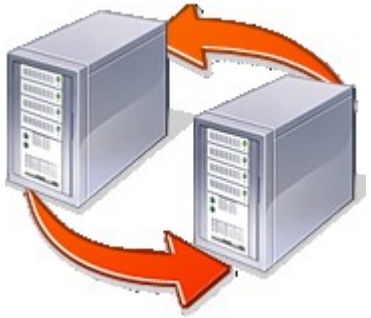
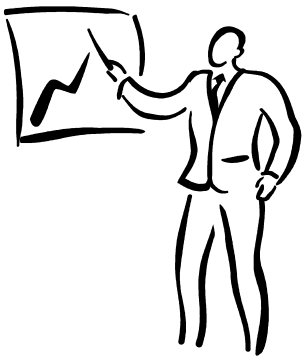
Le langage SQL
DML- Manipulation des données



Partie 1 : Rappels



Le langage SQL
DML- Manipulation des données

Les Objets logiques d'une Base de Données Relationnelle :

Les objets logiques représentent les objets perçus de l'extérieur de la base (voir Ansi/Sparc)

Les tables

Une BD relationnelle est perçue comme un ensemble de tables (ou relations).

Une table est constituée d'attributs (ou colonnes)

codecours	libellecours	nbjours
BR035	UML : Perfectionnement	2
BR034	UML : Initiation	2
BR013	Administration BDR	3
BR056	SQL2 : La norme	3

Le langage SQL
DML- Manipulation des données

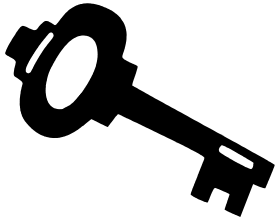
Les attributs (ou champs)

Ils constituent les colonnes des tables. Les valeurs d'un attribut sont élémentaires (ou atomiques), c'est-à-dire non décomposables.

Un certain nombre de contrôles à réaliser sur les valeurs permises pourront être renseignés au moment de la création de la table (constraint check) ou encore par l'intermédiaire de triggers (ou déclencheurs)

Le langage SQL
DML- Manipulation des données

Les Clés



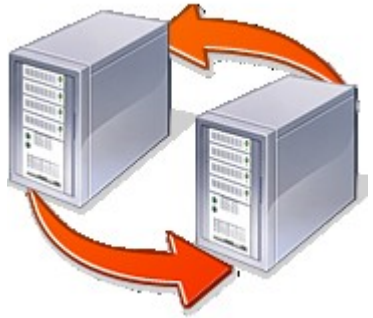
On trouvera :

- ✓ les ***clés primaires (primary key)***, une seule par table, qui sera constituée d'un ou plusieurs attributs, dont la valeur permet de déterminer un tuple (une ligne) unique dans une table (identifiant d'une entité ou association non fonctionnelle au niveau MCD)

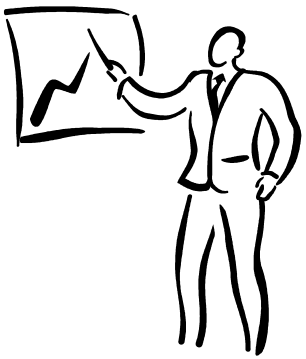
- ✓ les ***clés étrangères (foreign key)*** dont le rôle sera d'établir un lien logique entre deux tables.

- ✓ Une contrainte d'intégrité référentielle (constituée d'un ou plusieurs attributs) impose que toute valeur de la clé étrangère existe en tant que valeur de clé primaire de la table associée. Des restriction existent aussi sur les mises à jour.

Le langage SQL
DML- Manipulation des données

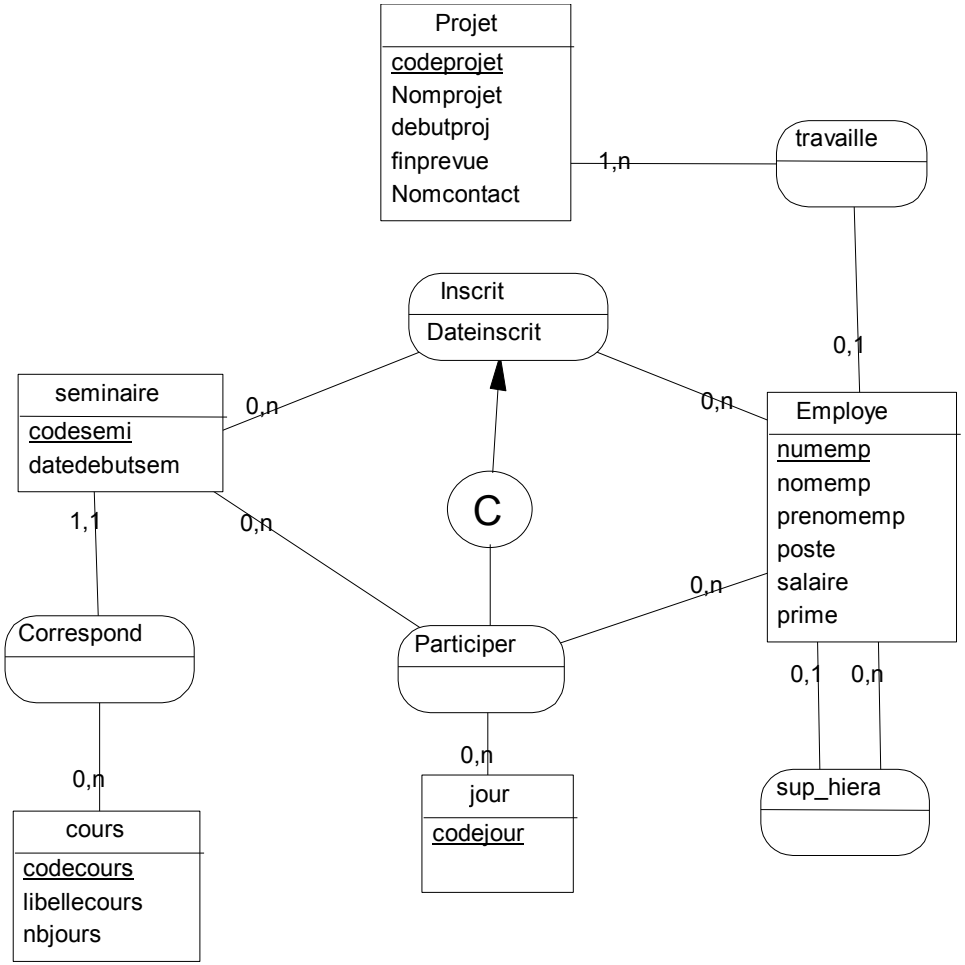


Partie 2 : La base exemple



Le langage SQL
DML- Manipulation des données

Le MCD



Le langage SQL
DML- Manipulation des données

Le MLD définitif

Le langage SQL
DML- Manipulation des données

Les fichiers nécessaires aux exemples

Les exemples peuvent être réalisés dans plusieurs environnements.

Pour chaque environnement, 2 fichiers :

Crebase : ordres DDL de création de la structure (tables, index,…)

Creocc : ordres DML de remplissage des tables

Environnement	***Fichier DDL***	***Fichier DML***
Oracle	crebase_oracle.sql	creocc_oracle09.sql
Mysql	crebase_Mysql09.sql	CREOCC_mysqlV09.sql
Postgresql	crebas_postgres.sql	creocc_postgres04.sql
Access	Basecours.mdb	

Didacticiels de mise en place : oracle et mysql : fichier

Le langage SQL
DML- Manipulation des données

Dans les syntaxes des commandes :

- ✓ ***Accolades {} :*** un des choix à l'intérieur est nécessaire
- ✓ ***Crochets [] :*** ce qui est à l'intérieur est facultatif
- ✓ ***Pipe*** **|** : séparation des choix à l'intérieur d'une accolade ou de crochets

Le langage SQL
DML- Manipulation des données

Partie 3 : Extraction des données de la base

La clause select

Le langage SQL
DML- Manipulation des données

Syntaxe générale :

SELECT [*DISTINCT* | *ALL*] {*nom-tab.nom-col1*[,*nom-tab.nom-col2*]}
[FROM] [*nom-createur.*] *nom-tab* [,[*nom-createur.*] *nom-tab2*]
[*WHERE prédicat(s)*]
[*GROUP BY nom-col3* [,*nom-col4*..] [*HAVING prédicat*]]
[*ORDER BY nom-col5* [*DESC*] [,*nom-col6 [desc]..*]] ;

- ✓ Seule la clause select est obligatoire,
- ✓ Suivant le SGBD, la clause From peut être obligatoire (oracle par ex)
- ✓ L'ordre des clauses doit être respecté
- ✓ La clause having n'existe qu'avec la clause Group by

Le langage SQL
DML- Manipulation des données

Nom d'une table :

NomBase.NomTable
ou
NomSchema.NomTable

On peut utiliser le nom d'une table sans préfixe, à condition l'on se soit positionné sur la base de données qui la contient (use unebase). ou connecté sur le schéma
Sinon, il faut faire précéder le nom de la table par le nom de la base ou du schéma si la table est dans un schéma différent de celui sur lequel on est connecté (oracle)

- **Exemple** : test.employe.

Le langage SQL
DML- Manipulation des données

Nom d'une colonne :

NomTable.NomColonne

Le nom de la table est facultatif quand il n'y a pas de risque d'ambiguïté sur le nom de la colonne à l'intérieur d'une requête

- **Exemple** : dans une même requête

cours.codecours et seminaire.codecours

Le langage SQL
DML- Manipulation des données

Sélection des colonnes d'une seule table :

SELECT * FROM *nom-table* **;**

Toutes les colonnes et toutes les lignes de la table seront sélectionnées

Exercice : afficher toutes les lignes et colonnes de la table cours

SELECT nom-col1, nom-col2,...
FROM nom-table ;

Seules les colonnes spécifiées seront affichées, par contre, toutes les lignes seront sélectionnées.

