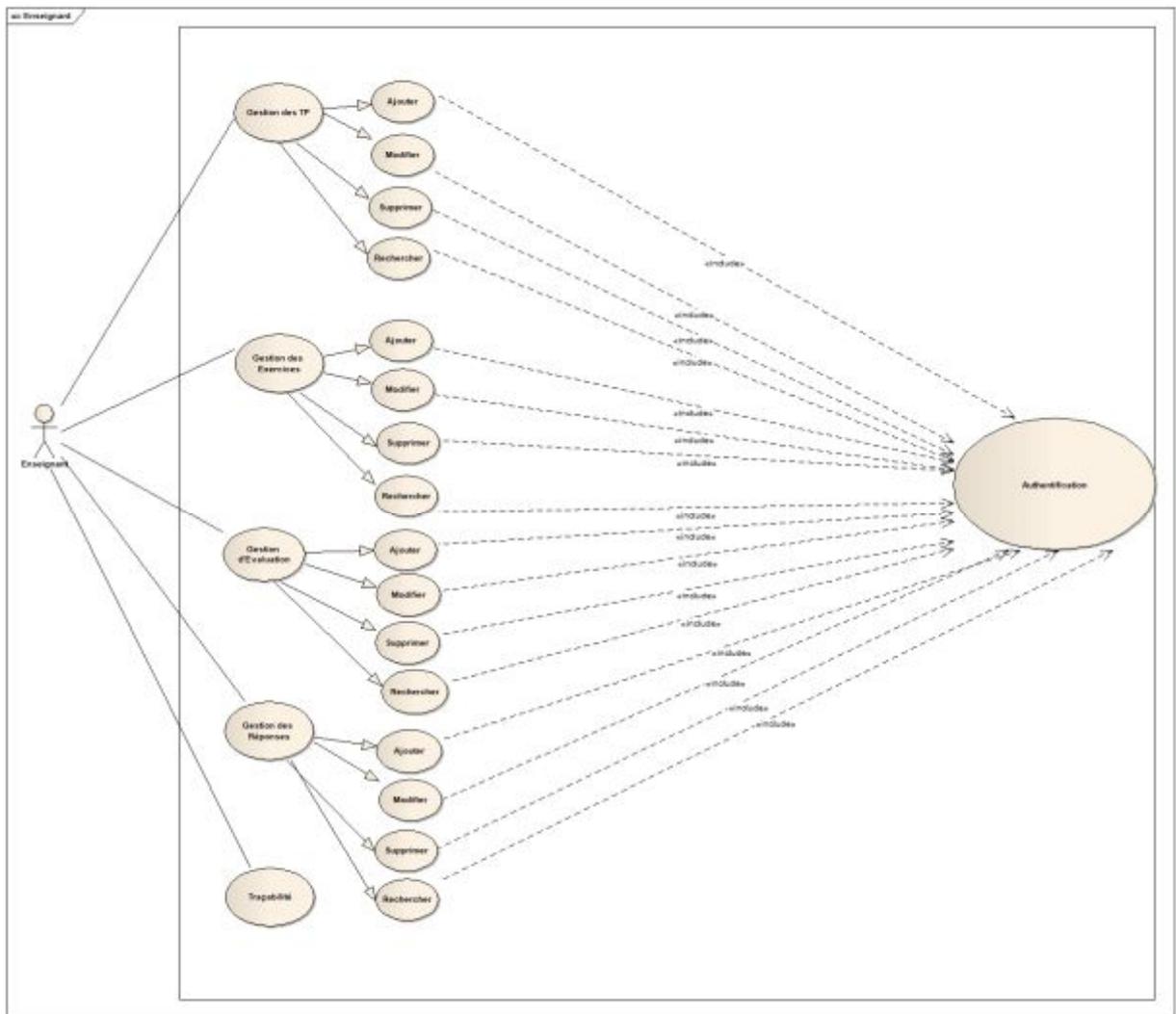


Figure III.9: Diagramme des cas d'utilisation de l'enseignant.

❖ **Etudiant :**

Acteur	Cas d'utilisation
<i>Etudiant</i>	Créer compte
	Choisir BD
	Choisir l'exercice
	Formuler la requête
	Tester Syntaxe
	Exécuter

Tableau III.9: Cas d'utilisation de l'apprenant

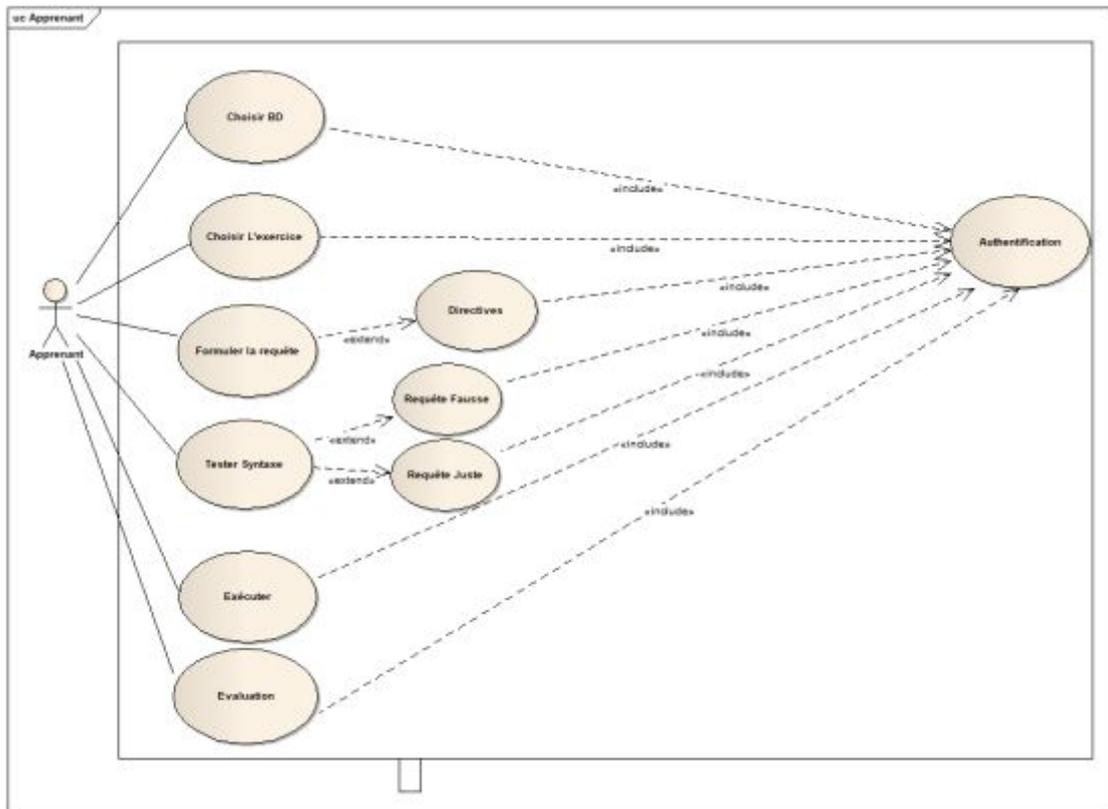


Figure III.10: Diagramme des cas d'utilisation de l'apprenant.

❖ **Administrateur :**

La plateforme doit lui fournir un accès à la base de données pour effectuer son rôle qui est :

- ❖ Consultation des liste des nouveaux inscrits, valider et les ajouter.
- ❖ Consultation des profils des membres, les modifier ou les supprimer.
- ❖ Consulter les événements de l'agenda, les modifier ou les supprimer.
- ❖ Consulter les discussions créés par les membres de la plateforme, modifier et le supprimer.

Acteur	Cas d'utilisation
<i>Administrateur</i>	Gestion des comptes
	Configuration serveur

