

T.C.
MALTEPE ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

**VERİTABANI UYGULAMALARINDA PERFORMANS VE GÜVENLİK
SORUNLARI VE ÖNERİLEN ÇÖZÜMLER**

YÜKSEK LİSANS TEZİ

Halil KAYA

Tez Danışmanı

Prof. Dr. Kemal KÖYMEN

İSTANBUL – 2011

Bu tez çalışması, Maltepe Üniversitesi Fen Bilimleri Enstitüsü Yönetim Kurulu'nun / / tarih ve / sayılı kararıyla oluşturulan jüri tarafından **Bilgisayar Mühendisliği Yüksek Lisansı Tezi** olarak kabul edilmiştir.

JÜRİ

Prof. Dr. Kemal KÖYMEN
Danışman

Prof. Dr.
Üye

Prof. Dr.
Üye

ÖZET

Yüksek Lisans Tezi, Veritabanı Uygulamalarında Performans ve Güvenlik Sorunları ve Önerilen Çözümler, T.C. Maltepe Üniversitesi, Fen Bilimleri Enstitüsü, Bilgisayar Mühendisliği Anabilim Dalı.

Bu tez çalışmasında veritabanında bahsedilmiş, veritabanı türleri ve günümüzde yaygın kullanılan ticari veritabanları incelenmiştir. En çok kullanılan oracle veritabanı üzerinden çalışan uygulamaların güvenlik ve performans sorunlarının tespit edilmesi ve çözüm önerilerinin yapan bir yazılım geliştirilmiştir. Geliştirilen yazılım, oracle veritabanı kullanan kullanıcıların gelecekle ilgili donanım ve kaynak planlamasının yapılmasına yardımcı olmaktadır. Yazılımın içindeki örnek uygulamalar ve sorun çözme yöntemleriyle veritabanı yöneticilerinin bilgilerinin gelişmesine katkı sağlamaktadır. Aynı zamanda veritabanı ve uygulamalarının sağlıklı ve güvenli çalışmasına yardımcı olmaktadır.

Tez 2011 yılında yapılmıştır ve 128 sayfadan oluşmaktadır.

Anahtar Kelimeler: Oracle Veritabanı, Veritabanı Performans, Veritabanı Güvenliği, Uygulama Güvenliği, DBA, Veritabanı Türleri, Oracle Veritabanı Yapısı.

ABSTRACT

Master Thesis, Performance and Security Problems of Database Applications and Proposed Solution, T.C. Maltepe University, Graduate School of Natural and Applied Sciences, Department of Computer Engineering.

In the scope of this thesis, database concept and the types of databases are mentioned, widely used commercial databases are examined. Since oracle database is the most used and widely known database, a software application has been developed that discover, determine, make suggestions in order to solve the security and performance issues in the oracle databases. Developed software helps oracle database users with hardware and resource planning in the future. The best practices and samples in the software contributes significantly to expand database administrator's knowledge and ability to manage the database. This software application ensures database and their applications running healthy and secure at the same time.

This thesis has been completed in 2011 and it has 128 pages.

Keywords: Oracle Database, Database Performance, Database Security, Application Security, DBA, Type of Database, Oracle Architecture.

TEŐEKKÜR

Öncelikle yüksek lisans öğrencilik yıllarım içerisinde beni akademik arařtırmaya yüreklendiren ve yönlendiren Maltepe Üniversitesi Fen Bilimleri Enstitüsü Müdürü Sayın Prof. Dr. İlhami YAVUZ'A, geliřtirmiş olduđum bu tez boyunca beni yönlendiren, yardımını esirgemeyen ve kendime örnek aldıđım Maltepe Üniversitesi Rektörü değerli danışman hocam Sayın Prof. Dr. Kemal KÖYMEN'E ve tez yazım aşamasından dokümanı gözden geçirip fikirlerini öne süren, yazım düzen ve kurallarıyla ilgili katkı sađlayan Maltepe Üniversitesi Yazılım Mühendisliđi bölümünden Sayın Yrd. Doç. Dr. Turgay Tugay BİLGİN'E teőekkürü borç bilirim.

Annem ve Babama...

İÇİNDEKİLER

Sayfa

ÖZET.....	III
ABSTRACT.....	IV
TEŞEKKÜR.....	V
1 GİRİŞ.....	1
2 VERİTABANI HAKKINDA GENEL BİLGİLER.....	2
2.1 Veritabanı.....	2
2.2 Veritabanı Türleri.....	2
2.2.1 Hiyerarşik Veritabanı.....	3
2.2.2 İlişkisel Veritabanı.....	3
2.2.3 Nesnesel Veritabanı.....	4
2.3 Ticari Veritabanı Yazılımları.....	4
2.3.1 Microsoft Access.....	4
2.3.2 MySQL.....	5
2.3.3 IBM DB2.....	6
2.3.4 Microsoft SQL Server.....	6
2.3.5 Oracle.....	7
3 ORACLE VERİTABANI.....	9
3.1 Oracle Çekirdek Katmanı.....	10
3.2 Oracle Veritabanı Mimari Yapısı.....	14
3.3 Oracle Veritabanı Bellek Yapısı.....	15
3.3.1 Sistem Genel Alanı (SGA).....	16
3.3.2 Program Genel Alanı (PGA).....	18
3.4 Görev Yapıları.....	19
3.4.1 Bir Kullanıcının Veritabanına Bağlanması.....	20
3.4.2 Arka Plan Görevler.....	22
3.5 Oracle Veritabanı Kütük Yapısı.....	23
3.6 Oracle Veritabanı Fiziksel Yapısı.....	24
3.6.1 Kontrol Kütükleri.....	25
3.6.2 Veri Kütükleri.....	25
3.6.3 Online Redo Log Kütükleri.....	26
3.6.4 Parametre Kütüğü.....	26
3.6.5 Yedek Kütükler.....	26
3.6.6 Arşiv Kütükleri.....	26
3.6.7 Şifre Kütüğü.....	27
3.6.8 Hata ve Uyarı İzleme Kütüğü.....	27
3.7 Oracle Veritabanı Mantıksal Yapısı.....	28
3.7.1 Tablo Uzayı.....	28
3.7.2 Veritabanı Sistem Tablo Uzayları (System ve Sysaux Tablespace).....	29
3.7.3 Veri Saklama Yapıları.....	30
3.8 Oracle iyileştirici.....	41
3.8.1 Optimize etme.....	41
4 VERİTABANI PERFORMANS VE GÜVENLİK ARAÇLARI.....	45
4.1 Toad 9.6 (Quest Software).....	45

4.2	KeepTool 7.2 (Tool for Oracle Database)	46
4.3	DBHawk (DataSparc, Inc.)	46
4.4	DB PowerStudio for Oracle (Embarcadero)	47
4.5	DBTools 6.1.0 (SoftTree Technologies)	47
4.6	Oracle Enterprise Manager ve Grid (Oracle)	48
5	VERİTABANI UYGULAMALARINDA PERFORMANS VE GÜVENLİK SORUNLARI VE ÖNERİLEN ÇÖZÜMLER	50
5.1	DAPT 'ın Tanımı	52
5.2	Uygulamanın Geliştirileceği Ortamın Belirlenmesi	54
5.3	Kullanım Durumu (Use Case)	54
5.4	UML Diyagram	56
5.5	Kullanıcı Arayüzleri	58
5.5.1	Sisteme Giriş	58
5.5.2	Ana Ekran	58
5.6	Performans	61
5.6.1	Sistem Global Alanı	62
5.6.1.1	Tespit Edilen Sorunlar ve Çözüm Önerileri	63
5.6.2	Oturum ve Görev Parametreleri	64
5.6.2.1	Tespit Edilen Sorunlar ve Çözüm Önerileri	64
5.6.3	Kaynak Limitleri	65
5.6.4	Kontrol Kütüğü	65
5.6.4.1	Tespit Edilen Sorunlar ve Çözüm Önerileri	65
5.6.5	Redo-Log Kütükleri	67
5.6.5.1	Tespit Edilen Sorunlar ve Çözüm Önerileri	68
5.6.6	Redo Log Anahtarlanması	69
5.6.6.1	Tespit Edilen Sorunlar ve Çözüm Önerileri	69
5.6.7	Arşiv Kütüklerinin Boyutu	71
5.6.8	Tablo Uzayı ve Veri Kütükleri	71
5.6.8.1	Tespit Edilen Sorunlar ve Çözüm Önerileri	72
5.6.9	Otomatik Disk Yönetimi	74
5.6.10	Otomatik Geri Alma	74
5.6.10.1	Tespit Edilen Sorunlar ve Çözüm Önerileri	75
5.7	Veritabanı Nesneleri	76
5.7.1	Tablo ve İndekslerin Farklı Tablo Uzayında Bulunması	77
5.7.1.1	Tespit Edilen Sorunlar ve Çözüm Önerileri	77
5.7.2	Üzerinde İndeks Olmayan Tablolar	78
5.7.2.1	Tespit Edilen Sorunlar ve Çözüm Önerileri	79
5.7.3	Üzerinde Fazla İndeks Olan Tablolar	80
5.7.3.1	Tespit Edilen Sorunlar ve Çözüm Önerileri	81
5.7.4	Birinci Anahtar Kısıtlaması Olmayan Tablolar	82
5.7.4.1	Tespit Edilen Sorunlar ve Çözüm Önerileri	82
5.7.5	İndeks Olmayan Yabancı Anahtar	83
5.7.5.1	Tespit Edilen Sorunlar ve Çözüm Önerileri	84
5.7.6	Pasif Kısıtlamalar	85
5.7.6.1	Tespit Edilen Sorunlar ve Çözüm Önerileri	85
5.7.7	Analizi Yapılmamış Tablo ve İndeksler	86
5.7.7.1	Tespit Edilen Sorunlar ve Çözüm Önerileri	86
5.7.8	Paralel Sorgu Kullanmak	88
5.7.8.1	Tespit Edilen Sorunlar ve Çözüm Önerileri	89
5.7.9	Veritabanı Kilitleri	90

5.8	Veritabanı Güvenliği	92
5.8.1	Güvenlik Ayarları	93
5.8.1.1	Tespit Edilen Sorunlar ve Çözüm Önerileri	93
5.8.2	Geçersiz Nesneler	94
5.8.2.1	Tespit Edilen Sorunlar ve Çözüm Önerileri	94
5.8.3	Veritabanı Kullanıcıları	96
5.8.3.1	Tespit Edilen Sorunlar ve Çözüm Önerileri	97
5.8.4	“With Admin” Seçeneği ile Verilen Yetkiler	102
5.8.5	Veritabanı Sistem Yetkileri	103
5.8.6	Kullanıcılarının Uygulama Tabloları Üzerindeki Yetkileri	105
5.8.6.1	Tespit Edilen Sorunlar ve Çözüm Önerileri	105
5.9	Oracle Performans Analiz Raporu	108
5.9.1	Bellek Kullanım Analizi	109
5.9.2	Merkezi İşlem Birimi Maliyet Analizi	110
5.9.2.1	Tespit Edilen Sorunlar ve Çözüm Önerileri	111
5.9.3	Girdi Çıktı Analizi	112
5.9.3.1	Tespit Edilen Sorunlar ve Çözüm Önerileri	113
5.9.4	Maliyetli SQL Sorguları	114
5.9.4.1	Tespit Edilen Sorunlar ve Çözüm Önerileri	114
5.9.5	Yüksek Kaynak Tüketimi	116
5.9.5.1	Tespit Edilen Sorunlar ve Çözüm Önerileri	117
5.10	Veritabanı Alert Log Analizi	118
5.10.1	Tespit Edilen Sorunlar ve Çözüm Önerileri	119
5.11	DAPT Özellikleri	121
6	SONUÇ	124

KISALTMALAR

Kısaltmalar	İngilizce	Türkçe
API	Application Programming Interface	Uygulama Geliştirme Ara yüzü
AWR	Automatic Workload Repository	Otomatik istatistiksel verileri raporu
ASM	Automatik Storage Management	Otomatik Disk Yönetimi
BDE	Borland Database Engine	Borland Veritabanı Motoru
CPU	Central Processing Unit	Merkezi İşlem Birimi
DDL	Data Definition Language	Veri ve Veri Modeli Tanımlama Dili
DML	Data Manipulation Languag	Veri İşleme Dili
DBA	Database Administrator	Veritabanı Yöneticisi
DOP	Degree of parallelism	Paralel Sayısı
DAPT	Database Application Performance Tool	Veritabanı Performans Uygulaması
IBM	International Business Machines	Uluslararası İş Makineleri
IMS	Information Management System	Bilgi Yönetim Sistemi
ITPM	Information Technology Project Management	Bilgi Teknoloji Proje Yönetimi
I/O	Input/Output	Girdi/Çıktı
LAN	Local Area Network	Yerel Alan Ağı
LOB	Large object	Büyük Nesne
Net8	Network product of Oracle	Oracle'ın ağ ürünü
OCI	Oracle Call Interface	Oracle Bağlantı Ara yüzü
OS	Operating System	İşletim Sistemi
OLAP	Online Analytical Processing	Çevrimiçi Analitik İşleme
OLTP	Online Transaction Processing	Çevrimiçi Hareket İşleme
ODAC	Oracle Data Access Component	Direkt Oracle Bağlantısı
ODBC	Open Database Connectivity	Serbest Veritabanı Bağlanabilirliği
OPI	Oracle Progame Interface	Oracle Program Ara yüzü
PL/SQL	Oracle Structured Query Language	Oracle'ın SQL komutları
PGA	Program Global Area	Program Genel Alanı
RDBMS	Relation Database Management Systems	İlişkisel Veri Tabanı Yönetim Sistemi
RAC	Real Application Cluster	Gerçek Uygulama Kümeleri
RAID	Redundant Array of Independent Disks	Bağımsız Disklerin Artıklıklı Dizisi
RAW	Raw Divece	Ham Disk
RMAN	Recovery Manager	Oracle Veritabanı Yedek Yönetimi
SQL	Structured Query Language	Yapısal Sorgulama Dili

SP	Stastpact Report	Oracle Performans Analiz Raporu
SNAP	Snapshot	Tarihsel Olarak İstatistiksel Veriler
SCRIPT	Script	SQL Komut Dizisi
SGA	System Global Area	Sistem Genel Alanı
UPI	User Program Interface	Kullanıcı Program Ara yüzü
VTYA	Database Administration Tool	Veritabanı Yönetim Aracı
VTYS	Database Administration System	Veritabanı Yönetim Sistemi
VT	Database	Veritabanı
VTY	Database Administration	Veritabanı Yöneticisi
WAN	Wide Area Network	Geniş Alan Ağı

ŞEKİLLER

	Sayfa
Şekil 3.1. Oracle8i Mimari Yapısı	10
Şekil 3.2. Oracle Çekirdek Katmanı	11
Şekil 3.3. Oracle Veritabanı Sunucu Yapısı	15
Şekil 3.4. Ortak Havuz (Shared Pool) ve Bileşenleri	18
Şekil 3.5. Oracle İşlem Görevleri	19
Şekil 3.6. Oracle Kullanıcısının Veritabanı Bağlanması	21
Şekil 3.7. Oracle VT Kütüklerinin Fiziksel ve Mantıksal Yapısı	24
Şekil 3.8. Tablo Uzayı, VT Kütük ve Nesne Yapısı	29
Şekil 3.9. Segment-Boyut-Blok ilişkisi	30
Şekil 3.10. Oracle Veritabanı Fiziksel ve Mantıksal Yapısı	33
Şekil 3.11. Blok Yapısı	34
Şekil 3.12. Kayıt (Row) Yapısı	35
Şekil 3.14. PCTFREE Parametresi	37
Şekil 3.15. PCTFREE ve PCTUSED Arasındaki İlişki	37
Şekil 3.16. Kayıt Zincirlemek	40
Şekil 3.17. Kayıt Taşımak	41
Şekil 5.1. DAPT'nin Genel Kullanım Durum Diagramı	55
Şekil 5.2. Veritabanı Nesnelere İlgili Delphi UML Diyagramı	57
Şekil 5.3. Programa Giriş Ekran Görünümü	58
Şekil 5.4. Program Ana Ekran Görünümü	60
Şekil 5.5. Performans Modülü Ana Ekran Görünümü	62
Şekil 5.6. Kontrol Kütüğü Analizi Ekran Görünümü	67
Şekil 5.7. Redo-Log Kütükleri Analiz Ekran Görünümü	68
Şekil 5.8. Tablo Uzayı ve Veri kütüklerinin Analiz Ekran Görünümü	73
Şekil 5.9. Otomatik Geri Alma Kullanım Analiz Ekran Görünümü	75
Şekil 5.10. Veritabanı Nesnelere İlgili Görünüm Ekranı	76
Şekil 5.11. Tablo ve İndeks'lerin Tablo Uzayı Üslerindeki Dağılım Analiz Ekranı	78
Şekil 5.12. Uygulamaların Tablo ve İndeks Analiz Ekranı	87
Şekil 5.13. Paralel Sorgu Kullanım Analiz Ekranı	89

Şekil 5.14. Veritabanı Güvenlik Ekranı	92
Şekil 5.15. Uygulama Kullanıcılarına Verilen Sistem Yetki Analiz Ekranı	104
Şekil 5.16. Veritabanı Bellek Kullanımını Analiz Ekranı	110
Şekil 5.17. Merkezi İşlem Birimi Maliyet Analizini Gösteren Ekran	112
Şekil 5.18. Veri Kütüklerindeki I/O Analizini Gösteren Ekran	113
Şekil 5.19. Maliyetli SQL Sorgularının Ekran Görünümü	115
Şekil 5.20. Maliyeti Yüksek Olan İlk Beş Olayın Analiz Ekranı	118
Şekil 5.21. Alert Log Analiz Ekranı	119

TABLÖLAR

	Sayfa
Tablo 5.1. SGA Bellek Durumunu Gösteren Örnek Uygulama	63
Tablo 5.2. Kullanıcı Oturum ve İşlemleri Gösteren Örnek Uygulama	64
Tablo 5.3. Kontrol Kütüğü Durumunu Gösteren Örnek Uygulama	65
Tablo 5.4. Log Anahtarlama Durumunu Gösteren Örnek Uygulama	70
Tablo 5.5. Üzerinde İndeks Olmayan Tabloları Gösteren Örnek Uygulama	79
Tablo 5.6. Üzerinde Fazla İndeks Olan Tabloları Gösteren Örnek Uygulama	81
Tablo 5.7. Birinci Anahtar Kısıtlaması Olmayan Tabloları Gösteren Örnek Uygulama	83
Tablo 5.8. İndeks Olmayan Yabancı Anahtarları Gösteren Örnek Uygulama	84
Tablo 5.9. Pasif Kısıtlamaları Gösteren Örnek Uygulama	85
Tablo 5.10. Oracle Veritabanı Kilit tanımları	91
Tablo 5.11. Veritabanı Odit Parametreleri	93
Tablo 5.12. Geçersiz Nesnelere Gösteren Örnek Uygulama	94
Tablo 5.13. Veritabanı Kullanıcıları Gösteren Örnek Uygulama	99
Tablo 5.14. Örnek Veritabanı Kullanıcı Profil Tanımlama	100
Tablo 5.15. Örnek With Admin Seçeneğiyle Verilen Yetkiler	103
Tablo 5.16. Uygulama Tabloları Üzerinde Yetkileri Gösteren Örnek Uygulama	106

1 GİRİŞ

Bilgisayar ve iletişim teknolojilerindeki gelişmelere paralel olarak donanımın ucuzlaması verilerin uzun süre depolanmasına dolayısıyla büyük kapasiteli veritabanlarının oluşmasına neden olmuştur. Bu nedenle büyük veritabanlarında istenilen anlamlı, kullanılabilir ve ilginç bilgiye erişmek yeni bir disiplinin doğmasına sebep olmuştur.[1]

Bir veritabanı tasarlandığında, veritabanındaki verilere erişimin hızlı, güvenilir ve tüm önemli işlevleri gerçekleştiren yapıya sahip olması gerekir. Genel olarak veritabanlarındaki performans sorunları; veritabanı tasarımının yanlış yapılması, veritabanı kullanan kullanıcıların mevcut durumu gözden geçirerek gelecekte artan verileri tam olarak tahmin edememesi ya da yanlış hesaplaması, kapasite planlamasını sağlıklı yapmaması ve veritabanından kullanılan uygulama kodlarındaki algoritmaların veritabanının yapısına uygun çalışacak şekilde yazılmaması nedeniyle performans sorunları ortaya çıkmaktadır.

Bu tez çalışması kapsamında bu konu ile ilgili araçlar ve çalışmalar incelenmiş, günümüzde en yaygın kullanılan oracle veritabanı üzerinden çalışan uygulamaların performans ve güvenlik problemleri incelenerek tespit edilmiş, çözüm önerilerini sunan bir uygulama gerçekleştirilmiş ve alınan sonuçlar değerlendirilmiştir.

2 VERİTABANI HAKKINDA GENEL BİLGİLER

2.1 Veritabanı

Veritabanı düzenli bilgiler topluluğudur. Kelimenin anlamı bilgisayar ortamında saklanan düzenli verilerle sınırlı olmamakla birlikte, daha çok bu anlamda kullanılmaktadır. Bilgisayar terminolojisinde, sistematik erişim imkânı olan, yönetilebilir, güncellenebilir, taşınabilir, birbirleri arasında tanımlı ilişkiler bulunabilen bilgiler kümesidir. Bir başka tanımı da, bir bilgisayarda sistematik şekilde saklanmış, programlarca işlenebilecek veri yığıdır.

Veritabanı süreçleri incelendiğinde en önemli basamaklardan birinin indeksleme ya da diğer bir deyişle dizinleme olduğu görülür. Dizin her türlü yayın ve/veya kayıt üzerindeki kavramların, konuların, kelimelerin yerini gösteren sistematik bir rehberdir. Bir dizin, konuların nerede bulunabileceklerini göstererek bunlara kolay ulaşımı amaçlayıp, genellikle alfabetik olmak üzere mantıksal bir düzen içerisinde sıralanan konu girişlerini içerir.

2.2 Veritabanı Türleri

Veritabanı türleri, verileri yapılandırma, saklama ve bu verilere erişim şekillerindeki farklılıklarla göre ayrılırlar. Yapısal olarak üç ana tür veritabanından söz edilebilir. [2]

1. Hiyerarşik veri tabanları
2. İlişkisel veri tabanları

3. Nesneye yönelik veri tabanları

2.2.1 Hiyerarşik Veritabanı

Hiyerarşik veritabanlarında farklı veri tipleri önceden tanımlanan ağaç yapısında saklanır. “Kök” olarak bir kayıt ve bu köke bağlı “dal” kayıtlar bu tip veritabanının yapısını oluşturur. Bu türde en çok kullanılan yazılım, IBM tarafından çıkarılan IMS’dir. [3] Çoğunlukla büyük bilgisayarlarda (mainframe) kullanılır. Verilerin bir ağaç şeklinde düzenlendiği ve kayıtların hiyerarşik bağımlı olduğu bu veritabanlarına duyulan ilgi günden güne azalmaktadır. Bu veritabanlarının terk edilmesinin sebeplerinden biri olarak da üzerinde çalıştığı büyük bilgisayar (mainframe) yapısının son derece hantal ve sorunlu olması, bu sistemlerde uygulama geliştirme ve tasarımının yeni sistemlere son derece güç olmasından kaynaklanmaktadır. [4]

2.2.2 İlişkisel Veritabanı

İlişkisel veritabanı mantığı ilk olarak 1969 yılında E.F. Codd tarafında formüle edilmiş ve Haziran 1970’te ilişkisel veritabanı modeli teorisini sunmuş. Böylece ticari ilişkisel veritabanları olan IBM DB2, Oracle Database, Microsoft SQL Server, PostgreSQL, MySQL, diğer birçok veritabanlarının doğmuştur. [3]

Aralarında ilişki kurulan ve birden fazla tablodan oluşan veritabanlarına İlişkisel Veritabanı (Relational Database) denir. Günümüzde sıklıkla kullanılan ilişkisel veritabanlarında veriler birden fazla tabloda bulunur ve önceden belirlenmiş ortak alanlar aracılığı ile birbirleriyle ilişkilendirilirler. [4] Bir biriyle ilişkili verilerin tablolar halinde tutulması, veriye erişim ve tasarımın kolay olmasıyla beraber hem PC de hem de

büyük bilgisayarlarda (mainframe) çalışması nedeniyle günümüzde en çok yaygın kullanılan veritabanıdır.

2.2.3 Nesnel Veritabanı

Veritabanı yapılarında en önemli gelişme 1990'lı yıllarda ortaya çıkar, nesneye yönelik veritabanı (Object Oriented Database) olmuştur. Burada nesne olgusu temel alınmaktadır. Kayıtlar nesnelere, kayıt grupları sınıflara karşılık gelmekte ve tablonun kendisi ise bir nesne tipi olarak kabul edilmektedir. Bilgisayar Destekli Tasarım (CAD) ve diğer mühendislik uygulamalarında karşılaşılan veritabanlarının işlenmesinde ilişkisel veritabanları yetersiz kalmakta ve yerine nesneye yönelik veritabanları kullanılmaktadır. AutoCad, CadCam gibi programlar bunlara örnek olarak verilebilir. [4]

2.3 Ticari Veritabanı Yazılımları

2.3.1 Microsoft Access

Microsoft Access, Microsoft firması tarafından Microsoft Windows işletim sistemi tabanında çalışmak üzere yazılan ve dağıtımı yapılan veritabanı programıdır. Veritabanı programları arasında Access çok sonradan girmiş olmasına rağmen bu alanda önemli ölçüde başarı sağlayarak en çok kullanılan veritabanı programlama dillerinden biri haline gelmiştir. Bunda, Access'in veritabanına getirdiği farklı boyutun etkisi büyüktür. Access'de veritabanında bulunan nesnelere birçoğu sihirbazlar yardımıyla kolayca hazırlanabilir. Çoğu zaman hiç tasarım ortamına girmeden, sadece sihirbazlar kullanılarak veritabanı dosyaları hazırlanabilir. Access'in iki yüzü vardır. Bunlardan

birinde hiç program kodu kullanmadan veritabanı hazırlamak mümkündür. Bu ortamda hiç programcılık bilgisi olmayan biri programcılığı öğrenmesine gerek kalmadan kendi veritabanı dosyalarını hazırlayabilir. Programcılık bilgisi olanlar Access'de hazırladıkları veritabanlarına kod yazarak daha ileri düzey işlemleri yapabilir. Microsoft Access tek bir veri tablosunda 2 GB a kadar veri depolayabilir ve aynı anda 255 bağlantıya izin verebilirsiniz. [5], [6]

Ticari veritabanları arasından diğerlerine göre daha ucuz olan Access küçük ölçekli uygulamalardaki gereksinimlerinizi karşılayabilir. Microsoft işletim sistemi tabanlı çalışan Access veri miktarı ve aynı anda yapılan işlem sayısı az olan projelerden kullanılır.

2.3.2 MySQL

MySQL, altı milyondan fazla sistemde yüklü bulunan çoklu iş parçacıklı (multi-threaded), çok kullanıcı (multi-user), hızlı ve sağlam (robust) bir veritabanı yönetim sistemidir. UNIX, OS/2 ve Windows platformları için ücretsiz dağıtılmakla birlikte ticari lisans kullanmak isteyenler için de ücretli bir lisans seçeneği de mevcuttur. Linux altında daha hızlı bir performans sergilemektedir. Kaynak kodu açık olan MySQL'in pek çok platform için çalıştırılabilir ikilik kod halindeki indirilebilir sürümleri de mevcuttur. Ayrıca ODBC sürücülere de bulunduğu için birçok geliştirme platformunda rahatlıkla kullanılabilir. 1994 MySQL'in geliştirilmesine başlandı, 2008 Ocak: Sun Microsystems, Bir Milyar Dolar'a MySQL'i satın aldı. 2009 Nisan: Oracle, Sun Microsystems'i satın aldı. Oracle, MySQL'i geliştirmeye devam edeceğini duyurdu. [7], [8] Ücretsiz olması ve kurulumu kolay olması nedeniyle küçük ölçekli işletmeler tarafından tercih ediliyor.

2.3.3 IBM DB2

IBM DB2, IBM tarafından geliştirilmiş ilişkisel veritabanı yönetim sistemidir. Unix (AIX Unix) başta olmak üzere Linux, IBM I (AS/400), Z/OS ve Windows sunucularında çalışır. IBM, DB2'nin yanı sıra bir başka veritabanı yönetim sistemi (RDBMS) olan Informix'i de 2001 yılında satın aldı ve günümüzde nesne-ilişkisel modelin geliştiricisi olarak kabul ediliyor. DB2 uzun bir geçmişe sahiptir. Oracle'ın ticari veritabanı ürünü daha önce üretilmesine rağmen IBM DB2 SQL kullanan ilk veritabanı yazılımı olarak kabul edilir. DB2 veritabanı yönetim sistemi IBM tarafından 1983'te yayınlandı. İlerleyen yıllarda IBM DB2 ürününü geliştirerek platform desteğini ve programlama dili desteğini arttırdı. Günümüzde DB2 PDA'lar da dahil olmak üzere 19 farklı platformda 15'ten fazla programlama dilini desteklemektedir. 2009'da DB2 motoru MySQL içerisinde kullanılabilir hale geldi. [9], [10], [11]

2.3.4 Microsoft SQL Server

Versiyon 7.0'dan önce "kod tabanı" Sybase SQL Server tarafından Microsoft'a satıldı ve bu Microsoft'un kurumsal seviyede veritabanı pazarına girişi oldu. Sybase SQL Server 3.0 ile esasen aynı olan ilk sürüm, SQL Server 1.0'ı yaratmak ve pazarlamak adına Microsoft, Sybase ve Ashton-Tate ile takım oluşturdu ve ilk sürüm 1989 da 16 bit olarak çıkartıldı. 1992'de Microsoft SQL Server 4.2 sevk edildi. Daha sonra Windows NT 3.1 ile aynı zamanda Microsoft SQL Server 4.21 piyasaya sunuldu. Microsoft SQL Server 6.0 Windows NT için dizayn edilmiş ilk sürüm olmasıyla birlikte Sybase'den talimat alınmaksızın piyasaya çıkartıldı. SQL Server 7.0, miras Sybase kodu ile yazılan bir "rewrite" sürümü oldu ve yerine SQL Server 2000 çıkartıldı. SQL Server 2000, IA-64 mimarisinden farklı olarak yazılan ilk sürüm oldu. [8] İyi bir performansa sahip olan MS SQL Server'in en büyük dezavantajı, sadece Windows üzerinde çalışabilmesidir.

Kullanım kolaylığı, güvenilirliği ve işlem gücüyle dikkat çekmektedir. Maliyeti diğer veritabanlarına göre yüksektir.

2.3.5 Oracle

Oracle'ın bu günlere gelmesinde emeği olan gizli kahraman IBM'dir. IBM nasıl Bill Gates'e altın tepside DOS'u teslim ettiyse, benzer bir stratejik hatayı "ilişkisel veritabanı yönetim sistemleri" konusunda da yapmıştır. İlişkisel veritabanlarının temeli, bir IBM çalışanı olan Edgar F. Codd'un 1970'lerde yazdığı 'A Relational Model of Data for Large Shared Data Banks' makalesine dayanır. IBM o zamanlar pazara IMS/DB ürünü ile hakim olduğundan bu yeni fikirleri benimsemekte direnmiştir. IBM, Codd'un fikirlerini göz ardı ederken, Larry Ellison (Oracle'ın kurucusu ve CEO'su), Codd'un makalesini okur ve ilişkisel veritabanı sistemlerindeki büyük potansiyeli fark eder. 1977 yılında Larry Ellison, iki arkadaşıyla beraber (Bob Miner ve Ed Oates) Software Development Laboratories firmasını kurar ve RDBMS üzerine çalışmaya başlarlar. 1978 yılında resmi olmayan, pazarlanmayan PDP-11 platformda ve RSX işletim sisteminde 128 KB bellek ile çalışan assembly programlama diliyle Oracle Versiyon 1 (Sürüm) yazılır. 1979 de PDP-11 assembly programlama dili ile yazılan ve VAX/VMS platformlarında uyumlu olarak çalışan SQL destekli ilk ticari RDBMS Oracle sürüm 2 piyasaya sürülür ve firma ismini Relational Software Inc. (RSI) olarak değiştirir. 1982 de RSI' nin ismi Oracle veritabanını yansıtan Oracle System adını alır. 1983 de Oracle sürüm 3 piyasaya sürülür. Bu sürüm Assembly ve C programlama diliyle yazıldığı için büyük bilgisayarlarda (mainframes), küçük bilgisayarlarda (minicomputers), PCs ve C ile derlenen herhangi bir donanımda çalışır. Böylece Oracle veritabanı piyasada güçlenerek yalınlaşmaya başlar. [12], [13], [14], [15]

Günümüzde, oracle veritabanı çok büyük hacimli verileri depolayan, veriye erişimde ki hızının yüksek olması, çok yönlü işlevsel ve güvenilir olması nedeniyle dünyanın en

güçlü veritabanı olarak gösterilmektedir. Ancak lisans ve iş gücü maliyetinin yüksek olması nedeniyle sadece büyük kurumlar tercih etmektedir.

3 ORACLE VERİTABANI

Oracle8i veritabanının mimari yapısı şekil 3.1’de gözüktüğü gibi birden fazla ana bileşenlerden oluşmaktadır. [16] Bu ana bileşenler [17] şunlardır:

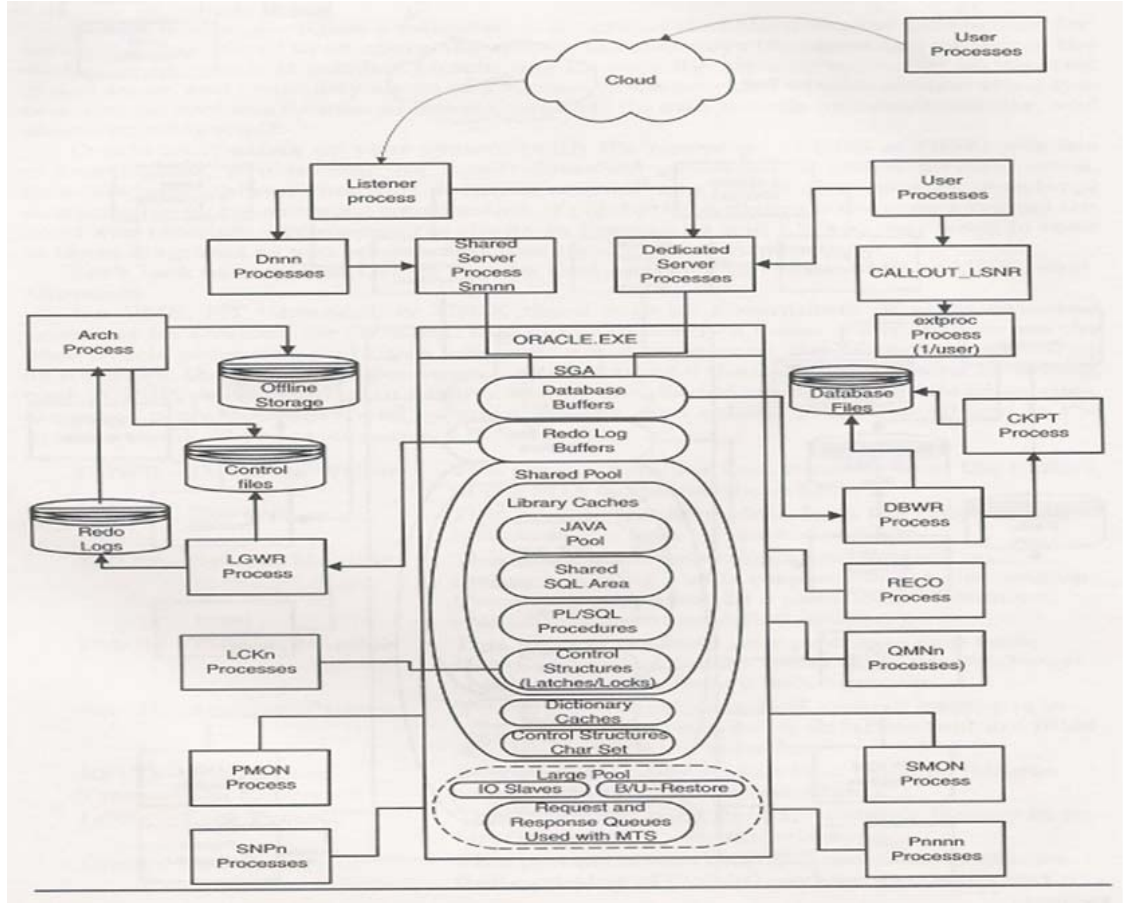
Oracle Sunucusu: Birkaç farklı dosya, işlemler (processes) ve bellek yapısı oracle sunucusunu (oracle server) oluşturur. Oracle sunucusu bir oracle anının (oracle instance) ve bir oracle veritabanı içermektedir. [16], [17]

Oracle Anı: Veritabanının bellek yapısı ve arka plan işlemleri (background processes) bir oracle anının (oracle instance) oluşmasını sağlar. Veritabanındaki verilere erişilmesi için oracle anının her zaman açık olması gerekmektedir. [16], [17]

Oracle Veritabanı: Bir oracle veritabanı işletim sistemi dosyalarını içermektedir, aynı zamanda bu dosyalar veritabanı dosyaları olarak da bilinir ve veritabanının fiziksel yapısını oluştururlar. [16], [17] Disk üzerindeki toplam veriler oracle veritabanı oluşturur. Oracle10g sürümünden ASM ya da RAW bölmeleri (partition) kullanıldığında veritabanı dosyalar OS’den gözükmeyebilirler fakat mantıksal olarak aynıdır. [18]

Diğer Anahtar Dosyalar: Bunlar veritabanı dosyalar olmayıp oracle anının konfigüre edilmesini, veritabanına uzaktan sistem kullanıcısı ile bağlanmasını ve herhangi bir felaket anında donanımdan kaynaklanan problemler nedeniyle bozulan dosyaları yeniden kurtarılması sağlar. [16], [17]

Kullanıcı ve Sunucu işlemleri: Veritabanından PL/SQL paketleri ve SQL komutların çalışması bu işlemler (server and user processes) aracılığıyla yapılır. [16], [17]

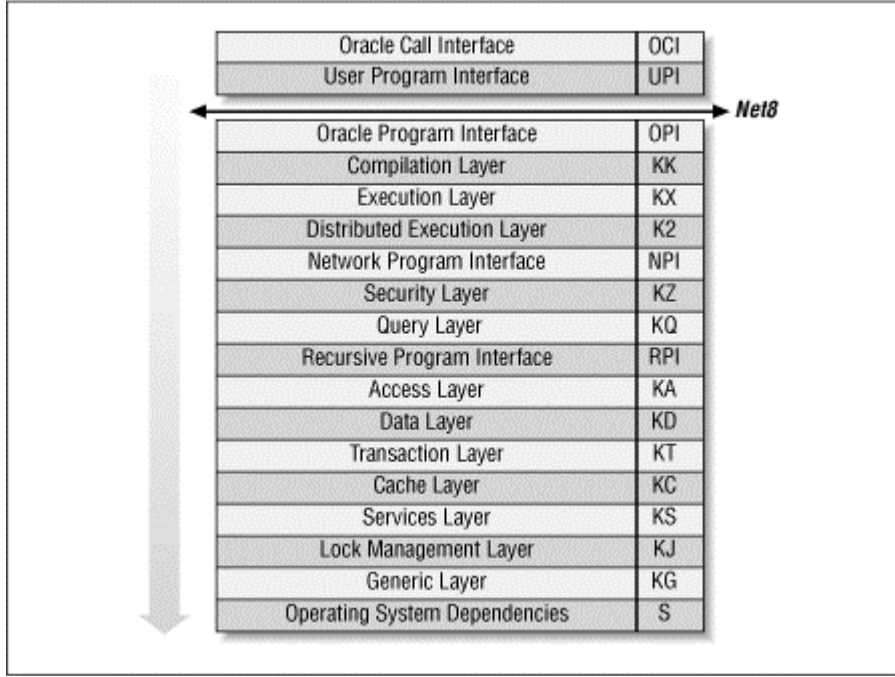


Şekil 3.1. Oracle8i Mimari Yapısı [16]

3.1 Oracle Çekirdek Katmanı

Oracle çekirdeği katmanlardan (Oracle kernel layers) oluşur ve her bir katman bir üstündeki servislerle ve altındaki katmanlarla ilişkilidir. Oracle'ın ana katmanları aşağıdaki şekilde gösterilmiştir. [19]

Her bir katman kısaltılmış ön eklerle adlandırılır. Örneğin; çekirdek önbellek (kernel cache) KC olarak adlandırılır. Şekil 3.2'de oracle'ın çekirdek katmanları gösterilmektedir.



