

Veritabanında Saklı Yordamlar:

Bir Veritabanı Tasarımı ve Web Uygulaması

Kemal Akyol¹, Baha Şen², İlhami Muharrem Orak¹

¹ Karabük Üniversitesi, Bilgisayar Mühendisliği Bölümü, Karabük

² Yıldırım Beyazıt Üniversitesi, Bilgisayar Mühendisliği Bölümü, Ankara

kemalakyol48@gmail.com , baha.sen@ybu.edu.tr , imorak@karabuk.edu.tr

Özet: Bu çalışmada bir veritabanı tasarımı ve web uygulaması geliştirilmiştir. Veritabanı sisteminde evrensel olarak tüm kitap bilgilerinin saklanacağı bir şema oluşturulmuştur. Varlık ilişki modeli esas alınarak oluşturulan ilişkişel şemaya göre Oracle veritabanı yönetim sistemi ile tablolar oluşturulmuştur. Çalışmanın web programlama kısmı JSP ile gerçekleştirilmiştir. Veritabanı üzerinde gerçekleştirilen veri çekme, ekleme, silme, güncelleme işlemlerinin tamamı saklı yordamlar ile icra edilmiştir. Saklı yordamlar veritabanı üzerinde yürütülürler ve iyi bir performans sunarlar. Geliştirilen bu çalışma ile kitap adı, kategorisi, yazarı basım yılı gibi bilgilere göre kitap taraması yapılabilmektedir.

Anahtar Sözcükler: Veritabanı Tasarımı, Kitap Bilgi Bankası, Saklı Yordam

Stored Procedures in Database: A Database Design and Web Application

Abstract: In this study, a database design and web-application have been developed. In database system, a relational scheme was built in order to save the whole book information universally. According to this scheme which was built regarding entity-relationship data model, the charts were built via Oracle database management. The web programming part of the work was made by JSP. All of the operations such as data retrieval, adding, deleting and updating were carried out with stored procedure. Stored procedures work on database and reveal excellent performance. Besides, the information such as the book names, their categories, authors and printing dates can be searched via this application.

Keywords: Database Design, Book Knowledge Base, Stored Procedure

1. Giriş

Veritabanı sistemleri sayesinde günümüzde çok büyük miktarda veriler güvenli ve tutarlı bir şekilde saklanmaktadır. Bu sistemlerin veri topluluklarını oluşturmasının yanı sıra daha önemli olan avantajı, veritabanı üzerinde gerçekleştirilen veri çekme, ekleme, silme, güncelleme işlemlerinin tamamını kendi içsel mekanizmalarıyla gerçekleştirebilmeleridir. Saklı yordamlar, veritabanı üzerinde yürütülen ve iyi bir performans sunan veritabanı yönetim sistemi yordamlarıdır. Bu çalışmada evrensel olarak tüm kitap ve yazar bilgilerinin saklanacağı bir veritabanı

tasarımı ve web uygulaması geliştirilmiştir. Varlık ilişki modeli esas alınarak oluşturulan ilişkişel şema doğrultusunda Oracle veritabanı yönetim sistemi ile tablolar oluşturulmuş olup web programlama kısmı JSP ile gerçekleştirilmiştir. Geliştirilen uygulama ile evrensel bir kitap bilgi bankasının oluşturulması hedeflenmiştir.

2. Veritabanı Tasarımı

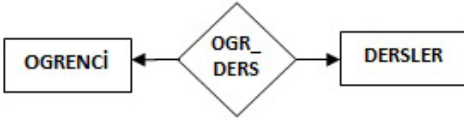
Veri tabanı, birbiriyle ilişkili verilerden oluşan ve bu veriler üzerinde işlemlerin yapıldığı ve yönetildiği bir ortam olarak düşünülebilir [1].

Veri tabanı, tipik olarak bir veya daha çok ilişkilendirmelerin olduğu veri koleksiyonlarıdır. Örneğin üniversite veri tabanı aşağıdaki bilgileri içerebilir:

- Öğrenciler, fakülte, dersler ve sınıfların olduğu varlıklar (Şekil 1.a)
- Derslere öğrencilerin kayıt olması, fakültedeki dersler ve dersler için sınıfların kullanımı gibi varlıklar arasındaki ilişkiler (Şekil 1.b)



a) Varlık ve nitelikleri



b) Varlıklar arası ilişki

Şekil 1. Varlık ve İlişki Örnekleri

Varlık-ilişki veri modeli, gerçek dünyadaki nesnelere ve ilişkilerine tanımlamamıza imkân verir. Veri tabanı tasarımında yaygın bir şekilde kullanılmaktadır. Bu çalışmada da varlık ilişki modelinden yararlanılarak uygulama hazırlanmıştır. Veri tabanı yönetim sistemleri, büyük çaptaki verilerin kullanımı ve bakımına yardımcı olmak için tasarlanan bir yazılımdır. Bu sistemlerin kullanımına olan ihtiyaç hızla artmaktadır [2].