Tableau III.10: Cas d'utilisation d'administrateur

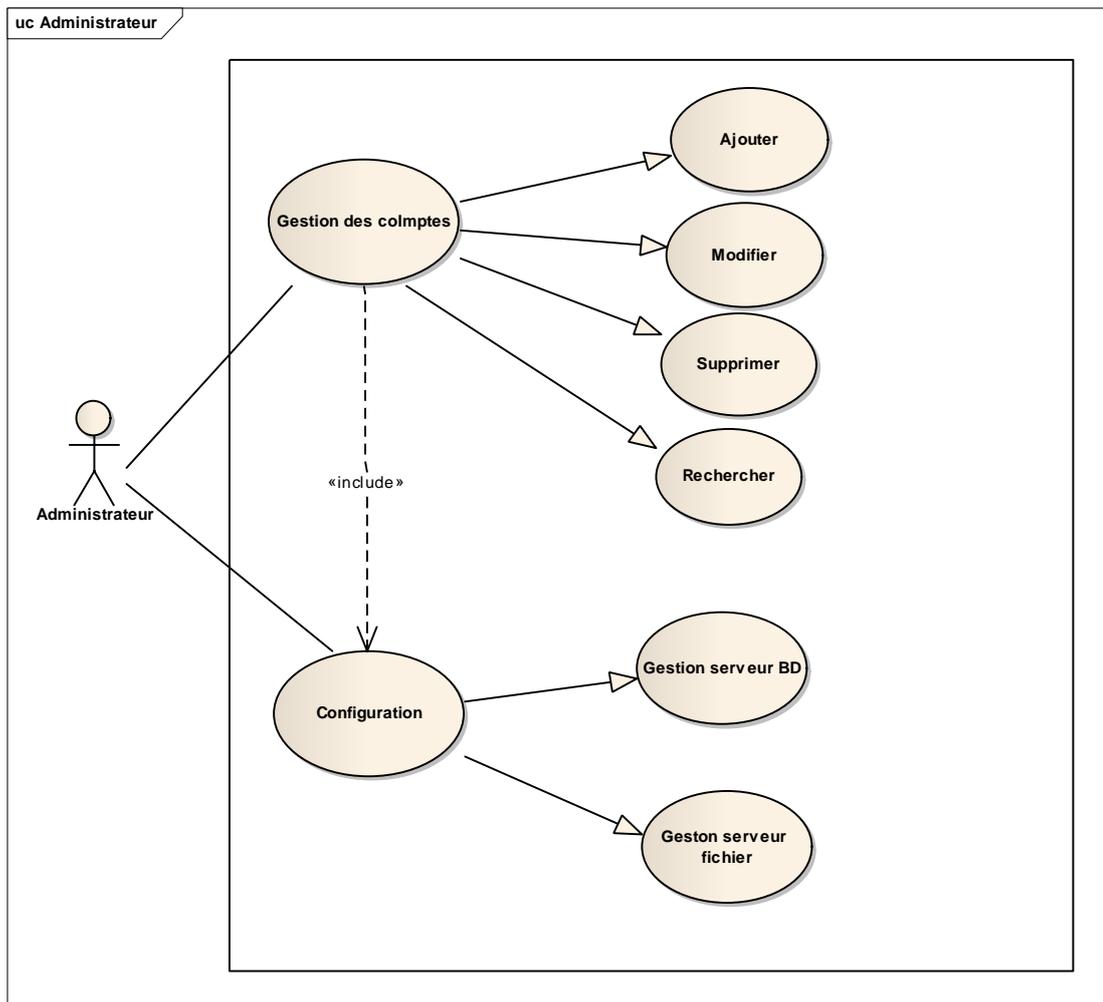


Figure III.11: Diagramme des cas d'utilisation de l'administrateur.

III.3 .2 Diagramme de classe :

Le diagramme des classes représente le côté statique de l'analyse du système, c'est le diagramme le plus largement répandu dans les spécifications d'UML, Le diagramme des classes identifie la structure de la classe d'un système, y compris les propriétés et les méthodes de chaque classe. Les diverses relations qui peuvent exister entre les classes y sont également représentées. Pour des raisons de lisibilité, nous présentons le diagramme de classe selon les modules.

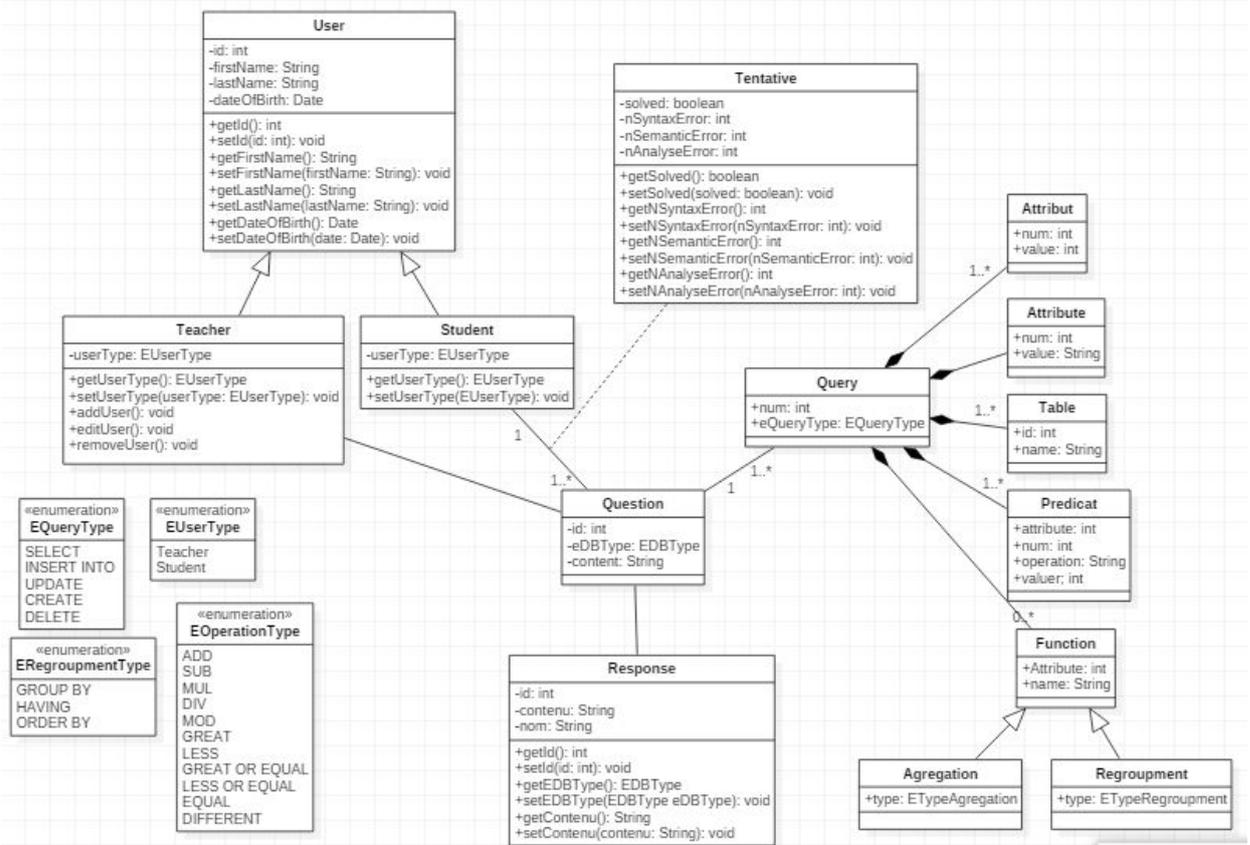


Figure III.12: Diagramme de classe.

III. 3 .3Diagramme de séquence :

Le diagramme de séquence permet de représenter les messages échangés entre les différents objets du système, avec un ordre chronologique représenté de haut en bas .Le diagramme de séquence n’est pas la seule modélisation possible des échanges de messages entre objets ; le diagramme de communication le permet aussi. La différence entre les deux diagrammes réside dont le fait que le diagramme de séquence « met l’accent sur l’aspect chronologique des communications »

Cas N°01 : Faire un TP

La Figure III.13 présente le diagramme de séquence pour le scénario « Faire un TP ».

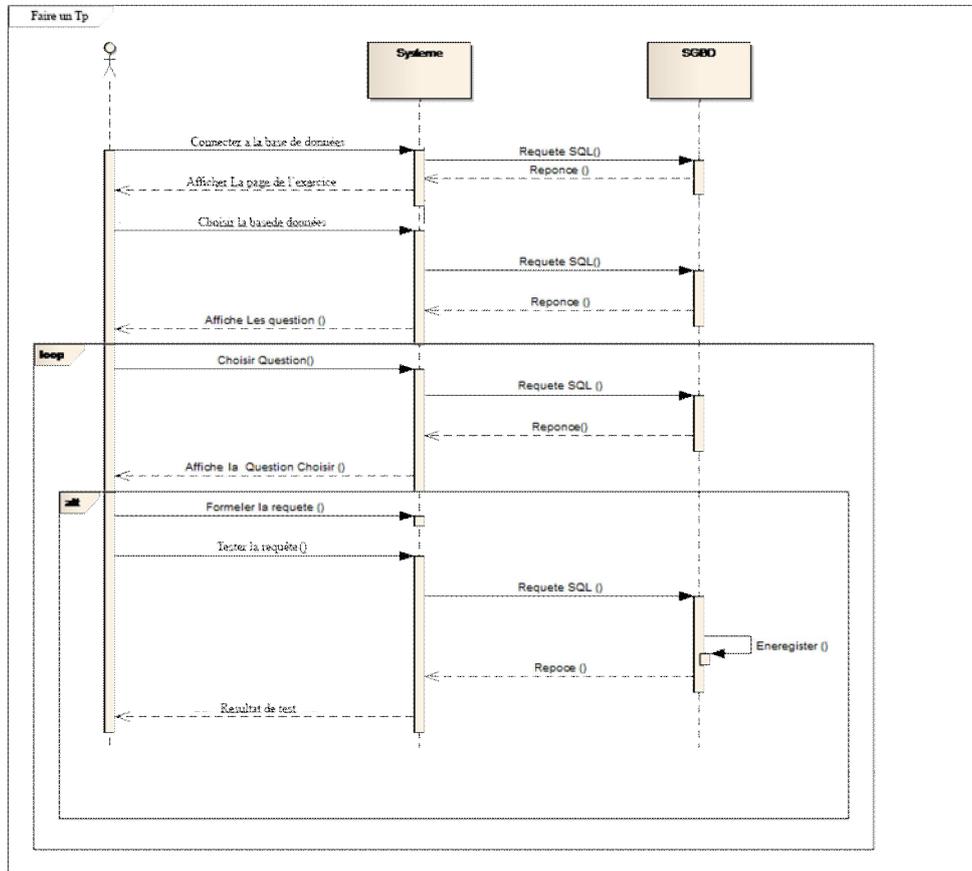


Figure III.13: Diagramme de séquence "Faire un TP".

Cas N°02 : *Voir la traçabilité*

La figure III.14 présente le diagramme de séquence pour le scénario « Voir traçabilité ».

