

Base de données

Chapitre 3 : Introduction aux bases de données relationnelles

Joel Cavat

2022

- Objectifs
 - Introduction aux BD relationnelles

Première réalisation commerciale d'une bd rel. dans les années 1980

BD rel. populaires

- Open Source
 - MySQL
 - PostgreSQL
 - MariaDB
 - SQLite
 - Firebird
- Commerciales
 - Oracle Database
 - DB2 d'IBM
 - Microsoft SQL Server
 - Microsoft Access

Les bases de données relationnelles

Base de données relationnelle

Une base de données relationnelle permet de regrouper des données structurées dans un ensemble de **tables** organisées en **lignes** et **colonnes**. Elle permet de renforcer les contraintes d'intégrité.

Remarques

- Le terme de **relation** ne décrit pas les association entre les tables.
- Une relation **correspond** à une table avec ses enregistrements.
 - Le terme de **relation** provient du formalisme mathématique décrit par Codd dans sa formulation du modèle relationnel.
- Dès lors, on préférera le terme d'**association** (*relationship*) pour représenter les liens entre les tables.
- Une table était représentée par type d'entité ou un type d'association du modèle EA.

Caractéristiques

- Structurée
- **Normalisée**
- Centralisée - scalabilité verticale
- Fortement cohérent
 - Modélisation des CI
 - Transactions (propriétés ACID)
 - Triggers (renforcement des CI)

Structurée

- Les données sont représentées sous forme de lignes et de colonnes (tables)
- Les valeurs des attributs sont atomiques (monovalués)
 - La valeur d'un attribut ne peut pas être une liste
- Tous les enregistrements d'une table contiennent les mêmes attributs, la même structure
 - Un attribut peut être facultatif (valeur `null`)

Terminologie de base et représentation

- Une **table** est un ensemble de données organisées sous forme de lignes et de colonnes.
- Une **ligne** correspond à un enregistrement, à une entité ou à un tuple.
- Une **colonne** correspond à une propriété d'une entité. Elle possède un type.
- Une **base de données relationnelle** est un ensemble de tables et de contraintes d'intégrité.

The diagram illustrates a database table named "Student". The table has four columns: **student_id**, **firstname**, **lastname**, and **university_id**. The first column, **student_id**, is highlighted with a red box and labeled as the "Clé primaire" (Primary Key). The other three columns are labeled as "Attributs" (Attributes). The table contains four rows of data, each labeled as a "Ligne (Tuple)". The columns are also labeled as "Colonne/Attributs". The table is labeled as "Table (Relation) 'Student'" and "Nom de la table".

student_id	firstname	lastname	university_id
23	Aurélia	Racloz	1
18	Stéphane	Ischi	4
16	Naïm	Daiz	4
29	Jérôme	Doutaz	2

Terminologie de base et représentation (suite)

- Chaque enregistrement possède un identifiant appelé **clé primaire**. Une telle clé doit être unique.
- Une **clé étrangère** est une colonne utilisée pour référencer une clé (candidate) d'une entité d'une autre table.

The diagram shows a table named 'Student' with four columns: 'student_id', 'firstname', 'lastname', and 'university_id'. The 'student_id' column is highlighted with a red box and labeled 'Clé primaire'. The 'university_id' column is also highlighted with a red box and labeled 'Clé étrangère'. The table contains four rows of data. Annotations with red arrows point to various parts of the table: 'Table (Relation) "Student"' points to the table title, 'Nom de la table' points to the 'Student' title, 'Attributs' points to the column headers, 'Clé étrangère' points to the 'university_id' column, 'Ligne (Tuple)' points to the rows of data, and 'Colonne/Attributs' points to the individual columns.

student_id	firstname	lastname	university_id
23	Aurélia	Racloz	1
18	Stéphane	Ischi	4
16	Naïm	Daiz	4
29	Jérôme	Doutaz	2

Theme

<u>id_theme</u>	theme
1	Architecture
2	Informatique
3	Mathématique

Book

<u>isbn</u>	title	id_theme
2-X-4	Les Acueducs	1
5-X-1	Python pour les nuls	2
7-X-3	Base de données avancée	2
9-X-4	Algèbre et géométrie	3

Figure 1: Exemple de deux tables avec clés étrangères

Définition (simpliste)

Une base de données **normalisée** est modélisée de manière à éviter des redondances, des incohérences et des anomalies

La normalisation repose sur un ensemble de règles à respecter : les **formes normales** (sujet d'un prochain chapitre).

