SMI-S5

SGBD ORACLE : LDD

Pr. Issam QAFFOU

Laboratoire Ingénierie des Systèmes d'Information
Département d'Informatique
FSSM-UCA

Commandes SQL

- Pour la manipulation des données on utilise les commandes SQL:
  - Insertion des tuples (lignes): insert
  - Modification des valeurs des tuples: update
  - Suppression des tuples: delete

Insertion des n-uplets: INSERT

La commande INSERT est utilisée pour ajouter des enregistrements ou des parties d'enregistrements dans des tables. Elle est utilisée généralement sous deux formes.

1ère forme

INSERT INTO nom_table (champ1, champ2, ...)
VALUES ('valeur1','valeur2',...);

2ème forme

INSERT INTO nom_table (champ1, champ2, ...)
(Requête)

Remarque: Si des valeurs doivent être insérées dans tous les champs de l'enregistrement de la table, la liste des noms des champs n'a pas besoin d'être explicitement indiquée dans la commande. Les valeurs des champs à insérer doivent cependant apparaître dans le même ordre que les noms des champs lors de la création de la table:

INSERT INTO nom_table VALUES ('valeur1', 'valeur2', 'valeur3',...);

Exemple 1

Nous souhaitons insérer des valeurs dans la table client déjà créée.

Solution 1

INSERT INTO client values (1,'Alami','15 rue agadir','12/11/1988');

Solution 2

INSERT INTO client (Id_client,nom,adresse,naissance) values (1,'Alami','15 rue agadir','12/11/1988');

Exercice

1/ La table Agence est définie par un numéro d'agence (nombre et clé primaire), son nom (chaîne de 10 caractère), la ville (chaîne de 8 caractères) et l'adresse (chaine de caractères). « Le nom de l'agence doit être saisie »

2/ La table Client définie par son numéro de CIN (clé primaire), son nom et prénom (chaînes de 20 caractères), son adresse (chaine de caractères), la ville (chaîne de 9 caractères) et sa date de naissance. la ville doit être saisie.

3/ Insérer dans la table agence les enregistrements suivants :

* 1,2 Mars, Casablanca, 26 BD 2 Mars
* 2,Agdal

4/ Insérer dans la table Client les enregistrements suivants :

* B2541,romani, hamid, 26 BD zarktouni, casablanca, 15/04/1990
* A231,23/06/1989,Agadir

Modification des n-uplets: Update

- La commande UPDATE est utilisée pour changer des valeurs dans des champs d'une table.

  Sa syntaxe est:

  UPDATE table SET champ1 = nouvelle_valeur1, champ2 = nouvelle_valeur2,

  champ3 = nouvelle_valeur3 WHERE condition;

- La clause SET indique quels champs de la table vont être mis à jour et avec quelles valeurs ils vont l'être. Les champs non spécifiés après la clause SET ne seront pas modifiés.

La clause WHERE précise les enregistrements qui seront modifiés.

Remarque : Si la clause WHERE est absente, tous les enregistrements de la table seront modifiés.

Exemple :

UPDATE produit SET prix_unitaire = 1000;

Cette requête va modifier le prix unitaire de TOUS les produits de la table produit, en effet aucune condition n'a été précisée.

UPDATE produit SET prix_unitaire = 1000 where code_produit=19;

Cette requête ne modifiera que le prix unitaire du produit ayant le code égal 19 et si le code produit représente la clé primaire dans la table produit, nous aurons un seul enregistrement qui sera modifié.

Exercice :

Soit la table Etudiant suivante:

N°	Nom	Prénom	Age	Sexe	Ville
1	KACHLOUL	Hassan	17	M	Rabat
2	MRABET	Salwa	19	F	Essaouira
3	ZAHOURI	Ali	18	M	Casablanca
4	SEBTI	Rachid	18	M	Marrakech

1/ Modifier l’âge de tous les étudiants et mettre la valeur 20.

2/ Modifier l’âge des étudiants qui habitent rabat et mettre la valeur 18.

3/ Modifier le prénom et le sexe de l’étudiant dont le nom est SEBTI (prénom aura la valeur Hind)

4/ Que se passera t-il si nous avions dans la table étudiant deux personnes qui ont le nom SEBTI ?

