INFO-H-415
Labs 2 and 3

SQL Server Triggers (2)

SQL Server Triggers : Types

- Triggered directly after an instruction (i.e. not after each row or each transation)

- `AFTER` triggers : triggered after the instruction takes place
- `INSTEAD OF` trigger : does not execute the instruction, executes in place of it

SQL Server Triggers : Syntax

```
CREATE TRIGGER <name> ON <table>
{AFTER|INSTEAD OF} <list of events>
AS
<transact-SQL-statements>
```

- events : INSERT, DELETE, UPDATE

SQL Server Triggers : Tables

- Inside the `<transact-SQL-statements>:`
  - Tables **`INSERTED`** and **`DELETED`** can be used
  - Since the trigger is at the instruction level, these tables can contain many rows
- On `DELETE` :
  - `DELETED` contains the removed rows
- On `INSERT` :
  - `INSERTED` contains the new rows
- On `UPDATE` :
  - `DELETED` contains the rows before the modification
  - `INSERTED` contains the rows after the modification

Example

- **Employee**(Name, Salary, Department)

  Department references Department.DeptNo
- **Department**(DeptNo, Manager)

  Manager references Employee.Name

- The salary of an employee cannot be greater than that of his manager.

- When can it happen?

Example

- The salary of an employee cannot be greater than that of his manager.

- When can it happen?
  - When adding an employee
  - When modifying an employee salary
  - When modifying an employee department
  - When modifying the manager of a department

Example

Employee(Name, Salary, Department)
Department(DeptNo, Manager)

- When adding an employee:

```
CREATE TRIGGER salaryEmployee ON Employee
AFTER INSERT
AS
IF EXISTS(
  SELECT * FROM Inserted NewE, Department D, Employee Mgr
  WHERE NewE.Department = D.DeptNo and
        D.Manager = Mgr.Name and
        Mgr.Salary < NewE.Salary
)
BEGIN
  RAISERROR 13000 'The salary of an employee
   cannot be greater than that of his manager'
  ROLLBACK
END
```

Other constraints

CHECK, FOREIGN KEY, UNIQUE

CHECK

- **`CHECK`** is used to set a constraint on a **single row**.
- Example:
  - The salary of an employee must be grater than 1000 €.

Employee(Name, Salary, Department)

```
ALTER TABLE Employee
ADD CONSTRAINT employee_salary_1000
CHECK (Salary >= 1000)
```

FOREIGN KEY

- Add a foreign key constraint.
- Example:

Employee(Name, Salary, Department)
Department **references** Department.DeptNo

Department(DeptNo, Manager)

```
ALTER TABLE Employee
ADD CONSTRAINT FK_employee_dep
FOREIGN KEY (Department)
REFERENCES Department (DeptNo)
```

FOREIGN KEY

- Notice: the fields to which a foreign key refer must be **unique**:
  - A primary key
  - A (set of) fields under a uniqueness constraint: **`UNIQUE`**

```
ALTER TABLE <t_name>
ADD CONSTRAINT <c_name>
UNIQUE (<field_list>)
```

Date-related Functions

- `getdate()`

- `dateadd(`*`interval, n, date`*`)`
  - *`interval`* : year, month, day, …
  - Returns the **date** (date + (n*interval))

- `datediff(`*`interval, start, end`*`)`
  - Returns the numberof *`intervals`* between *`start`* `and` *`end`*

Dataset

- Available on the labs web page:
  - http://cs.ulb.ac.be/public/teaching/infoh415/tp
- Installation :
  - Create the '`triggers`' database (drop it if it already exists)
  - Open and run `createtable.sql`
  - Open and run `dbload.sql`
  - (Select the right database before running these scripts!)