

Les bases de données relationnelles

Modèle relationnel

2


- Proposé en 1970, par E. F. Codd,
- Modèle fondé sur la notion mathématique de *relation*.
- Représentation simple et très puissante (tableaux).
- Une BD relationnelle est formée d'un ensemble de *tables*.
- Structures de données: *relations* et *tables*.
- Propose des méthodes de manipulation des données par l'*algèbre relationnelle*.

Base de données relationnelle

3

- Type de base de données qui utilise des **tables** pour le stockage d'informations.
- **Utilise des valeurs issues de deux tables** pour associer les données d'une table aux données d'une autre table.
- En règle générale, dans une BD relationnelle, les informations ne sont stockées qu'une seule fois.

Numéro	Prénom	Nom	Email
1	Jean	Dupont	jdupont@email.com
2	Marie	Malherbe	mama@email.com
3	Nicolas	Jacques	jacques.nicolas@email.com
4	Hadrien	Piroux	happie@mail.com



Numéro	Client	Produit	Quantité
1	3	Tube de colle	3
2	2	Rame de papier A4	6
3	2	Ciseaux	2

Schéma d'une BD relationnelle

Structure de données relationnelles

4

- **Table** : constituée de **lignes** (enregistrements) et de **colonnes** (champs).
 - **Domaine** : l'ensemble des valeurs de la colonne
- Exemple : table *Étudiants*

→ ÉTUDIANTS

no étudiant	nom étudiant	prénom étudiant	programme
122334	Legendre	Albert	420
123456	Grand	Justin	420
9876554	Durand	Simone	420
9988776	Berlier	Caroline	420

Table

Colonne

Ligne

Structure de données relationnelles

5

Table : Constituée de lignes et de colonnes

Voici les colonnes.

prenom	nom	adresse	ville	code_postal	num_id
Auguste	Préau	donnée	donnée	donnée	donnée
Albert	Durand	donnée	donnée	donnée	donnée
Marie	Herbier	donnée	donnée	donnée	donnée
Louis	Blanc	donnée	donnée	donnée	donnée

Voici les lignes.

Structure de données relationnelles

6

BD : Constituée de toutes les tables reliées entre elles

Voici une base de données contenant trois tables. La base de données s'appelle "mes_encas".

Les noms de bases de données et de tables sont normalement en minuscules.

mes_encas

crepes_confiture

boutique	heure	date	note	commentaires
Café Starbuzz	7h43	23/4	9	presque parfaite
La crêperie du port	8h56	25/4	5	grasse
Le roi de la crêpe	21h39	26/4	6	pas très chaude mais savoureuse

Table contenant les informations sur les crêpes à la confiture.

crepes_sucree

boutique	heure	date	note	commentaires
Le roi de la crêpe	21h39	26/4	8	tède mais pas chaud
Café Starbuzz	7h43	23/4	4	pas assez de sucre
La crêperie du port	8h56	25/4	6	glace
La crêperie du port	22h35	24/4	7	grasse
				pas frais

Table contenant les informations sur les crêpes au sucre.

autres_encas

boutique	heure	date	gâteau	note	commentaires
Café Starbuzz	22h35	24/4	pain d'épice	6	trop épicé
Café Starbuzz	7h43	23/4	éclair	8	chocolat !
Le roi de la crêpe	21h39	26/4	tartlette	4	manque de fruits
La crêperie du port	8h56	25/4	croissant	9	dur, croustillant

Table contenant les informations sur les encas autres que les crêpes.

Clés

7

- Clé : chaque table possède une *clé primaire*.
- Une clé primaire sert à identifier de façon unique chaque ligne d'une table.
- Une clé primaire composée de plus d'une colonne s'appelle une *clé primaire composée*

Exemple :

ailé	no_porte
G	8
G	10
T	16

- Clé étrangère : fait le lien entre les tables.

