PLAN :

I. Introduction

II. Notion Analyse et Conception

III. Concepts Analyse et Conception

   A. MCD

   B. MLD

I.V SQL

- ✓ Langage de définition des données (LDD) :
- ✓ Langage de manipulation de données (LMD) :
- ✓ Langage d'interrogation de données (LID) : SELECT ;

I. Introduction

Il est difficile de donner une définition exacte de la notion de base de données. Une définition très générale pourrait être : Un ensemble organisé d'informations avec un objectif commun.

Peu importe le support utilisé pour rassembler et stocker les données (papier, fichiers, etc.), dès lors que des données sont rassemblées et stockées d'une manière organisée dans un but spécifique, on parle de base de données.

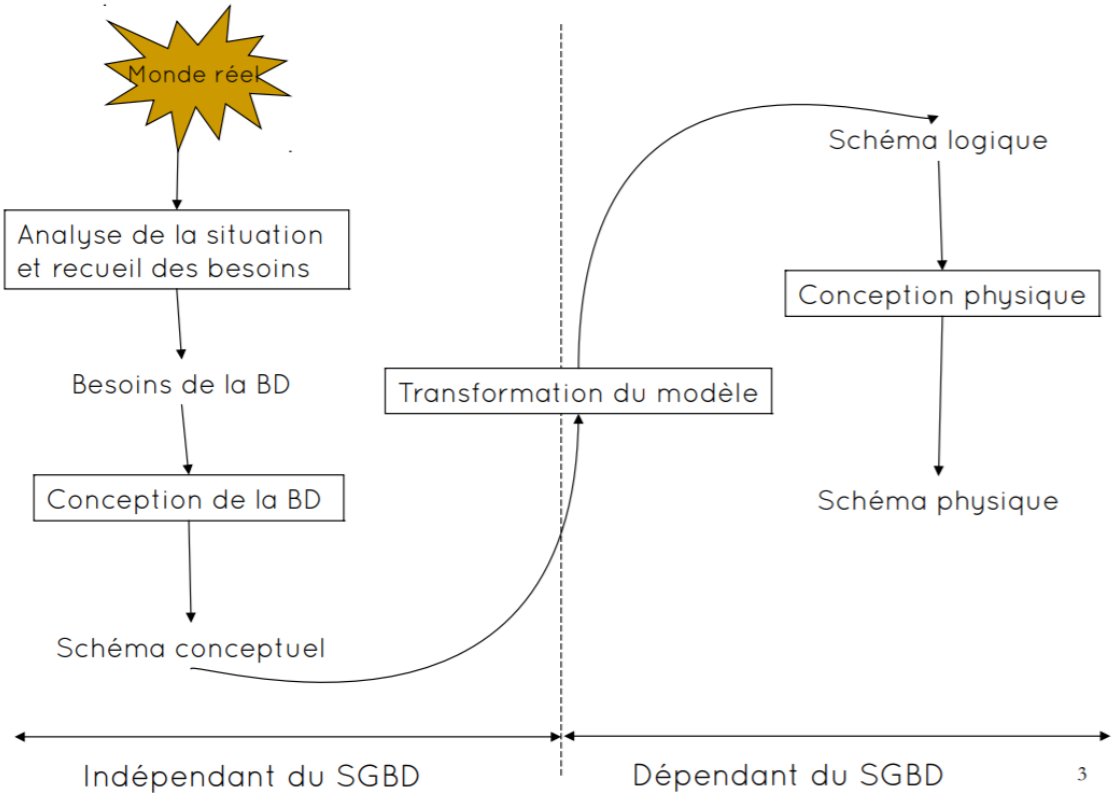
Plus précisément, on appelle base de données un ensemble structuré et organisé permettant le stockage de grandes quantités d'informations afin d'en faciliter l'exploitation (ajout, mise à jour, recherche de données). Bien entendu, dans le cadre de ce cours, nous nous intéressons aux bases de données informatisées.

II. Notion Analyse et Conception

On distingue quatre étapes dans la conception d'une base de données :

- Analyse de la situation existante et des besoins
- Création d'une série de modèles conceptuels pour représenter tous les aspects importants du problème
- Traduction des modèles conceptuels en modèle logique
- Implémentation d'une base de données dans un SGBD à partir du modèle logique

Conception d'une base de données



1) **Merise**

MERISE est une méthode française née dans les années 70, développée initialement par **Hubert Tardieu**. Elle fut ensuite mise en avant dans les années 80, à la demande du Ministère de l'Industrie qui souhaitait une méthode de conception des SI.

MERISE est donc une méthode d'analyse et de conception des SI basée sur le principe de la séparation des données et des traitements. Elle possède un certain nombre de **modèles** (ou **schémas**) qui sont répartis sur 3 niveaux :

- Le niveau **conceptuel**,
- Le niveau **logique** ou **organisationnel**,
- Le niveau **physique**.

2) **Uml :** Langage de Modélisation

UML est une notation graphique. UML intègre l'objet et est donc plus adaptée aux SGBDOO UML est adaptée à la programmation orientée-objet Le diagramme de classe UML repose sur le modèle entité-association. Quelques outils : – Modelio, ArgoUML, Papyrus (gratuit, UML) : génération code Java – Together (gratuit, commercial, UML) : génération code C++ et Java – Rational Rose (commercial, UML) : tables, C++, Java, …

3) **Etude comparative**

Les « méthodologues » disent qu'une méthode, pour être opérationnelle, doit avoir 3 composantes :

Une démarche (les étapes, phases et tâches de mise en oeuvre) ;

Des formalismes (les modélisations et les techniques de transformations) ;

Une organisation et des moyens de mise en oeuvre.

Merise s'est attachée, en son temps, à proposer un ensemble « cohérent » sur ces trois composantes. Certaines ont vieilli et ont dû être réactualisées (la démarche), d'autre « tienne encore le chemin » (la modélisation).

UML se positionne exclusivement comme un ensemble de formalismes. Il faut y associer une démarche et une organisation pour constituer une méthode.

