



Bases de données

Hamida LAGRAA

Maitre de conférences, Université Lyon 1

Hamida.lagraa@univ-lyon1.fr

R1.05 Introduction aux bases de données

- **1^{ère} année – Semestre 1 :**
 - ✓ **LMD : Initiation aux concepts de bases de données, Modèle relationnel et Langage de Manipulation de Données**
 - ✓ **LDD : Langage de Définition de Données**
 - ✓ **Introduction à la modélisation de données**

- **1^{ère} année – Semestre 2 :**
 - ✓ **Concepts avancés en conception et interrogation de données**
 - ✓ **Administration BD**
 - ✓ **Visualisation de données**

Organisation des cours

R1.05 Introduction aux bases de données

- **Cours :**
 - ✓ **LMD**
 - 2 séances de CM
 - 7 séances TP (2h)
 - ✓ **LDD**
 - 1 séance de CM
 - 3 séances de TP (4h)
 - ✓ **Intro Modélisation**
 - 1 séance CM
 - 2 séances TD (2h)
 - 3 séances TP (2h)
- **Evaluations :**
 - ✓ 3 TPs notés
 - ✓ 1 SAE
- **Enseignants :**
 - ✓ Hamida Lagraa (CM – TP)
 - ✓ Karim Benouaret (TP)
 - ✓ Jean-Pierre Boutin (TP)
- **Espace « IntroductionBD » sur Claroline**

Qu'est ce qu'une Base de Données ?

Qu'est ce qu'une Base de Données

Définition

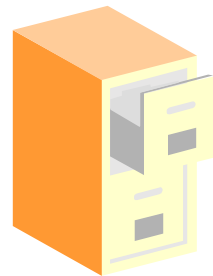
Utilité

Modèle de données

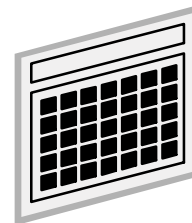
- **Besoin : Stocker des données, y accéder facilement, ne pas les perdre...**

DEPT		
DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

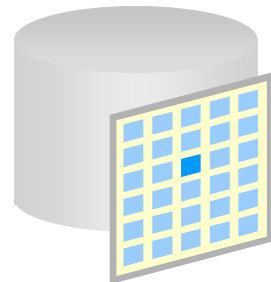
SALGRADE		
GRADE	LOSAL	HISAL
1	700	1200
2	1201	1400
3	1401	2000
4	2001	3000



Fiches de rangement



Tableur excel



Base de Données

Qu'est ce qu'une Base de Données

Définition

Utilité

Modèle de données

- **Besoin** : Stocker des données, y accéder facilement, ne pas les perdre...
 - **Critères pertinents** :
 - ✓ Rapidité et facilité pour stocker des données en grande quantité
 - ✓ Rapidité et simplicité pour récupérer des informations stockées parmi une foule de données
 - ✓ Gestion des conflits lors d'accès concurrentiels aux données
 - ✓ Fiabilité du support de stockage (haute disponibilité)
- ⇒ **Une base de données répond à ces critères**

Qu'est ce qu'une Base de Données

Définition

Utilité

Modèle de données



Définition

- ✓ Une BD est un ensemble **volumineux**, **structuré** et **minimalement redondant** de données, **reliées entre elles**, stockées sur supports numériques centralisés ou distribués, servant pour les besoins d'une ou plusieurs applications, interrogeables et modifiables par un ou plusieurs utilisateurs travaillant potentiellement en parallèle.



Exemples

- ✓ Université :
Gestion des personnels, étudiants, cours, inscriptions, ...
- ✓ Bibliothèque :
Gestion des livres et des prêts des lecteurs
- ✓ SCNF :
Système de réservation des billets
- ✓ ...

Qu'est ce qu'un SGBD?

Définition

Utilité

Modèle de données

□ Définition

- Un SGBD (Système de Gestion de Bases de Données) est un logiciel qui prend en charge la structuration, le stockage, la mise à jour et la maintenance d'une base de données.
- Il est l'unique interface entre les informaticiens et les données (définition des schémas, programmation des applications), ainsi qu'entre les utilisateurs et les données (consultation et mise à jour).

□ Exemple

- Oracle: SGBD relationnel pour applications professionnelles
- MariaDB et PostGres: SGBD relationnels libres
- ACCESS: SGBD relationnel Microsoft pour des applications de petites envergures
- MySQL, ...

Définition

Utilité

Modèle de données

En quoi un simple fichier texte ou Excel n'est-il pas viable pour stocker de l'information ?

- ✓ Difficulté de gestion : l'utilisateur doit connaître la structure pour manipuler les données;
- ✓ Incohérence ou redondance des données;
- ✓ Coûts élevés : un programme par fichier est nécessaire;
- ✓ Maintenance difficile;
- ✓ Gestion des pannes non prise en charge par défaut;
- ✓ Partage des données compliquée

Définition

Utilité

Modèle de données

❑ Modèle de données

- Ensemble de **concepts et de règles de composition de ces concepts** permettant de décrire des données (Gardarin, 1999).
- Un modèle est souvent représenté au moyen **d'un formalisme graphique** permettant de décrire les données (ou plus précisément les types de données) et les relations entre les données.
- On distingue 3 niveaux de modélisation: **Le modèle conceptuel, le modèle logique, le modèle physique**

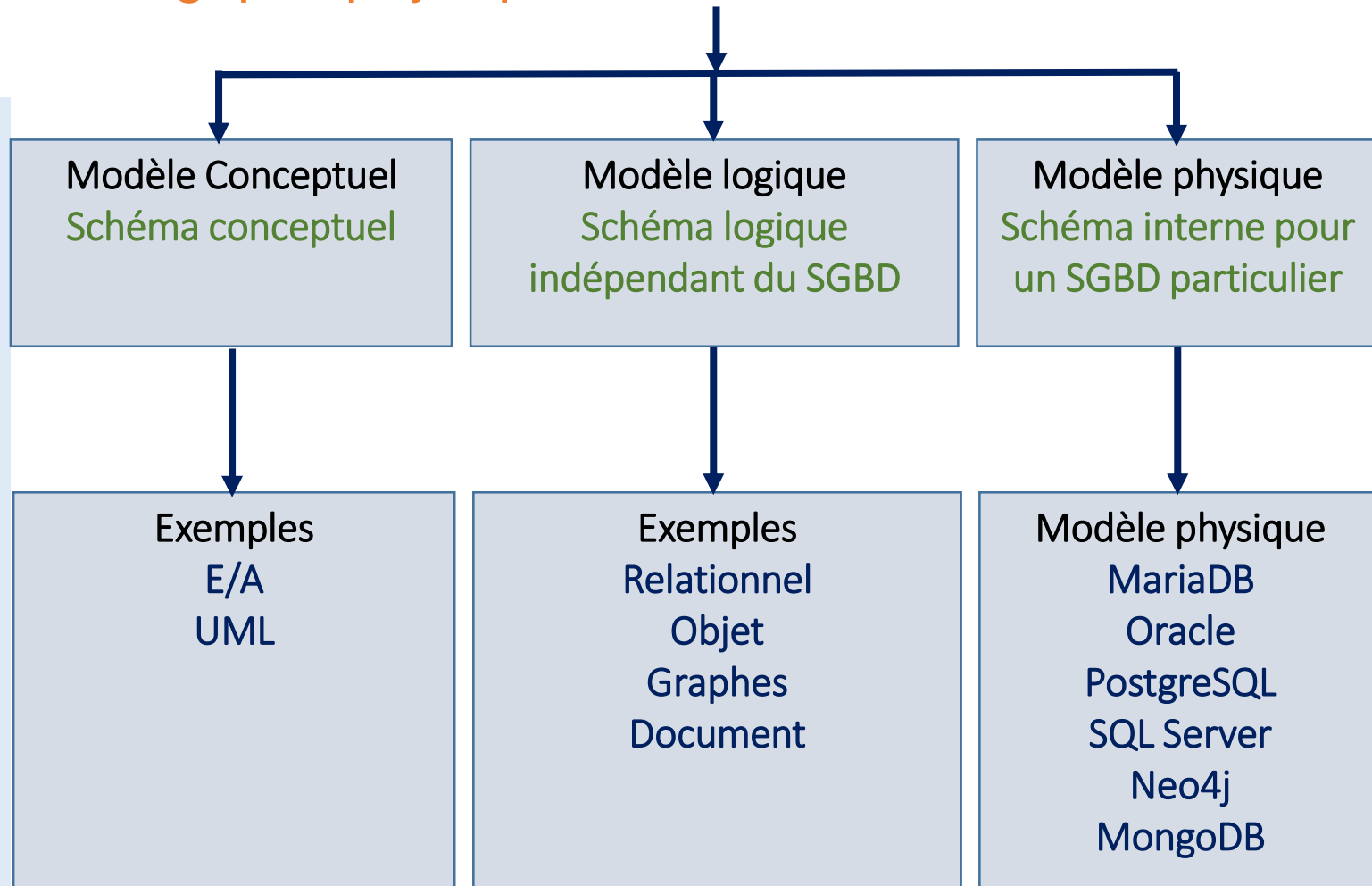
Introduction aux Bases de Données

Définition

Utilité

Modèle de
données

□ Une approche à trois niveaux: conceptuel, logique, physique



Introduction aux Bases de Données

Définition

Utilité

Modèle de données

❑ Processus de conception d'une base de données

Une problématique de gestion de données qui se pose

On utilise outil de modélisation conceptuel

Modéliser

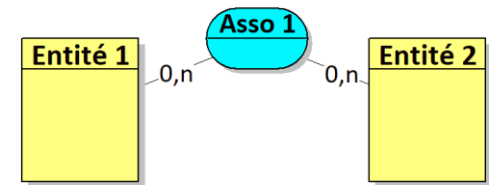


Schéma conceptuel de données

Traduire

On utilise un algorithme (ensemble de règles)

On utilise un SGBD particulier

Implémenter

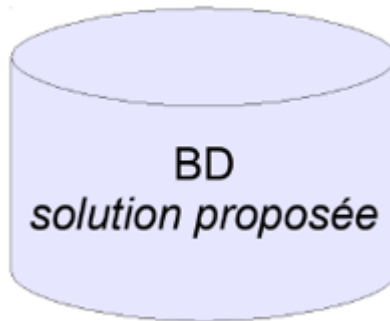


Schéma physique de données

EMP			
EMPNO	int(11)	<pk>	
ENAME	varchar(10)		
JOB	varchar(9)		
MGR	int(11)	<ak1, fk2>	
HIREDATE	date		
SAL	decimal(7,2)		
COMM	decimal(7,2)		
DEPTNO	int(11)	<ak2, fk1>	

Schéma logique de données

DEPT	
DEPTNO	int(11) <pk>
DNAME	varchar(50)
LOC	varchar(50)

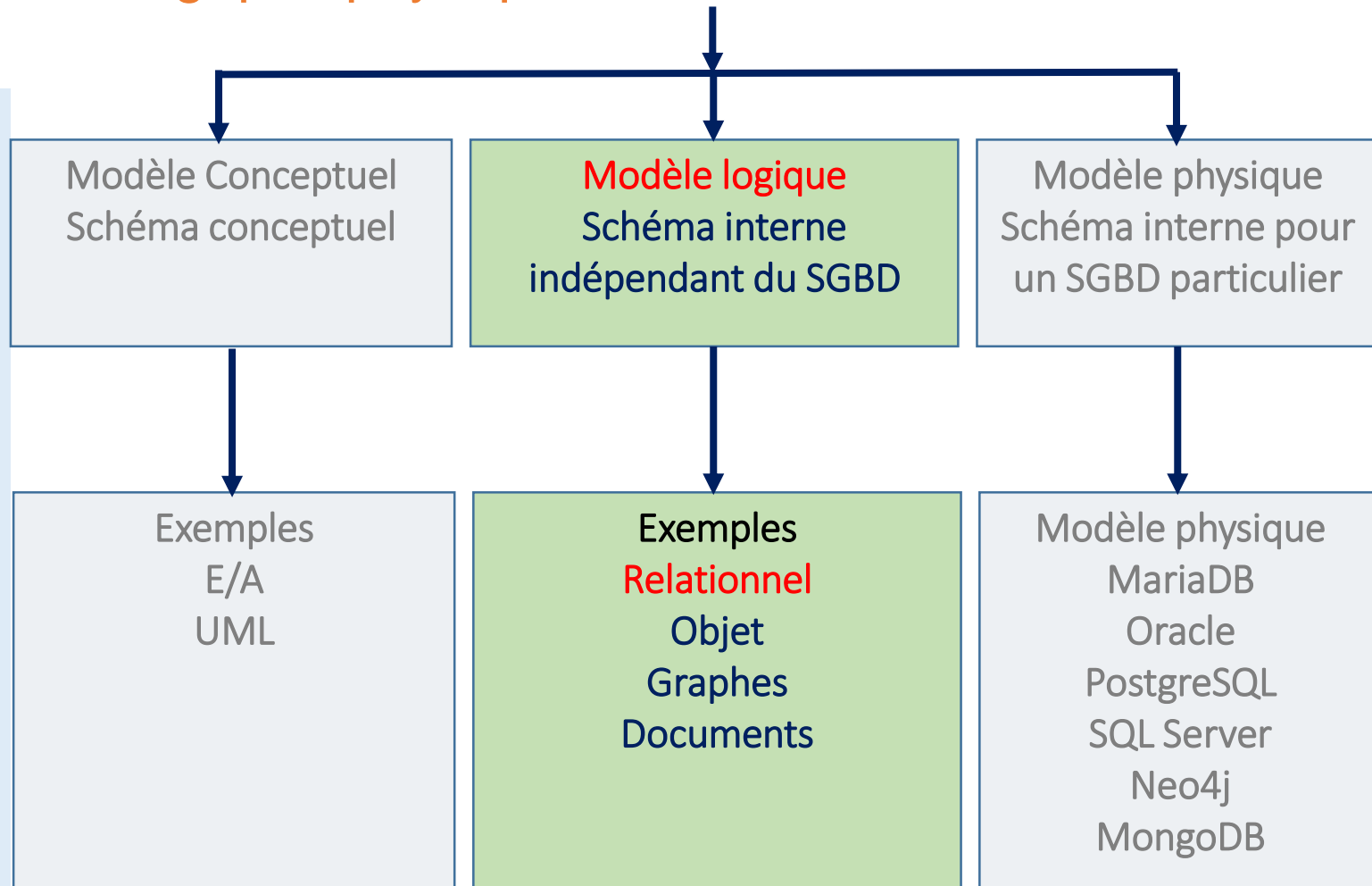
Introduction aux Bases de Données

Définition

Utilité

Modèle de données

❑ Une approche à trois niveaux: conceptuel, logique, physique



Le Modèle Relationnel

Le Modèle Relationnel

Définition

Concepts clés

Domaine

Relation, attribut et tuple

Notion de Clé

Schéma relationnel

- Introduit par *Codd*, en 1970 au laboratoire de recherche d'IBM de San José, pour :
 - ✓ Assurer l'indépendance des applications et de la représentation interne des données
 - ✓ Gérer les problèmes de cohérence et de redondance des données
- **Définition:**
 - ✓ On appelle modèle relationnel un ensemble de concepts permettant de formaliser logiquement la description de données, indépendamment de la façon dont ces données sont physiquement stockées dans une mémoire numérique.

