

175956

**ORACLE VERİTABANI PROGRAMINI (UYGULAMA SİSTEMİNİ)
KULLANARAK “ÜNİVERSİTE BİLGİ SİSTEMİ” YAZILIMINI
GERÇEKLEŞTİRMEK**

YÜKSEK LİSANS TEZİ

Ahmet Yücel ÜRÜŞAN

0309040005

Tezin Enstitüye Verildiği Tarih : 27 Ocak 2006
Tezin Savunulduğu Tarih : 02 Şubat 2006

Tez Danışmanı: Prof. Dr. Behiç ÇAĞAL

Diğer Jüri Üyeleri: Yrd. Doç. Dr. Hikmet ÇAĞLAR
Yrd. Doç. Dr. Işım Genç DEMİRİZ

ŞUBAT 2006

ÖNSÖZ

Çalışmalarında bana destek olan ailem ve eşime çok teşekkür ediyorum.

ARALIK 2005

Ahmet Yücel ÜRÜŞAN

İÇİNDEKİLER

Sayfa

Giriş.....	1
1 Veritabanı ve Oracle.....	2
1.1 Oracle Veritabanının Özellikleri.....	3
1.2 Oracle Veritabanı Yapısı.....	4
1.2.1 Kontrol Dosyaları (Control Files).....	6
1.2.2 Log Dosyaları (Log Files).....	6
1.2.3 Veri Dosyaları (Data Files).....	7
1.2.4 Veri Blokları.....	8
1.2.5 Genişletmeler.....	9
1.2.6 Parçalar.....	9
1.2.7 Bellek Yapıları SGA ve PGA (System Global Area-Sistem Genel Alanı, Program Global Area – Program Genel Alanı)	10
1.2.7.1 Veritabanı Tamponları (Database Buffer Cache).....	11
1.2.7.2 Geri Alma Tamponları (Redo Log Buffer).....	11
1.2.7.3 Paylaşılmış Havuz (Shared Pool).....	11
1.2.7.4 Büyük Havuz (Large Pool).....	11
1.2.8 Artalan İşlemleri.....	12
1.2.8.1 Veritabanı Yazıcısı (Database Writer, DBWR).....	13
1.2.8.2 Yedek Yazıcısı (Log Writer, LGWR).....	13
1.2.8.3 Kontrol Noktası (Checkpoint, Ckpt).....	14
1.2.8.4 Sistem Monitörü (System Monitor-SMON).....	14
1.2.8.5 İşlem Analizcisi (Process Monitor-PMON).....	15
1.2.8.6 Yedekleyici (Archiver-ARC).....	15
1.2.8.7 Kurtarma İşlemi (Recoverer-RECO).....	15
1.2.8.8 Aracı İşlem (Dispatcher-DNNN).....	15
1.2.8.9 Kilit (Lock-LCKN).....	15
1.2.9 Veri Sözlüğü (Data Dictionary).....	15
1.2.9.1 Veri Sözlüğü Tabloları (Data Dictionary Tables)	16
1.2.9.2 Veri Sözlüğü Görüntüleri (Data Dictionary Views).....	16
1.2.10 Başlangıç Dosyası Parametreleri (INITSID.ORA Parameters).....	19
1.2.11 İz (Trace) Dosyaları.....	21
1.2.12 Veri Tabanı Şema Nesneleri.....	23
1.2.12.1 Küme (Cluster).....	23
1.2.12.2 İndeks (Index).....	23
1.2.12.3 Rol(Role).....	24
1.2.12.4 Veritabanı Nesneleriyle İlgili Komutlar.....	25
Create Table.....	25
Alter Table.....	26
Drop Table.....	26
Create View.....	27
Create Tablespace.....	27
Create User.....	28

	Create Role.....	28
	Create Index.....	29
	Create Sequence.....	29
	Create Rollback Segment.....	29
	Grant.....	30
	Revoke.....	30
1.2.12.5	Geri Alma Parçası (Rollback Segment).....	30
1.2.12.6	Sıra (Sequence).....	30
1.2.12.7	Kayıtlı Fonksiyonlar (Stored Function) ve Kayıtlı Prosedürler (Stored Procedure).....	31
1.2.12.8	Eşanlam (Synonym).....	31
1.2.12.9	Tablo (Table).....	32
1.2.12.10	Tablo Uzayı (Tablespace).....	33
1.2.12.11	Kullanıcı (User).....	34
1.2.12.12	Gerçekleştirilmiş Görüntüler (Materialized Views).....	34
1.2.12.13	Veritabanı Tetiklemeleri (Database Triggers).....	34
2	Yedek Alma ve Geri Getirme.....	35
2.1	Fiziksel Yedek Alma.....	36
2.1.1	Noarchivelog Modunda Yedek Alma.....	36
2.1.2	Archivelog Modunda Yedek Alma.....	37
2.2	Mantıksal Yedek Alma.....	38
2.2.1	Veritabanı Yedeği Alma ve Geri Getirme (Export, Import).....	38
2.2.1.1	Export.....	39
2.2.1.2	Import.....	40
2.2.2	Oracle'ın Yedek Alma Programı (Recovery Manager - RMAN)	41
3	Başka Ortamlardan Getirilen Bilgilerin Veritabanına Yükleme İşlemi (SQL*Loader) .	41
4	Oracle İşlemlerinin Birbirleriyle Etkileşimi.....	43
4.1	Veriye Aynı Anda Erişim ve Veri Tutarlılığının Korunması.....	43
4.2	Kilitleme Mekanizmaları.....	44
4.3	Veri Aktarım Araçları.....	44
4.4	Net8 Bağlantısı.....	45
5	Dağıtık Veritabanı Yapısı.....	45
6	Veritabanı Güvenliği.....	46
6.1	Güvenlik Mekanizmaları.....	46
6.2	Yetki Verme.....	47
6.3	Disk Kullanımını Sınırlama.....	48
6.4	Profil (Profile) Kullanımı ve Kaynak Sınırlama.....	48
6.5	İzleme (Audit).....	48
7	Oracle Ürünleri	49
7.1	Oracle RDBMS.....	49
7.2	Oracle Uygulama Geliştirme Araçları.....	49
7.2.1	SQL* Plus.....	49
7.2.2	PL/SQL (Procedural Language).....	50
7.2.3	Net8.....	51
7.2.4	SQL*Net.....	55
8	Projeye Başlamak.....	56

KISALTMALAR

RDBMS	: Relational DataBase Management System – İlişkisel Veritabanı Yönetim Sistemi
SQL	: Structured Query Language – Yapısal Sorgu Dili
PL/SQL	: Procedural Language / Structured Query Language
SMON	: System Monitor – Sistem Ekranı
PMON	: Process Monitor – İşlem Ekranı
SGA	: System Global Area – Sistem Genel Alanı
PGA	: Program Global Area – Program Genel Alanı
DML	: Data Management Language – Veri İşlem (Yönetim) Dili
DBWR	: DataBase Writer – Veritabanı Yazıcısı
LGWR	: Log Writer – Log Yazıcısı
ARC	: Archiver - Arşivleyici
RECO	: Recoverer – Kurtarıcı işlem
DNNN	: Dispatcher – Aracı işlem
LCKN	: Lock – Kilit işlem
SID	: System Identification – Veritabanı ismi
MB	: Mega Byte – Hafıza birimi
RMAN	: Recovery Manager – Kurtarıcı Yönetimi
BDE	: Borland Database Engine
ADO	: ActiveX Data Object
ODBC	: Open DataBase Connectivity
OLE	: Object Linked or Embedded
QREPORT	: Quick Report
ODAC	: Oracle Data Access Component

TABLO LİSTESİ

Tablo 1.1	: Sık Kullanılan Görüntüler.....	17
Tablo 1.2	: Oracle tarafından önceden tanımlanan roller.....	28

ŞEKİL LİSTESİ

Şekil 1.1	: Oracle Giriş ekranı.....	4
Şekil 1.2	: Açılmış Oracle oturumunda Prjiku kaydedilmiş konfigürasyonu.....	5
Şekil 1.3	: Datafiles görünümüleri.....	6
Şekil 1.4	: Schema içerisindeki kullanıcılar ve bunlardan USERA için açılmış tablolar....	7
Şekil 1.5	: Bir tablo kayıtlarının görünümü.....	9
Şekil 1.6	: Genişletme içerisinde Veri Bloğunun yeri.....	9
Şekil 1.7	: Parça, Genişleme ve Veri Bloğunun görünümü.....	10
Şekil 1.8	: Sistem Genel Alanı şematik görünümü.....	11
Şekil 1.9	: Oracle'da Prjiku oturumu içerisinde memory konfigürasyonunun görünümü...	12
Şekil 1.10	: Artalan işlemlerinin şematik görünümü.....	13
Şekil 1.11	: Oturum içerisinde bağlantı ve dinleyici adresleri.....	14
Şekil 1.12	: Tablo alanlarının, veri tiplerinin, varsayılan değerlerinin ve diğer tanımlamalarının yapıldığı ekran.....	33
Şekil 1.13	: Veritabanı, tablo uzayı ve veri dosyaları arasındaki ilişkiyi açıklamaktadır.....	33
Şekil 2.1	: Yedekleme yönetimi ekranı.....	35
Şekil 2.2	: Export – Import – Load ekranı.....	39
Şekil 3.1	: SQL*Load işlemlerinin akış şeması.....	42
Şekil 6.1	: USERA kullanıcılarına atanmış rollerin görünümü ve atanabilecek roller, Privileges_for_A rolü dahilinde belirlenmiş yetkiler USERA kullanıcılarına verilmiştir.....	47
Şekil 7.1	: SQL*Plus ekranı ve komut satırı.....	50
Şekil 7.2	: Listener ekranı.....	51
Şekil 7.3	: User ve Server arasında Net 8 bağlantı şeması.....	52
Şekil 7.4	: Oturum içerisinde açık olan Listeners' ların görünümü.....	55
Şekil 8.1	: Program Giriş ekranı. Bu ekranda girilen kullanıcı adının uzunluğuna göre, 4 karakter ise idari ve öğr. üyeleri, 10 karakter ise öğrenci girişi gerçekleşecektir..	57
Şekil 8.2	: Örnek bir tablonun tasarım ekranı. UserA kullanıcılarına ayrılmış tablo uzayında açılmış bir tablonun alanları, bu alanlara atanmış veri tipleri, bu verilerin maksimum uzunlukları ve veri girşinin gerekli olup olmadığı yer alıyor.....	58
Şekil 8.3	: Öğrenci sicil ekranı ve aşağı doğru açılabilen menüler.....	58
Şekil 8.4	: Filtreleme yöntemi ile veri girişini kolaylaştıran ekranlar.....	59
Şekil 8.5	: Örnek bir ekran. Bu ekranda veriler istenirse yazıcıya, istenirse excel programına gönderilebiliyor. Excel dosyası şeklinde kaydedilen veriler daha sonra Tecil Belgesi gibi toplu yazıcı çıktılarının alınabilmesi için kullanılabilir.....	60
Şekil 8.6	: Excel'e gönderilmiş veriler.....	60
Şekil 8.7	: Word'de hazırlanmış Askerlik Tecil Belgesi. Mail Merge özelliği ile değişik öğrenciler için bir seferde Tecil Belgesi düzenlenebilmektedir.....	61
Şekil 8.8	: Resim eklemek veya değiştirmek için resmin bulunduğu (Şekil 8.3 te de görölüyor) yerde çift tıklama yapılır açılan listyeden resim seçilir ve open uygulanır. Resim bu şekilde o öğrenci için ekrana atanmış olur.....	62
Şekil 8.9	: Öğretim planları ekranı.....	62
Şekil 8.10	: Açılan Dersler ekranı.....	63

Şekil 8.11 : Ders İsteme ekranı.....	64
Şekil 8.12 : Not Kartı ekranı.....	65
Şekil 8.13 : Not Kartı önizleme	65
Şekil 8.14 : Transkript önizleme	66
Şekil 8.15 : Arşiv ekranı.....	66
Şekil 8.16 : Danışmanlar ekranı.....	67
Şekil 8.17 : Örnek Rapor ekranı.....	67
Şekil 8.18 : Örnek Rapor ekranı yazıcı çıktısı	68
Şekil 8.19 : Program Hakkında Bilgi ekranı.....	68

ÖZET

Gelişen teknoloji beraberinde de her zaman bazı problemleri getirir. Bilgisayar donanımı, internet, yazılım ve ağ yapıları konularındaki gelişmeler de yine problemleriyle birlikte hayatımızın vazgeçilmez parçaları olmuşlardır. Her gün dünyamızın değişik bölgelerinde meydana gelen Hacker olaylarını duyarız. Bunlar iyi huylu ya da kötü huylu olabilmektedir. Kurumların Internet'e bağlı ağ üzerinden çalışma oranları arttıkça bu tip saldırılardan etkilenme olasılığında haliyle artmaktadır. Örneğin banka hesaplarından para aktarma, stratejik kurumlardan bilgi sızdırma gibi güvenliği tehdit eden saldırılar, özellikle bu tip kuruluşların daha güvenli bir veri tabanına sahip olmasını gerektirmektedir. İşte bu ihtiyaçtan kaynaklanarak geliştirilen ORACLE RDBMS en çok tercih edilebilen bir yapıdadır. Burada, İstanbul Kültür Üniversitesi'nin önümüzdeki yıllarda veri güvenliğini en iyi şekilde sağlayabileceği düşünülen bu veritabanına bağlı olacak şekilde bir Öğrenci Otomasyon Programı geliştirilmiştir. Bu bağlamda, yapılan bu çalışmada, halen üniversitemizde kullanılan, Sn. Prof . Dr. Behiç ÇAĞAL tarafından geliştirilmiş Öğrenci, Muhasebe ve Yönetim otomasyonundaki hazır verileri Oracle' a uygun şekilde transfer edilmiş ve bunun yanısıra yeni veriler girebilmek ve bunları işlemek maksadı ile Delphi ortamında, Delphi ve SQL komutlarını kullanarak bir kullanıcı arayüzü de oluşturulmuştur.

ABSTRACT

Advanced technology in computer systems brings advantages and disadvantages to our life. However, we can state that despite the fact of some disadvantages which we are facing in our daily lives, advances in the internet such as computer programming and computer engineering made our life much practical and easier and none of us can neglect such advances in computer technology in these days. Almost everyday, we are hearing some hacker events and different kinds of viruses which are negatively effecting our organizations. Moreover, other than simple viruses, we noted serious criminal events such as withdrawals from individual bank accounts or such events like illegally providing data from institutions. ORACLE RDBMS was developed in order to prevent such instances as mentioned at above. ORACLE RDBMS will provide a better solution for Istanbul Kultur University as far as student registration and grading are concerned. Also, this study combined Prof. Dr. Behic Cagal's student automation program in Istanbul Kultur University with Oracle according to Delphi condition and SQL direction in order to gather more data for improvement.

GİRİŞ

ORACLE'ın detaylarına inmeden önce sözcük anlamına ve nasıl doğduğuna kısaca bir göz atalım.

Eski Yunanistan ve Roma'da Oracle, "Gaipten haber veren kahin ve bu kahinin kehanette bulunduğu kutsal yer".

İngilizce' de kehanet, kahin.

Oracle'ı iyi bilen uygulayıcılar "Oracle bir veri tabanı değildir, O bir kariyer seçimidir" derler.

Bundan sonra anlatılacak detayların daha kolay anlaşılabilmesi için bir adım ileri giderek İlişkisel Veritabanı Yönetim Sistemleri dendiğinde ne anladığımızı dile getirmemiz faydalı olacaktır.

Bu model ilk önce 1970 yılında Dr. E. F. Codd tarafından tanımlanmıştır. İlişkisel Veritabanı Yönetim Sistemleri (Relational Database Management Systems - RDBMS) büyük miktarda verilerin güvenli bir şekilde tutulabildiği bilgilere hızlı erişim imkanlarının sağlandığı, bilgilerin bütünlük içinde tutulabildiği ve birden fazla kullanıcıya aynı anda bilgiye erişim imkanının sağlandığı programlardır.

Hemen burada SQL ve Oracle arasındaki ilişki şöyle açıklanabilir. SQL veya Structured English Query Language (SEQUEL), IBM firması tarafından Codd'un RDBMS modelini kullanmak için geliştirilmiştir. 1979 yılında, Relational Software (şu an Oracle Corporation), SQL' in ilk ticari uygulamasını geliştirmiştir. Bugün SQL, ilişkisel veritabanı yönetim sistemleri standardı olarak kabul edilmektedir.

1 VERİTABANI VE ORACLE

Veriler üzerindeki çalışmalarda öncelikle zorunlu olan şey iyi planlanmış bir veri tabanıdır. Bu, üzerinde çalışılan projeyi kullanacak olan firma veya gruplardan toplanan ön bilgiler, tesbit edilen ihtiyaçlar, verilerden elde edilmek istenen sonuçlar ve bu bilgilerin tecrübelerle yoğrulmasıyla elde edilebilir. Veritabanı tasarımını etkileyen ve bunun için dikkat edilmesi gereken noktalar şunlardır :

- ✓ Veritabanının yoğun olarak veri girişi için mi yoksa analitik sorgulamalar için mi kullanılacağı
- ✓ Veritabanının gerçek uygulama için mi veya uygulama geliştirme için mi kullanılacağı
- ✓ Veritabanı üzerinde işlem yapacak toplam kullanıcı sayısı ve aynı anda işlem yapacak kullanıcı sayısı
- ✓ Veritabanının büyüme hızının ne olacağı ve hangi tabloların daha hızlı büyüyeceği
- ✓ Tablo, indeks, saklanmış prosedür, ve tetikleyici gibi veritabanı nesnelere kaç tane oluşturulacağı
- ✓ Veritabanının aralıksız çalışıp çalışmayacağı
- ✓ Veritabanından beklenen performans

Veritabanının tasarımı esnasında yukarıda verilen kararların yerine getirilebilmesi için aşağıda verilen seçenekler gözönünde bulundurulmalıdır.

- ✓ Veri Bloğu Boyu (Data Block Size)
- ✓ Başlangıç Parametreleri
- ✓ Geçici tablo uzayı
- ✓ Geri alma (rollback) parça (segment) konfigürasyonu
- ✓ Parça boyutlandırma faktörleri
- ✓ Veri dosyalarının yerleşimi
- ✓ Yedekleme planları
- ✓ Veri dosyası boyutlandırma ve konfigürasyonu

- ✓ Veri sözlüğü boyutlandırma faktörleri
- ✓ Veritabanının işlemsel konfigürasyonu
- ✓ Arşivleme modunda çalışma
- ✓ İşlem saklama (Redo Log) dosyalarının yerleşimi
- ✓ İşlem saklama dosyalarının boyu ve sayısı

Tasarlanan veri tabanının esnek bir yapıda olması, sonradan geliştirilmesi için çok önemlidir. Tabii veriler üzerinde bu planlamaları yaparken, verilerin güvenli şekilde tutulabilmesi konusunda da düşünmek gereklidir. Bu konuda dünyada en güvenli veritabanı programı olarak ORACLE görülmektedir. Sadece Oracle'ın kullanılması hiç bir zaman verilerin artık yüzde yüz güvenli olduğu anlamına da gelemez. İşletim sistemlerinin kendi savunma mekanizmalarını da ihmal etmemek ve iyi kullanabiliyor olmak, internetten gelecek hacker saldırılarına karşı savunma duvarları oluşturmak ve bunları her zaman güncellemek, virüs saldırılarına karşı sistemi sürekli güncellenen güvenilir anti virüs programları ile korumak, çalışanların kötü niyetlerine karşı onları sistem hakkında sahip olabileceklerinin üzerinde yetkilendirmemek, güvenliğin sağlanmasında akla gelebilecek ilk tedbirlerdir.

1.1 ORACLE VERİTABANININ ÖZELLİKLERİ

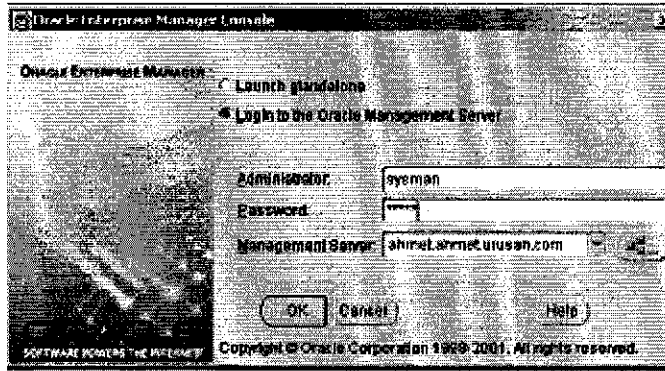
Oracle, bir ilişkisel veritabanı yönetim sistemidir. Özelliklerini kısaca şu şekilde sıralayabilir :

- ✓ Büyük miktarda veri tutabilmekte ve verilerin depolandığı alanları ayarlama imkanı vermektedir.
- ✓ 8 Sürümü itibarı ile onbinlerce kullanıcıya verilerin bütünlüğünü bozmadan hizmet verebilmektedir.
- ✓ Haftalar boyu hiç kapatılmadan çalışabilmektedir.
- ✓ İşletim sistemi, veri erişim dilleri ve ağ iletişim protokolleri standartlarıyla uyumludur.
- ✓ Yetkisiz erişimleri engelleme ve kontrol edebilme imkanı sağlamaktadır.
- ✓ Bütünlüğü veritabanı düzeyinde sağlayabilmektedir. Böylece daha az kod yazılmaktadır.
- ✓ İstemci/Sunucu mimarisinin bütün avantajlarını kullanabilmektedir.
- ✓ Diğer veritabanları ile uyumlu çalışabilmektedir.

Oracle, Delphi veya Visual Basic gibi bir programlama dili değildir. Fakat Oracle tarafından geliştirilen ve Oracle'ın kendi uygulama geliştirme araçları içerisinde kullanılan bir programlama dili vardır. Oracle genellikle büyük çaplı veri kontrolünü gerektiren uygulamalarda kullanılır. Öncelikle, bir ağda Oracle veritabanı, sadece sunucu olarak adlandırılan bilgisayara yüklenir. Bu sunucu, Oracle'ın desteklediği herhangi bir işletim sistemi ile birlikte çalışıyor olabilir. Yani Oracle'ın farklı işletim sistemleri için farklı sürümleri vardır. Bu sunucu, bilgisayara kurulan veritabanı üzerinde tablolar, indeksler, eşanımlar, tablo uzayı ve ihtiyaç duyulan kayıtlı prosedürler oluşturulur. Oracle'ın, bu veritabanına erişerek uygulama programı geliştirmeye yarayan diğer ürünleri de istemci bilgisayara kurulur. Tabii sunucu bilgisayara da isteğe bağlı olarak bu ürünler kurulabilir.

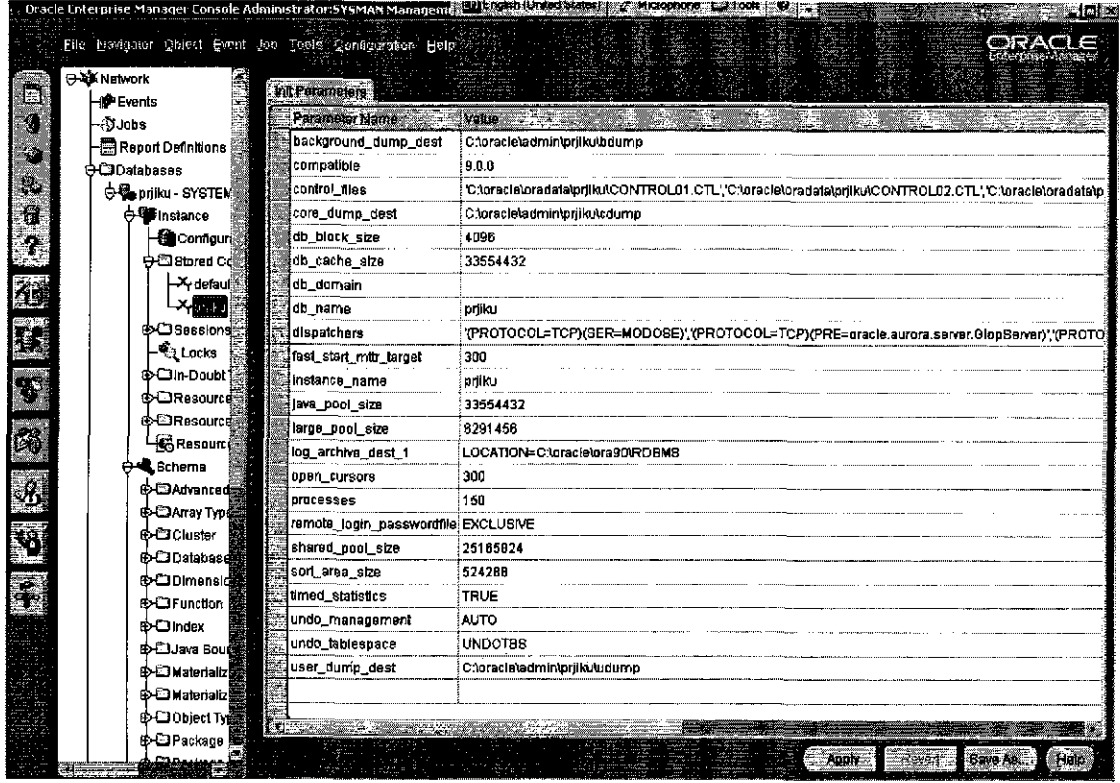
1.2 ORACLE VERİTABANI YAPISI

Oracle veritabanının işletim sistemi tarafından bakıldığında, biri fiziksel diğeri mantıksal olmak üzere iki bölümü vardır. Fiziksel bölüm, işletim sisteminden görünen kısımdır. Bu Data File (Veri dosyası), Control File (Kontrol dosyası) ve Log File (Log dosyası)'dan oluşmaktadır. Mantıksal bölüm, bir ya da daha fazla tablo uzayı (Tablespace) ve tablolar (table), görüntüler (View), sıralar (sequence), eşanımlar (synonym), indeksler (index), kümeler (cluster), veritabanı bağlantıları (database link), prosedürler (procedure), fonksiyonlar (function) ve paketler (package) diye adlandırılan şema nesnelere oluşmaktadır. Fiziksel bölüm, işletim sistemi tarafından görülebilmesine rağmen, mantıksal bölüm ancak Oracle'a bağlanıp, SQL komutları çalıştırılarak görülebilmektedir. Yani, Oracle kurulu herhangi bir makinede, SQL bilgisi olmayan bir insan, Oracle'ın sadece fiziksel bölümünü görebilmektedir.



Şekil 1.1 Oracle Giriş Ekranı

Oracle veritabanındaki her nesnenin bir sahibi (kullanıcı) vardır. Her kullanıcı, bir veya daha fazla tablo uzayına sahip olabilir. Her nesne, ait olduğu kullanıcının herhangi bir tablo uzayında (mantıksal olarak) bulunur. Her tablo uzayı da, kendisine sahip olan kullanıcının nesnelere tutmak için, işletim sisteminde bir veya daha fazla veri dosyasına sahip olabilmektedir.



