

Pamukkale Üniversitesi Mühendislik Fakültesi

Bilgisayar Mühendisliği Bölümü

2019-2020 Güz Dönemi - Veritabanı Yönetimi ve Modellemesi

Final Sınavı

Numara:

Adı Soyadı:

1.Soru	2.Soru	3. Soru	4. Soru	Toplam

1- (40 P) Aşağıda verilen boşluklara uygun olan ifade/ifadeleri yazınız.

a)(5 P) _____; Varlık, Özellik ve ilişkileri kullanarak veri modellemesi yapılmasını sağlamaktadır. Grafikselleştirimi sayesinde veri modellemede kolaylık sağlamaktadır. Chen, Crow's Foot ve UML olmak üzere 3 farklı gösterimi vardır.

b) (12 P) Büyük veri kavramında V olarak adlandırılan kavramlardan üç tanesi şu şekildedir:

c) (8 P) _____ veri tabanı üzerinde raporlama ve analiz işlemleri daha çok yapılırken, _____ veri tabanları üzerinde; insert, update, delete gibi işlemler daha sık yapılmaktadır.

d) (5 P) Join işlemi yapılan tabloların kartezyen kümesini oluşturan join türüne _____ denir.

e) (10 P) Transaction işlemlerinin taşınması gereken 4 temel özellik şu şekildedir (ACID):

- 2- (15 P) Aşağıda verilen problem hibrit (birden fazla veritabanının kullanıldığı) bir veritabanı ile modellenmek istenmektedir. Buna göre problemin hangi adımının hangi tür veritabanı ile modellenmesi gerektiğini verilen boşluklara yazınız.

Problem:

Akademik çalışmaların ve kitapların yer aldığı hibrit bir veri tabanı tasarlanmak istenmektedir. Veri tabanı için istenen temel özellikler aşağıda sıralanmıştır.

a) _____ Akademik çalışmayı ya da kitabı yazan yazar/yazarların; adı, soyadı, e-mail adresi, telefon numarası gibi temel bilgiler tutulmalıdır. Ayrıca, çalışmanın indirilme sayısı ve çalışmanın referans gösterilme sayısı da görüntülenmek istenmektedir.

b) _____ Kişilerin yaptığı çalışmalar ve/veya kitaplar sistemde tutulmak istenmektedir.

c) _____ Ayrıca kişilerin birlikte çalıştıkları kişiler ile arasındaki ilişkiler sistem üzerinde modellenmek istenmektedir. Sistem, arama yapılarak erişilen çalışmalara benzer bir çalışmanın olup olmadığı görülebilir olmalıdır ve sistem kişilere birlikte çalışma önerisi yapabilir konumda olmalıdır.

- 3- (15 P) Parametre olarak hayvan türü bilgisini alan ve hayvanların id, isim ve tür bilgisini (isim olarak) döndüren bir UDF oluşturunuz. Oluşturulan UDF parametre olarak hayvan türü bilgisi aldığı takdirde o türe ait hayvanların listesini döndürmelidir. Parametre olarak " (boş) bir değer verildiğinde ise; hayvan listesinin tamamını döndürmelidir. Bu bilgilere göre aşağıdaki T-sql ifadesinde eksik kalan kısımları tamamlayınız. (Zoo veritabanı tabloları en arka sayfada verilmiştir.)

```
CREATE FUNCTION udf_getAnimalTypes (@typeName AS NVARCHAR(50))
```

```
RETURNS TABLE AS RETURN
```

```
    SELECT tbl_animal.id,name,
```

```
           info FROM tbl_animal
```

```
           INNER JOIN tbl_animalType
```

```
           ON tbl_animal.animalType_Id=tbl_animalType.id
```

```
WHERE _____
```

Fonksiyonun Çalışması:

```
SELECT * FROM dbo.udf_getAnimalTypes('')
```

```
SELECT * FROM dbo.udf_getAnimalTypes('Keçi')
```

4- (30 P) Aşı takviminde yer alan aşıları dikkate alarak, hangi hayvana hangi aşının yapılması gerektiğini, hayvana en son yapılan aşının zamanını, bu süre üzerinden geçen gün sayısını, yapılacak olan aşı için kalan gün sayısını ekranda listeleyen bir **udf_GetVaccinePeriod** adında bir UDF yazılmıştır. Tanımlanan UDF iki adet scaler değer döndüren UDF çağırılmaktadır. Bunlar;

- dbo.udf_FLastVT(@animalId): Hayvan id'sine göre en son aşılanma zamanını datetime olarak döndürmektedir.
- dbo.udf_FindDayCount(@time,@period): Time parametresi olarak 3,5,7,1 gibi değerler alırken, @period parametresi olarak 'gün','hafta','ay' ve 'yıl' değerlerini alır. Fonsiyona gönderilen değerlere bağlı olarak toplam gün sayısını int olarak döndürmektedir. Örneğin; UDF, dbo.udf_FindDayCount(3,'ay') değeri ile çalıştırıldığında 90 sayısını geri döndürmektedir.

