

UNIVERSITE DE LUBUMBASHI

FACULTE DES SCIENCES

DEPARTEMENT DE MATHEMATIQUES ET INFORMATIQUE



**Implémentation d'un système
d'interrogation floue intégré à un
gestionnaire de base de données
relationnelle.**

Cas de « l'agence automobile RHM_Unit Business »

Dieumerci KUNZI MATONDO

*Travail présenté en vue de l'obtention
du titre de bachelier en Sciences
Informatiques.*

Octobre 2020

UNIVERSITE DE LUBUMBASHI

FACULTE DES SCIENCES

DEPARTEMENT DE MATHEMATIQUES ET INFORMATIQUE



**Implémentation d'un système
d'interrogation floue intégré à un
gestionnaire de base de données
relationnelle.**

Cas de « l'agence automobile RHM_Unit Business »

Dieumerci KUNZI MATONDO

*Travail présenté en vue de l'obtention
du titre de bachelier en Sciences
Informatiques.*

*Directeur : CT **Serge BUKASA***

Octobre 2020

Epigraphe

À l'heure actuelle, la commande floue est la partie médiatiquement émergée d'un iceberg scientifique. (D. Dubois)

Dédicaces

A ma mère,

A mon père (d'heureuse mémoire),

A mes sœurs,

A toute ma famille,

A tous ceux que j'aime,

A tous mes enseignants.

Table des matières

Table des matières	i
Liste des figures	ii
Remerciements	iii
Introduction générale	1
Chapitre I. DEVELOPPEMENT THEORIQUE	5
I.1. Problème	5
I.2. Solution	6
I.3. Les prérequis	6
<i>I.3.1. La logique floue</i>	6
<i>I.3.2. La théorie des ensembles flous</i>	9
<i>I.3.3. Types d'informations d'une base de données floue</i>	14
I.4. Les bases de données floues	18
I.5. SQLf et l'évaluation de requêtes floues	24
Chapitre II. CADRE D'EXPERIMENTATION	29
II.1. Cadre géographique	30
II.2. Organisation et structure	30
II.3. Présentation et discussion autour de la base de données de l'agence	34
Chapitre III. CONSTRUCTION DE LA SOLUTION	38
III.1. Configuration de la BDR	38
III.2. Préparation des requêtes flexibles	39
III.3. Interrogation flexible	40
III.4. Présentation et interprétation des résultats	43
Conclusion	46
Références	47

Liste des figures

Figure 1 : Schéma de la solution.....7

Figure 2 : Forme trapézoïdale de la fonction d'appartenance.....12

Figure 3 : Organigramme de l'agence RHM_Unit Business.....32

Figure 4 : Création de la base de données agenceautorhm36

Figure 5 : Création de l'extension sqlf.....37

Figure 6 : Création de la table auto.....37

Figure 7 : Copie des tuples dans la table.....38

Figure 8 : Définition des prédicats flous.....39

Figure 9 : 1^{ière} requête.....40

Figure 10 : 2^{ième} requête.....40

Figure 11 : Quantificateur OWA.....41

Figure 12 : 3^{ième} requête.....41

Figure 13 : 4^{ième} requête.....41

Remerciements

Je tiens à remercier Monsieur Serge BUKASA, Chef des travaux à l'UNILU, pour avoir bien voulu accepter de nous encadrer tout au long de cette démarche.

Je tiens à remercier également Monsieur Daniel BAVUEZA, Professeur à l'UNILU de qui j'ai reçu les prérequis sur les bases de données.

Je tiens à remercier aussi Monsieur Raymond MULUMBA, Chef des travaux à l'UNILU, pour les rudiments de la logique booléenne grâce à quoi j'ai pu aborder avec lucidité la logique floue.

Je tiens également à remercier Monsieur André MAKITU, de qui j'ai eu le goût très jeune d'apprendre l'outil informatique.

Mes remerciements s'adressent aussi à Madame Gloire KUNZI et Monsieur l'abbé Germain NZINGA par qui le bon Dieu est passé pour nous soutenir tout au long de mon cursus.

J'aimerais signifier, sincèrement, toute ma reconnaissance à Madame Tabitha MABUNGU, pour ses conseils précieux, sa présence et ses motivations ; Je ne pourrai oublier la famille MPHANZU et la famille BAFUNA.

Je ne devrais pas oublier de remercier mes collègues de promotion pour les encouragements et le soutien moral qu'ils m'ont apporté.

Introduction générale

Les Systèmes de Gestion de Bases de Données relationnelles (SGBDR) sont devenus, sans conteste, le noyau de tout système informatique. Cependant, la diversification des applications des bases de données a montré les limites des SGBDR notamment sur le plan de modélisation des données imprécises et de l'interrogation flexible (BEN HASSINE M. , 2005).

Les systèmes de gestion de base de données relationnelle (SGBDR) commerciaux reposent essentiellement sur un langage *booléen* d'interrogation, à savoir Structured Query Language (SQL). L'interprétation d'une requête floue sur un SGBDR nécessite donc soit une étape de dérivation afin d'être transformée en requête *booléenne* soit l'utilisation d'une extension dédiée aux requêtes floues (SMITS, PIVERT, & GIRAULT, 2006).

C'est dans cette perspective que nous menons notre étude autour du sujet intitulé : « **Implémentation d'un système d'interrogation floue intégré à un gestionnaire de base de données relationnelle (PostgreSql) cas de (l'agence automobile RHM_Unit Business) »** ».

Plusieurs raisons nous poussent à élaborer ce travail avec une thématique complexe et ce choix ne repose pas sur le hasard, il est plutôt soutenu par des raisons valables que nous classons à 3 niveaux :

- Notre choix se justifie par notre esprit de curiosité sur la logique floue et ses applications, défendre notre formation en tant que informaticien et aussi approfondir la notion sur la logique floue appliquée aux bases de données.

- Par ce travail, nous voulons appuyer, amplifier la position des scientifiques qui estiment les BDR présentent des limites par rapport aux demandes et requêtes des utilisateurs finaux qui n'obtiennent généralement pas satisfaction.
- Nous sommes par ailleurs convaincus que tous les domaines d'activité reposent sur des systèmes d'information, qui ne sont rien d'autres que des ensembles organisés qui permettent de collecter, stocker, traiter et diffuser l'information. Il s'agit d'une application des notions liées aux bases de données floues dans le domaine de la vente ou de la location des véhicules.

Plusieurs travaux ont été proposés dans la littérature pour introduire la flexibilité dans l'interrogation des bases de données. La majorité de ces travaux ont utilisé le formalisme des ensembles flous et de la logique floue pour modéliser les termes linguistiques tels que ("modéré", "moyen") et pour évaluer des prédicats comportant de tels termes. L'idée essentielle dans ces travaux consiste à étendre le langage SQL et à ajouter une couche supplémentaire d'un SGBD classique pour évaluer les prédicats flous. Parmi ces travaux, nous citons :

a. MOHAMED ALI BEN HASSINE: dans son ouvrage, il affirme que pour stocker de grandes quantités de données, les bases de données ont des mécanismes incorporés seulement pour des relations exactes. Comme la technologie est appliquée pour atteindre des domaines très variés, qui supportent des données vagues, imprécises et/ou des préférences dans les attributs, parfois les relations usuelles de SQL ne peuvent pas complètement satisfaire ou modéliser les relations et les attributs de ces domaines. L'interrogation flexible constitue alors une alternative à l'interrogation booléenne pour certains types d'applications (BEN HASSINE, Contribution à l'implémentation d'une

base de données floue sous un système de gestion de base de données relationnel, 2005).

b. GREGORY SMITS, OLIVIER PIVERT ET THOMAS GIRAULT: dans leur article, ils confirment que Les ensembles flous constituent un cadre théorique et méthodologique intéressant pour l'interrogation flexible de Base de Données Relationnelles (BDR).

En effet, associé à une fonction d'appartenance et une étiquette linguistique, un ensemble flou formalise une propriété graduelle pouvant être ensuite intégrée en tant que préférence dans une requête adressée à une BDR. En ce sens, les requêtes floues étendent les requêtes booléennes à travers la prise en compte de tolérance et de gradualité dans la définition des intentions des utilisateurs. La logique floue apporte également un riche ensemble de connecteurs permettant de combiner ces préférences de manière compensatoire ou non. Un langage d'interrogation nommé SQLf a ainsi été proposé pour exploiter l'expressivité des requêtes floues (SMITS, PIVERT, & GIRAULT, 2010).

c. BARTEL SAMUEL : à son tour, affirme que chaque utilisateur possède son propre contexte flou où chaque terme défini est associé à un attribut et à une base précis et donc, nous pouvons dire que l'utilisation de termes flous est plus souple et plus proche du raisonnement humain: on ne cherche plus à savoir si des éléments satisfont ou pas une condition, mais à quel degré ils la satisfont. Nous allons maintenant voir comment modéliser un terme flou (BARTEL, 2006).

d. ANNE LAURENT : Quant à lui, il estime que les données issues du monde réel sont souvent entachées d'imperfection et les requêtes que les utilisateurs souhaitent poser ont souvent des formulations vagues. La théorie des sous-ensembles flous permet d'améliorer les systèmes d'information, notamment pour l'interprétation qualitative de données

quantitatives (numériques). Les informations présentées à l'utilisateur (sous forme de règles par exemple) sont plus compréhensibles et se généralisent mieux (ANNE, 2002).

Quant à nous, nous prenons la position de nos prédécesseurs en adhérant à leur paradigme. L'objectif ici est de montrer l'utilité des bases de données floues au travers d'un cas concret tout en tenant compte des notions évoquées ci-haut.

Pour ce qui est de la méthode, d'abord analyser la base de données de l'agence pour ensuite en implémenter des requêtes flexibles (faisant d'elle une base de données floue), grâce au gestionnaire PostgreSQL sur quoi nous avons ajouté une surcouche de support pour les requêtes flexibles à l'aide du langage Structured Query Language fuzzy (SQLf).

Ce travail, outre l'introduction et la conclusion générale, il comprend trois chapitres :

- Le premier chapitre traite sur les généralités et est consacré à la présentation des concepts de base des ensembles flous et des bases de données, puis à la présentation des principales approches de modélisation des requêtes flexibles. Aussi au développement théorique du problème, de la solution et de la méthode.
- Le deuxième chapitre portera sur la présentation du lieu d'investigation et les besoins d'analyse ou interrogation floue.
- Enfin le troisième chapitre portera sur la construction de la solution, accompagnée du test et de la validation de la solution.

Chapitre I. DEVELOPPEMENT THEORIQUE

Dans cette partie du travail, il est question de développer théoriquement le problème ainsi que les solutions qui vont avec, sans oublier toutes les approches liées à la logique floue et ses applications sur les ensembles étant donné tous ces outils nous aideront dans la construction de notre solution.

I.1. Développement théorique du problème

En ce qui concerne l'interrogation classique d'une base de données relationnelle, le premier problème est que cette interrogation est qualifiée par "interrogation booléenne" dans la mesure où l'utilisateur formule une requête, avec le SQL par exemple, qui retourne un résultat ou rien du tout. Cette interrogation pose un problème pour certaines applications. En effet, l'utilisateur doit connaître tous les détails sur le schéma et sur les données de la base de données (BD). Le deuxième problème est que l'interrogation booléenne ne permette pas à l'utilisateur ni d'utiliser des termes linguistiques vagues et imprécis dans les critères de qualification des données recherchées ni d'exprimer des préférences entre ces critères, ce qui est souvent une demande légitime des utilisateurs finaux.

