LE LANGAGE SQL

LMD : Langage de Manipulation des Données (Suite)

M. RABI

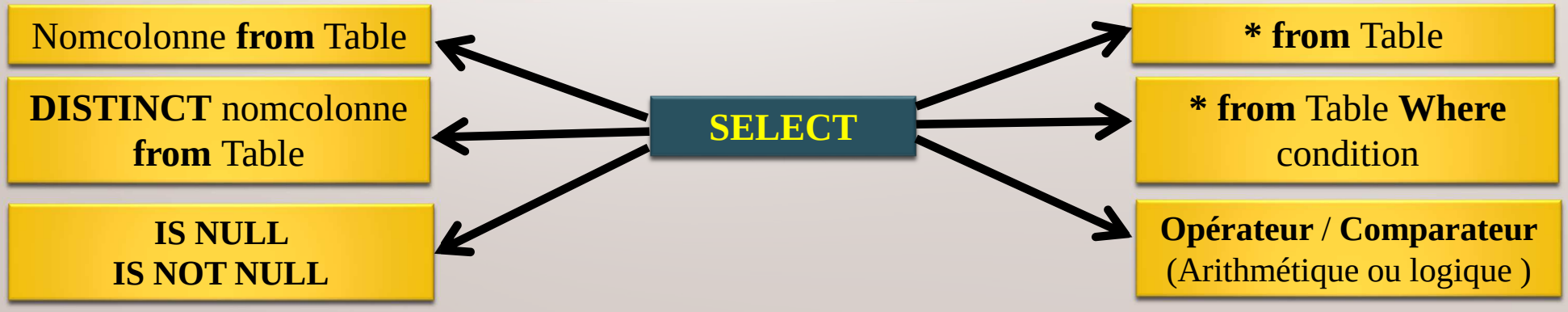
LE LANGAGE SQL (Rappel)

Définition

Le SQL (Structured Query Language) est un langage permettant la manipulation et la communication avec une base de données.

Fonctionnalités

Langage Manipulation de données (LMD)
Langage Définition de données (LDD)
Langage Contrôle de données (LCD)



2. Le langage de manipulation de données (LMD)

2.3 Le tri des résultats

La commande ORDER BY

La commande ORDER BY permet de trier les lignes d'une requête SQL. Il est possible de trier les données sur une ou plusieurs colonnes, par ordre ascendant ou descendant.

Syntaxe :

```
SELECT colonne1, colonne2
FROM table
ORDER BY colonne1
```

Par défaut les résultats sont classés par ordre ascendant, toutefois il est possible d'inverser l'ordre en utilisant le suffixe DESC après le nom de la colonne. Par ailleurs, il est possible de trier sur plusieurs colonnes en les séparant par une virgule.

```
SELECT colonne1, colonne2, colonne3
FROM table
ORDER BY colonne1 DESC, colonne2 ASC
```

2. Le langage de manipulation de données (LMD)

2.3 Le tri des résultats

La commande ORDER BY

Exemple 1

Utilisateur

id	nom	prenom	date_inscription	tarif_total
1	Durand	Maurice	2012-02-05	145
2	Dupond	Fabrice	2012-02-07	65
3	Durand	Fabienne	2012-02-13	90
4	Dubois	Chloé	2012-02-16	98
5	Dubois	Simon	2012-02-23	27

Pour récupérer la liste des utilisateurs classée par ordre alphabétique du nom de famille.

```
SELECT * FROM utilisateur ORDER BY nom
```

id	nom	prenom	date_inscription	tarif_total
4	Dubois	Chloé	2012-02-16	98
5	Dubois	Simon	2012-02-23	27
2	Dupond	Fabrice	2012-02-07	65
1	Durand	Maurice	2012-02-05	145
3	Durand	Fabienne	2012-02-13	90