Şekil 3.2 Oracle Çekirdek Katmanı [19]

The oracle call interface (OCI): Oracle'ın en küçük katmanı olup oracle ile işlemci ya da diğer programları (client programs) ile oracle arasında ilişki kurulan katmandır. Bu katman C ve C++ programlama diliyle yazılmış olup oracle veritabanı yönetim sisteminin birçok fonksiyonlarını yerine getirmektedir. Örneğin; oracle veritabanına erişim, oturum kontrolü (session control), karmaşık işlemleri ve nesnelerin akışını (object navigation) sağlamakta. [19]

The user program interface (UPI): Bazı UPI arabirimin olanaklarının OCI ile yapılması yetersiz olması nedeniyle oracle uygulamaları bu arabirimi direkt olarak ilişki kurarlar. Uygulamaların ön derlemeleri dolaylı olarak SQLLIB Kütüphanesi (SQLLIB library) kullanarak yapılır. [19]

Oracle program katmanı (The oracle program interface (OPI)): Kullanıcı program arabirimi (user program interface) işlemci tarafındaki (client-side call stack) adlandırılan en küçük katman ve oracle programlama katmanının ana sunucu (server-side call stack) katmanının en büyüğüdür. Net8, UPI ile OPI arasında ki boşluk köprü görevini yapar. Ancak, tek işlemlerin (single-task) çalışmasında bu boşluk yoktur. Kullanıcı program ara yüzü (UPI) doğrudan Oracle Program ara yüzü (OPI) ile haberleşir. [19]

Net8 Oracle'ın bir ağ (network) ürünüdür. Farklı bilgisayarların veritabanına bağlantı kurması ve istemci-sunucu arasında veri alışverişinin sağlanabilmesi Net8'in ana görevidir. Net8 protokol bağımsız olup geniş kapsamlı platformları destekler. Bütün veri dönüşümleri net8'i kullanarak yapılır. [20]

The compilation layer (KK): Oracle çekirdeğinin en üst katmanı olan KK, SQL kodlarının yazım kontrolü (parse) ve iyileştirme (optimize) ederek PL/SQL programlarını derlemekten sorumludur. Aynı zamanda veri sözlüğü (Data Dictionary) deki tablo ve indeks istatistiklerine göre SQL ve PL/SQL sorgularının çalışma planını (execution plan) oluşturur. [19]

The execution layer (KX): Bu katman PL/SQL kodlarını parçalayarak SQL cümlelerini çalıştırır ve programın bağlayıcı değerlerini (binding variable) tutar. Aynı zamanda tekrar tetikleyici (tirgger) çalışmasını yerine getirir. [19]

The distributed execution layer (K2): Parçalanmış işlemler (distributed transactions) çalışması ve çift yönlü geçerli kılma (two-phase commit) protokolünün yönetimin bu katmanda yapılır. [19]

The network program interface (NPI): Kullanıcıların ana sunucuya göndermiş olduğu SQL cümle parçalarının işlenmesi ve verileri tekrar kullanıcıya gönderme işlemi yapılır. [19]

The security layer (KZ): Objelerin derlenmesi, çalıştırılması, rol ve sistem yetkilerinin doğrulanması bu katmanda yapılır. [19]

The query layer (KQ): Veri sözlüğü (data dictionary) deki kayıtların saklanır, derleme (KK) ve güvenlik (KZ) katmanları sorguların derlenmesi aşamasında verileri bu katmanda alırlar. [19]

The recursive program interface (RPI): Veri sözlüğündeki (data dictionary) objeleri ya da temel veri (meta data) bilgilerini yeniden düzenler. [19]

The access layer (KA): Veri parçalarına (data segment) erişimde sorumlu katman olup oracle çekirdeğinin en orta katmanıdır. [19]

The data layer (KD): Verilerin yönetimi ve kontrolünde sorumlu olan katmandır. Örneğin tablo, indeks, cluster (küme) içerisindeki verilerin saklanması ve parça blokların kontrolü yapılır. [19]

The transaction layer (KT): İşlemlerin kontrolü, bölümlerdeki boş alanlar (segment free list) yönetimi ve tablo uzaylarının (tablespace) büyümesini kontrol eder. [19]

The cache layer (KC): Veri havuzu (Database Buffer Cach) yönetiminde sorumlu olan katmandır. Bu katmanda işletim sistemi ile birlikte bellek alanı (buffe cach) ve paylaşılmış bellek alanı (shared memory) yönetimini sağlar. Ayrıca verilerin redo log dosyalarının yazılması, redo log dosyalarının üretiminde ve kontrol dosyalarındaki bilgilerinin saklanmasından sorumludur. [19]

The services layer (KS): Üst düzeydeki kullanılan bütün servislerin en alt servis katmanı olup bu servislerde oluşan yanlışların işlenmesi, hataların ayıklanması ve izlenmesi işlemleri yapılır. Ayrıca bu katman veritabanında oluşan kilitler (lock, latch), beklemler (wait events) istatistiklerin yönetiminde, oracle anı (instance) ve kullanıcı oturumuna (session) ait veritabanı parametrelerini başlatılması ile sorumludur. [19]

The lock management layer (KJ): Bu katman veritabanı kilitlerinin yönetimi ve kontrolünde sorumludur. [19]

The generic layer (KG): Bu katman üst düzeylerdeki katmanlarda kullanılan veri yapılarının yönetiminde sorumludur. Örneğin bağlantı listesi (linked lists) gibi. [19]

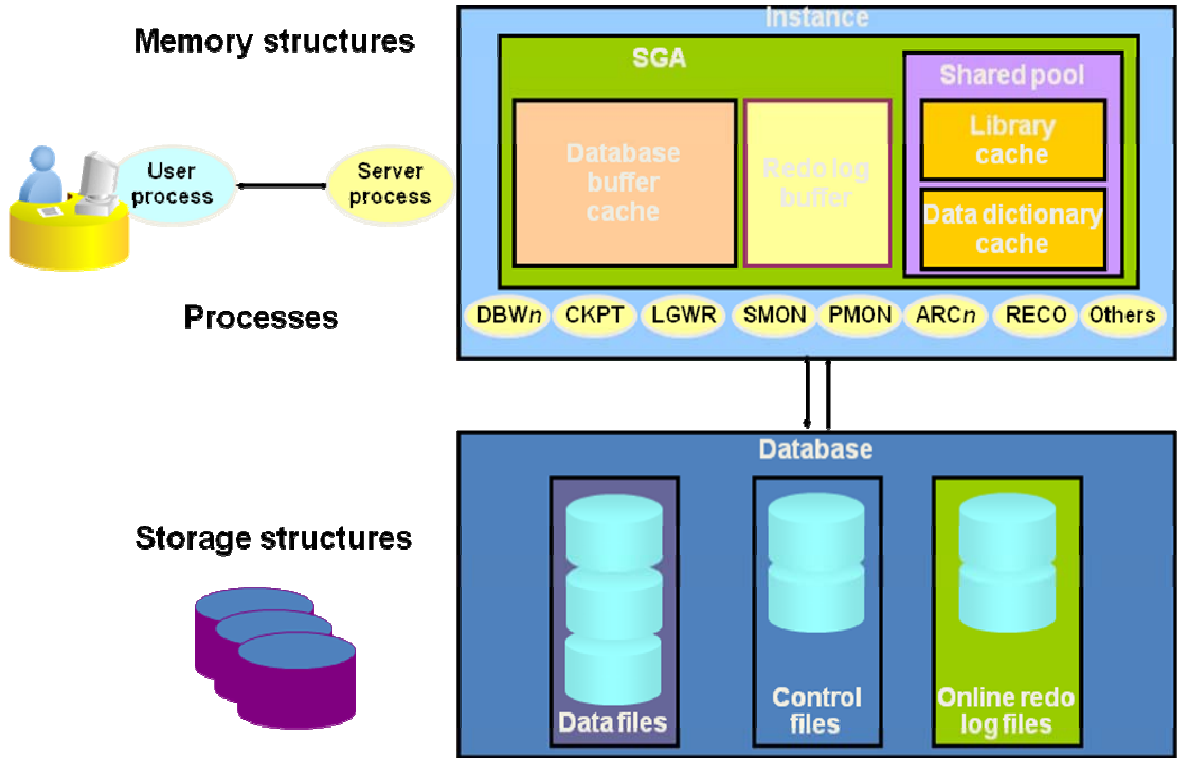
The operating system dependencies (S): Oracle çekirdeğinin en küçük katmanı olup işletim sistemi ile oracle arasındaki soket haberleşmesi, bellek yönetimi, kilitleme ve I/O işlemlerini programlar. [19]

3.2 Oracle Veritabanı Mimari Yapısı

Oracle veritabanı çok sayıda kullanıcıyı aynı anda destekleyen karmaşık denilebilecek bir yapıya sahiptir. Bu yapıyı fiziksel olarak üçe ayırabiliriz. [21]

- Bellek Alanları
- İşletim Sistemi Süreçleri
- Veritabanı Kütükleri (dosyaları)

Şekil 3.3'de oracle veritabanının sunucu yapısı gösterilmektedir.



Şekil 3.3. Oracle Veritabanı Sunucu Yapısı [21]

3.3 Oracle Veritabanı Bellek Yapısı

Oracle çalışırken birçok iş için bellek kullanır. Örneğin, veritabanında çalışan program kodlarının ya da verinin ortak bir alanda kullanıcılara sunulması gibi.

Bu işlemler için iki temel bellek alanı kullanılır [22]:

- SGA (System Global Area)
- PGA (Program Global Area)

3.3.1 Sistem Genel Alanı (SGA)

Sistem Genel Alanı (SGA) bir oracle anının (instance) veri ve kontrol bilgisi içeren ortak kullanım alanıdır. Bir oracle anı bir SGA ve ilgili arka plan proseslerden (background proseses) oluşur. Veritabanı her açıldığında SGA oluşturulur, kapatıldığında ise silinir. Her anın SGA'sı ayrıdır. Sistemin en iyi şekilde çalışabilmesi için gerektiği kadar büyük olmalıdır ki disk I/O minimuma insin. [23]

SGA üç ana kısımdan oluşur: [23]

- Veritabanı Tampon Belleği (Database Buffer Cache)
- “Redo Log” Tampon Belleği (Redo Log Buffer)
- Paylaşım Havuzu (Shared Pool)

Bu alanların büyüklükleri “init.ora” dosyası parametreleri ile belirlenebilir ve bu değerler performansa direk etki eder. Sistemdeki kullanıcı sayısı, veri büyüklüğü, sunucu kapasitesi gibi özellikler göze alınarak başlangıç değerler belirlenir.

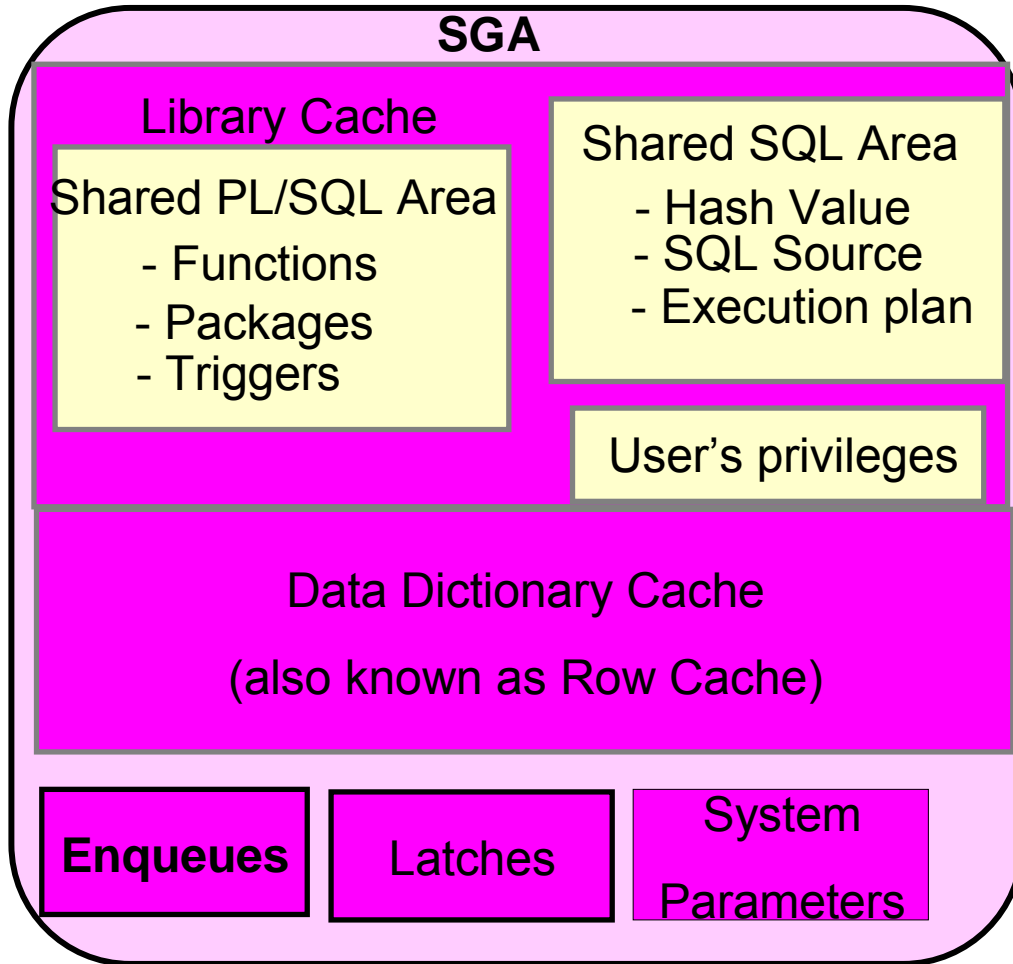
Database buffer cache: Bu alanda en son kullanılan veri blokları tutulur. Bu sayede çok kullanılan verinin hep bellekte tutularak disk I/O azaltılır ve performans kazanılır. Doğal olarak veri dosyalarındaki bilgi büyüklüğü kadar bellek olamayacağından en az kullanılan bloklar bellekten atılır. Bu algoritma LRU (Last Recentl Used) olarak adlandırılır. Bütün veriler mutlaka bellekte işlenir. Her veri bloğu kullanımına göre “LRU List” olarak adlandırılan bir sıralama listesine sahip tutulur. “Database Buffer Cache” de olmayan bir veri üzerinde işlem yapılmak istendiğinde eğer “Database Buffer Cache” de yer yoksa bu listenin en sonlarındaki bloklardan ihtiyaç olduğu kadarı bu alandan çıkartılır ve liste düzeltilir. Bu işlem tüm sistemin yaşamı boyunca sürekli tekrar eder. Bu alanın büyüklüğü “db_block_buffers init.ora” parametresi ile belirlenir.[17], [23], [24], [25], [26]

Redo log buffer: SGA'nın "redo log buffer" bileşeni üzerinde redo kayıtları (veritabanı üzerinde yapılan bütün değişiklikler) kaydı saklanır. Bu alan üzerindeki kayıtlar sistemde tanımlı olaylar dahilinde "on-line redo log" dosyasına yazılır. Bilindiği gibi "redo log" dosyaları veritabanı ile ilgili kurtarıma (recovery) ihtiyaç duyulduğu zamanlar için kullanılır. [17], [23], [24],[25], [26]

Shared pool: SGA'nın "Shared Pool" bileşeni üzerinde, ortak SQL alanları gibi veritabanı üzerinde ortak kullanılan bellek yapıları tutulur. Ortak bir SQL alanı her bir SQL cümlesinin veritabanı üzerinde işlenmesi için gerekmektedir. Bir SQL cümlesi ile ilgili "parse tree" , çalışma planı (execution plan) gibi bilgiler aynı cümleyi çalıştıracak bütün kullanıcılara ortak olması için o cümle ile ilgili özel bir alanda tutulur. Böylelikle birbirinin aynı olan SQL cümleleri için veri bloklarında olduğu gibi bellekten yer kazanılmış olur.[17], [23], [24], [25], [26]

"Library Cache" ve "Dictionary Cache" bu alanı oluşturan en önemli iki bileşendir. "Dictionary Cache" oracle'ın kendi iç nesnelere bulunurken "Library Cache" üzerinde tıpkı "Data Buffer Cache" de olduğu gibi PL/SQL paketleri (package/procedure) ve ortak SQL kullanım alanları ile ilgili bilgiler bulunur. [23] Şekil 3.4 'de Ortak havuz (Shared pool) ve bütün bileşenleri görülmektedir.

Shared pool



Şekil 3.4. Ortak Havuz (Shared Pool) ve Bileşenleri

3.3.2 Program Genel Alanı (PGA)

Program Genel Alanı (PGA) bir sunucu işleminin (server process) kullandığı veri ve kontrol bilgisinin tutulduğu alandır. PGA, bir sunucu işlemi başlatıldığında yaratılmaktadır ve o işleme ait olmaktadır. “Dedicated Server” mimarisinde her oracle sunucu işlemine ait bir PGA mevcuttur. PGA büyüklüğünü etkileyen en önemli iki

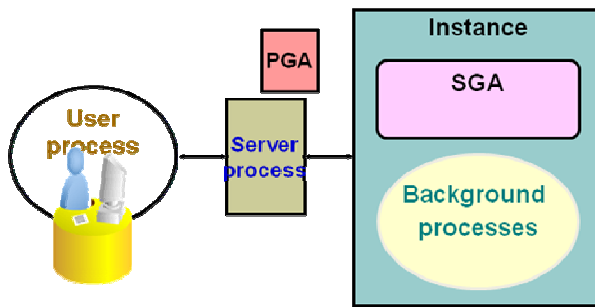
faktör sıralama için ayrılan alanlar ve SQL cümleleri için açılan imleçlerin (cursor) büyüklükleridir. PGA içinde değişken, izin ve diğer nesnelerin bulunduğu bir alan (stack space) ile birlikte kullanıcının oturumu ile ilgili bilgiler (özel SQL alanları) bulundurulur. Bu alanın kullanımı da SGA kullanımı kadar kritik ve o derecede önem verilmesi gereken bir konu olmasa da performansa benzer derecede etki eder. Özellikle sıralama işlemlerinde ve büyük verilerin istemcilere gönderilmesi gerektiği durumlarda, diğer bir deyişle uygulama tarafından açılan SQL imleçlerin büyük olduğu durumlarda önemlidir. [17], [23], [24], [25]

3.4 Görev Yapıları

Oracle bütün işlemlerini doğal olarak prosesler (görevler) yardımı ile yapmaktadır. Bu görevleri de bellek konusunda olduğu gibi fonksiyonel olarak uç gruba ayırmak mümkündür: [24]

- Oracle sunucu görevleri (Oracle Server Processes)
- Oracle kullanıcı görevleri (Oracle User Processes)
- Arka plan görevleri (Background Processes)

Şekil 4.5’de oracle işlem görev (Oracle Process) yapısı gösterilmektedir.



Şekil 3.5 Oracle İşlem Görevleri [25]

Oracle sunucu görevleri: Oracle sunucu görevleri (oracle server processes), veritabanına bağlanıp çalışan uygulamanın ihtiyaçlarına cevap vermek için sunucu üzerinde yaratılan oracle işlemidir. Oracle kullanıcısı bir oturum (session) açtığı anda çalışmaya başlar. Amaca özel sunucu (dedicated server) denilen böyle bir yapıda ilgili uygulama istemci üzerinde çalışabileceği gibi sunucu üzerinde de çalışabilir. İlgili uygulama fonksiyonel olarak kullanıcı işlem görevi (client process) olarak da adlandırılabilir. Bir oracle sunucu işlemi, ilgili işlem görevi (client process) ile sürekli konuşmak, istekleri oracle veritabanına ulaştırmak ve sonucunu iletmekle yükümlüdür. Örneğin; kullanıcı bir sorgu yaptığı zaman ilgili veri blokları 'Data Buffer'da yok ise bu verilerin veri dosyalarından okunma işlemini yaparlar. [17], [23], [24], [25], [26]
Oracle, sunucu işlem görevlerinin durumuna göre iki şekilde çalışabilir: [17]

Dedicated server: Her kullanıcı işlem görevi için bir sunucu işlem görevi yaratılır.

Multi-threaded server: Kullanıcı sayısının çok olduğu sistemlerde sunucu kaynakları daha etkin kullanılabilir. Birden fazla istemci görevi için bir sunucu işlem görevinin kullanıldığı sunucudur.

Oracle kullanıcı görevleri: Oracle kullanıcı işlem görevleri (oracle user processes), kullanıcı veritabanına bağlantı kurmak istediği anda yaratılır ve sürekli sunucu işlem görevi (process) ile ilişki içerisinde. [17]

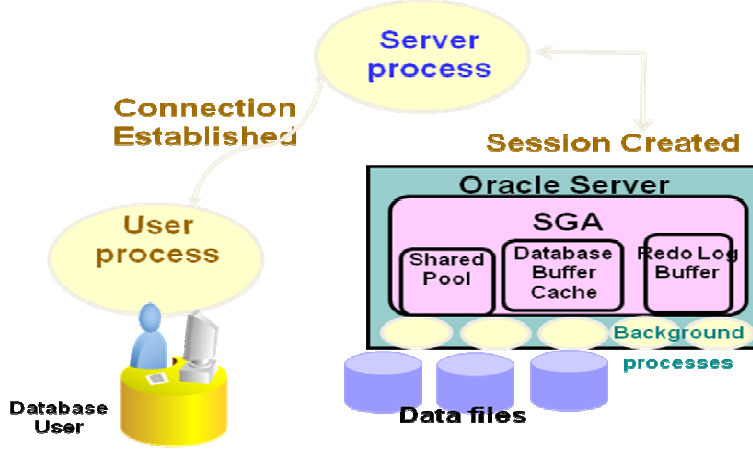
3.4.1 Bir Kullanıcının Veritabanına Bağlanması

Bir kullanıcı Oracle Server'a üç değişik şekilde bağlanabilir. [17], [23], [26]

- Sunucu üstünden doğrudan
- İki katman kullanılarak (Two-tiered (Client-Server))

- Üç katman kullanılarak (Three-tiered (Internet Computing))

Şekil 1.6’da herhangi bir oracle kullanıcısının veritabanı bağlantısı gösterilmiştir.



Şekil 3.6. Oracle Kullanıcısının Veritabanı Bağlanması [17]

Sunucu üstünden: Önce oracle’ın çalıştığı sunucuya doğrudan bağlanılır (login) olunur. (Örneğin bir unix sunucuya oracle kullanıcısı ile girip “server manager” çalıştırmak.) [17]

İki katman kullanılarak: LAN veya WAN üzerindeki bir bilgisayar veya sunucudan aynı ağ üzerindeki bir oracle sunucuya yapılan bağlantı. Client-Server mimarisi. [17]

Üç katman kullanılarak: Internet veya LAN, WAN üzerinde istemci ve sunucu arasında bir katmanda uygulama sunucu aracılığı ile yapılan bağlantıdır. Aslında İki katmanlı bağlantıdan bir farkı yoktur. Haberleşme, uygulama sunucusu aracılığı ile kullanıcı bilgisayarı ve sunucu arasında ya da kullanıcı bilgisayarı ile veritabanı sunucusu arasında olmaktadır. [17]

Uygulama başladığında net8 istemci ayağa kalkarak ayrı giriş/çıkış (port) dan (genelde 1521) bağlantı isteğini sunucu üstündeki dinleyiciye (listener) iletir. Dinleyici ilgili oracle sunucu işlemini başlatır ve istemci görevi ile sunucu görevi arasındaki

haberleşmenin kullanılacağı giriş/çıkış istemciye iletir. İstemci giriş/çıkış alır ve her iki süreç (proses) bu giriş/çıkış üzerinden haberleşmeye başlar. Dinleyici aradan çekilir. [23], [24], [25], [26]

3.4.2 Arka Plan Görevler

Bir oracle anı içinde birçok kullanıcıya hizmet edecek şekilde çalışan genel görev işlemleridir. Oracle sunucu birçok arka plan işlem görevi (background processes) kullanır. Bazıları oracle veritabanının çalışması için zorunluken bazıları ihtiyaç anında opsiyonel olarak kullanılır. Hazırda zorunlu olarak beş çeşit sunucu görevleri kullanılır. Aşağıda zorunlu ve opsiyonel olan işlem görevi kısaca tanımlanmaktadır. [16], [17], [18], [23], [25], [26]

Zorunlu işlem görevleri:

- DBWR (Database Writer)
- LGWR (Log Writer)
- SMON (System Monitor)
- PMON (Process Monitor)
- CKPT (Check Point)

Opsiyonel işlem görevleri:

- ARC (Archiver)
- RECO (Recoverer)
- Dxxx (Dispatcher)
- LCK (Lock)
- SNP (Job Que)
- QMN (Queue Monitor)

Bu işlem görevleri kısa açıklaması:

DBWR: “Buffer Manager” olarak görülebilir. “Data Buffer Cache” deki deęiřtirilmiř (dirty) oracle veri bloklarını veri dosyalarına yazar.

LGWR: “Redo Log Buffer” daki bilgileri “on-line redo log” dosyasına yazar.

SMON: Sistemin genel durumunu gözlemleyici iřlem görevidir. Veritabanı anı içinde sistemin akıřı ile ilgili iřleri yapar. Bu iřlerden en önemlileri; ‘Instance Recovery’, ‘temporary tablespace’ düzenlenmesi, boř ‘extent’lerin düzenlenmesidir.

PMON: Temel görevi herhangi bir sebeple yarım kalan iřlerle ilgili iřlemlerin yapılmasıdır (Process Recovery). Bu iřler yarım kalan iřlerin (transaction) geri alınması, sistem kaynaklarının temizlenmesi gibi iřlerdir.

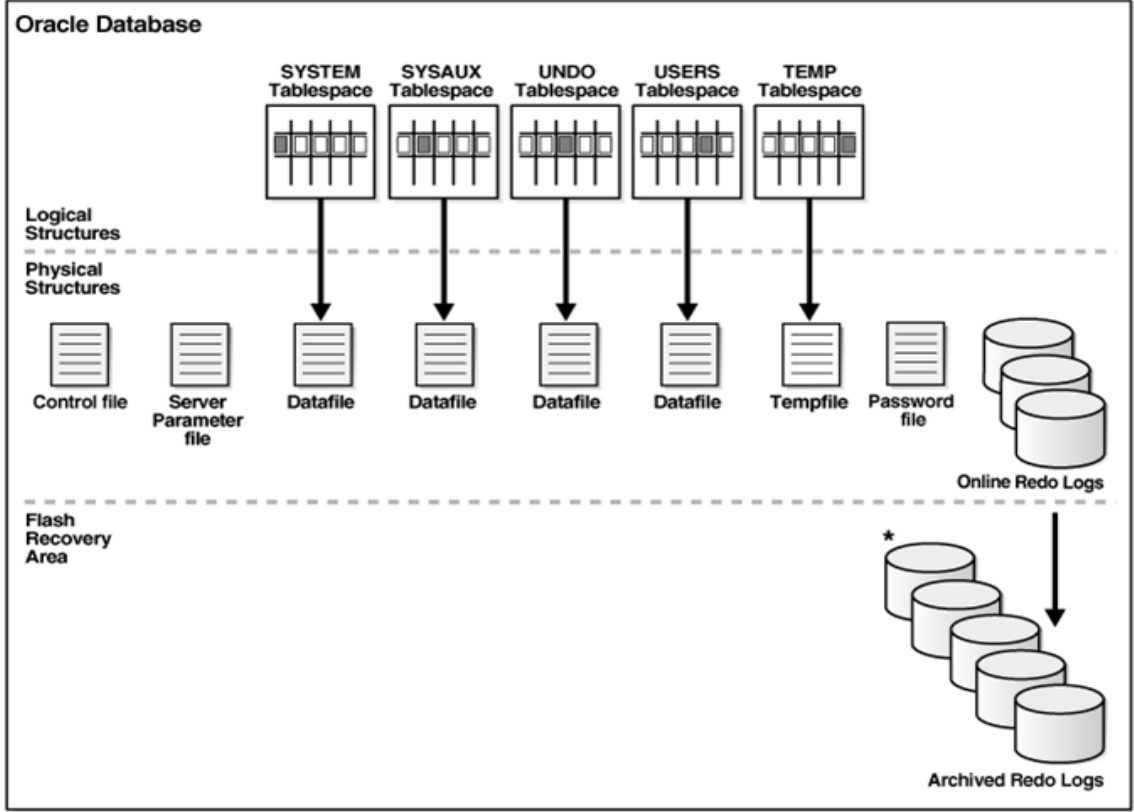
CKPT: Buffer Cache’deki bilgiler DBWR tarafından kalıcı olarak veritabanına yazıldıęında veritabanı durum bilgisini günceller.

Dięer iřlem görevleri bir anın alıřması için zorunlu deęildir fakat her birinin görevleri vardır. Örneęin ARC iřlem görevi arřiv modadaki bir oracle anının dolan redo-log dosyalarını arřivlemek için ilgili arřiv dizinine sırayla kopyalar.

3.5 Oracle Veritabanı Kütük Yapısı

Oracle VT kütük yapısı (storage structures) fiziksel ve mantıksal olmak üzere iki katmandan oluşur. Fiziksel ve mantıksal katman birbirinden ayrı olduęu için verinin fiziksel olarak saklanma řekli mantıksal yapıya eriřimi etkilemez. [22]

řekil 3.7’de oracle veritabanının fiziksel ve mantıksal veri kütüklerinin yapısı gösterilmektedir.



Şekil 3.7 Oracle VT Kütüklerinin Fiziksel ve Mantıksal Yapısı [22]

3.6 Oracle Veritabanı Fiziksel Yapısı

Oracle VT dosyalarının fiziksel yapısı (Physical Structure), işletim sisteminden görünen kısımdır. Bunlar veri kütükleri, kontrol kütükleri, parametre, şifre ve log kütüklerinden oluşmaktadır. [22]

3.6.1 Kontrol Kütükleri

Her oracle veritabanını bir kontrol kütüğü (control file) içerir. Kontrol kütükleri VT fiziksel yapısı ve statüsü ile ilgili kayıtlar vardır.

Bir kontrol dosyası aşağıda listelenen çeşitte bilgiler içerir: [23], [24], [25], [26]

- Veritabanı ismi (Database Name)
- Veri ve “redo-log” dosyaları isimleri ve lokasyonları ve sıra numaraları
- Veritabanı yaratılma zaman bilgisi
- Veritabanı üzerinde yapılan değişikliklerin veri dosyalarına kaydedildiği son “Checkpoint” anı bilgisi
- Eğer yedekleme için RMAN kullanılıyorsa yedekler ile ilgili bilgiler

Oracle anının her başlangıcında ilgili kontrol dosyası kullanılarak açılması gereken veri ve “redo-log” dosyaları tespit edilir ve senkronize olup olmadıkları kontrol edilir. Eğer dosya ekleme gibi veritabanı fiziksel yapısını değiştiren işlemler yapılırsa bu işlemler oracle tarafından kontrol dosyasına yansıtılır.

3.6.2 Veri Kütükleri

Bir veri dosyası (data file) yaratıldığında ilgili disk alanı ayrılır ve oracle’a göre formatlanır ve herhangi bir kullanıcı verisi içermez. İlgili nesnelere oluşturulduklarında bu veri dosyaları üzerinde boş alanlar kullanılarak oluşturulurlar. Nesne yapıları tamamen mantıksaldır ve fiziksel olarak veri dosyaları üzerinde dururlar. [23]

3.6.3 Online Redo Log Kütükleri

“Redo-Log” dosyalarında veritabanında yapılan bütün değişiklikler tutulur. En az iki log grup olmak zorundadır. Bu gruplara sırayla ve dönerek bütün değişiklikler yazılır. “Redo-Log” dosyalarındaki bilgiler sistem ya da medya çökmesi olduğunda veritabanını kurtarmak için kullanılır. Bu dosyalar sayesinde hiç bir şekilde bilgi kaybedilmez. [26]

3.6.4 Parametre Kütüğü

Bir oracle anının (instance) hangi parametrelerle konfigüre edilip çalışacağını parametre dosyasındaki (parameter file) bilgilere göre belirlenir ve veritabanı açılma aşamasında kullanılır. [21]

3.6.5 Yedek Kütükler

Veritabanı orijinal dosyaların bozulması ya da herhangi bir felaket anında medya bozulması durumlarında yedek dosyaları (online ya da offline backup files) kullanılarak sistemin kurtarılması sağlanır.[26]

3.6.6 Arşiv Kütükleri

Oracle veritabanı “archivelog” ya da “noarchivelog” durumunda çalışır. Hazır durum “noarchivelog” dir. Veritabanı “archivelog” durumunda çalıştırılarak, medya bozulması durumlarında komple veya geçmişteki bir zamana dönülecek şekilde (on-line ya da

offline yedeklerle) sistemin kurtarılması sağlanır. “Redo-log” dosyalarında veritabanında yapılan bütün değişiklikler tutulmaktadır. Veritabanı, “archivelog” durumunda iken yazılmakta olan “redo-log” dosyası dolduğunda bizim belirleyebileceğimiz bir dizine veya yedek ünitesine “ARC process” tarafından kopyalanır. Her “log switch” olduğunda “on-line redo log” dosyası arşivlenir. Bu olay veritabanı yaşadıkça sürekli devam eder. Böylece tam yedek (full-backup) alındıktan sonra sadece arşiv dosyalarının kopyaları alınarak veri tabanı komple yedeklenmiş olur. [22], [23]

3.6.7 Şifre Kütüğü

Şifre dosyası binary bir dosyadır ve “sysdba” ve “sysoper” hakları verilmiş kullanıcıları tutmak içindir. Bu bağlamda DBA veritabanı işlemlerini “sysdba” ve “sysoper” olarak uzaktan bağlanıp yapmak için şifre dosyası (password file) kullanılır. [27]

3.6.8 Hata ve Uyarı İzleme Kütüğü

Oracle veritabanında ki hatalar “alert” ve “trace” dosyalarında izlenir. Bu dosyalar aynı zamanda veritabanı ve uygulamalardaki performans problemleri hakkında bilgi de içerir. [27]

3.7 Oracle Veritabanı Mantıksal Yapısı

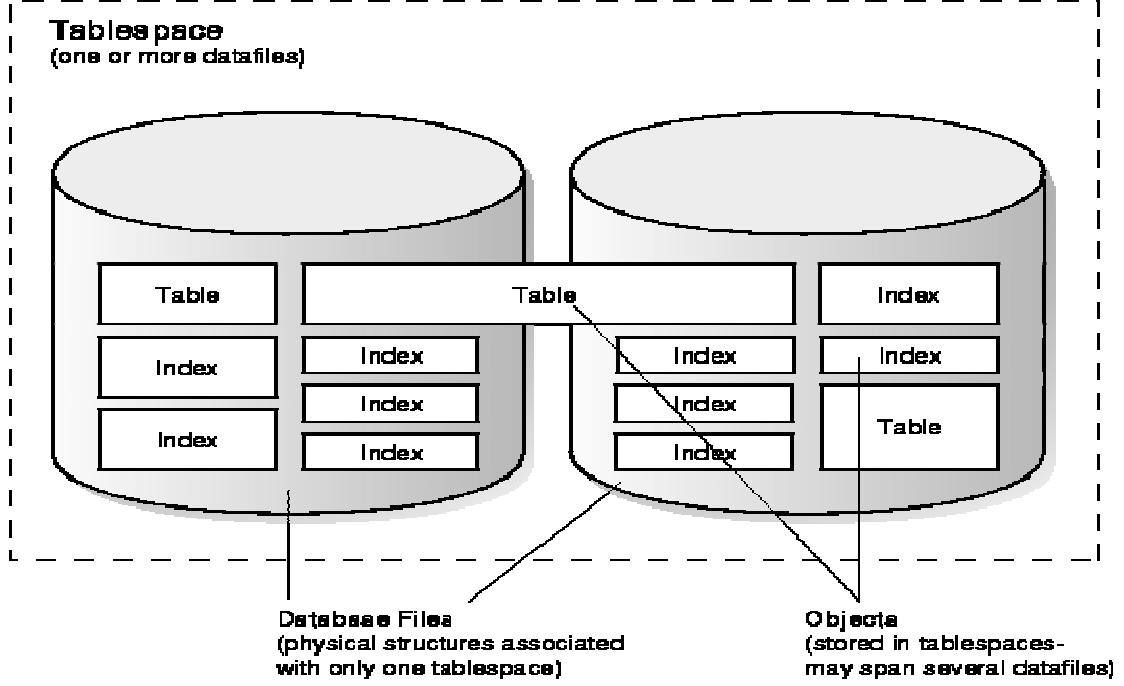
Oracle veritabanı dosyalarının (kütüklerin) mantıksal yapısı (Logical Structure) bir ya da daha fazla tablo uzayı (tablespace), tablolar, görüntüler, sıralar, eşanlımlar, indisler, kümeler, VT bağlantıları, prosedürler (izlek), fonksiyonlar ve paketlerden oluşan şema nesnelere oluşmaktadır. Fiziksel yapıya işletim sistemi tarafından erişebilmesine rağmen, mantıksal yapıya ancak oracle'a bağlanıp, SQL komutları çalıştırılarak erişilebilir. Oracle veritabanı dosyalarının fiziksel ve mantıksal yapısı arasındaki ilişki Şekil 3.7'de gösterilmektedir. [22], [23], [27]

3.7.1 Tablo Uzayı

Bir veritabanı bir veya daha fazla tablo uzayından (tablespace) oluşur, bu tablo alanlarının ise uzantılar (extent) ve bölmelerden (segment) oluşmaktadır. Oracle veritabanı içindeki mantıksal yapı mantıksal nesnelere birbirlerini kapsayacak şekilde kurulması gerçeğiyle oluşmaktadır. Bu yapıya veri saklayan bileşenler ve bu bileşenlerin birbirlerini oluşturma sırası şöyledir. [23], [27]

Database-Tablespace-Segment-Extent-Block-Row.