Pour illustrer ce problème, considérons un utilisateur qui consulte une BD d'offres de location de biens immobiliers. L'utilisateur souhaite, d'une part trouver un appartement de préférence dans la 2ième ou la 3ième Avenue au quartier industriel ayant une surface "moyenne" et un loyer "modérée" avec une place de parking si possible.

D'autre part, l'utilisateur souhaite que ses préférences soient considérées selon l'ordre décroissant : Quartier, Avenue, Loyer,

Surface et Garage. Ainsi, les données retournées par le système de gestion de base de données doivent être ordonnées et présentées à l'utilisateur selon ces préférences. Sans cette flexibilité, l'utilisateur doit affiner ces critères de recherche jusqu'à obtenir éventuellement satisfaction puisqu'il n'a pas des connaissances à priori sur les données qu'il consulte. Ce problème est analogue à celui des moteurs de recherche sur Internet (BEN HASSINE, Contribution à l'implémentation d'une base de données floue sous un système de gestion de base de données relationnel, 2005).

I.2. Développement théorique de la solution proposée

Plusieurs extensions du modèle relationnel et du langage SQL ont été proposées pour introduire une certaine incertitude dans la modélisation des données et une certaine flexibilité dans l'interrogation des BD (BEN HASSINE, Contribution à l'implémentation d'une base de données floue sous un système de gestion de base de données relationnel, 2005).

Pour résoudre le problème évoqué au point I.1, nous proposons d'ajouter une couche à la base de données relationnelle pour soutenir les requêtes floues à l'aide de Structured Query Language fuzzy (SQLf), une extension floue du Structured Query Language (SQL) sous le gestionnaire PostgreSQL 9.5.19. La figure 1 illustre notre solution.

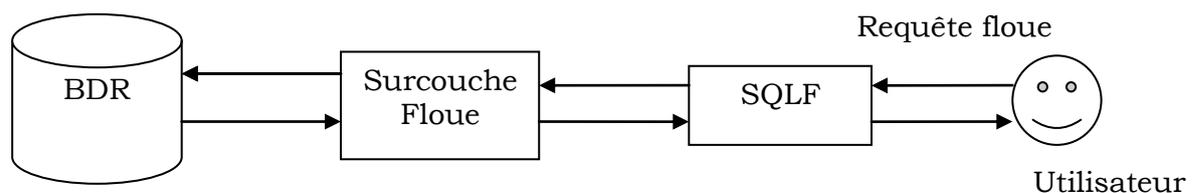


Figure 1. Schéma de la solution

L'utilisateur lance des requêtes flexibles directement vers la base de données relationnelle sans passer par des dérivations à l'aide du Structured Query Language fuzzy (SQLf), ces requêtes seront interprétées grâce à la couche floue intégrée à la base de données pour enfin de compter retournée des résultats à l'utilisateur.

I.3. Les prérequis

Aborder les bases de données floues sans une connaissance préalable sur la logique floue, sur la théorie des ensembles flous, sur les types d'information de la dite base de données est une aberration, raison pour laquelle nous présentons ici les notions requises qui nous serviront de base dans le troisième chapitre dédié à la construction de la solution.

I.3.1. La logique floue

Les connaissances dont disposent les humains sur le monde ne sont presque jamais parfaites. Ces imperfections peuvent être distinguées en deux classes :

- Incertitudes pour désigner les connaissances dont la validité est sujette à question. Par exemple, si nous savons qu'une personne s'est cognée la tête sur le plafond, nous devinons qu'il est probable qu'elle soit très grande.
- Imprécisions pour désigner les connaissances qui ne sont pas perçues ou définies nettement. Par exemple, au lieu de dire qu'une personne mesure 2 mètres et 3 centimètres, nous disons usuellement que cette personne est très grande.

Comment faire en sorte d'exprimer ces imprécisions et ces incertitudes en termes logiques ?

En logique classique, une proposition est vraie ou fausse. Exemple : C'est personne est grande. Vrai ou faux ? (pas flexible)

En logique multivaluée, une proposition peut avoir plusieurs valeurs. Exemple (ternaire) : Cette personne est grande. Vrai, à moitié vrai ou faux ? (un peu plus flexible)

En logique floue, une proposition peut avoir autant de valeurs que l'on veut. Exemple : Cette personne est grande. C'est vrai à 30% (flexible) (DERNONCOURT, introduction à la logique floue appliquée aux systèmes décisionnels, 2011).

La logique floue est une extension de la logique classique qui permet la modélisation des imperfections des données et se rapproche dans une certaine mesure de la flexibilité du raisonnement humain (DERNONCOURT, introduction à la logique floue, 2012).

La logique floue est une théorie qui, appliquée dans un ordinateur, permet de gouverner une régulation, comme le ferait manuellement un opérateur expert (BENJAMIN, XAVIER, & FLORENT, 2001).

Exemple : quelle sera la décision du montant du pourboire à l'issue d'un repas au restaurant, en fonction de la qualité du service ressentie ainsi que de la qualité de la nourriture (exemple souvent utilisé pour introduire à la logique floue).

La logique floue, ou plus généralement le traitement des incertitudes, a pour objet d'étude la représentation des connaissances imprécises et le raisonnement approché. On peut donc la situer à côté des heuristiques de résolutions de problèmes, des systèmes experts, de l'apprentissage, de l'intelligence artificielle distribuée et même du

traitement de la langue naturelle, domaines qui composent les techniques d'intelligence artificielle au sein des sciences cognitives.

Dans les problèmes de prise de décision, d'aide au diagnostic et plus généralement dans tous les systèmes à base de connaissances, on souhaite, à partir d'observations, parvenir à une conclusion qui peut être la détermination d'un objet ou une action à prendre. Or lors du fonctionnement de ces systèmes, interviennent des connaissances mal définies, mal décrites et imparfaitement connues, puis au niveau des règles d'inférence, intervient un traitement imparfait et incomplet du déroulement de la déduction, enfin survient le problème du traitement des contradictions et de la fusion (agrégation) de données voisines. Tous les problèmes concrets sont, en fait, confrontés aux notions d'incertitude et d'imprécision. Ces deux notions sont habituellement mêlées et c'est essentiellement l'observation statistique qui induisait, jusqu'à présent dans la pratique, la mesure probabiliste des incertitudes (GACOGNE, 2003).

La logique floue est une extension de la logique booléenne créée par Lotfi ZADEH en 1965 en se basant sur sa théorie mathématique des ensembles flous, qui est une généralisation de la théorie des ensembles classiques. En introduisant la notion de degré dans la vérification d'une condition, nous permettons à une condition d'être dans un autre état que vrai ou faux. La logique floue confère ainsi une flexibilité très appréciable aux raisonnements qui l'utilisent, ce qui rend possible la prise en compte des imprécisions et des incertitudes. Un des intérêts de la logique floue pour formaliser le raisonnement humain est que les règles sont énoncées en langage naturel. Un système complet de règles basées sur la logique floue et permettant de prendre des décisions est appelé **un système d'inférence flou** (DERNONCOURT, introduction à la logique floue, 2012).

La notion d'appartenance est très importante car la logique floue se base sur le concept d'appartenance floue. Cela signifie simplement que l'on peut appartenir par exemple à 0,8 à un ensemble, contrairement à la théorie des ensembles classiques où comme nous venons de le voir l'appartenance est soit 0 (n'appartient pas) ou 1 (appartient). En introduisant la notion de degré dans la vérification d'une condition, permettant ainsi à une condition d'être dans un autre état que vrai ou faux, la logique floue confère une flexibilité très appréciable aux raisonnements qui l'utilisent, ce qui rend possible la prise en compte des imprécisions (DERNONCOURT, introduction à la logique floue appliquée aux systèmes décisionnels, 2011).

La logique trouve tout son sens dans son application sur les ensembles. Après avoir dit un mot sur la logique floue, nous abordons dans les lignes qui suivent la théorie des ensembles flous.

I.3.2. La théorie des ensembles flous

L'idée de la logique floue est de transmettre cette richesse du raisonnement humain à un ordinateur. En effet, c'est une logique qui ne se limite plus à deux valeurs de vérité mais à une infinité de valeurs dans l'intervalle $[0,1]$, ce qui permet de traduire des affirmations du type un peu vrai, très faux, autant vrai que faux. En effet, un objet peut appartenir à un ensemble et en même temps à son complément. Ainsi, un individu de 1m63 pourra être à la fois grand et petit.

La logique floue repose sur la théorie des ensembles flous, qui est une généralisation de la théorie des ensembles classiques. Dire que la théorie des ensembles flous est une généralisation de la théorie des ensembles classiques signifie que cette dernière n'est qu'un cas particulier la théorie des ensembles flous. Pour faire une métaphore en langage ensembliste, la théorie des ensembles classiques n'est qu'un

sous-ensemble de la théorie des ensembles flous. **"La théorie des ensembles classiques n'est qu'un sous-ensemble de la théorie des ensembles flous"**. Par abus de langage, suivant les us de la littérature, nous utiliserons indifféremment les termes sous-ensembles flous et ensembles flous. Les ensembles classiques sont également appelés ensembles nets, par opposition à flous, et de même la logique classique est également appelée logique booléenne ou binaire (DERNONCOURT, introduction à la logique floue, 2012).

Comme objectif, l'utilisation de la logique floue permet d'écarter et de définir les nuances, d'explicitier les choix, d'affiner les modèles grâce à sa flexibilité et sa gradualité, de modéliser le langage naturel en respectant la réalité des tendances. En fait, c'est une théorie mathématique qui modélise les notions vagues du langage naturel, pour palier l'inadéquation de la théorie des ensembles classiques (BEN HASSINE, contribution à l'implémentation d'une base de donnée floue sous un système de gestion de base de donnée relationnel, 2005).

a. Les ensembles flous

Basés sur la logique floue, les ensembles flous ont été introduits pour définir les ensembles qui n'ont pas de limites précises. Il y a donc une transition graduelle et non brutale entre l'appartenance complète et la non-appartenance (Zadeh, Fuzzy sets. Information and control,, 1965).

Un ensemble F d'univers X est défini par une fonction d'appartenance μ_F qui à chaque élément x de X attribue une valeur de l'intervalle $[0,1]$.

$$F = \{(x, \mu_F(x)); x \in X, \mu_F(x) \in [0, 1]\}$$

Cette valeur ($\mu_F(x)$) représente le degré d'appartenance de x à l'ensemble F . Par définition, si ($\mu_F(x) = 0$) alors x n'appartient pas du tout à F et plus ($\mu_F(x)$) se rapproche de 1, plus la valeur de x appartient à F (si ($\mu_F(x) = 1$) alors x appartient complètement à F) (LAMBALGEN, 2000).

Soit X un ensemble. **Un sous-ensemble flou** A de X est **caractérisé** par une **fonction d'appartenance** $f^a : X \rightarrow [0,1]$. Cette notation veut simplement dire que quel que soit l'entrée X donnée à la fonction f^a , sa sortie est un réel entre 0 et 1. En théorie, il est possible que la sortie soit supérieure à 1, mais en pratique cela n'est quasiment jamais utilisé (DERNONCOURT, introduction à la logique floue, 2012).

b. Propriétés

Soit X un ensemble, A un sous-ensemble flou de X et μ_A la fonction d'appartenance le caractérisant,

1. la hauteur (noté $h(A)$) : correspond à la borne supérieure de l'ensemble d'arrivée de sa fonction d'appartenance $h(A) = \sup\{\mu_A(x) \mid x \in X\}$.