Şekil 1.2 Açılmış Oracle oturumunda Prjiku kaydedilmiş konfigürasyonu

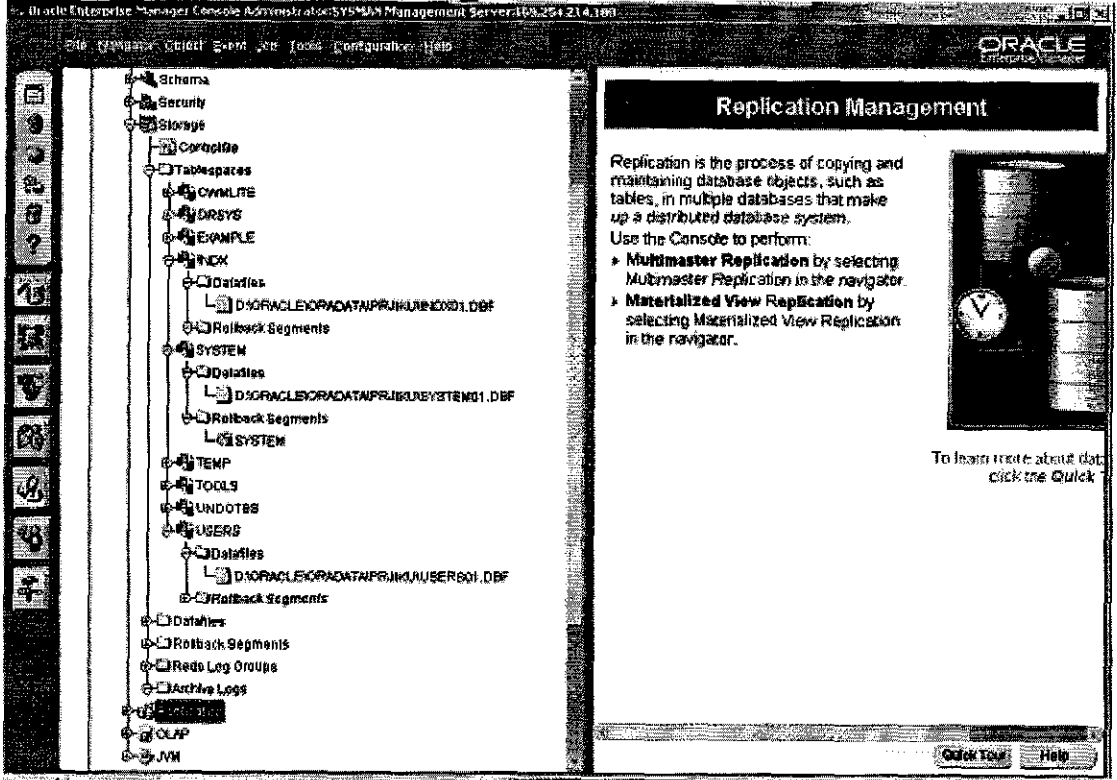
Sonuç olarak, veritabanındaki her bir nesnenin bir kullanıcısı vardır ve bu nesnelere, mantıksal olarak o kullanıcının sahip olduğu tablo uzaylarının herhangi birinin (hangisi olduğu komutlarla öğrenilebilir) içerisinde, fiziksel olarak da o kullanıcının sahip olduğu tablo uzayının herhangi bir veri dosyasında bulunur. Fakat o veri dosyasının içerisine işletim sisteminden bu nesneyi bulabilmek için bakılamaz. Bu nesnenin sahibi ve mantıksal yeri “DML” (veri işleme dili) komutları ile bulunabilmektedir.

Veritabanındaki mantıksal nesnelere oluşturduğu gruba “SCHEMA” denmektedir. Yani SCHEMA, tablolar, kümeler (cluster), indeksler (indexes), veritabanı tetiklemeleri, gibi nesnelere oluşmaktadır.

Fiziksel bölüm, veritabanını oluşturan işletim sistemi dosyalarıdır. Bir Oracle veritabanı, fiziksel olarak bir ya da daha fazla veri dosyası, iki ya da daha fazla log dosyası, bir yada daha fazla kontrol dosyasından oluşur.

1.2.1 KONTROL DOSYALARI (CONTROL FILES)

Tüm Oracle veritabanları kontrol dosyasına sahiptir. Bir kontrol dosyası veritabanı adı, veri dosyaları ve log dosyalarının adı ve diskteki yeri, veritabanının oluşturulma tarihi gibi veritabanı ile ilgili bilgileri tutar.



Şekil 1.3 Datafiles görüntüleri

Her veritabanı oturumu açıldığında Oracle bu dosyayı kontrol ederek gerekli bilgileri alır, eğer veritabanında fiziksel bir değişme olursa (yeni bir log dosyası veya veri dosyası oluşturulması gibi), yapılan değişiklikler Oracle tarafından otomatik olarak kontrol dosyalarına yansıtılır.

1.2.2 LOG DOSYALARI (LOG FILES)

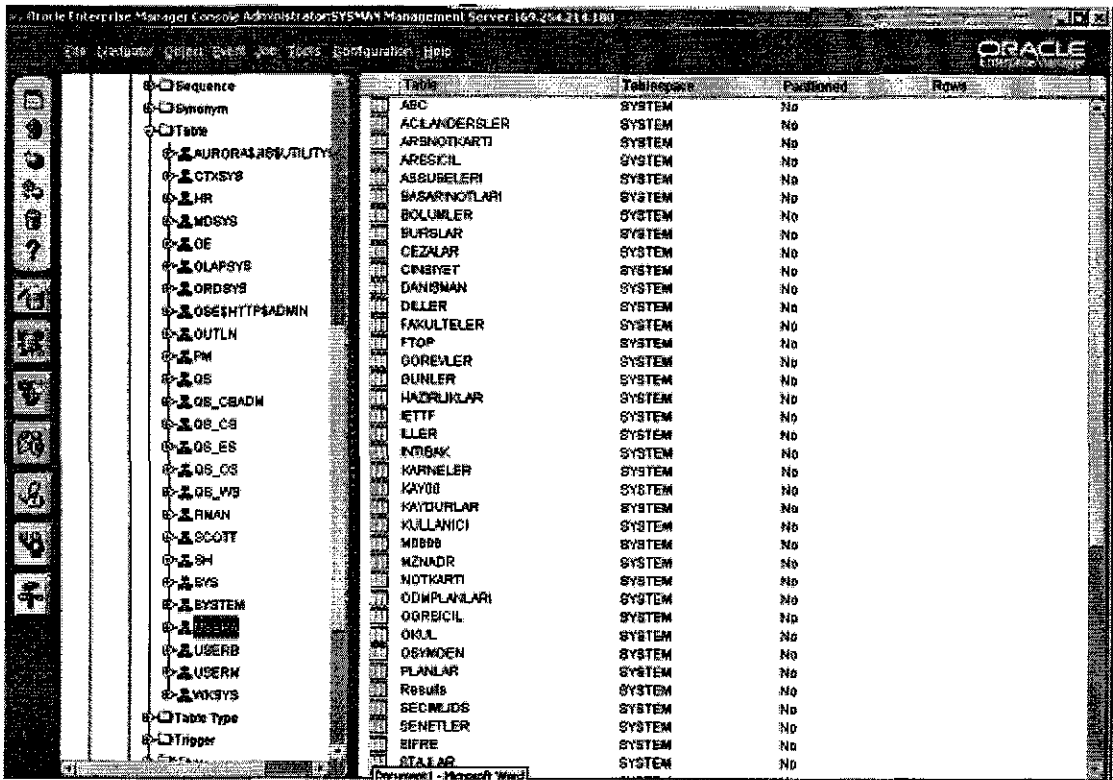
Redo log dosyaları olarak bilinen bu dosyaların amacı veriler üzerinde yapılan tüm değişiklikleri kaydetmektir. Eğer veri dosyalarına kalıcı olarak kaydedilmiş olan , değişikliğe uğramış kayıtlarda bir bozukluk olursa yapılan değişiklikler redo log dosyalarından sağlanabilir ve işlemler kaybolmaz. Birden fazla tekrarlanan bozukluk

durumlarında, redo log dosyalarında bozulmasını engellemek için Oracle, farklı diskler üzerinde redo log dosyalarının birden fazla kopyasının alınmasına olanak sağlar.

Bir veritabanı işlemi sırasında elektrik kesilirse, bellekteki veriler veri dosyalarına kaydedilmeyecek ve verilerin kaybolması durumuyla karşılaşılacaktır. Oracle veritabanı tekrar açıldığında redo log dosyalarında yapılan son değişiklikler veri dosyalarına yansıtılarak verilerin kaybolması engellenir.

1.2.3 VERİ DOSYALARI (DATA FILES)

Veri dosyaları veritabanındaki tüm verileri tutan dosyalardır. Tablo, index gibi mantıksal veritabanı yapılarının içerisindeki veriler, fiziksel olarak veri dosyalarında tutulurlar. Bir veri dosyası kendisi için ayrılan alan dolduğunda, kendi sahip olduğu alanı arttıracak özelliklere sahiptir. Bir ya da daha fazla veri dosyası mantıksal bir veritabanı depolama ünitesi olan bir tablo uzayını oluşturur.



The screenshot shows the Oracle Enterprise Manager console interface. On the left, there is a tree view showing the database structure, with 'Table' selected under the 'USER' schema. On the right, a table lists the tables in the USER schema. The table has columns for 'Table', 'Tablespace', 'Partitioned', and 'Rows'. The tables listed include: ABC, ACELANDERSLER, ARSNOTKARTI, ARSICIL, ASSUBELERI, BASARINOTLARI, BOLLUMLER, BURSLAR, CEZALAR, CINSIYET, DANISMAN, DILLER, FAKULTELER, FTOP, GOREVLER, GUNLER, HAZIRLIKLAR, IETTIF, ILLER, INDIKAK, KARNELER, KAYID, KAYITLAR, KULLANICI, MIBDD, MZNADR, NOTKARTI, ODMPLANLARI, OGRENCIL, OKUL, OBYMOEN, PLANLAR, RBBUS, SECMLDS, SENETLER, SIFRE, and STALEAD.

Table	Tablespace	Partitioned	Rows
ABC	SYSTEM	No	
ACELANDERSLER	SYSTEM	No	
ARSNOTKARTI	SYSTEM	No	
ARSICIL	SYSTEM	No	
ASSUBELERI	SYSTEM	No	
BASARINOTLARI	SYSTEM	No	
BOLLUMLER	SYSTEM	No	
BURLAR	SYSTEM	No	
CEZALAR	SYSTEM	No	
CINSIYET	SYSTEM	No	
DANISMAN	SYSTEM	No	
DILLER	SYSTEM	No	
FAKULTELER	SYSTEM	No	
FTOP	SYSTEM	No	
GOREVLER	SYSTEM	No	
GUNLER	SYSTEM	No	
HAZIRLIKLAR	SYSTEM	No	
IETTIF	SYSTEM	No	
ILLER	SYSTEM	No	
INDIAK	SYSTEM	No	
KARNELER	SYSTEM	No	
KAYID	SYSTEM	No	
KAYITLAR	SYSTEM	No	
KULLANICI	SYSTEM	No	
MIBDD	SYSTEM	No	
MZNADR	SYSTEM	No	
NOTKARTI	SYSTEM	No	
ODMPLANLARI	SYSTEM	No	
OGRENCIL	SYSTEM	No	
OKUL	SYSTEM	No	
OBYMOEN	SYSTEM	No	
PLANLAR	SYSTEM	No	
RBBUS	SYSTEM	No	
SECMLDS	SYSTEM	No	
SENETLER	SYSTEM	No	
SIFRE	SYSTEM	No	
STALEAD	SYSTEM	No	

Şekil 1.4 Schema içerisindeki kullanıcılar ve bunlardan USERA için açılmış tablolar

Normal veritabanı işlemleri boyunca bir veri dosyası içerisindeki veriler okunur ve Oracle için ayrılan belleğe getirilirler. Örneğin bir kullanıcının veritabanındaki bir

tablonun verilerine erişmek istediğini varsayalım. Eğer istenilen veriler bellekte yer almıyorsa, ancak o zaman uygun veri dosyasından okunur ve belleğe getirilirler.

Değişikliğe uğrayan veriler ya da yeni eklenen veriler veri dosyalarına hemen yazılmazlar. Sabit diske erişimi azaltmak ve böylece sistemin performansını artırmak için veriler bellek havuzunda tutulur ve gerektiğinde hepsi birden uygun veri dosyalarına kaydedilirler. Bunu, Oracle artalan işlemleri belirler.

1.2.4 VERİ BLOKLARI

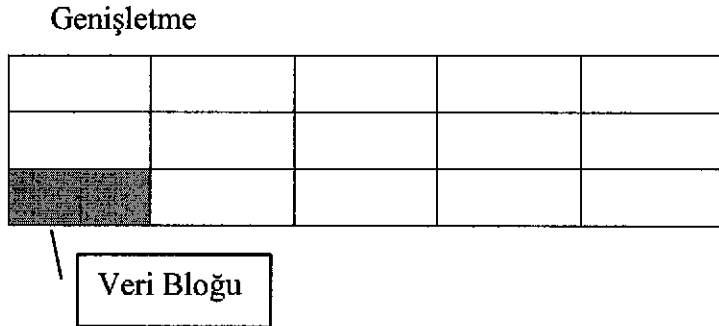
Oracle, diski daha etkin kullanabilmek için işletim sistemine ait bazı konularla da ilgilenmektedir. Veri blokları bunlardan birisidir. Veri blokları, Oracle veritabanında verilerin tutulduğu en küçük birimdir. Veri bloğu, işletim sistemindeki veri dosyalarının belli bir sayıdaki byte'ı kadar uzunluğa sahiptir. Bir tablo uzayı yaratırken, işletim sisteminde belirttiğimiz veri dosyasının bulunduğu yer, Oracle tarafından kendi blok yapısına göre düzenlenir. Tablo uzayı yaratırken beklenen süre, bu süredir. Veri dosyası tamamen dolu olmasa bile, işletim sistemi veri dosyasına dolu gözüyle bakmaktadır. Bu bölümün yönetimini, Oracle işletim sisteminden almıştır. Veriler, veri bloklarında tutulduğundan dolayı, veritabanında nesnelere için kullanılan yer byte sayısı olarak görülebildiği gibi, blok sayısı olarak da görülebilir.

DERSNO	FAKBOLKOD	AD	SOYAD	ACRESB1	ACRESB2
0001010005	0101	CAN	KUŞEYRİOĞULLU	MODA MEKTEP...	K...
0001010001	0101	MUSTAFA BEH.	ÖZALP	ATAKÖYS KIS...	BL
0001010004	0101	ÖZGÜR	AYDINOĞULLU	HÜRRİYET MA...	CA
0001010006	0101	EDAL	ALKAN	ÇAMLIK CD. KA...	DC
0001010007	0101	BERDAR	ATAK	HAZNEGAR MA...	CA
0001010008	0101	YUSUF	KAYA	MARAZAL FEVZ...	3.
0001010011	0101	HAKAN	JAHYN	ATIB NO:119...	
0001010012	0101	ÖZKAN	ÖZALP	ETHEMEFENO...	AP
0001010013	0101	MEHMET	FATMA BAĞCI	EMYT HALIL...	UC
0001010015	0101	EMREK	BAHAR	ALTAYÇE ME...	SO
0001010016	0101	OSMAN	ÖZYİT	DEKİZ KÖ. İL...	SH
0001010017	0101	FERİT	PALA	BYLEZYKÖYS...	AP
0001010020	0101	METİN	ALYBARAN	YSKOP SK NO...	ÇS
0001010021	0101	AYTEK	GÜLSER	HÜRRİYET CD...	AP
0001010022	0101	KEMAN	SERT	ULKLAR MH. Ö...	GE
0001010024	0101	M	FARUK SARINL	ARPA EMYNY H...	PA
0001010025	0101	HURİY	DEMİR	FLORYA MAH...	CA
0001010026	0101	YAVUZ	BELLYN AYGÜN	FLORYA MAH S...	AP
0001010028	0101	EMİN	BYTİKBAI	YANVERDİTE...	CA

Şekil 1.5 Bir tablo kayıtlarının görünümü

1.2.5 GENİŞLETMELER

Oracle veritabanında, nesnelerin büyümek için aldıkları yerlere genişleme denmektedir. Genişletmeler bir veya daha fazla veri bloğundan oluşabilmektedir.



Şekil 1.6 Genişletme içerisinde Veri Bloğunun yeri

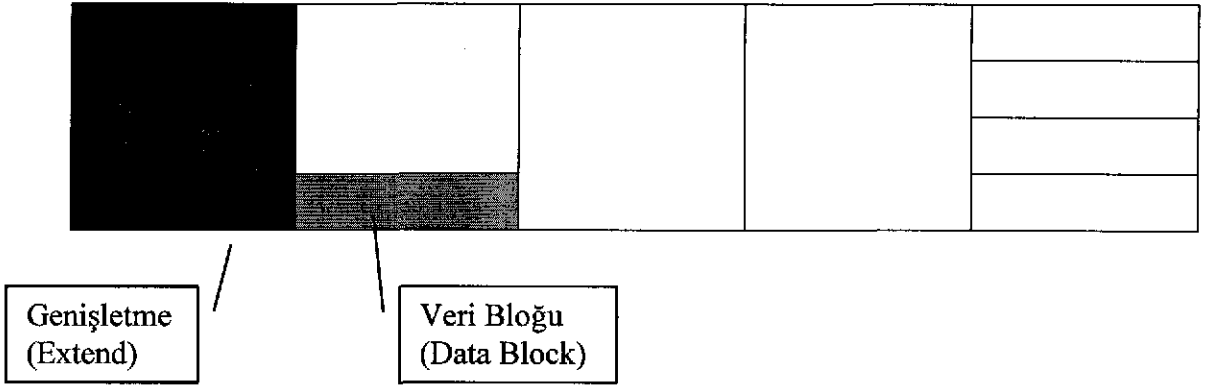
1.2.6 PARÇALAR

Parça (segment), veritabanında alan kullanan her bir nesne için ayrılmış bölümlere denmektedir. Nesneler ilk yaratıldığında onlar için bir yer ayrılmaktadır. Ayrılan yer yetmezse parça bir genişleme olarak büyür. Bu büyüme ilk tanımlamada öngörülen maksimum genişleme sayısına erişene kadar devam edebilir. Bir parça, bir veya daha çok genişlemeden oluşmaktadır.

Parçalar veri, indeks veya geçici bilgileri saklamak için kullanılır. İndeks yaratırken kullanılan parçalar indeks parçası olarak adlandırılmaktadır. Ayrıca,

```
CREATE INDEX
SELECT ... INDEX
SELECT DISTINCT ....
SELECT .... GROUP BY
SELECT ... UNION
SELECT ... INTERSECT
SELECT ... MINUS
```

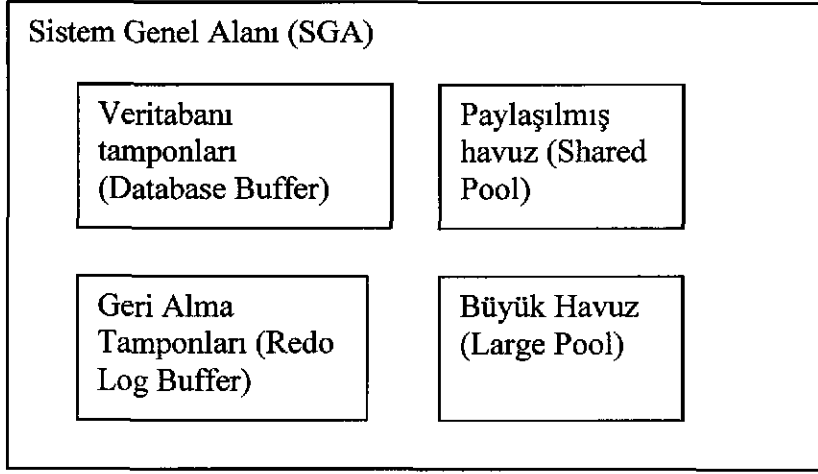
gibi işlemler esnasında da bazı parçalar kullanılırlar. İşlem bittikten sonra bu parçalar serbest bırakılırlar.



Şekil 1.7 Parça, Genişleme ve Veri Bloğunun görünümü

1.2.7 BELLEK YAPILARI SGA VE PGA (SYSTEM GLOBAL AREA- SİSTEM GENEL ALANI, PROGRAM GLOBAL AREA – PROGRAM GENEL ALANI)

Bu yapılar ile, bellek daha etkin olarak kullanılmaya çalışılmaktadır. SGA, herhangi bir Oracle veritabanı için gereken belleğe denir. Bir Oracle instance (oturum)' ı için gerekli veri ve kontrol bilgilerini tutar. Oracle oturumu açıldığında, SGA için gerekli bellek, işletim sisteminden alınmaktadır ve Oracle oturumu kapatılana kadar başka hiç bir program tarafından kullanılamaz. SGA'da, veri dosyalarından okunan veri blokları, kullanıcılara ait bilgiler, değişiklikler ara belleği bilgileri ve paylaşımlı sql bellek alanları bulunmaktadır. PGA, bir kullanıcı veya bağlantı için ayrılan belleğe denir. Bu belleği kullanıcıya ait işlemler kullanmaktadır.



Şekil 1.8 Sistem Genel Alanı şematik görünümü

1.2.7.1 VERİTABANI TAMPONLARI (DATABASE BUFFER CACHE)

Veritabanında son kullanılan veri bloklarının tutulduğu bellek alanlarıdır. Bu bölgede değişikliğe uğramış fakat henüz diske kaydedilmemiş veriler bulunur. Yazma işlemini, Transaction (kesin kayıt işlemi) bitiminden sonra yapmaktadır. Sık kullanılan veri blokları bellekte bulunduğu için disk erişimi azalarak performans artmaktadır.

1.2.7.2 GERİ ALMA TAMPONLARI (REDO LOG BUFFER)

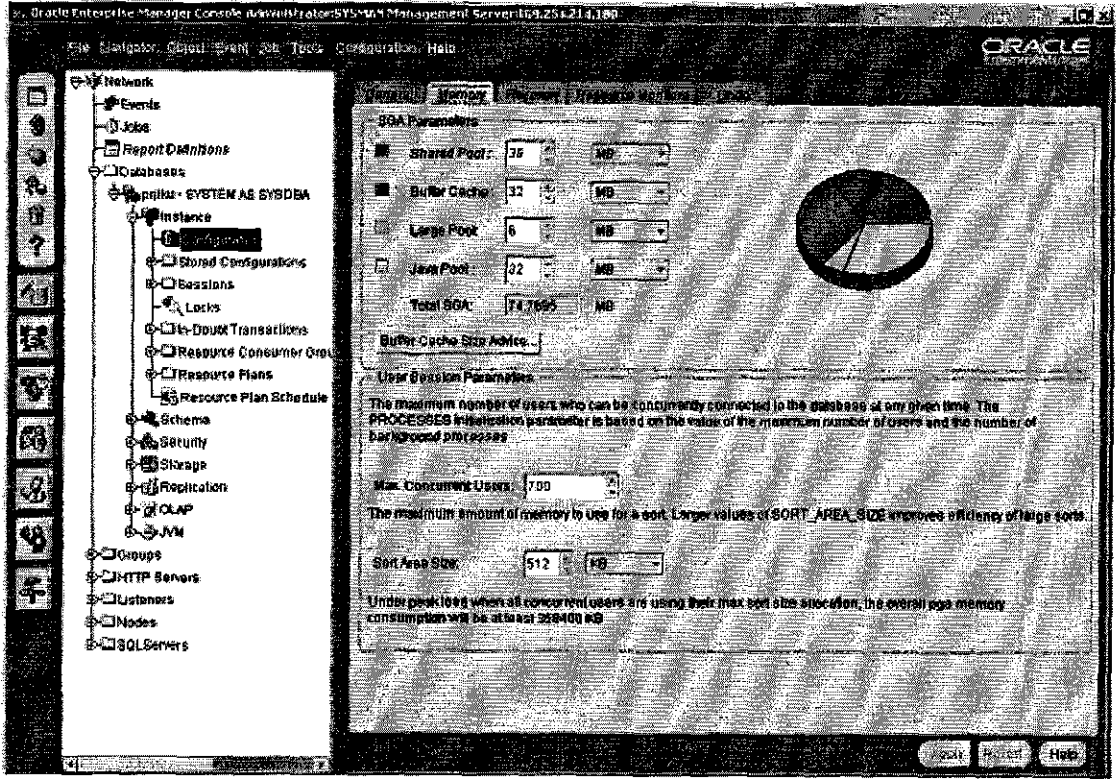
Veritabanında yapılan değişikliklerin tutulduğu alandır. Bu bilgiler herhangi bir kurtarma anında kullanılmak üzere saklanılmaktadır.

1.2.7.3 PAYLAŞILMIŞ HAVUZ (SHARED POOL)

Kullanıcılar tarafından veritabanına gönderilen SQL cümleleri ile ilgili bilgilerin bir daha kullanılmak ve böylece etkinliği artırmak için saklandığı yerdir. Bir SQL komutu kullanılmadan önce ayrıştırılır ve çalıştırılması için bir çalışma planı yapılır. Bu sorgu daha sonra başka kullanıcılar tarafından kullanılmak istendiğinde bellekten direk çalıştırıldığı için performans artışı sağlanır.

1.2.7.4 BÜYÜK HAVUZ (LARGE POOL)

Opsiyonel olarak tutulabilmektedir. Büyük miktarda hafıza ihtiyacı olan işlemler için (yedek alma, yedekten kurtarma gibi) ayrılmaktadır.

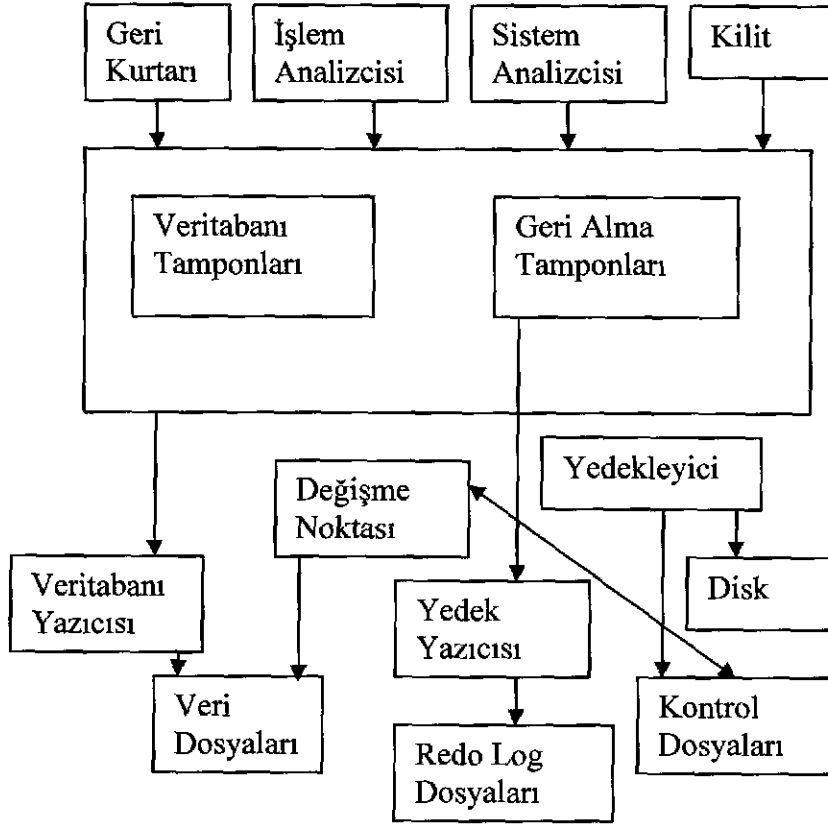


Şekil 1.9 Oracle'da Prjiku oturumu içerisinde memory konfigürasyonunun görünümü

1.2.8 ARTALAN İŞLEMLERİ

Her Oracle Instance (oturum) için bir grup işlem, diske belirli amaçlar için okuma ve yazma yaparak ve birbirleriyle haberleşerek, daha iyi performans ve daha güvenilir işletim ortamı için paralel bir şekilde arka arkaya çalışırlar. Aşağıda bu ilişkisel çalışma şeması gösterilmektedir.

