

Cours de base de données

Par: Kamal BAL
 Université AMOB de Bouira
 Faculté des sciences et des sciences appliquées
 Département d'informatique

Kamal BAL - Cours SGBD - Université Akli Mohand Oulhadj de Bouira - Février 2015

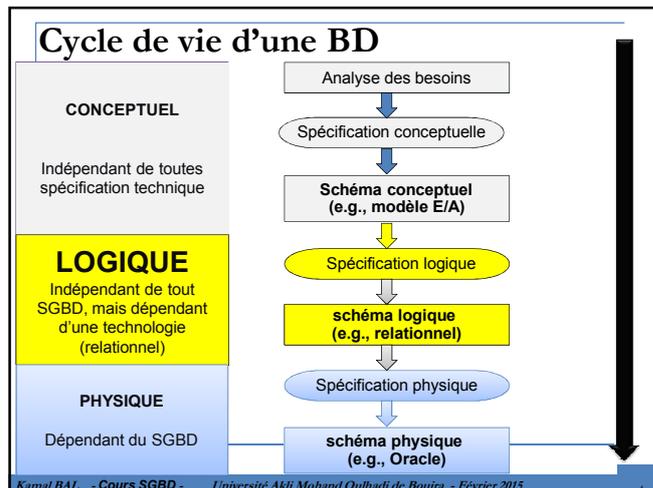
Contenu du cours

- Introduction aux bases de données
- Conception - le modèle entité association
- **Le modèle relationnel**
- L'algèbre relationnelle
- Dépendances fonctionnelles et normalisation
- Les contraintes d'intégrité
- Accès concurrents et gestion des transactions

Kamal BAL - Cours SGBD - Université Akli Mohand Oulhadj de Bouira - Février 2015

Le modèle relationnel

Kamal BAL - Cours SGBD - Université Akli Mohand Oulhadj de Bouira - Février 2015



Le modèle relationnel

Historique

- Travaux de Edgar F. Codd 1970
- Codd critiqua les modèles existant
 - Ils mélangeaient la description abstraite de la structure de l'information et les descriptions des mécanismes physiques d'accès.
- Modèle logique de données formalisé par F. CODD en 1970.

Kamal BAL - Cours SGBD - Université Akli Mohand Oulhadj de Bouira - Février 2015

Le modèle relationnel

- **Idée initiale** : Utiliser un modèle **ensembliste** pour décrire et manipuler un ensemble d'enregistrements.

Kamal BAL - Cours SGBD - Université Akli Mohand Oulhadj de Bouira - Février 2015

Objectifs :

- ❑ **Indépendance** entre programmes d'application et représentation interne des données.
- ❑ **Base théorique solide** pour la cohérence et la non redondance des données.
 - Théorie des ensembles
 - La logique mathématique(théorie des prédicats de premier ordre).

Les 12 règles de Codd

1. L'information est représentée de façon **logique** sous forme de **tables**.
2. Les données doivent être accessibles de façon **logique** par les **tables**, les **clés** primaires et les **colonnes**.
3. Les **valeurs nulles** doivent être traitées uniformément comme des "informations absentes", et non pas comme des chaînes vides, ni des blancs, ni des zéros.
4. Les **métadonnées** doivent être stockées dans la base au même titre que les données normales.
5. Un **langage unique** doit permettre de définir les données, les vues sur ces données, les contraintes d'intégrité, les autorisations d'accès, les transactions, et enfin, la manipulation des données.

Les 12 règles de Codd

6. Les **vues** doivent refléter les mises à jour de leurs tables de base, et vice-versa.
7. Chaque action suivante doit pouvoir être réalisée par une et une seule action : **Retrouver**, **Insérer**, **Modifier** et **Supprimer** une donnée.
8. Il doit y avoir séparation logique entre les opérations et le stockage physique des données et **leurs méthodes d'accès**.
9. Les opérations peuvent modifier le schéma de la base de données sans qu'elle n'ait à être recrée, et sans que les applications construites au dessus d'elle n'aient à être réécrites.
10. Les contraintes d'intégrité doivent être disponibles, et stockées dans les métadonnées et non pas dans un quelconque programme d'application.