Bütün bileşenler mantıksaldır ve kendi başlarına oracle sistemi içinde ifade edilirler. Fakat veritabanı ve tablo uzayı (tablespace) yaratıldıktan sonra veritabanı fiziksel yapısı hazır demektir. Bundan sonra bu yapı üzerinde tablo, indeks gibi mantıksal bileşenleri bu fiziksel yapı üzerinde oluşturma aşaması başlar. Tablo uzayı içindeki tüm mantıksal yapıları fiziksel olarak tutması için bir veya birden fazla veri kütükleri yaratılır. Veri kütüklerin toplam büyüklüğü, tablo uzayının toplam depolama kapasitesini verir. Tablo uzaylarının toplam depolama kapasitesi, veritabanının toplam büyüklüğünü verir. [17], [23] Şekil 3.8'de Tablo uzayı, veri kütükleri ve VT nesnelere arasındaki ilişki belirtilmektedir.



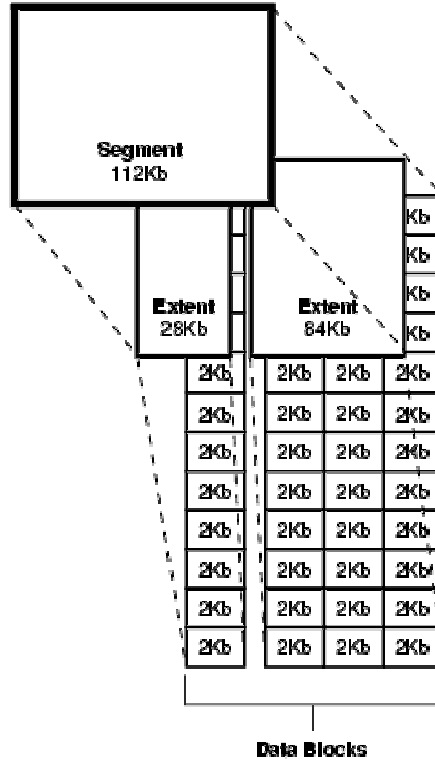
Şekil 3.8. Tablo Uzayı, VT Kütük ve Nesne Yapısı [23]

3.7.2 Veritabanı Sistem Tablo Uzayları (System ve Sysaux Tablespace)

Veritabanının yaratılmasıyla birlikte bir tane “SYSTEM” ve “SYSAUX” tablo uzayları zorunlu olarak yaratılır ve veritabanı yaşadığı sürece aktif olması gerekir. Veritabanının ana fonksiyonlarını yerine getiren veriler “SYSTEM” tablo uzayında depolanır. Örneğin, veri sözlüğü tabloları (data dictionary tables). Veritabanına ait birçok bileşenler ise “SYSAUX” tablo uzayında depolanmaktadır. Örneğin; tablo ve indeks istatistiksel verileri bu tablo uzayında tutulur. [25], [26]

3.7.3 Veri Saklama Yapıları

Veri blokları, uzantılar (extent) ve bölmelerden (segment) oluşan mantıksal saklama yapıları sayesinde oracle'ın disk alanı üzerinde veritabanındaki bütün veriler için mantıksal veritabanı alanı ayırır. Veri blokları, uzantılar ve bölmeler (segment) veritabanı alanı ayırma birimleridir. Şekil 3.9'da segmentler, uzantılar ve veri blokları arasındaki ilişki gösterilmektedir. [23]



Şekil 3.9. Segment-Boyut-Blok ilişkisi [23]

Segment Çeşitleri: Bir segment ilgili tablo uzayı (tablepace) içinde belirli bir mantıksal yapı ile ilgili bütün verinin tutulduğu boyut (extent) topluluğudur. Bir segment'i

oluşturan boyutlar değişik dosyalar üzerinde dağıtık şekilde oluşabilir. Örneğin oracle her bir tabloyu oluşturmak için ilgili tablonun veri segment'ini oluşturmak için bir veya daha fazla boyut (extent) yaratıp kullanabilir. [23]

Bu bağlamda oracle içinde kullanılan segment çeşitleri ve tanımları aşağıdaki gibidir: [23], [24] [25], [27]

Table: Verilerin en genel biçimde tutulduğu segment.

Table Partition: Table segment'i ile aynıdır. Bir tablo çeşitli tablo uzaylarına bölünmek istendiğinde kullanılır.

Cluster: Bir veya birden fazla tabloyu bir anahtar değere göre bir yapıda tutmak için kullanılır.

Index: indeks'lerin tutulduğu yapıdır.

Index-organized table: Tablodaki bilgi indeks yapısına göre bu yapıda tutulur. Bu yapıdaki tablolar için ilgili indeks ihtiyaç yoktur.

Index partition: İndeks segment ile aynıdır. Bir indeks'i çeşitli tablo uzaylarına bölmek için kullanılır.

Undo segment: SQL işlemleri (transaction) tarafından kullanılır. Veri değiştirilmeden önce ilgili bloğun orjinal halini tutmak için kullanılır.

Temporary segment: Sıralama işlemleri için kullanılır.

Lob segment: Bir tablonun bir veya daha fazla alanında LOB tutulmak istenebilir. Bu LOB'lar alışılmışın dışında bilgi oldukları için başka yapılarda tutmak gerekir.

Kullanıldığı zaman ilgili tablo LOB verinin yerine sadece LOB ile ilgili yer işareti (pointer) tutar.

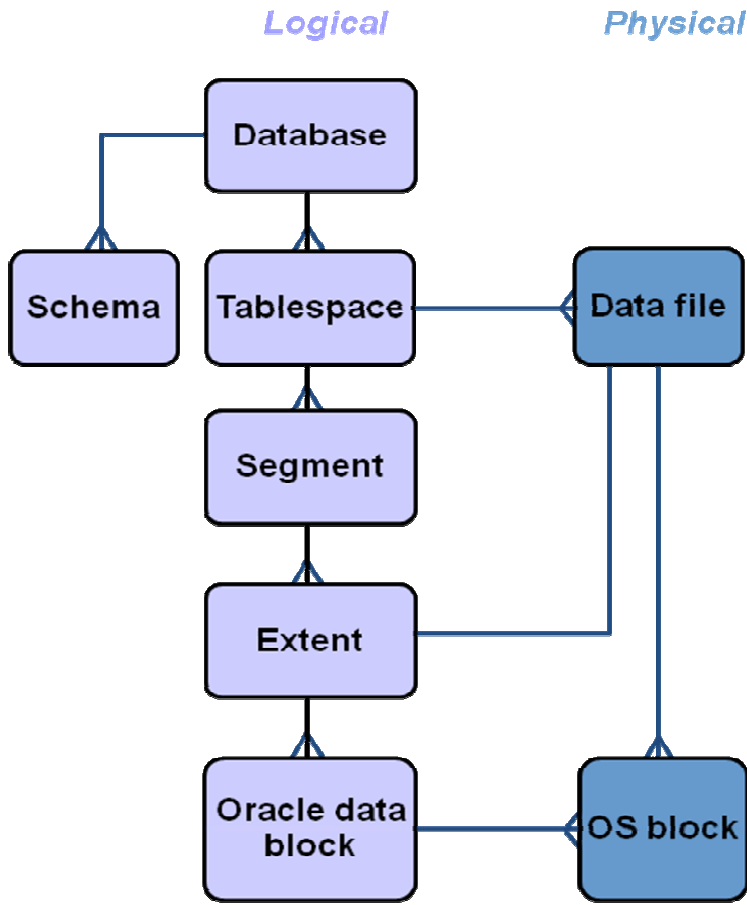
Lob index: LOB segment yaratıldıktan sonra dolaylı olarak yaratılır. LOB alanları sorgulamak için kullanılır.

Nested table: Bir tablonun bir alanı kullanıcı tanımlı bir tablo olarak tutulmak isteniyor olabilir. Bu durumda tıpkı LOB'larda olduğu gibi bu tabloların da tutulduğu yer ayrı olabilir.

Bootstrap: Saklama (cache) segment olarak bilinir. Veritabanı yaratılırken oluşturulur. Veritabanı açılırken "Data Dictionary Cache" başlatılmak için kullanılır.

Extent: Bir boyut (extent) ilgili segment dahilinde verinin tutulduğu ardışık bloklar topluluğudur. Bir veya daha fazla boyut bir segment'i oluşturur. Boyut'lar içindeki tüm alan dolduktan sonra oracle yeni bir boyut yaratarak ilgili segment'in büyümesini sağlar.

Storage cümlecığı: Segment'ler yaratılırken bazı özellikleri (initial extent, next extent vs.) "storage" cümlecığı ile verilebilir. Verilmez ise ilgili değerler segment'in yaratıldığı tablo uzayın hazır değerlerinden alınır. Tablo uzayların hazır değerleri ise yaratılırken verilmez ise sistem hazır değerleri geçerli olur. Depo (storage) cümlesi özellikle verisi büyük olan sistemlerde mutlaka kullanılmalıdır. Yer yönetiminin performansa etki eden en önemli etkenlerden biridir. Şekil 3.10'da oracle veritabanının mantıksal ve fiziksel yapısı gösterilmektedir.



Şekil 3.10. Oracle Veritabanı Fiziksel ve Mantıksal Yapısı [24]

Oracle blokları: Oracle veri dosyalarındaki yeri veri bloğu (data block) olarak adlandırılan üniteler ile yönetir. Bir veri bloğu veritabanı tarafından yapılan en küçük I/O ünitesidir. İşletim sisteminde her şey fiziksel olarak “byte” larda tutulur. Her işletim sistemi kullanılan dosya sistemine göre değişik büyüklüklerde kendi bloklarına sahiptir. Oracle içinde veri mantıksal olarak işletim blokları sayesinde değil oracle blokları sayesinde işlenir. Bir oracle bloğunun doğal olarak bir formatı vardır. İlgili kayıtlar bu format dâhilinde bloklar üzerinde tutulmaktadır. [23]

Şekil 3.11’de bir oracle bloğunun yapısı, Şekil 3.12’de ise blok içindeki kayıt yapısı gösterilmektedir.

Bu bağlamda bir oracle bloğunu oluşturan bileşenler ve açıklamaları şöyledir: [23], [24], [25], [27]

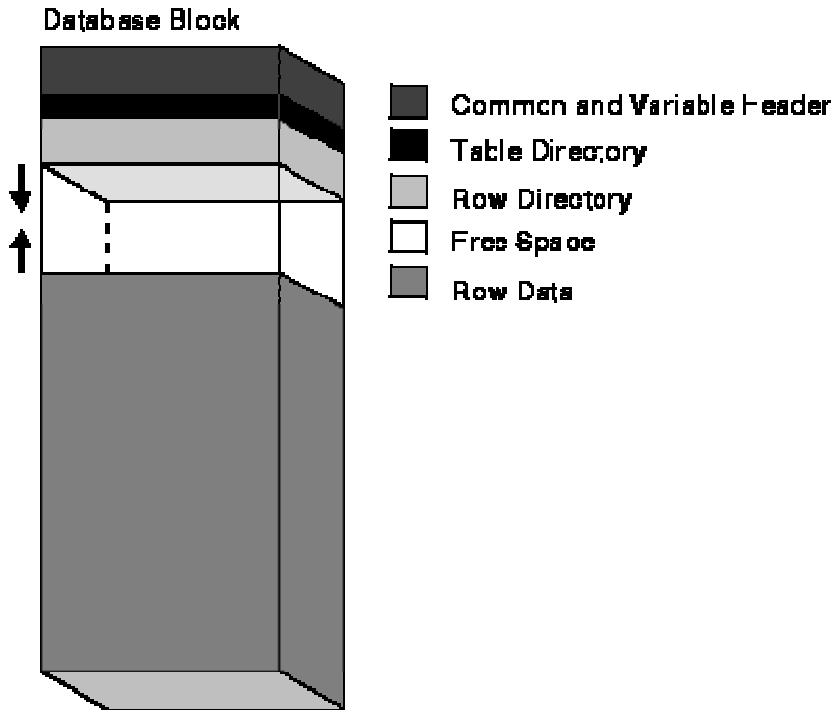
Header: Genel blok bilgisini içermektedir. Bu bilgi içinde bloğun adresi ve segment çeşidi bulunmaktadır.

Table directory: Bu bloğun içinde tutulan tablolar ile ilgili bilgi tutulur.

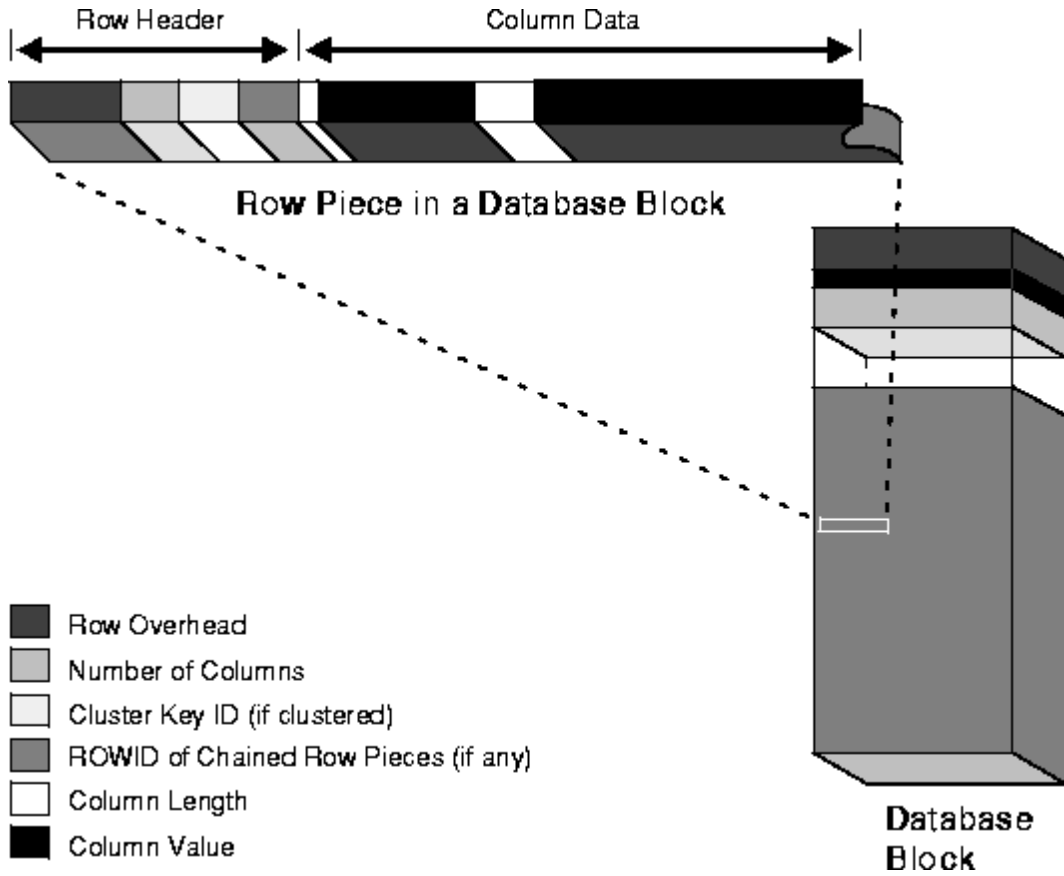
Row directory: Bloğun içinde gerçekten duran kayıtlarla ilgili adres bilgiler tutulur. Kayıt silinse bile yer silinmez başka kayıtlar için ayrılır.

Row data: Kayıtların kendisi tutulur.

Free space: Boş alandır sonradan gelecek kayıtlar içindir



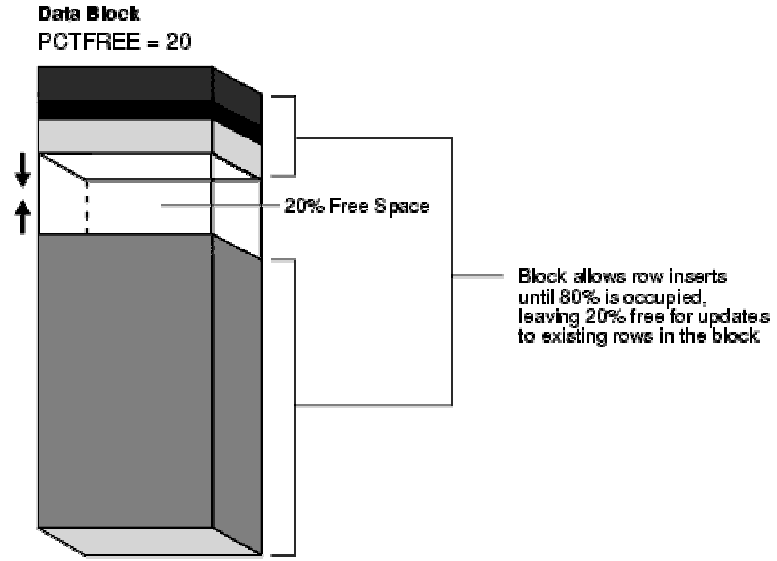
Şekil 3.11. Blok Yapısı [23]



Şekil 3.12 Kayıt (Row) Yapısı [23]

“PCTFREE” ve “PCTUSED” Yöntemleri: İki yer yönetimi parametresi “PCTFREE” ve “PCTUSED, insert, update ve delete” işlemlerinde bloklardaki boş alanın kullanımı ile ilgilidir. Bu parametreler tablo ve indekslerin yaratılma anında verilir. [23]

“PCTFREE” Parametresi: Blok dolduktan sonra gelecekteki Update’ler için önceden ayrılan alan yüzdesidir. Örneğin bir tablonun yaratılma anında bu değer 20 olarak verilirse bu tablo ile ilgili bütün bloklardan %20’lik bir alan *sadece update* işlemleri için boş bırakılacak demektir. Bu da örneğin kabaca 100Mb. Veri içerecek bir tablo segment’inin veritabanında 120Mb. yer kaplaması demektir. [18], [23], [26]



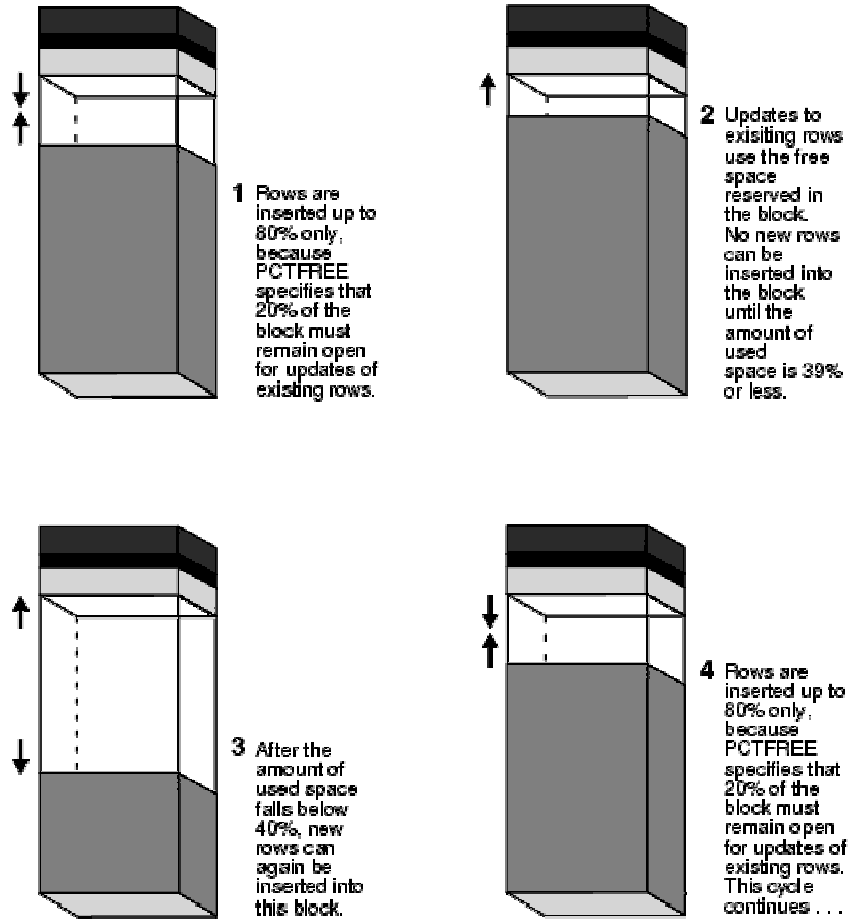
Şekil 3.14. “PCTFREE” Parametresi [23]

“PCTUSED” Parametresi: İlgili bloğun free-list’ten çıkması için gereken doluluk oranıdır. Free-list ise insert yapılabilecek blokların listesidir ve her tablo ve *indeks segment*’i için ayrı ayrı tutulur. Eğer bir bloğun doluluk oranı bu parametre ile verilen değerden daha düşük ise bu bloğa kayıtlar girebilir demektir. [18], [23]

Örneğin bir “CREATE TABLE” cümlesinde “PCTUSED” 40 şeklinde verildiğini düşünülün. Bu durumda bu tablonun bir bloğu içindeki kullanım (doluluk) yüzdesi %39 ve daha altına düşene kadar (tablonun daha önce tabii ki “PCTFREE” değerine ulaşması kabul ediliyor) bu blok için yeni *insert* kabul edilmez. [23]

“PCTFREE” ve “PCTUSED” parametreleri yerin daha etkili kullanılması için birbirleriyle ilişkili çalışmaktadır. Şekil 3.15’de bu çalışma anlayışını sırasıyla göstermektedir. [23] Bir “INSERT” cümlesi çalıştırıldığında, Oracle tablonun “free-list” ini kontrol ederek müsait olan ilk bloğu seçer ve mümkünse kullanır. Eğer bu blok içindeki boş alan “INSERT” cümlesini kaldırarak kadar fazla değilse bu blok en azından

“PCTUSED” değerindedir ve blok listeden çıkartılır. Bir segment başına birden fazla “free-list” aynı anda birden fazla “insert” olan sistemlerde bu tür beklemleri azaltır. Bir “DELETE” ya da “UPDATE” cümlesi çalıştırıldığında, oracle cümleyi işler ve blok üzerinde kullanılan alanın şimdi “PCTUSED” dan az olduğunu kontrol edip görmek ister. Eğer öyle ise blok free-list’in en başına gider ve bu transaction (işlem) için kullanıma hazır ilk blok olur. İşlem onaylandıktan (commit) sonra boş alan diğer transactionlar (işlemler) için kullanıma hazır olur. [23]



Şekil 3.15. “PCTFREE” ve “PCTUSED” arasındaki ilişki [23]

Bir veri bloğunda boş alan bulunması ve sıkıştırılması: İki çeşit cümle bir ya da daha fazla blok üzerinde yer açılmasına yol açar; “DELETE” cümleleri ve olan değerleri küçülten “UPDATE” cümleleri. Boşalan bu alan aşağıdaki koşullarda izleyen “INSERT” cümleleri için müsaittir: [23], [26]

- Eğer “INSERT” cümleciği aynı işlem içinde ise ve alan açan cümleden sonra geliyorsa açılan bu alan “INSERT” cümlesi için kullanılabilir.
- Eğer “INSERT” cümlesi yer boşaltan cümleden ayrı işlem (başka kullanıcıların çalıştırdığı işlem) içinde ise , “INSERT” cümlesi boş alanı ancak yer açan işlem onaylandıktan (commit) sonra kullanılabilir.

Bloktaki boş alanlar ardışık olabilir ya da olmayabilir. Oracle bir bloğun boş alanlarını aşağıdaki durumlarda birleştirir: [23], [26]

- Bir “INSERT” ya da “UPDATE” cümleciği yeni kayıt parçasını içerecek kadar yeterli boş alan bulduğunda
- Kayıt parçası sığacak kadar ardışık alan bulunmadığı ama boş yer yeterli olduğunda performans etkileneyeceği için başka hiç bir durumda otomatik olarak birleştirme yapılmamaktadır.

Veri bloklarının büyük veya küçük olmasının bazı avantajları ve dezavantajları vardır. Bu avantaj ve dezavantajlar şunlardır: [23], [26], [28]

Küçük blok avantajı

- Küçük bloklar, blok taşmasını azaltır çünkü her blokta daha az satır vardır.
- Küçük bloklar, küçük satırlar için daha iyidir.
- Küçük bloklar rastgele okumalar için iyidir. VT gizli yerinin boyutu sınırlandırılabileninden, küçük blok boyutu, tampon gizli yerinin daha çok etkin kullanımını sağlar. Bu durum bellek kaynakları sınırlandırıldığında önemlidir.

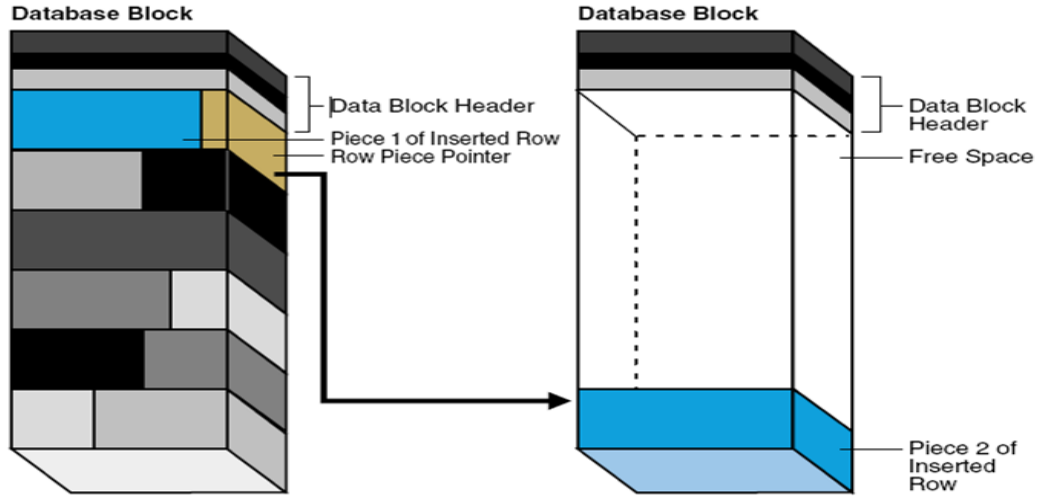
Küçük blok dezavantajı

- Küçük bloklar göreceli olarak tepe üst düzeyine sahip olurlar.
- Daha fazla blok okunmasına neden olurlar.

Büyük blok avantajları

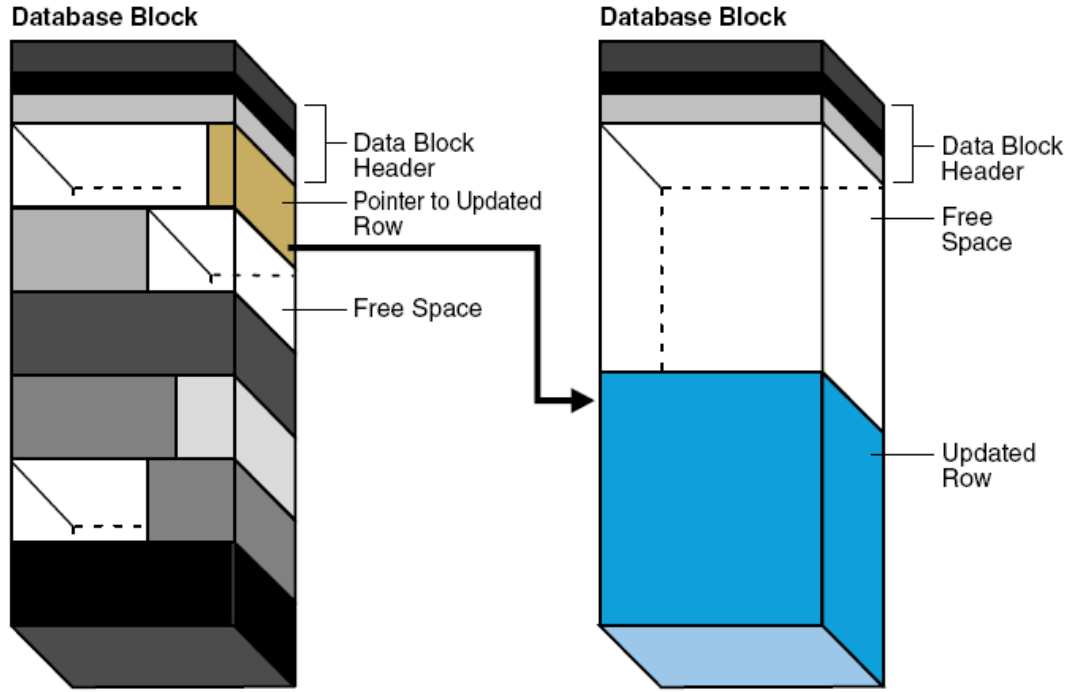
- Büyük bloklar büyük satırlar için daha iyidir.
- Büyük bloklar sıralı okumalar için daha iyidir.
- Büyük bloklar, indis okumalar için performans sağlar. Büyük bloklar her bir blok içerisinde daha çok dizin girişi tutabilirler.

Kayıt zincirleme ve taşınma (row chaining and migration) : İki durumda bir kaydın tek bir bloğa sığmaması söz konusudur. Birinci durum, kaydın ilk “insert” anında bir bloğa sığmadığı durumdur. Bu durumda oracle ilgili kaydı birbirlerine bağlı bir veya daha fazla blokta tutar. Kayıt zincirleme (Row Chaining) denilen bu durum daha çok büyük “LONG, LONG RAW ve VARCHAR2(4000)” gibi veri tipleri içeren kayıtlarda olur. Bu durumda zaten kaçınılmazdır. [26] Şekil 3.16’da kayıt zincirleme durumu gösterilmektedir.



Şekil 3.16. Kayıt Zincirlemek [26]

İkinci durum, başlangıçta bir bloğa sığan herhangi bir kaydın, blokta güncellenmek (update) için ayrılan alana sığmayacak şekilde “update” edilip büyütülmesi sonucu oluşur. Bu durumda oracle bütün kaydı yeni bir bloğa (sığacağını kabul ederek) taşır. Oracle, orijinal parçayı taşınan parçanın yeni bloğunun yerini gösterecek şekilde saklar. Taşınan kaydın sıra numarası (rowid) değişmez. Bu durum “Migration” olarak adlandırılır. [26] Kayıtlar zincirleme ve taşıma durumuna maruz kalırsa bu kayıtlar ile ilgili I/O performansı daha fazla bloğun taranması ihtiyacından dolayı azalacaktır. Her iki durumdan kaçınmak için önlemler alınmalıdır. Şekil 3.17’de kayıt taşıma durumu gösterilmektedir.



Şekil 3.17. Kayıt Taşımak [26]

3.8 Oracle iyileştirici

Bu bölümde oracle iyileştirici (optimizer) tarafından yapılan optimize etme işlemi ilgili konular kısaca tanıtılmaya çalışılmıştır. Amaç; beşinci bölümde tez kapsamında geliştirilen uygulamanın anlaşılmasına ve geliştirmesine katkı sağlamaktadır.

3.8.1 Optimize etme

Optimize etme (optimizasyon) işlemi bir SQL sorgusunun çalışması için en hızlı ve verimli yolun seçilme işlemidir. Bu işlem herhangi bir DML sorgusu (Select, Insert,

Update, Delete) ya da DDL (mevcut indeksi yeniden yapılandırma-rebuild index) çalışmasında önemli bir aşamadır. Bir SQL cümlesinin çalışması için birçok yol mevcuttur. Örneğin ilgili SQL cümledeki tablolara veya indekslere ulaşımdaki sıra verilebilir. Oracle'ın bir sorguyu çalıştırırken izlediği yöntem bu sorgunun çalışma hızını büyük ölçüde etkilemektedir. [29], [30]

DML ve Commit işlemlerinin çalışma mantığı : [23], [29]

DML işlemi: Oracle veritabanında DML cümlelerinin çalıştırılması üç aşamada sırayla gerçekleşir. Bu aşamalar aşağıdaki gibidir.

- Parse
- Execute
- Fetch

Parse: Kullanıcı işlem görevi (user process) oracle sunucu işlem görevine ilgili SQL cümlesini yazım kontrolü (parse) ya da derlemek (compile) için gönderir. Sunucu işlem görevi (server process) bu cümle üzerinde yetki, derleme (compilation) ve sözdizimi (syntax) kontrolü yapmak için de “library cache” i kullanır. Bu aşamanın sonunda sunucu işlem görevi kullanıcı işlem görevine bu işlemlerde bir bilgilendirme çıkarsa bunu gönderir.

Execute: Bu aşamada ilgili sunucu işlem görevi (server process) kayıtları seçer ve almak için hazırlık yapar.

Fetch: Sorgu tarafından seçilen kayıtlar sunucudan istemciye gönderilir. Gönderilen bilginini büyüklüğüne bağlı olarak bir veya daha fazla gönderim olabilir. “UPDATE” ve “DELETE” işlemlerinde doğal olarak “fetch” olmaz. Sadece sonuç gönderilir.

Güncelleme ve silme (Update and Delete) işlemleri: Sorgular çalıştırılırken geçilen aşamalara benzerdir. “Parse” ve “Execute” aşamaları yine vardır fakat fetch aşaması yoktur. [23]

Bu işlemlerin detayı aşağıdaki gibidir: [23], [29]

- Eğer ilgili bloklar “Data Buffer Cache” de yok ise veri dosyaları ve “rollback” segmentler taranarak ilgili bilgiler getirilir.
- Sunucu işlem görevi veri üzerine ilgili kilitleri koyar.
- Sunucu işlem görevi verinin değişiklik öncesi (before-image) ve değişiklik sonrası (after-image) üzerinde yapılacak değişiklikleri önce log kütüklerine (log buffer) kaydeder.
- Değişiklik öncesi bilgisi “rollback” bloğuna, değişiklik ise veri bloğuna “Buffer Cache” içinde uygulanır. Değiştirilen her iki blok “dirty buffer” olarak daha sonra “DBWR” tarafından veri dosyalarına yazılması için işaretlenir.

Onay (commit) işlemi: Bir commit (onay) yapıldığında aşağıdaki aşamalar yapılır : [23]

- Sunucu işlem görevi “Redo Log Buffer” da “System Change Number (SCN)” ile birlikte bir onay kaydı oluşturur.
- “LGWR Log Buffer “ daki bütün bilgileri onay (commit) kaydı ile beraber “online redo-log” dosyasına yazar. Bu sayede bu noktadan itibaren bu değişikliklerin kaybedilmeyeceği garanti edilmiş olur
- Kullanıcıya onaylamanın bittiğine dair bilgi gönderilir.
- Sunucu işlem, işlemin (transaction) bittiğine dair bilgiyi kaydeder ve ilgili kaynakları (kilit) serbest bırakır.

Oracle uygulama kodu iyileştirici (optimizer), “Cost-Based” ya da “Rule-Based” olarak çalışır. Ancak uygulama kodu iyileştiriciler (oracle optimizer) , oracle sürümleri

arasında aynı kararları vermeyip bu konuda deęişiklikler gösterebilir. Oracle9i ve sonraki sürümlerden “Rule-Based optimizer” kullanılmamaktadır.

“RBO (Rule Based Optimizer)” çalışma mantığı belli kurallar kümesine göre çalışır. Örneğin; bir SQL cümlesinin indeksin kullanılıp kullanılmayacağına veya birleştirme (join) işleminde hangi tablonun önce kullanılacağına hep kurallar dâhilinde karar verir.[23]

“CBO (Cost Based Optimizer)” ise veriye ayrı ulaşım yollarının maliyetlerini değerlendirir ve bunlardan en az maliyetli olanını seçer. Bu işlem için tablo ve indeksler üzerinden toplanan istatistikler kullanılır. [30]

Bir sorgunun (SQL cümleleri) veritabanında çalışmasının üç temel maliyeti vardır.[29], [30]

- I/O Maliyeti
- CPU Maliyeti
- Ağ bağlantı (Network) Maliyeti

I/O maliyeti ilgili verinin disklerden belleğe taşınma maliyetidir. CPU maliyeti bellekteki bilgilerin işlenmesi maliyetidir. Ağ bağlantı (network) maliyeti ise farklı makineler arasından sorgu çalıştırılıyorsa ilgili bilginin ağ üzerinden transferi maliyeti oluşturur. [29],[30]

4 VERİTABANI PERFORMANS VE GÜVENLİK ARAÇLARI

Piyasada en yaygın kullanılan VTYA araştırılmış ve belirlenen kıstaslara göre karşılaştırma yapılmış ve fayda/maliyet analizi açısında en uygun aracın seçimi sağlanmıştır. [48]

Buradaki veriler incelendiğinden tez kapsamında geliştirilen yazılımın bu yöntemin sağlıklı sonuç çıkarmayacağı kanısında olduğum için sayısal rakamlar yerine aradaki farklar belirtilmektedir.

