

Les Bases de Données

Malek RAHOUAL
mrahoul@ibisc.univ-evry.fr

Bases de Données - 2015-2016 - Malek RAHOUAL

PLAN

1. Introduction
 - 1.1. Les BDD
 - 1.2. Les SGBD
 - 1.3. Modèle relationnel
2. Le langage Algébrique
3. Le langage SQL

Bases de Données - 2015-2016 - Malek RAHOUAL

Bibliographie

- Livres
 - ↳ Bases de données Objet et relationnel de Georges Gardarin (Edition Eyrolles)
 - ↳ Mastering Oracle SQL (O'Reilly)
- Web
 - ↳ <http://www-inf.int-evry.fr/COURS/BD/accueil-ext.html> (modèle relationnel, SQL, Conception Entité-Association)
 - ↳ http://cui.unige.ch/~nerima/index_bd.html (modèle relationnel, SQL, Conception Entité-Association)

Bases de Données - 2015-2016 - Malek RAHOUAL

1. Introduction

Bases de Données - 2015-2016 - Malek RAHOUAL

Définition

- Base de données : ensemble structuré de données pour satisfaire simultanément plusieurs utilisateurs
 - ↳ de façon sélective
 - ↳ enregistrées avec le minimum de redondance
 - ↳ en un temps opportun.
- Stockage des données :
 - ↳ Sur support permanent
- Administration des données :
 - ↳ Systèmes de Gestion de Base de Données (SGBD)

Bases de Données - 2015-2016 - Malek RAHOUAL

Système de Gestion de Base de Données

- Un **Système de Gestion de Base de Données (SGBD)** est un ensemble de logiciels permettant la recherche, la mise à jour et la sauvegarde de données stockées sur mémoire secondaire.

```

graph TD
    User((User)) <--> SGBD[SGBD]
    SGBD <--> OS[Système d'exploitation]
    OS <--> Disk[(Disque)]
  
```

Place du SGBD dans le système

Bases de Données - 2015-2016 - Malek RAHOUAL

Système de Gestion de Base de Données

- Un SGBD assure :
 - ☞ la description des données,
 - ☞ leur recherche et mise à jour,
 - ☞ la sûreté :
 - vérifier les droits d'accès des utilisateurs;
 - limiter les accès non autorisés ;
 - crypter les informations sensibles
 - ☞ la sécurité :
 - sauvegarde et restauration des données;
 - limiter les erreurs de saisie, de manipulation.
 - ☞ l'intégrité : définir des règles qui maintiennent l'intégrité de la base de données (contraintes d'intégrité)
 - ☞ la concurrence d'accès : détecter et traiter les cas où il y a conflit d'accès entre plusieurs utilisateurs.

7

Bases de Données - 2015-2016 - Malek RAHOUAL

Système de Gestion de Base de Données

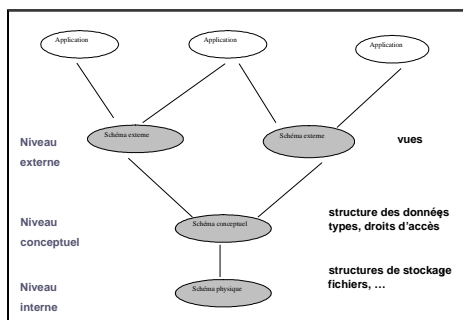
- Domaines d'utilisation :
 - ☞ Bases de Connaissances,
 - ☞ Systèmes Experts,
 - ☞ Systèmes d'Information Géographique,
 - ☞ Edition de Documents Informatisés (EDI),
 - ☞ Systèmes d'Information Documentaire (SID),
 - ☞ Conception Assistée par Ordinateur (CAO),
 - ☞ Gestion de Production Assistée par Ordinateur (GPAO)
- Ce sont des domaines où une information structurée est
 - ☞ enregistrée et gérée par un SGBD
 - ☞ traitée selon des besoins spécifiques (l'interface et les outils de traitement dépendant plus particulièrement de l'application).

8

Bases de Données - 2015-2016 - Malek RAHOUAL

Modélisation des données

- Trois niveaux de description des données:



9

Bases de Données - 2015-2016 - Malek RAHOUAL

Le niveau interne

- Spécifie comment les données sont enregistrées sur les mémoires secondaires (disque, bandes, ...). Le schéma physique décrit les données telles qu'elles sont stockées dans la machine
- Exemple :
 - ☞ les fichiers qui les contiennent (nom, localisation, taille)

10

Bases de Données - 2015-2016 - Malek RAHOUAL

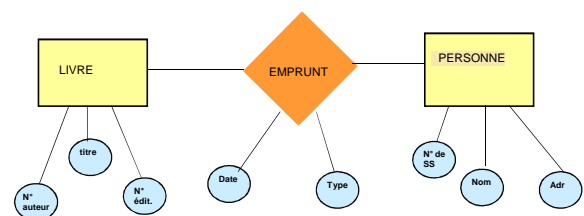
Le niveau conceptuel

- Le schéma conceptuel : description des données qui existent dans l'entreprise sans souci d'implantation en machine:
 - ☞ les types de données élémentaires
 - (par ex. titre de livre, nom d'auteur, éditeur, nombre de pages)
 - ☞ les types de données composés (description des objets)
 - (par ex. livre, personne...)
 - ☞ les types de données composés (description des associations entre les objets)
 - (par ex. un livre emprunté par une personne,...)
 - ☞ certaines règles
 - (par ex. une personne ayant plus d'une semaine de retard pour rendre un livre à la bibliothèque n'a pas le droit d'emprunter des livres pendant deux semaines, ...)

11

Bases de Données - 2015-2016 - Malek RAHOUAL

Exemple de schéma conceptuel



12

Bases de Données - 2015-2016 - Malek RAHOUAL

Le niveau externe

- C'est le plus haut niveau d'abstraction de la base de données.
- Il est aussi appelé niveau vue et est propre à un utilisateur ou à un groupe d'utilisateurs et ne lui présente qu'une vue partielle de la réalité : celle qui intéresse son service.
- Il y a bien entendu plusieurs vues d'une même BD.

