
Le modèle conceptuel de données

Définition

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.

I) Notion Analyse et Conception

De nombreux modèles de données sont conçus selon un processus bien particulier : le modelleur crée un modèle logique, puis un modèle physique. En général, les modèles logiques décrivent des entités et des attributs et les relations qui les lient, offrant ainsi une représentation claire de la finalité des données dans le métier. Les modèles physiques mettent ensuite en œuvre le modèle logique sous forme de tables, de colonnes, de types de données et d'index, associés à de brèves règles d'intégrité des données. Ces règles définissent les clés primaires et étrangères et les valeurs par défaut. En outre, des vues, des déclencheurs et des procédures stockées peuvent être définis afin de faciliter la mise en œuvre, au besoin. Le modèle physique définit également l'allocation de mémoire sur le disque sur la base des options de configuration fournies par la plupart des systèmes hôtes (Oracle, MS SQL Server, MySQL, etc.).

a) Merise

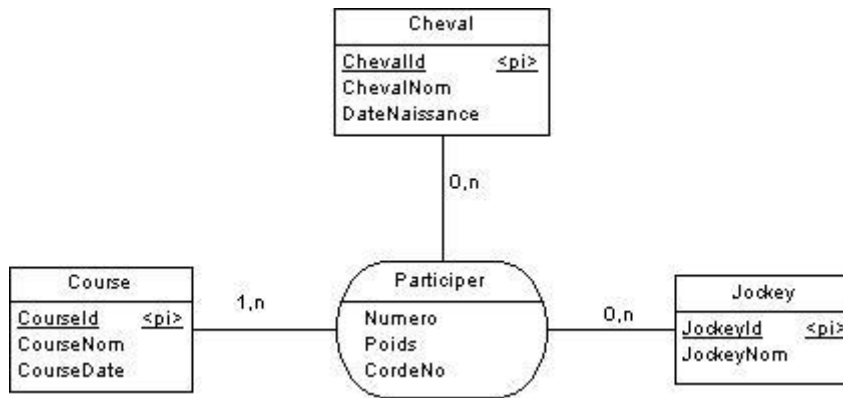
MERISE est une méthode de conception, de développement et de réalisation de projets informatiques.

Le but de cette méthode est d'arriver à concevoir un système d'information. La méthode MERISE est basée sur la séparation des données et des traitements à effectuer en plusieurs modèles conceptuels et physiques. La séparation des données et des traitements assure une longévité au modèle. En effet, l'agencement des données n'a pas à être souvent remanié, tandis que les traitements le sont plus fréquemment.

b)Uml Unified Modeling Language

UML ,On le traduit par « Langage de modélisation unifié ». La notation UML est un **langage visuel** constitué d'un ensemble de schémas, appelés des **diagrammes**, qui donnent chacun une vision différente du projet à traiter. UML nous fournit donc des diagrammes pour **représenter** le logiciel à développer : son fonctionnement, sa mise en route, les actions susceptibles d'être effectuées par le logiciel, etc.

c) Etude comparative



II) Concepts Analyse et Conception A) MCD

Modèle conceptuel des données

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.

a) Les entités

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

On appelle **classe d'entité** un ensemble composé d'entités de même type, c'est-à-dire dont la définition est la même. Le classement des entités au sein d'une classe s'appelle *classification* (ou *abstraction*). Une entité est une *instanciation* de la classe. Chaque entité est composée de propriétés, données élémentaires permettant de la décrire.

b) Les attributs

Les attributs d'entité sont les éléments d'information de base attachés à une entité. Lorsque vous générez un Modèle Physique de Données (MPD) à partir d'un Modèle Conceptuel de Données (MCD), les attributs d'entité deviennent des colonnes de tables dans le MPD.

c) Les 'occurrences

On définit l'occurrence ou l'instance la réalisation particulière d'une propriété d'une association ou d'une entité.

La cardinalité

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

Formalisme de Représentation

Le modèle conceptuel des données (**MCD**) a pour but de **représenter** de façon structurée les données qui seront utilisées par le système d'information. Le modèle conceptuel des données décrit la sémantique c'est-à-dire le sens attaché à ces données et à leurs rapports et non à l'utilisation qui peut en être fait

