



LES SYSTEMES DE GESTION DE BASES DE DONNEES

Agnès HERRMANN



UNE SOCIETE DE L'INFORMATION

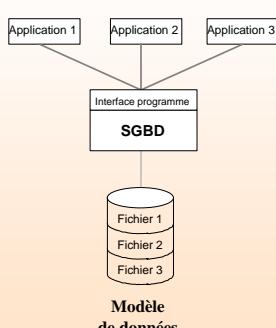
- Des besoins de stockage en croissance exponentielle :
 - Imagerie et graphisme
 - Applications scientifiques
 - Gestion commerciale ou bancaire

- Gestion de données
 - Retrouver rapidement des informations
 - Les traiter pour en extraire de nouvelles données



POURQUOI UN SGBD ?

- Une interface entre applications et stockage de l'information
- Le SGBD masque à l'utilisateur :
 - le niveau du stockage physique des données
 - la structure des fichiers





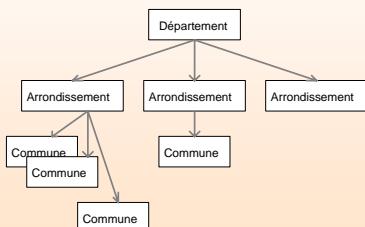
MODELES DE DONNEES

- SGBD = modèle de données
- Un modèle est une description logique :
 - des structures de données
 - des contraintes d'intégrité
 - des opérations de manipulation
 - des mécanismes de définition de règles d'intégrité
- Historiquement, plusieurs modèles ont été proposés :



MODELE HIERARCHIQUE

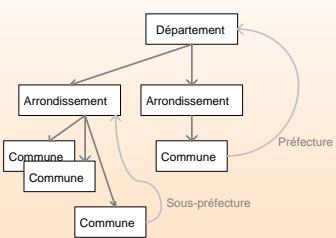
- Un père, des fils.
- On obtient une arborescence





LE MODELE RESEAU

- Les liens sont quelconques.
- On définit un graphe au lieu d'un arbre.





LE MODELE RELATIONNEL

- Il est fondé sur la notion mathématique de relations entre valeurs
- Très utilisé actuellement en gestion, il est parfois insuffisant pour les applications techniques (CAO, Génie logiciel, SIG, ...)
- Extensions en cours pour pallier ses limites
- Il sera vu en détail (Access est un SGBDR)



LE MODELE ORIENTE OBJET

- À chaque structure de donnée (ensemble de propriétés) est associé un ensemble de méthodes (opérations)
- Les objets sont groupés par classe, avec une notion d'héritage (filiation entre classes)
- Ne sera pas vu dans le cadre de ce cours



Ecole Nationale du Génie de l'Eau et de l'Environnement de Strasbourg

LE MODELE RELATIONNEL



OBJECTIFS DU MODELE RELATIONNEL

- Indépendance entre application et représentation interne des données
- Gérer les problèmes de cohérence des données et de redondance



STRUCTURE DE DONNEE SIMPLE : LA TABLE

- Une table est caractérisée par un **nom**.
- Les tables sont indépendantes entre elles
- Une table est un ensemble **d'enregistrements** indépendants (les *lignes* de la table)
- Un **enregistrement** est une liste de valeurs, chacune appartenant à un domaine de valeurs.
- On appelle **champ** (ou attribut) une colonne d'une table, caractérisé par un nom



EXEMPLES DE TABLES

Personne	N° insee	nom	adresse
	1541136345214	Charles	Paris
	1640267482220	Jean	Strasbourg
	2700613163021	Marie	Marseille

Voiture	type	immatriculation	couleur
	Renault 5	435 LN 74	rouge
	Fiat 500	356 XG 67	blanche
	Peugeot 205	123 KC 09	rouge



VALEURS DES CHAMPS

- Chaque champ possède un type...
 - Types de champ : chaîne de caractère, entier, réel, date...)
- ...et un domaine de valeur
 - positif, < à une valeur donnée, ...
- Ces règles d'intégrité sémantiques sont vérifiées à chaque mise à jour
- **Valeur nulle** : cette valeur particulière est utilisée lorsqu'un champ n'est pas renseigné pour certains enregistrements.

Attention : la **valeur nulle** est différente de la **valeur 0** !!