13

Bases de Données - 2015-2016 - Malek RAHOUAL

Historique des SGBD

- Trois générations dans l'évolution des systèmes de gestion de base de données (SGBD).
- Chaque génération est caractérisée par le modèle de données qu'elle propose.
 - Les systèmes hiérarchiques et réseaux
 - Les systèmes relationnels
 - Les systèmes objets

14

Bases de Données - 2015-2016 - Malek RAHOUAL

Les systèmes hiérarchiques et réseaux

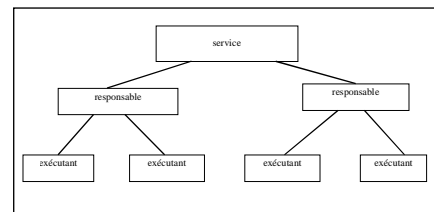
- Les systèmes hiérarchiques et réseaux
 - Ces systèmes sont apparus dans les années 60 et constituent la première génération de SGBD.
 - Ils résultent d'une approche de conception système qui vise à étendre le système de fichier par des possibilités de liaison inter-fichiers matérialisées par des pointeurs.

15

Bases de Données - 2015-2016 - Malek RAHOUAL

Les systèmes hiérarchiques et réseaux

- Modèle hiérarchique (1960)
 - Les données sont représentées dans la base sous la forme d'un arbre
 - La structure d'arbre utilise des pointeurs et détermine le chemin d'accès aux données.

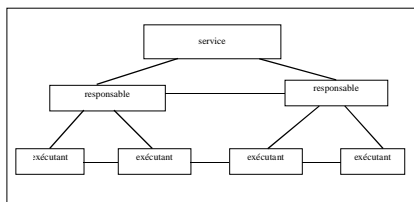


16

Bases de Données - 2015-2016 - Malek RAHOUAL

Les systèmes hiérarchiques et réseaux

- Modèle réseau (1970)
 - Les données sont représentées dans la base sous la forme d'un graphe quelconque
 - La structure de graphe utilise des pointeurs et détermine le chemin d'accès aux données.



17

Bases de Données - 2015-2016 - Malek RAHOUAL

Les systèmes relationnels

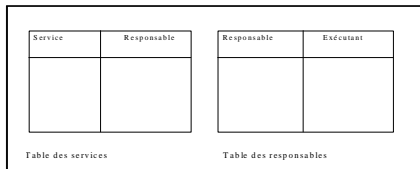
- Ces systèmes constituent la deuxième génération de SGBD. Ils sont fondés sur le concept de relation, d'où leur nom relationnel. Ces systèmes ont été assez long à s'imposer et représentent aujourd'hui la grande majorité des SGBD.
- On note plusieurs étapes dans l'apparition de ces systèmes :
 - **1970** : apparition du modèle relationnel
 - **1980** : apparition du premier SGBD relationnel : SGBDR
 - **1990** : les SGBDR représente plus de 50% du parc des SGBD

18

Bases de Données - 2015-2016 - Malek RAHOUAL

Les systèmes relationnels

- La théorie des SGBDR est fondée sur la théorie mathématique des relations
- Représentation des données sous la forme de tables constituées de lignes et de colonnes
- Dominent le marché des SGBD

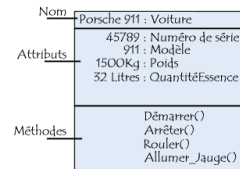


19

Bases de Données - 2015-2016 - Malek RAHOUAL

Les systèmes objets

- Ils sont fondés sur le modèle à objets qui permet de représenter beaucoup de situations du monde réel, par la richesse des concepts qu'il propose.
- Le modèle est apparu dans les années 1980, et les premiers systèmes sont apparus dans les années 1990.



20

Bases de Données - 2015-2016 - Malek RAHOUAL

Le Modèle Relationnel

- Le modèle relationnel a été inventé par E. F. Codd en 1970. Il se fonde sur le concept mathématique de relation.



- **Définition** : Un domaine est un ensemble de valeurs.
 - Exemple :
 - domaine des entiers (domaine infini)
 - domaine de chaînes de caractères

21

Bases de Données - 2015-2016 - Malek RAHOUAL

Le Modèle Relationnel

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

- **Exemple** :

- Soit $D1 = \{\text{France, Royaume Uni, Argentine}\}$ et $D2 = \{\text{Euro, Livre, Peso}\}$.
- La relation PAYS :

Pays	NomPays	Monnaie
	France	Euro
	Royaume Uni	Livre
	Argentine	Peso

22

Bases de Données - 2015-2016 - Malek RAHOUAL

L'unicité de clé

- **Définition** : Une **clé de la relation** est un ensemble minimal d'attributs dont la connaissance des valeurs permet d'identifier un tuple unique de la relation considérée.
- Plus formellement, une clé d'une relation R est un ensemble d'attributs X tel que quels que soient les tuples t1 et t2 d'une instance de R, $t1(X) \neq t2(X)$.
- **Toute relation possède au moins une clé.**
 - S'il existe plusieurs clés, on en choisit une que l'on appelle clé primaire.
- **Exemple** :
 - L'attribut NomPays est la clé de la relation PAYS (NomPays, Monnaie).

23

Bases de Données - 2015-2016 - Malek RAHOUAL

Schéma de relation

- **Attribut** : colonne d'une relation caractérisée par un nom.
 - NomPays, Monnaie
- Le schéma de relation est composé du nom de la relation suivi de la liste des attributs et de la définition de leurs domaines.
 - Un schéma de relation est noté sous la forme :
 - $R (A1 : D1, A2 : D2, \dots, An : Dn)$

24

Bases de Données - 2015-2016 - Malek RAHOUAL

Schéma de relation

- **Exemple :**
 - PAYS (NomPays : VARCHAR2(25), Monnaie : VARCHAR2(25) , Population : INTEGER)
- Le **schéma** d'une relation représente son intention.
- Une **table** représente une extension d'une relation.