Les 12 règles de Codd

11. Le langage de manipulation des données (LMD) ne doit pas se soucier d'où ni du comment les données sont stockées et/ou distribuées.
12. Le traitement d'une ligne doit respecter les mêmes règles et contraintes d'intégrité que les opérations portant sur des ensembles.

Concepts de base du modèle relationnel

Théorie relationnelle est fondée sur théorie des ensembles.

■ Briques principales :

- ❑ **Domaine**,
- ❑ **Attribut**,
- ❑ **Relation**

■ Autres briques

- ❑ **N_tuplet**
- ❑ **Schéma d'une relation**
- ❑ **Schéma d'une base de données**

Concepts de base du modèle relationnel

■ Domaine

Domaine :

Un domaine est un ensemble de valeurs caractérisé par un **nom**. C'est un ensemble **nommé** dans lequel les données peuvent prendre leur valeur

Concepts de base du modèle relationnel

■ Domaine :

- **Domaine défini en intention** : via les **propriétés** respectées par ses valeurs (éléments).
 - Ex.: **N_plus** = entiers positifs
Age = entiers compris entre 0 et 150
- **Domaine défini en extension** : via l'ensemble de toutes les valeurs qui le composent.
 - Ex.: **Domaine (devise)** = {Dollar, Euro, Livre_sterling, Yen}
Domaine (couleurs) = {Bleu, Rouge, Blanc, ...}
Domaine (noms) = {Ali, Salah, Yasmine...}

Concepts de base du modèle relationnel

■ Relation :

■ Relation :

- Une relation est un **sous-ensemble du produit cartésien** d'une liste **finie** de **domaines**, caractérisé par un **nom**.

Structure véritablement centrale /
fondamentale du modèle relationnel.

Concepts de base du modèle relationnel

■ Relation :

■ Produit cartésien :

le produit cartésien d'un ensemble de domaines d_1, d_2, \dots, d_n est l'ensemble des vecteurs $\langle v_1, v_2, \dots, v_n \rangle$ ou $\forall i, v_i \in d_i$.

■ Exemple :

- Pays = {France, Italie, Japon} et Devise = {euro, yen}
- Pays x Devise = { (France, euro), (France, yen), (Italie, euro), (Italie, yen), (Japon, euro), (Japon, yen) }.

Concepts de base du modèle relationnel

■ Relation :

■ Relation :

- Une relation R est un **sous-ensemble du produit cartésien** d'une liste **finie** de domaines, caractérisé par un **nom**.

- $R \subset D_1 \times D_2 \times \dots \times D_n$

- D_1, D_2, \dots, D_n sont les domaines de R

- Une relation est donc un **ensemble de vecteurs**.

■ Exemple:

- $R = \text{Devise_Courante} = \{(France, euro), (Italie, euro), (Japon, yen)\}$.
- $R \subset \text{Devise} * \text{Courante}$

Concepts de base du modèle relationnel

■ Relation :

■ Exemple :

- Les domaines :
 - NOM = { Zaoui, Badaoui }
 - PRENOM = { Ali, Nora, Larbi }
 - DATE_NAISS = {Date entre 1/1/1990 et 31/12/2020}
 - NOM_SPORT = { judo, tennis, foot }
- La relation ELEVE
 - PERSONNE \subset NOM_ELV \times PREN_ELV \times DATE_NAISS
 - PERSONNE = {(Zaoui, Ali, 1/1/1992), (Badaoui, Nora, 2/2/1994) }
- La relation INSCRIPTION
 - INSCRIPTION \subset NOM_ELV \times NOM_SPORT
 - INSCRIPT = { (Zaoui, judo), (Zaoui, foot), (Badaoui, tennis) }

Concepts de base du modèle relationnel

■ Relation :

- Pour visualiser facilement le contenu d'une relation on utilise la représentation **tabulaire**.

