



Université Djilali Liabès- Sidi Bel Abbès
2ème Année Master Probabilités Appliquées

Base de données avancée et système répartis

Mohammed Fethi KHALFI

2023-2024

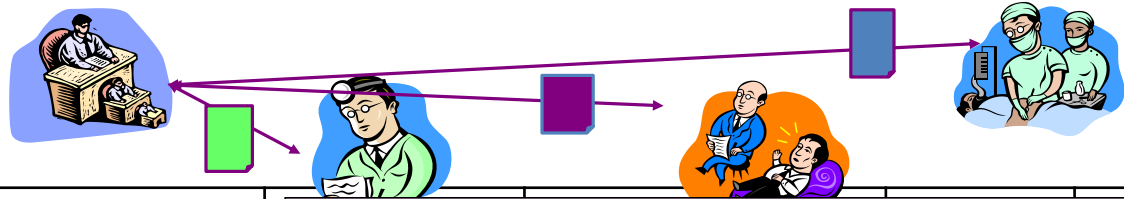
Fethi.Khalfi@yahoo.fr

Conception d'une base de données Relationnelle

SQL

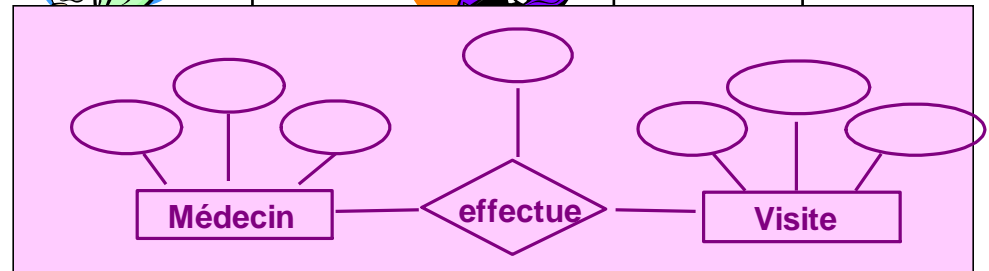
Modélisation à plusieurs niveaux

Réel



Modèle
conceptuel

- ♦ Indépendant du modèle de données
- ♦ Indépendant du SGBD



Modèle
logique

- ♦ Dépendant du modèle de données
- ♦ Indépendant du SGBD

Relationnel

Objet

XML

Modèle
Physique

- ♦ Dépendant du modèle de données
- ♦ Dépendant du SGBD





Introduction

- Le langage **SQL** est l'un des éléments qui ont contribué au développement et au succès de l'approche relationnelle dans le monde des bases de données.
- La normalisation internationale du langage garantit la stabilité indépendamment du SGBD utilisés.



Introduction

- Langage de manipulation des données: Un ensemble d'instructions SQL se nomme une **requête : (Query)** . Il permet de :
 - **Faire des requêtes (Query)**
 - Insérer des uplets
 - supprimer des uplets
 - modifier des uplets

- Toutes les requêtes d'interrogations retournent comme résultat une **table ou une Valeur**.

Opérations relationnelles avec SQL

- Cette section présente l'expression en SQL des opérations de l'algèbre relationnelle vues précédemment :
 - projection,
 - sélection,
 - produits cartésiens et
 - jointures.
-
- Les exemples simples qui présentent les opérations relationnelles utilisent la base de données « casse »

Opérations relationnelles avec SQL

Voiture

NumVoit	Marque	Type	Type
1	Peugeot	404	Rouge
2	Citroen	SM	Noire
3	Opel	GT	Blanche
4	Peugeot	403	Blanche
5	Renault	Alpine A310	Rose
6	Renault	Floride	Bleue

Personne

NumAch	Nom	Age	Ville	Sexe
1	Nestor	96	Paris	M
2	Irma	20	Lille	F
3	Henri	45	Paris	M
4	Josette	34	Lyon	F
5	Jacques	50	Bordeaux	M

Vente

DateVente	Prix	NumVoit	NumAch
1985-12-03	10 000	1	1
1996-03-30	70 000	2	4
1998-06-14	30 000	4	1
2000-04-02	45 000	5	2

Opérations relationnelles avec SQL

■ PROJECTION (SELECT)

- L'opération de projection consiste à sélectionner la (les) colonne(s) que l'on veut voir figurer dans la table « résultat ».
- On spécifie la liste des colonnes à inclure dans cette table derrière l'instruction **SELECT** en les séparant par des virgules.
- Si l'on désire afficher toutes les colonnes, on les désigne par le caractère « * ».

Opérations relationnelles avec SQL

- **PROJECTION (SELECT)**
- Projection sur les colonnes '**Nom**' et '**Ville**' de la table 'Personne'.