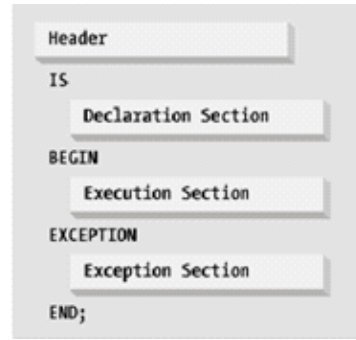
3. PL/SQL Nedir?

PL/SQL, yapısal sorgulama dili SQL'in yordamsal dil uzantısıdır. SQL, ilişkisel veritabanlarında veri sorgulama ve güncelleme işlemleri için kullanılan Oracle veritabanına sistemine has bir dildir. Oracle şirketi SQL dilinin bazı sınırlamalarından dolayı ve daha eksiksiz prog-

ramlama çözümleri sağlamak için bu yordamsal dili geliştirmiştir. PL/SQL dili blok esasına dayanan bir dildir[3,4]. Bu bloklar, yordam ve fonksiyon bloklarıdır. Her bir PL/SQL bloğu, sadece bir tanesi zorunlu olan 4 farklı bölüm içerir:

- Başlık: Sadece isimli bloklar için kullanılır. Zorunlu değildir.
- Bildirim bölümü: Değişken ve alt blokların tanımlamaların yapıldığı bölümdür. Kullanımı zorunlu değildir.
- Kod yürütme bölümü: Çalışma zamanında yürütülen kodların olduğu zorunlu bölümdür.
- İstisna bölümü: Uyarı ve hata durumlarının ele alındığı bölümdür.

Şekil 2'de PL/SQL blok yapısını görülmektedir.



Şekil 2. PL/SQL blok yapısı [3]

PL/SQL son derece yapılandırılmış, okunabilir ve erişilebilir bir dildir. Bir PL/SQL kodu geliştirdiğimizde bu kodu hiçbir değişiklik yapmadan başka bir sisteme taşıyarak aynı işlevleri yürütebiliriz. Şart cümleleri ile akış kontrolleri sağlanabilmekte ve döngüler kullanılabilir. Böylece yazılan programlar verimli ve hızlı bir şekilde yürütülebilmektedir.

3.1. Saklı Yordamlar

Bir veritabanı uygulamasında gerçekleştirilecek olan işlemler için gerekli kodlamalar veritabanı sisteminde iki türlü yürütülür. Birinci yöntem, ilgili kodların web programlama arayüzünden gelmesidir. İkinci yöntemde ise

yapılacak işlemler veritabanında saklanır ve programdan bu saklı yordamlar çağrılır. Bu yöntemin birçok üstünlükleri vardır. Bunlar:

- Yordam, veritabanında saklı olduğu için veritabanı yöneticisi gerekli gördüğü zamanlarda program kodlarına girmeden doğrudan veritabanı üzerinde gerekli değişiklikleri yapabilir. Bu işlemde geliştirilen yada geliştirilecek uygulamalara modülerlik kazandırır.
- Saklı yordam bir kez derlenip defalarca aynı kod parçacıkları çalıştırıldığı için yüksek seviyede bir performans sağlar. Aynı zamanda ağ ortamında çalışan bir uygulamada ağ trafiğini azaltır.
- Saklı yordam üzerindeki bir değişikliği ancak veritabanı yöneticisi gerçekleştirebilir. Bu işlem, veri güvenliğine yardımcı olur.

Tablodan veri çekme, silme, tabloya veri ekleme ve tablodaki veriyi güncelleme işlemleri saklı yordamların kullanıldığı başlıca alanlardır.