Bu işlemler olmasaydı gerekli işleri yapmak için her kullanıcıya özel ayrı ayrı programlar çalıştırmak gerekirdi. Artalan işlemleri I/O (Input / Output – Giriş / Çıkış) işlemleri yapmakta ve performans ve güvenliği artırmak için Oracle işlemlerini kontrol etmektedirler. Şimdi şemada görülen artalan işlemlerini şöyle açıklanabilir.



Şekil 1.10 Artalan işlemlerinin şematik görünümü

1.2.8.1 VERİTABANI YAZICISI (DATABASE WRITER, DBWR)

SGA'daki tampon alanında değiştirilen blokları, diskteki veri dosyalarına yazar. Verilerin yoğun biçimde değiştirildiği sistemlerde 9 adete kadar DBWR işlemi atanarak yazma işleminin hızı artırılabilir. DB_Writer Processes (Veritabanı yazıcısı işlemleri) adlı başlangıç parametresi, DBWR process lerin sayısını belirler. DBWR, bütünsel bir veritabanı işlemi (transaction) tarafından açık olarak veritabanına kaydedilen (commit) işlemlere ait veri bloklarını yazmaz. Genellikle, bellekte bulunan değişikliğe uğramış blok sayısı artıp boş blok sayısı azalmaya başladığı zaman, en az bloklardan başlamak üzere yazma işlemi yapar.

1.2.8.2 YEDEK YAZICISI (LOG WRITER, LGWR)

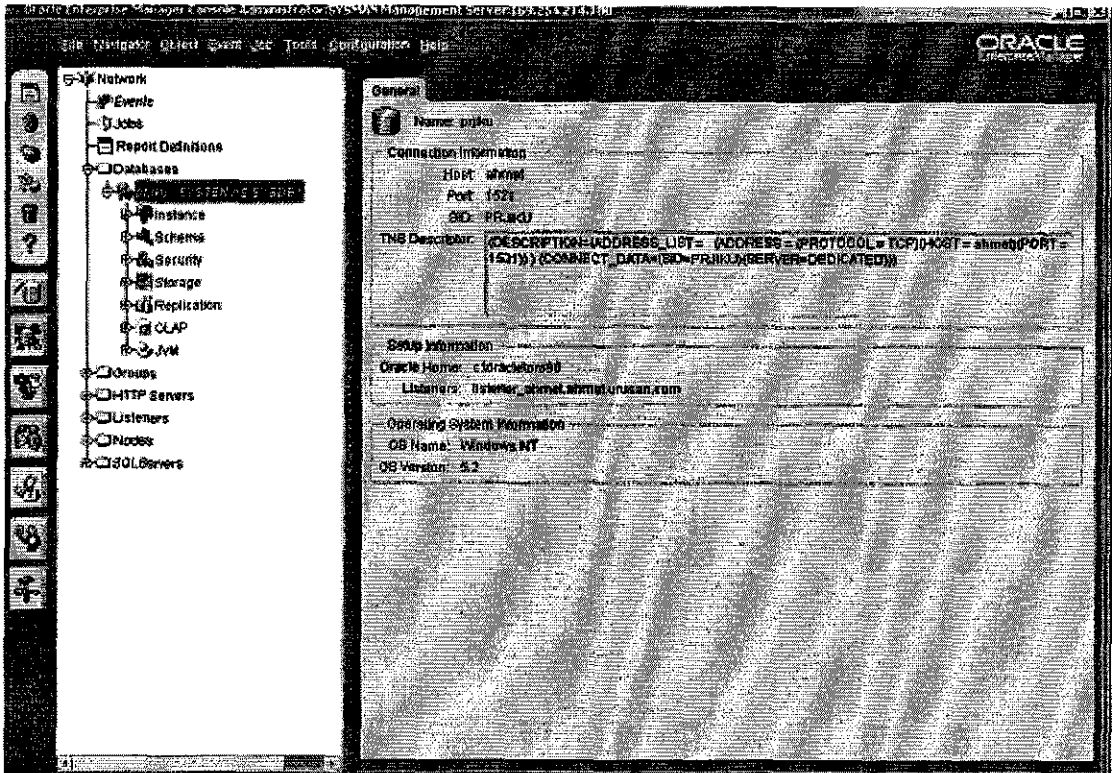
Değişiklikler dosyasını (redo log) diske yazmaktan sorumludur. Değişiklikler dosyası verileri, SGA'daki geri alma tamponunda üretilmektedir. Veritabanı işlemi kaydedilir ve değişiklikler tamponu dolarsa, LGWR bilgileri etkin olan bir veritabanı dosyasına yazar. Eğer veritabanında birden fazla işlem saklama dosyası varsa, LGWR her birine aynı anda yazma yapar. Her Oracle (Instance) oturumu için bir tane LGWR işlemi vardır.

1.2.8.3 KONTROL NOKTASI (CHECKPOINT, CKPT)

Belirli zamanlarda, SGA'daki bütün deęişen veritabanı tamponları, DBWR tarafından diske, veri dosyalarına yazılmaktadır. Zaman içindeki bu kritik anlara "kontrol noktası-checkpoint" denilir. Kontrol noktası işlemi, bu anlarda DBWR işlemini uyararak tüm veri ve kontrol dosyalarını güncelleme işlemi yapmasını sağlar.

1.2.8.4 SİSTEM MONİTÖRÜ (SYSTEM MONITOR-SMON)

Arızalanan bir Oracle oturumu yeniden başlatıldığında, SMON işlemi otomatik olarak bir kurtarma işlemi yapar. Birden fazla veritabanı oturumunun olduğu ortamlarda, SMON aynı zamanda bozulan sistemler içinde, oturum kurtarma işlemi yapmaktadır. Bunun yanısıra, kullanılmayan geçici parçaları (temporary segment) temizlemekte ve herhangi bir problemten dolayı bozulan işlemleride kurtarmaktadır. Bozulan işlemlerin veritabanı işlemleride, SMON tarafından tablo boşluğu veya veri dosyası tekrar aktif hale getirildikten sonra kurtarılır. Son olarak SMON, veritabanında daha fazla boş yer açılmasını diye, boş genişlemeleri, DOS'taki defragmentasyon gibi birleştirmektedir.



Şekil 1.11 Oturum içerisinde bağlantı ve dinleyici adresleri

1.2.8.5 İŞLEM ANALİZCİSİ (PROSESS MONITOR-PMON)

Herhangi bir kullanıcı işlemi bozulduğunda, o işlemin kurtarılmasını (recovery) yapmaktadır. Kurtarıma ile, kaydedilmemiş değişiklikleri eski haline getirme (rollback), işletimi kesilen görevin tuttuğu kilitleri kaldırma, duraklayan görev için ayrılan SGA kaynaklarını serbest bırakma anlaşılmaktadır. İşlemin kullandığı kaynakları temizler ve belleği boşaltır, ilgili sunucu işlemlerini de kontrol ederek, bir arıza oluşmuş ise bunları yeniden başlatır.

1.2.8.6 YEDEKLEYİCİ (ARCHIVER-ARC)

Aktif geri alma dosyalarını (redo log file), dolduklarında, yedekleme dizinine kopyalar. Bu işlemin aktif olması için, veritabanı ARCHIVELOG modunda çalışıyor olmalı ve otomatik arşivleme açık olmalıdır. Bu işlem çalışırken tablo uzaylarını (tablespace) yedekler, aynı zamanda günlük dosyaların otomatik olarak arşivlenmesine olanak tanır.

1.2.8.7 KURTARMA İŞLEMİ (RECOVERER-RECO)

Dağıtık veritabanında, sistem veya ağ hatalarından dolayı bekleyen veritabanı işlemlerini düzenler. Belirli aralıklarla, yerel RECO uzaktaki veritabanına bağlanıp, yereldeki dağıtık veritabanı işlemleri ile ilgili kayıt ve eski haline getirme işlemlerini yapmaktadır.

1.2.8.8 ARACI İŞLEM (DISPATCHER-DNNN)

Çoklu ortamlarda isteğe bağlı olarak çalıştırılırlar. Her iletişim protokolü için en az bir (D000,...,Dnnn) yaratılmaktadır. Her Aracı işlemi kullanıcı işlemlerinden gelen istekleri server işlemlerine yönlendirmekte ve gelen cevaplarında uygun kullanıcılara tekrar döndürmekten sorumludur.

1.2.8.9 KİLİT (LOCK-LCKN)

Oracle Paralel Server'da, oturumlar arasındaki kilitleme işlemleri için kullanılırlar.

1.2.9 VERİ SÖZLÜĞÜ (DATA DICTIONARY)

Veri sözlüğü, veritabanı hakkında salt okunabilir nitelikte bilgiler (veritabanındaki kullanıcılar, kullanıcıların hakları, veritabanı nesnelere, tabloların kısıtlamaları hakkında bilgiler gibi) için bir grup tablo ve görüntüden oluşan yapıdır. Veri sözlüğü Oracle veritabanının en önemli bölümüdür. Veritabanı hakkındaki tüm bilgiler, veri sözlüğünden SQL komutları yazarak öğrenilebilir. Veri sözlüğü, veritabanı

kurulurken standart olarak oluşturulur ve yapılan değişiklikler otomatik olarak veri sözlüğüne yansıtılır. Örneğin, bir kullanıcı bir tabloya bir alan eklerse, veri sözlüğünde o tabloya ilişkin yapı, kullanıcı farketmeden değiştirilir, böylece veri sözlüğünden veritabanı hakkında sürekli olarak güncel ve sağlıklı bilgi almak mümkün olur. Veri sözlüğünün sahibi SYS kullanıcısıdır.

Veri sözlüğünden, veritabanındaki her kullanıcı, hakkına göre faydalanabilmektedir. Veri sözlüğü içerisindeki bilgiler, SQL komutlarıyla sorgulanıp görülebilir. Fakat hiçbir kullanıcı veri sözlüğünü değiştiremez, ekleme yapamaz ve kayıt silemez. Bunu yapabildiği takdirde, veritabanının bütünlüğü bozulur. İlişkisel Veritabanı Yönetim Sistemi, veritabanında olan değişiklikleri otomatik olarak anında veri sözlüğüne yansıtmaktadır.

1.2.9.1 VERİ SÖZLÜĞÜ TABLOLARI (DATA DICTIONARY TABLES)

Veritabanı yaratıldığı anda, veri sözlüğü tablolarında otomatik olarak (CREATE DATABASE cümlesi çalışırken) oluşturulur. Çünkü, bu tablolar, daha sonra kullanıcılar tarafından oluşturulacak nesnelere için gereklidir. Veritabanı oluşturulurken oluşturulan veri sözlüğü tabloları, SYS adlı kullanıcıya aittir. Veri sözlüğü tablolarının içindeki bilgiler, kullanıcılar için çok anlaşılır olmadığından dolayı çok az sorgulanmaktadır.

1.2.9.2 VERİ SÖZLÜĞÜ GÖRÜNTÜLERİ (DATA DICTIONARY VIEWS)

Veri Sözlüğü görüntüleri, veritabanı hakkında bilgileri tutan ve kullanıcılar tarafından kolay anlaşılabilir bir şekilde olan sanal tablolardır. Veri sözlüğü görüntülerinin sahibi SYS kullanıcısıdır ve bu görüntülerde veritabanı ile ilgili, anlık olarak değişebilen bilgiler tutulmaktadır. Mesela, anlık kullanıcı sayısı ve anlık kilit (lock) sayısı gibi. Veri sözlüğü görüntülerine her kullanıcı tarafından erişilebilmektedir. Fakat, her kullanıcı farklı görüntülere erişme hakkına sahiptir. Veri sözlüğü görüntüleri, başlarındaki ön eklerine göre üç gruba ayrılır:

USER_XXX : Veritabanına o an bağlı olan kullanıcının sahip olduğu nesnelere ile ilgili görüntülerdir.

ALL_XXX : Veritabanına o an bağlı olan kullanıcının sahip olduğu ve başkalarına ait olan ve kendisine kullanma hakkı verilmiş olan tüm nesnelere ilgili görüntülerdir.

DBA_XXX : DBA (veritabanı yöneticisi) veya DBA hakkına sahip kullanıcıların görebileceği görüntülerdir.

Örnek : Veri sözlüğü görüntülerinin listesi, SYS veya SYSTEM kullanıcısı tarafından aşağıdaki şekilde alınabilmektedir :

```
SELECT * FROM V$FIXED_TABLE;
```

Örnek : DICT görüntüsünün alanlarını listelleyiniz.

```
DESC DICT
```

Name	Null ?	Type
-----	-----	-----
TABLE_NAME		VARCHAR2 (30)
COMMENTS		VARCHAR2 (2000)

Tablo 1.1 Sık kullanılan görüntüler

Görüntü İsimleri	Açıklama
USER_TABLES	Kullanıcıların sahip olduğu tabloların yapısını gösteren görüntüdür.
USER_SEQUENCES	Kullanıcının sahip olduğu sıralar hakkında bilgi içeren görüntüdür.
USER_VIEWS	Kullanıcının sahip olduğu görüntüler hakkında bilgi içeren görüntüdür.
ALL_OBJECTS	Kullanıcının erişebileceği bütün nesnelere hakkında bilgi içeren görüntüdür.
DBA_USERS	Veritabanındaki bütün kullanıcılar hakkında bilgi içeren görüntüdür.
DBA_TABLESPACES	Veritabanındaki bütün tablo uzayları hakkında bilgi içeren görüntüdür.

Örnek : USER_OBJECTS görüntüsünde bağlı olunan kullanıcının bütün nesnelere hakkında bilgi :

DESC USER_OBJECTS

Name	Null ?	Type
-----	-----	-----
OBJECT_NAME		VARCHAR2 (128)
SUBOBJECT_NAME		VARCHAR2 (30)
OBJECT_ID		NUMBER
DATA_OBJECT_ID		NUMBER
OBJECT_TYPE		VARCHAR2 (18)
CREATED		DATE
LAST_DDL_TIME		DATE
TIMESTAMP		VARCHAR2 (19)
STATUS		VARCHAR2 (7)
TEMPORARY		VARCHAR2 (1)
GENERATED		VARCHAR2 (1)
SECONDARY		VARCHAR2 (1)

Object_name = nesnenin ismi

Object_id = veritabanı tarafından nesneye verilen numara

Object_type = nesnenin tipi (tablo, indeks gibi)

Created = nesnenin yaratıldığı tarih

Last_ddl_time = o nesneyle ilgili veritabanındaki son veri tanımlama dili Data Definition Language) kullanım tarihi (create, alter gibi komutlardan biri)

Timestamp = o nesneyle ilgili son DDL zamanı

Status = Nesnenin durumunu yani geçerli olup olmadığını göstermektedir.

Örnek : SCOTT kullanıcısının bütün nesnelerini görmesi

```
SELECT * FROM USER_OBJECTS
```

Örnek : SCOTT kullanıcısını nesne tipi "INDEX" olan bütün nesnelerini listeleyiniz.

```
SELECT * FROM USER_OBJECTS  
WHERE OBJECT_TYPE = 'INDEX';
```

Örnek : Bir kullanıcının kendine ait bütün tabloların adlarını listelemesi

```
SELECT TABLE_NAME FROM USER_TABLES
```

Örnekl : Bir kullanıcının tüm indeks bilgilerini listelemesi

```
SELECT INDEX_NAME, TABLE_NAME, TABLESPACE_NAME  
FROM ALL_INDEXES
```

1.2.10 BAŞLANGIÇ DOSYASI PARAMETRELERİ (INITSID.ORA PARAMETERS)

Başlangıç parametreleri dosyası (init.ora), ilişkisel veritabanı yönetim sisteminin temel konfigürasyon yapısıdır. Başlangıç parametresi dosyası'nda temel konfigürasyon bilgileri vardır ve bu değerlerin her birisi veritabanının işlemlerini kontrol etmektedir. Bu dosya, text dosyası yapısındadır ve Windows NT sistemlerde \$ORACLE_HOME\DATABASE dizininde bulunmaktadır. Dosyanın standart ismi INITSID.ORA dır. SID, veritabanının ismidir. Her Oracle veritabanının ve veritabanı oturumunun (Instance) kendine has bir init.ora'sı vardır.

Init.ora dosyası, veritabanı çalıştırılmadan önce, yeni bir oturum yaratılmadan önce veya kontrol dosyaları okunmadan önce okunur. Bu dosyanın içindeki değerler, veritabanının özelliklerini belirlemektedir. Bu değerler, shared pool, buffer cache, redo log buffer, memory allocations değerlerini, okunacak kontrol dosyalarını, otomatik olarak açılacak geri alma parçalarını, belirlemektedirler. Başlangıç parametresi dosyalarında yapılan değişiklikler, veritabanı kapatılıp açılmadan geçerli olmazlar. Aşağıda PRJIKU veritabanının konfigürasyon bilgilerini içeren init.ora dosyasının yol bilgisi ve içeriği görülmektedir.

C:\oracle\admin\prjiku\pfile

```
#####  
#####  
# Copyright (c) 1991, 2001 by Oracle Corporation  
#####  
#####  
  
#####  
# Cache and I/O  
#####  
db_block_size=4096  
db_cache_size=33554432  
  
#####  
# Cursors and Library Cache
```

```

#####
open_cursors=300

#####
# Diagnostics and Statistics
#####
background_dump_dest=C:\oracle\admin\prjiku\bdump
core_dump_dest=C:\oracle\admin\prjiku\cdump
timed_statistics=TRUE
user_dump_dest=C:\oracle\admin\prjiku\udump

#####
# Distributed, Replication and Snapshot
#####
db_domain=""
remote_login_passwordfile=EXCLUSIVE

#####
# File Configuration
#####
control_files=( "C:\oracle\oradata\prjiku\CONTROL01.CTL",
"C:\oracle\oradata\prjiku\CONTROL02.CTL",
"C:\oracle\oradata\prjiku\CONTROL03.CTL" )

#####
# MTS
#####
dispatchers="(PROTOCOL=TCP) (SER=MODESE) ",
"(PROTOCOL=TCP) (PRE=oracle.aurora.server.GiopServer) ",
"(PROTOCOL=TCP) (PRE=oracle.aurora.server.SGiopServer) "

#####
# Miscellaneous
#####
compatible=9.0.0
db_name=prjiku

#####
# Network Registration
#####
instance_name=prjiku

#####
# Pools
#####
java_pool_size=33554432
large_pool_size=1048576
shared_pool_size=33554432

#####
# Processes and Sessions
#####
processes=150

#####
# Redo Log and Recovery
#####
fast_start_mttr_target=300

#####
# Sort, Hash Joins, Bitmap Indexes

```

```
#####  
sort_area_size=524288
```

```
#####  
# System Managed Undo and Rollback Segments  
#####  
undo_management=AUTO  
undo_tablespace=UNDOTBS
```

1.2.11 İZ (TRACE) DOSYALARI

Bütün Oracle veritabanlarında sistem mesajlarını, hataları ve temel olayları kaydedebilmek için en az bir dosya bulunmaktadır. Bu dosya “sidALTR.LOG” (sid veritabanının ismidir), init.ora parametresi BACKGROUND_DUMP_DEST’in belirttiği dizindedir. Veritabanı ile ilgili problemlerin incelenmeye başlanacağı ilk yer bu dosyadır. Temel problemler, her zaman veritabanı tarafından otomatik olarak buraya kaydedilir.

Artalan ve kullanıcı işlemleri problem olduğunda, kendi trace dosyalarını oluştururlar. Artalan işlemleri trace dosyaları, BACK_GROUND_DUMP_DEST adresinde bulunur. USER_DUMP_DEST parametresini kullanmakla kullanıcılara ait değişik trace dosyaları oluşturmak mümkün olmaktadır. Artalan işlem trace dosyaları “sidPROC.TRC” diye isimlendirilirler. Burada Sid, veritabanı adı, PROC ise artalan işleminin (DBWR, LGWR, SMON, PPMON gibi) adı olmaktadır. Aşağıda PRJIKUSMON.TRC dosyasının yol bilgisi ve içeriği görülmektedir.

```
Dump file C:\oracle\admin\prjiku\bdump\prjikuSMON.TRC  
Sat Mar 12 18:12:07 2005  
ORACLE V9.0.1.1.1 - Production vsnsta=0  
vsnsql=10 vsnxtr=3  
Windows 2000 Version 5.2 , CPU type 586  
Oracle9i Enterprise Edition Release 9.0.1.1.1 - Production  
With the Partitioning option  
JServer Release 9.0.1.1.1 - Production  
Windows 2000 Version 5.2 , CPU type 586  
Instance name: prjiku
```

Redo thread mounted by this instance: 1

Oracle process number: 6

Windows thread id: 3040, image: ORACLE.EXE

```
*** SESSION ID:(5.1) 2005-03-12 18:12:07.000  
*** 2005-03-12 18:12:07.000  
SMON: following errors trapped and ignored:  
ORA-00604: error occurred at recursive SQL level 1  
ORA-01001: invalid cursor  
ORA-00601: cleanup lock conflict
```

Dump file C:\oracle\admin\prjiku\bdump\prjikuSMON.TRC
Tue May 31 14:07:49 2005
ORACLE V9.0.1.1.1 - Production vsnsta=0
vsnsql=10 vsnxtr=3
Windows 2000 Version 5.2 , CPU type 586
Oracle9i Enterprise Edition Release 9.0.1.1.1 - Production
With the Partitioning option
JServer Release 9.0.1.1.1 - Production
Windows 2000 Version 5.2 , CPU type 586
Instance name: prjiku

Redo thread mounted by this instance: 1

Oracle process number: 6

Windows thread id: 2980, image: ORACLE.EXE

*** SESSION ID:(5.1) 2005-05-31 14:07:49.000
*** 2005-05-31 14:07:49.000
SMON: following errors trapped and ignored:
ORA-00604: error occurred at recursive SQL level 1
ORA-01001: invalid cursor
ORA-00601: cleanup lock conflict
Dump file C:\oracle\admin\prjiku\bdump\prjikuSMON.TRC
Wed Jun 15 10:23:52 2005

ORACLE V9.0.1.1.1 - Production vsnsta=0
vsnsql=10 vsnxtr=3
Windows 2000 Version 5.2 , CPU type 586
Oracle9i Enterprise Edition Release 9.0.1.1.1 - Production
With the Partitioning option
JServer Release 9.0.1.1.1 - Production
Windows 2000 Version 5.2 , CPU type 586
Instance name: prjiku

Redo thread mounted by this instance: 1

Oracle process number: 6

Windows thread id: 2880, image: ORACLE.EXE

*** SESSION ID:(5.1) 2005-06-15 10:23:52.000
*** 2005-06-15 10:23:52.000
SMON: following errors trapped and ignored:
ORA-00604: error occurred at recursive SQL level 1
ORA-01001: invalid cursor
ORA-00601: cleanup lock conflict
Dump file C:\oracle\admin\prjiku\bdump\prjikuSMON.TRC
Thu Jun 23 13:06:33 2005

ORACLE V9.0.1.1.1 - Production vsnsta=0
vsnsql=10 vsnxtr=3
Windows 2000 Version 5.2 , CPU type 586
Oracle9i Enterprise Edition Release 9.0.1.1.1 - Production
With the Partitioning option
JServer Release 9.0.1.1.1 - Production
Windows 2000 Version 5.2 , CPU type 586
Instance name: prjiku

Redo thread mounted by this instance: 1

Oracle process number: 6

Windows thread id: 3012, image: ORACLE.EXE

```
*** SESSION ID:(5.1) 2005-06-23 13:06:32.000
*** 2005-06-23 13:06:32.000
SMON: following errors trapped and ignored:
ORA-00604: error occurred at recursive SQL level 1
ORA-01001: invalid cursor
ORA-00601: cleanup lock conflict
```

1.2.12 VERİ TABANI ŞEMA NESNELERİ

Şema nesneleri, mantıksal veri depolama yapıları olarak bilinir. Veritabanı üzerinde kullanıcının belirli işleri yapabilmesi için tanımlanan bu yapılar tablolar, görüntüler, sıralar, eşanlamlar, indeksler, kümeler, veritabanı bağlantıları, prosedürler, fonksiyonlar, ve paketlerdir. Bir şema ise, bu nesnelere oluşturduğu gruptur.

1.2.12.1 KÜME (CLUSTER)

Aynı anda sorgulanan birden fazla tablonun bir arada kaydedilmesine "cluster" denir. Bu yapı, beraber sorgulanan tablolarda hız kazanmak için çok önemlidir. Örneğin "ogrsicil" tablosu ile "notkarti" tablosunun (bu tabloların yapısı için SQL bölümüne bkz.) her ikisi de ortak olarak "ogrno" ve "ogrno" alanlarını içermekte. Yani "ogrsicil" tablosundaki "ogrno", "notkarti" tablosundaki "ogrno" alanına karşılık geliyor. Burada oluşturulacak bir cluster'da her iki tablo, veritabanında yan yana getirilerek aynı veri bloğu içerisine kaydedilir. Böylece daha hızlı erişim sağlanır.

1.2.12.2 İNDEKS (INDEX)

İndeksler, tablo ve cluster'lar için kullanılan veri tabanı nesnelere aittir. Burada amaç, aranan bir kayda daha hızlı erişimdir. Özellikle, üzerinde çok arama yapılan alan veya alanlar üzerinde indeks oluşturmak çok etkilidir. İndeks oluşturulduğunda ilgili tablonun kayıtları yer değiştirmez. Sadece ilgili kayıtların kayıt numaraları olarak adlandırılan "**rowid**" ler alınarak sıralama yapılır.

Bir tablo üzerinde oluşturulabilecek indeks sayısı, sütunların kombinasyonları farklı olduğu müddetçe sınırsızdır. Bir sütun, diğer sütunlarla değişik kombinasyonlarda kullanıldığı müddetçe, birden fazla indeks içerisinde yer alabilir. Aynı sütun kombinasyonlarının indeksi, farklı indeks ismi kullanarak oluşturulmaya çalışılsa bile gerçekleştirilemez.

İndeksleme işleminin gerçekleştirilmesinde Oracle çoğunlukla B-tree indeksleme metodunu kullanır. Bunun yanında "hash cluster indexes", "reverse key indexes" ve "bitmap indexes" yöntemlerini de kullandığı bilinmektedir. İndeksler, mantıksal ve fiziksel olarak oluşturuldukları tablodan bağımsızdırlar. Eğer bir indeks silinirse, ilgili tablo zarar görmez, çalışmaya devam eder. Fakat indeks olmadığı için veri erişim süresi artacaktır.