25

Bases de Données - 2015-2016 - Malek RAHOUAL

Langage de requêtes

- **Structured Query Language (SQL)**
 - Langage de requêtes prédominant au niveau applicatif
 - Déclaratif
- **Algèbre relationnelle**
 - Langage intermédiaire utilisé sans SGBD
 - Procédural

26

Bases de Données - 2015-2016 - Malek RAHOUAL

ALGÈBRE RELATIONNELLE

27

Bases de Données - 2015-2016 - Malek RAHOUAL

1. Introduction

L'algèbre relationnelle est le support mathématique cohérent sur lequel repose le modèle relationnel. L'algèbre relationnelle propose un ensemble d'opérations élémentaires formelles sur les relations dans le but de créer de nouvelles relations. Ces opérations permettent de représenter des requêtes sur la base de données dont le résultat s'exprime sous la forme d'une relation (i.e. table). C'est ce formalisme qui est au cœur du langage de requête de SQL.

Nous pouvons distinguer trois familles d'opérateurs relationnels :

- **Les opérateurs unaires** (la sélection et la projection), qui sont les plus simples, permettent de produire une nouvelle table à partir d'une autre table.
- **Les opérateurs binaires ensemblistes** (l'union, l'intersection et la différence) permettent de produire une nouvelle relation à partir de deux relations de même degré et de même domaine.
- **Les opérateurs binaires ou n-aires** (le produit cartésien, la jointure et la division) permettent de produire une nouvelle table à partir de deux ou plusieurs autres tables.

Remarque : Les notations de l'algèbre relationnelle ne sont pas standardisées.

28

Bases de Données - 2015-2016 - Malek RAHOUAL

2. Sélection

Soit la relation PERSONNE suivante :

PERSONNE		
idPersonne	nom	prénom
5	Durand	Caroline
1	Germain	Stan
12	Dupont	Lisa
3	Germain	Rose-Marie

Ci-dessous un exemple de sélection $\sigma_{(idPersonne \geq 5)}$ PERSONNE sur la relation PERSONNE :

$\sigma_{(idPersonne \geq 5)}$ PERSONNE		
idPersonne	nom	prénom
5	Durand	Caroline
12	Dupont	Lisa

27

Bases de Données - 2015-2016 - Malek RAHOUAL

Définition : La *sélection* génère une relation regroupant exclusivement toutes les occurrences de la relation R qui satisfont l'expression logique E. **Notation :** $\sigma_E R$.

Il s'agit d'une opération unaire essentielle dont la signature est :

relation \times expression logique \rightarrow relation

La sélection permet ainsi de choisir (i.e. sélectionner) certaines lignes dans une table. Le résultat de la sélection est donc une nouvelle relation qui a les mêmes attributs que R. Si R est vide (c'est-à-dire sans aucune occurrence), la relation qui résulte de la sélection est vide.

28

Bases de Données - 2015-2016 - Malek RAHOUAL

3. Projection

Définition : La *projection* consiste à supprimer les attributs autres que A_1, A_2, \dots, A_n d'une relation et à éliminer les n-uplets en double apparaissant dans la nouvelle version.

Notation : $\Pi_{(A_1, A_2, \dots, A_n)} R$

Il s'agit d'une opération unaire essentielle dont la signature est :

relation \times liste d'attributs \rightarrow relation

En d'autres termes, la projection permet de choisir des colonnes dans une table. Si R est vide, la relation qui résulte de la projection est vide, mais pas forcément équivalente étant donné qu'elle contient généralement moins d'attributs.

Exemple de projection sur la table PERSONNE :

$\Pi_{(nom)} PERSONNE$	
nom	
Durand	
Germain	
Dupont	

Bases de Données - 2015-2016 - Malek RAHOUAL

4. Union

Définition : L'*union* est une opération portant sur deux relations R_1 et R_2 ayant le même schéma et construisant une troisième relation constituée des n-uplets appartenant à l'une ou l'autre des deux relations R_1 et R_2 sans doublon. **Notation :** $R_1 \cup R_2$

Il s'agit d'une opération binaire ensembliste commutative essentielle dont la signature est :

relation \times relation \rightarrow relation

Bases de Données - 2015-2016 - Malek RAHOUAL

Comme nous l'avons déjà dit, R_1 et R_2 doivent avoir les mêmes attributs et si une même occurrence existe dans R_1 et R_2 , elle n'apparaît qu'une seule fois dans le résultat de l'union. Le résultat de l'union est une nouvelle relation qui a les mêmes attributs que R_1 et R_2 . Si R_1 et R_2 sont vides, la relation qui résulte de l'union est vide. Si R_1 (respectivement R_2) est vide, la relation qui résulte de l'union est identique à R_2 (respectivement R_1).

Ci-dessous un exemple d'union :

R_1		R_2		$R_1 \cup R_2$	
nom	prénom	nom	prénom	nom	prénom
Durand	Caroline	Dupont	Lisa	Durand	Caroline
Germain	Stan	Juny	Carole	Germain	Stan
Dupont	Lisa	Fourt	Lisa	Dupont	Lisa
Germain	Rose-Marie			Germain	Rose-Marie
				Juny	Carole
				Fourt	Lisa

Bases de Données - 2015-2016 - Malek RAHOUAL

5. Intersection

Définition : L'*intersection* est une opération portant sur deux relations R_1 et R_2 ayant le même schéma et construisant une troisième relation dont les n-uplets sont constitués de ceux appartenant aux deux relations. **Notation :** $R_1 \cap R_2$

Il s'agit d'une opération binaire ensembliste commutative dont la signature est :

relation \times relation \rightarrow relation

Comme nous l'avons déjà dit, R_1 et R_2 doivent avoir les mêmes attributs. Le résultat de l'intersection est une nouvelle relation qui a les mêmes attributs que R_1 et R_2 . Si R_1 ou R_2 ou les deux sont vides, la relation qui résulte de l'intersection est vide.