Merise se positionne comme une méthode de conception de système d'information organisationnel, plus tournée vers la compréhension et la formalisation des besoins du métier que vers la réalisation de logiciel. En

sens, Merise se réclame plus de l'ingénierie du système d'information que du génie logiciel. Jamais Merise ne s'est voulu une méthode de développement de logiciel ni de programmation.

UML, de par son origine (la programmation objet) s'affirme comme un ensemble de formalismes pour la conception de logiciel à base de langage objet.

Merise est encore tout à fait valable pour :

La modélisation des données en vue de la construction d'une base de données relationnelles, la modélisation des processus métiers d'un système d'information automatisé en partie par du logiciel,

La formalisation des besoins utilisateur dans la cadre de cahier des charges utilisateur, en vue de la conception d'un logiciel adapté.

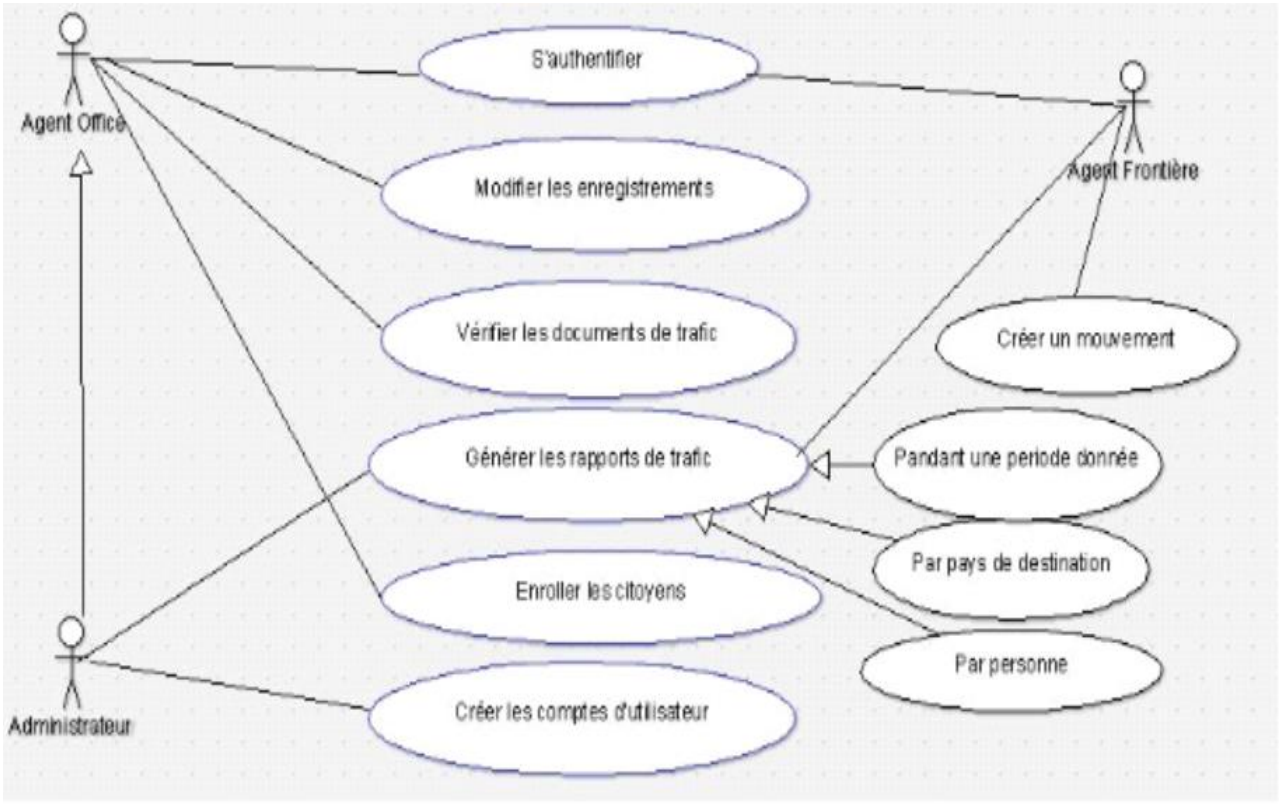
UML estidéal pour:

Concevoir et déployer une architecture logiciel développée dans un langage orienté objet (Java, C++, VB.Net,...).

Pour modéliser les données (le modèle de classe réduit sans méthodes et stéréotypé en entités), mais avec des lacunes que ne présentait pas l'entité relation de Merise.

Pour modéliser le fonctionnement métier (le diagramme d'activité et de cas d'utilisation) qui sont des formalismes très anciens qu'avait, en son temps, amélioré Merise...

Après cette étude comparative, il est certes que nous allons adopter UML comme langage de modélisation puisque nous allons utiliser le concept de l'orienté objet ainsi en VB.Net comme langage, pour développer l'application de synthèse et de reconnaissance biométrique de documents de trafic.



III. Concepts Analyse et Conception

Élaboration du concept

L'ingénieur doit élaborer plusieurs concepts pour parvenir à une solution optimale répondant aux besoins du client. Parmi les tâches accomplies par l'ingénieur, cette étape est sans doute celle qui sollicite le plus sa créativité.

Les séances de remue-méninges (*brainstorming*) favorisent le processus en mettant en commun la créativité de plusieurs personnes. Il est parfois indiqué d'inviter à ces séances des personnes ayant peu de connaissances dans le domaine d'activité couvert par le mandat afin d'amener une nouvelle vision ou une approche différente. Par sa connaissance approfondie du domaine, le client apporte souvent une contribution importante.

Les objectifs des mandats de conception qu'ont à atteindre les ingénieurs sont si nombreux et de nature si différente qu'il est impossible de décrire ici les façons d'y parvenir.

Les efforts fournis par l'ingénieur à cette étape conduisent à la solution optimale avant même de commencer les calculs ou autres tâches plus complexes demandant temps et énergie. Le client se trouve ainsi à être mieux servi, à moindre coût.