Exercice : afficher les codes et libellés des différents cours

Le langage SQL
DML- Manipulation des données

La clause Distinct / All

La clause **DISTINCT** permet de retourner un seule fois les lignes dupliquées. Cette clause génère un tri, il faut donc l'utiliser quand cela est vraiment nécessaire.

On n'utilise quasiment jamais la clause ALL (clause par défaut)

Exercice : afficher les codes des cours pour lesquels il y a au moins un séminaire

Le langage SQL
DML- Manipulation des données

SÉLECTION DE LIGNES : LA CLAUSE WHERE

SELECT * FROM nom-table
WHERE prédicat ;

prédicats simples :

Ils expriment condition simple qui renvoie une valeur booléenne.
Ainsi, si le prédicat est vérifié, le tuple fera partie de la table résultat.

Les opérateurs de comparaison sont :

égal (=)	**différent (<>)**
inférieur (<)	**supérieur (>)**
inférieur ou égal (<=)	**supérieur ou égal (>=)**

BETWEEN
IN
LIKE

Le langage SQL
DML- Manipulation des données

Dans un prédicat simple, on compare toujours :

- ✓ Une colonne à une constante
- ✓ Une colonne à une colonne
- ✓ Une colonne au résultat d'une requête. (***requêtes imbriquées)***.

Les chaînes de caractères se présenteront entre simples quotes ('JOLIBOIS').
On met deux simples quotes lorsqu'il faut insérer une apostrophe dans la chaîne de caractères ('L''OISEAU').

Le langage SQL
DML- Manipulation des données

Les dates se présenteront ***entre simples quotes***. Voir la documentation du SGBD pour les différents formats qui peuvent être configurés.

Les majuscules sont différentes des minuscules.
On prêtera une attention particulière à la façon dont on va stocker les données dans la base

Le langage SQL
DML- Manipulation des données

Exercices :

✓ Afficher les employés (numéro, nom et salaire) dont le salaire est égal à 10000.

✓ Afficher le nom et le salaire de l'employé numéro 2

Le langage SQL
DML- Manipulation des données

Between **:**

expr1 BETWEEN expr2 AND expr3

L'évaluation renvoie VRAI si expr1(qui est une colonne) est compris entre expr2 et expr3(qui sont des valeurs Les bornes sont comprises dans le résultat

Les bornes sont comprises dans le résultat

Exercices :

✓numéro, nom et salaire des employés dont le salaire est compris entre 10000 et 11000.

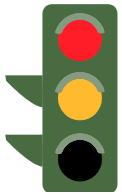
✓Numéros des employés qui se sont inscrits à un séminaire en octobre 2009

Le langage SQL
DML- Manipulation des données

In **:**

expr1 in (expr2, expr3,...)

L'évaluation renvoie vrai si expr1 est égal à l'une des expressions entre parenthèses. Expr1 correspond à une colonne, alors que expr2 et expr3 sont des constantes.



On verra plus tard que l'on peut remplacer la liste de valeurs par une requête renvoyant plusieurs valeurs

Exercice :

✓ numéro, nom et salaire des employés dont le nom est DUPONT ou RIVIERE

Le langage SQL
DML- Manipulation des données

***Like* :**

expr LIKE chaîne

Le prédicat évalue la ressemblance à une chaîne de caractères.
La ressemblance s'établit avec 2 jockers :

- ✓ : underscore (_) : remplace 1 caractère
- ✓ : pourcentage (%) : remplace de 0 à n caractères .

*Dans les bases microsoft, les jokers sont les mêmes que sous windows : * et ?*

Exercices :

- ✓ nom et salaire des employés dont le nom commence par G
- ✓ nom et salaire des employés dont le nom a un E en deuxième position

Le langage SQL
DML- Manipulation des données

Prédicats composés

On peut combiner les prédicats à l'aide des opérateurs ***AND, OR et NOT***.

✓ L'opérateur AND signifie que le prédicat est vrai si les deux conditions formant le AND sont vérifiées.

✓ L'opérateur OR signifie que le prédicat est vrai si une des deux conditions formant le OR sont vérifiées.

✓ L'opérateur NOT inverse le sens du prédicat

Exercices :

✓ nom des employés dont le poste est C17 et dont le salaire est supérieur à 11000

✓ nom des employés dont le poste est C17 ou B14

Le langage SQL
DML- Manipulation des données

La valeur Null

- ✓ La valeur Null signifie que la valeur du champ na pas été renseignée.
- ✓ ***Null est un état et non une valeur***
- ✓ C'est donc incomparable avec un 0, un espace ou une double quote.
- ✓ Toute opération arithmétique avec un null fournit un résultat null.
- ✓ L'opérateur ***isnull*** permet de tester si un champ est à l'état null (mysql)

Exercices :

- ✓ nom salaire et poste des employés dont la prime est nulle
- ✓ nom salaire et poste des employés dont la prime n'est pas nulle

Le langage SQL
DML- Manipulation des données

La jointure

✓ la JOINTURE permet d'obtenir dans une même ligne des informations provenant de plusieurs tables.

✓On peut faire des jointures sur toutes les colonnes du moment qu'elles sont compatibles

✓Souvent une jointure entre la clé primaire de la première table et la clé étrangère de la seconde table.

✓On peut faire des jointures entre un nombre de tables supérieur à 2

✓ Il existe plusieurs façons d'effectuer des jointures notamment SQL89 et SQL92

Les syntaxes peuvent légèrement varier d'un éditeur de SGBD et d'une version à l'autre d'un même SGBD… vérifier les syntaxes

Le langage SQL
DML- Manipulation des données

La jointure SQL 89

- ✓ Les tables à accéder apparissent dans la clause from
- ✓ La condition de jointure et dans la clause where
- ✓ Le nombre de tables apparaissant dans la clause from peut être supérieur à 2
- ✓ L'opérateur de comparaison n'est pas obligatoirement l'égalité

Si la condition de jointure n'est pas mentionnée, le SGBD effectue un produit cartésien des tables ….

Le langage SQL
DML- Manipulation des données

La jointure SQL 89

Exercices :

✓ code séminaire, libellé cours, nbjours de chaque séminaire

✓ codesemi, codecours, datedbutsem, nbjours des séminaires du cours BR035

✓ codecours, libellecours des cours où se sont inscrits les employés travaillant sur le projet dont le code est égal à PR1

Le langage SQL
DML- Manipulation des données

La jointure SQL 92 (à privilégier)

✓ Le verbe inner Join et la condition de jointure apparaissent dans la clausse from

From T1 inner join T2 on condition de jointure

Exercice :

✓ Ecrire les deux premières requêtes SQL89 avec la syntaxe SQL92

Le langage SQL
DML- Manipulation des données

La jointure SQL 92

✓ La condition de jointure, en plus des opérateurs traditionnels peut être ***between***, ***like*** ou ***in***

Exercice :
✓numéros et dates d'inscription des employés qui se sont inscrits à un séminaire et dont la date d'inscription est comprise entre la date de début et la date de fin prévue du projet dont le code est PR1

✓ On peut effectuer des jointures sur plus de deux tables, ***en imbriquant les clauses inner join***

Exercice :
✓ codecours, libellé cours des cours où se sont inscrits les employés travaillant sur le projet dont le code est égal à PR1

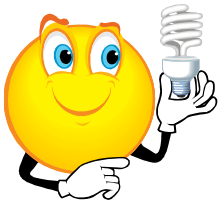
Le langage SQL
DML- Manipulation des données

La jointure Externe SQL92

Dans une jointure classique, tout tuple de la première table qui ne vérifie pas la condition de jointure avec un tuple de la deuxième table n'apparaît pas dans la table résultat.

Dans certain cas il est nécessaire de faire apparaître tous les tuples de la première table.

On peut alors pour obtenir ce résultat utiliser une jointure externe



Une jointure externe est une jointure qui favorise une table par rapport à une autre.

Ainsi, les lignes de la table dominante seront affichées même si la condition n'est pas réalisée.

Le langage SQL
DML- Manipulation des données

La jointure Externe SQL92

Pour réaliser une jointure externe, on utilise un des opérateurs suivants :

- **Left join** (on privilégie la table de gauche)
- **Right join** (on privilégie la table de droite)
- **Full join** (on privilégie .. Les deux tables)

Le langage SQL
DML- Manipulation des données

La jointure Externe SQL92

Exercices :

✓ Afficher les noms des employés dont le numéro est supérieur à 17 avec les codes des séminaires auxquels ils se sont inscrits. Si un employé ne s'est inscrit à aucun séminaire, son nom doit apparaitre.