Le Modèle Relationnel

Définition

Concepts clés

Domaine

Relation, attribut et tuple

Notion de Clé

Schéma relationnel

- Le modèle relationnel inclut des concepts pour la **description** de données, ainsi que des concepts pour la **manipulation** de données.
- Ces concepts ont été normalisés par l'ISO dans le langage **SQL**
- Ils sont implémentés par la plupart des SGBDs relationnels : ORACLE, POSTGRE, MYSQL, MARIADB, ...

Le Modèle Relationnel

Définition

Concepts clés

Domaine

Relation, attribut et tuple

Notion de Clé

Schéma relationnel

Représenter le monde en tables

- Les **relations** ou **tables** : des lignes et des colonnes
- Les **domaines** de valeurs (**type**): chaque case d'une table prend une unique valeur dans un domaine pré-défini
- Les **clés** : il existe des cases dont les valeurs doivent être **uniques** et **non nulles**
- Les **clés étrangères** : il existe des cases qui doivent prendre une valeur existante dans les cases d'une autre table

Nom	Prénom	Club
Mbapé	Kylian	PSG
Messi	Lionel	PSG
Benzema	Karim	Real Madrid

Le Modèle Relationnel

Objectifs

Concepts clés

Domaine

Relation,
attribut et tuple

Notion de Clé

Schéma
relationnel

- **Définition:** Un domaine est un ensemble de **valeurs**, caractérisé par un nom, dans lequel des données peuvent prendre leurs valeurs. **C'est un type de données**
- Un domaine peut-être défini :
 - ✓ en *intension* (c'est à dire en définissant les propriétés caractéristiques des valeurs du domaine, on parle aussi de compréhension) :
 - Tous les entiers
 - Les réels inférieur à 5
 - ✓ en *extension* (c'est à dire en énumérant toutes les valeurs du domaine) :
 - Couleur : {Bleu, Vert, Rouge, Jaune, Blanc, Noir}
 - SGBD : {Relationnel, Objet, Relationnel-Objet, graphes}

Le Modèle Relationnel

Objectifs

Concepts clés

Domaine

Relation, attribut et tuple

Notion de Clé

Schéma relationnel

- Dans la norme SQL, il y a 3 grandes familles de domaines :
 1. **Les chaines de caractères:**
 - ✓ **CHAR**: 1 seul caractère
 - ✓ **VARCHAR(n)**: chaine de caractères de longueur au plus n
 - ✓ ...
 2. **Les numériques :**
 - ✓ **INTEGER, (ou INT)**: entier
 - ✓ **FLOAT, DOUBLE, REAL**: nombre réel
 - ✓ ...
 3. **Les dates :**
 - ✓ **DATE**: date
 - ✓ **TIME**: heure, minute et secondes
 - ✓ ...

Le Modèle Relationnel

Objectifs

Concepts clés

Domaine

Relation, attribut et tuple

Notion de Clé

Schéma relationnel

- Relation ou table
 - ✓ Une relation est une table à deux dimensions
 - ✓ Un nom est associé à chaque colonne afin de la repérer indépendamment de son numéro d'ordre
- Attribut (propriété, champ ou colonne)
 - ✓ Nom donné à une colonne d'une relation
 - ✓ Prend ses valeurs dans un domaine
- Tuple (enregistrement, ligne, N-uplet, vecteur)
 - ✓ Une ligne de la table
 - ✓ Un enregistrement prend une valeur pour chaque attribut de la relation.

JEDI		
Nom	Prénom	Midichloriens
Skywalker	Anakin	90
Organa	Leia	70
Skywalker	Luke	80

Relation
Attribut
Tuple

Le Modèle Relationnel

Objectifs

Concepts clés

Domaine

Relation,
attribut et tuple

Notion de Clé

Schéma
relationnel

Remarques :

- ✓ Les noms des attributs sont uniques dans la relation
- ✓ L'ordre des lignes et des colonnes est indifférent
- ✓ Une relation ne contient pas deux fois la même ligne
- ✓ Un attribut se distingue d'un domaine car il peut ne comporter que certaines valeurs de ce domaine
- ✓ Un enregistrement peut ne pas avoir de valeur pour certains attributs de la relation, parce que cette valeur est inconnue ou inapplicable, sa valeur est alors "null"

Le Modèle Relationnel

Objectifs

Concepts clés

Domaine

Relation,
attribut et tuple

Notion de Clé

Schéma
relationnel

- **Définition** : Une clé est un groupe d'attributs minimum qui permet d'identifier de façon univoque un tuple dans une relation.
- Toute relation doit comporter au moins une clé, ce qui explique pourquoi une relation ne peut pas contenir deux tuples identiques.
- **Exemple**
<Nom, Prénom> est une clé pour la relation ETUDIANTS

ETUDIANTS		
Nom	Prénom	Parcours
Bernard	Eliot	Développement
Souti	Jeremy	null
Bernard	Thomas	Données
Tourelle	Linda	Développement
Emilion	Lila	Données

Le Modèle Relationnel

Objectifs

Concepts clés

Domaine

Relation, attribut et tuple

Notion de Clé

Schéma relationnel

➤ Important:

- ✓ Tous les attributs d'une clé doivent être valués, c'est-à-dire qu'aucun ne peut avoir de valeur *null*.
- ✓ Dire qu'un groupe d'attributs est une clé implique qu'il est unique et non *null*.

➤ Clé primaire et clés candidates

- ✓ Si plusieurs clés existent dans une relation, on en choisit une parmi celles-ci. Cette clé est appelée **clé primaire**.
- ✓ **Pas plus d'une clé primaire par table**
- ✓ La clé primaire est généralement :
 - **Immuable** : **ne** change jamais une fois valuée pour la première fois
 - la plus **simple** : **elle** porte sur le moins d'attributs et sur les attributs de domaine les plus basiques (entiers ou chaînes courtes typiquement).
- ✓ **clés candidates** : l'ensemble des clés d'une relation qui n'ont pas été choisies comme clé primaire

Le Modèle Relationnel

Objectifs

Concepts clés

Domaine

Relation,
attribut et tuple

Notion de clé

Schéma
relationnel

Exercice :
Quelle(s) sont les clé(s) candidates pour cette relation ?

REPARATIONS		
Garagiste	Intervention	Modèle
Martin	Electricité	Citroën
Piquard	Carrosserie	Ford
Martin	Mécanique	Citroën
Tussier	Electricité	Peugeot
Tussier	Dépannage	Peugeot
Piquard	Alarme	Ford

Le Modèle Relationnel

Objectifs

Concepts clés

Domaine

Relation,
attribut et tuple

Notion de Clé

Schéma
relationnel

➤ Clé étrangère :

- ✓ Une clé étrangère est un attribut ou un groupe d'attributs d'une relation R1 devant apparaître comme clé primaire dans une relation R2 afin de matérialiser une référence entre les tuples de R1 et les tuples de R2.
- ✓ Une clé étrangère d'un tuple référence une clé primaire d'un autre tuple.

➤ Contrainte d'intégrité référentielle

- ✓ Une clé étrangère respecte la contrainte d'intégrité référentielle si sa valeur est effectivement existante dans la clé primaire d'un tuple de la relation référencée, ou si sa valeur est *null*.
- ✓ Une clé étrangère qui ne respecte pas la contrainte d'intégrité référentielle exprime un lien vers un tuple qui n'existe pas et donc n'est pas cohérente.

Le Modèle Relationnel

Objectifs

Concepts clés

Domaine

Relation,
attribut et tuple

Notion de Clé

Schéma
relationnel

➤ Clé étrangère :

Table : **EMP**

EMPNO	ENAME	JOB	DEPTNO
7839	KING	PRESIDENT	10
7698	BLAKE	MANAGER	30
7782	CLARK	MANAGER	10
7566	JONES	MANAGER	20

Table : **DEPT**

DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON



Le Modèle Relationnel

Objectifs

Concepts clés

Domaine

Relation,
attribut et tuple

Notion de clé

Schéma
relationnel

➤ Référence (lien) entre relations

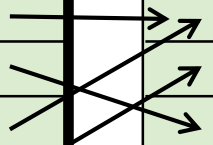
- ✓ La référence entre deux tuples T1 et T2 de deux relations différentes est exprimable par une valeur identique entre une clé étrangère du tuple T1 et la clé primaire de l'autre tuple T2.
- ✓ Ces références créent des liens entre les relations

Table : EMP

EMPNO	ENAME	JOB	DEPTNO
7839	KING	PRESIDENT	10
7698	BLAKE	MANAGER	30
7782	CLARK	MANAGER	10
7566	JONES	MANAGER	20

Table : DEPT

DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON



Le Modèle Relationnel

Objectifs

Concepts clés

Domaine

Relation,
attribut et tuple

Notion de clé

Schéma
relationnel

- **Schéma d'une relation** : Le schéma d'une relation définit cette relation en intension. Il est composé :
 - ✓ du nom de la relation,
 - ✓ de la liste de ses attributs avec les domaines respectifs dans lesquels ils prennent leurs valeurs,
 - ✓ de la clé primaire,
 - ✓ des clés étrangères
 - ✓ Par convention
 - la clé primaire est soulignée
 - les clés étrangères sont précédées (ou suivies) d'un #

Le Modèle Relationnel

Objectifs

Concepts clés

Domaine

Relation,
attribut et tuple

Notion de clé

Schéma
relationnel

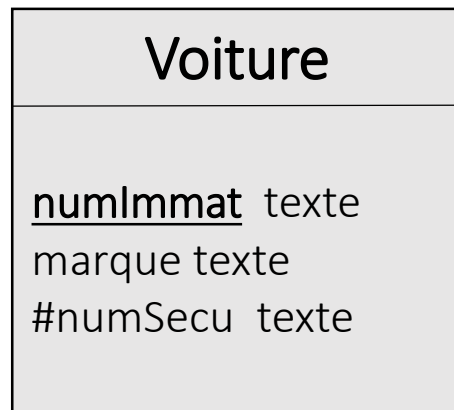
➤ Schéma d'une relation :

✓ 2 représentations possibles:

- Représentation textuelle

DEPT (numImmat:texte, marque : texte, #numSecu: texte)

- Représentation graphique (dépend du logiciel)



Le Modèle Relationnel

Objectifs

Concepts clés

Domaine

Relation,
attribut et tuple

Notion de clé

Schéma
relationnel

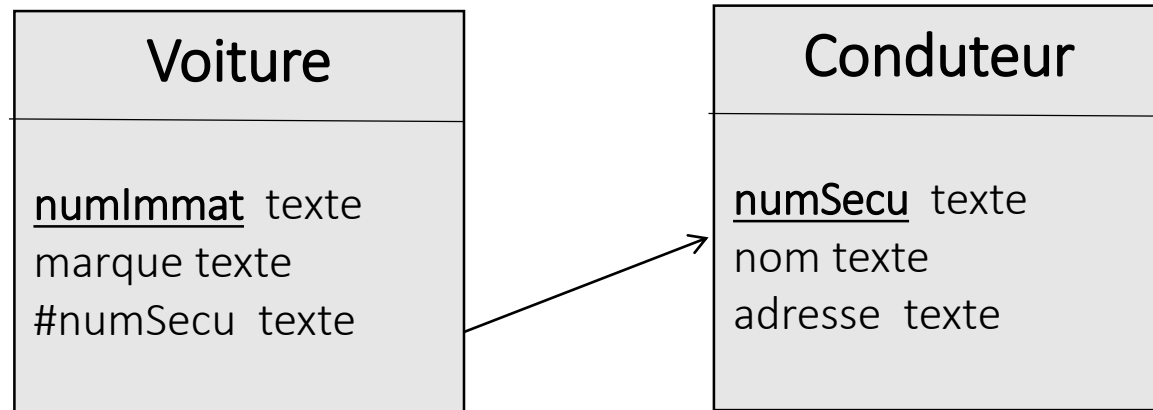
➤ Schéma relationnel d'une base de données

- ✓ Le schéma relationnel d'une BD est la définition en intension de cette BD (par opposition à l'instance de la BD qui est une extension de la BD). Il est composé de l'ensemble des schémas des relations qui le composent.