B) MLD

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

C) Règles de passage du MCD au MLDR

Règle numéro 1 :

a) Une entité du MCD devient une relation, c'est à dire une table. Dans un SGBD (Système de Gestion de base de données) de type relationnel, une table est une structure tabulaire dont chaque ligne correspond aux données d'un objet enregistré (d'où le terme enregistrement) et où chaque colonne correspond à une propriété de cet objet. Une table contiendra donc un ensemble d'enregistrements. Une ligne correspond à un enregistrement. Une colonne correspond à un champ. La valeur prise par un champ pour un enregistrement donné est située à l'intersection ligne-colonne correspondant à enregistrement-champ. Il n'y a pas de limite théorique au nombre d'enregistrements que peut contenir une table. Par contre, la limite est liée à l'espace de stockage. b) Son identifiant devient la clé primaire de la relation. La clé primaire permet d'identifier de façon unique un enregistrement dans la table. Les valeurs de la clé primaire sont donc uniques. Les valeurs de la clé primaire sont obligatoirement non nulles. Dans la plupart des SGBDR (Système de Gestion de Base de Données Relationnelle), le fait de définir une clé primaire donne lieu automatiquement à la création d'un index. Un index est un fichier interne au SGBD. L'utilisateur standard n'a pas besoin d'y accéder. L'index a pour but d'accélérer les

traitements de recherche, de tri, de filtre et notamment sur les tables avec de nombreux enregistrements. La contrepartie est que l'index nécessite de l'espace mémoire et surtout, les temps d'insertion, de suppression d'enregistrements sont plus importants car il faut mettre à jour à la fois la table et l'index.

Règle numéro 2 : Une association de type 1:N (c'est à dire qui a les cardinalités maximales positionnées à « 1 » d'une côté de l'association et à « n » de l'autre côté) se traduit par la création d'une clé étrangère dans la relation correspondante à l'entité côté « 1 ». Cette clé étrangère référence la clé primaire de la relation correspondant à l'autre entité.

Le SQL

Le langage **SQL** (Structured Query Language) est un langage informatique utilisé pour exploiter des bases de données. Il permet de façon générale la définition, la manipulation et le contrôle de sécurité de données.

Dans la pratique, le langage **SQL** est utilisé pour créer des tables, ajouter des enregistrements sous forme de lignes, interroger une base de données, la mettre à jour, ou encore gérer les droits d'utilisateurs de cette base de données. Il est bien supporté par la très grande majorité des systèmes de gestion de base de données (SGBD). Créé au début des années 1970 par Donald D. Chamberlin et Raymond F. Boyce, tous deux chez IBM, le langage **SQL** est aujourd'hui reconnu comme une norme internationale.

SQL est un langage déclaratif, il n'est donc pas à proprement parlé un langage de programmation, mais plutôt une interface standard pour accéder aux bases de données.

Il est composé de trois sous ensembles :

- Le Langage de Définition de Données (LDD, ou en anglais DDL, *Data Definition Language*) pour créer et supprimer des objets dans la base de données (tables, contraintes d'intégrité, vues, etc.).

Exemple de commandes : CREATE DROP ALTER

- Le Langage de Manipulation de Données (LMD, ou en anglais DML, *Data Manipulation Language*) pour la recherche, l'insertion, la mise à jour et la suppression de données. Le LMD est basé sur les opérateurs relationnels, auxquels sont ajoutés des fonctions de calcul d'agrégats et des instructions pour réaliser les opérations d'insertion, mise à jour et suppression.

Exemple de commandes : INSERT UPDATE DELETE SELECT

- LID : « Langage d'Interrogation de Données » : permet de rechercher des informations utiles en interrogeant la base de données.

Le langage d'interrogation de données (LID) permet **d'établir une combinaison d'opérations** portant sur des tables (relation). Le résultat de cette combinaison d'opérations est lui-même une table dont l'existence ne dure qu'un temps.

LES jointures en Lid

Les jointures permettent d'exploiter pleinement le modèle relationnel des tables d'une base de données.

Elle sont faites pour mettre en relation deux (ou plus) tables concourant à rechercher la réponse à des interrogations. Une jointure permet donc de combiner les colonnes de plusieurs tables.

La Projection

(SELECT ... FROM ...) L'opération de projection consiste à choisir le nom des attributs de la (ou des) table(s) que l'on souhaite voir apparaître dans la réponse. Si l'on veut afficher toutes les colonnes, il faut utiliser « * ». Les noms des attributs sont données dans la clause SELECT. Si l'on veut éliminer les doublons, il faut utiliser « DISTINCT ». Il est utilisé pour chaque attribut non clé. Si l'on souhaite créer un alias pour une meilleure lisibilité (attributs et tables), il faut utiliser « AS ». Exemple : SELECT idclient AS identifiant FROM client AS

Sous-requête

Dans le langage SQL une sous-requête (aussi appelé “requête imbriquée” ou “requête en cascade”) consiste à exécuter une requête à l'intérieur d'une autre requête. Une requête imbriquée est souvent utilisée au sein d'une clause WHERE ou de HAVING pour remplacer une ou plusieurs constante.

