Oracle DBA Concise Handbook

Covers 9i to 11g

Saikat Basak

ORACLE

Certified Professional

Published by Ensel Software
www.enselsoftware.com

1st Edition 2004
2nd Edition 2009 – updated for 10g & 11g

Last updated Jul 2010

Contents

Introduction

There are several books available in the market for Oracle DBAs. So what is special about this book?

Well, the differences are several. First of all, this book is available in electronic format. You can carry this book wherever you want. It is written in simple Word format (PDF format is also available). So, you can open it in almost anywhere. This book is based on the commands necessary for regular DBA activities. This is a concise book. It attempts to be your first reference for DBA job. You will still require looking at Oracle manuals and other huge references for advanced commands and features. Please note that this book does not try to teach you how to be a DBA. It is assumed that you know a bit of DBA activities. It only acts as a DBA handbook.

I included a section on common DBA interview questions. Where do you find such information in other books? Some avid readers wanted the answers to be given along with. However, some questions are too easy. Some difficult answers/hints have been provided. But I don't appreciate providing all answers because unless you have sound understanding of the concept, you can be very easy bogged down by some twists in the questions.

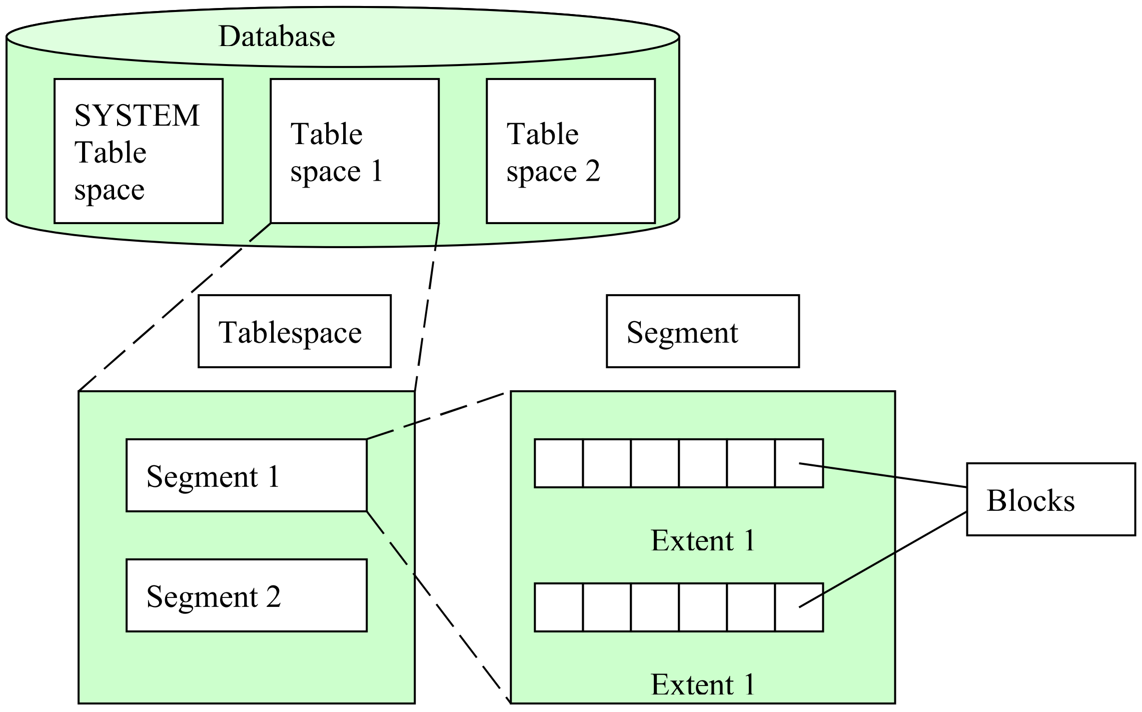
This is really a CONCISE handbook. That's why I couldn't cover every nuts and bolts of Oracle. Had I do so, this book would have consisted of more than 1000 pages and its charm would have been lost! Only the most basic aspects have been touched. So, please don't yell that why such and such topics have been left. However, if readers strongly feel that some topics need to be added, I shall definitely honor that demand.

I shall occasionally update the book. So, please visit the website frequently to download latest version of the book.

Please note that originally I wrote this book for Oracle 9i and then updated for 11g.

1 Oracle Server – an overview

1.1 Logical structure



Physical storage structures

Oracle database consists of 3 types of physical files –

- Data files – contains all database data
- Redo log files – record all changes made to data. Used for recovery
- Control files – maintains information about physical structure of database

1.2 Oracle memory structure

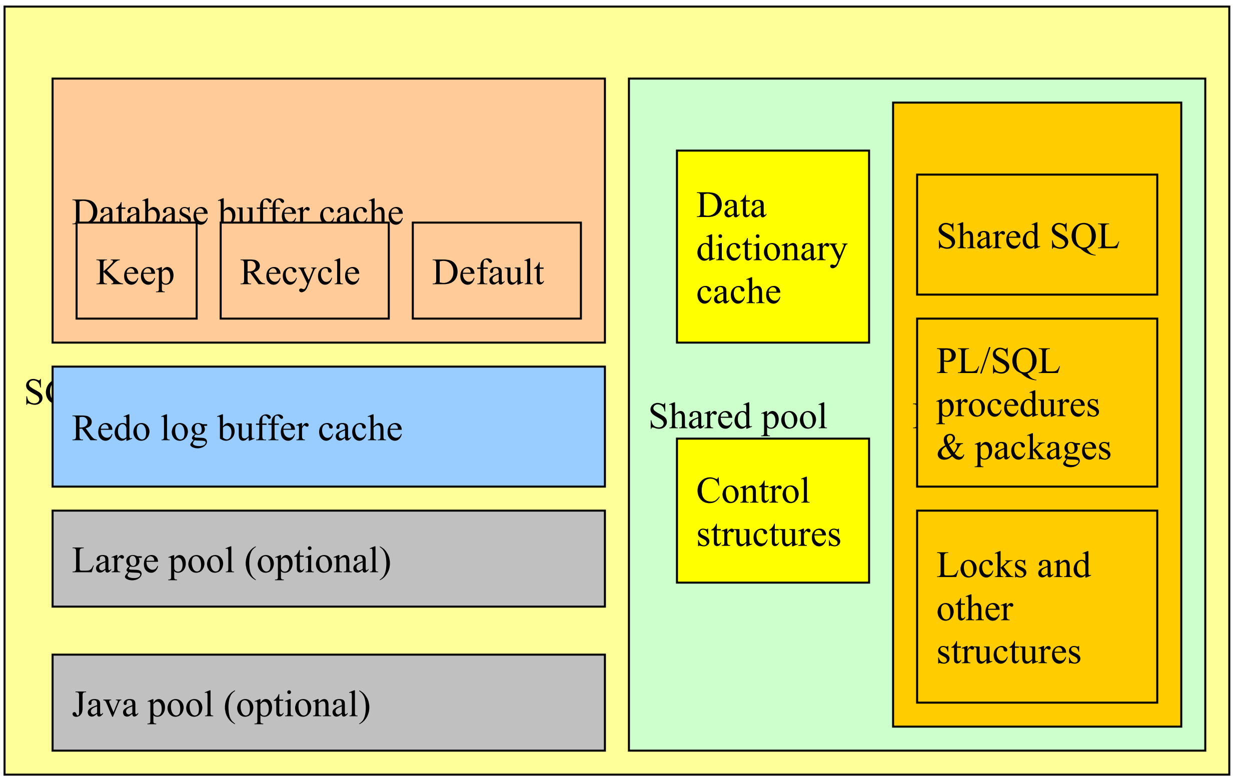
1.2.1 SGA

System Global Area (SGA) is shared memory area. All users of database share information maintained in this area. The SGA and other background processes constitute an Oracle instance.

SGA size is limited by SGA_MAX_SIZE initialization parameter.

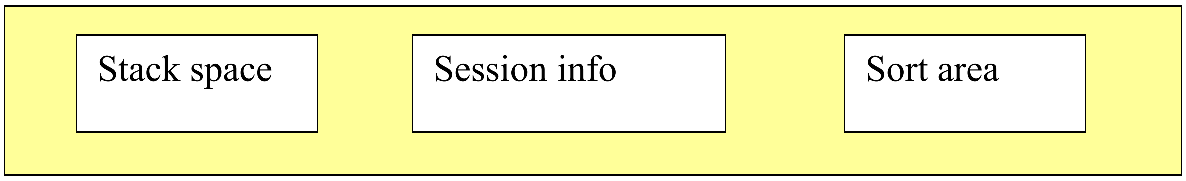
From 11g, Oracle can manage SGA and PGA completely automatically.

Shared memory



From 11g, there are two new components in SGA viz Streams pool and Result cache.

Non-shared memory
PGA



The database buffer cache is the area of memory that caches database data, holding blocks from data files that have been read recently. Before a user can look at a piece of information in an Oracle database, it must first reside in the database buffer cache. Data gets into this cache based upon the Most Recently Used algorithm. Because the most recently and most frequently used data is kept in memory, less disk I/O is necessary, and overall database performance is improved.

There are 3 types of buffers – dirty, free and pinned.

Oracle uses an LRU mechanism to remove data from DB cache.

DB_CACHE_SIZE, DB_KEEP_CACHE_SIZE,
DB_RECYCLE_CACHE_SIZE determines DB cache size.

DB_CACHE_ADVICE can be set to ON/OFF/READY and the result can be viewed from V$DB_CACHE_ADVICE.

Redo log buffer is a circular buffer in SGA that holds information about changes made to data.

LOG_BUFFER determines its size.

1.2.2 Shared pool

Library cache contains shared SQL area, PL/SQL procedures and packages etc. It is used for maintaining recently executed SQL commands and their execution plans.

Data dictionary cache is a collection of database tables and views containing metadata about the database, its structures, its privilege and its users. Oracle accesses data dictionary frequently during parsing of SQL statements. Data dictionary cache holds most recently used data dictionary information.

SHARED_POOL_SIZE determines size of shared pool and can be dynamically altered.

PGA contains data and information for single server process. PGA_AGGREGATE_TARGET specifies total amount of memory that can be used by all server processes.

1.3 Background processes

1.3.1 Database Writer (DBWn)

Oracle marks buffers in memory as dirty when the data they contain is changed. DBWn writes content of dirty buffer to data file when – a server process can't find a clean buffer after searching set threshold of buffers, a checkpoint occurs, change table space to read only/offline/backup mode, drop/truncate table etc.

1.3.2 Log Writer (LGWR)

It is responsible to redo log buffer management. Almost all activities against the database are tracked in the online redo logs. As transaction are initiated and eventually committed or rolled back, a record of this activity is written to these log files.

Log writer writes to redo logs sequentially.

1.3.3 Checkpoint (CKPT)

Helps to reduce time required for instance recovery. A checkpoint is an event that flushes modified data from buffer cache to disk and updates control file and data files. The CKPT process updates header of data files and control files and DBWn writes actual blocks to file. Checkpoint occurs automatically when an online redo log file fills (log switch).

1.3.4 System Monitor (SMON)

At startup, SMON’s job is to ensure that all the database files are consistent and perform recovery if required. There is also an assortment of other cleanup activities that may need to be done, which are SMON’s responsibility. The SMON process by itself checks every so often to see whether there are any tasks waiting for its attention.

1.3.5 Process Monitor (PMON)

Cleans up failed user processes and frees all resources used by failed process.

1.3.6 Archiver (ARCn)

It automatically saves copies of redo logs in a DBA specified storage location when media recovery is enabled.

1.3.7 Recover (RECO)

Is used with distributed transaction to resolve failure.

1.3.8 Lock (LCKn)

It is used in RAC.

1.3.9 Dispatcher (Dnnn) & Shared Server (Snnn)

These are discussed later.

1.3.10 Manageability monitor (MMON)

From 10g, it makes snapshots of the database's health (statistics) and stores this information in the automatic workload repository.

1.3.11 Processing SQL

The following steps show how Oracle processes SQL

1. Statement is passed to Oracle for processing
2. Before it is placed in the library cache, a hash value is computed that represent s a number of characteristics of the SQL.
3. Oracle compares the computed hash value against those values in a hash table where it maintains for SQL statements already in the cache.
4. If a match is found, the new SQL statement is thrown away and the one sitting in the cache is executed on its behalf.
5. In no match is found, further processing is done on the new SQL statement, an entry is made in the library cache hash table for newly arrived code, and it is placed in the library cache.
6. There are 3 stages of SQL processing – parse, execute and fetch
   - During parsing, Oracle server checks the syntax and validates table, column names against data dictionary
   - Determines whether user has privilege to execute the statement
   - Determines optimal execution plans for statement
   - Finds a shared SQL area for the statement
   - In execution stage, for UPDATE and DELETE statement, Oracle locks the affected rows, looks for data blocks in DB buffer cache, if found, executes becomes faster, if not then Oracle has to read from physical data files to buffer cache. For SELECT and INSERT statements, locking is not necessary.
   - During fetch operation, rows are fetched to user process.