À cette étape, l'ingénieur doit exécuter les principales actions suivantes :

- la revue technologique et des règles de l'art;
- l'élaboration des concepts;
- l'établissement des démarches préparatoires.

Revue technologique et des règles de l'art

La revue technologique permet à l'ingénieur :

- de mettre à jour l'information technique ainsi que les normes et règles de l'art;
- de réviser l'état des technologies disponibles ou existantes.

L'ingénieur complète sa revue technologique par une recherche bibliographique permettant de localiser les sources de données qu'il croit pouvoir utiliser pour la conception.

L'ingénieur doit particulièrement s'assurer de la fiabilité des sources. La littérature en format papier — revues ou livres — demeure une source fiable.

Par contre, l'ingénieur doit demeurer vigilant en ce qui concerne les données et les renseignements trouvés sur Internet ou obtenus par l'intermédiaire de ses contacts professionnels. Il est recommandé de valider ces données avec celles obtenues d'autres sources. Si ce n'est pas possible, l'ingénieur doit évaluer la crédibilité de la source elle-même en communiquant avec son responsable et en posant des questions.

Élaboration des concepts

L'ingénieur fait une description détaillée des différents concepts et solutions envisagés, en exposant leurs avantages et leurs inconvénients respectifs. Les concepts peuvent parfois être représentés sommairement, sous forme graphique (schéma ou dessin).

Dès lors, l'ingénieur peut amorcer une analyse de ces concepts, notamment en effectuant des calculs sommaires afin d'établir un plan de mise en œuvre et d'en estimer les coûts et les échéances.

L'ingénieur choisit les concepts les plus avantageux et les documente dans un rapport d'ingénierie conceptuelle qu'il remet au client. Ce rapport contient les principaux éléments suivants :

- les objectifs, les données de base, les besoins et les contraintes du client;
- les exigences fixées par les différents codes, normes et règlements;
- l'analyse des concepts, appuyée par des calculs sommaires et des dessins préliminaires;
- les coûts et les échéances;
- le plan de mise en œuvre du projet;
- les plans et les schémas;
- une grille de comparaison des concepts indiquant l'option privilégiée par l'ingénieur concepteur ainsi que les raisons de son choix.

Établissement des démarches préparatoires

Afin de réduire les délais parfois importants liés à la préparation de la réalisation de l'ouvrage ou du processus, l'ingénieur établit une liste des démarches préparatoires. Ces démarches

correspondent aux différentes modalités par lesquelles l'ingénieur s'assure de la disponibilité :

- des espaces (en fonction des contraintes de production, des échéances, des prévisions pour l'avenir, etc.);
- des services (énergie, téléphone, câble, eau, air, vapeur, etc.);
- des terrains qui seront requis au cours de la réalisation.

L'ingénieur tient aussi compte des différentes demandes :

- de permis;
- d'accès à des services;
- d'acquisition de terrains ou autres, incluant la vérification de la conformité de l'ouvrage prévu aux lois relatives à l'aménagement ou à l'environnement (urbanisme, environnement, zonage agricole et réglementations municipales).

L'ingénieur fournit généralement au client la liste des demandes préparatoires, à laquelle il joint une estimation des délais prévus. À la demande du client, l'ingénieur peut être appelé à préparer et à faire le suivi et la coordination de ces demandes et démarches.

Analyse des besoins

La première étape de la conception consiste à analyser la situation pour tenir compte des contraintes, des risques et de tout autre élément pertinent et assurer un ouvrage ou un processus répondant aux besoins du client.

À cette étape, l'ingénieur doit accomplir les principales actions suivantes :

- connaître le contexte;
- déterminer les besoins et les contraintes;
- déterminer les paramètres de conception;
- préparer le cahier des charges.

Connaître le contexte

Cette activité permet à l'ingénieur de bien comprendre le contexte du moment et d'obtenir de l'information supplémentaire permettant d'en élargir les horizons ou de tenir compte des considérations futures.

La source principale d'information étant généralement le client lui-même ou l'employeur, il ne faut pas hésiter à poser des questions à cette personne pour obtenir des renseignements

utiles. Ceux-ci sont parfois si évidents pour le client qu'il n'a tout simplement pas pensé à les mentionner.

En parallèle, l'ingénieur cherche de l'information supplémentaire en se documentant sur le sujet. Il visite généralement le site de l'ouvrage et l'équipe de conception. Au cours de cette visite, l'ingénieur peut effectuer des mesurages, prendre des photos et réaliser différents tests. Plusieurs visites pourraient être nécessaires.

Déterminer les besoins et les contraintes

Une fois mis en perspective avec le contexte, les besoins réels du client permettent de redéfinir ou de valider ceux que ce dernier a proposés. L'ingénieur devrait structurer les besoins par degré d'importance ou par thèmes, afin de mieux cerner à quelle étape de la conception ces besoins doivent être pris en compte.

L'ingénieur, en collaboration avec le client, doit également établir les contraintes susceptibles de nuire à l'atteinte des objectifs par l'ouvrage ou le processus.

L'ingénieur ne doit pas se limiter aux contraintes physiques, techniques et économiques. Il doit aussi élargir sa vision et tenir compte des contraintes environnementales, humaines, sociales, légales et de tout autre élément pertinent.

Déterminer les paramètres de conception

Pour répondre aux besoins, l'ingénieur doit déterminer les paramètres de conception à partir des mesures prises sur le site de l'ouvrage ou du processus, de l'information technique disponible et de son expérience. L'ingénieur doit être réaliste quant au nombre de paramètres.

Il est parfois possible de regrouper les paramètres de conception par types ou par besoins.

L'ingénieur devrait également considérer le développement durable comme un critère de conception en soi, tant au stade de l'ingénierie préliminaire qu'à celui de l'ingénierie détaillée. Pour en savoir plus, consulter la section Développement durable.

Préparer le cahier des charges (ou plan de travail)

Les étapes précédentes permettent à l'ingénieur de proposer au client un cahier des charges réaliste et convenant à ses besoins, dans lequel il décrit de façon claire et précise les tâches à effectuer et une évaluation du temps pour les accomplir.

