



**МІНІСТЕРСТВО ОСВІТИ, НАУКИ, МОЛОДІ ТА СПОРТУ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»
ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ**

Лабораторна робота №8

**Інформаційно-комунікаційні системи:
Бази даних**

Варіант № 4

Підготував:

студент 4 курсу

групи ФІ-84

Коломієць Андрій Юрійович

E-mail: *andkol-ipt22@lil.kpi.ua*

СТВОРЕННЯ І ВИКОРИСТАННЯ ЗБЕРЕЖЕНИХ ПРОЦЕДУР

Завдання

Предметна область:

Навчально-методичне управління (професорсько-викладацький склад).

Основні предметно-значущі сутності:

Співробітники, Підрозділи, Дисципліни.

Основні предметно-значущі атрибути сутностей:

- Співробітники - прізвище, ім'я, по батькові, стать, дата народження, адреса прописки, посада, підрозділ;
- Підрозділи - назва, вид підрозділу;
- Дисципліни — назва.

Основні вимоги до функцій системи:

- Вибрати дисципліни, що читаються співробітниками або певним співробітником;
- Вибрати список співробітників по підрозділам або певного підрозділу;
- Вибрати дисципліни, що читаються співробітниками по підрозділам або певного підрозділу.

Тригери:

1. На додавання запису в таблицю «Працівники». Якщо в таблиці вже існує запис про співробітника з збігаються предметно-значущими атрибутами, заборонити додавання нового запису.
2. Створити представлення «Дисципліни» з полями «Код_сотрудніка», «ФІО_сотрудніка», «Дисципліна». Оновлювати представлення «Дисципліни».

Процедура:

Процедура повинна повертати кількість дисциплін, що читаються кожним співробітником зазначеного підрозділу.

Необхідно додатково

Створити тригери, які вказані у вашому завданні.