4.1 Toad 9.6 (Quest Software)

Toad kullanılan diğer VT araçları arasında geçmişi daha çok olan bir araçtır. Geliştirilmesi kişisel programcılıkla 1995 yılında başlamıştır. 1998 yılında Quest yazılım ürün geliştirmeye devam etmiş ve ilk resmi sürümü Toad 5, Ekim 1998 yılında çıkmıştır. Yaklaşık olarak her yıl iki ürün çıkartılmıştır. 2007’de çıkan Toad 9.5 sürümü, yalnız DBA izleme (monitor) ve uygulama geliştirme ürünün değil aynı zamanda gelişmiş yönetim ve analiz özellikleri sağlanmıştır. Toad, birden fazla VT için geliştirici ve yönetici aracıdır aynı zamanda çeşitli VT teknolojilerini desteklemekte ve buna göre şekil almaktadır. Desteklenen veri tabanları: Oracle, SQL Server, MySQL, ve DB2’dir. [31] Genel özellikler şu şekilde özetlenebilir:

- Uygulama geliştirme aracı,
- VT şama ve farklı sürümler arasında nesne taşıma,
- Veritabanı yönetim aracı, DBA veritabanı ile ilgili işlemler hızlı ve basit olarak yapmaktadır.
- Veritabanı performans/durum izleme (database monitoring),
- Veritabanı sağlık kontrolü (database health check),

- Canlı sistemde şema (schema) ve konfigrasyon karşılaştırarak farkların çıkartılması,
- Canlı sistem ile test sistem karşılaştırarak farkların çıkartılması,
- SQL ve PL/SQL çalışma planının çıkartılması (execution plan),
- SQL ve PL/SQL kodlarının analizi ve iyileştirme özelliği,
- Excel ve “txt” dosyalarını VT tablolarına taşıma gibi özellikleri içermektedir.

4.2 KeepTool 7.2 (Tool for Oracle Database)

1996 Yılında Alman yazılım firması olan PSI’da 3 programcı, oracle’da SQL yazmak için delphi arabirimini kullanmaya karar verdi. 1997 Yılında KeepTool GbR, daha sonra 2000 yılında KeepTool Gmbh geliştirildi. 2010’da sürüm 9’u piyasaya çıkardılar. Ürünün oracle ile ilgili temel özellikleri gözden geçirerek her yıl yeni bir sürümü piyasaya çıkarmaktadırlar. [32] Genel özellikler şu şekilde özetlenebilir:

- Uygulama geliştirme aracı
- Veritabanı yönetim aracı
- Veritabanı performans/durum izleme (database monitoring)
- ER diyagram ve PL/SQL hata ayıklayıcı (ER Diagrammer and PL/SQL Debugger) özelliği vardır.

4.3 DBHawk (DataSparc, Inc.)

DataSparc firması tarafından Haziran 2002’de DBAConnect 1.0 sürümü çıkartıldı. İlk olarak “SQL Browser” adı ile SQL aracı olarak çıkarıldı, daha sonra bazı özellikler ve yenilikler katılarak adı DBHawk olarak değiştirildi. DBHawk, herhangi bir web gezgini

ile oracle VT'nın yönetimini sağlayan web tabanlı bir araçtır. [33] Genel özellikler şu şekilde özetlenebilir:

- Web tabanlı uygulama geliştirme aracı
- Web tabanlı veritabanı yönetim aracı
- Veritabanı performans/durum izleme (database monitoring)

4.4 DB PowerStudio for Oracle (Embarcadero)

DB PowerStudio for Oracle Embarcadero firması tarafından geliştirilen uygulama geliştirme, veritabanı yönetim ve performans iyileştirme aracıdır. [34] Genel özellikler şu şekilde özetlenebilir:

- SQL Optimize ve performans aracı
- Canlı sistemde şema (sema) ve yapılandırma karşılaştırarak farkların çıkartılması
- Canlı sistem ile test sistem karşılaştırarak farkların çıkartılması
- SQL ve PL/SQL çalışma planının çıkartılması (execution plan)
- Tek bir ara yüz ile Oracle'ın bütün sürümlerinin çalıştırılması
- SQL ve PL/SQL kodlarının analizi ve iyileştirme
- VT nesnelere için rapor üretmekte
- Oracle kapasite ve kaynak planlaması
- VT şema'lar (schema) arası nesne taşıma

4.5 DBTools 6.1.0 (SoftTree Technologies)

DBTools'un 6.1 sürümü Kasım 2009'da yayınlanmıştır. [35] Genel özellikler şu şekilde özetlenebilir:

- Veritabanı performans/durum izleme (database monitoring)

- Canlı sistemde şama (sema) ve konfigürasyon karşılaştırarak farkların çıkartılması
- Canlı sistem ile test sistem karşılaştırarak farkların çıkartılması
- VT performans analizi ve VT güvenlik (Db Audit) yönetimi
- VT şama ve farklı sürümler arasında nesne taşıma
- PL/SQL editör ve hata ayıklayıcı
- Java uygulama geliştirme editörü
- Kullanıcı ve kapasite yönetimi
- Oracle kapasite ve kaynak planlaması
- Örnek veri oluşturulması ve görev planlaması yönetimi
- Excel ve txt dosyalarını VT tablolarına taşıma
- VT nesnelerinin tarihsel olarak sürümlerini saklama gibi özellikler bulunur.

4.6 Oracle Enterprise Manager ve Grid (Oracle)

Oracle'ın kendi ürünü olup oracle9i veritabanı ve sonraki sürümlerin içerisinde hazır gelmektedir. Diğer ürünlere göre oldukça gelişmiş özelliklere sahiptir. [17], [24], [25] Genel özellikler şu şekilde özetlenebilir:

- Veritabanı yönetim aracı
- Veritabanı performans/durum izleme (database monitoring)
- Tek bir ekran ile birden fazla farklı Oracle sürümlerinde veritabanılarını yönetme ve performans iyileştirme özelliği
- Yedek alma ve yedekten geri dönme özelliği
- Kapasite yönetimi ve kaynak planlaması
- Veritabanı güvenlik (Database Audit) yönetimi
- Veritabanı log inceleme
- SQL ve PL/SQL kodlarının analizi ve iyileştirme

- VT ile oracle desteke (Oracle support-Metalink) doğrudan haberleşme
- İşletim sistemi kaynak kullanım kontrolü
- AWR ya da SP raporlarının oluşturulması
- VT ya da OS ile ilgili hata teşhisi gibi birçok özellikleri içermektedir.

5 VERİTABANI UYGULAMALARINDA PERFORMANS VE GÜVENLİK SORUNLARI VE ÖNERİLEN ÇÖZÜMLER

SQL Server, Oracle ve Sybase gibi birçok ticari veritabanları hem Çevrimiçi Analitik İşleme (OLAP) hem de Çevrimiçi Hareket İşleme (OLTP) sistemi özelliklerine sahiptirler. Ancak bir veritabanı yöneticisinin veritabanı ile ilgili ayarlamaları yapması gerekir. Veritabanının kurulumu, yedeklenmesi, bakımı, performansı ve güncellenmesi gibi düzenli bakım gerektiren işlemlerinin gerçekleştirilmesi de bu ayarlamalar ile birlikte sağlanır ve bu tür konularla ilgilenen kişiye “Veritabanı Yönetimi” (Database Administration) adı verilir.

Veritabanı yöneticisinin görev tanımları

- Veritabanı yönetim politikası oluşturmak ve koordine etmek,
- Veritabanı kapasite planlamasını; cpu, bellek, veri depolama ünitesi (storage) dosya yapısı ve network gibi donanım gereksinimlerini doğru seçmek ve gelecekteki artan iş gereksinimini karşılayacak şekilde planlanması,
- Veritabanı sunucu (database server) ve uygulama sunucuların kurulumu ve güncellenmesini,
- Veritabanı tasarımını yapmak ve yönetmek,
- Veritabanı uygulama geliştiricilerle beraber çalışarak uygulamanın ihtiyacına göre ana nesnelere (tablo, indeks, veri kütüğü, görüntü vb.) oluşturmak,
- Veritabanı yedek ve geri dönüş (backup and recovery) sistem planını yapmak ve yönetmek,
- Veritabanı güvenliğini sağlamak,
- Veritabanı performans çözümlemesini Pro-Active (Performans sorunları çıkmadan önce) ve Re-Active (Performans sorunları çıktıktan sonra, sorunun teşhisinin yapılması) olarak yapmak,

- Veritabanının sürekliliğini sağlamak,
- Çalıştığı kurum ile veritabanı firması arasındaki teknik iletişim görevini üstlenmek [16], [27] olarak listelenebilir.

Veritabanı yöneticisi seçimi: Bir veritabanı yöneticisi seçilirken iki konuya dikkat edilmeli; birincisi yeterli teknik ve donanım bilgisine sahip olması, diğeri ise mesleki ahlak açısından uygun kişiliğe sahip olması gerekmektedir.

Veritabanı kurumların her çeşit bütün verilerini içermesi nedeniyle veritabanı yöneticisinin kişiliği çok önem arz etmektedir. Bu bağlamda kişiliğini ve mesleki bilgisi kısaca aşağıdaki şekilde tanımlanabilir.

Kişilik Olarak: [16], [27]

- Kendine güvenen
- Meraklı
- Azimli
- Yaratıcı
- Bir işi kendi kendine başlatmak ve olumlu sonuçlandırmak (self-starter)
- Detaya yönelen (detail oriented)

Mesleki bilgi becerisi yukarıdaki veritabanı yönetim görev tanımını kapsamaması gerekmektedir. Bir veritabanı programcısı; veritabanı üzerinde genellikle SQL, PL/SQL komutları ile uygulama geliştiren, sistem analizinin (iş gereksinimlerini tespit eden kişi) verilmiş olduğu bilgi doğrultusundan uygulamanın ihtiyacına göre tablo tanımlayan ve tablolar arasındaki ilişkiyi kuran, ihtiyaca göre geliştirmiş olduğu kodun hızlı çalışması için tablo üzerinde indeks tanımlayan ve veri bütünlüğünü sağlamak gibi mesleki bilgiye sahip olan kişidir. Veritabanı yazılımcıları çoğunlukla veritabanı yöneticileriyle beraber çalışırlar ve uygulama geliştirirken zaman zaman veritabanı yöneticilerin teknik bilgilerine başvururlar.

Veritabanı uygulamalarında güvenlik ve performans problemlerini tespit eden yeni bir uygulama geliştirilmesindeki temel amaç, gelişen teknoloji ve artan ülke sorunları karşısında verilerin giderek daha önem kazanması ve veri depolama ünitelerindeki maliyetlerin giderek ucuzlaması nedeniyle artan veriler karşısında istenilen verilere hızlı bir şekilde ulaşmak ve yönetmek büyük bir önem kazanmıştır. Bu bağlamda dünyada kullanılan benzer ürünlerin maliyetinin yüksek ve yetersiz olması, ülkemizde bu tarz ürünlerin yaygın olmaması nedeniyle daha düşük maliyete ve bilimsel akademik çalışmalara katkı sağlamak amaçlanmıştır. Diğer önemli bir etken ise, piyasada yeterli veritabanı yöneticisi kaynağının olmaması ve maliyetlerinin oldukça yüksek olması nedeniyle uygulama içerisinde tanımlanan “örnek uygulamalar” (best practice) ile oracle veritabanı yöneticilerine katkı sağlamak amaçlanmış ve kurum yöneticilerinin anlayabileceği dilde ilgili raporlar çıkartılması amaçlanmıştır.

5.1 DAPT ‘ın Tanımı

DAPT uygulaması oracle veritabanı üzerinde çalışan uygulamaların performans ve güvenlik sorunlarını tespit eden ve bu sorunlarla ilgili çözüm önerileri sunan bir veritabanı analiz aracıdır. Oracle veritabanı performans, güvenilirlik, erişilebilirlik, ölçeklenebilirlik, güvenlik ve yönetim özelliklerini bir arada barındırmaktadır. Ancak ülkemiz de ve dünyada ki kullanıcılar yeterli teknik donanıma sahip olmadıkları için oracle veritabanında tam verimlikli alamamaktadır. Bu bağlamda geliştirilen bu uygulama kullanıcıların iş ihtiyaçları doğrultusunda analizi, sistemdeki sıkışma noktasını tespiti, teknolojik ihtiyaçlar gereksinimi belirterek gerek oracle veritabanı özelliklerini verimli kullanılmasını, gerekse olası bir donanım kaynak ihtiyacının (disk, cpu, ram vb.) yükseltimi hakkında önerilerde bulunmaktadır.

Programdan tespit edilen sorun ve çözüm önerilerinin oluşturduğu raporlar excel ve pdf formatında dosyalara aktarılabilir. Oracle veritabanında performans sorunları

genelde veritabanı sunucusu (database server) ve uygulama olmak üzere iki farklı şekilde ortaya çıkmaktadır. DAPT uygulaması mevcut veritabanı yapısını analiz ederek her seviyedeki karmaşık problemlerin çözümlenmesine katkı sağlamaktadır.

Bunları kısaca şu şekilde açıklanır: Veritabanı sunucusuyla ilgili sistem parametrelerinin düzgün yapıp yapılmadığını, kullanılan donanımın yeterli olup olmadığını, disk I/O, CPU, işletim sistemi ile oracle sunucusu arasındaki bellek dağılımının kontrolünü, veritabanının fiziksel ve mantıksal mimari yapısının kontrolünü, network sisteminin düzgün ve güvenli olup olmadığını tespit eder.

Veritabanı üzerindeki uygulamalarındaki performans ve güvenlik sorunlarını tespit eder. Veritabanı sistemindeki CPU, Bellek, disk I/O vb. gibi yoğun kaynak kullanımına neden olan SQL, PL/SQL kodlarını tespit ederek oracle veritabanı sistemlerinin istikrarlı çalışabilmesi için farklı metot ve özelliklerin kullanılmasını önerir.

Veritabanlarındaki bir diğer önemli konu ise güvenlik sorunudur. Bu konuyla ilgili piyasada çok üst düzey gelişmiş uygulamalar mevcuttur. Özellikle oracle firmasının çıkarmış olduğu Oracle Db Vault ve Oracle Audit Vault ürünleri çok iyidir. Ancak, ürünün maliyetinin oldukça yüksek olması nedeniyle çok büyük firmalar tarafında tercih edilmektedir. DAPT ise oracle'ın kendi içinde gelen hazır paketleri kullanarak daha düşük maliyete benzer hizmeti sunmaktadır. Güvenlikle ilgili özellikle küçük ve orta ölçekli firmaların ihtiyaçlarına cevap vermekle beraber akademik çalışmalara katkı sağlamayı hedeflemektedir.

DAPT dünya çapında kullanımı ve akademik çalışmalara katkı sağlamayı amaçladığı için ürünün bütün ekranları ve kullanımı İngilizce'dir. DAPT uygulaması işletim sistemi ve veritabanı sunucusu bağımsız çalışan önyüzü Delphi 2010 programlama dili, arka tarafta SQL, PL/SQL paket ve script'ler (kod dizisi) ile geliştirilmiş bir uygulamadır.

5.2 Uygulamanın Geliştirileceği Ortamın Belirlenmesi

Uygulamanın geliştirileceği ortam belirlenirken, aşağıdaki kıstaslar göz önüne alınmıştır:

- Görsel ve anlaşılabilir olması için grafik ara yüzü olması,
- Veritabanına uzaktan bağlantı kurmayı desteklemesi,
- Veritabanıyla iletişimin hızlı olması,
- Arada hiçbir katman bulunmadan VT'na direk bağlanması,
- Yüksek hızda çalışarak çoklu işlemlerde izlek (thread) kullanması,
- En kısa zamanda en az hatayla kolay uygulama geliştirmek,

Bu kıstaslar göz önüne alındığında, uygulamayı geliştirmek üzere Delphi 2010 programlama dili ve bileşenleri seçilmiştir.

5.3 Kullanım Durumu (Use Case)

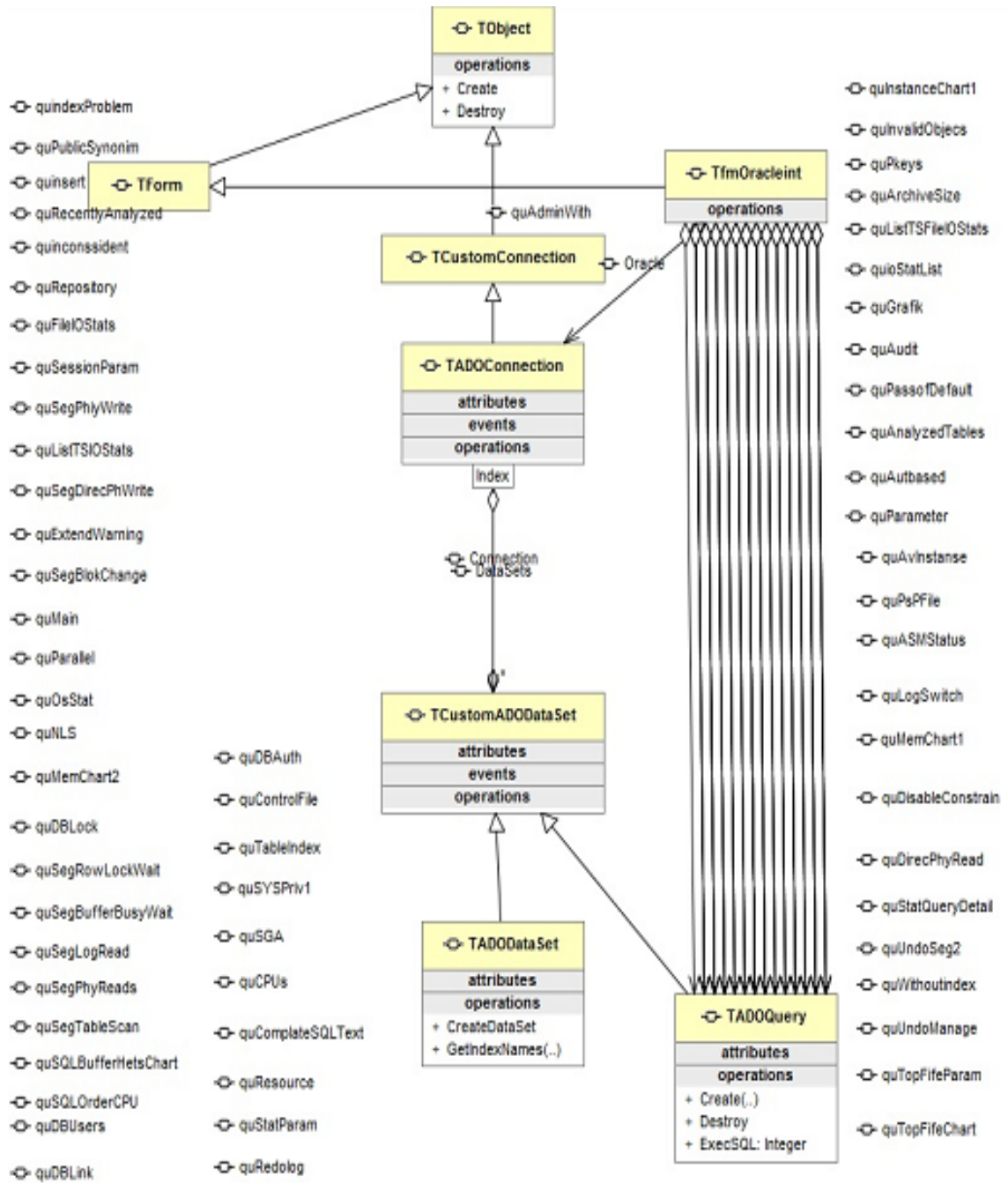
Sistemin genel işleyişi ile ilgili kullanım durumu Şekil 5.1'de gösterilmiştir. Programın kullanımında “*Veritabanı Yöneticisi*” ve “*Kullanıcı*” diye iki görüntü (profile) vardır. Kullanıcı görüntü aslına birim yöneticisidir ve yalnız ihtiyaç duyduğunda sistemi izleyerek sistem hakkında bilgi alır. Veritabanı yöneticisi (DBA) ise veritabanının tamamında sorumlu olan kişi olup veritabanının yönetim ve bakım işlerini (veritabanı güvenliğini sağlama, oturumları izleme, performans sorunlarını çözümleme, yedek alma ve hataları izleme, sistemdeki kaynak kullanımını izleme, veritabanı kontrolü vb.) gerçekleştirmektedir. Bu iki görüntü (profile) veritabanı üzerinde verilen yetki ve haklar ile belirlenir. DAPT uygulamasının kullanımı kullanıcıya verilen yetki ve haklar doğrultusunda kullanılır.



Şekil 5.1. DAPT'nin Genel Kullanım Durum Diyagramı

5.4 UML Diyagram

Tez kapsamında geliştirilen uygulamanın tablolar arasındaki ilişkiler Şekil 5.2’de gösterilmektedir. Bu tablolar, indeksler ve aralarındaki ilişkilerin yapıları oracle veritabanında tutulmaktadır. Bu ilişkileri kullanarak SQL, PL/SQL paket ve script’leri (sql komut dizisi) çalıştırılarak tez kapsamındaki uygulama geliştirilmiştir. Ayrıca, tablo ve indeks üzerinde ilgili kullanıcıya verilen hak ve yetkiler doğrultusunda veritabanına Toad ya da SQLPLS vb. gibi veritabanında uygulama geliştirilen ürünleri kullanarak tablodaki verilere erişilir ve yapısı hakkında bilgi edilebilir.

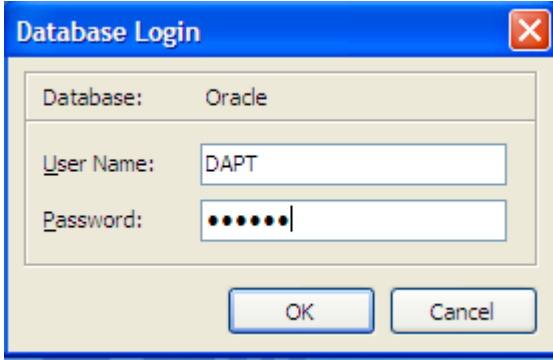


Şekil 5.2. Veritabanı Nesneleri ile İlgili Delphi UML Diyagramı

5.5 Kullanıcı Arayüzleri

5.5.1 Sisteme Giriş

Sisteme giriş ekranı Şekil 5.3’de gösterildiği gibi oracle DAPT’nin ilk giriş ekranıdır. Bu ekran kullanılarak hangi oracle VT’ya hangi kullanıcı/şifre ile girileceği belirtilir. Uygulamanın kullanılması için veritabanı da önce DAPT kullanıcı ve şifresi tanımlanmalıdır. Bu uygulama aynı zamanda oracle’ın sistem kullanıcıları ile de çalışabildiği gibi farklı bir kullanıcı ile çalışması için veritabanında ilgili kullanıcı ve şifre yaratılarak uygulamada kullanılan nesnelere erişim ve çalıştırma yetkileri verilmelidir.



Şekil 5.3. Programa Giriş Ekran Görünümü

5.5.2 Ana Ekran

Ana ekran DAPT’nin açılış ekranıdır. Bu ekranda oracle anının (instance), veritabanı ve işletim sistemi hakkında genel bilgi gösterilmektedir.

İşletim sistemi bilgileri olarak:

- Sunucu adı
- Platform, hangi işletim sistemi kurulu olduğu
- İşletim sistemi sürümü
- Kaç adet CPU olduğu
- Soket sayısı
- Fiziksel bellek miktarı (Mb)

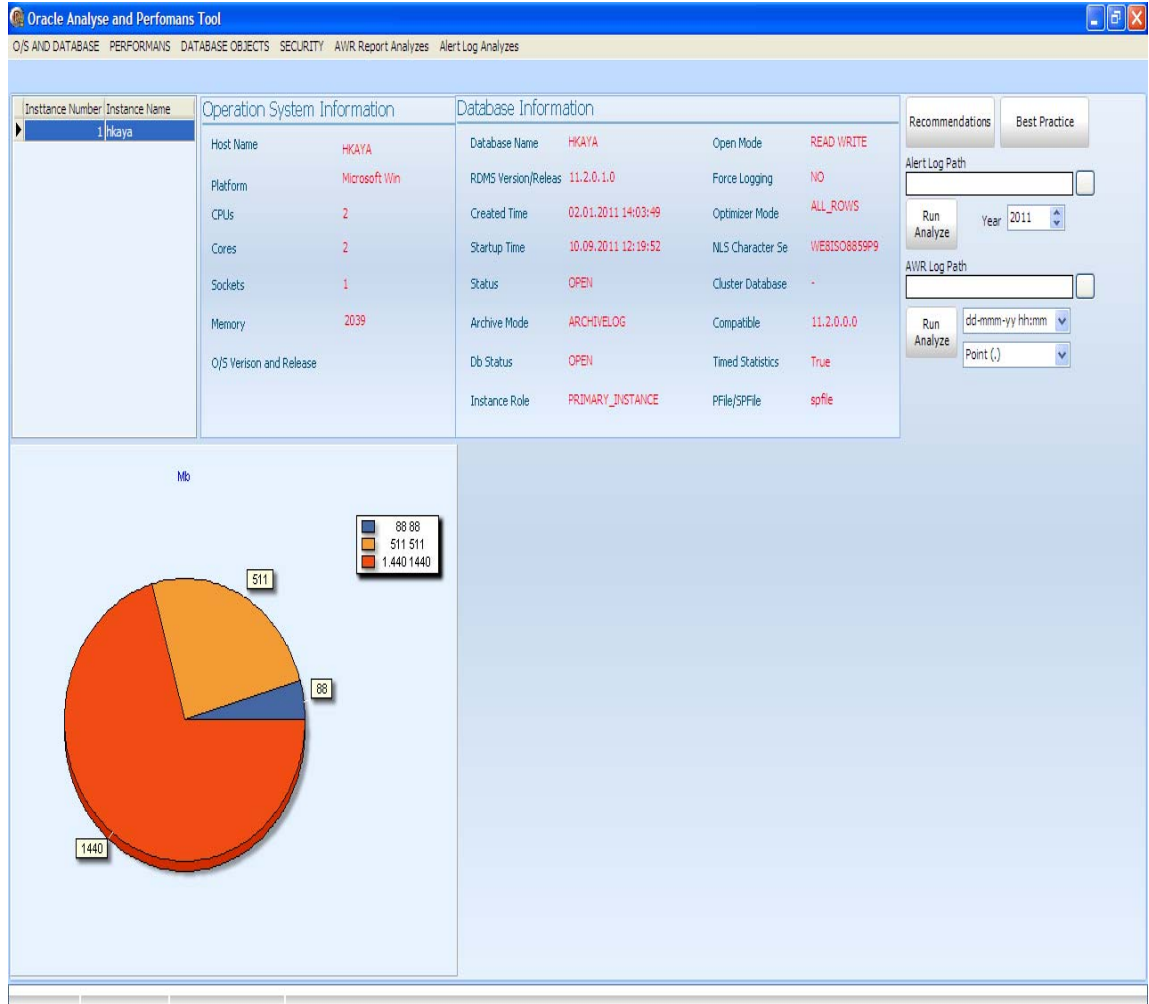
Veritabanı bilgileri olarak:

- Veritabanı ismi
- Veritabanı sürümü
- Veritabanı yaratılma tarihi
- Veritabanının en son açılış tarihi
- Veritabanının durumu
- Oracle anının rolü
- Veritabanının arşive moda çalışıp çalışmadığı
- Veritabanının hangi moda çalıştığı (okuma-yazma, sadece okuma)
- Hangi iyileştirici (optimizer) moda çalıştığı
- Karakter seti
- Veritabanı istatistiklerinin otomatik olarak alınıp alınmadığı
- Veritabanı hangi parametre kütüğü ile açıldığı

Bu ekranda aynı anda birden fazla oracle anına (oracle instance) bağlanmayı ve Gerçek Uygulama Kümeleri (RAC) veritabanına bağlanmayı desteklemektedir. Bağlanılan oracle anının isimleri ve sayısı gösterilmektedir. Hangi oracle anı seçilmiş ise bundan

sonra yapılacak olan işlemler seçilen oracle anı (oracle instance) oturumu geçerli olacaktır.

Bu bilgiler doğrultusunda veritabanıyla ilgili bazı sistem parametrelerinin kontrolü yapılıyor ve veritabanının evrensel yapısına uygun olmayan parametre tanımları varsa önerilerde bulunuyor. Aynı zaman bu ekranda veritabanı hata izleme dosyası (alert log file) ve veritabanı performansı ile ilgili istatistiksel bilgilerin tutulduğu rapor dosyalar (AWR Files) okunarak veritabanı tablolarına aktarılıyor. Şekil 5.4'de ana ekran görünümü gösterilmektedir.



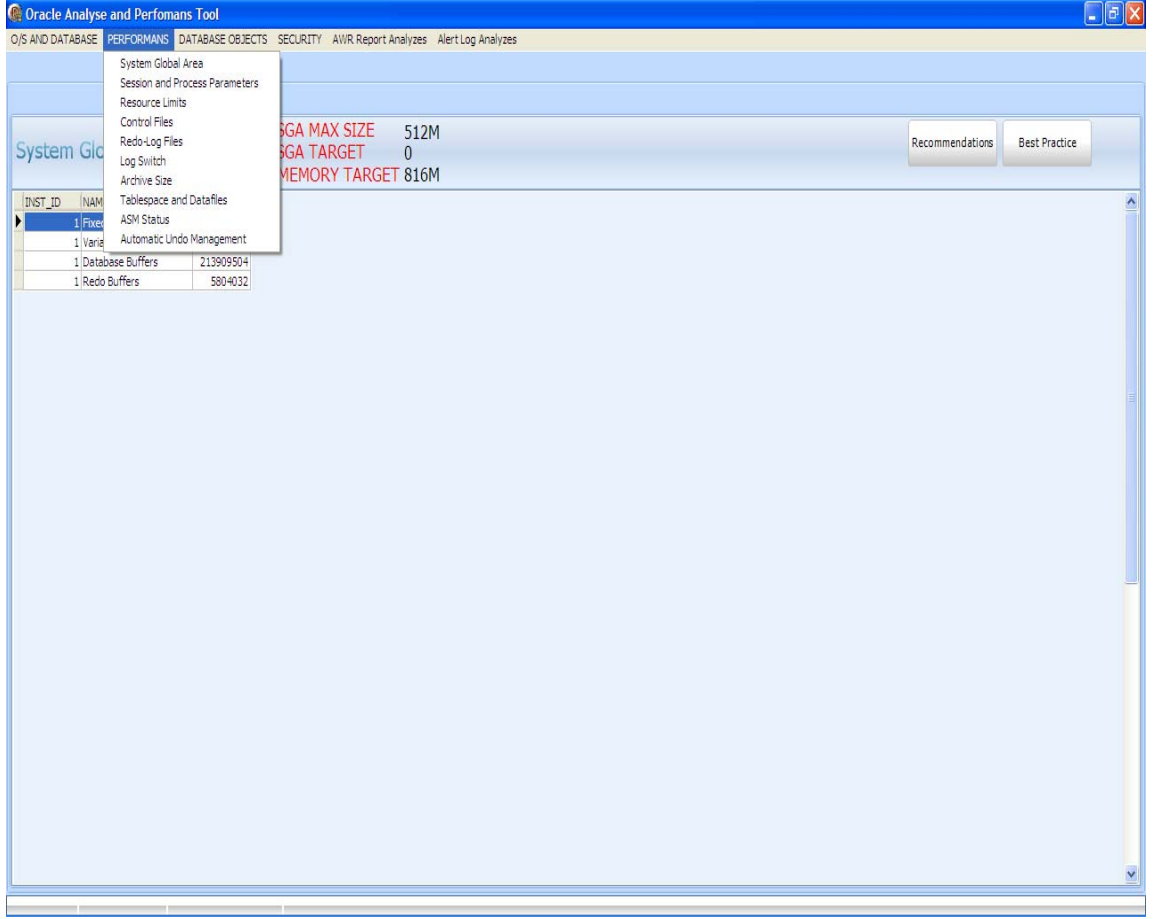
Şekil 5.4 Program Ana Ekran Görünümü

5.6 Performans

Bu bölümde oracle veritabanı üzerinde çalışan uygulamaların oracle veritabanı yapısındaki kaynakların tam olarak kullanılıp kullanılmadığı ve bu kaynakların yapısındaki değişmelerin veritabanı güvenliği ve uygulamaların performans sorunlarının tespiti ve çözüm önerileri incelenmiştir. Asıl amaç, veritabanının kullanıcılarının iş ihtiyaçları doğrultusunda analiz ederek oracle veritabanı özelliklerini tam anlamıyla verimli kullanır hale getirmektir. Şekil 5.5’da performans ekranı ve alt ekran görünümü gösterilmektedir.

Bu modül aşağıdaki bileşenleri içermektedir:

- Sistem Global Alanı (SGA)
- Oracle oturumu ve görev parametreleri (Session and Process Parameters)
- Kaynak Limitleri (Resource Limits)
- Kontrol Kütükleri (Control Files)
- Redo Log Tampon Belleği (Redo-Log Files)
- Redo Log Anahtarlanması (Log Switch)
- Arşive Kütüklerin Boyutu (Archive Size)
- Tablo Uzayı ve Veri kütükleri (Tablespace and Datafiles)
- Otomatik Disk Yönetimi (ASM)
- Otomatik Geri Alma (Automatic Undo Management)



Şekil 5.5. Performans Modülü Ana Ekran Görünümü

5.6.1 Sistem Global Alanı

Sunucu üzerinde bulunan fiziksel belleğin işletim sistemi ve oracle sunucusu arasındaki paylaşımı incelenerek fiziksel belleğin daha etkin kullanımı amaçlanmaktadır. Eğer fiziksel bellek dağılımında kaynaklanan performans sorunu varsa tespit edilir ve yeni dağılımın mevcut iş ihtiyacına göre nasıl yapılması gerektiği önerilmektedir. Ayrıca örnek bellek kullanımının nasıl olması gerektiğine ait bilgi (best practice) vermektedir.

5.6.1.1 Tespit Edilen Sorunlar ve Çözüm Önerileri

Oracle kullanan kullanıcıların veritabanları incelendiğinden yaklaşık %90'inde fiziksel belleğin OS ve Oracle veritabanı arasından düzensiz paylaşıldığı ve %95'inde ise oracle veritabanının paylaşılmış bellek yönetimini oracle'ın kendi yönetimine otomatik yönetilmesine bırakıldığı gözlemlenmiştir.

Tablo 5.1 SGA Bellek Durumunu Gösteren Örnek Uygulama

Region	Size	Size (Mb)
Database Buffers	3288334336	3136
Fixed Size	2160352	2.060
Redo Buffers	34377728	32.78
Variable Size	3489663264	3328

Toplam SGA Hesaplanması: 6814535680 (6499 Mb)

memory_max_target 6528Mb

memory_target 6528Mb

Örneğin. DWH veritabanında sunucusunda 32 G fiziksel bellek bulunmaktadır. Ancak oracle'ın SGA için tahsis edilen bellek 6.499Mb dir.

Normal şartlarda OS üzerinde yalnız Oracle veritabanı çalışıyor ise yani Oracle haricinde başka bir uygulama yoksa OS için 2 G'lik bellek yeterlidir. Bu bağlamda 25,5 G' lık bellek kullanılmıyor. Dolayısıyla kullanılmayan bu bellek oracle veritabanı için tahsis edilmelidir.

Bir diğer sorun ise oracle'a tahsis edilen belleğin oracle tarafından otomatik yönetiliyor olmasıdır. Oracle 10 G sürümüyle beraber gelen bu özellik oracle tarafından sağlıklı çalışmamaktadır. SGA içerisindeki bellek kullanım dağılımını zaman zaman sağlıklı yapılmadığı gözlemlendiği için otomatik bellek yönetimi özelliği kullanılmamalı SGA parametreleri el ile yapılmalıdır.

5.6.2 Oturum ve Görev Parametreleri

Oracle veritabanındaki oturum ve görev parametreleri (Session and Process Parameters) kontrol ediliyor. Veritabanı üzerinde çalışan uygulamaların bu parametrelerde kaynaklanan bir sorun varsa onlar tespit ederek çözüm önerileri sunmaktadır.

5.6.2.1 Tespit Edilen Sorunlar ve Çözüm Önerileri

Oracle veritabanında işlemlerin ve uygulamayı kullanacak olan oturumların (processes and sessions) sayısı “init.ora” içerisindeki parametrelerle belirlenmiştir. Oracle kullanan kullanıcıların uygulamaları incelendiğinden genelinde bu parametrelerin yanlış değer verilmesinden kaynaklanan performans sorunları olduğu tespit edilmiştir.

Tablo 5.2 Kullanıcı Oturum ve İşlemleri Gösteren Örnek Uygulama

Parametre İsmi	Değer
Processes	3000
sessions	2000

- Bu parametreler OS kaynaklarına göre değer verilmeli,
- Sessions (oturumlar) parametresinin Processes (işlemler) parametresine bağlı olarak değiştirilmeli ve $SESSIONS = (1.1 * PROCESSES + 5)$ formülü ile hesaplanmalı,
- Bu parametrelerin daha etkin kullanılması için kullanıcılar profile (görüntü) atanmalı ve aylak kalma (idle time) değeri aktif edilmelidir.

5.6.3 Kaynak Limitleri

Veritabanında kullanılan mevcut kaynakların iş ihtiyacını karşılayıp karşılamadığı kontrol edilir ve herhangi bir sorun gözlemlendiğinde çözüm önerisi yapılmaktadır.

5.6.4 Kontrol Kütüğü

Oracle veritabanı varsayılan (default) değerlerle kurulduğunda bir tane kontrol kütüğü yaratılmaktadır. Veritabanının güvenli ve verimli çalışabilmesi için en az iki tane kontrol kütüğüne ihtiyaç var. Kontrol kütükleri incelenerek veritabanının daha verimli çalışabilmesi için çözüm ve örnek uygulama metotları öneriliyor.

5.6.4.1 Tespit Edilen Sorunlar ve Çözüm Önerileri

Örneğin: INTDBPR veritabanı incelendiğinde bir tane kontrol dosyası bulunduğu tespit edilmiştir.