Les limites du mandat et des besoins inscrits dans le cahier des charges doivent être clairement indiquées et éviter toute ambiguïté.

De plus, cet exercice permet à l'ingénieur d'évaluer adéquatement l'envergure du mandat de conception et de vérifier s'il a toutes les compétences pour le réaliser ou s'il devra recourir à des ressources externes, en accord avec son client.

A. MCD

Le MCD (Modèle Conceptuel des Données) est utilisé par les concepteurs et les analystes pour décrire sous forme d'un schéma les données relatives au sujet à traiter (en gros les entités, leurs attributs et les relations qu'elles entretiennent). Le MCD ne tient pas compte du SGBD et du langage de programmation à suivre. Le formalisme utilisé pour le MCD est convenu à l'avance. En France, on utilise essentiellement celui qui a été

défini pour Merise fin des années soixante-dix, augmenté de fonctionnalités supplémentaires dans les années quatre-vingt (par exemple l'héritage). Il suffit de connaître la norme pour lire sans ambiguïté un MCD.

- **Les entités**

  Une entité est la représentation d'un élément matériel ou immatériel ayant un rôle dans le système que l'on désire décrire.

  Une entité est une *instanciation* de la classe. Chaque entité est composée de propriétés, données élémentaires permettant de la décrire.

- Dans un MCD, les *attributs* sont des éléments d'information de base attachés à une entité, une association ou un héritage. Dans un MLD, il n'y a pas d'informations, les attributs existent alors dans les entités sans origine conceptuelle.

- Les Occurrences

  Une **occurrence**, c'est tout simplement une **« ligne » de valeurs**. Dans une entité, une occurrence correspond à l'ensemble des valeurs des propriétés rattachées à un seul identifiant. Dans une relation, une occurrence correspond à l'ensemble des valeurs des propriétés de la relation (représenté par les clés de chaque entité liée) : on l'appelle alors une **occurrence de relation**.

- **Les Cardinalités**

  Lorsque l'on conçoit une base de données avec le MCD de Merise, on obtient un schéma avec des entités et des associations. Pour préciser au mieux les associations, on utilise des cardinalités. Les cardinalités sont des caractères (0,1, n) qui fonctionnent par couple et qui sont présents de chaque côté d'une association (sur chaque « patte »). Ils donnent des indications très intéressantes et permettent par la suite de construire la base de données :

  -avec la création de clés étrangères dans le cas d'une CIF

  -avec la création d'une table intermédiaire dans le cas d'une CIM

Les cardinalités possibles sont :

- 0,1 : au minimum 0, au maximum 1 seule valeur (CIF) ;
- 1,1 : au minimum 1, au maximum 1 seule valeur (CIF) ;
- 0,n : au minimum 0, au maximum plusieurs valeurs ;
- 1,n : au minimum 1, au maximum plusieurs valeurs.

Les cardinalités maximales (le côté droit ou le dernier caractère) définissent si la relation est une CIM ou une CIF.

Dans l'image, on a « 1,1 » en cardinalité du côté gauche de l'association et une cardinalité « 0,n » du côté droit. Concrètement, il faut lire les cardinalités ainsi :

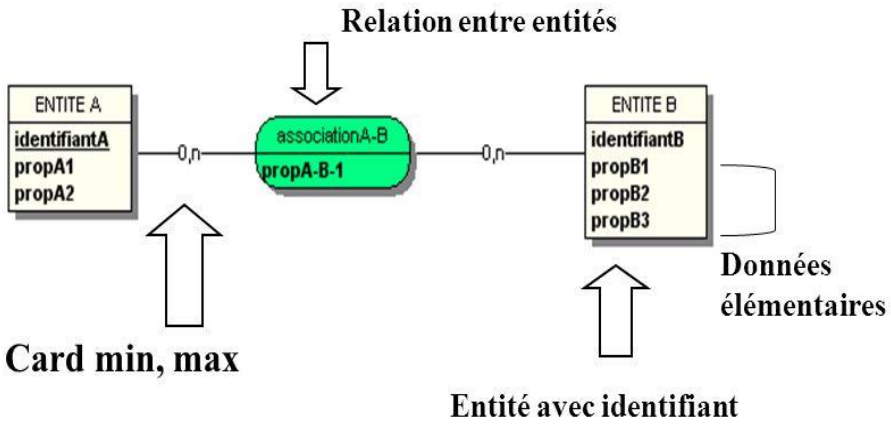
- Une structure se situe dans 1 et 1 seule ville ;
- Dans une ville peut se situer 0 ou plusieurs (n) structures.

Les cardinalités sont importantes car elles traduisent les règles de gestion, elles doivent être validées avec l'utilisateur final.

- **Formalisme et Représentation**

3- éléments d'un MCD

- Récapitulatif du formalisme:



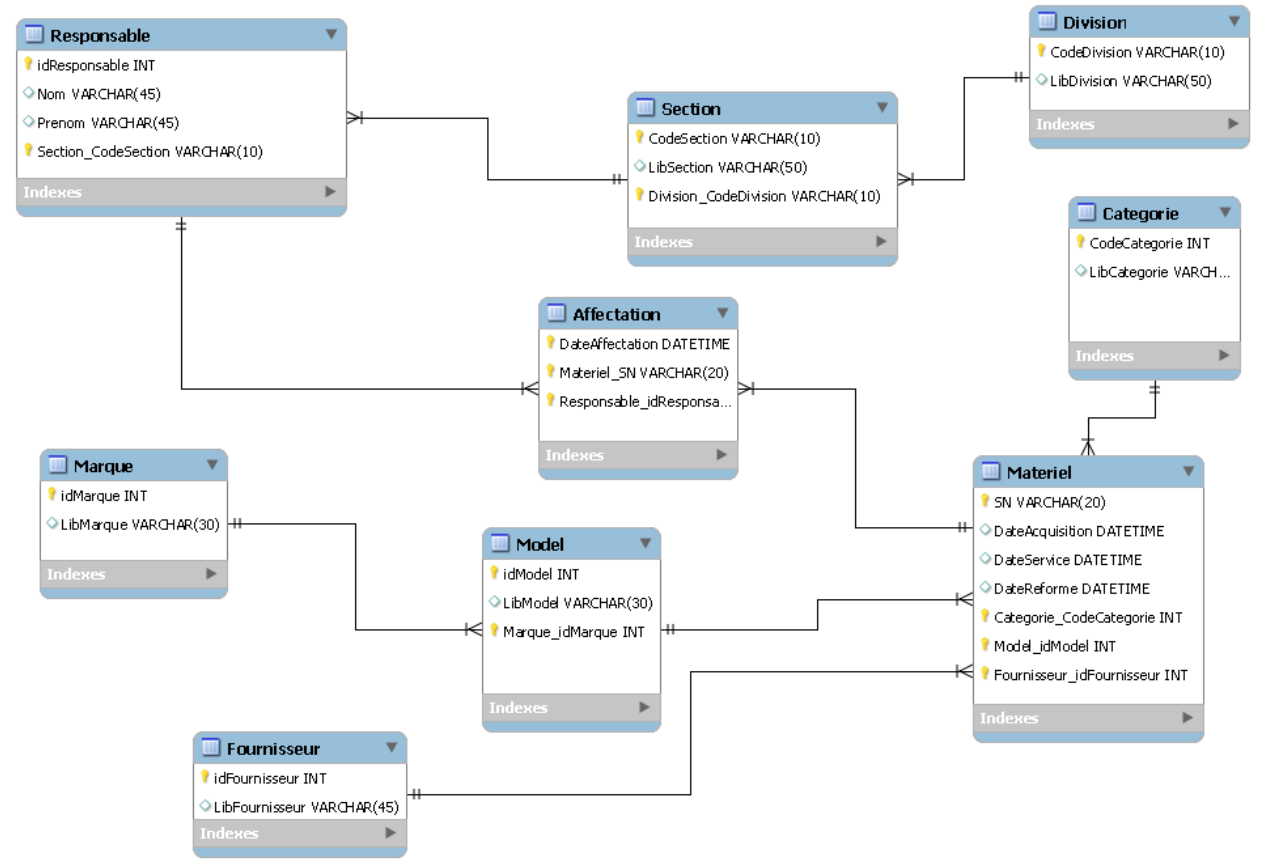
B. MLD

L’étape MLD (Modèle Logique de Données) se situe chronologiquement juste après l’étape MCD et revient à présenter les objets du MCD sous une forme compréhensible par un SGBD. Pour faire court, dans un contexte SGBD relationnel, les objets représentés sont désormais des tables (disons SQL) et les liens qui les unissent.

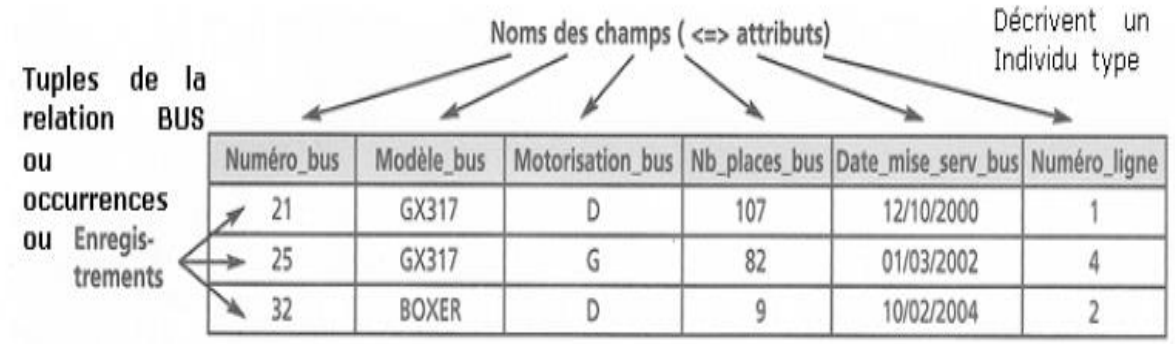
- MLDR

Le Modèle Logique de Données Relationnelles (MLDR)

C'est grâce à toutes les opérations précédentes que l'ensemble des tables de la base de données vont pouvoir être structurées de manière simple et rapide. Ce sont les entités, les cardinalités maximales et minimales qui vont jouer le rôle essentiel pendant la réalisation du MLDR. Le modèle logique de données relationnel s'obtient soit à partir du modèle conceptuel de données ou soit à partir du modèle organisationnel de données.


Responsable
idResponsable INT
Nom VARCHAR(45)
Prenom VARCHAR(45)
Section_CodeSection VARCHAR(10)
Indexes
Section
CodeSection VARCHAR(10)
LibSection VARCHAR(50)
Division_CodeDivision VARCHAR(10)
Indexes
Division
CodeDivision VARCHAR(10)
LibDivision VARCHAR(50)
Indexes
Categorie
CodeCategorie INT
LibCategorie VARCH...
Indexes
Affectation
DateAffectation DATETIME
Materiel_SN VARCHAR(20)
Responsable_idResponsa...
Indexes
Marque
idMarque INT
LibMarque VARCHAR(30)
Indexes
Model
idModel INT
LibModel VARCHAR(30)
Marque_idMarque INT
Indexes
Materiel
SN VARCHAR(20)
DateAcquisition DATETIME
DateService DATETIME
DateReforme DATETIME
Categorie_CodeCategorie INT
Model_idModel INT
Fournisseur_idFournisseur INT
Indexes
Fournisseur
idFournisseur INT
LibFournisseur VARCHAR(45)
Indexes

