

Documentation sur les MCD, MLD et SQL

I. Notion Analyse et Conception

➤ Les systèmes d'information (SI)

🚦 Définition Exemples de SI :

- Logiciel des gestions d'étudiants
- Logiciel des gestions d'étudiants
- Application de gestion d'un restaurant

« Un système d'information est un ensemble des ressources (matériel, logiciel, personnel, données, procédures,...) permettant d'acquérir, traiter, stocker l'information (sous forme de données, textes, sons, images, ...) dans et entre les organisations. » Robert Reix

Mais qui sont ces organisations qui manipulent les systèmes d'information ? Plus simplement, il s'agit d'un ensemble de personnes. Dès qu'une activité dépasse une seule personne, on parle d'organisation. Les différents acteurs de l'organisation partagent les mêmes objectifs et ont chacun un rôle bien défini. Si des individus se constituent en organisation, c'est parce qu'ils ont un objectif de résultat.

Exemples de résultat :

- Produire un logiciel
- Mettre en ligne un commerce

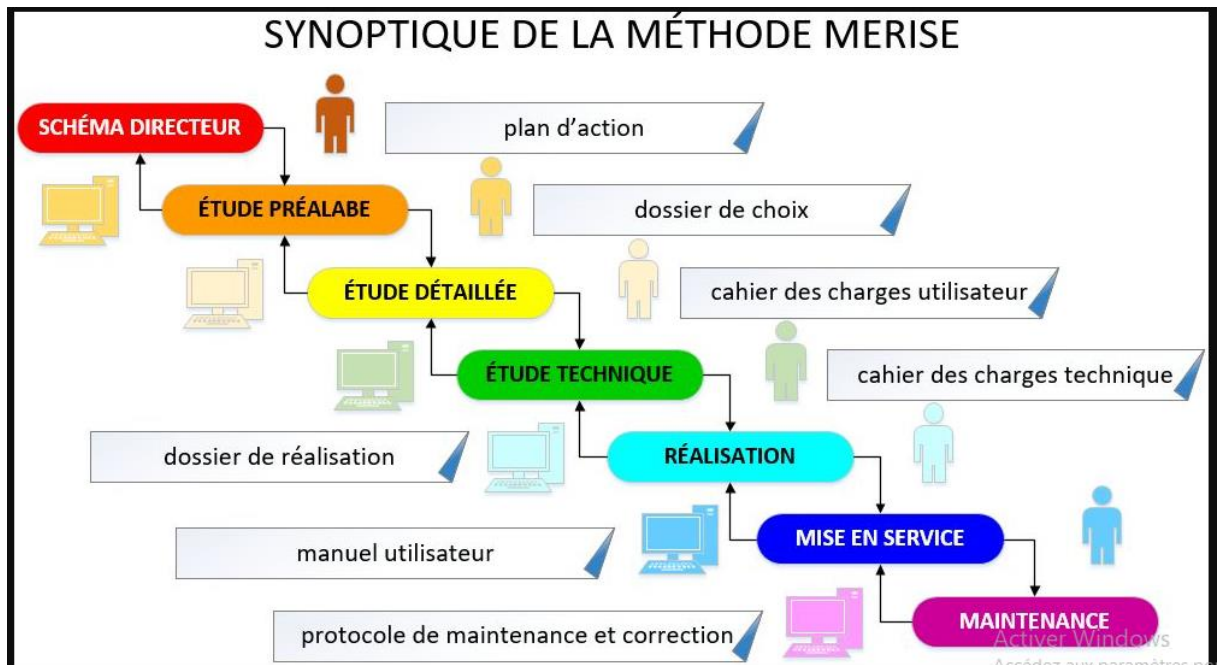
Pour atteindre cet objectif, le travail (le projet) est divisé en plusieurs tâches de manière cohérente et chaque acteur de l'organisation possède une tâche bien déterminée.