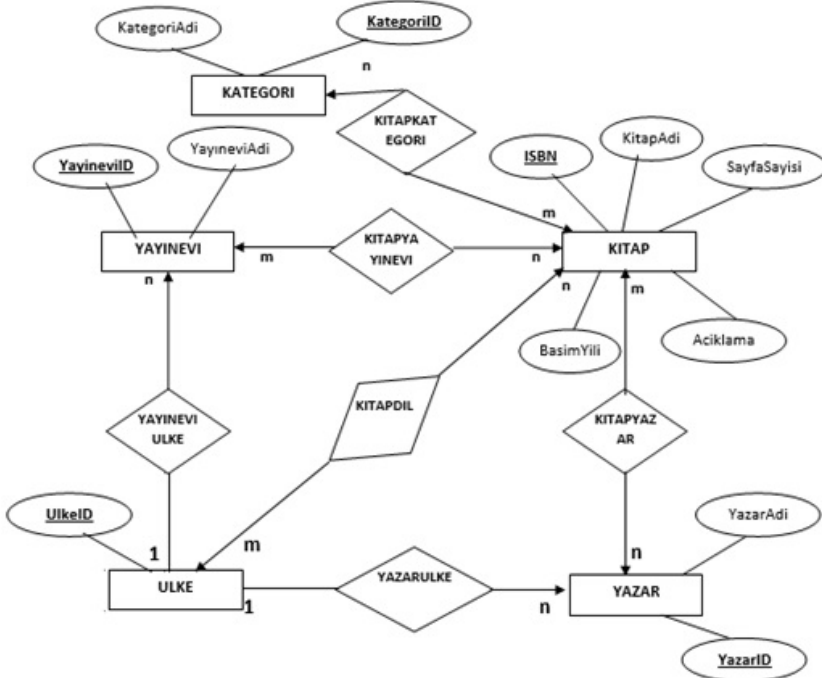
Tetikleyiciler, veriler üzerinde gerçekleştirilecek işlemlerde veri bütünlüğünün sağlanmasına yardımcı olan bir nevi özelleştirilmiş saklı yordamlardır. Bir tablo üzerinde belirli bir olaya bağlı olarak tetiklenip çalışan SQL kodlarıdır. Tablo üzerindeki tetikleyicileri tetikleyen olaylar silme, ekleme ve güncelleme işlemleridir. Yordam, kullanıcı, uygulama veya tetikleyici tarafından koşturulur. Tetikleyiciler ise, veritabanı yönetim sisteminde dolaylı bir tetikleme işlemi yürütürler [5].

4. Uygulama

4.1. Tasarım Aşaması

Kitap bilgi bankası için belirlenen amaç doğrultusunda “Varlık İlişki Modeli” kullanılarak Şekil 3’de verilen kavramsal tasarım gerçekleştirilmiştir.

Kavramsal tasarım doğrultusunda oluşturulan Şekil 4’deki ilişkisel veri tabanı şemasına göre Oracle 10G veri tabanı üzerinde tüm tablolar



Şekil 3. Varlık İlişki Modeli Kullanılarak Veri Modellemesi

KITAP BİLGİ BANKASI [Sistemden Güvenli Çıkış]

Sayın, **Kemal AKYOL** Sistemimize Hoşgeldiniz
Bugün:18 Mayıs 2010 Salı 18:15:47
Sisteme Son Giriş Tarihi:18.05.2010

Kitap Bilgileri	Kategori-Yazar-Yayınevi-Dil Bilgileri
Kitap Ekleme İçin Tıklayınız	Dil Ekleme İçin Tıklayınız
Kitap Silme İçin Tıklayınız	Yazar Ekleme İçin Tıklayınız
Kitap Kategorisi Belirlemek İçin Tıklayınız	Yayınevi Ekleme İçin Tıklayınız
Kitabın Yazıldığı Dili Belirlemek İçin Tıklayınız	Kategori Ekleme İçin Tıklayınız
Kitap Yazarını Belirlemek İçin Tıklayınız	İstatistik Bilgileri Görmek İçin Tıklayınız
Kitap Yayınevinin Belirlemek İçin Tıklayınız	

Şekil 6. Kitap Bilgi Bankası Yönetici Bilgi Giriş Ekranı

KITAPBILGI tablosuna bilgi kaydetmek için kullanılan saklı yordam aşağıdaki gibidir.