CLES ET CONTRAINTES D'INTEGRITE

- Les enregistrements d'une table sont identifiés de manière unique par un ou l'association de plusieurs champs, c'est la **clé**
- Il ne peut exister qu'un seul enregistrement par valeur de clé : c'est **l'unicité de la clé**
- Pour créer un lien entre 2 tables dans le modèle relationnel, on utilise la clé primaire d'une table (champ père) en tant que **clé étrangère** dans une autre table (champ fils)
- Les clés étrangères doivent être définies dans le schéma : c'est la **contrainte d'intégrité référentielle**
- L'intégrité référentielle est vérifiée à chaque mise à jour de la base de données



EXEMPLE DE TABLES AVEC CLE ETRANGERE

Personne	N° insee	nom	adresse
	1541136345214	Charles	Paris
	1640267482220	Jean	Strasbourg
	2700613163021	Marie	Marseille

Voiture	type	immat.	couleur	Propriétaire
	Renault 5	435 LN 75	rouge	1541136345214
	Fiat 500	356 XG 67	blanche	1640267482220
	Peugeot 205	123 KC 09	rouge	1541136345214



FONCTIONNALITES DES SGBD



FONCTIONNALITES DES SGBD

- Description des données (définition du *schéma de la base de donnée*)
- Manipulation des données
- Maintien de l'intégrité des données
- Maintien de la sécurité



DESCRIPTION DES DONNEES

- Langage de Définition de Données (LDD)
 - définir la structure de données
 - fixer les attributs des objets de la base
 - établir les liens entre les objets
 - donner des mécanismes de définition des contraintes voulues par l'utilisateur



MANIPULATION DES DONNEES

- Langage de Manipulation de Données (LMD)

- rechercher des données
- modifier des données
- insérer des données
- supprimer des données



MAINTIEN DE L 'INTEGRITE DES DONNEES

- Les contraintes d'intégrité sont définies par le modèle (*cohérence interne de la base*)
- Les règles d'intégrité sont définies par l'utilisateur (*caractéristiques des données de l'application*)



MAINTIEN DE LA SECURITE

- Gestion des utilisateurs et de leurs droits d'accès (confidentialité)
- Gestion de la concurrence d'accès
- Sûreté de fonctionnement avec mécanisme de reprise :
 - à chaud : si les données n'ont pas été détruites
 - à froid : à partir de la sauvegarde et du journal des transactions



FIN...