A est dit normalisée ssi $h(A) = 1$

2. le support de A est l'ensemble des éléments de X appartenant au moins un peu à A ; $\text{supp}(A) = \{x \in X \mid \mu_A(x) > 0\}$.
3. Le noyau de A est l'ensemble des éléments de X appartenant totalement à A ;
 $\text{noy}(A) = \{x \in X \mid \mu_A(x) = 1\}$.
4. Une α -coupe de A est le sous-ensemble classique des éléments ayant un degré d'appartenance supérieur ou égal à α ;
 $\alpha\text{-coupe}(A) = \{x \in X \mid \mu_A(x) \geq \alpha\}$.

La figure 2 illustre mieux les propriétés ci-haut : pour une fonction d'appartenance $\mu_A(x)$ de forme trapézoïdale, le support, la hauteur et le noyau sont définis comme suit :

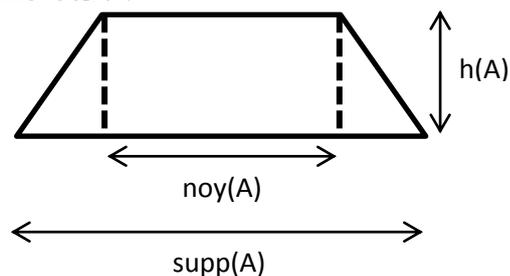


Figure 2 : Forme trapézoïdale de $\mu_A(x)$

L'utilité des sous-ensembles flous repose sur l'identification d'un univers de discours convenable et la détermination des fonctions d'appartenance appropriées (BEN HASSINE, contribution à l'implémentation d'une base de donnée floue sous un système de gestion de base de donnée relationnel, 2005).

c. Variable linguistique

Étant donné que les termes sont moins précis que les nombres, le concept de variable linguistique se révèle ainsi approprié pour la description des connaissances imprécises et vagues.

Une variable linguistique est une fonction dont le domaine est un ensemble classique, et les valeurs sont des ensembles flous (LAMBALGEN, 2000). Chaque variable linguistique est représentée par un quintuple :

$(x, T(x), U, G, M)$ où

- x est le nom de la variable,
- $T(x)$ est l'ensemble des valeurs linguistiques que peut prendre x ,
- U est l'univers de discours associé avec la valeur de base,
- G est la règle syntaxique pour générer les valeurs linguistiques de x ,
- M est la règle sémantique pour associer un sens à chaque valeur linguistique.

Exemple

La variable linguistique $x =$ taille moyenne peut être définie avec un ensemble des termes : $T = \{\text{petite, moyenne, grande}\}$ qui forment son univers de discours $U = [30\text{cm}, 220\text{cm}]$. La variable de base est taille. Le terme petite représente une valeur linguistique. Nous pouvons l'interpréter, par exemple comme "les tailles plus petites que 150cm".

Les fonctions d'appartenance trapézoïdales sont les plus utilisées dans la littérature pour exprimer l'imprécision des termes linguistiques. Cependant, il est difficile d'utiliser la même fonction d'appartenance pour modéliser un terme linguistique donné dans toutes les circonstances. Ainsi, il n'existe pas de concepts linguistiques ayant des distributions universelles. (BEN HASSINE, contribution à l'implémentation d'une base de donnée floue sous un système de gestion de base de donnée relationnel, 2005).

d. Les modificateurs linguistiques

Toutes les descriptions imprécises d'une même variable linguistique V ne peuvent pas être décrites par la liste Tv . C'est pourquoi, nous préférons recourir à des caractéristiques intermédiaires à l'aide des modificateurs tels que "très", "environ" qui permettent de moduler une description en l'atténuant ou en la renforçant. Ces modificateurs sont eux aussi définis par des sous-ensembles flous (HACHANI, 2001).

e. Fuzzification et Défuzzification

La fuzzification est le processus de conversion d'une donnée précise en une donnée floue. Elle consiste à définir les fonctions d'appartenance de toutes les variables précises (d'entrée) et à déterminer le degré d'appartenance de chacune d'elles (BEZDEK, 1993).

D'une façon similaire, la défuzzification se définit comme le processus de passer d'une donnée floue (variable linguistique) en une donnée exacte. Plusieurs méthodes sont utilisées dans la défuzzification, la plus souvent utilisée est le calcul du centre de gravité. La fuzzification et la défuzzification sont utilisées également dans les bases de données (HACHANI, 2001).

Dans ce cadre, elles consistent à assouplir les critères de recherche figurant dans les requêtes au moyen de sous-ensembles flous et transformer ainsi la requête originale en une requête flexible (et

inversement). Ceci donne à l'utilisateur la possibilité d'avoir une image plus réaliste des réponses à sa requête. Dans le cas, où aucun élément ne satisfait pas complètement les conditions de la requête, l'utilisateur dispose des réponses les plus proches de la réponse désirée.

Maintenant que les différentes bases sur la logique floue et sur la théorie des ensembles flous sont posés, il sied d'aborder les différents types d'informations que traitent les bases de données floues ; Les lignes suivantes seront consacrées quant à ce.

I.3.3. Types d'informations d'une base de données floue

Il existe plusieurs types d'informations pour une base de données floue. Dans la suite, une liste non exhaustive est présentée :

a. Information imparfaite

Une partie importante de toutes les informations collectées par l'homme est intrinsèquement **imparfaite**. Cette imperfection découle de la façon dont les humains utilisent le langage naturel pour communiquer, penser, se comporter et travailler. Par exemple, pour conduire une voiture, personne n'a besoin d'informations parfaites sur le fait que «le prochain chemin de traverse est exactement à 214,83 mètres », que« nous devrions commencer à utiliser les freins à exactement 1 minute et 43 secondes ", etc. C'est juste la capacité humaine à faire des abstractions et à estimer par exemple le temps et la distance qui nous permettent d'effectuer des tâches complexes telles que la conduite d'une voiture.

Les imperfections de l'information peuvent être classées comme suit (TRE & SLAWOMIR, 2015) :

- Les informations sont *imprécises* si elles ne sont pas spécifiées aussi précisément qu'elles devraient l'être.
- L'information est *floue* si elle est intrinsèquement vaguement décrite.

- L'information est *incertaine* si elle n'est pas connue avec certitude.
- Les informations sont *incomplètes* si certaines données sont manquantes.
- Les informations sont *incohérentes* s'il y a deux ou plusieurs déclarations contradictoires.

b. Informations incomplètes et manquantes

Avec le concept d'**incomplétude**, nous constatons le manque d'une partie des informations. Des informations sont **manquantes** si, pour certaines parties, aucune description n'est disponible, même pas une description imprécise, vague ou incertaine. Notez qu'en cas d'imprécision, ou incertitude. Cependant, dans la classification considérée, la notion «d'informations manquantes» est utilisé pour désigner uniquement les cas où, pour le sujet considéré, l'information est disponible à tous.

Quatorze sources différentes d'incomplétude ont été identifiées. Dans la recherche scientifique, ces sources sont généralement réduites aux cinq suivantes:

- Les données *n'existent pas* ou *ne s'appliquent pas*. Par exemple, les scores d'un examen n'existent pas avant la fin de l'évaluation.
- Données *inconnues*. Les données existent mais ne sont pas connues de la personne qui doit les traiter.
- *Aucune* information. On ne sait rien. Il est possible que les données existent, mais il est également possible que ce ne soit pas le cas.
- Les données sont connues mais *ne peuvent pas être saisies*. Par exemple, pour des raisons de sécurité.
- Les données ne sont que *partiellement fournies*. Les données sont parfaitement connues, mais ne sont entré dans le système. Par exemple, les codes ABC pour les résultats des examens.

Le quatrième cas de données qui ne peuvent pas être saisies sera généralement traité par le système de gestion de base de données. Dans le cinquième cas de données qui ne sont que partiellement fournies, les

données résultent généralement d'un processus d'agrégation et peuvent donc être considéré comme des données dérivées (TRE & SLAWOMIR, 2015).

c. Informations incohérentes

L'incohérence (ou l'ambiguïté) décrit une situation où deux descriptions ou plus sont contradictoires, par exemple «John est 1m 72» et «John est plus grand que 1m 80». Dans ces cas, il n'est pas possible de combiner les données de telle manière qu'un compromis se présente.

Une solution possible consiste à supprimer les informations de la source la moins fiable (sous l'hypothèse que cette source est connue). Une telle information incohérente peut être formellement représentée comme, par exemple, " $X = 175$ ET $X = 185$ ".

d. Notion d'incertitude

Comme les concepts d'*imprécision* et de vague, le concept d'**incertitude** désigne un manque d'information. Les informations imprécises et vagues ne sont pas suffisamment spécifiques mais est supposé être précis. Nous désignerons explicitement les informations comme incertaines si la confiance en elle est limitée. Un exemple d'informations incertaines est si John raconte nous que la hauteur de Jack est de 175 cm mais nous ne croyons pas si John nous dit vérité. Une telle information incertaine peut être formellement représentée comme « $X = 175$ est probable".

e. Notion d'imprécision et de vague

Les notions d'*imprécision* et de vague dénotent une restriction en raison de laquelle l'information ne peut être décrite qu'en partie. L'incapacité de donner une description exacte peut être due, par exemple, à des erreurs d'arrondi dans les calculs, à un manque de connaissances ou aux limites des équipements d'observation ou de mesure. Si les données imparfaites sont approximativement modélisées par des limites

précises comme déviations ou marges de défaut, on parle alors d'**imprécision**. De cette façon la longueur du corps d'une personne pourrait par exemple être indiquée comme «comprise entre 175 cm et 180 cm » et le prix d'un objet pourrait être décrit comme 500 ± 10 Euro.

Des telles informations imprécises peuvent être formellement représentées par « $X \in [175,180]$ ». L'**imprécision** du concept est associée à l'incapacité de décrire l'information approximativement à l'intérieur de limites précises et représente donc une sorte sa propre imprécision. Pour gérer le flou, les gens utilisent généralement des termes linguistiques pour décrire l'imprécision inhérente. Exemples de descriptions vagues de la hauteur d'une personne sont les termes linguistiques «court», «plutôt court», «grand» et «très grand». Examen-pleins de descriptions vagues des indications de prix sont les termes linguistiques «très bon marché», «bon marché», «cher» et «inabordable». Une telle information vague peut être formellement représentée comme « $X \text{ IS } \textit{court}$ ». D'un point de vue sémantique, ces termes linguistiques ont un caractère *conjonctif* car ils représentent généralement plusieurs valeurs candidates. À ce titre, il existe exemple des longueurs multiples qui correspondent en même temps au terme «court» (TRE & SLAWOMIR, 2015).

Les notions essentielles et requises abordées, les bases de données floues deviennent accessibles.

I.4. Les bases de données floues

I.4.1. Présentations générales

Lorsque vous utilisez un logiciel informatique ordinaire pour le traitement et la gestion, les utilisateurs sont presque toujours obligés de décrire les données de manière parfaite. C'est parce que la modélisation des données se fait généralement au moyen de structures de données et de modèles de données qui sont développés pour modéliser les informations de manière parfaite, en tenant compte des limitations du

ystème informatique. Ce qui est bien sûr une conséquence directe de la nature binaire inhérente au stockage et au traitement des données.

Il y a encore quelques années, les restrictions qui accompagnent une telle approche ne valaient pas la peine d'être mentionnés. D'autant plus que par exemple aussi en sciences exactes comme la physique, nous observons que les gens recherchent la certitude et la précision et faire avec succès des abstractions de la réalité pour développer des théories basées sur une parfaite l'information est un monde idéal (TRE & SLAWOMIR, 2015).