An Oracle instance consists of memory structures (SGA) and background processes (DBWn, LGWR, CKPT, SMON, PMON, MMON and optionally ARCn, Dnnn, Snnn etc.)

1.4 Installing and managing Oracle database

To use operating system authentication, set REMOTE_LOGIN_PASSWORDFILE parameter to NONE (default).

OS authenticated users can connect as CONNECT / AS SYSDBA (or SYSOPER).

When using password file authentication, users connect to database by specifying username and password.

1. Using ORAPWD utility, create a password file with SYS password.
2. Set REMOTE_LOGIN_PASSWORDFILE parameter.
3. Grant appropriate users SYSDBA or SYSOPER privilege.

1.5 Oracle Managed Files (OMF)

Set following parameters in initialization file.

DB_CREATE_FILE_DEST – default location of new datafiles
DB_CREATE_ONLINE_LOG_DEST_n – specifies location for online log files and control files (max. 5 locations)

To create database using OMF, use

CREATE DATABASE MYDB DEFAULT TEMPORARY TABLESPACE TMP;

1.6 Creating a new database

Usually when you install Oracle, it automatically creates a database for you (though you need to specify a database name for this). Otherwise, you usually use graphical Database Configuration Assistant (DBCA) to create/manage databases. So, you may not even require creating a database from command prompt. However, in case you need to, the steps are shown below.

In OS command prompt, use

```
ORADIM -NEW -SID sid_name -INTPWD sys_password -MAXUSERS 10
```

In Windows, go to Control Panel – Services
Check service named OracleServiceSID started.

In My Computer, Environment tab, define
ORACLE_SID=sid_name

Create folders where you want to keep database files eg.
F:\MYDB\DATAFILES, UDUMP, BDUMP, CDUMP, LOGS, ARCHIVE etc.

Create a new initSID.ora file with relevant parameters

```
BACKGROUND_DUMP_DEST=F:\MYDB\BDUMP
CORE_DUMP_DEST=F:\MYDB\CDUMP
USER_DUMP_DEST=F:\MYDB\UDUMP
DB_NAME= sid_name
INSTANCE_NAME= sid_name
UNDO_MANAGEMENT=AUTO
UNDO_TABLESPACE=UNDO01
```

Start SQL Plus
/ AS SYSDBA

```
STARTUP NOMOUNT

CREATE DATABASE MYDB
MAXDATAFILES 30
MAXLOGFILES 10
MAXLOGMEMBERS 5
MAXINSTANCES 1
MAXLOGHISTORY 1
DATAFILE 'F:\MYDB\DATAFILES\SYSTEM01.DBF' SIZE 100M
LOGFILE
GROUP 1 'F:\MYDB\LOG\LOG01.DBF' SIZE 10M,
GROUP 2 'F:\MYDB\LOG\LOG02.DBF' SIZE 10M
UNDO TABLESPACE UNDO01
DATAFILE 'F:\MYDB\UNDO\UNDO01.DBF' SIZE 50M
DEFAULT TEMPORARY TABLESPACE TEMP
TEMPFILE 'F:\MYDB\TEMP\TEMP01.DBF' SIZE 10M
CHARACTER SET WE8MSWIN1252
/
```

CREATE SPFILE FROM PFILE;

Note: Above command was for Oracle 9i. From 10g, you need Sysaux table space as well.

Now run calalog.sql (creates data dictionary views) and catproc.sql (creates PL/SQL packages) from $ORACLE_HOME\RDBMS\ADMIN

After installing Oracle, several services are registered in the server computer. In Windows, (for Oracle 9.2) following services must be started as a minimum to run Oracle – OracleOraHome92Agent, OracleOraHome92TNSListener and OracleServiceSID (where SID is the name of database you created).

In Windows Vista running Oracle 11g, I usually start/stop my test database via following batch file (say *StartOracle.bat – MYDB is name of my database*)

```
net start OracleServiceMYDB
net start OracleOraDb11g_home1TNSListener
net start OracleVssWriterMYDB
net start OracleDBConsoleMYDB
```

To stop database services, you can create a similar batch file by replacing `start` with `stop`. In Vista, you need to run these batch files as administrator.

If you do not start OracleDBConsole<SID> service, then you won't be able to start web based Enterprise Manager – which you usually start from https://localhost:1158/em (where localhost may be substituted for your computer name)

You usually log on to Oracle server from "*SQL Plus*" as – SYS/PASSWORD@DATABASENAME AS SYSDBA. Sometimes you can also log in as simply "/ AS SYSDBA" only when you are physically in the same computer where Oracle server is installed. Please note that Server Manager tool is no longer available from Oracle 9i onwards. You can do everything using SQL Plus!

For various day-to-day database works, you may find SQL Plus cumbersome to work with. For this, several 3rd party GUI tools are available. Two most popular tools are – PL/SQL Developers and TOAD. From 11g, Windows SQL Plus has been replaced by SQL Developer suite bundled with 11g. DOS version of SQL Plus is still available though!

1.7 Starting up database instance

SQL> STARTUP NOMOUNT
This state is used for creating new database or creating new control file. At this state, Oracle allocates SGA and starts background processes.

SQL> STARTUP MOUNT
This state is used for performing specific maintenance operations like renaming data files, enabling/disabling archive log mode, adding/dropping/renaming redo log files, recovering database etc. Control file is read at this stage but the data files are not open.

SQL> STARTUP OPEN or simply SQL>STARTUP
Database is available for normal operations.

NB: Oracle 11g requires at least 1 GB of RAM and 5 GB disk space to install. For an operational production database, more RAM and disk space are required.

1.8 Shutting down database instance

SQL> SHUTDOWN NORMAL or simply SQL> SHUTDOWN
Waits for all database users to disconnect then closes database.

SQL> SHUTDOWN IMMEDIATE
Terminates all user connections, rolls back uncommitted transactions, closes database.

SQL> SHUTDOWN TRANSACTIONAL
Waits for all transactions to commit or roll back, then closes database.

SQL> SHUTDOWN ABORT
Immediately closes database leaving it in inconsistent state. SMON automatically performs instance recovery during next startup.

1.9 Control file

Control file contains –

- Database name
- Database creation timestamp
- Data files – name, location, on/off line status
- Redo log files – name, location
- Redo log archive information

- Table space names
- Current log sequence number
- Most recent checkpoint information
- Begin and end of undo segments
- RMAN backup information

Oracle backs up control file after any structural changes in database. LGWR updates control file with current log sequence number. CKPT updates control file with recent checkpoint information. ARCn updates with archiving information.

1.9.1 Multiplexing control files

Using init.ora

```
CONTROL_FILES =
('/ora/oradata/mydb/control1.ctl', '/ora/oradata/mydb/control1.ctl')
```

Using spfile

1. SQL> ALTER SYSTEM SET CONTROL_FILES = ('/ora/oradata/mydb/ control1.ctl', '/ora/oradata/mydb/control1.ctl') SCOPE=SPFILE

2. SQL>SHUTDOWN NORMAL
3. Copy control file to new location
4. SQL>STARTUP

To create OMF control files, don't specify CONTROL_FILES parameter in initialization file; rather specify DB_CREATE_ONLINE_LOG_DEST_n parameter n times starting with 1.

1.9.2 Creating new control file

Make sure you have complete list of all data and log files.

In SQL Plus, write (the database should be in STARTUP NOMOUNT)

```
CREATE CONTROLFILE SET DATABASE "MYDB"
        NORESTLOGS NOARCHIVELOG
        MAXDATAFILES 30
        MAXLOGFILES 10
        MAXLOGMEMBERS 5
        MAXINSTANCES 1
        MAXLOGHISTORY 1
DATAFILE
        'F:\MYDB\DATAFILES\SYSTEM01.DBF'
        'F:\MYDB\DATAFILES\USERS01.DBF'
        'F:\MYDB\DATAFILES\UNDO01.DBF'
        'F:\MYDB\DATAFILES\TEMP01.DBF'
LOGFILE
        GROUP 1 'F:\MYDB\LOG\LOG01.DBF' SIZE 10M,
        GROUP 2 'F:\MYDB\LOG\LOG02.DBF' SIZE 10M
/
```

To back up control file when database is running, use

ALTER DATABASE BACKUP CONTROLFILE TO filename REUSE
Or
ALTER DATABASE BACKUP CONTROLFILE TO TRACE – it places a text copy of control file to USER_DUMP_DEST directory.

If Oracle can’t update control file, instance crashes.

1.9.3 Control file related data dictionary views

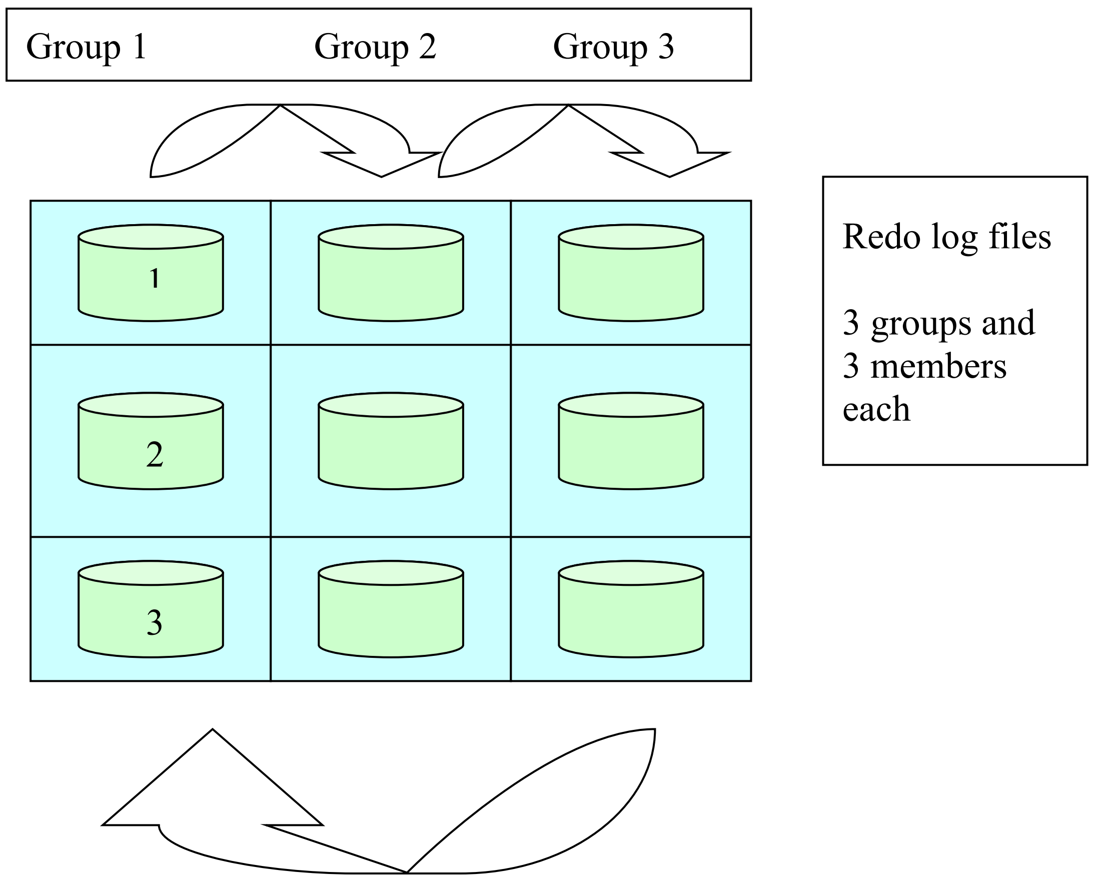
V$CONTROLFILE
V$CONTROLFILE_RECORD_SECTION

1.10 Redo log files

Redo entries record data changes that can be used to reconstruct all changes made to database. Whenever you do any change to database (DML or DDL), it is recorded in redo logs.

To operate successfully, an Oracle instance requires at least 2 redo log groups. Each group must have at least 1 redo log file.

Usually in production databases, there are at least 3 redo log groups and each group has at least 2 redo log members. Note that, all member files under same group are identical. Members are multiple copies to protect against data loss in case of disk failure.