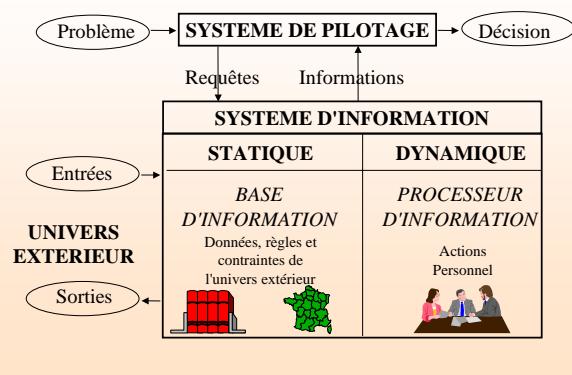


LA MODELISATION DES DONNEES

Agnès HERRMANN

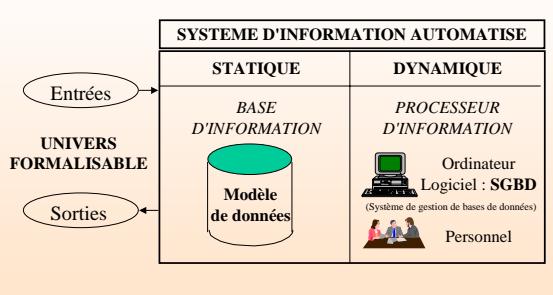


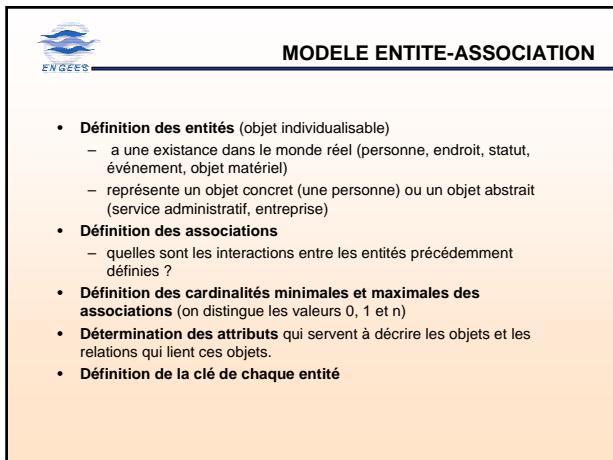
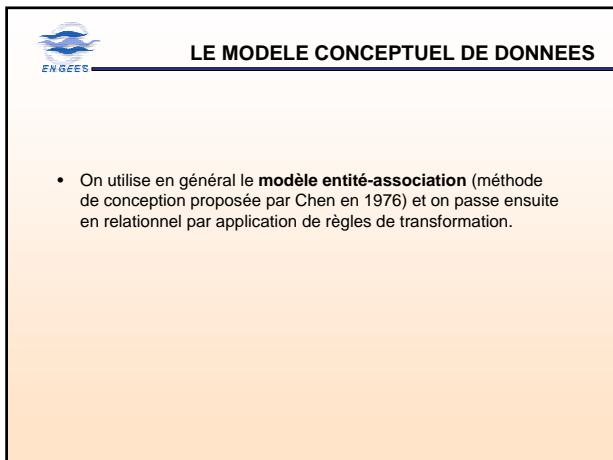
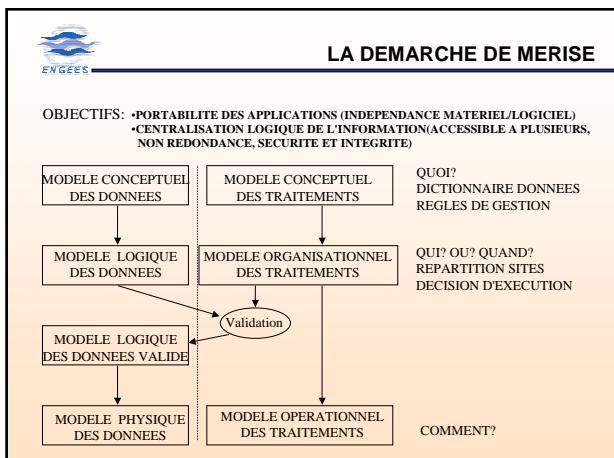
LE SYSTEME D'INFORMATION

