INFO-H-415
Séance d'exercices 2 et 3

Triggers SQL Server (2)

Triggers SQL Server

- Se déclenche directement après une instruction (ni ligne, ni transaction)

- `AFTER` trigger : déclenche le trigger après l'instruction
- `INSTEAD OF` trigger : n'exécute pas l'instruction mais fait un traitement à la place

Rappel : Triggers SQL Server

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

- events : INSERT, DELETE, UPDATE

Triggers SQL Server

- Dans le `<transact-SQL-statements>` :
  - On a les tables **`INSERTED`** et **`DELETED`**
  - Attention, niveau instruction -> plusieurs lignes
- Dans le cas d'un `DELETE` :
  - `DELETED` contient les lignes supprimées
- Dans le cas d'un `INSERT` :
  - `INSERTED` contient les nouvelles lignes
- Dans le cas d'un `UPDATE` :
  - `DELETED` contient les lignes avant modification
  - `INSERTED` contient les lignes après modification

Exemple

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

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

  Manager references Employee.Name

- Le salaire d'un employé ne peut être supérieur à celui de son manager.

- Dans quels cas cela peut-il arriver ?

Exemple

- Le salaire d'un employé ne peut être supérieur à celui de son manager.

- Dans quels cas cela peut-il arriver ?
  - On ajoute un employé
  - On modifie le salaire d'un employé
  - On modifie le département d'un employé
  - On modifie le manager d'un département

Exemple

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

- On ajoute un employé

```
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 'Le salaire d'un employé ne peut être supérieur à celui
   de son manager'
  ROLLBACK
END
```

Autres types de contraintes

CHECK, FOREIGN KEY, UNIQUE

Contrainte `CHECK`

- **`CHECK`** permet de poser une condition sur les valeurs des champs **d'une ligne**.
- Exemple :
  - Le salaire d'un employé doit être supérieur à 1000 €.

Employee(Name, Salary, Department)

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

Contrainte `FOREIGN KEY`

- Contrainte de clés étrangères.
- Exemple :

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)
```

Contrainte `FOREIGN KEY`

- Attention, la référence d'une clé étrangère doit être **unique** :
  - Une clé primaire
  - Un champ ou un ensemble de champs sous une contrainte d'unicité : **`UNIQUE`**

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

Fonctions sur les dates

- `getdate()`

- `dateadd(`*`interval, nombre, date`*`)`
  - *`interval`* : year, month, day, …
  - Retourne une **date** (date+(nombre*interval))

- `datediff(`*`interval, start, end`*`)`
  - Retourne le **nombre** d'*`intervals`* entre *`start`* et *`end`*

Jeu de données

- Disponible sur la page web des tp :
  - http://cs.ulb.ac.be/public/teaching/infoh415/tp
- Installation :
  - Créer une base de données '`triggers`' (la supprimer si elle existe déjà)
  - Ouvrir et exécuter le script `createtable.sql`
  - Ouvrir et exécuter le script `dbload.sql`
  - (! Exécuter ces scripts dans le bon contexte)