1. ცხრილი არის სვეტების და სტრიქონების ერთობლიობა რომელსაც ვადგენთ ამოცანის შესაბამისად მას შეიძლება გონდეს პირველადი ან მეორეული გასაგები რომლიოთაც დაუკავშირდება სხვა ცხრილს . ცხრილები მონაცემთა ბაზაში აუცილებლად უნდა იყოს ერთმანეთთან დაკავშირებული

2.

1)ერთი-ერთთან

2) ერთი-ბევრთან

3)ბევრი-ბევრთან

4) განმარტეთ მარტივი გასაღები და შედგენილი გასაღები.

3.

Simply key- მარტივი გასაღები ხშირად ასოცირდება როგორც მთავარი გასარები და ეს არის ერთი სვეტი ან სვეტების ერთობლიობა მონაცემთა ბაზის ცხრილში, რომელიც ცალსახად განსაზღვრავს ამ ცხრილის თითოეულ სტრიქონს ან ჩანაწერს.

Compose KEY -შედგენილი გასაღები ეწოდება გასაღებს რომელიც შედგება ორი ან მეტი სვეტისგან, რათა ცალსახად იდენტიფიცირდეს ჩანაწერის ცხრილში.

4. ნორმალიზაცია ეწოდება პროცესს, რომელიც ახდენს მონაცემთა ბაზის სტრუქტურის ოპტიმიზაციას. ნორმალური ფორმები წარმოადგენენ ამ პროცესის ეტაპებს.

5.

View- ხედვა ჩვენ შეგვიძლია გამოვიდზახოთ ვირტუალური ცხრილი

6.

tempDB-ის შექმნისას იქმნება დროებითი მონაცემთა ბაზა. რომელშიც შეგვიძლია შევჩვალოთ რაგაცეები. ის იქმნება მონაცემთა ბაზის შექმნისას და ასევე როგორც კი გამოვრთავთ იშლება (დროებითი მონაცემთა ბაზაა)

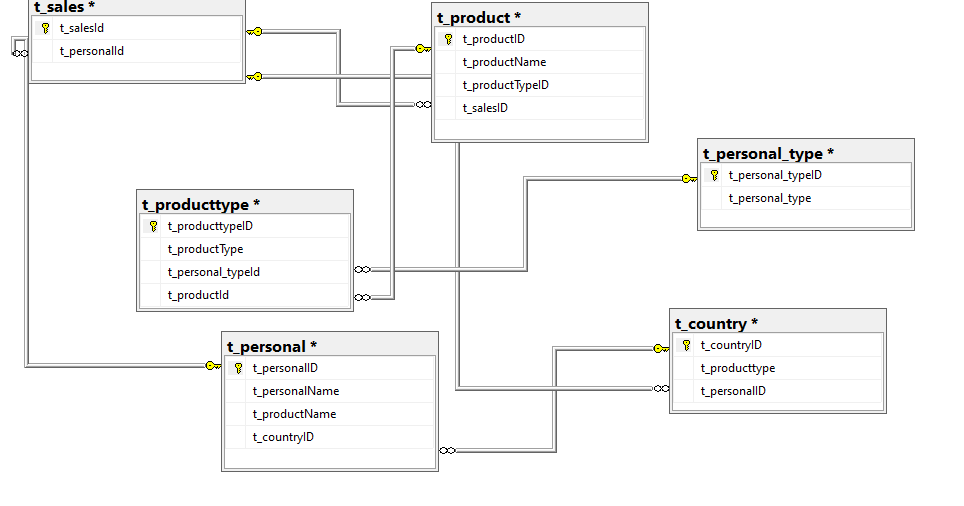
1. master, model, msdb, and tempdb

9.

ტრიგერი წარმოადგენს სპეციალური სახის შენახულ პროცედურას, რომელიც სრულდება, როცა ადგილი აქვს გარკვეულ ცვლილებებს. არსებობს ორი სახის ტრიგერი: **DML** ტრიგერი და **DDL** ტრიგერი

პრაქტიკული ნაწილი

N2



პროგრამული დავალება

N2

select \* V-Info\_filter

where type like '%saweri kalami%'or like '%fanqari%'

group by sum (price)>=150

N8

(A3 +B2 )-(A2+B2 )

create function F\_math

(

@A int,

@b int

)

returns int

Begin

As

@result int

set @result=((A\*A\*A)+(B\*B))-(A\*A)+(B\*B))

return @result;

end;

N9

CREATE FUNCTION f\_sales()

RETURNS TABLE

AS

RETURN (

SELECT product\_name, product\_type, price

FROM products

WHERE product\_type = 'Pencil' AND price < 1.5

);

SELECT \* FROM f\_sales();

N10

select \* t\_personal

where type like '%menejer%' or like'%directoti%'

drop t\_personal

CREATE TRIGGER tr\_check\_type

ON t\_personal

AFTER INSERT, UPDATE

AS

BEGIN

SET NOCOUNT ON;

IF EXISTS (

SELECT 1

FROM inserted

WHERE type IN ('manager', 'director')

)

BEGIN

RAISEERROR('Invalid type. Type must not be "manager" or "director".', 16, 1);

ROLLBACK;

END

END;

N6

SELECT product\_id, product\_name, product\_type, price

FROM products

WHERE product\_type = 'paper' AND price > 6;

N7

CREATE PROCEDURE p\_sales

@p\_sale\_date DATE,

@p\_ personalID INT

AS

BEGIN

SELECT

product\_name,

quantity,

sale\_price

FROM

sales

WHERE

sale\_date = @p\_sale\_date

AND personalID = @p\_personnelID;

END;