Виконання завдання

Microsoft SQL

```
USE Lab_7
```

```
DROP TABLE IF EXISTS MAIN, EMPLOYEE,DEGREE,DEPARTMENT,SUBJECTS;
```

```
CREATE TABLE DEGREE
```

```
(  
    ID_DEGREE INT IDENTITY PRIMARY KEY NOT NULL,  
    NAME_DEGREE VARCHAR (20) NOT NULL,  
);
```

```
CREATE TABLE SUBJECTS
```

```
(  
    ID_SUBJECT INT IDENTITY PRIMARY KEY NOT NULL,  
    NAME_SUBJECT VARCHAR (20) NOT NULL,  
);
```

```
CREATE TABLE DEPARTMENT
```

```
(  
    ID_DEPARTMENT INT IDENTITY PRIMARY KEY NOT NULL,  
    NAME_DEPARTMENT VARCHAR (20) NOT NULL,  
);
```

```
CREATE TABLE EMPLOYEE
```

```
(  
    ID_EMPLOYEE INT IDENTITY PRIMARY KEY NOT NULL,  
    NAME_EMPLOYEE VARCHAR (20) NOT NULL,  
    SURNAME_EMPLOYEE VARCHAR (20) NOT NULL,  
    FATHER_NAME_EMPLOYEE VARCHAR (20) NOT NULL,  
    AGE_EMPLOYEE INT,  
    ADDRESS_EMPLOYEE VARCHAR (25),  
);
```

```
CREATE TABLE MAIN
```

```
(  
    EMPLOYEE_NAME INT,  
    EMPLOYEE_DEGREE INT,  
    EMPLOYEE_DEPARTMENT INT,  
    EMPLOYEE_SUBJECT INT,  
    FOREIGN KEY(EMPLOYEE_NAME) REFERENCES EMPLOYEE(ID_EMPLOYEE) ON DELETE CASCADE,  
    FOREIGN KEY(EMPLOYEE_DEGREE) REFERENCES DEGREE(ID_DEGREE) ON DELETE CASCADE,  
    FOREIGN KEY(EMPLOYEE_DEPARTMENT) REFERENCES DEPARTMENT(ID_DEPARTMENT) ON DELETE CASCADE,  
    FOREIGN KEY(EMPLOYEE_SUBJECT) REFERENCES SUBJECTS(ID_SUBJECT) ON DELETE CASCADE  
);
```

```
INSERT INTO DEGREE VALUES
```

```
('DEGREE_1'),  
( 'DEGREE_2'),  
( 'DEGREE_3'),  
( 'DEGREE_4'),  
( 'DEGREE_5'),  
( 'DEGREE_6'),  
( 'DEGREE_7'),  
( 'DEGREE_8'),  
( 'DEGREE_9'),  
( 'DEGREE_10');
```

```
INSERT INTO DEPARTMENT VALUES
```

```
(' DEPARTMENT_1'),  
( ' DEPARTMENT_2'),  
( ' DEPARTMENT_3'),  
( ' DEPARTMENT_4'),  
( ' DEPARTMENT_5'),  
( ' DEPARTMENT_6'),  
( ' DEPARTMENT_7'),  
( ' DEPARTMENT_8'),  
( ' DEPARTMENT_9'),  
( ' DEPARTMENT_10');
```

```
INSERT INTO SUBJECTS VALUES
```

```
('SUBJECTS_1'),  
( 'SUBJECTS_2'),  
( 'SUBJECTS_3'),  
( 'SUBJECTS_4'),  
( 'SUBJECTS_5'),  
( 'SUBJECTS_6'),  
( 'SUBJECTS_7'),  
( 'SUBJECTS_8'),  
( 'SUBJECTS_9'),  
( 'SUBJECTS_10');
```

```
INSERT INTO EMPLOYEE VALUES
```

```
('Name_1','Surname_1','Father_Name_1','20','Street_1'),  
( 'Name_2','Surname_2','Father_Name_2','66','Street_2'),  
( 'Name_3','Surname_3','Father_Name_3','25','Street_3'),  
( 'Name_4','Surname_4','Father_Name_4','34','Street_4'),  
( 'Name_5','Surname_5','Father_Name_5','61','Street_5'),  
( 'Name_6','Surname_6','Father_Name_6','27','Street_6'),  
( 'Name_7','Surname_7','Father_Name_7','60','Street_7'),  
( 'Name_8','Surname_8','Father_Name_8','46','Street_8'),  
( 'Name_9','Surname_9','Father_Name_9','35','Street_9'),  
( 'Name_10','Surname_10','Father_Name_10','24','Street_10');
```

```

INSERT INTO MAIN VALUES
(1,2,3,1),
(2,3,2,5),
(2,3,3,4),
(3,2,1,2),
(4,3,5,1),
(5,2,3,2),
(5,2,3,3),
(5,2,3,7),
(6,3,3,10),
(7,7,3,1),
(8,8,3,3),
(8,8,3,2),
(8,8,3,4),
(8,8,3,8),
(9,2,1,6),
(10,10,2,4);

GO

CREATE TRIGGER NOT_REPEAT
ON EMPLOYEE
INSTEAD OF INSERT
AS
BEGIN

IF EXISTS
(
    SELECT * FROM inserted AS NEW
    INNER JOIN EMPLOYEE ON EMPLOYEE.NAME_EMPLOYEE=NEW.NAME_EMPLOYEE AND
                           EMPLOYEE.SURNAME_EMPLOYEE=NEW.SURNAME_EMPLOYEE AND
                           EMPLOYEE.FATHER_NAME_EMPLOYEE=NEW.FATHER_NAME_EMPLOYEE AND
                           EMPLOYEE.AGE_EMPLOYEE=NEW.AGE_EMPLOYEE
)
    BEGIN
        RAISERROR('DATA ABOUT EMPLOYEE ALREADY EXISTS!',16,1);
        ROLLBACK TRAN;
    END
ELSE
    BEGIN
        DECLARE @NAME_EMPLOYEE VARCHAR(20)
        DECLARE @SURNAME_EMPLOYEE VARCHAR (20)
        DECLARE @FATHER_NAME_EMPLOYEE VARCHAR(20)
        DECLARE @AGE_EMPLOYEE INT
        DECLARE @ADDRESS_EMPLOYEE VARCHAR(25)

        SELECT @NAME_EMPLOYEE=inserted.NAME_EMPLOYEE FROM inserted
        SELECT @SURNAME_EMPLOYEE=inserted.SURNAME_EMPLOYEE FROM inserted
        SELECT @FATHER_NAME_EMPLOYEE=inserted.FATHER_NAME_EMPLOYEE FROM inserted
        SELECT @AGE_EMPLOYEE=inserted.AGE_EMPLOYEE FROM inserted
        SELECT @ADDRESS_EMPLOYEE=inserted.ADDRESS_EMPLOYEE FROM inserted
    END
END

