Modélisation de Données

Avec UML

Table des matières

[I. Introduction2](#_Toc347950326)

[II. Les Niveaux d’abstraction selon l’ANSI4](#_Toc347950327)

[III. Le Modèle Conceptuel de Données – MCD – *Concepts de bases*5](#_Toc347950328)

[III.1 Entité, attribut et association5](#_Toc347950329)

[III.2 Contraintes sur les attributs et les associations6](#_Toc347950330)

[III.2.1 Contrainte d’intégrité d’entité6](#_Toc347950331)

[III.2.2 Multiplicité (association binaire)7](#_Toc347950332)

[III.2.3 Contrainte de domaine des attributs7](#_Toc347950333)

[III.2.4 Dépendance fonctionnelle des attributs à l’identifiant8](#_Toc347950334)

1. Introduction

**Donnée (Data) :** Représentation d’un élément d’information, tel qu’un chiffre ou un fait, codé dans un format permettant son stockage et son traitement par ordinateur. Le type de donnée est la nature du codage utilisé pour représenter une donnée élémentaire et les opérations applicables à cette donnée. Les types courants sont : entier, réel, texte, date, image.

**Information (Information) :** Une donnée ou un ensemble de données qui a ou ont été interprétée(s).

**Base de données BD (Database)** **:** Ensemble structuré d’éléments d’*information* agencés sous forme de *tables*, dans lesquels les *données* sont organisées selon certains critères en vue de permettre leur exploitation pour répondre aux besoins d’information d’une organisation.

**Application de bases de données (Database application) :** Utilisation de moyens informatiques pour répondre à **un besoin déterminé** en faisant appel à une ou plusieurs base de données à travers un système de gestion de bases de données (SGBD).

Exemples :

Commerce électronique – Accès au catalogue, transaction d’achat avec autorisation de paiement par carte de crédit ne sauraient être mis en œuvre sans l’utilisation de plusieurs bases de données gérées soit par l’entreprise, soit par une institution financière partenaire. Si le niveau de stock d’un produit atteint un seuil de rupture, le système d’information du commerçant peut émettre sur-le-champ une commande auprès d’un fournisseur par voie électronique. Une base de données fournisseur va permettre de recevoir et de donner suite à la commande en confirmant une date de livraison, puis en procédant à la facturation.

Support à la décision –Systèmes d’information permettant la recherche active et l’exploitation, sur le plan décisionnel, de l’ensemble des renseignements stratégiques essentiels qu’une entreprise doit posséder si elle veut faire face à la concurrence. Ce type d’application appelé système de *veille stratégique (business intelligence system BI)* fait appel à une masse considérables de données provenant de sources multiples recueillies sur une large échelle de temps, regroupées dans une base de données que l’on appelle **entrepôt de données**.

**Entrepôt de données (Data warehouse) :** Base de données spécialisée dans laquelle est centralisé un volume important de données consolidées à partir de différentes sources de renseignement d’une entreprise (ex : BD internes + données démographiques, socioéconomiques) et conçue de manière à ce que les personnes intéressées aient accès rapidement à l’information stratégique dont elles ont besoin.

**Système de gestion de bases de données SGBD (Database management system) :** Logiciels conçus pour exploiter les bases de données.

SGBD commerciaux : DB2 d’IBM, Oracle9i ou Oracle10g de Oracle, SQL Server ou Access de Microsoft.

SGBD libre : MySQL

Caractéristiques des SGBD :

1. **Indépendance entre les données et les applications**. Le programmeur d’une application ne s’occupe pas des aspects structurels des fichiers d’une base de données. Si la structure de la base de données devait être changée, les programmes ne seront pas impactés.
2. **Contrôle centralisé des données pour éviter toute redondance.** La redondance des informations pose un problème lors des mises à jour et au niveau des performances.
3. **Partage des données et accès concurrents.** Mécanisme d’accès concurrent qui doit éviter que des opérations en simultanées mènent à des incohérences. Ce mécanisme porte le nom de *verrou,* il interdit tout accès à une donnée tant qu’un autre utilisateur procède à leur traitement.
4. **Gestion de la cohérence et de l’intégrité des données.** Une contrainte d’intégrité des données est un ensemble de règles, définies par le concepteur d’une base de données, qui devront en tout temps être respectées. Ces règles sont gérées par le SGBD qui en assure l’application et informe l’utilisateur lorsque l’une d’elles est transgressée. On écarte ainsi, dès leur saisie, des valeurs jugées inacceptables. On peut établir pour une donnée son type ou sa taille mais aussi fixer une plage de valeurs acceptables.
5. **Description des données stockées sous forme de métadonnées (Metadata).** Schéma physique d’une base de données : Collection des composants constitutifs de la structure d’une base de données, notamment les propriétés des données, les domaines (types des données), les fichiers, les contraintes d’intégrité. Ce schéma est défini par le biais d’un langage de définition de données faisant partie intégrante du SGBD.
6. **Gestion de la sécurité.** Privilèges d’accès aux données accordés aux utilisateurs, cryptage de certaines informations, sauvegarde pour la récupération des données suite à une panne. Administrateur de base de données est responsable de la réalisation physique de la base de données sous forme de schéma physique, du contrôle de la sécurité et de l’intégrité, de la maintenance du SGBD et des performances des applications de bases de données.