Avec l'introduction de logiques à plusieurs valeurs (Black, 1937) (et plus tard aussi (Rescher, 1969)) un changement de la vision du monde qui repose sur une base définie s'est produit, ce qui constitue entre autres le début des ensembles dits *flous théorie* (ZADEH, 1965), *théorie des possibilités* (Zadeh, ensembles flous comme base d'une théorie de la possibilité, 1978) (DUBOIS & PRADE, 1988) et *logique floue* (ZADEH, Logique floue et raisonnement approximatif, 1975). Ces théories sont à ce jour suffisamment avancés pour fournir un cadre très pratique et adéquat pour le traitement formel d'informations *imprécises*, *vagues* et *incertaines*.

La théorie des ensembles flous est bien adaptée pour la modélisation des imprécises et vagues informations, alors que la théorie des possibilités peut être utilisée pour le traitement des informations incertaines et incomplètes.

Les bases de données sont un élément très important des systèmes informatiques car ce sont les sources d'informations (structurées) pour de nombreuses applications. Les systèmes de gestion ne permettent que de modéliser et de gérer efficacement des informations parfaites. Les développements de la théorie des ensembles flous, de la théorie des possibilités et de la logique floue ont également déclenché la recherche de modèles de bases de données avancés et de techniques qui

peuvent en outre traiter des informations imparfaites. Il en résulte la solution appelée **bases de données floues** qui visent à saisir des informations imparfaites sur une partie du monde et la représenter directement dans une base de données, de préférence que possible et en préservant sa sémantique (TRE & SLAWOMIR, 2015).

En fournissant également un niveau des possibilités de modélisation de base de données pour modéliser des données imparfaites aussi adéquates que possible, une plus grande flexibilité des systèmes de bases de données pourrait être obtenue. Une telle approche donne des bases de données sémantiquement plus riches. Si la modélisation des imperfections est basée sur *la théorie des ensembles flous* on parle d'une **base de données "floue"**.

RADECKI a été l'un des premiers à reconnaître le pouvoir de la théorie des ensembles flous en ce qui concerne la gestion de l'information (RADECKI, 1975). Après RADECKI, plusieurs autres chercheurs ont proposé des modèles de base de données «flous».

I.4.2. Caractéristiques générales d'un modèle de base de données floues (BDF)

Les aspects les plus importants de l'information vague que nous traitons habituellement sont : l'incertitude et l'imprécision. Le premier est dérivé des appréciations réalisées dans mon observation de la réalité.

Exemple : "Il me semble que Zied est un professeur d'Université". Cependant, l'imprécision signifie à travers l'énoncé des concepts qui ne sont pas bien différenciés ou définis.

Exemple : "Zied est jeune", où jeune est un concept dont la sémantique peut varier suivant celui qui l'utilise et qui implique un ensemble de valeurs possibles pour l'âge de Zied.

Il est habituel que, dans nos conversations, plusieurs aspects se mélangent et c'est évident que les énoncés ont montré dans les deux exemples qu'ils fournissent un degré d'information sur le monde réel. Dans quelques cas, l'information qu'ils fournissent peut devenir insuffisante, mais dans d'autres peut ne pas l'être. Toutes ces considérations ont mené à poser une convention que doit satisfaire un SGBD basé sur ces modèles. Cette convention possède les caractéristiques suivantes (PONS, 1995) :

1. Le système doit fournir les mécanismes appropriés pour représenter l'information floue dans toutes ses formes.
2. Il doit offrir le cadre adéquat pour représenter et entreposer la signification de l'information floue qu'il héberge.
3. Il doit fournir un ensemble minimum d'opérateurs pour récupérer et traiter l'information floue.
4. Il doit satisfaire le plus possible les exigences du modèle relationnel. Comme nous avons vu, le modèle relationnel présente une série de caractéristiques qui ne sont pas complètes dans tout système de BD commercial.

Il sera pris en compte, pour ce système, de penser que les modèles flous de BDR présentent, également, différentes adaptations à ce modèle.

I.4.3. Principaux modèles de bases de données floues

Le problème de la représentation et du traitement des informations "imprécises" a été étudié largement par plusieurs auteurs. Cependant, tous les modèles publiés pour donner la solution à ce problème ont leurs avantages, inconvénients et leurs limitations. Aussi, le terme "imprécision" inclut plusieurs significations ; Il est peut être intéressant de distinguer : que l'information que nous avons est incomplète, que nous ne savons pas si elle est certaine ou pas (incertitude), que nous l'ignorons totalement (inconnue) ou qu'elle n'est pas applicable à une certaine entité (indéterminée). Ces significations ne

sont pas disjonctives quelquefois, mais peuvent plutôt être unies dans une certaine information (GALINDO, 1999). Le problème n'est pas trivial, il est nécessaire de modifier la structure des relations et, avec celles-ci, les opérations définies sur elles. Pour permettre d'entreposer l'information imprécise et de la consulter d'une façon flexible, cette information nécessite l'étude d'une multitude de cas particuliers qui ne se produisent pas dans le modèle classique. Plusieurs modèles sont apparus dans la littérature, nous pouvons citer :

- le modèle Relationnel Flou (PONS, 1995),
- le modèle de Buckles-Petry (BUCKLES & PETRY, 1982),
- le modèle d'Umano-Fukami (UMANO, FUKAMI, MIZUMOTO, & TANAKA, 1980),
- le modèle de Prade Testemale (PRADE & TESTEMALE, 1987),
- le modèle de Zemankova-Kaendel (ZEMANKOVA LEECH & KANDEL, 1985)

I.4.4. Requêtes flexibles et requêtes floues

Il existe une utilisation répandue des technologies pour gérer le multimédia et les grandes bases de données (par exemple, les bases de données SIG et les bases de données biologiques). De plus, les nouveaux développements de la technologie des réseaux et internet exigent pour les bases de données distribuées qui sont connectées les unes aux autres et créent des sources de données techniques. En raison du nombre et du volume croissant de bases de données, une bonne accessibilité précise à une base de données devient encore plus importante. Plus d'une recherche ont déjà été effectuée pour améliorer l'accès à la base de données. Dans cette recherche, de nombreux aspects ont été traités, parmi lesquels on mentionne l'organisation des fichiers, l'indexation, techniques d'interrogation, langages d'interrogation et autres techniques d'accès aux données (TRE & SLAWOMIR, 2015).

Les techniques destinées à rendre la requête de base de données plus flexible pour les utilisateurs sont généralement appelées techniques d'**interrogation flexible**. Ces techniques comprennent entre autres:

- Des systèmes d'interrogation à correction automatique qui peuvent corriger les erreurs syntaxiques et sémantiques dans les formulations de requêtes.
- Des systèmes de recherche de navigation qui permettent une navigation intelligente à travers les bases de données.
- Des systèmes d'interrogation coopératifs qui prennent en charge les réponses «indirectes» comme les résumés, les réponses conditionnelles et les informations contextuelles pour les résultats.

Un type spécifique de techniques d'interrogation flexibles est basé sur la théorie des ensembles flous (ZADEH, 1965) et la théorie des possibilités qui lui est associée (Zadeh, ensembles flous comme base d'une théorie de la possibilité, 1978) (DUBOIS & PRADE, 1988) et peut donc être appelée technique d'**interrogation floue**. En général, les techniques d'interrogation floue visent à améliorer l'accès à la base de données en introduisant des préférences floues dans les formulations des requêtes (BOSC, KRAFT, & PETRY, 2005). L'introduction des préférences floues dans les requêtes peuvent se faire à deux niveaux: à l'intérieur des conditions de requête et entre les conditions de requête. Des préférences floues sont introduites dans des critères de recherche flexibles et permettent d'exprimer que certaines valeurs sont plus souhaitables que d'autres de manière progressive. Les préférences floues entre les conditions de requête sont exprimées par des degrés d'importance attribués à des conditions de requête indiquant que la satisfaction de certaines conditions de requête est plus souhaitable que la satisfaction des autres.

La recherche sur l'interrogation floue a déjà une longue histoire. Elle a été inspirée par le succès de la logique floue dans la modélisation des propositions de langage naturel. L'utilisation de telles propositions dans les requêtes, à leur tour, semblent être très naturelles pour les utilisateurs système d'information, notamment les systèmes de gestion de bases de données. Plus tard, l'intérêt pour l'interrogation floue a été renforcé par l'omniprésence des applications basées sur le réseau liées aux mots à la mode des technologies de l'information modernes, comme le commerce électronique, e-gouvernement, etc. Ces applications nécessitent évidemment une capacité de requête flexible lorsque les utilisateurs recherchent des biens, des hôtels, etc., qui peuvent mieux être décrit en utilisant des termes en langage naturel comme bon marché, grand, proche de port, etc. Une autre amplification de l'intérêt pour les requêtes floues vient du développement dans le domaine des applications liées à l'entreposage de données et à l'exploration de données (TRE & SLAWOMIR, 2015).

Comme dit tout haut, nous avons étendu le SQL en SQLf pour nous permettre formuler les requêtes floues. Dans les lignes qui suivent, il sera question de donner des détails quant à ce.

I.5. SQLf et l'évaluation de requêtes floues

I.5.1. SQLf (Structured Query Language fuzzy)

Une requête SQLf intègre des prédicats graduels la clause WHERE de la requête. La satisfaction d'un tuple vis-à-vis d'une requête floue étant désormais graduelle (degré de satisfaction dans $[0,1]$) et non plus binaire, la requête peut être complétée par un seuil qualitatif α exprimant le degré minimum de satisfaction acceptable pour qu'un tuple fasse partie de la relation floue résultat et/ou un

seuil quantitatif K indiquant le nombre de tuples attendus. Le squelette d'une requête SQLf est le suivant :

```
SELECT [distinct] [K α|α|K] attributs FROM relations WHERE
condition-floue ;
```

Où condition-floue peut comporter à la fois des prédicats booléens et flous. Cette expression est interprétée comme la sélection floue du produit cartésien des relations de la clause from, la projection des attributs spécifiés dans la clause select (par défaut les tuples dupliqués sont conservés à moins que le mot clé distinct soit spécifié et l'on ne conserve alors que le degré maximal de satisfaction pour chaque résultat) (SMITS, PIVERT, & GIRAULT, 2006).

Les opérateurs relationnels et ensemblistes utilisés dans les requêtes booléennes ont été étendus pour prendre en compte des prédicats flous et retourner des relations floues.

Exemple : `SELECT DISTINCT 12 0.5 annee FROM auto WHERE (annee IS recent)`

1.5.2. Evaluation et interprétation des requêtes floues

Le développement d'un système d'interrogation floue au-dessus d'un SGBDR peut s'effectuer selon trois stratégies. La première, qualifiée de faiblement intégrée, repose sur une couche logicielle développée au-dessus du SGBDR et chargée de récupérer tous les tuples de la BD et de calculer pour chacun d'eux leur degré de satisfaction vis-à-vis des prédicats flous spécifiés par l'utilisateur. Les tuples dont le degré de satisfaction est supérieur ou égal à α sont conservés pour former la relation floue résultat. Cette version du processus d'interprétation d'une requête floue peut être améliorée à

l'aide d'un prétraitement consistant à effectuer une dérivation de la requête floue. Cette étape de dérivation consiste à traduire la requête floue en requête booléenne permettant de retourner uniquement les tuples satisfaisant les contraintes floues à un degré supérieur ou égal à α (alpha) (SMITS, PIVERT, & GIRAULT, 2006).

La seconde stratégie d'implémentation, qualifiée de semi-intégrée, est fondée sur l'utilisation de procédures stockées ou externes qui définissent à la fois les fonctions d'appartenance des prédicats flous et les opérateurs nécessaires au calcul de la relation floue résultat. La troisième stratégie, qualifiée de pleinement intégrée, s'appuie sur une implémentation des opérateurs relationnels flous au cœur même du SGBD.