2. Le langage de manipulation de données (LMD)

2.3 Le tri des résultats

La commande ORDER BY

Exemple 2 : Si on souhaite afficher la liste des utilisateurs classée par ordre alphabétique du nom et pour ceux qui ont le même nom de famille, les trier par ordre décroissant d'inscription.

```
SELECT *
FROM utilisateur
ORDER BY nom, date_inscription DESC
```

```
SELECT *
FROM utilisateur
ORDER BY nom ASC, date_inscription DESC
```

id	nom	prenom	date_inscription	tarif_total
5	Dubois	Simon	2012-02-23	27
4	Dubois	Chloé	2012-02-16	98
2	Dupond	Fabrice	2012-02-07	65
3	Durand	Fabienne	2012-02-13	90
1	Durand	Maurice	2012-02-05	145

2. Le langage de manipulation de données (LMD)

2.4 Regroupement des résultats

La commande GROUP BY

Il peut être intéressant de regrouper des résultats afin de faire des opérations par groupe (opérations statistiques par exemple). Cette opération se réalise à l'aide de la clause *GROUP BY*, suivie du nom de chaque colonne sur laquelle on veut effectuer des regroupements.

Syntaxe :

```
SELECT colonne1, fonction(colonne2)
FROM table
GROUP BY colonne1
```

Fonctions :

AVG	Calcule la moyenne d'un set de valeurs
COUNT	Calcule le nombre de lignes
MAX	Récupérer la valeur maximale
MIN	Récupérer la valeur minimale
SUM	Calcule la somme d'un set de valeurs

2. Le langage de manipulation de données (LMD)

2.4 Regroupement des résultats

Exemple : Prenons en considération une table «achat » qui résume les ventes d'une boutique.

id	client	tarif	date_achat
1	Pierre	102	2012-10-23
2	Simon	47	2012-10-27
3	Marie	18	2012-11-05
4	Marie	20	2012-11-14
5	Pierre	160	2012-12-03

Pour obtenir le coût total de chaque client :

```
SELECT client, SUM(tarif)
FROM achat
GROUP BY client
```

client	SUM(tarif)
Pierre	262
Simon	47
Marie	38

2. Le langage de manipulation de données (LMD)

2.4 Regroupement des résultats

La commande HAVING

La condition HAVING en SQL est presque similaire à WHERE à la seule différence que HAVING permet de filtrer en utilisant des fonctions telles que SUM(), COUNT(), AVG(), MIN() ou MAX().

Syntaxe :

```
SELECT * FROM nom_table
GROUP BY colonne1
HAVING fonction(colonne2) operateur valeur
```

Cela permet donc de sélectionner toutes les colonnes de la table « nom_table » en GROUPANT les lignes qui ont des valeurs identiques sur la colonne « colonne1 » et que la condition de HAVING soit respectée.

2. Le langage de manipulation de données (LMD)

2.4 Regroupement des résultats

La commande HAVING

Exemple:

Achat

id	client	tarif	date_achat
1	Pierre	102	2012-10-23
2	Simon	47	2012-10-27
3	Marie	18	2012-11-05
4	Marie	20	2012-11-14
5	Pierre	160	2012-12-03

Si on souhaite récupérer la liste des clients qui ont commandé plus de 40€.

```
SELECT client, SUM(tarif)
FROM achat
GROUP BY client
HAVING SUM(tarif) > 40
```

client	SUM(tarif)
Pierre	262
Simon	47

2. Le langage de manipulation de données (LMD)

2.5 La jointure

Les jointures en SQL permettent d'associer plusieurs tables dans une même requête. Cela permet d'exploiter la puissance des bases de données relationnelles pour obtenir des résultats qui combinent les données de plusieurs tables de manière efficace.

Une jointure est un produit cartésien de deux tables. On appelle équijointure une jointure dont la qualification est une égalité entre deux colonnes.

En SQL, l'expression d'une jointure se fait en précisant le nom des colonnes des tables sur lesquelles on fait la jointure. La clause WHERE permet de préciser la qualification de la jointure.

