

INTRODUCTION A SQL (Structured Query Language)

INTRROGATION D'UNE BASE DE DONNEES

Syntaxe d'une requête SQL de base

<requete> := **SELECT** <liste_resultat | * >
 FROM <liste_relations>
 [**WHERE** <liste_conditions>]
 [**ORDER BY** <liste_attributs_a_trier>]

où :

liste_resultat := [**DISTINCT**] <attribut1 | expression1 > [<alias1>] [, <attribut2 | expression2> [<alias2> ...]

<liste_relations> := <relation1> [alias1] [, <relation2> [alias2] ...]

<liste_conditions> := [**NOT**] <condition1> [**AND** | **OR** <condition2> ...]

Expression des Projections et Sélections

Condition de sélection (Format général) :

<conditioni>:= <attribut | expression> <comparateur | predicat_cond> <constante>

<predicat_cond> := IS NULL | IN | BETWEEN ... AND | LIKE | IS NOT NULL | NOT IN | NOT BETWEEN | NOT LIKE

Remarques

- <attribut> est le nom d'un attribut
- Par la suite <attribut> peut désigner « expression » comme Sal*1.1
- θ est un comparateur appartenant à {'=', '!=', '>=', '>', '<=', '<'}

- <attribut> θ constante

Donner le nom et l'adresse des pilotes qui gagnent plus de 6000 €.

```
SELECT   NOMPIL, ADR  
FROM     PILOTE  
WHERE    SAL > 6000
```

En plus des opérateurs de comparaison classiques, SQL dispose des prédicats spécifiques suivants :

- <attribut> IS NULL

Teste si la valeur d'une colonne est une valeur nulle (manquante).

Rechercher le nom des pilotes dont l'adresse est inconnue.

```
SELECT   NOMPIL  
FROM     PILOTE  
WHERE    ADR IS NULL
```

- **<attribut> IN (<liste_valeurs>)**

Vérifie si la valeur d'un attribut coïncide avec l'une des valeurs de la liste. Rechercher toutes les informations sur les avions de nom A310, A320, A330 et A340.

```
SELECT  *
FROM    AVION
WHERE   NOMAV IN ('A310', 'A320', 'A330', 'A340')
```

- **<attribut> BETWEEN v1 AND v2**

Teste si la valeur d'un attribut est comprise entre les valeurs v1 et v2 avec v1 <= valeur <= v2.

Quel est le nom des pilotes qui gagnent entre 6.000 et 7000 € ?

```
SELECT  NOMPIL
FROM    PILOTE
WHERE   SAL BETWEEN 6000 AND 7000
```

- **<attribut> LIKE 'chaîne_générique'**

Teste si la valeur d'un attribut alphanumérique est équivalente à une chaîne de caractère obtenue à partir de la chaîne générique. Dans cette dernière, le symbole % remplace une série de caractères quelconque, y compris le vide.

Quelle est la capacité des avions de type Airbus (i.e. dont le nom commence par A) ?

```
SELECT  CAPACITE
FROM    AVION
WHERE   NOMAV LIKE 'A%'
```

- **ROWNUM <= n (LIMIT n avec MySQL)**

Sélectionne les n premiers tuples résultats.

Quelle sont les noms et les capacités des avions. Afficher uniquement les premiers avions

```
SELECT  NOMCAPACITE
FROM    AVION
WHERE   ROWNUM <= 2
```

Conditions négatives

Tous les prédicats spécifiques peuvent être mis sous forme négative en les combinant avec l'opérateur de négation NOT. Nous obtenons alors IS NOT

NULL, NOT IN, NOT LIKE et NOT BETWEEN. De plus la négation de toute condition peut être obtenue en faisant précéder la condition de NOT.

Quels sont les noms des avions différents de A310, A320, A330 et A340 ?

```
SELECT  NOMAV
FROM    AVION
WHERE   NOMAV NOT IN ('A310', 'A320', 'A330', 'A340')
```

Quels sont les noms des avions différents de B747 ?

```
SELECT  NOMAV
FROM    AVION
WHERE   NOT NOMAV = 'B747'
```

Combinaisons de conditions

La condition dans la clause WHERE peut être une condition composite, i.e. une combinaison de conditions reliées par les opérateurs logiques AND et OR. L'opérateur AND est prioritaire et, si nécessaire, le parenthèse doit être utilisé pour spécifier un ordre d'évaluation particulier.