Oracle, bir indeks oluşturulduğunda onu otomatik olarak kullanmaya başlar ve indeksin oluşturulduğu tablodaki silme, güncelleme ve ekleme işlemleri indekse otomatik olarak yansıtılır.

1.2.12.3 ROL (ROLE)

Oracle veritabanında her nesnenin ait olduğu bir kullanıcı vardır. Bir kullanıcı bir başka kullanıcının nesnelere üzerinde işlem yapmak isterse, buna hakkı olması gerekir. Bir nesne üzerinde işlem yapabilme yetkisine hak denir. Örneğin, veritabanına bağlanma, tablo oluşturma, bir başkasına ait tablodan kayıt listeleme, bir başkasının prosedürünü çalıştırma birer haktır. Bu haklar kullanıcılara atanmak suretiyle, kullanıcıların bu işlemleri gerçekleştirmeleri sağlanır. Hakların kullanıcılara atanması iki şekilde olabilir. Birinci olarak bir tabloya kayıt ekleme, kayıt silme vb. haklar bir kullanıcıya ya da kullanıcılara ayrı ayrı atanır. İkinci şekilde ise verilmek istenen haklar bir rol altında birleştirilir ve bu rol istenen kullanıcılara aktarılır.

Haklar, "sistem hakları" ve "nesne hakları" olmak üzere ikiye ayrılır. Sistem hakları, veritabanı ile ilgili olarak önceden tanımlanmış rollerdir. Oracle'da 60'tan fazla sistem hakkı tanımlanmıştır. Nesne hakları ise, veri tabanı nesnelere üzerinde işlem yapma haklarıdır. "Create Table", "Create TableSpace", "Drop Any Index" gibi haklar sistem haklarına örnek olarak verilebilir. Nesne hakları ise, "Select", "Insert", "Update", "Delete", "Alter", "Index", "Execute", "References", "All" olmak üzere 8 çeşittir. "All", ayrı bir hak olarak adlandırılmaz. Diğer tüm hakları kapsar. Bu haklar sırayla kayıt seçme, kayıt güncelleme, kayıt silme, nesnelere yapısını değiştirme, indeks oluşturma, alt program çalıştırma, yabancı anahtar tanımlayabilme işlemlerini içerirler.

1.2.12.4 VERİTABANI NESNELERİYLE İLGİLİ KOMUTLAR

CREATE TABLE

Her ne kadar veritabanınındaki tabloları oluştururken ORACLE'nin "Create" özelliğinden faydalanılmış olsa da, yine de burada komut satırlarıyla tablo yaratılması ve yaratılan bu tabloda bir takım kısıtlamaların ne şekilde verilebileceğini basit bir örnekle incelemekte fayda vardır.

```
CREATE table ogrsicil
(Ogrno VARCHAR2 (10) NOT NULL,
Ad -----VARCHAR2 (15),
Soyad --VARCHAR2 (15),
Dogumyeri -NUMBER (2),
CONSTRAINT cst_Ogrno PRIMARY KEY (Ogrno)
CONSTRAINT cst_dogumyeri FOREIGN key (dogumyeri) REFERENCES
iller (il))
```

Bu komut "ogrsicil" adında bir tablo oluşturuyor. Tablonun dört alanı var. Bunlardan biri özel anahtar (Primary Key) olarak tanımlanmış. Bu alanın değeri boş olamaz ve tabloda aynı iki değer bulunamaz.. Bir tablo ile başka bir tablo arasında ilişki kurulacaksa bu ana tabloda PRIMARY KEY tanımıyla, diğer tabloda FOREIGN KEY tanımıyla yapılır. Yani doğum yeri alanı için buradaki tablomuz ana tablo değil. Çünkü, aynı yerde doğmuş kişilerin kaydı tablomuzda olabilir. Böylece aynı il kodu birden fazla kayıta yer alabilir. İlişki kurulan diğer tablonun adını yukarıdan çıkarabiliriz. Bu tablo "iller" isminde. Her il bu tabloda kayıtlı ve her birinin bir kodu var. Bu tabloda bir ilin kaydı iki sütunda yer alamaz. Bu yüzden "il" alanı "iller" tablosu için PRIMARY KEY olarak tanımlanmış.

PRIMARY KEY, NOT NULL gibi bildirimler, kısıtlamalar olarak tanımlanır. Bir tabloda farklı şekilde kısıtlamalar yapılabilir. Kısıtlama için kullanılan bildirimler 5 adettir.

- ✓ NOT NULL : Bu kısıtlamanın konulduğu bir alanın mutlaka bir değeri olmak zorundadır.
- ✓ UNIQUE : Bu kısıtlamanın konulduğu bir alan, bir değerden sadece bir adet içerebilir. Bu kısıtlamaya sahip alan, NOT NULL ile tanımlanmadıysa, NULL değer alabilir.

- ✓ PRIMARY : Bu kısıtlama, bu alanın özel anahtar olmasını sağlar. Bu alan boş geçilemez ve yine tablo içerisinde bir değer iki kayıta bulunamaz.
- ✓ FOREIGN KEY : Bu kısıtlama, başka bir tablonun PRIMARY KEY olan alanıyla ilişki kurmak için gerekir.
- ✓ CHECK : Bu kısıtlama ile birlikte bir şart kullanılır. Tabloya kayıt girilirken, değerlerin bu şarta uyması beklenir.

ALTER TABLE

Daha önceden oluşturulmuş bir tablonun yapısını değiştirmek için ALTER TABLE komutu kullanılır.

```
ALTER TABLE tablo_adi
```

```
ADD | MODIFY | DROP (<sütun ismi> veri tipi <sütun kısıtlaması>)
```

```
ENABLE ifade1
```

```
DISABLE ifade2
```

ALTER komutuyla tablolara yeni bir alan ve kısıtlama eklenebilir, var olan alan ve kısıtlamaların durumu değiştirilebilir veya tablodan kısıtlamalar düşürülebilir. Fakat ALTER komutuyla var olan sütunlar düşürülemez. Bir alanın değeri büyütülebilir, fakat küçültülemez.

```
ALTER TABLE ogrsicil
```

```
ADD CONSTRAINT xBolum
```

```
FOREIGN KEY (Bolum)
```

```
REFERENCES userb.bolumler (BolKod);
```

Yukarıdaki örnekte ogrsicil adlı tablonun yapısı değiştiriliyor. Bir başka kullanıcının bir tablosu ile ilişki kuruluyor.

```
ALTER TABLE acilandersler
```

```
MODIFY dersadi varchar2 (40);
```

Bu örnekte de acilandersler tablosunun bir alanının uzunluğu değiştiriliyor.

DROP TABLE

Daha önceden oluşturulmuş bir tablonun düşürülmesi için kullanılan komuttur.

Kullanımı:

```
DROP TABLE <tablo_ismi> [CASCADE CONSTRAINTS]
```

Köşeli parantez içerisindeki tanım kullanılırsa, master-detay ilişkili tablolarda master tablo düşürülünce detay tabloların da otomatik olarak düşürülmesi sağlanır. Eğer bu

seçenek kullanılmazsa, diğer tablolarla ilişkisi bulunan bir tablo silinemez. Ancak bu ilişkiler kaldırıldıktan sonra tablo silinebilir.

DROP TABLE ogrsicil CASCADE CONSTRAINT

CREATE VIEW

Görüntü oluşturmak için bir SELECT cümlesi kullanmak gerekir. Bir görüntü bir ya da daha fazla tablodan oluşturulabileceği gibi, bir başka görüntüden de oluşturulabilir. Aşağıdaki örnekte üç farklı kullanıcının aynı yapıdaki tabloları birleştirilip bir görüntü oluşturuluyor.

```
CREATE VIEW ogrsicil
AS SELECT * from userb.ogrsicil
UNION ALL
SELECT * FROM userm.ogrsicil
UNION ALL
SELECT * FROM usera.ogrsicil;
```

CREATE TABLESPACE

Tablo uzayı, kullanıcılara ait olan nesnelerin veritabanında mantıksal olarak tutulduğu yere denmektedir. Bir tablo uzayı oluştururken, bu tablo uzayının verilerinin hangi veri dosyasına konulacağı ve bu dosyanın dizini ile büyüklüğü bildirilmelidir.

```
CREATE TABLESPACE userb
DATAFILE 'c:\orasql\userb.dat' SIZE 30M
DEFAULT STORAGE (INITIAL 30K NEXT 70K
MINEXTENTS 1 MAXEXTENTS 999)
ONLINE;
```

SIZE bildirisi, veri dosyasının diskte kaplayacağı yeri belirler. Burada 10M, 5K gibi değerler girilebilir. INITIAL bildirisi, tablo uzayı oluşturulduğunda, ilk alacağı genişleme'nin büyüklüğünü belirler. NEXT, tablo uzayı oluşturulduktan sonra alacağı genişlemelerin büyüklüğünü belirler. MINEXTENTS, tablo uzayı oluşturulduğunda ilk olarak alacağı minimum genişleme sayısının belirtildiği bölümdür. MAXEXTENTS, bir tablo uzayının ilk olarak aldığı genişleme de dahil olmak üzere alabileceği maksimum genişleme sayısının belirtildiği bölümdür.

CREATE USER

"CREATE USER" komutunu SYS ve SYSTEM kullanıcıları standart olarak kullanabilir. Bu hak diğer kullanıcılara da verilebilir. Her kullanıcının nesnelere tutmak için bir tablo uzayı oluşturmak sistemin performansı açısından gereklidir. Kullanıcı oluşturulurken bu tablo uzayı o kullanıcıya atanır.

```
CREATE USER usera
IDENTIFIED BY a
DEFAULT TABLESPACE usera
QUOTA UNLIMITED ON usera
```

Yukarıdaki örnekte "usera" adında bir kullanıcı oluşturuluyor. Kullanıcının şifresi IDENTIFIED BY ile "a" olarak bildiriliyor. Kullanıcının kendi nesnelere oluşturacağı tablo uzayı için ise "usera" tablo uzayı bildiriliyor. Kullanıcının bu tablo uzayındaki tüm alanı kullanabileceği QUOTA UNLIMITED ile belirleniyor.

CREATE ROLE

Rol, veritabanındaki hakların toplanmış haline denir. Veritabanı yöneticisi rolleri kullanarak sistemin güvenliğini daha kolay sağlayabilir. Roller, Oracle tarafından, önceden tanımlanmış roller ve kullanıcı tanımlı roller olarak iki şekilde düşünülebilir. Oracle tarafından önceden tanımlanan roller beş tanedir:

Tablo 1.2 Oracle tarafından önceden tanımlanan roller

Rol	Atanmış Haklar
CONNECT	ALTER SESSION, CREATE CLUSTER, CREATE DATABASE LINK, CREATE SEQUENCE, CREATE SESION, CREATE SYNONYM, CREATE TABLE, CREATE VIEW
RESOURCE	CREATE CLUSTER, CREATE PROCEDURE, CREATE SEQUENCE, CREATE TABLE, CREATE TRIGGER
DBA	"WITH ADMIN OPTION" ile birlikte bütün sistem hakları
EXP_FULL_DATABASE	SELECT ANY TABLE, BACKUP ANY TABLE, SYS.INCVID, SYS.INCFIL ve SYS.INCEXP tablolarına INSERT, UPDATE ve DELETE hakkı
IMP_FULL_DATABASE	BECOME USER, WRITEDOWN

```
CREATE ROLE tabloma_bak
GRANT SELECT ON ogrsicil TO tabloma_bak
```

Yukarıdaki örnekte tabloma_bak adında bir rol oluşturuluyor. Daha sonra bu role ogrsicil tablosu üzerinde listeleme işlemi yapma hakkı veriliyor. Böylece bu rolün atandığı kullanıcı ogrsicil tablosu üzerinde "SELECT" komutunu çalıştırabilecek.

CREATE INDEX

Daha öncede bahsedildiği gibi indeks tablodaki kayıtlara daha hızlı erişim için kullanılan nesnelere. Bir indeks oluşturabilmek için "CREATE ANY INDEX" sistem hakkına sahip olmak gerekir. İndeks, bir tablonun bir alanı üzerinde tanımlanabileceği gibi birden fazla alan üzerinde de tanımlanabilir.

```
CREATE INDEX indeks1 ON ogrsicil (ogmo,ad)
```

CREATE SEQUENCE

Sıra, sıralı olarak artan alanlar için veritabanında tutulan nesnedir. (Bkz.: 1.2.12.6)

Örneğin birden başlayan ve birer birer artan bir sıra yaratmak için:

```
CREATE SEQUENCE sıra1 START WITH 1 INCREMENT BY 1
```

CREATE ROLLBACK SEGMENT

Geri alma parçaları SELECT, INSERT, DELETE, UPDATE gibi komutlarla yapılan işlemlerin gerektiğinde geri alınabilmesi için veritabanında ayrılan alanlara denir.

(Bkz.: 1.2.12.5)

"usera" tablo uzayını kullanan, kayıt parametrelerinden ilk parçanın büyüklüğü 10M, sonraki genişlemelerin büyüklüğü 1M, minimum genişleme sayısı 2, maksimum genişleme sayısı 121 ve ortalama değeri 30M olan ve ismi "rol_seg" olan bir geri alma parçası şöyle oluşturulabilir:

```
CREATE ROLLBACK SEGMENT rol_seg
TABLESPACE "usera"
STORAGE ( INITIAL 10M
NEXT 1M
MINEXTENTS 2
MAXEXTENTS 121 OPTIMAL 30M)
```

GRANT

Sistem ya da nesne haklarının kullanıcılara veya rollere atanması için kullanılan komuttur.

GRANT DELETE ON ogrsicil TO userb

Yukarıdaki örnekte, userb kullanıcıasına, ogrsicil tablosunda silme yapma yetkisi verilir.

REVOKE

Sistem ya da nesne haklarının kullanıcılardan veya rollerden geri alınması için kullanılan komuttur.

REVOKE DELETE ON ogrsicil FROM userb

Yukarıdaki örnekte, userb kullanıcıısından, ogrsicil tablosunda silme yapma yetkisi geri alınır.

Veritabanı nesnelерinin yönetilmesiyle ilgili daha bir çok veri tanımlama dili komutu vardır. Bu komutların diğeri burada anlatılmayacaktır. Bir nesneyi oluşturmak için "CREATE", silmek için "DROP", değiştirmek için "ALTER" komutlarının kullanıldığı bilinmesi diğeri komutlar için bir temel teşkil edecektir. Komutlar içerisinde kullanılan parametreler Oracle yardım dokümanlarında ayrıntılı olarak elde edilebilmektedir.

1.2.12.5 GERİ ALMA PARÇASI (ROLLBACK SEGMENT)

Oracle veritabanının güvenliği açısından "Select", "Insert", "Update", "Delete" gibi işlemlerin yedeğini almaktadır. Alınan bu yedeklerin konulduğu yerlere geri alma parçası denir. Kullanıcı bu tip işlemleri yaptıktan sonra "Rollback" komutunu uygularsa, yaptığı değişiklikler geri alma parçalarından getirilir ve böylece kayıtlar eski haline dönmüş olur. Kullanıcı "Commit" komutunu verirse yaptığı değişiklikler geri alma parçalarından geri getirilemez. Her veritabanında bir ya da birkaç tane geri alma parçası olabilir.

1.2.12.6 SIRA (SEQUENCE)

Tablolardaki kayıtlar için otomatik sıra numarası verilmesi isteniyorsa sıra nesnesi kullanılabilir. Bu sıra numarası veritabanı tarafından otomatik olarak üretilir.

Özellikle çok kullanıcıli ortamlarda tekil olarak numara üretilmek istendiğinde çok

kullanılışlıdır. Birden fazla kullanıcı aynı anda böyle bir sayı üretmek isterse bunun program koduyla yapılması işlem hızını yavaşlatır. Çünkü bir kullanıcı, diğer kullanıcı işini bitirene kadar beklemek zorundadır. Sıra nesnesi bu işi otomatik olarak ve çok seri bir şekilde başarır.

Sıra numaraları, Oracle'da tanımlı 38 rakama kadar tamsayılarından oluşur. Bir sıra tanımlaması sıranın adını, artan ya da azalan olacağını, iki sayı arasındaki fark miktarını içerir. Oracle, tüm sıra numarası tanımlarını, SYSTEM tablo uzayının içerisindeki bir veri sözlüğü tablosuna kaydeder. SYSTEM tablo uzayı sürekli çalışır durumda olduğu için tüm sıra numaraları da aktiftir. Sıra numaraları tablolardan bağımsız olarak üretilir. Yani bir sıra numarası bir ya da daha çok tablo için kullanılabilir.

1.2.12.7 KAYITLI FONKSİYONLAR (STORED FUNCTION) VE KAYITLI PROSEDÜRLER (STORED PROCEDURE)

Bir grup SQL ya da PL/SQL komutunun belli bir işi gerçekleştirmek için bir araya getirilip veritabanına kaydedilmesi kayıtlı prosedürler ve fonksiyonlar sayesinde olur. Birden fazla uygulama programı içerisinde aynı işi yapan kodları sürekli yazmak yerine kayıtlı prosedür ve fonksiyonlar bir kere yazılır ve tüm uygulamalar tarafından kullanılır. SQL*Plus, Oracle Forms ya da Oracle Reports içerisinde bu prosedürler çağrılabilir. Prosedürler geriye birden fazla değer döndürebilirken, fonksiyonlar sadece bir değer döndürürler.

1.2.12.8 EŞANLAM (SYNONYM)

Eşanlam, bir tablo, görüntü, sıra, prosedür, fonksiyon ya da paket için "alias" olarak adlandırılan bir takma isimdir. Eşanlam bir takma isim olduğu için veri sözlüğü içerisindeki tanımının kapladığı yer haricinde, veritabanında yer kaplamaz.

Eşanlamlar güvenlik ve daha rahat kod yazma amacıyla kullanılırlar. Bir eşanlam kullanıldığında, ilgili nesnenin adı ve sahibi gizlenir ve SQL komutu içerisinde kullanımı kolaylaştırır.

Eşanlamlar, genel (Public) ve özel (Private) olarak tanımlanabilirler. "Genel" olarak tanımlanan eşanamlara tüm veritabanı kullanıcıları erişebilir. "Özel" olarak tanımlanan eşanamlara sadece ilgili nesnenin sahibi ve sahibi tarafından hak verilmiş bir başka kullanıcı erişebilir.

Bir nesnenin adı değiştirilmek istendiğinde ya da silinmek istendiğinde, bu nesneyi kullanan tüm uygulama programları değiştirilmek zorundadır. Oysa ki bu nesnenin bir eşanlamı oluşturulursa ve uygulama programları bu eşanlamı kullanırsa, eşanlam üzerinde yapılacak değişikliklerle bu silme vb. işlemlerden uygulama programlarının etkilenmesi önlenir.

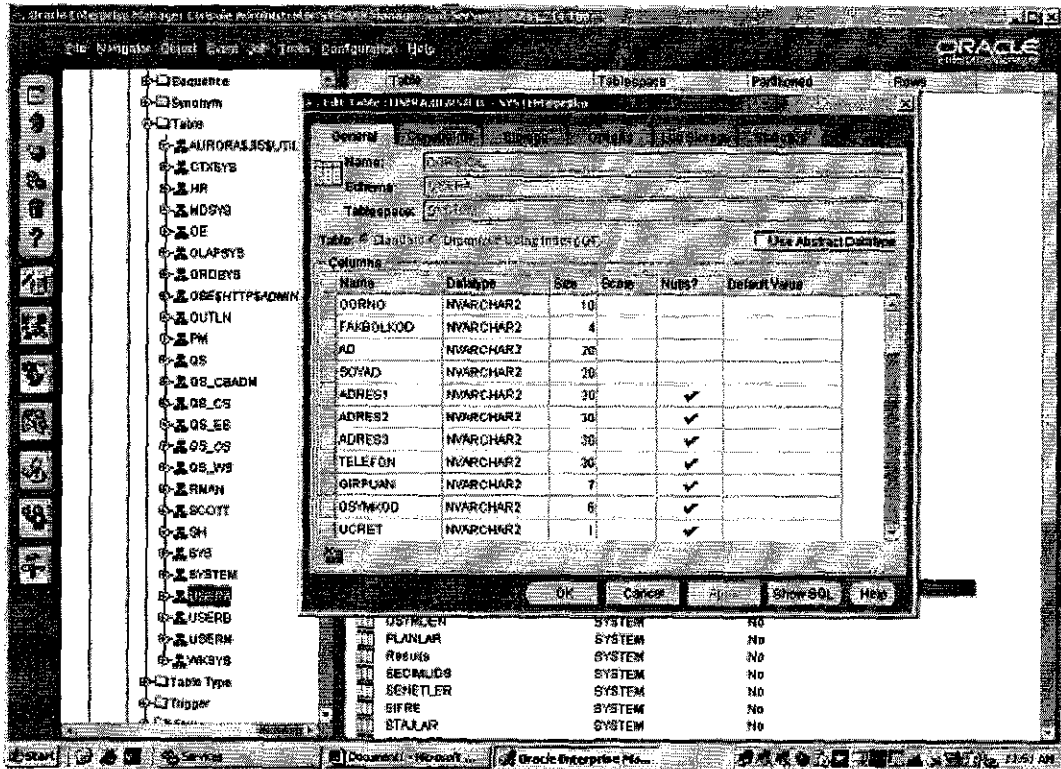
Bir kullanıcı bir başka kullanıcının nesnesini kullanmak istediğinde

"kullanıcı_adi.nesne_adi" şeklinde bir yazım kuralına uymak zorundadır. Eğer bir "genel" eşanlam tanımlanırsa, tüm kullanıcılar direk eşanlam ismini kullanarak işlemlerini gerçekleştirebilirler.

1.2.12.9 TABLO (TABLE)

İlişkisel Veri Tabanı Yönetim Sistemleri'nde veriler tablolar içerisinde yer alır. Her tablo bir isimle tanımlanır ve her biri bir "kayıt" olarak adlandırılan satırlar ile bu kayıtlardaki verilerin özelliklerini belirleyen sütunlardan oluşur. Her tablo bir ya da daha fazla sütuna sahip olabilir. Her sütunun bir adı ve veri tipi vardır.

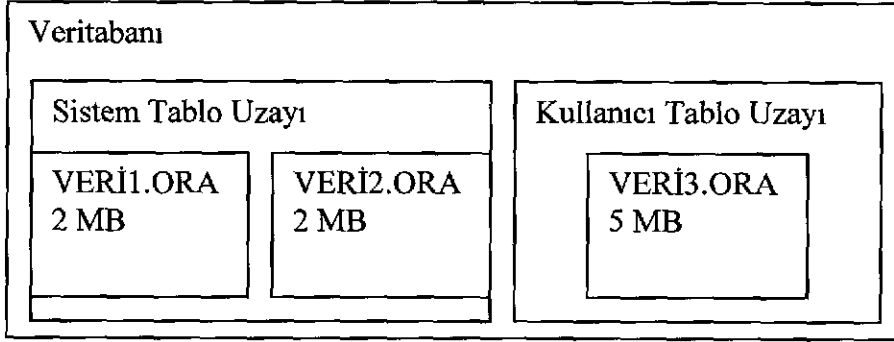
Bir tablo oluşturulduğunda Oracle verileri depolamak için bir tablo uzayı içerisinde bir veri segmenti ayırır. Veri segmentinin değerleri değiştirilerek bir tablo için ayrılan yer miktarı da değiştirilebilir.



Şekil 1.12 Tablo alanlarının, veri tiplerinin, varsayılan değerlerinin ve diğer tanımlamalarının yapıldığı ekran

1.2.12.10 TABLO UZAYI (TABLESPACE)

Bir veritabanı, ilişkili mantıksal yapıların gruplanmasını sağlayan ve tablo uzayı olarak bilinen mantıksal depolama ünitelerine bölünmüştür.



Şekil 1.13 Veritabanı, tablo uzayı ve veri dosyaları arasındaki ilişkiyi açıklamaktadır.

Yukarıdaki şekil veritabanı, tablo uzayı ve veri dosyaları arasındaki ilişkiyi açıklamaktadır. Buna göre:

Bir veritabanı bir ya da daha fazla tablo uzayına bölünmüştür.

Tablo uzayı içerisindeki tüm mantıksal yapıları fiziksel olarak depolayabilmek için, her tablo uzayı bir ya da daha fazla veri dosyasına sahip olabilir.

Tablo uzaylarının toplam kapasitesi, sahip oldukları veri dosyalarının toplam kapasitesine eşittir. (Yukarıdaki şekil için SYSTEM tablo uzayı 4MB, USERS tablo uzayı 5MB).

Tablo uzaylarının toplam kapasitesi veritabanının toplam kapasitesini belirler (9 MB)

Bir tablo uzayı açık ya da kapalı olabilir. Tablo uzayı kapalı olduğunda bu tablo uzayının içerisindeki nesnelere erişilemez. Bir tablo uzayı yönetim amaçlı olarak kapalı duruma alınabilir.

1.2.12.11 KULLANICI (USER)

Veritabanındaki nesnelere sahiplerine kullanıcı (user) denmektedir. Kullanıcılar nesnelere yaratır, kullanır ve silerler. Oracle veritabanı kurulduğunda, standart olarak 3 (üç) tane kullanıcı tanımlanır. Bunlar, SYS, SYSTEM, SCOTT.

SYS, sistem kullanıcısıdır. Veri sözlüğünün sahibidir. Önemli nesnelere (tablespace, user, role gibi) yaratma hakkına standart olarak sahiptir. Bütün kullanıcıların bütün nesnelere kullanma hakkına sahiptir. Kısaca veritabanının en önemli kullanıcısıdır. Varsayılan şifresi "change_on_install" dur.

SYSTEM, sistem kullanıcısıdır. Veri sözlüğünün hepsini kullanma ve önemli nesnelere (tablespace, user, role gibi) yaratma hakkına sahiptir. Standart olarak

veritabanındaki bütün kullanıcıların bütün nesnelere kullanma hakkına sahiptir. Varsayılan şifresi “manager” dir.

SCOTT, Oracle’ın deneme tablolarının sahibi olan kullanıcıdır. SQL öğrenenler bu kullanıcının nesnelere kullanabilirler. Varsayılan şifresi “tiger” dir.