If LGWR can write to at least 1 member of the group, database functions normally, but otherwise Oracle shuts down the instance.

1.10.1 Creating new groups and members

ALTER DATABASE ADD LOGFILE GROUP 3
('/oracle/oradata/log/redo31.log', '/oracle/oradata/log/redo32.log') SIZE 10M

ALTER DATABASE ADD LOGFILE MEMBER
'/oracle/oradata/log/redo31.log' TO GROUP 3

For OMF
ALTER DATABASE ADD LOGFILE

1.10.2 Renaming log members

Follow these steps

- Shutdown database
- Copy/rename redo log file member to new location
- SQL> STARTUP MOUNT
- ALTER DATABASE RENAME FILE old redo log file TO new redo log file
- ALTER DATABASE OPEN
- Backup control file

1.10.3 Dropping redo log file

First, make the log file inactive, if necessary, issue ALTER SYSTEM SWITCH LOGFILE

ALTER DATABASE DROP LOGFILE GROUP 3

ALTER DATABASE DROP LOGFILE MEMBER '/oracle/oradata/log/redo31.log'

1.10.4 Running database in archive log mode

Specify LOG_ARCHIVE_DEST_n parameters in initialization file.

E.g. LOG_ARCHIVE_DEST_1 = ((LOCATION = '/oradata/mydb/archive1') MANDATORY REOPEN = 60)

In 11g, to enable database archive log mode, define archive location like this
ALTER SYSTEM SET LOG_ARCHIVE_DEST_1='LOCATION=C:\Oracle11g\oradata\MYDB\archivelog' SCOPE=SPFILE; (in mount state)

LOG_ARCHIVE_FORMAT = 'arch_%t_%s'

%s – log sequence number
%S – log sequence number, zero filled
%t – thread number
%T – thread number, zero filled

To enable/disable archive log mode, follow these steps

- Shutdown database (log in via conn / as sysdba)
- Startup and mount database
- SQL> ALTER DATABASE ARCHIVELOG (use NOARCHIVELOG to disable)
- SQL> ALTER DATABASE OPEN

Till 9i, in the initialization file, set LOG_ARCHIVE_START = TRUE (if it is not set, once a redo log file is full, Oracle hangs until redo log file is archived). From 10g, this parameter is deprecated.

To manually initiate automatic archiving, issue ALTER SYSTEM ARCHIVE LOG START and ALTER SYSTEM SWITCH LOGFILE commands.

To see whether database is in archive log mode, use

```
SQL> ARCHIVE LOG LIST
```

1.10.5 Redo log related data dictionary views

V$LOG – shows redo log status
V$LOGFILE – shows redo log files' location
V$THREAD
V$LOG_HISTORY
V$ARCHIVED_LOG
V$ARCHIVE_DEST
V$ARCHIVE_PROCESSES

1.11 Table spaces

The database's data is stored logically in table spaces and physically in data files corresponding to the table spaces. One table space can have multiple data file but one data file must belong to only one table space. A single object (say a table) may span multiple data files but must reside within a single table space.

1.11.1 Creating table space

```
CREATE TABLESPACE supermarket DATAFILE
'e:\oracle\oradata\mdb\supermarket.dbf' SIZE 30M
EXTENT MANAGEMENT LOCAL AUTOALLOCATE SEGMENT SPACE
MANAGEMENT AUTO
```

```
CREATE TABLESPACE supermarket DATAFILE
'e:\oracle\oradata\mdb\supermarket.dbf' SIZE 30M
AUTOEXTEND ON NEXT 10M MAXSIZE UNLIMITED
```

1.11.2 Dropping table space

```
DROP TABLESPACE supermarket INCLUDING CONTENTS AND
DATAFILES
```

1.11.3 Renaming table space

From Oracle 10g onwards, you can rename a table space (except System and Sysaux)

```
ALTER TABLESPACE old_name RENAME TO new_name
```

When you rename a table space, all corresponding data dictionary entries are updated.

1.11.4 Availability of table space

```
ALTER TABLESPACE supermarket OFFLINE
NORMAL/TEMORARY/IMMEDIATE/FOR RECOVER
```

You can’t place System table space in offline mode.

To make a table space read only,

```
ALTER TABLESPACE supermarket READ ONLY
```

To change it to read write mode,

```
ALTER TABLESPACE supermarket READ WRITE
```

1.11.5 Adding space to table space

```
ALTER TABLESPACE supermarket_DATA ADD DATAFILE
'c:\oracle\oradata\mydb\supermarket_data2.dbf' SIZE 30M
```

```
ALTER DATABASE DATAFILE
'c:\oracle\oradata\mydb\supermarket_data2.dbf' RESIZE 300M
```

1.11.6 Table space related data dictionary views

DBA_TABLESPACES – all table space information
V$_TABLESPACE
DBA_FREE_SPACE
V$SORT_USAGE
DBA_SEGMENTS
DBA_USERS – shows default and temporary table space for users

1.11.7 Renaming and relocating file

Follow these steps to rename data file (for single table space except System table space)

- ALTER TABLESPACE supermarket OFFLINE
- Copy or move the file to new location with OS commands
- ALTER DATABASE RENAME FILE
  'c:\oracle\oradata\mydb\supermarket_data2.dbf' TO
  'c:\oracle\oradata\mydb\supermarket_new.dbf'
  or
  ALTER TABLESPACE supermarket RENAME DATAFILE
  'c:\oracle\oradata\mydb\supermarket_data2.dbf' TO
  'c:\oracle\oradata\mydb\supermarket_new.dbf'
- ALTER TABLESPACE supermarket ONLINE

In case of System table space or table spaces with multiple data files

- Shutdown database
- Copy or move the file to new location with OS commands
- Startup database in mount state
- ALTER DATABASE RENAME FILE
  'c:\oracle\oradata\mydb\supermarket_data2.dbf' TO
  'c:\oracle\oradata\mydb\supermarket_new.dbf'
- Open database

Please note in case of Windows, the file may get locked unless database is shutdown.

1.11.8 Data file related data dictionary views

V$DATAFILE
V$TEMPFILE
DBA_DATA_FILES
DBA_TEMP_FILES

1.12 Segment and storage structures

PCTFREE = specifies what percentage of block should be allocated as free space for future updates (default 10)
PCTUSED = specifies when the block can be considered for adding new rows (default 40)
PCTFREE + PCTUSED <= 100

Blocks are smallest logical unit of storage in Oracle database.
An extent is logical storage unit made of contiguous data blocks.
Segment is logical storage unit made up of one or more extents.

Types of segments are – table, table partition, cluster, nested table, index, index organized table, index partition, temporary, LOB, undo, bootstrap.

1.12.1 Undo segment

When a user performs an update or deletes operation, the earlier data is saved to undo segments and then actual data is modified to new value. In case of insert operation, rowid of new rows are stored in undo segments.

Undo data is not deleted immediately after commit or rollback. How long it will stay in undo segment depends on UNDO_RETENTION parameter in initialization file.

When a transaction is rolled back, Oracle restores the earlier data from undo segments.

From Oracle 9i, undo management can be automatically controlled.

To use automatic undo management, set following parameters in initialization file.

UNDO_MANAGEMENT=AUTO

```
UNDO_TABLESPACE=table space name
```

1.12.2 Creating undo segment

```
CREATE UNDO TABLESPACE undo DATAFILE
'/oradata/mydb/undo01.dbf' SIZE 20M
```

To specify different table space as undo table space dynamically – issue
ALTER SYSTEM SET UNDO_TABLESPACE=undo02

1.12.3 Extent or segment related data dictionary views

DBA_EXTENTS
DBA_FREE_SPACE
DBA_SEGMENTS
V$SORT_SEGMENT

DBA_ROLLBACK_SEGS
V$ROLLNAME
V$ROLLSTAT
V$UNDOSTAT

1.13 Tables

1.13.1 Creating tables

```
Create table MANUFACTURER
(
  MFDNO     NUMBER not null,
  MFDNAME   VARCHAR2(200) not null,
  ADDRESS   VARCHAR2(200),
  CITY      VARCHAR2(50),
  STATE     CHAR(2),
  COUNTRY   VARCHAR2(100),
  POSTCODE  VARCHAR2(10),
  PHONE     VARCHAR2(50),
  EMAIL     VARCHAR2(100),
  USERNAME  VARCHAR2(20) default USER,
  DATESTAMP DATE default SYSDATE
) tablespace SUPERMARKET_DATA
```

```
CREATE TABLE MANUFACTURER
NOLOGGING PARALLEL
AS
SELECT * FROM COMPANY
```

1.13.2 Reorganizing tables

To move a table to a different table space

```
ALTER TABLE product MOVE TABLESPACE supermarket
```

1.13.3 Dropping a table

```
DROP TABLE schema.table_name (CASCADE CONSTRAINTS)
```

```
TRUNCATE TABLE table_name
```

Truncate resets HWM where delete does not.
Truncate is not logged but delete is logged.

1.13.4 Modifying columns

```
ALTER TABLE schema.table_name DROP COLUMN column_name
(CASCADE CONSTRAINTS)
```

```
ALTER TABLE schema.table_name SET UNUSED COLUMN
column1_name, column2_name (CASCADE CONSTRAINTS)
```

```
ALTER TABLE product ADD Mfd VARCHAR2(30) DEFAULT ‘abc’
```

1.13.5 Table related data dictionary views

DBA_TABLES
DBA_TAB_COLUMNS – all columns of all tables

1.14 Indexes

Oracle has mainly 2 types of indexes, B+ Tree and Bitmap.

1.14.1 Creating index

Normal B tree index

```
CREATE INDEX index_name ON table_name (column_names)
TABLESPACE app_indx
```

Bitmap index

```
CREATE BITMAP INDEX emp_gender_idx ON employee (sex)
TABLESPACE app_indx
```

Reverse key index

```
CREATE INDEX index_name ON table_name (column_names)
REVERSE
```

Function based index

```
CREATE INDEX index_name ON UPPER(product (prodname))
```

To use function based index, set following parameters in initialization file

```
QUERY_REWRITE_ENABLED=TRUE
QUERY_REWRITE_INTEGRITY=TRUSTED
```

Cost based optimizer must be used.

1.14.2 Index Oraganized Table (IOT)

Create the table normally, with ORGANIZATION INDEX keyword. It is suitable when data access is mostly thru primary key. In IOT, rows are physically stored in sorted order of the primary key.

```
Create table MANUFACTURER
(
  MFDNO     NUMBER not null,
  MFDNAME   VARCHAR2(200) not null,
  ADDRESS   VARCHAR2(200),
  CITY      VARCHAR2(50),
  STATE     CHAR(2),
  COUNTRY   VARCHAR2(100),
  POSTCODE  VARCHAR2(10),
  PHONE     VARCHAR2(50),
  EMAIL     VARCHAR2(100),
  USERNAME  VARCHAR2(20) default USER,
  DATESTAMP DATE default SYSDATE
) tablespace SUPERMARKET_DATA
ORAGANIZATION INDEX
OVERFLOW TABLESPACE ovfl_tblsp
INCLUDING address
PCTTHRESHOLD 25
MAPPING TABLE
```

1.14.3 Rebuilding index

```
ALTER INDEX pk_customer REBUILD ONLINE
```

```
DROP INDEX pk_customer
```

You can also move index to a different table space using *ALTER INDEX index_name TABLESPACE new_table_space* command.

1.14.4 Monitoring index usage

```
ALTER INDEX index MONITORING USAGE
```

The V$OBJECT_USAGE view populated with index usage information.

```
ALTER INDEX index NOMONITORING USAGE
```

1.14.5 Index related data dictionary views

DBA_INDEXES
DBA_IND_COLUMNS

1.15 Constraints

Types of constraints – not null, check, unique, primary key, foreign key

```
ALTER TABLE table MODIFY column NOT NULL
ALTER TABLE table MODIFY column NULL
```

```
ALTER/CREATE TABLE table (….)
```

CONSTRAINT ck_bonus check (bonus > 0)

ALTER TABLE table ADD CONSTRAINT ck_bonus check (bonus > 0)

CREATE TABLE product (….)
CONSTRAINT pk_prodcode PRIMARY KEY (prodcode)

ALTER TABLE product ADD CONSTRAINT pk_prodcode PRIMARY KEY (prodcode)

ALTER TABLE product ADD CONSTRAINT fk_mfd FOREIGN KEY (mfdby) REFERENCES manufacturer(mfdno) ON DELETE CASCADE

Constraints created are enabled by default. You can create it as disabled by using DISABLED keyword at end of command.