➤ Exemple

Voiture (numImmat:texte, marque : texte, #numSecu: texte)

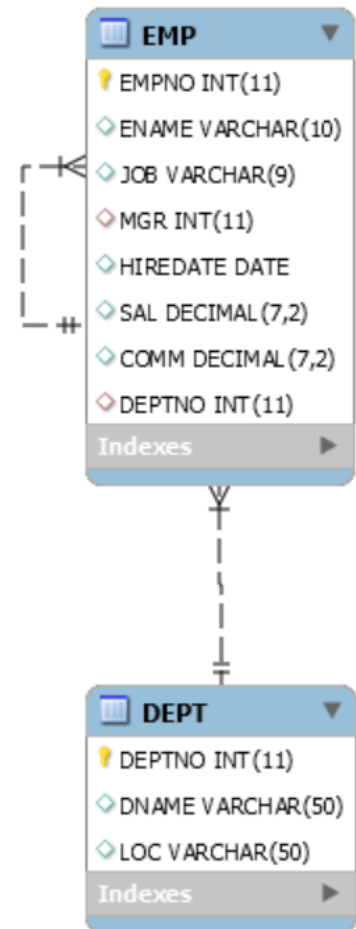
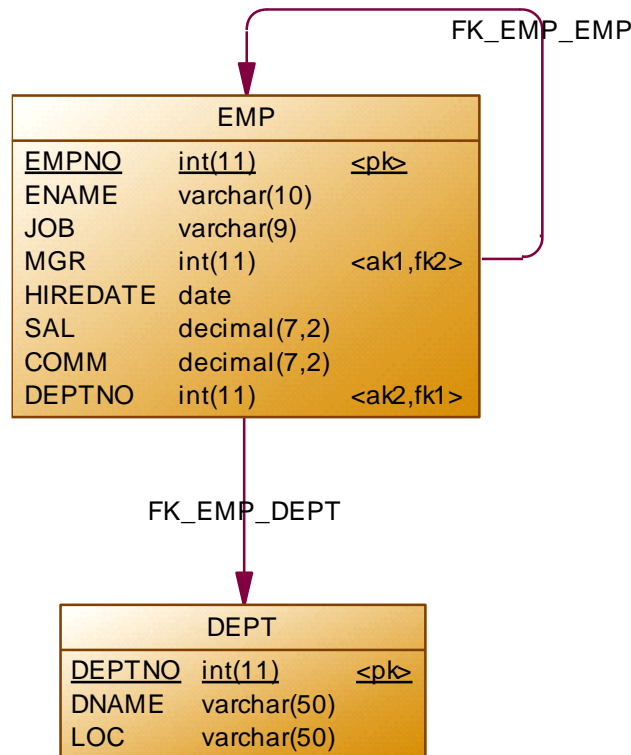
Conducteur (numSecu: texte, nom : texte, adresse: texte)



Le Modèle Relationnel

Objectifs
Concepts clés
Domaine
Relation,
attribut et tuple
Notion de clé
Schéma
relationnel

- **Schéma relationnel d'une base de données**
 - ✓ **Autres représentations graphiques**



Le Modèle Relationnel

Objectifs

Concepts clés

Domaine

Relation,
attribut et tuple

Notion de Clé

Schéma
relationnel

Exercice : Indiquer les phrases justes

- A. Une CP implique l'existence d'une CE**
- B. Une CE implique l'existence d'une CP**
- C. Une CE est un lien vers une autre table**
- D. Il peut exister plusieurs CP dans une table**
- E. Il peut exister plusieurs CE dans une table**

Manipulation de données et langage SQL

Langage de Manipulation de Données

Opérations

Projection

Restriction

Union

Différence

Intersection

Produit
cartésien

Jointures

- Le modèle relationnel offre un ensemble d'opérations formelles pour manipuler les données
- Ces opérations permettent d'exprimer les requêtes sur les données de la base
- Elles sont la base du langage **SQL**
 - ✓ Structured Query Language
 - ✓ langage standardisé, implémenté par tous les *SGBDR*
 - ✓ Il y a *quelques légères différences* entre les implémentations des différents *SGBD*

Langage de Manipulation de Données

Opérations

Projection

Restriction

Union

Différence

Intersection

Produit cartésien

Jointures

- Opérations de base
 - ✓ PROJECTION
 - ✓ RESTRICTION
 - ✓ UNION
 - ✓ PRODUIT CARTESIEN
- Opérations dérivées : (obtenues par combinaison d'opérations de base)
 - ✓ DIFFERENCE
 - ✓ INTERSECTION
 - ✓ JOINTURE
- Une même commande SQL avec plusieurs options: *SELECT ... FROM ...*

Langage de Manipulation de Données

Opérations

Projection

Restriction

Union

Différence

Intersection

Produit
cartésien

Jointures

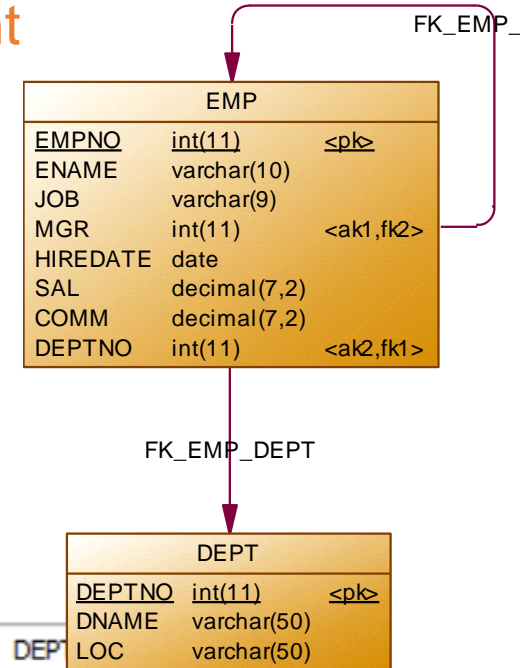
➤ Tous les exemples qui suivront utiliseront la base suivante sauf si indiqué autrement

DEPT

DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW-YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

EMP

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPT
7369	SMITH	CLERK	7902	2019-12-17	800.00	NULL	20
7499	ALLEN	SALESMAN	7698	2020-02-20	1600.00	300.00	30
7521	WARD	SALESMAN	7698	2020-02-22	1250.00	500.00	30
7566	JONES	MANAGER	7839	2020-04-02	2975.00	NULL	20
7654	MARTIN	SALESMAN	7698	2020-09-28	1250.00	1400.00	30
7698	BLAKE	MANAGER	7839	2020-05-01	2850.00	NULL	30
7782	CLARK	MANAGER	7839	2020-06-09	2450.00	NULL	10
7839	KING	PRESIDENT	NULL	2020-11-17	5000.00	NULL	10
7844	TURNER	SALESMAN	7698	2020-09-08	1500.00	0.00	30
7900	JAMES	CLERK	7698	2020-12-03	950.00	NULL	30
7902	FORD	ANALYST	7566	2020-12-03	3000.00	NULL	20
7934	MILLER	CLERK	7782	2021-01-23	1300.00	NULL	10



Opérations

Projection

Restriction

Union

Différence

Intersection

Produit
cartésien

Jointures

Opérations de base

➤ Opérations de base

- ✓ PROJECTION
- ✓ RESTRICTION
- ✓ UNION
- ✓ PRODUIT CARTESIEN

Opération sur une
seule table
(opération unaire)

➤ Opérations dérivées : (obtenues par combinaison d'opérations de base)

- ✓ DIFFERENCE
- ✓ INTERSECTION
- ✓ JOINTURE
- ✓ DIVISION

Opérations

Projection

Restriction

Union

Différence

Intersection

Produit
cartésien

Jointures

La projection

- Cette opération permet de construire une relation contenant seulement les attributs choisis :
 - ✓ Enlève les colonnes non choisies
 - ✓ La projection peut se faire sur plusieurs attributs

➤ SQL:

```
SELECT [ALL | DISTINCT] liste_attributs  
FROM relation
```

Où

- liste_attributs = * | attribut1 [AS] "alias colonne1" ,
attribut2 [AS] "alias colonne2", ...
- attribut = nom_relation.nom_attribut | nom_attribut |
expression arithmétique | chaîne de caractères

Langage de Manipulation de Données

Opérations

Projection

Restriction

Union

Différence

Intersection

Produit
cartésien

Jointures

➤ **Exemple :**

```
SELECT deptno, loc  
FROM   DEPT;
```

DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW-YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON



deptno	loc
10	NEW-YORK
20	DALLAS
30	CHICAGO
40	BOSTON

Langage de Manipulation de Données

Opérations

Projection

Restriction

Union

Différence

Intersection

Produit
cartésien

Jointures

- L'opérateur * : il permet la sélection de toutes les colonnes.

DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW-YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

SELECT *
FROM DEPT ;



DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW-YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

Langage de Manipulation de Données

Opérations

Projection

Restriction

Union

Différence

Intersection

Produit
cartésien

Jointures

➤ Les expressions arithmétiques :

```
SELECT ename, sal, 12*sal+300  
FROM EMP;
```

ename	sal	12*sal+300
SMITH	800.00	9900.00
ALLEN	1600.00	19500.00
WARD	1250.00	15300.00
JONES	2975.00	36000.00
MARTIN	1250.00	15300.00
BLAKE	2850.00	34500.00
CLARK	2450.00	29700.00
KING	5000.00	60300.00
TURNER	1500.00	18300.00
JAMES	950.00	11700.00
FORD	3000.00	36300.00
MILLER	1300.00	15900.00

Langage de Manipulation de Données

Opérations

Projection

Restriction

Union

Différence

Intersection

Produit
cartésien

Jointures



La valeur NULL

- ✓ Un NULL est une valeur inconnue ou non évaluée
- ✓ Un NULL n'est pas équivalent à un 0 ou un caractère blanc
- ✓ Les expressions arithmétiques contenant un null sont évaluées à null.

```
SELECT  ename, job, comm + 100  
FROM    EMP;
```

ename	job	comm + 100
SMITH	CLERK	NULL
ALLEN	SALESMAN	400.00
WARD	SALESMAN	600.00
JONES	MANAGER	NULL
MARTIN	SALESMAN	1500.00
BLAKE	MANAGER	NULL
CLARK	MANAGER	NULL
KING	PRESIDENT	NULL
TURNER	SALESMAN	100.00
JAMES	CLERK	NULL
FORD	ANALYST	NULL
MILLER	CLERK	NULL

Langage de Manipulation de Données

Opérations

Projection

Restriction

Union

Différence

Intersection

Produit
cartésien

Jointures

➤ Les alias :

```
SELECT ename AS NAME, sal SALARY
FROM EMP;
```

NAME	SALARY
...	

```
SELECT ename NAME, sal*2 "Annual
Salary"
FROM EMP;
```

NAME	Annual Salary
...	

➤ L'option **LIMIT**: affiche un nombre de ligne donné

```
SELECT *
FROM EMP;
LIMIT 100
```

Langage de Manipulation de Données

Opérations

Projection

Restriction

Union

Différence

Intersection

Produit
cartésien

Jointures

- L'opérateur ALL : l'affichage par défaut d'une requête inclut les lignes redondantes.

```
SELECT deptno  
FROM EMP;
```

```
SELECT ALL deptno  
FROM EMP;
```

deptno
10
10
10
20
20
20
30
30
30
30
30
30

- L'opérateur DISTINCT : permet d'éliminer les lignes redondantes.

```
SELECT DISTINCT deptno  
FROM EMP;
```

deptno
10
20
30

Opérations

Projection

Restriction

Union

Différence

Intersection

Produit
cartésien

Jointures

La restriction

- Cette opération permet de construire une relation contenant seulement les tuples correspondant à un prédicat évalué à vrai :
 - ✓ Enlève les lignes dont le prédicat est évalué à faux
 - ✓ Le prédicat peut contenir plusieurs paramètres
- Un prédicat est un énoncé dont le sens logique peut être faux ou vrai en fonction de la valeur des arguments.
- C'est une condition
- Exemples :
 - ✓ $SAL < 1000$
 - ✓ $JOB = 'SALESMAN'$
 - ✓ $COMM \text{ IS NOT NULL}$

Langage de Manipulation de Données

Opérations

Projection

Restriction

Union

Différence

Intersection

Produit
cartésien

Jointures

Instruction SQL pour la restriction

```
SELECT  [ALL|DISTINCT] liste_attributs  
FROM    relation  
WHERE   liste_conditions
```

Où

- **liste_conditions = condition [AND condition | OR condition]**
 - ✓ condition = [NOT] prédicat
 - ✓ prédicat = attribut op_comp expression | attribut IS [NOT] NULL
 - expression = attribut | constante
 - op_comp = < | > | = | <> | <= | >= | LIKE | IN | BETWEEN...

Langage de Manipulation de Données

Opérations

Projection

Restriction

Union

Différence

Intersection

Produit
cartésien

Jointures

➤ Exemple :

```
SELECT *  
FROM EMP  
WHERE deptno=20;
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	7902	2019-12-17	800.00	NULL	20
7566	JONES	MANAGER	7839	2020-04-02	2975.00	NULL	20
7902	FORD	ANALYST	7566	2020-12-03	3000.00	NULL	20

Opérations

Projection

Restriction

Union

Différence

Intersection

Produit
cartésien

Jointures

Instruction SQL pour la restriction

➤ Types Chaîne de caractères et Date :

- ✓ Les chaînes de caractères et les dates sont comprises entre simples apostrophes.
- ✓ Les chaînes de caractères sont sensibles à la casse.
- ✓ Les dates ont un format d'affichage standard :
 - Oracle : 'DD-MON-YY' (28 juillet 2003 => '23-JUL-03')
 - MariaDB/MySQL : 'YYYY-MM-DD' (28 juillet 2003 => '2003-07-23')

```
SELECT  ename, job, deptno
FROM    EMP
WHERE   ename = 'JAMES' ;
```

ename	job	deptno
JAMES	CLERK	30

Opérations

Projection

Restriction

Union

Différence

Intersection

Produit
cartésien

Jointures

Instruction SQL pour la restriction

➤ Opérateurs de comparaison :

= > >= <= < <>

- ✓ **BETWEEN ... AND ...** : valeur comprise dans un intervalle
- ✓ **IN (élément-1, élément-2, ...)** : égalité avec au moins l'un des membres de la liste
- ✓ **LIKE** : correspond à un patron de caractères
- ✓ **IS NULL** : est un NULL

Opérations

Projection

Restriction

Union

Différence

Intersection

Produit
cartésien

Jointures

Instruction SQL pour la restriction

- Opérateur BETWEEN : Permet d'afficher les lignes dont une valeur appartient à un intervalle défini par une valeur minimum et une valeur maximum

```
SELECT  ename, sal
FROM    EMP
WHERE    sal BETWEEN 1000 AND 1500;
```

ename	sal
WARD	1250.00
MARTIN	1250.00
TURNER	1500.00
MILLER	1300.00

Opérations

Projection

Restriction

Union

Différence

Intersection

Produit
cartésien

Jointures

Instruction SQL pour la restriction

- Opérateur IN : Permet de tester l'appartenance à une liste de valeurs finie.