Motivations

Grâce à la normalisation, le modèle relationnel permet d'éviter:

- les redondances
- anomalies de mise-à-jour
- anomalies d'insertions

Redondances et anomalies

- Prenons un modèle simplifié de l'emprunt de livres à la bibliothèque par des étudiants de l'école.
- Utilisons un simple tableur pour enregistrer toutes les informations (emprunteur, livre, emprunt et thème du livre)

NoEtu	Prénom	Nom	ISBN	Titre	Echéance	Thème
1	Bill	Torvalds	2-234432-54-3	Les Aqueducs	18.06.1999	Architecture
2	Linus	Gates	2-234432-54-3	Les Aqueducs	23.04.2012	Architecture
2	Linus	Gates	1-657879-34-6	Python pour les nuls	19.07.2013	Informatique
3	Mark	Jobs	4-493828-43-2	Base de données avancée	01.05.2011	Informatique
4	Steve	Zuckerberg	8-098384-98-7	Algèbre et géométrie	05.12.2005	Mathématique
4	Steve	Zuckerberg	4-493828-43-2	Base de données avancée	06.08.2013	Informatique

Redondances et anomalies

NoEtu	Prénom	Nom	ISBN	Titre	Echéance	Thème
1	Bill	Torvalds	2-234432-54-3	Les Aqueducs	18.06.1999	Architecture
2	Linus	Gates	2-234432-54-3	Les Aqueducs	23.04.2012	Architecture
2	Linus	Gates	1-657879-34-6	Python pour les nuls	19.07.2013	Informatique
3	Mark	Jobs	4-493828-43-2	Base de données avancée	01.05.2011	Informatique
4	Steve	Zuckerberg	8-098384-98-7	Algèbre et géométrie	05.12.2005	Mathématique
4	Steve	Zuckerberg	4-493828-43-2	Base de données avancée	06.08.2013	Informatique

Problèmes

- Espace de stockage non optimal
- Problème de cohérence des données : Les anomalies

Redondances \Rightarrow risque d'anomalies

Anomalie de mise-à-jour

- Problèmes
 - Comment classer un livre dans un autre thème ?
 - Comment corriger le nom d'un étudiant ?
- Solution naïve
 - Trouver toutes les occurrences et les modifier
- Risque
 - Oubli d'une modification a pour conséquence des données incohérentes

NoEtu	Prénom	Nom	ISBN	Titre	Echéance	Thème
1	Bill	Torvalds	2-234432-54-3	Les Aqueducs	18.06.1999	Architecture
2	Linus	Gates	2-234432-54-3	Les Aqueducs	23.04.2012	Architecture
2	Linus	Gates	1-657879-34-6	Python pour les nuls	19.07.2013	Informatique
3	Mark	Jobs	4-493828-43-2	Base de données avancée	01.05.2011	Informatique
4	Steve	Zuckerberg	8-098384-98-7	Algèbre et géométrie	05.12.2005	Mathématique
4	Steve	Zuckerberg	4-493828-43-2	Base de données avancée	06.08.2013	Informatique

Anomalie d'insertions

- Problèmes
 - On ne peut inscrire un étudiant dans la base de données que s'il a emprunté un livre au moins une fois
 - On ne peut inscrire un livre dans la base de données que s'il a été emprunté au moins une fois
- Solution naïve
 - Laisser des champs vides

NoEtu	Prénom	Nom	ISBN	Titre	Echéance	Thème
1	Bill	Torvalds	2-234432-54-3	Les Aqueducs	18.06.1999	Architecture
2	Linus	Gates	2-234432-54-3	Les Aqueducs	23.04.2012	Architecture
2	Linus	Gates	1-657879-34-6	Python pour les nuls	19.07.2013	Informatique
3	Mark	Jobs	4-493828-43-2	Base de données avancée	01.05.2011	Informatique
4	Steve	Zuckerberg	8-098384-98-7	Algèbre et géométrie	05.12.2005	Mathématique
4	Steve	Zuckerberg	4-493828-43-2	Base de données avancée	06.08.2013	Informatique
5	Georges	Codd				

Solutions proposées par le modèle relationnel

- Séparer les entités (Séparer les concepts)