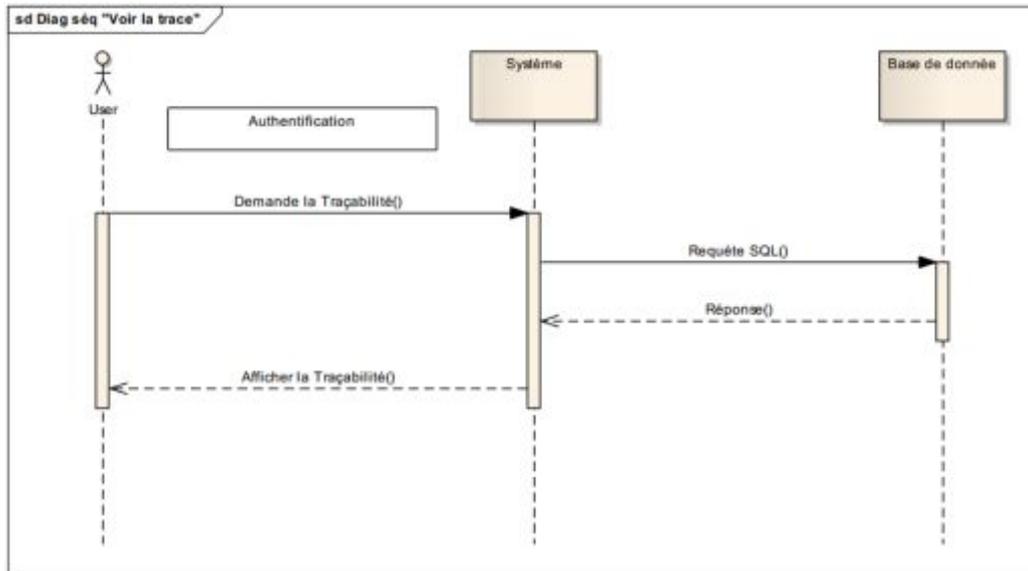


Figure III.14: Diagramme de séquence "Voir la traçabilité"

Cas N°03 : *Ajouter compte*

La figure III.15 présente le diagramme de séquence pour le scénario « Ajouter compte ».

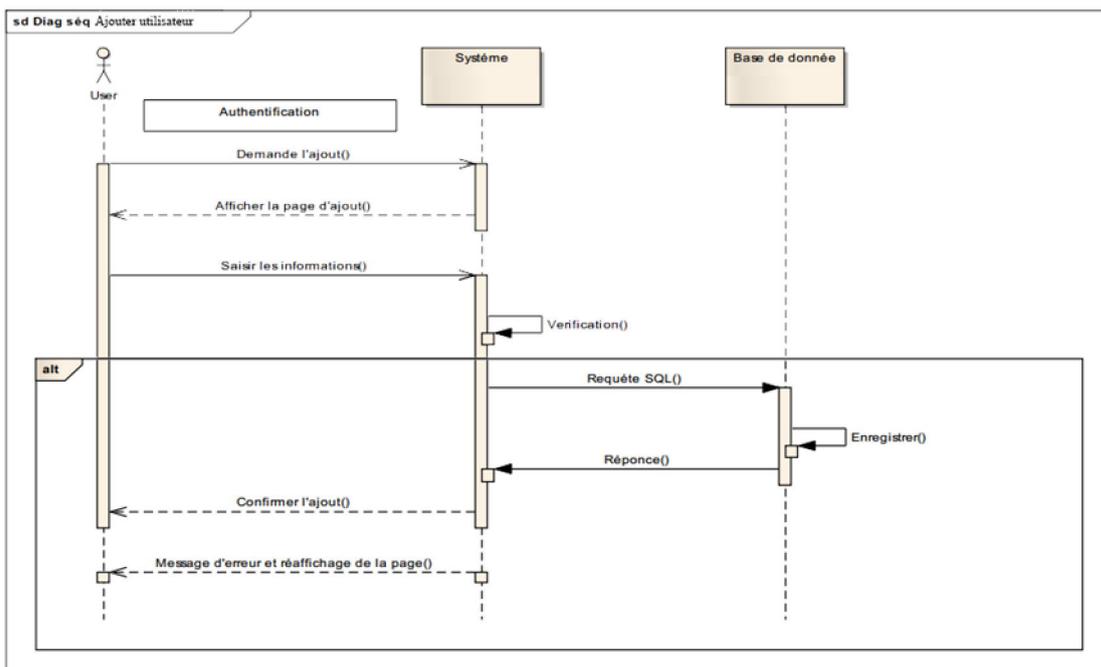


Figure III.15: Diagramme de séquence "Ajouter compte".

Cas N°04 : Modifier compte

La figure III.16 présente le diagramme de séquence pour le scénario « Modifier compte ».

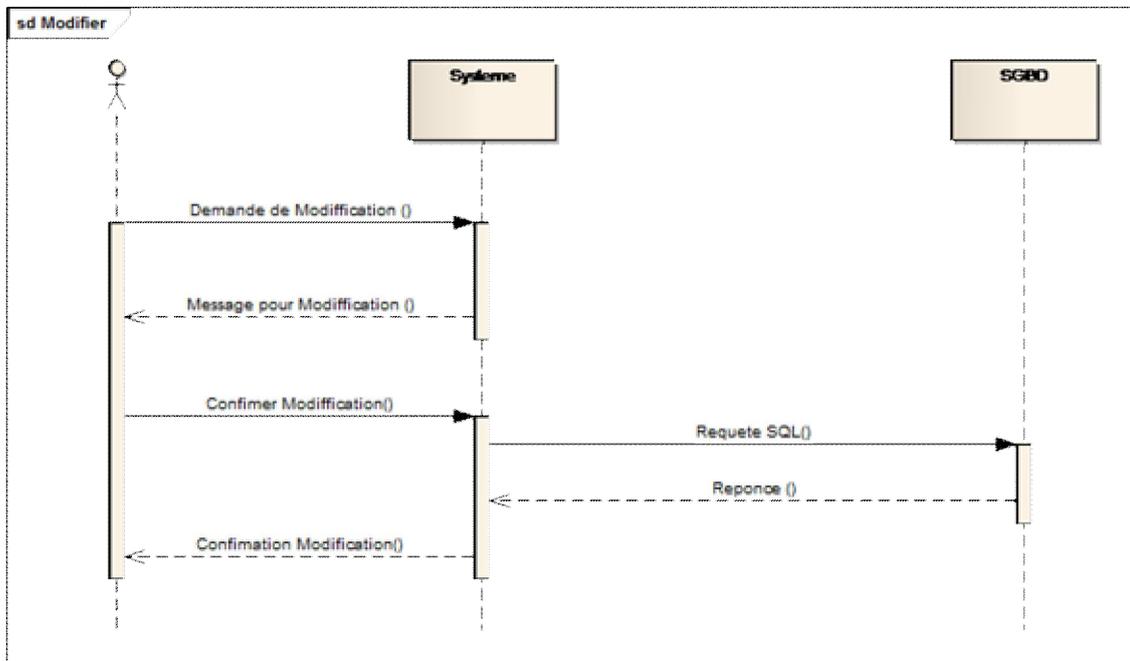


Figure III.16: Diagramme de séquence "Modifier compte".

Cas N°5 : *Ajouter question*

La figure III.17 présente le diagramme de séquence pour le scénario « Ajouter question ».

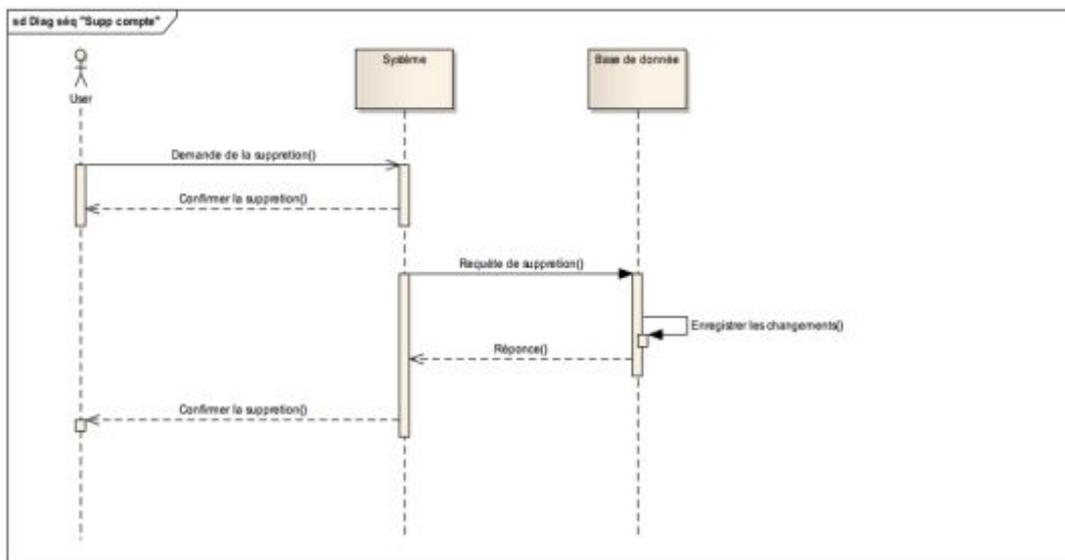


Figure III.17: Diagramme de séquence "Supprimer compte".

Cas N°6 : *Ajouter question*

La figure III.18 présente le diagramme de séquence pour le scénario « Ajouter question ».