Les principales fonctions d'un SI sont :

- Collecter des informations
- Traiter et transmettre des informations
- Mémoriser des informations

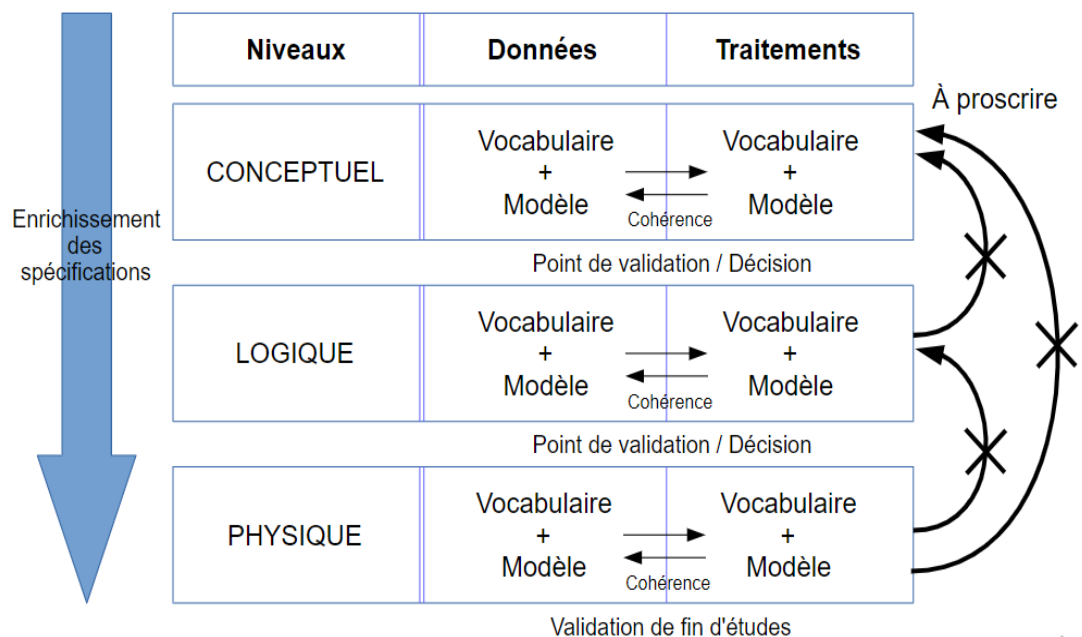
Quel est le lien avec l'informatique ? L'informatique facilite la gestion d'un SI. L'informatisation nécessite la mise en place de supports matériels et logiciels de stockage des données.

➤ **MERISE**



- La méthode Merise est une méthode d'analyse, de conception et de réalisation de systèmes d'informations.
- En amont, elle se situait dans le prolongement naturel d'un schéma directeur, souvent conduit suivant la méthode RACINES, très présente notamment dans le secteur public.
- Les projets Merise étaient généralement des projets de grande ampleur de refonte d'un existant complexe, dans un environnement grand système. La méthode a aussi connu des tentatives d'adaptation avec les SGBD relationnels, les différentes interfaces homme-machine IHM, l'Orienté objet, le développement micro, les outils CASE, la rétro-ingénierie... mais qui n'ont pas connu le même succès.
- La méthode est essentiellement française. Elle a des équivalents à l'étranger en ce qui concerne les modèles de données (avec des différences, par exemple les cardinalités ne sont pas aussi détaillées dans les modèles anglosaxons). En revanche la modélisation des traitements est beaucoup plus complexe que dans les méthodes anglo-saxonnes.