Ci-dessous un exemple d'intersection :

R_1		R_2		$R_1 \cap R_2$	
nom	prénom	nom	prénom	nom	prénom
Durand	Caroline	Dupont	Lisa	Durand	Caroline
Germain	Stan	Juny	Carole	Dupont	Lisa
Dupont	Lisa	Fourt	Lisa	Juny	Carole
Germain	Rose-Marie	Durand	Caroline		
Juny	Carole				

Bases de Données - 2015-2016 - Malek RAHOUAL

6. Différence

Définition : La *différence* est une opération portant sur deux relations R_1 et R_2 ayant le même schéma et construisant une troisième relation dont les n-uplets sont constitués de ceux ne se trouvant que dans la relation R_1 . **Notation :** $R_1 - R_2$

Il s'agit d'une opération binaire ensembliste non commutative essentielle dont la signature est :

relation \times relation \rightarrow relation

Comme nous l'avons déjà dit, R_1 et R_2 doivent avoir les mêmes attributs. Le résultat de la différence est une nouvelle relation qui a les mêmes attributs que R_1 et R_2 . Si R_1 est vide, la relation qui résulte de la différence est vide aussi. Si R_2 est vide, la relation qui résulte de la différence est identique à R_1 .

Bases de Données - 2015-2016 - Malek RAHOUAL

Exemple de différence entre deux relations :

R_1		R_2		$R_1 - R_2$	
nom	prénom	nom	prénom	nom	prénom
Durand	Caroline	Dupont	Lisa	Germain	Stan
Germain	Stan	Juny	Carole	Germain	Rose-Marie
Dupont	Lisa	Fourt	Lisa		
Germain	Rose-Marie	Durand	Caroline		
Juny	Carole				

Bases de Données - 2015-2016 - Malek RAHOUAL

7. Produit cartésien

Définition : Le produit cartésien est une opération portant sur deux relations R_1 et R_2 et qui construit une troisième relation regroupant exclusivement toutes les possibilités de combinaison des occurrences des relations R_1 et R_2 . **Notation :** $R_1 \times R_2$.

Il s'agit d'une opération binaire commutative essentielle dont la signature est :

$$\text{relation} \times \text{relation} \rightarrow \text{relation}$$

Le résultat du produit cartésien est une nouvelle relation qui a tous les attributs de R_1 et tous ceux de R_2 . Si R_1 ou R_2 ou les deux sont vides, la relation qui résulte du produit cartésien est vide. Le nombre d'occurrences de la relation qui résulte du produit cartésien est le nombre d'occurrences de R_1 multiplié par le nombre d'occurrences de R_2 .

Exemple :

Bases de Données - 2015-2016 - Malek RAHOUAL

PERSONNE		CADEAU		PERSONNE x CADEAU			
nom	prénom	article	prix	nom	prénom	article	prix
Fourt	Lisa	livre	45	Fourt	Lisa	livre	45
Fourt	Lisa	poupée	25	Fourt	Lisa	poupée	25
Juny	Carole	montre	87	Fourt	Lisa	montre	87
				Juny	Carole	livre	45
				Juny	Carole	poupée	25
				Juny	Carole	montre	87

Bases de Données - 2015-2016 - Malek RAHOUAL

8. Jointure, théta-jointure, jointure naturelle

➤ Jointure

Définition : La jointure est une opération portant sur deux relations R_1 et R_2 qui construit une troisième relation regroupant exclusivement toutes les possibilités de combinaison des occurrences des relations R_1 et R_2 qui satisfont l'expression logique E. La jointure est notée : $R_1 \bowtie_E R_2$.

Il s'agit d'une opération binaire commutative dont la signature est :

$$\text{relation} \times \text{relation} \times \text{expression logique} \rightarrow \text{relation}$$

Si R_1 ou R_2 ou les deux sont vides, alors la relation qui résulte de la jointure est vide. En fait, la jointure n'est rien d'autre qu'un produit cartésien suivi d'une sélection :

$$R_1 \bowtie_E R_2 = \sigma_E (R_1 \times R_2)$$

Exemple de jointure :

PERSONNE			CADEAU		
nom	prénom	Age	âgeC	article	prix
Fourt	Lisa	6	99	livre	30
Juny	Carole	42	6	poupée	60
Fidus	Laure	16	20	baladeur	45
			10	déguisement	15

Bases de Données - 2015-2016 - Malek RAHOUAL

		PERSONNE $\bowtie_{(\text{âge} \leq \text{âgeC}) \wedge (\text{prix} \leq 50)}$		CADEAU	
nom	prénom	âge	âgeC	article	prix
Fourt	Lisa	6	99	livre	30
Fourt	Lisa	6	20	baladeur	45
Fourt	Lisa	6	10	déguisement	15
Juny	Carole	42	99	Livre	30
Juny	Carole	42	99	Livre	30
Juny	Carole	42	20	baladeur	45

Bases de Données - 2015-2016 - Malek RAHOUAL

Cette jointure permet de générer toutes les possibilités d'association entre un cadeau et une personne en respectant l'âge maximum conseillé pour un cadeau et la somme de 50€ à ne pas dépasser.

Bases de Données - 2015-2016 - Malek RAHOUAL

➤ Thêta-jointure

Définition : La thêta-jointure est une jointure dans laquelle l'expression logique E est une simple comparaison entre un attribut A_1 de la relation R_1 et un attribut A_2 de la relation R_2 . La thêta-jointure est notée $R_1 \bowtie_{A_1=A_2} R_2$.

➤ Équi-jointure

Définition : Une équi-jointure est une thêta-jointure dans laquelle l'expression logique E est un test d'égalité entre un attribut A_1 de la relation R_1 et un attribut A_2 de la relation R_2 . L'équi-jointure est notée $R_1 \bowtie_{A_1=A_2} R_2$.