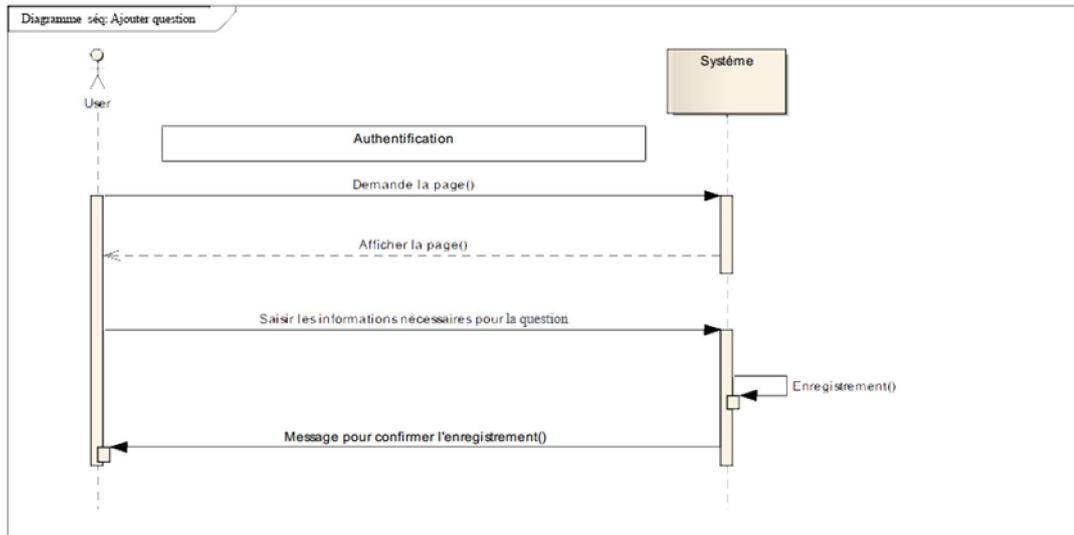


Figure III.18: Diagramme de séquence "Ajouter question".

Conclusion :

Tout au long de ce chapitre, nous avons présenté la description du domaine avec les méthodes de collecte de données, puis nous avons cité les anomalies rencontrées avec des suggestions d'amélioration proposées, ensuite nous avons entamé la partie de conception dont on a fait la modélisation en utilisant UML.

Dans le chapitre qui suit, nous présenterons les outils utilisés, l'environnement de développement, et le langage de programmation pour réaliser notre outil prototype nommée « Learn SQL ».

Chapitre IV

Implémentation de notre outil
prototype de faisabilité

Chapitre IV : Implémentation de notre outil prototype de faisabilité

IV .1 Introduction :

Après une étude large et approfondie du projet, nous arrivons dans ce chapitre à la mise en place de notre application. Ce chapitre comporte entre autre, la présentation des outils de développement utilisés, l'implémentation du système ainsi que des scenarios de déroulement de certaines fonctionnalités de l'outil.

IV.2 Technologies utilisées pour le développement :

Nous présentons dans cette section les outils utilisés pour le développement de notre Application (, *windev17*, *Enterprise Architect*, *HFSQL*. . etc.).



Figure IV.1: Les Technologies utilisées.

IV.2.1 HFSQL :

(Hyper File SQL) est un *moteur de base de données* lancé en 1988 (sous le nom de Hyper File) par la société française PC SOFT et incorporé en standard de l'*environnement de développement WinDev* du même auteur. Il est prévu pour être utilisé dans les logiciels créés avec **WinDev**, **WebDev** et **WinDev Mobile**, dans lesquels il peut être incorporé et ne nécessite aucune maintenance. Il existe en version Locale, Réseau, Cluster, Client/serveur (Windows et Linux), Cloud et Mobile, qui fonctionnent sur les *systèmes d'exploitation Windows, Linux, Mac OS, iOS et Android*.

IV.2.2 Entreprise Architect :

C'est un logiciel de modélisation et de conception UML, édité par la société Australienne **Sparx** Systèmes. Couvrant, par ses fonctionnalités, l'ensemble Des étapes du cycle de conception d'application, il est l'un des logiciels de Conception et de modélisation les plus reconnus.

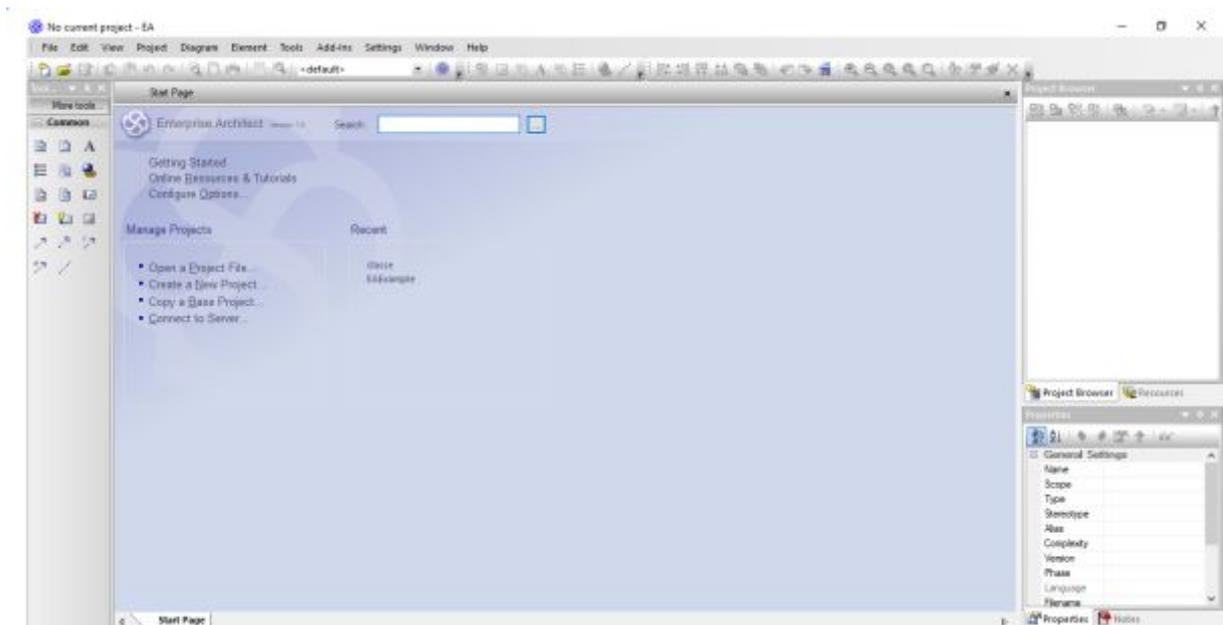


Figure IV.2 : Logiciel Enterprise Architect.

IV.2.3 Langage de Programmation WinDev :

WinDev est un AGL (Atelier de génie logiciel) complet permettant de développer des applications Windows dans de nombreux domaines : *gestion, industrie, médical,....*

Les applications développées peuvent inclure l'accès à des informations stockées dans des bases de données.

En standard, WinDev inclut un puissant moteur de base de données Hyper File SQL. Le moteur Hyper File SQL est disponible en version réseau, et en version Client/serveur.



Figure IV.3: logiciel WinDev.

IV.3 WinDev et le Client/ Serveur :

WinDev permet de créer des applications accédant à des bases de données Hyper File Client / Serveur.

Une application Hyper File Client / serveur consiste à exécuter l'application sur différents postes utilisateur (appelés machines clientes) et déporter la ou les bases de données sur un poste serveur.

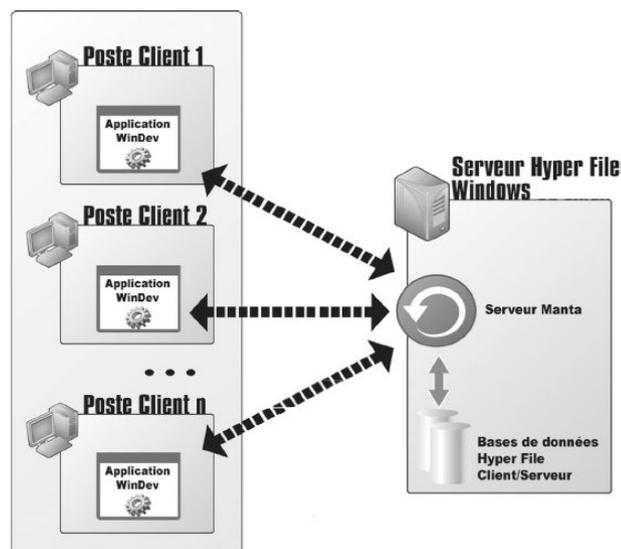


Figure IV.4 : Client serveur.

Grâce à l'application X-Manta on peut :

- Administrer entièrement les différents serveurs utilisés.
- Gérer les utilisateurs de base de données.
- Visualiser et/ ou modifier le contenu d'une base de données.
- Effectuer des recherches dans une base de données.
- Visualiser les connexions en cours sur une base de données ou sur un fichier de données.
- Arrêter, démarrer, installer et/ ou désinstaller un serveur.

Win Dev permet de :

- ❖ Créer entièrement une application Hyper File Client / serveur.
- ❖ Modifier une application WinDev existante en une application Hyper File Client / serveur.