Quels sont les vols au départ de Nice desservant Paris ?

```
SELECT  *
FROM    VOL
WHERE   VILLE_DEP = 'NICE' AND VILLE_ARR = 'PARIS'
```

Quels sont les pilotes niçois et marseillais gagnant plus de 6000 € ?

```
SELECT  *
FROM    PILOTE
WHERE   (ADR = 'NICE' OR ADR = 'MARSEILLE')
        AND SAL > 6000
```

Quels sont les pilotes n'étant ni niçois ni marseillais et gagnant plus de 6.000 € ?

```
SELECT  *
FROM    PILOTE
WHERE   NOT (ADR = 'NICE'
OR       ADR = 'MARSEILLE')
AND     SAL > 6000
```

Calculs verticaux (fonctions agrégatives)

<nom_fonction> ([**DISTINCT**]<attribut>)

où :

<nom_fonction> := **SUM** | **AVG** | **COUNT** | **MAX** | **MIN** | **STDDEV** | **VARIANCE**

Les fonctions agrégatives disponibles sont les suivantes :

SUM	somme,
AVG	moyenne arithmétique,
COUNT	nombre ou cardinalité,
MAX	valeur maximale,
MIN	valeur minimale,
STDDEV	écart type (<i>standard deviation</i>), racine carrée de la variance,
VARIANCE	variance.

```
SELECT  AVG(SAL)
FROM    PILOTE
WHERE   ADR = 'NICE'
```

Trouver le nombre de vols au départ de Marseille.

```
SELECT  COUNT(VOLNUM)
FROM    VOL
WHERE   VILLE_DEP = 'MARSEILLE'
```

Dans cette requête, COUNT(*) peut être utilisé à la place de COUNT(VOLNUM).

Donnez les capacités minimale et maximale des appareils localisés à Paris.

```
SELECT  MIN(CAPACITE), MAX(CAPACITE)
FROM    AVION
WHERE   LOC = 'PARIS'
```

Combien de destinations sont desservies au départ de Marseille ?

```
SELECT  COUNT(DISTINCT VILLE_ARR)
FROM    VOL
WHERE   VILLE_DEP = 'MARSEILLE'
```

Tri des résultats

ORDER BY <attribut1|expression1> [ASC | DESC] [NULLS FIRST | NULLS LAST] [, <attribut2|expression2> [ASC | DESC] [NULLS FIRST | NULLS LAST]...]

Quelle est la liste des pilotes, selon l'ordre alphabétique de leur nom, effectuant un vol au départ de Paris ?

```
SELECT DISTINCT NOMPIL
FROM VOL, PILOTE
WHERE VOL.NUMPIL = PILOTE.NUMPIL AND VILLE_DEP = 'PARIS'
ORDER BY NOMPIL
```

Donner la liste des pilotes niçois par ordre de salaire décroissant, puis par ordre alphabétique des adresses.

```
SELECT      NOMPIL, SAL
FROM P      PILOTE
WHERE       ADR = 'NICE'
ORDER      BY SAL DESC, ADR
```

Expression des jointures

Jointures Prédicatives

Condition de jointure prédictive (Format général) :

<conditionjp>:= < attribut1|expression1> <comparateur> <attribut2|expression2>

Quels sont les avions (numéros et noms) localisés dans la même ville que l'avion numéro 103 ?

```
SELECT  AUTRES.NUMAV AUTRES.NOMAV
FROM    AVION AUTRES, AVION AV103
WHERE   AV103.NUMAV = 103
AND     AUTRES.NUMAV <> 103
AND     AV103.LOC = AUTRES.LOC
```

Jointure Imbriquées

Condition de jointure imbriquée (Format général) :

<conditionji>:= (<attribut1|expression1>[, < attribut2|expression2>,...]) θ (<requete>)
| (<attribut1|expression1>[, < attribut2|expression2>,...]) θ ANY | IN (<requete>)
| (<attribut1|expression1>[, < attribut2|expression2>,...]) θ ALL (<requete>)

Remarques

- Par la suite « attribut » peut désigner « expression » comme Sal*1.1
- θ est un comparateur appartenant à {'=', '!=', '>=', '>', '<=', '<'}
- Les parenthèses ne sont pas obligatoires si <attribut1|expression1> est unique

- (<attribut1>[, < attribut2>,...]) θ (<requete>)

Quel est le nom des pilotes gagnant plus que le salaire moyen des pilotes ?

```
SELECT  NOMPIL
FROM    PILOTE
WHERE   SAL > (SELECT AVG(SAL) FROM PILOTE)
```

- (<attribut1>[, < attribut2>,...]) θ ANY | IN (<requete>)

Quel est le nom des pilotes en service au départ de Nice ?

```
SELECT  NOMPIL
FROM    PILOTE
WHERE   NUMPIL IN      (SELECT NUMPIL
                        FROM VOL
                        WHERE VILLE_DEP = 'NICE')
```

On peut remplacer IN par =ANY

Quel est le numéro des avions localisés à Nice dont la capacité est supérieure à celle de l'un des appareils effectuant un Paris-Nice ?

```
SELECT NUMAV
FROM    AVION
WHERE   LOC = 'NICE'
AND CAP > ANY
        (SELECT CAP
         FROM    AVION
         WHERE NUMAV = ANY
              (SELECT NUMAV
               FROM VOL
               WHERE VILLE_DEP = 'PARIS' AND
```