- Sa mise en œuvre peut paraître lourde. On consacre beaucoup de temps à concevoir et à pré-documenter avant de commencer à coder, ce qui pouvait sembler nécessaire à une époque où les moyens informatiques n'étaient pas aussi diffusés qu'aujourd'hui. Cela dit, elle évite l'écueil inverse du développement micro, qui souffre du manque de documentation, et où les erreurs sont finalement très coûteuses à réparer a posteriori.
- Même si les échanges et la consultation entre concepteurs et utilisateurs sont formellement organisés, on a aussi reproché à Merise d'utiliser un formalisme jugé complexe (surtout pour les modèles de données), qu'il faut d'abord apprendre à manier, mais qui constitue ensuite un véritable langage commun, puissant et rigoureux pour qui la maîtrise.
- L'articulation très codifiée et bien balisée des différentes étapes, avec un descriptif très précis des résultats attendus est ce qui reste aujourd'hui de mieux connu et de plus utilisé.



- La méthode Merise d'analyse et de conception propose une démarche articulée simultanément selon 3 axes pour hiérarchiser les préoccupations et les questions auxquelles répondre lors de la conduite d'un projet :
- NIVEAU CONCEPTUEL

- L'étude conceptuelle Merise s'attache aux invariants de l'entreprise ou de l'organisme du point de vue du métier : quelles sont les activités, les métiers gérés par l'entreprise, quels sont les grands processus traités, de quoi parle-t-on en matière de données, quelles notions manipule-t-on ?... et ce indépendamment des choix techniques (comment fait-on ?) ou d'organisation (qui fait quoi ?) qui ne seront abordés que dans les niveaux suivants.
- Au niveau conceptuel on veut décrire, après abstraction, le modèle (le système) de l'entreprise ou de l'organisme :
- le Modèle conceptuel des données (ou MCD), schéma représentant la structure du système d'information, du point de vue des données, c'est-à-dire les dépendances ou relations entre les différentes données du système d'information (par exemple : le client, la commande, les produits, etc.),
- et le Modèle conceptuel des traitements (ou MCT), schéma représentant les traitements, en réponse aux événements à traiter (par exemple : la prise en compte de la commande d'un client).

➤ UML

UML (Unified Modeling Language), traduit en français « langage de modélisation objet unifié » est né en 1995 de la fusion des trois méthodes qui ont le plus influencé la modélisation objet : OMT, Booch et OOSE. Il convient à tous les langages objets (C++ (Héritage multiple, Template), Java (Interface), C#, . . .) UML est normalisé par l'OMG (Object Management Group), en fin 1997. OMG regroupe plus de 80 des plus grandes Entreprises du monde (HP, IBM, SONY, ERICSON, MICROSOFT,...) UML est un langage de modélisation orientée objet : toutes les entités modélisées sont des objets ou se rapporte à des objets. C'est n'est pas une méthode, cependant il a été adopté par toutes les méthodes orientées objet.

➤ Avantages d'UML

- UML présente plusieurs avantages :
- Transcender les contraintes d'implémentation liées aux langages et aux systèmes;
- Représenter des concepts abstraits (graphiquement);

- Limiter les ambiguïtés (parler un langage commun, au vocabulaire précis, indépendant des langages orientés objet);
- Faciliter l'analyse (simplifier la comparaison et l'évaluation de solutions);
- Ne pas effectuer une analyse fonctionnelle et se contenter d'une implémentation objet, mais penser objet dès le départ,
- Définir les vues qui permettent de décrire tous les aspects d'un système avec des concepts objets.
- Sécuriser d'avantage les données (par l'encapsulation)
- Réutiliser efficacement l'existant (par l'héritage et le polymorphisme)
- Une panoplie d'outils et de langages performants pour le développement.
- Bases de données orientées objet (O2, ObjectStore, Versant...)

➤ Inconvénients d'UML

Cependant, UML présente quelques inconvénients. En effet, comme le langage est précis, il nécessite un effort d'apprentissage. De plus, pour un projet donné, il est souvent difficile de naviguer dans l'abstrait, donc difficulté d'identifier les objets.

UML est avant tout un support de communication performant, qui facilite la représentation et la compréhension de solutions objet :

Sa notation graphique permet d'exprimer visuellement une solution objet, ce qui facilite la comparaison et l'évaluation de solutions.