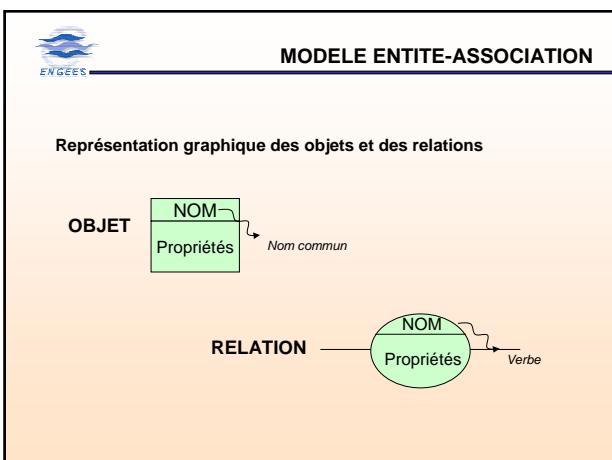


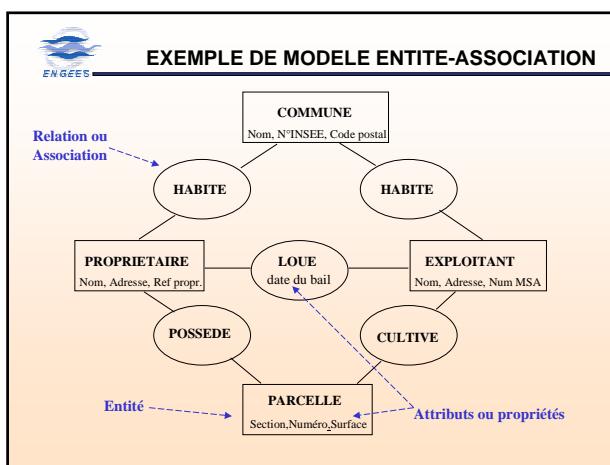


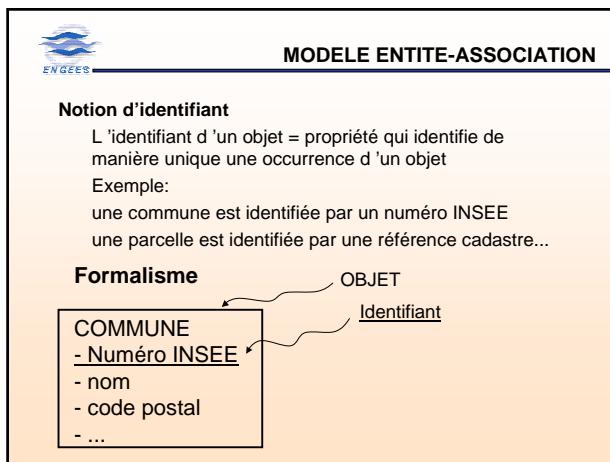
AUTOMATISATION DU S.I.

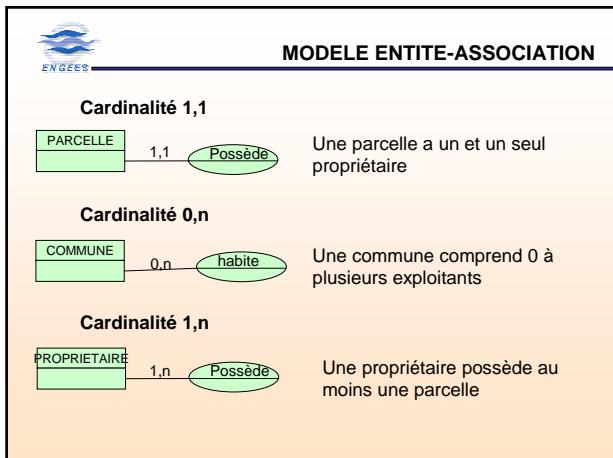
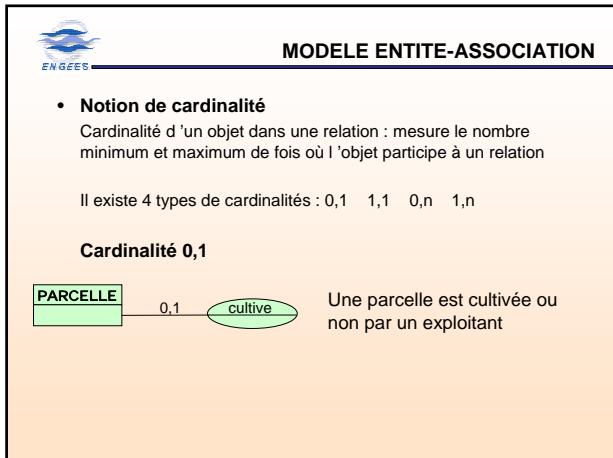
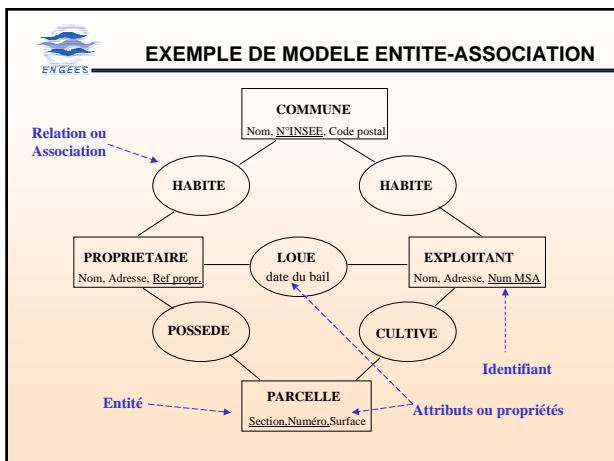