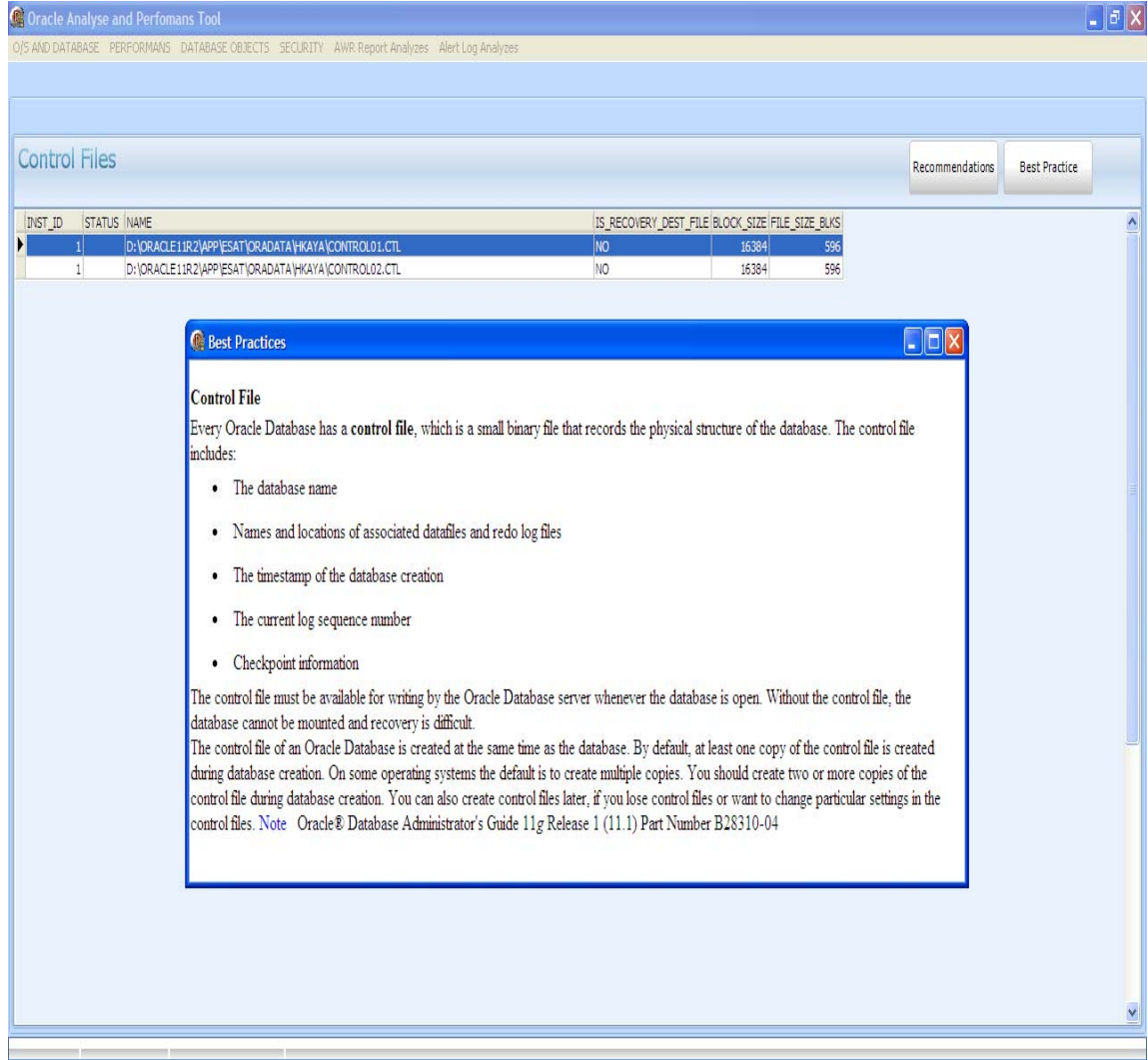
Tablo 5.3 Kontrol Kütüğü Durumunu Gösteren Örnek Uygulama

Oracle adı	İsim	Durum
INTDBPR	+DATA/intdbpr/controlfile/current.266.739130825	null

- Kontrol dosyası kayıpları zaman zaman olabilmektedir. Redo-log dosyaları gibi, kontrol dosyaları da muhtemel kayıplara karşı mirror (aynısının kopyası) edilmelidir. Kontrol dosyası kaybedildiği zaman kurtarma yolu için iki alternatif vardır:

1. Kontrol dosyasını yeniden yaratmak,
 2. Geçmişteki bir kontrol dosyasını kullanmak ile olur.
- Kontrol Dosyasını yeniden yaratmak: Kontrol dosyası, *'create controlfile'* komutu ile yaratılır. Oracle, kontrol dosyası yaratan script'i (sql kod dizisi), *'alter database backup controlfile to trace'* komutu ile *'background_dump_dest'*'de oluşturur. Fakat bu komutu çalıştırmak için veritabanı "mount" durumunda olmalıdır. Veritabanını "mount" durumunda almak için kontrol dosyası gerekeceğinden, bu yöntem kontrol dosyası kaybedildikten sonra kullanılamaz. Bu sebeple *'create controlfile ...'* script'i (sql kod dizisi) daha önceden hazırlanmış olmalı ve hazırlandıktan sonrada veritabanına yapılan fiziksel değişiklikler biliniyor olmalıdır.

Şekil 5.6'de kontrol kütüğü analiz durumuyla ilgili ekran gösterilmektedir. Bu örnekte, örnek uygulama (Best Practices) yöntemleri önerilmektedir.



Şekil 5.6. Kontrol Kütüğü Analizi Ekran Görünümü

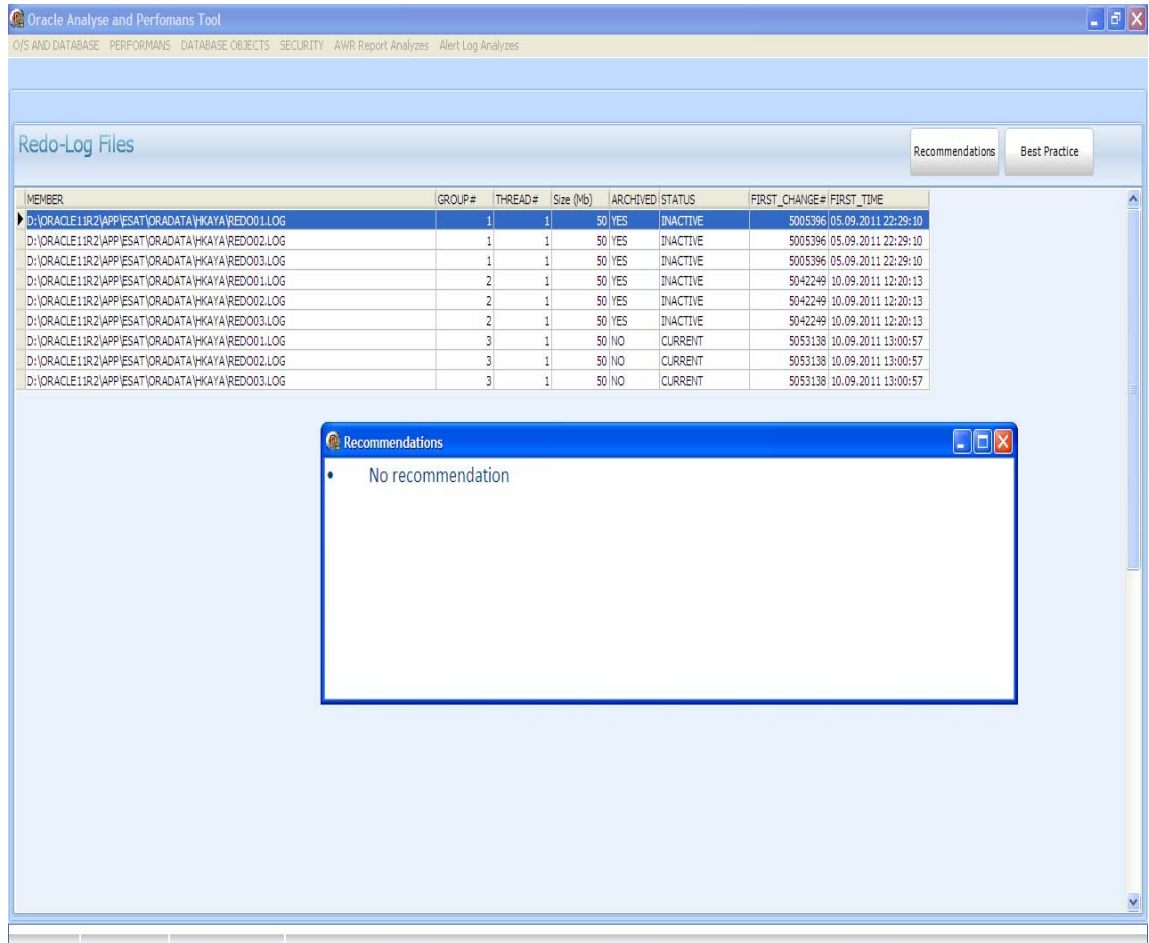
5.6.5 Redo-Log Kütükleri

Veritabanı çalışırken verileri geçici olarak Redo-Log kütüklerinde saklarlar. Dolayısıyla bu kütüklerin sayısı ve boyut veritabanı üzerinde çalışan uygulamaların hem performans bağlamında hem de medya arızasından (media failure) kaynaklanan veritabanı çöktüğünde ya da veritabanı dosyalarının, uygulama objelerinin (tablo, indeks vb.)

herhangi birinde bozulmalar (corruption) olması nedeniyle veritabanı kurtarılmak istendiği zaman bu kütükler hayati önem teşkil etmektedir. Bu bağlamda bu kütükler incelenir, mevcut iş ihtiyacı ve gelecekte doğabilecek iş yükünü karşılayabilecek çözüm önerileri sunulmaktadır.

5.6.5.1 Tespit Edilen Sorunlar ve Çözüm Önerileri

Şekil 5.7’de Redo-Log kütüklerinin analiz ekranı gösterilmektedir. Bu örnekte yapılan analiz sonucundan Redo-Log kütükleriyle ilgili herhangi bir sorun tespit edilmemiştir.



MEMBER	GROUP#	THREAD#	Size (Mb)	ARCHIVED	STATUS	FIRST_CHANGE#	FIRST_TIME
D:\ORACLE1\OR2\APP\ESAT\ORADATA\HKAYA\REDO01.LOG	1	1	50	YES	INACTIVE	5005396	05.09.2011 22:29:10
D:\ORACLE1\OR2\APP\ESAT\ORADATA\HKAYA\REDO02.LOG	1	1	50	YES	INACTIVE	5005396	05.09.2011 22:29:10
D:\ORACLE1\OR2\APP\ESAT\ORADATA\HKAYA\REDO03.LOG	1	1	50	YES	INACTIVE	5005396	05.09.2011 22:29:10
D:\ORACLE1\OR2\APP\ESAT\ORADATA\HKAYA\REDO01.LOG	2	1	50	YES	INACTIVE	5042249	10.09.2011 12:20:13
D:\ORACLE1\OR2\APP\ESAT\ORADATA\HKAYA\REDO02.LOG	2	1	50	YES	INACTIVE	5042249	10.09.2011 12:20:13
D:\ORACLE1\OR2\APP\ESAT\ORADATA\HKAYA\REDO03.LOG	2	1	50	YES	INACTIVE	5042249	10.09.2011 12:20:13
D:\ORACLE1\OR2\APP\ESAT\ORADATA\HKAYA\REDO01.LOG	3	1	50	NO	CURRENT	5053138	10.09.2011 13:00:57
D:\ORACLE1\OR2\APP\ESAT\ORADATA\HKAYA\REDO02.LOG	3	1	50	NO	CURRENT	5053138	10.09.2011 13:00:57
D:\ORACLE1\OR2\APP\ESAT\ORADATA\HKAYA\REDO03.LOG	3	1	50	NO	CURRENT	5053138	10.09.2011 13:00:57

Şekil 5.7. Redo-Log Kütükleri Analiz Ekran Görünümü

5.6.6 Redo Log Anahtarlanması

Bir Redo-Log dosyası dolduđu zaman bir sonraki redo log dosyasına geilir ve bu olaya redo log anahtarlanması (Log Switch) denir. Veritabanında ok sık redo log anahtarlanmasının olmaması gerekiyor, 15 dakikada bir redo log anahtarlanması olması oracle tarafında kabul edilmektedir. Ancak oracle veritabanı kullanan kullanıcılar genellikle bilimsel olmayan deneme yanılma yöntemiyle redo log anahtarlanmaların sayısını belirlemeye alışmaktadırlar. DAPT ise bu konuya formüsel bir yaklaşımla iş ihtiyacını karşılayabilecek özüm sunmaktadır.

- Oracle en ok 15 dakikada bir Log-switch olmasını önerir. Normal şartlarda 15–30 dakikada bir redo log anahtarlanması yapılmalıdır. Redo log dosyalarının boyutu aşıđıdaki formüle göre hesaplanır. Yeni Redo Log Size= (Bir saat için oluřan güncel Redo Log Anahtarlanması * Redo Log Size) / (Bir saat için istenilen Redo Log Anahtarlanma Sayısı)


5.6.6.1 Tespit Edilen Sorunlar ve özüm Önerileri


Tablo 5.4’de DWH veritabanına ait örnek bir uygulama gösterilmiştir. Tabloda 24 saat bir saat’lik dilimlere bölünmüş ve her saat içerisinde oluřan log anahtarlanma sayıları gösterilmektedir. Bu uygulamada ok sık redo log anahtarlanması yapıldığı tespit edilmiştir.

Tablo 5.4. Log Anahtarlama Durumunu Gösteren Öretek Uygulama

DAY	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23
2010-07-05	18	0	3	16	5	22	37	1	0	3	5	3	5	3	22	10	3	3	1	0	0	1	0	1
2010-07-06	17	1	2	16	4	2	21	2	0	3	5	3	5	3	24	3	2	3	2	1	0	1	0	1
2010-07-07	17	1	5	15	9	19	5	1	0	3	5	4	6	2	24	6	3	2	1	0	1	0	1	0
2010-07-08	18	0	3	16	18	2	41	28	1	1	3	1	3	1	24	2	1	3	0	1	0	1	0	1
2010-07-09	18	0	2	16	19	5	24	42	2	5	5	4	8	5	27	3	5	3	0	1	0	1	0	1
2010-07-10	16	1	2	16	13	24	1	0	0	1	3	0	4	41	29	0	1	0	1	0	1	0	1	0
2010-07-11	17	0	2	17	13	12	13	0	1	0	3	0	3	1	67	6	0	1	0	1	0	1	0	1
2010-07-12	16	1	2	16	13	23	0	1	0	3	6	3	5	6	27	10	3	3	0	1	0	1	0	1
2010-07-13	17	1	0	1	0	3	0	1	0	3	6	3	6	3	25	4	3	3	0	1	0	1	0	1
2010-07-14	19	13	5	3	19	2	1	0	1	4	5	7	9	28	41	8	3	3	1	0	1	0	0	1
2010-07-15	19	18	18	38	7	3	0	1	0	3	5	3	5	23	5	3	3	2	1	0	1	0	1	0
2010-07-16	18	6	21	14	41	5	1	0	1	3	5	1	4	22	4	2	1	2	0	1	0	1	0	1
2010-07-17	16	23	17	42	6	2	1	0	1	0	3	0	3	21	3	1	0	1	0	1	0	1	0	1
2010-07-18	16	18	21	43	5	3	0	1	0	1	3	0	3	21	3	1	0	1	0	1	0	1	0	1
2010-07-19	18	22	29	25	8	2	1	0	1	1	2	1	4	23	5	1	2	1	1	0	1	0	1	0
2010-07-20	18	21	14	36	8	2	1	0	1	1	2	5	5	23	1	2	3	2	1	1	0	1	0	1
2010-07-21	17	20	37	41	6	2	1	0	2	3	5	7	5	24	6	2	3	3	1	0	2	1	0	1
2010-07-22	17	16	40	41	7	2	1	0	10	2	6	3	5	37	3	3	3	3	0	2	1	0	1	0
2010-07-23	18	18	40	44	1	3	0	1	0	3	4	4	5	25	5	3	4	3	1	2	0	1	0	1
2010-07-24	19	4	55	42	8	2	0	1	1	0	3	0	3	24	3	0	1	0	1	2	0	1	0	0
2010-07-25	17	7	19	37	37	10	1	0	1	0	3	1	3	23	3	1	0	1	0	2	1	0	1	0
2010-07-26	20	27	69	5	2	3	0	1	0	3	3	3	5	26	5	3	4	4	1	2	1	0	0	1
2010-07-27	0	1	0	0	0	2	0	0	1	49	5	3	6	2	3	2	3	3	1	2	0	1	0	0
2010-07-28	17	3	17	27	42	3	7	1	0	3	29	3	5	3	2	3	3	3	1	2	1	0	0	1
2010-07-29	19	5	43	13	41	9	1	0	1	3	5	3	5	3	5	3	27	3	1	2	1	0	1	0
2010-07-30	18	6	37	18	43	9	1	0	1	3	2	3	5	3	5	6	27	3	0	3	0	1	0	1
2010-07-31	15	4	17	28	16	50	1	0	1	0	3	0	3	1	3	0	25	1	0	3	0	1	0	1
2010-08-01	17	11	50	40	3	8	0	1	0	1	3	0	3	1	3	0	26	0	1	2	1	0	1	0
2010-08-02	19	14	42	41	7	2	1	0	1	2	3	2	5	3	3	17	40	6	0	3	1	1	6	1
2010-08-03	16	7	16	36	45	8	0	1	0	3	6	2	6	4	5	3	29	3	0	3	0	1	0	0
2010-08-04	17	7	38	15	41	9	1	0	1	5	6	5	5	2	0	0	0	0	0	0	0	0	0	0

 Tablodaki kırmızı bloklarda redo log anahtarlama aktiviteleri çok çok yüksek.

 Tablodaki mavi bloklarda redo log anahtarlama aktiviteleri çok yüksek.

 Tablodaki sarı bloklarda Redo log anahtarlama aktiviteleri biraz yüksek.

Tablodaki diğer kalan zaman aralıklarında oluşan redo log anahtarlanması (Log-switch) ler normal gözükmemektedir. Büyük değişikliklerin olduğu zaman aralıklarında özel bir işlem yapılmışsa, örneğin test işlemi, veri yükleme vb. tablonun NOLOGGING opsiyonunun kullanılması önerilir. Ancak her gün aynı yoğunlukta redo log anahtarlanması oluyorsa, yukarıdaki formüle göre oracle redo log dosyalarının boyutunun artırılmasını önerilir.

5.6.7 Arşiv Kütüklerinin Boyutu

Veritabanı arşiv (archive) durumunda çalışırken yazılmakta olan redo-log dosyası dolduğunda bizim belirleyebileceğimiz bir dizine veya yedek kasete “ARC” işlem görevi tarafından kopyalanır. Her log anahtarlandığında “on-line redo log” dosyası arşivlenir. Bu olay veritabanı yaşadıkça sürekli devam eder. Dolayısıyla arşiv alanı dolduğu zaman veritabanı pasif (database hang) olur. Bu durum genellikle ya veritabanı kapasite planlamasının düzgün yapılmadığı durumlarda ya da veritabanı üzerinde özel büyük bir değişiklik durumlarda oluşur. Örneğin çok büyük boyutta veri yükleme, veri güncelleme, veri silme vb. durumlar.

Bu bağlamda arşiv boyutları günlük olarak kontrol edilmeli ve veritabanının sürekliliği açısından gerekli önlemler alınmalıdır. DAPT uygulaması sadece tarihsel olarak arşiv kütüklerinin boyutu ve log switch sayısı hakkında bilgi vermektedir.

5.6.8 Tablo Uzayı ve Veri Kütükleri

Veritabanında veriler veri kütüklerinde tutulmaktadır, bir ve birden fazla veri kütüğü tablo uzayını (tablespace) oluşturur. Veri kütükleri dolduğu zaman otomatik genişleme

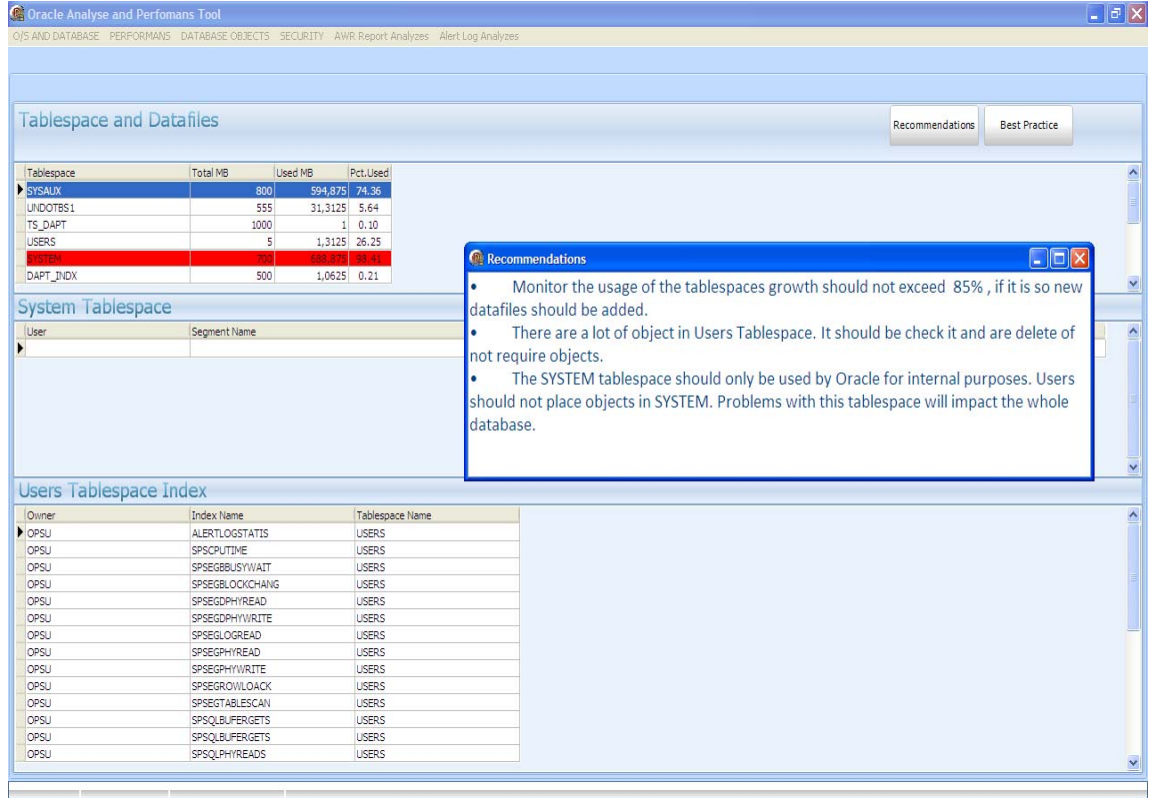
özelliđi (auto extent) kullanılmamış ise ilgili kütük üzerinde işlem yapılamayacağı için sorunları ortaya çıkmaktadır. Şayet veri kütüğü otomatik genişleme özelliđi kullanılmış ise veri kütüğü dolduđunda belirtilen deđer kadar disk üzerinde alan oracle özgü biçimlenerek genişleyecektir ki bu da belli bir zaman alacaktır.

Dolayısıyla bu işlemler uygulamada performans sorunlarına sebep olacaktır. Bu bağlamda tablo uzayı ve veri kütüklerinin kullanım oranları kontrol ediliyor ve bu oran %85 geçince gerekli önlemlerin alınması için çözüm önerileri sunmaktadır.

5.6.8.1 Tespit Edilen Sorunlar ve Çözüm Önerileri

Şekil 5.8’de tablo uzayı ve veri kütüklerinin kullanımıyla ilgili ekran örneđi gösterilmektedir. Bu örnekten “System” ve “User” tablo uzaylarında problem olduđu tespit edilmiştir.

Tablo uzayı ve veri kütüğü (Tablespace and Datafile) ile kullanıcı veri kütüğündeki indeks’lerin (Users Tablespace Index) oluşturduđu raporlar sağ kılık fare (maus) ile Excel dosyasına kayıt edilmektedir. Böylece programın oluşturmuş olduđu bütün raporla dışarı aktarılabilmektedir.



Şekil 5.8 Tablo Uzakı ve Veri kütüklerinin Analiz Ekran Görünümü

SYSTEM veri kütüğü yalnızca oracle internal (kurulumla beraber gelen kullanıcılar) kullanıcıları için kullanılır, diğer kullanıcılara ait nesnelere bu tablo uzayında olmamalıdır. Oracle veritabanında USER veri kütüğü sadece veritabanı kullanıcı bilgilerini tutmaktadır, özellikle şema kullanıcı bilgileri için özel veri kütüğü yaratılmalı ve nesnelere (tablo, indeks vb.) bu kütüklerden saklanmalıdır. Oracle10g ile beraber veritabanındaki istatistiklerin toplanması otomatik yapılır duruma getirilmiştir. Her saat veritabanının resmi (snapshot) alınıyor ve bu bilgiler SYSAUX tablo uzayında tutuluyor. Veritabanı üzerinde çalışan uygulamaların tablo ve indeksleri aynı veri kütüğünde tutulduğu zaman disk üzerinde I/O sorunlarına sebep olacağı için farklı veri kütüklerinde saklanması oracle tarafında önerilmektedir. [36]

Tablo uzayı ve veri kütüklerinin kullanım artış oranları izlenmeli, kullanım oranları %85'i geçmemeli. Bu oranın geçilmesi durumunda yeni veri kütüğü eklenerek ya da veri kütüklerinin boyutu artırılarak önlem alınmalıdır.

USER ve SYSTEM tablo uzaylarında birçok uygulama şema kullanıcılarına ait nesnelere bulunmaktadır. Bu nesnelere kontrol edilerek kullanılmayanlar düşürülmeli (drop edilmeli) ve uygulama şema kullanıcılarına ait olan nesnelere uygulamanın ait olduğu tablo uzayına taşınmalıdır.

5.6.9 Otomatik Disk Yönetimi

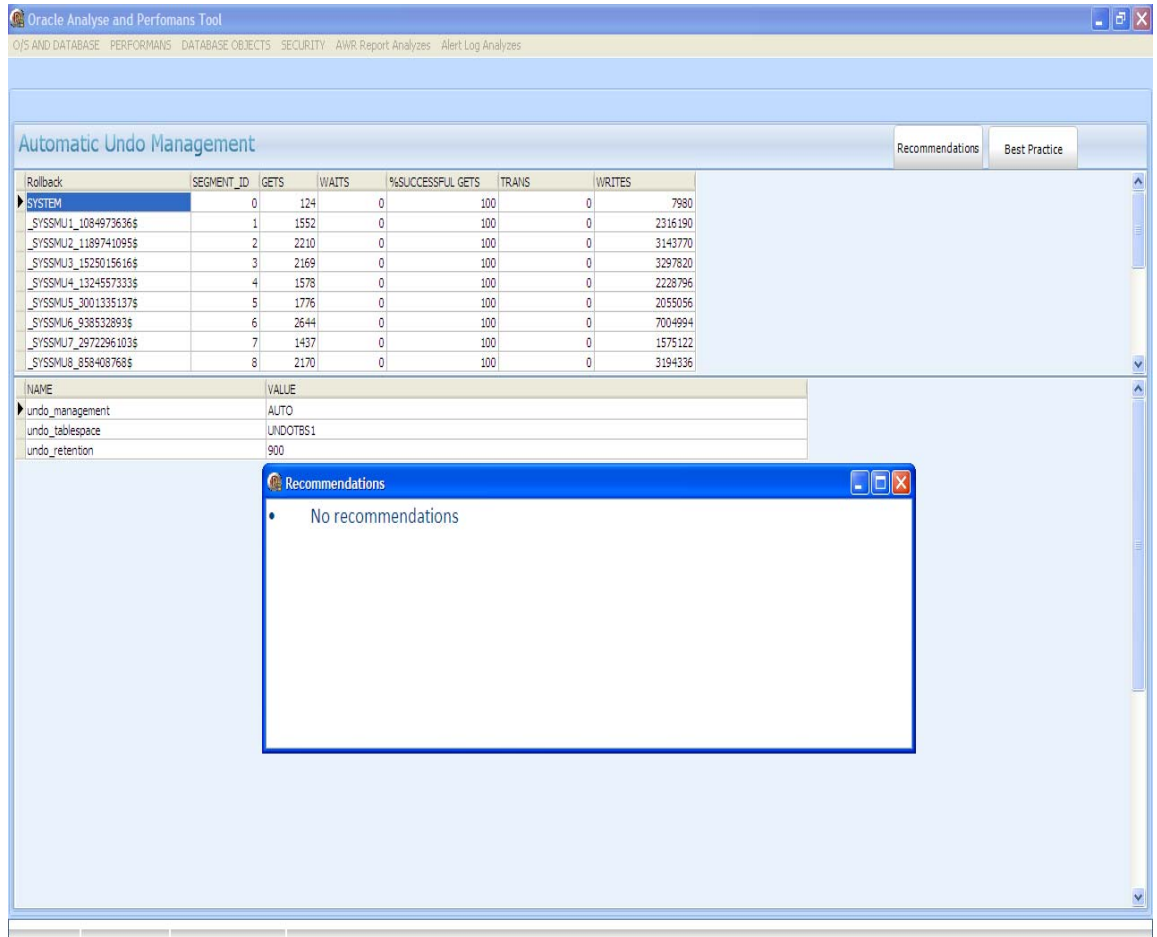
Veritabanında otomatik disk yönetim (Automatic Storage Management (ASM)) kullanılmış ise kullanım durumu hakkında (dolu-boş) bilgi göstermektedir. Amaç, donanımla ilgili kaynak ihtiyacının olup olmadığının araştırılmasıdır.

5.6.10 Otomatik Geri Alma

Oracle9i ile beraber otomatik geri alma (Automatic Undo Management) kullanılmaya başlandı, daha önceki sürümlerden Rollback Segment'leri kullanılıyordu. Otomatik geri almayı aktif duruma getirmek için UNDO_MANAGEMENT = AUTO parametresini set etmek gerekiyor ki bu durum ORA-1555 hatasının oluşumunu engelliyor. UNDO_RETENTION parametresinde belirtilen onaylama (commit) aralığındaki oracle oluşan bilgileri tutulmaktadır. [17]

5.6.10.1 Tespit Edilen Sorunlar ve Çözüm Önerileri

Şekil 5.9’da otomatik geri alma (Automatic Undo Management) ekranı gösterilmektedir. Bu örnekte otomatik geri almanın kullanımıyla ilgili herhangi bir problem tespit edilmemiştir. Otomatik geri alma alanı inceleniyor, performans problemi teşkil eden durum tespit edilerek çözüm ve örnek uygulamalar önerisi vermektedir. Örneğin: “Seccessfully Gets” lerin başarısız sonuçlanması, sorguların uzun sürmesi, alma (gets) ve bekleme (waits) değerlerinin aşırı yüksek olması durumlarından ortaya çıkmaktadır. Bu durumda performans problemine neden olan sorgu iyileştirilmelidir.



Oracle Analyze and Perfomans Tool

Q/S AND DATABASE | PERFORMANS | DATABASE OBJECTS | SECURITY | AWR Report Analyzes | Alert Log Analyzes

Automatic Undo Management

Recommendations | Best Practice

Rollback	SEGMENT_ID	GETS	WAITS	%SUCCESSFUL GETS	TRANS	WRITES
SYSTEM	0	124	0	100	0	7980
_SYSMU1_1084973636\$	1	1552	0	100	0	2316190
_SYSMU2_1189741095\$	2	2210	0	100	0	3143770
_SYSMU3_1525015616\$	3	2169	0	100	0	3297820
_SYSMU4_1324557333\$	4	1578	0	100	0	2228796
_SYSMU5_3001335137\$	5	1776	0	100	0	2055056
_SYSMU6_938532893\$	6	2644	0	100	0	7004994
_SYSMU7_2972296103\$	7	1437	0	100	0	1575122
_SYSMU8_858408768\$	8	2170	0	100	0	3194336

NAME	VALUE
undo_management	AUTO
undo_tablespace	UNDOTBS1
undo_retention	900

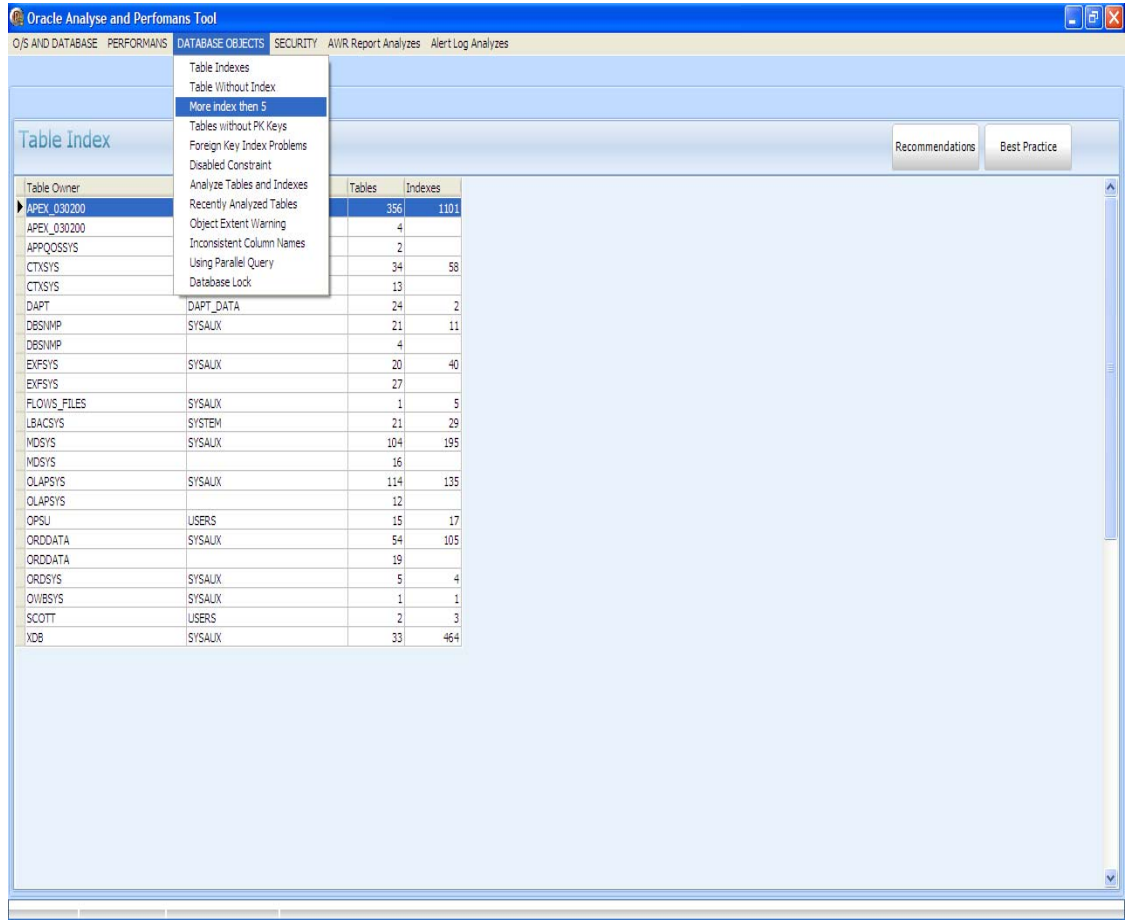
Recommendations

- No recommendations

Şekil 5.9. Otomatik Geri Alma Kullanım Analiz Ekran Görünümü

5.7 Veritabanı Nesneleri

Bu bölümde veritabanı üzerindeki uygulamaların nesneleri (obje) incelenmektedir. Uygulamalardaki performans problemlerine sebep olan dar boğazlar tespit edilerek çözüm önerileri ve örnek uygulama metotları verilmektedir. Şekil 5.10'da veritabanı nesnelерinin modülü ekranı gösterilmektedir.



The screenshot shows the Oracle Analyze and Performans Tool interface. The 'Table Index' module is active, displaying a list of database users and their associated tables and indexes. A context menu is open over the 'Table Index' header, showing options such as 'Table Indexes', 'Table Without Index', 'More Index than 5', 'Tables without PK Keys', 'Foreign Key Index Problems', 'Disabled Constraint', 'Analyze Tables and Indexes', 'Recently Analyzed Tables', 'Object Extent Warning', 'Inconsistent Column Names', 'Using Parallel Query', and 'Database Lock'. The table below shows the following data:

Table Owner	Tables	Indexes
APEX_030200	336	1101
APEX_030200	4	
APPOSSYS	2	
CTXSYS	34	58
CTXSYS	13	
DAPT	24	2
DBSNMP	21	11
DBSNMP	4	
EXFSYS	20	40
EXFSYS	27	
FLWIS_FILES	1	5
LBACSYS	21	29
MDSYS	104	195
MDSYS	16	
OLAPSYS	114	135
OLAPSYS	12	
OPUS	15	17
ORDDATA	54	105
ORDDATA	19	
ORDSYS	5	4
OWBSYS	1	1
SCOTT	2	3
XDB	33	464

Şekil 5.10 Veritabanı Nesnelерinin Görünüm Ekranı

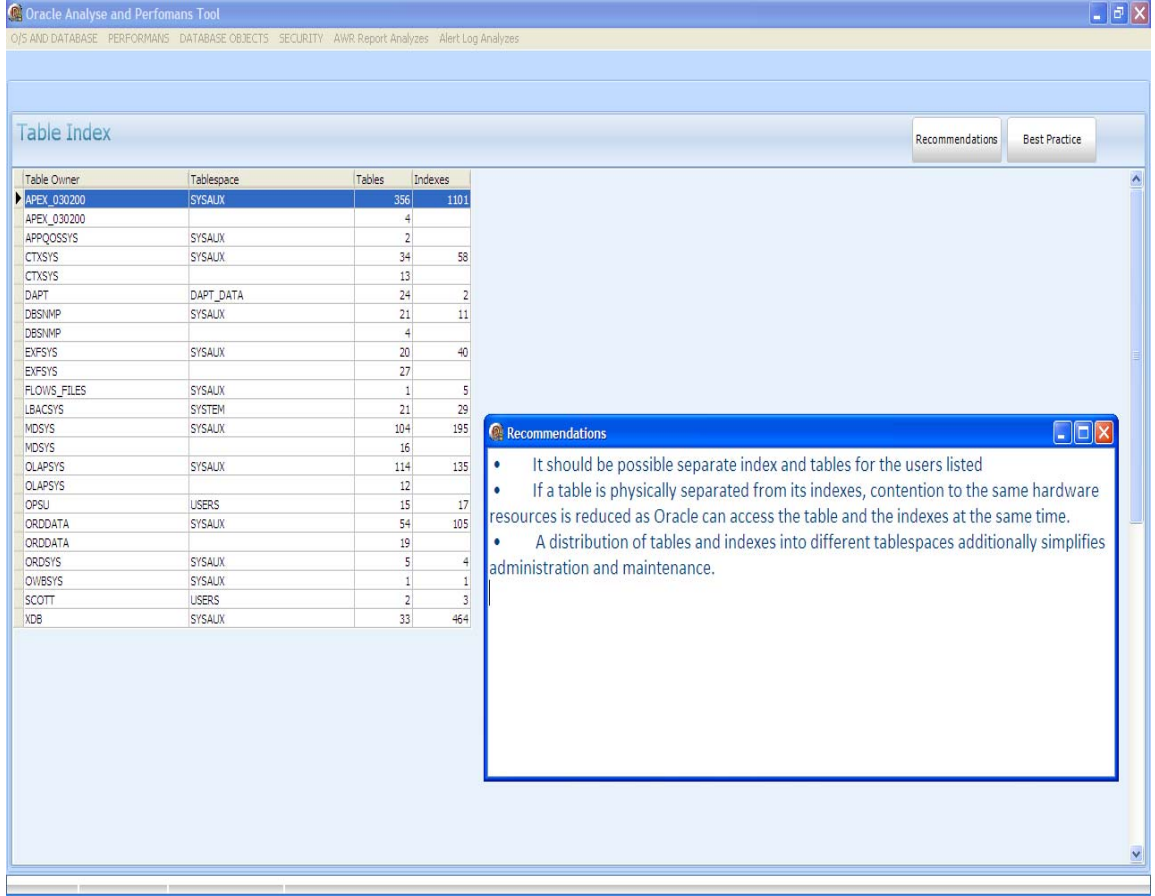
5.7.1 Tablo ve İndekslerin Farklı Tablo Uzayında Bulunması

Veritabanı üzerinde çalışan uygulamaların tablo ve indekslerin aynı tablo uzayı içerisinde saklanıp saklanmadıkları inceleniyor. Çünkü indeks ve tabloların aynı tablo uzayı içerisinde olması veritabanında I/O sorununa sebep olduğu için tablo ve indekslerin farklı tablo uzayına (tablespace) taşınması önermektedir. [23]

5.7.1.1 Tespit Edilen Sorunlar ve Çözüm Önerileri

Şekil 5.11'de uygulama tablo ve indekslerin tablo uzaylarındaki kullanımı durumu gösterilmektedir. Bu örnekte tablo ve indekslerin aynı tablo uzaylarında yer aldığı tespit edilmiştir.