- **SELECT Nom, Ville
FROM personne ;**

Nom	Ville
Nestor	Paris
Irma	Lille
Henri	Paris
Josette	Lyon
Jacques	Bordeaux

Personne

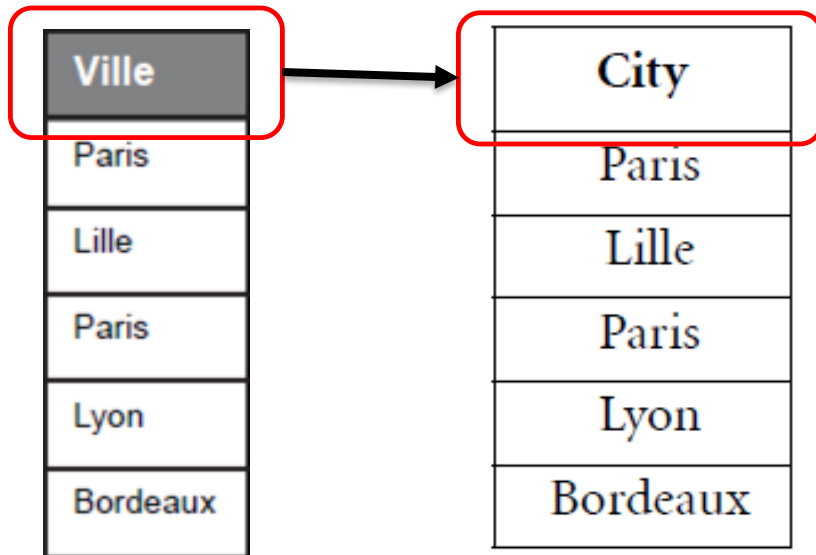
NumAch	Nom	Age	Ville	Sexe
1	Nestor	96	Paris	M
2	Irma	20	Lille	F
3	Henri	45	Paris	M
4	Josette	34	Lyon	F
5	Jacques	50	Bordeaux	M

- **SELECT *
FROM personne ;**



Opérations relationnelles avec SQL

- **PROJECTION (SELECT)** : Renommer une colonne
- Les colonnes de la table « résultat » peuvent être renommées par le mot clé **AS**.
- **SELECT** Ville **AS** City
FROM personne ;



Personne

NumAch	Nom	Age	Ville	Sexe
1	Nestor	96	Paris	M
2	Irma	20	Lille	F
3	Henri	45	Paris	M
4	Josette	34	Lyon	F
5	Jacques	50	Bordeaux	M

EXAMPLE

emp

EMP NO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	7902	17.12.1980	800		20
7499	ALLEN	SALESMAN	7698	20.2.1981	1600	300	30
7521	WARD	SALESMAN	7698	22.2.1981	1250	500	30
7566	JONES	MANAGER	7839	02.4.1981	2975		20
7654	MARTIN	SALESMAN	7698	28.9.1981	1250	1400	30
7698	BLAKE	MANAGER	7839	01.5.1981	2850		30
7782	CLARK	MANAGER	7839	09.6.1981	2450		10
7788	SCOTT	ANALYST	7566	19.4.1987	3000		20
7839	KING	PRESIDENT		17.11.1981	5000		10
7844	TURNER	SALESMAN	7698	08.9.1981	1500		30
7876	ADAMS	CLERK	7788	23.5.1987	1100		20
7900	JAMES	CLERK	7698	03.12.1981	950		30
7902	FORD	ANALYST	7566	03.12.1981	3000		20
7934	MILLER	CLERK	7782	23.1.1982	1300		10

Donner les Job et MGR de la table emp ?

Projection

```
SELECT job, mgr  
FROM emp;
```

JOB	MGR
CLERK	7902
SALESMAN	7698
SALESMAN	7698
MANAGER	7839
SALESMAN	7698
MANAGER	7839
MANAGER	7839
ANALYST	7566
PRESIDENT	
SALESMAN	7698
CLERK	7788
CLERK	7698
ANALYST	7566
CLERK	7782

Opérations relationnelles avec SQL

- **PROJECTION (SELECT)** : Valeurs distinctes d'une colonne
- Afin d'éliminer les doublons éventuels des valeurs d'une colonne de la table « résultat », on fait précéder le nom de la colonne par le mot clé **DISTINCT**.

Opérations relationnelles avec SQL

- **PROJECTION (SELECT)** : Valeurs distinctes d'une colonne

SELECT **DISTINCT** Marque
FROM voiture ;

Marque
Peugeot
Citroen
Opel
Renault

SELECT Marque
FROM voiture ;

Marque
Peugeot
Citroen
Opel
Peugeot
Renault
Renault

Voiture

NumVoit	Marque	Type	Type
1	Peugeot	404	Rouge
2	Citroen	SM	Noire
3	Opel	GT	Blanche
4	Peugeot	403	Blanche
5	Renault	Alpine A310	Rose
6	Renault	Florde	Bleue

Projection

Donner les Job et MGR uniques
de la table emp ?

JOB	MGR
CLERK	7902
SALESMAN	7698
SALESMAN	7698
MANAGER	7839
SALESMAN	7698
MANAGER	7839
MANAGER	7839
ANALYST	7566
PRESIDENT	
SALESMAN	7698
CLERK	7788
CLERK	7698
ANALYST	7566
CLERK	7782

Projection

```
SELECT DISTINCT job, mgr  
FROM emp;
```

JOB	MGR
CLERK	7902
SALESMAN	7698
MANAGER	7839
ANALYST	7566
PRESIDENT	
CLERK	7788
CLERK	7698
CLERK	7782

Fonctions

- *Fonctions d'agrégat* - elles permettent d'effectuer des traitements de globalisation sur un ensemble de uplets.
 - *COUNT* : retourne le nombre de uplets sélectionnés
 - *SUM* : retourne la somme des valeurs d'un attribut.
 - *AVG* : retourne la moyenne des valeurs d'un attribut.
 - *MIN, MAX* : retourne les valeurs mini et maxi, respectivement, d'un attribut
 - *VARIANCE* : retourne la variance d'un attribut.