- Chaque ligne correspond à un **vecteur**
- Chaque colonne correspond à un **domaine**.

■ Exemple : Devise_Courante \subset Pays x Devise

Pays	devise
France	Euro
Italie	Euro
Japan	Yen
Canada	Dollars Canadien

Concepts de base du modèle relationnel

■ Relation :

- une relation est un ensemble \Rightarrow on ne peut pas avoir 2 vecteurs (lignes) identiques
- En SQL on dit *table* à la place de relation.

Concepts de base du modèle relationnel

■ Relation :

- Degré d'une relation : c'est le nombre de colonnes (domaines) dans une relation .
 - $R \subset D1 \times D2 \times \dots \times Dn$
 - n est le degré de la relation R

Ex. Degré de PRODUIT=3

Relation PRODUIT		
NUM_PDT	DES_PDT	COUL_PDT
P1	D1	C1
P2	D2	C2

Domaine / Colonne

Concepts de base du modèle relationnel

■ Attribut

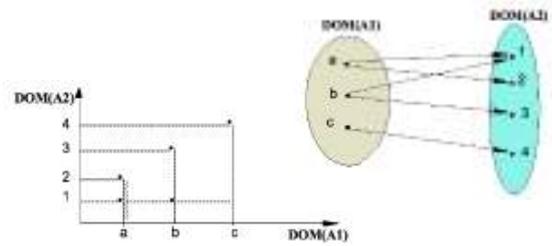
- Un attribut est une **colonne** caractérisée par un **nom unique** dans cette relation.

Ex. NUM_PDT, DES_PDT, COUL_PDT attributs de la relation PRODUIT.

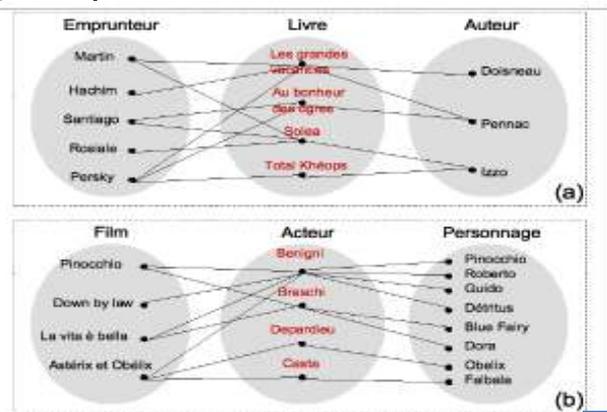
NUM_PDT	DES_PDT	COUL_PDT
P1	D1	C1
P2	D2	C2

Graphe d'une relation

- Relation binaire R(A1,A2)
- Une relation n-aire est une généralisation à n dimensions



Exemples



Pratique (10 Minutes)

- Donner d'après les relation (a) et (b) du slide précédent
 - les titres des livres empruntés par « Persky »
 - Les noms des emprunteurs du livre « Soleda »
 - Les auteurs lus par « Martin »
 - Les auteurs des livres empruntés par « Hachim » et « Persky »
 - Les noms des auteurs qui ont un seul livre
 - Les noms des auteurs qui ont 3 livres.
 - Les différents personnages interprétés par l'acteur « Benigni »

Schéma d'une relation

- Le schéma d'une relation est défini par :
 - le **nom** de la relation
 - la liste de ses **attributs**
- on note : **R (A1, A2, ... , An)**
 - **Ex.:**
 - **Eleve** (nom, prenom, naiss)
 - **Produit** (num_pdt, des_pdt, coul_pdt)
- Le schéma d'une base de données est défini par **l'ensemble des schémas des relations** qui la composent
 - le schéma de la BD dit comment les données sont **organisées** dans la base.