Remarque : Il vaut mieux écrire $R_1 \bowtie_{A_1=A_2} R_2$ que $R_1 \bowtie_{A_1, A_2} R_2$, car cette dernière notation, bien que parfois dans la littérature, prête à confusion avec une jointure naturelle explicite.

Bases de Données - 2015-2016 - Malek RAHOUAL

➤ Jointure naturelle

Définition : Une jointure naturelle est une jointure dans laquelle l'expression logique E est un test d'égalité entre les attributs qui portent le même nom dans les relations R₁ et R₂. Dans la relation construite, ces attributs ne sont pas dupliqués, mais fusionnés en une seule colonne par couple d'attributs. La jointure naturelle est notée R₁ ⋈ R₂. Si la jointure ne doit porter que sur un sous-ensemble des attributs communs à R₁ et R₂ il faut préciser explicitement ces attributs de la manière suivante : R₁ ⋈_{A1,...,An} R₂.

Généralement, R₁ et R₂ n'ont qu'un attribut en commun. Dans ce cas, une jointure naturelle est équivalente à une équi-jointure dans laquelle l'attribut de R₁ et celui de R₂ sont justement les deux attributs qui portent le même nom.

Pour effectuer une jointure naturelle entre R₁ et R₂ sur un attribut A1 commun à R₁ et R₂, il vaut mieux écrire R₁ ⋈_{A1} R₂ que R₁ ⋈ R₂. En effet, si R₁ et R₂ possèdent deux attributs portant un nom commun, A1 et A2, R₁ ⋈_{A1} R₂ est bien une jointure naturelle sur l'attribut A1, mais R₁ ⋈ R₂ est une jointure naturelle sur le couple d'attributs A1, A2, ce qui produit un résultat très différent !

PERSONNE			CADEAU		
nom	prénom	âge	âge	article	prix
Fourt	Lisa	6	40	livre	45
Juny	Carole	40	6	poupée	25
Fidus	Laure	20	20	montre	87
Choupy	Emma	6			

Ci-dessous la jointure naturelle PERSONNE ⋈ CADEAU qui peut également s'écrire PERSONNE ⋈_{âge} CADEAU.

PERSONNE ⋈ CADEAU					
nom	prénom	âge	article	prix	
Fourt	Lisa	6	poupée	25	
Juny	Carole	40	livre	45	
Fidus	Laure	20	montre	87	
Choupy	Emma	6	poupée	25	

9. Division

Définition : la division est une opération portant sur deux relations R₁ et R₂, telles que le schéma de R₂ est strictement inclus dans celui de R₁, qui génère une troisième relation regroupant toutes les parties d'occurrences de la relation R₁ qui, associées à toutes les occurrences de la relation R₂, se retrouvent dans R₁. **Notation :** R₁ ÷ R₂

Il s'agit d'une opération binaire non commutative dont la signature est :

$$\text{relation} \times \text{relation} \rightarrow \text{relation}$$

Autrement dit, la division de R₁ par R₂ (R₁ ÷ R₂) génère une relation qui regroupe tous les n-uplets qui, concaténés à chacun des n-uplets de R₂, donne toujours un n-uplet de R₁.

La relation R₂ ne peut pas être vide. Tous les attributs de R₂ doivent être présents dans R₁ et R₂ doit posséder au moins un attribut de plus que R₂ (inclusion stricte). Le résultat de la division est une nouvelle relation qui a tous les attributs de R₁ sans aucun de ceux de R₂. Si R₁ est vide, la relation qui résulte de la division est vide.

Ci-dessous, un exemple de division ENSEIGNEMENT ÷ ETUDIANT qui permet de dresser la table R de tous les enseignants de la relation ENSEIGNEMENT qui enseignent à tous les étudiants de la relation ETUDIANT.

ENSEIGNEMENT		ETUDIANT
enseignant	nom	nom
Germain	Dubois	Dubois
Fidus	Pascal	Pascal
Robert	Dubois	
Germain	Pascal	
Fidus	Dubois	
Germain	Durand	
Robert	Durand	

ENSEIGNEMENT ÷ ETUDIANT	
enseignant	
Germain	
Fidus	

3. Le langage SQL

Définition

• **Structured Query Language (SQL) :**

- Ensemble de commandes utilisées par les utilisateurs et les programmes pour
 - définir,
 - rechercher
 - ou modifier
 des données dans une base de données relationnelle.

➤ C'est le langage standard des SGBDR.

Définition

- Le langage SQL se divise en trois parties :
 - ☞ Le **Data Manipulation Language (DML)** : langage d'interrogation et de manipulation de données (ordres SELECT, UPDATE, INSERT, DELETE)
 - ☞ Le **Data Definition Language (DDL)** : langage de définition des données (ordres CREATE, ALTER, DROP)
 - ☞ Le **Data Control Language (DCL)** : langage de contrôle de l'accès aux données (ordres GRANT, REVOKE)

49

Bases de Données - 2015-2016 - Malek RAHOUAL

Les types de données de SQL

- Type Caractère : CHAR
 - ☞ permet de stocker une chaîne de caractères de longueur fixe.
 - CHAR [(longueur)]
 - ☞ Par défaut longueur est égale à 1, maximale: 2000.
- Type Caractère : VARCHAR2
 - ☞ permet de stocker une chaîne de caractères de longueur variable.
 - VARCHAR2(longueur_max)
 - ☞ Longueur maximale: 4000 caractères
 - (VARCHAR2 remplace l'ancien type VARCHAR dans les SGBD Oracle)

50

Bases de Données - 2015-2016 - Malek RAHOUAL

Les types de données de SQL

- Type numérique : NUMBER
 - ☞ permet de représenter les entiers, les nombres décimaux et les nombres en virgule flottante.
 - NUMBER [(précision [, échelle])]
 - ☞ précision : nombre de chiffres significatifs
 - ☞ échelle : nombre de chiffres à droite de la marque décimale
 - ☞ La valeur 123456.78 est de type NUMBER (8 , 2).
- Type date : DATE
 - ☞ Formats : '25-OCT-05', '25 oct 2005', '25/OCT/2005', '25/10/2005', '25-10-2005', '25-OCT-2005'

