

--Inainte de executarea acestui laborator, se executa comenzile:
--(crearea de obiecte se va face de catre profesor sau de student, acasa)

```
USE Northwind --sau North
GO
```

```
IF OBJECT_ID ('dbo.employees_n', 'U') IS NOT NULL
    DROP TABLE dbo. employees_n;
GO
```

```
--Create si populare cu date tabela Employees_n
SELECT * INTO Employees_n FROM Employees
```

```
select * from Employees_n
```

```
IF OBJECT_ID ('dbo.customers_n', 'U') IS NOT NULL
    DROP TABLE dbo. customers_n;
GO
```

```
--Create si populare cu date tabela customers_n
SELECT * INTO customers_n FROM customers
select * from customers_n
```

```
IF OBJECT_ID ('dbo.orders_n', 'U') IS NOT NULL
    DROP TABLE dbo. orders_n;
GO
```

```
--Create si populare cu date tabela orders_n
SELECT * INTO orders_n FROM orders
```

--Exercitiul 1

--Definirea constrangerilor DEFAULT

--In acest exercitiu veti adauga constrangeri DEFAULT la tabela Employees_n
--din baza de date Northwind .

--Definirea unei constrangeri DEFAULT

--In aceasta procedura, veti executa un script care creaza o constrangere DEFAULT pentru
-- coloana Regiondin tabela Employees_n si apoi veti adapta acelasi script pentru a
-- modifica aceasta constrangere.

--1. Logati-va la serverul utm-vac-ts1 si apoi conectati-va la sql server folosind consola
-- Sql Server Management Studio , cu Windows Authentication

--2. In lista Databases, click Northwind sau North.

--3. Revedeti si executati scriptul:

```
/*
Adaugati o constrangere DEFAULT "NY" pentru New York
coloanei Region din tabela Employees_n.
*/
```

```
/* Stergeti constrangerea daca exista deja */
```

```
IF OBJECT_ID('DF_Region_n') IS NOT NULL
    ALTER TABLE Employees_n
        DROP CONSTRAINT DF_Region_n
GO
```

```

/* Adaugati constrangerea */

ALTER TABLE Employees_n
ADD CONSTRAINT DF_Region_n DEFAULT 'NY' FOR Region
GO
-- Scriptul va crea constrangerea DEFAULT care adauga NY (New York) ca valoare implicita
-- pentru coloana Region din tabela Employees_n.

--4. Executati sp_helpconstraint tablenam si sp_help constraintname pentru a vedea
-- informatiile despre constrangerea DEFAULT creata.

EXEC SP_HELPCONSTRAINT Employees_n
EXEC SP_HELP DF_Region_n

--5. Modificati scriptul de mai sus pentru a inlocui constrangerea DEFAULT creata la
-- pasul 4 cu o constrangere care face valoarea implicita WA(Washington) pentru coloana
-- Region din tabela Employees_n:

ALTER TABLE Employees_n DROP CONSTRAINT DF_Region_n
GO
ALTER TABLE Employees_n
ADD CONSTRAINT DF_Region_n DEFAULT 'WA' FOR Region
GO
--Verificare inlocuire constrangere:
EXEC SP_HELPCONSTRAINT Employees_n
EXEC SP_HELP DF_Region_n

--Exercitiul 2
--Definirea constrangerilor CHECK
--In acest exercitiu veti adauga doua constrangeri CHECK la tabela Employees_n
-- din baza de date Northwind sau North.

--Definirea unei constrangeri CHECK
-- In aceasta procedura veti executa un script pentru a adauga o constrangere CHECK numita
-- titlu_de_politete tabeli Employees_n si apoi veti adauga o constrangere de tip CHECK
-- aceluiasi tabel pentru data nasterii.
-- La sfarsit se vor testa constrangerile CHECK create.
--1. Revedeti si executati scriptul:
/*
Adaugati constrangerea CHECK CK_titlu_de_politete tabeli Employees_n
Din baza de date Northwind. Titlurile de politete trebuie sa fie Ms., Mrs., Mr. sau Dr.
*/

USE Northwind
GO

/* Drop constrangere daca exista deja */

IF EXISTS
(SELECT * FROM INFORMATION_SCHEMA.CHECK_CONSTRAINTS
where CONSTRAINT_NAME = 'CK_titlu_de_politete')
ALTER TABLE Employees_n
DROP CONSTRAINT CK_titlu_de_politete
GO

/* Acum adaugati constrangerea CK_titlu_de_politete' */