L'aspect formel de sa notation, limite les ambiguïtés et les incompréhensions.

Son indépendance par rapport aux langages de programmation, aux domaines d'application et aux processus, en font un langage universel.

La notation graphique d'UML n'est que le support du langage. La véritable force d'UML, c'est qu'il repose sur un métamodèle.

En d'autres termes : la puissance et l'intérêt d'UML, c'est qu'il normalise sémantique des concepts qu'il véhicule !

Qu'une association d'héritage entre deux classes soit représentée par une flèche terminée par un triangle ou un cercle, n'a que peu d'importance par rapport au sens que cela donne à votre modèle. La

notation graphique est essentiellement guidée par des considérations esthétiques, même si elle a été pensée dans ses moindres détails.

Par contre, utiliser une relation d'héritage, reflète l'intention de donner à votre modèle un sens particulier. Un "bon" langage de modélisation doit permettre à n'importe qui de déchiffrer cette intention de manière non équivoque ! Il est donc primordial de s'accorder sur la sémantique des éléments de modélisation, bien avant de s'intéresser à la manière de les représenter.

Le métamodèle UML apporte une solution à ce problème fondamental.

UML est donc bien plus qu'un simple outil qui permet de "dessiner" des représentations mentales... Il permet de parler un langage commun, normalisé mais accessible, car visuel. Il représente un juste milieu entre langage mathématique et naturel, pas trop complexe mais suffisamment rigoureux, car basé sur un méta modèle.

UML comme cadre d'une analyse objet:

Une autre caractéristique importante d'UML, est qu'il cadre l'analyse. UML permet de représenter un système selon différentes vues complémentaires : les diagrammes. Un diagramme UML est une représentation graphique, qui s'intéresse à un aspect précis du modèle ; c'est une perspective du modèle.

Chaque type de diagramme UML possède une structure (les types des éléments de modélisation qui le composent sont prédéfinis) et véhicule une sémantique précise (il offre toujours la même vue d'un système).

Combinés, les différents types de diagrammes UML offrent une vue complète des aspects statiques et dynamiques d'un système. Les diagrammes permettent donc d'inspecter un modèle selon différentes perspectives et guident l'utilisation des éléments de modélisation (les concepts objet), car ils possèdent une structure.

Une caractéristique importante des diagrammes UML, est qu'ils supportent l'abstraction. Cela permet de mieux contrôler la complexité dans l'expression et l'élaboration des solutions objet.

UML opte en effet pour l'élaboration des modèles, plutôt que pour une approche qui impose une barrière stricte entre analyse et

conception. Les modèles d'analyse et de conception ne diffèrent que par leur niveau de détail, il n'y a pas de différence dans les concepts utilisés. UML n'introduit pas d'éléments de modélisation propres à une activité (analyse, conception...) ; le langage reste le même à tous les niveaux d'abstraction.

Cette approche simplificatrice facilite le passage entre les niveaux d'abstraction. L'élaboration encourage une approche non linéaire, les "retours en arrière" entre niveaux d'abstraction différents sont facilités et la traçabilité entre modèles de niveaux différents est assurée par l'unicité du langage.

UML favorise donc le prototypage, et c'est là une de ses forces. En effet, modéliser une application n'est pas une activité linéaire. Il s'agit d'une tâche très complexe, qui nécessite une approche itérative, car il est plus efficace de construire et valider par étapes, ce qui est difficile à cerner et maîtriser.

UML permet donc non seulement de représenter et de manipuler les concepts objet, il sous-entend une démarche d'analyse qui permet de concevoir une solution objet de manière itérative, grâce aux diagrammes, qui supportent l'abstraction.

Comme UML n'impose pas de méthode de travail particulière, il peut être intégré à n'importe quel processus de développement logiciel de manière transparente. UML est une sorte de boîte à outils, qui permet d'améliorer progressivement vos méthodes de travail, tout en préservant vos modes de fonctionnement.