**Système d’information :** système constitué des ressources humaines, des ressources informatiques (équipement, logiciel, données) et des procédures permettant d’acquérir, de stocker, de traiter et de diffuser les éléments d’information pertinents au fonctionnement d’une entreprise ou d’une organisation.

1. Les Niveaux d’abstraction selon l’ANSI (American National Standards Institue)

Domaine à modéliser

Vue externe :

Utilisateur 1

Vue externe :

Utilisateur 2

Vue externe :

Utilisateur 3

Plus haut niveau d’abstraction. Façon dont un utilisateur perçoit les données. Chaque utilisateur selon ses besoins peut avoir une vue différente

**Niveau externe**

**Niveau interne**

Schéma Conceptuel

Schéma

Logique

Schéma Physique

BD

Modèle conceptuel de données (**MCD**). Le niveau conceptuel assure une couche d’***isolation*** entre les besoins des utilisateurs et l’organisation physique de la base de données. Chaque vue est un sous-ensemble du schéma conceptuel.

Au plus bas niveau d’abstraction se situe le niveau interne. Il représente la manière dont le SGBD perçoit les données stockées dans une base de données. La description des données au niveau interne est donnée par deux modèles : modèle logique des données (**MLD**) et modèle physique des données (**MPD**). Ces modèles sont fortement dépendants du type de SGBD choisi pour mettre en œuvre la base de données.

**Niveau conceptuel**

Organisation

Physique

des données

1. Le Modèle Conceptuel de Données – MCD – *Concepts de bases*

Un modèle conceptuel de données est une représentation des besoins en matière de données pour un système d’information. Il met en évidence les entités, leurs attributs, les associations et contraintes entre ces entités pour un domaine donné. Cette représentation ne comporte aucune indication concernant la structure de mémorisation des données associées aux entités. Le modèle conceptuel est généralement représenté à l’aide du formalisme entité-association (EA).

## III.1 Entité, attribut et association

**Entité (Entity) :** Objet concret ou abstrait au sujet duquel une organisation est susceptible de conserver des données.

Toute entité possède des propriétés, appelés *attributs.*

**Attribut :** Donnée élémentaire qui sert à caractériser une propriété des entités et des associations dans un MCD.

« entité »

**Nom de l’entité**

id

attribut 1

attribut 2

« entité »

**Client**

No Client

Nom client

Prénom client

Représentation graphique : entité

**Occurrence d’une entité :** Elément particulier d’une entité type, identifiable de façon unique. (Instance).

« entité »

**Client**

CL113

Dupuis

Francky

« entité »

**Client**

No Client

Nom client

Prénom client

Deux occurrences de l’entité Client

« entité »

**Client**

CL256

Martin

Bob

Représentation graphique : **occurrence**

**Association :** Lien qui existe entre deux entités ou plus. Elle représente souvent la mémoire d’un évènement qui a permis d’établir un lien logique entre ces entités.

« entité »

**Client**

No Client

Nom client

Prénom client

« entité »

**Commande**

effectuer

No Commande

Date commande

Total commande

Représentation graphique : association

**Association réflexive**

« entité »

**Personne**

Id personne

Nom personne

Prénom personne

0..\*

0..\*

épouser

Représentation graphique : association réflexive

## III.2 Contraintes sur les attributs et les associations

### III.2.1 Contrainte d’intégrité d’entité

Pour chaque entité il doit exister un attribut ou un groupe d’attributs permettant d’identifier de façon unique chaque occurrence de cette entité.

Cet attribut ou ce groupe d’attributs est appelé **Identifiant**. L’identifiant doit **obligatoirement** avoir une valeur pour chaque occurrence de l’entité. Cette valeur doit être unique et non nulle.

La plupart des notations utilisent le soulignement pour marquer l’identifiant. En notation UML, on **ajoute une contrainte {Unique}** placée à la suite du nom de l’attribut qui indique que la valeur de cet attribut est unique.

La présence d’un identifiant marqué de la contrainte {Unique} pour une entité représente :

**Une contrainte d’intégrité d’entité**

effectuer

« entité »

**Client**

**No Client : {Unique}**

Nom client

Prénom client

« entité »

**Commande**

**No Commande : {Unique}**

Date commande

Total commande

Représentation graphique identifiant simple

**Identifiant simple :** Identifiant formé d’un seul attribut.

**Identifiant composé :** Identifiant formé de plusieurs attributs.

No Commande : {Unique}

Date commande

Total commande

« entité »

**Client**

**{Unique Nom client, prénom client}**

Nom client

Prénom client

Représentation graphique identifiant composé

**Règle d’identité :**Toute entité présente dans un MCD doit comporter **obligatoirement** un identifiant. Chaque occurrence de l’entité doit posséder une valeur pour cet attribut. La valeur de l’attribut identifiant devra être **stable**, cette valeur ne pourra changer. Deux occurrences de l’entité ne pourraient avoir la même valeur pour leur identifiant. La même valeur accordée plus d’une fois à un identifiant représente un **doublon**.