```

```

ALTER TABLE Employees_n WITH NOCHECK
ADD CONSTRAINT CK_titlu_de_politete CHECK (TitleOfCourtesy
IN ('Ms.', 'Mrs.', 'Mr.', 'Dr. '))
GO
-- Verificare adaugare constrangere CHECK CK_titlu_de_politete:
EXEC SP_HELPCONSTRAINT Employees_n
GO
EXEC SP_HELP CK_titlu_de_politete
GO

```

```

--2. Scrieti si executati instructiunea care adauga o constrangere CHECK
-- coloanei BirthDate,
-- din tabela Employees, numita CK_data_nasterii.
-- Valoarea coloanei BirthDate sa fie mai mica decat data actuala.

```

```

ALTER TABLE Employees_n
ADD CONSTRAINT CK_data_nasterii
CHECK (BirthDate < GETDATE())
GO
--Verificare adaugare constrangere CHECK CK_data_nasterii:
EXEC SP_HELPCONSTRAINT Employees_n
GO
EXEC SP_HELP CK_data_nasterii
GO

```

```

--3. Executati instructiunile care violeaza fiecare constrangere:

```

```

UPDATE Employees_n SET TitleOfCourtesy = 'None'
WHERE EmployeeID = 1
GO

```

```

UPDATE Employees_n SET BirthDate = (GETDATE()+1)
WHERE EmployeeID = 1
GO

```

```

--Ce se intampla?

```

```

--In fiecare caz, comanda este intrerupta, cu mesajul :

```

```

--The UPDATE statement conflicted with the CHECK constraint "CK_titlu_de_politete",
-- respectiv
--The UPDATE statement conflicted with the CHECK constraint "CK_data_nasterii"

```

```

/*Exercitiul 3

```

```

Definirea constrangerilor PRIMARY KEY

```

```

In acest exercitiu veti adauga constrangeri PRIMARY KEY la tabela Employees_n si
Customers_n din baza de date Northwind sau North.

```

```

Definirea constrangerii PRIMARY KEY:

```

```

In aceasta procedura veti crea, mai intai, o cheie primara pe tabela Employees_n
si apoi veti crea o constrangere PRIMARY KEY pe tabela Customers_n.

```

```

*/

```

```

-- 1. Revedeti si executati scriptul:

```

```

/*Adaugati o constrangere primary key la tabela Employees_n.
Setati NOCOUNT on pt. a nu afisa mesajul cu nr. de linii afectate.
*/

```

```

SET NOCOUNT ON

```

GO

/* Tabela Employees_n */

```
IF EXISTS
  (SELECT * FROM INFORMATION_SCHEMA.TABLE_CONSTRAINTS
   where CONSTRAINT_SCHEMA = 'dbo'
        AND CONSTRAINT_NAME = 'PK_Employees_n'
        AND CONSTRAINT_TYPE = 'PRIMARY KEY')
ALTER TABLE Employees_n
  DROP CONSTRAINT PK_Employees_n
```

GO

```
ALTER TABLE Employees_n
  ADD CONSTRAINT PK_Employees_n PRIMARY KEY NONCLUSTERED (EmployeeID)
GO
```

/* Resetare optiune NOCOUNT*/

```
SET NOCOUNT OFF
GO
```

-- 2. Scrieti si executati instructiunea care adauga constrangerea PRIMARY KEY numita
-- PK_Customers_n pe coloana CustomerID din tabela Customers_n:

```
ALTER TABLE Customers_n
ADD CONSTRAINT PK_Customers_n PRIMARY KEY CLUSTERED(CustomerID)
GO
```

/* Care este impactul indecsilor noncluster asociati cu tabela, cand creati
o constrangere PRIMARY KEY ?
O constrangere PRIMARY KEY creaza automat un index clustered si reconstruieste automat
toti indecsii noncluster(daca nu exista deja un index cluster).
*/

--3. Executati procedura stocata sistem sp_ HELPCONSTRAINT nume_tabela si
-- SP_HELP nume_constr_PK:

```
EXEC SP_HELPCONSTRAINT customers_n
GO
EXEC SP_HELP PK_Employees_n
GO
```

-- Exercitiul 4

/*Definirea constrangerilor FOREIGN KEY

In acest exercitiu veti adauga constrangeri FOREIGN KEY tabellei Orders_n din baza de
date

Northwind sau North.