Bu verileri kullanarak aşağıda tanımlanan UDF'de eksik kalan kısımları tamamlayınız. (Zoo veritabanı tabloları en arka sayfada verilmiştir.)

```
CREATE FUNCTION [dbo].[udf_GetVaccinePeriod]()  
RETURNS TABLE  
AS RETURN  
(
```

```
    SELECT  
        AnimalId=A.id,  
        AnimalName=A.Name,  
        VaccineName=VT.description,  
        SonAsiZamani=_____,  
        AsiPeriyodu_g=_____,  
        AsilcinKalanZaman=_____
```

```
FROM tbl_Animal AS A INNER JOIN  tbl_VaccineType AS VT
```

```
    ON A.animalType_Id=vt.animalType_id  
LEFT JOIN  tbl_VaccineDosage AS VD  
    ON A.id=VD.animalId
```

```
)
```

30.12.2019

Pamukkale Üniversitesi Mühendislik Fakültesi

Bilgisayar Mühendisliği Bölümü

2019-2020 Güz Dönemi - Veritabanı Yönetimi ve Modellemesi Laboratuvarı

Final Sınavı

Numara:

Adı Soyadı:

1.Soru	2.Soru	3. Soru	Toplam

- 1- **(20 Puan)** Uluslararası bir Gemi Lojistik firmasına ait veritabanı tabloları ve kayıtları aşağıdaki gibidir. Bu tablolara göre aşağıda istenen sorguların T-Sql ifadelerini yazınız.

Tbl_GemiTuru

Id	Tanim
1	Genel kargo
2	Konteyner
...	...

Tbl_gemi

Id	Ad	Uretim_Tarihi	Kapasite	GemiTur_Id
1	Aframax	01.12.2000	500	5
2	Handysize	05.11.2017	400	1
...

- a) **(5 p)** Firmanın sahip olduğu gemilerin adı, gemi türü ve kapasitesi
- b) **(5 P)** Yaşı 10'dan büyük olan gemilerin tüm bilgileri
- c) **(5 P)** Firmanın sahip olduğu gemi türleri
- d) **(5 P)** Gemilerin tür bazında toplam kapasiteleri

- 2- **(40 Puan)** Hayvanların beslenme programlarındaki eksik gün sayılarını while yapısı kullanarak hesaplayan bir store procedure(SP) yazılmıştır. Yazılan SP'de bazı satırlar eksik bırakılmıştır. Yazılan SP parametre olarak "@dayNumber" adında bir parametre alır ve bu parametre hayvanlara kaç günlük beslenme programı planlandığını göstermektedir. Parametre 7, 15, 30 gibi herhangi bir sayı değeri atandığında planlamadaki eksik gün sayısı bu gönderilen sayıya göre hesaplanmaktadır. SP'ye parametre gönderilmeden de çalışabilmektedir. Ancak böyle bir durumda SP 1 haftalık yemek planlaması olduğunu varsayıp hesaplamaları buna göre yapmaktadır. (Hayvanların yemek planlaması hiç yapılmamış olabilir.)

```
CREATE PROC SP_getDeficentMealTime (_____)
AS BEGIN
DECLARE @tmp TABLE(
    name NVARCHAR(50),
    deficentdayCount INT
)
DECLARE @name AS NVARCHAR(50)
DECLARE @dayinfo AS INT
DECLARE MealControl Scroll CURSOR FOR
    SELECT A.Name,COUNT(DISTINCT AFS.day)
    FROM tbl_Animal A
        _____
        tbl_Animal_Food_Schedule AFS
    ON A.id=AFS.animal_Id
        GROUP BY A.id,A.Name
        _____ MealControl
    FETCH MealControl INTO @name,@dayinfo
    WHILE (_____) %Liste sonu kontrolü
    BEGIN
        IF(_____)
            INSERT INTO @tmp values(@name,0)
        ELSE
            INSERT INTO @tmp
                values(@name,(@dayNumber-@dayinfo))
        FETCH MealControl INTO _____
        End
        _____ MealControl
    DEALLOCATE MealControl
    _____
END
```

- 3- (40 Puan)Aşağıda bir şirkete ait çalışanların (Employee) ve çalışanların yöneticilerinin (Manager) isimlerini tutan **tbl_Employee** tablosu ve içeriği verilmiştir. Bu verilere göre; “EmployeeHierarchy” isimli bir UDF tanımlanmıştır. Bu UDF çalışan ismini alır ve sonuç olarak çalışanların bağlı olduğu yönetici bilgilerini yinelemeli olarak gösterir. Örneğin; “James” kişisi parametre olarak gönderildiğinde UDF, “James->Amanda->Charlie” sonucunu döndürmektedir. Yöneticinin kendisi gönderildiğinde ise; sadece yöneticinin ismi ekranda görüntülenmektedir. Yani “Elai” gönderildiğinde, UDF “Elai” sonucunu üretir.

DESKTOP-LC0CKVR\...dbo.tbl_Employee		
	Employee	Manager
	Charlie	NULL
	Peter	James
	Elai	NULL
	Graham	Emanuel
	Amanda	Charlie
	Sen	Graham
	Emanuel	NULL
	James	Amanda
	Elai	NULL
	Victor	Elai

Verilen bilgilere göre tanımı verilen UDF’de eksik kalan kısımları doldurunuz. (Her bir boşluk 10 puandır.)