NoEtu	Prénom	Nom	ISBN	Titre	Echéance	Thème
1	Bill	Torvalds	2-234432-54-3	Les Aqueducs	18.06.1999	Architecture
2	Linus	Gates	2-234432-54-3	Les Aqueducs	23.04.2012	Architecture
2	Linus	Gates	1-657879-34-6	Python pour les nuls	19.07.2013	Informatique
3	Mark	Jobs	4-493828-43-2	Base de données avancée	01.05.2011	Informatique
4	Steve	Zuckerberg	8-098384-98-7	Algèbre et géométrie	05.12.2005	Mathématique
4	Steve	Zuckerberg	4-493828-43-2	Base de données avancée	06.08.2013	Informatique



NoEtu	Prénom	Nom	ISBN	Titre	Echéance	Thème
1	Bill	Torvalds	2-234432-54-3	Les Aqueducs	18.06.1999	Architecture
2	Linus	Gates	2-234432-54-3	Les Aqueducs	23.04.2012	Architecture
2	Linus	Gates	1-657879-34-6	Python pour les nuls	19.07.2013	Informatique
3	Mark	Jobs	4-493828-43-2	Base de données avancée	01.05.2011	Informatique
4	Steve	Zuckerberg	8-098384-98-7	Algèbre et géométrie	05.12.2005	Mathématique
4	Steve	Zuckerberg	4-493828-43-2	Base de données avancée	06.08.2013	Informatique

Solutions proposées par le modèle relationnel

- **Séparer les entités** : Attention à la perte des associations

NoEtu	Prénom	Nom	ISBN	Titre	Echéance	Thème
1	Bill	Torvalds	2-234432-54-3	Les Aqueducs	18.06.1999	Architecture
2	Linus	Gates	2-234432-54-3	Les Aqueducs	23.04.2012	Architecture
2	Linus	Gates	1-657879-34-6	Python pour les nuls	19.07.2013	Informatique
3	Mark	Jobs	4-493828-43-2	Base de données avancée	01.05.2011	Informatique
4	Steve	Zuckerberg	8-098384-98-7	Algèbre et géométrie	05.12.2005	Mathématique
4	Steve	Zuckerberg	4-493828-43-2	Base de données avancée	06.08.2013	Informatique



NoEtu	Prénom	Nom
1	Bill	Torvalds
2	Linus	Gates
3	Mark	Jobs
4	Steve	Zuckerberg

ISBN	Titre
2-234432-54-3	Les Aqueducs
1-657879-34-6	Python pour les nuls
4-493828-43-2	Base de données avancée
8-098384-98-7	Algèbre et géométrie

NoThm	Thème
1	Architecture
2	Informatique
3	Mathématique

Echéance
18.06.1999
23.04.2012
19.07.2013
01.05.2011
05.12.2005
06.08.2013

Redondances et anomalies

Solutions proposées par le modèle relationnel

- Séparer les entités
- Représenter les associations en utilisant des attributs communs à chaque table associée.

NoEtu	Prénom	Nom	ISBN	Titre	Echéance	Thème
1	Bill	Torvalds	2-234432-54-3	Les Aqueduc	18.06.1999	Architecture
2	Linus	Gates	2-234432-54-3	Les Aqueduc	23.04.2012	Architecture
2	Linus	Gates	1-657879-34-6	Python pour les nuls	19.07.2013	Informatique
3	Mark	Jobs	4-493828-43-2	Base de données avancée	01.05.2011	Informatique
4	Steve	Zuckerberg	8-098384-98-7	Algèbre et géométrie	05.12.2005	Mathématique
4	Steve	Zuckerberg	4-493828-43-2	Base de données avancée	06.08.2013	Informatique



NoEtu	Prénom	Nom
1	Bill	Torvalds
2	Linus	Gates
3	Mark	Jobs
4	Steve	Zuckerberg

ISBN	Titre	NoThm
2-234432-54-3	Les Aqueduc	1
1-657879-34-6	Python pour les nuls	2
4-493828-43-2	Base de données avancée	2
8-098384-98-7	Algèbre et géométrie	3

NoThm	Thème
1	Architecture
2	Informatique
3	Mathématique

Echéance	NoEtu	ISBN
18.06.1999	1	2-234432-54-3
23.04.2012	2	2-234432-54-3
19.07.2013	2	1-657879-34-6
01.05.2011	3	4-493828-43-2
05.12.2005	4	2-234432-54-3
06.08.2013	4	4-493828-43-2

Centralisée - scalabilité verticale

- En cas de montée en charge -> migration vers un serveur plus puissant
- (Possible de la décentraliser, mais difficile et mal adapté)

Contraintes d'intégrité

Les contraintes d'intégrité décrivent des règles métiers à respecter. Elles décrivent des situations (ou comportements) admissibles ou non admissibles pour le domaine d'application.

