

Persistence des données :

Placentino

May 28, 2015

Contents

Evaluations du cours :	2
Objectifs du cours :	2
Plan du cours :	2
Introduction	4
Etude du monde réel	4
S.G.F.	4
S.G.B.D.	5
ANSI- X3 - SPARC	6
Les schémas de définition de données.	6
Modèle relationnel.	6
Les produits relationnels :	7
Les concepts de base	8
Algèbre relationnelle	9
Intersection (\cap)	9
Union (\cup)	9
différence ($-$)	9
projection :	9
Selection :	9

Exemple :	10
Produit cartésien :	10
Normalisation	10
Exercice	12
S.Q.L.	14
Introduction	14
4.2. La structure de SQL	14
DML : DATA MANIPULATION LANGUAGE	14
DDL : DATA DEFINITION LANGUAGE	14
DCL : DATA CONTROL LANGUAGE	14
Quelques règles syntaxiques :	15
DML de SQL	15
D.C.L.	17
ACID :	17
Le DDL définitions des schémas de données	17
Le schéma conceptuel	17

Evaluations du cours :

- Interrogation en Janvier 25%
- Examen en Mai/Juin 75%

Objectifs du cours :

- Pouvoir percevoir et comprendre les différents niveau de description des données.
- Maitriser du SQL-DML (DML = Data Manipulation Language, c'est la partie C.R.U.D.)
- Bonne connaissance du DDL (Data Definition Language).

Plan du cours :

1. Introduction
2. Modèle Relationnel
3. S.Q.L. (Non procédurale, le S.G.B.D. le traduit en procédural)
4. Embedded-SQL (Utiliser SQL pour interroger, manipuler des données persistantes)
5. Programmation sur serveur
6. Sécurité
 - Détails sécurité physique
 - Détails sécurité d'accès
 - Sécurité logique (les données soient bien coérentes)
 - Concurrence d'accès

Il existe plusieurs familles de S.G.B.D.¹²

- Relationnel
- hiérarchique
- réseau
- ...

-
- Systeme = Ensemble de programmes offrant des services, des facilités aux utilisateurs.
 - Gestion =
 - Manipulations des données (manipulations = C.R.U.D.)
 - sécurisation des données(octroiement des droits d'accès, physique, cryptage, logique)
 - définir (définir la structure des données),
 - ...

Contrainte d'intégrité : une donnée non-intègre != une donnée incorrecte. C'est une règle que les données doivent respecter ³. Notion qui existe dans le monde réel et qui ne provient pas explicitement du monde informatique.

"C'est une règle que les données doivent suivre pour représenter un monde réel possible" **ADT, 2014.**

Une base de donnée est cohérente quand toutes les contraintes d'intégrité sont respectées.

¹Système de Gestion de Base de Données

²D.B.M.S. (Data Base Managment System)

³Ex. : Le prix d'un article DOIT être positif

Introduction

Etude du monde réel

La redondance est génératrice d'incohérences -> On essayera toujours d'avoir une modélisation unique du monde réel.

Modèle Conceptuel de Données, une représentation du réel qui semble nous suffire, conceptuel car aucune notions technique.

S.G.F.

- M.C.D. -> { de fichiers représentant cela } .
- M.P.D. -> Modèle Physique des Données DOIT représenter tout le M.C.D.
- M.C.D. est l'agrégation des visions de chacun

Problèmes :

- **Sécurité** : Demander la liste des employés nous donne accès a toutes les informations, même son salaire.
- **Réinterprétation des données.**
- **Evolution du M.C.D.**
- **Performance** : elles se dégradent en fonction du temps⁴.
- **Contrainte d'Intégrité** écrite par le programmeur et redondant sur l'ensemble des programmes qui utilisent les données régies par ces contraintes. Cette contrainte prend sa source dans le monde réel, par l'analyste. Si les C.I. sont respectées alors les données sont cohérentes.
- D.D. : **Description des données** est éclatée entre la document du S.G.F.⁵ et la documentation des données et des programmes (+ redondance).

⁴évolution du volume des données, ...

⁵Système de Gestion de Fichiers

S.G.B.D.