IV.3.1 Mettre en place une application Hyper File Client / serveur :

Pour mettre en place une application Hyper File Client / serveur, les étapes sont les suivantes :

- 1) Installation du Hyper File Client / serveur
- 2) Connexion des postes client au serveur.
- 3) Utilisation de l'application.

a. Installation du Hyper File Client / serveur :

WinDev 17 détecté (installer un serveur Hyper File Client / serveur)

Choix de la plateforme (pour Windows sur cette machine)

Choix du serveur hyper file SQL (cocher sur installer un nouveau serveur)



Figure IV.5 : Installation du Hyper File Client / serveur.

b. Connexion des postes client au serveur :

Pour que l'application WinDev puisse manipuler les fichiers de données présents sur le serveur Hyper File, il est nécessaire de définir dans le projet la connexion qui sera utilisée par l'application.

Soit depuis l'éditeur d'analyses ou bien par programmation.

Nous avons choisi de définir la connexion par **l'éditeur d'analyses**.

1V.3.2 Définition de la connexion par l'éditeur d'analyses :

Voiser les étapes suivantes :

Configurer les paramètres de serveur de l'entrepôt de données (**Serveur, Port, Base de données, User, MPD**), puis importé la base de données et choisie les tables nécessaires après activer le serveur

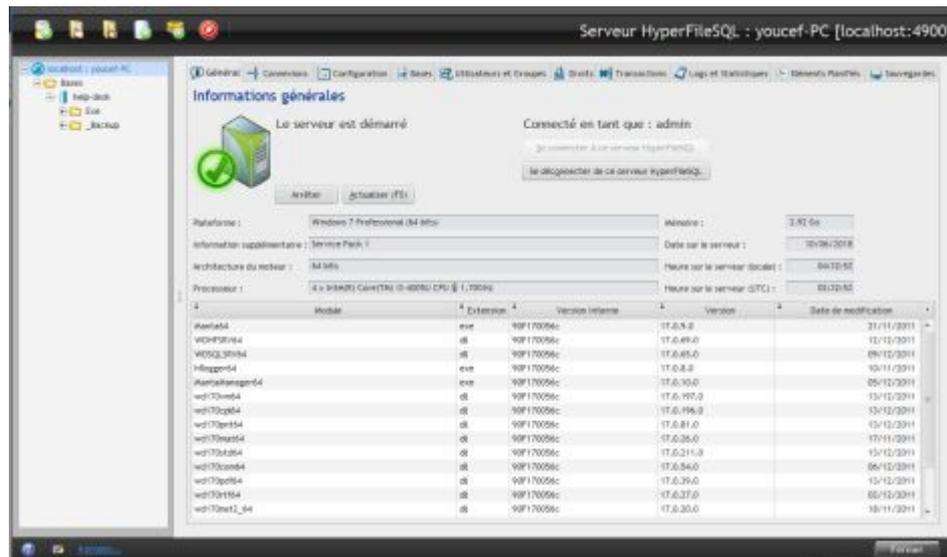


Figure IV.6: server Hyper File Client / serveur.

IV.4 Démonstrations de notre application ;

Nous présentons dans cette partie un ensemble de captures d'écran qui illustrent quelques fonctionnalités principales du système développé.

IV.4.1 Authentification :

Cette interface constitue la fenêtre d'accueil de notre application. À travers cette fenêtre l'utilisateur s'authentifie impérativement pour utiliser l'application. Cette étape met en valeur l'aspect sécurité, nous verrons la disponibilité du compte utilisateur et nous lui attribuons les droits et les privilèges nécessaires, pour qu'à la suite on lui affiche les fonctionnalités sur lesquelles il a les droits d'accès.

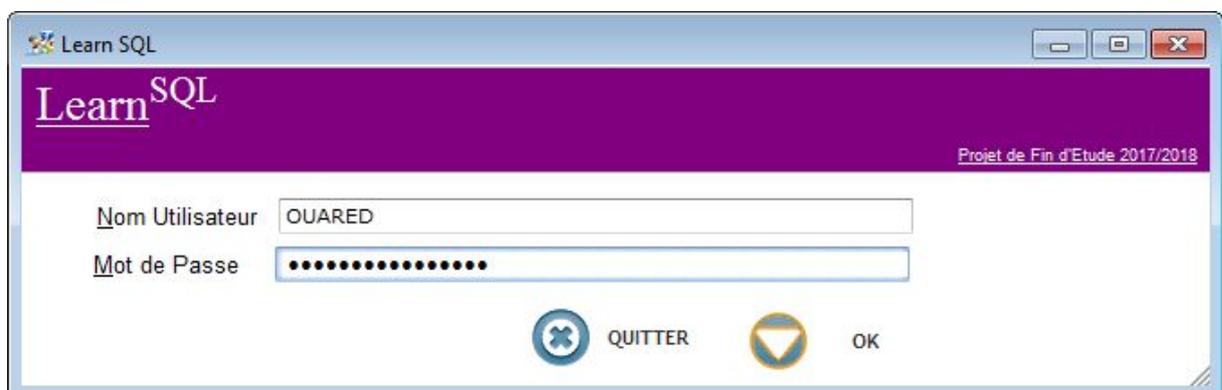


Figure IV.7 : Fenêtre d'authentification.

IV.4.2 L'interface principale :

Une fois l'utilisateur authentifié, il sera redirigé vers la page principale. L'interface de l'application est composée de plusieurs éléments : une page accueil qui affiche la définition d'une application et la fonctionnalité avec les conditions d'utilisation, **Catalogue SQL**, **Gestion et le Scenario TP**, **Traçabilité** et dernièrement **la gestion des utilisateurs avec la configuration de server SGBD**

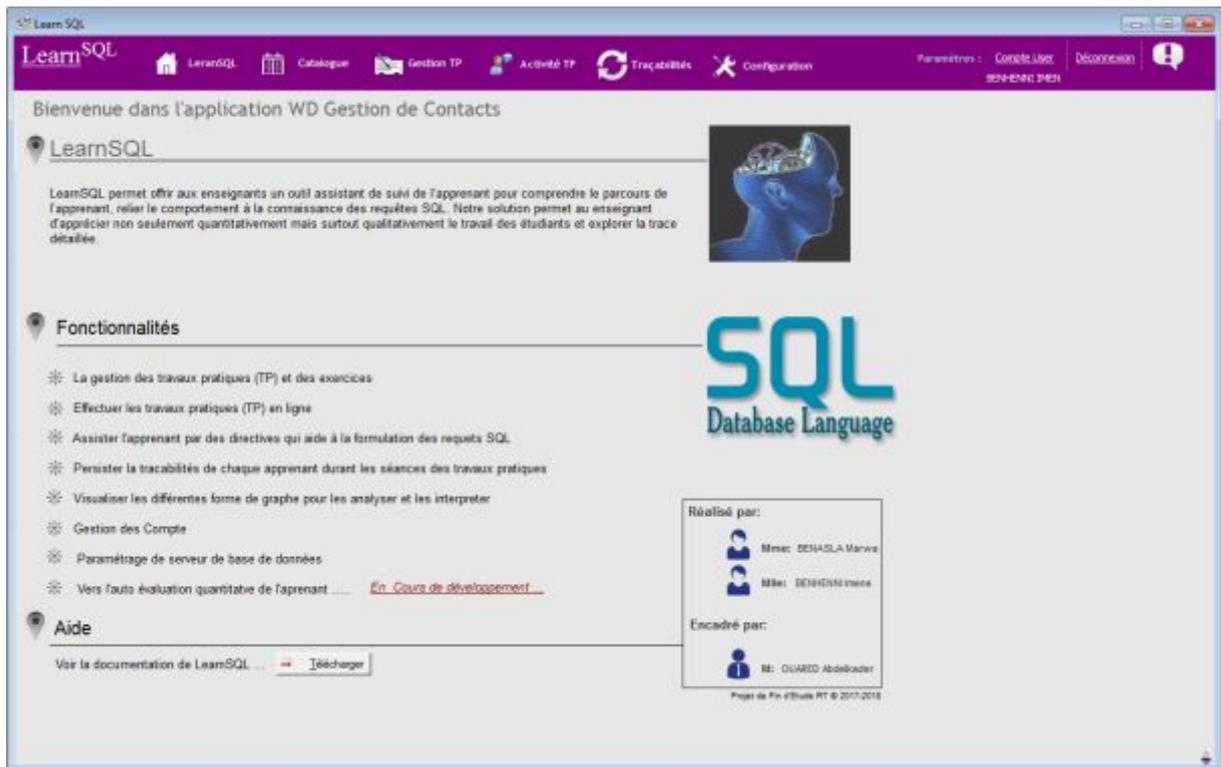


Figure IV.8 : page accueil.

IV.4.3 Catalogue SQL :

L'étudiant peut accéder à tous les ressources du langage SQL, en allant dans la rubrique «Catalogue SQL». Il sélectionne ensuite, celui qui l'intéresse, et accède à son contenu (ici, le chapitre n°1 a été sélectionné).