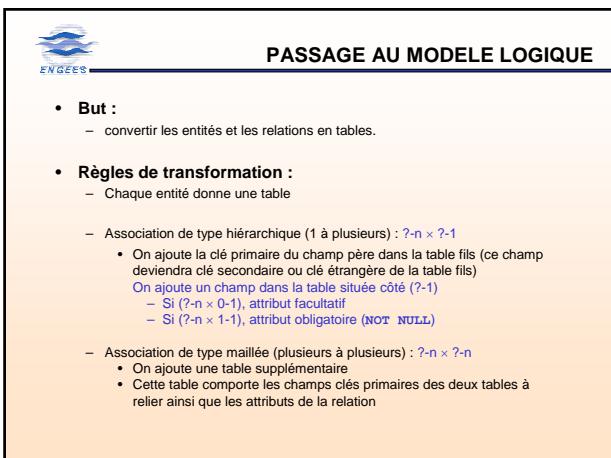
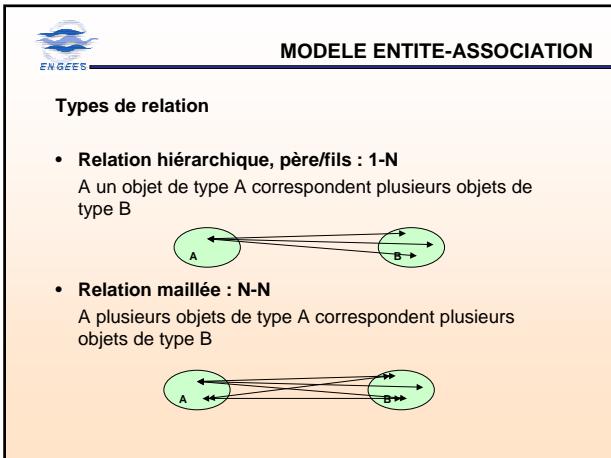
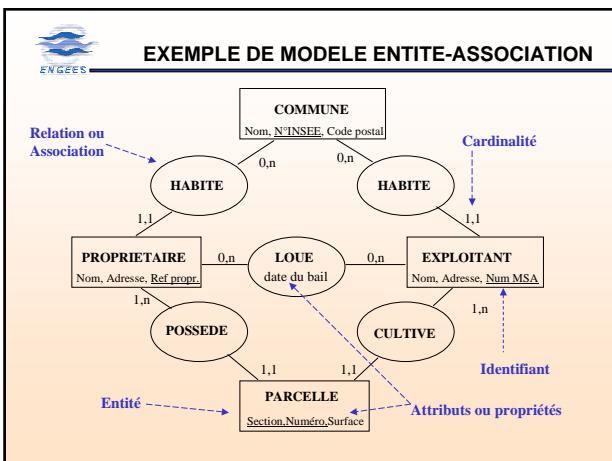


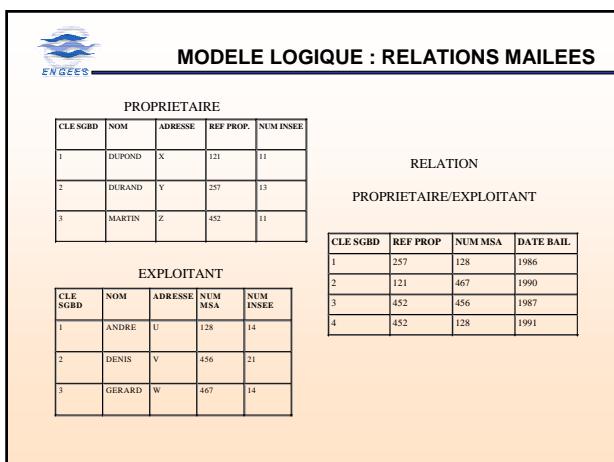
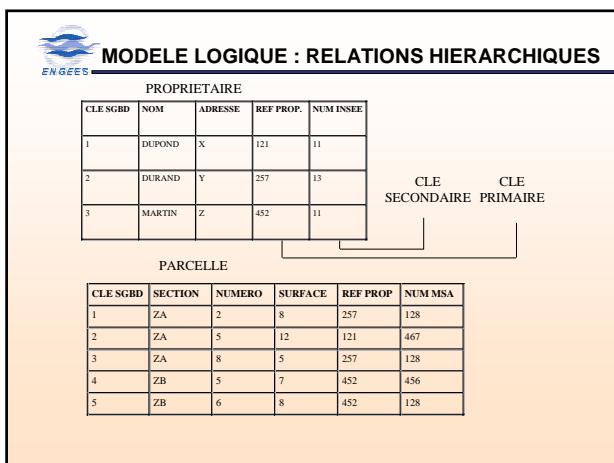
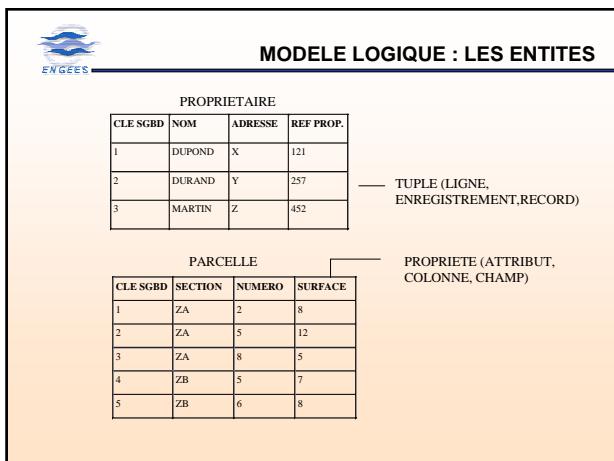