Opérations relationnelles avec SQL

- **PROJECTION (SELECT)** : Utilisation de fonctions statistiques
- **SELECT** **AVG**(Prix) AS Prix_Moyen
FROM vente ;

Vente

DateVente	Prix	NumVoit	NumAch
1985-12-03	10 000	1	1
1996-03-30	70 000	2	4
1998-06-14	30 000	4	1
2000-04-02	45 000	5	2

Prix_Moyen
38 750.000 0

Opérations relationnelles avec SQL

- **PROJECTION (SELECT)** : Utilisation de fonctions statistiques
- **SELECT COUNT(*) AS Nombre_Personne**
FROM personne ;

Personne

NumAch	Nom	Age	Ville	Sexe
1	Nestor	96	Paris	M
2	Irma	20	Lille	F
3	Henri	45	Paris	M
4	Josette	34	Lyon	F
5	Jacques	50	Bordeaux	M

Nombre_Personne
5

Fonctions

- ***Fonctions d'agrégat.***

```
SELECT count(*) FROM emp WHERE deptno=20;
```

Count 5

```
SELECT AVG(sal) average FROM emp WHERE deptno=20;
```

Average 2175

```
SELECT count(Job) as Jobs FROM emp;  
SELECT count(DISTINCT Job) as Jobs FROM emp;
```

Jobs 14

Jobs 5

N'existe pas en Access

Opérations relationnelles avec SQL

- **SÉLECTION OU RESTRICTION (WHERE)**
- L'opération de **sélection** (ou **restriction**) consiste à indiquer un ou plusieurs critères pour choisir **les lignes** à inclure dans la table « résultat ».
- Ces critères utilisent évidemment le contenu des valeurs des colonnes.

Personne

NumAch	Nom	Age	Ville	Sexe
1	Nestor	96	Paris	M
2	Irma	20	Lille	F
3	Henri	45	Paris	M
4	Josette	34	Lyon	F
5	Jacques	50	Bordeaux	M

Opérations relationnelles avec SQL

- **SÉLECTION OU RESTRICTION (WHERE)**
- Le critère de sélection est indiqué à la suite du mot clé **WHERE**.
- Il est constitué d'expressions de conditions composées à l'aide **d'opérateurs de comparaison** et combinées à l'aide de **connecteurs logiques**.

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

Opérateurs

Voici la liste des opérateurs de comparaison classiques :

Opérateur	Opération
+, -	plus, négation
*, /	multiplication, division
+, -, (&)	addition, soustraction, concaténation
=, !=, <, >, <=, >=, IS NULL, LIKE, BETWEEN, IN	comparaison
NOT	booléenne négation
AND	conjonction
OR	disjonction

Opérations relationnelles avec SQL

■ SÉLECTION OU RESTRICTION (WHERE)

```
SELECT *  
FROM vente  
WHERE Prix > 50 000 ;
```

Vente

DateVente	Prix	NumVoit	NumAch
1985-12-03	10 000	1	1
1996-03-30	70 000	2	4
1998-06-14	30 000	4	1
2000-04-02	45 000	5	2

DateVente	Prix	NumVoit	NumAch
1996-03-30	70 000	2	4

Restriction

emp

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	7902	17.12.1980	800		20
7499	ALLEN	SALESMAN	7698	20.2.1981	1600	300	30
7521	WARD	SALESMAN	7698	22.2.1981	1250	500	30
7566	JONES	MANAGER	7839	02.4.1981	2975		20
7654	MARTIN	SALESMAN	7698	28.9.1981	1250	1400	30
7698	BLAKE	MANAGER	7839	01.5.1981	2850		30
7782	CLARK	MANAGER	7839	09.6.1981	2450		10
7788	SCOTT	ANALYST	7566	19.4.1987	3000		20
7839	KING	PRESIDENT		17.11.1981	5000		10
7844	TURNER	SALESMAN	7698	08.9.1981	1500		30
7876	ADAMS	CLERK	7788	23.5.1987	1100		20
7900	JAMES	CLERK	7698	03.12.1981	950		30
7902	FORD	ANALYST	7566	03.12.1981	3000		20
7934	MILLER	CLERK	7782	23.1.1982	1300		10

SELECT *
FROM emp

WHERE deptno=10;

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7782	CLARK	MANAGER	7839	09.6.1981	2450		10
7839	KING	PRESIDENT		17.11.1981	5000		10
7934	MILLER	CLERK	7782	23.1.1982	1300		10

Opérations relationnelles avec SQL

- SÉLECTION OU RESTRICTION (WHERE)

```
SELECT *  
FROM voiture  
WHERE Couleur IN ("Blanc","Rouge") ;
```

NumVoit	Marque	Type	Couleur
1	Peugeot	404	Rouge

Voiture

NumVoit	Marque	Type	Type
1	Peugeot	404	Rouge
2	Citroen	SM	Noire
3	Opel	GT	Blanche
4	Peugeot	403	Blanche
5	Renault	Alpine A310	Rose
6	Renault	Floride	Bleue

Opérations relationnelles avec SQL

- SÉLECTION OU RESTRICTION (WHERE)
- SELECT *
FROM personne
WHERE Age BETWEEN 40 AND 60;

NumAch	Nom	Age	Ville	Sexe
3	Henri	45	Paris	M
5	Jacques	50	Bordeaux	M

Personne

NumAch	Nom	Age	Ville	Sexe
1	Nestor	96	Paris	M
2	Irma	20	Lille	F
3	Henri	45	Paris	M
4	Josette	34	Lyon	F
5	Jacques	50	Bordeaux	M

Prédicats

BETWEEN / NOT BETWEEN

```
SELECT ename  
FROM emp  
WHERE hiredate BETWEEN  
1.1.81 AND 31.12.81;
```

