

Régon 28 : Modèle relationnel et conception de bases de données

Niveau : Prépa

On illustrera les concepts avec un exemple concret minimal : un grossiste a une liste de produits dont les clients font la commande.

Objectif 1 : Stocker ces données, de façon à les rendre facilement utilisables, modifiables, etc...

Solution naïve 2 : Convertir une commande en chaîne de caractères et les stocker dans un grand fichier.

Exemple 3 : {produit : tomate, prix : 3, quantité : 50, client : Hervé, adresse : 113 rue du Swag}, {produit : tomate, prix : 3, qté : 10, ...}, ...

Problème 4 : Beaucoup de redondance, recherche compliquée

I-Schéma Entité - Association

Idée 5 : Pour éviter les redondances, on stocke à part les clients, les produits. Ensuite, on stocke les liens entre les deux.

Remarque 6 : On se place alors dans un paradigme appelé relationnel.

Définition 7 : Un schéma entité association est un graphe non orienté, composé de :

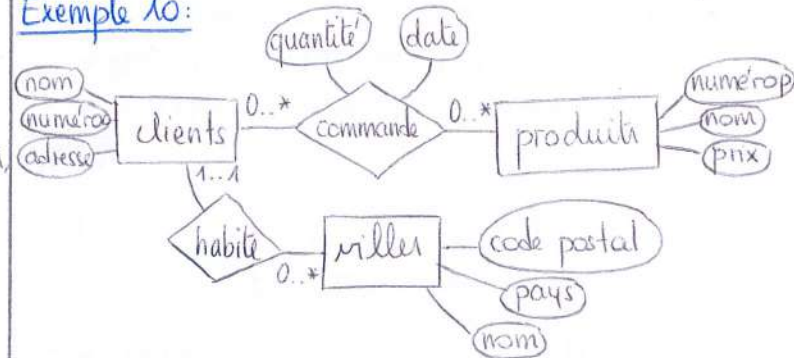
- * sommets appelés entités
- * arêtes appelées relations
- * étiquettes sur les arêtes : A chaque arête (u, v) , on associe une étiquette (x, y) , avec x et y pouvant prendre les valeurs $0..1$, $0..*$, $1..1$, $1..*$. (x, u) sera étiquetée par (y, x)
- * attributs sur les sommets et les arêtes précisant ce qu'ils représentent.

Explication 8 :

- les entités représentent les types des objets (ex : clients, produits, ...)
- les arêtes représentent les liens (ex : un client habite dans une ville)
- les étiquettes indiquent combien un objet précis peut avoir de liens, la première valeur étant le minimum et la deuxième le maximum (* représente +∞)
- les attributs représentent les composants (ex : le nom du client)

Représentation 9 : Les entités sont des carrés, les relations des traits avec un losange, les étiquettes sont aux bords des arêtes et les attributs des bulles liées à leur objet.

Exemple 10 :



Remarque 11 : Si on veut que chaque client ait au moins une commande, on met $1..*$ sur son étiquette associée à commande.

Propriété 12 : On peut convertir les relations $..* ..*$ en entité.

Exemple 13 :



II - Modèle relationnel

II.1 - Définitions

On cherche maintenant une manière de modéliser ce schéma pour le stocker en machine.

Def 14 : * Un domaine est un ensemble, fini ou non, de valeurs possibles ayant un nom (ex : $(\mathbb{N}, \text{"poids"})$, $(\text{flottant}, \text{"taille"})$, $(\text{chaîne de caractères}, \text{"nom"})$, ...).

* Si $(D_1, \text{nom}_1), \dots, (D_n, \text{nom}_n)$ sont des domaines, on appelle schéma de table (ou schéma de relation) le produit cartésien $D_1 \times \dots \times D_n$ où chaque D_i représente une colonne (ou attribut) représentée par le nom nom_i .

* Une table (ou relation) est un sous-ensemble du produit cartésien d'un schéma de table.

* Un enregistrement (ou ligne, n-uplet, entrée) est un élément d'une table.

Remarque 15 : Quand on a un ensemble de tables, on veut qu'elles puissent respecter des règles.