1.2.12.12 GERÇEKLEŞTİRİLMİŞ GÖRÜNTÜLER (MATERIALIZED VIEWS)

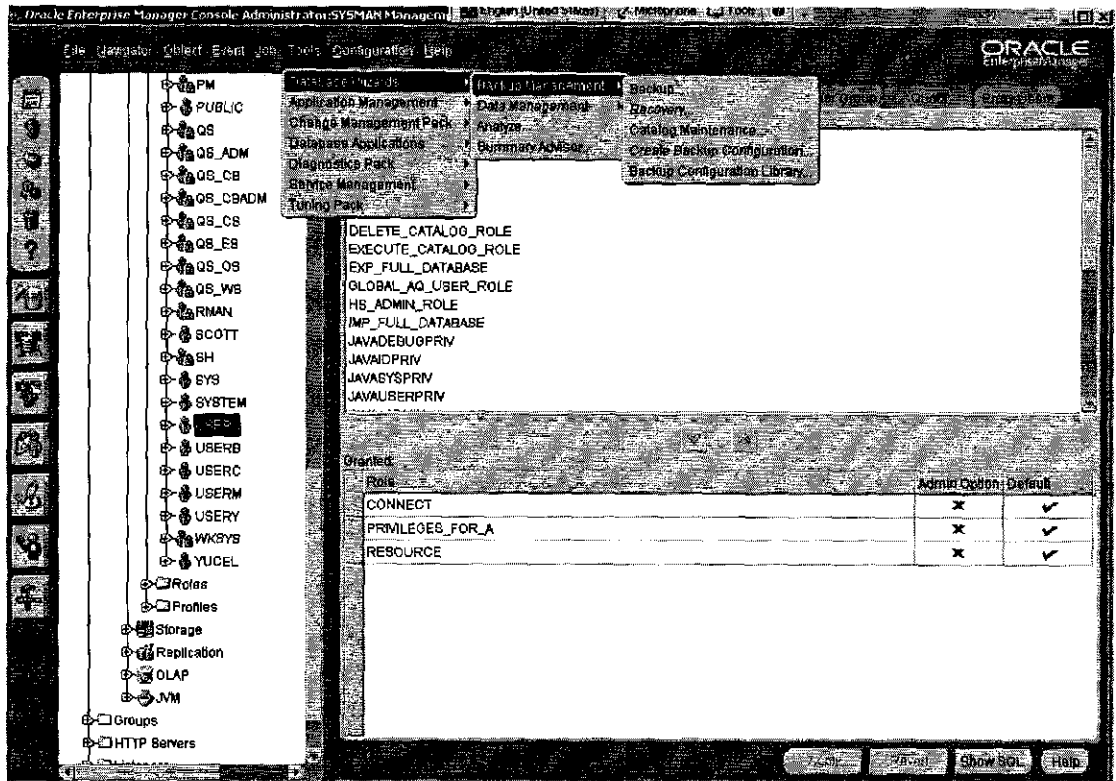
SQL sonucu dönen kayıtların ayrı bir tabloda saklanmasından oluşmaktadır. Normal görüntüler veritabanında görüntünün sql’i hariç herhangi bir yer kaplamamasına rağmen, gerçekleştirilmiş görüntüler içerisinde tuttukları kayıtlar için yer kullanmaktadırlar. Bu tip görüntüler daha çok aylık ya da belirli dönemlerde rapor veya istatistik şeklinde alınan ve bir çok kullanıcı tarafından sorgulanabilen verileri içermek için kullanılmaktadır.

1.2.12.13 VERİTABANI TETİKLEMELERİ (DATABASE TRIGGERS)

Tetiklemeler bir tabloda belirtilen işlem olduğunda veritabanı tarafından otomatik olarak çalıştırılan kaydedilmiş yordamlardır (stored procedures). Tetiklemeler tablolara insert, update, delete veya bunların kombinasyonları yapıldığında çalışacak şekilde düzenlenebilir. Tetiklemeler, genellikle tablolarda belirtilemeyecek kadar karmaşık olan bütünlük kısıtlamalarını (integrity constraints) sağlamak için kullanılmaktadır.

2 YEDEK ALMA VE GERİ GETİRME

Oracle, verileri her ne kadar güvenli tutsa da, fiziksel sebeplerden (disk hataları), kullanıcı hatalarından veya benzer durumlardan dolayı yedeklere ihtiyaç duyulmaktadır. Yedek alma, veritabanı yöneticisinin yapacağı önemli işlerden birisidir.



Şekil 2.1 Yedekleme yönetimi ekranı

Oracle'da, yedek almayla ilgili değişik yöntemler vardır. Veritabanı yöneticisi, kendi veritabanının durumuna göre, yedek almayla ilgili bir veya birden fazla yöntemi kullanmaya karar vermelidir. Bu kararı vermede, yönetici, yedeği kimin alacağı, hangi tür depolama ünitesine alınacağı, ne kadar sıklıkla alınacağı, alınan yedeklerin büyüklüğü, veritabanında küçük de olsa bir kayıba tahammül olup olmadığı gibi çok değişik soru ve bunların cevaplarını gözönünde bulundurmak zorundadır.

2.1 FİZİKSEL YEDEK ALMA

Fiziksel yedek alma veri dosyalarının, redo log dosyalarının ve kontrol dosyalarının yedeklerinin alınması işlemidir. Bir veritabanı ARCHIVELOG ve NOARCHIVELOG olmak üzere iki farklı modda çalışabilir.

2.1.1 NOARCHIVELOG MODUNDA YEDEK ALMA

İşletim sistemi yedeği alma olarak ta bilinen fiziksel yedek alma işleminin dezavantajı, yedek alma işlemi boyunca veritabanının kapatılması gereğidir. Eğer veritabanı NOARCHIVELOG modda çalışıyorsa, çalışma sırasında yapılan değişiklikler bir yere kaydedilmeyecektir. Bu yüzden sistemin bozulması durumunda ancak son alınan yedekler geri getirilebilir. Yani son alınan yedekten sonra yapılan değişiklikler kaybolur. Bu yöntemde bütün log, kontrol ve veri dosyalarının teker teker yedeği alınacaktır ve bu işlem uzun sürebilir, bu nedenle NOARCHIVELOG modu, daha çok veritabanındaki verisi az olan ve veritabanının 24 saat açık kalması zorunluluğu bulunmayan ve az da olsa veri kaybına tahammül edebilecek veritabanı yöneticilerinin kullanacağı bir yöntemdir. Bu modda en ideal zaman, günde bir defa geceleri yedek almaktır. Yedek almak için komut moduna geçilerek aşağıdaki işlemleri yapmak gerekir. Bundan önce SQL*Plus içerisinde yedeği alınacak dosyaların tespiti yapılmalıdır.

```
SQL> spool yedek
SQL> select name from v$controlfile
 2 union
 3 select member from v$logfile
 4 union
 5 select file_name from dba_data_files;
```

NAME

```
-----
C:\ORACLE\ORADATA\OGR\CONTROL01.CTL
C:\ORACLE\ORADATA\OGR\CONTROL02.CTL
C:\ORACLE\ORADATA\OGR\INDX01.DBF
C:\ORACLE\ORADATA\OGR\DEMREP01.DBF
C:\ORACLE\ORADATA\OGR\RBS01.DBF
C:\ORACLE\ORADATA\OGR\REDO01.LOG
C:\ORACLE\ORADATA\OGR\REDO02.LOG
C:\ORACLE\ORADATA\OGR\REDO03.LOG
C:\ORACLE\ORADATA\OGR\REDO04.LOG
C:\ORACLE\ORADATA\OGR\SYSTEM01.DBF
C:\ORACLE\ORADATA\OGR\TEMP01.DBF
C:\ORACLE\ORADATA\OGR\USERS01.DBF
C:\ORASQL\TBS_ESEF.DAT
C:\ORASQL\TBS_GAZI.DAT
C:\ORASQL\TBS_GEF.DAT
C:\ORASQL\TBS_TEF.DAT
```

16 satırları seçildi.

Burada kullanılan V\$LOGFILE, V\$CONTROLFILE ve DBA_DATA_FILES dosyaların veri sözlüğündeki görüntüleridir. Bu görüntüleri sorgulayarak yedeklememiz gereken dosyaları ve bunların nerede bulduklarını öğreniyoruz. Burada yapacağımız iş veritabanı oturumunu kapattıktan sonra, bu dosyaları bir başka depolama ünitesine kopyalamak. Veritabanını Windows NT veya Windows 2000 içerisinde, Denetim Masası'na girip "Sevices" simgesine çift tıkladıktan sonra burada "Oracle" ile başlayan tüm servisleri durdurarak kapatabiliriz. Bu işlemler komut modundan da yapılabilir. Bunun için veritabanı yüklenirken belirtilen dizin içerisindeki "SvrMgr" programı çalıştırılır. Eğer varsayılan dizini seçtiyseniz bu "c:\oracle\ora90\bin" dizinidir. Komut modunda şu adımlar gerçekleştirilir:

```
>svrmgr
svrmgr>connect internal
svrmgr>shutdown immediate
svrmgr>exit
>lsnrctl stop
```

Buradaki işlemlerde önce sunucu yöneticisine (server manager) bağlanılıyor ve sonra da veritabanı kapatılıyor. En sonda da Dinleyici (Listener) durduruluyor. İlgili dosyalar kopyalandıktan sonra :

```
>svrmgr
svrmgr>connect internal
svrmgr>startup
```

Bu dosyaları bir bozulma anında tekrar geri yüklemek için, veritabanını kapatıp dosyaları geriye kopyalamak ve sonra yeniden veritabanını açmak gerekir. İşte burada son yapılan değişiklikler kaybolur.

2.1.2 ARCHIVELOG MODUNDA YEDEK ALMA

ARCHIVELOG modunda yapılan tüm işlemler, redo log dosyalarına otomatik olarak kaydedilir. Bundan dolayı dosyaları tek tek kopyalamak gerekmez. Bunun anlamı, veritabanındaki değişikliklerin sürekli dosyalara kaydedilmesidir. Kaydedilen bu dosyaların hangileri olduğu ve nerede buldukları gibi bilgiler init<SID>.ora dosyası içerisinde yer alır. ARCHIVELOG modda çalışan bir veritabanında, veri kaybı söz konusu değildir. Fakat bu değişikliklerin kaydedildiği dosyaları yedeklemek yararlı olabilir.

Veritabanında bir bozukluk durumunda yapacağımız iş, veritabanını kapatıp "Recover Database" komutunu kullanmak ve sonra veritabanını yeniden açmaktır. "Recover Database" komutunu kullanmak için veritabanı "Mount" modunda açılır.

```
>svrmgr
svrmgr>connect internal
svrmgr>shutdown immediate
>exit
  >lsnrctl stop
  >svrmgr
svrmgr>connect internal
svrmgr>startup mount
svrmgr>recover database
```

2.2 MANTIKSAL YEDEK ALMA

2.2.1 VERİTABANI YEDEĞİ ALMA VE GERİ GETİRME (EXPORT, IMPORT)

Oracle'da mantıksal yedek alma denince Export ve Import anlaşılır. Mantıksal yedek alma, daha çok veritabanının 24 saat açık kalması zorunluluğu bulunan ve az da olsa veri kaybına tahammül edebilecek veritabanı yöneticilerinin kullanacağı bir yöntemdir. Çünkü, bu yöntemde yedek alırken veritabanı kapatılmak zorunda değildir, bu yüzden sistem 24 saat açık kalabilmektedir. Bu sistemde yaşanabilecek bir çökmede, bir önceki alınan yedeklerle, sonra alınan yedekler arasında girilen veriler kaybolabilir. Yedek alma zamanları arasındaki süre kısaltılabilir. Bu da performansı düşürecektir. Bu nedenle en uygun yedekleme zamanı günde bir kez geceleri olmasıdır.

Mantıksal yedek veya veritabanı yedeğinin diğer yedek alma yöntemlerine göre bir kısım ekstra avantajları vardır. Bunlar :

- Tarihlerle göre yedek almak daha kolaydır. Yani, veritabanının yapısına göre alınacak günlük ihraçlar (export), veritabanının durumunu ve gelişmesini günlük olarak takip etmede en kolay yöntemdir.
- Uygulamalara ait tabloların ve veritabanı yapısının verileri olmaksızın yedeği alınabilmektedir. Bu imkanla, herhangi bir uygulamaya ait tablo, indeks, kısıtlama ve sıra gibi nesnelere başka bir ortama, başka bir veritabanına veya başka bir kullanıcıya taşımak için kolaylık sağlanmış olmaktadır.

- Makinalar ve veritabanları arasında veri kopyalama işlemi için kullanılabilir.
- Oracle'ın değişik versiyonları arasında veri taşımak için kullanılabilir.

2.2.1.1 EXPORT

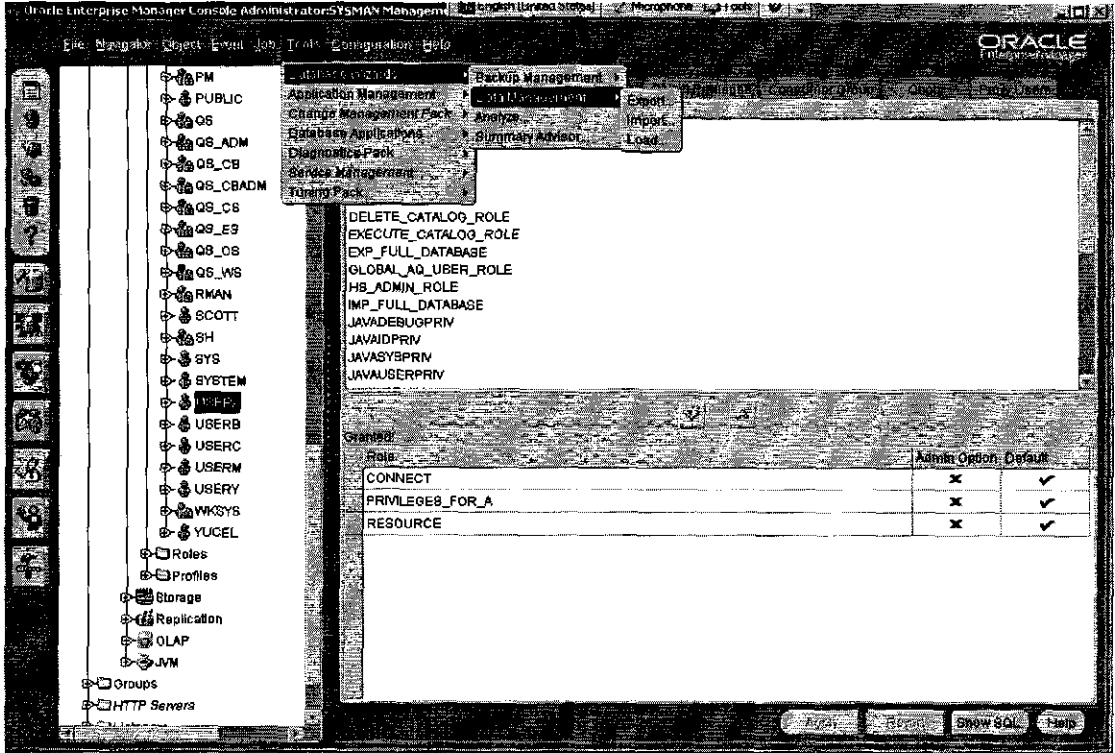
Export, Oracle'ın mantıksal yedek alma işlemleri için geliştirdiği bir üründür. Veritabanı verilerinin anlık olarak görüntüsünün, işletim sistemi dosyası haline dönüştürülerek yedek alınması işlemidir. Yani ihraç alındığı andaki tabloların, indekslerin veya diğer veritabanı nesnelerinin işletim sistemine bir ikili (binary) dosya halinde çevrilerek kopyalanmasıdır. Export ile alınan yedekler, ancak Import ile geri yüklenebilir.

Oracle'ın ihraç uygulamasını kullanabilmek için veritabanı yöneticisi sys kullanıcısı ile "catexp.sql" veya "catalog.sql" dosyalarını bir defalığına çalıştırmalıdır. Bu dosyalar ihraç uygulaması için gerekli görüntüleri yaratmakta ve rolleri ve hakları dağıtmaktadır.

Kullanımı şu şekildedir :

EXP <anahtar>=<değer> veya

EXP <anahtar>=(<değer>,<değer>, ...)



Şekil 2.2 Export – Import – Load ekranı

Export almada 4 mod vardır: Birincisi Tablo Modu (Table Mode)' dur. Bu modda kullanıcılar kendi tablolarının yedeklerini alabilirler ya da hakkı olan kullanıcı, bir başka kullanıcının tablolarının yedeklerini alabilir. İkincisi, Kullanıcı Modu (User Mode)' dur. Bu modda bir kullanıcının nesnelere yedeği alınabilir. Üçüncüsü, Tam Veritabanı Modu' dur (Full Database Mode). Bu modda tüm veri tabanının yedeği alınabilir. Bu modu gerçekleştirmek için kullanıcının bazı özel haklara sahip olması gerekir. EXP_FULL_DATABASE rolüne sahip olan kullanıcı ya da DBA rolüne sahip olan kullanıcılar bu modu kullanabilir. Dördüncüsü, Tablo Uzayı Modu (Tablespace Mode)' dur. Bu modda da tablo uzaylarının yedeği alınır.

Tablo modu ile tablo yedeği alınırken, tablonun tamamının yedeği alınabileceği gibi, herhangi bir PARTITION (bölüm)' in yedeği de alınabilir.

Export yardımcı programını çalıştırmak için komut moduna geçilmelidir. Exp.exe dosyası, Oracle9i kurulurken varsayılan dizin adı kabul edildiyse,

"c:\oracle\ora90\bin" dizini içerisinde bulunabilir. Aşağıdaki örnekler bu dizine geçip komut modunda yazılarak çalıştırılabilir.

- EXP Kullanıcı_Adı / Şifre FILE=DATABASE.DMP FULL=Y
- EXP Kullanıcı_Adı / Şifre FILE=KULLANICI.DMP OWNER=(Kullanıcı_Adı[, Kullanıcı_Adı])
- EXP Kullanıcı_Adı / Şifre FILE=TABLO.DMP TABLES=(Tablo_Adı[:Partition_Adı][, Tablo_Adı[:Partition_Adı]])
- EXP Kullanıcı_Adı / Şifre FILE=TABLESPACE.DMP TABLESPACES=(Tablespace_Adı[, Tablespace_Adı])

2.2.1.2 IMPORT

İthal (import), ihraç (export) ile alınan verilerin veritabanına geri yüklenmesi işlemine denmektedir.

Oracle'ın ithal uygulamasını kullanabilmek için veritabanı yöneticisi SYS kullanıcısı ile "catexp.sql" veya "catalog.sql" dosyalarını bir defalığına çalıştırılmalıdır. Bu dosyalar ithal uygulaması için gerekli görüntüleri yaratmakta ve rolleri ve hakları dağıtmaktadır.

Kullanımı şu şekildedir :

IMP <anahtar>=<değer> veya

IMP <anahtar>=(<değer>,<değer>, ...)

İthal işlemi dört temel şekilde yapılmaktadır. Bunlar :

FULL; veritabanının tamamını geri yükleme,

USER; herhangi bir kullanıcının yedeğini geri yükleme,

TABLE; herhangi bir tablonun yedeğini geri yükleme,

TABLESPACE; herhangi bir tablo uzayının yedeğini geri yüklemedir.

```
IMP Kullanıcı_Adı / Şifre FILE=DATABASE.DMP FULL=Y
[TOUSER=Kullanıcı_Adı]
EXP Kullanıcı_Adı / Şifre FILE=KULLANICI.DMP
FROMUSER=(Kullanıcı_Adı[,Kullanıcı_Adı])
[TOUSER=(Kullanıcı_Adı[,Kullanıcı_Adı])][TABLES=(Tablo_Adı[:Partition_Adı][
,Tablo_Adı[:Partition_Adı]])]
EXP Kullanıcı_Adı / Şifre FILE=TABLESPACE.DMP
TRANSPORT_TABLESPACE=(Y|N)
TABLESPACE=(Tablespace_Adı[,tablespace_Adı])[DATAFILES=(Dosya_Adı[,
Dosya])]
```

2.2.2 ORACLE'IN YEDEK ALMA PROGRAMI (RECOVERY MANAGER - RMAN)

Tanımlanan kurallara göre otomatik olarak yedek alan bir programdır. RMAN, Enterprise Manager'dan veya komut satırından erişilerek kullanılabilir.

3 BAŞKA ORTAMLARDAN GETİRİLEN BİLGİLERİN VERİTABANINA YÜKLENME İŞLEMİ (SQL*LOADER)

SQL*Load, Oracle veritabanı dosyası haricindeki bazı dosya tiplerinden veritabanına bilgi yüklenmesi işleminde kullanılan bir araçtır.

SQL*Loader, kontrol dosyasında adı geçen tablo veya tablolara verileri yüklemektedir. Ayrıca yüklenecek verilerin bulunduğu dosya veya dosyaların adresi de kontrol dosyasında belirtilmektedir. Aynı zamanda SQL*Load'ın çalışması sırasındaki tüm işlemler, uzantısı log olan bir dosyaya kaydedilmektedir. Eğer dosyada veritabanındaki tablonun yapısına uymayan kayıtlar var ise, bu kayıtlar tabloya alınmamakta ve işletim sisteminde uzantısı bad olan bir dosyaya atılmaktadır. Ayrıca, belirtilen şartlara uymayan kayıtlarda işletim sisteminde bir discard dosyasına atılmaktadır.

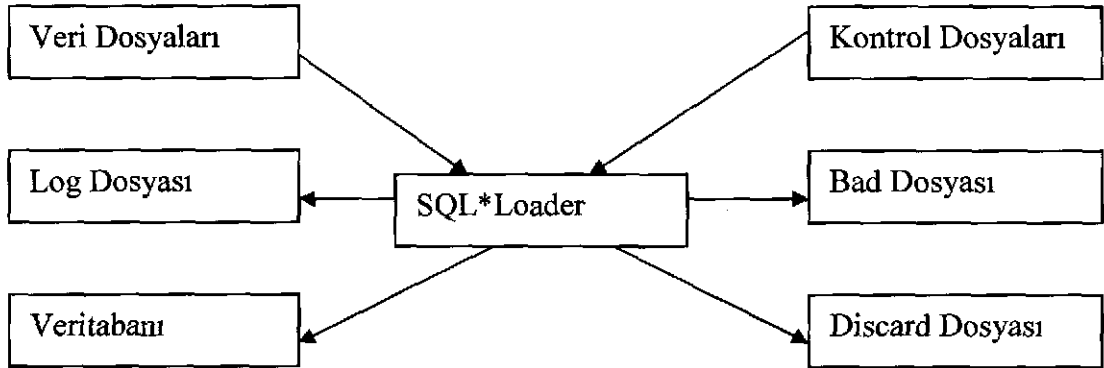
SQL*Loader ile aşağıdakiler yapılabilmektedir :

- Değişik tipteki birden fazla dosyadan veri tabanına veri yüklenebilmektedir.

- Sabit formattaki, herhangi bir karakterle ayrılmış alanların bulunduğu formattaki kayıtlar veritabanına atılabilmektedir.
- SQL fonksiyonları ile veriler veritabanına atılmadan önce gerekli işlemler yapılabilmektedir.
- Birden fazla tabloya aynı anda veri yüklenebilmektedir.
- Tabloya veri yüklerken eleme yapılabilmektedir.
- Birden fazla dosya bir tabloya aynı anda atılabilmektedir.
- Yükleme işlemi esnasında tekil sıra numarası üretilip, tabloya girilebilmektedir.
- Hatalar, ayrıntıları ile birlikte kaydedildiğinden hata kontrolü kolay yapılabilmektedir.
- DIRECT yükleme opsiyonu ile, geri alma parçalarına yazılmadan veriler yüklenebilmektedir. Böylece, yükleme işlemi hızlandırılmaktadır.

SQL*Load işleminin üç temel bölümü vardır. Bunlar :

- Kontrol dosyasının yapısı
- İşletim sistemindeki dosyalarının yapısı
- Komut satırı parametreleri



Şekil 3.1 SQL*Load işlemlerinin akış şeması

4 ORACLE İŞLEMLERİNİN BİRBİRLERİYLE ETKİLEŞİMİ

Kullanıcı işlemi, uygun Net8 sürücüsü kullanarak Oracle sunucusuna bağlanmaya çalışır. Sunucu bağlantı isteğini, uygun Net8 sürücüsü ile algılayarak, kullanıcı işleminin isteklerini karşılayacak sunucu işlemini bellekte oluşturur.

Kullanıcı bir SQL komutu çalıştırarak, veritabanı işleminin (transaction) sonucunu veritabanına kaydeder (commit). Oracle sunucusu, SQL komutunu algılar ve buna karşılık gelen SQL komutunun ortak bellek havuzunda olup olmadığını denetler.

Eğer SQL komutu yoksa, SQL komutu için yeni bir bellek alanı açar. Kullanıcının ilgili veriye erişim izninin olup olmadığını denetler.

Sunucu, istenen veri bellekte değilse, ilgili veri dosyasından okuyarak bellekteki SGA'ya aktarır. Daha sonra bu veriyi değiştirir. DBWR işlemi, değişen veri bloğunu diske yazar. İşlem veritabanına kaydedildiği için, LGWR işlemi, bununla ilgili kaydı işlem saklama (redo log) dosyasına yazar.

İşlemin başarılı olup olmadığı hakkında geri bildirim mesajı kullanıcıya iletilir.

Bu işlemler sırasında arka planda çalışan işlemler müdahaleyi gerektirecek bir durum olup olmadığını kontrol ederler. Sunucu bu esnada oluşacak diğer kullanıcı isteklerine de yanıt verir ve bunların arasında aynı veriye ulaşma konusunda anlaşmazlık olursa bunu giderir.

4.1 VERİYE AYNI ANDA ERİŞİM VE VERİ TUTARLILIĞININ KORUNMASI

Oracle, performans düşüklüğüne neden olmadan, veritabanına erişen birden fazla kullanıcının işlemlerini çeşitli yöntemlerle düzenleyerek veri tutarlılığını sağlamaktadır. Bunu, kilitleme mekanizmasını ve bütünsel veritabanı işlemi (transaction) yaklaşımını birlikte kullanarak yapmaktadır. Oracle, veri tutarlılığını sağlamada şu kriterleri göz önünde tutmaktadır :

- ✓ Belli bir anda çalıştırılan SQL komutunun kullanacağı veriler tutarlı olmalıdır.
- ✓ Veritabanındaki herhangi bir veriyi okumak isteyenler, o veriyi yazanları ve okuyanları beklememeliler.
- ✓ Veritabanına herhangi bir veriyi yazan kullanıcılar, aynı veriyi okuyanları beklememeliler.

- ✓ İki kullanıcı aynı veriye yazmak durumunda ise, bir diğerinin yazma işini beklemelidir.
- ✓ Bir kullanıcı veri değişikliği yaptığı zaman bunu, COMMIT komutunu çalıştırana kadar, diğer kullanıcı göremez, önceki veriyi görür. Ancak bu komut sonrasında veriler kalıcı hale gelir ve kullanıcılar tarafından artık bu değiştirilmiş verilerin son halini görürler.
- ✓ Tek bir sorgu veritabanı üzerinde çalışırken, veri tutarlılığı, Oracle tarafından garanti edilmektedir. Birden fazla tablo üzerinde sorgulama yapılıyorsa, işlem salt okunur olarak belirtilmelidir.

4.2 KİLİTLEME MEKANİZMALARI

Kullanıcıların verilere aynı anda erişimini denetim altında tutmak için Oracle, kilitleri kullanır. Bunu kullanıcıya hissettirmeden yapar ve böylece veri tutarlılığını ve bütünlüğünü sağlar.

Kilitleme yöneticisi, satır seviyesinde kilitleme yaparak, aynı satıra birden fazla güncelleme için erişimi önler. Oracle, exclusive (dışlayan) ve share (paylaşımli) şeklinde iki farklı kilitleme yöntemi kullanır. Her iki kilit türünde de, kilit koyan kullanıcı haricindeki kullanıcılar sadece okumak için erişim yapabilirler, kaynak üzerinde değişiklik yapamazlar.