✓ numéro, nom, code projet et nom du projet. Tous les employés doivent apparaitre

Le langage SQL
DML- Manipulation des données

Les qualificateurs courts ou alias

Les qualificateurs courts permettent de renommer une table dans une requête. On les définit dans la clause from après le nom de la table que l'on veut renommer

From nomtable [AS] alias,…

Exercice :

✓ codecours, libellé cours des cours où se sont inscrits les employés travaillant sur le projet dont le code est égal à PR1 … en utilisant les qualificateurs courts

On peut aussi utiliser ces qualificateurs pour renommer une colonne résultat dans la clause select

Le langage SQL
DML- Manipulation des données

Cas particulier : l'auto jointure

Il s'agit d'un cas particulier où l'on a besoin de faire une jointure sur des colonnes provenant de la même table.
Il sera alors nécessaire d'utiliser deux fois la même table en attribuant à chacune un alias

Exercice :
✓ Retourner le supérieur hiérarchique de chaque employe (numemp, nomemp, prenomemp, superieur, nomsuperieur, prenomsuperieur)

Tous les employés de la table employe doivent apparaitre

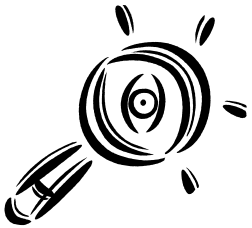
Le langage SQL
DML- Manipulation des données

Le classement : la clause order by

Elle permet de présenter la requête résultat triée sur une ou plusieurs colonnes.

ORDER BY nom-col5 [DESC] [,nom-col6 [desc]..]] ;

L'option DESC permet de demander un tri décroissant. Par défaut l'option ASC est exécutée.



il n'est pas nécessaire de faire apparaître le nom de la colonne sur laquelle on veut effectuer le tri dans la clause select.

Cette fonctionnalité est particulièrement utilisée quand on va inclure des colonnes virtuelles dans la clause select. Voir "les colonnes virtuelles"

Le langage SQL
DML- Manipulation des données

Le classement : la clause order by

Exercices :

✓Afficher les employés, numemp, nomemp, prenomemp, codeprojet, prime, triés sur la valeur de prime en ordre croissant.

✓Que remarque-t-on pour les primes nulles ?

✓Afficher les employés, numemp, nomemp, prenomemp, codeprojet, libelleprojet, salaire, triés sur le codeprojet en ordre croissant et la salaire en ordre décroissant.

✓ afficher le code projet, le nom projet, la date de début du projet, trié sur la date de fin de projet prévue.

Le langage SQL
DML- Manipulation des données

Les colonnes virtuelles :

Une colonne virtuelle est une colonne que l'on crée dans une requête et qui contient une expression, ou une fonction prédéfinie de colonne

On pourra trouver :

- ✓ ***Les colonnes contenant une expression***
- ✓ ***Les fonctions colonne***

Le langage SQL
DML- Manipulation des données

Les colonnes virtuelles : Les colonnes contenant une expression

On peut placer dans la clause select une expression portant sur
- ✓ deux colonnes de table,
- ✓ une colonne et une valeur numérique ou chaîne de caractère.

Exercice :
- ✓ trouver le gain (salaire + prime) de chaque employé (numemp, nomemp, total)

On remarquera les résultats des additions impliqaunt les valeurs ***null*** dont le résultat fournit la valeur ***null***

Le langage SQL
DML- Manipulation des données

Les colonnes virtuelles : Les colonnes contenant une expression

Il existe des fonctions permettant de substituer une valeur à un null

✓ La fonction qui est normalisée est ***coalesce***
COALESCE retourne la première expression non-NULL dans la liste d'arguments. Si toutes les expressions sont NULL, elle renvoie NULL
• ***Exemple : select coalesce(null, null, 10) … renverra 10 !***

✓ sur Oracle, ***la fonction NVL(expr1,expr2),*** qui permet d'affecter expr2 à expr1 si celle-ci est nulle,

✓ sur Mysql, c'est la fonction ***IFNULL(expr1,expr2)*** qui en est l'équivalente.

Exercice : Corriger la requête

Le langage SQL
DML- Manipulation des données

Les colonnes virtuelles : Les fonctions colonne

Ce sont des fonctions qui permettent d'obtenir des informations concernant l'ensemble de la colonne citée en paramètre.
La syntaxe est :

Fonction([distinct] nomcol)

- ✓ Le mot clé Distinct permet d'éliminer les valeurs répétées.
- ✓ Les valeurs Null ne sont pas prises en considération (sauf count)
- ✓ On trouvera donc ces fonctions :
  - soit dans la clause select
  - soit dans la clause having (voir les regroupements)

Le langage SQL
DML- Manipulation des données

Les colonnes virtuelles : Les fonctions colonne

Les principales fonctions :

- ✓ ***SUM(NOMCOL|EXPRESSION)*** : somme de toutes les valeurs de la colonne
- ✓ ***AVG(NOMCOL|EXPRESSION)*** : moyenne de toutes les valeurs d'une colonne
- ✓ ***MAX(NOMCOL|EXPRESSION)*** : trouve la valeur maximum de la colonne
- ✓ ***MIN(NOMCOL|EXPRESSION)*** : trouve la valeur minimum de la colonne
- ✓ ***COUNT(*)*** : compte le nombre d'occurrences
- ✓ ***COUNT(NOMCOL)*** : compte le nombre de valeurs not null de la colonne nomcol

On ne peut intégrer ces fonctions dans la clause where

On utilisera ces fonctions avec les regroupements

Le langage SQL
DML- Manipulation des données

Les colonnes virtuelles : Les fonctions colonne

Exercices :

✓ quel est le plus gros salaire ?

✓ quelle est la plus petite prime attribuée à un employé ?

✓ quelle est la moyenne des salaires et des primes ? Comment se comporte la valeur null?

✓ combien y a-t-il d'employés dans le poste A12 ?

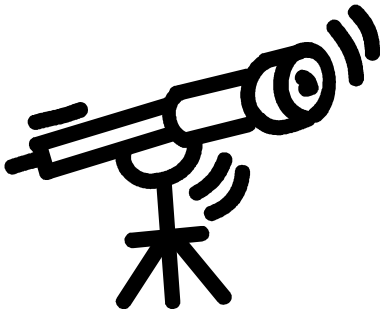
✓ combien y a-t-il d'employés touchant une prime ? (2 manières)

✓Dans combien de postes travaillent les employés?

Le langage SQL
DML- Manipulation des données

Les Regroupements : La clause group by

✓ Elle consiste à subdiviser la table en un ou plusieurs groupes.
✓ Un groupe étant un ensemble de lignes ayant une ou plusieurs valeurs égales.
✓ A l'intérieur d'un groupe, les valeurs de l'attribut spécifié seront égales.



la requête suivante va retourner autant de tuples qu'il y a de codeprojets différents dans la table employe :

```
select codeprojet
from employe
group by codeprojet;
```

On utilise beaucoup les fonctions colonnes avec la clause group by. Ainsi on peut calculer un certain nombre d'informations à l'intérieur des groupes formés

Le langage SQL
DML- Manipulation des données

Les Regroupements : La clause group by

Exercices :

✓ trouver la somme des salaires par poste (poste, somme)

✓ trouver le nombre de séminaires par cours (codecours, nombre)

✓ trouver le nombre de séminaires par cours (codecours, libellecours, nombre) … ***Attention, il y a un petit piège !!!***
✓ trouver le nombre d'inscrits par jour pour chacun des séminaires du cours 'BR013'
✓ trouver le nombre de séminaires où se sont inscrits chaque employés ? … ***Attention, il y a un petit piège !!!***

Le langage SQL
DML- Manipulation des données

Les Regroupements : La clause group by

L'ensemble des valeurs null forment un groupe

La clause Group By doit comporter OBLIGATOIREMENT les colonnes de tables figurant dans le select.

Exercices :

- ✓ Nombre d'employés travaillant par projet
- ✓ Réécrire la requête en écrivant "***aucun***" à la place de null

Le langage SQL
DML- Manipulation des données

Les Regroupements : La clause having

- ✓ ***elle s'utilise uniquement avec la clause group by.***
- ✓ **elle permet d'effectuer une restriction sur le résultat du group by.**

La clause HAVING n'empêche pas que l'on trouve une clause WHERE dans la requête. SQL va ici aussi effectuer le WHERE en premier puis le GROUP BY puis enfin le HAVING.

Le langage SQL
DML- Manipulation des données

Les Regroupements : La clause having

Exercices :

✓ salaires moyens par poste, pour les postes comportant plus de deux employés

✓ liste des séminaires (codesemi, codecours, libellecours), où il y a plus de 3 inscrits

✓ salaire moyen par poste, pour les employés dont la prime est à null, et pour les postes comportant plus de deux employés

Le langage SQL
DML- Manipulation des données

Les opérateurs ensemblistes

Ils permettent de combiner le résultat d'au moins 2 ordres select
Ils sont au nombre de 3 :

- ✓ ***UNION :*** réalise la réunion du résultat de deux select
- ✓ ***INTERSECT*** : réalise l'intersection du résultat de deux select
- ✓ ***EXCEPT (ou MINUS)*** : garde les occurrences du premier select en y enlevant les occurrences du deuxième select

Dans tous les cas, il est impératif que le nombre et le type des colonnes issues des select soient compatibles sinon identiques. Les longueurs des colonnes peuvent toutefois être différentes.

Le langage SQL
DML- Manipulation des données

Les opérateurs ensemblistes

Syntaxe :

SELECT [*DISTINCT*] {*nom-tab.nom-col1*[,*nom-tab.nom-col2*]}
FROM [*nom-createur.*] *nom-tab1* [,[*nom-createur.*] *nom-tab1*]
[*WHERE prédicat*]
[*GROUP BY nom-col3* [,*nom-col4*..] [*HAVING prédicat*]]

UNION [ALL] *ou* **INTERSECT** *ou* **EXCEPT**

SELECT [*DISTINCT*] {*nom-tab.nom-col1*[,*nom-tab.nom-col2*]}
FROM [*nom-createur.*] *nom-tab1* [,[*nom-createur.*] *nom-tab1*]
[*WHERE prédicat*]
[*GROUP BY nom-col3* [,*nom-col4*..] [*HAVING prédicat*]]
[***ORDER BY*** *numero-col1* [***DESC***] [,*numero-col2 [desc]..]*] ;

Dans la relation résultat on trouvera le nom des colonnes du premier select.

