

Le Langage SQL

<http://www.lirmm.fr/~poncelet/HLIN605.html>

Pascal Poncelet
LIRMM
Pascal.Poncelet@lirmm.fr
http://www.lirmm.fr/~poncelet

Introduction

- Origine : développé chez IBM par l'équipe de recherche de Codd - Naissance de Sequel
- Devenu un standard ANSI en 89 (American National Standard Institute)
 - Existence de « différents dialectes » mais les différences de syntaxe sont minimes
- « Aujourd'hui, un SGBDR ne se vend pas sans une interface SQL »

Introduction

- 3 facettes
 - Langage de définition de données (LDD)
 - Crédit du schéma
 - Langage de manipulation de données (LMD)
 - Mise à jour et interrogation du schéma
 - Langage de contrôle des données (LCD)
 - Autorisations
- SQL est basé sur des mots clés (en anglais) et se veut proche du langage naturel
- Table = Relation ; Colonne = Attribut ; Ligne = tuple

Langage de Manipulation (LMD)

- Structure du bloc de base

SELECT {liste des attributs résultats}	<i>Obligatoire</i>
FROM {liste des relations concernées}	<i>Obligatoire</i>
WHERE {liste des conditions}	<i>Facultatif</i>
;	<i>Fin de requête</i>

LMD - Projections

- Expression des projections
 - Dans la clause SELECT
 - Les différents attributs sont séparés par des virgules (,)
- Liste des villes desservies par la compagnie ?
 - **SELECT VA FROM VOL;**
- Noms et Adresses des pilotes ?
 - **SELECT PINom, Adr FROM PILOTE;**

LMD - Projections

- Attention : avec SQL il n'y a pas d'élimination automatique des dupliques, c'est à l'utilisateur de le spécifier avec la clause **DISTINCT**
SELECT DISTINCT VA FROM VOL ;
- Pour ne pas réaliser de projection, soit on cite tous les attributs de la (les) relations, soit on utilise *
Toutes les informations sur les pilotes
SELECT * FROM PILOTE;

LMD - Sélections

- Expression des sélections
 - Dans la clause **WHERE** sous la forme d'une condition :
Attribut θ Constante où $\theta \subseteq \{=, \neq, <, \leq, >, \geq\}$
Quels sont les pilotes Niçois ?
SELECT * FROM PILOTE WHERE Adr='Nice' ;
- Possibilité d'utiliser des connecteurs logiques **AND, OR, NOT**

LMD - Sélections

- Définition d'appartenance à un intervalle
WHERE att BETWEEN borneinf AND bornesup
- Appartenance à une liste
WHERE att IN (val1, val2, ..., valn)
att=val1 ou att = val2 ou ... att = valn
- Recherche de sous chaînes
WHERE att LIKE « chaîne générique »
où chaîne générique est une suite de caractères quelconques pouvant inclure (_) pour un caractère quelconque ou % pour une chaîne quelconque). Sous Unix _ = ? et % = *

LMD - Sélections

- Donner tous les informations sur les Airbus dont le numéro est compris entre 100 et 150 et qui sont localisés à Nice, Marseille, Toulouse ou Bordeaux

```
SELECT * FROM AVION  
WHERE Avnom LIKE « Airbus% » AND  
Avnum BETWEEN 100 AND 150 AND  
Loc IN (« Nice », « Marseille », « Toulouse »,  
« Bordeaux ») ;
```

LMD - Exercice

- Exprimer en SQL les requêtes suivantes :

Donner les horaires des vols au départ de Paris à destination de Nice

Quels sont les numéros et noms des avions, localisés à Nice et Marseille et dont la capacité est supérieure à 150 places ?

Quels sont les noms des pilotes gagnant entre 2000 et 3000 € et dont le nom commence par B ou C ?