Ce dernier type d'implémentation, qui est évidemment le plus efficace, nécessite un effort d'implémentation très conséquent et conduit à un système délicat à maintenir sans nuire à l'efficacité du système d'interprétation des requêtes booléennes (SMITS, PIVERT, & GIRAULT, 2006).

Dans le cadre de ce travail, il sera question d'une implémentation semi-intégrée au SGBDR PostgreSQL avec un développement des prédicats flous en tant que procédures stockées et des connecteurs flous en tant que fonctions C. Elle nous permet également d'obtenir directement une relation floue comme résultat de l'exécution de la requête par le SGBDR étant donné que nous voulons que ce dernier puisse nous retourner des résultats telle que décrit au point I.2.

I.5.3. PostgreSQLf

Le module PostgreSQLf permet de manipuler les principaux éléments qui composent une requête SQLf à savoir :

- a. Les prédicats flous : de forme trapézoïdale pour les attributs numériques avec spécification de l'étiquette linguistique associée et des bornes du trapèze (SMITS, PIVERT, & GIRAULT, 2006)

Exemple : `SELECT newTrapezoidalFuzzySet('recente', 2005, 2006, 2008, 2010) ;`

Ou de forme discrète pour les attributs catégoriels avec spécification de l'étiquette linguistique, des catégories concernées et de leurs degrés de satisfaction respectifs (SMITS, PIVERT, & GIRAULT, 2006).

Exemple: `SELECT newDiscreteFuzzySet('française', ['peugeot', 'citroen', 'dacia'], [1.0, 1.0, 0.6]) ;`

- b. Les conditions floues : dans la clause WHERE, où `~=` correspond à l'opérateur IS (SMITS, PIVERT, & GIRAULT, 2006).

Exemple: `SELECT * FROM voitures occasion WHERE annee ~= 'recente' ;`

- c. Les modificateurs : pour modifier le degré d'appartenance retourné par un prédicat ou afin de le rendre plus restrictif ou tolérant (SMITS, PIVERT, & GIRAULT, 2006).

Exemple: `SELECT * FROM voitures occasion WHERE annee ~='tres_recente';`

- d. Les conjonctions et disjonctions : Les opérateurs SQL AND et OR ont été respectivement étendus à l'aide des opérateurs `&&` et `||`, où par défaut l'opérateur minimum est utilisé pour la conjonction et le maximum pour la disjonction. Différentes implémentations (zadeh, probabiliste, lukasiewicz et weber) de ces opérations ont été réalisées et peuvent être sélectionnées à l'aide de l'instruction: `SELECT set_norme('norme')` (SMITS, PIVERT, & GIRAULT, 2006).

Exemple : `SELECT * FROM voitures occasion WHERE annee ~='tres recente' && km ~='faible';`

- e. Les seuils : pour calibrer la qualité des résultats (α) ou la quantité (K)

Exemple : `SELECT set alpha(0.4) ; SELECT set K(20) ;`

- f. Les quantificateurs : intervenant dans des conditions floues du type `QX are A`, où A est un prédicat flou, X un ensemble de tuples et Q (le quantificateur) exprime une opération d'agrégation telle que "la plupart". Différentes méthodes d'interprétation des quantificateurs (Zadeh, CTA et OWA) ont été implémentées et peuvent être sélectionnées à l'aide de l'instruction :

Ex : `SELECT set_quantifier('owa')` ; Le quantificateur « la plupart » est prédéfini (SMITS, PIVERT, & GIRAULT, 2006).

Exemple : `SELECT * FROM voitures occasion WHERE la plupart (annee ~= 'tres recente', km ~= `faible', marque ='peugeot', consommation ~= 'faible', prix ~= 'raisonnable') ;`

- g. Les opérateurs graduels : pour effectuer des inclusions (opérateur `in~`) ou comparaisons (opérateur `~`) graduelles en s'appuyant sur des mesures de distance définies sur le domaine de définition des attributs concernés. La requête suivante permet de récupérer les voitures de marque 'peugeot', 'renault' ou d'une marque proche sémantiquement (selon la mesure de distance définie sur cet attribut) (SMITS, PIVERT, & GIRAULT, 2006).

Exemple : `SELECT * from voiture occasion WHERE annee ~ 2008 && marque in ~('peugeot', 'renault') ;`

Chapitre II. CADRE D'EXPERIMENTATION

Comme annoncé dans notre introduction, ce travail porte sur l'application des bases de données floues dans le domaine de vente et de location des véhicules. Pour ce faire, notre choix s'est porté sur l'agence automobile « RHM_Unit Business » qui fera l'objet de notre étude dans cette partie du travail et lors de la construction de la solution, nous utiliserons sa base de données déjà existante en vue d'en implémenter les requêtes flexibles.

II.1. Cadre géographique

« RHM_Unit Business » est une agence à la fois spécialisée dans la vente et dans la location des véhicules. Elle vend et facilite l'obtention des véhicules de toutes marques. La direction générale de l'agence est située au numéro 32 du quartier plateau II dans la commune annexe, ville de Lubumbashi.

II.2. Organisation et structure

L'organisation au sein de toute entreprise reste une condition sine qua non de l'efficacité et de la rentabilité, pour sa part l'agence est dirigée par l'administrateur général (A.D.G), il est secondé par l'administrateur général adjoint (A.D.G.A) ainsi que la direction générale adjoint.

RHM_Unit Business possède les directions suivantes :

- la Direction générale
- la direction administrative
- la direction comptable et financière,
- la direction technique
- la direction commerciale

Voici la description de ces différentes directions :

a. La direction générale

Elle constitue la haute hiérarchie de l'organisation, elle supervise et planifie l'ordonnancement des opérations, d'approvisionnement des véhicules, elle connaît le plan d'action en tenant compte des besoins de l'exploitation du siège de Lubumbashi.

b. La direction administrative

Elle s'occupe de l'organisation de l'agence, des actionnaires qui dirigent l'organisation du siège, elle assure la fonction administrative.

Elle est chargée de bien mener la politique arrêtée par la direction générale en dotant l'agence du meilleur employé, elle commande et donne des ordres et les résultats seront contrôlés par la direction générale.

c. La direction comptable et financière

Ce service est dirigé par des experts comptables. Il s'occupe de l'évolution financière de l'ensemble de toutes les opérations qui se déroulent à l'intérieur de la société. Il ressort de ce service 2 types de comptabilités :

- La comptabilité générale (CG)
- La comptabilité Analytique d'Exploitation (CAE)

d. La direction Commerciale

Ce service s'occupe de l'achat, de la vente et de la publicité, voir le marketing et comprend deux secteurs :

- Celui de vente ;
- Celui d'achat qui s'occupe de l'approvisionnement en véhicules et du stockage.

e. La direction technique

Ce service s'occupe de la maintenance, de la réparation des éventuels pannes ainsi de l'entretien des véhicules.

La description ci-dessus est mieux illustrée par l'organigramme de la figure 3 :

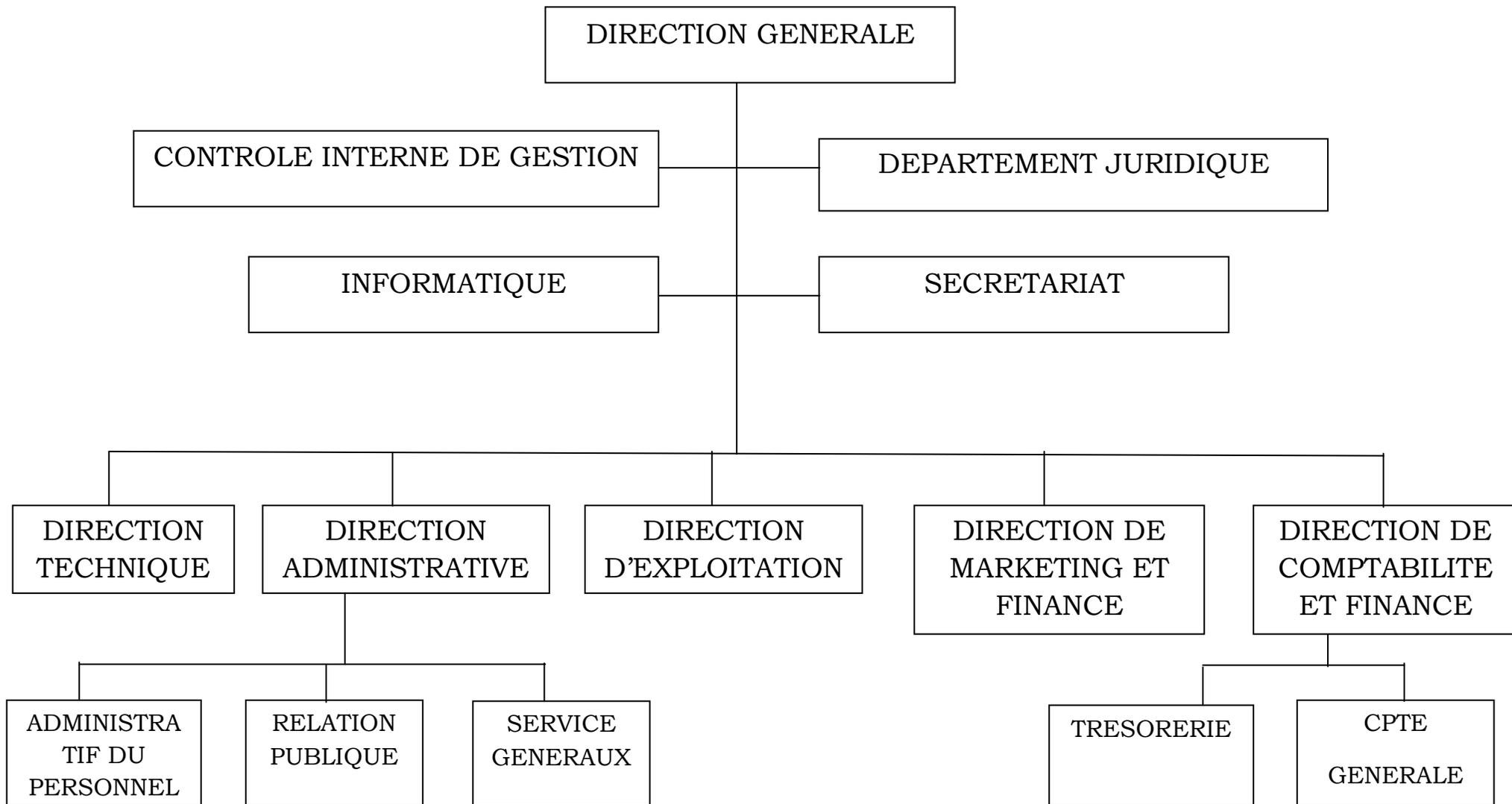


Figure 3. Organigramme de l'agence RHM_Unit Business

Source: RHM_Unit Business, février 2019

II.3. Présentation et discussion autour de la base de données de l'agence

II.3.1. Présentation de la BD Auto de RHM_Unit Business

Dans cette section, nous allons présenter la base de données de l'agence automobile « RHM_Unit Business », ses requêtes pour ensuite justifier notre motivation de migrer vers des requêtes flexibles.