Le langage SQL
DML- Manipulation des données

Les opérateurs ensemblistes : l'opérateur Union

L'opérateur union retourne une table ayant les mêmes colonnes que les 2 tables issues des ordres select et ***dont les tuples appartiennent à l'une ou à l'autre des deux tables de départ.***

L'option ALL permet de ne pas enlever les doublons.

Exercices :

✓ Trouver les employés dont le salaire est inférieur à 10000 ou bien qui se sont inscrits au séminaire 'BR0571012'

✓ Même requête avec la clause union all (que remarque-t-on ?)

Le langage SQL
DML- Manipulation des données

Les opérateurs ensemblistes : l'opérateur Intersect

L'opérateur intersect retourne une table ayant les mêmes colonnes que les 2 tables issues des ordres select et ***dont les tuples appartiennent à l'une et à l'autre des deux tables de départ***.

Non encore implémenté sous Mysql

Exercice :

✓ Trouver les employés dont le salaire est inférieur à 10000 et qui se sont inscrits au séminaire 'BR0571012'

(on peut aussi retrouver ce résultat avec une requête imbriquée)

Le langage SQL
DML- Manipulation des données

Les opérateurs ensemblistes : l'opérateur Except

L'opérateur except retourne une table ayant les mêmes colonnes que les 2 tables issues des ordres select et ***dont les tuples appartiennent à la première table mais pas à la deuxième***.

Non encore implémenté sous Mysql

Remplacé par l'opérateur Minus sous Oracle

Exercice :

✓ Trouver les employés dont le salaire est inférieur à 10000 et qui ne sont pas inscrits au séminaire 'BR0571012', résultat trié sur le nom de l'employé

Le langage SQL
DML- Manipulation des données

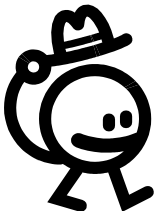
Les requêtes imbriquées : Les select imbriqués

Les requêtes imbriquées ne sont implémentées que depuis la version 4.1 de Mysql

✓ On peut imbriquer un ordre select dans les clauses :

- ***Where***
- ***Having***

✓ Le nombre de niveaux d'imbrication est suffisamment important pour qu'on ne l'atteigne jamais… voir la doc du SGBD…



La seule syntaxe à respecter est de placer la requête imbriquée entre parenthèses.

Le langage SQL
DML- Manipulation des données

Les requêtes imbriquées : Les select imbriqués

Le select imbriqué ne retourne qu'une seule valeur

Lorsque l'on attend du select imbriqué qu'il ne retourne qu'une seule valeur, on utilise les opérateurs de comparaison que l'on a déjà vus, à savoir : =, <, <= ,> ,>=, <>

Exercices :

✓ liste des employés (numéro, nom) travaillant dans le même poste que l'employé numéro 2

✓ liste des employés (numéro, nom) gagnant plus que l'employé numéro 2

Le langage SQL
DML- Manipulation des données

Les requêtes imbriquées : Les select imbriqués

Le select imbriqué retourne plus d'une seule valeur

- ✓ ***On utilise en général l'opérateur IN :***
  la comparaison prendra la valeur vrai si la colonne spécifiée avant le IN comprend au moins une valeur retournée dans le select suivant le IN.,
- ✓ ***De la même façon, on peut utiliser l'opérateur not in***
- ✓ ***on peut aussi utiliser les opérateurs any et all***

Exercices :

- ✓ liste des employés (numéro, nom) inscrits à au moins un séminaire auquel s'est inscrit JOLIBOIS
- ✓ liste des employés du poste B12 dont le salaire est égal à l'un des salaires des employés du poste B15 (n°, nom, prénom)
- ✓Liste des employés inscrits à un des séminaires d'un cours de UML(n°, nom, prénom)

Le langage SQL
DML- Manipulation des données

Les requêtes imbriquées : Le prédicat Exists

- ✓ ***Le prédicat EXISTS permet de tester l'existence ou l'absence de données dans la sous-requête.***
- ✓ ***Retourne vrai si la sous-requête renvoie au moins une ligne (même constituée uniquement de nulls)***
- ✓ ***Renvoie faux dans le cas contraire.***
- ✓ ***Le prédicat peut être combiné avec l'opérateur NOT***
- ✓ ***On le préférera en général au prédicat IN***
- ✓ ***On l'utilise en général avec une sous-requête corrélée***
- ✓ ***Il convient d'indiquer * dans le select ou 1 plutôt que des noms de colonne.***

Le langage SQL
DML- Manipulation des données

Les requêtes imbriquées : Le prédicat Exists

Syntaxe :

Select ….
From …
Where EXISTS (Select….)

Exercices :

- ***Nombres d'inscriptions par séminaires à condition qu'il y ait une inscription après le 1er octobre 2009***
- ***La même mais après le 1er novembre 2009***
- ***Nombre d'inscrits par séminaire pour les séminaire où aucun employé travaillant sur le projet PR1 n'est inscrit.***

Le langage SQL
DML- Manipulation des données

Les requêtes imbriquées : Les All et Any

✓**ALL** : la comparaison prendra la valeur vrai si la comparaison de la valeur de la colonne spécifiée avec toutes les valeurs retournées par le select imbriqué est vérifiée.

✓ **ANY** : la comparaison prendra la valeur vrai si la comparaison de la valeur de la colonne spécifiée est vérifiée avec au moins une valeur retournée par le select imbriqué

Le langage SQL
DML- Manipulation des données

Les requêtes imbriquées : Le prédicat All et Any

Syntaxe :

Select ….
From …
Where colonne ALL|ANY operateur (Select….)

Exercices :

- ✓***liste des employés gagnant plus que tous les employés travaillant dans le poste B14***

- ✓***Employés du poste A25 qui gagnent plus que l'un des employé du poste B14***

Le langage SQL
DML- Manipulation des données

Jointures et requêtes imbriquées

Les deux requêtes suivantes fournissent le même résultat :

```
select numemp, nomemp
from employe
where codeprojet = (select codeprojet
                    from projet
                    where nomprojet = 'Paye
ADCV');
```

```
select numemp, nomemp
from employe inner join  projet on
employe.codeprojet=projet.codeprojet
where nomprojet = 'Paye ADCV';
```

✓ **On privilégiera la rapidité d'exécution …**

✓ **En général, une jointure traditionnelle est plus rapide qu'une jointure par requête imbriquée**

Exercice :

✓ liste des employés travaillant sur le projet Paye ADCV et dont le salaire est supérieur à celui de GALLI

Le langage SQL
DML- Manipulation des données

Les vues : les vues scalaires ou inline

✓Il est possible, à partir de Oracle 8i de créer des vues temporaires dont la durée de vie est la durée de l'exécution de la requête.

✓Ces vues peuvent se trouver dans toutes les clauses de l'ordre select, sauf la clause group by.

✓Il suffit de remplacer le nom de la table dans la clause from par la requête.

Exercice :

✓ Afficher les noms et prénoms des supérieurs hiérarchiques de chaque employé : numéro, nom, prenom, numsup, nomsup, prenomsup.

Le langage SQL
DML- Manipulation des données

Partie 4 : Mise à jour des données de la base

La clause update

La clause insert

La clause delete

Le langage SQL
DML- Manipulation des données

Optimisation des requêtes :

http://tuningsql.metouyou.fr/?Itemid=1

Le langage SQL
DML- Manipulation des données

Mise à jour des données

✓ ***Le langage de manipulation permet de traiter en une commande un ensemble de une à n tuples d'une table.***

✓ ***L'unité de manipulation est donc le tuple.***

✓ ***Ces commandes sont au nombre de trois :***

- **UPDATE : pour la mise à jour de tuples,**
- **INSERT : pour l' insertion de tuples,**
- **DELETE : pour la suppression de tuples.**

Le langage SQL
DML- Manipulation des données

Mise à jour des données : Update

Il permet de modifier les valeurs de un ou de plusieurs champs, dans une ou plusieurs lignes existantes d'une table.

La syntaxe est la suivante :

```
UPDATE nom-table
SET nom-champ1 = nouvelle-valeur1
      [,nom-champ2 = nouvelle-valeur2,...]
[WHERE condition] ;
```

nouvelle-valeur1 est une expression qui peut être :

- une constante,
- un nom de colonne
- le résultat d'une requête imbriquée.

Les mises à jour sont effectuées sur les lignes satisfaisant la condition ou sur toutes les lignes si la clause where n'est pas spécifiée

Le langage SQL
DML- Manipulation des données

Mise à jour des données : Update

Exercices :

✓ Mettre à jour ANTHONY (numéro 11) dont la prime passe de 0 à 1000 (contrôler sa prime avant et après la requête de mise à jour)

✓ augmenter le salaire de tout le monde de 150 (contrôler les valeurs avant et après la requête de mise à jour)

✓ augmenter les salaires des employés travaillant sur le projet PR1 de 5150 (contrôler les valeurs avant et après la requête de mise à jour)

✓ augmenter le salaire de 15 % pour les employés qui ont le salaire le moins élevé des employés travaillant sur le même projet. (contrôler les valeurs avant et après la requête de mise à jour)

✓ Attribuer à l'employé 16 le même salaire que l'employé 17 (contrôler les valeurs avant et après la requête de mise à jour)

Le langage SQL
DML- Manipulation des données

Mise à jour des données : Delete

Il permet de supprimer des lignes d'une table

DELETE FROM nom-table
[WHERE prédicat] ;

Quand la condition WHERE n'est pas spécifiée, tout le contenu de la table est effacé.