G. Gardarin, **BD objets et relationnelles**.

- **Disposer de l'indépendance logique** : Une seule réinterprétation pour l'ensemble des programmes liés à UNE vision des données. Si les données d'une vision change, il suffit de changer la vision. ⁶ Apprendre au SGBD a travailler avec une vision des données.
- **Disposer de l'indépendance physique** : Pouvoir modifier la structure des données, cette modification sera prise en charge par le SGBD et rien est a changé au programme. Les performances vont donc évoluer (+ ou - si je fais de la merde).
- **Absence de redondance** (espérée) : On espere que tout fait du monde réel soit exprimé une fois au maximum.
- **Intégrité des données** = cohérence des données = Data consistency. Faire apprendre les contraintes d'intégrités par le SGBD, on ne l'écrit donc qu'a un seul endroit => absence de redondance
- **Partagabilité des données** : Ex : on peut utiliser un ATM même si qqun retire déjà de l'argent autre part.
- **Sécurité des données** :
 - **sécurité physique** : Se mettre à l'abri des conséquences de problème physique⁷
 - **sécurité d'accès** : Seul les personnes autorisées ont le droit de faire ..
 - **sécurité logique** : gérer la concurrence d'accès en disposant d'outil nous permettant d'éviter les problèmes. - Administration centralisée des données - Manipulation des données par des non-informaticiens

⁶Ex. : le cube.

⁷crash disque, coupure courant, ...

ANSI- X3 - SPARC

Les schémas de définition de données.

Les descriptions des données doivent être fournies en **3 type de schémas distincts** :

- Schéma central
 - = schéma conceptuel le plus proche du monde réel pour celui qui s'occupe de BD
 - = Ensemble de tables à décrire (traduction du M.C.D.)
- Schéma externe
 - = Constructions réalisées sur le schéma conceptuel⁸ => PROMET L INDEPENDANCE LOGIQUE
- Schéma (unique) interne = Comment sont stockées les différents élément et quel chemin d'accès on peut utiliser pour retrouver telle ou telle donnée. Modifier le schéma interne n'affectera rien.

ANALYSTE	BDISTE

M.C.D.	
M.L.D.	SCHEMA CONCEPTUEL (trad. du M.C.D.)
M.P.D.	SCHEMA CONCEPTUEL + SCHEMA INTERNE

L'ensemble des données de la description des données = **METADONNES**, elles sont stockées dans un BDD = **METABASE**.

SYSTABLES	Tno	Tnom
	423	Etudiant

SYSCOLUMNS	Cno	Cnom	Ctable	Cordre	Ctype
	1737	EtuNo	423	1	int
				

Modèle relationnel.

1. Le modèle hiérarchique

⁸ce sont les perceptions des données par les utilisateurs

- Le schéma conceptuel est nécessairement un arbre.
 - Chaque element peut avoir de 0 à n enfants, et ainsi de suite.
 - Default majeur : lors de la conception du schéma conceptuel, on privilège déjà certains chemins, des choix sont ainsi fait dès le départ.
2. Modèle réseau : Même concept mais on est plus limité aux arbres mais on peut utiliser des graphes
 3. Modèle relationnel
 4. Modèle Orienté Objet (on vera en 3ème que dans certains environnement c'est mega performant)
 5. Modèle Relationnel Objet : le relationnel vie à l'intérieur du R.O. car le relationnel pur est mort.
 6. NoSQL (= Not Only SQL)

Les produits relationnels :

- MySQL
- PostgreSQL
- Oracle
- SQLite
- javaDB (- derby)
- Acces
- SQLserver
- DB2
- Ingres
- Rbase
- ...

Les concepts de base

Codd a défini les bases du modèle relationnel à partir de la théorie des ensembles. Il a exprimé des choses de manière non procédurale (A U B en procédurale s'exprime à l'aide de boucle).

- 1ère notion : **Domaine** : est un ensemble⁹ Un ensemble ne peut pas présenter deux fois le même élément. L'ordre n'a pas de sens.
- 2ème : **Domaines compatibles** : lorsque nous pouvons comparer des éléments de chacun d'eux.
- 3ème : **Relation** : sous-ensemble d'un produit cartésien

ex : R1 = {(dupont, vrai), (dupuis, faux)}

R1		Nom : D1		Cadre : D2	:	SCHEMA DE LA RELATION
		DUPONT		VRAI	:	EXTENSION DE LA RELATION
		DUPUIS		FAUX		

- 4ème : **sémantique** : quel est le fait ou les faits représentés par les données. /!\ IMPORTANT /!\

Relation	Table

tuple	ligne (row)
attribut	nom de colonne

==> Base de données relationnelle : est une base de données dont le schéma conceptuel est constitué d'un ensemble de schéma de relation.