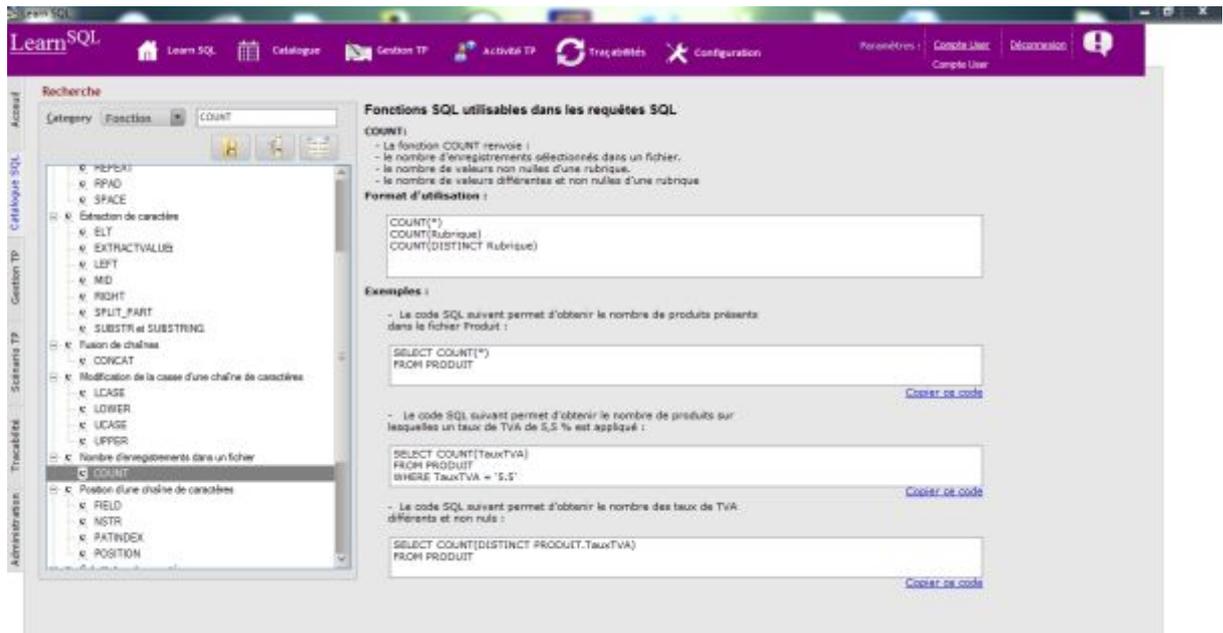


Figure IV.9: Catalogue SQL.

IV.4.4 Gestion TP :

L'enseignant peut faire la gestion des TPs et les exercices, il doit ajouter ces informations concernant le TP (N°TP, nom, objectif, date, ressource,..... etc)

Chaque TP concerne un schéma de base de données qui ça peut être existant ou bien non existant.

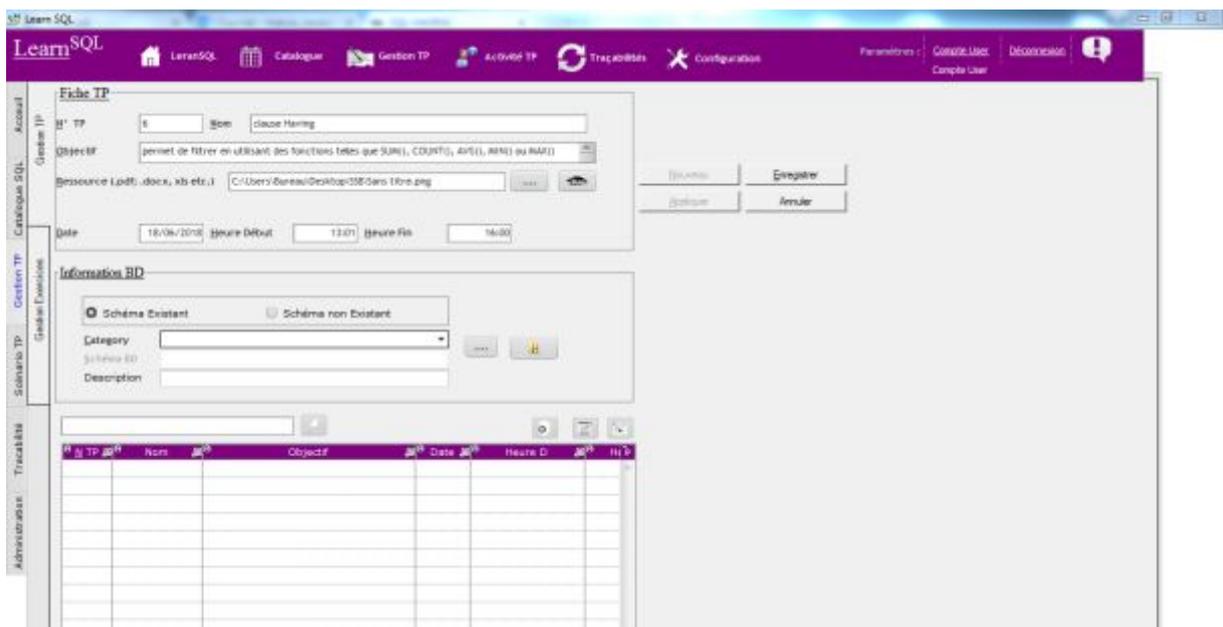


Figure IV/10 : Gestion TP

IV.4.5 Scénario TP :

Une fois familiarisé avec le catalogue, l'étudiant peut s'auto évaluer grâce aux questions, Le rôle de ce volet c'est de permettre au responsable du partage des Connaissances il contient des champs de saisie avec table de tentative et Description pour partager nouveau solution Ce qu'il à découvert.

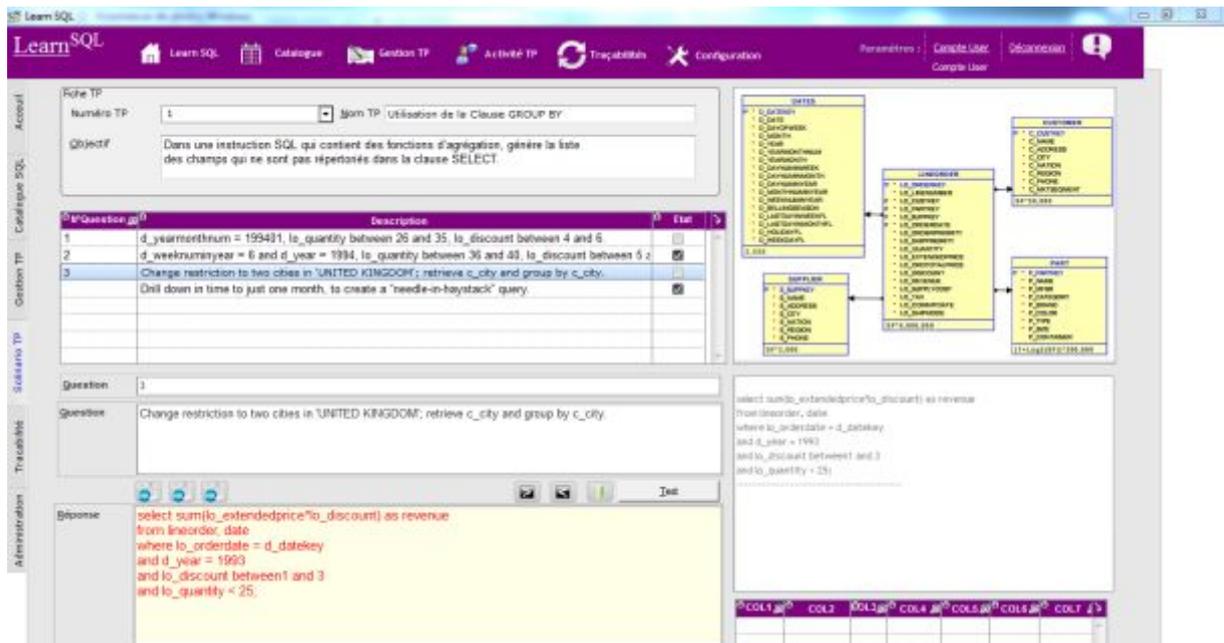


Figure IV.11: Scénario TP

IV.4.6 Traçabilité :

Le rôle de ce volet est de permet de donner un maximum d'information sur le comportement de l'apprenant durant les séances de travaux pratiques.

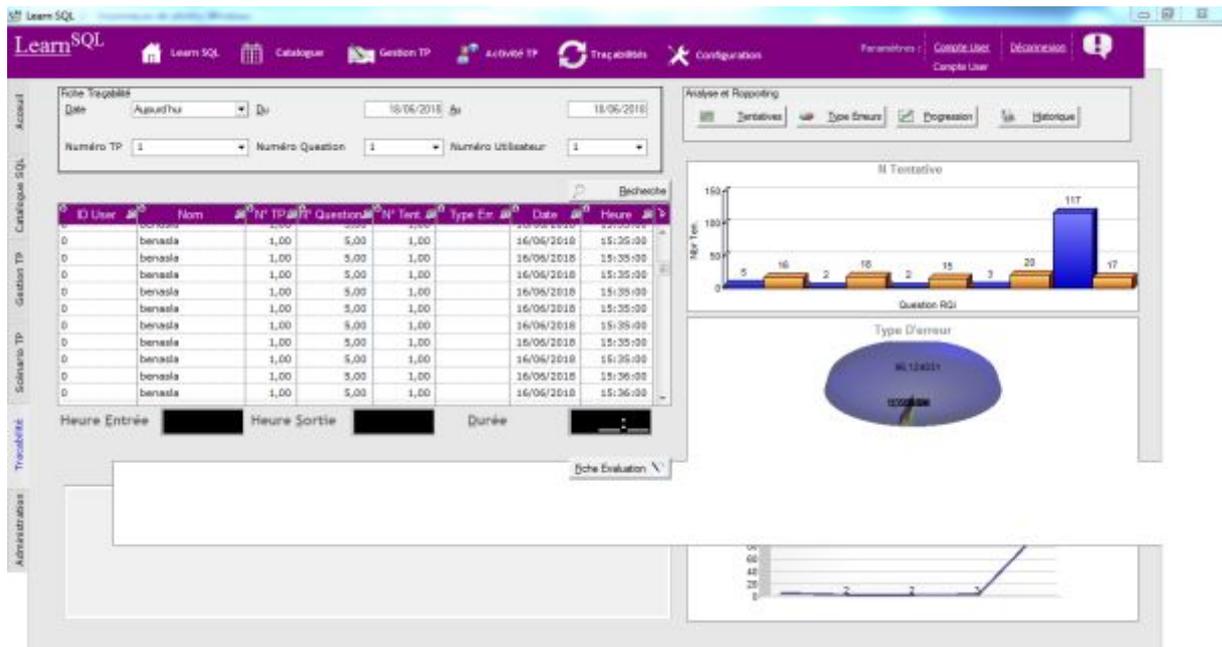


Figure IV.12 : Traçabilité.