Elles permettent de renforcer la cohérence des données.

Il existe trois moyens de renforcer la cohérence des données

- La modélisation logique
- Les déclencheurs (*triggers*)
- Les transactions

CI statiques

- décrivent des états valides
- modélisables
 1. une personne doit avoir un nom de famille
 2. une conférence doit avoir un thème
- non modélisables
 3. le prix d'une conférence doit être supérieur à 0
 4. les intervenants pour les conférences "blockchain" logent dans des hôtels quatre ou cinq étoiles
 5. les thèmes sont obligatoirement associés à une conférence
 6. les conférences n'ont lieu que dans des capitales européennes

CI dynamiques

- décrivent des changements d'états valides
- non modélisables
 7. pour un intervenant, il est possible de changer l'attribution d'un hôtel vers un autre de même classe ou de classe supérieure
 8. il est impossible d'augmenter le prix d'une conférence de plus de 10%
 9. l'état civil d'une personne peut passer de "célibataire" à "marié" mais pas aux statuts "veuf" ou "divorcé"

Classement des contraintes d'intégrité

	Dynamique	Statique
Modélisable	-	1, 2
Non modélisable	7, 8, 9	3, 4, 5, 6

Toutes les contraintes non modélisables **doivent être décrites** en annexe du modèle

Types de contraintes d'intégrité

- **existence**

- p. ex.: un livre est classé dans une catégorie existante, une personne a obligatoirement un nom

- **unicité**

- la valeur d'un attribut (ou d'un ensemble d'attributs) doit être unique
- **unicité clé**
- chaque entité doit avoir un identifiant unique

- **domaine**

- ensemble des valeurs possibles / restreindre
- clause CHECK/WITH CHECK OPTION ou colonne générée (c.f. chapitre SQL)
- p. ex.: le numéro d'un employé est un nombre à six chiffres, le degré d'alcool est compris entre 0 et 100

Types de contraintes d'intégrité (suite)

- **dépendances fonctionnelles**

- les valeurs d'un groupe d'attributs permet de déterminer un autre groupe d'attributs
- p: ex.: $no_avs \rightarrow nom, prenom,$
 $no_commande, no_article \rightarrow quantité$

- **inclusion/référentielle**

- est une contrainte d'existence qui garanti que les valeurs d'un groupe d'attributs d'une table doivent exister dans une autre table
- une clé étrangère doit contenir une valeur existante comme clé primaire d'une autre table
- p. ex.: il est possible d'acheter un article auprès d'un fournisseur uniquement si celui-ci livre l'article en question

Transaction

Une transaction est une suite d'opérations réalisées de manière indivisible (c.f. cours de concurrence).

Propriété d'une transaction (acronyme ACID):

- Atomicité
- Cohérence
- Isolation
- Durabilité

Problématique d'un accès concurrent (sans transaction)

Démonstration en classe

Caractéristiques ACID d'une transaction

- **A**tomicité
 - suite d'opérations indivisibles
 - l'opération est faite dans son intégralité ou elle est annulée (commit/rollback)
 - transaction indivisible
- **C**ohérence
 - la base de données est toujours dans un état cohérent
 - une transaction fait passer d'un état valide à un autre état valide

Caractéristiques ACID

Ce sont les quatre propriétés d'une transaction

- **I**solation
 - chaque transaction est indépendante
 - une transaction n'a pas accès aux données d'une autre transaction en cours
 - une transaction verrouille les ressources
- **D**urabilité
 - une fois qu'une transaction est terminée, elle est persistée même en cas de crash du système
 - stockage sur disque non volatile
 - résultat d'une transaction persistée à long terme

Les déclencheurs (triggers)

Trigger

Un *trigger* est une vérification qui est activée lors d'un événement particulier (insertion, mise-à-jour...). Ils permettent de vérifier des contraintes non modélisables.

LORS DE <événement>

SI <condition>

ALORS <action/exception>