ENAME
ALLEN
WARD
JONES
MARTIN
BLAKE
CLARK
KING
TURNER
JAMES
FORD

IN / NOT IN

```
SELECT ename  
FROM emp  
WHERE job In  
('ANALYST','MANAGER')  
;  
;
```

ENAME
SMITH
JONES
BLAKE
CLARK
ADAMS
JAMES
MILLER

Prédicats

- **LIKE/NOT LIKE** compare des chaînes de caractères.

```
SELECT ename, job  
FROM Emp  
WHERE ename LIKE "b*";
```

ENAME	JOB
BLAKE	MANAGER

```
SELECT ename, job, sal  
FROM emp  
WHERE ename LIKE '%mi%';
```

ename	job	sal
SMITH	CLERK	800
MILLER	CLERK	1300

Opérateurs

Voici la liste des opérateurs et connecteurs logiques:

Opérateur	Opération
+, -	plus, négation
*, /	multiplication, division
+, -, (&)	addition, soustraction, concaténation
=, !=, <, >, <=, >=, IS NULL, LIKE, BETWEEN, IN	comparaison
NOT	booléenne négation
AND	conjonction
OR	disjonction

Opérations relationnelles avec SQL

- SÉLECTION OU RESTRICTION (WHERE)
- SELECT *
FROM voiture
WHERE Couleur="Blanche" OR Marque="Peugeot";

NumVoit	Marque	Type	Couleur
1	Peugeot	404	Rouge
3	Opel	GT	Blanche
4	Peugeot	403	Blanche

Voiture

NumVoit	Marque	Type	Type
1	Peugeot	404	Rouge
2	Citroen	SM	Noire
3	Opel	GT	Blanche
4	Peugeot	403	Blanche
5	Renault	Alpine A310	Rose
6	Renault	Florde	Bleue

Opérations relationnelles avec SQL

- SÉLECTION OU RESTRICTION (WHERE)
- SELECT *
FROM personne
WHERE NOT (Ville='Paris');

NumAch	Nom	Age	Ville	Sexe
2	Irma	20	Lille	F
4	Josette	34	Lyon	F
5	Jacques	50	Bordeaux	M

Personne

NumAch	Nom	Age	Ville	Sexe
1	Nestor	96	Paris	M
2	Irma	20	Lille	F
3	Henri	45	Paris	M
4	Josette	34	Lyon	F
5	Jacques	50	Bordeaux	M

Restriction et projection

emp

EMP NO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	7902	17.12.1980	800		20
7499	ALLEN	SALESMAN	7698	20.2.1981	1600	300	30
7521	WARD	SALESMAN	7698	22.2.1981	1250	500	30
7566	JONES	MANAGER	7839	02.4.1981	2975		20
7654	MARTIN	SALESMAN	7698	28.9.1981	1250	1400	30
7698	BLAKE	MANAGER	7839	01.5.1981	2850		30
7782	CLARK	MANAGER	7839	09.6.1981	2450		10
7788	SCOTT	ANALYST	7566	19.4.1987	3000		20
7839	KING	PRESIDENT		17.11.1981	5000		10
7844	TURNER	SALESMAN	7698	08.9.1981	1500		30
7876	ADAMS	CLERK	7788	23.5.1987	1100		20
7900	JAMES	CLERK	7698	03.12.1981	950		30
7902	FORD	ANALYST	7566	03.12.1981	3000		20
7934	MILLER	CLERK	7782	23.1.1982	1300		10

```
SELECT ename, job, sal
FROM emp
WHERE job = 'MANAGER' AND sal>2500;
```

ENAME	JOB	SAL
JONES	MANAGER	2975
BLAKE	MANAGER	2850

Opérations relationnelles avec SQL

- **GROUPE (GROUP BY)**
- Les opérations de « groupage » regroupent les lignes d'une table par **valeurs** contenues dans une colonne.
- Pour réaliser cette opération avec SQL, on utilise le mot clé **GROUP BY** suivi du nom de la colonne sur laquelle s'effectue l'agrégat.

Opérations relationnelles avec SQL

- AGRÉGATS OU GROUPEMENT (GROUP BY)

- SELECT Marque
FROM voiture
GROUP BY Marque ;

Marque
Citroen
Opel
Peugeot
Renault

Voiture

NumVoit	Marque	Type	Type
1	Peugeot	404	Rouge
2	Citroen	SM	Noire
3	Opel	GT	Blanche
4	Peugeot	403	Blanche
5	Renault	Alpine A310	Rose
6	Renault	Floride	Bleue

Opérations relationnelles avec SQL

- **AGRÉGATS OU GROUPAGE (GROUP BY)**
- `SELECT Ville, AVG(Âge)`
`FROM personne`
`GROUP BY Ville ;`

Ville	Age
Bordeaux	50.0000
Lille	20.0000
Lyon	34.0000
Paris	70.5000

Personne

NumAch	Nom	Age	Ville	Sexe
1	Nestor	96	Paris	M
2	Irma	20	Lille	F
3	Henri	45	Paris	M
4	Josette	34	Lyon	F
5	Jacques	50	Bordeaux	M