51

Bases de Données - 2015-2016 - Malek RAHOUAL

Syntaxe d'une requête de sélection

- La commande **SELECT** permet d'effectuer les opérations suivantes :
 - ☞ Projection,
 - ☞ Restriction,
 - ☞ Jointure, ...
- Syntaxe générale :

```
SELECT ...  
FROM ...  
WHERE ...  
GROUP BY ...  
HAVING ...  
ORDER BY
```

} optionnelles
- L'ordre des clauses est imposé

52

Bases de Données - 2015-2016 - Malek RAHOUAL

Recherche de données

- Projection

```
SELECT [DISTINCT] {*, attribut , ... }  
FROM table ;
```

 - ☞ « * » signifie que toutes les colonnes de la table sont sélectionnées.
 - ☞ **DISTINCT** permet de supprimer les duplications.

53

Bases de Données - 2015-2016 - Malek RAHOUAL

Recherche de données : Exemple

- Soit la base de données de gestion des réservations de places de spectacles pour un théâtre:

```
Spectacle (NumSpec, NomSpec, DatDebSpec, DatFinSpec)  
Représentation (NumRep, NumSpec, DatRep, HeureRep)  
Place (NumRep, NumPlace, Prix)  
Réservation (NumRes, NumRep, NumPlace, NomDem, TelDem)
```

```
SELECT NomSpec  
FROM Spectacle ;
```
- Cette requête permet d'obtenir le nom de tous les spectacles figurant dans la base de données.

54

Bases de Données - 2015-2016 - Malek RAHOUAL

Recherche de données

● Restriction

```
SELECT *  
FROM table  
WHERE condition ;
```

● Exemple:

```
SELECT DISTINCT NomSpec  
FROM Spectacle  
WHERE DatDebSpec = '02-jan-2006';
```

55

Bases de Données - 2015-2016 - Malek RAHOUAL

Prédicats de sélection

● Prédicat de comparaison à l'aide des opérateurs : <, >, <=, >=, =, != ou <>.

● Prédicat d'intervalle BETWEEN AND :

- expr BETWEEN expr1 AND expr2
- teste si la valeur d'une expression est comprise entre deux valeurs.
 - Prix BETWEEN 100 AND 200

● Comparaison de chaînes de caractères LIKE :

- expr LIKE chaîne,
- où chaîne peut contenir le caractère % qui remplace une chaîne de caractères de longueur quelconque ou le caractère _ qui remplace un seul caractère.
 - NomSpec LIKE 'Rev%'

● Prédicat de test IS (NOT) NULL

- vérifie si une valeur est connue/inconnue.
 - TelDem IS NOT NULL)

56

Bases de Données - 2015-2016 - Malek RAHOUAL

Prédicats de sélection

● Prédicat d'appartenance IN : expr IN (exp1, exp2, ...)

- vrai si la valeur de expr est égale à l'une des celles de exp1, exp2, ...
 - NumRep IN (1, 3, 13)

● Prédicat EXISTS

- retourne une valeur vraie s'il existe au moins une ligne satisfaisant les conditions de la sous-interrogation

● Prédicat ANY ou ALL

- doit être précédé par un opérateur de comparaison et retourne une valeur vraie en fonction de l'opérateur de comparaison utilisé et des tuples résultant de la sous-requête exprimée

● Les prédicats peuvent être composés de plusieurs conditions grâce aux opérateurs logiques : NOT, AND, OR.

57

Bases de Données - 2015-2016 - Malek RAHOUAL

Prédicats de sélection : exemples

● Toutes les places qui coûtent moins de 100 francs pour la représentation N°7:

```
SELECT *  
FROM Place  
WHERE Prix <= 100 AND NumRep = 7;
```

● Toutes les représentations du spectacle N°13 entre le 15 septembre 2005 et le 15 janvier 2006 ou entre le 15 février 2006 et le 15 mai 2006:

```
SELECT *  
FROM Représentation  
WHERE DatRep BETWEEN '15-SEP-2005' AND '15-JAN-2006'  
OR DatRep BETWEEN '15-FEV-2006' AND '15-MAI-2006'  
AND NumSpec = 13;
```

58

Bases de Données - 2015-2016 - Malek RAHOUAL

Jointure

● Une jointure permet de sélectionner des données de plusieurs tables.

```
SELECT [table1.]attribut [, table2.attribut, ...]  
FROM table1, table2 [, table3,...]  
WHERE conditions ;
```

● Il est nécessaire de préciser le nom de la table dans laquelle on sélectionne un attribut si le même attribut est présent dans d'autres tables participant à cette jointure.

59

Bases de Données - 2015-2016 - Malek RAHOUAL

Jointure : Exemple

● Toutes les représentations du spectacle " Révisor " entre le 15 septembre 2005 et le 15 janvier 2006 :

```
SELECT DatRep, HeurRep  
FROM Représentation, Spectacle  
WHERE DatRep BETWEEN '15-SEP-05' AND '15-JAN-06'  
AND NumSpec = 'Révisor'  
AND Spectacle.NumSpec = Représentation.NumSpec ;
```

● Remarque : on peut remplacer les noms des tables par les alias plus courts :

```
SELECT DatRep, HeurRep  
FROM Représentation R, Spectacle S  
WHERE DatRep BETWEEN '15-SEP-05' AND '15-JAN-06'  
AND NumSpec = 'Révisor'  
AND S.NumSpec = R.NumSpec ;
```

60

Bases de Données - 2015-2016 - Malek RAHOUAL

Tri des données

- Pour trier/ordonner les données résultant d'une requête on utilise la clause **ORDER BY**.

```
SELECT [DISTINCT] {*, attribut , ... }  
FROM table  
[ WHERE conditions ]  
[ ORDER BY { colonne, expr } [ASC | DESC] ];
```