Oracle, ayrıca manuel olarakta kaynak ve tablo seviyesinde kullanıcılara kilitleme izni verir.

4.3 VERİ AKTARIM ARAÇLARI

Veritabanındaki bilgilerin tamamını veya bir kısmını başka bir veritabanına taşıyabilmek için Oracle'da 3 farklı araç (Export, Import, SQL*Loader) vardır. Export aracı, veritabanının tamamını veya bir kısmını parametre olarak belirtilen bir işletim sistemi dosyasına kaydeder. Bu dosyanın içeriğini okuyup, aynı veritabanına veya başka bir veritabanına aktarmayı sağlayan uygulama da Import'tur. Import olmadan Export dosyasını okuyup veritabanına aktarmak mümkün değildir. Sabit bir formatta veya belli karakterlere ayrılmış olarak herhangi bir dosyada bulunan verileri bir Oracle veritabanına aktarmak için ise SQL*Loader kullanılabilir.

4.4 NET8 BAĞLANTISI

Kullanıcı ve sunucu işlemleri, ağ üzerinde farklı bilgisayarlarda ise birbiriyle haberleşmek için Net8 arayüzünü kullanırlar. Net8, standart iletişim protokolleri ile verinin bilgisayarlar arasında uygun biçimde aktarımını sağlayan Oracle arayüzüdür. (Bkz.: 7.2.3)

5 DAĞITIK VERİTABANI YAPISI

Oracle, birden fazla veritabanı sunucusunu kullanıcılara, birbiriyle bağlantılı biçimde çalışan tek bir veritabanı olarak gösterebilme olanağı sunmaktadır. Her kullanıcı sadece tek bir veritabanı sunucusuna bağlanmakta, fakat diğer veritabanlarına erişebilmektedir. Kullanıcının bağlandığı veritabanı yerel, diğer bağlantılı veritabanları ise uzak veritabanı olarak adlandırılır. Kullanıcılar, sanki tüm veriler ortak bir veritabanında saklanıyor izlenimindedirler.

- Dağıtık veritabanındaki her veritabanı, diğerinden bağımsız yönetildiği için karmaşıklık azalmış olur.
- Uzak veritabanındaki herhangi bir veriye erişim oldukça basit komutlarla yapılabilmektedir. Dağıtık veritabanındaki bütünsel veri işlemleride yerel veritabanında çalışanlarda olduğu gibi kaydedilir (COMMIT) veya eski haline getirilir (ROLLBACK).
- Bu yapıdaki her veritabanı sunucusunda ayrı bir kurtarma işlemi (RECO) çalışır. Herhangi bir dağıtık işlem sırasında ağ veya sistem arızası olmuşsa, RECO, ilgili uzak veritabanındaki bütünsel işlemleri otomatik olarak kayıt eder veya eski haline getirir.
- Dağıtık veritabanı sistemindeki her yönetici kendi yerel veritabanındaki kilitleri bularak sonlandırabilir, tabloların bir yedeğini yerel veritabanında saklayabilir. Bütün veritabanı sunucularını ve kullanıcıları birbirine bağlamak için Net8 kullanılır.

6 VERİTABANI GÜVENLİĞİ

Oracle'daki güvenlik mekanizmaları, izinsiz veritabanı erişimini ve şema nesnelere erişimi engellemektedir. Ayrıca disk ve işlemci gibi sistem kaynaklarının kullanımını kontrol etmekte ve kullanıcıların yaptığı işlemleri izleme olanağı sağlamaktadır. Her veritabanı kullanıcısı için aynı isimde şema kullanılır. Şemalar, veritabanı nesnelere birarada bulunduğu mantıksal yapılardır. Kullanıcılar ilgili şemalar içinde nesne yaratma ve kullanma yetkisine sahiptir.

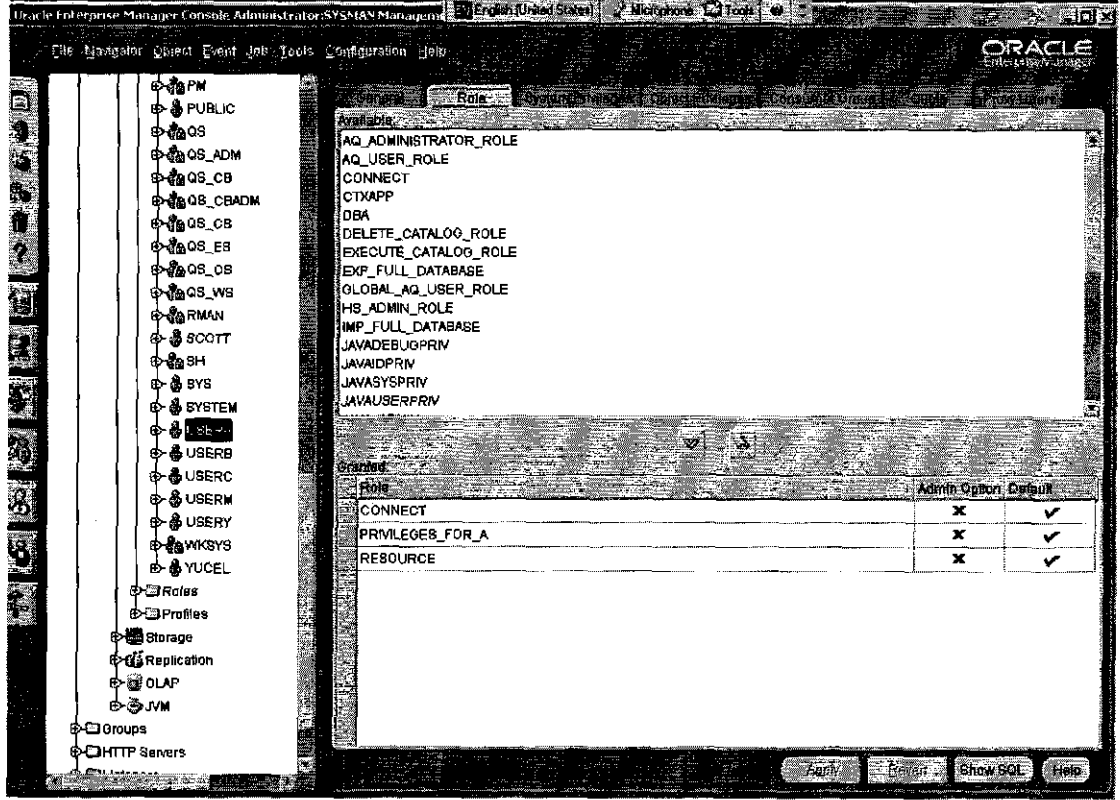
Kullanıcıların kullanıcı adı ve şifresinin doğruluğunun denetlenmesi, kullanıcıya ait şema nesnelere için kalan disk alanının kontrolü ve kullanıcının kaynak limitlerinin izlenmesi, sistem güvenliğinin içerdiği konulardır. Kullanıcının bağlanma, izleme izinleri ve kullanıcının veritabanında ne tür işler yapabileceğini kontrol eder.

Veri güvenliği, ilgili şema nesnelere erişimi ve kullanımı denetleyen mekanizmaları içerir. Örnek olarak; A kullanıcısının X tablosuna SELECT ve INSERT haklarının olup, DELETE ve UPDATE haklarının olmaması gibi bir tanımlama yapılabilir.

6.1 GÜVENLİK MEKANİZMALARI

Oracle'da veri ve nesnelere erişim için kullanıcıların yetkilendirilmiş olması gereklidir. Bir kullanıcı başka kullanıcılara yetki verme hakkına sahip olabilir. Her kullanıcı için yetkiler, roller, tablo uzayı kotaları ve kaynak limitleri vardır. Yetki, belli bir SQL komutunu çalıştırma hakkı demektir. Veritabanına bağlanma, şema içerisinde tablo yaratma, başka birinin tablosundaki kayıtları seçme, başka birine ait saklanmış prosedürü çalıştırma hakkı gibi yetkilerden söz edilebilir. Yetki, sistem ve şema nesne yetkileri şeklinde iki kısımda düşünülebilir. Bir tablo uzayı yaratmak veya veritabanındaki herhangi bir tablodaki kayıtları silmek, sistem yetkilerindedir. Bunların çoğu veritabanı yöneticileri ve uygulama geliştiricilere aittir. Şema nesne yetkileri, şema içindeki belli bir nesne üzerinde yapılacak işlemlerle ilgili yetkililerdir.

Bu yetkiler ise son kullanıcılara verilir ve uygulamalar yardımıyla veriler üzerinde kendilerine tayin edilen işlemleri yaparlar.



Şekil 6.1 USERA kullanıcıasına atanmış rollerin görünümü ve atanabilecek roller. Privileges_for_A rolü dahilinde belirlenmiş yetkiler USERA kullanıcıasına verilmiştir.

6.2 YETKİ VERME

Yetkiler kullanıcılara iki şekilde verilebilir.

- ✓ SQL komutları yardımıyla belli kullanıcılara belli yetkiler doğrudan verilir. Örneğin; OGRENCI tablosuna kayıt ekleme yetkisi AHMET kullanıcıasına verilir.
- ✓ Yetkiler roller şeklinde grup halinde tanımlanır ve bu roller kullanıcılara verilir. Örneğin; OGRENCI tablosuna kayıt ekleme yetkisi MEMUR rolüne verilir ve bu rol AHMET ve MUGE kullanıcılarına atanır.

Rolleri kullanmak, yetkileri daha iyi ve daha kolay kontrol etmeyi sağlar. Tek rol bir defaya mahsus tanımlanır. Sonra istenilen kullanıcıya atanır. Veritabanı yöneticisi rol için bir şifre tanımlayabilir ve böylelikle diğer kullanıcıların rolde tanımlı yetkileri öğrenmelerini engelleyebilir.

6.3 DİSK KULLANIMINI SINIRLAMA

Oracle, kullanıcılar için ayrı ayrı disk kullanımını yönlendirip sınırlayabilir. Her kullanıcı varsayılan olarak bir tablo uzayına atanarak yaratılır. Kullanıcılar, başka bir tablo uzayını açık olarak belirtmedikleri sürece yeni bir tablo, indeks, öbek gibi nesnelere oluşturmak istediklerinde, default tablo uzayı kullanılır. Ancak tersi için yetki ve tablo uzayında yeterli kotası olması gereklidir.

6.4 PROFİL (PROFILE) KULLANIMI VE KAYNAK SINIRLAMA

Profiller tanımlanarak kullanıcılar için pek çok kısıtlama getirilebilir. Bunlardan bazıları

- ✓ Kullanıcının aynı anda başlatabileceği oturum sayısı
- ✓ Kullanıcı oturum veya bir SQL komutu için ayrılan işlemci zamanı ve mantıksal disk erişim imkanı
- ✓ Kullanıcı oturumu için bekleme durumunda kalma süresi
- ✓ Kullanıcı oturumu için bağlantı kurma süresi
- ✓ Şifreler için kaç denemeden sonra şifrenin kilitleneceği, şifrenin geçerliliğini yitirme süresi, tekrar kullanımını ve karmaşıklığının derecesi

şeklindedir.

Profil tanımlanmayan kullanıcıya varsayılan profil otomatik olarak atanır. Kaynak kullanımı limiti, veritabanı sistem kaynaklarının ekstra tüketilmesini önler.

6.5 İZLEME (AUDIT)

Veritabanındaki şüpheli erişimleri araştırmak için Oracle, kullanıcı işlemlerini izleme imkanını sunmaktadır. İzleme aşağıdaki üç seviyede yapılır :

- ✓ Komut izleme, belirli SQL komutlarını izleme imkanıdır.
- ✓ Yetki izleme, sistem açısından önemli yetkilerin hangi kullanıcılar tarafından kullanıldığını izleme işlemidir.
- ✓ Şema nesnesi izleme, belirli nesnelere kullanıcıların erişimini izleme işlemidir.

Oracle, yapılan işlemlerin başarı ile sonuçlanıp sonuçlanmadığı bilgisini de saklar.

İzleme işleminin sonuçları SYS.AUD\$ adlı görüntüde saklanmaktadır.

7 ORACLE ÜRÜNLERİ

7.1 ORACLE RDBMS

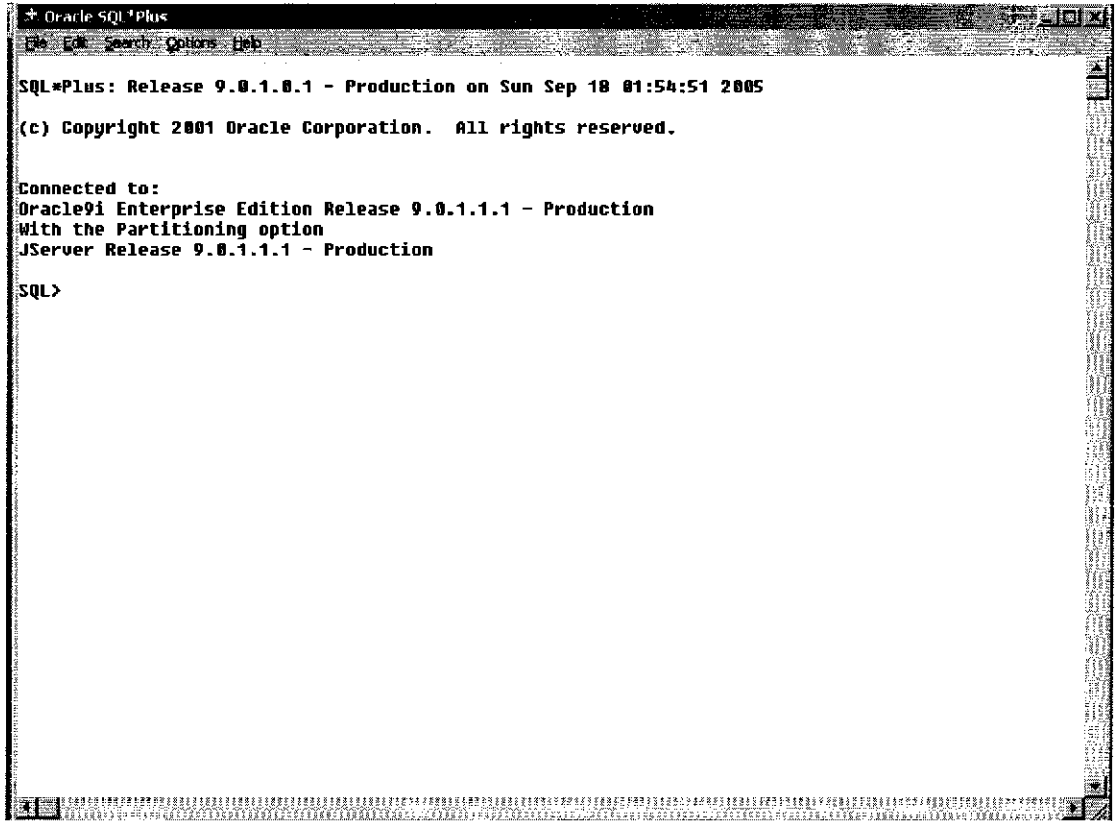
Ana ürün. En yeni sürümü 11i dir. Projede 9i kullanılıyor. Oracle 6, 7, 8i, 9i, 10i,11i sürümleri mevcuttur.

7.2 ORACLE UYGULAMA GELİŞTİRME ARAÇLARI

Oracle'ın veritabanı uygulamaları geliştirmek için görsel ürünleri bu grupta yer alır. Eski adı Oracle Power Objects olan bu ürünler, Oracle Developer 2000 ve sonra da Oracle Developer 6 'dır. Developer 2000 ve Developer 6, Form Builder, Report Builder, Graphics Builder, Procedure Builder ve Schema Builder ürünlerini içerir. Form Builder, kullanıcı arayüzü geliştirilen programdır. Report Builder, veritabanından elde edilen bilgiler ile raporlar hazırlamaya yarar. Özellikle banka ve kamu kuruluşlarının ihtiyacı olan fatura, evrak gibi belgelerin hazırlanıp çıktı alınması için geliştirilmiş bir programdır. Procedure Builder, sürekli kullanılacak kodların prosedürler halinde oluşturulup, kütüphaneler olarak kaydedilmesi işlemine yarayan bir programdır. Schema Builder ise, veritabanı nesnelerinin görsel olarak oluşturulabilmesi için geliştirilmiş bir programdır. Burada tablo, görüntü, eşanlam oluşturma, tablolar arası ilişki kurma, tablo kısıtlamaları oluşturma işlemleri görsel olarak yapılabilir.

7.2.1 SQL* Plus

SQL (Structure Query Language) * Plus, Oracle veritabanında sorgulama yapmak için kullanılan bir araçtır. SQL * Plus' da SQL komutlarının yanısıra PL/SQL komutlarında kullanılabilir.



Şekil 7.1 SQL*Plus Ekranı ve komut satırı

7.2.2 PL/SQL (Procedural Language)

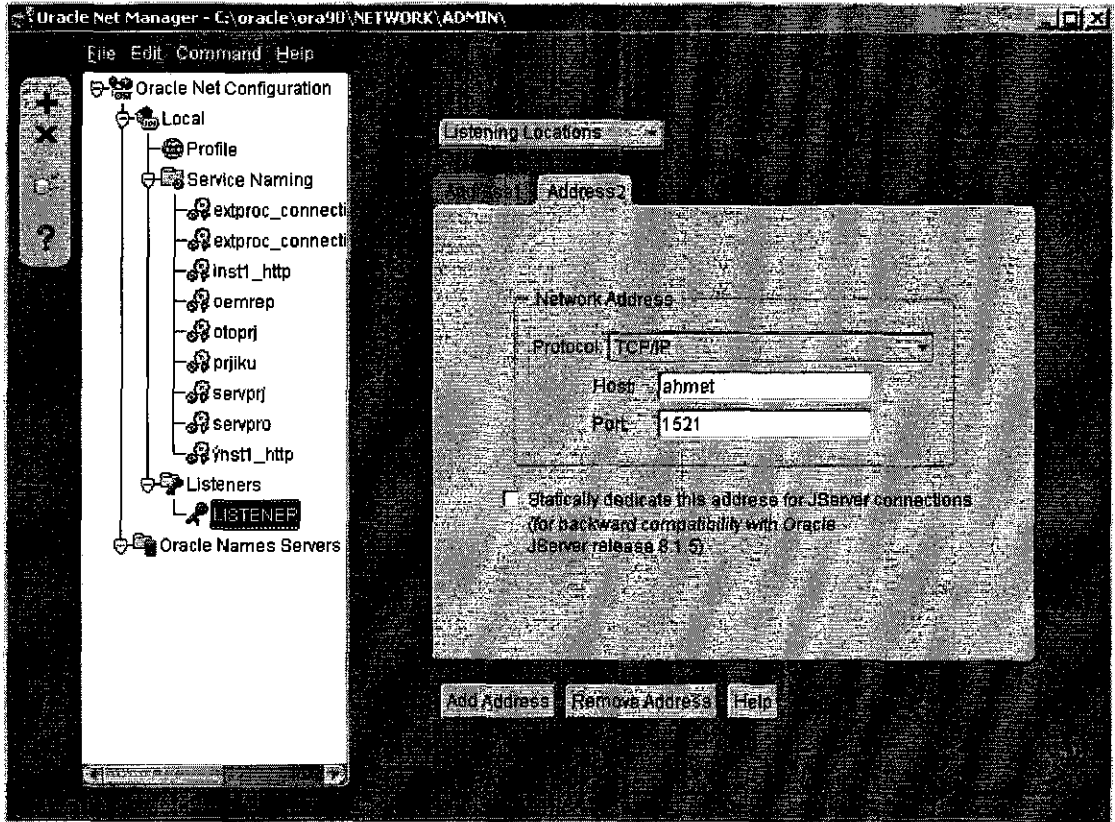
PL/SQL, yapısal dillere ait özelliklerin standart SQL'e eklenmesiyle ortaya çıkan Oracle'a has bir dildir. Kullanıcılar PL/SQL'i kullanarak programlarına bir çok yeni özellikler katmaktadırlar. Bunlardan bazıları :

- PL/SQL tabanlı kaydedilmiş yordamlar (stored procedures) ve veritabanı tetiklemeleri (triggers) ile önemli iş kurallarını programlamak
- Oracle developer ürünleri ile güçlü ve kolay kullanılabilir görsel ortamlarla programları detaylandırmak ve daha rahat kontrol etmek
- Oracle tabanlı uygulamalarda nesne-tabanlı (object-oriented) tasarımlar yapma WEB sayfalarından Oracle veritabanlarına link yapma
- En önemlisi de istemci - sunucu ortamları tasarlama ile ağ üzerinde dağıtık işlemler yapma ve bir çok sorguyu çalıştırma imkanı sunar.

PL/SQL'de, en gelişmiş yordamsal dillerde olan veri tipleri, blok yapısı, şartlara bağlı ve bağlı olmayan kontrol cümleleri, bir çok döngü tipi, hata durumlarında kullanılacak yordamlar ve kullanıcı tarafından tanımlanabilecek nesnelere yer almaktadır.

7.2.3 Net8

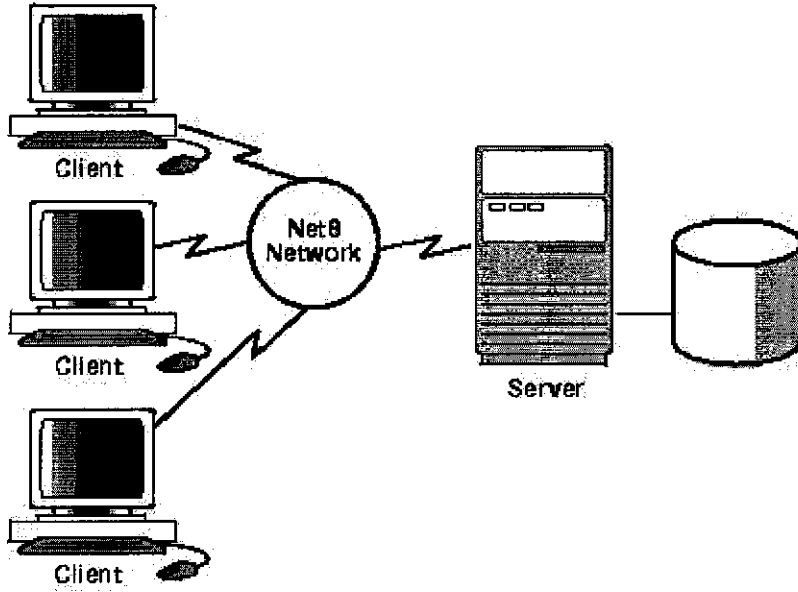
Net8 Oracle'ın bir ağ ürünüdür. Farklı bilgisayarların veritabanına bağlantı kurması ve istemci sunucu arasında veri alışverişinin sağlanabilmesi Net8 in ana görevidir. Net8 ağıdaki her bilgisayara kurulur. Ağ bağlantısı sağlandığında Net8 istemci ile sunucu arasında bir veri taşıyıcı gibi işlem görür. Ağ üzerinde bir istemcinin sunucudaki veritabanına erişebilmesi için, sunucuda bir ağ servis adının (Net Servis Name) ve bir dinleyicinin (Listener) oluşturulması gerekir.



Şekil 7.2 Listener Ekranı

Oracle Enterprise edition ile birlikte gelen Net8 Assistant ile bu gerçekleştirilebilir. Ağ servis adları, veritabanı gibi bir servisi ağda tanımlamak amacıyla basit bir yöntem olarak kullanılır. Bir kullanıcı, ağ servis adını kullanarak bağlantısını şöyle sağlar :

Connect kullanıcı_adi/şifre@ağ_servis_adi



Şekil 7.3 User ve Server arasında Net 8 bağlantı şeması

Ağ servis adları TNSNAMES.ORA adlı dosyaya kaydedilir. Mevcut sistemimizde ki dosya içeriği aşağıdaki gibidir.

TNSNAMES.ORA Network Configuration File:

C:\oracle\ora90\network\admin\tnsnames.ora

Generated by Oracle configuration tools.

SERVPRO =

(DESCRIPTION =

(ADDRESS_LIST =

(ADDRESS = (PROTOCOL = TCP)(HOST = ahmet)(PORT = 1521))

)

(CONNECT_DATA =

(SERVER = SHARED)

(SERVICE_NAME = servpro)

)

)

SERVPRJ =

(DESCRIPTION =

(ADDRESS_LIST =

(ADDRESS = (PROTOCOL = TCP)(HOST = ahmet)(PORT = 1521))

```

)
(CONNECT_DATA =
  (SERVICE_NAME = servprj)
)
)
OTOPRJ =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = ahmet)(PORT = 1521))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = otoprj)
    )
  )
)
OEMREP =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = ahmet)(PORT = 1521))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = OEMREP.iku.edu.tr)
    )
  )
)
EXTPROC_CONNECTION_DATA.AHMET.URUSAN.COM =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC0))
    )
    (CONNECT_DATA =
      (SID = PLSExtProc)
      (PRESENTATION = RO)
    )
  )
)
PRJIKU =

```

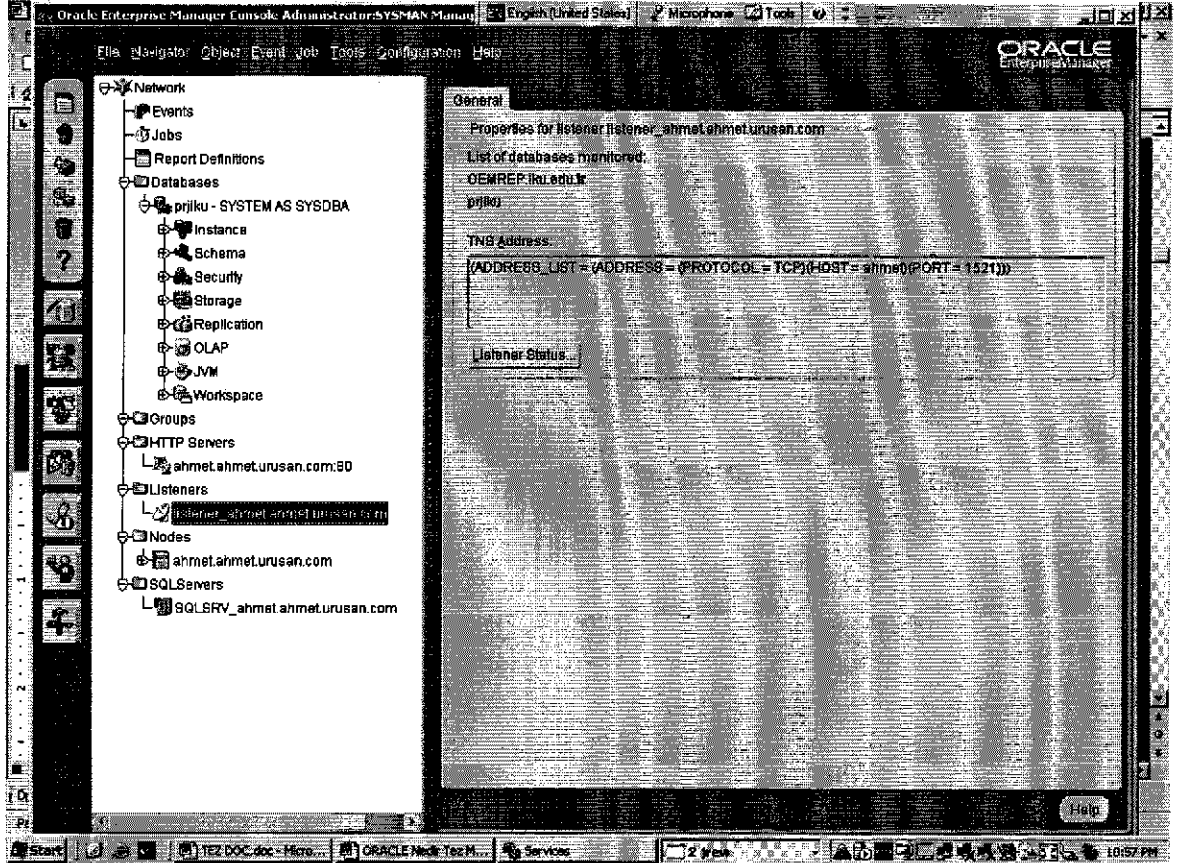
```

(DESCRIPTION =
  (ADDRESS_LIST =
    (ADDRESS = (PROTOCOL = TCP)(HOST = ahmet)(PORT = 1521))
  )
  (CONNECT_DATA =
    (SERVICE_NAME = prjiku)
  )
)
INST1_HTTP =
(DESCRIPTION =
  (ADDRESS_LIST =
    (ADDRESS = (PROTOCOL = TCP)(HOST = ahmet)(PORT = 1521))
  )
  (CONNECT_DATA =
    (SERVER = SHARED)
    (SERVICE_NAME = MODOSE)
    (PRESENTATION = http://HRService)
  )
)
EXTPROC_CONNECTION_DATA =
(DESCRIPTION =
  (ADDRESS_LIST =
    (ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC0))
  )
  (CONNECT_DATA =
    (SID = PLSExtProc)
    (PRESENTATION = RO)
  )
)
YINST1_HTTP =
(DESCRIPTION =
  (ADDRESS_LIST =
    (ADDRESS = (PROTOCOL = TCP)(HOST = ahmet)(PORT = 1521))
  )
)

```



```
(CONNECT_DATA =
(SERVER = SHARED)
(SERVICE_NAME = MODOSE)
(PRESENTATION = http://HRService) )
)
```



Şekil 7.4 Oturum içerisinde açık olan Listeners' ların görüntüsü

7.2.4 SQL*Net

SQL*Net, Net8'in yardımcı programıdır. Bir istemci bilgisayara Oracle Forms ürününün yüklendiğini düşünelim. Bunun için istemci bilgisayar SQL*Net ile oluşturulan veritabanı takma adını (alias) kullanır. Net8 konfigürasyonu istemci bilgisayarlarda SQL*Net sayesinde yapılır. Burada veritabanına verilecek takma isim, protokol adı, ana bilgisayar adı ve veritabanının SID olarak girilen ismi bilgileri girilerek düzenleme yapılır.