```
SELECT empno, ename, sal, mgr  
FROM EMP  
WHERE mgr IN (7902, 7566, 7788) ;
```

empno	ename	sal	mgr
7369	SMITH	800.00	7902
7902	FORD	3000.00	7566

Opérations

Projection

Restriction

Union

Différence

Intersection

Produit
cartésien

Jointures

Instruction SQL pour la restriction

- Opérateur LIKE : Permet d'utiliser des caractères jokers pour définir des patrons de chaînes de caractère; la condition porte sur des caractères alphanumériques.

- ✓ % remplace un ou plusieurs caractères
- ✓ _ remplace un caractère exactement

```
SELECT ename  
FROM emp  
WHERE ename LIKE 'S%';
```

ename
SMITH

Opérations

Projection

Restriction

Union

Différence

Intersection

Produit
cartésien

Jointures

Instruction SQL pour la restriction

- Opérateur IS NULL : Permet de tester si une valeur est nulle.

```
SELECT  ename, mgr
FROM    EMP
WHERE   mgr IS NULL;
```

ename	mgr
KING	NULL

Opérations

Projection

Restriction

Union

Différence

Intersection

Produit
cartésien

Jointures

Instruction SQL pour la restriction

- **Les opérateurs logiques : permettent de combiner plusieurs prédicats.**

Opérateur	Signification
AND	= TRUE si les deux prédicats sont Vrai
OR	= TRUE si l'un des deux prédicats est Vrai
NOT	= TRUE si le prédicat suivant est faux

Opérations

Projection

Restriction

Union

Différence

Intersection

Produit
cartésien

Jointures

Instruction SQL pour la restriction

- L'opérateur logique AND : est vrai si les 2 prédicats sont vrais.

```
SELECT empno, ename, job, sal
FROM EMP
WHERE sal >= 1100 AND job = 'CLERK';
```

empno	ename	job	sal
7934	MILLER	CLERK	1300.00

Langage de Manipulation de Données

Opérations

Projection

Restriction

Union

Différence

Intersection

Produit
cartésien

Jointures

Instruction SQL pour la restriction

- L'opérateur logique OR : est vrai si au moins l'un des 2 prédicats est vrai.

```
SELECT empno, ename, job, sal
FROM EMP
WHERE sal >= 1100 OR job = 'CLERK';
```

empno	ename	job	sal
7369	SMITH	CLERK	800.00
7499	ALLEN	SALESMAN	1600.00
7521	WARD	SALESMAN	1250.00
7566	JONES	MANAGER	2975.00
7654	MARTIN	SALESMAN	1250.00
7698	BLAKE	MANAGER	2850.00
7782	CLARK	MANAGER	2450.00
7839	KING	PRESIDENT	5000.00
7844	TURNER	SALESMAN	1500.00
7900	JAMES	CLERK	950.00
7902	FORD	ANALYST	3000.00
7934	MILLER	CLERK	1300.00

Langage de Manipulation de Données

Opérations

Projection

Restriction

Union

Différence

Intersection

Produit
cartésien

Jointures

Instruction SQL pour la restriction

- **Opérateur logique NOT : est vrai si le prédicat est faux.**

```
SELECT ename, job  
FROM EMP  
WHERE job NOT IN ( 'CLERK', 'MANAGER' );
```

ename	job
ALLEN	SALESMAN
WARD	SALESMAN
MARTIN	SALESMAN
KING	PRESIDENT
TURNER	SALESMAN
FORD	ANALYST

Opérations

Projection

Restriction

Union

Différence

Intersection

Produit
cartésien

Jointures

Instruction SQL pour la restriction

- Les opérateurs et leurs règles de priorité :

Ordre d'évaluation	Type d'opérateur
1	Opérateurs de comparaison
2	NOT
3	AND
4	OR

- Utiliser les parenthèses pour changer l'ordre des priorités.

Langage de Manipulation de Données

Opérations

Projection

Restriction

Union

Différence

Intersection

Produit
cartésien

Jointures

Instruction SQL pour la restriction

➤ Exemples de règles de priorité :

```
SELECT ename, job, sal
FROM EMP
WHERE (job='SALESMAN'
      AND sal>1500;
```

OR job='MANAGER')

ename	job	sal
ALLEN	SALESMAN	1600.00
JONES	MANAGER	2975.00
BLAKE	MANAGER	2850.00
CLARK	MANAGER	2450.00

```
SELECT ename, job, sal
FROM EMP
WHERE job='SALESMAN' OR
      job='MANAGER' AND
      sal>1500;
```

ename	job	sal
ALLEN	SALESMAN	1600.00
WARD	SALESMAN	1250.00
JONES	MANAGER	2975.00
MARTIN	SALESMAN	1250.00
BLAKE	MANAGER	2850.00
CLARK	MANAGER	2450.00
TURNER	SALESMAN	1500.00

Opérations
Projection
Restriction
Union
Différence
Intersection
Produit
cartésien
Jointures

Le tri des lignes en SQL

```
SELECT      [ALL|DISTINCT]
liste_attributs
FROM        relation
WHERE       liste_conditions
ORDER BY    liste_colonnes
```

Où

- **liste_colonnes = nom_colonne1 [ASC | DESC] ,
nom_colonne2 [ASC | DESC], ...**
- **liste_colonnes = numéro_colonne1 [ASC |
DESC], numéro_colonne2 [ASC | DESC], ...**

Le tri des lignes en SQL

- Trier les lignes : ASC, par défaut, ordre croissant
DESC ordre décroissant
- La clause ORDER BY doit être la dernière clause d'une requête

```
SELECT   ename, job, deptno, hiredate
FROM     EMP
ORDER BY hiredate ASC;;
```

ename	job	deptno	hiredate
SMITH	CLERK	20	2019-12-17
ALLEN	SALESMAN	30	2020-02-20
WARD	SALESMAN	30	2020-02-22
JONES	MANAGER	20	2020-04-02
BLAKE	MANAGER	30	2020-05-01
CLARK	MANAGER	10	2020-06-09
TURNER	SALESMAN	30	2020-09-08
MARTIN	SALESMAN	30	2020-09-28
KING	PRESIDENT	10	2020-11-17
JAMES	CLERK	30	2020-12-03
FORD	ANALYST	20	2020-12-03
MILLER	CLERK	10	2021-01-23

Opérations

Projection

Restriction

Union

Différence

Intersection

Produit
cartésien

Jointures

Le tri des lignes en SQL

- La clause **ORDER BY** accepte les noms de colonne ou les numéros de colonnes
- Oracle et MariaDB acceptent également les alias de colonne

```
SELECT    ename, job, deptno, hiredate
FROM      EMP
ORDER BY  ename;
```

```
SELECT    ename, job, deptno, hiredate
FROM      EMP
ORDER BY  1;
```

```
SELECT    ename Nom, job, deptno,
hiredate
FROM      EMP
ORDER BY  Nom;
```

Opérations
Projection
Restriction
Union
Différence
Intersection
Produit
cartésien
Jointures

Le tri des lignes en SQL

- **Tri sur plusieurs colonnes : l'ordre des colonnes dans la clause ORDER BY a une importance**

```
SELECT  ename, deptno, sal  
FROM    EMP  
ORDER BY deptno DESC, sal DESC;
```

ename	deptno	sal
BLAKE	30	2850.00
ALLEN	30	1600.00
TURNER	30	1500.00
WARD	30	1250.00
MARTIN	30	1250.00
JAMES	30	950.00
FORD	20	3000.00
JONES	20	2975.00
SMITH	20	800.00
KING	10	5000.00
CLARK	10	2450.00
MILLER	10	1300.00

Langage de Manipulation de Données

Opérations

Projection

Restriction

Union

Différence

Intersection

Produit
cartésien

Jointures

- Opérations de base
 - ✓ PROJECTION
 - ✓ RESTRICTION
 - ✓ **UNION**
 - ✓ PRODUIT CARTESIEN

- Opérations dérivées : (obtenues par combinaison d'opérations de base)
 - ✓ **DIFFERENCE**
 - ✓ **INTERSECTION**
 - ✓ JOINTURE

Opérations sur
deux tables de
même schéma
Opérations
ensemblistes

Langage de Manipulation de Données

Opérations

Projection

Restriction

Union

Différence

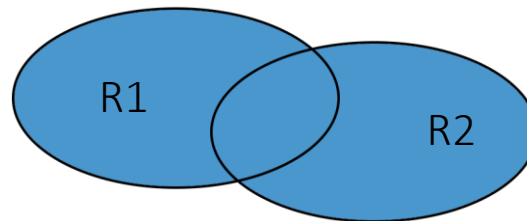
Intersection

Produit
cartésien

Jointures

Union

- L'union permet de construire une relation contenant la fusion de deux relations sans doublons
- L'union nécessite d'avoir des relations qui possèdent les mêmes schémas. Les colonnes n'ont pas forcément le même nom mais obligatoirement le même type.



Relation résultante en bleu

Langage de Manipulation de Données

Opérations

Projection

Restriction

Union

Différence

Intersection

Produit
cartésien

Jointures

Exemple d'union

CUSTOMER

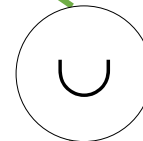
NAME	CUSTID
JOCKSPORTS	100
SPORT SHOP	101
VOLLYRITE	102

EMP

ENAME	DEPTNO
SMITH	10
ALLEN	20

CUSTOMER \cup EMP

NAME	CUSTID
JOCKSPORTS	100
SPORT SHOP	101
VOLLYRITE	102
SMITH	10
ALLEN	20



Langage de Manipulation de Données

Opérations

Projection

Restriction

Union

Différence

Intersection

Produit
cartésien

Jointures

UNION en SQL

➤ Syntaxe :

```
SELECT liste_attributs
  FROM   liste_relations
[WHERE liste_conditions]
UNION
  SELECT liste_attributs
  FROM   liste_relations
[WHERE liste_conditions]
```

➤ Exemple : "Quel est l'ensemble des noms d'employés et de départements ?"

```
SELECT ename
  FROM   EMP
  UNION
  SELECT dname
  FROM   DEPT;
```

ename
SMITH
ALLEN
WARD
JONES
MARTIN
BLAKE
CLARK
KING
TURNER
JAMES
FORD
MILLER
ACCOUNTING
RESEARCH
SALES
OPERATIONS

Opérations

Projection

Restriction

Union

Différence

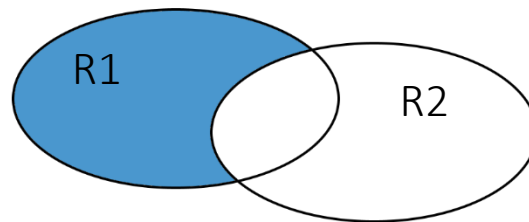
Intersection

Produit
cartésien

Jointures

Différence

- La différence permet d'obtenir les tuples de la relation 1 qui n'appartiennent pas à la relation 2
- La différence nécessite d'avoir des relations qui possèdent les mêmes domaines (pour les colonnes concernées par la différence)
- Cette opération n'est pas commutative



Relation résultante en bleu

Langage de Manipulation de Données

Opérations

Projection

Restriction

Union

Différence

Intersection

Produit
cartésien

Jointures

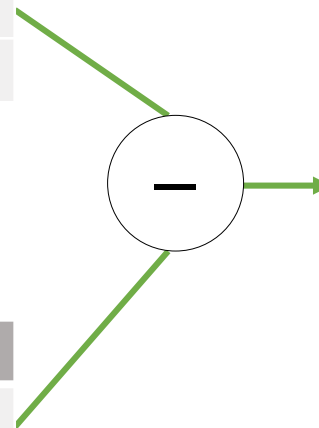
Exemple de différence

EMP

ENAME	DEPTNO
SMITH	10
ALLEN	30
CLARK	10
JONES	20
DOE	

EMPLOYEE

ENAME	DEPTNO
SMITH	10
CLARK	10



EMP - EMPLOYEE

ENAME	DEPTNO
ALLEN	30
JONES	20
DOE	

Langage de Manipulation de Données

Opérations

Projection

Restriction

Union

Différence

Intersection

Produit
cartésien

Jointures

Différence

- **Syntaxe SQL (SQL Standard/oracle/Postgre)**

```
SELECT liste_attributs  
FROM     liste_relations  
[WHERE liste_conditions]  
MINUS | EXCEPT
```

```
SELECT     liste_attributs  
FROM       liste_relations  
[WHERE liste_conditions]
```

- **Exemple : "Quels sont les départements qui n'ont pas d'employés ?"**