KITAP KAYIT EKRANI

ISBN Numarası : 2896-5736-1240-1992
Kitap Adı :
Sayfa Sayısı :
Basım Yılı :
Açıklama :

```
create or replace PROCEDURE  
KitapBilgiKayitSP(id IN nvarchar2,adi in  
nvarchar2,ssayisi in number,basimYili in  
number,aciklama in nvarchar2) is  
  
begin  
  
insert into kitapbilgi (ISBN,KitapAdi,SayfaSa  
yisi,BasimYili,Aciklama) values (id,adi,ssay  
isi,basimYili,aciklama);
```

```
END KitapBilgiKayitSP;
```

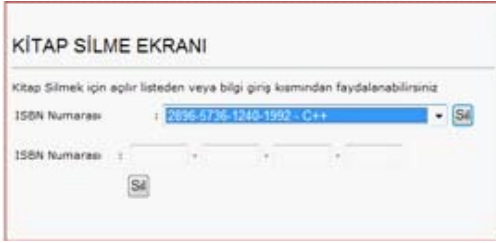
KITAPBILGI tablosuna kitap bilgisi girişi yapıldıktan sonra kitaba ait diğer yazar, yayınevi, kategori ve dil bilgilerinin girişi esnasında işlem koduna göre ilgili tabloya veri girişini gerçekleştiren KITAPDETAYKAYITSP yordamı aşağıdaki gibidir. Bu yordamın çalışması esnasında yapılacak olan işleme göre ilgili kod yürütülmektedir.

```
create or replace PROCEDURE  
KitapDetayKayitSP(isbn_no in  
nvarchar2,kriter_kodu IN number,islem_kodu  
in number)is  
  
begin  
  
if (islem_kodu=1) then  
  
insert into KitapKategori (ISBN,KategoriKodu)  
values (isbn_no,kriter_kodu);  
  
elsif (islem_kodu=2) then  
  
insert into KitapDil (ISBN,YazildigiDil)  
values (isbn_no,kriter_kodu);  
  
elsif (islem_kodu=3) then  
  
insert into KitapYazar (ISBN,YazarKodu)  
values (isbn_no,kriter_kodu);  
  
elsif (islem_kodu=4) then  
  
insert into KitapYayınevi (ISBN,YayinEviKodu)  
values (isbn_no,kriter_kodu);  
  
end if;  
  
END KitapDetayKayitSP;
```

Bir kitap ile ilgili tüm detay veri girişlerinde saklı yordamlar ile işlemler yürütülmüştür. Böylelikle, sisteme modülerlik kazandırılmış

ve aynı kodlar defalarca yazılmamıştır. Veritabanı sisteminde bulunan saklı bir yordam ile gönderilen parametre bilgilerine göre tüm işlemler(kategori, dil, yayınevi, yazar bilgileri) gerçekleştirilmektedir.

Uygulamada, yönetici veri girişi yapabildiği gibi veri silme işlemlerini de saklı yordamlar ile yapabilmektedir. Şekil 8’de kitap silme ile ilgili ekran görüntüsü yer almaktadır. Bu ekrandan bilgi girişi, sistemdeki kitaplardan birinin açılır listeden seçimi veya ISBN bilgisinin klavyeden girişile yapılabilir. Açılır listeye gelen veriler de KİTAPISBNLISTE-LESP yordamı ile gelmektedir. Görüldüğü üzere, bir yordam istenilen yerde kullanılmakta ve böylece daha az kod ile daha çok işlem elde edilmektedir.



Şekil 8. Kitap Bilgi Bankası Kitap Silme Ekranı

Silinmek istenen kitaba ait veri girişi esnasında sistemde o ISBN numarasının olup olmadığının kontrolünü KitapISBNSorgulaSP yordamı yürütmektedir. Bu yordamdan dönen değer “true(doğru)” ise o zaman KitapSilSP yordamı çalışmaktadır.

```
create or replace PROCEDURE
KitapISBNSorgulaSP(isbn_recordset OUT SYS_
REFCURSOR, isbn_no in nvarchar2) IS
BEGIN
    OPEN isbn_recordset FOR
        SELECT ISBN FROM KitapBilgi where
ISBN=isbn_no;
END KitapISBNSorgulaSP;
```

Şekil 9’da girilmiş olan ISBN numarasının sistemde bulunmadığını belirten bilgilendirme mesajı yer almaktadır. Bu işlemi gerçekleştiren yani sistemde o ISBN numarasıyla ilgili veri kontrolünü yapan yordam KitapISBNSorgulaSP’dir.