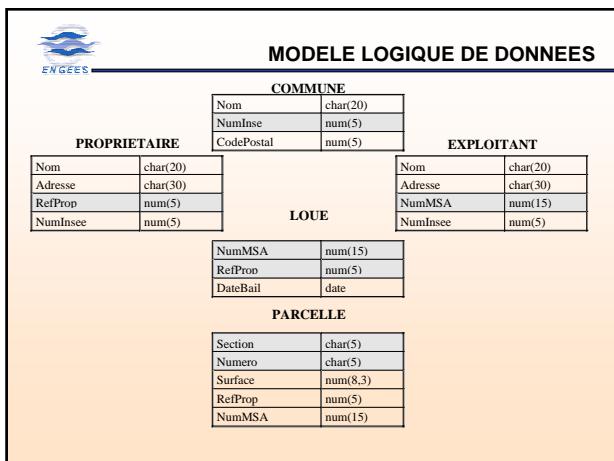


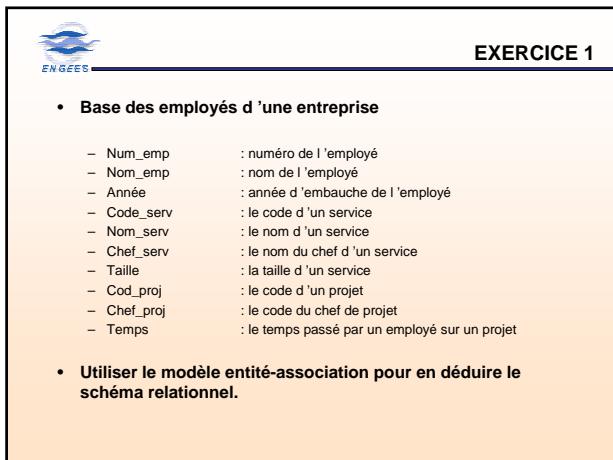


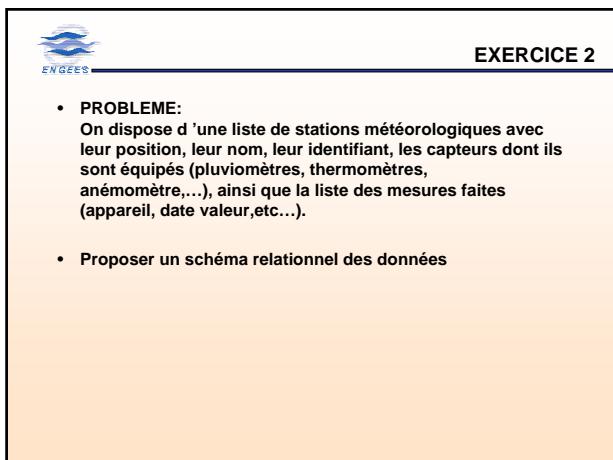














EXERCICE 3

- **PROBLEME:**

Soit un organisme départemental centralisateur d'information devant répondre aux demandes d'agriculteurs concernant la location de matériel (outils, engins de chantier, matériel agricole)

- **Mise en place d'une base de données pour renseigner les exploitants sur les dates et conditions de location, sur les matériels disponibles et les entreprises offrant ce type de service**

• Eléments disponibles :

- adresse des entreprises
- liste d'exploitants demandeurs
- liste du personnel des entreprises
- nom du propriétaire des entreprises
- raison sociale des entreprises
- tarifs et conditions de locations

Règles

- en plus de la location du matériel, l'exploitant peut demander une personne qui l'utilise
- certains matériels ne peuvent être conduits que par du personnel qualifié
- conserver un historique des locations



FIN...