```
SELECT deptno  
FROM   DEPT  
EXCEPT  
SELECT deptno  
FROM   EMP;
```

deptno
40

Opérations

Projection

Restriction

Union

Différence

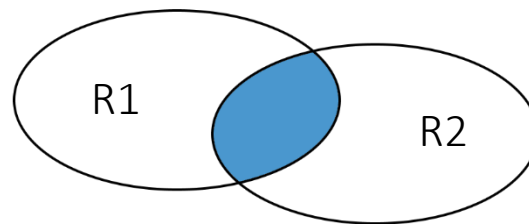
Intersection

Produit
cartésien

Jointures

Intersection

- L'intersection permet de construire une relation contenant la les tuples présents dans deux relations sans doublons
- L'intersection nécessite d'avoir des relations qui possèdent les mêmes domaines (pour les colonnes concernées)



Relation résultante en bleu

L'algèbre relationnelle

Opérations

Projection

Restriction

Union

Différence

Intersection

Produit
cartésien

Jointures

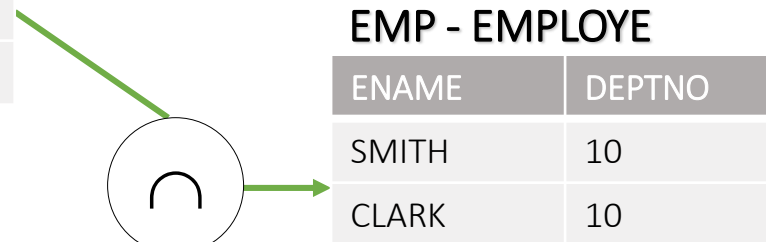
Exemple d'intersection

EMP

ENAME	DEPTNO
SMITH	10
ALLEN	30
CLARK	10
JONES	20
DOE	

EMPLOYEE

ENAME	DEPTNO
SMITH	10
CLARK	10
JONES	30



Langage de Manipulation de Données

Opérations

Projection

Restriction

Union

Différence

Intersection

Produit
cartésien

Jointures

Intersection

➤ Syntaxe (SQL Standard/Oracle/Postgre)

```
SELECT liste_attributs
FROM     liste_relations
[WHERE liste_conditions]
INTERSECT
SELECT   liste_attributs
FROM     liste_relations
[WHERE liste_conditions]
```

➤ **Exemple:** Quels sont les départements qui contiennent à la fois des managers et des analystes ?"

```
SELECT deptno FROM EMP WHERE job = 'MANAGER'
INTERSECT
SELECT deptno FROM EMP WHERE job = 'ANALYST';
```

deptno
20

Opérations

Projection

Restriction

Union

Différence

Intersection

Produit

cartésien

Jointures

Règles communes à l'union, l'intersection et la différence

- **Tous les ordres SELECT doivent avoir le même nombre de colonnes sélectionnées**
- **Les doublons sont éliminés**
- **Les noms des colonnes sont ceux du premier ordre SELECT**
- **Si une clause ORDER BY est utilisée, elle doit faire référence au numéro de la colonne et non pas à son nom**

Opérations

Projection

Restriction

Union

Différence

Intersection

Produit
cartésien

Jointures

Exercice :

Ecrire les requêtes suivantes

- **Donner les numéros de tous les employés**
- **Chercher tous les employés qui sont managers**
- **Chercher tous les employés qui sont managers du département 10**
- **Donner les numéros et les noms de tous les départements**
- **Donner les noms et les salaires des employés embauchés après le 01/01/1981**

Langage de Manipulation de Données

Opérations

Projection

Restriction

Union

Différence

Intersection

Produit
cartésien

Jointures

- Opérations de base
 - ✓ PROJECTION
 - ✓ RESTRICTION
 - ✓ UNION
 - ✓ **PRODUIT CARTESIEN**

Opérations sur deux
tables ou plus, de
schémas différents

- Opérations dérivées : (obtenues par
combinaison d'opérations de base)
 - ✓ DIFFERENCE
 - ✓ INTERSECTION
 - ✓ **JOINTURE**

Langage de Manipulation de Données

Opérations

Projection

Restriction

Union

Différence

Intersection

Produit
cartésien

Jointures

Produit cartésien

- Le produit cartésien se construit en combinant toutes les possibilités de "liaison" des deux relations

EMP (3 lignes)

EMPNO	ENAME
7369	SMITH
7499	ALLEN
7566	JONES

DEPT (2 lignes)

DNAME	LOC
ACCOUNTING	NEW YORK
RESEARCH	DALLAS

Produit cartésien ($3 \times 2 = 6$ lignes)

EMPNO	ENAME	DNAME	LOC
7369	SMITH	ACCOUNTING	NEW YORK
7369	SMITH	RESEARCH	DALLAS
7499	ALLEN	ACCOUNTING	NEW YORK
7499	ALLEN	RESEARCH	DALLAS
7566	JONES	ACCOUNTING	NEW YORK
7566	JONES	RESEARCH	DALLAS

Langage de Manipulation de Données

Opérations

Projection

Restriction

Union

Différence

Intersection

Produit
cartésien

Jointures

Le produit cartésien en SQL

```
SELECT liste_attributs  
FROM   relation_gauche, relation_droite
```

ou

```
SELECT liste_attributs  
FROM   relation_gauche  
CROSS JOIN relation_droite
```

Le produit cartésien en SQL

- **Exemple : "Afficher toutes les combinaisons possibles entre les noms d'employés et les noms de département"**

```
SELECT  ename,  dname  
FROM    EMP,  DEPT
```

- **ou**

```
SELECT  ename,  dname  
FROM    EMP CROSS JOIN DEPT
```

Jointures

- La jointure est un produit cartésien restreint entre deux relations R1 et R2
- Le prédicat (condition) de la restriction est appelé Pivot de jointure

$\text{pivot_jointure} = R1.\text{attribut Op } R2.\text{attribut [AND | OR } R1.\text{attribut Op } R2.\text{attribut]}$

- Elle peut donc s'écrire avec un produit cartésien et une restriction
- Plusieurs types de jointure
 - ✓ Interne
 - ✓ Naturelle
 - ✓ Externe
 - ✓ Droite
 - ✓ Gauche
 - ✓ Auto jointure

Langage de Manipulation de Données

Opérations

Projection

Restriction

Union

Différence

Intersection

Produit
cartésien

Jointures

- ✓ Interne
- ✓ Naturelle
- ✓ Externe
- ✓ Droite
- ✓ Gauche
- ✓ Auto jointure

Jointure interne

- La jointure interne ne retient que les tuples qui vérifient le pivot de jointure
- SQL

```
SELECT liste_attributs  
FROM relation_gauche  
[INNER] JOIN relation_droite ON  
    pivot_jointure  
[WHERE liste_conditions]
```

Langage de Manipulation de Données

Opérations

Projection

Restriction

Union

Différence

Intersection

Produit
cartésien

Jointures

- ✓ Interne
- ✓ Naturelle
- ✓ Externe
- ✓ Droite
- ✓ Gauche
- ✓ Auto jointure

Exemple de jointure interne

EMP (14 lignes) Clé étrangère Clé primaire DEPT (4 lignes)

EMPNO	ENAME	DEPTNO
7369	SMITH	10
7499	ALLEN	30
7782	CLARK	10
7566	JONES	20
7463	WARD	30
7896	FORD	30
7632	SMITH	20

DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

```
SELECT      *
FROM      EMP
JOIN DEPT ON
EMP.DEPTNO=DEPT.DEPTNO
```

EMPNO	ENAME	DEPTNO	DNAME	LOC
7369	SMITH	10	ACCOUNTING	NEW YORK
7499	ALLEN	30	SALES	CHICAGO
7782	CLARK	10	ACCOUNTING	NEW YORK
7566	JONES	20	RESEARCH	DALLAS
7463	WARD	30	SALES	CHICAGO
7896	FORD	30	SALES	CHICAGO
7632	SMITH	20	RESEARCH	DALLAS

Opérations

Projection

Restriction

Union

Différence

Intersection

Produit
cartésien

Jointures

- ✓ Interne
- ✓ Naturelle
- ✓ Externe
- ✓ Droite
- ✓ Gauche
- ✓ Auto jointure

La jointure interne par un produit cartésien

➤ Exemple :

```
SELECT  EMP.empno, EMP.ename, EMP.deptno,  
        DEPT.deptno, DEPT.loc  
FROM    EMP INNER JOIN DEPT  
        ON EMP.deptno=DEPT.deptno;
```

```
SELECT  EMP.empno, EMP.ename, EMP.deptno,  
        DEPT.deptno, DEPT.loc  
FROM    EMP, DEPT  
WHERE   EMP.deptno=DEPT.deptno;
```

Opérations

Projection

Restriction

Union

Différence

Intersection

Produit
cartésien

Jointures

- ✓ Interne
- ✓ Naturelle
- ✓ Externe
- ✓ Droite
- ✓ Gauche
- ✓ Auto jointure

La jointure interne

Pourquoi préférer le JOIN à la jointure par le WHERE ?

- ✓ Les jointures faites dans la clause WHERE (ancienne syntaxe de 1986 !) ne permettent pas de faire la distinction de prime abord entre ce qui relève de la restriction et ce qui relève de la jointure
- ✓ La lisibilité des requêtes est plus grande en utilisant la syntaxe à base de JOIN
- ✓ L'optimisation d'exécution de la requête est souvent plus pointue du fait de l'utilisation du JOIN (pas besoin de passer par le produit cartésien très coûteux en temps et espace)

Langage de Manipulation de Données

Opérations

Projection

Restriction

Union

Différence

Intersection

Produit
cartésien

Jointures

- ✓ Interne
- ✓ Naturelle
- ✓ Externe
- ✓ Gauche
- ✓ Droite
- ✓ Auto jointure

Jointure naturelle

- La jointure naturelle entre R1 et R2 est une jointure pour laquelle la condition est l'égalité entre les **attributs de même nom** de R1 et de R2.
- Il est donc inutile de spécifier la condition dans une jointure naturelle, elle reste toujours implicite



SQL

```
SELECT      liste_attributs
FROM        relation_gauche
NATURAL JOIN      relation_droite
[WHERE      liste_conditions]
```

Langage de Manipulation de Données

Opérations

Projection

Restriction

Union

Différence

Intersection

Produit

cartésien

Jointures

- ✓ Interne
- ✓ Naturelle
- ✓ Externe
- ✓ Gauche
- ✓ Droite
- ✓ Auto jointure

Exemple de jointure naturelle

EMP

EMPNO	ENAME	DEPTNO
7369	SMITH	10
7499	ALLEN	30
7782	CLARK	10
7566	JONES	20
7463	WARD	30
7896	FORD	30
7632	SMITH	20

DEPT

DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

SELECT *
FROM EMP
NATURAL JOIN DEPT

EMPNO	ENAME	DEPTNO	DNAME	LOC
7369	SMITH	10	ACCOUNTING	NEW YORK
7499	ALLEN	30	SALES	CHICAGO
7782	CLARK	10	ACCOUNTING	NEW YORK
7566	JONES	20	RESEARCH	DALLAS
7463	WARD	30	SALES	CHICAGO
7896	FORD	30	SALES	CHICAGO
7632	SMITH	20	RESEARCH	DALLAS

Langage de Manipulation de Données

Opérations

Projection

Restriction

Union

Différence

Intersection

Produit
cartésien

Jointures

- ✓ Interne
- ✓ Naturelle
- ✓ Externe
- ✓ Gauche
- ✓ Droite
- ✓ Auto jointure

Jointure externe

- La jointure externe est une jointure qui fait figurer les tuples qui n'ont pas "d'associé" dans l'autre relation
- On associe à ces tuples des NULL
- SQL

```
SELECT      liste_attributs
FROM        relation_gauche
[FULL]OUTER JOIN      relation_droite
ON pivot_jointure
[WHERE      liste_conditions]
```

Langage de Manipulation de Données

Opérations

Projection

Restriction

Union

Différence

Intersection

Produit
cartésien

Jointures

- ✓ Interne
- ✓ Naturelle
- ✓ Externe
- ✓ Gauche
- ✓ Droite
- ✓ Auto jointure

Exemple de jointure externe

EMP	ENAME	DEPTNO	DEPTNO	DNAME	DEPT
	SMITH	10	10	ACCOUNTING	
	ALLEN	30	30	SALES	
	CLARK	10	10	ACCOUNTING	
	JONES	20	20	RESEARCH	
	DOE		40	OPERATIONS	

L'employé DOE
n'a pas de
département

Aucun employé dans
le département
OPERATIONS

EMPNO	ENAME	DEPTNO	DEPTNO	DNAME	LOC
7369	SMITH	10	10	ACCOUNTING	NEW YORK
7499	ALLEN	30	30	SALES	CHICAGO
7782	CLARK	10	10	ACCOUNTING	NEW YORK
7566	JONES	20	20	RESEARCH	DALLAS
7590	DOE				
			40	OPERATION	BUFALO

Opérations

Projection

Restriction

Union

Différence

Intersection

Produit
cartésien

Jointures

- ✓ Interne
- ✓ Naturelle
- ✓ Externe
- ✓ Gauche
- ✓ Droite
- ✓ Auto jointure

Jointure externe gauche

- La jointure externe gauche est une jointure externe qui fait figurer la totalité des tuples de la relation gauche, mais seulement les tuples de la relation droite qui sont "liés" à la relation 1
- Cette opération n'est pas commutative
- SQL

```
SELECT      liste_attributs
FROM        relation_gauche
LEFT [OUTER] JOIN      relation_droite
ON  pivot_jointure
WHERE      liste_conditions]
```

Langage de Manipulation de Données

Opérations

Projection

Restriction

Union

Différence

Intersection

Produit
cartésien

Jointures

- ✓ Interne
- ✓ Naturelle
- ✓ Externe
- ✓ Gauche
- ✓ Droite
- ✓ Auto jointure

Exemple de jointure externe gauche

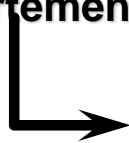
EMP

ENAME	DEPTNO
SMITH	10
ALLEN	30
CLARK	10
JONES	20
DOE	

DEPT

DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

L'employé
DOE n'a pas
de
département