- (<attribut1>[, < attribut2>,...]) θ ALL (<requete>)

Rechercher le nom des pilotes dont l'adresse et le salaire sont différents de ceux de tous les pilotes appelés Dupont.

```
SELECT NOMPIL
FROM    PILOTE
WHERE   (ADR, SAL) <> ALL
        (SELECT ADR, SAL
         FROM PILOTE
         WHERE NOMPIL = 'DUPONT')
```

Jointure algébrique

```
SELECT < attribute1|expression1> [, <attribut2|expression2>...]  
FROM <nom_table1> [alias]  
[INNER] JOIN <nom_table2> [alias]  
ON < attribute1|expression1>  $\theta$  < attribute1|expression2> [AND <condition>...]  
[[INNER] JOIN <nom_table3> [alias]...]  
[WHERE <condition>]
```

Nom des pilotes en service au départ de Marseille

```
SELECT  NOMPIL  
FROM    PILOTE  
JOIN    VOL  
PILOTE. NUMPIL = VOL.NUMPIL  
WHERE   VILLE_DEP = 'MARSEILLE')
```

Nom des pilotes effectuant des vols au départ de leur ville de résidence

```
SELECT  NOMPIL  
FROM    PILOTE PIL  
JOIN VOL VO  
ON      PIL.NUMPIL = VO.NUMPIL AND ADR = VILLE_DEP
```

Opérateurs ensemblistes

La forme d'une opération d'union, d'intersection ou de différence est :

```
<requete1>  
UNION | INTERSECT | MINUS  
<requete2>
```

Les résultats de requête 1 et requête2 doivent être uni-compatibles (même nombre d'attributs et attributs respectifs comparables)

UNION

Quel est le nom des avions de capacité supérieure à 250 ou localisés à Paris ?

```
SELECT NOMAV FROM  AVION WHERE CAP > 250  
UNION  
SELECT NOMAV FROM  AVION WHERE  LOC = 'PARIS'
```

INTERSECT

Quel est le numéro des pilotes qui conduisent les avions n° 2 et 4 ?

```
SELECT NUMPIL FROM VOL WHERE NUMAV = 2
```

INTERSECT

```
SELECT NUMPIL FROM
```

```
VOL WHERE NUMAV = 4
```

MINUS

Quels sont les numéros des pilotes qui conduisent l'avion n° 2 sans jamais conduire l'avion n° 4 ?

```
SELECT NUMPIL FROM VOL WHERE NUMAV = 2
```

MINUS

```
SELECT NUMPIL FROM VOL WHERE NUMAV = 4
```


INTRODUCTION A SQL

MISE A JOUR D'UNE BASE DE DONNEES

Modification d'un tuple

UPDATE <nom_table>
SET <attribut1> = <expression1>[, <attribut2> = <expression2> ...]
[**WHERE** <condition_selection>]

Augmenter de 10% les salaires des pilotes Parisiens

UPDATE PILOTE
SET S **AL = SAL*1.1**
WHERE **ADR = 'PARIS'**

UPDATE <nom_table>
SET (<attribut1>[, <attributn>...]) =
(**SELECT** <attribut1>[, <attributn>...] **FROM** ... **WHERE** ...)
WHERE <condition_selection>

Affecter le salaire maximal des pilotes Marseillais aux salaires des pilotes Parisiens

UPDATE PILOTE
SET SAL = (**SELECT MAX(SAL)**
FROM PILOTE
WHERE ADR = 'MARSEILLE')
WHERE ADR = 'PARIS'

Ajout d'un tuple

INSERT INTO <nom_table> [(<liste_attributs>)]
VALUES (<valeur1>[, <valeur2> ...])

Ajouter le pilote (110, MARTIN, NICE, 10000) dans la table PILOTE

INSERT INTO PILOTE
VALUES(110, 'MARTIN', 'NICE', 10000)

Ajouter le pilote (112, 'DUPONT, NULL, NULL) dans la table PILOTE

INSERT INTO PILOTE (NUMPIL, NOMPIL)
VALUES(112, 'DUPONT')

Autre Expression

INSERT INTO PILOTE (NUMPIL, NOMPIL)
VALUES(112, 'DUPONT', NULL, NULL)

INSERT INTO <nom_table> [(<liste_attributs>,...)] <requête>

Ajouter les pilotes Marseillais dans la table PILMARS

INSERT INTO PILMARS

SELET * FROM PILOTE

WHERE ADR = 'MARSEILLE'

Suppression d'un tuple

DELETE FROM <nom_table> **WHERE** <condition>

Supprimer tous les pilotes

DELETE FROM PILOTE

Supprimer les pilotes habitant Paris

DELETE FROM PILOTE

WHERE ADR = 'PARIS'

COMMIT valide une opération de mise à jour

ROLLBACK Invalide une opération de mise à jour