ALGEBRE RELATIONNELLE ET SQL

Agnès HERRMANN



ALGEBRE RELATIONNELLE

- Pour obtenir une nouvelle table, on applique sur une ou plusieurs tables une requête
- Requêtes et tables sont donc équivalentes...
- Une requête est composée d'opérateurs (ensemblistes) de l'algèbre relationnelle
- Les opérateurs peuvent s'enchaîner : une requête est une succession d'opérateurs



ALGEBRE RELATIONNELLE

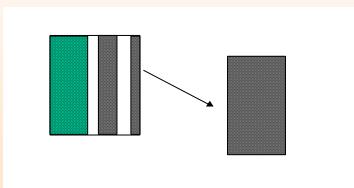
- Les 9 opérateurs principaux

Nom	type	opérateur
– Projection	unaire	relationnel
– Extension	unaire	relationnel
– Sélection	unaire	relationnel
– Union	binaire	ensembliste
– Intersection	binaire	ensembliste
– Différence	binaire	ensembliste
– Produit cartésien	binaire	ensembliste
– Jointure	binaire	relationnel
– Agrégat	unaire	relationnel



PROJECTION

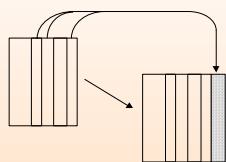
- La projection consiste à ne conserver que certains champs de la table (sélection de colonnes de la table)





EXTENSION

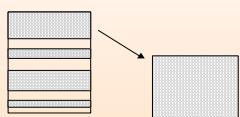
- L'extension consiste à rajouter un champ (colonne de la table), calculé à partir des autres champs de la table





SELECTION

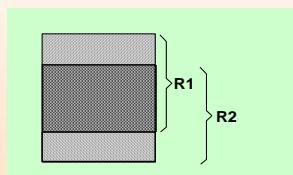
- La sélection consiste à ne conserver que les enregistrements de la table vérifiant un certain critère (sélection de lignes de la table)





UNION

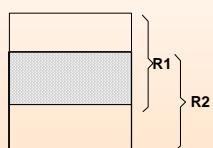
- L'union s'applique à deux tables de même schéma (ayant les mêmes champs) et fournit une table comportant les enregistrements de l'une ou l'autre des tables de départ





INTERSECTION

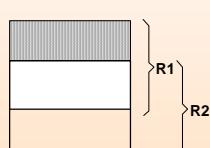
- L'intersection, dans les mêmes conditions, ne retourne que les enregistrements faisant partie des deux tables de départ





DIFFERENCE

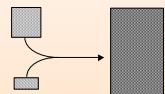
- La différence entre deux tables de même schéma renvoie les enregistrements de la première table qui n'appartiennent pas à la seconde





PRODUIT CARTESIEN

- Le produit cartésien de deux tables fournit une table ayant les champs des deux tables
- Le nombre d'enregistrements est le produit des nombres d'enregistrements initiaux !





JOINTURE

- La jointure revient à un produit cartésien suivi d'une sélection
- Cette dernière se fait en général sur l'égalité de deux champs (un de chaque table)
- Elle permet d'associer à un enregistrement d'une table des champs d'une autre table



AGREGAT

- L'agrégat regroupe les enregistrements sur un critère d'égalité de certains champs, projette ces champs, puis ajoute à la table un champ calculé par une fonction d'aggrégation (minima, maxima, moyenne, somme, compte, etc.) appliquée à chaque ensemble de valeurs regroupées
- Cela permet de répondre à des questions du genre "Combien de voitures y a-t-il par couleur ?" ou "Quelle est la puissance moyenne par marque ?"



AUTRES OPERATIONS

- **Division de deux tables (R1 + R2)**
 - sous-ensemble des enregistrements de R1 tels que le produit cartésien avec R2 soit sous-ensemble de R1.
- **Opérateurs irréductibles (projection, extension, sélection, produit, union et différence)**
- **Opérateurs non irréductibles :**
 - jointure = produit cartésien + sélection
 - intersection = union + différence
 - division = jointure + projection + différence



VUE RELATIONNELLE

- La table résultant d'une requête peut être utilisée comme une table de base
- Ces tables s'appellent *relations déduites ou vues*
- Le mécanisme des vues forme le niveau externe ; un utilisateur peut ne voir que les vues que l'administrateur lui a préparées



LANGAGES DE REQUETES

- Appelés aussi L4G (langages de quatrième génération)
- Ils permettent d'exprimer les requêtes
- Le langage (normalisé) SQL s'est imposé pour les SGBD relationnels
- Fonctionnalités du langage SQL
 - Langage de Manipulation de Données (LMD)
 - Langage de Définition de Données (LDD)



POURQUOI SQL ?