L'opérateur WHERE accepte :

- ✓ les opérateurs logiques AND et OR,
- ✓ les opérateurs de comparaison (=, <>, <=, >=, <, >),
- ✓ le BETWEEN,
- ✓ le SELECT.

Le langage SQL
DML- Manipulation des données

Mise à jour des données : Delete

Exercices :

✓ supprimer l'inscription de l'employé numéro 1 au séminaire BR0350216

✓ supprimer toutes les inscriptions des employés travaillant sur le projet PR1 pour les cours BR035

Le langage SQL
DML- Manipulation des données

Mise à jour des données : Truncate alternative au Delete

✓ L'instruction ***truncate*** permet de supprimer TOUS les enregistrements d'une table, en redonnant à celle-ci tous les paramètres initiaux de stockage.

✓ Cette instruction est particulièrement intéressante pour les tables volumineuses.

Syntaxe :

TRUNCATE TABLE nomtable

Exercice :

✓ Exécuter le script test_truncate.sql sous oracle- analyser les résultats

Le langage SQL
DML- Manipulation des données

Mise à jour des données : Insert

Il permet d'insérer une ou plusieurs lignes dans une table, en spécifiant les valeurs à insérer.

On trouvera trois façons différentes d'insérer des tuples dans une table :

- ✓ ***Insertion "manuelle" d'une ligne ou plusieurs***
- ✓ ***Insertion de plusieurs lignes provenant d'une ou plusieurs tables***
- ✓ ***Création et remplissage d'une table dans une seule commande SQL***

Le langage SQL
DML- Manipulation des données

Mise à jour des données : Insert

Insertion d'une seule ligne

La syntaxe est la suivante :

```
INSERT INTO nom-table [(nom-col1,nom-col2,...)]
  VALUES (val1,val2,...) [,(val1,val2,...) , (val1,val2,...) ] ;
```

- ✓ ***La liste des colonnes est optionnelle mais il est fortement conseillé de la mettre, notamment dans les programmes***
- ✓ ***Si elle n'est pas spécifiée, la liste des colonnes sera par défaut la liste ordonnée de la table définie lors de la création.***

Le langage SQL
DML- Manipulation des données

Mise à jour des données : Insert

Insertion d'une seule ligne

Cas des valeurs que l'on ne peut renseigner lors de l'insertion d'un tuple :

✓ Il est alors nécessaire d'indiquer la valeur ***NULL*** ou la valeur ***DEFAULT*** si toutes les colonnes de la table sont spécifiées dans l'ordre insert

✓ On n'indique pas les colonnes non renseignées dans la liste des colonnes de l'ordre insert

✓ Dans les deux cas, Il s'entend que les colonnes acceptent ces valeurs (null ou default)

Le langage SQL
DML- Manipulation des données

Mise à jour des données : Insert

Insertion d'une seule ligne

Exercices :

✓ insérer un nouveau séminaire, code :BR0351123 codecours :BR035 date début :23/11/2010

✓ Insérer un nouvel employé numéro 22, nom :BILLONEAU, prénom : Jean-Jacques, poste : A22 ,salaire : 9500, supérieur:4)

✓ Insérer son frère jumeau dont le prénom est Jean-François, en utilisant une syntaxe différente

✓ Insérer les cours suivants en une seule instruction :

Code	libellé	nbjours
BR043	Initiation à la programmation objet	3
BR044	Programmation objet : les langages	2

Le langage SQL
DML- Manipulation des données

Mise à jour des données : Insert

Insertion de plusieurs lignes provenant de plusieurs tables

La syntaxe est la suivante :

```
INSERT INTO nom-table [(nom-col1,nom-col2,...)]
 SELECT … ;
```

Exercice :

✓ Créer la table EMP_RIEN avec les champs numero, nom, prenom, gain (mêmes caractéristiques que les champs numemp, nomemp, prenomemp, salaire de la table employe)

✓ Remplir cette table avec une seule instruction insert : Seuls les employés n'ayant aucun projet, ou travaillant sur le projet PR1 figureront dans la nouvelle table. La colonne gain sera la somme du salaire et de la prime.

Le langage SQL
DML- Manipulation des données

Mise à jour des données : Insert

Création et remplissage d'une table en une seule instruction

La syntaxe est la suivante :

```
CREATE TABLE nom-table [(nom-col1,nom-col2,...)]
 AS SELECT … ;
```

Exercice :

✓ même exercice que précédemment, mais avec la table EMP_PR2, qui contiendra les employés travaillant sur le projet PR2 ou sur aucun projet.

Le langage SQL
DML- Manipulation des données

Mise à jour des données : Insert

Cas des colonnes en numérotation séquentielle

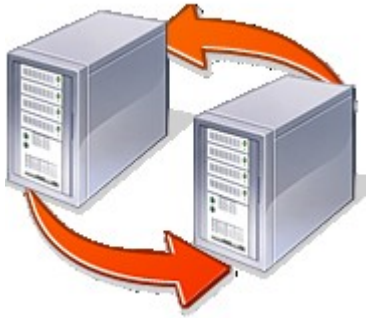
✓Sous Mysql (et les Sgbd acceptant le type autoincrement ou numauto) : on ne renseigne pas la colonne dans la liste des valeurs de l'ordre insert

✓Sous Oracle (et les SGBD acceptant l'objet sequence) : voir documentation pour l'utilisation des séquences

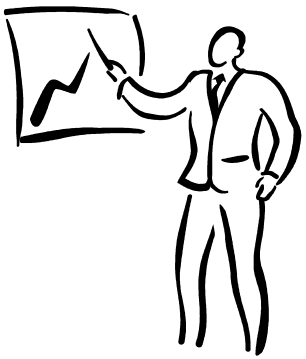
Exercice :

✓Créer graphiquement sous phpmyadmin la table STAGIAIRE, dont les colonnes sont numéro, nom, prénom et diplôme (acceptant la valeur NULL). La colonne numero est en auto incrément et sert de clé primaire.
✓ Insérer à l'aide de la commande insert plusieurs occurrences. (avec ou sans diplôme)

Le langage SQL
DML- Manipulation des données



Partie 5 :
Bonus au DML



Le langage SQL
DML- Manipulation des données

Partie 5a : connect by … start with

Le langage SQL
DML- Manipulation des données

Connect by … start with :

✓ Cette partie de cours a été traitée sous Oracle, cette fonctionnalité n'est pas généralisée.

✓ Cette clause permet la représentation et la manipulation de données ayant une structure arborescente. Elle correspond à la relation père/fils, à savoir qu'une ligne fasse référence à une autre ligne de la même table.

exemple : la colonne SUPERIEUR (supérieur hiérarchique) de la table EMPLOYE fait référence à un autre employé de la même table.

Le langage SQL
DML- Manipulation des données

La clause connect by :

- ✓**Elle permet de spécifier le lien de parenté entre les différentes lignes.**
- ✓**Elle doit obligatoirement contenir la clause PRIOR.**
- ✓**La clause PRIOR doit porter sur une des deux colonnes afin de préciser le père dans chacune des relations père/fils.**
- ✓**Ainsi, le SGBD détermine la ligne père et cherche ensuite dans la table toutes les lignes fils de la ligne père.**
- ✓**Il boucle sur les fils qui deviennent des pères et cherche leurs fils et ainsi de suite.**

Le langage SQL
DML- Manipulation des données

La clause start with :

✓ Elle permet d'indiquer la ligne à utiliser comme racine de l'arbre à rechercher.

✓ Si cette clause est omise, on affichera tous les arbres de la table (sous arbres), où chaque ligne sera une racine.

✓ Cette clause peut également servir pour commencer le parcours d'un arbre à partir d'une certaine position.

✓Il vaut donc mieux spécifier la condition dans cette clause.

Le langage SQL
DML- Manipulation des données

La clause start with :
Exemple :

```
select  level niveau , lpad(' ',2*level) || nomemp nom
        from employe
        connect by prior numemp=superieur
        start with nomemp='RUSSOT';
```

- ✓ **la pseudo-colonne LEVEL, qui permet de savoir à quel niveau hiérarchique on se trouve**
- ✓ **La fonction LPAD permet de rajouter des espaces au début de la chaine de caractères**
- ✓ **On a renommé toutes les colonnes**

Le langage SQL
DML- Manipulation des données

La clause connect by …start with :

Résultat :

```
NIVEAU     NOM
---------- ------------------------
         1   RUSSOT
         2     LUNEAU
         3       JACONO
         4         ANTHONY
         2     GIMOND
         3       DUCHATEL
         4         DUPONT
         3       MARTIN
         4         JOLIBOIS
         4         GALLI
         2     SAULT
         3       FARNY
         4         BERNARDI
         3       MAZAUD
         4         BEAUMONT
         3       CANE
         4         GOMEZ
         3       ESTIVAL
         2     RIVIERE
         3       BEUGNIES
         2     BRESSON
         2     CHAMPION
22 lignes sélectionnées
```

Le langage SQL
DML- Manipulation des données

La clause connect by …start with :

On peut aussi commencer l'arborescence à partir de Sault :

```
select  level niveau , lpad(' ',2*level) || nomemp nom
        from employe
        connect by prior numemp=superieur
        start with nomemp='SAULT'
```

```
NIVEAU      NOM
---------- --------------------
         1   SAULT
         2     FARNY
         3       BERNARDI
         2     MAZAUD
         3       BEAUMONT
         2     CANE
         3       GOMEZ
         2     ESTIVAL
8 lignes sélectionnées
```

