

ნანტი - 1

Time left 1:15:24

--შექმენით ტრიგერი, რომელიც პროდუქტების ცხრილში ყოველი ახალი პროდუქტის წაშლისას გამოიტანს შეტყობინებას პროდუქტის წაშლის შესახებ N'სისტემიდან წაიშალა პროდუქტი

Create	Trigger
DeleteProductTrigger	
ON	Product
Delete	For

AS  
BEGIN  
Print N'sistemidan waishala...!  
END

N'სისტემიდან წაიშალა პროდუქტი	Insert
	Trigger
	shop
	Print
Create	
Update	
For	

Next page

page



რომელი ჯგუფის ბრძანებები გამოიყენება მონაცემთა ბაზაში ჩანაწერების განახლებისათვის?

- ☐ a. DLS
- ☐ b. DCL
- ☒ c. DML
- ☐ d. DDL

DML (INSERT, UPDATE, DELETE)

DDL => CREATE TABLE, DROP TABLE, ALTER TABLE.

ჩამოთვლილთაგან, რომელია არასწორი მტკიცებულება?

- ☐ a. ფუნქციაში არ ხდება DML გამოყენება
- ☒ b. პროცედურის გამოძახება შესაძლებელია SELECT-ში
- ☐ c. პროცედურაში შესაძლებელია DML ბრძანებების გამოყენება
- ☐ d. ფუნქცია შესაძლებელია გამოძახებული იქნეს SELECT წინადადებაში



n 2

ed  
out of

n

შექმენით პროცედურა (AddShop) output პარამეტრით (@Error varchar(15)), რომელიც დაამატებს ახალ მაღაზიას და თუ სისტემაში უკვე არსებობს მაღაზია კონკრეტული დასახელებით ან ნომრით დააბრუნებს შეცდომას output პარამეტრში: Time left 0:46:44

CREATE PROCEDURE AddShop

@shop\_id int,

@shop\_name varchar (15),

@Error varchar(15) Output

AS

IF EXISTS(

BEGIN

END

ELSE

BEGIN

END

1. SELECT shop\_id FROM shop WHERE shop\_name=@shop\_name OR shop\_id=@shop\_id

DELETE FROM shop WHERE shop\_id=@shop\_id

2. SET @Error='Duplicate shop'

SET @Error='Invalid shop.'

UPDATE shop SET shop\_name=@shop\_name WHERE shop\_id=@shop\_id

SELECT shop\_id FROM shop WHERE shop\_name=@shop\_name AND shop\_id!=@shop\_id

SELECT shop\_id FROM shop WHERE shop\_id=@shop\_id

3. INSERT INTO shop VALUES (@shop\_id,@shop\_name)

Time left 1:05:16

ჩამოთვლილთაგან, რომელია არასწორი?

- ☒ a. ფუნქციაში შესაძლებელია გამოყენებულ იქნეს Try-catch ბლოკი
- ☐ b. პროცედურაში ყოველთვის შესაძლებელია DML ბრძანებების გამოყენება.
- ☐ c. პროცედურა შესაძლებელია გამოიყენებოდეს ფუნქციით.
- ☐ d. View შეიცავს მონაცემებს ერთი ან რამდენიმე ცხრილიდან.

Time left 1:05:07

in 6

ed

d out of

g question

shop ცხრილში გააკეთეთ ჩანაწერი (108, 'shop8'); და შედეგში დააბრუნეთ აბლა  
შეტანილი ჩანაწერის სტრიქონი OUTPUT.

DECLARE @Shop TABLE (shop\_id INT, shop\_name VARCHAR(15))

INSERT INTO shop  
OUTPUT inserted shop\_id, inserted shop\_name INTO @

shop  
VALUES (108, 'shop8')

SELECT \* FROM @ shop Product Deleted shop\_name

@ inserted shop

shop\_id

Next page



შექმენით პროცედურა DeleteProduct, რომელიც პროდუქციის კოდის გადაცემის შემთხვევაში Product ცხრილიდან წაშლის პროდუქციას.

CREATE PROCEDURE **DeleteProduct**

@ **pr\_id** **int**, -- პროდუქციის კოდი

AS

BEGIN

**Delete** **from** **product**  
**where** **pr\_id** = @ **pr\_id**

END

**Delete** **From** **pr\_id** **Set** **DeleteProduct**  
**@** **Where** **product** **int**

Please put an answer in each box.

--შექმენით მაღაზიაში ახალი პროდუქტის შეტანის პროცედურა

AddNewProductToShop რომელსაც გადაეცემა ( pr\_id,Shop\_id, pr\_quantity) და Time left 0:41:12

პროდუქტს დაამატებს მაღაზიაში არსებული პროდუქტების სიაში. (პროდუქტი აქამდე არ ჰქონდა მაღაზიას).

CREATE PROCEDURE

AddNewProductToShop

@

pr\_id

int

-- პროდუქტის კოდი

@

shop\_id

int

--მაღაზიის ნომერი

@

pr\_quantity

int

--რაოდენობა

AS

BEGIN

insert into

pr\_shop

pr\_id

-- პროდუქტის კოდი

shop\_id

--მაღაზიის ნომერი

pr\_quantity

--რაოდენობა

values

@

pr\_id

@

shop\_id

@

pr\_quantity

END

pr\_shop

@

pr\_quantity

pr\_id

AddNewProductToShop

int

shop\_id

Where

Next page



Time left 1:23:23

დაწერეთ მოთხოვნა რომელიც დააბრუნებს იმ მაღაზიების დასახელებებს რომლებსაც აქვთ ერთი მაინც პროდუქტი შეტანილი.

```
SELECT shop_name
FROM shop
WHERE shop_id in
(select shop_id FROM pr_shop )
shop_name NOT IN IN shop_id shop pr_shop
```

Finish attempt ...

შექმენათ სკალარული ფუნქცია GetMinPrice რომელიც დააბრუნებს საწყობში არსებული ყველაზე დაბალ ფასს @MinPrice

```
CREATE FUNCTION GetMinPrice()
returns decimal(5,2)
AS
BEGIN
DECLARE @minPrice int
SET @MinPrice =(SELECT Min (Unit_Price ) FROM product)
return @MinPrice
END
Min @MinPrice Decimal(5,2) RETURNS GetMinPrice()
RETURN Unit_price
```

Next page