UNIVERSITÉ DE NANTES

IUT NANTES
Université de Nantes

Département Informatique

BUT 1

Ressource R1.05 :
Introduction aux bases de données et SQL

4 décembre 2021

Cours et exercices

Table des matières

1 Le langage SQL : consultation des données

1 Introduction

SQL (Structured Query Language) a été conçu et implémenté initialement au centre de recherche de IBM

SQL est un langage

1. de définition de bases de données,
2. de manipulation de bases de données (consultation et mise à jour de données),

SQL est désormais un standard désigné sous le nom SQL-92

Consultation et manipulation de données

1. Consultation de données
   - SELECT : consulter de données
2. Création et manipulation de données :
   - CREATE : création de tables
   - ALTER : modifier la structure d'une table
   - INSERT : insérer de données
   - UPDATE : modifier de données
   - DELETE : supprimer de données

Consultation de données : types de requêtes Une requête de consultation de données SQL peut être :

1. Requête de base
2. Requête de groupement
3. Requête multitables
4. Requête imbriquée

2 Requête de base

Une requête de base est composée essentiellement de trois clauses :

- SELECT : renvoie le résultat sous la forme d'une liste d'attributs. Des fonctions d'agrégats peuvent être appliquées,
- FROM : liste des relations impliquées dans la requête
- WHERE : permet de spécifier une condition qui sera respectée. Cette clause est facultative

Requête de base Une requête SQL est formulée suivant la syntaxe suivante :

SELECT [DISTINCT | ALL] $\{*|[ExpressionColonne\,[As\ nouveauNom]]\,[,...]\}$
FROM table [alias] [,...]
[WHERE condition]
$[\{UNION|INTERSECT|MINUS\} < SELECT >]$
[ORDER BY <nom d'attribut> | <numéro | colonne> [ASC | DESC] [...]]

Requête de base

- Retrouver toutes les colonnes et toutes les lignes :
  *select * from R*,
- Retrouver quelques colonnes et toutes lignes :
  select A, B from R,
- Retrouver quelques lignes et toutes les colonnes :
  *select * from R where P*,
- Utilisation de distinct :
  select distinct A from R,
- Champs calculés et alias :
  *select A*10/100 from R*,

Requête de base Conditions de recherche (clause Where) :

- Comparaison : {>, <, =, <>, <=, >=, ! =},
  *select * from R where A > 2*
- Plusieurs critères de recherche : $\{OR, AND\}$,
  *select * from R where A = a and B > b*
- Conditions de recherche de type intervalle : Between, NOT Between,
  *select * from R where C [not] between v1 and v2*

Requête de base Conditions de recherche (clause Where) :

- Conditions de recherche de type appartenance : In, NOT in,
  *select * from R where c [not] in (v1,v2, v3)*
- Conditions de recherche de type correspondance : Like, NOT like,
  select * from R where c [not] like 'Na_tes' ; select * from R where c like 'N%tes'

— Conditions de recherche de type null : is null, is not null,
*select * from R where c is [not] null*
— Trier les résultats : Order By ASC, DSC,
select A,B from R where P order by A
select A,B from R where P order by 1 DESC, 2 ASC

3 Requête de groupement

— Group By : forme des groupes de lignes de même valeur de colonne sur lesquelles on peut appliquer des fonctions d'agrégats
— Utilisation des fonctions d'agrégat : count(*), count(distinct), SUM(Attribut), MIN(Attribut), MAX(Attribut), AVG(Attribut)
— Having : exprime une condition sur les groupes

A	B	C
1	2	3
1	0	1
5	1	1
5	2	3

TABLE 1.1: Table R

select A, sum(B) from R group by A

A	sum(B)
1	2
5	3

TABLE 1.2: Table Résultat

A	B	C
1	2	3
1	0	1
5	1	1
5	2	3

TABLE 1.3: Table R

Requête de groupement *select A, count(*) from R where C=1 group by A*

A	count(*)
1	1
5	1

TABLE 1.4: Table Résultat

A	B	C
1	2	3
1	0	3
5	1	1
5	2	3

TABLE 1.5: Table R

Requête de groupement *select B, count(*) from R where C=3 group by B having count(*) > 1*

B	count(*)
2	2

TABLE 1.6: Table Résultat

Requête de groupement L'évaluation d'une telle requête peut être décomposée en quatre actions :

- Evaluation de la clause Where
- Evaluation de la clause Group by
- Evaluation des fonctions reprises dans la clause Select
- Evaluation de la clause Having
- Evaluation de la projection

4 Requêtes multitables (jointures entre tables)

Une requête multitables permet d'obtenir des informations à partir de plusieurs tables.