Definirea constrangerii FOREIGN KEY:

In aceasta procedura veti scrie si executa un script care creaza un foreign key pe tabela
Orders_n.

*/

--1. Revedeti si executati scriptul care creaza o constrangere FOREIGN KEY pe tabela
Orders_n:

```
SET NOCOUNT ON
GO
```

IF EXISTS

```

    (SELECT * FROM INFORMATION_SCHEMA.TABLE_CONSTRAINTS
     WHERE CONSTRAINT_SCHEMA = 'dbo' AND CONSTRAINT_NAME = 'FK_Orders_Customers_n'
                                     AND CONSTRAINT_TYPE = 'FOREIGN KEY')
ALTER TABLE dbo.Orders_n DROP CONSTRAINT FK_Orders_Customers_n
GO

ALTER TABLE dbo.Orders_n WITH NOCHECK
    ADD CONSTRAINT FK_Orders_Customers_n
        FOREIGN KEY(CustomerID) REFERENCES dbo.Customers_n(CustomerID)
GO

/* Reset NOCOUNT */
SET NOCOUNT OFF
GO

EXEC sp_helpconstraint Orders_n --verificare existenta constrangeri pe tabela Orders_n

-- De ce nu este necesar sa se execute instructiunea DROP INDEX?
-- Crearea unui FOREIGN KEY nu creaza automat si un index.

-- Exerciitiul 5
-- (daca timpul permite)
/* Crearea obiectelor Defaults si Rules
In acest exercitiu veti adauga obiecte defaults si rules la baza de date Northwind.
*/
-- Crearea unui obiect de tip default
-- 1. Revedeti si executati scriptul:

/* Daca obiectul exista il sterge */

IF OBJECT_ID('DF_Country') IS NOT NULL
BEGIN
    EXEC sp_unbindefault 'dbo.Suppliers.Country'
    DROP DEFAULT DF_Country
END
GO

/* Recreare obiect default si legare la o coloana de tabel */

CREATE DEFAULT DF_Country AS 'Singapore'
GO
EXEC sp_bindefault DF_Country, 'dbo.Suppliers.Country'
GO
/*Acest script creaza si leaga un obiect default la coloana Suppliers.Country.
Valoarea default este Singapore.
*/
--2. Executati o instructiune care insereaza un articol in tabela Suppliers, pt. a
-- verifica faptul ca obiectul default DF_Country lucreaza adecvat :

INSERT Suppliers (CompanyName) VALUES ('Compania mea')
GO
--verificati rezultatul:

SELECT * FROM Suppliers
WHERE Country = 'Singapore'
GO

```

```

-- Crearea unui obiect de tip rule
-- 1. Revedeti si executati scriptul:
IF OBJECT_ID('R_PhotoPath') IS NOT NULL
BEGIN
    EXEC sp_unbindrule 'dbo.Employees.PhotoPath'
    DROP RULE R_PhotoPath
END
GO

-- Creati si legati obiectul Rule.
CREATE RULE R_PhotoPath
    AS @PhotoPath LIKE 'http://accweb/emmployees/%'
GO
EXEC sp_bindrule R_PhotoPath, 'dbo.Employees.PhotoPath'
GO

--2. Executati urmatoarea instructiune UPDATE pentru a testa regula,
-- incercand sa updatati coloana PhotoPath cu o cale de fisier invalida.
-- Instructiunea va esua , deoarece violeaza regula .

UPDATE Employees
SET PhotoPath = 'http://accweb/xemmployees/fuller.bmp'
WHERE LastName = 'Fuller'
GO

--afisarea numelor coloanelor si ale regulilor la care sunt legate
-- pt. o tabela(aici Employees):

SELECT name [Nume coloana],rule_object_id,OBJECT_NAME (rule_object_id) [Nume regula]
FROM sys.columns
WHERE object_id = object_id('Employees') AND rule_object_id > 0

```