



BSc Shkenca Kompjuterike dhe Inxhinieri – Sistemet e Bazës së të Dhënave
Ligjëruesi i lëndës: Arbër Perçuku, PhD

Zgjidhjet/përgjigjet - provimi afati Shkurt 2022

Grupi A

Detyra 1 (10 pikë):

```
SELECT od.SalesOrderID, od.ProductID, od.OrderQty
FROM Sales.SalesOrderDetail od
WHERE od.OrderQty =
      (SELECT MAX(OrderQty)
       FROM Sales.SalesOrderDetail AS d
       WHERE od.ProductID = d.ProductID)
ORDER BY od.ProductID desc, od.OrderQty asc;
```

Detyra 2 (12 pikë):

a. Tabela Customer_depo, e dhënë në këtë formë është e normalizuar në Formën e Parë të Normalizimit (1NF) sepse vlerat në kolonat janë të vetme (single atomic), dhe në Formën e Dytë të Normalizimit (2NF) - nuk ka varësi të pjesëshme (no partial dependencies), por nuk është e normalizuar edhe në Formën e Tretë të Normalizimit (3NF).

Në tabelën Customer_depo, e dhënë në këtë formë, ka varësi funksionale kalimtare (transitive functional dependency):

Customer_name, Depo_no varen funksionalishtë nga Customer_no, pra Customer_no --> Customer_name, Depo_no

Depo_location varet funksionalishtë nga Depo_no, pra Depo_no --> Depo_location

b. Normalizimi i tabelës Customer_depo në Formën e Tretë të Normalizimit bëhet duke e dekompozuar tabelën Customer_depo në tabela si më poshtë:

Customer (Customer_no, Customer_name, Depo_no)

Depo (Depo_no, Depo_location)

Detyra 3 (10 pikë):

Create Function dbo.getemrimbiemri (@b_id int)

Returns varchar (100)

AS

Begin

declare @Emri_plote varchar(100);

select @Emri_plote = concat (FirstName, LastName)

from person.person

where BusinessEntityID =@b_id;

Return @Emri_plote;

End;

```
Select ph.BusinessEntityID, dbo.getemrimbiemri (ph.BusinessEntityID),  
       ph.phoneNumber  
from Person.PersonPhone ph  
where ph.BusinessEntityID=2;
```

Detyra 4 (10 pikë):

```
CREATE PROCEDURE dbo.GetReviewer  
@Emri_produktit nvarchar(40)  
AS  
Select ppr.ProductReviewID, pp.Name, ppr.ReviewerName, ppr.EmailAddress  
from Production.Product pp, Production.ProductReview ppr  
where pp.ProductID=ppr.ProductID  
and pp.Name = @Emri_produktit  
GO  
EXEC dbo.GetReviewer @Emri_produktit = 'HL Mountain Pedal';
```

Pyetja 1 (2 pikë):

d). Të gjitha opsionet e paraqitura

Pyetja 2 (2 pikë):

c) Grumbull i organizuar e të dhënave ose informacioneve që mund të qasen, përditësohen dhe menaxhohen

Pyetja 3 (2 pikë):

c). Gjen emrin_klienti, numrin_huanë dhe shumën nga natural join i huamarrësit dhe huasë

Pyetja 4 (2 pikë):

b. Pa sakt

Pyetja 5 (2 pikë):

a). Pa sakt

Pyetja 6 (2 pikë):

b). Not Null

Pyetja 7 (2 pikë):

Në mënyrë konceptuale, modeli relacional/lidhje (relation) përbëhet nga këto komponente: Relacioni/Lidhja, Atributi (Attribute), Domeni (domain), Rreshti (tuple), Shkalla/Niveli (Degree), Cardinality, Baza e të dhënave relacionale

Pyetja 8 (2 pikë):

d). Durability

Pyetja 9 (2 pikë):

b. Abort

Pyetja 10 (2 pikë):

RAID (Redundant Arrays of Independent Disks) është një mënyrë e teknikave të organizimit të disqeve, dhe shërbejnë për të përmirësuar performancën dhe besueshmërinë.