```
SELECT      *  
FROM      EMP  
LEFT JOIN DEPT ON EMP.DEPTNO=DEPT.DEPTNO
```

EMPNO	ENAME	DEPTNO	DNAME	LOC
7369	SMITH	10	ACCOUNTING	NEW YORK
7499	ALLEN	30	SALES	CHICAGO
7782	CLARK	10	ACCOUNTING	NEW YORK
7566	JONES	20	RESEARCH	DALLAS
7590	DOE			

Langage de Manipulation de Données

Opérations

Projection

Restriction

Union

Différence

Intersection

Produit
cartésien

Jointures

- ✓ Interne
- ✓ Naturelle
- ✓ Externe
- ✓ Gauche
- ✓ Droite
- ✓ Auto jointure

Jointure externe droite

- La jointure externe droite est une jointure externe qui fait figurer la totalité des tuples de la relation droite, mais seulement les tuples de la relation gauche qui sont "liés" à la relation droite
- Cette opération n'est pas commutative
- SQL

```
SELECT      liste_attributs
FROM        relation_gauche
RIGHT [OUTER] JOIN  relation_droite
ON pivot_jointure
WHERE      liste_conditions]
```

Langage de Manipulation de Données

Opérations
Projection
Restriction
Union
Différence
Intersection
Produit
cartésien
Jointures

- ✓ Interne
- ✓ Naturelle
- ✓ Externe
- ✓ Gauche
- ✓ Droite
- ✓ Auto jointure

EMP

ENAME	DEPTNO
SMITH	10
ALLEN	30
CLARK	10
JONES	20

Exemple de jointure externe droite

DEPT

DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

Aucun
employé dans
le
département
OPERATIONS

```
SELECT *  
FROM EMP  
RIGHT JOIN DEPT ON EMP.DEPTNO=DEPT.DEPTNO
```

EMPNO	ENAME	DEPTNO	DEPTNO	DNAME	LOC
7369	SMITH	10	10	ACCOUNTING	NEW YORK
7499	ALLEN	30	30	SALES	CHICAGO
7782	CLARK	10	10	ACCOUNTING	NEW YORK
7566	JONES	20	20	RESEARCH	DALLAS
			40	OPERATION	BUFALO

Langage de Manipulation de Données

Opérations

Projection

Restriction

Union

Différence

Intersection

Produit cartésien

Jointures

- ✓ Interne
- ✓ Naturelle
- ✓ Externe
- ✓ Gauche
- ✓ Droite
- ✓ Auto jointure

Auto-jointure

- L'auto-jointure est une jointure d'une relation avec elle-même
- SQL et représentation graphique
 - ✓ Comme une jointure interne
- Exemple : comment récupérer le nom des managers de chaque employé ?

```
SELECT      concat (e.ENAME, ' travaille pour ',  
m.ENAME)  
FROM      EMP e  
JOIN EMP m ON e.mgr = m.ename;
```

Les jointures et les alias de tables

- **Ajouter des alias aux tables jointées :**
- ✓ simplifie les requêtes
 - ✓ améliore les performances
 - ✓ lève les ambiguïtés sur l'appartenance des colonnes

```
SELECT    EMP.empno, EMP.ename, EMP.deptno,
          DEPT.deptno, DEPT.loc
FROM      EMP INNER JOIN DEPT
          ON EMP.deptno=DEPT.deptno;
```

```
SELECT    e.empno, e.ename, e.deptno,
          d.deptno, d.loc
FROM      EMP e INNER JOIN DEPT d
          ON e.deptno=d.deptno;
```

Opérations
Projection
Restriction
Union
Différence
Intersection
Produit
cartésien
Jointures

Les jointures sur plus de deux relations

```
SELECT      name, itemid
FROM CUSTOMER c
INNER JOIN ORD o ON c.custid=o.custid
INNER JOIN ITEM i ON o.ordid=i.ordid;
```

CUSTOMER

NAME	CUSTID
JOCK	100
VOLLY	102
JUST	107

ORD

CUSTID	ORDID
101	610
102	611
104	604

ITEM

ORDID	ITEMID
610	3
611	1
601	2

Opérations

Projection

Restriction

Union

Différence

Intersection

Produit
cartésien

Jointures

Relations entre les jointures

**Jointure externe gauche \cap Jointure externe droite
= Jointure interne**

**Jointure externe gauche \cup Jointure externe droite
= Jointure externe**

Opérations
Projection
Restriction
Union
Différence
Intersection
Produit
cartésien
Jointures

Exercice : Ecrire les requêtes

- **Donner le nom et le lieu de travail de tous les employés**
- **Donner le lieu de travail de tous les employés s'appelant JONES**
- **Donner les noms de département, numéros de département et lieu de travail de tous les managers**

Les fonctions

Les fonctions

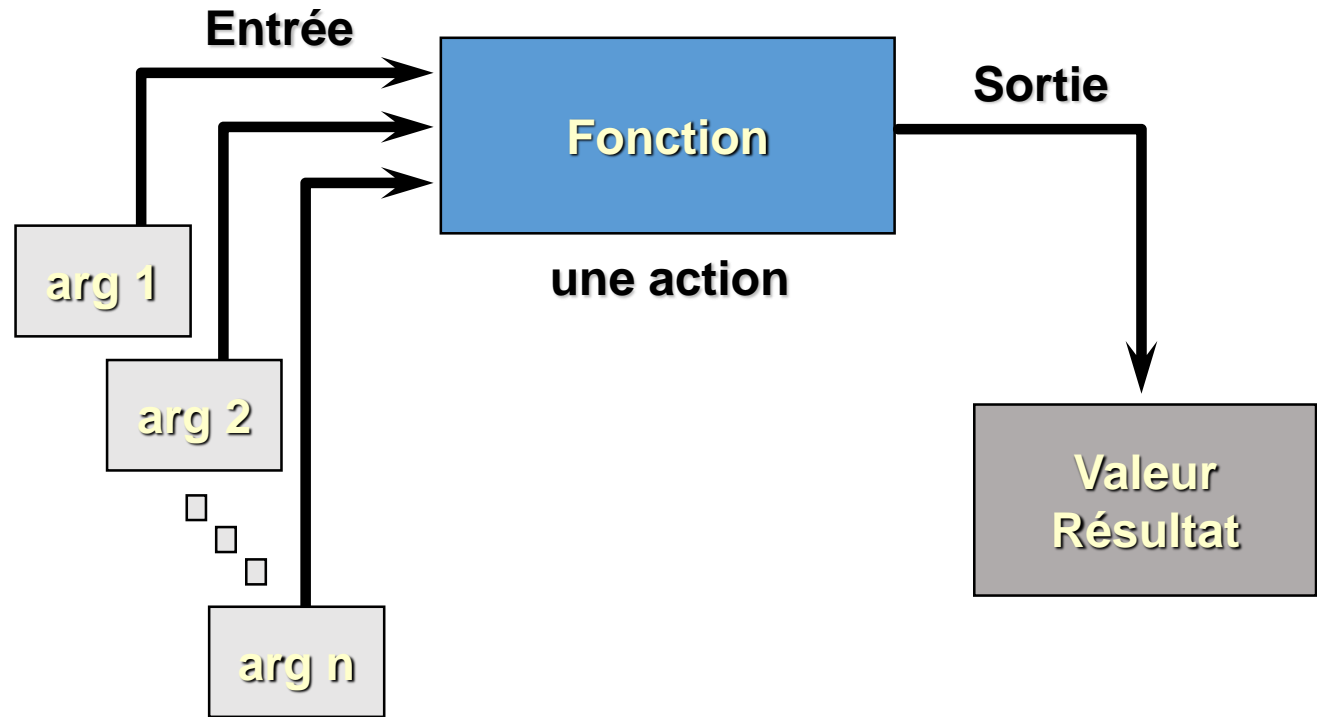
Définition

Chaînes

Numériques

Dates

- Une fonction permet de travailler avec autre chose que ce qui est directement dans la relation



- Une fonction peut être utilisée dans les clauses :
 - ✓ SELECT : pour traiter une valeur avant de l'afficher
 - ✓ WHERE : pour traiter une valeur avant de la tester

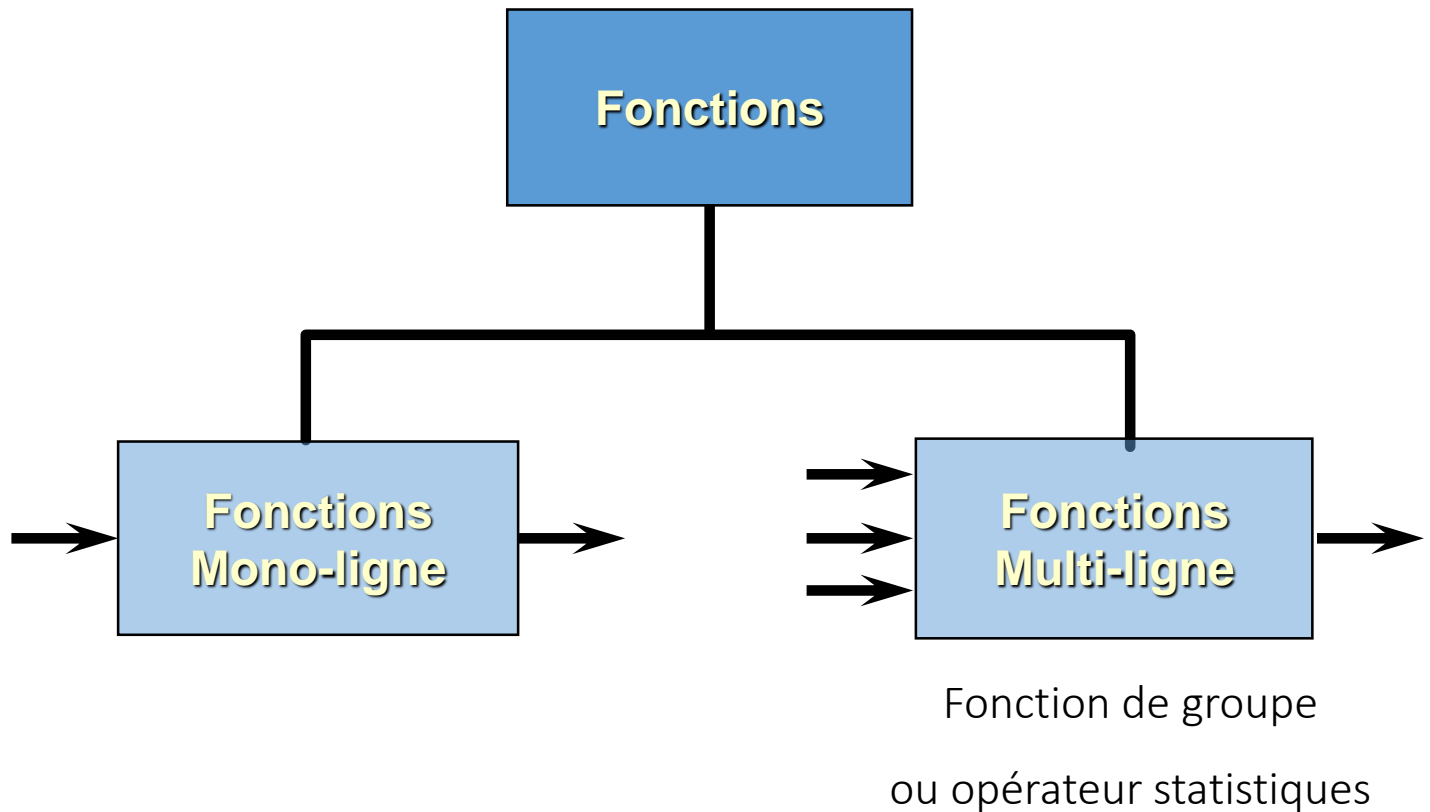
Définition

Chaînes

Numériques

Dates

- Il existe deux types de fonctions :



Définition

Chaînes

Numériques

Dates

- **Une fonction mono-ligne :**
 - ✓ **Manipule les données d'une ligne**
 - ✓ **Prend des arguments et retourne une valeur**
 - ✓ **Agit sur chaque ligne de la requête**
 - ✓ **Retourne un résultat par ligne**
 - ✓ **Peut modifier le type de donnée**
 - ✓ **Peut être imbriquée dans une ou plusieurs autres fonctions mono-ligne**

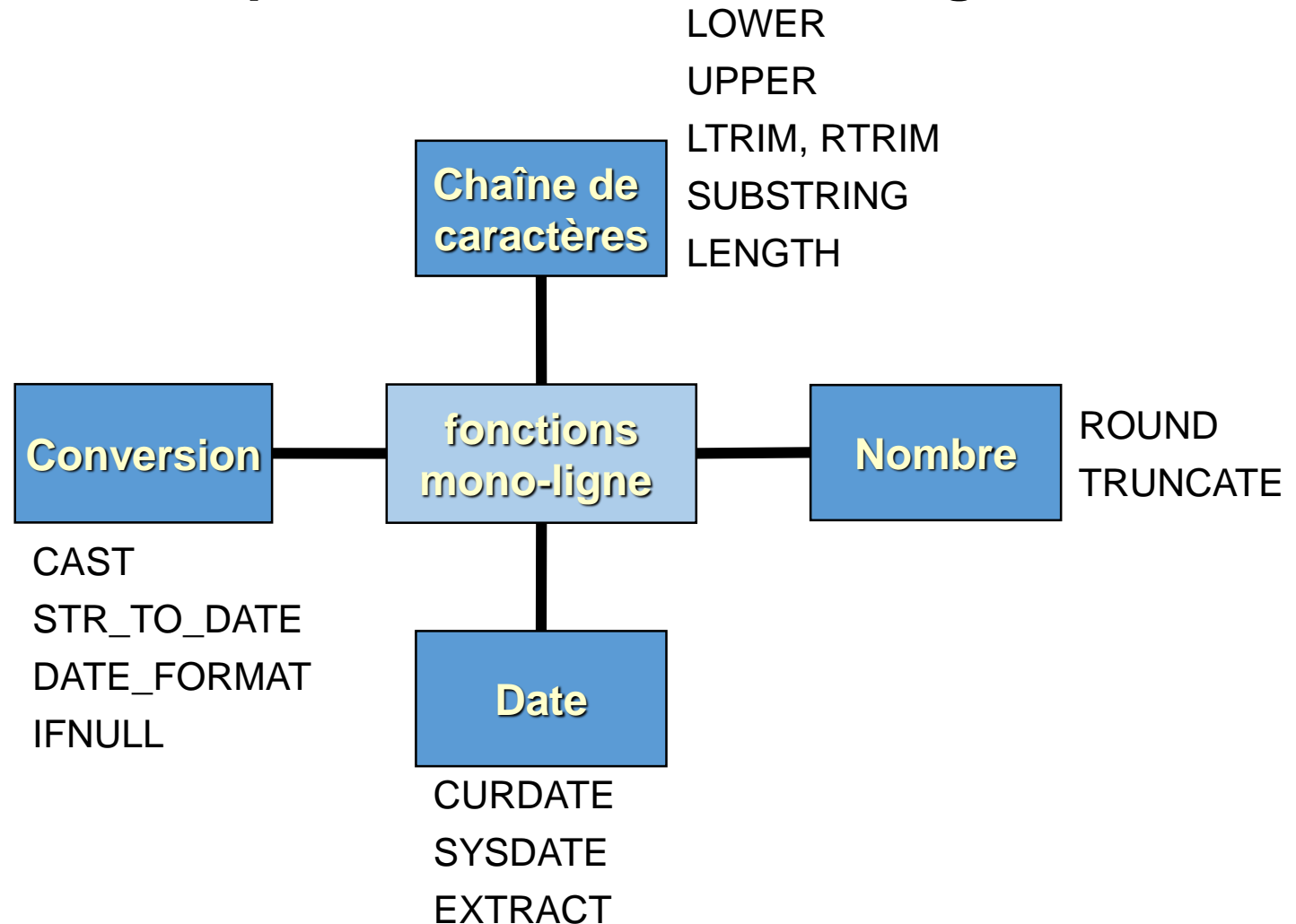
Définition

Chaînes

Numériques

Dates

➤ Exemples de fonctions mono-ligne :



Les fonctions

Définition

Chaînes

Numériques

Dates

➤ Exemple de fonctions de chaîne :

Fonction	Description
CONCAT(c1,c2, ...)	Concatène plusieurs chaînes
LOWER(c)	Tout en minuscules.
LOCATE(c1,c2,pos)	Premier indice d'une sous-chaîne c1 dans une chaîne c2 à partir de la position pos.
CHAR_LENGTH(c)	Longueur de la chaîne en nombre de caractères.
LEFT(c,n)	Extrait les n premiers caractères de c en partant de la gauche.
RIGHT(c,n)	Extrait les n derniers caractères de c en partant de la droite.
SUBSTR(c,n,[t])	Extraction de la sous-chaîne c commençant à la position n et de longueur t caractères.
UPPER (c)	Tout en majuscules.
REPLACE(c1,c2,c3)	Recherche les chaînes c2 présentes dans c1 et les remplace par c3.

Les fonctions

Définition

Chaînes

Numériques

Dates

➤ Exemple de fonctions de chaîne :