Intégrer UML par étapes dans un processus, de manière pragmatique, est tout à fait possible. La faculté d'UML de se fondre dans le processus courant, tout en véhiculant une démarche méthodologique, facilite son intégration et limite de nombreux risques (rejet des utilisateurs, coûts...).

➤ **ETUDE COMPARATIVE**

Merise	UML
méthode d'analyse et de conception de système d'information	langage de représentation d'un système d'information.
méthode de modélisation de données et traitements orienté bases de données relationnelles.	système de notation orienté objet.
relationnel	objet.
Franco-français	International
schéma directeur, étude préalable, étude détaillée et la réalisation.	langage de modélisation des systèmes standard, qui utilise des diagrammes pour représenter chaque aspect d'un systèmes ie: statique, dynamique,...en s'appuyant sur la notion d'orienté objet
plus adapté à une approche théorique	plus orientée vers la conception
du "bottom up" de la base de donnée vers le code	du "top down" du modèle vers la base de donnée.

➤ CONCEPTS ANALYSE ET CONCEPTION

❖ MCD

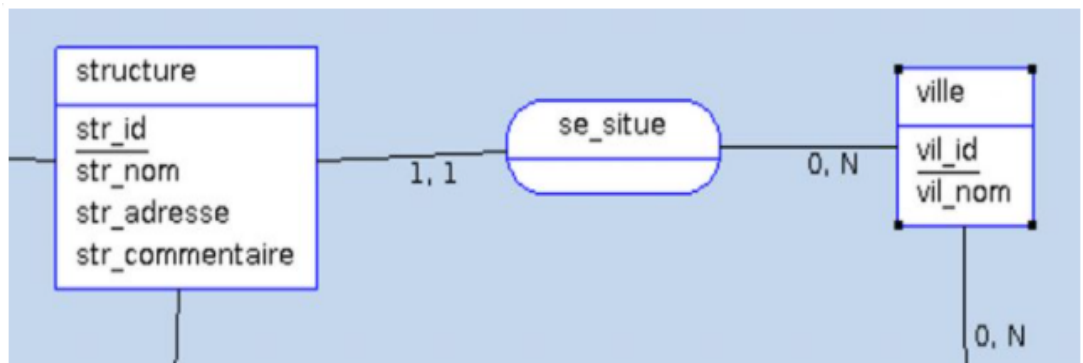
Le modèle conceptuel des données (MCD) a pour but d'écrire de façon formelle les données qui seront utilisées par le système d'information. Il s'agit donc d'une représentation des données, facilement compréhensible, permettant de décrire le système d'information à l'aide d'entités.

➤ LES ENTITES

- Dans la méthode Merise au niveau du MCD, une entité correspond à un ensemble homogène d'informations qui correspondent au même « objet » à informatiser. Cette entité a un nom unique afin de la manipuler facilement. Plus tard dans l'analyse, l'entité se transforme en table et devient concrètement une table lors de la réalisation effective de la base de données.
- Cet ensemble d'informations, l'entité, partage les mêmes caractéristiques et peut être manipulé au sein du système d'information mais aussi en discutant entre informaticiens et

personnes du métier. L'entité « contrat » et l'entité « client » sont typiques des bases de données de gestion. Isoler, comprendre le rôle et les informations qui se trouvent dans ces coquilles est une des premières étapes de l'analyse.