Les contraintes d'intégrité

Contraintes d'intégrité

- Une **Contrainte d'Intégrité (CI)** est une **règle** qui doit être **vérifiée** au moment de la création et de la manipulation de données afin que le résultat soit considéré **correct et cohérent**.
- A tout instant de l'existence d'une BD, on doit pouvoir ajouter, modifier ou supprimer une contrainte d'intégrité et le SGBD doit être capable de vérifier que la base est toujours cohérente vis à vis du changement apporté à son environnement; dans le cas contraire, il doit rejeter notre intervention.

Contraintes d'intégrité

- **Contraintes de structure (contraintes de clés)**
- **Contraintes de référence**
- **Contraintes d'unicité**
- **Contraintes de valeurs**

Contraintes d'intégrité

- **Contrainte de structure (contraintes de clés)**
 - Permettent de faciliter la gestion de la **cohérence** des données.
 - Ces règles sont énoncées par des **assertions** sur les relations et les attributs.
 - **Notion principale : Clef.**

Contrainte de clé

- Relation = ensemble.
 - Donc pas de **doublons** de tuples!
- A chaque tuple est associé **une clé unique**, définie au niveau de la relation.
- Une clé est un **ensemble minimal** d'attributs dont la connaissance des valeurs permet d'identifier un tuple de façon **unique** au sein de la relation considérée.
- Une clé n'est pas forcément un unique attribut.

Contrainte de structure (contraintes de clés)

- Une **clé** d'une relation est un attribut (ou groupe minimum d'attributs) qui détermine tous les autres
- Ex.: PRODUIT (no_prod, nom, prixUHT)
 - no_prod → nom
 - no_prod → prixUHT)
 - no_prod est une clé
- Une clé détermine un n-uplet de façon unique
- Une relation peut **posséder plusieurs** clés, on les appelle clés **candidate**s
- Ex.: dans la relation PRODUIT, nom est une clé **candidate**

Clés primaire

- Si **R** est une relation décrite par un ensemble d'attributs **A**, un sous-ensemble **K** est une clé de **R** si :
 - pour tout couple de tuples (t, t') de R, t(K)≠t'(K) où t(K) est la liste des valeurs données par t aux attributs composant K.
- On choisit une **clé** parmi les possibles (**candidate**s) pour identifier chaque tuple : **c'est la clé primaire**.
- Ce choix est guidé par la **sémantique** de la relation.
- Par convention, on représente la clé primaire en la **soulignant** dans l'énoncé de la relation.

Exemple

- **Acteur** (nom, prénom, numéro).
- **Casting** (film_acteur, personnage).
 - Le même personnage peut être joué dans plusieurs films.
- **Minimalité de la clé**: l'ensemble des attributs formant la clé primaire doit être minimal.

Pratique

- Soit la relation suivante
 - **Achat** (Numero_client, numero_produit, date_achat, quantité_achetée)
- Proposer une clé primaire à cette relation

Numero-clinet	Numero_produit	Date_achat	Quantité_achetée
C1	P1	D1	23
C1	P2	D1	56
C2	P7	D5	65
C1	P2	D4	5

Contrainte d'unicité

- La définition d'une clé primaire contient automatiquement une **contrainte d'unicité**.
- On peut aussi **spécifier** que la valeur d'un (groupe d'attributs) doit être **unique**, c'est-à-dire que deux tuples ne peuvent avoir la même valeur pour cet attribut.
- Exemple :
 - Film(numFilm, titre, réalisateur, année).
 - numFilm est la clé primaire mais on peut spécifier que le couple (titre, réalisateur) est unique.