Regroupements

emp

EMP NO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	7902	17.12.1980	800		20
7499	ALLEN	SALESMAN	7698	20.2.1981	1600	300	30
7521	WARD	SALESMAN	7698	22.2.1981	1250	500	30
7566	JONES	MANAGER	7839	02.4.1981	2975		20
7654	MARTIN	SALESMAN	7698	28.9.1981	1250	1400	30
7698	BLAKE	MANAGER	7839	01.5.1981	2850		30
7782	CLARK	MANAGER	7839	09.6.1981	2450		10
7788	SCOTT	ANALYST	7566	19.4.1987	3000		20
7839	KING	PRESIDENT		17.11.1981	5000		10
7844	TURNER	SALESMAN	7698	08.9.1981	1500		30
7876	ADAMS	CLERK	7788	23.5.1987	1100		20
7900	JAMES	CLERK	7698	03.12.1981	950		30
7902	FORD	ANALYST	7566	03.12.1981	3000		20
7934	MILLER	CLERK	7782	23.1.1982	1300		10

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

deptno	Minsal	Maxsal
10	1300	5000
20	800	3000
30	950	2850

Regroupements

emp

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	7902	17.12.1980	800		20
7499	ALLEN	SALESMAN	7698	20.2.1981	1600	300	30
7521	WARD	SALESMAN	7698	22.2.1981	1250	500	30
7566	JONES	MANAGER	7839	02.4.1981	2975		20
7654	MARTIN	SALESMAN	7698	28.9.1981	1250	1400	30
7698	BLAKE	MANAGER	7839	01.5.1981	2850		30
7782	CLARK	MANAGER	7839	09.6.1981	2450		10
7788	SCOTT	ANALYST	7566	19.4.1987	3000		20
7839	KING	PRESIDENT		17.11.1981	5000		10
7844	TURNER	SALESMAN	7698	08.9.1981	1500		30
7876	ADAMS	CLERK	7788	23.5.1987	1100		20
7900	JAMES	CLERK	7698	03.12.1981	950		30
7902	FORD	ANALYST	7566	03.12.1981	3000		20
7934	MILLER	CLERK	7782	23.1.1982	1300		10

```
SELECT deptno, MIN(sal), MAX (sal)
FROM emp WHERE job = 'CLERK'
GROUP BY deptno;
```

deptno	Minsal	Maxsal
10	1300	1300
20	800	1100
30	950	950

Opérations relationnelles avec SQL

- TRI DU RÉSULTAT D'UNE REQUÊTE
- On utilise le mot clé **ORDER BY** pour spécifier la (les) colonne(s) sur laquelle (lesquelles) on souhaite trier le résultat.
- Il est possible de préciser l'ordre de tri par les mots clés **ASC** (croissant par défaut) ou **DESC** (décroissant).

Opérations relationnelles avec SQL

- TRI DU RÉSULTAT D'UNE REQUÊTE
- SELECT Prix, DateVente
FROM vente
- ORDER BY Prix DESC ;

Prix	DateVente
70 000	1996-03-30
45 000	2000-04-02
30 000	1998-06-14
10 000	1985-12-03

Vente

DateVente	Prix	NumVoit	NumAch
1985-12-03	10 000	1	1
1996-03-30	70 000	2	4
1998-06-14	30 000	4	1
2000-04-02	45 000	5	2

Opérations relationnelles avec SQL

- **TRI DU RÉSULTAT D'UNE REQUÊTE**
- On peut indiquer **plusieurs critères de tri**, qui sont lus et traités de **gauche à droite** (ici, on trie d'abord par villes puis par âges).
- **SELECT** Nom, Age, Ville
FROM personne
ORDER BY Ville, Age ;

Nom	Age	Ville
Jacques	50	Bordeaux
Irma	20	Lille
Josette	34	Lyon
Henri	45	Paris
Nestor	96	Paris

Personne

NumAch	Nom	Age	Ville	Sexe
1	Nestor	96	Paris	M
2	Irma	20	Lille	F
3	Henri	45	Paris	M
4	Josette	34	Lyon	F
5	Jacques	50	Bordeaux	M

Ordonnancement

SELECT ename, deptno, sal
FROM emp
ORDER BY deptno, sal **DESC**;

ename	deptno	sal
KING	10	5000
CLARK	10	2450
MILLER	10	1300
FORD	20	3000
SCOTT	20	3000
JONES	20	2975
ADAMS	20	1100
SMITH	20	800
BLAKE	30	2850
ALLEN	30	1600
TURNER	30	1500
MARTIN	30	1250
WARD	30	1250

Opérations relationnelles avec SQL

- Gestion de tables
- Le langage SQL comprend une partie manipulation de données (LMD) pour gérer les tables. Les opérations sont :
 - Faire des requêtes (Query)
 - Insérer des uplets
 - supprimer des uplets
 - modifier des uplets

Opérations relationnelles avec SQL

- Gestion de tables : Création
- La création d'une table est une opération importante qu'il faut entreprendre avec soin.
- La forme générale de l'instruction de création de table est la suivante :
- **CREATE TABLE** <Nom de la table> (liste des colonnes avec leur type séparé par ,) ;

Opérations relationnelles avec SQL

- Gestion de tables : Création
- CREATE TABLE <Nom de la table> (liste des colonnes avec leur type séparé par ,) ;

```
CREATE TABLE voiture (
NumVoit INT,
Marque CHAR(40),
Type CHAR(30),
Couleur CHAR(20)
) ;
```

[illegible]

Opérations relationnelles avec SQL

■ Gestion de tables : Création : Type de données

INT	Entier standard (32 bits)
SMALLINT	Entier « petit » (16 bits)
REAL	Réel (taille spécifique au SGBD)
FLOAT(n)	Réel (représenté sur « n » bits)

CHAR(n)	Chaîne de caractères de longueur « n » (codage ASCII 1 octet)
VARCHAR(n)	Chaîne de caractères de longueur maximale « n » (codage ASCII 1 octet)
NCHAR(b)	Chaîne de caractères de longueur « n » (codage Unicode sur 2 octets)
NVARCHAR(b)	Chaîne de caractères de longueur maximale « n » (codage Unicode sur 2 octets)

BOOLEAN	Booléen
BLOB	<i>Binary Large Object</i> : permet de stocker tout type binaire (photo, fichier traitement de texte...)