IV.4.7 Configuration de serveur SGBD et des utilisateurs :

Cette fenêtre compose par deux volets le premier **pour la gestion des Utilisateurs** et le deuxième **pour la configuration de serveur SGBD**.

IV.4.7.1 gestion utilisateur :

Le rôle de ce volet est de créer des utilisateurs avec les opérations insincère comme (modifier et supprimer..etc.).

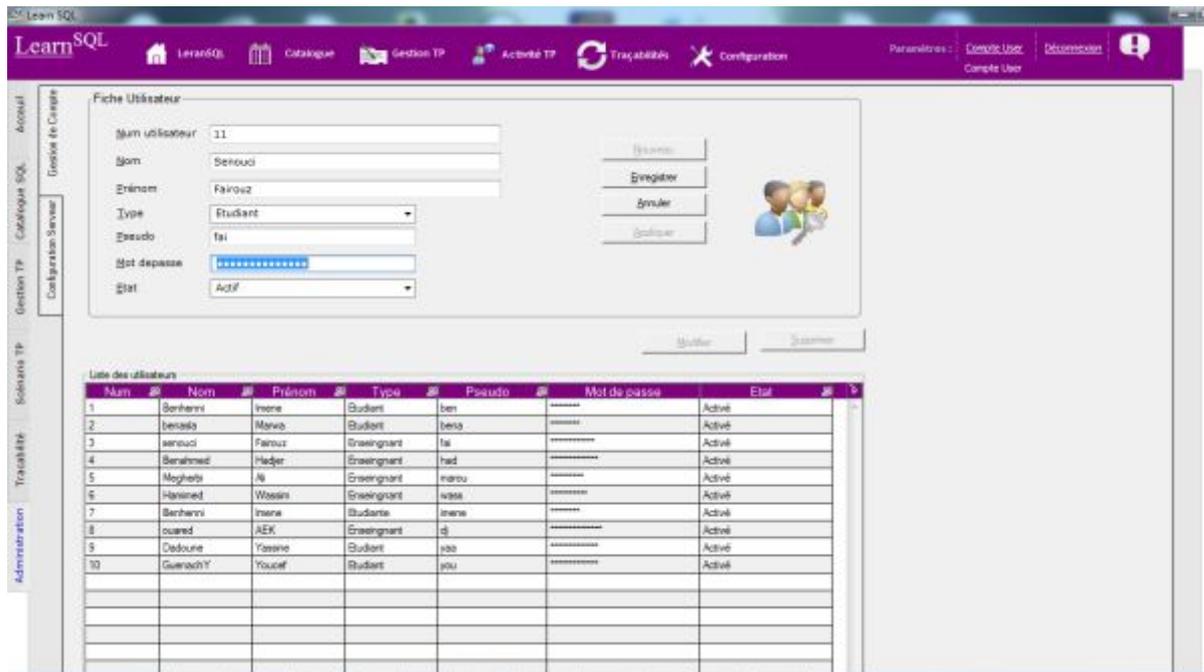


Figure IV.13 : Gestion des utilisateurs.

IV.4.7.2 Configuration le SGBD :

L'administrateur du système commence par configurer les paramètres de serveur de l'entrepôt de données (*Serveur, Port, Base de données, User, MPD*), et avant de valider il doit tester la connectivité.

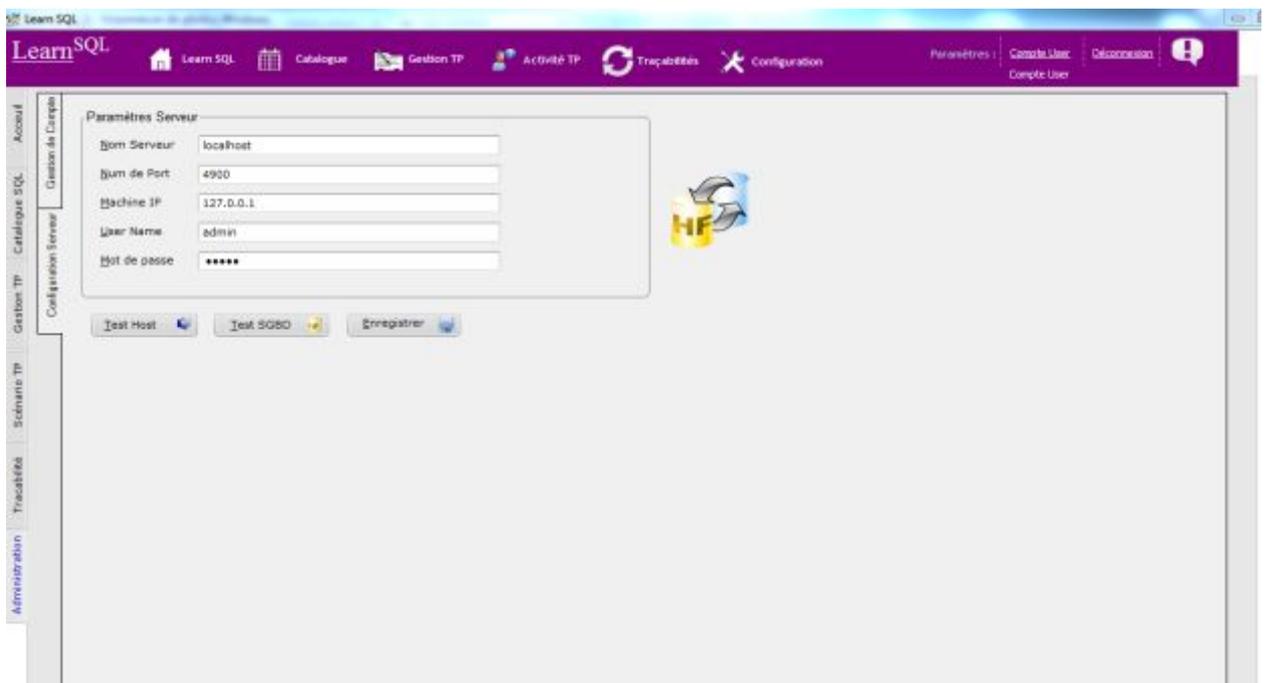


Figure IV.14: Configuration Serveur.

IV.4.8 Les tables de base de données :

Cette capture pour pressante les tables de base de données qu'est utilisé dans Notre application chaque table contient des attribue et des clés primaires et Secondaires avec liaison entre les tables.

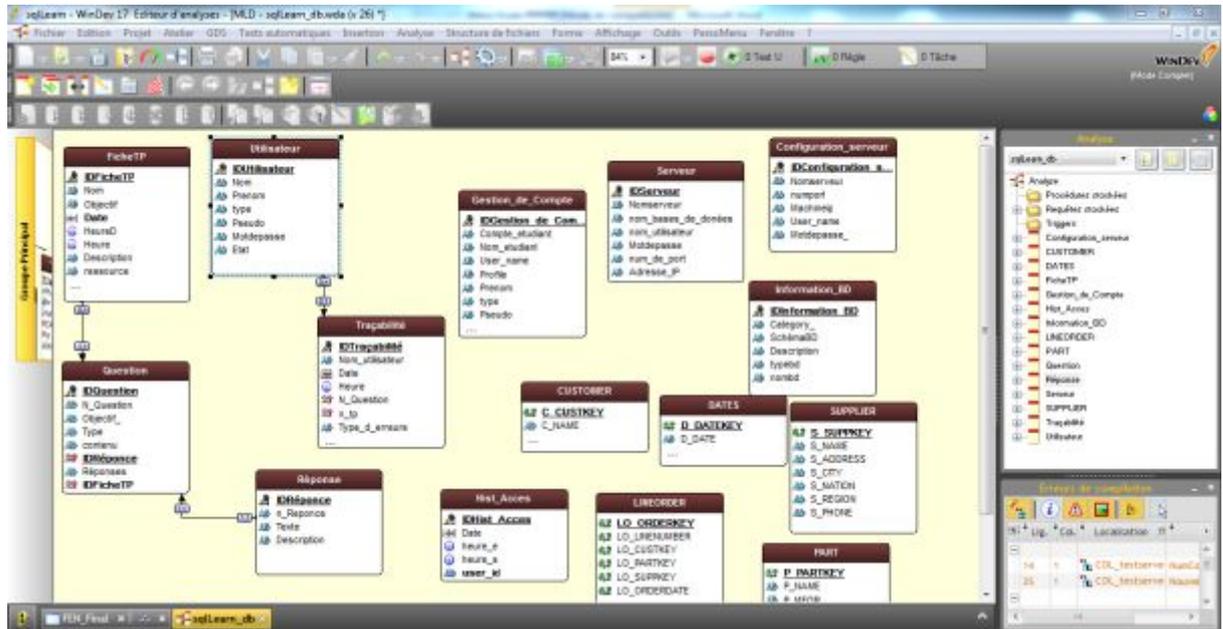


Figure IV.15 : Les tables de base de données.