Syntaxe : requêtes multitables SELECT [DISTINCT | ALL] * | [ExpressionColonne [As nouveauNom]] [,...]
FROM table [alias] [,...]
WHERE <predicat de jointure>

Une jointure externe est identifiée par le symbole (+) dans la clause where.

Requêtes multitables

- Produit cartésien : *select * from R1, R2*
- Θ-Jointure : *select * from R1, R2 where A < B*
- Jointure naturelle : *select * from R1, R2 where R1.A=R2.B* (A est une clé primaire dans R1 et B est une clé étrangère dans R2)
- Jointure externe gauche : *select * from R1, R2 where R1.A(+)=R2.B*
- Auto-jointure : select r1.A, r2.B from R r1, R r2 where r1.C > r2.C

Requêtes multitables - les jointures normalisées

- Jointure interne : *Select ... From nom_table_1 [INNER] JOIN nom_table_2 On Condition_de_Jointure*
  *select * from R1 INNER join R2 on A < B*
- Jointure externe : *Select ... From nom_table_1 [INNER] LEFT | RIGHT | FULL OUTER JOIN nom_table_2 On Condition_de_Jointure*
  *Jointure externe gauche : select * from R1 Left Outer Join R2 ON R1.A=R2.A*

- Jointure naturelle :*Select * From nom_table_1 NATURAL JOIN nom_table_2*

5 Requête imbriquée (sous_requête)

Dans une requête imbriquée une instruction SELECT (sélection interne) est imbriquée dans une autre instruction SELECT (sélection principale).
Les requêtes imbriquées peuvent être des :

- requêtes indépendantes renvoyant une seule ligne
  *select * from R1 where P = (select max(P) from R2)*
- requêtes indépendantes renvoyant plusieurs lignes
  *select * from R1 where P IN (select P from R2)*
  *select * from R1 where P < ANY (select P from R2)*
  *select * from R1 where P > ALL (select P from R2)*

- requêtes avec plusieurs colonnes
  *select * from R1 where (P,K) IN (select D,E from R2)*

Requête imbriquée (sous_requête)

- sous_requêtes multiples. Une requête princpale peut contenir plusieurs sous_requêtes reliées par les connecteurs AND et OR :
  *select * from R1 where P IN (select P from R2) AND K > ANY (select K from R3 where j=val)*
- sous_requêtes dépendantes de la requête principale. Une sous_requête est dite dépendante lorsqu'elle fait référence à un attribut issu d'une relation qui se trouve dans

la clause From de la requête principale
*select * from R1 where P IN (select D from R2 where **R1.A** = R2.A)*

Requête imbriquée (sous_requête) Une des formes particulières des sous_requêtes dépendantes de la requête principale est celle testant l'existence de ligne de valeurs répondant à une condition. Le mot clé EXISTS est placé devant une sous_requête.

*select * from R1 where [NOT] EXISTS (select * from R2 where **R1.A** = R2.A)*

2 TD/TP

0.1 *La base de données Basetd*

Nous utiliserons la base de données *Basetd* déjà installée sur le serveur Oracle. Elle est composé de quatre tables : Employé, Service, Projet, Travail et Concerne. Vous avez des droits de consultation et donc vous ne pourrez pas modifier ni les données ni la structure des relations de cette base de données.

1 Requêtes de base SQL de consultation de données

1.1 *Exercice :*

En utilisant la base de données 'Basetd' sous Oracle, exprimez en langage naturel les requêtes SQL suivantes :

1 ► select nomempl from basetd.employe,basetd.travail where employe.nuempl=travail.nuempl ;

2 ► select nomempl from basetd.employe where nuempl in (20,30,42) ;

3 ► select nuproj from basetd.projet where nomproj Like 'C%' ;

4 ► select nuempl from basetd.employe where hebdo is NULL ;

5 ► select * from basetd.employe, basetd.travail where travail.nuempl(+) = employe.nuempl ;
Quel nom peut-on donner à cette requête ? ;

6 ► select * from basetd.employe e NATURAL JOIN basetd.travail t ;

7 ► ***Select nuempl,nomempl from basetd.employe union select nuempl, nuproj from basetd.travail*** *;*

8 ► select nuproj from basetd.projet intersect select nuproj from basetd.travail ;

9 ► select nuempl from basetd.employe minus select nuempl from basetd.travail ;

1.2 *Exercice :*

En se basant sur le schéma de la base de données Basetd, exprimez en SQL les interrogations suivantes :

1 ► Sélectionnez les numéros et noms de tous les employés dans la table Employé ;

2 ► Sélectionnez le nombre d'employés ;

3 ► Sélectionnez le temps hebdomadaire moyen de travail des employés ;

4 ► Sélectionnez la somme des durées consacrées par les employés aux projets ;

5 ► Affichez les noms des employés par ordre croissant ;

6 ► Affichez les numéros des employés et la durée de travail consacrée par chacun des employés sur chacun des projets. Les résultats doivent être triés par numéro employé (ordre décroissant) ;

7 ► Affichez le nom du service numéro 1 ;

8 ► Affichez les noms des autres services ;

9 ► Affichez les noms des employés qui ne travaillent pas ;

10 ► Affichez les noms des employés dont le temps de travail hebdomadaire est compris entre 20 et 30 (deux versions)

1.3 *Exercice :*

En se basant sur le schéma relationnel du "Basetd" , traduisez chacune des questions suivantes sous forme de requêtes SQL.