La base de données de l'agence contient une seule table, en l'occurrence la table « auto » possédant les attributs suivants :

- **id** qui est l'identifiant de type entier attribué par l'agence pour chaque véhicule,
- **type** désigne si le véhicule est personnel ou professionnel avec le type texte,
- **désignation** met ensemble la marque et le nom du véhicule de type texte,
- **kilométrage** qui représente le kilométrage de chaque véhicule de type entier,
- **prix** représente le prix du véhicule de type entier,
- **année** désigne l'année de fabrication de type entier,
- **fabricant** représente le nom du fabricant de type texte,
- **nom voiture** représente le nom du véhicule,
- **nom complet** de type texte,
- **modèle** représente le modèle de type texte,
- **longueur** représente la longueur du véhicule de type petit entier,
- **largeur** représente la largeur du véhicule de type petit entier,
- **hauteur** désigne la hauteur du véhicule de type petit entier,
- **nombre de portes** représente le nombre de porte du véhicule de type petit entier,
- **nombre de sièges** précise le nombre de siège du véhicule de

type petit entier,

- **accélération** désigne la limite de l'accélération du véhicule de type numérique,
- **consommation** précise le niveau de consommation du véhicule de type numérique,
- **émission CO2** désigne la quantité d'émission du CO2 du véhicule de type entier.

II.3.2. Requêtes courantes sur la BD de RHM_Unit Business

Voici les différentes requêtes que l'entreprise réalise très souvent sur sa base de données :

- R1 : donner les noms, modèle et prix des véhicules :
SELECT auto.nom, auto.modele, auto.prix
FROM auto;
- R2 : donner les différents fabricants des véhicules :
SELECT DISTINCT auto.fabriquant
FROM auto;
- R3 : donner tous les renseignements sur les véhicules :
SELECT *
FROM auto;
- R4 : donner tous les noms des véhicules fabriqués par OPEL :
SELECT auto.nom
FROM auto
WHERE auto.fabriquant = 'OPEL';

- R5 : donner les voitures dont les prix est compris entre 5000\$ et 10.000\$:

SELECT *

FROM auto

WHERE auto.prix BETWEEN 5000 AND 10000;

- R6 : donner les voitures dont le fabricant est PEUGEOT dont le prix est inférieur à 8000 :

SELECT *

FROM auto

WHERE auto.fabricant = 'PEUGEOT' AND auto.prix <8000;

II.3.3. Nécessite des requêtes flexibles

L'agence rencontre certaines difficultés qui nécessitent une approche différente quant aux formulations des requêtes en tenant compte des préférences du requérant et des résultats attendus. Illustrons ces difficultés par deux exemples : Ex1 : le client désire acheter une voiture avec un kilométrage faible à un prix réduit ;

Ex2 : le client souhaite louer une voiture récente en termes d'année de fabrication avec un degré de satisfaction au-dessus de la moyenne.

En voulant tenir compte de ces préférences, les requêtes SQL présenteront des limites et des insuffisances d'où la nécessité de migrer vers les requêtes flexibles où les préférences, les prédicats flous, le degré de satisfaction qualitatif, l'étiquette linguistique seront définis au préalable.

Après constat des problèmes, la dernière partie s'articulera essentiellement autour des solutions, c'est-à-dire l'implémentation des requêtes flexibles (tout en rendant la base de données floues). Il sied de noter qu'il ne s'agira que de quelques requêtes flexibles considérées comme des prototypes, de ce fait l'agence développera dans la suite d'autres requêtes selon les besoins et les demandes.

Chapitre III. CONSTRUCTION DE LA SOLUTION

III.1. Configuration de la BDR

Notre résultat est obtenu à l'aide d'un système d'interprétation de requêtes floues intégré au cœur d'un SGBDR commercial, en l'occurrence PostgreSQL version 9.5.19.

Comme évoqué ci-haut il s'agit de la même base de données que nous venons de présenter dans le chapitre précédent (aucune modification n'a été apportée). Seulement elle vient d'être créée cette fois-ci sous PostgreSQL puisque c'est avec ce gestionnaire que nous ajouterons notre surcouche.

Dans le souci d'être plus explicite, nous avons divisé cette démarche par des étapes que voici :

- a. Création de la base de données « agenceautorhm » sous postgresql avec la commande :

CREATE DATABASE agenceautorhm ; (illustrer à la figure 4)

```
tabitha@tabitha-HP-ProBook-4430s ~ $ sudo su postgres
postgres@tabitha-HP-ProBook-4430s /home/tabitha $ psql
psql (9.5.19, server 9.5.23)
Type "help" for help.

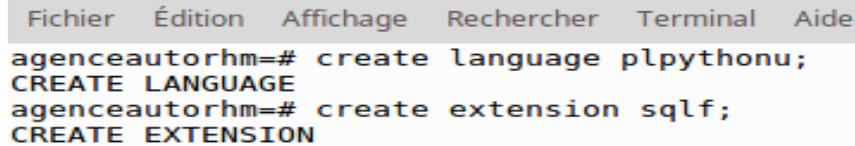
postgres=# CREATE DATABASE agenceautorhm;
CREATE DATABASE
```

Figure 4 : création de la base de données

- b. création du langage plpythonu puisque sqlf repose sur lui l'extension sqlf avec les commandes suivantes :

CREATE LANGUAGE PLPYTHONU;
CREATE EXTENSION SQLF;

La figure 5 illustre ces commandes

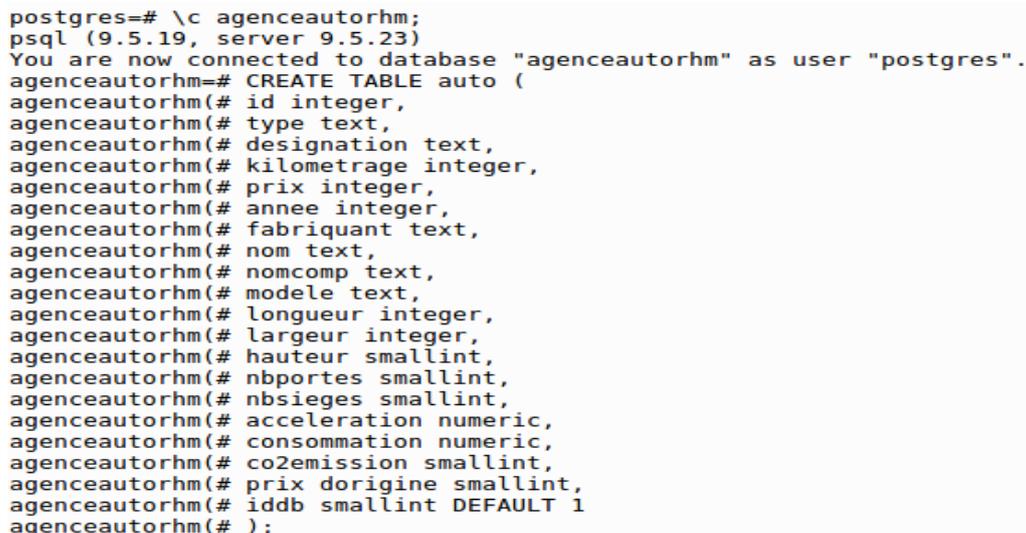


```
Fichier  Édition  Affichage  Rechercher  Terminal  Aide
agenceautorhm=# create language plpythonu;
CREATE LANGUAGE
agenceautorhm=# create extension sqlf;
CREATE EXTENSION
```

Figure 5 : ajout de l'extension sqlf

- c. Création de la table « auto » avec les mêmes attributs et les mêmes types, avec la commande suivante :

CREATE TABLE auto (id integer, type text, designation text, kilometrage integer, prix integer, annee integer, fabricant text, nom text, nomcomp text, modele text, longueur integer, largeur integer, hauteur smallint, nbportes smallint, nbsieges smallint, acceleration numeric, consommation numeric, co2emission smallint); illustrer à la figure 6



```
postgres=# \c agenceautorhm;
psql (9.5.19, server 9.5.23)
You are now connected to database "agenceautorhm" as user "postgres".
agenceautorhm=# CREATE TABLE auto (
agenceautorhm(# id integer,
agenceautorhm(# type text,
agenceautorhm(# designation text,
agenceautorhm(# kilometrage integer,
agenceautorhm(# prix integer,
agenceautorhm(# annee integer,
agenceautorhm(# fabricant text,
agenceautorhm(# nom text,
agenceautorhm(# nomcomp text,
agenceautorhm(# modele text,
agenceautorhm(# longueur integer,
agenceautorhm(# largeur integer,
agenceautorhm(# hauteur smallint,
agenceautorhm(# nbportes smallint,
agenceautorhm(# nbsieges smallint,
agenceautorhm(# acceleration numeric,
agenceautorhm(# consommation numeric,
agenceautorhm(# co2emission smallint,
agenceautorhm(# prix dorigine smallint,
agenceautorhm(# iddb smallint DEFAULT 1
agenceautorhm(# );
```

Figure 6: création de la table auto

- d. Copie des tuples dans la table : il s'agit ici de l'insertion des données dans notre base de données à l'aide de la commande :

COPY auto (id integer, type text, designation text, kilometrage integer, prix integer, annee integer, fabricant