ALTER TABLE table DROP CONSTRAINT constraint
ALTER TABLE table DROP PRIMARY KEY CASCADE

1.15.1 Enabling/disabling constraints

ALTER TABLE table DISABLE CONSTRAINT constraint
ALTER TABLE table ENABLE CONSTRAINT constraint

ALTER TABLE table MODIFY CONSTRAINT constraint ENABLE
ALTER TABLE table MODIFY CONSTRAINT constraint DISABLE

1.15.2 Validated constraints

Enable validate – default, existing rows and future rows are checked
Enable novalidate – existing rows not checked but future rows are checked
Disable validate – existing rows checked but future rows are not checked (no DML is allowed on table)
Disable novalidate – no check done on existing or future rows

ALTER TABLE table MODIFY CONSTRAINT constraint ENABLE NOVALIDATE

1.15.3 Deferring constraints

If constraint is created with DEFERABLE clause, you can define whether constraint checking will be done immediately (INITIALLY IMMEDIATE, default) or later (INITIALLY DEFEREED).

```
ALTER TABLE name MODIFY CONSTRAINT cons_name INITALLY
DEFFERED (or IMMEDIATE)
```

1.15.4 Index related data dictionary views

DBA_CONSTRAINTS
DBA_CONS_COLUMNS

1.16 Users and security

1.16.1 Profile

Profiles are used to control database resource usage. DEFAULT profile is created at creation time of database. To enforce resource limit, RESOURCE_LIMIT=TRUE should be set in initialization file.

```
CREATE PROFILE OFFICE_USER LIMIT
SESSIONS_PER_USER          6
CONNECT_TIME               1440
IDLE_TIME                  120
FAILED_LOGIN_ATTEMTS  3
PASSWORD_LOCK_TIME    UNLIMITED
```

```
ALTER USER clerk PROFILE office_user
```

1.16.2 Users

```
CREATE USER MKM
IDENTIFIED BY MKM
DEFAULT TABLESPACE SUPERMARKET_DATA
TEMPORARY TABLESPACE TEMP
QUOTA UNLIMITED ON SUPERMARKET_DATA
PROFILE DEFAULT
```

```
GRANT CONNECT, RESOURCE, SELECT_CATALOG_ROLE,
EXECUTE_CATALOG_ROLE TO MKM
```

DROP USER mkm CASCADE

1.16.3 Privilege

Privileges control what users can or can't do in database.

Object privilege – provides permission to access schema objects. Granted for specific objects.

GRANT SELECT, UPDATE ON product, price TO clerk (WITH GRANT OPTION)

System privilege – provide right to perform structural change in database level.

GRANT CREATE ANY TABLE TO john (WITH ADMIN OPTION)

REVOKE CREATE ANY TABLE FROM john

For object privileges, both grantor and grantee information is stored in data dictionary; where as for system privilege, only grantee information is stored.

1.16.4 Roles

A role is named set of privileges.

CREATE ROLE CLERK
GRANT SELECT, INSERT, UPDATE ON TRANSACTION TO CLERK
ALTER USER john DEFAULT ROLE CLERK (NONE)

1.16.5 User related data dictionary views

DBA_USERS
DBA_TS_QUOTA – space assigned to users
V$SESSION – users currently connected to database

DBA_TAB_PRIVS
DBA_COL_PRIVS
DBA_SYS_PRIVS
SESSION_PRIVS

DBA_ROLES
DBA_ROLES_PRIVS
ROLE_ROLE_PRIVS

2 Backup and Recovery

2.1 Introduction

When instances crashes for any reason (e.g. Power failure) Oracle automatically recovers when database starts next time.

The most important decision you need to take for backup is to decide whether the database will run in archive or no archive log mode. Usually production databases *always* run in archive log mode.

Ideally, each member of archive log groups should reside in different physical disks so that if one disk gets corrupt, identical copies can be retrieved from another disk.

Backup can be either user managed (copying files with OS commands) or server managed (RMAN – Recovery Manager based). If you use RMAN, you can have incremental backups (i.e. backing up only changed blocks since last backup). RMAN is not discussed in this book.

2.2 Backup and recovery in no archive log mode

When the database is running in no archive log mode, only cold back up can be taken. The steps are –

1. Shutdown the database
2. Copy data files, control files, redo log files using OS commands to a backup location.
3. Startup database

If any of the files (data/control/redo) is corrupt in no archive log mode, you need to restore all files from backup. If relocation of files is necessary, you need to open the database in mount stage and issue rename command to specify new location of the files. Then you should open database for normal use. In no archive log mode, you can only recover data, which was taken during back up period. Even if only one file is corrupt, you need to restore all files and recover entire database.

2.3 Backup in archive log mode

In archive log mode, backup can be taken while database is running.

To take backup of data file, follow these steps –

1. ALTER TABLESPACE name BEGIN BACKUP
2. Use OS commands to copy data file
3. ALTER TABLESPACE name END BACKUP

Data file headers don't get updated until backup ends. Ideally, perform hot backup when there is less DML is occurring in the database.

2.4 User managed complete recovery

In all the following cases, it is assumed that database is running in archive log mode and backup was already taken.

2.4.1 System table space lost/corrupt

1. STARTUP MOUNT
2. Restore only system table space file
3. If required, relocate file – ALTER DATABASE RENAME file TO new file
4. RECOVER AUTOMATIC DATABASE (or DATAFILE file name)
5. ALTER DATABASE OPEN

2.4.2 Non-system table space lost/corrupt

1. STARTUP MOUNT
2. Make data file offline – ALTER DATABASE file name OFFLINE
3. ALTER DATABASE OPEN
4. Restore data file from backup to database location
5. If required, relocate file – ALTER TABLESPACE table space name OFFLINE IMMEDIATE for issuing check point
6. RECOVER AUTOMATIC TABLESPACE table space name
7. ALTER TABLESPACE table space name ONLINE

2.4.3 Non-system table space lost/corrupt when no backup taken

1. STARTUP MOUNT
2. ALTER DATABASE CREATE DATAFILE filename with path AS new filename (only if relocate)

3. RECOVER DATABASE (DATAFILE filename)
4. ALTER DATABASE OPEN

2.4.4 Instance failed during backup

Entire database won't start. Data file header freezes during backup, so no checkpoint information is written.

1. STARTUP MOUNT
2. ALTER DATABASE END BACKUP
3. ALTER DATABASE OPEN

2.5 User managed incomplete recovery

2.5.1 Time/Cancel/Change (SCN) based

1. STARTUP MOUNT
2. Restore all data files (including system, data, index, undo but not control file or redo log files)
3. RECOVER DATABASE UNTIL TIME 'dd-mon-yyyy hh24:mi:ss' (or CANCEL or CHANGE number)
4. ALTER DATABASE OPEN RESETLOGS

Log sequence number will be re-initialized.

SHUTDOWN
Take full backup of database

2.6 Logical backup – export/import and Data Pump

Export may be either through conventional path or direct path. In direct path export, the evaluating buffer is bypassed and makes it faster.

Most common export command is (from OS command prompt) –
EXP 'sys/password@mydb as sysdba'

Most common parameters are –
FILE – output file (default expdat.dmp)
DIRECT – direct path (N)
OWNER – list of owners/users

TABLES – list of table names
PARFILE – parameter file name (in this file you can store all options)
TABLESPACES – list of table spaces to export
TRANSPORT_TABLESPACE – export transportable table space metadata (N)

Similarly, import is run as –
IMP 'sys/password@mydb as sysdba'
FILE – input file (expdat.dmp)
IGNORE – ignore create object errors (N)
ROWS – import data rows (Y)

Example of export,
exp system/*password* file=F:\Database\exp_mkm.dmp owner=mkm rows=y

You can run export/import in interactive mode i.e. it will ask you about parameters during runtime.

Data pump

From 10g onward, Oracle introduces new utilities EXPDP and IMPDP (as run from OS command prompt) to fast data loading (export/import). As it uses API, it is significantly faster than export/import utility. You can even monitor data pump progress from data dictionary views.

Data pump is run as – IMPDP user/password DIRECTORY=data_pump_dir DUMPFILE=dpdump.dmp JOB_NAME=my_import

Where, DIRECTORY is an external directory name already created. EXPDP has similar syntax.

Oracle automatically creates data_pump_dir in $ORACLE_HOME/ admin/ database_name/ dpdump. However if you want you can create a directory of your own like this –

```
create directory data_pump_dir as
'C:\Oracle11g\admin\MYDB\dpdump';
```

You can view location of directory by
```
select * from dba_directories where DIRECTORY_NAME = 'DATA_PUMP_DIR'
```

Example:
```
expdp system/password DIRECTORY=data_pump_dir
DUMPFILE=supermarket.dmp SCHEMAS=mkm
```

Oracle now advises use of data pump over export/import.

2.7 SQL Loader

Using SQL Loader, you can load data from text file to Oracle tables.

The components you need to run SQL Loader are –

1. Control file – specifies how to load data in Oracle
2. Data file – the data in text file which will be loaded
3. Discard file – data that is discarded by SQL Loader because of not matching load condition
4. Bad file – data that SQL Loader could not load because of error
5. Log file – synopsis of loading operation

If you specify DIRECT=Y option, SQL Loader will bypass buffer and save data directly into data blocks in the disk. It makes data loading very fast however, this option does not enforce constraints, does not fire insert triggers, does not allow SQL functions in control file and locks entire table during loading.

You can run SQL Loader from OS command prompt as –

```
SQLLDR user/password@db CONTROL=myfile.ctl
```

A sample control file is shown below (assuming input data file is | delimited).

```
LOAD DATA
INFILE 'F:\DUMP\CUSTOMER.TXT'
BADFILE 'F:\DUMP\CUSTOMER.BAD'
DISCARDFILE 'F:\DUMP\CUSTOMER.DSC'
APPEND
INTO TABLE CUSTOMER
FIELDS TERMINATED BY "|"
TRAILING NULLCOLS
(
CUSTNO          DECIMAL EXTERNAL "sq_cust_no.nextval",
CUSTNAME        CHAR,
SEX             CHAR,
ADDRESS         CHAR,
CITY            CHAR,
STATE           CHAR,
COUNTRY         CHAR,
POSTCODE        CHAR,
PHONE           CHAR,
EMAIL           CHAR,
NOTE            CHAR,
DATESTAMP       DATE "YYYY-MM-DD HH24:MI:SS”
)
```

Note for NUMBER columns you specify DECIMAL EXTERNAL and for VARCHAR2 columns you specify CHAR in SQL Loader control file.

2.8 Networking fundamentals

Oracle Net is an internal layer which manages communication between client and server. It is configured on server, client, web server etc.

2.8.1 Configuring Oracle Net on server

Listener is a server-side networking component, which listens for requests from client on server. To communicate with Oracle server, Listener service must be started on server. Listener is configured in server using listener.ora file.

Following is a sample listener.ora file.

```
# LISTENER.ORA Network Configuration File:
E:\oracle\ora92\network\admin\listener.ora
# Generated by Oracle configuration tools.

LISTENER =
  (DESCRIPTION_LIST =
    (DESCRIPTION =
      (ADDRESS_LIST =
        (ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC0))
      )
      (ADDRESS_LIST =
        (ADDRESS = (PROTOCOL = TCP)(HOST = ensel)(PORT =
1521))
      )
    )
  )

SID_LIST_LISTENER =
  (SID_LIST =
    (SID_DESC =
      (GLOBAL_DBNAME = MDB)
      (ORACLE_HOME = E:\oracle\ora92)
      (SID_NAME = MDB)
    )
  )
```

You can manage listener from OS command prompt using LSNRCTL utility.

Oracle Net Manager is a tool using which you can manage most client/server configuration files.

2.8.2 Configuring Oracle Net on client

Usually you connect to Oracle server from client as USER/PASSWORD@DB (this is known as connect descriptor).

How the database name you specified in connection is resolved to exact database in the server is known as name resolution method.

Most popular name resolution methods are – host naming, local naming (most common using tnsnames.ora) and Oracle Internet Directory naming.

In tnsnames.ora file (usually in client machine it resides in $ORACLE_HOME / network/admin folder). A sample tnsnames.ora file is shown below.

```
# TNSNAMES.ORA Network Configuration File:
E:\oracle\ora92\network\admin\tnsnames.ora
# Generated by Oracle configuration tools.

MDB =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = ensel)(PORT = 1521))
    )
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = MDB)
    )
  )

MARKET =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = 109.125.257.250)(PORT
= 1521))
    )
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = MARKET)
    )
  )
```

Most common connection problem is “ORA-12154 TNS could not resolve service name”. When this occurs, check the client is looking at correct tnsnames.ora file (there may be multiple version in computer). You can verify this by checking TNS_ADMIN environment variable. Also check whether client computer can talk to server computer by running “TNSPING server ip address” command. Or you can use PING SID command as well.