Exemple BD : employés-départements

8

Représentation visuelle :

EMPLOYÉS		
<u>nas employé</u>	<u>nom employé</u>	<u>no département</u>
111-222-333	Roger-Dupuis	3
444-555-666	Marie-Dubois	1
777-888-999	Justin-Maurois	1

DÉPARTEMENTS	
<u>no département</u>	<u>nom département</u>
1	Ressources-humaines
2	Ventes
3	Stocks

Représentation textuelle :

Clé primaire

Clé étrangère

- EMPLOYÉ (nas employé, nom_employé, no_département)
- DÉPARTEMENT (no département, nom_département)

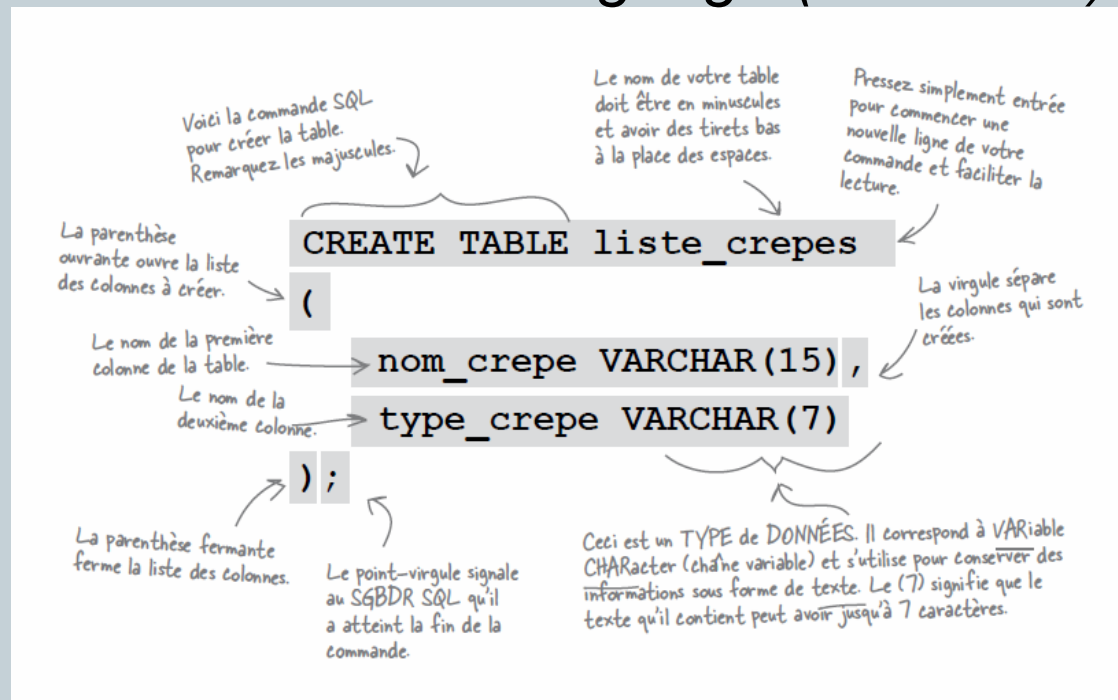
SQL - Structured Query Language

9

- Le langage SQL permet de dialoguer avec la BD.

Exemple : Création d'une table avec SQL

Data Definition Language (SQL DDL)



Exemple : employés-départements (SQL DDL)

10

- ```
CREATE TABLE employé (
 nas_employé NUMBER(4) PRIMARY KEY,
 nom_employé CHAR(20),
);
```

# Exemple : employés-départements (SQL DDL)

11

- ```
CREATE TABLE employé (  
    nas_employé      NUMBER(4) PRIMARY KEY,  
    nom_employé      CHAR(20),  
    no_département   NUMBER(2),  
    CONSTRAINT employé_no_département_fk  
        FOREIGN KEY (no_département)  
        REFERENCES département(no_département)  
);
```

Propriétés des tables

12

- Chaque table doit avoir un nom distinct.
- Chaque champ (cellule) doit contenir une seule valeur (par opposition à une liste de valeurs).
- Chaque colonne doit avoir un nom distinct.
- Toutes les valeurs d'une colonne font partie du même domaine.
- L'ordre des colonnes n'a pas de signification.
Par convention, la première colonne représente la clé primaire de la table.
- Toutes les lignes d'une table sont distinctes.
- L'ordre des lignes n'a pas de signification.

Algèbre relationnelle

13

- Opérateurs ensemblistes qui permettent de manipuler le contenu des tables.
- Opérateurs de relation.
- Opérateurs unaire
sélection : `SELECT`
projection : `WHERE`
- Opérateurs binaires (produit cartésien, jointure).

Contraintes d'intégrité

14

- Une **contrainte d'intégrité** est une condition définie sur le schéma de données qui doit être vérifiée à chaque fois qu'il y a mise à jour de la BD.

