

INFO-H-403 Bases de données

Séance d'exercices 7

SQL (1)

F. Servais et B. Verhaegen

6 novembre 2009

SQL

- ▶ Structured Query Language
- ▶ Langage de définition de de manipulation de données relationnelles

SQL DDL (Data Definition Language)

- Sous ensemble de SQL pour **définir la structure** de la base de données.

Création d'une table

```
CREATE TABLE Employee (  
    SSN          VARCHAR(9)          NOT NULL,  
    Name         VARCHAR(15)         NOT NULL,  
    SuperSSN     VARCHAR(9),  
    PRIMARY KEY (SSN),  
    FOREIGN KEY (SuperSSN) REFERENCES Employee(SSN)  
)
```

Suppression d'une table

```
DROP TABLE Employee
```

SQL DDL

Ajout d'une colonne

```
ALTER TABLE Employee  
    ADD BDate    DATE
```

Suppression d'une colonne

```
ALTER TABLE Employee  
    DROP BDate
```

Requêtes SQL

Structure générale

```
SELECT attributs  
FROM relations  
WHERE conditions
```

- ▶ Les attributs et les relations sont **séparés par des virgules**.
- ▶ Dans la clause WHERE, on peut placer :
 - ▶ des **comparaisons** binaires d'attributs et de valeurs
 - ▶ des **connecteurs** AND, OR, NOT
 - ▶ attribut **IN** ensemble de valeurs
 - ▶ **EXISTS**(sous-requête)

Requêtes SQL

Adresse et date de naissance des employés nommés John Smith

```
SELECT Address, BDate  
FROM Employee  
WHERE FName='John' AND LName='Smith'
```

Jointure : Noms des employés du département recherche

```
SELECT E.FName  
FROM Employee E, Department D  
WHERE E.DNum = D.DNum AND D.DName='Research'
```

Remarquez l'utilisation de variables.

Requêtes SQL : inclusion

Noms des employés du département 1, 2 ou 3

```
SELECT E.FName  
FROM Employee E  
WHERE E.Dno IN {1,2,3}
```

Noms des employés des départements dirigés par Bill

```
SELECT E.FName  
FROM Employee E  
WHERE E.Dno IN (SELECT D.Dno  
                 FROM Department D  
                 WHERE D.Manager = 'Bill')
```

Requêtes SQL : existence

Noms des employés qui ont au moins un dépendant

```
SELECT E.FName  
FROM Employee E  
WHERE EXISTS ( SELECT *  
                FROM Dependent D  
                WHERE E.SSN = D.ESSN )
```


Requêtes SQL : divers

- ▶ 'SELECT *' renvoie toutes les colonnes
- ▶ 'SELECT DISTINCT' renvoie les tuples distincts
- ▶ Le mot clé 'UNION' fait l'union de deux tables ayant le même nombre de colonnes en supprimant les doublons.
- ▶ La clause 'ORDER BY attributs' trie les résultats

Requêtes SQL et TRC

- ▶ Les requêtes en TRC peuvent se traduire facilement en SQL.
- ▶ Par contre, il n'y a pas de quantificateur universel en SQL.

$$\{y \mid \forall x (P(x) \rightarrow Q(x, y))\} \equiv \{y \mid \neg \exists x (P(x) \wedge \neg Q(x, y))\}$$

Formes équivalentes :

- ▶ Les employés qui travaillent sur **tous** les projets.
- ▶ Les employés pour lesquels **il n'existe pas** de projet sur lequel ils ne travaillent pas.

Quantificateur universel

Les employés pour lesquels il n'existe pas de projet sur lequel ils ne travaillent pas.

Traduction en SQL

```
SELECT FName, LName
FROM Employee
WHERE NOT EXISTS
  ( SELECT *
    FROM Project P
    AND NOT EXISTS
      ( SELECT *
        FROM WorksOn W
        WHERE W.ESSN = SSN AND W.PNo = P.PNumber) )
```

Requêtes SQL : modifications

Insertion d'une ligne

```
INSERT  
INTO Employee(SSN, Name)  
VALUES (9857234, 'John')
```

Modification d'une ligne

```
UPDATE Employee  
SET Name = 'Bill'  
WHERE SSN = 9857234
```

Suppression d'une ligne

```
DELETE FROM Employee  
WHERE SSN = 9857234
```

SQL Server dans la salle Platon

- ▶ Démarrer en Windows XP
- ▶ Utilisateur : “db2”
- ▶ Mot de passe : voir tableau
- ▶ Programmes - Microsoft SQL Server - Analyseur de requêtes
- ▶ Login “sa”, même mot de passe
- ▶ Ouvrir et exécuter le script SQL qui se trouve sur la page des TP