Check sqlnet.ora file and see whether it specifies local naming first. This file resides in both client and server.

Sample of what an sqlnet.ora file will contain.
NAMES.DIRECTORY_PATH= (TNSNAMES, ONAMES, HOSTNAME)

2.9 Log Miner

Using log miner, you can examine redo log files!

2.9.1 Running log miner

Specify a directory by `UTL_FILE_DIR=E:\ORACLE\ORADATA\UNLOAD` parameter and bounce database.

Create a dictionary file

```
EXECUTE DBMS_LOGMNR_D.BUILD
('dictionary.ora','E:\ORACLE\ORADATA\UNLOAD', OPTIONS =>
DBMS_LOGMNR_D.STORE_IN_FLAT_FILE);
```

Add log files

```
exec dbms_logmnr.add_logfile ('E:\oracle\oradata\MDB\Archive1\
ARC00026.001', dbms_logmnr.new);
exec dbms_logmnr.add_logfile ('E:\oracle\oradata\MDB\Archive1\
ARC00027.001', dbms_logmnr.addfile);
exec dbms_logmnr.add_logfile ('E:\oracle\oradata\MDB\Archive1\
ARC00028.001', dbms_logmnr.addfile);
exec dbms_logmnr.add_logfile ('E:\oracle\oradata\MDB\Archive1\
ARC00029.001', dbms_logmnr.addfile);
```

Start log miner session

```
exec dbms_logmnr.start_logmnr (dictfilename=>
'E:\ORACLE\ORADATA\UNLOAD\dictionary.ora');
```

Once the redo logs were analyzed, all the DDL (and some DML) statements applied in the source database will be found in the V$LOGMNR_CONTENTS view. Important columns of this view are - SQL_UNDO, SQL_REDO, USERNAME, SCN, TIMESTAMP, COMMIT_TIMESTAMP, TABLESPACE, SEG_NAME, SEG_TYPE, and OPERATION.

The Log Miner session is closed by executing following command in the same session.

```
EXEC DBMS_LOGMNR.END_LOGMNR
```

After the session is complete, all data in the v$logmnr_contents table are deleted. Be sure to execute CREATE TABLE my_logmnr AS SELECT ... to copy the data before analyzing the contents.

2.9.2 Filtering data that is returned

Log Miner can potentially be dealing with large amounts of information. There are several methods you can use to limit the information that is returned to the V$LOGMNR_CONTENTS view, as well as the speed at which it is returned. These options are specified when you start LogMiner.

Showing Only Committed Transactions

At the time of starting

```
EXECUTE DBMS_LOGMNR.START_LOGMNR(OPTIONS =>
DBMS_LOGMNR.COMMITTED_DATA_ONLY);
```

Filtering Data By Time

```
EXECUTE DBMS_LOGMNR.START_LOGMNR(DICTFILENAME =>
'/oracle/dictionary.ora', STARTTIME => TO_DATE('01-Jan-2004
08:30:00', 'DD-MON-YYYY HH:MI:SS'), ENDTIME => TO_DATE('01-
Jan-2004 08:45:00', 'DD-MON-YYYY HH:MI:SS'));
```

Filtering Data By SCN

```
EXECUTE DBMS_LOGMNR.START_LOGMNR(DICTFILENAME =>
'/oracle/dictionary.ora', STARTSCN => 100, ENDSCN => 150);
```

2.9.3 Querying on log miner

Some examples are shown below.

```
SELECT USERNAME, SQL_REDO FROM V$LOGMNR_CONTENTS
```

```
WHERE USERNAME <> 'SYS';

SELECT OPERATION, SQL_REDO, SQL_UNDO FROM V$LOGMNR_CONTENTS
WHERE SEG_OWNER = 'MKM' AND SEG_NAME = 'PRODUCT' AND OPERATION
= 'INSERT' AND USERNAME = 'MKM';
```

2.9.4 Log miner related data dictionary views

V$LOGMNR_CONTENTS - Shows changes made to user and table information.

V$LOGMNR_DICTIONARY - Shows information about the Log Miner dictionary file, provided the dictionary was created using the STORE_IN_FLAT_FILE option.

V$LOGMNR_LOGS - Shows information about specified redo logs. There is one row for each redo log.

V$LOGMNR_PARAMETERS - Shows information about optional Log Miner parameters, including starting and ending system change numbers (SCNs) and starting and ending times.

2.10 Real Application Clusters (RAC)

An RAC database is a clustered database. In RAC environment, multiple Oracle instances (which are connected via *interconnect*) operate on same database files on disk. Different servers must access set of shared disks and should communicate through high-speed interconnection. Changed database blocks and log information on one instance can be moved to another instance through memory interconnection without writing on disk. This is known as *cache fusion* (see later). The main advantage of RAC is scalability and high availability. Several servers with Oracle instances can be added to system. If one of them fails, others continue to work without affecting operations. For this reason RAC is also known as High Availability. When work load grows, you can simply add another server to the grid (RAC is a type of grid computing after all).

The data files are stored in several disk drives which are connected by *cluster aware storage*. By adding multiple instances, we can add and remove and single Oracle instance without bringing the database down. So, database always remains available. That is the essence of RAC - high availability.

An RAC database is managed by *Oracle Clusterware*. RAC operates in *shared everything* architecture mode.

Any storage (eg. SAN, SCSI etc.) can be used with RAC. However, good I/O speed is required for scalability of data volume.

RAC supports up to 100 clusters which may be different at hardware level but must run same operating system.

Oracle recommends using ASM (Automatic Storage Management) for ease of dealing with clustered storage.

An Oracle RAC database requires three components - cluster nodes (the servers or computers running Oracle instances), shared storage (disk drives) and Oracle Clusterware (software application).

Installing RAC

The first step of working with RAC is to install "Oracle Clusterware" which is installed via Universal Installer.

Then you have to configure the clusterware.

Then install ASM (Automatic Storage Management).

Now install Oracle 11g database.

Then perform post installation tasks. This ensures that clusterware and database are installed properly and they are aware of each other.

It is possible to convert your normal single instance Oracle database to an RAC database. You can achieve this via Enterprise Manager or *rconfig* utility.

Administering Clusterware

Oracle Clusterware includes two important components: the voting disk and the OCR. The voting disk is a file that manages information about node membership, and the OCR is a file that manages cluster and Oracle RAC database configuration information.

RAC can be administered via Oracle Enterprise manager. On EM's web console mode, click on Availability tab to see details of Clusterware. You can click on Topology tab to see a visual representation of your nodes. The Interconnect tab shows you info on interfaces. You can also add new instance to clusterware via EM (under Server tab).

Oracle Clusterware posts alert messages in alert log – which is under $CRS_home. RAC data dictionary views are created by catclust.sql.

Cache Fusion

Oracle RAC uses *Cache Fusion* to synchronize the data stored in the buffer cache of each database instance – i.e. to keep track of which nodes are writing to which blocks and to ensure that two nodes do not update duplicate copies of the same block. Since all computers/instances in an RAC access the same database, the overall system must guarantee the coordination of data changes on different computers such that whenever a computer queries data it receives the current version – even if another computer recently modified that data. Oracle RAC refers to this functionality as Cache Fusion. It involves the ability of RAC to *fuse* in-memory data cached physically separately on each computer into a single global cache. This saves a lot of resource time.

2.11 Standby database (also Oracle Data Guard)

Stand by database is used as a protection against failure of main production database.

2.11.1 Physical stand by

This may be created from last back up taken for the production database and there after applying archived redo log files on it. This is similar to recovering a database. However, for stand by database, this recovery is like a continuous process, because, archived logs (from production database) are continuously being applied on it to make it sync with production database. However, stand by database won’t be available for use until recovery is completed.

2.11.2 Logical stand by

Instead of applying archived redo log files, SQL statement is constructed from log files are being applied to stand by database (similar to Log Miner method). The database is available for use during application of SQLs.

2.12 Replication

It allows you to have multiple copies of same database at different locations. At most basic level, replication is implemented using materialized views. In case of synchronous replication, transactions are either successfully propagated to all sites or it is rolled back. It ensures no conflict will occur between sites. It requires stable network environment and mostly used for read only materialized views. In asynchronous replication, changes to each site is stored locally and forwarded to other sites. In this case, conflict resolution processes are required.

Replication generally improves performance across network. For example, if you use a database for online transactions as well as reporting, you can replicate the database and run transactions and reporting on different versions.

3 Performance Tuning

3.1 Tuning development and production systems

Oracle's top down approach for development systems

1. Tune data design
2. Tune application design
3. Tune memory allocation
4. Tune I/O and physical structures
5. Tune resource contention
6. Tune for underlying platform (OS)

Oracle's production performance tuning principles

1. Define problem clearly and formulate a tuning goal
2. Examine host system and gather statistics
3. Compare common problems with Oracle's documentation
4. Get a conceptual picture of what went wrong from gathers statistics
5. Identify changes to be done and implement those changes
6. Check whether tuning objective has been met or not, repeat steps until tuning is complete.

3.2 Sources of tuning information

3.2.1 Alert log file

This file records information and error messages for various database activities. This file is located at BACKGROUND_DUMP_DEST folder. The name format is alert_SID.log.

3.2.2 Trace files

Background process trace files contains session information for process that created them. These files are available in BACKGROUND_DUMP_DEST folder. Name format for the trace file is usually SID_PROCESS_nnnn.trc e.g. mdb_lgwr_1598.trc.

User trace files are found in USER_DUMP_DEST folder. These are created when error occurs in user's server process. User tracing can be enabled using

ALTER SESSION SET SQL_TRACE = TRUE command or by executing package as
EXEC
SYS.DBMS_SYSTEM.SET_SQL_TRACT_IN_SESSION
(sid, serial#, TRUE)

3.2.3 Views commonly used in tuning

V$SGASTAT
V$EVENT_NAME
V$SYSTEM_EVENT
V$SESSION_EVENT
V$SESSION_WAIT
V$STATNAME
V$SYSSTAT
V$SESSTAT
V$SESSION
V$WAITSTAT

DBA_TABLES
DBA_INDEXES
DBA_STATS
DBA_DATA_FILES
DBA_SEGMENTS
DBA_HISTOGRAMS

3.3 Collecting statistics

Run UTLBSTAT and UTLESTAT to collect all database activities in the given time period in a single file.

Run from $ORACLE_HOME/rdbms/admin/utlbstat.sql and utlestat.sql
The resulting file is named as REPORT.TXT

3.3.1 Statspack

- Run spcreate.sql script from $ORACLE_HOME/rdbms/admin folder.
- It will create PERFSTAT schema with all required objects.
- To collect statistics, use EXECUTE STATSPACK.SNAP procedure (as PERFSTAT user). A snapshot of statistics will be collected and stored in PERFSTAT schema tables.

- Use $ORACLE_HOME/rdbms/admin/spauto.sql to run automatic statistics collection at specific intervals.
- Once you gathered enough statistics, you can generate a report using $ORACLE_HOME/rdbms/admin/spreport.sql script.