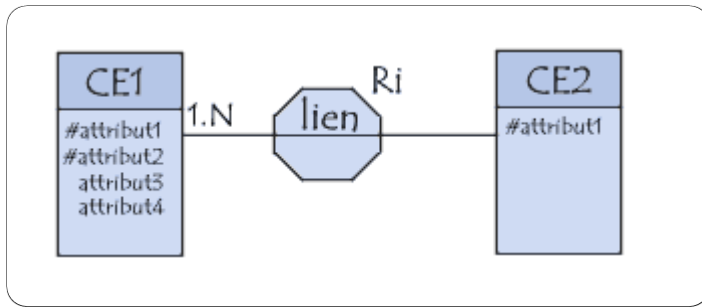
- Une entité reste cependant peu utile si elle n'est pas remplie avec des informations calibrées et si elle ne discute pas avec les autres entités de la base de données : on parle alors de propriétés et d'associations ce qui a donné le nom de modèle entités-associations que l'on croise parfois.
- Dans un MCD, les entités sont représentées par des rectangles et dans le cartouche supérieur de chaque rectangle est écrit au singulier le nom de l'entité.



2 entités (structure et ville) représentant des objets du monde réel qui seront informatisés

➤ LA CARDINALITE

- Les cardinalités permettent de caractériser le lien qui existe entre une entité et la relation à laquelle elle est reliée. La cardinalité d'une relation est composée d'un couple comportant une borne maximale et une borne minimale, intervalle dans lequel la cardinalité d'une entité peut prendre sa valeur :
- la borne minimale (généralement 0 ou 1) décrit le nombre minimum de fois qu'une entité peut participer à une relation
- la borne maximale (généralement 1 ou n) décrit le nombre maximum de fois qu'une entité peut participer à une relation



❖ MLD

➤ 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 .

✚ Représentation du MLDR

On représente le modèle logique de données relationnel de deux manières, soit en intention ou soit en extension.

a) Représentation en intention

T_RECEPTION (numrec, nomal, datearrive, nomrec)

T_MALADE (numal, nufich, nomal, fonction, sexe, adresse, nompere, nomere, nomconj, tel, reference, nomeglise, #numrec, #numcons, #numlab, #numtr)

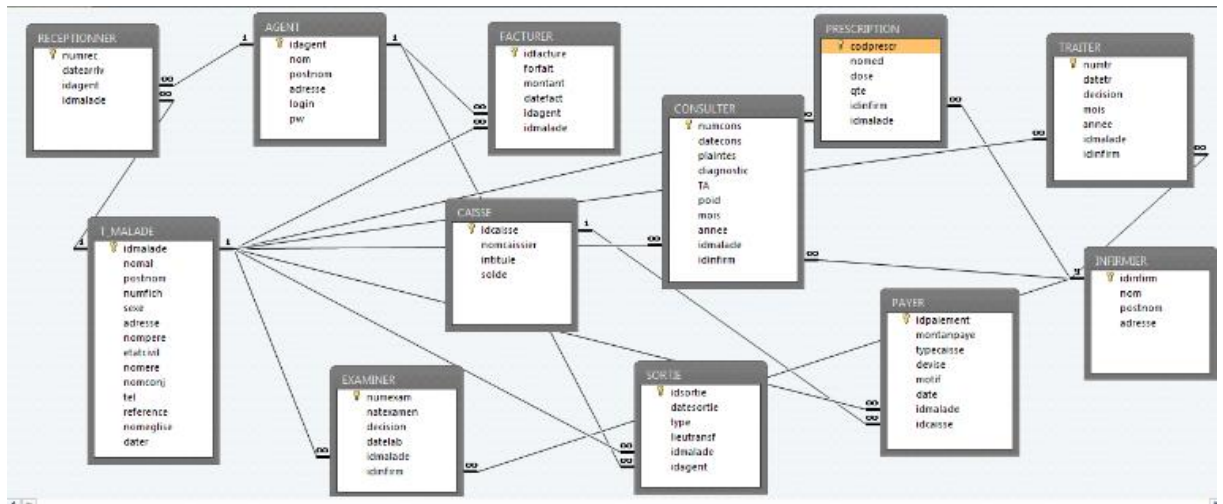
T_CONSULTATION (numcons, nomal, nominfir, plaintes, diagnostic, datecons, mois, année)

T_LABORATOIRE (numlab, nomal, natexam, decision, datelab)