- **Model Relationnel**

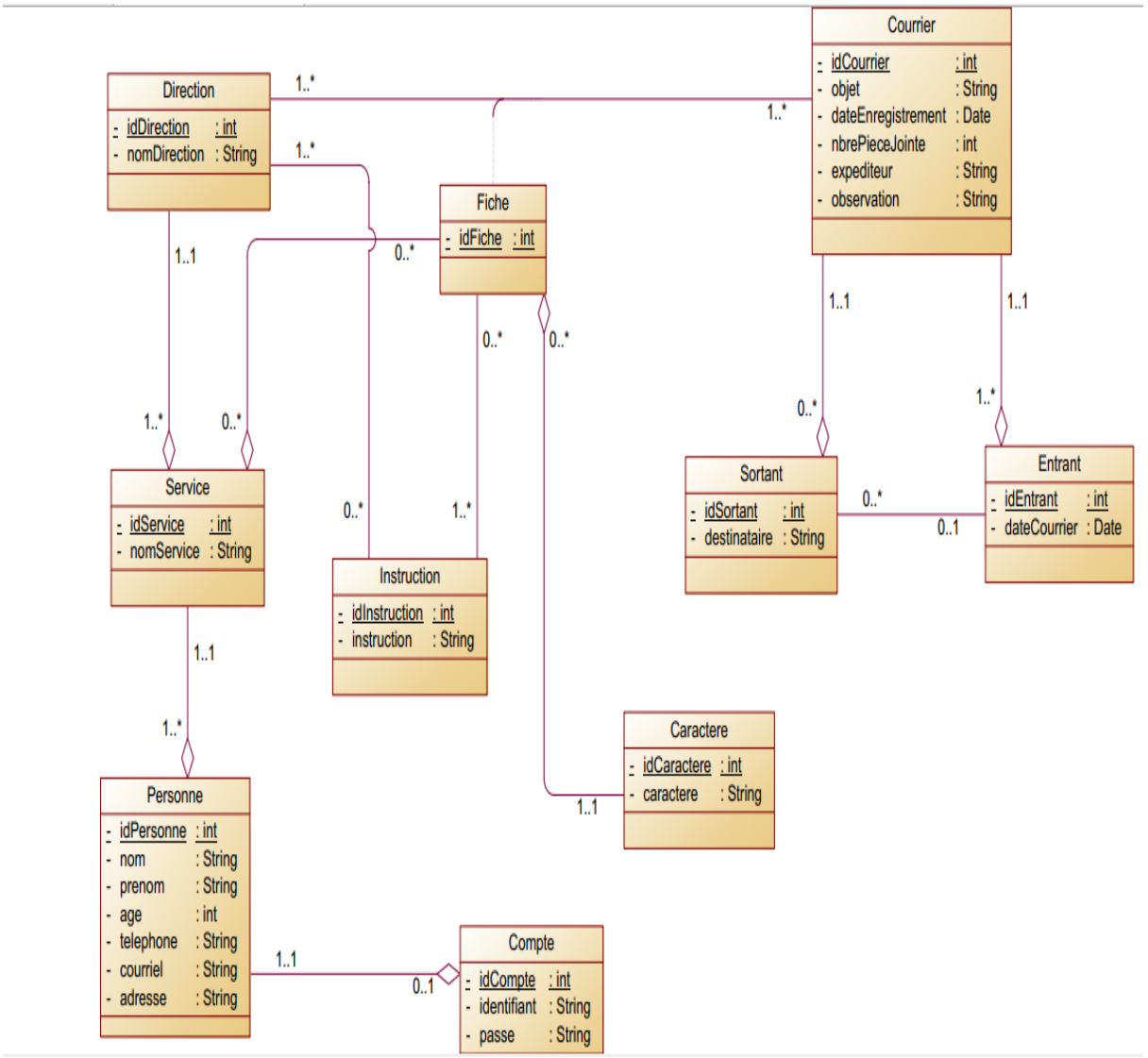


Numéro_bus	Modèle_bus	Motorisation_bus	Nb_places_bus	Date_mise_serv_bus	Numéro_ligne
21	GX317	D	107	12/10/2000	1
25	GX317	G	82	01/03/2002	4
32	BOXER	D	9	10/02/2004	2

