--- 14 лаба

--- 1 задание

CREATE TABLE TR\_AUDIT

(

ID int identity,

STMT varchar(20) check (STMT in ('INS','DEL','UPD')),

TRNAME varchar(50),

CC varchar(300)

);

GO

CREATE TRIGGER TR\_TEACHERS\_INS on TEACHER after INSERT AS

insert into TR\_AUDIT values (

'INS',

'TR\_TEACHER\_INS',

'Преподаватель:' + rtrim((SELECT TEACHER from INSERTED)) +

'. Имя:' + rtrim((SELECT TEACHER\_NAME from INSERTED)) +

'. Пол:' + rtrim((SELECT GENDER from INSERTED)) +

'. Кафедра: ' + rtrim((SELECT PULPIT from INSERTED))

);

INSERT into TEACHER(TEACHER,TEACHER\_NAME,GENDER,PULPIT)

VALUES ('test','test','м','ИСиТ');

SELECT \* from TR\_AUDIT;

--- 2 задание

GO

CREATE TRIGGER TR\_TEACHER\_DEL on TEACHER after DELETE AS

insert into TR\_AUDIT values (

'DEL',

'TR\_TEACHER\_DEL',

'Преподаватель:' + rtrim((SELECT TEACHER from DELETED)) +

'. Имя:' + rtrim((SELECT TEACHER\_NAME from DELETED)) +

'. Пол:' + rtrim((SELECT GENDER from DELETED)) +

'. Кафедра: ' + rtrim((SELECT PULPIT from DELETED))

);

GO

DELETE FROM TEACHER where TEACHER = 'test';

SELECT \* from TR\_AUDIT;

--- 3 задание

GO

CREATE TRIGGER TR\_TEACHER\_UPD on TEACHER after UPDATE AS

insert into TR\_AUDIT values (

'UPD',

'TR\_TEACHER\_UPD',

'Преподаватель:' + rtrim((SELECT TEACHER from INSERTED)) +

'. Имя:' + rtrim((SELECT TEACHER\_NAME from INSERTED)) +

'. Пол:' + rtrim((SELECT GENDER from INSERTED)) +

'. Кафедра: ' + rtrim((SELECT PULPIT from INSERTED))

);

GO

UPDATE TEACHER SET PULPIT = 'ТЛ' where TEACHER = 'РЖК';

SELECT \* from TR\_AUDIT;

--- 4 задание

GO

CREATE TRIGGER TR\_TEACHER on TEACHER after INSERT, DELETE, UPDATE AS

DECLARE @upd int = (select count(\*) from inserted),

@del int = (select count(\*) from deleted);

if @upd > 0 and @del = 0

BEGIN

insert into TR\_AUDIT values (

'INS',

'TR\_TEACHER',

'Преподаватель:' + rtrim((SELECT TEACHER from INSERTED)) +

'. Имя:' + rtrim((SELECT TEACHER\_NAME from INSERTED)) +

'. Пол:' + rtrim((SELECT GENDER from INSERTED)) +

'. Кафедра: ' + rtrim((SELECT PULPIT from INSERTED))

);

END

else if @upd = 0 and @del > 0

BEGIN

insert into TR\_AUDIT values (

'DEL',

'TR\_TEACHER',

'Преподаватель:' + rtrim((SELECT TEACHER from DELETED)) +

'. Имя:' + rtrim((SELECT TEACHER\_NAME from DELETED)) +

'. Пол:' + rtrim((SELECT GENDER from DELETED)) +

'. Кафедра: ' + rtrim((SELECT PULPIT from DELETED))

);

END

else if @upd > 0 and @del > 0

BEGIN

insert into TR\_AUDIT values (

'UPD',

'TR\_TEACHER',

'Преподаватель:' + rtrim((SELECT TEACHER from INSERTED)) +

'. Имя:' + rtrim((SELECT TEACHER\_NAME from INSERTED)) +

'. Пол:' + rtrim((SELECT GENDER from INSERTED)) +

'. Кафедра: ' + rtrim((SELECT PULPIT from INSERTED))

