Veritabanı ve Yönetim Sistemleri

Öğr. Gör. M. Mutlu YAPICI

Ankara Üniversitesi
Elmadağ Meslek Yüksekokulu

Ders İzlencesi

Hafta	Modüller/İçerik/Konular
1. Hafta	Temel Kavramlar
2. Hafta	Veri Modelleri
3. Hafta	
4. Hafta	
5. Hafta	
6. Hafta	
7. Hafta	
8. Hafta	
9. Hafta	
10. Hafta	
11. Hafta	
12. Hafta	
13. Hafta	
14. Hafta	

Kavramsal, Mantıksal ve Fiziksel Model Hazırlama

Örnek Soru1.

Bir şirkete ait aşağıda belirtilen gereksinimler doğrultusunda veritabanı tasarımı yapılması istenmektedir.

- Şirket birden fazla bölümden oluşmaktadır ve her bölüme ait bölüm adı, bölüm numarası, adresi, telefonu ve bölüm yöneticisinin sicil nosu saklanmak isteniyor.
- Bölüm yöneticisi çalışanlar arasından bir kişi olarak seçiliyor ve bir bölümde sadece bir yönetici bulunabiliyor. Bölüm yöneticisinin göreve başlama tarihi, yönetici olan kişinin tc nosu ve yönettiği bölüm no yönetici ilişkisi üzerinde tutuluyor.
- Bölüm adresi bina no, kapı no, cadde, sokak ve şehir niteliklerinin birleşiminden oluşmaktadır ve telefon ise birden fazla değer alabilir.
- Bölümler proje geliştirmektedir ve her proje bir proje id, proje adı ve açıklaması bilgilerine sahiptir.
- Bir bölümde birden fazla proje geliştirilmektedir ancak, her proje sadece bir bölüm tarafından gerçekleştirilebilinir.

Kavramsal, Mantıksal ve Fiziksel Model Hazırlama

Örnek Soru1.(Devamı)

- Çalışanların, tc no, adres, maaş, doğum tarih ve işe başlama tarihleri veritabanında saklanmaktadır.
- Çalışan adresi bina no, kapı no, cadde, sokak ve şehir niteliklerinin birleşiminden oluşmaktadır. Çalışanların yaş bilgisi türetilmiş bir niteliktir.
- Her çalışan sadece bir bölümde çalışabilir ancak bir bölümde birden fazla çalışan bulunabilir.
- Her projede birden fazla çalışan bulunabilir ve bir çalışan birden fazla projede çalışabilir.
- Çalışanların çalıştıkları proje ile ilgili projeye başlama tarihi bilgisi proje çalışanı ilişkisi üzerinde tutulmaktadır.
- Her varlığın birincil anahtarlarını gösteriniz.

Yukarıdaki özelliklere uygun ER diyagramını çiziniz ve bu kavramsal modele uygun mantıksal modeli oluşturarak oradan da fiziksel modele aktarınız.

Kavramsal, Mantıksal ve Fiziksel Model Hazırlama

Örnek Soru 2.

Ev Kiralama şirketi için Varlık-İlişki şemasını çiziniz.

- Sistemde ev, kiracı ve ev sahibi bilgileri saklanacaktır.
- Her ev için ev no'su, ev tipi, oda sayısı, kira ücreti, ev Yaşı, Yapılış tarihi veritabanında saklanacaktır.
- Ev yaşı türetilmiş veridir.
- Bir evin bir veya birden fazla sahibi olabilir ,bir ev sahibinin de bir veya birden fazla evi olabilir.
- Ev sahibinin kimlikno'su, isim, telefon numaraları ve cinsiyeti veritabanında saklanmalıdır.
- Telefon no birden fazla değer alabilir, ev sahibinin isim verisi ad ve soyadtan oluşmaktadır.
- Bir kiracı (bir kişi veya aile olabilir) bir ev kiralayabilir, bir ev ise sadece bir kişi (kiracı) tarafından kiralanabilir.
- Kiracılar için kimlik no'su, adı, soyadı, telefonu, cinsiyeti ve medeni hali veri tabanında tutulmalıdır.
- Her ev bir kiracıya sahip olmayabilir ama her kiracı mutlaka bir ev kiralamıştır
- Kiracıların evleri ne zaman kiraladıkları (kira sözleşmesinin başlangıç ve bitiş tarihleri) bilgisinin de veritabanında tutulması gerekmektedir.

Not: Her varlığın birincil anahtarını gösteriniz.