Dno : naturel
Dchar50 :
Dsec : { I, R, G}
Dan : {1, 2, 3}

Etudiant(etuNo : Dno, etuNom : Dchar50, etuSec : Dsec, etuAn : Dan)

Cette relation représente les élèves régulièrement inscrits à l'esi durant l'année 2014 2015 pour lesquels etuno représente son numero, ...

- **Degré d'une relation** : nombre d'attributs de la relation, (de colonnes de la table).

⁹ex.: l'ensemble des entiers, des dates, des entiers pairs, ..

- **Deux relations sont compatibles** : les deux relations doivent avoir le même degré et les colonnes doivent être compatibles deux à deux
1^{ere} - 1^{ere}, 2^{eme} - 2^{eme}, ...

Le modèle relationnel (pas sql) accepte plusieurs types d'absence de valeurs. L'absence de valeur est très problématique :

- comparaison avec NULL

Algèbre relationnelle

Intersection (\cap)

Soient R_1, R_2 2 relations compatibles, $R_1 \cap R_2$, telles que :

SuitCoursC \cap SuitCoursBD	etuNo	etuNom
-----	26	Durant

Union (\cup)

Soient R_1, R_2 2 relations compatibles, $R_1 \cup R_2$,

$$R_1 \cup R_2 = R_1 + R_2 - R_1 \cap R_2$$

difference ($-$)

soient R_1, R_2 deux relations compatibles

$$R_1 - R_2 = R_1 - (R_1 \cap R_2)$$

les tuples présents en R_1 mais pas dans R_2 .

projection :

Soit relation $R(a, b, c, d, e)$
 $R[a]$ ne garde que les valeurs de la colonne a
 $R[b]$..
 $R[c]$..

Selection :

$R(\text{cond})$

Exemple :

voir feuille annexe.

Produit cartésien :

Soient $R_1(A_1, \dots, A_n)$ et $R_2(B_1, \dots, B_m)$
tels que $R_1 \times R_2$ de degré $n + m$

jointure :

interne : INNER JOIN

JOIN (A, B; $a_1 = b_3$)

traduit en : $A \times B$ ($a_1 = b_3$)

externe : OUTER JOIN

- Droite (right)

- Gauche (LEFT)

- Complete (FULL)

Normalisation

Dépendance fonctionnelle : (propriétés du monde réel, contraintes d'intégrité)

$R(A, B, C, \dots, N)$
 $A \rightarrow B$

Commande(com, cli, date, pro, Q, pu)

com \rightarrow cli, date

cli \rightarrow

com, pro \rightarrow Q, pu

cli, date \rightarrow com

Relations soient $R(A_1, A_2, \dots, A_n)$ une relation et X, Y des sous-ensembles de $\{A_1, \dots, A_n\}$, on dit que $X \rightarrow Y$ si pour toute extension r de R , pour tout t_1, t_2 appartenant à r on a

$\text{'R[X].t1 = R[X].t2'}$ (projection de R selon X)
 $\Rightarrow \text{'R[Y].t1 = R[Y].t2'}$

Prouvez que telle dépendance est fautive en regardant une table.

Propriétés des dépendances fonctionnelles

- Transitivité :
 $A \rightarrow B \text{ et } B \rightarrow D \Rightarrow A \rightarrow D$
- Reflexivité :
 $A \rightarrow A$
- Augmentation :
si $A \rightarrow B$ alors $A, C \rightarrow B$
- Additivité :
si $A \rightarrow B$ et $C \rightarrow D$ alors $A, C \rightarrow B, D$
- Pseudotransitivité :
si $A \rightarrow B$ et $B, C \rightarrow D$ alors $A, C \rightarrow D$
- Fermeture transitive :
 - Dépendances fonctionnelles \rightarrow X dépendances fonctionnelles
 - (fermeture transitive) \rightarrow couverture minimale (plus petits nombre de dépendances fonctionnelles mais ayant la même fermeture transitive)

Clé de relation : Sous ensemble minimal d'attributs qui déterminent tous les autres

Commande(com, cli, date, pro, Q, pu)

com, pro \rightarrow cli, date, Q, pu
cli, date, pro \rightarrow com, Q, pu

\Rightarrow 2 clefs de relation;

- clé : sous ensemble x d'attributs de R(A1, ..., An):
 1. $X \rightarrow A1, \dots, An$
 2. Il n'existe pas de y non inclut dans X : $y \rightarrow A1, \dots, An$

Exemple :

com, pro, Q n'est pas une clé car si on retire Q on peut toujours déterminer tous les attributs.