Exemple

2. Le langage de manipulation de données (LMD)

2.5 La jointure

Exemple 1:

Voiture

Marque	Modele	Serie	Numero	Compteur
Renault	18	RL	4698 SJ 45	123450
Renault	Kangoo	RL	4568 HD 16	56000
Renault	Kangoo	RL	6576 VE 38	12000
Peugeot	106	KID	7845 ZS 83	75600
Peugeot	309	chorus	7647 ABY 82	189500
Ford	Escort	Match	8562 EV 23	
Fiat	Punto	GTI	8941 UD 61	
Audi	A4	Quattro	7846 AZS 75	21350

Societe

Nom	Pays
Renault	France
Fiat	Italie
Peugeot	France
Volkswagen	Allemagne
Ford	Etats-Unis

Afficher le pays d'origine des voitures par marque/modèle .

2. Le langage de manipulation de données (LMD)

2.5 La jointure

L'affichage des pays d'origine des voitures par marque/modèle se fait par la requête suivante :

```
SELECT Marque, Modele, Pays
FROM VOITURE,SOCIETE WHERE
Voiture.Marque = Societe.Nom
```

Version 2 :

```
SELECT
Voiture.Marque, Voiture.Modele, Societe.Pays
FROM
VOITURE,SOCIETE
WHERE
Voiture.Marque = Societe.Nom
```

Marque	Modele	Pays
Renault	18	France
Renault	Kangoo	France
Renault	Kangoo	France
Peugeot	106	France
Peugeot	309	France
Ford	Escort	Etats-Unis
Fiat	Punto	Italie

Il est possible de donner des alias aux noms des tables pour diminuer la taille des requêtes.

Version :3

```
SELECT V.Marque, V.Modele, S.Pays
FROM VOITURE V,SOCIETE S
WHERE V.Marque = S.Nom
```

2. Le langage de manipulation de données (LMD)

2.5 La jointure

Exemple 2:

Etudiant

CodeEtudiant	NomEtudiant	PrenomEtudiant	DateNaissance	VilleEtudiant	IdFiliere
E1	Rabi	Mouhcine	1987-12-23	Agadir	1
E2	Semlai	Laila	2000-12-10	Tanger	1
E3	Filali	Amine	1998-05-02	Tanger	2
E4	Amraoui	Najlae	2000-06-25	Agadir	3

Filiere

IdFiliere	NomFiliere
1	Informatique
3	Commerce
4	Chimie

Afficher le nom et prénom des étudiants avec leurs filières.

NomEtudiant	PrenomEtudiant	NomFiliere
Rabi	Mouhcine	Informatique
Semlai	Laila	Informatique
Filali	Amine	Chimie
Amraoui	Najlae	Commerce

```
SELECT
E.NomEtudiant, E.PrenomEtudiant, F.NomFiliere
FROM
etudiant E,filiere F
WHERE
E.IdFiliere = F.IdFiliere;
```

2. Le langage de manipulation de données (LMD)

2.6 Les Sous-requêtes

Effectuer une sous-requête consiste à effectuer une requête à l'intérieur d'une autre, ou en d'autres termes d'utiliser une requête afin d'en réaliser une autre (on entend parfois le terme de *requêtes en cascade*).

Syntaxe:

```
SELECT ……. FROM ……..
WHERE
…….. Opérateur (SELECT ….. FROM …..)
```

Opérateur (= , > , < , >= , <= …)

Renvoie une seule réponse (valeur)

Opérateur (IN, EXISTS, ALL, ANY)

Renvoie une seule ligne

2. Le langage de manipulation de données (LMD)

2.6 Les Sous-requêtes

Exemple 1 : Soit la table Voiture.

Marque	Modele	Serie	Numero	Compteur
Renault	18	RL	4698 SJ 45	123450
Renault	Kangoo	RL	4568 HD 16	56000
Renault	Kangoo	RL	6576 VE 38	12000
Peugeot	106	KID	7845 ZS 83	75600
Peugeot	309	chorus	7647 ABY 82	189500
Fiat	Punto	GTI	8941 UD 61	80232
Audi	A4	Quattro	7846 AZS 75	21350

La liste des voitures dont le compteur est inférieur à la moyenne des compteurs

2. Le langage de manipulation de données (LMD)

2.6 Les Sous-requêtes

La liste des voitures dont le compteur est inférieur à la moyenne

```
SELECT * FROM Voiture WHERE
Compteur < (SELECT AVG(Compteur) FROM Voiture)
```