```

	ID_EMPLOYEE	NAME_EMPLOYEE	SURNAME_EMPLOYEE	FATHER_NAME_EMPLOYEE	AGE_EMPLOYEE	ADDRESS_EMPLOYEE
1	1	Name_1	Surname_1	Father_Name_1	20	Street_1
2	2	Name_2	Surname_2	Father_Name_2	66	Street_2
3	3	Name_3	Surname_3	Father_Name_3	25	Street_3
4	4	Name_4	Surname_4	Father_Name_4	34	Street_4
5	5	Name_5	Surname_5	Father_Name_5	61	Street_5
6	6	Name_6	Surname_6	Father_Name_6	27	Street_6
7	7	Name_7	Surname_7	Father_Name_7	60	Street_7
8	8	Name_8	Surname_8	Father_Name_8	46	Street_8
9	9	Name_9	Surname_9	Father_Name_9	35	Street_9
10	10	Name_10	Surname_10	Father_Name_10	24	Street_10
11	11	Name_new	Surname_new	Father_Name_new	32	Street_new

```

(10 rows affected)
Msg 50000, Level 16, State 1, Procedure NOT_REPEAT, Line 14 [Batch Start Line 151]
DATA ABOUT EMPLOYEE ALREADY EXISTS!

```

```
INSERT INTO EMPLOYEE(NAME_EMPLOYEE, SURNAME_EMPLOYEE, FATHER_NAME_EMPLOYEE, AGE_EMPLOYEE, ADDRESS_EMPLOYEE) VALUES
(@NAME_EMPLOYEE, @SURNAME_EMPLOYEE, @FATHER_NAME_EMPLOYEE, @AGE_EMPLOYEE, @ADDRESS_EMPLOYEE);
```

```
END
```

```
END
```

```
GO
```

```
INSERT INTO EMPLOYEE(NAME_EMPLOYEE, SURNAME_EMPLOYEE, FATHER_NAME_EMPLOYEE, AGE_EMPLOYEE, ADDRESS_EMPLOYEE) VALUES
('Name_1', 'Surname_1', 'Father_Name_1', '20', 'Street_1');
```

```
GO
```

```
INSERT INTO EMPLOYEE(NAME_EMPLOYEE, SURNAME_EMPLOYEE, FATHER_NAME_EMPLOYEE, AGE_EMPLOYEE, ADDRESS_EMPLOYEE) VALUES
('Name_new', 'Surname_new', 'Father_Name_new', '32', 'Street_new');
```

```
GO
```

```
SELECT * FROM EMPLOYEE
```

```
GO
```

```
DROP VIEW IF EXISTS SUBJECT_VIEW;
```

```
GO
```

```
CREATE VIEW SUBJECT_VIEW
```

```
AS
```

```
SELECT ID_EMPLOYEE, NAME_EMPLOYEE, SURNAME_EMPLOYEE, NAME_SUBJECT
```

```
FROM EMPLOYEE, SUBJECTS, MAIN
```

```
WHERE EMPLOYEE.ID_EMPLOYEE=MAIN.EMPLOYEE_NAME AND SUBJECTS.ID_SUBJECT = MAIN.EMPLOYEE_SUBJECT
```

```
GO
```

```
CREATE TRIGGER UPDATING
```

```
ON MAIN
```

```
FOR INSERT, UPDATE, DELETE
```

```
AS
```

```
SELECT * FROM SUBJECT_VIEW
```

```
GO
```

```
SELECT * FROM SUBJECT_VIEW
```

```
INSERT INTO MAIN VALUES
```

```
(11,1,4,1);
```

	ID_EMPLOYEE	NAME_EMPLOYEE	SURNAME_EMPLOYEE	NAME_SUBJECT
4	4	Name_4	Surname_4	SUBJECTS_1
5	5	Name_5	Surname_5	SUBJECTS_7
6	6	Name_6	Surname_6	SUBJECTS_10
7	7	Name_7	Surname_7	SUBJECTS_1
8	8	Name_8	Surname_8	SUBJECTS_8
9	9	Name_9	Surname_9	SUBJECTS_6
10	10	Name_10	Surname_10	SUBJECTS_4
11	11	Name_new	Surname_new	SUBJECTS_1