Le langage SQL
DML- Manipulation des données

La clause connect by …start with :

Si on ne précise pas la racine, on retrouvera une arborescence pour chacun des employés

```
select  level niveau , lpad(' ',2*level) || ename nom
        from emp
        connect by prior empno=mgr;
```

Extrait du résultat :

NIVEAU	NOM
1	RUSSOT
2	LUNEAU
3	JACONO
4	ANTHONY
2	GIMOND
3	DUCHATEL
4	DUPONT
3	MARTIN
4	JOLIBOIS
4	GALLI
2	SAULT
3	FARNY
4	BERNARDI
3	MAZAUD
4	BEAUMONT

Le langage SQL
DML- Manipulation des données

La clause connect by …start with :

Si on change le prior et qu'on le fait porter sur SUPERIEUR, on obtient la hiérarchie inverse, à savoir pour chaque employé, quels sont ses supérieurs hiérarchiques.

```
select  level niveau , lpad(' ',2*level) || ename nom
        from emp
        connect by  numemp = prior superieur
        start with nomemp= 'SAULT';
```

résultat :

```
NIVEAU     NOM
---------- --------------------
         1   SAULT
         2     RUSSOT
2 lignes sélectionnées
```

On constate que Sault n'a qu »un seul supérieur hiérarchique

Le langage SQL
DML- Manipulation des données

La clause connect by …start with :

Si on veut connaître les supérieurs des autres :

```
select  level niveau , lpad(' ',2*level) || ename nom
        from emp
        connect by  empno=prior mgr
        start with SUPERIEUR is not null;
```

Extrait du résultat :

```
 NIVEAU      NOM
---------- ----------------
         1   LUNEAU
         2     RUSSOT
         1   GIMOND
         2     RUSSOT
         1   DUCHATEL
         2     GIMOND
         3       RUSSOT
         1   MARTIN
         2     GIMOND
         3       RUSSOT
         1   SAULT
         2     RUSSOT
         1   FARNY
         2     SAULT
         3       RUSSOT
         1   DUPONT
         2     DUCHATEL
         3       GIMOND
         4         RUSSOT
```

Le langage SQL
DML- Manipulation des données

Partie 5b : Limiter le nombre de lignes (Oracle)

Le langage SQL
DML- Manipulation des données

Limiter le nombre de lignes retournées :

Il est possible de limiter le nombre de lignes retournées par un requête grâce à l'utilisation de la pseudo colonne rownum

Syntaxe :

SELECT ...WHERE ROWNUM < n ...
| = 1

- ✓ **= 1 ne renverra qu'une seule ligne,**
- ✓ **<n renverra un nombre de lignes strictement inférieur à n**
- ✓ **L'expression ROWNUM est évaluée avant le order by.**
- ✓ **Il n'est donc pas possible de récupérer les les n premières lignes d'une table suite à un order by.**

Le langage SQL
DML- Manipulation des données

Limiter le nombre de lignes retournées :

Il est possible de limiter le nombre de lignes retournées par un requête grâce à l'utilisation de la pseudo colonne rownum

Syntaxe :

SELECT ...WHERE ROWNUM < n ...
| = 1

- ✓ **= 1 ne renverra qu'une seule ligne,**
- ✓ **<n renverra un nombre de lignes strictement inférieur à n**

Le langage SQL
DML- Manipulation des données

Limiter le nombre de lignes retournées :

Exercice :

✓ Afficher les 3 employés gagnant le plus …

Bien vérifier le résultat !!!!!…

Le langage SQL
DML- Manipulation des données

Partie 5c : L'instruction Case

Le langage SQL
DML- Manipulation des données

L'instruction case :
Elle permet de mettre en place une structure alternative sans passer par un bloc d'instructions PL/SQL.

- ✓ **On peut directement l'insérer dans l'ordre select.**
- ✓ **On pourra donc présenter de manière plus ergonomique le résultat de requêtes ou bien obtenir des résultats directement exploitables par des programmes.**
- ✓ **L'instruction est limitée à 128 choix. C'est déjà beaucoup !!!**
- ✓ **On peut toutefois aller au delà en imbriquant les instructions !!! bon courage !!!**

Le langage SQL
DML- Manipulation des données

L'instruction case :
Syntaxes possibles :

Syntaxe 1 : une expression d'égalité

```
CASE expression
WHEN expression_comparaison1 THEN expression_retournée1
WHEN expression_comparaison2 THEN expression_retournée2
…
[ELSE expression_retournée_autre]
END
```

La clause ELSE est facultative

Le langage SQL
DML- Manipulation des données

L'instruction case :

Syntaxe 1 : avec une expression d'égalité

```
select numemp, nomemp, prenomemp, case nvl(superieur,0)
                                  when 0  then 'Grand chef'
                                  when 15 then 'adjoint du grand chef'
                                  else 'Employé de base ou petit chef'
                                  end as "Situation hierarchique"
from employe;
```

Le langage SQL
DML- Manipulation des données

L'instruction case :
Syntaxe 1 : avec une expression d'égalité

NUMEMP	NOMEMP	PRENOMEMP	Situation hiérarchique
1	DUPONT	Pierre	Employé de base ou petit chef
2	JOLIBOIS	Rolland	Employé de base ou petit chef
3	BEAUMONT	Jean	Employé de base ou petit chef
4	DUCHATEL	Mireille	Employé de base ou petit chef
5	MARTIN	Robert	Employé de base ou petit chef
6	MAZAUD	Patricia	Employé de base ou petit chef
7	GIMOND	Antoine	adjoint du grand chef
8	SAULT	Jean	adjoint du grand chef
9	GALLI	Jean Daniel	Employé de base ou petit chef
10	JACONO	Marie	Employé de base ou petit chef
11	ANTHONY	Henri	Employé de base ou petit chef
12	CANE	Michel	Employé de base ou petit chef
13	GOMEZ	Joseph	Employé de base ou petit chef
14	RIVIERE	Maurice	adjoint du grand chef
15	RUSSOT	Eric	Grand chef
16	BERNARDI	Patrick	Employé de base ou petit chef
17	BEUGNIES	Maurice	Employé de base ou petit chef
18	FARNY	Daniel	Employé de base ou petit chef
19	ESTIVAL	Sophie	Employé de base ou petit chef
20	LUNEAU	Henri	adjoint du grand chef
21	BRESSON	Pierre	adjoint du grand chef

Le langage SQL
DML- Manipulation des données

L'instruction case :
Syntaxes possibles :

Syntaxe 2 : avec des conditions

```
CASE
WHEN condition1 THEN expression_retournée1
WHEN condition2 THEN expression_retournée2
…
[ELSE expression_retournée_autre]
END
```

La clause ELSE est facultative

Le langage SQL
DML- Manipulation des données

L'instruction case

Syntaxe 2 : avec des conditions

```
select codesemi, codecours , datedebutsem,
  case  when to_char(datedebutsem, 'MM')
                between 4 and 6 then 'Printemps'
        when to_char(datedebutsem, 'MM')
                between 7 and 9 then 'Eté'
        when to_char(datedebutsem, 'MM')
                between 10 and 12 then 'Automne'
        when to_char(datedebutsem, 'MM')
                between 1 and 3 then 'Hiver'
        else 'Saison non définie'
    end Saison
from seminaire;
```

Le langage SQL
DML- Manipulation des données

L'instruction case :

Syntaxe 2 : avec un opérateur de comparaison

CODESEMI	CODECO	DATEDEBU	SAISON
BR0350216	BR035	16/02/98	Hiver
BR0350316	BR035	16/03/98	Hiver
BR0350525	BR035	25/05/98	Printemps
BR0350907	BR035	07/09/98	Eté
BR0351019	BR035	19/10/98	Printemps
BR0340413	BR034	13/04/98	Printemps
BR0340518	BR034	18/05/98	Printemps
BR0340921	BR034	20/09/98	Eté
BR0341020	BR034	20/10/98	Automne
BR0341207	BR034	07/12/98	Automne
BR0130223	BR013	23/02/98	Hiver
BR0130914	BR013	14/09/98	Eté
BR0560210	BR056	10/02/98	Hiver
BR0560525	BR056	25/05/98	Printemps
BR0560914	BR056	14/09/98	Eté
BR0570323	BR057	23/03/98	Hiver
BR0570928	BR057	28/09/98	Eté
BR0571012	BR057	12/10/98	Automne
BR0701012	BR070	12/10/98	Automne

Partie 5d : Group by avec Rollup et Cube

Le langage SQL
DML- Manipulation des données

Group by … Rollup / Cube

Syntaxes possibles :

SELECT … FROM…
***GROUP BY* rollup (*nom-col1 ,nom-col2,…)* [*HAVING prédicat*]]**

SELECT … FROM…
GROUP BY* cube (*nom-col1 ,nom-col2,…)
[*HAVING prédicat*]]

Le langage SQL
DML- Manipulation des données

Group by … Rollup

Le Rollup permet de faire des regroupements de plus en plus généraux sur les colonnes mentionnées dans le group by.
Avec 2 colonnes le niveau le plus fin sera sur la deuxième colonne,
Le SGBD des regroupements sur la première colonne seulement puis sur l'ensemble.