DATE	Date
TIME[(n)]	Heure, n (optionnel) est le nombre de décimales représentant la fraction de secondes

Opérations relationnelles avec SQL

- **Gestion de tables : Suppression**
- La commande **DROP TABLE** permet de supprimer une table.

DROP TABLE *name*;

DROP TABLE Voiture;

[illegible]

Opérations relationnelles avec SQL

- Gestion de tables : Modification
- La commande **ALTER TABLE** permet de modifier la structure de la table, c'est-à-dire **d'ajouter** des colonnes.
- **ALTER TABLE** voiture
- **ADD COLUMN** enplus INT ;

NumVoit	Marque	Type	Couleur	enplus
1	Peugeot	404	Rouge	NULL
2	Citroen	SM	Noire	NULL
3	Opel	GT	Blanche	NULL
4	Peugeot	403	Blanche	NULL
5	Renault	Alpine A310	Rose	NULL
6	Renault	Floride	Bleue	NULL

Opérations relationnelles avec SQL

- Gestion de tables : Modification
- La commande **ALTER TABLE** permet de modifier la structure de la table, c'est-à-dire de **supprimer** des colonnes.
- **ALTER TABLE** voiture
- **DROP COLUMN** Couleur;

NumVoit	Marque	Type	enplus
1	Peugeot	404	NULL
2	Citroen	SM	NULL
3	Opel	GT	NULL
4	Peugeot	403	NULL
5	Renault	Alpine A310	NULL
6	Renault	Floride	NULL

Opérations relationnelles avec SQL

- Gestion de tables : Modification
- Il n'est pas possible de **modifier** directement le nom d'une colonne ou son type. Il faut pour cela écrire une série d'opérations.
- Exemple : Voici la suite d'instructions permettant la modification du nom de la colonne de nom '**Couleur**' de la table '**voiture**' en '**Teinte**' en changeant son type.

Opérations relationnelles avec SQL

- Gestion de tables : Modification

- ALTER TABLE voiture
ADD COLUMN Teinte CHAR(60);

NumVoit	Marque	Type	Couleur	Teinte
1	Peugeot	404	Rouge	NULL
2	Citroen	SM	Noire	NULL
3	Opel	GT	Blanche	NULL
4	Peugeot	403	Blanche	NULL
5	Renault	Alpine A310	Rose	NULL
6	Renault	Floride	Bleue	NULL

Opérations relationnelles avec SQL

- Gestion de tables : Modification
- **UPDATE** voiture
SET Teinte=Couleur ;

NumVoit	Marque	Type	Couleur	Teinte
1	Peugeot	404	Rouge	Rouge
2	Citroen	SM	Noire	Noire
3	Opel	GT	Blanche	Blanche
4	Peugeot	403	Blanche	Blanche
5	Renault	Alpine A310	Rose	Rose
6	Renault	Floride	Bleue	Bleue

Opérations relationnelles avec SQL

- Gestion de tables : Modification
- ALTER TABLE voiture
DROP COLUMN Couleur;

NumVoit	Marque	Type	Teinte
1	Peugeot	404	Rouge
2	Citroen	SM	Noire
3	Opel	GT	Blanche
4	Peugeot	403	Blanche
5	Renault	Alpine A310	Rose
6	Renault	Floride	Bleue

Opérations relationnelles avec SQL

- Gestion des données
- On dispose classiquement de trois opérations : l'insertion, la suppression et la mise à jour, pour gérer les **données** d'une table.

Opérations relationnelles avec SQL

- Gestion des données : INSERTION (INSERT INTO)
- L'insertion d'enregistrements dans une table peut se réaliser enregistrement par enregistrement (INSERT INTO).
- Insertion d'un enregistrement dans la table 'voiture'

```
INSERT INTO voiture (NumVoit, Marque, Couleur)  
VALUES (10,'Triumph','Bleue') ;
```

Opérations relationnelles avec SQL

- Gestion des données : INSERTION (INSERT INTO)
- INSERT INTO voiture
(NumVoit, Marque, Type, Couleur)
VALUES (1,'Peugeot',404,'Rouge') ;

NumVoit	Marque	Type	Teinte
1	Peugeot	404	Rouge
2	Citroen	SM	Noire
3	Opel	GT	Blanche
4	Peugeot	403	Blanche
5	Renault	Alpine A310	Rose
6	Renault	Floride	Bleue

NumVoit	Marque	Type	Couleur
1	Peugeot	404	Rouge
2	Citroen	SM	Noire
3	Opel	GT	Blanche
4	Peugeot	403	Blanche
5	Renault	Alpine A310	Rose
6	Renault	Floride	Bleue
10	Triumph	NULL	Bleue

Opérations relationnelles avec SQL

- **SUPPRESSION (DELETE FROM).**
- L'opération de suppression permet de supprimer un ensemble d'enregistrements (**lignes**).