```
SELECT ENAME, CONCAT(ENAME, JOB),  
Char_LENGTH(ENAME), INSTR(ENAME, 'A') FROM EMP  
WHERE SUBSTR(JOB, 1, 5) = 'SALES';
```

ENAME	CONCAT(ENAME, JOB)	Char_LENGTH(ENAME)	INSTR(ENAME, 'A')
ALLEN	ALLENSALESMAN	5	1
WARD	WARDSALESMAN	4	2
MARTIN	MARTINSALESMAN	6	2
TURNER	TURNERSALESMAN	6	0

Définition

Chaînes

Numériques

Dates

➤ Exemple de fonctions numériques :

Fonction	Description
POW(m,n)	m puissance n.
RADIANS(n)	Conversion de degrés en radians.
RAND()	Flottant aléatoire (à 14 décimales) entre 0 et 1.
ROUND(m,n)	Arrondi à une ou plusieurs décimales.
SIGN(n)	Retourne le signe d'un nombre (-1, 0 ou 1).
SIN(n)	Sinus de n
SINH(n)	Sinus hyperbolique de n.
SQRT(n)	Racine carrée de n.
TAN(n)	Tangente de n .
TRUNCATE(n,m)	Coupure de n à m décimales.

➤ Exemple de fonctions numériques :

- ✓ Affiche la valeur 45.923 arrondi au centième, à l'unité et à la dizaine

```
SELECT ROUND (45.923, 2) ,  
ROUND (45.923, 0) , ROUND (45.923, -1) ;
```

ROUND(45.923,2)	ROUND(45.923,0)	ROUND(45.923,-1)
45.92	46	50

Les fonctions

Définition

Chaînes

Numériques

Dates

➤ Exemple de fonctions de date :

Fonction	Description
MONTH(<i>date</i>), MONTHNAME(<i>date</i>)	Retourne respectivement le numéro et le nom du mois d'une date-heure.
NOW()	Date et heure courantes au format 'YYYYMM-DD HH:MM:SS' ou YYYYMMDDHHMMSS.
SECOND(<i>time</i>)	Extrait les secondes d'un temps.
SEC_TO_TIME(<i>secondes</i>)	Construit une heure au format 'HH:MM:SS' ou HHMMSS.
STR_TO_DATE(<i>c,format</i>)	Construit une date (heure) selon un certain format. C'est l'inverse de DATE_FORMAT().
SYSDATE()	Date et heure courantes au format 'YYYYMM-DD HH:MM:SS' ou YYYYMMDDHHMMSS
DAYOFYEAR(<i>date</i>)	Numéro du jour dans l'année (0 à 366).
DAY(<i>date</i>) ou DAYOFMONTH(<i>date</i>)	Numéro du jour dans le mois (0 à 31).
TIMEDIFF(<i>tdate1,tdate2</i>)	Temps entre 2 temps ou 2 dates ou 2 dates-heure.

Les fonctions

Définition

Chaînes

Numériques

Dates

➤ Exemple d'une fonction de date :

```
SELECT ENAME, (SYSDATE() - HIREDATE) / 7 WEEKS  
FROM EMP  
WHERE DEPTNO = 10
```

ENAME	WEEKS
CLARK	2887257136185.0000
KING	2887257136112.4286
MILLER	2887257134825.8571

Les fonctions

Définition

Chaînes

Numériques

Dates

- Une date possède un format d'affichage :
 - ✓ Les formats sont différents selon le SGBD
 - ✓ Le format est défini entre simples apostrophes (il s'agit d'une chaîne de caractères)

- Exemple :

```
Select ENAME, DATE_FORMAT(HIREDATE, '%d %M %Y') HIREDATE FROM EMP;
```

ENAME	HIREDATE
SMITH	17 December 2019
ALLEN	20 February 2020
WARD	22 February 2020
JONES	02 April 2020
MARTIN	28 September 2020
BLAKE	01 May 2020
CLARK	09 June 2020
KING	17 November 2020
TURNER	08 September 2020
JAMES	03 December 2020
FORD	03 December 2020
MILLER	23 January 2021

Les fonctions

Définition

Chaînes

Numériques

Dates

- Conversion d'une valeur NULL : le domaine de l'attribut doit correspondre à la valeur de remplacement du NULL

- ✓ IFNULL(HIREDATE, '2000-01-01')

- ✓ IFNULL(JOB, 'No job yet')

- Exemple :

```
SELECT ENAME, SAL, COMM, (SAL * 12) +  
IFNULL(COMM, 0) FROM EMP
```

ENAME	SAL	COMM	(SAL * 12) + IFNULL(COMM, 0)
SMITH	800.00	NULL	9600.00
ALLEN	1600.00	300.00	19500.00
WARD	1250.00	500.00	15500.00
JONES	2975.00	NULL	35700.00
MARTIN	1250.00	1400.00	16400.00
BLAKE	2850.00	NULL	34200.00
CLARK	2450.00	NULL	29400.00
KING	5000.00	NULL	60000.00
TURNER	1500.00	0.00	18000.00
JAMES	950.00	NULL	11400.00
FORD	3000.00	NULL	36000.00
MILLER	1300.00	NULL	15600.00

Définition

Chaînes

Numériques

Dates

Exercice : Ecrire les requêtes

- Donner les noms des employés en minuscule.
- Donner le numéro et la commission reçue par chaque employé embauché avant le 1^{er} août 1981, uniquement pour les employés qui en ont une.
Le résultat est affiché par ordre décroissant de commission.
- Donner les noms et les numéros de département des employés. Si le numéro de département n'est pas renseigné, alors afficher « Département inconnu ». L'affichage comporte deux colonnes nommées « nom » et « dept ».

Groupes de données

Les groupes de données

Définition

Fonctions

La clause GROUP BY

La clause HAVING

- Les fonctions de groupe retournent un seul résultat par groupe de lignes
- Un groupe peut être constitué de toute la table ou d'un sous-ensemble de lignes de la table
- Les fonctions de groupe permettent de faire des statistiques

EMP		
SAL	DEPTNO	
800	20	“le salaire maximum de la relation EMP”
1600	30	
1250	30	
2975	20	
		MAX(SAL)
		2975

Les groupes de données

Définition

Fonctions

La clause GROUP BY

La clause HAVING

- Calcul sur des groupes de données de types numériques
 - ✓ `AVG([All|Distinct] attribut)` : retourne la moyenne
 - ✓ `SUM ([All|Distinct] attribut)` : retourne la somme
 - ✓ `VARIANCE ([All|Distinct] attribut)` : retourne la variance

- Calcul sur des groupes de n'importe quel type
 - ✓ `MIN ([All|Distinct] attribut)` : renvoie le minimum
 - ✓ `MAX ([All|Distinct] attribut)` : renvoie le maximum
 - ✓ `COUNT (* | [All|Distinct] attribut)` : renvoie le nb de valeurs connues.

- Les fonctions de groupe ne tiennent pas compte des NULLs.

Les groupes de données

Définition

Fonctions

La clause
GROUP BY

La clause
HAVING

Les fonctions de groupe sur toute la table

```
SELECT      fonction_groupe(attributes)  
FROM        liste_relations  
WHERE       liste_conditions];
```

- Renvoie une seule valeur
- Exemple:

```
SELECT count(comm) from EMP;
```

count(comm)
4

```
SELECT avg(sal) from EMP;
```

avg(sal)
2077.083333

Les groupes de données

Définition

Fonctions

La clause
GROUP BY

La clause
HAVING

La clause GROUP BY

```
SELECT      fonction_groupe(attributs) |  
            attributs_groupe  
FROM        liste_relations  
[WHERE      liste_conditions]  
[GROUP BY   attributs_groupe]  
[ORDER BY   liste_colonnes];
```

- La clause GROUP BY découpe une relation en plusieurs groupes
- Un groupe se compose des lignes qui ont la même valeur pour les attributs_groupe
- Un attribut_groupe est un attribut qui va avoir une valeur par groupe
- Un ordre SELECT avec une clause GROUP BY rend une ligne résultat pour chaque groupe

Les groupes de données

Définition

Fonctions

La clause
GROUP BY

La clause
HAVING

La clause GROUP BY

- Exemple: quel est le salaire moyen par département?

```
SELECT deptno, avg(sal)
FROM EMP
GROUP BY deptno;
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7839	KING	PRESIDENT	NULL	2020-11-17	5000.00	NULL	10
7934	MILLER	CLERK	7782	2021-01-23	1300.00	NULL	10
7369	SMITH	CLERK	7902	2019-12-17	800.00	NULL	20
7566	JONES	MANAGER	7839	2020-04-02	2975.00	NULL	20
7902	FORD	ANALYST	7566	2020-12-03	3000.00	NULL	20
7499	ALLEN	SALESMAN	7698	2020-02-20	1600.00	300.00	30
7521	WARD	SALESMAN	7698	2020-02-22	1250.00	500.00	30
7654	MARTIN	SALESMAN	7698	2020-09-28	1250.00	1400.00	30
7698	BLAKE	MANAGER	7839	2020-05-01	2850.00	NULL	30
7844	TURNER	SALESMAN	7698	2020-09-08	1500.00	0.00	30
7900	JAMES	CLERK	7698	2020-12-03	950.00	NULL	30

→ 10, avg(5000, 1300)

→ 20, avg(800, 2975, 3000)

→ 30, avg(1600, 1250, ...)

deptno	avg(sal)
10	2916.666667
20	2258.333333
30	1566.666667

Les groupes de données

Définition

Fonctions

La clause
GROUP BY

La clause
HAVING

Propriétés

- Dans le SELECT tous les attributs qui ne sont pas dans une fonction de groupe doivent apparaître après la clause GROUP BY

```
SELECT    deptno, AVG(sal)
FROM      emp
GROUP BY  deptno;
```

- L'attribut du GROUP BY ne doit pas forcément être présent dans le SELECT

```
SELECT          AVG(sal)
FROM            emp
GROUP BY        deptno;
```