İndeks ve tabloların aynı tablo uzayında olması veritabanında I/O sorununa sebep olduğu için tablo ve indeks'lerin farklı tablo uzaylarına taşınmalıdır. Tablo ve indeksler için farklı diskler üzerinden tablo uzayları yaratılmalı ya da RAID 10 yapısı kullanılmalıdır.



Şekil 5.11. Tablo ve İndeks'lerin Tablo Uzayı Üslerindeki Dağılım Analiz Ekranı

5.7.2 Üzerinde İndeks Olmayan Tablolar

Veritabanında sorgulama yapılırken eğer tablo üzerinde indeks yoksa sorgu tablonun bütün bloklarını okur (full table scan) , yani tablodaki bütün dolu ve boş olan blokları okur. Bu durumda veritabanında I/O sorunu ortaya çıkar. Dolayısıyla I/O maliyetini azaltmak için tablo üzerinde uygun indeksleri oluşturularak çözümlene yapılır.

DAPT veritabanı üzerinde çalışan uygulamaların kullandığı bütün tabloları inceleyerek indeksi olmayanların tespit eder ve boyutuyla beraber rapor oluşturur. Ayrıca, hangi kıstaslara göre indeks yaratılması konusunda en iyi çözüm önerileri sunmaktadır.

5.7.2.1 Tespit Edilen Sorunlar ve Çözüm Önerileri

Tablo 5.5’de örnek uygulamaya ait üzerinde indeks olmayan tablo sahibi, tablo ismi ve tablo boyutu (Mb) olarak gösterilmiştir. Bu örnekte oracle veritabanı üzerinde çalışan uygulamadaki sorgular tablolara erişmek istediklerinde sorgu tablonun bütün bloklarını okur (full table scan) , yani tablodaki bütün dolu ve boş olan blokları okur. Bu durum ise uygulamada performans problemine yol açar.

Tablo 5.5. Üzerinde İndeks Olmayan Tabloları Gösteren Örnek Uygulama

Table Owner	Table Name	Table Size(Mb)
ABONE	SOZMT999	5.566
ABONE	ELBMT016OLD	5.426
ABONE	THSMT019_LG	5.183
ABONE	DBAMT010	2.920
ABONE	THKMT316_08	2.112
ABONE	THSMT251_LG	1.986
ABONE	THSRP083	10
ABONE	THSRP081	10
SACMGR	SACLG001	160
SACMGR	LG_SACMT020	112
SACMGR	SACMT060	55
SACMGR	SACLG016	12
SACMGR	SACLT001	10
VEZNE	LG_IEMMT060	9.578
VEZNE	LG_THKMT320	8.945
VEZNE	LG_SOZLT033	6.049
VEZNE	LG_SOZMT010	3.768
VEZNE	IEMMT171	3.008
VEZNE	LG_TGAMT100	2.424
VEZNE	LG_KRTMT022	24
VEZNE	LG_COMLT033	11
VEZNE	INTMT174	11
VEZNE	SIL_INTMT020	10
VEZNE	THKLT002	10

İndeks olmayan tablolar kontrol edilerek 10Mb'den büyük olanlar için uygun indeksler yaratılmalıdır. İndeks aşağıdaki kurallara uygun olarak yaratılmalıdır:

- Sorgu içerisindeki WHERE ifadesine kullanılan uygun kolon üzerinde,
- Tabloların JOINS ifadelerinden kullanılan kolon ya da kolonlar üzerinde,
- Seçiciliği yüksek olan kolon üzerinde,
- Çok fazla güncellenmeyen ve çok fazla sayıda satıra sahip tablolardaki düşük çeşitlilik (cardinality) sahibi kolonlar üzerinden "Bitmap" indeks yaratılmalıdır.
- Çok sık UPDATE, INSERT, DELETE ifadeleri kullanılan kolon üzerinde indeks yaratılması tercih edilmemeli.
- Sorgu içerisinde kullanılan WHERE ifadesindeki kolon MAX ve MIN fonksiyonunda kullanılmış ise bu kolonlar üzerinde yaratılan normal indeks RDBMS tarafında kullanılmaz, dolayısıyla normal indeks yerine function-base (fonksiyon indeks) indeks yaratılmalıdır.

5.7.3 Üzerinde Fazla İndeks Olan Tablolar

DAPT veritabanı üzerinde çalışan uygulamaların kullandığı bütün tabloları inceleyerek beşten fazla indeksi olanların listesini çıkartmaktadır ve bunlarla ilgili çözüm önerileri sunmaktadır.

5.7.3.1 Tespit Edilen Sorunlar ve Çözüm Önerileri

Tablo 5.6’da tablo sahibi, tablo ismi ve tablo üzerindeki indeks sayısı belirtilmiştir. Bu örnekte oracle üzerinde çalışan uygulamalara ait tablolar üzerinden beş’den fazla olan indeksler tespit edilmiştir. Tablo üzerinde indekslerin fazla olması, tablolar üzerinde DML işleri yapılması durumunda veritabanı uygulamasında performans problemine sebep olur.

Tablo 5.6. Üzerinde Fazla İndeks Olan Tabloları Gösteren Örnek Uygulama

Owner	Table Name	Index Count
ABONE	COMMT020	20
VEZNE	COMMT020	19
ABONE	ITRMT010	14
VEZNE	ITRMT010	14
ABONE	THKMT021	12
VEZNE	THKMT021	12
ABONE	IEMMT060	11
VEZNE	TGAMT100	10
VEZNE	COMLT033	9
VEZNE	SOZMT010	8
VEZNE	TGAMT102	8
ABONE	THKMT020	7
ABONE	THSMT037	6
VEZNE	SOZMT030	6

- Veritabanında DML (Data Manipulation Language) yapıldığı zaman tablo üzerinde bulunan bütün indeksler okunması sorgunun maliyetini artırır. Bu bağlamda indeksler kontrol edilmeli, kullanılmayan ve mükerrer olanlar düşürülmelidir.
- Oracle9i ile beraber indeksler izlenerek kullanılıp kullanılmadığının tespit ediliyor. Aşağıdaki yöntemle indeksler bir ay süreyle izlenmeli, kullanılmayanlar düşürülmelidir:

1. İndekslerin izlenmesi başlatılır

Örnek: alter index products_pk monitoring usage;

2. v\$object_usage sorgusuyla indeks'lerin kullanılıp kullanılmadığına MONITORING alanı YES ise kullanıldığı, NO ise kullanılmadığı anlaşılır. [37]

5.7.4 Birinci Anahtar Kısıtlaması Olmayan Tablolar

Veri bütünlüğünün sağlanması için birincil anahtar kısıtlamaları (Primary key constraints) kullanılır birincil anahtar kısıtlaması (Primary key Constraints) verilerin bütünlüğünü tablolara çift kayıt ve boş değer girilmesini izin vermeyerek sağlar. [38]

DAPT veritabanı üzerinde çalışan uygulamaların kullandığı bütün tabloları inceleyerek birincil anahtar kısıtlamaları (Primary key constraints) olmayanların listesini çıkartmaktadır ve bunlarla ilgili çözüm önerileri sunmaktadır.

5.7.4.1 Tespit Edilen Sorunlar ve Çözüm Önerileri

Tablo 5.7'de tabloya ait uygulamanın kime ait olduğu ve üzerinde birinci anahtar kısıtlaması olmayan tabloların sayısı belirtilmiştir. Bu örnekte oracle üzerinde çalışan uygulamalara ait birinci anahtar kısıtlaması olmayan tabloların sayısı tespit edilmiştir. Bu tablolar kontrol edilmeli ve iş kurallarına göre birinci anahtar (Primary key) kısıtlaması yaratılmalıdır.

Tablo 5.7. Birinci Anahtar Kısıtlaması Olmayan Tabloları Gösteren Örnek Uygulama

Owner	No.Tables without PK
ABONE	828
AKSYAPI	12
CKOC	20
DB_DRC	72
SACDEV	4
SACMGR	43
SU	111
VEZNE	802
XDB	9

5.7.5 İndeks Olmayan Yabancı Anahtar

Detay tablo üzerine yapılan bir DML (insert, delete, update) işlemi ana tablo üzerinde herhangi bir kilitleme işlemine yol açmaz. Fakat kayıt ekleme (insert) ve kayıt güncelleme (update) işlemleri ana tablo üzerindeki indekse ulaşım için ana tablo üzerindeki kilidi varsa bekleyeceklerdir.

Ana tablo üzerinde yapılan güncelleme (update), silme (delete) cümleleri detay tablonun kilitlemesine yol açar ancak ekleme (insert) cümlesi detay tablonun kilitlemesine yol açmaz. Ana tablo onaylama (commit) ya da yapılan değişikliği geri alma (rollback) yapana kadar detay tabloya işlem yapamaz. Örneğin kayıt ekleme (insert) yapılamaz.

Dolayısıyla bu işlem veritabanı üzerinde çalışan uygulamalarda performans sorununa yol açmaktadır. Bu bağlamda detay tablo üzerindeki yabancı anahtar kısıtlaması (FK-foreign key) indeks olarak tanımlandığında sorun çözümlenmiş olur. Yani ana tablo üzerinde yapılan kayıt güncelleme (update) , kayıt silme (delete) cümleleri eğer detay tablo yabancı anahtar kısıtlaması (foreign key) üzerinde indeks varsa detay tablo üzerine herhangi bir kilit koyulmaz. Ana tablo üzerine doğal olarak kayıt düzeyinde kilit (row level lock) konur. [23]

5.7.5.1 Tespit Edilen Sorunlar ve Çözüm Önerileri

Tablo 5.8’de oracle veritabanı üzerinde çalışan örnek uygulamalara ait yabancı anahtar üzerinden indeks olmayan tablolar tespit edilmiştir. Bu durum uygulamada performans problemine sebep olur. Bu problemin çözülmesi için yabancı anahtar üzerinde indeks yaratılmalıdır.

Tablo 5.8. İndeks Olmayan Yabancı Anahtarları Gösteren Örnek Uygulama

Owner	Table Name	Constraint Name	Column Name	Position	Problem
PR_OER	REVIEWS	FL_FK_090041	ASSETID	1	No Index
PR_OER	REGISTRARQUEUES	FL_FK_090100	QUEUETYPE	1	No Index
PR_OER	SCMHOSTS	FL_FK_090113	TYPEID	1	No Index
PR_OER	SEARCHLOGPARTS	FL_FK_090115	SEARCHLOGID	1	No Index
PR_OER	ENTSECUSERROLETYPES	FL_FK_1000052	SECUSERID	1	No Index
PR_OER	ENTSECROLETYPES	FL_FK_100046	PARENTID	1	No Index
PR_OER	ENTSECROLECONTEXTS	FL_FK_100047	PARENTID	1	No Index
PR_OER	ENTSECROLES	FL_FK_100048	ROLETYPEID	1	No Index
PR_OER	ENTSECROLES	FL_FK_100049	ROLECONTEXTID	1	No Index
PR_OER	ENTSECPERMISSIONS	FL_FK_100050	ROLEID	1	No Index
PR_OER	ENTSECPERMISSIONS	FL_FK_100051	PERMISSIONTYPEID	1	No Index
PR_OER	FORMLETTERSUBSTITUTIONINDEX	FL_FK_100126	FORMLETTERID	1	No Index
PR_OER	SFIDCANDIDATEATTRS	FL_FK_100161	SFIDCANDIDATEID	1	No Index
PR_OER	ASSETTYPEARCHETYPE	FL_FK_100165	ARCHETYPEID	1	No Index
PR_OER	ASSETTYPEARCHETYPE	FL_FK_100166	ASSETTYPEID	1	No Index
PR_OER	METADATAINDEX	FL_FK_100201	PARENTID	1	No Index
PR_OER	METADATAENTRYTYPEENTITYTYPE	FL_FK_100204	ENTITYTYPEDEFID	1	No Index
PR_OER	POLICYASSERTIONS	FL_FK_100208	ASSERTIONTYPEID	1	No Index
PR_SOAINFRA	BPM_ACTIVITY_INSTANCE	BPM_ACTIVITY_INSTANCE_FK2	ACTIVITY	2	No Index
PR_SOAINFRA	BPM_ACTIVITY_INSTANCE	BPM_ACTIVITY_INSTANCE_FK2	PROCESS	1	No Index
PR_SOAINFRA	BPM_ACTIVITY_PROPERTY	BPM_ACTIVITY_PROPERTY_FK1	ACTIVITY	2	No Index
PR_SOAINFRA	BPM_ACTIVITY_PROPERTY	BPM_ACTIVITY_PROPERTY_FK1	PROCESS	1	No Index
PR_SOAINFRA	BPM_CUBE_ACTIVITY_INSTANCE	BPM_CUBE_ACTIVITY_INST_FK2	ACTIVITYID	2	No Index
PR_SOAINFRA	BPM_CUBE_TASKPERFORMANCE	BPM_CUBE_TASKPERF_FK1	ACTIVITYID	2	No Index
PR_SOAINFRA	BPM_CUBE_TASKPERFORMANCE	BPM_CUBE_TASKPERF_FK1	PROCESSID	1	No Index
PR_SOAINFRA	BPM_CUBE_WORKLOAD	BPM_CUBE_WORKLOAD_FK1	ACTIVITYID	2	No Index
PR_SOAINFRA	BPM_CUBE_WORKLOAD	BPM_CUBE_WORKLOAD_FK1	PROCESSID	1	No Index
PR_SOAINFRA	BPM_PARTNER_SERVICE	BPM_PARTNER_SERVICE_FK1	PARTNER	2	No Index
PR_SOAINFRA	BPM_PARTNER_SERVICE	BPM_PARTNER_SERVICE_FK1	PROCESS	1	No Index

5.7.6 Pasif Kısıtlamalar

Veritabanı uygulamalarındaki verilerin bütünlüğü tablo üzerindeki kısıtlamalar (constraint) ile sağlanır. Ancak, kısıtlamalar pasif (disabled) duruma getirildiğinde verilerin tabloya girişine izin verileceği için veri bütünlüğü sağlanmaz, tutarsız veriler oluşur. DAPT veritabanındaki bütün uygulamaların tablolar üzerindeki kısıtlamalarını kontrol eder ve pasif olanların sahibini, tablo ismini, kısıtlama ismini, kısıtlama türünü ve durumu bilgilerini içeren raporunu oluşturarak çözüm önerisi yapar.

5.7.6.1 Tespit Edilen Sorunlar ve Çözüm Önerileri

Tablo 5.9’de oracle üzerinde çalışan örnek uygulamalara ait tablolardaki pasif kısıtlamalar tespit edilmiştir. Pasif kısıtlamalar uygulamalarda tutarsız verilerin oluşmasına sebep olur. Bu sorunun çözülmesi için pasif olan kısıtlamalar (Disabled Constraints) kontrol edilmeli ve ihtiyaç yok ise düşürülmelidir. Normal şartlarda veritabanından pasif kısıtlama olmamalıdır.

Tablo 5.9. Pasif Kısıtlamaları Gösteren Örnek Uygulama

Owner	Table Name	Constrant Name	Type	Status
SACMGR	SACDT032	FK_SACDT032_SACMT032	Foreign Key	DISABLED
SACMGR	SACMT016	FK_SACMT016_SACLT016	Foreign Key	DISABLED
SACMGR	SACMT033	FK_SACMT032_SACMT033	Foreign Key	DISABLED
VEZNE	OFFMT050	OFFMT050_ISLEM_NO	Unique	DISABLED
VEZNE	SIL_HUKMT020	R_249	Foreign Key	DISABLED

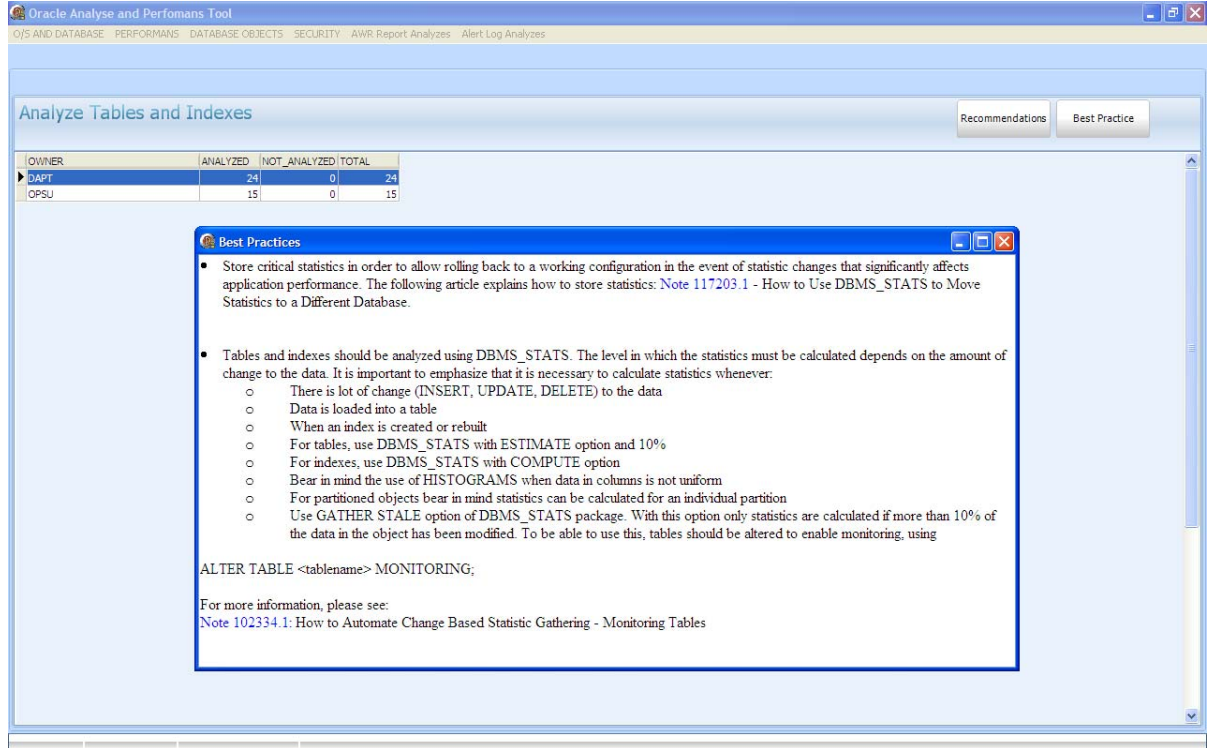
5.7.7 Analizi Yapılmamış Tablo ve İndeksler

Veritabanı maliyet iyileştirme tabanlı (cost based optimizer) durumunda çalışırken belli sürelerle, tablo ve indeksler analiz edilerek, istatistiklerin toplanması sorgunun tabloya doğru bir şekilde en iyi çalışma planını seçerek tablo ya da indekse erişimini sağlamaktadır. Oracle10g sürümüyle beraber veritabanındaki tablo ya da indekslerde bulunan verilerin yapısında büyük değişimler olduğunda istatistikleri otomatik olarak gece yarısı toplanmaktadır. [36]

5.7.7.1 Tespit Edilen Sorunlar ve Çözüm Önerileri

Şekil 5.12'de veritabanı üzerinde çalışan uygulamaların tablo ve indekslerin analiz durumlarını belirten ekran gösterilmiştir. Bu örnekte tablo ve indeks analizlerinin güncel olduğu tespit edilmiş ancak, hangi durumlarda analiz (tablo ve indeks istatistiklerinin toplanması) yapılması gerektiğini ve örnek uygulama yöntemlerini belirtmektedir.

İstatistikler alındıkları ana özgü oldukları için periyodik olarak tekrar toplanmasından fayda vardır. Özellikle büyük değişimlere yol açan veri işlemlerinden (kayıt ekleme, güncelleme, silme) ilgili veri istatistikleri değiştireceği için akabinde istatistik toplamak gerekir. Tablo ve indeks'lerin istatistiklerinin toplanma (analize) işlemini DBMS_STATS [36] Oracle veritabanı paketi kullanılarak yapılmalıdır.



Şekil 5.12. Uygulamaların Tablo ve İndekslerin Analiz Ekranı

Tablo ve indekslerin analyze (istatistikleri toplamak) kullanımı ile aşağıda bahsi geçen bazı önemli hususları göz önüne almak gerekir:

- Tablo üzerinde çok fazla kayıt ekleme, kayıt silme, kayıt güncelleme (insert, delete, update) işlemlerin yapılmasında
- Bir tabloya çok fazla veri aktarıldığında (data loaded)
- Yeni bir indeks yaratma ya da mevcut indeksi yenide oluşturmada (rebuild index)
- Bir tablo üstünde analiz o tablo ile ilgili tüm indekslere yansır.
- Bir tablo üzerinde tahmini istatistikler toplanırsa ilgili indeksler üzerinde tam istatistiklerin toplanması mantıklıdır. İndeksli alanların istatistiklerinin doğru olduğundan emin olunur.
- İndekslerin analizi esnasında geçici alan (temporary space) kullanılmaz.

- İndeksler analiz edilir fakat tablo edilmezse maliyet iyileştirme durumu (COST-BASED optimizer) seçilmez.
 - Eğer estimate (tahmini) opsiyonu seçilirse (veri çok ise) %10'luk bir veri üzerinde yapılması mantıklıdır. %50'nin üzeri seçilirse bütün veri analiz edilir.
 - Veri sözlüğü (data dictionary) tabloları yani SYS tabloları analiz edilmemelidir. Tablo kilitleri (deadlock) ya da performans problemleri çıkar.
 - Mümkün olduğu kadar istatistiklerin toplanması veritabanının müsait olduğu bir zaman yapılmalıdır. Özellikle çok hareket gören tabloların istatistiklerinin toplanmasına dikkat edilmeli, işlem yapılmadığı bir zamandan toplanmalıdır.
 - Boş tabloların istatistikleri toplanmamalıdır
- [15], [17], [21], [22], [27], [24], [27], [30], [39], [40]

5.7.8 Paralel Sorgu Kullanmak

“init. Ora” parametre dosyasında PARALLEL_MAX_SERVERS parametresinin değeri CPU sayısına göre belirlenmelidir. Böylece veritabanı üzerinden çalışan uygulama sorguların hızlı çalışması sağlanmış olunur. Özellikle sorgu ağırlıklı veritabanlarından bu parametreler düzgün olarak set edilmelidir, parametrelerin hazır değeri kullanılmamalıdır.

Aşağıdaki formül ile PARALLEL_MAX_SERVERS parametresinin hazır (default) değeri hesaplanır. [44]

$$(CPU_COUNT \times PARALLEL_THREADS_PER_CPU \times (2 \text{ if } PGA_AGGREGATE_TARGET > 0; \text{ otherwise } 1) \times 5)$$

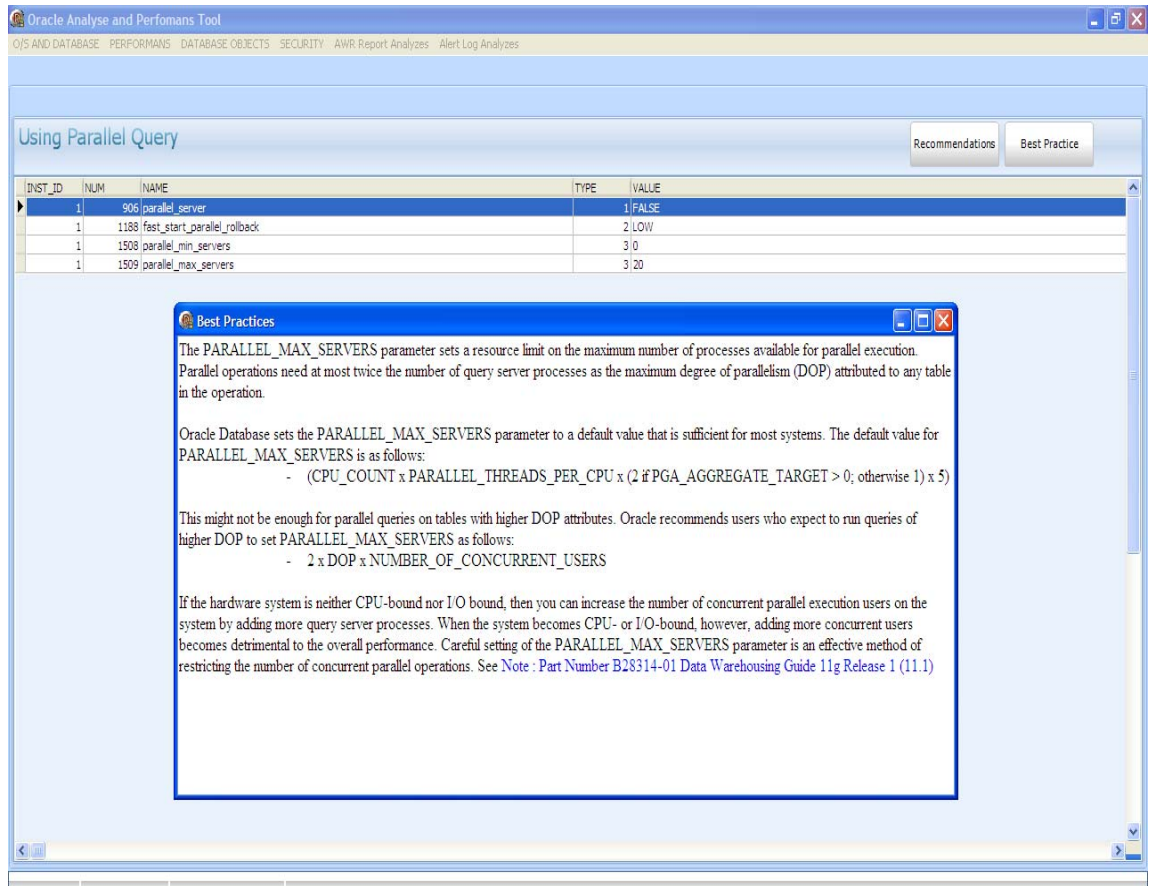
Parametrenin gerçek değeri aşağıdaki formül ile hesaplanarak belirlenir.

$$- 2 \times DOP \times NUMBER_OF_CONCURRENT_USERS$$

5.7.8.1 Tespit Edilen Sorunlar ve Çözüm Önerileri

Şekil 5.13’de veritabanı paralel sorgu ekranı gösterilmiştir. Bu örnekte veritabanındaki paralellik parametrelerinin sunucu üzerindeki kaynaklara göre doğru tanımlanmamış, hazır (default) değer kullanılmıştır. Bu bağlamda parametrenin gerçek değeri örnek uygulamadaki formüle göre hesaplanarak belirtilmelidir.

$$- 2 \times DOP \times \text{NUMBER_OF_CONCURRENT_USERS}$$



The screenshot shows the Oracle Analyze and Performans Tool interface. The main window displays a table of parallel query parameters. A 'Best Practices' dialog box is open, providing detailed information about the PARALLEL_MAX_SERVERS parameter and its recommended values based on system characteristics.

INST_ID	NUM	NAME	TYPE	VALUE
1	906	parallel_server		1 FALSE
1	1188	fast_start_parallel_rollback		2 LOW
1	1508	parallel_min_servers		3 0
1	1509	parallel_max_servers		3 20

Best Practices

The PARALLEL_MAX_SERVERS parameter sets a resource limit on the maximum number of processes available for parallel execution. Parallel operations need at most twice the number of query server processes as the maximum degree of parallelism (DOP) attributed to any table in the operation.

Oracle Database sets the PARALLEL_MAX_SERVERS parameter to a default value that is sufficient for most systems. The default value for PARALLEL_MAX_SERVERS is as follows:

- $(\text{CPU_COUNT} \times \text{PARALLEL_THREADS_PER_CPU} \times (2 \text{ if } \text{PGA_AGGREGATE_TARGET} > 0; \text{ otherwise } 1) \times 5)$

This might not be enough for parallel queries on tables with higher DOP attributes. Oracle recommends users who expect to run queries of higher DOP to set PARALLEL_MAX_SERVERS as follows:

- $2 \times \text{DOP} \times \text{NUMBER_OF_CONCURRENT_USERS}$

If the hardware system is neither CPU-bound nor I/O bound, then you can increase the number of concurrent parallel execution users on the system by adding more query server processes. When the system becomes CPU- or I/O-bound, however, adding more concurrent users becomes detrimental to the overall performance. Careful setting of the PARALLEL_MAX_SERVERS parameter is an effective method of restricting the number of concurrent parallel operations. See Note : Part Number B28314-01 Data Warehousing Guide 11g Release 1 (11.1)

Şekil 5.13. Paralel Sorgu Kullanım Analiz Ekranı

Paralellik Kullanıldığı Durumlar: Küçük tablo ve indekslerde paralellik kullanılmamalıdır. Paralellik genelde büyük tablo ve uzun süren sorgularda kullanılmalıdır. Normal bir veritabanında aşağıda belirtilen objelerin durumuna göre DOP (degree of parallelism) değeri tanımlanır.

- 200 MB'ın altındaki objelerde paralellik kullanılmamalıdır
- 200 MB ile 5 G arasında DOP değeri 4 olarak tanımlanır.
- 5 G üstündeki objelerde DOP değeri 32 olarak tanımlanır.

Yukarıdaki değerler iş ihtiyaçları ve aynı anda çalışan sorgular dikkate alarak değiştirilir. [35], [42],[43],[44]

5.7.9 Veritabanı Kilitleri

Veritabanı sunucusu üzerinden bulunan bellek (RAM), CPU, network, disk I/O gibi kaynakların kullanımı ve uygulamalardan çalışan SQL sorgularının çalışma planı (Execution Plane) iyi olmasına rağmen hala veritabanı uygulamalarında performans problemleri olabildiği durumlar ortaya çıkmaktadır.

Bu performans problemleri iki farklı şekilde kendini göstermektedir. Bunlar:

1. **Latch:** Oracle tarafında bellek kilitlerini yönetmek için kullanılan içsel mekanizmadır.
2. **Lock:** Veritabanı tarafında herhangi bir kullanıcı veriye erişip işlem yaparken diğer kullanıcıların erişimini engellemek için kullanılan kilit mekanizmasıdır.

Latch ve Lock Arasındaki Fark

- Latch çok sınırlayıcı bir mekanizma olup aynı anda bellek'te bir tek işlemin veriye erişimini sağlar. Lock ise aynı anda birden fazla prosesin (işlemin) veriye erişimine izin verir.
- Latch den veriye erişim için herhangi bir kuyruk mekanizması yoktur, lock (kilit) de veriye erişim için işlemler bir kuyruksa bekletilir sırası gelince işlem yaparlar.
- Latch işlemleri kısa olup mikro saniye sürer, lock ise çok uzun sürelidir.
- Latch veriye erişimin sınırlanmasını bellek üzerinde bir tek işlem yapar, transaction (sorgu işlemi) işlemi yapmaz. Lock ise tablo, indeks, veri blokları üzerinde sınırlama koyup sorgu işlemi bazında işlem yapmaktadır.
- Latch mekanizması işlem yaparken iki model kullanır; bellek üzerindeki blok müsait değilse beklemez ya da bakar blok boş değilse tekrar dener (willing-to-wait or no-wait). Lock ise 6 farklı model mekanizmasını kullanabilir. Bunlar çizelgede belirtilmiştir.

Oracle veritabanı birden fazla kullanıcıya aynı anda hizmet verdiği için veri bütünlüğü ve veri tutarlılığını kilitleme mekanizmalarıyla çözümlenmektedir. Tablo 5.10'da VT kilit tanımları gösterilmektedir.

Tablo 5.10. Oracle Veritabanı Kilit tanımları [14],[19],[16],[45],[46]

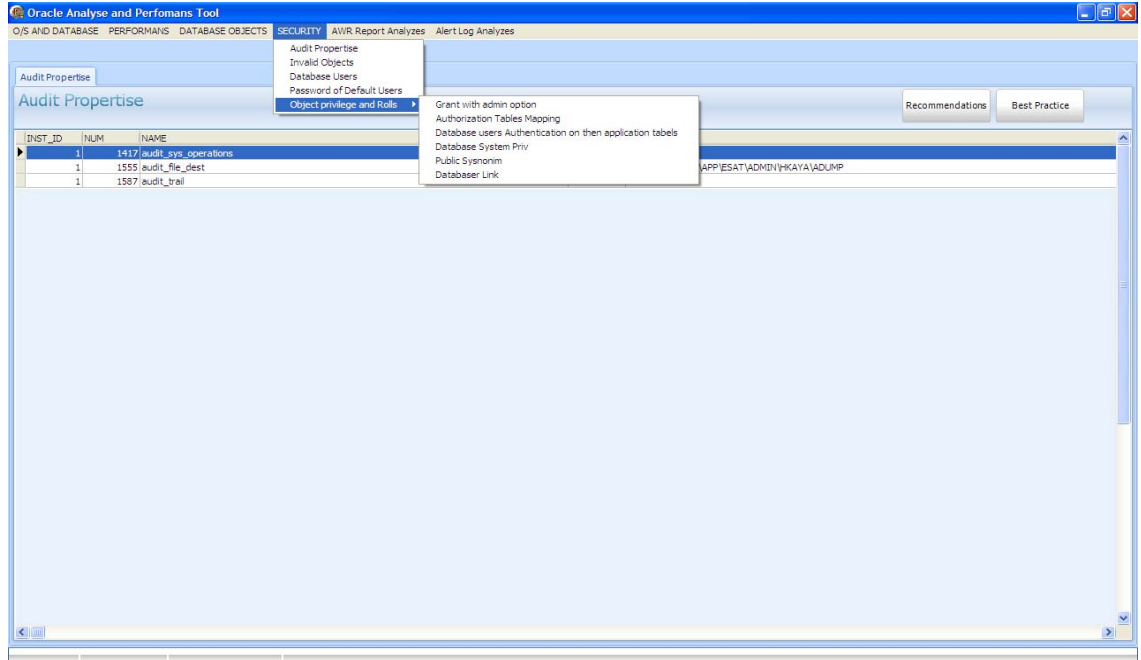
Mod	Tanım
1	Boş-Null (N)
2	Satır paylaşımı-Row Share(RS), ya da alt paylaşım kilidi (SS)
3	Satır ayrıcalıklı-Row Exclusive (RX), ya da alt ayrıcalıklı kilit(SX)
4	Paylaşım-Share(S)
5	Paylaşım ayrıcalıklı satır-Share Row Exclusive(SRX), ya da alt satır ayrıcalıklı kilit (SSX)
6	Ayrıcalıklı-Exclusive(X)

DAPT Oracle veritabanındaki kilitleme sorunlarını inceleyip hangi tür kilitleme var ise onları tespit ederek çözümlenme yöntemi sunmaktadır.

5.8 Veritabanı Güvenliği

Veritabanlarındaki güvenlik sorunları genellikle yazılım, donanım ve uygulama seviyelerinde gözlemlenmektedir. Oracle, tespit edilen güvenlik açıklıklarına yönelik Ocak, Nisan, Temmuz ve Ekim aylarında olmak üzere yılda dört kez kritik yama güncellemesi (Critical Patch Update - CPU) yayınlamaktadır. Yayımlanan yamaları düzenli olarak takip etmek için oracle web sitesinden (<http://www.oracle.com>) Oracle Technology Network → Security → Critical Patch Updates yolu takip edilmelidir. [47]

DAPT bu bölümde veritabanının güvenlik açıklıklarının tespit edilmesi ve çözüm öneriler yapılmaktadır. Şekil 5.14'de veritabanının güvenlik ana ekranı ve alt ekran görünümü gösterilmektedir.



Şekil 5.14 Veritabanı Güvenlik Ekranı

5.8.1 Güvenlik Ayarları

Veri güvenliğinin sağlanması ve veritabanından tanımlı olan kullanıcılarının yapmış olduğu işlemlerin izlenmesi için veritabanının güvenlik ayarları (database audit) yapılmalıdır. DAPT oracle veritabanının hazır (default) odit ayalarını (setlerini) kontrol eder ve tespit etmiş olduğu güvenlik açıklarıyla ilgili alternatif çözüm metotları sunmaktadır.

5.8.1.1 Tespit Edilen Sorunlar ve Çözüm Önerileri

Şekil 5.14’de ve Tablo 5.11’de veritabanı güvenlik ekranı ve güvenlik ayarları (database audit) parametreler gösterilmiştir. Bu örnekte oracle veritabanının hazır (default) odit ayalarının aktif edilmediği tespit edilmiştir.

Tablo 5.11 Veritabanı Odit Parametreleri

INST_ID	NUM	NAME	TYPE	VALUE	DESCRIPTION
1	1417	audit_sys_operations	1	FALSE	enable sys auditing
1	1555	audit_file_dest	2	D:\OracleR2\esat\hkayaladump	Directory in wich auditing files are to reside
1	1587	audit_trail	2	DB	enable system auditing

- Veritabanının oditleri aktif edilmelidir. Ancak odit işlemi veritabanı bazında full (tam) yapılmamalı bunun yerine belli kritik tablolar üzerinde odit tutulmalıdır.
- Veritabanının da DBA’lerin yapmış olduğu bütün işlemlerin oditi tutularak kayıt altına alınmalıdır.
- Oracle’ın SYS ve SYSTEM kullanıcılarının oditi tutularak yapmış olduğu işlemler kayıt altına alınmalıdır.
- Odit dosyası belli dönemlerde yedeği alınarak silinmelidir.