DELETE FROM table [WHERE **condition**];

DELETE FROM voiture

WHERE **Couleur='Rouge'** ;

NumVoit	Marque	Type	Teinte
1	Peugeot	404	Rouge
2	Citroen	SM	Noire
3	Opel	GT	Blanche
4	Peugeot	403	Blanche
5	Renault	Alpine A310	Rose
6	Renault	Floride	Bleue

NumVoit	Marque	Type	Couleur
2	Citroen	SM	Noire
3	Opel	GT	Blanche
4	Peugeot	403	Blanche
5	Renault	Alpine A310	Rose
6	Renault	Floride	Bleue

Opérations relationnelles avec SQL

- Gestion des données : MODIFICATION (UPDATE).
- Pour cette opération, il faut préciser :
 - la (les) colonne(s) concernée(s) ;
 - les enregistrements pour lesquels on modifiera les nouvelle(s) valeurs.

Opérations relationnelles avec SQL

- Gestion des données : MODIFICATION (UPDATE).
- UPDATE personne
SET Ville='Paris-Centre'
WHERE Ville='Paris' ;

Personne

NumAch	Nom	Age	Ville	Sexe
1	Nestor	96	Paris	M
2	Irma	20	Lille	F
3	Henri	45	Paris	M
4	Josette	34	Lyon	F
5	Jacques	50	Bordeaux	M

NumAch	Nom	Age	Ville	Sexe
1	Nestor	96	Paris-Centre	M
2	Irma	20	Lille	F
3	Henri	45	Paris-Centre	M
4	Josette	34	Lyon	F
5	Jacques	50	Bordeaux	M

Opérations relationnelles avec SQL

- Gestion de tables : REQUÊTES SUR PLUSIEURS TABLES.
- Lorsque l'on utilise **plusieurs tables** dans une requête SQL, il peut exister une ambiguïté dans les expressions sur les noms de colonnes.
- En effet, deux tables peuvent avoir une colonne de **nom identique**.

Opérations relationnelles avec SQL

- Gestion de tables : REQUÊTES SUR PLUSIEURS TABLES.
- Pour cette raison, on préfixera le nom de la colonne par le nom de la table.
- `SELECT voiture.Marque, voiture.Couleur`
`FROM voiture ;`

Marque	Couleur
Peugeot	Rouge
Citroen	Noire
Opel	Blanche
Peugeot	Blanche
Renault	Rose
Renault	Bleue

Opérations relationnelles avec SQL

- Cette notation peut devenir rapidement fastidieuse si le nombre de tables est **élevé** et si leurs noms sont **longs**.
- Dans ce cas, on désigne la table par un **alias** plus commode, qui peut être réduit à une **simple lettre**, plutôt que par son nom complet.

Opérations relationnelles avec SQL

- L'alias est indiqué simplement à la suite du nom de la table ou à l'aide du mot clé **AS** qui est optionnel.
- **SELECT Vo.Marque, Vo.Couleur**
FROM voiture AS Vo;

Marque	Couleur
Peugeot	Rouge
Citroen	Noire
Opel	Blanche
Peugeot	Blanche
Renault	Rose
Renault	Bleue

Opérations relationnelles avec SQL

- Gestion de tables : REQUÊTES SUR PLUSIEURS TABLES: Produit cartésien
- Le produit cartésien est la combinaison de toutes les lignes d'une table avec toutes les lignes d'une autre table sans tenir aucun compte du « sens » associé aux données. C'est une opération qui n'a guère d'intérêt en pratique. En SQL, cette opération s'écrit simplement.
- `SELECT *`
`FROM personne, voiture ;`

Opérations relationnelles avec SQL

- Gestion de tables : REQUÊTES SUR PLUSIEURS
- SELECT *
FROM personne, voiture ;

Personne

NumAch	Nom	Age	Ville	Sexe
1	Nestor	96	Paris	M
2	Irma	20	Lille	F
3	Henri	45	Paris	M
4	Josette	34	Lyon	F
5	Jacques	50	Bordeaux	M

Voiture

NumVoit	Marque	Type	Type
1	Peugeot	404	Rouge
2	Citroen	SM	Noire
3	Opel	GT	Blanche
4	Peugeot	403	Blanche
5	Renault	Alpine A310	Rose
6	Renault	Florie	Bleue

Opérations relationnelles avec SQL

- Gestion de tables : REQUÊTES SUR PLUSIEURS
- SELECT *
FROM personne, voiture ;

NumAch	Nom	Age	Ville	Sexe	Num Voit	Marque	Type	Couleur
1	Nestor	96	Paris	M	1	Peugeot	404	Rouge
2	Irma	20	Lille	F	1	Peugeot	404	Rouge
3	Henri	45	Paris	M	1	Peugeot	404	Rouge
4	Josette	34	Lyon	F	1	Peugeot	404	Rouge
5	Jacques	50	Bordeaux	M	1	Peugeot	404	Rouge
1	Nestor	96	Paris	M	2	Citroen	SM	Noire

Opérations relationnelles avec SQL

- Gestion de tables : REQUÊTES SUR PLUSIEURS
- SELECT *
FROM personne, voiture ;