SQL - Komutlar

- SQL dilinde bir tablodan kayıt çekmek için SELECT komutu kullanılır.

- SELECT * FROM tablo_adi

SQL - Komutlar

- Görev 10’da oluşturulan veri tabanındaki tüm kullanıcı bilgilerini görüntülemek için;

- SELECT * FROM kullanici_bilgileri;

SQL - Komutlar

- Öğrencilerin tümü tarafından yazılımda alınan notların tamamını listeleyin.

SQL - Select Komutu

- Tablo içindeki sadece belli alanlara ilişkin verileri listelemek için;

- SELECT alanadi1,alanadi2 FROM tablo_adi;

SQL - Select Komutu

- Örneğin notlar tablosunda sadece notu alan ogrenci numarası ve not aldığı tarihi listelemek için;

- SELECT ogrno,tarih FROM notlar;

SQL - Select Komutu

- Verileri listelerken belli bir alandaki verilere göre sıralamak için;

- SELECT * FROM tablo_adi ORDER BY alan1;

SQL - Komutlar

- Örneğin alınan notları tarihe göre sıralamak için;

- SELECT * FROM notlar ORDER BY tarih;

SQL - Komutlar

- SELECT * FROM notlar ORDER BY tarih;
- Kodu şu şekilde de yazılabilir;
- SELECT * FROM notlar ORDER BY tarih ASC;

SQL - Komutlar

- Örneğin alınan notları tarihe göre azalan sırada sıralamak için;

- SELECT * FROM notlar ORDER BY tarih DESC;

SQL - Komutlar

- Verileri listelerken birden fazla alan baz alınarak listeleme yapılabilir

- SELECT * FROM tabloadi ORDER BY alanadi1 DESC, alanadi2 ASC;

SQL - Komutlar

- Örneğin alınan notları alındığı içerik numaralarına göre azalan, öğrenci numaralarına göre de artan sırada listelemek için;

- SELECT * FROM notlar ORDER BY icerikno DESC, ogrno ASC ;

SQL - Komutlar

- Tablodaki belli bir alanın bir koşula uymasına göre verileri listelemek için;

- SELECT * FROM tablo_adi WHERE alanadi1=koşul;

SQL - Komutlar

- Örneğin sadece 9801 no'lu öğrencinin aldığı notları görüntülemek için;

- SELECT * FROM notlar WHERE ogrno=9801;

SQL - Komutlar

- SQL kodlarında metin değerleri tırmak içinde yazılmalıdır.
- Örneğin koşul metin ise;
- SELECT * FROM tabloadi WHERE alanadi1=“metin”;

SQL - Komutlar

- Örneğin kullanıcı bilgileri tablosundan sadece adı Sema olan öğrencilerin bilgilerini görüntülemek için;

- SELECT * FROM kullanici_bilgileri WHERE ad="Sema";

SQL - Komutlar

- Tablodaki verileri birden fazla koşula göre listelemek için;

- SELECT * FROM tablo_adi WHERE alanadi1=“metin“ AND alanadi2=“metin”;

SQL - Komutlar

- Örneğin adı Serdar Öztürk olan kullanıcıların bilgilerini listelemek için;

- SELECT * FROM kullanici_bilgileri WHERE ad="Sema“ AND soyad=“Öztürk”;

SQL - Komutlar

- Tablodaki verileri iki koşuldan birini sağlama durumuna göre listelemek için;

- SELECT * FROM tablo_adi WHERE alanadi1=“metin“ OR alanadi2=“metin”;

SQL - Komutlar

- Örneğin adı Serdar ya da soyadı Kubalı olan kullanıcıların bilgilerini listelemek için;

- SELECT * FROM kullanici_bilgileri WHERE ad="Serdar“ OR soyad=“Kubalı”;

SQL - Komutlar

- Hem ve hem de veya koşulunu içeren sorgulamalar da olabilir

- SELECT * FROM tablo_adi WHERE (alanadi1=“metin“ OR alanadi2=“metin”) AND alanadi3=“metin”;

SQL - Komutlar

- Örneğin öğrenci numarası 9801 ya da 9802 olan 1 nolu içeriği önemli sayfa yapan kullanıcılar varsa listelemek için;

- SELECT * FROM onemli_sayfalar WHERE (ogrno=9801OR ogrno=9802) AND icerikno=1;

SQL - Komutlar

- SELECT (tüm alanlar veya belli alanlar)
- ORDER BY
- ASC-DESC
- WHERE (sayı veya “metin”)
- AND-OR
- (a AND b) OR c
- a AND (b OR c)

SQL - Komutlar

- LIKE
- DISTINCT
- COUNT
- İÇİ İÇE SELECT KULLANIMI
- BETWEEN
- INNER JOIN

SQL - Like Komutu

- SQL dilinde bir tablodan kayıtları çekerken ismi A ile başlayan soyadı B ile biten tarzında sorgulamalar yapmak için LIKE komutu kullanılır.