Pyetja 11 (2 pikë):

c. Baza e të Dhënave Document

Pyetja 12 (2 pikë):

b. select, from, where

Pyetja 13 (2 pikë):

a. CREATE VIEW AS SELECT

Pyetja 14 (2 pikë):

d). Primary files and Log files

Grupi B**Detyra 1 (10 pikë):**

```
SELECT ProductID, [Name], SafetyStockLevel, DaysToManufacture
FROM Production.Product
WHERE SafetyStockLevel > ANY (SELECT AVG(SafetyStockLevel)
                              FROM Production.Product)
Order by [Name] desc, DaysToManufacture asc;
```

Detyra 2 (12 pikë):

a. Tabela Studentet, e dhënë në këtë formë është e normalizuar në Formën e Parë të Normalizimit (1NF) sepse vlerat në kolonat janë të vetme (single atomic), dhe në Formën e Dytë të Normalizimit (2NF) - nuk ka varësi të pjesëshme (no partial dependencies), por nuk është e normalizuar edhe në Formën e Tretë të Normalizimit (3NF).

Në tabelën Studentet, e dhënë në këtë formë, ka varësi funksionale kalimtare (transitive functional dependency):

Stud_distrikt varet nga Stud_no, Stud_distrikt varet edhe nga Stud_country, dhe Stud_country është i varur nga Stud_no.

b. Normalizimi i tabelës Studentet në Formën e Tretë të Normalizimit (3NF) bëhet duke e dekompozuar tabelën Studentet në tabela si më poshtë:

Student_info (Stud_no, Stud_name, Stud_distrikt, Age)

Distrikt_info (Stud_distrikt, Stud_country)

Detyra 3 (10 pikë):

```
CREATE FUNCTION dbo.GetDepartment (@DeptID INT)
```

```
RETURNS VARCHAR(40)
```

```
AS
```

```
BEGIN
```

```
    DECLARE @var VARCHAR(40)
```

```
    SET @var =
```

```
    (SELECT [Name]
```

```
        FROM HumanResources.Department
```

```
        WHERE DepartmentID = @DeptID)
```

```
    RETURN @var
```

```
END;
```

```
SELECT
```

```
    BusinessEntityID AS EmployeeID, dbo.GetDepartment(DepartmentID) AS DepartmentName,
    StartDate
```

```
FROM HumanResources. EmployeeDepartmentHistory;
```

Detyra 4 (10 pikë):

```
CREATE PROCEDURE dbo.GetBrand  
@Emri_brandit nvarchar(40)  
AS  
Select pm.Name, mi.IllustrationID, mi.ModifiedDate  
from Production.ProductModelIllustration mi, Production.ProductModel pm  
where mi.ProductModelID=pm.ProductModelID  
and pm.Name = @Emri_brandit  
GO
```

EXEC dbo.GetBrand @Emri_brandit = 'HL Touring Frame';

Pyetja 1 (2 pikë):

d. Të gjitha opsionet e paraqitura

Pyetja 2 (2 pikë):

c). Grumbull i tabelave

Pyetja 3 (2 pikë):

b. Pa sakt

Pyetja 4 (2 pikë):

c). where

Pyetja 5 (2 pikë):

d). Schema, Instance

Pyetja 6 (2 pikë):

c). Null

Pyetja 7 (2 pikë):

Çelësi i huaj (foreign key) përdoret për të gjeneruar marrëdhëniet midis tabelave. Foreign key është një fushë në tabelën e bd, që është primary key tek një tabelë tjetër. Foreign key mund të ketë NULL dhe vlera të duplikuara.

Pyetja 8 (2 pikë):

b. Atomicity

Pyetja 9 (2 pikë):

e. Të gjitha opsionet e paraqitura

Pyetja 10 (2 pikë):

Dy qëllimet kryesore të RAID janë:

1. Përmirësimi i performances përmes striping
2. Përmirësimi i fault tolerance përmes redundancy (mirroring)

Pyetja 11 (2 pikë):

b. Key-Value

Pyetja 12 (2 pikë):

d. Të gjitha opsionet e paraqitura

Pyetja 13 (2 pikë):

a. DROP VIEW

Pyetja 14 (2 pikë):

b). Secondary files

Grupi C

Detyra 1 (10 pikë):

```
SELECT BusinessEntityID, LoginID, JobTitle, VacationHours
FROM HumanResources.Employee e1
WHERE
    VacationHours > (SELECT
                        AVG(VacationHours)
                        FROM HumanResources.Employee e2)
Order by VacationHours desc, JobTitle asc;
```