3.4 Oracle supplied GUI tuning tools

Capacity Planner
Performance Manager – observe database performance.
Top Sessions – see which users are consuming most resource
Trace Data Viewer
Lock Monitor
Top SQL – most resource consuming SQLs
Performance Overview – see current performance of database
Oracle Expert – gathers statistics and gives recommendations for tuning
Index Tuning Wizard – identify unused indexes

3.5 SQL application tuning and design

3.5.1 TKPROF

It is used to format user trace files. Usage example is (from OS command prompt) – TKPROF ORA_1234.TRC TRACE.TXT
Some of its parameters are –
EXPLAIN – generates explain plan for each statement in trace file
SYS – output file includes recursive SQL statements (i.e. those involving data dictionary queries)
RECORD – specifies a file where SQL statements of trace file are written

Example usage
TKPROF ORA_1234.TRC TRACE.TXT SYS=NO
EXPLAIN=MKM/MKM@MDB RECORD=SQL.TXT

To identify SQL statements, which may require tuning, look for statements –

- Consuming excess CPU resource
- Taking long time to parse, execute and fetch
- Reading too many data blocks from disk and too few from SGA
- Access many data blocks but return only few rows

3.5.2 Explain plan

- Create plan table using $ORACLE_HOME/rdbms/admin/utlxplan.sql
- Populate PLAN_TABLE using EXPLAIN PLAN FOR… command. For example,
  ```
  EXPLAIN PLAN
  SET STATEMENT_ID = 'PRICE'
  FOR
  SELECT P.PRODCODE, P.PRODNAME, J.SELLPRICE, M.MFDNAME,
         NVL(SUM(I.QTYAVAILABLE),0) "ALL STORES QTY"
  FROM PRODUCT P
  JOIN PRICE J ON (P.PRODCODE = J.PRODCODE)
  LEFT JOIN MANUFACTURER M ON (P.MFDBY = M.MFDNO)
  LEFT JOIN INVENTORY I ON (P.PRODCODE = I.PRODCODE)
  WHERE SYSDATE BETWEEN J.STARTDATE AND J.ENDDATE
  AND P.HASSERIALNO=0
  GROUP BY P.PRODCODE, P.PRODNAME, J.SELLPRICE, M.MFDNAME
  ```

- To view the plan, you may issue following statement

  ```
  SELECT LPAD(' ',4*(LEVEL-2)) || OPERATION || ' ' ||
  OPTIONS || ' ' ||
  OBJECT_NAME "EXECUTION_PLAN", IO_COST,
  CPU_COST,TEMP_SPACE
  FROM PLAN_TABLE
  START WITH ID = 0 CONNECT BY PRIOR ID = PARENT_ID
  ```

- Remember that, if you run the above query, you will see the plan in an indented view. The innermost operations are executed first. If two operations appear at same level (with same inner level), the top one is executed first.

3.5.3 Auto trace

Unlike explain plan, auto trace executes the actual SQL statement before generating the plan.

You need to create PLUSTRACE role by running plustrce.sql as SYS user.
Then you need to assign PLUSTRACE role to users who will use auto trace. They also need PLAN_TABLE in their schemas.

To use auto trace, issue following command from SQL Plus.
SET AUTOTRACE ON
Then issue any SQL statement and you will always see statistics after result is displayed.

3.6 Optimizer

Earlier versions of Oracle used Rule Base Optimizer (RBO). But latest versions use Cost Based Optimizer (CBO) by default.

Statistics must be gathered to benefit from CBO.

3.6.1 Gathering statistics

ALTER INDEX index_name COMPUTE STATISTICS
ALTER TABLE table_name COMPUTE STATISTICS

ALTER TABLE table_name ESTIMATE STATISTICS FOR TABLES/COLUMNS col1, col2/ALL COLUMNS/ALL INDEXES

EXECUTE DBMS_UTILITY.ANALYZE_SCHEMA('MKM','COMPUTE')

Oracle now recommends that you use DBMS_STATS package instead of Analyze.

EXEC DBMS_STATS.gather_schema_stats (ownname => 'MKM', cascade =>true, estimate_percent => dbms_stats.auto_sample_size)

3.6.2 Optimizer modes

In initialization file, you can set optimizer mode as shown below OPTIMIZER_MODE=CHOOSE (this is default) or FIRST_ROWS or ALL_ROWS or RULE or FIRST_ROWS_n where n = 1/10/100/1000.

You can also change optimizer mode at session level.

ALTER SYSTEM SET OPTIMIZER_MODE=CHOOSE

To change optimizer mode at statement level, you should use hints using /*+ … */.

SELECT /*+ RULE */ …
Commonly used hints are – FULL, INDEX, REWRITE, and PARALLEL etc.

If statistics exists for any one table or index involved in SQL statement, CBO is used, otherwise RBO is used.

To reuse a saved execution plan, you can use “plan stability” (in the form of stored outline) feature or “materialized view”.

3.6.3 Materialized view

Unlike in normal view, materialized view actually stores the data in tables.

To create materialized view, issue

```
CREATE MATERIALIZED VIEW schema.mview_name
BUILD IMMEDIATE
REFRESH ON DEMAND (or REFRESH COMPLETE ON COMMIT)
ENABLE QUERY REWRITE
AS
SELECT query;

EXEC DBMS_MVIEW.REFRESH('schema.mview_name','C');
EXEC DBMS_MVIEW.REFRESH_ALL_DEPENDENT('table_name');
EXEC DBMS_MVIEW.REFERSH_ALL_MVIEWS;
```

3.6.4 Partitioned tables

3.6.4.1 Range partition

It uses range of column values to determine where the record will be inserted.

CREATE TABLE student (…) PARTITION BY RANGE (graduation_year)
(PARTITION p_2000 VALUES LESS THAN 2000 TABLESPACE student_1,
PARTITION p_2002 VALUES LESS THAN 2002 TABLESPACE student_2,
PARTITION p_2004 VALUES LESS THAN 2004 TABLESPACE student_3,
PARTITION p_error VALUES LESS THAN (MAXVALUE) TABLESPACE student_4);

3.6.4.2 List partition

It is based on set of specified value instead of range of values.

CREATE TABLE student (…) PARTITION BY LIST (degree)
(PARTITION p_engg VALUES (‘BTech’,’MTech’) TABLESPACE stu_engg,
PARTITION p_comm VALUES (‘Mcom’,’Bcom’) TABLESPACE stu_comm,
PARTITION p_arts VALUES (‘MA’,’BA’) TABLESPACE stu_arts);

3.6.4.3 Hash partition

It uses a hashing algorithm to assign records in particular partition. It usually keeps almost equal number of records in each partition.
CREATE TABLE student (…) PARTITION BY HASH (roll_no) PARTITIONS 3 STORE IN (stu_1, stu_2, stu_3);

3.6.4.4 Composite partition

It uses range partition and inside it uses hash sub-partitions. Data is physically stored in sub-partition level.

```
CREATE TABLE student (…)
PARTITION BY RANGE (graduation_year)
SUBPARTITION BY HASH (roll_no) SUBPARTITIONS 3
STORE IN (stu_1, stu_2, stu_3)
(PARTITION p_2000 VALUES LESS THAN 2000,
PARTITION p_2002 VALUES LESS THAN 2002,
PARTITION p_2004 VALUES LESS THAN 2004,
PARTITION p_error VALUES LESS THAN (MAXVALUE));
```

Note: Any bitmap indexes created on partitioned table must be local (to the partition).

3.6.4.5 Cluster

A cluster is a group of one or more tables whose data is stored at same place (physically). This helps faster access of data columns, which are often queried as joins because Oracle server needs to read less number of physical data blocks.

```
CREATE CLUSTER dept_emp
(dno NUMBER) SIZE 1000 TABLESPACE clst

CREATE INDEX dept_emp_idx
ON CLUSTER dept_emp TABLESPACE idx

CREATE TABLE dept (dno NUMBER, dname VARCHAR2(100))
CLUSTER dept_emp(dno);

CREATE TABLE emp (eno NUMBER, ename VARCHAR2(100), dno
NUMBER) CLUSTER dept_emp(dno);
```

Oracle 11g allows reference partition and interval partition.

Take an example of Transaction and Transaction Detail tables. These two are linked by foreign key TransId. Now, master table has transaction date but detail table do not. We want both tables be partitioned by transaction date. In 11g, we can specify details tables be partitioned based on foreign key reference TransId.

Interval partition is useful when we do not know how many partitions we need beforehand. In above example, we can specify the table such a way whenever new month begins, a new partition will be created automatically.

3.7 Tuning shared pool

Shared pool consists of library cache and data dictionary cache. Library cache caches most recently used SQL and PL/SQL statements. Data dictionary cache caches data dictionary information. Shared pool is managed by a Least Recently Used algorithm.

Finding a matching SQL statement in shared pool is known as cache hit. For cache hit to occur, two SQL statements must be exactly same i.e. their ASCII value equivalent should be same. The aim of tuning shared pool is to maximize cache hit ratio.

3.7.1 Measuring shared pool performance

High cache hit ratio indicates that your application users are getting results of SQL and PL/SQL mostly from memory rather than reading from disk.

```
SELECT * FROM V$LIBRARYCACHE
```

The value in GETHITRATIO column for SQL AREA, TABLE/PROCEDURE, BODY and TRIGGER rows should be very close to 1 i.e. 100%.

Similarly PINHITRATIO should also be very close to 1.

GET is referred to parse lock, while PIN is referred to execution time locks.

A well-tuned OLTP system should have GETHITRATIO and PINHITRATIO 90% or higher for SQL AREA.

Data dictionary performance is measured by

```
SELECT 1-(SUM(GETMISSES)/SUM(GETS)) "Data dictionary hit
ratio" FROM V$ROWCACHE
```

This value should be over 0.85 for OLTP systems.

3.7.2 Improving shared pool performance

3.7.2.1 Add more memory to shared pool

The size of shared pool is determined by SHARED_POOL_SIZE initialization parameter.

```
SELECT POOL, SUM(BYTES) "SIZE" FROM V$SGASTAT WHERE POOL =
'shared pool' GROUP BY POOL
```

Use following command to change shared pool size

```
ALTER SYSTEM SET SHARED_POOL_SIZE=200M
```

3.7.2.2 Make space for large PL/SQL statements

You can set aside a reserved area in SGA for large PL/SQL packages. This area is controlled by SHARED_POOL_RESERVED_SIZE parameter.

Determine which packages are loaded into memory from following command

```
SELECT * FROM V$DB_OBJECT_CACHE WHERE TYPE IN ('PACKAGE',
'PACKAGE BODY')
```

3.7.2.3 Keep important PL/SQL code in memory

You can “pin” most frequently used PL/SQL code in memory. To do this –

1. Build DBMS_SHARED_POOL package by running $ORACLE_HOME /rdbms/admin/dbmspool.sql script.
2. Load a package using EXECUTE DBMS_SHARED_POOL.KEEP(‘PACKAGE_NAME’)

You can see which packages are pinned by issuing

```
SELECT OWNER, NAME, TYPE FROM V$DB_OBJECT_CACHE WHERE KEPT =
'YES'
```

You must have very good knowledge of application to determine which objects to pin right after instance startup. You can audit PL/SQL packages, triggers, sequences etc. and find out which objects are most frequently accessed.

3.7.2.4 Code reuse

Use bind variables like SELECT * FROM CUSTOMER WHERE CUST_ID = :cust_id

Implement strict coding standard (i.e. capitalization, indentation etc.) so that chances of finding identical statement in memory will increase.

3.7.2.5 Tune cache specific initialization parameters

OPEN_CURSORS (default 50, increase if necessary)
CURSOR_SPACE_FOR_TIME (set to FALSE for Forms based applications)
SESSION_CACHED_CURSORS
CURSOR_SHARING (use FORCE, SIMILAR or default EXACT)

3.8 Tuning database buffer cache

It caches most recently accessed data blocks into memory. If data is not found from cache, it is fetched from disk. It is operated on LRU mechanism (except, only for full table scan, it is placed on MRU end.)

Buffer may be free, pinned, clean or dirty.

DBWn writes dirty (changed) buffer to disk.

3.8.1 Measuring buffer cache hit ratio

```
SELECT 1-(physical.VALUE - direct.VALUE - lobs.VALUE) /
(logical.VALUE) "Buffer cache hit ratio"
FROM v$sysstat physical, v$sysstat direct, v$sysstat lobs,
v$sysstat logical
WHERE physical.NAME = 'physical reads'
AND direct.NAME = 'physical reads direct'
AND lobs.NAME = 'physical reads direct (lob)'
AND logical.NAME = 'session logical reads'
```

The ratio should be more than 0.90 for OLTP systems.

3.8.2 Improving buffer cache hit ratio

3.8.2.1 Add more space

ALTER SYSTEM SET DB_CACHE_SIZE = 100M

Set DB_nK_CACHE_SIZE parameters in initialization file to access multi-block database.

To get an idea how much you should increase buffer cache size

- Set DB_CACHE_ADVICE = ON initialization parameter.
- Query V$DB_CACHE_ADVICE view.

3.8.2.2 Use multiple buffer pools

Use keep, recycle and default pool by setting DB_KEEP_CACHE_SIZE, DB_RECYCLE_CACHE_SIZE and DB_CACHE_SIZE (default) initialization parameters.

Determining which segments to cache in keep and recycle pools, you must have in depth knowledge of the application.

```
SELECT username "owner", NAME "seg name", kind "seg type",
COUNT(DISTINCT BLOCK#) "No. buffers"
FROM v$cache v, dba_users d
WHERE v.owner# = d.user_id
GROUP BY NAME, username, kind
HAVING COUNT(DISTINCT BLOCK#)>10
ORDER BY 3 DESC
```

Oracle recommends you consider caching of segments in keep pool whose total size is less than 10% of default pool size.