8 PROJEYE BAŞLAMAK

Bu hatırlatmaların ardından bir taraftan veritabanı dosyasını oluşturacak ön bilgileri toplarken diğer taraftan da eldeki sistemin, donanımsal ve yazılımsal olarak, kullanılmak istenen Oracle veritabanı programı için yeterli olup olmadığına gözatmak gereklidir. Oracle 9i Release 1 versiyonu için gerekli minimum konfigürasyon :

Pentium III 866 MHz. İşlemci, 256 MB RAM, 13 GB boş hard disk alanı (bütün Oracle ürünleri yüklenirse), Windows NT 4.0 with Service Pack 5 işletim sistemi şeklindedir.

Bu çalışmada, Intel Centrino 1.6 GHz. işlemci, 512 MB RAM, 60 GB hard diskli bir Server makine ve Active Directory, Application, Web, Printer ve Client server özellikleri yüklenmiş İngilizce Windows 2003 Server işletim sistemi, SQL Server, Oracle Enterprise Manager 9i Release 9.0.0 veri tabanı programları ve Delphi 7 ara yüz programlama dili kullanılmıştır.

Oracle Enterprise Manager (OEM) 9i'nin kurulum aşamasında pek çok hata mesajı ile karşılaşmış, bu mesajların içerdiği hatalar ve bunların çözümleri Oracle'ın resmi internet sitesinden bulunarak çözümlenmiştir. OEM'in kurulumu ile birlikte PRJIKU isimli bir veritabanı dosyasıda oluşturulmuştur.

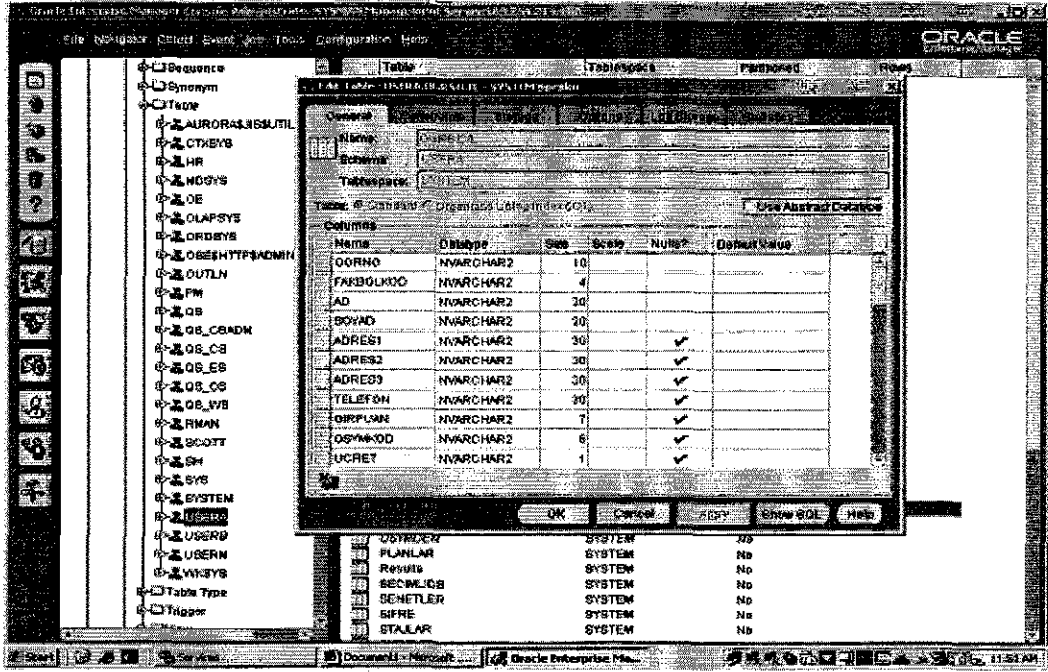
Yine, Delphi üzerinden Oracle'daki tablolara bağlantı sağlamak için normal koşullarda BDE kullanılması gerekir. Ancak bu program geliştirilirken internette edinilen Oracle Data Access Components 4.5 (ODAC 4.05) bileşeni Delphi'ye eklenerek bu konuda bir avantaj yakalamanın yanısıra, kullanıcı terminale sadece program kurulumunu gerçekleştirmek yetmektedir. Böylelikle BDE, ADO, ODBC'den ve Alias tanımlamadan daha başka bir yol kullanılmış, böyle olunca da kullanıcı makineye ORACLE CLIENT kurmak ihtiyacı ortadan kalkmıştır. Bu durumda hem Delphi'den Oracle'a erişimde, hem de, Oracle Client kurulmadan da yalnızca yazılan programın ilgili istemci makineye kurulumu ile, server üzerinde varolan

Oracle Enterprise Manager içerisindeki tablolara erişim imkanı elde edilmiştir. Bu, normal koşullarda toplamda oldukça maliyetli olabilecek Oracle istemci programlarının satın alınmından, şu anki gözlemlere göre, bir kar sağlamaktadır. Ayrıca BDE ve ADO bileşenlerinin kullanımı, dosya boyutu büyüdükçe sorgulamaların uzamasına da neden olmaktadır. Yaklaşık 1 sene kadar yüklenecek veriler nedeniyle dosya boyutunun artması sonrasında, programda bir çok probleme neden olabilecek durumlarda ortadan kalkmıştır.



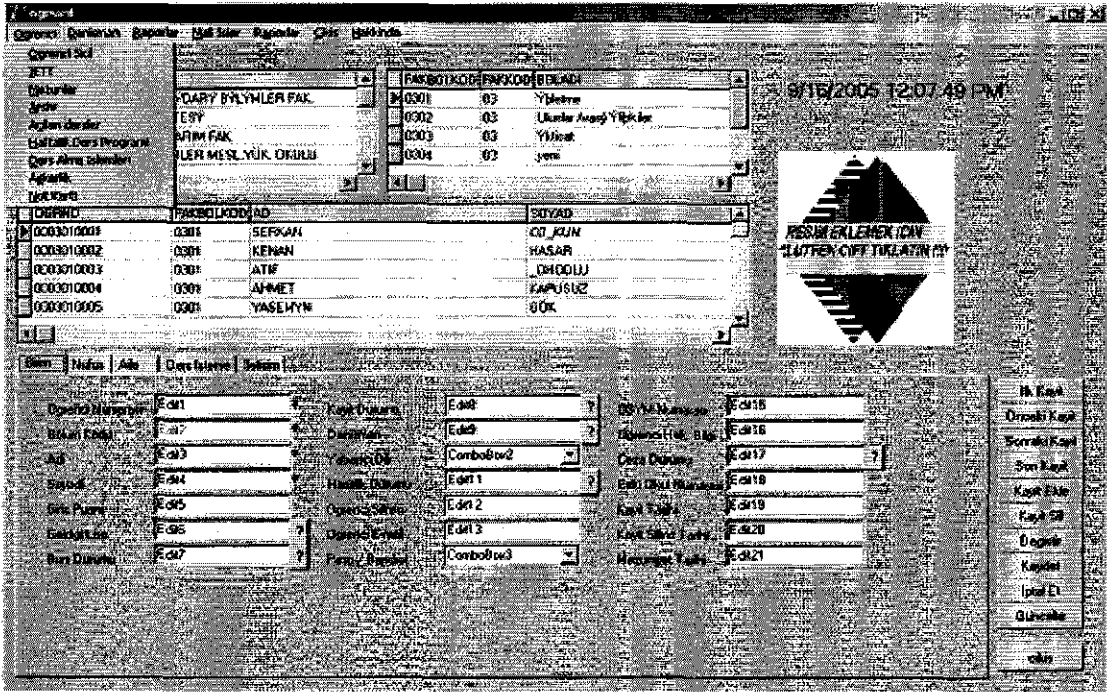
Şekil 8.1 Program Giriş Ekranı. Bu ekranda girilen kullanıcı adının uzunluğuna göre, 4 karakter ise idari ve öğr. üyeleri, 10 karakter ise öğrenci girişi gerçekleşecektir.

İstanbul Kültür Üniversitesi bünyesinde halen kullanılmakla birlikte yerini yeni bir programa devretmek üzere olan Sn. Prof. Dr. Behiç ÇAĞAL tarafından yazılmış öğrenci, personel ve mali işler otomasyon programının, access veritabanında hazırlanmış ve 5 senelik (1999-2004) bütün verileri içeren veritabanı dosyası alınarak Oracle'da, PRJIKU veritabanı altında, 'UserA' kullanıcı adı altında açılmış olan tablolara SQL server aracılığı ile aktarılmıştır.



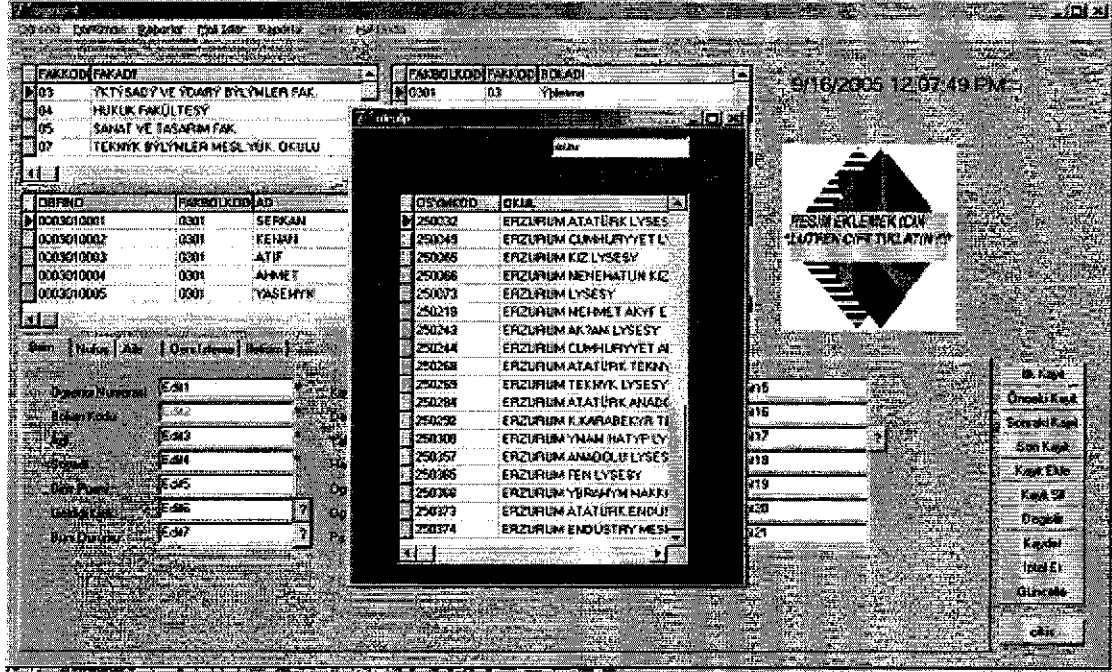
Şekil 8.2 Örnek bir tablonun tasarım ekranı. UserA kullanıcıasına ayrılmış tablo uzayında açılmış bir tablonun alanları, bu alanlara atanmış veri tipleri, bu verilerin maksimum uzunlukları ve veri girşinin gerekli olup olmadığı yer alıyor

Programda, genel olarak bir giriş ekranı ve sonrasında üzerinde Öğrenci, Danışmalar, İdari, Mali İşler, Raporlar, Çıkış, Hakkında gibi aşağı doğru açılabilen menüler bulunan diğer ekranlar tasarlanmıştır. Bu ekranlarda çoğunlukla kullanıcıya kullanım kolaylığı sağlaması amacıyla birbirleriyle aile-çocuk ilişkisi olan Oratable veya Oraquery'lere bağlı DBGrid nesnelere kullanılmıştır.



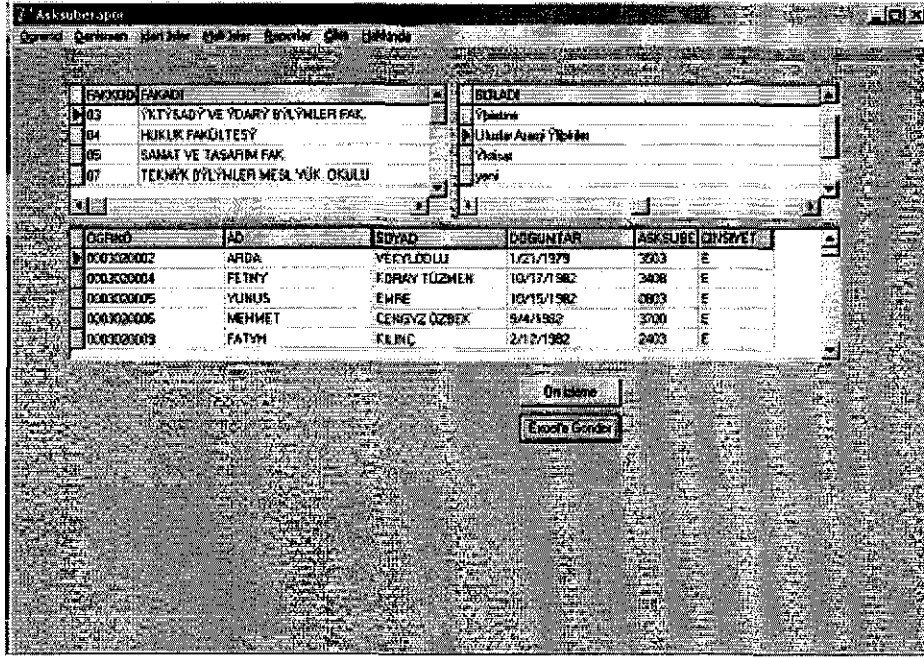
Şekil 8.3 Öğrenci sicil ekranı ve aşağı doğru açılabilen menüler

Formlar üzerinde zaman zaman yazım hanelerinin sağ taraflarında yer alan “ ? ” butonları, ilgili verinin alınabileceği ekranlara ulaşılmasını sağlayıp buradan filtreleme yapılarak, istenilen verinin bulunduğu hücre üzerinde çift tıklama ile, kayıt girilmekte olan formun ilgili yazım hanesine yerleşmesini sağlayan önemli bileşenler olarak kullanıcıya sunulmuştur.



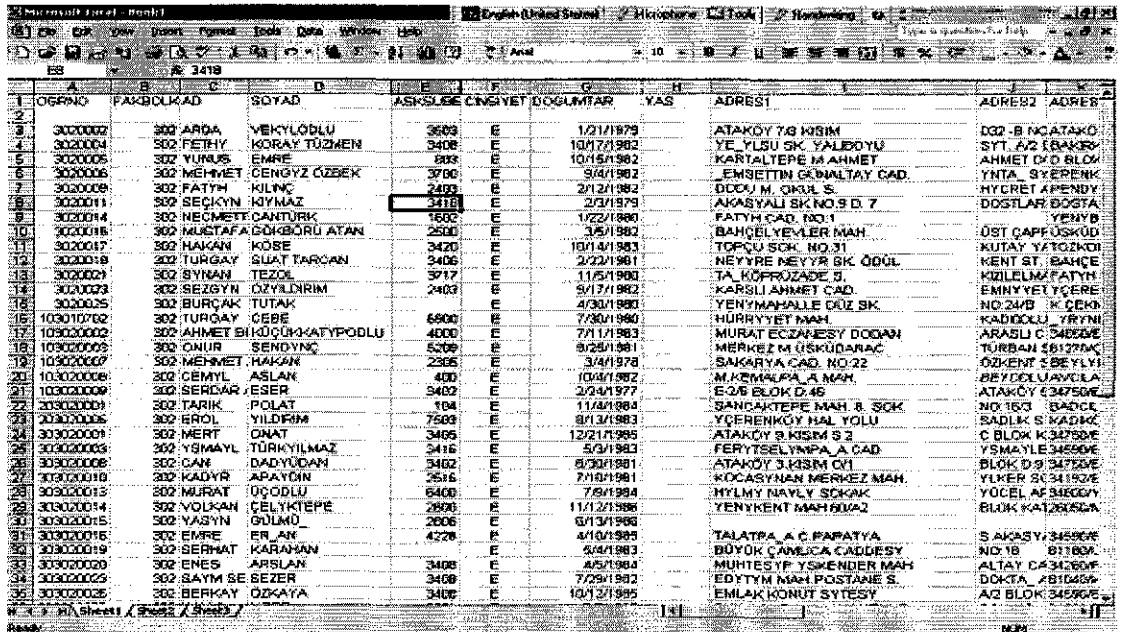
Şekil 8.4 Filtreleme yöntemi ile veri girişini kolaylaştıran ekranlar

Raporlarda tabii olarak filtrelemelerin yanı sıra, gerek QReport ve gerekse bilgilerin Excel ve Word'e aktarıldığı OLE nesnelere yer verilmiştir. Kullanıcıların en çok sıkıntı çektiği yazıcı çıktılarından biri olabileceği düşünülen 'Askerlik Tecil Belgeleri' nin bölümlere göre tamamının bir seferde yazıcıdan çıkarılması seri dökümlere örnek olarak geliştirilmiştir.

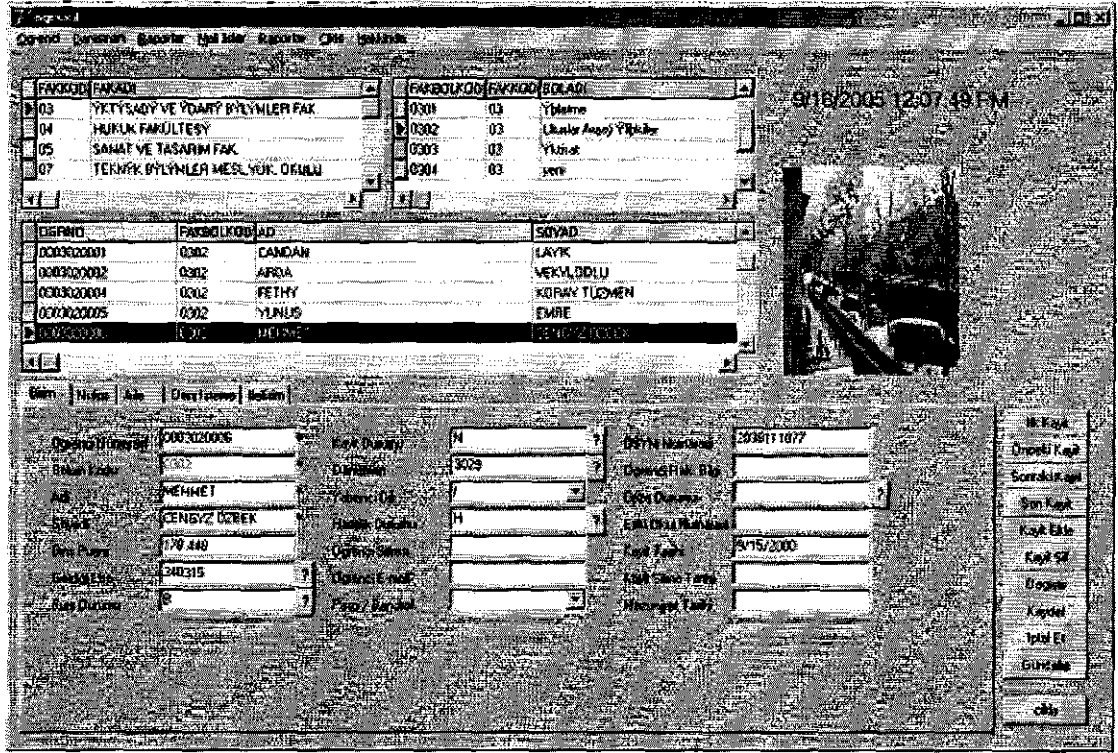


Şekil 8.5 Örnek bir ekran. Bu ekranda veriler istenirse yazıcıya, istenirse excel programına gönderilebiliyor. Excel dosyası şeklinde kaydedilen veriler daha sonra Tecil Belgesi gibi toplu yazıcı çıktıların alınabilmesi için kullanılabilir

Bu yapılırken, önce ilgili bölüm veya programdaki erkek öğrencilere ait bilgiler Excel'e aktarılıyor. Bu bilgiler Excel'de bir isim altında kaydediliyor ve bu bilgilerin Word'de hazırlanan Tecil belgesi taslağında, Mail Marge özelliği kullanılarak ilgili alanlara yerleştirilmesi sağlanıp tamamının yazıcı çıktısı bir seferde alınabilmiştir. Bu yöntem, 'Transkript' gibi her bölüm için aynı dönemde seri olarak çıkarılması gereken belgeler için de kullanılabilir.

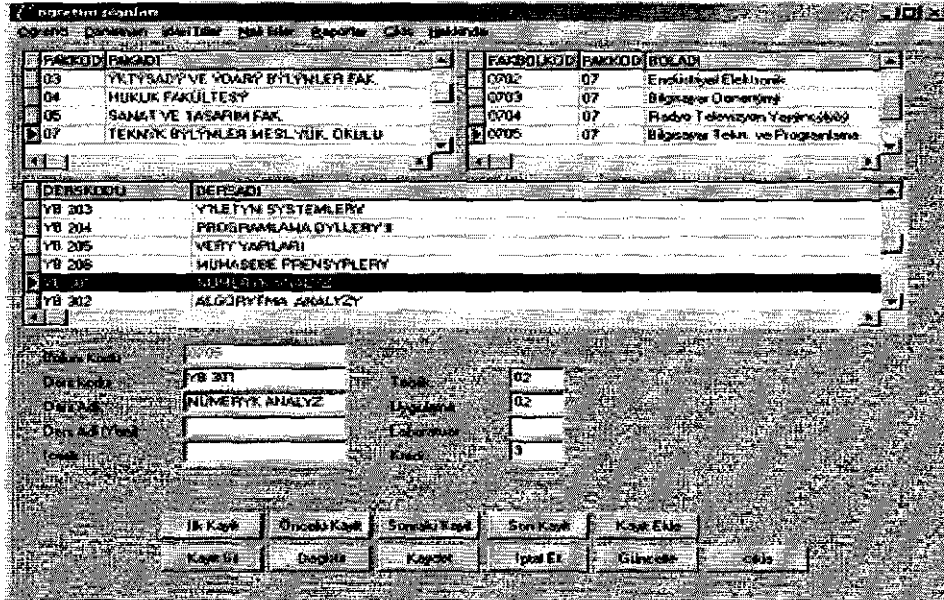


Şekil 8.6 Excel'e gönderilmiş veriler



Şekil 8.8 Resim eklemek veya değiştirmek için resmin bulunduğu (Şekil 8.3 te de görüldüğü) yerde çift tıklama yapılır açılan listeden resim seçilir ve open uygulanır. Resim bu şekilde öğrenci için ekrana atanmış olur.

Yeni başlayacak öğretim yılı için öncelikle ders planları yapılmak zorundadır. Bu amaçla hazırlanan “Öğretim Planları” formunda fakülte ve bölüm bilgilerinin seçiminin ardından açılacak olan dersler girilmekte ve kaydedilmektedir.