Nombre d'inscrits par cours et par séminaire :

```
select libellecours, seminaire.codesemi, count(numemp) nombre
from inscrit inner join seminaire on seminaire.codesemi=
inscrit.codesemi
right join cours on cours.codecours=seminaire.codecours
group by rollup (libellecours, seminaire.codesemi);
```

Le langage SQL
DML- Manipulation des données

Group by … Rollup

Résultat :

LIBELLECOURS	CODESEMI	NOMBRE
SQL2 : La norme	BR0560210	4
SQL2 : La norme	BR0560525	4
SQL2 : La norme	BR0560914	4
SQL2 : La norme		12
UML : Initiation	BR0340413	3
UML : Initiation	BR0340518	5
UML : Initiation	BR0340921	5
UML : Initiation	BR0341020	3
UML : Initiation	BR0341207	4
UML : Initiation		20
Administration BDR	BR0130223	4
Administration BDR	BR0130914	5
Administration BDR		9
UML : Perfectionnement	BR0350216	3
UML : Perfectionnement	BR0350316	4
UML : Perfectionnement	BR0350525	5
UML : Perfectionnement	BR0350907	4
UML : Perfectionnement	BR0351019	3
UML : Perfectionnement		19
Conception BD relationnelle	BR0570323	3
Conception BD relationnelle	BR0570928	5
Conception BD relationnelle	BR0571012	3
Conception BD relationnelle		11
		71

Le langage SQL
DML- Manipulation des données

Group by … cube

L'option cube permet de réaliser toutes les combinaisons possibles de regroupements par rapport aux colonnes mentionnées dans la clause.

Statistiques d'inscrits par cours et par séminaire :

```
select libellecours, seminaire.codesemi, count(numemp) nombre
from inscrit inner join seminaire on seminaire.codesemi=
inscrit.codesemi
right join cours on cours.codecours=seminaire.codecours
group by cube (libellecours, seminaire.codesemi);
```

Le langage SQL
DML- Manipulation des données

Group by … cube

Extrait du résultat :

LIBELLECOURS	CODESEMI	NOMBRE
		71
	BR0130223	4
	BR0130914	5
	BR0340413	3
	BR0340518	5
	BR0340921	5
	BR0341020	3
	BR0341207	4
	BR0350216	3
	BR0350316	4
	BR0350525	5
	BR0350907	4
	BR0351019	3
	BR0560210	4
	BR0560525	4
	BR0560914	4
	BR0570323	3
	BR0570928	5
	BR0571012	3
SQL2 : La norme		12
SQL2 : La norme	BR0560210	4
SQL2 : La norme	BR0560525	4
SQL2 : La norme	BR0560914	4
UML : Initiation		20
UML : Initiation	BR0340413	3
UML : Initiation	BR0340518	5
UML : Initiation	BR0340921	5

Le langage SQL
DML- Manipulation des données

Partie 6 : les fonctions prédéfinies

Varient beaucoup d'un SGBD à l'autre… malgré la norme !

Le langage SQL
DML- Manipulation des données

Les fonctions prédéfinies

(D'après Frédéric Brouard http://sql.developpez.com/sqlaz/fonctions/)

Légende :
O : Oui
N : Non
X : Existe mais syntaxe hors norme
! : Même nom mais fonction différente

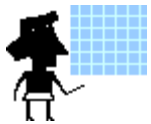
Voir la doc du SGBD…

Et actualiser le tableau avec les nouvelles versions des SGBD….

Le langage SQL
DML- Manipulation des données

I- AGREGATION STATISTIQUE

Fonction	Description	Norme SQL	Paradox	Access	MySQL	PostGreSQL	SQL Server	Oracle	Interbase
AVG	Moyenne	O	O	O	O	O	O	O	O
COUNT	Nombre	O	O	X	O	O	O	O	O
MAX	Maximum	O	O	O	O	O	O	O	O
MIN	Minimum	O	O	O	O	O	O	O	O
SUM	Total	O	O	O	O	O	O	O	O



Le langage SQL
DML- Manipulation des données

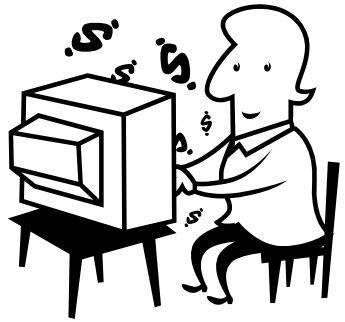
II- FONCTION "SYSTEME"

Fonction	Description	Norme SQL	Paradox	Access	MySQL	PostGreSQL	SQL Server	Oracle	Interbase
CURRENT_DATE	Date courante	O	N	N	O	O	N	N	O
CURRENT_TIME	Heure courante	O	N	N	O	O	N	N	O
CURRENT_TIMESTAMP	Date et heure courante	O	N	N	O	O	O	N	O
CURRENT_USER	Utilisateur courant	O	N	N	N	O	O	N	N
SESSION_USER	Utilisateur autorisé	O	N	N	X	O	O	N	N
SYSTEM_USER	Utilisateur système	O	N	N	X	O	O	N	N
CURDATE	Date du jour	N	N	N	O	N	N	N	N
CURTIME	Heure courante	N	N	N	O	N	N	N	N
DATABASE	Nom de la bases de données courante	N	N	N	O	N	O	O	N
GETDATE	Heure et date courante	N	N	N	N	N	O	N	N
NOW	Heure et date courante	N	O	O	O	O	O	O	N
SYSDATE	Date et/ou heure courante	N	N	N	O	N	N	O	N
TODAY	Date du jour	N	O	N	N	N	N	N	N
USER	Utilisateur courant	N	N	N	O	N	O	O	O
VERSION	Version du SGBDR	N	N	N	O	O	N	N	N

Le langage SQL
DML- Manipulation des données

III- FONCTIONS GENERALES

Fonction	Description	Norme SQL	Paradox	Access	MySQL	PostGreSQL	SQL Server	Oracle	Interbase
CAST	Transtypage	O	O	N	O	O	O	O	O
COALESCE	Valeur non NULL	O	N	N	O	O	O	N	N
NULLIF	Valeur NULL	O	N	N	O	O	O	N	N
OCTET_LENGTH	Longueur en octet	O	N	N	O	O	N	O	N
DATALENGTH	Longueur	N	N	N	N	N	O	N	N
DECODE	Fonction conditionnelle	N	N	N	N	N	N	O	N
GREATEST	Plus grande valeur	N	N	N	O	N	N	O	N
IFNULL	Valeur non NULL	N	N	N	O	O	O	N	N
LEAST	Plus petite valeur	N	N	N	N	O	N	O	N
LENGTH	Longueur	N	N	O	O	O	O	O	N
NVL	Valeur non NULL	N	N	N	N	N	N	O	N
TO_CHAR	Conversion de données en chaîne	N	N	N	N	N	N	O	N
TO_DATE	Conversion en date	N	N	N	N	O	N	O	N
TO_NUMBER	Conversion en nombre	N	N	N	N	N	N	O	N



Le langage SQL
DML- Manipulation des données

IV- FONCTIONS DE CHAINES DE CARACTERES

Fonction	Description	Norme SQL	Paradox	Access	MySQL	PostGreSQL	SQL Server	Oracle	Interbase
||	Concaténation	O	O	N	X	O	N	O	O
CHAR_LENGTH	Longueur d'une chaîne	O	N	N	X	O	N	N	N
CHARACTER_LENGTH	Longueur d'une chaîne	O	N	N	O	O	O	N	N
COLLATE	Substitution à une séquence de caractères	O	N	N	N	N	N	N	O
CONCATENATE	Concaténation	O	N	N	N	N	O	N	N
CONVERT	Conversion de format de caractères	O	N	N	N	N	!	O	O
LIKE (prédicat)	Comparaison partielle	O	O	X	O	O	O	O	O
LOWER	Mise en minuscule	O	O	N	O	O	O	O	N
POSITION	Position d'une chaîne dans une sous chaîne	O	N	N	O	O	N	N	N
SUBSTRING	Extraction d'une sous chaîne	O	O	N	O	O	N	N	N
TRANSLATE	Conversion de jeu de caractères	O	N	N	N	X	N	X	N
TO_CHAR	Conversion de données en chaîne	N	N	N	N	N	N	O	N

Le langage SQL
DML- Manipulation des données

IV- FONCTIONS DE CHAINES DE CARACTERES

Fonction	Description	Norme SQL	Paradox	Access	MySQL	PostGreSQL	SQL Server	Oracle	Interbase
TRIM	Suppression des caractères inutiles	O	O	N	O	O	N	O	N
UPPER	Mise en majuscule	O	O	N	O	O	O	O	O
CHAR	Conversion de code en caractère ASCII	N	N	O	O	N	O	N	N
CHAR_OCTET_LENGTH	Longueur d'une chaîne en octets	N	N	N	N	N	O	N	N
CHARACTER_MAXIMUM_LENGTH	Longueur maximum d'une chaîne	N	N	N	N	N	O	N	N
CHARACTER_OCTET_LENGTH	Longueur d'une chaîne en octets	N	N	N	N	N	O	N	N
CONCAT	Concaténation	N	N	O	O	N	O	O	N
ILIKE	LIKE insensible à la casse	N	N	N	N	O	N	N	N
INITCAP	Initiales en majuscule	N	N	N	N	O	N	O	N
INSTR	Position d'une chaîne dans une autre	N	N	O	O	N	N	O	N
LCASE	Mise en minuscule	N	N	O	O	N	O	O	N