To assign segments to pool, use

ALTER TABLE schema.package STORAGE (BUFFER_POOL KEEP);

ALTER TABLE schema.package STORAGE (BUFFER_POOL RECYCLE);

Monitor buffer pool statistics

```
SELECT NAME "Buffer pool",
 1-(physical_reads/(db_block_gets+consistent_gets)) "Buffer
pool hit ratio"
```

```
FROM v$buffer_pool_statistics
ORDER BY NAME
```

3.8.2.3 Cache tables in memory

CREATE TABLE (…) CACHE

ALTER TABLE name CACHE

3.8.2.4 Use proper indexes

Build indexes on foreign key columns of tables that reference a primary key column in another table.

3.9 Shared server

By default, Oracle runs on dedicated server mode where for each user session connection, a server process is created on server and a user process is created on client machine. When number of concurrent users becomes very high, shared server (known as Multi Threaded Server in earlier versions) is used.

In shared server, several (up to 5) dispatcher background processes are run in server, which accept user requests. They then put the request in a queue. Thereafter, shared server processes pick the request from queue and process the SQL (i.e. parsing the statement, reading segments from disk and placing them in buffer cache etc.). After processing, they put the result in response queue. The corresponding dispatcher sends back the result to user process.

Shared server is useful for these scenarios – many application users, short application transactions (e.g. railway reservation system, order entry system), non-continuous transactions etc.

Shared server is a scalability improvement feature not performance improvement feature. You will always have same or better performance by using dedicated server feature.

3.9.1 Configuring shared servers

Set following parameters in initialization file

DISPATCHERS = n (0 to 5)

SHARED_SERVERS = n (1 to OS dependent max. value)

Other optional parameters are – MAX_DISPATCHERS, CIRCUITS, MAX_SHARED_SERVERS, and PROCESSES

3.9.2 Measuring shared server performance

Query on V$SHARED_SERVER_MONITOR view.

The following query shows shared server busy ratio

```
SELECT NAME, decode(busy+idle,0,0,round((busy/
(busy+idle))*100,4)) "busy rate" FROM v$shared_server WHERE
status != 'QUIT'
```

The next query shows dispatcher busy ratio

```
SELECT NAME "dispatcher", NETWORK, (round(SUM(busy)/
(SUM(busy)/SUM(idle)),4))*100 "busy rate"
FROM v$dispather GROUP BY NAME, NETWORK
```

Additional dispatcher is necessary is busy ratio found from above query is more than 50%.

Please note that DBA must connect through dedicated server to perform various DBA activities (e.g. startup/shutdown database).

3.10 Large pool and Java pool

Large pool in SGA is used for Recovery Manager (RMAN) operations and parallel query.

It is defined by LARGE_POOL_SIZE (usual values from 600K to 2GB) parameter in initialization file and cannot be changed dynamically.

Java pool is used to keep session specific Java application code and variables. Its size is determined by JAVA_POOL_SIZE (default 20M) initialization parameter and cannot be changed dynamically.

3.11 Tuning Redo

Its performance is measured by the amount of time server process waits to access redo log files. The following queries can be used to measure redo log waits.

```
SELECT r.VALUE/e.VALUE "Redo log buffer retry ratio"
FROM v$sysstat r, v$sysstat e
WHERE r.NAME = 'redo buffer allocation retries'
AND e.NAME = 'redo entries'
```

Its value should be less than 1%.

```
SELECT NAME,VALUE FROM v$sysstat WHERE NAME = 'redo log space
requests'
```

Constantly increase value indicates too small redo log files.

3.11.1 Improving redo log performance

Make it bigger. Redo log space is specified by LOG_BUFFER initialization parameter.

Reduce redo log generation by using UNRECOVERABLE and NOLOGGING keywords in SQL statements.

3.12 Tuning archiving operations

Common problems are –

- Archive location getting full – you can temporarily change archive log location by ALTER SYSTEM ARCHIVE LOG ALL TO '/oradata/archive'.
- ARCn process can't keep up with redo log generations – create more ARCn process.

3.13 Tuning disk I/O

Keep separate table spaces for different functional segments.

Measure data file I/O from following views – V$FILESTAT, V$DATAFILE, V$TEMPFILE.

3.14 Tuning sorts

Minimize sorting activity. If required, perform in memory whenever possible.

Sort may be measured by following query

```
SELECT m.VALUE/(d.VALUE+m.VALUE) "In memory sort ratio"
FROM v$sysstat m, v$sysstat d
WHERE m.NAME = 'sorts (memory)' AND d.NAME = 'sorts (disk)'
```

This ratio should be more than 0.95

Increase value for SORT_AREA_SIZE initialization parameter if necessary.

Disk sort is performed in temporary table space file.

3.15 Tuning rollback segments

From Oracle 9i, automatic undo management does away manual rollback segment management. However, you can set a specific rollback segment for large transaction by this command –

- CREATE PRIVATE ROLLBACK SEGMENT rbs STORAGE (INITIAL 1M NEXT 10M) TABLESPACE rbstb
- ALTER ROLLBACK SEGMENT rbs ONLINE
- SET TRANSACTION USE ROLLBACK SEGMENT rbs
- EXEC procedure
- ALTER ROLLBACK SEGMENT rbs ONLINE

3.16 Locks

Lock contention occurs when multiple users try to obtain lock on specific object at the same time. Lock is released only after COMMIT or ROLLBACK statement.

See lock contention from following views.

V$LOCK
V$LOCKED_OBJECT
DBA_WAITERS
DBA_BLOCKERS

To resolve lock contention, avoid coding long transactions and don't use restrictive explicit locks. It is always better to let Oracle handle all locks.

You can kill any session using ALTER SYSTEM KILL SESSION 'sid, serial#' command.

3.17 Tuning operating system

Try to avoid paging and swapping.

If server has multiple CPUs, use Oracle's parallel query, parallel DML, parallel ANALYZE and parallel index creation features. In multiple CPU machines, Oracle will itself change many parameters.

You can allocate and manage server resources using Oracle's Resource Manager feature (not discussed here).

On Unix, all server processes and background processes are run as separate executables (you can see them using ps command). However, on Windows, all those processes run as threads inside ORACLE*servicename*.EXE process.

3.18 Automatic Workload Repository (AWR)

From 10g, AWR continuously captures database statistics and derives metrics from it. This is of immense help during database tuning.

Among many others, AWR collects following statistics

- Wait events used to identify performance problems.
- Some system and session statistics from the V$SYSSTAT and V$SESSTAT views.
- Object usage statistics.
- Resource intensive SQL statements

AWR information is used by ADDM and SQL Tuning Advisor (discussed below)

AWR related package is DBMS_WORKLOAD_REPOSITORY. The DBA_HIST% views contain historical data stored in the database.

3.19 Automatic Database Diagnostic Monitor (ADDM) and SQL Tuning Advisor

From 10g, ADDM constantly monitors database and in case of any problem, it will suggest what you should do next. The ADDM findings can be queried from DBA_ADVISOR% objects on data dictionary. Note that ADDM analysis is based on AWR snapshots, which have a default frequency of once an hour and a default retention period of 8 days.

An ADDM analysis is performed after each AWR snapshot (every hour by default), and the results are saved in the database.

Remember, the goal of database performance tuning is to reduce the DB time of the system for a given workload.

4 SQL and PL/SQL

All source code of packages, procedures and functions are written in DBA_SOURCE, ALL_SOURCE or USER_SOURCE views.

4.1 Compiling packages

ALTER PROCEDURE procedure COMPILE
ALTER PACKAGE package COMPILE
ALTER PACKAGE package COMPILE BODY

4.2 Records and Tables

PL/SQL records are similar to C structures.

```
DECLARE
TYPE tEmp IS RECORD (
ENO  NUMBER, FNAME VARCHAR2(50), LNAME  VARCHAR2(50));
vEmp tEmp;
BEGIN
SELECT id, fname, lname INTO vEmp FROM employee;
END;
```

You can use %ROWTYPE to specify a variable of table row type. For example,
`vEmp employee%ROWTYPE;`

PL/SQL Tables are similar to 1-dimensional array in C. You can visualize it as a table with only 2 columns – KEY and VALUE.

```
DECLARE
TYPE tEmp IS TABLE OF VARCHAR2(50) INDEX BY BINARY_INTEGER;
vEmp tEmp;
BEGIN
vEmp(1) = 'Saikat';
END;
```

Some table attributes are – tablename.COUNT, DELETE, DELETE(i), FIRST, LAST, NEXT(i), PRIOR(i).

4.3 Cursor

Declaring cursor

CURSOR cursor_name IS SELECT statement;

Processing cursor

OPEN cursor_name;
FETCH cursor_name INTO variable(s);
CLOSE cursor_name;

Cursor attributes

%FOUND, %NOTFOUND, %ISOPEN, %ROWCOUNT

Cursor FOR loop

FOR variable IN cursor_name LOOP
Statement(s);
END LOOP;

4.4 Merge, multi – table insert and pivot insert

An example of merge command is shown below.

```
MERGE INTO MANUFACTURER M
USING NEW_MANUFACTURER WM
ON (M.MFDNO = WM.MFDNO)
WHEN MATCHED THEN UPDATE
   SET M.MFDNAME = WM.MFDNAME, M.ADDRESS = WM.ADDRESS, M.CITY = WM.CITY,
   M.STATE = WM.STATE, M.COUNTRY = WM.COUNTRY
WHEN NOT MATCHED THEN
   INSERT (M.MFDNO, M.MFDNAME, M.ADDRESS, M.CITY, M.STATE, M.COUNTRY)
   VALUES (WM.MFDNO, WM.MFDNAME, WM.ADDRESS, WM.CITY, WM.STATE, M.COUNTRY);
```

Similarly, an example of multi – table insert follows.

```
INSERT ALL
WHEN MEDIA = ‘BOOK’ THEN
 INTO BOOK VALUES (NO, TITLE, PRICE)
WHEN MEDIA = ‘CD’ THEN
 INTO SOFTWARE VALUES (NO, TITLE, PRICE)
WHEN MEDIA = ‘VCD’ THEN
 INTO VIDEO VALUES (NO, TITLE, PRICE)
SELECT MEDIA, NO, TITLE, MEDIA FROM PRODUCT;
```

Using pivot insert, you can create multiple rows of data from single record. Say, SALES_SOURCE_DATA comes from a non-relational source and it contains following columns – emp_id, sales_mon, sales_tue, sales_wed, sales_thu and sales_fri. We like to store this information in SALES_INFO table which has following fields – emp_id, week, sales. We can achieve it by using the statement shown below.

```
INSERT INTO SALES_INFO VALUES (emp_id, week, sales_mon)
INSERT INTO SALES_INFO VALUES (emp_id, week, sales_tue)
INSERT INTO SALES_INFO VALUES (emp_id, week, sales_wed)
INSERT INTO SALES_INFO VALUES (emp_id, week, sales_thu)
INSERT INTO SALES_INFO VALUES (emp_id, week, sales_fri)
SELECT emp_id, sales_mon, sales_tue, sales_wed, sales_thu, sales_fri FROM
SALES_SOURCE_DATA.
```

Thus for 1 row in SALES_SOURCE_DATA, we shall have 5 rows in SALES_INFO table. This feature is also known as normalization is some other database applications.

4.5 SQL Joins

Oracle now follows ANSI/ISO join syntax (similar to SQL Server). You can also use earlier versions join syntax. However, new ANSI/ISO standard syntax is more comprehensive. Here are few examples.

Say we have 2 tables – EMP (eno, ename, dno) and DEPT (dno, dname). To list all employee names and department names we can now write following query.

```
SELECT E.ENAME, D.DNAME FROM EMP JOIN DEPT ON (E.DNO = D.DNO)
```

We can use LEFT, RIGHT or FULL keywords for outer join. For example, to list all employees even when some employees may not belong to any department, we can issue this query.

```
SELECT E.ENAME, D.DNAME FROM EMP LEFT JOIN DEPT ON (E.DNO =
D.DNO)
```

This is definitely more intuitive than earlier syntax with "+" operator!