Contrainte de référence (clé étrangère)

- Référencer un tuple : utiliser sa valeur de clé primaire.
 - ('Yahia', 'Ben Mabrouk', 5), tuple de la relation Acteur(Nom, Prénom, numéro), est **référencé** par le numéro 5:
 - Acteur[numéro=5] le désigne exactement.
- Une **clé étrangère** = attribut (s) dans une relation qui fait **référence** à une clé primaire d'une autre relation.
- La valeur des **attributs référentiels** dans le tuple référent est la valeur de la **clé primaire** dans le tuple référencé.
- CLÉ ÉTRANGÈRE = CLÉ PRIMAIRE dans une autre relation

Contrainte de référence (exemple)

- Pays(nom, devise, capitale, superficie)
- Devise(numéro_devise, nbpieces, nbillets, nom)
- Le Pays référence sa monnaie via la clé primaire de la relation Devise
 - les valeurs de l'attribut *devise* de la relation Pays sont les valeurs de l'attribut *numéro_devise* de la relation Devise.
- Ex.:
 - Devise(4, 8, 8, 'euro'),
 - Devise(1,9,8,'Dollars US')
 - Pays('France', 4, 'Paris', 544435)
 - Pays (USA, 1, Washington, 78898676)

Contrainte de référence (clé étrangère)

- Une contrainte référentielle exprime un lien **obligatoire** entre deux relations.
- Une clé étrangère (ex.: *devise*) est un groupe d'attributs dans une relation R (ex.: Pays) qui **doit correspondre** à la clé primaire d'une autre relation R' (ex.: Devise).
- Une relation possède une clé primaire, peut spécifier **plusieurs** clefs étrangères.
- Convention d'écriture : *italique*.

Contrainte de référence (clé étrangère)

- Un attribut peut être à la fois une clé primaire et étrangère :
 - Film(numFilm, titre, année),
 - Acteur(numActeur, nom, prénom),
 - Casting(numFilm, numActeur, personnage).

Casting.numacteur **reference** Acteur.numActeur

Casting.numFilm **reference** Film.numFilm

Exercice (15 minutes)

- Une relation peut s'auto-référencer via une clef étrangère.
- Ex.: Personne(numéro, nom, prénom, *père*, *mère*)
 - *Père* et *mère* sont des clés étrangères qui font référence à la clé primaire numéro dans la table **Personne**
- Retrouver l'arbre généalogique :
 - (12,'Martin','Ali', NULL, NULL),
 - (56, 'Castard', 'Joséphine', NULL, NULL),
 - (77, 'Frandler', 'Ernest', NULL, NULL),
 - (74, 'Bisteur', 'Annie', NULL, NULL),
 - (89, 'Martin', 'Eric', 12, 74),
 - (91, 'Frandler', 'Alain', 77, 56),
 - (94, 'Bisteur', 'Virginie', NULL, 74),
 - (103, 'Frandler', 'Mélanie', 91, 94).

Contraintes de valeurs d'attribut (CHECK)

- On ne connaît pas forcément les valeurs de tous les attributs lors de l'insertion d'un tuple.
- Convention : valeur NULL

Contrainte de non-vide (NOT NULL) : clé primaire; autres attributs

Restriction sur les valeurs d'un attribut (CHECK) : contrainte de domaine.

- Par extension : exemple : attribut saison IN {'hiver', 'printemps', 'été', 'automne'}
- Par restriction : durée > 0 (CHECK)
- Date_bac > date_nai

Contraintes référentielles et SGBD

- Importance des contraintes référentielles pour le maintiens de la cohérence lors des actions en écriture :
 - **Insertion d'un tuple** dans une relation avec clé étrangère : le SGBD doit vérifier que la valeur donnée par ce tuple à la clé étrangère correspond bien à une valeur de clé primaire.
 - Même vérification en cas de modification d'une valeur de clé étrangère dans un tuple existant
 - **Suppression d'un tuple référencé via une clé étrangère** par un ensemble d'autres tuples : plusieurs options :
 - supprimer "en cascade" tous les tuples référençant,
 - ou donner la valeur NULL à l'attribut de clé étrangère ("set null")
 - ou refuser la suppression ("restrict").