NumAch	Nom	Age	Ville	Sexe	Num Voit	Marque	Type	Couleur
2	Irma	20	Lille	F	2	Citroen	SM	Noire
3	Henri	45	Paris	M	2	Citroen	SM	Noire
4	Josette	34	Lyon	F	2	Citroen	SM	Noire
5	Jacques	50	Bordeaux	M	2	Citroen	SM	Noire
1	Nestor	96	Paris	M	3	Opel	GT	Blanche
2	Irma	20	Lille	F	3	Opel	GT	Blanche
3	Henri	45	Paris	M	3	Opel	GT	Blanche
4	Josette	34	Lyon	F	3	Opel	GT	Blanche
5	Jacques	50	Bordeaux	M	3	Opel	GT	Blanche
1	Nestor	96	Paris	M	4	Peugeot	403	Blanche
2	Irma	20	Lille	F	4	Peugeot	403	Blanche
3	Henri	45	Paris	M	4	Peugeot	403	Blanche
4	Josette	34	Lyon	F	4	Peugeot	403	Blanche
5	Jacques	50	Bordeaux	M	4	Peugeot	403	Blanche
1	Nestor	96	Paris	M	5	Renault	AlpineA 310	Rose

Opérations relationnelles avec SQL

- Gestion de tables : REQUÊTES SUR PLUSIEURS
- SELECT *
FROM personne, voiture ;

2	Irma	20	Lille	F	5	Renault	AlpineA 310	Rose
3	Henri	45	Paris	M	5	Renault	AlpineA 310	Rose
4	Josette	34	Lyon	F	5	Renault	AlpineA 310	Rose
5	Jacques	50	Bordeaux	M	5	Renault	AlpineA 310	Rose
1	Nestor	96	Paris	M	6	Renault	Floride	Bleue
2	Irma	20	Lille	F	6	Renault	Floride	Bleue
3	Henri	45	Paris	M	6	Renault	Floride	Bleue
4	Josette	34	Lyon	F	6	Renault	Floride	Bleue
5	Jacques	50	Bordeaux	M	6	Renault	Floride	Bleue

Opérations relationnelles avec SQL

- Gestion de tables : REQUÊTES SUR PLUSIEURS TABLES: Jointure interne (INNER JOIN)
- Il s'agit de l'opération de base de l'algèbre relationnelle. Elle permet de lier deux tables entre elles en introduisant un critère de « sens des données issu du monde réel.

Opérations relationnelles avec SQL

- Gestion de tables : REQUÊTES SUR PLUSIEURS TABLES: Jointure interne (INNER JOIN)
- SELECT voiture.Marque, voiture.Couleur, vente.Prix
FROM voiture, vente
WHERE voiture.NumVoit=vente.NumVoit ;

Voiture

NumVoit	Marque	Type	Type
1	Peugeot	404	Rouge
2	Citroen	SM	Noire
3	Opel	GT	Blanche
4	Peugeot	403	Blanche
5	Renault	Alpine A310	Rose
6	Renault	Floride	Bleue

Vente

DateVente	Prix	NumVoit	NumAch
1985-12-03	10 000	1	1
1996-03-30	70 000	2	4
1998-06-14	30 000	4	1
2000-04-02	45 000	5	2

Opérations relationnelles avec SQL

- Gestion de tables : REQUÊTES SUR PLUSIEURS TABLES: Jointure interne (INNER JOIN)
- SELECT voiture.Marque, voiture.Couleur, vente.Prix
FROM voiture, vente
WHERE voiture.NumVoit=vente.NumVoit ;

Marque	Couleur	Prix
Peugeot	Rouge	10 000
Citroen	Noire	70 000
Peugeot	Blanche	30 000
Renault	Rose	45 000

Opérations relationnelles avec SQL

- Gestion de tables : REQUÊTES SUR PLUSIEURS TABLES: Jointure interne (INNER JOIN)
- l'opération de jointure est complexe à réaliser pour un SGBD et est coûteuse en temps et en ressources.
- Le nombre de lignes de la table « résultat » est égal cette fois au nombre de lignes contenues dans les deux tables pour lesquelles le critère d'égalité des colonnes est respecté.
- Il est bien sûr possible d'effectuer la jointure sur plus de deux tables : on indique alors les différents critères de jointure entre les tables.

Opérations relationnelles avec SQL

- Gestion de tables : REQUÊTES SUR PLUSIEURS TABLES: Jointure interne (INNER JOIN)

- SELECT vo.Marque, vo.Couleur, ve.Prix, pe.Nom, pe.Age

FROM voiture AS vo, vente AS ve, personne AS pe

WHERE (vo.NumVoit=ve.NumVoit) AND
(pe.NumAch=ve. NumAch);

Opérations relationnelles avec SQL

- SELECT vo.Marque, vo.Couleur, ve.Prix, pe.Nom, pe.Age
FROM voiture AS vo, vente AS ve, personne AS pe
WHERE (vo.NumVoit=ve.NumVoit) AND
(pe.NumAch=ve. NumAch);

Vente

DateVente	Prix	NumVoit	NumAch
1985-12-03	10 000	1	1
1996-03-30	70 000	2	4
1998-06-14	30 000	4	1
2000-04-02	45 000	5	2

Personne

NumAch	Nom	Age	Ville	Sexe
1	Nestor	96	Paris	M
2	Irma	20	Lille	F
3	Henri	45	Paris	M
4	Josette	34	Lyon	F
5	Jacques	50	Bordeaux	M

Voiture

NumVoit	Marque	Type	Type
1	Peugeot	404	Rouge
2	Citroen	SM	Noire
3	Opel	GT	Blanche
4	Peugeot	403	Blanche
5	Renault	Alpine A310	Rose
6	Renault	Florie	Bleue

Marque	Couleur	Prix	Nom	Age
Peugeot	Rouge	10 000	Nestor	96
Citroen	Noire	70 000	Josette	34
Peugeot	Blanche	30 000	Nestor	96
Renault	Rose	45 000	Irma	20