5/ Proposer une requête qui répondra à la question « 3 » en traitant le cas de la question « 4 »

6/ Modifier la ville des personnes ayant l’âge entre 20 et 30 en mettant la valeur « Taza »

Suppression des n-uplets: DELETE

- Pour supprimer *des enregistrements* d'une table, on utilise la commande DELETE, sa syntaxe est la suivante :

  DELETE FROM nom_table WHERE condition;

  Remarque 1 : On ne peut pas supprimer une donnée d'un champ précis mais on supprime toute la ligne

  Remarque 2 : Si la clause WHERE est absente, tous les enregistrements de la table seront supprimés.

Remarque 3: La commande DELETE ne supprime pas la table, elle ne supprime que des enregistrements de la table.

Exemple :

Delete from produit; ou Delete * from produit;

Cette requête supprimera tous les produits de la table produit.

Et pour ne supprimer que le produit ayant le code_produit=19 la requête sera :

Delete from produit where code_produit=19;

Exercice : Soit la table Etudiant Suivante

N°	Nom	Prénom	Age	Sexe	Ville
1	KACHLOUL	Hassan	17	M	Rabat
2	MRABET	Salwa	19	F	Essaouira
3	ZAHOURI	Ali	18	M	Casablanca
4	SEBTI	Rachid	18	M	Marrakech

1. Supprimer les étudiants qui habitent Casablanca
2. Supprimer les étudiants qui habitent Casablanca et qui ont l'âge supérieur ou égal = 20.
3. Supprimer le 3ème enregistrement.
4. Supprimer le prénom de la personne qui se nomme SEBTI.
5. Supprimer tous les enregistrements de la table étudiant

La commande SELECT

- La commande select est la commande la plus utilisée de SQL, elle permet de récupérer des données dans des tables.

Syntaxe réduite:

SELECT champ1,champ2 … from nom_table;

- Les champs sont retournés dans l'ordre spécifié dans la clause SELECT et non pas dans l'ordre qu'il ont été créés dans la table.

Syntaxe générale

SELECT [{ DISTINCT | UNIQUE }] liste_Colonnes
FROM nomTable1 [,nomTable2]…
[WHERE condition]
[GROUP BY col1, col2..]
[HAVING condition]
[{ UNION | UNION ALL | INTERSECT | MINUS } (sous_Requête)]
[ORDER BY col1] ;

Exemple 1

Pour récupérer le nom et le prénom de tous les enregistrements de la table client :

```
SELECT nom,prénom from client
```

Pour avoir tous les champs de la table client :

```
SELECT * from client
```

Exercice 1

1/ Sélectionner la date de naissance, nom et le prénom de tous les clients.

2/ Sélectionner tous les champs de la table client.

3/ Sélectionner le numéro d'agence, ville et le nom des toutes les agences.

4/ sélectionner toutes les lignes de la table Agence.

Remarque: Les clauses SELECT et FROM doivent obligatoirement apparaître au début de chaque requête, on peut, ensuite, indiquer des critères de sélection avec la clause WHERE :

SELECT * FROM table WHERE condition;

La condition peut être formée en utilisant les opérateurs de condition, les opérateurs logiques, les clauses IN, BETWEEN, LIKE, GROUP BY

Les opérateurs de condition

On peut utiliser les opérateurs suivants dans les conditions:

= Égale

<> Différent

< Inférieur

> Supérieur

>= Supérieur ou égale

<= Inférieur ou égale

Pour sélectionner tous les articles dont le prix est supérieur à 100DH :

```
SELECT * FROM produit WHERE prix > 100;
```

Les opérateurs logiques

L'opérateur AND

Il est possible de combiner plusieurs conditions avec des opérateurs logiques AND et OR :

SELECT * FROM Client WHERE nom = 'Dupond' AND ville = 'Paris';

Sélectionne tous les clients dont le nom est Dupond et qui habitent Paris

L’opérateur OR

SELECT * FROM Clients WHERE ville = ‘Toulouse' OR ville = ‘Paris';

Sélectionne tous les clients qui habitent Paris ou Toulouse

L’opérateur AND et OR

Il est possible de combiner plusieurs conditions avec des opérateurs logiques AND et OR :

SELECT * FROM Client WHERE nom = 'Dupond' AND (ville = ‘Paris‘ OR ville = ‘Toulouse’);

Sélectionne tous les clients dont le nom = Dupond et qui habitent Paris ou Toulouse

La clause IN et BETWEEN

Elles servent à sélectionner des enregistrements dont la valeur d'un champ peut être comprise dans une liste où entre deux valeurs.

SELECT * FROM Clients WHERE ville IN ('Paris','Toulouse');

Sélectionne tous les clients dont la ville est Paris ou Toulouse.