1 ► Liste des noms de services avec le nom du chef de service

2 ► Liste des noms d'employés avec pour chacun d'eux la liste des projets sur lesquels il travaille

3 ► Pour le service Achat, trouvez le nom du chef de service et les numéros de projets sur lesquels il travaille.

4 ► Liste des noms de projets avec le nom du responsable

5 ► Pour le projet Zorro, donnez le nom du responsable et les employés qui y travaillent.

6 ► Liste de tous les employés avec les projets sur lesquels ils travaillent. Si un employé n'est pas encore affecté à un projet il doit sortir dans le résultat.

2 Les sous_requêtes SQL de consultation de données

2.1 *Exercice :*

En utilisant "Basetd", exprimez en langage naturel les requêtes SQL suivantes. Justifiez votre réponse lorsque vous estimez qu'une opération ne peut pas être satisfaite.

1 ► Sous-requête indépendante renvoyant une seule ligne

1 ► select nomempl from basetd.employe where affect = (select nuserv from basetd.service where nomserv='achat') ;

2 ► select nomproj from basetd.projet where nuproj = (select nuproj from basetd.travail where nuempl=20) ;

2 ► Sous-requête indépendante renvoyant plusieurs lignes

1 ► select nomempl from basetd.employe where nuempl in (select nuempl from basetd.travail where duree=5) ;

2 ► select nomempl from basetd.employe where nuempl not in (select nuempl from basetd.travail) ;

3 ► select nomempl from basetd.employe where nuempl != All (select nuempl from basetd.travail) ;

4 ► select nomempl from basetd.employe where nuempl = Any (select nuempl from basetd.travail where nuproj = 103) ;

3 ► Sous-requête dépendante de la requête principale

1 ► select distinct nomempl from basetd.travail t, basetd.employe e where t.nuempl = e.nuempl and t.nuproj = (select nuproj from basetd.projet p where p.nuproj = t.nuproj and p.resp=30) ;

2 ► select distinct nomempl from basetd.employe e where exists (select * from basetd.travail t where e.nuempl = t.nuempl) ;

3 ► select distinct nomempl from employe e where not exists (select * from basetd.travail t where e.nuempl = t.nuempl) ;

4 ► Sous-requête avec plusieurs colonnes

1 ► select nomempl from basetd.employe where (nuempl,affect) in (select chef,affect from basetd.service) ;

2 ► select nomempl,nomproj from basetd.projet p , employe e where (p.nuproj,e.nuempl) in (select nuproj, nuempl from basetd.travail) ;

5 ► Groupement des données

1 ► Calcul sur un seul groupe

1 ► select count(*) from basetd.employe ;

2 ► select avg(hebdo) from employé where affect = 2 ;

2 ► Calcul sur plusieurs groupes

1 ► select affect,count(*) from basetd.employe group by affect ;

2 ► select resp "Responsable", count(*) as NbProjet from basetd.projet group by resp ;

3 ► Sélection de groupe

1 ► select affect "Service", count(*) as Nbemploye from basetd.employe group by affect having count(*) > 4 ;

2 ► select e.nuempl,e.nomempl from basetd.employe e, basetd.travail t, basetd.projet p where e.nuempl=t.nuempl and t.nuproj = p.nuproj group by e.nuempl, e.nomempl having count(*) = (select count(*) from basetd.projet) ;

2.2 *Exercice :*

1 ► Exprimez en SQL les requêtes suivantes :

- 1 ► Liste des noms de projets avec le nom du responsable et le nombre d'employés qui y travaillent
- 2 ► Liste des noms de projets avec la totalisation du nombre d'heures passées par les employés qui y travaillent
- 3 ► Liste des noms de projets avec pour chaque service concerné, le nom du service et le nombre d'employés de ce service qui travaillent sur ce projet
- 4 ► Liste des employés qui travaillent sur tous les projets
- 5 ► Pour le service Achat, trouvez le nom du chef de service et le nombre d'employés qui y sont affectés
- 6 ► Liste des employés qui travaillent sur au moins un des projets sur lesquels 'Sophie' travaille.