Şekil 10. Kitap Bilgi Bankası
Kitap Silme Sonuç Ekranı-2

Şekil 11’deki uygulama arayüzünde görüldüğü üzere geliştirilen bu uygulama ile ziyaretçi sistemde şu işlemleri gerçekleştirilebilmektedir:

KitapSilSP yordamı, KİTAPBILGI tablosundan ilgili kitabın silinmesini sağlayan bir yordamdır. Bu silinen kitap ile ilişkili yazar, yayınevi, kategori, yazıldığı dil gibi diğer tüm verilerinde otomatik olarak yani arka planda sistem tarafından ilgili tablolardan silinmesini sağlayan trigger olan KİTAPSILTRIGGER triggeri aşağıda verilmiştir. Bu trigger ile KİTAPBILGI tablosundaki silme işleminin gerçekleşmesinden sonra(after delete on kitapbilgi) silinen ISBN numarasına göre sistemdeki ilişkili tüm tablolardan veri silme işlemi gerçekleşmektedir. Şekil 10’da girilmiş olan ISBN numarasının sistemde bulunduğunu belirten bilgilendirme mesajı yer almaktadır.

- ISBN numarasına göre arama yapabilir,
- Kategorisine, yazarına, yayınevine, yazıldığı dile, basım yılına göre kitap listeleyebilir. Listelenen kitaplar içerisinde istediği kitabın üzerine gelip seçim yaparak o kitap ile ilgili tüm detay bilgilere ulaşabilir,
- Yaptığı seçime göre sistemde kaç kitap var görebilir,
- Yukarıdaki kriterlerin hepsini ayrı ayrı gerçekleştirebildiği gibi tüm kriterlerin bir arada olduğu sorgulama yapabilir, ayrıca kelime girerek de arama yapabilir. Kullanı-



Şekil 11. Kitap Bilgi Bankası Bilgi Giriş Ekranı

cının yapacağı seçimler doğrultusunda Şekil 12'deki gibi bir listeleme ekranı oluşacaktır.

KATEGORİLERİNE GÖRE KİTAP LİSLEME EKRANI



Şekil 12. Kitap Bilgi Bankası Listeleme Ekranı

5. Sonuç ve Öneriler

Veritabanının birbiriyle alakalı verilerin topluluğu olduğunu düşünürsek, bu topluluk üzerinde gerçekleştirilecek olan işlemler için veritabanı yönetim sistemleri tasarlanmıştır. Veritabanı yönetim sistemleri verilerin güvenliğini ve tutarlılığını sağlar. Uygulama geliştiricilerin veritabanı üzerinde gerçekleştireceği işlemlerin verimli ve hızlı bir şekilde veritabanına zarar vermeden gerçekleştirilmesi için veritabanı içinde yürütülen yapılar vardır. Saklı yordamlar ve tetikleyiciler veri tabanı programlamanın vazgeçilmez unsurlarıdır. Sistemin düzgün çalışması ve veri tutarlılığının sağlanması

açısından mükemmel yapılarıdır. 3-4 satırlık bir tetikleyici kodunun yürütülmesi ile icra edilen işlemlerin web uygulaması tarafında yazılması hem zaman maliyeti hem de bellek maliyeti açısından çok fazla maliyet gerektirir. Uygulama geliştirme arayüzünde yapılması istenilen işleme göre ilişkili veriler bulunacak ve bu veriler üzerinde işlem yapılacak ki, bu da çok fazla emek ve maliyet demektir. Saklı yordamlar ile yapılacak işlemler bir kez hazırlanır ve daha sonra istenilen yerde ilgili saklı yordamlar çağrılarak sistemin hızlı veri işleyebilmesi sağlanmış olur.

6. Kaynaklar

- [1] Özkan, Y., "Veri Madencili Yöntemleri", Papatya Yayıncılık Eğitim, İstanbul, 14 (2008).
- [2] Ramakrishnan, R., Gehrke, J., Database Management Systems, 3rd Edition, McGraw-Hill, Singapore, 4,25 (2003).
- [3] Oracle PL/SQL Programming, 3rd edition, O'Reilly & Associates, 1005 Gravenstein Highway, Sebastopol, CA 95472 ,p 54,55,
- [4] <http://www.bilisimweb.org/ders/veri/plsql.pdf> (Erişim Tarihi:24.11.2013)
- [5] Cyran, M., "Oracle Database Concepts, 10g Release 2 (10.2)", 376, 393, 440-443 (2005).