);

END

return;

GO

INSERT into TEACHER(TEACHER,TEACHER\_NAME,GENDER,PULPIT) VALUES

('test','test','м','ИСиТ');

DELETE FROM TEACHER where TEACHER = 'test';

UPDATE TEACHER SET PULPIT = 'ЛВ' where TEACHER = 'РЖК';

SELECT \* from TR\_AUDIT;

--- 6 задание

drop trigger TR\_TEACHER\_DEL2;

GO

CREATE TRIGGER TR\_TEACHER\_DEL1 on TEACHER after DELETE AS

insert into TR\_AUDIT values (

'DEL',

'TR\_TEACHER\_DEL1',

'Преподаватель:' + rtrim((SELECT TEACHER from DELETED)) +

'. Имя:' + rtrim((SELECT TEACHER\_NAME from DELETED)) +

'. Пол:' + rtrim((SELECT GENDER from DELETED)) +

'. Кафедра: ' + rtrim((SELECT PULPIT from DELETED))

);

GO

GO

CREATE TRIGGER TR\_TEACHER\_DEL2 on TEACHER after DELETE AS

insert into TR\_AUDIT values (

'DEL',

'TR\_TEACHER\_DEL2',

'Преподаватель:' + rtrim((SELECT TEACHER from DELETED)) +

'. Имя:' + rtrim((SELECT TEACHER\_NAME from DELETED)) +

'. Пол:' + rtrim((SELECT GENDER from DELETED)) +

'. Кафедра: ' + rtrim((SELECT PULPIT from DELETED))

);

GO

GO

CREATE TRIGGER TR\_TEACHER\_DEL3 on TEACHER after DELETE AS

insert into TR\_AUDIT values (

'DEL',

'TR\_TEACHER\_DEL3',

'Преподаватель:' + rtrim((SELECT TEACHER from DELETED)) +

'. Имя:' + rtrim((SELECT TEACHER\_NAME from DELETED)) +

'. Пол:' + rtrim((SELECT GENDER from DELETED)) +

'. Кафедра: ' + rtrim((SELECT PULPIT from DELETED))

);

GO

exec SP\_SETTRIGGERORDER @triggername = 'TEACHER.TR\_TEACHER\_DEL3',

@order = 'First', @stmttype = 'DELETE';

exec SP\_SETTRIGGERORDER @triggername = 'TEACHER.TR\_TEACHER\_DEL2',

@order = 'Last', @stmttype = 'DELETE';

select t.name, e.type\_desc

from sys.triggers t

join sys.trigger\_events e on t.object\_id = e.object\_id

where OBJECT\_NAME(t.parent\_id) = 'TEACHER' and e.type\_desc = 'DELETE';

DELETE FROM TEACHER where TEACHER = 'test';

SELECT \* from TR\_AUDIT;

--- 8 задание

use UNIVER;

set nocount on;

GO

create trigger FACULTY\_INSTEAD\_OF on FACULTY instead of DELETE AS

PRINT 'Instead of delete';

return;

delete from FACULTY where FACULTY ='ИТ';

--- 9 задание

GO

create trigger DDL\_UNIVER on database for DDL\_DATABASE\_LEVEL\_EVENTS AS

declare @t varchar(50) = EVENTDATA().value('(/EVENT\_INS-TANCE/EventType)[1]', 'varchar(50)');

declare @t1 varchar(50) = EVENTDATA().value('(/EVENT\_INS-TANCE/ObjectName)[1]', 'varchar(50)');

declare @t2 varchar(50) = EVENTDATA().value('(/EVENT\_INS-TANCE/ObjectType)[1]', 'varchar(50)');

print 'Тип события: '+@t;

print 'Имя объекта: '+@t1;

print 'Тип объекта: '+@t2;

raiserror( N'Операции с таблицами запрещены ', 16, 1);

rollback;

CREATE TABLE BUBUB (

ID int identity(1,1)

)