Şekil 8.9 Öğretim planları ekranı

Açılan dersleri atamak için fakülte, bölüm, şube, yıl / s0mester seçimlerinin ardından Kayıt ekle butonu tıklanır ve Planlardan aktarılmak istenen dersin üzerinde tıklama

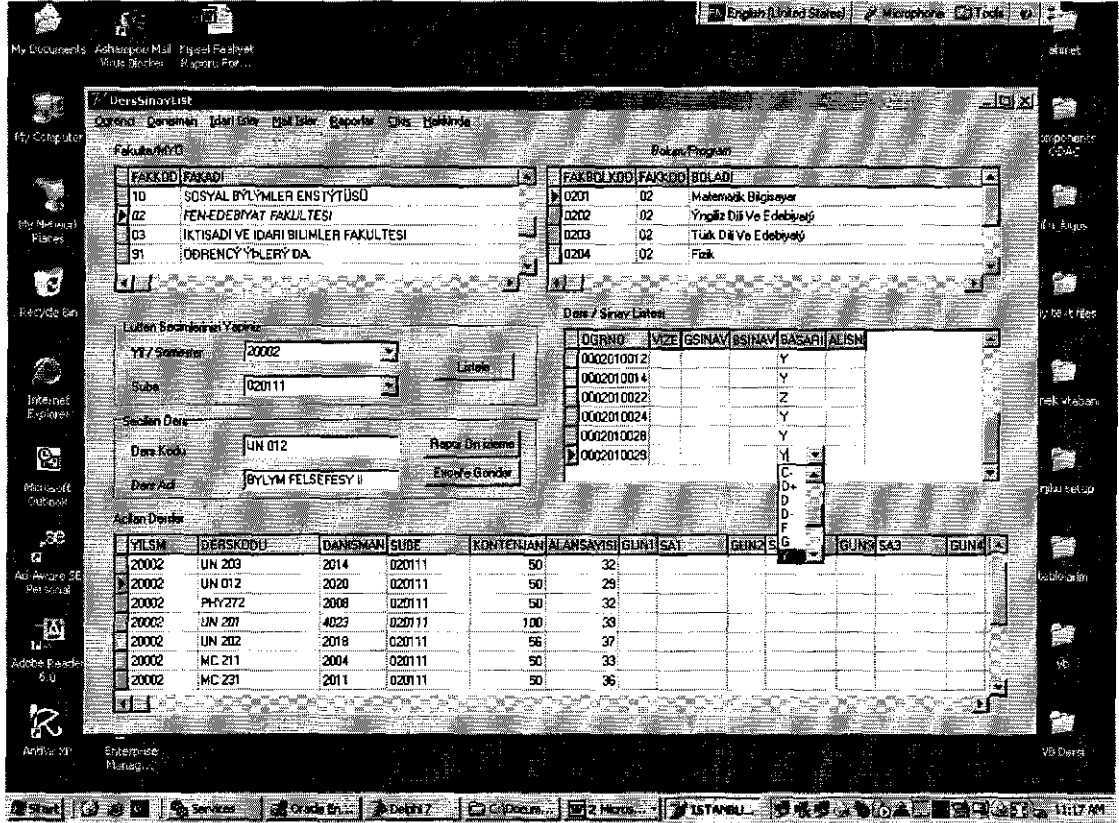
yapılır. Yazım kutularına eksik kalan bilgiler girilerek Kaydet butonu ile planlanan ders, açılan ders tablosuna eklenir. Burada “Kontenjan” bilgisi mutlaka bilgi girilmesi gerekli olan bir alandır.

The screenshot shows a software interface for selecting courses. It includes several input fields and a table. The table at the bottom has the following data:

YILIM	DERSKODU	DANISMAN	SURE	KONTENJAN	ALANSAYISI	GUN1	SAT	GUN2	GAZ
20042	YB 201	7003	070511	100		1	3		

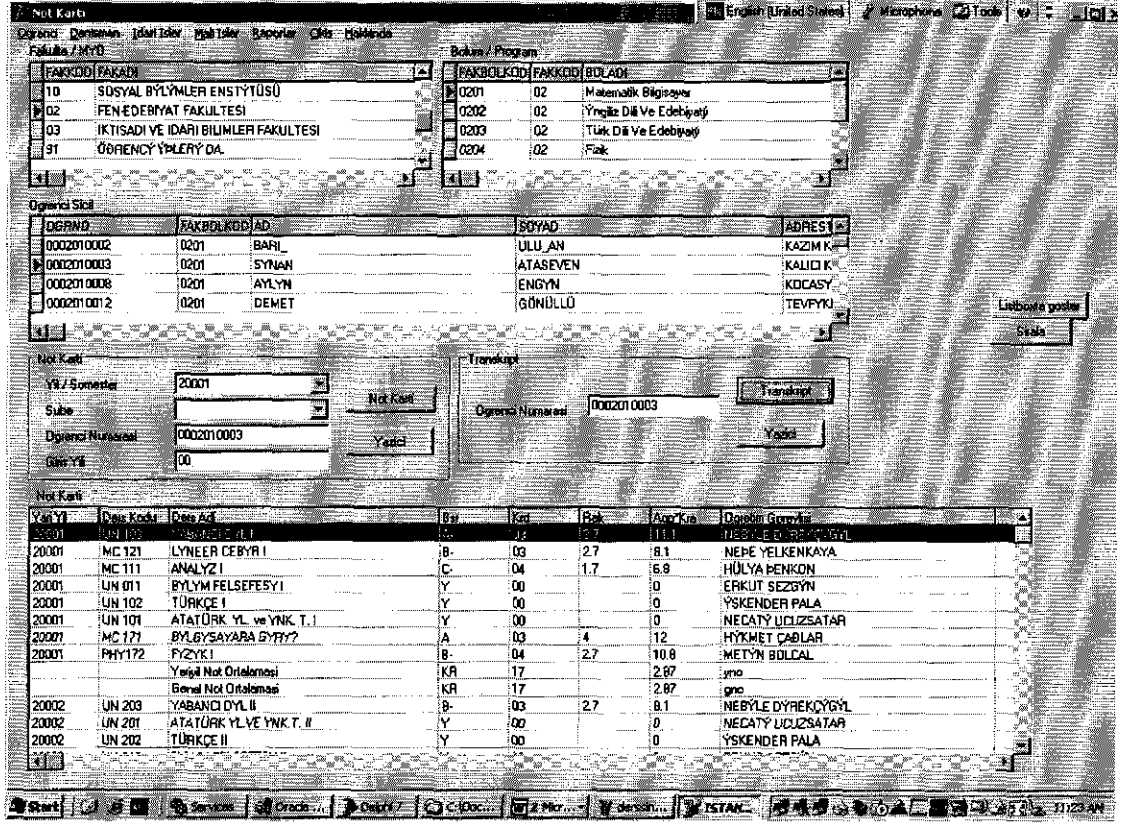
Şekil 8.10 Açılan Dersler Ekranı

Ders alma işlemlerinde fakülte, bölüm, yıl/sömester, şube ve öğrenci seçimlerinin ardından, açılmış derslerden alınması istenilen ders üzerinde çift tıklama ile dersi ilgili öğrencinin alınmış dersler listesine eklemek mümkündür.



Şekil 8.11 Ders İsteme Ekranı

Not Kartı'nda fakülte, bölüm, öğrenci, şube ve yıl / yarıyıl seçimleri yapıldıktan sonra "Not Kartı" botunu tıklanır. En altta görülen Not Kartı penceresinde öğrencinin aldığı bütün dersler ve bunlara ilişkin başarı durumu ekrana gelmektedir.



Şekil 8.12 Not Kartı Ekranı

Gerekirse “Not Kartı Önizleme” butonu ile dönem raporu önizleme ekranı veya buradan yazıcı çıktısı elde edilebilir. Ayrıca ilgili öğrenciye ait Transkript’ de bu ekranda önce “Transkript” butonu ile Not Kartı alanında görüntülenir ve “Transkript Önizleme” butonu ile de önizleme ekranı veya Not Kartı’nda olduğu gibi yazıcı çıktısı alınabilir.

Yıl Yıl	Ders Kodu	Ders Adı	Snr	Krs	Psk	Avt/Krs	Dersin Grubu
20001	UN 002	YABANCI DYL II	B-	03	2,7	8,1	NEBYLE DYREKÇYGYL
20001	MC 121	LYNEER CEBYR I	B-	03	2,7	8,1	NEBE YELKENKAYA
20001	MC 111	ANALYZ I	C-	04	1,7	6,9	HÜLYA BENKON
20001	UN 011	BYLYM FELSEFESY I	Y	00		0	ERKUT SEZGYN
20001	UN 102	TÜRKÇE I	Y	00		0	YSKENDER PALA
20001	UN 101	ATATÜRK YL ve YNK T. I	Y	00		0	NECATY ULUZSATAR
20001	MC 177	BYLYSAYABA B'YRY??	A	03	4	12	HYKMET CADLAR
20001	PHY172	FYZYK I	B-	04	2,7	10,8	METYN BOLCAL
		Yarijl Not Ortalamasi	KR	17		2,87	yno
		Genel Not Ortalamasi	KR	17		2,87	yno
20002	UN 203	YABANCI DYL II	B-	03	2,7	8,1	NEBYLE DYREKÇYGYL
20002	UN 201	ATATÜRK YL VE YNK T. II	Y	00		0	NECATY ULUZSATAR
20002	UN 202	TÜRKÇE II	Y	00		0	YSKENDER PALA
20002	UN 012	BYLYM FELSEFESY II	Y	00		0	ERKUT SEZGYN
20002	MC 271	PROGRAMLAMA DYLLERY I	A	03	4	12	HYKMET CADLAR
20002	MC 231	ANALYTYK GEOMETRY	C-	04	2	8	AYTEN KOÇ
20002	MC 211	ANALYZ II	F	04	0	0	HÜLYA BENKON
20002	PHY272	FYZYK II	F	04	0	0	METYN BOLCAL
		Yarijl Not Ortalamasi	KR	18		1,56	yno
		Genel Not Ortalamasi	KR	35		2,20	yno

Şekil 8.13 Not Kartı önizleme

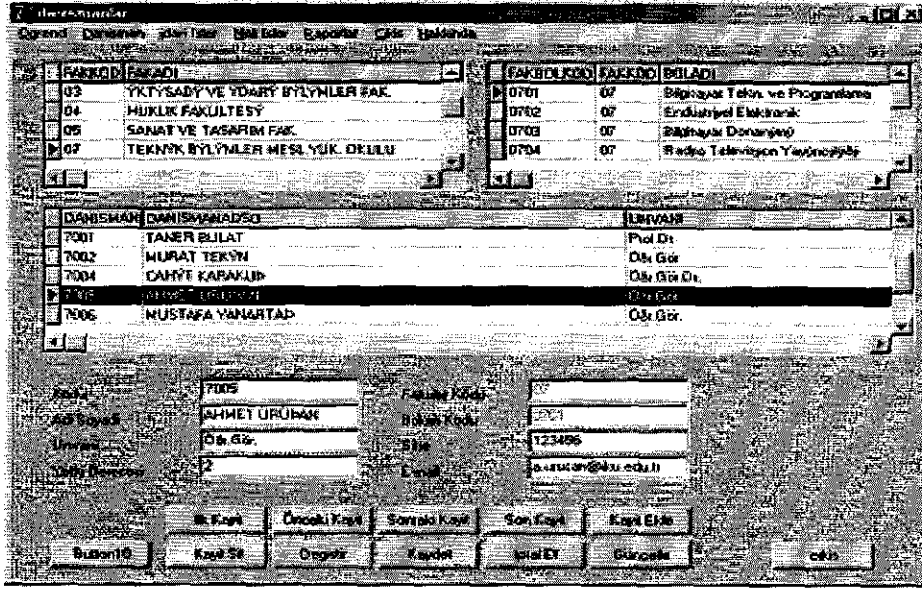
Yarıyıl	Ders Kodu	Ders Adı	Bir	Kr	Bak	Adı/Kı	Öğren Görevi
20001	UN 103	YABANCI DYL I	A	03	2,7	11,1	NEBİLE DİREKÇİYİL
20001	MC 121	LYNEER CEBYR I	B-	03	2,7	8,1	NEPE YELKENKAYA
20001	MC 111	ANALYZI	C-	04	1,7	6,8	HÜLYA BENKON
20001	UN 011	BYLYM FELSEFESY I	Y	00		0	ERKUT SEZGYN
20001	UN 102	TÜRKÇE I	Y	00		0	YSKENDER PALA
20001	UN 101	ATATÜRK YL ve YNK. T. I	Y	00		0	NECATY UCLZSATAR
20001	MC 171	DYLGYSAYARA GYRY?	A	03	4	12	HYKMET ÇADLAR
20001	PHY172	FYZYK I	B-	04	2,7	10,8	METYN BOLCAL
		Yarıll Not Ortalamasi	KR	17		2,87	no
		Genel Not Ortalamasi	KR	17		2,87	no
20002	UN 203	YABANCI DYL II	B-	03	2,7	8,1	NEBYLE DİREKÇİYİL
20002	UN 201	ATATÜRK YL VE YNK. T. II	Y	00		0	NECATY UCLZSATAR
20002	UN 202	TÜRKÇE II	Y	00		0	YSKENDER PALA

Şekil 8.14 Transkript önizleme

Mezun olmuş öğrencilerin kayıtlarının arşivlenebilmesi için geliştirilen “Arşiv” formuna bakıldığında ise, mezunlar tablosuna taşınan öğrenci bilgisi öğrenci sicil tablosundan silinir.

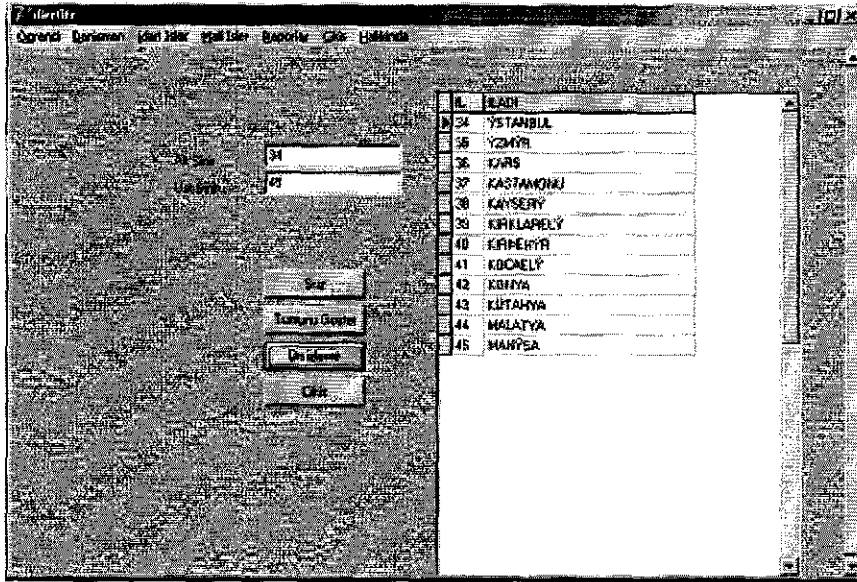
Şekil 8.15 Arşiv Ekranı

İdari görevler, Bölümler, Danışmanlar, İller, Okullar, Burslar gibi fazla hareketli olmayan tablolar aşağıda görüldüğü şekilde tasarlanmıştır. Gereğinde ilgili alanlara yeni bilgiler girilerek tabloya eklenir.



Şekil 8.16 Danışmanlar ekranı

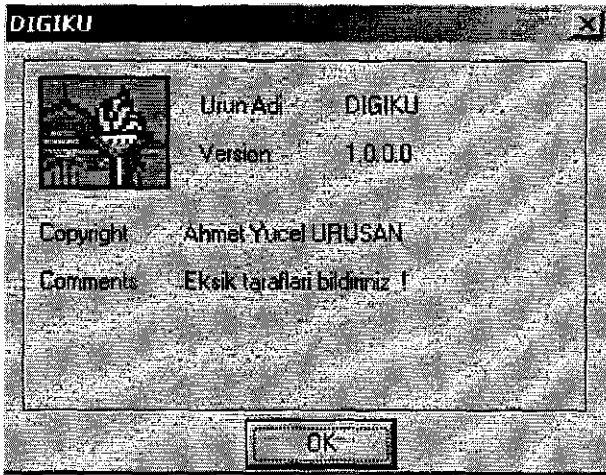
Raporlar menüsünde yer alan bazı raporlar için şu söylenebilir. İller tablosunda olduğu gibi alt ve üst sınır alanlarından herhangi birine veya ikisine birden tablodaki alanlardan, iller tablosu için il kodu, girilerek filtreleme işlemi yapılır. Gerekli görüldüğü takdirde “Önizleme” butonu ve peşinden yazıcıya gönderilebilir.



Şekil 8.17 Örnek Rapor Ekranı

İler	İ adı
34	İSTANBUL
35	İZMİR
36	KARS
37	KASTAMONU
38	KAYSERİ
39	KIRLAELİ
40	KIRSEHR
41	KOCAELİ
42	KONYA
43	KÜTAHYA
44	MALATYA

Şekil 8.18 Örnek Rapor ekranı yazıcı çıktısı



Şekil 8.19 Program Hakkında Bilgi Ekranı

Yeni geliştirilmeye çalışılan programların hatalarının giderilmesi, ilgili kurumun yapısına tam olarak uyumlu hale gelerek güvenli bir şekilde kullanılması, zaman alan bir konudur. Bu anlamda öncelikle yazılımı yapan kişi veya grupların ellerindeki verilerin tamamını kullanarak ve programı kullanacak kişilerle zaman zaman görüşerek, gereğinde saatlerce üzerinde birlikte çalışarak, büyük emek sarfetmeleri

gerekmektedir. Programın hazır hale gelip kullanılmaya başlanması ile birlikte bu kez kullanıcılar, yazılım aşamasında görülemeyen hataları yazılımcı gruba bildirerek programdaki hataların en kısa zamanda giderilmesine yardımcı olurlar. Örnek pencerede programı yazan programcı, programın adı ve versiyon bilgileri yer almaktadır. Kullanıcılar hangi versiyonu kullandıkları bilgisini bu pencereden edinerek, yeni versiyonu edinmek konusunda harekete geçebilirler.

SONUÇ

Çalışmanın başında amaçlandığı şekilde Oracle Veritabanı Uygulama Sistemine bağlı olarak çalışan “Üniversite Bilgi Sisteminin” geliştirilmesi Öğrenci Otomasyonu bazında gerçekleştirilmiştir.

Bu yapılırken 1999 senesinden beri kullanılmakta olan programdaki veriler başarıyla Oracle’a aktarılmıştır. Oracle’ın kurulmuş olduğu makine, Active Directory oluşturulmuş ve Domain tanımı yapılmış ‘ahmet.urusan.com’, Windows 2003 Server işletim sistemine sahiptir. Active Directory, Oracle ve programda tanımlanmış kullanıcılar haricinde veritabanına erişim tamamıyla yasaklanmıştır. Program kullanıcıları da aldıkları kodlara göre erişim hakkına sahiptirler. Burada basit olarak Öğrenci ve Öğretim üyesi için değişik menülü ekran düzenlenmiştir.

Bu temel işlemlerden hareketle kullanıcıların ve üniversitenin diğer birimlerinin taleplerini karşılayacak şekilde programın geliştirilmesi mümkündür.

KAYNAKÇA

- Ankara Üniversitesi Bilgi İşlem Daire Başkanlığı. Kullanımda Olan Projeler. 02 / 08 / 2004. < <http://www.bid.ankara.edu.tr/projeler.html> >.
- Bayramlı, Burak. Oracle SQL*Loader. 02 / 08 / 2004. < <http://www.bilgiyönetimi.org> >.
- Computer World Türkiye. Oracle Türkiye, 15 Yıllık Yerel Bilgi Birikimiyle Geleceğe Bakıyor. 03 / 07 / 2004. 02 / 08 / 2004. < <http://computerworld.com.tr/printbottom.php?qid=32559> >.
- Computer World Türkiye. Oracle Openworld Kapılarını Londra'da Açıyor. 07 / 07 / 2004. 02 / 08 / 2004. < <http://computerworld.com.tr/printbottom.php?qid=32559> >.
- Constraints and Triggers. < <http://www-db.stanford.edu/~ullman/fcdb/oracle/or-triggers.html> >.
- DataFlux, Statistical Solutions. dfPower DBMS / Copy. 24 / 05 / 2005. < <http://www.statsol.ie/dbmscopy/dbmscopy.htm> >.
- Demirli, Nihat and Yüksel İnan. Zirvedeki Beyinler – 6 Doruk Noktası Delphi 7 Veritabanı ve Network Programcılığı. İstanbul : Prestige Education Center, 2003.
- Getting Started With Oracle. < <http://www-db.stanford.edu/~ullman/fcdb/oracle/or-intro.html> >.
- Grof, R. James and Paul N. Weinberg. SQL. Osborne : McGraw-Hill, 2002.
- Karagülle, İhsan. Borland Delphi 7. İstanbul : Türkmen, 2003.
- Kıral, Cenk. Kariyer ve Yönetim Kulübü. Oracle ve CRM. 2004. 02 / 08 / 2004. < http://www.kykonline.com/index.php?sk_id=kariyer&rp_id=6 >.
- Koopman, James. Transferring Operating System Files from within Oracle. 24 / May / 2005 < <http://www.dbasupport.com/oracle/ora9i/transOSfiles02.shtml> >.
- Note : 146599.1. Diagnosing and Resolving Error ORA-04031. Oracle Metalink Page. 23 / May / 2001. 23 / June / 2005. < <http://metalink.oracle.com> >.
- Note : 1012046.6. Oracle 7: How to Calculate Your Shared Pool Size. Oracle Metalink Page. 24 / Aug / 1995. 21 / Dec / 2004. 28 / 07 / 2005. < <http://metalink.oracle.com> >.
- Note : 1019814.6. Win, WinNT, Win 95 : NLS_Date_Format in init.ora is not used by client. 19 / July / 1996. 05 / Nov / 2003. 18 / May / 2005. < <http://metalink.oracle.com> >.
- Note : 227334.1. Dates, Timestamps & Timezones – Frequently Asked Questions. 13 / Apr / 2005. 18 / 05 / 2005. < <http://metalink.oracle.com> >.

- Note : 18622.1. OERR : ORA 1031 “insufficient privileges”. 25 / Feb / 1998. 18 / Apr / 2004. 11 / 03 / 2005. < <http://metalink.oracle.com> >.
- Note : 1005146.6. ORA-942 Even Though User Has Been Granted Privileges on Object. 22 / Aug / 1994. 10 / Jan / 2003. 11 / 03 / 2005. < <http://metalink.oracle.com> >.
- Note : 1062335.6. ORA-942 when select from any v\$view within stored PL/SQL procedure 26 / Nov / 1998. 01 / Feb / 2002. 11 / 03 / 2005. < <http://metalink.oracle.com> >.
- Note : 107843.1. Grant Object Privileges on Another Schema Object to Other Users as SYSTEM or SYS. 06 / May / 2000. 12 / May / 2003. 11 / 03 / 2005. < <http://metalink.oracle.com> >.
- Note : 222860.1. ORA-00942 When Commit DML Transaction on a Base Table Belonging to a MV. 09 / Dec / 2002. 02 / Jan / 2003. 11 / 03 / 2005. < <http://metalink.oracle.com> >.
- Note : 170740.1. 9i:External Tables Overview and Usage. 27 / Jun / 2001. 25 / Jun / 2002. 27 / 04 / 2005. < <http://metalink.oracle.com> >.
- Note : 264893.1. TMS Release 4.0.5 iPatch 4.0.5.11 Release Notes, Rev. 1.0. 03 / Mar / 2004. 27 / 04 / 2005. < <http://metalink.oracle.com> >.
- Note : 241047.1. The Priority of NLS Parameters Explained. 27 / 04 / 2005. < <http://metalink.oracle.com> >.
- Note : 1022579.6. ORA-1031 “Insufficient Privileges” or Access Denied When Creating Instance Via ORADIM7X. 22 / Oct / 1996. 13 / May / 2004. 09 / 11 / 2004. < <http://metalink.oracle.com> >.
- Note : 213416.1. RAC: Troubleshooting Windows NT/2000 Service Hangs. 04 / Oct / 2002. 22 / Jan / 2003. 09 / 11 / 2004. < <http://metalink.oracle.com> >.
- Note : 168168.1. Getting ORA-942 or ORA-1031 and PLS-201 or ORA-28111 in PL/SQL, works in SQL*Plus. 13 / Dec / 2001. 08 / Apr / 2003. 09 / 11 / 2004. < <http://metalink.oracle.com> >.
- Note : 121443.1 Error: ORA3113 When Deleting Records from Designer Generated Table. 05 / Oct / 2000. 11 / Oct / 2000. 05 / 11 / 2004. < <http://metalink.oracle.com> >.
- Note : 1005393.6 ORA-1034 ORACLE NOT AVAILABLE Connectint to Database PROBLEM. 09 / Sep / 1994. 20 / Dec / 2002. 08 / 11 / 2004. < <http://metalink.oracle.com> >.
- Note : 121737.1 TNS-12500 “Failed to start a dedicated server process” in Windows OPS environment. 10 / Oct / 2000. 20 / Mar / 2003. 08 / 11 / 2004. < <http://metalink.oracle.com> >.
- Note : 1036696.6. ORA-12505 and TCP/IP ERROR 61 Connecting to Datamart Repository. 07 / Oct / 1997. 26 / Mar / 2003. 09 / 11 / 2004. < <http://metalink.oracle.com> >.
- Note : 168164.1. ORA-00900 Error Source. 09 / 11 / 2004. < <http://metalink.oracle.com> >.

Note : 107321.1. Logging Into Enterprise Manager Console Fails with VTK-1000 If OMS isn't Started. 02 / May / 2000. 12 / Apr / 2002. 09 / 11 / 2004. < <http://metalink.oracle.com> >.

Note : 1079800.6. FRM-40510 / ORA-01031 Insufficient privileges. 09 / 11 / 2004. < <http://metalink.oracle.com> >.

ODAC Component. < <http://component.sitemynet.com/odac510.rar> >.

ODAC Component Kurulumu. < <http://component.sitemynet.com/kurulum.ppt> >.

ODTU Güvenlik sayfası. Oracle Veritabanı'nda Veri ve Sistem Güvenliği. 17 / 03 / 2002. 02 / 08 / 2004. < <http://security.metu.edu.tr/belge.php?oracle.html> >.

Oracle 9i SQL Versus Standart SQL. < <http://www-db.stanford.edu/~ullman/fcdb/oracle/or-nonstandart.html> >.

Oracle Dates and Times. < <http://www-db.stanford.edu/~ullman/fcdb/oracle/or-time.html> >.

Oracle Veritabanı Dersi. 19 / 07 / 2004. < <http://serdar.ktg.com.tr/oracle.html> >.

Oracle Sozluk. < <http://sozluk.sourtimes.org/show.asp?t=oracle> >.

Oracle ile İlgili Web Adreslerinden bazıları. 02 / 08 / 2004. < <http://angelfire.com/ak4/ozansari/oracle.html> >.

ORA-00942. 11 / 03 / 2005.< http://www.adp-gmbh.ch/ora/err/ora_00942.html>.

Öztürk, Ali. Oracle Veritabanı Yönetim Sistemi. İstanbul : Atlas, 2003.

Pala, Zeydin. Borlan Delphi 7 Uygulama ve Geliştirme Rehberi 34 Konuda Delphi. İstanbul : Türkmen, 2003.

Sun Microsystem Türkiye – Basın Bültenleri – Oracle. Oracle, ERP Konsolidasyonu için SUN'ı seçti. 2004. 02 / 08 / 2004. < <http://tr.sun.com/media/bulten/2004/040428.html> >.

Şen, Osman Nihat. Oracle (9i) SQL, SQL*PLUS, PL/SQL ve Veritabanı Yönetimi. İstanbul : Beta, 2004.

Turk.Internet.Com. Oracle'ın EMEA bölgesinde Pazar Payı % 13 arttı. 14 / 07 / 2004. < <http://turk.internet.com/haber/yaziyaz.php3?yaziid=10454> >.

Using The Oracle Bulk Loader. < <http://www-db.stanford.edu/~ullman/fcdb/oracle/or-load.html> >.

Using Oracle PL/SQL. < <http://www-db.stanford.edu/~ullman/fcdb/oracle/or-plsql.html> >.

Yağımlı, Mustafa and Fevzi Akar. Delphi 5 ile İnternet Programlama. İstanbul : Beta, 2001.