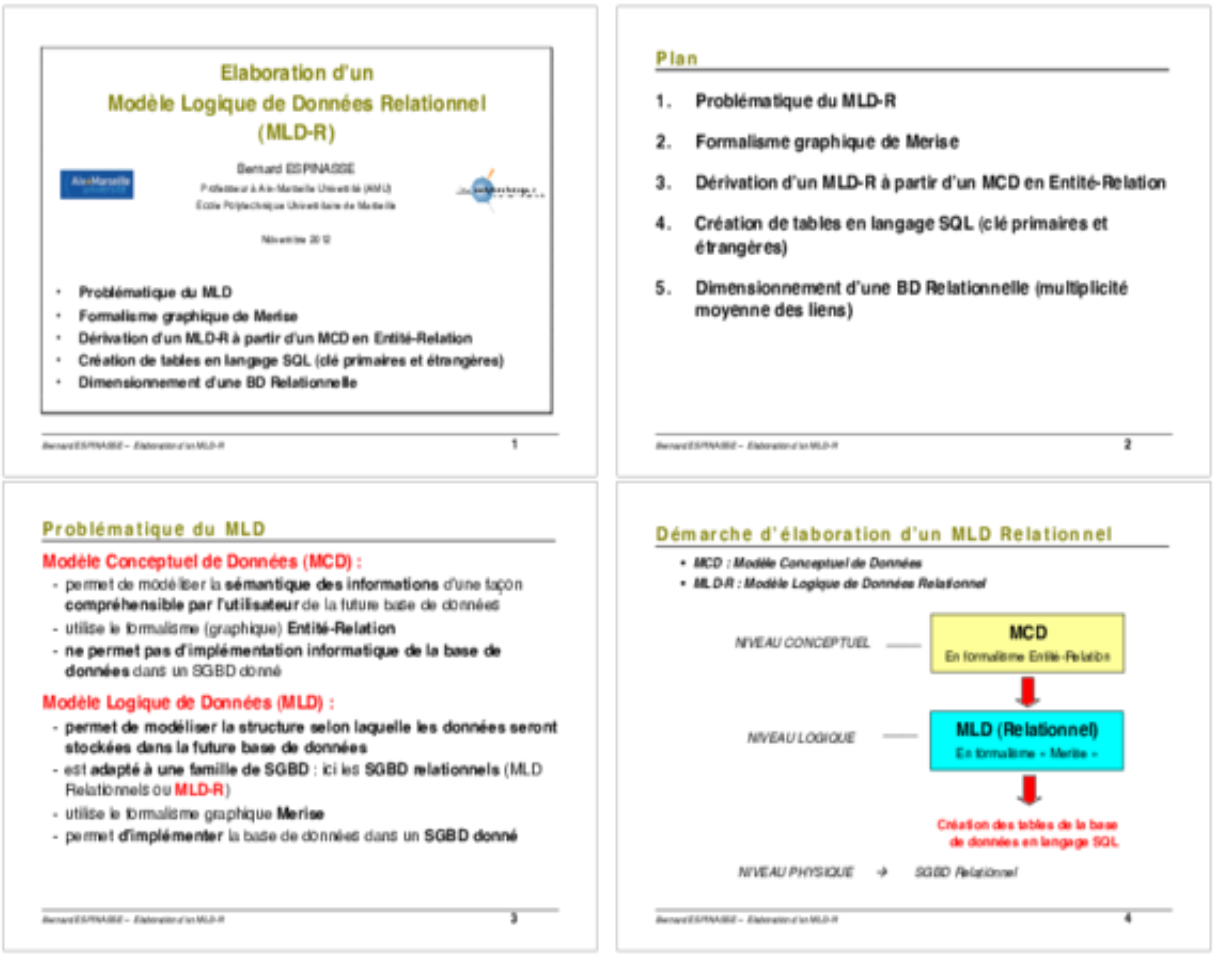
- **Régles de passage du MCD au MLDR**

- Il y a 3 règles à appliquer pour transformer les entités.
- **Règle E1**: une entité devient une relation, dont les attributs sont les propriétés de l'entité, et la clef primaire l'identifiant de l'entité
- **Règle E2**: une sous-entité devient une relation dont la clef primaire et étrangère est celle de l'entité dont elle dépend, et les attributs sont les propriétés de l'entité
- **Règle E3**: une entité faible devient une relation dont les attributs sont les propriétés de l'entité, et la clef primaire composite associe l'identifiant de l'entité faible et celui de l'entité dont elle

Formalisme de Représentation



Schema Récapitulatif



Elaboration d'un
Modèle Logique de Données Relationnel
(MLD-R)

Bernard ESPINASSE
Professeur à Aix-Marseille Université (AMU)
Ecole Polytechnique Universitaire de Marseille

Novembre 2012

- Problématique du MLD
- Formalisme graphique de Merise
- Dérivation d'un MLD-R à partir d'un MCD en Entité-Relation
- Création de tables en langage SQL (clé primaires et étrangères)
- Dimensionnement d'une BD Relationnelle

Bernard ESPINASSE – Elaboration d'un MLD-R 1

Plan

1. Problématique du MLD-R
2. Formalisme graphique de Merise
3. Dérivation d'un MLD-R à partir d'un MCD en Entité-Relation
4. Création de tables en langage SQL (clé primaires et étrangères)
5. Dimensionnement d'une BD Relationnelle (multiplicité moyenne des liens)

Bernard ESPINASSE – Elaboration d'un MLD-R 2

Problématique du MLD

Modèle Conceptuel de Données (MCD) :
- permet de modéliser la **sémantique des informations** d'une façon **compréhensible par l'utilisateur** de la future base de données
- utilise le formalisme (graphique) **Entité-Relation**
- **ne permet pas d'implémentation informatique de la base de données** dans un SGBD donné

Modèle Logique de Données (MLD) :
- **permet de modéliser la structure selon laquelle les données seront stockées dans la future base de données**
- est **adapté à une famille de SGBD** : ici les **SGBD relationnels** (MLD Relationnels ou **MLD-R**)
- utilise le formalisme graphique **Merise**
- permet **d'implémenter** la base de données dans un **SGBD donné**

Bernard ESPINASSE – Elaboration d'un MLD-R 3

Démarche d'élaboration d'un MLD Relationnel

- *MCD : Modèle Conceptuel de Données*
- *MLD-R : Modèle Logique de Données Relationnel*

Bernard ESPINASSE – Elaboration d'un MLD-R 4

IV. SQL

SQL est l'abréviation de Structured Query Language ou langage de requêtes structuré en français.

Le SQL est le langage principal utilisé pour accéder aux bases de données et les manipuler. Nous allons utiliser ce langage pour exécuter toutes sortes de requêtes dans une base de données : récupérer des données, les mettre à jour, en insérer de nouvelles ou même créer de nouvelles bases de données.

Le SQL est un langage à part entière : il possède sa propre syntaxe que nous allons découvrir dans les chapitres suivants.

- ✓ Langage de définition des données (LDD) :

Un **langage de définition de données** (**LDD** ; en anglais *data definition language*, DDL) est un langage de programmation et un sous-ensemble de SQL pour manipuler les structures de données d'une base de données, et non les données elles-mêmes.

Il permet de définir le domaine des données, c'est-à-dire l'ensemble des valeurs que peut prendre une donnée : nombre, chaîne de caractères, date, booléen. Il permet aussi de regrouper les données ayant un lien conceptuel au sein d'une même entité. Il permet également de définir les liens entre plusieurs entités de nature différente. Il permet enfin d'ajouter des contraintes de valeur sur les données.

On distingue typiquement quatre types de commandes SQL de définition de données :

- `CREATE` : création d'une structure de données ;
- `ALTER` : modification d'une structure de données ;
- `DROP` : suppression d'une structure de données ;
- `RENAME` : renommage d'une structure de données.