Exemple :

Intégrité référentielle

no_département	nom_département
1	Ressources humaine
2	Ventes
3	Stocks

nas_employé	nom_employé	no_département
111-222-333	Roger Dupuis	3
444-555-666	Marie Dubois	4
777-888-999	Justin Maurois	1
123-456-7890	PMC	a

Intégrité sur le format

Intégrité sur le domaine

Contraintes d'intégrité

15

- Nul
- **Intégrité référentielle** : vérifie que la valeur prise par une clé étrangère existe dans la table où la clé étrangère est une clé primaire.
- Contraintes générales :
 - Nombre positif ou dans un intervalle
 - Liste de valeurs
 - Domaine, format, etc.

Normalisation

16

- Un bon schéma de données doit :
 - Respecter les **règles** du modèle relationnel;
 - Être **NON redondant**;
Exemples : salaire_horaire
 nb_heures_travaillées
 salaire_total_semaine
 - Respecter les **exigences** de l'entreprise.

Normalisation

17

- Processus formel qui permet de transformer le schéma brut en schéma relationnel non redondant.
- Le cadre proposé pour cette démarche repose sur le concept de *dépendance fonctionnelle* et fait passer le schéma par différentes étapes de normalisation :
 - 1FN, 2FN, 3FN

Dépendance fonctionnelle

18

- Lorsque la valeur prise par une colonne (ou groupe de colonnes) détermine de manière unique la valeur d'une autre colonne (ou groupe de colonnes) de la même table, on dit qu'il existe une *dépendance fonctionnelle* entre la (les) première colonne (ou groupe de colonnes) et la (les) seconde.

Exemples :

no_étudiant	→ nom_étudiant, prénom_étudiant	correct
no_cours	→ no_local	?
no_cours	→ no_unique_employé_professeur	?
no_local	→ no_cours	?

Formes normales - 1FN

19

- Une table est en *première forme normale* lorsque chaque colonne de la table ne prend que des valeurs atomiques.

Exemple :

ÉTUDIANT (no_étudiant, nom_étudiant, prénom_étudiant, programme)

En forme normale 1 (1FN) :

ÉTUDIANT (no_étudiant, nom_étudiant, prénom_étudiant)

INSCRIPTION (no_étudiant, no_programme, date_admission)

PROGRAMME (no_programme, nom_programme)

Formes normales – 2FN

20

- Une table est en *deuxième forme normale* :
 - Si elle est en 1FN.
 - Si toutes les colonnes non clés sont complètement dépendantes de la clé primaire.

Par *dépendance fonctionnelle complète*, on entend qu'il n'y a pas de dépendance fonctionnelle entre une partie de sa clé primaire et une de ses colonnes non clés.

Formes normales – 2FN

21

- La 2FN se teste que dans les cas de tables ayant une clé primaire composée (plus d'une colonne).
- Dans le cas de clé primaire simple, une table en première forme normale est automatiquement en deuxième forme normale.

Formes normales - 2FN

22

Exemple :

INSCRIPTION_COURS (no_cours, no_étudiant, date_inscription,
nom_département)

En forme normale 2 (2FN) :

INSCRIPTION_COURS (no_cours, no_étudiant, date_inscription)

COURS (no_cours, nom_cours)

DÉPARTEMENT (no_département, nom_département)

Formes normales – 3FN

23

- Une table est en troisième forme normale :
 - Si elle est en 2FN.
 - Et s'il n'y a pas de dépendances fonctionnelles transitives, c'est-à-dire de dépendances fonctionnelles entre ses colonnes non clés.
- Se teste que sur des colonnes non clés.

Exemple :

```
SESSION (code_session, date_début_session,  
         date_fin_session, durée_session) incorrect !
```

Avantages de la normalisation

24

- Permet de vérifier que le schéma logique est relationnel et non redondant.
- Un schéma en 3FN possède :
 - de «bonnes» clés primaires
 - Des tables décomposées en structures stables
- Permet d'éviter les problèmes d'anomalies lors des mises à jour de la BD.

Source pour les images

25

- Beighley L, *SQL Tête la première - Un guide d'apprentissage*, 2009, 613 pages.
 - Diapositive 16 : page 12
 - Diapositive 17 : page 13
 - Diapositive 20 : page 19