### III.2.2 Multiplicité (association binaire)

Contrainte inscrite à chaque extrémité d’une association binaire comportant un couple de valeurs (minimum – maximum) qui établit, pour chaque entité de l’association, les nombres minimum et maximum d’occurrences de l’autre entité qui peuvent lui être associées.

|  |  |
| --- | --- |
| **Multiplicité UML** | **Signification** |
| 0..1 | Au plus un |
| 1..1 (ou 1) | Exactement un |
| 0..\* (ou \*) | Plusieurs (indéterminé) |
| 1..\* | Au moins un |

effectuer

« entité »

**Client**

No Client : {Unique}

Nom client

Prénom client

« entité »

**Commande**

**0..\***

**1..1**

Représentation graphique avec multiplicités

### III.2.3Contrainte de domaine des attributs

**Les contraintes de domaine** de l’attribut vont fixer le type de données de l’attribut et les valeurs admissibles de l’attribut. Les outils de modélisation permettent d’inscrire ces contraintes dans ce qui est souvent appelé le **dictionnaire de données** du modèle. Elles ne sont pas toujours affichées dans les entités pour éviter d’alourdir le modèle.

Leur formulation peut être libre ou donnée grâce à un langage spécialisé.

|  |  |
| --- | --- |
| **Type** | **Signification** |
| int | Valeur numérique entière |
| real | Valeur numérique réelle |
| string | Chaîne de caractères |
| date | Date |
| boolean | Vrai ou faux |
| enum{} | Liste de valeurs admissibles |

S’il y a lieu, le type de données sera suivi d’une contrainte indiquant une restriction sur les valeurs que l’attribut pourra prendre.

No article : {Unique}

Nom article

Description article

« entité »

**Client**

No Client:**string**{Unique}

Nom client :**string {Non nul}**

Prénom client : **string**

Sexe : **enum{M, F}**

« entité »

**Commande**

No Commande :**string** {Unique}

Date commande : **date**

Total commande : **real {total commande >0}**

effectuer

Représentation graphique avec typage des attributs

### III.2.4Dépendance fonctionnelle des attributs à l’identifiant

**Règle de non redondance :** Chaque attribut du modèle conceptuel est unique, il ne peut y apparaître plus d’une fois à travers plusieurs entités du modèle.

« entité »

**Client**

No Client : {Unique}

Nom client

Prénom client

« entité »

**Commande**

No Commande : {Unique}

Date commande

Total commande

**No Client**

**Règle de construction :** Une entité ne peut en contenir une autre. Elle ne peut contenir un attribut d’une autre entité.

« entité »

**Commande**

contenir

« entité »

**Commande**

No Commande : {Unique}

Date commande

Total commande

« entité »

**Article**

0..\*

No Commande : {Unique}

Date commande

**No article**

**Nom article**

**Description article**

Total commande

1..\*

Une notion théorique importante est à la base de la règle de construction : **La dépendance fonctionnelle**.

Pour que la règle de construction soit respectée, **il faut que chaque attribut autre que l’identifiant *dépende fonctionnellement de l’identifiant*.** Une valeur de l’identifiant détermine une et une seule valeur pour chaque attribut.

**Cas d’un attribut qui dépend fonctionnellement de deux entités ou plus :**

Un attribut qui dépend fonctionnellement de deux entités ou plus doit être considéré comme un ***attribut de l’association***. Les attributs d’une association sont représentés en UML à l’aide d’une entité rattachée à l’association par une ligne en pointillée. Cette entité d’association est nécessaire pour regrouper les attributs qui ne peuvent appartenir ni à l’une ni à l’autre des entités associées. Une entité d’association est ce que les modélisateurs appellent une entité faible.

**Entité faible :** Type d’entité dont l’existence dépend de deux ou plusieurs entités dites fortes. Son identifiant se définit en fonction des identifiants des entités dont elle dépend.

**Client**

No Client : {Unique}

Nom client

Prénom client

**Commande**

1..1

comporte

**Article**

No article : {Unique}

Nom article

Description article

1..\*

0..\*

effectue

No Commande : {Unique}

Date commande

Total commande

0..\*

**Article commandé**

**Quantité commandée**

**Prix unitaire négocié**

Représentation graphique avec attributs associés

Représentation textuelle :

Client(No Client, Nom client, Prénom client)

Commande(No Commande, Date commande, Total commande)

Article(No article, Nom article, Description article)

Article\_commande(No Commande, No article, Quantite\_commandée, PrixU\_negocie)