- Langage **SEQUEL** en 1975 (issu de **SQUARE** pour la base de donnée relationnelle R d'IBM)
- Evolution en **SQL**, qui devient un standard de fait car utilisé/copié par de nombreux autres produits (DB2, Oracle, ...)
- **Portabilité**
 - programme SQL réutilisable (SGBD différent)
- **Partageabilité des données**
 - SQL permet l'échange de données entre SGBD



SQL EN INTERROGATION

- La forme générale est

```
SELECT [DISTINCT] attribut, f(attribut)
FROM tables[WHERE condition]
```
- ◆ la clause **SELECT** donne les attributs ou les fonctions à retourner (fonctions : + - / *)
- ◆ la clause **FROM** donne les tables sur lesquelles porte la requête
- ◆ la clause **WHERE** donne la condition de sélection sur les enregistrements



SQL EN INTERROGATION

- Les requêtes peuvent être emboîtées (semi-jointure).
- La condition peut être complexe.
 - combinaisons : OR AND NOT
 - prédictats variés := <> < > <= >=



AUTRES CONDITIONS

- valeur [NOT] BETWEEN valeur AND valeur
- valeur [NOT] IN (valeur, valeur, ...)
- attribut [NOT] LIKE pattern
- attribut IS [NOT] NULL
- **quantificateurs possibles :** ALL SOME ANY suivis d'une liste de valeurs ou d'une sous-requête.



AGREGATS

- Pour les agrégats, on utilisera des fonctions d'agrégats :
AVG MAX MIN SUM COUNT

```
SELECT [DISTINCT] fonction d'agrégat (attribut)
FROM relations [WHERE condition]
GROUP BY attribut [HAVING condition]
```



SQL - INSERTION

- Ajout d'enregistrements dans une table :
 - `INSERT INTO rel.[att.,att.] VALUES(value, NULL)`
 - `INSERT INTO rel.[att.,att.] requête`
- Exemples :
 - `INSERT INTO voitures SELECT * FROM [voitures bis] WHERE immat LIKE "*2A" OR immat LIKE "*2B"`



SQL - SUPPRESSION

- Suppression de n-uplets dans une relation :
 - `DELETE FROM relation WHERE condition`
- Exemples :
 - `DELETE FROM voitures WHERE immat LIKE "*20"`



SQL EN MISE A JOUR

- Modification de valeurs dans une relation :
 - `UPDATE relation
SET attribut = expression
WHERE condition`
- Exemples :
 - `UPDATE voitures
SET couleur = "bleu"
WHERE immat = "435WZ89"`



SQL POUR LES DEFINITIONS

- Types de données
- Création d'une table
- Création d'une vue
- Destruction d'une table ou d'une vue
- Gestion des droits



TYPES DE DONNEES

- **Entiers** : INTEGER, INT, SMALLINT
- **Virgule fixe** : NUMERIC, DECIMAL, DEC
- **Nombres flottants** : FLOAT, REAL, DOUBLE PRECISION
- **Chaînes de caractères** : CHARACTER, CHAR



CREATION D 'UNE TABLE

- On peut préciser pour chaque attribut, ou pour la table entière des contraintes ("NOT NULL", "UNIQUE")
- On peut définir une clé étrangère par la contrainte globale :
FOREIGN KEY (attr.,...)
REFERENCE table(attr.,...)



CREATION ET DESTRUCTION

- **Création d'une vue**
CREATE VIEW nom AS requête
- **Destruction d'une table ou d'une vue**
DROP TABLE nom
DROP VIEW nom



GESTION DES DROITS

- GRANT privilège ON table TO utilisateur

- Les priviléges sont :

- SELECT
- INSERT
- DELETE
- UPDATE(attribut)
- ou bien "ALL PRIVILEGES"



FIN...