- clé candidate : clé qui n'a pas été choisie pour être primaire.

Exemple :

VEH(chassis, plaque, marque, ...)

Exercice

Exercice 1

- a) faux
- b) vrai
- c) faux
- d) faux plus petit ou égal

Les formes normales :

DF :

```
etuId -> etuNom, etuNat, etuDom
etuNat -> etuNatId
etuId, insAnnac -> anet, catId, oriId
oriId -> oriNom, oriTp
crsId -> crsLib, crsNbH, crsType
catId -> catLib
```

1ère FN (Forme Normale) :

```
etuId, insAnnac, crsId
```

On parle de $2^n - 1$ choses (7 là).

=> On est assuré maintenant qu'un eleve ne suit pas deux fois le même cours.(gain faible).

2ème FN :

création de 7 tables :

```
Etudiant(etuId, etuNom, etuNat, etuNatLib, etuDom) | etuId = cle CANDIDATE
AnneeAcad(insAnnac) | insAnnac = cle CANDIDATE
Cours(crsId, crsLib, crsNbH, crsType)
Inscription(etuId, insannac, anet, oriId, oriNom, oriTpEm, catId,
catLib)
ASuivi(etuId, crsId) [Table un peu compliqué..]
EstOrganisé(insAnac, crsId)
Suit(etuId, insAnac, crsId)
```

- de redondances
- gain de représentativité (représenter des choses sensé que je ne pouvais pas me représenter en première forme normale. Ex: ajouter un cours sans encore aucun étudiant qui le suis).

- d'incohérences

C.I.R. : Contraintes d'Intégrité Référencielles

```
Inscription[etuId] C Etudiant[etuId]
Inscription[insAnnac] C AnneeAcad[insAnnac]
aSuivi[etuId] C Etudiant[etuId]
aSuivi[crsId] C Cours[crsId]
estOrganise[insAnnac] C AnneeAcad[insAnnac]
estOrganise[crsId] C Cours[crsId]
Suit[etuId, insAnnac] C Inscription[etuId, insAnnac]
Suit[insAnnac, crsId] C estOrganise[insAnnac, crsId]
Suit[etuId, crsId] C ASuivi[etuId, crsId]
```

=> Apres les CIR on vient d'arriver en 2eme forme normale

2eme FN SSI tout attribut n'appartenant pas à une clef ne dépend pas d'une partie d'une clef + 1FN

3FN :

```
Etudiant(etuId, etuNom, etuNat, etuDom)
Pays(paysId, paysLib)

Inscription(etuId, insanac, anet, oriId, catId)
Orientation(oriId, oriNom, oriTpEns)
Categorie(catId, catLib)
```

C.I.R. en plus :

```
Etudiant[etuNat] C pays[paysId]
Inscription[oriId] C Orientation[oriId]
Inscription[catId] C Categorie[catId]
```

3eme FN SSI 2FN + tout attribut n'appartenant pas à une clef ne dépend pas d'un attribut non clef

RECAPITULATION :

R(A, B, C, D, E, F)

Si A, B = clef => 1FN

Si B -> C => pas 2FN

si D -> E => pas 3FN

/!!\ Interrogation : ajout de table(s) intermédiaire(s)

etrangere -> primaire
etrangere C primaire

S.Q.L.

Introduction

Language normalisé¹⁰ et donc partiellement portable.

4.2. La structure de SQL

DML : DATA MANIPULATION LANGUAGE

4 instructions :

- R = SELECT
- D = DELETE
- U = UPDATE
- C = INSERT

DDL : DATA DEFINITION LANGUAGE

- CREATE : Créer une table ou
- DROP {table, view, cluster} : Supprimer
- ALTER {nomObjet} : Modifier définition d'objet

DCL : DATA CONTROL LANGUAGE

- REVOKE
- GRANT

¹⁰ISO SQL86-SQL92 et ANSI

Quelques règles syntaxiques :

Délimiteur d'instruction :

- ; \
- espaceS = espace
- commentaires :
 1. -- (= //)
 2. /* */ (= /* */)
- délimiteurs de chaînes :
 1. ' doublé si utile dans la chaîne
 2. " délimiteur d'identifiant **CASE SENSITIF**, sinon tout en MAJ.