- SELECT * FROM tablo_adi WHERE alanadi1 LIKE ‘deger1*’;

- SELECT * FROM tablo_adi WHERE alanadi1 LIKE ‘*deger1’;

SQL - Like Komutu

- Görev 10’da oluşturulan veri tabanındaki adı S ile başlayan kullanıcıların kimler olduğunu görüntülemek için;

- SELECT * FROM kullanici_bilgileri WHERE ad LIKE 'S*';

SQL - Distinct Komutu

- SQL dilinde bir tablodan tekrarlayan kayıtları yalnız bir sefer çekmek için DISTINCT komutu kullanılır.

- SELECT DISTINCT alanadi1 FROM tablo_adi

SQL - Distinct Komutu

- Görev 10’da oluşturulan veri tabanındaki önemli sayfa yapılan sayfaların neler olduğunu görüntülemek için (her bir sayfa bir kez listelenecek şekilde);

- SELECT DISTINCT icerikno FROM onemli_sayfalar;

SQL - Count Komutu

- SQL dilinde bir tablodaki kayıtların toplam sayısını belirlemek için COUNT komutu kullanılır.

- SELECT COUNT(*) FROM tablo_adi;

SQL - Count Komutu

- Görev 10’da oluşturulan veri tabanındaki kaç adet kullanıcı olduğunu belirlemek için;

- SELECT COUNT(*) FROM kullanici_bilgileri;

SQL – İç İçe Select

- Görev 10'da oluşturulan veri tabanında kaç farklı arayüz seçimi yapıldığını belirleyin.
- Örneğin 3 öğrenci 3,3 ve 5 no'lu arayüzü seçtiklerinde seçilen farklı arayüz sayısı 2'dir.

SQL – İç İçe Select

- SELECT Count(*) FROM arayuz_tercihleri WHERE id IN (SELECT DISTINCT arayuzno FROM arayuz_tercihleri);

SQL - Between Komutu

- SQL dilinde bir tablodaki kayıtlardan belli aralıktakileri seçmek için BETWEEN komutu kullanılır.

- SELECT alanadi1,alanadi2 FROM tablo_adi where tarih>=01.03.2009 AND tarih<=03.03.2009;

SQL - Between Komutu

- Görev 10'da oluşturulan veri tabanındaki 01.03.2009 ile 03.03.2009 tarihleri arasında not alan öğrencileri ve aldıkları notları listelemek için;

- SELECT ogrno, notu FROM notlar WHERE tarih BETWEEN #03/01/2009# AND #03/03/2009#;

- Tarih: Ay/gün/yıl

SQL – Inner Join Komutu

- İki farklı tablodaki kayıtları eşleştirmek amacıyla INNER JOIN komutu kullanılır.

- SELECT tablo1.alanadi1,tablo2.alanadi2 FROM tablo1 INNER JOIN tablo2 ON tablo1.alanadi3=tablo2.alanadi3;

SQL – Inner Join Komutu

- Örneğin tartışma bölümünde sorulan soruları ve bu sorulara verilen cevapları listelemek için;

- SELECT tartisma_sorular.soru,tartisma_cevaplar.cevap FROM tartisma_sorular INNER JOIN tartisma_cevaplar ON tartisma_sorular.id=tartisma_cevaplar.soruno;

SQL - Komutlar

- IN
- SUM
- AVG
- ROUND
- MAX
- MIN
- Fark bulma komutu
- GROUP BY

SQL - IN Komutu

- SQL dilinde bir tablodan kayıtları çekerken belli kriterleri uyan verileri çekmek için IN komutu kullanılır.

- SELECT * FROM tablo_adi WHERE alanadi1='deger1' OR alanadi1='deger2'OR alanadi1='deger3';
- komutu yerine

- SELECT * FROM tablo_adi WHERE alanadi1 IN ('deger1','deger2','deger3');

SQL - IN Komutu

- Eğitselyazılım veri tabanındaki adı Sema veya Serdar olan kullanıcıları görüntülemek için;

- SELECT * FROM kullanici_bilgileri WHERE ad IN ("Sema" ,"Serdar");

SQL - Sum Komutu

- SQL dilinde alanlardaki verileri toplamak için SUM komutu kullanılır.