LMD - Calculs verticaux

- Dans un mapping, il est possible d'utiliser des fonctions agrégatives, appliquées sur les valeurs d'attributs. Ces fonctions sont :
 - Sum, Avg, Min, Max, Stddev, Variance, Count
 - Quel est le total des salaires des pilotes ?
SELECT SUM(Sal) FROM PILOTE ;
- Utiliser **DISTINCT** si vous ne voulez pas que le calcul se fasse sur les dupliques
- **COUNT** admet * comme argument = rend le nombre de tuples sélectionnés

LMD - Calculs verticaux

- Exemples
- Quel est le nombre de villes desservies par la compagnie ?
SELECT COUNT (DISTINCT VA) FROM VOL ;
- Quel est le nombre de Vols à destination de Nice ?
SELECT COUNT (*) FROM VOL WHERE VA = « Nice »;

LMD - Calculs horizontaux

- On peut également exprimer des calculs dans un mapping SQL en utilisant :
 - des opérateurs : +, -, *, / et || (concaténation de chaînes)
 - des fonctions : ABS, SQRT, COS, ...

Quels sont les noms des pilotes qui avec une augmentation de 10% de leur salaire gagnent moins de 2000 € ?

```
SELECT P.nom FROM PILOTE WHERE Sal * 1.1 < 2000;
```

LMD - Jointures prédictives

- Exprimées dans la clause WHERE sous forme $att1 \theta att2$ où $\theta \subseteq \{=, \neq, \leq, \geq\}$ et att1 est un attribut de R1, att2 est un attribut de R2 et att1, att2 sont compatibles
- Si les attributs de jointure portent le même nom, on lève l'ambiguïté en préfixant par le nom de la relation

Donner les numéros et horaires des vols au départ de Paris assurés par un A320 ?

```
SELECT Volnum, HD, HA FROM VOL, AVION  
WHERE VOL.Avnum = AVION.Avnum AND VD =  
« PARIS » AND Avnom = « A320 » ;
```

LMD - Autojointures

- Lorsque l'on utilise 2 fois la même relation dans un bloc, il faut utiliser des alias ou synonymes pour différencier les rôles joués par la relation

Numéros des pilotes gagnant le même salaire que Dupont ?

```
SELECT P1.Pnum FROM PILOTE P1, PILOTE P2  
WHERE P2.Pnom = « Dupont » AND  
P1.Pnum >> P2.Pnum AND P1.Sal = P2.Sal ;
```

LMD - Jointures imbriquées

- Il est possible d'imbriquer les blocs SQL en les parenthésant :
- 1er cas : Le résultat de la sous-requête est une unique valeur => utilisation d'un opérateur de comparaison entre les 2 blocs

Nom des pilotes qui gagnent plus que la moyenne

```
SELECT P.nom FROM PILOTE WHERE Sal >  
(SELECT AVG(Sal) FROM PILOTE) ;
```

LMD - Jointures imbriquées

- 2nd cas : le résultat de la sous requête est un ensemble de valeurs
 - Si la condition doit être vérifiée pour une des valeurs de la liste, on fait précéder la sous requête de **IN** ou **=ANY**

Nom des pilotes assurant un vol au départ de Nice

**SELECT Plnom FROM PILOTE WHERE Plnum IN
(SELECT Plnum FROM VOL WHERE VD = « Nice »);**

LMD - Jointures imbriquées

- 2nd cas : le résultat de la sous requête est un ensemble de valeurs
 - Si la condition doit être vérifiée pour toutes les valeurs de la liste, on fait précéder la sous requête de **θ ALL** où **θ** est un opérateur de comparaison

Noms des pilotes Niçois qui gagnent plus que les pilotes Parisiens ?

**SELECT Plnom FROM PILOTE WHERE Adr =
« Nice » **AND** Sal > ALL (SELECT DISTINCT Sal
FROM PILOTE WHERE Adr = « Paris »);**

LMD - Jointures imbriquées

- Possibilités de travailler sur un ensemble d'attributs

Quels sont les avions de même nom et localisés au même endroit que l'avion Numéro 105 ?

**SELECT * FROM AVION WHERE (Avnom, Loc) =
(SELECT Avnom, Loc FROM AVION WHERE Avnum
= 105);**

LMD - Exercices

- Quels sont les vols (numéros et horaires) assurés par le Pilote Jean Dupont ?
- Quelles sont les villes desservies par des A320 ?
- Quels sont les numéros et noms des avions localisés dans la ville de départ d'un vol à destination de Nice ?
- Quels sont les noms des pilotes Niçois pilotant un Airbus ?