SELECT * FROM Clients WHERE age BETWEEN 17 AND 20;

La clause LIKE

Elle permet de faire des recherches approximatives sur le contenu d'un champ en utilisant des caractères jokers comme ‘*’, ‘?’, ‘%’ le symbole * remplace un ensemble de caractères alors que le symbole ? ne remplace qu'un seul caractère.

Si on a deux clients nommés Dupond et Dupont, on utilisera 'Dupon?'.

SELECT * FROM Client WHERE nom LIKE 'S*';

La clause GROUP BY

Elle permet de regrouper la sélection par un des champs de sélection.

Select ville, count(*) from client group by ville

Cette requête va retourner les villes des clients avec un compteur de nombre de client par ville

Remarque importante: Le critère du group by doit absolument figurer comme critère de sélection.

La clause HAVING

Elle précise quels groupes doivent être sélectionnés, elle se place après la clause group by

Select ville, count(*) from client group by ville

Cette requête va retourner les villes des clients avec un compteur de nombre de client par ville, si nous souhaitons afficher rien que les ville ayant le nombre de client > 20 la requête sera :

Select ville, count(*) from client group by ville
HAVING count(*) > 20

La clause ORDER BY

Pour trier le résultat d'une requête on peut utiliser la clause order by qui fait le tri soit par ordre croissant ou décroissant :

SELECT Nom, Prénom FROM Client ORDER BY Nom, Prénom DESC;

La requête précédente effectue un tri croissant sur les noms, suivi d'un tri décroissant sur les prénoms.

Les fonctions d’ensemble

SQL dispose de cinq fonctions importantes : SUM, AVG, MAX, MIN et COUNT. On les appelle fonctions d'ensemble parce qu'elles résument le résultat d'une requête plutôt que de renvoyer une liste d'enregistrements.

Exemple :

SELECT MIN(prix_unitaire), MAX(prix_unitaire), AVG(prix_unitaire)

Va retourner le prix le plus petit de la table Produit, le prix le plus élevé et le prix moyen.

Les opérations ensemblistes

1- L'union (union)

Pour fusionner les résultats de deux requêtes ou deux tables on utilise l'opérateur UNION :

```
SELECT nom_client FROM client_Janvier UNION
SELECT nom_client FROM client_Mars;
```

Remarque: L'union se fait sur des résultats de même structure.

2- La différence (-)

A-B Consiste à récupérer les enregistrements qu'on a dans la table A et qui ne figurent pas dans la table B

3- La produit cartésien (X)

Sert à croiser les enregistrement de deux tables qui peuvent avoir deux structures différentes

4- L'intersection (intersect)

Permet d'obtenir l'ensemble des lignes communes à deux tables

EXERCICE D'APPLICATION

Soit la table Etudiant suivante:

N°	Nom	Prénom	Age	Sexe	Ville
1	KACHLOUL	Hassan	17	M	Rabat
2	MRABET	Salwa	19	F	Essaouira
3	ZAHOURI	Ali	18	M	Casablanca
4	SEBTI	Rachid	18	M	Marrakech
5	EDDAHIR	Nourddine	23	M	Agadir
6	EL HILALI	Khalil	21	M	Rabat
7	RIZKI	Laila	21	F	Tanger
8	MOUJAHID	Samir	19	M	Essaouira
9	RACHACH	Hamid	20	M	Casablanca
10	BAKIZ	Fatima	22	F	Oujda

Données à sélectionner :

- Tous les étudiants;
- Les étudiants qui ont l'âge égal à 21;
- Les étudiants qui ont l'âge supérieur à 20;
- Les étudiants qui habitent Casablanca;
- Les étudiants qui habitent Casablanca et qui ont l'âge supérieur à 20;
- Les étudiants qui habitent Casablanca ou Rabat;
- Les étudiants qui habitent Casablanca et de sexe masculin;
- Les étudiants qui ont l'âge entre 19 et 23;

- Le nom et le prénom des étudiants qui habitent rabat et qui ont l'âge inférieur ou égal à 22
- La moyenne d'âge des étudiants
- Le nom des étudiants classés par ordre décroissant
- Le nom, le prénom et la ville des étudiants regroupés par ville
- Le nom, le prénom et la ville des étudiants regroupés par ville et qui ont l'âge supérieur à 20
- Les étudiants qui habitent casa ou rabat et qui ont le nom qui commence par 'm'