T_TRAITEMENT (numtr, nomal, nominf, décision, datetr, mois, année)

T_PRESCRIPTION (codmed, nomed, dose, qté, #numcons)

b) Présentation en extension



MODELE RELATIONNEL

- ❖ Les données sont organisées en relations
- ❖ Tables: relations
- ❖ Colonnes: attributs Lignes: n-uplets (ou tuples)

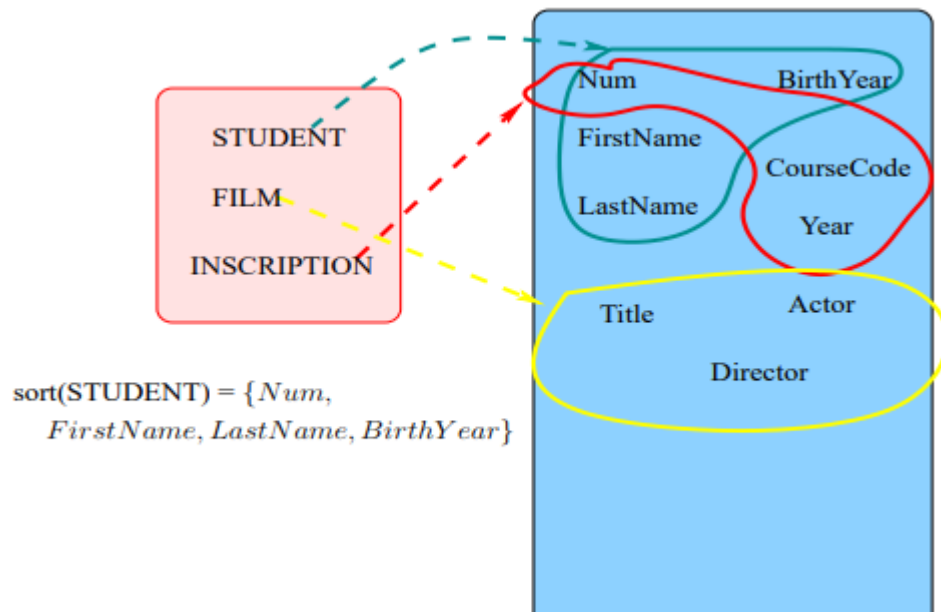
STUDENT	Num	FirstName	LastName	BirthYear
	2008120	Dumont	Marie	1980
	2008122	Dubois	Paul	1980
	2008125	Martin	Jean	1981
	

Schéma d'une base de données relationnel

- ❖ Ensemble de noms de tables
 - ❖ Ensemble d'attributs pour chaque table
- STUDENT [Num, FirstName, LastName, BirthYear]
- INSCRIPTION[Num, CourseCode, Year]

Modèle relationnel - Définitions formelles

- ❖ Nous considérons trois ensembles (infinis et dénombrables) disjoints:
- ❖ att: attributs
- ❖ dom: domaine
- ❖ relname: noms de relations
- ❖ Un schéma de relation est un nom de relation (dans relname).
- ❖ Le sort d'une relation est une fonction qui associe à chaque nom de relation un ensemble fini d'attributs (un sous ensemble de att)



EXEMPLE

dom = {The Cameraman, Hitchcock ,Grace Kelly, Lubitsch, 10, tata . . .}

att = {title, director, actor, . . .}

relname = {Film, Schedule, . . .}

Le sort (Film) = {title, director, actor} sort(Schedule) = {theater, title, director}

➤ **SQL**

✓ 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.

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

✓ 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.

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

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

- Jointure
- Projection
- Sous requête
- Alias (AS)
- Commande

- UNION

- GROUP BY

- ORDER BY

- HAVING o Fonction

- DISTINCT

- COUNT

- LIMIT

- LIKE %% (B%A)

- IN & NOT IN

- BETWEEN

- AVG

- MIN

- MAX

- SUM