5.8.2 Geçersiz Nesnelere

Bir nesnenin yapısında deęişiklik yapıldığı zaman obje (nesne) geçersiz (invalid) durumuna düşer. Eğer bu nesneye tekrar erişilmek istenildiği zaman nesne recompiled (yeniden derlenmek) olarak kullanılabilir (valid) durumuna geçer. Şayet, nesne derlenmiyorsa kontrol edilerek hatanın düzeltilmesi ya da kullanılmıyorsa veritabanında silinmelidir.

5.8.2.1 Tespit Edilen Sorunlar ve Çözüm Önerileri

Tablo 5.12’de Oracle veritabanı üzerinde çalışan uygulamaların geçersiz nesnelere tespit edilerek nesnelere hangi uygulamaya ait oldukları, türü, yaratılma tarihi, en son güncelleme tarihi ve durum bilgilerini içeren rapor DAPT tarafından oluşturulmuştur.

Tablo 5.12. Geçersiz Nesnelere Gösteren Örnek Uygulama

Owner	Object Name	Object Type	Created	Last Ddl Time	Status
AKSUAVT	CDLOG_MAHALLE	TRIGGER	8 Haziran 2009 23:00	7 Eylül 2009 18:23	INVALID
AKSUAVT	CDLOG_KOY	TRIGGER	9 Haziran 2009 23:00	8 Eylül 2009 18:23	INVALID
AKSUAVT	CDLOG_ILCE	TRIGGER	10 Haziran 2009 23:00	9 Eylül 2009 18:23	INVALID
AKSUAVT	CDLOG_IL	TRIGGER	11 Haziran 2009 23:00	10 Eylül 2009 18:23	INVALID
AKSUAVT	CDLOG_CSBM	TRIGGER	12 Haziran 2009 23:00	11 Eylül 2009 18:23	INVALID
AKSUAVT	CDLOG_BUCAK	TRIGGER	13 Haziran 2009 23:00	12 Eylül 2009 18:23	INVALID
AKSUAVT	CDLOG_BINA	TRIGGER	14 Haziran 2009 23:00	13 Eylül 2009 18:23	INVALID
AKSUAVT	CDLOG_BAGIMSIZBOLUM	TRIGGER	15 Haziran 2009 23:00	14 Eylül 2009 18:23	INVALID

- Veritabanı düzenli olarak analiz edilmeli, geçersiz (invalid) nesne olmamalıdır.
- Veritabanında geçersiz olan nesnelere aktif edilmesi için aşağıdaki paket sistem (SYS) kullanıcısı tarafında yaratılarak çalıştırılmalıdır.

```

CREATE OR REPLACE PROCEDURE SYS.COMPILE_INVALID_OBJECTS
AS
    V_COUNTER          NUMBER (4) := 1;
    V_INVALID_COUNTER  NUMBER (4) := 0;
    V_REPEAT_COUNTER   NUMBER (2) := 0;
    V_REPEAT_LIMIT     NUMBER (2) := 10;
BEGIN
    WHILE V_COUNTER > 0 AND V_REPEAT_COUNTER <= V_REPEAT_LIMIT
    LOOP
        V_COUNTER := 0;
        V_INVALID_COUNTER := 0;

        FOR X
        IN ( SELECT 'ALTER '
              || CASE OBJECT_TYPE
                 WHEN 'PACKAGE BODY' THEN 'PACKAGE'
                 ELSE OBJECT_TYPE
              END
              || ' '
              || OWNER
              || '.'
              || OBJECT_NAME
              || ' COMPILE '
              || CASE OBJECT_TYPE
                 WHEN 'PACKAGE BODY' THEN ' BODY'
                 ELSE ''
              END
              || ' '
              || COMPILE_SCRIPT
        FROM DBA_OBJECTS
        WHERE STATUS = 'INVALID'
              AND OBJECT_TYPE IN
                 ('PACKAGE BODY',
                  'PACKAGE',
                  'FUNCTION',
                  'PROCEDURE',
                  'TRIGGER',
                  'VIEW')
        ORDER BY OBJECT_TYPE, OBJECT_NAME)
        LOOP
            BEGIN
                EXECUTE IMMEDIATE X.COMPILE_SCRIPT;

                V_COUNTER := V_COUNTER + 1;
            EXCEPTION
            WHEN OTHERS
            THEN
                V_INVALID_COUNTER := V_INVALID_COUNTER + 1;
            END;
        END LOOP;

        DBMS_OUTPUT.PUT_LINE (
            'Toplam ' || TO_CHAR (V_COUNTER) || ' obje compile edildi.'

```

```

);
V_REPEAT_COUNTER := V_REPEAT_COUNTER + 1;
END LOOP;

DBMS_OUTPUT.PUT_LINE (
  'Toplam ' || TO_CHAR (V_INVALID_COUNTER) || ' obje invalid.'
);

FOR X
IN ( SELECT OBJECT_TYPE || ': ' || OWNER || '.' || OBJECT_NAME
      COMPILE_SCRIPT
      FROM DBA_OBJECTS
      WHERE STATUS = 'INVALID'
      AND OBJECT_TYPE IN
          ('PACKAGE BODY',
           'PACKAGE',
           'FUNCTION',
           'PROCEDURE',
           'TRIGGER',
           'VIEW')
      ORDER BY OBJECT_TYPE, OBJECT_NAME)
LOOP
  DBMS_OUTPUT.PUT_LINE ('Invalid: ' || X.COMPILE_SCRIPT);
END LOOP;
END;
/

```

5.8.3 Veritabanı Kullanıcıları

Oracle veritabanının kurulumuyla beraber gelen sistem ve içsel (internal) kullanıcılar vardır. Bu kullanıcıların şifreleri hazır (default) şifre olup, çoğu oracle kullanıcısı tarafında bilinmektedir. Ayrıca bu kullanıcılar aktif kullanıcılardır. Dolayısıyla bu durum veritabanı güvenliği sorununu teşkil etmektedir. Kullanıcılarla ilgili bir diğer güvenlik açığı şifrelerinin basit verilmesi, belli bir yaşam süresinin tanımlanmamasıdır. Kullanıcıların veritabanında göre süresinin dolmasına rağmen yani görev değişikliği olması ya da çalışmış olduğu birimden ayrılması halinde hala kullanıcı veritabanında aktif olması ya da tanımlı olan bir kullanıcı veritabanında uzun süre işlem yapmaması

durumlar güvenlik sorunlarını doğurmaktadır. DAPT bu ve buna benzer kullanıcılarla ilgili güvenlik açıklarını tespit diyor ve farklı metotlarla çözüm önerileri yapmaktadır.

5.8.3.1 Tespit Edilen Sorunlar ve Çözüm Önerileri

Tablo 5.13’de Oracle veritabanı üzerinde tanımlı olan kullanıcıların yaratılış tarihi ve durum bilgilerini içeren rapor DAPT tarafından oluşturulmuştur. Bu örnek uygulama rapordan gözüktüğü gibi oracle’ın içsel (internal) kullanıcılar aktif, hazır (default) profile (görüntü) kullanılmakta ve kullanıcının yaratıldığı tarih ile zaman aşım süresi (expired date) arasında farklar gözükmemektedir. Dolayısıyla bütün bunlar bir güvenlik açığıdır.

Veritabanında kullanıcı tanımları, hangi durumlarda nasıl kullanılması gerektiği ve şifre tanımları aşağıdaki yöntemlere göre yapılmalıdır.

İçsel (internal) kullanıcılar: Oracle veritabanının oluşturulması ile birlikte gelen hazır (default) kullanıcılar olup, veritabanının içsel fonksiyonlarının bakımı amacıyla kullanılmaktadır. Bu kullanıcılar veritabanı yaşadığı süre içinde kilitli olmalıdır. Ancak ihtiyaç duyulduğu an kilit kaldırılmalıdır.

Sistem kullanıcıları (SYS ve SYSTEM): Veritabanı üzerinde her türlü işlem yapabilme yetkisine sahip olan bu kullanıcılar aşağıdaki durumlarda kullanılmalıdır.

- SYS kullanıcı ile veritabanına bağlanma durumu
 - Veritabanı yaratma (create) etmek,
 - Veritabanını kapatıp açma,
 - Veritabanı kurtarma (database recover),
 - Parametre kütüğü oluşturma (spfile, pfile),
 - Sistem tetikleyicisi (system trigger) oluşturma,

- Oracle'ın hata düzeltici kodlarını (built-in scripts) çalıştırma,
 - Standby veritabanını durum kontrolü,
 - Oracle'ın güncel paket'lerini yükleme (patch install),
 - Veritabanı güncelleme (database upgrade) işlemini yapma için SYS kullanıcısıyla işlem yapılır.
- SYSTEM kullanıcı ile veritabanına bağlanma durumu
 - Veritabanı canlı yedek (hot backup) ve resmini çekmek (snap backup) için,
 - Kontrol kütüğü (controlfile backup) yedeğini almak,
 - Hata izlemek (trace) için,
 - Tüm veritabanı verilerini dışarı alma (full export) işlemlerinde,
 - Veritabanı uygulama tablo ve indekslerin istatistiklerini toplamak için SYSTEM kullanıcısıyla işlem yapılır.
- Veritabanı yöneticisi kullanıcıları ile bağlanma durumu
 - SYS ve SYSTEM kullanıcılarının kullanımı dışındaki bütün veritabanı işlemlerinin yapılması durumunda
 - Oracle'ın parametre kütüklerinde değişiklik ihtiyacı olduğu durumlarda (bellek ve oracle anı iyileştirmeleri vb.) DBA yetkili kullanıcı ile işlem yapılır.
- Şema kullanıcıları (uygulama kullanıcıları) ile bağlanma durumu
 - Uygulama devreye alırken,
 - Uygulama ortamında ani çıkan problemlerde uygulamayı test yapma durumunda,
 - Şemanın kendi nesnelere görünümlere (view) yetki vermek veya yetki alma durumunda,

- Şema kullanıcısı ile birden fazla veritabanının konuşması için “db link” yaratma ya da silmek,
 - Şemanın kendi indekslerini izlemek için uygulama kullanıcılarıyla işlem yapılır.
- “Db Link” kullanıcıları ile veritabanına bağlanma durumu
 - Veritabanları arasında bilgi taşıma amacıyla kullanılmaktadır.

Tablo 5.13. Veritabanı Kullanıcıları Gösteren Örnek Uygulama

USERNAME	ACCOUNT STATUS	LOCK DATE	EXPIRY DATE	CREATED	PROFILE
OUTLN	EXPIRED & LOCKED	3/30/2009	3/30/2009	8/3/2007	DEFAULT
TSMSYS	EXPIRED & LOCKED	3/30/2009	3/30/2009	8/3/2007	DEFAULT
WMSYS	EXPIRED & LOCKED	3/30/2009	3/30/2009	8/3/2007	DEFAULT
EXFSYS	EXPIRED & LOCKED	3/30/2009	3/30/2009	8/3/2007	DEFAULT
CTXSYS	EXPIRED & LOCKED	3/30/2009	3/30/2009	8/3/2007	DEFAULT
XDB	EXPIRED & LOCKED	3/30/2009	3/30/2009	8/3/2007	DEFAULT
ANONYMOUS	EXPIRED & LOCKED	3/30/2009	3/30/2009	8/3/2007	DEFAULT
MDSYS	EXPIRED & LOCKED	3/30/2009	3/30/2009	8/3/2007	DEFAULT
ORDSYS	EXPIRED & LOCKED	3/30/2009	3/30/2009	8/3/2007	DEFAULT
ORDPLUGINS	EXPIRED & LOCKED	3/30/2009	3/30/2009	8/3/2007	DEFAULT
MDDATA	EXPIRED & LOCKED	3/30/2009	3/30/2009	8/3/2007	DEFAULT
WKPROXY	EXPIRED & LOCKED	3/30/2009	3/30/2009	8/3/2007	DEFAULT
WK_TEST	EXPIRED & LOCKED	3/30/2009	3/30/2009	8/3/2007	DEFAULT
FLows_FILES	EXPIRED & LOCKED	3/30/2009	3/30/2009	8/3/2007	DEFAULT
APEX_PUBLIC_USER	EXPIRED & LOCKED	3/30/2009	3/30/2009	8/3/2007	DEFAULT
FLows_030000	EXPIRED & LOCKED	3/30/2009	3/30/2009	8/3/2007	DEFAULT
OWBSYS	EXPIRED & LOCKED	3/30/2009	3/30/2009	8/3/2007	DEFAULT
BI	EXPIRED & LOCKED	3/30/2009	3/30/2009	3/30/2009	DEFAULT
PM	EXPIRED & LOCKED	3/30/2009	3/30/2009	3/30/2009	DEFAULT
OE	EXPIRED & LOCKED	3/30/2009	3/30/2009	3/30/2009	DEFAULT
SH	EXPIRED & LOCKED	3/30/2009	3/30/2009	3/30/2009	DEFAULT
HR	EXPIRED & LOCKED	3/30/2009	3/30/2009	3/30/2009	DEFAULT
SCOTT	OPEN		3/30/2009	8/3/2007	DEFAULT
MGMT_VIEW	OPEN		9/26/2009	8/3/2007	DEFAULT
SYSMAN	OPEN		10/4/2009	8/3/2007	DEFAULT
WKSYS	OPEN		3/30/2009	3/30/2009	8/3/2007
DWH_READ	EXPIRED		4/26/2010	4/1/2009	DEFAULT
SPATIAL	OPEN		2/27/2011	4/1/2009	DEFAULT
HALILKAYA	OPEN		2/28/2011	9/1/2010	DEFAULT
YEDEK	OPEN		3/5/2011	7/24/2009	DEFAULT

Profil tanımları: Oracle veritabanı güvenliğini sağlayabilmek için kullanıcılara değişik amaçlarla profil yaratılmalı ve kullanıcılara bu profil'e atanmalıdır. Veritabanında tanımlı olan kullanıcıların şifreleri ITPM standartlarına göre verilmeli, şifrelerin yaşam süreleri ve şifre verme kısıtları profil'de tanımlı olan SYS.VERIFY_FUNCTION ile yapılmalıdır. Örnek profil tanımları aşağıdaki şekilde yapılabilir.

FOTO : Son kullanıcıları,
 ENDUSER : Son kullanıcılar,
 SCHEMA : Sema kullanıcıları,
 ORAUSER : Veritabanı kullanıcıları,
 DBA : Veritabanı yöneten kullanıcıları,
 SELLING : Satış departmanı son kullanıcıları,
 ITUSER : Bilgi işlem kullanıcıları ifade edern örnek profil tanımlarıdır.

Tablo 5.14'den örnek veritabanı kullanıcı profil tanımlama gösterilmektedir.

Tablo 5.14. Örnek Veritabanı Kullanıcı Profil Tanımlama

	Aylak Zaman (dk.)	Eş zamanlı bağlantı sayısı	Şifrenin yaşam süresi (gün)	Bilgi indirme süresi (gün)	Şifre tarihçesi (sayı)	Şifre kriterlerini sağlayan kod	Kilit (sayı)	Kilit sonrası aktif olma süresi(saat)
FOTO	5	2	90	3	10	VERIFY_FUNCTION	5	1
ENDUSER	5	2	90	3	10	VERIFY_FUNCTION	5	1
SCHEMA	-	-	365	30	10	VERIFY_FUNCTION	-	-
ORAUSER	-	-	365	30	10	VERIFY_FUNCTION	-	-
DBA	-	5	30	3	10	VERIFY_FUNCTION	5	24
SELLING	5	2	90	3	10	VERIFY_FUNCTION	5	1
ITUSER	5	2	90	3	10	VERIFY_FUNCTION	5	1
DEFAULT	-	-	-	-	-	-	-	-

Kullanıcı şifresi tanımlarken aşağıdaki şekilde şifre kısıtlarını belirten fonksiyon yazılabilir.

```

CREATE OR REPLACE FUNCTION SYS.verify_function
(username          VARCHAR2,
PASSWORD          VARCHAR2,
old_password      VARCHAR2)

RETURN BOOLEAN
IS
digitarray        VARCHAR2 (20);
chararray         VARCHAR2 (52);
BEGIN
digitarray := '0123456789';
chararray := 'abcdefghijklmnopqrstuvwxyzaBCDEFGHIJKLMNOPQRSTUVWXYZ';

IF NLS_LOWER (PASSWORD) = NLS_LOWER (username)
THEN
raise_application_error (-20001, 'Sifre kullanıcı adı ile aynı
olamaz');
END IF;

IF LENGTH (PASSWORD) < 6
THEN
raise_application_error (
-20002,
'Sifrenizin uzunluğu en az 6 karakter olmalıdır'
);
END IF;

RETURN (TRUE);
END;
/

```

Uygulama veritabanında bazı kullanıcı ve veritabanında işlem yapmak için kullanılan uygulamaların veritabanı bağlantısını engellemek için aşağıdaki tetikleyiciyi (trigger) yazılmalıdır.

```

CREATE OR REPLACE TRIGGER SYS.BAN_TOOLS
AFTER LOGON
ON DATABASE
DECLARE

CURSOR user_prog
IS
SELECT  program
FROM    sys.V_$SESSION
WHERE   auid = SYS_CONTEXT ('USERENV', 'SESSIONID');

--Assign the cursor to a PL/SQL record.
user_rec  user_prog%ROWTYPE;

```

```

counter    NUMBER (2) := 0;
BEGIN
  SELECT    COUNT ( * )
           INTO    counter
           FROM    DBA_USERS
           WHERE   PROFILE NOT IN ( 'DBA', 'ORAUSER', 'SCHEMA', 'ITUSER',
'FOTO' )
           AND    username = USER;

  OPEN user_prog;

  FETCH user_prog INTO    user_rec;

  IF user_rec.program IN
    ( 'sqlplusw.exe', 'TOAD.exe', 'SQLNav5.exe', 'sqlplus.exe' )
    AND counter > 0
  THEN
    RAISE_APPLICATION_ERROR (-20001, 'You are not allowed to login');
  END IF;

  CLOSE user_prog;
END;
/

```

5.8.4 “With Admin” Seçeneği ile Verilen Yetkiler

Veritabanı nesnelere üzerinde yetkisi verilirken “With Admin=Yes” seçeneğinin kullanılması yetkiyi alan kullanıcının bir başka kullanıcıya ilgili yetkiyi vermesine sebep olur. Bu bağlamda “With Admin” seçeneği ile verilen yetkilere dikkat edilmesi gerekir.

Tablo 5.15’de “With Admin=Yes” seçeneği ile verilen bir örnek gösterilmektedir.

Tablo 5.15. Örnek With Admin Seçeneğiyle Verilen Yetkiler

Grantee	Privilege	Admin Option
FLows_030000	CREATE ANY CONTEXT	YES
FLows_030000	CREATE CLUSTER	YES
FLows_030000	CREATE DIMENSION	YES
FLows_030000	CREATE INDEXTYPE	YES
FLows_030000	CREATE JOB	YES
FLows_030000	CREATE MATERIALIZED VIEW	YES
FLows_030000	CREATE OPERATOR	YES
FLows_030000	CREATE PROCEDURE	YES
FLows_030000	CREATE SEQUENCE	YES
FLows_030000	CREATE SYNONYM	YES
FLows_030000	CREATE TABLE	YES
FLows_030000	CREATE TRIGGER	YES
FLows_030000	CREATE TYPE	YES
FLows_030000	CREATE VIEW	YES
FLows_030000	UNLIMITED TABLESPACE	YES
SCHEDULER_ADMIN	CREATE ANY JOB	YES
SCHEDULER_ADMIN	CREATE EXTERNAL JOB	YES
SCHEDULER_ADMIN	CREATE JOB	YES
SCHEDULER_ADMIN	EXECUTE ANY CLASS	YES
SCHEDULER_ADMIN	EXECUTE ANY PROGRAM	YES
SCHEDULER_ADMIN	MANAGE SCHEDULER	YES
SPATIAL	UNLIMITED TABLESPACE	YES
WMSYS	SELECT ANY DICTIONARY	YES

5.8.5 Veritabanı Sistem Yetkileri

Veri ve nesnelerin güvenliğinin sağlanması için sistem yetkilerinin (system prive) nesnelere üzerindeki DDL yetkileri yalnızca belli kullanıcılara tarafında yapılmalıdır. Genelde bu DDL işlemleri DBA ve uygulama kullanıcıları tarafında yapılır.

DAPT veritabanı kullanıcılarının hangi sistem yetkilerine (DDL yetkilerine) sahibi olduğu gösterilmektedir. Liste kontrol edilerek veritabanının güvenliğini teşkil edecek yetkiler ilgili kullanıcılardan geri alınmalıdır.

Şekil 5.15’de veritabanı kullanıcılarına verilen sistem yetkileri ekranı görülmektedir. Bu ekranda iki bölümden oluşmaktadır. Üsteki bölüm veritabanının içsel kullanıcıları hariç yani veritabanı kurulumundan sonra yaratılan uygulama kullanıcıları listesi ve aldıkları bölüm ise üstten seçilen kullanıcıya verilen sistem yetkileri belirtilmektedir. Her iki rapor listesi sağ kılık fare (mause) ile “Excel” dosyasına kayıt yapılmaktadır.

The screenshot shows the Oracle Enterprise Manager interface. The title bar reads 'Oracle Analyse and Performans Tool'. The main window is titled 'Database System Priv' and contains two sections:

Database System Priv (Top Section):

Grantee	Privilege	With Admin Option
DAPT	UNLIMITED TABLESPACE	NO
▶ HALIL	UNLIMITED TABLESPACE	NO
OPSU	UNLIMITED TABLESPACE	NO

USER PRIVS (Bottom Section):

HALIL
DBA
ADMINISTER ANY SQL TUNING SET
ADMINISTER DATABASE TRIGGER
ADMINISTER RESOURCE MANAGER
▶ ADMINISTER SQL MANAGEMENT OBJECT
ADMINISTER SQL TUNING SET
ADVISOR
ALTER ANY ASSEMBLY
ALTER ANY CLUSTER
ALTER ANY CUBE
ALTER ANY CUBE DIMENSION
ALTER ANY DIMENSION
ALTER ANY EDITION
ALTER ANY EVALUATION CONTEXT
ALTER ANY INDEX
ALTER ANY INDEXTYPE
ALTER ANY LIBRARY
ALTER ANY MATERIALIZED VIEW
ALTER ANY MINING MODEL
ALTER ANY OPERATOR
ALTER ANY OUTLINE
ALTER ANY PROCEDURE
ALTER ANY ROLE
ALTER ANY RULE
ALTER ANY RULE SET

A 'SaveToExcel' button is visible next to the selected 'ADMINISTER SQL MANAGEMENT OBJECT' privilege.

Şekil 5.15 Uygulama Kullanıcılarına Verilen Sistem Yetki Analiz Ekranı

5.8.6 Kullanıcılarının Uygulama Tabloları Üzerindeki Yetkileri

Veri güvenliği sağlanması kurum için önem teşkil etmektedir. Bu bağlamda veritabanındaki tanımlı olan kullanıcılara uygulama tabloları üzerinde yetki verilirken dikkat edilmeli ve yetkilendirme 5.8.6.1’de belirtilen örnek uygulamaya benzer belli bir prosedür’e göre verilmelidir.

DAPT veritabanı üzerinde çalışan bütün uygulamaların rol ve tablo bazında detay yetkilerinin haritasını çıkartıyor ve alınması gereken önlemleri belirtmektedir.

5.8.6.1 Tespit Edilen Sorunlar ve Çözüm Önerileri

DAPT veritabanı kullanıcılarının uygulamalara ait olan tablolar üzerindeki veri ekleme, veri silme, veri güncelleme, veri değiştirme (insert, delete, update, alter) gibi DML yetkilerinin raporu çıkarmaktadır. Tablo 5.16’da örnek uygulamaya ait tablolar üzerindeki yetkilendirme raporunda gözüktüğü gibi yetkiler rol’lere değil direkt kullanıcılara atanmıştır. Bu yöntemle veritabanında güvenlik bakım ve kontrolleri karmaşık olması nedeniyle sağlıklı olarak yapamamakta, güvenlik açığı oluşabilmektedir.

Tablo 5.16. Uygulama Tabloları Üzerinde Yetkileri Gösteren Örnek Uygulama

Owner	Table Name	Privileges
CW	CWDOCSEQ	ALTER
CW	SI_ACCESS_PORT_CONFIG	DELETE
CW	SI_CUSTOMER_SERVICE_ADDRESS	UPDATE
CW	SI_DISCOUNT_AGREEMENT	DELETE
CW	SI_DISCOUNT_AGREEMENT	INSERT
CW	TT_DELIVERY	ALTER
CW	TT_DELIVERY	DELETE
CW	TT_DELIVERY	INSERT
CW	TT_DELIVERY	UPDATE
CW	TT_MAINTENANCE	ALTER
CW	TT_MAINTENANCE	DELETE
CW	TT_MAINTENANCE	INSERT
CW	TT_MAINTENANCE	UPDATE
IPEX	DB_REV_INFO	DELETE
IPEX	DB_REV_INFO	INSERT
IPEX	DB_REV_INFO	UPDATE
IPEX	DETAIL_TREE_ORDER	DELETE
TT_R1	PORT_PAIR_MAP	INSERT
TT_R1	RESOURCE_INST	INSERT
TT_R1	RESOURCE_INST	UPDATE
TT_R1	RESOURCE_TEMPLATE_INST	UPDATE
TT_R1	RESOURCE_TPLT_CAPACITY_INST	DELETE
TT_R1	RESOURCE_TPLT_CAPACITY_INST	INSERT
TT_R1	RESOURCE_TPLT_CAPACITY_INST	UPDATE
TT_R1	RESOURCE_TPLT_DOMAIN_MAP	DELETE

Ana uygulama veritabanında nesnelere üzerindeki yetkiler kullanıcılara direkt atanmamalı, her uygulamaya ait rol'ler yaratılarak yetkiler rol'lere, rol'ler de kullanıcılara atanmalıdır. Ancak görünüm'ler (view) rol'lere atanamamasından dolayı ve bazı istisnai durumlarda (prosedür içinde kullanılan bir tablonun sahibinin farklı olması durumu) yetkiler kullanıcılara direkt atanabilir.

Veritabanındaki her kullanıcı belli kurallar çerçevesinde yaratılıp, şifrelerinin yaşam süreleri ve şifre verme özellikleri profil'ler atanarak yapılmalıdır.

Kullanıcıların veritabanındaki faaliyet alanı ise veritabanındaki kullanıcılara atanan rol'ler aracılığıyla yapılmalıdır.

Örnek uygulamalar: Oracle veritabanında beş farklı rol'le tanımlı bulunmakta olup, oracle rol'leri hariç diğer rol'lerin içerisinde DDL yetkileri bulunmamalıdır.

Oracle veritabanında hali hazırda olan kendi rol'leri haricinde, sonradan yaratılan rol'lerin güvenliğinin sağlanması için şifre verilmelidir.

OWNER_[şema_user] rol'ü yalnız şema kullanıcılarına atanır.

1. Oracle rol'leri: Veritabanının yaratılmasıyla birlikte gelen hazır rol'ler olup, sistem nesnelere ve veri sözlüklerinin (dictionary view) kullanımı, veritabanı yaratma ve veritabanı nesnelere oluşturma izin verilen yetkiler bu role atanmıştır. [48], [49]
2. Son Kullanıcı (Enduser Role) : Veritabanında tanımlı olan kullanıcıların veritabanına bağlanması ve kullanıcıların şifrelerini kendilerinin değiştirebilmesine izin veren yetkiler bu role atanmalıdır.
3. Belirli uygulamaların kullanılmasına izin verilen rol'ler: Veritabanındaki bazı özel uygulamaların kullanımına izin veren bölüm bazında tanımlanan özel rol'lerdir.
Örneğin; Satış bölümü için SELLING_ROLE, teknik dokümanlar için TECNICDOC_ROLE vb.
4. Uygulamaya özgü standart rol'ler: Veritabanında her şema'nın kendi standart rolü olup DDL ve DML işlemleri için hazırlanan ve şema ile ilişkilendirilen rol'lerdir.

Bu rol'lerin tanımlanması aşağıdaki şekilde yapılır:

OWNER_[şema_user] : Şemanın bütün nesnelere üzerinde veri okuma, veri ekleme, veri güncelleme, veri silme ve çalıştırma (select, delete, insert, update, execute) yetkilerini içerir.

SELECT_[şema_user] : Şemanın bütün nesneleri üzerinde yalnız okuma yetkisi içerir.

DML_[şema_user] : Şemanın bütün nesneleri üzerinde veri ekleme, veri güncelleme ve veri silme (insert, delete, update) yetkilerini içerir.

EXECUTE_[şema_user]: Şemanın bütün objelerini (procedure, function, trigger vb.) çalıştırma yetkilerini içerir.

ENDUSER: Veritabanına bağlanma (login olma) ve kullanıcı şifresini değiştirebilme yetkilerini içerir.

İşlemci - Sunucu (Client-Server) mimarisindeki uygulamalar için yetkilendirme ekran bazında yapılmalıdır.

5.9 Oracle Performans Analiz Raporu

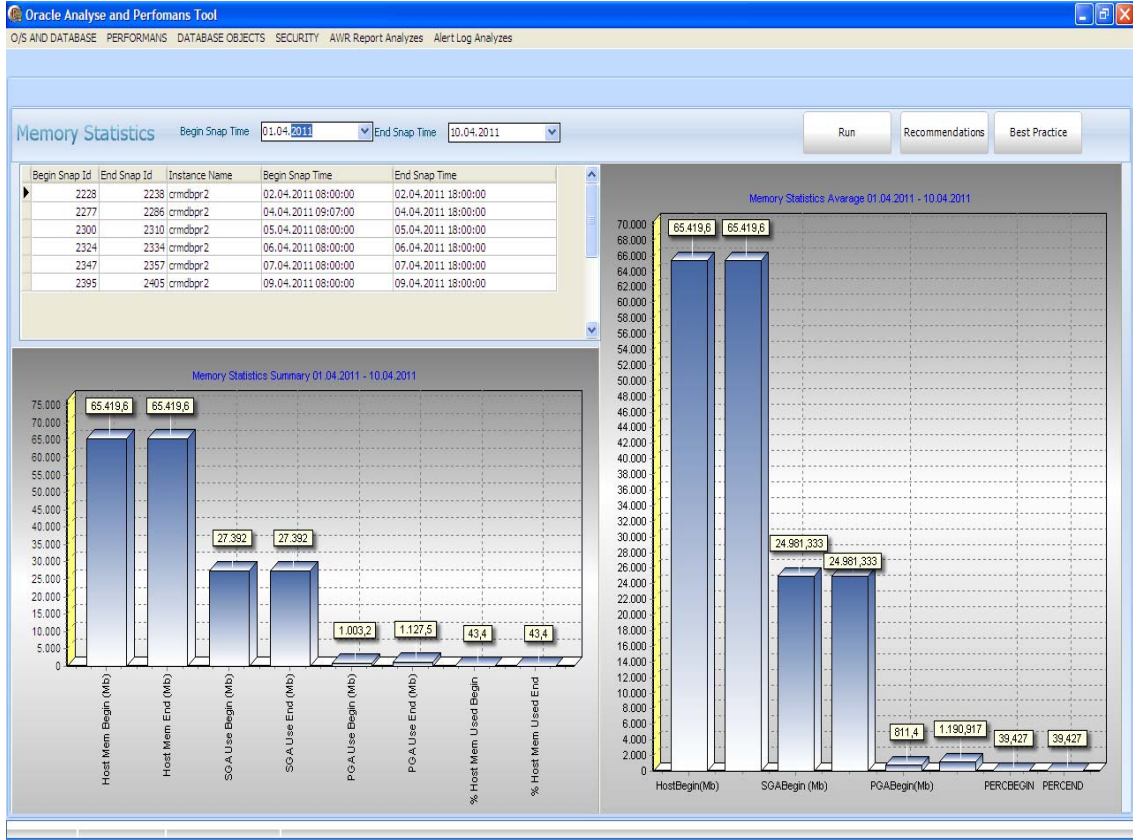
Bu bölümde Oracle içsel (internal) paketleri kullanılarak istatistiksel verilerden oluşan SP raporunun analizi yapılmaktadır. Oracle9i öncesi sürümlerden SP, yeni sürümlerden AWR raporu olarak adlandırılan bu rapor veritabanında oluşan problemlerin anlaşılmasına yardımcı olunmaktadır. Raporun istatistiksel verilerden oluşması ve çok karmaşık bir yapısı olması açısından analiz işlemi zordur. Günümüzden yeterli teknik bilgiye sahip DBA sayısı az olması nedeniyle çoğu oracle kullanıcısı olan firmalar oracle ya da oracle çözüm ortaklarından teknik danışmanlık almaktadırlar. Tez çalışması

içerisinde SP raporu analiz edilerek veritabanı problemleri tespiti ve çözüm önerileri yapılmaktadır.

5.9.1 Bellek Kullanım Analizi

Verilen tarih ya da iki tarih arasında veritabanının bellek kullanımını istatistiklerini analiz edere, SGA ve PGA'nın kullanım dağılımını Mb belirtmektedir. Şekil 5.16'da veritabanı üzerimde çalışan uygulamaların belirtilen tarih ya da tarihler arası bellek kullanım dağılımı belirten ekran gösterilmektedir. Sol taraftaki ekran bir tek SP raporunu bellek analiz sonucunu grafiğini gösterir. Sağ taraftaki ise belirtilen iki tarih arasındaki bütün SP raporlarının bellek kullanım bilgilerinin analizi sonucunu göstermektedir.

Şekil 5.16'daki ekran örneğinden SGA'nın kullanımında herhangi bir değişiklik gözlenmezken PGA'dan az da olsa farklılıklar gözükmektedir.



Şekil 5.16. Veritabanı Bellek Kullanımını Analiz Ekranı

5.9.2 Merkezi İşlem Birimi Maliyet Analizi

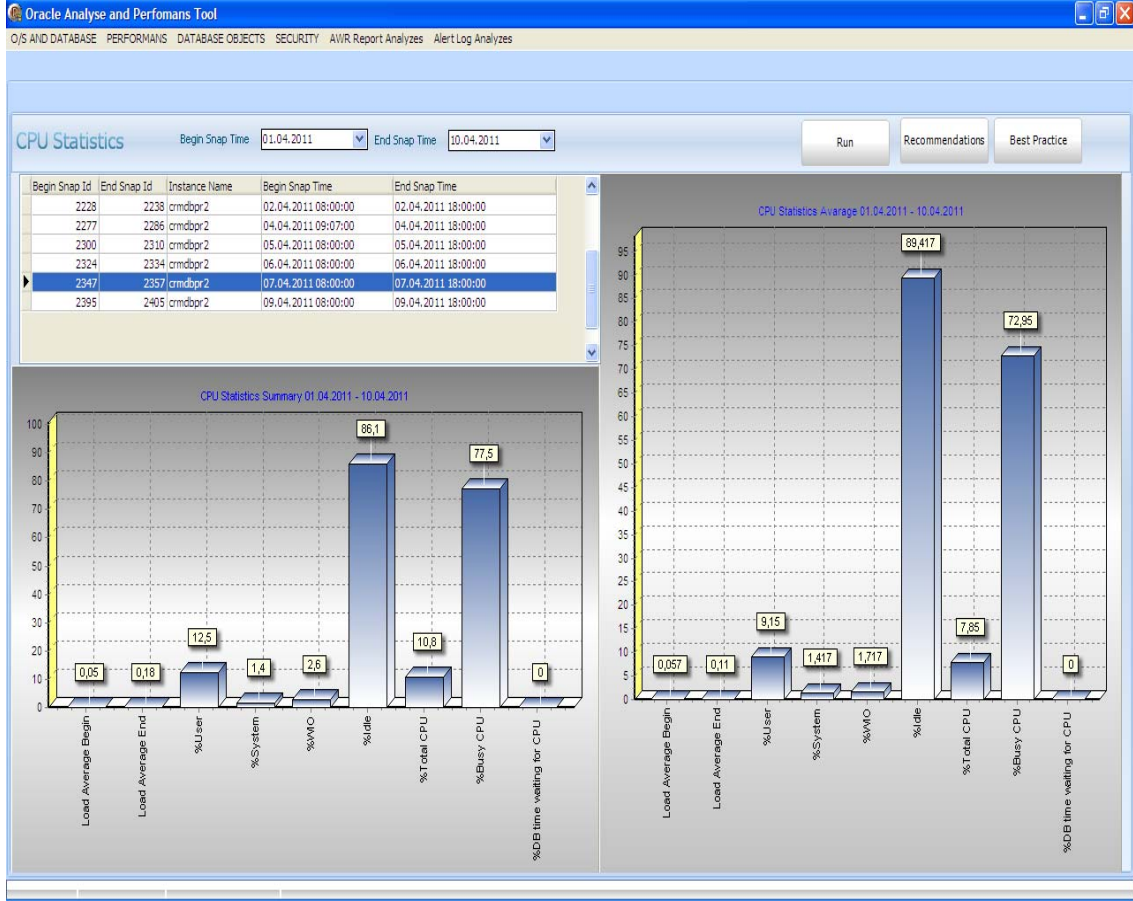
Verilen tarih ya da tarihler arasında veritabanı uygulamasının CPU kullanımını analiz ediyor ve CPU üzerindeki yük dağılımı grafiksel olarak belirtilmektedir. Şekil 5.17'deki örnekten sol taraftaki grafik belirtilen yalnız bir SP raporundaki CPU kullanım dağılım grafiği gösterilirken, sağ taraftaki grafik belirtilen iki tarih arasındaki bütün SP raporlarının CPU kullanım dağılımı gösterilmektedir.

5.9.2.1 Tespit Edilen Sorunlar ve Çözüm Önerileri

Crdbpr2 instance'ına ait örnekten 01.04.2011 ile 10.04.2011 tarihleri arasında CPU'nun %89,417 boş iken %72,95' i kullanılmaktadır. Bu durum da bize belirtilen tarihler arasından veritabanında CPU kullanımıyla ilgili herhangi bir sorun olmadığını belirtmektedir.

Sol taraftaki ekranda ise 07.04.2011 08.00 ile 07.04.2011 18.00 tarihleri arasında CPU'nun %77,5 kullanılmış, %86,1'i boş gözükmektedir. Bu da bize bu tarihler arasında veritabanında az da olsa bir CPU kullanım sorunu olduğunu göstermektedir.

DAPT yazılımın bu sorunun hangi SQL cümlesinin sebep olduğunu diğer ekranlardan (SQL kullanım ile ilgili ekranlar) belirtmektedir.