Marque	Modele	Serie	Numero	Compteur
Renault	Kangoo	RL	4568 HD 16	56000
Renault	Kangoo	RL	6576 VE 38	12000
Peugeot	106	KID	7845 ZS 83	75600
Audi	A4	Quattro	7846 AZS 75	21350

2. Le langage de manipulation de données (LMD)

2.6 Les Sous-requêtes

Exemple 2 : Soit les deux tables suivantes :

Etudiant

CodeEtudiant	NomEtudiant	PrenomEtudiant	DateNaissance	VilleEtudiant	IdFiliere
E1	Rabi	Mouhcine	1987-12-23	Agadir	1
E2	Semlai	Laila	2000-12-10	Tanger	1
E3	Filali	Amine	1998-05-02	Tanger	4
E4	Amraoui	Najlae	2000-06-25	Agadir	3

Filiere

IdFiliere	NomFiliere
1	Informatique
2	Gestion
3	Commerce
4	Chimie
5	Droit

La liste des filières n'ayant aucun étudiant.

```
SELECT * FROM Filiere WHERE
IdFiliere NOT IN (SELECT Distinct IdFiliere FROM Etudiant)
```

IdFiliere	NomFiliere
2	Gestion
5	Droit

2. Le langage de manipulation de données (LMD)

2.6 Les opérations ensemblistes

Les opérations ensemblistes en SQL, sont celles définies dans l'algèbre relationnelle. (Union, Intersection, Différence)

La commande UNION

La commande UNION de SQL permet de mettre bout-à-bout les résultats de plusieurs requêtes
utilisant elles-mêmes la commande SELECT. C'est donc une commande qui permet de concaténer les résultats de deux requêtes ou plus. Pour l'utiliser il est nécessaire que chacune des requêtes à concaténer retournes le même nombre de colonnes, avec les mêmes types de données et dans le même ordre.

Syntaxe :

```
SELECT * FROM table1
UNION
SELECT * FROM table2
```

A savoir : par défaut, les enregistrements exactement identiques ne seront pas répétés dans les résultats. Pour effectuer une union dans laquelle même les lignes dupliquées sont affichées il faut utiliser la commande **UNION ALL**.

2. Le langage de manipulation de données (LMD)

2.6 Les opérations ensemblistes

La commande UNION

Exemple : Imaginons une entreprise qui possède plusieurs magasins et dans chacun de ces magasins il y a une table qui liste les clients.

« magasin1_client »

prenom	nom	ville	date_naissance	total_achat
Marion	Leroy	Lyon	1982-10-27	285
Paul	Moreau	Lyon	1976-04-19	133
Marie	Bernard	Paris	1993-07-03	75
Marcel	Martin	Paris	1976-11-24	39

« magasin2_client »

prenom	nom	ville	date_naissance	total_achat
Léon	Dupuis	Paris	1983-03-06	135
Marie	Bernard	Paris	1993-07-03	75
Sophie	Dupond	Marseille	1986-02-22	27
Marcel	Martin	Paris	1976-11-24	39

2. Le langage de manipulation de données (LMD)

2.6 Les opérations ensemblistes

La commande UNION

Sachant que certains clients sont présents dans les deux tables, pour éviter de retourner plusieurs fois les mêmes enregistrements, il convient d'utiliser la requête UNION.

```
SELECT * FROM magasin1_client
UNION
SELECT * FROM magasin2_client
```

prenom	nom	ville	date_naissance	total_achat
Léon	Dupuis	Paris	1983-03-06	135
Marie	Bernard	Paris	1993-07-03	75
Sophie	Dupond	Marseille	1986-02-22	27
Marcel	Martin	Paris	1976-11-24	39
Marion	Leroy	Lyon	1982-10-27	285
Paul	Moreau	Lyon	1976-04-19	133

2. Le langage de manipulation de données (LMD)

2.6 Les opérations ensemblistes

La commande INTERSECT

La commande SQL INTERSECT permet d'obtenir l'intersection des résultats de deux requêtes. Cette commande permet donc de récupérer les enregistrements communs à deux requêtes. Cela peut s'avérer utile lorsqu'il faut trouver s'il y a des données similaires sur deux tables distinctes.