DML de SQL

SELECT Peut être imbriqué dans un **EXIST()**

UPDATE Mettre à jour, modifier, des tuples déjà existant.

```
UPDATE nomTable
  SET att1 = expr1
    [,att2 = expr2]
    [, ...]
  [WHERE cond]
```

Ex. : Multiplier par deux le salaire de Dupont :

```
UPDATE employe
  SET empSal = empSal * 2
  WHERE empNo = ..
```

Ex. :

```
UPDATE employe mgr
  SET empSal = 2 * (
    SELECT MAX(empSal)
      FROM employe
     INNER JOIN Departement
      ON empDpt = dptNo
     WHERE mgr.empNo = dptMgr
  )
  WHERE empNo IN ( SELECT dptMgr FROM Departement ) ;
```


!! :

- Toutes les instructions en DML disposent de "l'intégrité en lecture". => Thread safe, la lecture se fait sur une IMAGE figée. Soit, l'exemple précédent cohérent même si un employé est un manager. Le SGBD gère ça tout seul.
- Atomicité

Retourne un "SQLcode" qui vaut :

```
--> 0    "OK"
--> >0   "OK mais j'ai un truc à dire Farit" /WARNING\
        Ex : update avec un where il y a aucune ligne à modifier.
--> <0   "ERREUR"    Syntaxique "selct", Sémantique "select machin"
                    n'est pas un attribut de la table, de droit :
                    modifier une table de ADT.
```

DELETE

```
DELETE FROM table
        [WHERE cond]
```

Ex. :

```
DELETE FROM Employe
        where empNo = ..
```

INSERT

```
INSERT INTO nomTable[(attr1, attr2, attr3, ..)
        VALUES(.., .., .., ...)
```

OU

```
INSERT INTO nomTable[(attr1, attr2, attr3, ..)
        SELECT ...;
```

=> créer de la redondance, prendre des données quelque part et les recopier quelque part.

D.C.L.

Transaction = Sous ensemble minimum d'instructions qui fait passer une base de données d'un état cohérent à un état cohérent.

ACID :

- A : **Atomicité** = indivisible.
- C : **Consistency** = cohérence.
- I : **Isolation** = Si les transactions se passent en même temps, elle ne s'influent pas mutuellement.
- D : **Durability** = quand la transaction est terminée, les modifs sont répertoriées dans la BDD.

```
begin Transaction
- try {
|
|
|
|
| VALIDATION DE LA TRANSACTION
catch ( ERREUR )
|
| ANNULATION DE LA TRANSACTION
|
-
end Transaction
```

- COMMIT
- ROLLBACK

Ces deux instructions ne fonctionnent que sur instruction du DML.
DDL genere obligatoirement un commit par lui même.

Le DDL définitions des schémas de données

Le schéma conceptuel

1. Définition minimale d'une table

```
CREATE TABLE Departement(
    dptNo CHAR(3),
    dptLib VARCHAR(30),    // varchar = character varying
    dptAdm char(3)
);
```

2. Les types de données

- Les chaines
 - CHAR[ACTER] [(n)]
 - VARCHAR(n)
- Les chaines de bits
 - BIT[(n)]
 - BITVAR(n)
- Les nombres
 - NUMBER(precision, scale) // (nombre de chiffres, nombre apres la virg)
 - DECIMAL(precision, scale)
 - FLOAT, REAL, DOUBLE PRECISION
 - INT ou INTEGER
- Moments
 - DATE
 - TIME
 - TIMESTAMP
- LOB (Large Object) (stockage brut de gros fichiers)
 - CLOB // Characters LOB ex. : file.doc
 - BLOB // Binary LOB ex. : file.mp3

3. Caractère obligatoire d'un attribut

```
CREATE TABLE Departement(
    dptNo CHAR(3) NOT NULL,
    dptLib VARCHAR(30) NOT NULL,    // varchar = character varying
    dptMgr char(3) NOT NULL,
    dptAdm char(3) [NULL/NOT NULL] [DEFAULT {lit}]
);
```

4. Valeur par défaut

SQL INSERT INTO Departement(dptNo, dptLib, dptMgr) VALUES('A21', 'dfsdfs', 'B33'); => dptAdm sera par default à NULL si DEFAULT n est pas présent.

5. Contraintes :

```
CREATE TABLE Departement(  
    dptNo CHAR(3) NOT NULL,  
    dptLib VARCHAR(30) NOT NULL,          // varchar = character varying  
    dptMgr char(3) NOT NULL,  
    dptAdm char(3)  
    CONSTRAINT DepartementPK PRIMARY KEY(dptNo)  
);
```