Les groupes de données

Définition

Fonctions

La clause
GROUP BY

La clause
HAVING

Propriétés

- La clause GROUP BY peut s'utiliser avec plusieurs colonnes.
- Exemple: Dans EMP, quelle est la somme des salaires pour chaque job, regroupé par département ?

```
SELECT deptno, job, sum(sal)
FROM EMP
GROUP BY deptno, job
Order by deptno, job;
```

deptno	ename	job	sal
10	MILLER	CLERK	1300.00
10	CLARK	MANAGER	2450.00
10	KING	PRESIDENT	5000.00
20	FORD	ANALYST	3000.00
20	SMITH	CLERK	800.00
20	JONES	MANAGER	2975.00
30	JAMES	CLERK	950.00
30	BLAKE	MANAGER	2850.00
30	ALLEN	SALESMAN	1600.00
30	WARD	SALESMAN	1250.00
30	MARTIN	SALESMAN	1250.00
30	TURNER	SALESMAN	1500.00

deptno	job	sum(sal)
10	CLERK	1300.00
10	MANAGER	2450.00
10	PRESIDENT	5000.00
20	ANALYST	3000.00
20	CLERK	800.00
20	MANAGER	2975.00
30	CLERK	950.00
30	MANAGER	2850.00
30	SALESMAN	5600.00

Les groupes de données

Définition

Fonctions

La clause GROUP BY

La clause HAVING

- HAVING permet de mettre une condition sur les groupes : appliquer une restriction sur les groupes
 1. Les lignes sont regroupées
 2. Les fonctions de groupe sont calculées
 3. Les groupes sont sélectionnés en fonction des prédicats contenus dans la clause HAVING

```
SELECT      fonction_groupe(attributes) |  
              attributes_groupe  
FROM        liste_relations  
[WHERE      liste_conditions]  
[GROUP BY   attributes_groupe]  
[HAVING     liste_conditions_groupe]  
[ORDER BY   liste_colonnes];
```


Les groupes de données

Définition

Fonctions

La clause GROUP BY

La clause HAVING

- **Exemple :** Quels départements ont un salaire moyen supérieur à \$2900 ?

```
SELECT      deptno, avg(sal)
FROM        EMP
GROUP BY    deptno
HAVING      avg(sal) > 2000
ORDER BY    avg(sal) ;
```

deptno	avg(sal)
20	2258.333333
10	2916.666667

Les sous-requêtes

Les sous-requêtes

Définition

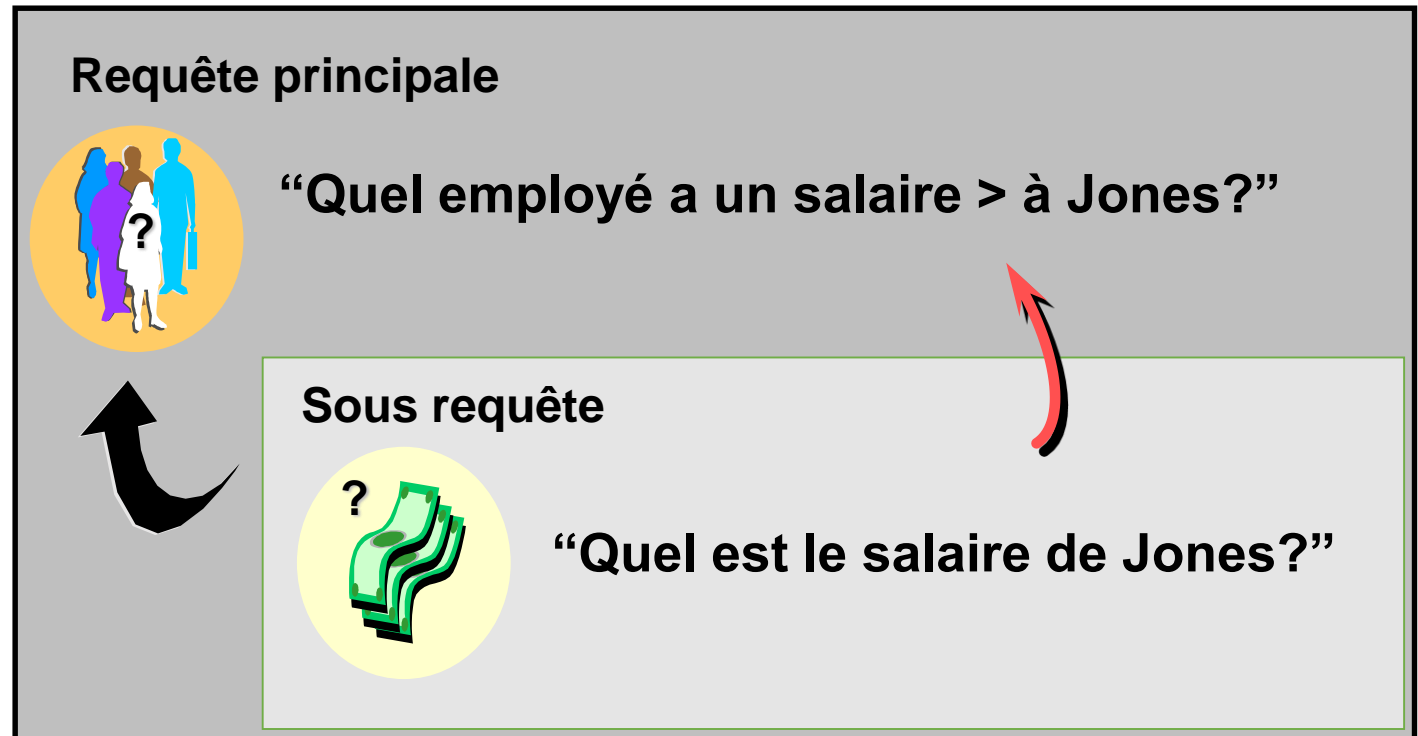
Mono-valeur

Multi-valeurs

Opérateur

EXISTS

- Une solution au problème :
“Qui a un salaire supérieur à celui de Jones?”



Les sous-requêtes

Définition

Mono-valeur

Multi-valeurs

Opérateur

EXISTS

```
SELECT      liste_attributs
FROM        liste_relations
WHERE       attribut opérateur
              (SELECT liste_attributs
                FROM    relation)
[ORDER BY   liste_colonnes];
```

- Les sous-requêtes sont entre ()
- Les sous-requêtes sont à droite de l'opérateur de comparaison
- Pas de ORDER BY dans une sous-requête (mais c'est possible dans la requête principale)
- La sous-requête s'exécute avant la requête principale
- Le résultat de la sous requête est utilisé dans la requête principale
- Si la sous-requête ne renvoie rien, elle fournit un NULL à la requête principale

Les sous-requêtes

Définition

Mono-valeur

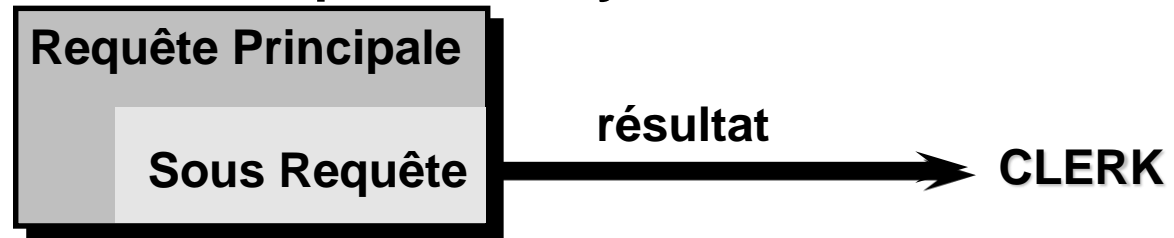
Multi-valeurs

Opérateur

EXISTS

➤ Types de sous-requêtes :

- ✓ Sous-requête renvoyant une seule valeur



- ✓ Sous-requête renvoyant plusieurs valeurs



- ✓ Sous-requête renvoyant une relation

Définition

Mono-valeur

Multi-valeurs

Opérateur

EXISTS

➤ Types de sous-requêtes :

- ✓ **La sous-requête est exécutée une seule fois avant la requête principale :**
 - Indépendance entre la sous-requête et la requête principale
 - Une seule exécution

- ✓ **La sous-requête est exécutée pour chaque ligne de la requête principale : sous-requête synchronisée (corrélée, liée, imbriquée) avec la requête principale :**
 - Dépendance entre la sous-requête et la requête principale
 - Plusieurs exécutions de la sous-requête (une par ligne de la requête principale)

Les sous-requêtes

Définition

Mono-valeur

Multi-valeurs

Opérateur

EXISTS

✓ La sous-requête renvoie une seule valeur (une seule colonne et une seule ligne)

✓ Opérateur de comparaison possible :
= < > <= >= <>

✓ Exemple:

```
SELECT ENAME, JOB
FROM EMP
WHERE JOB=(SELECT JOB FROM EMP WHERE EMPNO =
7369)
AND SAL > (SELECT SAL FROM EMP WHERE EMPNO =
7654) ;
```

ENAME	
MILLER	CLERK

Les sous-requêtes

Définition

Mono-valeur

Multi-valeurs

Opérateur

EXISTS

- **Sous-requête et fonctions de groupe :**
 - ✓ Une sous-requête peut contenir une fonction de groupe

Exemple: tous les employés ayant un salaire supérieur au salaire minimal

```
SELECT ENAME, JOB, SAL
```

```
FROM EMP
```

```
WHERE SAL > (SELECT MIN(SAL) FROM EMP) ;
```

ENAME	JOB	SAL
ALLEN	SALESMAN	1600.00
WARD	SALESMAN	1250.00
JONES	MANAGER	2975.00
MARTIN	SALESMAN	1250.00
BLAKE	MANAGER	2850.00
CLARK	MANAGER	2450.00
KING	PRESIDENT	5000.00
TURNER	SALESMAN	1500.00
JAMES	CLERK	950.00
FORD	ANALYST	3000.00
MILLER	CLERK	1300.00

Les sous-requêtes

Définition

Mono-valeur

Multi-valeurs

Opérateur

EXISTS

- **Sous-requête et fonctions de groupe :**
 - ✓ On peut avoir une sous-requête dans la clause **Having**

Exemple: les départements ayant un salaire minimal supérieur au salaire minimal du département 20

```
SELECT DEPTNO, MIN(SAL)
FROM EMP
GROUP BY DEPTNO
Having MIN(SAL) > (SELECT MIN(SAL) FROM EMP WHERE
DEPTNO=20) ;
```

DEPTNO	MIN(SAL)
10	1300.00
30	950.00

Les sous-requêtes

Définition

Mono-valeur

Multi-valeurs

Opérateur

EXISTS

Erreurs courantes

- Si la sous-requête renvoie plusieurs lignes, on ne peut pas utiliser les mêmes opérateurs de comparaison
- Exemple:

```
SELECT EMPNO, ENAME  
FROM EMP  
WHERE SAL=(SELECT MIN(SAL) FROM EMP  
GROUP BY DEPTNO) ;
```

Error Code: 1242. Subquery returns more than 1 row

Les sous-requêtes

Définition

Mono-valeur

Multi-valeurs

Opérateur

EXISTS

- ✓ **Le résultat de la sous requête contient plusieurs lignes (mais une seule colonne)**
- ✓ **Utilisation d'opérateurs de comparaison travaillant sur plusieurs valeurs**

Opérateur	Signification
IN	= à l'un des membres de la liste
ANY	Compare la valeur à chaque valeur résultat de la sous requête
ALL	Compare la valeur à toutes les valeurs résultat de la sous requête

Les sous-requêtes

Définition

Mono-valeur

Multi-valeurs

Opérateur EXISTS

➤ Opérateurs de comparaison :

- ✓ **attribut IN (sous-requête)**
le prédicat est vrai si la valeur de l'attribut existe dans les valeurs renvoyées par la sous-requête
- ✓ **attribut op_comp ALL (sous-requête)**
le prédicat est vrai s'il est vrai pour chacune des valeurs renvoyées par la sous-requête
- ✓ **attribut op_comp ANY (sous-requête)**
le prédicat est vrai s'il est vrai pour au moins une des valeurs renvoyées par la sous-requête
- ✓ **Avec op_comp = '<' | '>' | '<>' | '='**

Les sous-requêtes

Définition

Mono-valeur

Multi-valeurs

Opérateur

EXISTS

➤ Exemple d'opérateur ANY :

```
SELECT EMPNO, ENAME, JOB, SAL  
FROM EMP WHERE JOB <> 'CLERK'  
AND SAL < ANY (SELECT SAL FROM EMP WHERE  
JOB = 'CLERK' ) ;
```

EMPNO	ENAME	JOB	SAL
7521	WARD	SALESMAN	1250.00
7654	MARTIN	SALESMAN	1250.00

Les sous-requêtes

Définition

Mono-valeur

Multi-valeurs

Opérateur

EXISTS

➤ Exemple d'opérateur ALL :

```
SELECT EMPNO, ENAME, JOB, SAL
FROM EMP
WHERE SAL > ALL (SELECT AVG(SAL) FROM EMP GROUP
BY DEPTNO) ;
```

EMPNO	ENAME	JOB	SAL
7566	JONES	MANAGER	2975.00
7839	KING	PRESIDENT	5000.00
7902	FORD	ANALYST	3000.00

Les sous-requêtes

Définition

Mono-valeur

Multi-valeurs

Opérateur

EXISTS

➤ Exemple d'opérateur IN :

Exemple : Quels sont les départements qui n'ont pas d'employés ?

```
SELECT DISTINCT deptno
FROM   DEPT
WHERE  deptno NOT IN (SELECT deptno
                      FROM   EMP ) ;
```

deptno
40

Les sous-requêtes

Définition

Mono-valeur

Multi-valeurs

Opérateur

EXISTS

- L'opérateur **EXISTS** permet de construire un prédicat évalué à vrai si la sous-requête renvoie au moins une ligne

```
SELECT liste_attributs
FROM      liste_relations
WHERE [NOT] EXISTS
        (SELECT liste_attributs
         FROM      relation
         WHERE     liste_conditions);
```

- **Exemple:** Trouver les départements dans lesquels aucun employé n'y travaille

```
SELECT d.DEPTNO
FROM DEPT d
WHERE NOT EXISTS (SELECT * FROM EMP e WHERE
d.DEPTNO = e.DEPTNO );
```

DEPTNO
40