LMD - Op. ensemblistes

- Syntaxe :
SFW {UNION, INTERSEC, MINUS} SFW
- Contraintes : unicompatabilité des relations opérandes - Attention à l'ordre => compatibilité syntaxique des attributs projetés dans les deux clauses SELECT
- Equivalence avec d'autres opérations d'interrogation

LMD - Op. ensemblistes

- Equivalence avec d'autres opérations d'interrogation

**SELECT Plnom FROM PILOTE WHERE
Adr = « Nice » **UNION** SELECT Plnom FROM
PILOTE **WHERE** Adr = « Paris » ;**

SELECT Plnom FROM PILOTE **WHERE Adr =
« Nice » **OR** Adr = « Paris » ;**

SELECT Plnom FROM PILOTE **WHERE Adr IN
« Paris », « Nice »)**

LMD - Op. ensemblistes

- Numéros des pilotes habitant Nice et n'assurant aucun vol au départ de Nice

SELECT Plnum **FROM PILOTE **WHERE** Adr =
« Nice » **MINUS** SELECT DISTINCT Plnum **FROM**
VOL **WHERE** VD = « Nice » ;**

SELECT Plnum **FROM PILOTE **WHERE** Adr =
« Nice » **AND** Plnum **NOT IN** (SELECT DISTINCT
Plnum **FROM** VOL **WHERE** VD = « Nice »);**

LMD - Tri des résultats

- Il est possible d'ordonner les résultats en SQL, ordre croissant ou décroissant sur un ou plusieurs attributs en utilisant la clause

ORDER BY expression [ASC, DESC], ... où expression est un attribut ou ensemble d'attributs spécifiés dans le SELECT

- Dernière clause du bloc
- Si plusieurs expressions, d'abord tri sur le 1er, puis le 2nd, ...

Liste des pilotes Niçois par ordre de salaire décroissant puis par ordre alphabétique des noms

SELECT Plnom, Sal **FROM PILOTE **WHERE** Adr =
« Nice » **ORDER BY** Sal **DESC**, Plnom ;**

LMD - Mise à Jour

- Modification de la valeur d 'un attribut
UPDATE nomrelation **SET** att1=val1, att2=val2 ...
* condition
- Si absence de conditions, il y a une modification sur tous les tuples de la relations concernées
- La nouvelle valeur peut être fonction de l'ancienne ou être le résultat d'une requête
- Des jointures peuvent être exprimées dans la clause WHERE mais une seule relation est spécifiée dans la clause UPDATE

LMD - Mise à Jour

- Exemples
- Le pilote Dupond change d'adresse et son salaire est augmenté de 10 %
UPDATE PILOTE SET Adr=« PARIS », Sal=Sal*1.1
WHERE Plnom = « Dupont »
- Le pilote de numéro 105 a maintenant le même salaire que le pilote numéro 110
UPDATE PILOTE SET Sal = (**SELECT** Sal **FROM** PILOTE **WHERE** Plnum=110) **WHERE** Plnum=105;

LMD - Mise à Jour

- Insertion d'un tuple
INSERT INTO nomrelation (list_att) **VALUES** (list_val) ;
- Si la liste des attributs n'est pas spécifiée, il faut donner les valeurs pour chacun des attributs de la relation dans l'ordre de création
- On peut utiliser le mot clé NULL si l'attribut n'a pas de valeur

LMD - Mise à Jour

- Insertion d'un nouveau pilote
INSERT INTO PILOTE (Plnum, Plnom, Adr, Sal) **VALUES** (206, « Dupond », « Montpellier », 3000) ;
- Remarque : si le pilote 206 existe déjà étant donné que Plnum est clé primaire il ne pourra pas être inséré

LMD - Mise à Jour

- Suppression de tuples
DELETE FROM nomrelation **WHERE** condition
- Une seule relation dans le FROM
- **DELETE FROM PILOTE WHERE** P1num=206;

LMD - Mise à Jour

- Attention : Une opération de mise à jour n'est pas inscrite définitivement dans la base après son exécution
- Notion de transactions (voir en cours plus tard)