Detyra 2 (12 pikë):

a. Tabela Personi_shitjes, e dhënë në këtë formë është e normalizuar në Formën e Parë të Normalizimit (1NF) sepse vlerat në kolonat janë të vetme (single atomic), dhe në Formën e Dytë të Normalizimit (2NF) - nuk ka varësi të pjesëshme (no partial dependencies), por nuk është e normalizuar edhe në Formën e Tretë të Normalizimit (3NF).

Në tabelën Personi_shitjes, e dhënë në këtë formë, ka varësi funksionale kalimtare (transitive functional dependency):

Sales_Location, SalesP_name varen funksionalishtë nga SalesP_no, pra SalesP_no --> Sales_Location, SalesP_name

Sales_amount varet funksionalishtë nga Cust_no, pra Cust_no --> Sales_amount

b. Normalizimi i tabelës Personi_shitjes në Formën e Tretë të Normalizimit bëhet duke e dekompozuar tabelën Personi_shitjes në tabela si më poshtë:

SalesPerson (Sales_Location, SalesP_name, SalesP_no)

Sales (SalesP_no, Cust_no, Sales_amount)

Detyra 3 (10 pikë):

```
CREATE FUNCTION dbo.getmax_itv (@b_id int)
RETURNS TABLE
AS
RETURN
(
    select pp.ProductNumber, pp.name, max(pv.StandardPrice) as 'Cmimi_max'
    from Purchasing.ProductVendor pv, Production.Product pp
    where pv.ProductID=pp.ProductID
    and pp.ProductID=@b_id
    GROUP BY pp.ProductNumber, pp.name
);

SELECT * FROM dbo.getmax_itv (2);
```

Detyra 4 (10 pikë):

```
CREATE PROCEDURE dbo.ZbritjeSpeciale
```

```
@Emri_kategorise nvarchar(40)
```

```
AS
```

```
select sop.ProductID, ss.Type, ss.Category, ss.DiscountPct
```

```
from Sales.SpecialOffer ss, Sales.SpecialOfferProduct sop
```

```
where ss.SpecialOfferID=sop.SpecialOfferID
```

```
and ss.Category = @Emri_kategorise
```

```
GO
```

```
EXEC dbo.ZbritjeSpeciale @Emri_kategorise = 'Customer';
```

Pyetja 1 (2 pikë):

a. Sakt

Pyetja 2 (2 pikë):

d). DBMS ruan, modifikon dhe merr (select) të dhëna

Pyetja 3 (2 pikë):

c) Gjen të gjitha rreshtat (tupes) ku shuma është më e madhe se 1200

Pyetja 4 (2 pikë):

b). select

Pyetja 5 (2 pikë):

a). Sakt

Pyetja 6 (2 pikë):

e). opsionet a)., b). dhe c).

Pyetja 7 (2 pikë):

Rregullat për definimin e primary key janë si më poshtë:

Dy rreshta nuk mund të kenë të njëjtën vlerë të primary key; për secilin rresht duhet të ketë një vlerë të primary key; fusha e primary key nuk mund të jetë null; vlera në primary key nuk mundet të ndryshoj ose përditësohet nëse ndonjë foreign key i referohet këtij primary key

Pyetja 8 (2 pikë):

b. Transaction

Pyetja 9 (2 pikë):

d. Të gjitha opsionet e paraqitura

Pyetja 10 (2 pikë):

Tek RAID kuptimi i striping është që të distribuohen të dhënat në mes të disqeve për të shpërndarë ngarkesën, kurse kuptimi i redundancy (mirroring) është që të duplikohet çdo disk.

Pyetja 11 (2 pikë):

c. Pa strukturuara (unstructured)

Pyetja 12 (2 pikë):

d. të gjitha opsionet e paraqitura

Pyetja 13 (2 pikë):

b). Një pamje është një tabelë virtuale e cila rezulton nga ekzekutimi i një pyetësoi (query)

Pyetja 14 (2 pikë):

b). Pa sakt