text, nom text, nomcomp text, modele text, longueur integer, largeur integer, hauteur smallint, nbportes smallint, nbsieges smallint, acceleration numeric, consommation numeric, co2emission smallint) FROM stdin WITH DELIMITER ‘;’; (la figure 7 illustre mieux la réalité)

```
Fichier Édition Affichage Rechercher Terminal Aide
agencesautorhm# COPY auto (id, type, designation, prix, kilometrage, annee, fabriquant, nom, nomcomp, modele, longueur, largeur, hauteur, nbportes, nbsieges, accelerati
on, consommation, co2emission, prixdorigine, iddb) FROM stdin WITH DELIMITER ‘;’;
Enter data to be copied followed by a newline.
End with a backslash and a period on a line by itself, or an EOF signal.
>> 265391;particulier;opel meriva;10500;33300;2005;OPEL;Meriva 1.6i;Ecotec Enjoy;MERIVA;4042;1950;1624;5;5;13.3;7.3;175;0;4
265407;particulier;308 berline;16300;17500;2008;PEUGEOT;308 1.6 VTI 120ch;Premium;308;4276;1815;1498;3;5;12.7;6.7;159;0;1
265423;professionnel;toyota yaris (2) 95 vti1 linea terra Sp;5800;58594;2004;TOYOTA;Yaris Verso 85 VTI 1;Linéa Terra;YARIS VERSO;3880;1690;1715;5;5;12.7;6.4;154;0;6
265480;particulier;mini cooper park lane;12900;55000;2006;MINI;Mini 1.6i - 170;Cooper S Park La>> ne;MINI;3655;1688;1416;3;4;9.1;8.6;207;0;1
265518;particulier;406 coupe 2.0l 16s;4000;168000;2000;PEUGEOT;406 Coupé 2.0i;406 COUPE;4615;1780;1352;2;4;10.4;8.3;197;0;3
265529;professionnel;peugeot 807 2.2 navteq;10490;92058;2005;PEUGEOT;807 2.2 16V;Na>> vteq on Board;807;4727;1850;1752;5;5;11.6;9.7;231;0;5
265539;particulier;bmw 118d 143ch 3 portes;18800;40000;2008;BMW;Touring 318i 143ch;Luxe;SERIE 3;4520;1817;1418;5;5;9.5;6.0;144;0;4
265546;particulier;206 xs;4000;12358;2001;PEUGEOT;206 1.4i;XS;206;>> 3835;1652;1432;3;5;15.2;6.6;162;0;6
265551;particulier;jaguar x-type 2.0 d 2004 130cv;10000;148000;2004;JAGUAR;X-Type 2.0i V6;Classique;X-TYPE;4672;2003;1392;4;5;9.4;9.2;219;0;2
265558;particulier;206 xbox noire serie linitee 27000 km (1ère main);10800;> 0;27000;2008;PEUGEOT;206 1.4e 75ch;XBOX 360;206;3835;1652;1432;3;5;12.5;6.4;152;0;6
265566;particulier;206 s16;5500;72000;2001;PEUGEOT;206 2.0e 16V;S16;206;3835;1652;1432;3;5;8.4;7.9;187;0;5
265570;particulier;ave 406 coupé 2.0 pack;6200;132000;1999;PE>> UGEOT;406 Coupé 2.0i;Pack;406 COUPE;4615;1780;1352;2;4;10.4;8.3;197;0;2
265582;professionnel;porsche 997 carrera 4s bv6 3,8 l 2 p - 43500 kms;65000;43500;2006;PORSCHE;911 Carrera 4S Coupé 3.8i;911 TYPE 997;4427;1852;1310;2;2;4.9;11.8;285;0
1;
265608;>> particulier;c4 coupe;17000;10500;2004;CITROEN;C4 Coupé 2.0i 16V 138;VTS;C4;4274;1773;1458;3;5;12.8;7.8;186;0;1
265625;professionnel;peugeot 307 16 v confort pack 1,4 l 3 p - 60350 km;9800;60350;2006;PEUGEOT;307 1.6e 16V;Confort Pack;307;4218;1757;1510;3;>> ;5;10.7;7.4;174;0;5
265627;particulier;clio 3 rs siruis;16500;41556;2000;> 7;RENAULT;Clio TCE 100;Initiale;CLIO III;3986;1707;1496;5;5;13.4;5.9;139;0;5
265642;particulier;c4 picasso collection 110ch;20800;9000;2009;CITROEN;C4 Picasso VTI 120;Collection;C4;4470;1830;1660;5;5;11.1;7.5;177;0;4
265654;particulier;renault mégane 2>> carminat;12500;26000;2008;RENAULT;Mégane II Coupé 2.0T 16V;F1 Team 2;MEGANE II;4228;1777;1437;3;5;6.5;8.5;200;0;1
265656;particulier;clio 3;12500;29000;2007;RENAULT;Clio TCE 100;Initiale;CLIO III;3986;1707;1496;5;5;13.4;5.9;139;0;3
265662;particulier>> er;espace 3 rte;4500;251000;1999;RENAULT;Grand Espace 2.0i 16V;RTE;GRAND ESPACE;4787;1810;1700;5;7;13.7;9.7;227;0;3
265671;particulier;peugeot 2006 xbox;8500;45000;2007;PEUGEOT;206 1.4e 75ch;XBOX 360;206;3835;1652;1432;3;5;12.5;6.4;152;0;6
265682;particulier>> culier;bmw xs ph2 2004 3.0d 218cv pack luxe/sport;17400;166000;2004;BMW;XS 3.0i;Pack Luxe;XS;4667;1872;1715;5;5;8.3;12.7;307;0;5
265688;particulier;microcar mc city s;7500;17000;2008;TOYOTA;Corolla Verso MC 110 VVT-i 5pl;Limited Edition;COROLLA VERSO;436>> 0;1770;1620;5;5;12.7;7.5;178;0;2
265701;particulier;107;9500;33000;2007;PEUGEOT;107 1.0e 12v;Urban Move;107;3430;1630;1470;5;4;13.7;4.6;109;0;5
265709;particulier;polo 1.2 match 2005;6800;60500;2005;VOLKSWAGEN;Polo 1.2 65;Match;POLO 2002 - 3897;1650;1>> 465;5;5;14.9;5.9;142;0;5
265718;particulier;audi tt 1.8 turbo quattro 225cv;19900;149500;2006;AUDI;TT 1.8T;Quattro;TT;4041;1856;1349;3;4;16.8;9.5;228;0;4
265726;particulier;renault scenic ii latitude;10500;76500;2006;RENAULT;Scenic 1.6 16V Euro 4;Latitu>> de;SCENIC II;4259;2087;1620;5;5;12.5;7.2;173;0;4
265744;particulier;twingo zenos 1.2 16v clim;4500;90000;2005;RENAULT;Twingo 1.2 16V;Kenzo;TWINGO;3433;1630;1423;3;4;11.7;5.8;138;0;4
265755;particulier;citroen xsara accidentée;470;10000;1999;CITROEN;Xs>> ara Break 1.4i;X;XSARA;4354;1698;1420;5;5;14.6;6.9;170;0;4
265759;particulier;audi a4 2.4 v6;8800;145000;2002;AUDI;A4 Avant 2.4 V6;Pack Plus;A4 AVANT;4544;1937;1428;5;5;9.0;9.6;230;0;5
265760;particulier;clio rs 2000 ch;14500;29000;2007;RE>> AUDI;Clio TCE 100;Initiale;CLIO III;3986;1707;1496;5;5;13.4;5.9;139;0;4
265766;professionnel;c4 1.6 16v bioflex pack ambiance;13500;16180;2008;CITROEN;C4 Coupé 1.6i 16V 110;Pack Ambiance;C4;4274;1773;1458;5;5;10.6;7.1;169;0;2
265770;particulier;peugeo>> t 308 1.6 vti premium essence 3 portes;14500;24500;2008;PEUGEOT;308 1.6 VTI 120ch;Premium Pack;308;4276;1815;1498;5;5;10.8;6.7;159;0;2
265781;particulier;opel astra gtc sport;10000;80000;2005;OPEL;Astra GTC 1.6 Twinport;Sport;ASTRA;4290;1794;1435;3;5;12.3;>> 6.6;158;0;5
265782;particulier;suzuki grand vitara 119 d luxe;17700;59000;2007;SUZUKI;GRAND VITARA 2.0;LUXE;GRAND VITARA;4470;1810;1695;5;2;12.5;9.1;208;0;4
265792;particulier;pt cruiser pack crom;9500;119000;2004;CHRYSLER;PT Cruiser 2.4 Turbo;Touring;>> PT CRUISER;4288;1748;1601;5;5;11.1;10.0;237;0;3
265797;particulier;bmw serie 1;21000;22000;2008;BMW;125i 218 ch;Confort;SERIE 1;4360;1748;1423;2;4;6.4;7.9;190;0;2
265812;particulier;polo match;3500;107000;2001;VOLKSWAGEN;Polo 1.4i;Match;POLO 2000 - 3743>> -1832;1418;5;5;14.5;6.1;146;0;1
265821;particulier;nissan 350z pack noir 300cv gtie avril2010 16000km;27500;16000;2007;NISSAN;350Z;Pack;350Z;4315;1815;1325;2;4;5.8;11.7;280;0;5
265826;particulier;peugeot 206 s16 noire - 03/2002 - 4990 euros -;4990;1290>> 00;2002;PEUGEOT;206 SW 2.0i 16V;S16;206;4028;1652;1460;5;5;9.9;7.9;189;0;5
```

Figure 7 : copie des tuples dans la table

III. 2. Préparation des requêtes flexibles

Définition et utilisation des prédicats flous et du seuil qualitatif : nous avons défini 3 prédicats :

SELECT create_predicate('recent', 1995., 2011., 0., 0.);

SELECT create_predicate('reduit', 0., 0., 5000.0, 20000.0);

SELECT create_predicate('faible', 0., 0., 1000., 150000.);

Il s'agit des prédicats prédéfinis par l'utilisateur dans le but de satisfaire à sa demande, il s'accompagne toujours d'une étiquette

linguistique et des valeurs qui représentent des bornes trapézoïdales, voir le point I.5.3.

Ils seront donc utilisés dans les conditions floues pour exprimer les souhaits ou désirs de l'utilisateur.

La figure 8 décrit mieux la réalité.

```
agenceautorhm=# SELECT create_predicate('recent', 1995., 2011., 0., 0.);
create_predicate
-----
(1 row)

agenceautorhm=# SELECT create_predicate('reduit', 0., 0., 5000.0, 20000.0);
create_predicate
-----
(1 row)

agenceautorhm=# SELECT create_predicate('reduit', 0., 0., 1000., 150000.);
create_predicate
-----
(1 row)
```

Figure 8 : définitions des prédicats flous

III.3. Interrogation flexible

Dans cette section nous présenterons quelques requêtes flexibles selon les demandes des clients et des utilisateurs.

1^{ière} requête : le client veut avoir les dix premiers véhicules **récents** avec les attributs suivants : fabricant, nom, modèle, prix, kilométrage, année. Elle se traduit en requête flexible par :

```
SELECT DISTINCT fabricant, nom, modele, prix,  
kilometrage, annee, get_mu() as mu  
FROM auto  
WHERE (annee ~= 'recent')  
ORDER BY mu DESC LIMIT 10;
```

La figure 9 illustre mieux la réalité

```
Fichier  Édition  Affichage  Rechercher  Terminal  Aide
tabitha@tabitha-HP-ProBook-4430s ~ $ sudo su postgres
[sudo] password for tabitha:
postgres@tabitha-HP-ProBook-4430s /home/tabitha $ psql agenceautorhm
psql (9.5.19, server 9.5.23)
Type "help" for help.

agenceautorhm=# SELECT DISTINCT fabricant, nom, modele, prix, kilometrage, annee, get_mu() as mu
agenceautorhm=# FROM auto
agenceautorhm=# WHERE (annee <= 'recent')
agenceautorhm=# ORDER BY mu DESC LIMIT 10;
 fabricant |          nom          | modele | prix | kilometrage | annee | mu
-----|-----|-----|-----|-----|-----|-----
CITROEN    | C4 Picasso VTi 120   | C4     | 20800 | 9000        | 2009 | 0.875
BMW        | Touring 318i 143ch  | SERIE 3 | 18800 | 40000       | 2008 | 0.8125
CITROEN    | C4 Coupé 1.6i 16V 110 | C4     | 13500 | 16188       | 2008 | 0.8125
PEUGEOT    | 206 1.4e 75ch       | 206    | 10000 | 27000       | 2008 | 0.8125
PEUGEOT    | 308 1.6 VTi 120ch   | 308    | 14500 | 24500       | 2008 | 0.8125
PEUGEOT    | 308 1.6 VTi 120ch   | 308    | 16300 | 17500       | 2008 | 0.8125
RENAULT    | Mégane II Coupé 2.0T 16V | MÉGANE II | 12500 | 26000       | 2008 | 0.8125
TOYOTA     | Corolla Verso MC 110 VVT-i 5pl | COROLLA VERSO | 7500 | 17000       | 2008 | 0.8125
PEUGEOT    | 107 1.0e 12v        | 107    | 9500  | 33000       | 2007 | 0.75
PEUGEOT    | 206 1.4e 75ch       | 206    | 8500  | 45000       | 2007 | 0.75
(10 rows)
```

Figure 9 : requête n°1

2^{ème} requête : le client veut avoir les neuf premiers véhicules avec des prix **réduits** et des kilométrages **faibles** avec les attributs suivants : fabricant, nom, modèle, prix, kilométrage, année. Elle se traduit en requête flexible par :

SELECT DISTINCT fabricant, nom, modele, prix, kilometrage, annee, get_mu() as mu

FROM auto

WHERE (prix <= 'reduit') && (kilometrage <= 'faible')

ORDER BY mu DESC LIMIT 9;

La figure 10 illustre mieux la réalité

```
agenceautorhm=# SELECT DISTINCT fabricant, nom, modele, prix, kilometrage, annee, get_mu() as mu
agenceautorhm=# FROM auto
agenceautorhm=# WHERE (prix <= 'reduit') && (kilometrage <= 'faible')
agenceautorhm=# ORDER BY mu DESC LIMIT 9;
 fabricant |          nom          | modele | prix | kilometrage | annee | mu
-----|-----|-----|-----|-----|-----|-----
TOYOTA     | Corolla Verso MC 110 VVT-i 5pl | COROLLA VERSO | 7500 | 17000       | 2008 | 0.8333333333333333
PEUGEOT    | 206 1.4e 75ch       | 206    | 8500  | 45000       | 2007 | 0.704697986577181
PEUGEOT    | 107 1.0e 12v        | 107    | 9500  | 33000       | 2007 | 0.7
PEUGEOT    | 206 1.4e 75ch       | 206    | 10000 | 27000       | 2008 | 0.6666666666666666
OPEL       | Meriva 1.6i         | MERIVA | 10500 | 33300       | 2005 | 0.6333333333333333
TOYOTA     | Yaris Verso 85 VVT-i | YARIS VERSO | 5800 | 58594       | 2004 | 0.613463087248322
PEUGEOT    | 307 1.6e 16V        | 307    | 9890  | 60359       | 2006 | 0.60161744966443
VOLKSWAGEN | Polo 1.2 - 65       | POLO 2002- | 6800 | 60500       | 2005 | 0.600671140939597
PEUGEOT    | 206 2.0e 16V        | 206    | 5500  | 72000       | 2001 | 0.523489932885906
(9 rows)
```

Figure 10 : requête n°2

Quantificateur OWA : il fait appel au quantificateur la plupart avec la commande : **SELECT set_quantifier('owa');**

Figure 11 illustre mieux la réalité

```
Fichier  Édition  Affichage  Rechercher  Terminal  Aide
agenceautorhm=# SELECT set_quantifier('owa');
set_quantifier
-----
owa
(1 row)
```

Figure 11 : quantificateur OWA

Le quantificateur OWA nous permettra d'utiliser dans la troisième requête d'utiliser le quantificateur « la plupart ».