Le langage SQL
DML- Manipulation des données

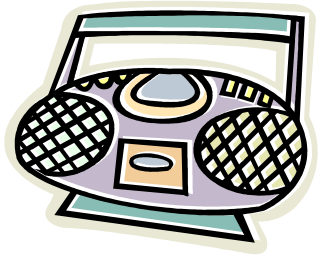
IV- FONCTIONS DE CHAINES DE CARACTERES

Fonction	Description	Norme SQL	Paradox	Access	MySQL	PostGreSQL	SQL Server	Oracle	Interbase
LOCATE	Position d'une chaîne dans une autre	N	O	O	O	N	O	O	N
LPAD	Remplissage à gauche	N	N	N	O	O	N	O	N
LTRIM	TRIM à gauche	N	O	O	O	O	O	O	N
NCHAR	Conversion de code en caractère UNICODE	N	N	N	N	N	O	N	N
PATINDEX	Position d'un motif dans une chaîne	N	N	N	N	N	O	N	N
REPLACE	Remplacement de caractères	N	N	N	O	N	O	O	N
REVERSE	Renversement	N	N	N	O	N	O	O	N
RPAD	Remplissage à droite	N	N	N	O	O	N	O	N
RTRIM	TRIM à droite	N	N	O	O	O	O	O	N
SPACE	Génération d'espaces	N	N	O	O	N	O	O	N
SUBSTR	Extraction d'une sous chaîne	N	N	N	N	N	N	O	N
UCASE	Mise en majuscule	N	N	O	O	N	O	O	N

Le langage SQL
DML- Manipulation des données

V- FONCTIONS DE CHAINES DE BITS

Fonction	Description	Norme SQL	Paradox	Access	MySQL	PostGreSQL	SQL Server	Oracle	Interbase
BIT_LENGTH	Longueur en bit	O	N	N	N	N	N	N	N
&	"et" pour bit logique	N	N	?	?	?	O	?	?
|	"ou" pour bit logique	N	N	?	?	?	O	?	?
^	"ou" exclusif pour bit logique	N	N	?	?	?	O	?	?



Le langage SQL
DML- Manipulation des données

V- FONCTIONS NUMERIQUES

Fonction	Description	Norme SQL	Paradox	Access	MySQL	PostGreSQL	SQL Server	Oracle	Interbase
%	Modulo	N	N	N	O	O	O	N	N
+ - * / ( )	Opérateurs et parenthésage	O	O	O	O	O	O	O	O
ABS	Valeur absolue	N	N	O	O	O	O	O	N
ASCII	Conversion de caractère en code ASCII	N	N	O	O	O	O	O	N
ASIN	Angle de sinus	N	N	N	O	O	O	O	N
ATAN	Angle de tangente	N	N	N	O	O	O	O	N
CEILING	Valeur approchée haute	N	N	O	O	N	O	N	N
COS	Cosinus	N	N	O	O	O	O	O	N
COT	Cotangente	N	N	O	O	O	O	N	N
EXP	Exponentielle	N	N	O	O	O	O	O	N
FLOOR	Valeur approchée basse	N	N	O	O	O	O	O	N
LN	Logarithme népérien	N	N	N	N	N	N	O	N
LOG	Logarithme népérien	N	N	O	O	N	O	O	N
LOG(n,m)	Logarithme en base n de m	N	N	N	N	O	N	O	N
LOG10	Logarithme décimal	N	N	N	O	N	O	O	N
MOD	Modulo	N	N	O	O	O	O	O	N
PI	Pi	N	N	N	O	O	O	O	N
POWER	Elévation à la puissance	N	N	O	O	N	O	O	N
RAND	Valeur aléatoire	N	N	O	O	N	O	N	N
ROUND	Arrondi	N	N	O	O	O	O	N	N
SIGN	Signe	N	N	O	O	O	O	O	N
SIN	Sinus	N	N	O	O	O	O	O	N
SQRT	Racine carrée	N	N	O	O	O	O	N	N
TAN	Tangente	N	N	O	O	O	O	O	N
TRUNC	Troncature	N	N	N	N	N	N	O	N
TRUNCATE	Troncature	N	N		O	O	O	O	N
UNICODE	Conversion de caractère en code UNICODE	N	N	N	N	N	O	?	N

Le langage SQL
DML- Manipulation des données

VI- FONCTIONS TEMPORELLES

Fonction	Description	Norme SQL	Paradox	Access	MySQL	PostGreSQL	SQL Server	Oracle	Interbase
EXTRACT	Partie de date	O	O	N	O	O	N	O	N
INTERVAL (opérations sur)	Durée	O	N	N	N	N	N	O	N
OVERLAPS (prédicat)	Recouvrement de période	O	N	N	N	O	N	N	N
ADDDATE	Ajout d'intervalle à une date	N	N	N	O	N	N	N	N
AGE	Age	N	N	N	N	O	N	N	N
DATE_ADD	Ajout d'intervalle à une date	N	N	N	O	N	N	N	N
DATE_FORMAT	Formatage de date	N	N	N	O	N	N	N	N
DATE_PART	Partie de date	N	N	N	N	O	N	N	N
DATE_SUB	Retrait d'intervalle à une date	N	N	N	O	N	N	N	N
DATEADD	Ajout de date	N	N	N	N	N	O	N	N
DATEDIFF	Retrait de date	N	N	N	N	N	O	N	N
DATENAME	Nom d'une partie de date	N	N	N	N	N	O	N	N
DATEPART	Partie de date	N	N	N	N	N	O	N	N
DAY	Jour d'une date	N	N	N	N	N	O	N	N
DAYNAME	Nom du jour	N	N	O	O	N	O	N	N
DAYOFMONTH	Jour du mois	N	N	N	O	N	N	N	N
DAYOFWEEK	Jour de la semaine	N	N	N	O	N	N	N	N
DAYOFYEAR	Jour dans l'année	N	N	N	O	N	N	N	N
HOUR	Extraction de l'heure	N	N	O	O	N	O	N	N
LAST_DAY	Dernier jour du mois	N	N	N	N	N	N	O	N
MINUTE		N	N	O	O	N	O	N	N
MONTH	Mois d'une date	N	N	O	O	N	O	O	N
MONTH_BETWEEN	MONTH_BETWEEN	N						N	N
MONTHNAME	Nom du mois	N	N	O	O	N	O	N	N
NEXT_DAY	Prochain premier jour de la semaine	N	N	N	N	N	N	O	N
SECOND	Extrait les secondes	N	N	O	O	N	O	N	N
SUBDATE	Retrait d'intervalle à une date	N	N	N	O	N	N	N	N
WEEK	Numéro de la semaine	N	N	O	O	N	O	O	N
YEAR	Année d'une date	N	N	O	O	N	O	O	N

Le langage SQL
DML- Manipulation des données

VII- PREDICAT, OPERATEURS ET STRUCTURES DIVERSES

Fonction	Description	Norme SQL	Paradox	Access	MySQL	PostGreSQL	SQL Server	Oracle	Interbase
CASE	Structure conditionnelle	O	N	N	O	O	O	X	O
IS [NOT] TRUE	Vrai	O	N	N	N	N	N	N	N
IS [NOT] FALSE	Faux	O	N	N	N	N	N	N	N
IS [NOT] UNKNOWN	Inconnu	O	N	N	N	N	N	N	N
IS [NOT] NULL	NULL	O	O	X	O	O	O	O	O
INNER JOIN	Jointure interne	O	O	O	O	O	O	N	O
LEFT, RIGHT, FULL OUTER JOIN	Jointure externe	O	O	O	O	O	O	N	O
NATURAL JOIN	Jointure naturelle	O	N	N	O	O	N	N	N
UNION JOIN	Jointure d'union	O	N	N	N	N	N	N	N
LEFT, RIGHT, FULL OUTER NATURAL JOIN	Jointure naturelle externe	O	N	N	X	O	N	N	N
INTERSECT	Intersection (ensemble)	O	?	N	N	O	N	X	N
UNION	Union (ensemble)	O	?	O	N	O	O	O	O
EXCEPT	Différence (ensemble)	O	?	N	N	O	N	N	N
[NOT] IN	Liste	O	O	O	X	O	O	O	O
[NOT] BETWEEN	Fourchette		O	O	O	O	O	O	O
[NOT] EXISTS	Existence	O	?	?	N	O	O	O	O
ALL	Comparaison à toutes les valeurs d'un ensemble	O	?	O	N	O	O	O	O
ANY / SOME	Comparaison à au moins une valeur de l'ensemble	O	?	O	N	O	O	O	O
UNIQUE	Existance sans doublons	O	N	N	N	N	N	N	N
MATCH UNIQUE	Correspondance	O	N	N	N	N	N	N	N
row value construteur	Construteur de ligne valuées	O	N	N	N	N	N	O	N
MINUS	Différence (ensemble)	N	N	N	N	O	N	O	N
LIMITE	nombre de ligne retournée	N	N	TOP	LIMIT	LIMIT	TOP	N	ROWS
identifiant de ligne		N	N	N	_rowid	oid	N	rowid	?

Le langage SQL
DML- Manipulation des données

IX- SOUS REQUETES

Fonction	Norme SQL	Paradox	Access	MySQL	PostGreSQL	SQL Server	Oracle	Interbase
Imbriquées	O	O	O	O	O	O	O	O
Corrélées	O	O	O	O	O	O	O	O
Dans la clause SELECT	O	X	O	O	O	O	O	O
Dans la clause FROM	O	N	N	O	O	O	O	N
Dans la clause WHERE	O	O	O	O	O	O	O	O
Dans la clause HAVING	O	O	N	O	O	O	O	O

Le langage SQL
DML- Manipulation des données

À suivre ... cours PL/SQL, administration Oracle...