Exemples

- *Création* d'une structure de données :

```
1 CREATE VIEW bts
2 AS SELECT nom, prenom, age
3 FROM eleves
4 WHERE classe = 'BTS';
```

- *Modification* d'une structure de données :

```
ALTER TABLE eleves ADD COLUMN moyenne_annuelle INTEGER NULL;
```

- *Suppression* d'une structure de données :

```
DROP VIEW bts;
```

- *Renommage* d'une structure de données :

```
RENAME VIEW bts TO eleves_bts;
```

✓ Langage de manipulation de données (LMD) :

Un **langage de manipulation de données** (**LMD** ; en anglais *data manipulation language*, DML) est un langage de programmation et un sous-ensemble de SQL pour manipuler les données d'une base de données.

Ces commandes de manipulation de données doivent être validées à l'issue d'une transaction pour être prises en compte.

On distingue typiquement quatre types de commandes SQL de manipulation de données :

- `SELECT` : sélection de données dans une table ;
- `INSERT` : insertion de données dans une table ;
- `DELETE` : suppression de données d'une table ;
- `UPDATE` : mise à jour de données d'une table.

Exemples

- *Sélection* de données dans une table :

```
SELECT nom, prenom, classe FROM eleves;
```

- *Insertion* de données dans une table :

```
INSERT INTO eleves (nom, prenom)
VALUES ('Dupont', 'Matthieu');
```

- *Suppression* de données dans une table :

```
DELETE FROM eleves
WHERE prenom = 'Paul' and nom = 'Durand';
```

- *Mise à jour* de données dans une table :

```
UPDATE eleves
SET prenom = 'Henry'
WHERE nom = 'Leroy';
```

✓ Langage d'interrogation de données (LID) : SELECT ;

SQL ou le langage d'interrogation de données (*LID*)

Une base de données peut être interrogée de manière formelle par le langage SQL ou par un langage algébrique. Le langage SQL (Structured Query Langage) est une évolution de SEQUEL développé en 1976 par IBM comme un langage de recherche. SQL est devenu un standard des bases de données relationnelles (en 1987, normalisation de ce langage par ANSI). SQL s'utilise sous deux formes, soit d'une manière interactive, soit à l'intérieur d'un langage hôte (C, fortran, cobol…). SQL ne comporte qu'une vingtaine d'instructions, il est dit procédural (l'accès aux données se fait par leur contenu et non par leur chemin). SQL est un langage de définition des données (LDD), de manipulation de données (LMD), de contrôle des données (LCD) et d'interrogation des données (LID).Malgré le succès du langage SQL qui a suivi, Edgar F. Codd dénoncera cet outil qu'il considère comme une interprétation incorrecte de ses théories.

Le langage algébrique (LA) correspond à un pseudo algorithme du SQL. Il est composé de peu d'opérateurs (sélection, projection et jointure étant les opérateurs de base).

La jointure

C'est une opération permettant de ramener sur une même ligne des données venant de plusieurs tables. Une jointure s'effectue grâce à un produit cartésien de plusieurs tables (256 au maximum) et l'application de sélections.

```
R3 = JOIN(R1, R2; R1.propriété opérateur R2.propriété)
```

La projection

La projection est une opération qui consiste à ne sélectionner que certaines données pour l'affichage. La syntaxe est la suivante :

```
R1 ← PROJ(nomdelatable; liste des propriétés)
SELECT liste des propriétés
FROM nomdelatable ;
```

La sélection

La sélection consiste à sélectionner des lignes répondant à certains critères. La syntaxe est la suivante :

```
R1 ← SEL(R; propriété opérateur valeur ET propriété opérateur valeur)
```

Le tri

Pour obtenir un affichage trié, il est nécessaire de le préciser. L'opérateur de tri est à utiliser en langage algébrique. La clause **order by** est à ajouter à la fin de la requête SQL, suivi du nom de la colonne qui doit être utilisé pour ce tri Le tri par défaut s'effectue par ordre croissant ; pour obtenir un tri par ordre décroissant, il faut rajouter le paramètre **DESC**.

```
R2 ← TRI(R1; attribut du tri croissant)
```

. La commande SELECT et la clause FROM

Le **SELECT** est la commande de base du SQL destinée à extraire des données d'une base ou calculer de nouvelles données à partir d'existantes. La syntaxe est la suivante :

SELECT [DISTINCT ou ALL] * ou liste de colonnes
FROM nom de table ou de la vue
[WHERE prédicats]
[GROUP BY ordre des groupes]
[HAVING condition]
[ORDER BY] liste de colonnes

A côté de cette syntaxe, il existe les agrégats suivants :
- **SUM** […] : renvoie à la **somme** d'un champ (valeurs de données de type numérique ou date/heure).
- **AVG** […] : renvoie la **moyenne** d'un champ (valeurs de données de type numérique ou date/heure).
- **MIN** […] : renvoie la **valeur minimale** d'un champ (valeurs de données de type numérique, date et texte).
- **MAX** […] : renvoie la **valeur maximale** d'un champ (valeurs de données de type numérique, date et texte).
- **COUNT** […] : renvoie le **nombre d'enregistrements** de la table.

• Il est possible de surnommer une table dans la clause **FROM**, dans ce cas, la syntaxe de la partie FROM de la commande SELECT est la suivante :
FROM nom_de_table ou nom_de_la_vue surnom .

On observera qu'en fait l'ordre SQL SELECT est composé de six clauses dont quatre sont optionnelles. La plupart du temps, la difficulté réside dans la compréhension de la différence entre le filtre WHERE et le filtre HAVING. Le **filtre WHERE** permet de **filtrer les données des tables** tandis que le **filtre HAVING** permet de **filtrer les données du résultat**.

Voici les clauses de l'ordre SELECT :

SELECT	NomChamp1, NomChamp2	Champs à projeter ou à calculer ou fonction d'agrégat
FROM	TABLE1, TABLE2…	Tables utiles à la requête
WHERE	Expression AND/OR	Jointures et restrictions
GROUP BY	NomChamp	Regroupement de résultat d'opérations d'agrégat
HAVING	Expression	Restriction sur l'affichage des résultats d'opérations d'agrégat
ORDER BY	NomChamp [ASC]/DESC	Critères de tri