Rappel:

- Le Modèle relationnel
 - principes théoriques supportant le développement des SGBD
 - *utiliser un modèle ensembliste pour décrire et manipuler un ensemble d'enregistrements.*
 - Domaine,
 - attribut,
 - relation
 - Contraintes d'intégrité:
 - Clé primaire
 - Clé étrangère
 - Unicité
 - Sur les valeurs des attribut (check)

Kamal BAL - Cours SGBD - Université Akli Mohand Oulhadj de Bouira - Février 2015 43

Règles de passage

du modèle Entité-Association au modèle Relationnel

Entité :

- Chaque Entité devient une Relation (ou table).
- Chaque attribut de l'entité devient un attribut de la relation, y compris l'identifiant.
- Les attributs issus de l'identifiant constituent la clé de la relation.

Remarque : les attributs formant la clé de la relation sont soulignés dans le schéma de la relation.

Kamal BAL - Cours SGBD - Université Akli Mohand Oulhadj de Bouira - Février 2015

Règles de passage

du modèle Entité-Association au modèle Relationnel

Entité :

Etudiant (Matricule, Nom, Prénom, Datenaiss, Année_bac)

Kamal BAL - Cours SGBD - Université Akli Mohand Oulhadj de Bouira - Février 2015 45

Règles de passage

du modèle Entité-Association au modèle Relationnel

Association :

- Plusieurs cas
- Selon les cardinalités

Kamal BAL - Cours SGBD - Université Akli Mohand Oulhadj de Bouira - Février 2015

Règles de passage

du modèle Entité-Association au modèle Relationnel

Association 1.N

L'entité **A** devient une relation **A(lda, , , , *idb*)**

L'entité **B** devient une relation **B(idb, , , ,)**

La clé primaire **idb** de l'entité **B** migre comme **clé étrangère** dans la relation **A**.

Kamal BAL - Cours SGBD - Université Akli Mohand Oulhadj de Bouira - Février 2015 47

Règles de passage

du modèle Entité-Association au modèle Relationnel

Association 1.N

Personne (N°Personne, Nom, Prenom)

Maison (N°maison, Superficie, Adresse, *N°Personne*)

Kamal BAL - Cours SGBD - Université Akli Mohand Oulhadj de Bouira - Février 2015 48

Règles de passage du modèle Entité-Association au modèle Relationnel

Association N.N sans propriétés

A (ida, ...)

B (idb, ...)

R (ida, idb)

La clé de R est composée des clés des deux relations A et B

Kamal BAL - Cours SGBD - Université Akli Mohand Oulhadj de Bouira - Février 2015

Règles de passage du modèle Entité-Association au modèle Relationnel

Association N.N sans propriétés

Etudiant (idEtudiant, nomEtudiant, prenomEtudiant)

Module (idModule, nomModule)

Suivre (idEtudiant, idModule)

Kamal BAL - Cours SGBD - Université Akli Mohand Oulhadj de Bouira - Février 2015

Règles de passage du modèle Entité-Association au modèle Relationnel

Association N.N avec propriétés

A (id_a, ...)

B (id_a, ...)

R (ida, idb, propriété1)

Les propriétés de l'association R devient des attributs de la relation R
La clé de R est composée des clés des deux relation A et B

Kamal BAL - Cours SGBD - Université Akli Mohand Oulhadj de Bouira - Février 2015

Règles de passage du modèle Entité-Association au modèle Relationnel

Association N.N avec propriétés

Personne (idPersonne, nom)

Poste (nomPoste)

Occupe (idPersonne, nomPoste, durée)

Kamal BAL - Cours SGBD - Université Akli Mohand Oulhadj de Bouira - Février 2015

Règles de passage du modèle Entité-Association au modèle Relationnel

Entité faible 1.N

Hotel (NomHotel, Nb_etoiles)

Chambre (N°chambre, NomHotel, Type)

Kamal BAL - Cours SGBD - Université Akli Mohand Oulhadj de Bouira - Février 2015

Règles de passage du modèle Entité-Association au modèle Relationnel

Association 1.1 - 0.1

Personne (N_personne, nom)

Permis (N_permis, N_personne)

Kamal BAL - Cours SGBD - Université Akli Mohand Oulhadj de Bouira - Février 2015