ALIAS sur une table. Permet d'attribuer un autre nom à une table dans une requête **SQL**. Cela peut aider à avoir des noms plus court, plus simple et plus facilement compréhensible. Ceci est particulièrement vrai lorsqu'il y a des jointures.

LES COMMANDES SQL sont généralement classées en deux catégories : Les **commandes** de manipulation de données, utilisées pour récupérer, ajouter, supprimer et/ou modifier des informations dans la base de données. En particulier, cette catégorie inclut les **commandes** SELECT, INSERT, UPDATE et DELETE.

SQL UNION

La commande UNION de SQL permet de mettre bout-à-bout les résultats de plusieurs requêtes utilisant elles-même la commande SELECT. C'est donc une commande qui permet de concaténer les résultats de 2 requêtes ou plus. Pour l'utiliser il est nécessaire que chacune des requêtes à concaténer retournes le même nombre de colonnes, avec les mêmes types de données et dans le même ordre.

SQL GROUP BY

La commande GROUP BY est utilisée en SQL pour grouper plusieurs résultats et utiliser une fonction de totaux sur un groupe de résultat. Sur une table qui contient toutes les ventes d'un magasin, il est par exemple possible de lister regrouper les ventes par clients identiques et d'obtenir le coût total des achats pour chaque client.

SQL ORDER BY

La commande ORDER BY permet de trier les lignes dans un résultat d'une requête SQL. Il est possible de trier les données sur une ou plusieurs colonnes, par ordre ascendant ou descendant

HAVING

La condition HAVING en SQL est presque similaire à WHERE à la seule différence que HAVING permet de filtrer en utilisant des fonctions telles que SUM(), COUNT(), AVG(), MIN() ou MAX().

Fonction – DISTINCT

L'utilisation de la commande SELECT en SQL permet de lire toutes les données d'une ou plusieurs colonnes. Cette commande peut potentiellement afficher des lignes en doubles. Pour éviter des redondances dans les résultats il faut simplement ajouter DISTINCT après le mot SELECT.

COUNT()

En SQL, la fonction d'agrégation COUNT() permet de compter le nombre d'enregistrement dans une table. Connaître le nombre de lignes dans une table est très pratique dans de nombreux cas, par exemple pour savoir combien d'utilisateurs sont présents dans une table ou pour connaître le nombre de commentaires sur un article.

LIMIT

La clause LIMIT est à utiliser dans une requête SQL pour spécifier le nombre maximum de résultats que l'on souhaite obtenir. Cette clause est souvent associée à un OFFSET, c'est-à-dire effectuer un décalage sur le jeu de résultat. Ces 2 clauses permettent par exemple d'effectuer des systèmes de pagination (exemple : récupérer les 10 articles de la page 4).

LIKE

L'opérateur LIKE est utilisé dans la clause WHERE des requêtes SQL. Ce mot-clé permet d'effectuer une recherche sur un modèle particulier. Il est par exemple possible de rechercher les enregistrements dont la valeur d'une colonne commence par telle ou telle lettre. Les modèles de recherches sont multiples.

L'opérateur BETWEEN

est utilisé dans une requête SQL pour sélectionner un intervalle de données dans une requête utilisant WHERE. L'intervalle peut être constitué de chaînes de caractères, de nombres ou de dates. L'exemple le plus concret consiste par exemple à récupérer uniquement les enregistrements entre 2 dates définies.

AVG()

La fonction d'agrégation AVG() dans le langage SQL permet de calculer une valeur moyenne sur un ensemble d'enregistrement de type numérique et non nul.

MIN()

La fonction d'agrégation MIN() de SQL permet de retourner la plus petite valeur d'une colonne sélectionnée. Cette fonction s'applique aussi bien à des données numériques qu'à des données alphanumériques.

MAX()

Dans le langage SQL, la fonction d'agrégation MAX() permet de retourner la valeur maximale d'une colonne dans un set d'enregistrement. La fonction peut s'appliquée à des données numériques ou alphanumériques. Il est par exemple possible de rechercher le produit le plus cher dans une table d'une boutique en ligne.

SUM()

Dans le langage SQL, la fonction d'agrégation SUM() permet de calculer la somme totale d'une colonne contenant des valeurs numériques. Cette fonction ne fonction que sur des colonnes de types numériques (INT, FLOAT ...) et n'additionne pas les valeurs NULL.