3^{ème} requête : le client veut avoir les neuf premiers véhicules dont **la plupart** ont des prix **réduits**, des kilométrages **faibles** avec les attributs suivants : fabricant, nom, modèle, prix, kilométrage, année. Elle se traduit en requête flexible par :

```
SELECT DISTINCT fabricant, nom, modele, prix,
kilometrage, annee, get_mu() as mu
FROM auto
WHERE la_plupart(prix ~= 'reduit', kilometrage ~= 'faible')
ORDER BY mu DESC LIMIT 9;
```

La figure 12 illustre mieux la réalité

```
agenceautorhm=# SELECT DISTINCT fabricant, nom, modele, prix, kilometrage, annee, get_mu() as mu
agenceautorhm=# FROM auto
agenceautorhm=# WHERE la_plupart(prix ~= 'reduit', kilometrage ~= 'faible')
agenceautorhm=# ORDER BY mu DESC LIMIT 9;
```

fabriquant	nom	modele	prix	kilometrage	annee	mu
TOYOTA	Corolla Verso MC 110 VVT-i 5pl	COROLLA VERSO	7500	17000	2008	0.848154362416107
PEUGEOT	107 1.0e 12v	107	9500	33000	2007	0.721308724832215
PEUGEOT	206 1.4e 75ch	206	8500	45000	2007	0.720190156599553
PEUGEOT	206 1.4e 75ch	206	10000	27000	2008	0.706375838926175
TOYOTA	Yaris Verso 85 VVT-i	YARIS VERSO	5800	58594	2004	0.696763982102908
OPEL	Meriva 1.6i	MERIVA	10500	33300	2005	0.670805369127517
VOLKSWAGEN	Polo 1.2 - 65	POLO 2002-	6800	60500	2005	0.670503355704698
PEUGEOT	206 2.0e 16V	206	5500	72000	2001	0.634284116331096
PEUGEOT	307 1.6e 16V	307	9890	60359	2006	0.619713087248322

(9 rows)

Figure 12 : requête n°3

4^{ème} requête : le client veut avoir les quinze premiers véhicules dont les **distances avoisinent** celles des véhicules de **très bons marchés** en terme de prix avec les attributs suivants : fabricant, nom, modèle, prix, kilométrage, année. Elle se traduit en requête flexible par :

```
SELECT DISTINCT fabricant, nom, modele, prix,
kilometrage, annee, get_mu() as mu
FROM auto
WHERE kilometrage :: kilometrage <~~ ARRAY (SELECT
kilometrage :: kilometrage (
FROM auto
WHERE (prix ~= 'reduit'))
ORDER BY mu DESC LIMIT 15;
```

La figure 13 illustre mieux la réalité

```
agenceautorhm=# SELECT DISTINCT fabricant, nom, modele, prix, kilometrage, annee, get_mu() as mu
agenceautorhm=# FROM auto
agenceautorhm=# WHERE kilometrage::kilometrage <~~ ARRAY(SELECT kilometrage::kilometrage
agenceautorhm=# FROM auto
agenceautorhm=# WHERE (prix ~= 'reduit'))
agenceautorhm=# ORDER BY mu DESC LIMIT 15;
```

fabriquant	nom	modele	prix	kilometrage	annee	mu
CITROEN	Xsara Break 1.4i	XSARA	470	100000	1999	1
PEUGEOT	206 1.4i	206	4000	123358	2001	1
PEUGEOT	406 Coupé 2.0i	406 COUPE	4000	168000	2000	1
RENAULT	Grand Espace 2.0i 16V	GRAND ESPACE	4500	251000	1999	1
RENAULT	Twingo 1.2 16v	TWINGO	4500	90000	2005	1
BMW	X5 3.0i	X5	17400	166000	2004	0.999856589703141
PEUGEOT	807 2.2 16V	807	10490	92058	2005	0.999489066070807
PEUGEOT	406 Coupé 2.0i	406 COUPE	6200	132000	1999	0.995434378457725
AUDI	TT 1.8T	TT	10900	149500	2000	0.986556420806615
OPEL	Astra GTC 1.6 Twinport	ASTRA	10000	80000	2005	0.986301369863014
JAGUAR	X-Type 2.0i V6	X-TYPE	10000	148000	2004	0.984167194426852
AUDI	A4 Avant 2.4 V6	A4 AVANT	8800	145000	2002	0.978745630599863
RENAULT	Scenic 1.6 16V Euro 4	SCENIC II	10500	76500	2006	0.974212034383954
PEUGEOT	206 2.0e 16V	206	5500	72000	2001	0.966666666666667

Figure 13 : requête n°4

III.4. Présentation et interprétation des résultats

Les résultats ainsi présentés dans les figures 9,10, 12, 13. Une implémentation semi-intégrée permet non seulement de manipuler plus facilement des relations floues depuis les applications puisqu'elles sont directement retournées par le système de gestion de

base de données relationnelle, offre également un gain non négligeable en termes de temps d'exécution.

Les résultats obtenus sont exposés dans les figures 9, 10, 12 et 13 ; Cette implémentation permet non seulement de manipuler plus facilement des relations floues depuis les applications puisqu'elles sont retournées par le SGBR comme le montre les différentes figures.

Pour chaque requête, le gain obtenu permet à l'agence de maximiser ses recettes. Puisque le client obtient satisfaction des informations attendues dans l'objectif de prendre une décision de prendre ou de louer un véhicule. Une requête SQLf intègre des prédicats graduels de la clause WHERE. La satisfaction d'un tuple vis-à-vis d'une requête floue étant cette fois-ci graduelle (0,1) et non plus binaire. Les requêtes sont complétées par un seuil qualitatif exprimant le degré minimum de satisfaction acceptable pour qu'un tuple fasse partie de la relation.

Conclusion et perspective

Il est évident que la technologie est appliquée pour atteindre des domaines très variés, qui supportent des données vagues, imprécises ; parfois les relations usuelles SQL ne peuvent pas complètement satisfaire ou modéliser les relations et les attributs de ces domaines. L'interrogation flexible constitue alors une alternative à l'interrogation booléenne pour certains types d'applications.

Nous avons tenté, dans ce mémoire, d'étudier ce type d'interrogation dans les bases de données. Nous avons commencé par faire un tour des publications faites sur le sujet, il a été constaté la flexibilité dans l'interrogation des bases de données est utilisée dans les systèmes relationnels. Le développement de la théorie des sous-ensembles flous a constitué un cadre général pour supporter cette flexibilité, notamment dans les SGBD relationnels. Nous avons ensuite présenté l'outil PostgreSqlf (avec le langage Sqlf) qui nous a permis d'avoir les résultats escomptés. A travers un exemple de 4 requêtes floues nous avons démontré l'utilité de cette technique dans la satisfaction des clients.

Nous comptons développer dans les jours qui viennent une interface graphique qui permettra aux utilisateurs de construire des requêtes floues sans disposer des connaissances sur les langages d'interrogation formels de base de données ni sur la logique floue.

Références

- ANNE, L. (2002). *Bases de données multidimensionnelles et leur utilisation pour la fouille de données*. Paris.
- BARTEL, S. (2006). *Interrogation floue de bases de données: extension iSQLf*. Lannion.
- BEN HASSINE, M. (2005). *contribution à l'implémentation d'une base de donnée floue sous un système de gestion d'une base de donnée relationnelle*. Alger.
- BEN HASSINE, M. A. (2005). *contribution à l'implémentation d'une base de donnée floue sous un système de gestion de base de donnée relationnel*. Tunis.
- BEN HASSINE, M. A. (2005). *Contribution à l'implémentation d'une base de données floue sous un système de gestion de base de données relationnel*. Tunis.
- BENJAMIN, F., XAVIER, N., & FLORENT, S. (2001). *La logique floue*.
- BEZDEK, J. (1993). *Fuzzy models- what are they, and why?*
- BOSC, P., KRAFT, D., & PETRY, F. (2005). *Ensembles flous dans les bases de données et les systèmes d'information*.
- BUCKLES, B., & PETRY, F. (1982). *Une représentation floue des données pour les bases de données relationnelles*.
- DERNONCOURT, F. (2011). *introduction à la logique floue appliquée aux systèmes décisionnels*. Paris.
- DERNONCOURT, F. (2012). *introduction à la logique floue*. OpenClassroom.
- DUBOIS, D., & PRADE, H. (1988). *Possibility Theory*. New York.
- GACOGNE, L. (2003). *logique floue et applications*. Evry.
- GALINDO, J. (1999). *Traitement de l'imprécisions dans les bases de données relationnelles*.
- HACHANI, N. (2001). *Conception et implémentation d'une approche de relaxation d'une requête SQL*. Tunis.
- LAMBALGEN. (2000). *Fuzzy logic*. Amesterdam.
- PONS, O. (1995). *Representacion Logica de Bases de Datos Difusas Fundamentos Teoricos y Implentacion*.
- PRADE, H., & TESTEMALE, C. (1987). *Fuzzy Relational Databases: Representational issues and Reduction Using Similarity Measures*.
- RADECKI, T. (1975). *Modèle mathématique de recherche d'informations basé sur la théorie du flou sets*. Pologne.
- SMITS, G., PIVERT, O., & GIRAULT, T. (2006). *PostgreSQL : un système d'interrogation floue*.

SMITS, G., PIVERT, O., & GIRAULT, T. (2010). PostgreSQLf, un système d'interrogation floue. *Université Rennes 1, France, 1.*

TRE, G., & SLAWOMIR, Z. (2015). *The essence of fuzzy databases.* Springer.

UMANO, M., FUKAMI, S., MIZUMOTO, M., & TANAKA, K. (1980). *Retrieval Processing from fuzzy Databases.* Tokyo.

Zadeh, L. (1965). *Fuzzy sets. Information and control,*

ZADEH, L. (1965). *Information and control "fuzzy sets".*

ZADEH, L. (1975). *Logique floue et raisonnement approximatif.*

Zadeh, L. (1978). *ensembles flous comme base d'une théorie de la possibilité.*

ZEMANKOVA LEECH, M., & KANDELI, A. (1985). *Implementing Imprecision in Information Systems.*