Fonksiyonun Çalışması:

```
SELECT dbo.EmployeeHierarchy('James')
```

Dönen Sonuç:

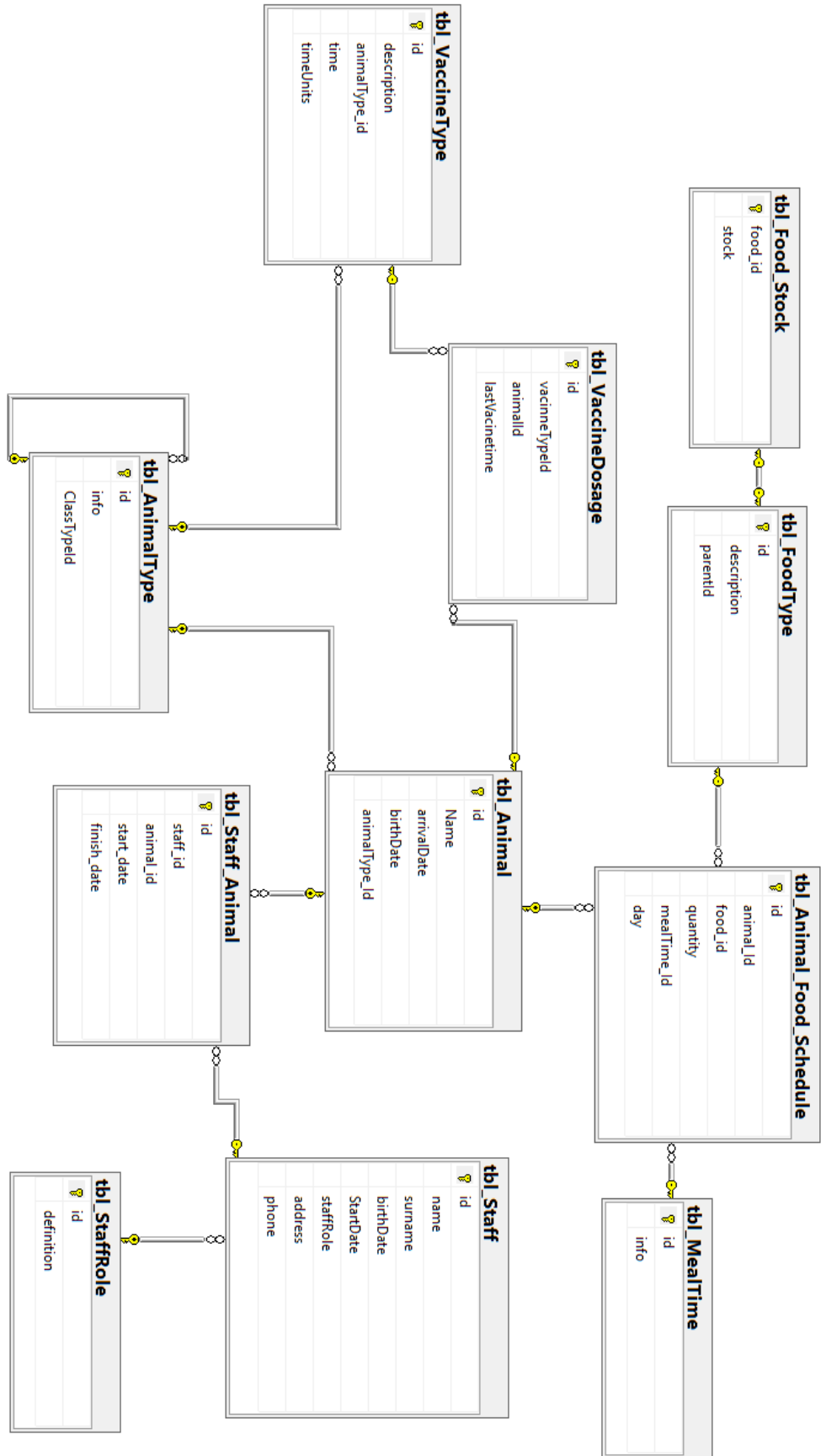
James->Amanda->Charlie

```

CREATE FUNCTION dbo.EmployeeHierarchy
(
    _____
)
RETURNS TABLE WITH AS
RETURN
WITH
    cte_Recur AS (
        SELECT
            CSH = CAST(CONCAT(e.Employee, '->' + e.Manager) AS VARCHAR(1000)),
            e.Manager,
            NodeLevel = 1
        FROM
            dbo.tbl_Employee e
        WHERE
            _____

        UNION ALL
        SELECT
            CSH = CAST(CONCAT(R.CSH, '->' + E.Manager) AS VARCHAR(1000)),
            E.Manager,
            NodeLevel = R.NodeLevel + 1
        FROM
            dbo.tbl_Employee AS E _____ cte_Recur AS R
            ON e.Employee = r.Manager
        WHERE
            E.Manager IS NOT NULL
    )
SELECT
    _____
FROM
    cte_Recur r;

```



Zoo Veritabanı diyagramı