IV.4 Conclusion :

Dans ce chapitre, dans un premier lieu nous avons présenté l'environnement et les technologies utilisées dans l'implémentation. En second lieu nous avons montré notre outil prototype « learn SQL » sous forme des captures d'écran.

Conclusion Générale

Conclusion générale

Conclusion et perspectives :

Dans ce projet de fin d'étude, nous avons abordé le problème d'observation de comportement de l'apprenant durant les séances de travaux pratiques. Ce travail est motivé par la difficulté de l'évaluation de l'apprenant et l'interprétation de comportement de l'apprenant vis-à-vis la connaissance pour résoudre ce problème, nous avons analysé le système existant et recenser les anomalies avec leur causes et leur conséquences. Cette analyse ça nous permis de suggérer des améliorations organisationnels, informationnels et techniques. Nous avons modélisé notre système par le langage UML. La solution proposée permet d'acquérir toutes les actions, les entrés des utilisateurs ainsi une classification des erreurs dans la formulation des requêtes SQL aide l'enseignant à catégoriser les apprenants. Afin de montrer la faisabilité, nous avons proposé un outil prototype qui assiste les enseignants et les étudiants pendant les séances de TP.

Ce travail ouvre plusieurs perspectives nous pouvons citer :

- Enrichir le module de l'observation et l'interprétation de comportement de l'apprenant.
- Considérer l'aspect de collaboration entre les différents étudiants.
- Evaluer la facilité d'utilisation (la mémorisation, apprentissage, nombre d'erreur, la satisfaction, composant jugé utilisable notre outil prototype par les enseignants de spécialités de base de données.
- Cons.
- idérer un autre TP comme les TP algorithmiques

Références Bibliographiques

Référence bibliographique

- [1] :H.GarciaMolina,J.D.UllmanetJ.Widom.DatabaseSystems:TheCompleteBook.2eéd.Up perSaddleRiver,NJ,USA:PrenticeHallPress,2008.isbn:978013187324(cf.p.33,36,37,39,40,43).
- [2] :R.Elmasri.Fundamentals.of.database.systems.PearsonEducationIndia,2008(cf.p.25,26, 33,36,37,39, 40,43–45).
- [3] :M.JarkeetJ.Koch.«Queryoptimizationindatabasesystems».In:ACM.Computing.surveys (CsUR).16.2(1984),p.111–152(cf.p.36).
- [4] :J.C.Freytag.Arule,based.view.oF.query.optimization.T.16.3.ACM.1987(cf.p.37).
- [5] :P.G.Selingeretal.«Accesspathselectioninarelationaldatabasemanagementsystem».In:Proce edings.of the 1979 ACM SIGMOD international conference on Management.of.data.ACM.1979,p.23–34(cf.p.40, 42,44,45,193).
- [6] :S.Chaudhuri.«Anoverviewofqueryoptimizationinrelationalsystems».In:ACM.SIGACT .SIGMODSIGART.ACM.1998,p.34–43(cf.p.40,104).
- [7] :R.RamakrishnanetJ.Gehrke.DatabaseManagementSystems.3eéd.NewYork,NY,USA: McGraw-Hill, Inc.,2003.isbn :0072465638,9780072465631(cf.p.41).
- [8] :<http://www.educnet.education.fr/dossier/eformation/eformationelearning/definitions?s earchterm=elearn>, 18/11/2008.
- [9] : <http://www.definition.be/defipoints/point2.asp>,18/11/2008.
- [10] : net nversite, une plateforme sur IMS LD pour la conception de cours en ligne dans le cadre du projet CEPIAH (conception et évaluation de la polycopie interactive pour apprentissage humaine), thèse de doctorat 2005, Ecaterina Pacurar Giacomini.
- [11] : MOOSER François, « évaluation des apprentissage », Mémoire de fin d'études pour l'obtention du diplôme de formation continue en enseignement supérieur et technologie de l'éducation, 2004
- [12] : Emna Ben Romdhane Hatem Skik « E-learning : élément de réflexion autour d'une expérience en "blended learning" développée dans le milieu universitaire », ESC Tunis Assistant – ESCE Tunis.
- [13] :Lahcen OUBAHSSI «Conception de plates-formes logicielles pour la formation à distance,présentant.des.propriétés.d'adaptabilité.à.différentes.catégories.d'usagers.et.d'interopé rabilité.avec.d'autres environnements logiciels », Thèse de doctorat, 2005, Université René Descartes – Paris V.
- [14] : évaluation formative : Réflexion sur une approche différente de la relation maître-élève, Jacques Belleau, juin 2000.
- [15] : Berthiaume, D. & Daele, A. (2013). Comment clarifier les apprentissages visés par un enseignement ? In D. Berthiaume & N. Rege Colet (Eds.), La pédagogie de l'enseignement supérieur : repères théoriques et applications pratiques, Vol. 1, 55-71. Berne : Peter Lang.

Référence bibliographique

- [16] : CEFES (Centre d'études et de formation en enseignement supérieur) de l'Université de Montréal. Guide d'élaboration d'un plan de cours (section « Cibles d'apprentissage – Formation.par.objectifs»). Consulté à l'adresse http://www.cefes.umontreal.ca/ressources/guides/Plan_cours/FPO.htm
- [17] : Krathwohl, D.R. (2002). A Revision of Bloom's Taxonomy, An Overview. *Theory into Practice*, 41(4), pp. 212-218.
- [18] : Bloom, B.S., et al., *Taxonomy of educational objectives: The classification of educational goals. Handbook 1: Cognitive domain* 1965, New York.: David McKay.
- [19] : Proceedings of the ninth annual international conference on International computing education research 2012, ACM: Auckland, New Zealand.
- [20] : Anderson, J., *The architecture of cognition*. Cambridge, Mass.: Harvard University Press, 1983.
- [21] : Gorman, M.E., Types of Knowledge and Their Roles in Technology Transfer, in *The Journal of Technology Transfer* 2002, Springer Netherlands. p. 219-231.
- [22] : Cutts, Q., et al., The abstraction transition taxonomy: developing desired learning outcomes through the lens of situated cognition
- [23] : Fuller, U., et al., Developing a computer science-specific learning taxonomy. *SIGCSE Bull.*, 2007. 39(4): p. 152-170.
- [24] : Lahtinen, E. A categorization of novice programmers: A cluster analysis study. in *Proceedings of the 19th annual Workshop of the Psychology of Programming Interest Group*. 2007.
- [25] : Bower, M., A taxonomy of task types in computing. *SIGCSE Bull.*, 2008. 40(3): p. 281-285.
- [26] : Amir Benmimoun, «Partie II: NetUniversité version 3 –conception et réalisation » Rapport de stage Master, 2006. www.inrp.fr/atief/theses.htm.
- [27] : Michel Desbiens , Enseignant en mathématiques et sciences ,Programme PROTIC , École des Compagnons-de-Cartier <http://www.protic.net/profs/micheldesbiens>
- [28] : Christian Martel, Christine Ferraris, Laurence Vignollet 2007 : Equipe Scénario, Université de Savoie Campus Scientifique 73376 Le Bourget-du-Lac cedex :{Christian.Martel, Christine.Ferraris, Laurence.Vignollet}@univ-savoie.fr : Scénariser les 4 piliers de la pédagogie : Environnements Informatiques pour l'Apprentissage Humain, Lausanne 2007 .
- [29] : FAERBER Richard « Indexer des situations d'apprentissage coopératif », 2004.

Référence bibliographique

- **[30]** : Everardo REYES GARCIA, « L'objet technique hypermédia : repenser la création de contenu éducatif sur le Web », Thèse de Doctorat de l'Université Paris VIII, Soutenue en Janvier 2007.
- **[31]** : Advanced Distributed Learning (ADL), « Sharable Content Object Reference Model (SCORM) - Content Aggregation Model (CAM) Version 1.3».
- **[32]** : « Communautés de Pratique du E-Learning - CoPE : définition des concepts et proposition d'un langage de spécification des scénarios d'apprentissage » rapport de mémoire de magister, l'Ecole Nationale Supérieure d'Informatique (ESI ex : INI) 2008.

Webographie

Webographie

[FRM, 09] : <http://www-fr.mysql.com/> , 06/06/2009.

[EDU,08] : <http://www.educnet.education.fr/dossier/eformation/eformationelearning/definitions?searchterm=elearn>, 18/11/2008.

[ELE 08] : les avantages d'e-Learning

<http://www.elearning-entreprise.com/se-former-a.../avantages-du-e-learning>

[CUR 08] : Pour bâtir des scénarios pédagogiques intégrant les TIC <http://cursus.edu/dossiers-articles/dossiers/18/art-recit-scenarisation-pedagogiquetelling/articles/17236/pour-batir-des-scenarios-pedagogiques-integrant/>.

[XPER, 08] : <http://www.x-perteam.fr/>, 18/11/2008.

[WIK, 08] : http://fr.wikipedia.org/wiki/Plate-forme_d'apprentissage_en_ligne, 18/11/2008.

[PRE 01] : Site officiel pour les prestataires e-Learning

<http://e-learning.prestataires.com/conseils/les-standards-de-le-learning>