- où :
 - ORDER BY précise que le résultat sera trié
 - ASC ordre croissant (ordre par défaut)
 - DESC ordre décroissant

61

Bases de Données - 2015-2016 - Malek RAHOUAL

Tri des données : Exemple

- La date et l'heure de toutes les représentations du spectacle "Révisor" qui sont programmées entre le 15 septembre 2005 et le 15 janvier 2006 :

```
SELECT DatRep, HeureRep  
FROM Représentation, Spectacle  
WHERE DatRep BETWEEN '15-SEP-05' AND '15-JAN-06'  
AND NomSpec = 'Révisor'  
AND Spectacle.NumSpec = Représentation.NumSpec  
ORDER BY DatRep ;
```

62

Bases de Données - 2015-2016 - Malek RAHOUAL

Groupement des données

- Les fonctions d'agrégats :
 - AVG(expression) : moyenne des valeurs d'une colonne
 - SUM(expression) : somme des valeurs d'une colonne
 - MIN(expression) : la plus petite des valeurs d'une colonne
 - MAX(expression) : la plus grande des valeurs d'une colonne
 - COUNT(expression) : compte le nombre de valeurs

- Exemple :
 - La moyenne des prix des places de toutes les représentations

```
SELECT AVG(Prix)  
FROM Place ;
```

63

Bases de Données - 2015-2016 - Malek RAHOUAL

Groupement des données

- On appelle **groupe** un ensemble de lignes, résultat d'une requête, qui ont une valeur commune dans une ou plusieurs colonnes.
- La clause **GROUP BY** permet de diviser les données d'une table en plusieurs groupes.

```
SELECT attribut , fonction_d'agrégat(attribut)  
FROM table [, table ... ]  
[ WHERE conditions ]  
[ GROUP BY expr [, expr ... ] ]  
[ ORDER BY attribut ];
```

- L'utilisation de la clause WHERE permet d'exclure certaines lignes du résultat avant de les diviser en groupes.

64

Bases de Données - 2015-2016 - Malek RAHOUAL

Groupement des données : Exemple

- La moyenne des prix des places pour chaque représentation :

```
SELECT AVG(Prix), NumRep  
FROM Place  
GROUP BY NumRep;
```

65

Bases de Données - 2015-2016 - Malek RAHOUAL

Groupement des données

- Pour spécifier quels groupes doivent être sélectionnés dans le résultat on utilise la clause **HAVING**.

```
SELECT attribut , fonction_d'agrégat(attribut)  
FROM table [, table ... ]  
[ WHERE conditions ]  
[ GROUP BY expr [, expr ... ] ]  
[ HAVING condition_de_groupe ]  
[ ORDER BY attribut ];
```

66

Bases de Données - 2015-2016 - Malek RAHOUAL

Groupement des données

- Pour restreindre les groupes :

```
SQL> SELECT AVG(Prix), NumRep
2 FROM Place
3 WHERE AVG(Prix) < 200
4 GROUP BY NumRep ;
WHERE AVG(Prix) < 200
*
```

ERREUR à la ligne 3:
ORA-00934: group function is not allowed here

- La requête précédente peut être réécrite de la manière suivante :

```
SELECT AVG(Prix), NumRep
FROM Place
GROUP BY NumRep
HAVING AVG(Prix) < 200;
```

Condition sur les groupes

67

Bases de Données - 2015-2016 - Malek RAHOUAL

Les sous-requêtes

- L'expression d'une clause WHERE peut prendre la forme :

... WHERE attribut opérateur (SELECT ...);

- Exemple :

Rechercher toutes les personnes ayant réservé une place pour la dernière représentation de l'année 2005.

Pour cela il est nécessaire de connaître la date de la dernière représentation, ce qui peut être exprimé par la requête suivante :

```
SELECT MAX(DatRep)
FROM Représentation
WHERE DatRep <= '31-12-2005';
```

68

Bases de Données - 2015-2016 - Malek RAHOUAL

Les sous-requêtes

- Voici la requête complète qui utilise la requête précédente comme une **sous-requête** :

```
SELECT NomDem
FROM Réservation Rs, Représentation Rp
WHERE Rp.NumRep = Rs.NumRep
AND Rp.DatRep = ( SELECT MAX(DatRep)
FROM Représentation
WHERE DatRep <= '31-12-2005');
```

- Plusieurs types de sous-requêtes :

- Sous-requête mono-ligne
- Sous-requête multi-ligne
- Sous-requête multi-colonne

69

Bases de Données - 2015-2016 - Malek RAHOUAL

Les sous-requêtes

- Une sous-requête **mono-ligne** doit être utilisée avec les opérateurs de comparaison tels que =, >, >=, <, <=, <>.

- Exemple : l'exemple précédent est une requête mono-ligne.

- Une sous-requête **multi-ligne** doit être utilisée avec les opérateurs pouvant traiter plusieurs lignes : IN, ALL, ANY :

- **Opérateur IN** : WHERE attribut IN (liste de valeurs)
- **Opérateur ALL** : WHERE attribut opér_compar ALL (liste de valeurs)
- **Opérateur ANY** : WHERE attribut opér_compar ANY (liste de valeurs)

70

Bases de Données - 2015-2016 - Malek RAHOUAL

Les sous-requêtes : Exemple

Les personnes ayant réservé le plus de places :

```
SELECT NomDem
FROM Réservation
GROUP BY NomDem, TelDem
HAVING COUNT (NomDem) = (SELECT MAX(COUNT(NomDem))
FROM Réservation
GROUP BY NomDem, TelDem);
```

C'est la clé de la table réservation

Sous-requête

71

Bases de Données - 2015-2016 - Malek RAHOUAL

Les sous-requêtes

- Une sous-requête **multi-colonnes** doit utiliser l'opérateur IN permettant de comparer les lignes colonne par colonne.