Exemple 16 : On veut que les clients mentionnés dans les commandes existent.

Def 17 : (Clef, clefs minimales, primaire, étrangère)

* Une clef d'une table est un ensemble de colonnes tq il n'existe pas deux enregistrements ayant les mêmes valeurs sur ces colonnes.

* Une clef est dite minimale si elle perd sa propriété lorsqu'on lui enlève un attribut.

* Chaque table doit avoir une clef primaire, qui est une clef minimale désignée unique.

* Une clef étrangère est un ensemble de colonnes d'une table t_1 , correspondant à un ensemble de colonnes de t_2 , et tq

pour tout enregistrement de t_1 , le p-uplet correspondant existe dans t_2 , et les colonnes associées font partie de la clef primaire de t_2 .

Def 18 : Une contrainte de domaine impose à chaque enregistrement d'une table de vérifier une assertion logique.

Représentation 19 :

On écrit sous la forme $\text{nom}_{\text{table}} (\text{attribut}_1, \dots)$ en soulignant les attributs issus d'une clef primaire et en mettant # devant les clés étrangères.

Exemple 20 :

client (numéro, nom, #code-postal, pays), adresse :

ville (code-postal, pays, nom)

II.2 - De l'entité association au modèle relationnel

Algo 21 :

* Les entités deviennent des relations

* Pour les associations 1..1 0..* :

on crée une contrainte clef étrangère dans la première relation vers la clé primaire de la deuxième.

* Pour les associations 0..* 0..* :

On crée une troisième relation contenant les clés primaires des deux tables, chacune étant des clés étrangères.

* Pour les associations 1..* 1..* :

On fait pareil, et on peut rajouter que les clés primaires des relations initiales sont des clés étrangères.

etc...

Exemple 22 : Clients (numeros, nom, #(code-postal, pays), adresse)
 Villes (code-postal, pays, nom)
 Produits (numeros, nom, prix, poids)
 Commande (#numeros, #numeros, quantité)

Développement 1 : Modélisation par une base de données relationnelle d'élèves inscrits à l'université.

III - Implémentation

III.1) SGBD (livre Tle)

Def 23 : Un SGBD (Système de Gestion de Bases de Données) est un système implémentant les bases de données, permettant d'effectuer des opérations dessus (création, insertion, modification, sélection...) tout en masquant la complexité des opérations.

Exemple 24 : Exemples de SGBD relationnel : PostgreSQL, duckDB, MySQL, Oracle

Remarque 25 Il existe des SGBD qui ne fonctionnent pas en relationnel (IMS).

Def 26 : Une transaction est un ensemble d'instructions qui doivent toutes être exécutées ou toutes annulées.

Remarque 27 : On veut cela pour que les données restent cohérentes.

Exemple 28 : On veut supprimer des produits et donc toutes les commandes attenantes. Si on commence par supprimer un produit, puis qu'on s'interrompt, la base

sera incohérente.

Propriété 29 : Un bon SGBD doit avoir la propriété ACID :

- Atomicité : Une transaction est "tout ou rien"
- Cohérence : les transactions doivent faire passer la base d'un état cohérent à un état cohérent. Les contraintes du modèle relationnel doivent être respectées (clé, clé étrangère, domaine...)

- Isolation : Si deux transactions s'exécutent en même temps, l'effet doit être le même que si l'exécution avait été séquentielle. En particulier, on ne doit pouvoir observer l'état intermédiaire d'une autre transaction.

- Durabilité : Une transaction validée l'est définitivement : un problème (panne, coupure de courant...) ne doit pas mettre à mal la pérennité de la mise à jour.

III.2) SQL

Pour communiquer avec un SGBD, on utilise le langage de programmation SQL (Structured Query Language).

Remarque 30 : Ce langage ne dépend pas de l'implémentation sous-jacente : on ne donne que le résultat que l'on souhaite, pas la manière de l'obtenir. C'est donc un langage déclaratif, dans le paradigme logique.

Remarque 31 : Grâce à SQL, les BDD sont très expressives et donnent de multiples possibilités.

TD 32 : Trouver la plus de manière possible de chercher le max et de faire la division.

Développement 2 : Correction du TD 32.