Syntaxe :

```
SELECT * FROM table1
INTERSECT
SELECT * FROM table2
```

A savoir : pour l'utiliser convenablement il faut que les deux requêtes retourne le même nombre de colonnes, avec les mêmes types et dans le même ordre.

2. Le langage de manipulation de données (LMD)

2.6 Les opérations ensemblistes

La commande INTERSECT

Exemple : Prenons l'exemple de deux magasins qui appartiennent au même groupe. Chaque magasin possède sa table de clients.

« magasin1_client »

prenom	nom	ville	date_naissance	total_achat
Marion	Leroy	Lyon	1982-10-27	285
Paul	Moreau	Lyon	1976-04-19	133
Marie	Bernard	Paris	1993-07-03	75
Marcel	Martin	Paris	1976-11-24	39

« magasin2_client »

prenom	nom	ville	date_naissance	total_achat
Léon	Dupuis	Paris	1983-03-06	135
Marie	Bernard	Paris	1993-07-03	75
Sophie	Dupond	Marseille	1986-02-22	27
Marcel	Martin	Paris	1976-11-24	39

2. Le langage de manipulation de données (LMD)

2.6 Les opérations ensemblistes

La commande INTERSECT

Exemple :

```
SELECT * FROM magasin1_client
INTERSECT
SELECT * FROM magasin2_client
```

prenom	nom	ville	date_naissance	total_achat
Marie	Bernard	Paris	1993-07-03	75
Marcel	Martin	Paris	1976-11-24	39

Le résultat présente deux enregistrements, il s'agit des clients qui sont à la fois dans la table « magasin1_client » et dans la table « magasin2_client ».

2. Le langage de manipulation de données (LMD)

2.6 Les opérations ensemblistes

La commande EXCEPT / MINUS

(Cette commande s'appelle différemment selon les Systèmes de Gestion de Base de Données).

Dans le langage SQL la commande EXCEPT s'utilise entre deux requêtes pour récupérer les enregistrements de la première requête sans inclure les résultats de la deuxième. Si un même enregistrement devait être présent dans les résultats des deux requêtes, ils ne seront pas présent dans le résultat final.

Syntaxe :

```
SELECT * FROM table1
EXCEPT/MINUS
SELECT * FROM table2
```

2. Le langage de manipulation de données (LMD)

2.6 Les opérations ensemblistes

La commande EXCEPT / MINUS

Exemple : Imaginons un système informatique d'une entreprise qui contient deux tables contenant des listes de clients.

- Une table « Clients_inscrits » qui contient les prénoms, noms et date d'inscription de clients
- Une table « Clients_refus_email » qui contient les informations des clients qui ne souhaitent pas être contacté par email

« Clients_inscrits »

id	prenom	nom	date_inscription
1	Lionel	Martineau	2012-11-14
2	Paul	Cornu	2012-12-15
3	Sarah	Schmitt	2012-12-17
4	Sabine	Lenoir	2012-12-18

« Clients_refus_email »

id	prenom	nom	date_inscription
1	Paul	Cornu	2013-01-27
2	Manuel	Guillot	2013-01-27
3	Sabine	Lenoir	2013-01-29
4	Natalie	Petitjean	2013-02-03

Afficher uniquement le prénom et le nom des utilisateurs qui accepte de recevoir des emails informatifs.

2. Le langage de manipulation de données (LMD)

2.6 Les opérations ensemblistes

La commande EXCEPT / MINUS

Exemple :

La requête SQL à utiliser est la suivante :

```
SELECT prenom, nom FROM clients_inscrits
EXCEPT
SELECT prenom, nom FROM clients_refus_email
```

prenom	nom
Lionel	Martineau
Sarah	Schmitt

Ce tableau montre bien les utilisateurs qui sont dans la table **« Clients_inscrits »** et qui ne sont pas présents dans la deuxième table **« Clients_refus_email ».**

LE LANGAGE SQL

LMD : Langage de Manipulation des Données (... A suivre)