- Exemple :

Toutes les personnes ayant réservé des places à la fois pour la représentation N°1 et pour la représentation N°2.

```
SELECT DISTINCT NomDem, TelDem
FROM Réservation
WHERE NumRep = 1
AND (NomDem, TelDem) IN ( SELECT NomDem, TelDem
FROM Réservation
WHERE NumRep = 2
);
```

72

Bases de Données - 2015-2016 - Malek RAHOUAL

Les opérateurs UNION, INTERSECT, MINUS

- Pour exprimer les opérateurs ensemblistes de l'algèbre relationnelle tels que : UNION, INTERSECTION et DIFFERENCE le langage SQL offre les opérateurs UNION , INTERSECT et MINUS.

```
SELECT attribut,...  
FROM table, ...  
[ WHERE condition ]
```

opérateur

```
SELECT attribut,...  
FROM table, ...  
[ WHERE condition ]
```

73

Bases de Données - 2015-2016 - Malek RAHOUAL

Les opérateurs UNION, INTERSECT, MINUS

- Exemples :

Toutes les personnes ayant réservé des places à la fois pour la représentation N°1 et la représentation N°2 peut être exprimée de la façon suivante :

```
SELECT DISTINCT NomDem, TelDem
```

```
FROM Réservation
```

```
WHERE NumRep = 1
```

INTERSECT

```
SELECT DISTINCT NomDem, TelDem
```

```
FROM Réservation
```

```
WHERE NumRep = 2 ;
```

74

Bases de Données - 2015-2016 - Malek RAHOUAL

Création et gestion des tables

- Création d'une table

```
CREATE TABLE table  
(colonne type [DEFAULT expr]  
[,colonne type [DEFAULT expr]]);
```

↳ Où DEFAULT expr : la valeur par défaut de l'attribut.

- Exemple :

```
CREATE TABLE Spectacle  
( NumSpec NUMBER,  
  NomSpec VARCHAR2(20),  
  DatDebSpec DATE,  
  DatFinSpec DATE DEFAULT '01-12-05' );
```

75

Bases de Données - 2015-2016 - Malek RAHOUAL

Création et gestion des tables

- Création d'une table avec insertion de données:

↳ Se fait par l'utilisation d'une **sous-requête**

```
CREATE TABLE table  
(attribut [, attribut ])
```

AS sous-requête ;

- Exemples :

```
CREATE TABLE Stat ( Nom, NbRep)  
AS SELECT NomDem, COUNT(NomDem)  
FROM Reservation  
GROUP BY NomDem, TelDem;
```

```
CREATE TABLE CopieSpectacle
```

```
AS SELECT * FROM Spectacle;
```

76

Bases de Données - 2015-2016 - Malek RAHOUAL

Création et gestion des tables

- Modification d'une table

```
ALTER TABLE table  
ADD ( attribut type [DEFAULT expr ]  
[, attribut type ] );
```

↳ Ou

```
ALTER TABLE table  
MODIFY ( attribut type [DEFAULT expr ]  
[, attribut type ] );
```

- Exemple :

```
ALTER TABLE Stat  
MODIFY ( NombreRep NUMBER DEFAULT 0 );
```

77

Bases de Données - 2015-2016 - Malek RAHOUAL

Création et gestion des tables

- Suppression d'une table

```
DROP TABLE table ;
```

- Renommer une table

```
RENAME table1 TO table2;
```

78

Bases de Données - 2015-2016 - Malek RAHOUAL

Manipulation des données

- Le langage de manipulation de données permet de modifier le contenu des objets de la base.
- L'unité de manipulation est une ligne.
- 4 commandes SQL permettent la modification des données :
 - **INSERT** : ajout de lignes
 - **UPDATE** : mise à jour de lignes
 - **DELETE** : suppression des lignes sélectionnées
 - **TRUNCATE** : suppression de toutes les lignes d'une table

79

Bases de Données - 2015-2016 - Malek RAHOUAL

Insertion de lignes - INSERT

- Permet d'ajouter une ligne dans une table en spécifiant les valeurs à insérer.

```
INSERT INTO table [(attribut1 [, attribut2 ])]  
VALUES (valeur1 [, valeur2 ]);
```

- **Exemple** :

```
INSERT INTO Réservation (NumRes, NumRep, NumPlace, NomDem)  
VALUES (23, 1, 126, 'Smith');
```

- Il est possible d'insérer dans une table des lignes provenant d'une **sous-requête** :

```
INSERT INTO table [(attribut1 [, attribut2 ])]  
SELECT ...;
```

80

Bases de Données - 2015-2016 - Malek RAHOUAL

Modification des données - UPDATE

- Permet de modifier les valeurs d'une ou plusieurs colonnes, dans une ou plusieurs lignes existantes d'une table.

```
UPDATE table  
SET attribut1 = expr1 [, attribut2 = expr2]  
[ WHERE condition ] ;
```

- **Exemple** :

```
UPDATE Place  
SET Prix = Prix + 100  
WHERE NumPlace <= 25;
```

81

Bases de Données - 2015-2016 - Malek RAHOUAL

Suppression de lignes - DELETE

- Permet de supprimer certaines lignes d'une table.

```
DELETE FROM table  
[ WHERE condition ]
```

- **Exemple** : Supprimer toutes les réservations du demandeur Smith.

```
DELETE FROM Réservation  
WHERE NomDem = 'Smith' ;
```

82

Bases de Données - 2015-2016 - Malek RAHOUAL

Suppression de lignes - TRUNCATE

- Permet de supprimer toutes les lignes d'une table.

```
TRUNCATE TABLE table  
[ DROP | REUSE STORAGE ]
```

- où les options **DROP STORAGE** et **REUSE STORAGE** permettent de gérer l'espace mémoire :
 - **DROP STORAGE** : après la suppression les blocs mémoire libérés sont réaffectés à la base
 - **REUSE STORAGE** : conserve l'espace mémoire alloué à la table

83

Bases de Données - 2015-2016 - Malek RAHOUAL