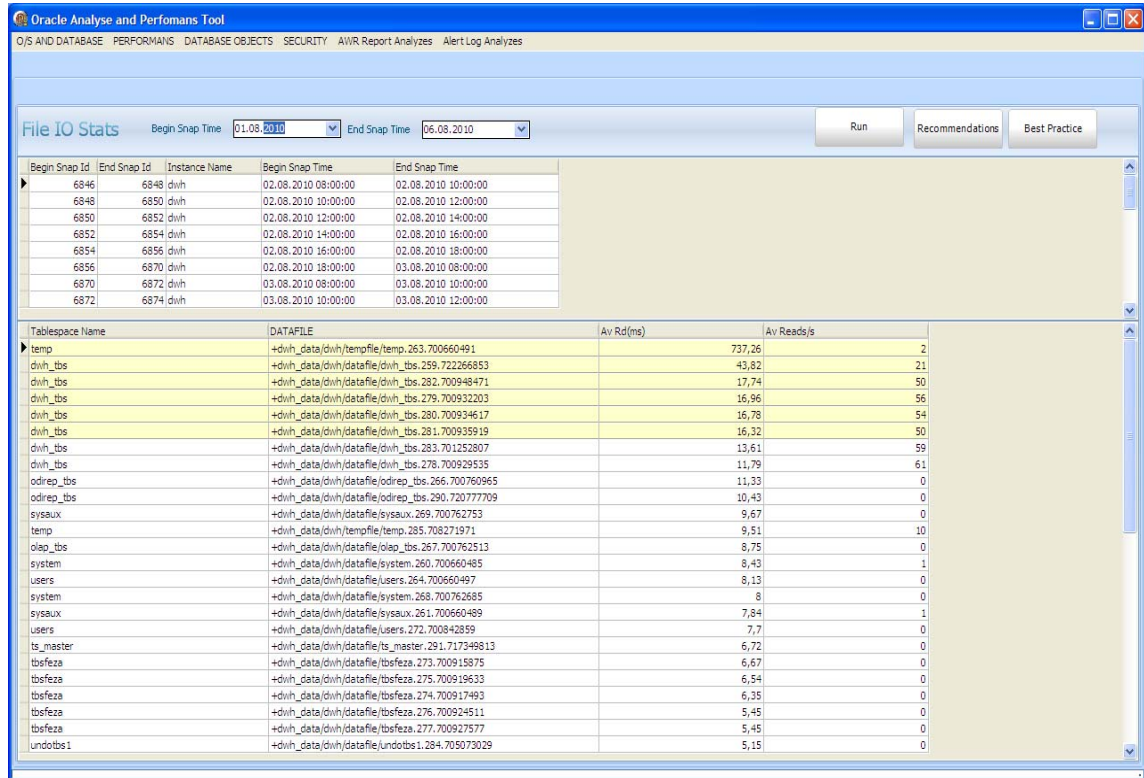
Şekil 5.17. Merkezi İşlem Birimi Maliyet Analizini Gösteren Ekran

5.9.3 Girdi Çıktı Analizi

Verilen tarihler arasında ilgili SP raporunu analiz edilir ve ekranda o tarihle ilgili SP raporundaki tablo uzayı ve veri kütüklerinin (tablespace and datafile) I/O durumlarını göstermektedir. Ekranda I/O problemi olan tablo uzayı ve veri kütükleri tespit edilerek gri renkli olarak tanımlanır ve çözüm önerisi yapar.

5.9.3.1 Tespit Edilen Sorunlar ve Çözüm Önerileri

Şekil 5.18’de veri kütüklerindeki I/O durumunu göstermektedir. Bu örnekten I/O sorunu olan veri kütükleri tespit edilmiş ve gri renkli olarak belirtilmiştir.



Begin Snap Id	End Snap Id	Instance Name	Begin Snap Time	End Snap Time
6846	6848	dwh	02.08.2010 08:00:00	02.08.2010 10:00:00
6848	6850	dwh	02.08.2010 10:00:00	02.08.2010 12:00:00
6850	6852	dwh	02.08.2010 12:00:00	02.08.2010 14:00:00
6852	6854	dwh	02.08.2010 14:00:00	02.08.2010 16:00:00
6854	6856	dwh	02.08.2010 16:00:00	02.08.2010 18:00:00
6856	6870	dwh	02.08.2010 18:00:00	03.08.2010 08:00:00
6870	6872	dwh	03.08.2010 08:00:00	03.08.2010 10:00:00
6872	6874	dwh	03.08.2010 10:00:00	03.08.2010 12:00:00

Tablespace Name	DATAFILE	Av Rd(ms)	Av Reads/s
temp	+dwh_data/dwh/temp/263.700660491	737,26	2
dwh_tbs	+dwh_data/dwh/datafile/dwh_tbs.259.722266853	43,82	21
dwh_tbs	+dwh_data/dwh/datafile/dwh_tbs.282.700948471	17,74	50
dwh_tbs	+dwh_data/dwh/datafile/dwh_tbs.279.700932203	16,96	56
dwh_tbs	+dwh_data/dwh/datafile/dwh_tbs.280.700934617	16,78	54
dwh_tbs	+dwh_data/dwh/datafile/dwh_tbs.281.700935919	16,32	50
dwh_tbs	+dwh_data/dwh/datafile/dwh_tbs.283.701252807	13,61	59
dwh_tbs	+dwh_data/dwh/datafile/dwh_tbs.278.700929535	11,79	61
odirep_tbs	+dwh_data/dwh/datafile/odirep_tbs.266.700760965	11,33	0
odirep_tbs	+dwh_data/dwh/datafile/odirep_tbs.290.720777709	10,43	0
sysaux	+dwh_data/dwh/datafile/sysaux.269.700762753	9,67	0
temp	+dwh_data/dwh/temp/265.708271971	9,51	10
olap_tbs	+dwh_data/dwh/datafile/olap_tbs.267.700762513	8,75	0
system	+dwh_data/dwh/datafile/system.260.700660485	8,43	1
users	+dwh_data/dwh/datafile/users.264.700660497	8,13	0
system	+dwh_data/dwh/datafile/system.268.700762685	8	0
sysaux	+dwh_data/dwh/datafile/sysaux.261.700660489	7,84	1
users	+dwh_data/dwh/datafile/users.272.700842859	7,7	0
ts_master	+dwh_data/dwh/datafile/ts_master.291.717349813	6,72	0
tsfeza	+dwh_data/dwh/datafile/tsfeza.273.700915875	6,67	0
tsfeza	+dwh_data/dwh/datafile/tsfeza.275.700919633	6,54	0
tsfeza	+dwh_data/dwh/datafile/tsfeza.274.700917493	6,35	0
tsfeza	+dwh_data/dwh/datafile/tsfeza.276.700924511	5,45	0
tsfeza	+dwh_data/dwh/datafile/tsfeza.277.700927577	5,45	0
undotbs1	+dwh_data/dwh/datafile/undotbs1.284.705073029	5,15	0

Şekil 5.18. Veri Kütüklerindeki I/O Analizini Gösteren Ekran

Yukarıdaki örnekten “Av Rd (ms)” gösteren sütunun değeri 10’u geçmemelidir. İdeal değer 3–5 olmalıdır. Bunun birkaç farklı sebebi vardır:

- Veri kütüklerinin bulunduğu diskler yavaş olması,
- Tablo ve indekslerin aynı veri kütüklerden saklanması,
- Bu veri kütükleri üzerinde maliyetli sorguların yapılıyor olması,
- Veri depolama ünitesinin (storage) disk bağlantılarından problem olması,

- o Oracle veritabanının kullandığı bellek yetersiz olduğu benzer durumlarında problem gözükmemektedir.

5.9.4 Maliyetli SQL Sorguları

Veritabanında kullanılan uygulamaların performans sorununa sebep olan SQL, PL/SQL kodlarının tespiti yapılmaktadır. Bu kodların sistemden ne kadar kaynak kullandıkları detay olarak gösterilmektedir. Ayrıca aynı SQL, PL/SQL kodunun belirtilen zaman aralığında toplam çalışma sayısı ve sorgunun hangi modülden (web, toad, sqlplus vb.) çalıştırıldığı belirtilmektedir.

5.9.4.1 Tespit Edilen Sorunlar ve Çözüm Önerileri

Şekil 5.19’da veritabanı uygulamalarında maliyetli SQL sorguları gösterilmektedir. Bu örnekten dwh oracle anının (Oracle anı) ait 01.08.2010 ile 06.08.2010 tarihleri arası 2 saatlik dönemlerle alınan AWR raporlarının analizi edildiğinde maliyetli sorguların ve kullandıkları kaynaklar gözükmemektedir.

Sorunun giderilmesi için maliyetli sorgu aşağıda belirtilen yöntemle “trace’i” alınarak iyileştirme yapılmalıdır.

10046 trace aşağıdaki şekilde alınır.

```
alter session set timed_statistics=true;
alter session set max_dump_file_size=UNLIMITED;
alter session set statistics_level=all;
alter session set tracefile_identifier=xxx_1;
```


alter session set events '10046 trace name context forever, level 8';

— run the statement(s) to be traced---

alter session set events '10046 trace name context off';

The screenshot shows the Oracle SQL Developer interface. The main window is titled "SQL ordered by Buffer Gets" and displays a table of SQL statements. The table has columns: Begin Snap Id, End Snap Id, Instance Name, Begin Snap Time, End Snap Time, Count, Buffer Gets, Total%, CPU Time, Elapsed Time, SQL Id, SQL Module, and SQL Text. A modal window titled "SQL Text" is open, showing the full text of a complex SQL query. The query is a SELECT statement with a complex WHERE clause involving multiple joins and subqueries. The SQL Text window is titled "SQL Text" and contains the following text:

```
select distinct t762180.apt_adi as c1,
t762018.dvnh_mahalle_kodu as c2,
t762018.dvnh_sokak_kodu as c3,
t762018.sokak_kodu as c4,
t761693.tc_kimlik_no as c5,
t761693.tahsilat_yapamaz as c6,
t761693.sozdurum_kodu as c7,
t761927.ipital_kodu as c8,
t761927.poz_acklamasi as c9,
t761583.dvnh_sayac_capi as c10,
t761583.kullanilabilir as c11,
t761583.montaj_bedei as c12,
t761864.doviz_kodu as c13
from
dvnh_comit032 t762018 /* dim_comit032 */ ,
dvnh_comit033 t762180 /* dim_comit033 */ ,
dvnh_mv_comit020 t761693 /* dim_mv_comit020 */ ,
dvnh_comit030 t761907 /* dim_comit030 */ ,
dvnh_iemlt051 t761864 /* dim_iemlt051_2 */ ,
dvnh_iemlt051 t761583 /* dim_iemlt051_1 */ ,
dvnh_iemlt060 t761927 /* dim_iemlt060_2 */ ,
dvnh_iemlt060 t761597 /* fact_iemlt060 */
where ( t761597.sozlesme_no = t761693.sozlesme_no and t761693.dvnh_ilce_kodu = t761907.dvnh_ilce_kodu and
t761597.dvnh_ilce_kodu = t761907.dvnh_ilce_kodu and t761597.dvnh_sayac_capi = t761864.dvnh_sayac_capi and
t761583.dvnh_sayac_capi = t761597.dvnh_sayac_capi and t761597.poz_kodu = t761927.poz_kodu and t761693.bina_no =
t762180.bina_no and t761693.blok = t762180.blok and t761693.dvnh_ilce_kodu = t762180.dvnh_ilce_kodu and
t761693.dvnh_mahalle_kodu = t762180.dvnh_mahalle_kodu and t761693.dvnh_sokak_kodu = t762018.dvnh_sokak_kodu and
t761693.dvnh_sokak_kodu = t762180.dvnh_sokak_kodu and t761693.ek_bina_no = t762180.ek_bina_no and t761693.si
te_bina_no = t762180.site_bina_no )
order by c1, c2, c3, c4, c5, c6, c7, c8, c9, c10, c11, c12, c13
```

Şekil 5.19. Maliyetli SQL Sorgularının Ekran Görünümü

5.9.5 Yüksek Kaynak Tüketimi

Yanıt süresi performans iyileştirmenin en önemli birimidir. [50] Bu bağlamdan oracle veritabanında performans iyileştirmesi yapılırken yanıt süresi (Response Time) analiz yöntemi kullanılır. Bu denklem şu şekilde ifade edilir. [50]

$$\text{Yanıt Süresi} = \text{Servis Süresi} - \text{Bekleme Süresi} \quad (\text{Response Time} = \text{Service Time} + \text{Wait Time})$$

Servis Süresi: AWR raporundan "CPU used by this session" ifadesine karşılık gelir yani bir prosesin (işlemin) CPU üzerinde harcadığı toplam zamandır.

Bekleme Süresi: Bir işlemin işlem yapmadan önce çeşitli kaynakların uygun olması için geçen süredir.

Örneğin: İki VT resminin alınması (snapshot) arası geçen süre 46 dakika olan bir awr raporunun maliyeti yüksek olan ilk beş olayını (event) inceleyelim.

Snapshot: awr-stastpact raporu oluşumu için ihtiyaç duyulan istatistiklerin toplanmasının başlama ve bitiş zamanını ifade eder.

Top 5 Wait Events

Event	Wait Waits	% Total Time(cs)	Wt Time
direct path read	4,232	10,827	52.01
db file scattered read	6,105	6,264	30.09
direct path write	1,992	3,268	15.70
control file parallel write	893	198	.95
db file parallel write	40	131	.63

Aynı raporda servis süresi aşağıdaki şekilde belirtilmiştir.

Statistic	Total	per Second	per Trans
CPU used by this session	358,806	130.5	12,372.6

Burada belirtilen verilere göre yanıt süresini ve her bir işlemin (event) kullanım oranları aşağıdaki şekilde hesaplanmaktadır. Önce "direct path read" işlemi için hesaplayalım.

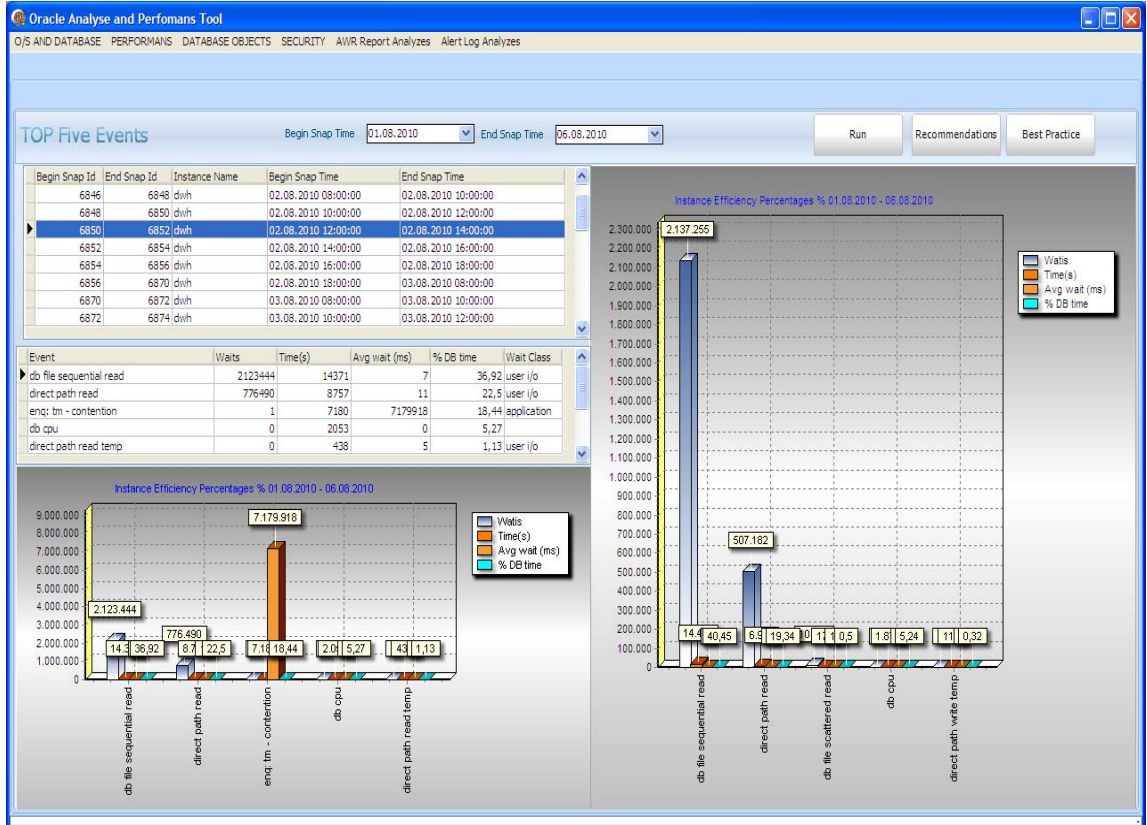
Bekleme Süresi (Wait Time) = $10,827 \times 100\% / 52,01\% = 20,817$ cs
 Servis Süresi (Service Time) = 358,806 cs
 Yanıt Süresi (Response Time) = $358,806 + 20,817 = 379,623$ cs

Bu verilere göre her bir işlemin(event) yanın sürelerinin kullanım oranları aşağıdaki şekilde olmaktadır. [49]

CPU time	= 94.52%	= (358,806/379,623)
direct path read	= 2.85%	
db file scattered read	= 1.65%	
direct path write	= 0.86%	
control file parallel write	= 0.05%	
db file parallel write	= 0.03%	

5.9.5.1 Tespit Edilen Sorunlar ve Çözüm Önerileri

Şekil 5.20’de en fazla kaynak kullanılan olaylardan ilk beş’i (top 5 events) belirtilmiştir. Sol taraftaki ekran tek AWR raporlarının analizini gösterirken sağ taraftaki ekran belirtilen iki tarih arasındaki AWR raporlarının analiz sonuçlarını göstermektedir. Bu örnekte dwh oracle anına (instance) ait 01.08.2010 ile 06.08.2010 arasında iki saatlik periyotlarla alınan AWR raporlarının analizi yapılmış ve analiz sonuçları gösterilmiştir. Saat 12.00 ile 14.00 arasından uygulamada VT kilitlerinde kaynaklanan bir sorun olduğu tespit edilmiştir.



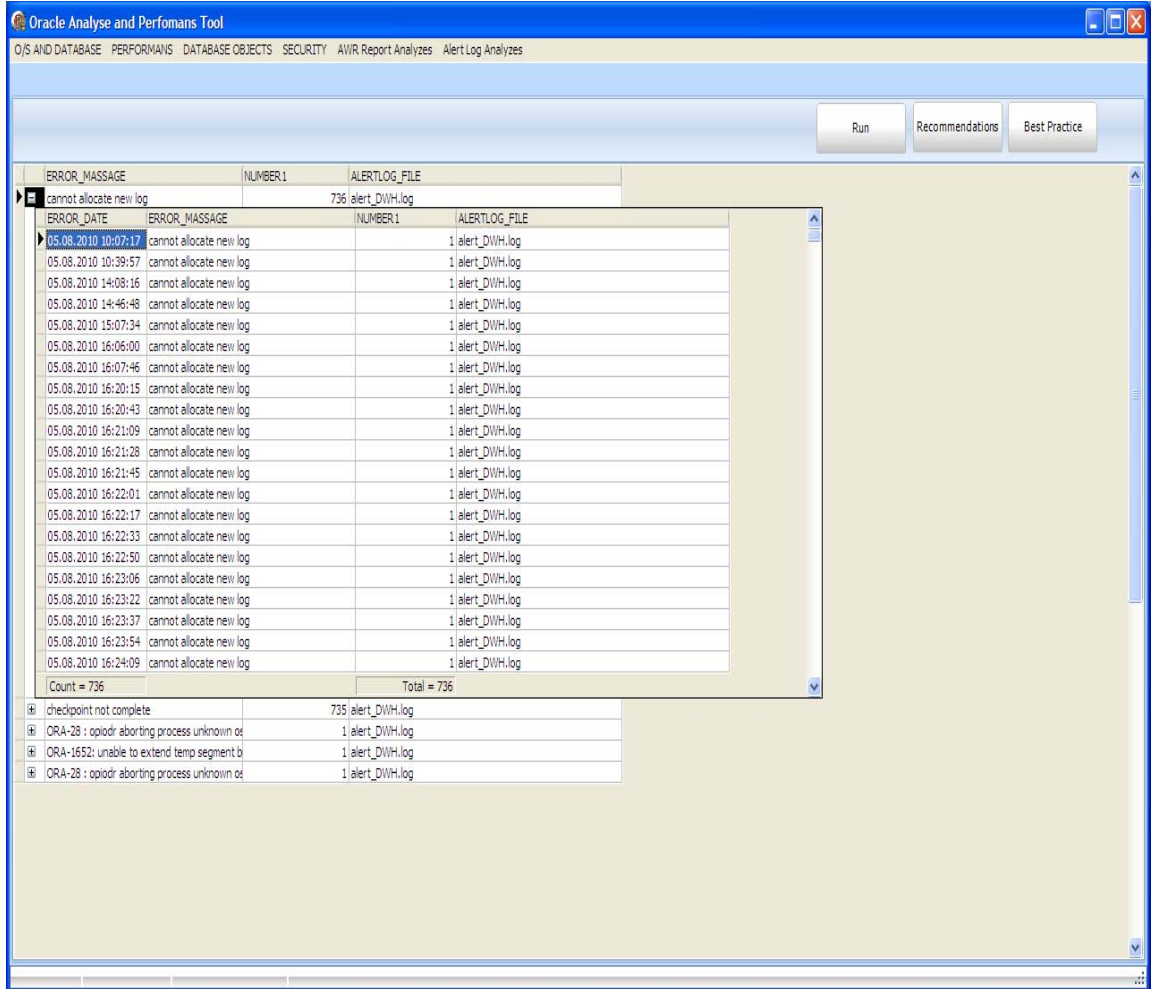
Şekil 5.20. Maliyeti Yüksek Olan İlk Beş Olayın Analiz Ekranı

5.10 Veritabanı Alert Log Analizi

DAPT ana ekranında alert log kütüğü (txt dosya) okunuyor ve içerisinde hata, uyarı geçen ifadeler tarih'leriyle beraber "DAPT.ALERTLOGSTATIS" veritabanı tablosuna yazılmaktadır ve hangi hatanın ve uyarının ne zaman, hangi oracle anına ait olduğu ve kaç kez tekrarlandığı bilgisi içermektedir. Buradaki hatalar incelenmeli özellikle "ORA-0600" hataları için metalink'de oracle firmasına hata bildirimini yapılmalıdır.

5.10.1 Tespit Edilen Sorunlar ve Çözüm Önerileri

Şekil 5.21’de alert log analiz ekranı görülmektedir. Bu örnek’te redo log dosyalarında kaynaklanan majör bir performans probleminin olduğu anlaşılmaktadır.



ERROR_MESSAGE	NUMBER1	ALERTLOG_FILE
cannot allocate new log	736	alert_DWH.log

ERROR_DATE	ERROR_MESSAGE	NUMBER1	ALERTLOG_FILE
05.08.2010 10:07:17	cannot allocate new log	1	alert_DWH.log
05.08.2010 10:39:57	cannot allocate new log	1	alert_DWH.log
05.08.2010 14:08:16	cannot allocate new log	1	alert_DWH.log
05.08.2010 14:46:48	cannot allocate new log	1	alert_DWH.log
05.08.2010 15:07:34	cannot allocate new log	1	alert_DWH.log
05.08.2010 16:06:00	cannot allocate new log	1	alert_DWH.log
05.08.2010 16:07:46	cannot allocate new log	1	alert_DWH.log
05.08.2010 16:20:15	cannot allocate new log	1	alert_DWH.log
05.08.2010 16:20:43	cannot allocate new log	1	alert_DWH.log
05.08.2010 16:21:09	cannot allocate new log	1	alert_DWH.log
05.08.2010 16:21:28	cannot allocate new log	1	alert_DWH.log
05.08.2010 16:21:45	cannot allocate new log	1	alert_DWH.log
05.08.2010 16:22:01	cannot allocate new log	1	alert_DWH.log
05.08.2010 16:22:17	cannot allocate new log	1	alert_DWH.log
05.08.2010 16:22:33	cannot allocate new log	1	alert_DWH.log
05.08.2010 16:22:50	cannot allocate new log	1	alert_DWH.log
05.08.2010 16:23:06	cannot allocate new log	1	alert_DWH.log
05.08.2010 16:23:22	cannot allocate new log	1	alert_DWH.log
05.08.2010 16:23:37	cannot allocate new log	1	alert_DWH.log
05.08.2010 16:23:54	cannot allocate new log	1	alert_DWH.log
05.08.2010 16:24:09	cannot allocate new log	1	alert_DWH.log

Count = 736 Total = 736

checkpoint not complete	735	alert_DWH.log
ORA-28 : opiodr aborting process unknown of	1	alert_DWH.log
ORA-1652: unable to extend temp segment b	1	alert_DWH.log
ORA-28 : opiodr aborting process unknown of	1	alert_DWH.log

Şekil 5.21. Alert Log Analiz Ekranı

Normal şartlarda 15–30 dakikalık periyotlarda bir Log-switch (log anahtarlama) yapılması gerekirken, sık sık bir saat içerisinde 24 kez log anahtarlama yapılmak istemiş ancak, redo log dosyasının boyutunun küçük ve sayısının yetersiz olması nedeniyle “Checkpoint not complete, cannot allocate new log” mesajları alınmıştır. Sorunun çözülmesi için redo log kütüklerinin sayısı ve boyutunu “5.6.6 Redo Log Anahtarlama” bölümünde belirtilen formüle göre artırılmalıdır.

5.11 DAPT Özellikleri

DAPT uygulaması 4. bölümde sözü edilen piyasada en yaygın olarak kullanılan benzer uygulamalardan farklı olan en temel özelliği; veritabanı ve üzerinde çalışan uygulamalardaki sorunların tespit edilmesi ve bu sorunlarla ilgili çözüm önerilerinin yapılmasıdır. Bir diğer temel farkı; ürünün içerisinde bulunan örnek uygulamalar (Best Practices) ile veritabanı yöneticilerinin bilgilerinin gelişmesine katkı sağlamadandır. Oracle veritabanıyla ilgili piyasada ya da internet ortamında birçok doküman ve kitaplar bulunmaktadır. Ancak bu kitapların araştırılması doğru ve sağlıklı bilgiye ulaşması oldukça zaman almaktadır. Sağlıklı ve verimli kaynağa ulaşmak uzun bir süreç sonucu yakalanabilir. Tez kapsamında geliştirilen DAPT uygulaması bu bağlamda veritabanı yöneticilerine bir yol gösterici olarak araştırmaların kısa sürede ve sağlıklı yapılmasına katkı sağlamaktadır.

Delphi 2010 programlama diliyle geliştirilen DAPT uygulaması bir veritabanı yönetim aracından daha çok oracle veritabanı üzerinde çalışan uygulamaların performans ve güvenlik sorunlarını tespit eden ve bu sorunlarla ilgili çözüm önerileri yapan bir veritabanı analiz aracıdır. Oracle ve üzerinde çalışan uygulamaları analiz ederken üç farklı şekilde yakalaşmaktadır:

1. Oracle veritabanının çalıştığı sunucunun donanımsal özellikleri ve hangi işletim sistemi olduğunu tespiti yapılır ve veriler doğrultusundan veritabanı ile işletim sistemi arasındaki kaynak planlamasının analizi yapılmaktadır. Yapılan analiz sonucu eksik ya da yanlış bir kaynak dağılımı tespit edilmişse çözüm önerileri yapmaktadır.
2. Oracle veritabanının fiziksel ve mantıksal yapısını analizini yapar. Veritabanında yaşanan performans ve güvenlik sorunlarının birçoğu oracle'ın kurulumunun sağlıklı yapılmaması, veritabanının fiziksel ve mantıksal yapısının düzgün

tasarlanmamasından kaynaklanmaktadır. Yapılan analiz oracle veritabanının evrensel yapısının kurulmasını sağlamaktır. Bu bağlamda veritabanının fiziksel ve mantıksal yapısında eksik ya da evrensel yapısına uymayan durumları örneğin; güvenlik ve performans problemi teşkil edici durumlar tespit ederek çözüm önerileri yapmaktadır.

3. Veritabanı üzerinde çalışan uygulama nesnelerinin güvenlik ve performans analizi yapılmaktadır. Performans ve güvenlik problemi teşkil eden durumlar tespit edilerek çözüm önerileri yapılmaktadır. Veritabanı performans ve güvenlik sorunlarının çok büyük bir bölümü uygulamaların yanlış tasarım ve SQL sorgularından kaynaklandığı gözlemlenmektedir.

DAPT' nin piyasadaki emsali ürünlerle arasındaki en temel farklar:

- Veritabanı analiz aracıdır. Yani oracle veritabanını ve üzerinde çalışan uygulamaların yapısında herhangi bir değişiklik yapmıyor, sadece sistemi analiz ediyor.
- Diğer uygulamalar sadece veritabanını anlık kontrol ediyor ve isteğe bağlı SP raporu üretiyor. DAPT ise oracle veritabanı ve üzerinden çalışan uygulamaları anlık kontrol ve analizinin yapılmasıyla beraber belirtilen zaman aralığında analiz işlemi yapıyor.
- Orac'ın kendisinin üretmiş olduğu SP ve AWR raporlarını analiz yapılarak performans probleminin tespiti ve çözüm önerisi yapılıyor.
- İsteğe bağlı olarak SP ve AWR raporları tek tek ya da belirtilen bir zaman aralığından aynı anda birden fazla raporların analizi yapılabiliniyor.
- Veritabanı ve uygulamanın genel yapısı kontrol edilerek sağlık taraması yapılıyor.
- Oracle veritabanının hata izleme kütüğü analizi yapılıyor.
- DAPT ürününün içerisinde bulunan örnek uygulamalar ile veritabanı yöneticilerinin bilgi ve deneyimlerinin artmasına katkı sağlıyor.

- Oracle veritabanını kullanan firma yöneticilerinin anlayacağı bir dilde raporlar üretiliyor ve bu raporlar istenirse excel ve pdf dosyası halinde dışarı alınıyor.

6 SONUÇ

Oracle veritabanı çok güçlü performans, güvenilirlik, erişebilirlik, ölçeklenebilirlik, güvenlik ve yönetim özelliklerini bir arada barındırdığı için günümüzde en yaygın kullanılan veritabanları arasında lider durumdadır. Eğitim ücret ve lisanslama maliyetlerinin çok fazla olması nedeniyle piyasada yeterli bilgi ve deneyime sahip kaynak bulunmamaktadır. Bu bağlamda geliştirilen uygulamayla bu eksikliği giderilmesi düşünülmektedir. Aynı zamanda oracle veritabanı yöneticilerinin bilgi ve deneyimlerinin artması kanaati taşımaktadır.

Bu tez kapsamında geliştirilen uygulama kullanıcıların iş ihtiyaçları doğrultusunda sistemlerinin analiz, sistemdeki sıkışma noktasını tespiti, teknolojik ihtiyaçlar gereksinimi belirterek gerek oracle veritabanı özelliklerini verimli kullanılmasına ve gelecekte olası bir donanım kaynak ihtiyacının (disk, cpu, ram vb.) giderilmesine fayda sağlayacağı kanaati taşımaktadır.

KAYNAKLAR

1. Kaya H. , Köymen K. ,Verimadancılığı Kavramı ve Uygulama Alanları, Doğu Anadolu Bölgesi Araştırmaları, 2008
2. <http://www.mis.boun.edu.tr/kutlu/bp175/vt1.doc>, (10.08.2011)
3. <http://en.wikipedia.org/wiki/Database>, (10.08.2011)
4. <http://www.scribd.com/doc/54622169/2/VER%C4%B0TABANI-TURLER%C4%B0>, (10.08.211)
5. http://tr.wikipedia.org/wiki/Veri_taban%C4%B1 , (11.08.2011)
6. <http://support.microsoft.com/kb/824263/tr>, (11.08.2011)
7. <http://tr.wikipedia.org/wiki/MySQL>, (11.08.2011)
8. http://tr.wikipedia.org/wiki/Microsoft_SQL_Server, (11.08.2011)
9. http://en.wikipedia.org/wiki/IBM_DB2, (12.08.2011)
10. <http://www-01.ibm.com/software/data/db2/>, (12.08.2011)
11. http://tr.wikipedia.org/wiki/IBM_DB2, (12.08.2011)
12. <http://www.oracle.com/technetwork/issue-archive/2007/07-jul/o4730-090772.html>, (15.08.2011)
13. http://www.dba-oracle.com/t_history_oracle.htm, (15.08.2011)
14. Eripek O. , VMWARE Server ile Enterprise Linux Üzerinde Oracle RAC Performans Analizi, Haliç Üniversitesi, Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Ana Bilim Dalı Yüksek Lisans Tezi, Aralık 2008
15. Womack J. , Han L. D. , Oracle9i Database: Dumps, Crashes, and Corruptions Student Guide, Oracle Corporation, Usa, February 2002
16. Ault M. , Oracle8i Administration and management, Wiley Computer Publishing, ISBN: 0-471-35453-8, Usa, 2000
17. Gelais M. , Oracle9i Database Administration Fundamentales 1, Volume 1. Student Guide, Oracle Corporation, Usa, September 2002
18. Kyte T. , Expert Oracle Database Architecture: 9i and 10g Programming Techniques and Solutions, Apress Media , ISBN: 1-59059-530-0, Usa, 2005
19. Adams S. , Oracle8i Internal Services for Waits, Latches, Locks, and Memory, O'reilly Media, ISBN: 1-56592-598-X, October 1999

20. Khan S. , Enterprise DBA Part3: Network Administration Student Guide, Oracle Corporation, Usa, August 1999
21. Billings M. , Matishak D. , Spiller J. , Vennapusa P. , Oracle Database 11g Administration Workshop 1. Student Guide, Oracle Corporation, Usa, October 2008
22. Fogel S. , Stern J. , McGregor C. , Oracle Database 2 Day DBA 11g Release 1 (11.1), Oracle Corporation, Part Number B28301-03, Usa, March 2008
23. Leverenz L. , Rehfield D. , Oracle8i Concepts Release 8.1.5, Oracle Corporation, Part No. A67781-01, Usa, February 1999
24. Dyke R. V. , Lowenthal R. , Oracle Database 10g Administration Workshop 1 Student Guide, Oracle Corporation, Usa, March 2004
25. Best T. , Billings M.J. , Oracle Database 10g: Administration Workshop I Student Guide , Volume I , D17090GC30 Edition 3.0, Usa, November 2005
26. Ashdown L. , Kyte T. , Oracle Database Concepts 11g Release 2 (11.2), E16508-05, Oracle, October 2010
27. Alapati S. R. , Expert Oracle Database 10g Administration, Apress Media, ISBN: 1-59059-451-7, Usa, 2005
28. Kılınçkaya A. , Oracle için Veritabanı Yönetim Aracı ve performans Analizi, TOBB Ekonomi ve Teknoloji Üniversitesi, Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Yüksek Lisans Tezi, Haziran 2008
29. Kyte T. , Effective Oracle by Design, McGraw-Hill/Osborne, ISBN: 0-07-223065-7, Usa, 2003
30. Lewis J. , Cost-Based Oracle Fundamentals, Apress Media, ISBN: 1-59059-636-6, Usa, 2006
31. <http://www.toadsoft.com>, (03.08.2011)
32. <http://www.keeptool.com>, (03.08.2011)
33. <http://www.datasparc.com>, (03.08.2011)
34. <http://www.embarcadero.com>, (08.08.2011)
35. <http://www.softtreetech.com>, (08.08.2011)
36. Dyke R.V. , Haan L. , Jeal C. , Stem J. , Verrier J.F. Oracle Database 10g:New Features for Administrators, Oracle Corporation, D17079GC10, Usa, February 2004
37. www.metalink.oracle.com Note: 311395.1, (05.09.2010)
38. www.metalink.oracle.com Note: 1019927.6, (05.09.2010)

39. www.metalink.oracle.com Note: 117203.1 - How to Use DBMS_STATS to Move Statistics to a Different Database, (01.09.2010)
40. www.metalink.oracle.com Note: 102334.1: How to Automate Change Based Statistic Gathering - Monitoring Table, (01.09.2010)
41. Lane P. , Oracle Database Data Warehousing Guide, 10g Release 2 (10.2), B14223-02, Oracle Corporation, December 2005
42. Chan I. , Oracle Database Performance Tuning Guide, 10g Release 2 (10.2), B14211-01, Oracle Corporation, June 2005
43. Best practices for a Data Warehouse on Oracle Database 11g, An oracle White Paper , September 2008
44. Bird M. , Lane P. , Oracle Database 2 Day + Data Warehousing Guide, 11g Release 1 (11.1), B28314-01, Oracle Corporation, July 2007
45. Shee R. , Deshpande K. , Gopalakrishnan K. , Oracle Wait Interface: A Practical Guide to Performance Diagnostics & Tuning, ISBN: 007222729x, McGraw-Hill/Osborne, Usa, 2004
46. Powell G. , Oracle High Performance Tuning for 9i and 10g, ISBN:1555583059, Usa, 2004
47. <http://www.oracle.com/technetwork/topics/security/alerts-086861.html>, (20.04.2010) [50]
48. Knox D. , Effective Oracle Database 10g Security by Design, ISBN: 0-07-223130-0, McGraw-Hill/Osborne, Usa, 2004 [47]
49. Spiller J. , Dyke R. V. , Ernst B. A. , Hale L. P. , Oracle Database 10g Security Student Guide Volume 1, D17499GC10, Oracle Corporation, May 2005 [48]
50. www.metalink.oracle.com Troubleshooting I/O-related waits Note.ID 223117.1, (07-05-2009) [49]

ÖZGEÇMİŞ

Halil Kaya, 1972 yılında Musabeyli Kilis’de doğdu. İlköğrenimini Tahtalıklaradut ilköğretim okulu ve lise öğrenimini Kilis Lisesi Matematik bölümünden 1990 yılında mezun oldu. 1997 yılında Yüzüncü Yıl Üniversitesi Bilgisayar Programlama Bölümünden, 2007 yılından Anadolu Üniversitesi İktisat bölümünden mezun oldu. 1997 Soyak A.Ş.’de bilgi işlem de programcı olarak işe başladı ve aynı yıl içerisinde Sentim Bilgisayarda (İstanbul) teknik destek uzmanı olarak çalıştı. 1998’den 2002 yılına kadar Obje Şirketler Topluluğunda (İstanbul) Oracle Veritabanı Yöneticisi ve Delphi Programcısı olarak çeşitli projelerde görev aldı. 2002’den 2008 yılına kadar Ford Otosan da (Ford Otomotiv Sanayi A.Ş.) veritabanı yöneticisi ve gurup lideri olarak çalıştı. 2008’den 2010 yılına kadar Oracle ve IBM şirketlerinden sözleşmeli teknik oracle danışmanı olarak yurt içi ve yurt dışı çeşitli kurumsal şirketlerin projelerde çalıştı. 2010 yılında kendi İnoTelica Limitet şirketini kurdu ve halan kendi şirketinde çalışmaktadır.