- SELECT SUM(alanadi1) FROM tablo_adi;

SQL - Sum Komutu

- Örneğin tartışma kısmında sorulan sorulara işlevselliği ölçüsünde puan verildiğini düşünelim. Bunun için bir tablo oluşturalım.

tartisma_soru_puanlar

Alan Adı	Veri Türü
id	Otomatik Sayı
soruno	Sayı
puanlayan_kisi	Sayı
puan	Sayı

SQL - Sum Komutu

- Bu tablonun içine veri girişi yapalım

tartisma_soru_puanlar

id	soruno	puanlayan_I	puan
1	1	9802	4
2	1	9803	5
3	1	9804	4
4	1	9805	3
5	2	9803	2
6	2	9805	4
*	Yeni		

SQL - Sum Komutu

- 1. soruya verilen puanların toplamını bulmak için;
- SELECT SUM(puan) FROM tartisma_soru_puanlar where soruno=1;
- Bu komut aşağıdaki şekilde de yazılabilir
- SELECT SUM(puan) AS toplam FROM tartisma_soru_puanlar where soruno=1;

SQL - AVG Komutu

- SQL dilinde belli bir alandaki verilerin ortalamasını bulmak için AVG komutu kullanılır.

- SELECT AVG(alanadi1) FROM tablo_adi;

SQL - AVG Komutu

- Örneğin 1 no'lu tartışma sorusuna verilen puanların ortalamasını belirlemek için;

- SELECT AVG(puan) FROM tartisma_soru_puanlar where soruno=1;

SQL - ROUND Komutu

- SQL dilinde sayıları yuvarlamak amacıyla ROUND komutu kullanılır.

- SELECT ROUND(alanadi1) * FROM tablo_adi

SQL - ROUND Komutu

- Örneğin 1 no'lu tartışma sorusuna verilen puanların ortalaması yuvarlayarak görüntülemek için;

- SELECT ROUND(AVG(puan)) FROM tartisma_soru_puanlar where soruno=1;

SQL - MAX Komutu

- SQL dilinde en yüksek değeri bulmak için MAX komutu kullanılır.

- SELECT MAX(alanadi1) FROM tablo_adi

SQL - MAX Komutu

- Örneğin 1 no'lu tartışma sorusuna verilen en yüksek puanı görüntülemek için;

- SELECT MAX(puan) FROM tartisma_soru_puanlar where soruno=1;

SQL - MIN Komutu

- SQL dilinde en düşük değeri bulmak için MIN komutu kullanılır.

- SELECT MIN(alanadi1) FROM tablo_adi

SQL - MIN Komutu

- Örneğin 1 no'lu tartışma sorusuna verilen en düşük puanı görüntülemek için;

- SELECT MIN(puan) FROM tartisma_soru_puanlar where soruno=1;

SQL – Fark bulma Komutu

- SQL dilinde iki değer arasındaki farkı bulmak için;

- SELECT X-Y FROM tablo_adi;

SQL – Fark bulma Komutu

- Örneğin 1 no'lu tartışma sorusuna verilen en yüksek puan ile en düşük puan arasındaki farkı görüntülemek için;

- SELECT MAX(puan)-MIN(puan) FROM tartisma_soru_puanlar where soruno=1;

SQL – GROUP BY

- Tablodaki kayıtların sayısının belli bir alandaki veri bazında gruplanarak listelenmesi için GROUP BY komutu kullanılır.

- SELECT alanadi1, COUNT(alanadi2) AS Toplam FROM tablo_adi GROUP BY alanadi1;

SQL – GROUP BY

- Örneğin her bir konuyu kaç kişinin önemli sayfa yaptığını belirlemek için;

- SELECT icerikno , COUNT (*) as toplamnot from notlar GROUP BY icerikno;

KAYNAKLAR

- **Yrd. Doç. Dr. Altan MESUT**
  http://altanmesut.trakya.edu.tr/vtys1/
- **Öğr. Gör. Dr. Sibel SOMYÜREK**
  http://sibelsomyurek.com/veritabani/ders_notlari.html
- Tokdemir, G. ve Çağıltay, N. E. (2010). *Veritabanı Sistemleri Dersi*. Seçkin yayıncılık, Ankara.