Example of updating columns of table with data from another table:

```
CREATE TABLE INFO1 (CODE VARCHAR2(3) PRIMARY KEY,
 COUNTRY VARCHAR2(100), CAPITAL VARCHAR2(100), PHONECODE NUMBER(4),
CURRENCYCODE VARCHAR2(3));

INSERT INTO INFO1 VALUES ('IND','India','New Delhi',NULL,NULL);
INSERT INTO INFO1 VALUES ('GBR','United Kingdom','London',NULL,NULL);
INSERT INTO INFO1 VALUES ('FRA','France','Paris',NULL,NULL);
INSERT INTO INFO1 VALUES ('USA','United States','Washington DC',NULL,NULL);
INSERT INTO INFO1 VALUES ('ITA','Italy','Rome',NULL,NULL);
INSERT INTO INFO1 VALUES ('SWE','Sweden','Stockholm',46,'SEK');
COMMIT;

CREATE TABLE INFO2 (CODE VARCHAR2(3), PHONECODE NUMBER(4), CURRENCYCODE
VARCHAR2(3) );

INSERT INTO INFO2 VALUES ('IND',91,'INR');
INSERT INTO INFO2 VALUES ('GBR',44,'GBP');
INSERT INTO INFO2 VALUES ('FRA',33,'EUR');
INSERT INTO INFO2 VALUES ('USA',1,'USD');
INSERT INTO INFO2 VALUES ('CAN',1,'CAD');
COMMIT;

/* case 1 - wrong */
UPDATE INFO1 i1 SET (PHONECODE,CURRENCYCODE) = ( SELECT PHONECODE,
CURRENCYCODE FROM INFO2 i2 WHERE i1.CODE = i2.CODE )
-- Sweden's data becomes null by running this!!

/* case 2 - correct */
UPDATE INFO1 i1 SET (PHONECODE,CURRENCYCODE) = ( SELECT PHONECODE,
CURRENCYCODE FROM INFO2 i2 WHERE i1.CODE = i2.CODE )
WHERE EXISTS ( SELECT 1 FROM INFO2 i22 WHERE i1.CODE = i22.CODE)
-- Sweden's existing data remains correct
```

4.6 Useful SQL functions

Decode – some examples have been provided in chapter “Useful scripts for DBAs”.

Case – example

```
SELECT COUNTRY, CONTINENT
CASE CONTINENT    WHEN 1 THEN ‘EUROPE’
                  WHEN 2 THEN ‘ASIA’
                  WHEN 3 THEN ‘AFRICA’
                  ELSE ‘OTHER’ END CONTINENT_NAME
FROM COUNTRIES
```

With clause example

```
WITH
x AS
(
    select 'Tom' as FirstName FROM dual union
    select 'Dick' FROM dual union
    select 'Barney' FROM dual union
    select 'Betty' FROM dual
),
z AS
(
    select 'Flintstone' as LastName FROM dual  union
    select 'Rubble' FROM dual
)
SELECT * FROM x,z
```

NVL (x, y) returns y if x is null or x otherwise.

NVLS (x, y, z) returns z if x is null, y if x is not null.

INSTR (string where to search, string what to search, start position default 1, what “n-th” occurrence default 1st) is used for searching for pattern inside a string. Returns the number position where occurrence has been found or 0 if not found.

SUBSTR (string, x, y) returns portion of string that is y characters long starting from position x.

ROUND (15.2568, 2) produces 15.26.

Pivot and Un-pivot examples – can be used to transpose rows/columns.

```
/* PIVOT */
WITH fly_table AS (
           SELECT 'John' "Customer",'Apple' "Item", 5 "Qty" FROM dual UNION
ALL
           SELECT 'Jill','Orange', 2 FROM dual UNION ALL
           SELECT 'Sally','Banana', 6 FROM dual UNION ALL
           SELECT 'John','Orange', 2 FROM dual UNION ALL
           SELECT 'Sally','Apple', 1 FROM dual UNION ALL
           SELECT 'John','Orange', 1 FROM dual UNION ALL
           SELECT 'Sally','Apple', 1 FROM dual
           )
SELECT * FROM (
SELECT "Customer","Item", SUM("Qty") "Qty" FROM fly_table GROUP BY
"Customer", "Item"
) PIVOT ( SUM("Qty") FOR "Item" IN ('Apple','Banana','Orange'))


/* UNPIVOT */
WITH fly_table AS (
           SELECT 'India' "Country",'New Delhi' "Capital" FROM dual UNION
           SELECT 'UK','London' FROM dual UNION
           SELECT 'USA', 'Washington DC' FROM dual UNION
           SELECT 'Germany','Berlin' FROM dual
           )
SELECT COL, VALUE FROM (
SELECT * FROM fly_table
)
UNPIVOT INCLUDE NULLS ( VALUE FOR COL IN ("Country","Capital"))
```

4.7 Dynamic PL/SQL

A sample is given below

```
CREATE TABLE TEST.EMP(NO NUMBER, NAME VARCHAR2(100));

CREATE OR REPLACE PROCEDURE DynSql
(pNo   IN test.emp.no%TYPE,
 pName IN test.emp.NAME%TYPE)
```

```
AS
cur    INTEGER;
p      INTEGER;
stmt   VARCHAR2(100);
BEGIN
     cur := dbms_sql.open_cursor;
     stmt := 'INSERT INTO TEST.EMP VALUES(:no,:name)';
     dbms_sql.parse(cur,stmt,dbms_sql.v7);

     dbms_sql.bind_variable(cur,':no',pNo);
     dbms_sql.bind_variable(cur,':name',pName);

     p := dbms_sql.EXECUTE(cur);

     COMMIT;

     dbms_output.put_line(p);
     dbms_sql.close_cursor(cur);
END;
```

Run the procedure from SQL Plus as `EXEC DYNSQL(1,'Saikat');`

It is also possible to execute dynamic SQL using EXECUTE IMMEDATE as shown below.

```
EXECUTE IMMEDIATE('INSERT INTO MY_TABLE SELECT * FROM
THAT_TABLE');
```

4.8 Nested table

```
create type objAddress as object
(HouseNo varchar2(20),
Street varchar2(200),
City varchar2(50),
State char(2),
PostCode varchar2(10))
/

create type oAddress as table of objAddress
/

create table People
(No number,
Name varchar2(100),
Address oAddress)
nested table Address store as tbAddress;

insert into people values (1,'Saikat Basak',
oAddress(objAddress('100','East West
Street','Kolkata','WB','700005')))
```

```
/
```

4.9 Types of joins

Suppose we are joining two tables in Oracle.

```
SELECT T1.X, T2.Y FROM T1 JOIN T2 ON T1.X = T2.X
```

Assume, T1 is driving table (i.e. more rows than T2)

Nested Loop join

This is executed like this:

```
For each record in T1
    find matching record in T2 where T1.X = T2.X
    fetch that record into result set
```

Now if there is no index on column X in T2, Oracle will perform full table scan for each record in T1.

Hash join

Oracle will load create an in-memory index (known as hash table) for T2 (i.e. smaller table)

It will still perform similar operation

```
for each record in T1
   find matching record in T2 where T1.X = T2.X
   fetch that record into result set
```

But even if there is no index on T2.X, Oracle will use the hash table which works similar to index!

Clearly this is suitable only when T2 is small otherwise [1] hash table may not fit in memory [2] creation of hash table may involve a long time (then we could have created the index on T2 in first place!)

Sorted Merge join

If both tables are very big, hash join is not suitable due to memory/time requirement and nested loop will be slow or will result in full table scan.

The alternative way to speed up performance is to use sorted merge join.

In this case, both tables are sorted by Oracle (using temporary table space) by joining key (X in this case).

The advantage is, when joining two tables, Oracle just needs to scan only a small part of the table (e.g. as they are sorted, to find a record in T2 when T1.X = 15 will require Oracle search just from X=10 to X=20 in T2 as it cannot be beyond this range and so on).

The trade off is, it will take some resource (space/time/memory) to sort the tables.

Usually Oracle chooses best method of join based on statistics available. However, user may force using a different type of join using hints like *use_hash(T1,T2)* etc.

4.10 External tables

By creating external tables, you can run SQL statements on simple delimited text files! However, external tables are read only i.e. you can't update their data using SQL (you need to modify them in text editor).

To create an external table, first you need to create a folder in your hard disk where you will place the text files.

```
CREATE OR REPLACE DIRECTORY EXT_TABLES AS
'E:\ORACLE\ORADATA\EXTERNAL_TABLES'
```

The user, who will access external table, should have CREATE ANY DIRECTORY system privilege.

Now create the external table in SQL Plus as –

```
CREATE TABLE AIRPORT_CODE
(
CODE      VARCHAR2(3),
CITY      VARCHAR2(100),
COUNTRY   VARCHAR2(50)
)
ORGANIZATION EXTERNAL
(
TYPE ORACLE_LOADER
DEFAULT DIRECTORY EXT_TABLES
ACCESS PARAMETERS
(
RECORDS DELIMITED BY NEWLINE
FIELDS TERMINATED BY '|'
MISSING FIELD VALUES ARE NULL
)
LOCATION ('AIRPORT.TXT')
)
REJECT LIMIT UNLIMITED
```

You can now run SQL on the table.

5 DBMS Packages

5.1 DBMS_JOB

Check JOB_QUEUE_PROCESSES parameter has non-zero value in initialization file.

```
BEGIN
SYS.DBMS_JOB.SUBMIT(
JOB => :JOB,
WHAT => 'BEGIN UPDATE_ALL_TRANS_PRICE; END;',
NEXT_DATE => TO_DATE('11-06-2004 09:54:15', 'DD-MM-YYYY
HH24:MI:SS'),
INTERVAL => 'SYSDATE+15/1440');
COMMIT;
END;
```

5.2 UTL_FILE

A sample procedure to unload a table data to a text file using UTL_FILE package is shown below.

Set UTL_FILE_DIR = ‘E:\ORACLE\ORADATA\UNLOAD’ (or path where from you need to read/write) in initialization file.

```
CREATE OR REPLACE PROCEDURE UNLOAD_TABLE (piTable IN VARCHAR2)
AS

CURSOR cTable IS
SELECT * FROM category;

f       utl_file.file_type;
buf     VARCHAR2(200);

BEGIN
     f:=UTL_FILE.FOPEN('E:\ORACLE\ORADATA\UNLOAD','CATEGORY.TX
T','w');
     FOR i IN cTable LOOP
         buf:=i.CategoryNo || '|' || i.Description || '|' ||
i.ParentCategoryNo;
         UTL_FILE.PUT_LINE(f,buf);
     END LOOP;
     UTL_FILE.FCLOSE(f);
     DBMS_OUTPUT.PUT_LINE('TABLE UNLOADED');
EXCEPTION
         WHEN OTHERS THEN
              DBMS_OUTPUT.PUT_LINE(SQLERRM);
```

```
END;
/
```

Similarly, GET_LINE is used to read data from file.

6 Useful scripts for DBAs

6.1 To show primary/foreign key relationships of all tables and views in a given schema

```
SELECT
D.TABLE_NAME "Table name",
D.CONSTRAINT_NAME "Constraint name",
DECODE(D.CONSTRAINT_TYPE,
                         'P','Primary Key',
                         'R','Foreign Key',
                         'C','Check/Not Null',
                         'U','Unique',
                         'V','View Cons') "Type",
D.SEARCH_CONDITION "Check Condition",
P.TABLE_NAME "Ref Table name",
P.CONSTRAINT_NAME "Ref by",
M.COLUMN_NAME "Ref col",
M.POSITION "Position",
P.OWNER "Ref owner"
FROM
     DBA_CONSTRAINTS D
LEFT JOIN
     DBA_CONSTRAINTS P
     ON (D.R_OWNER=P.OWNER AND
D.R_CONSTRAINT_NAME=P.CONSTRAINT_NAME)
LEFT JOIN
     DBA_CONS_COLUMNS M
     ON (D.CONSTRAINT_NAME=M.CONSTRAINT_NAME)
WHERE
     D.TABLE_NAME
     IN (
     SELECT TABLE_NAME FROM DBA_TABLES WHERE
OWNER=UPPER('mkm')
     UNION ALL
     SELECT VIEW_NAME FROM DBA_VIEWS WHERE OWNER=UPPER('mkm')
     )
ORDER BY 1,2,3
```

6.2 To see all objects in a table space file

```
SELECT
D.OWNER,D.SEGMENT_NAME,D.SEGMENT_TYPE,D.TABLESPACE_NAME,
D.HEADER_FILE,V.NAME
FROM
Dba_Segments D
```

```
JOIN V$DATAFILE V ON (D.HEADER_FILE=V.FILE#)
WHERE
D.OWNER IN ('MKM')
```

6.3 Which user executing what type of command

```
SELECT
       SID,
       SERIAL#,
       v.schemaname,
       DECODE(COMMAND
                      ,0,'None'
                      ,2,'Insert'
                      ,3,'Select'
                      ,6,'Update'
                      ,7,'Delete'
                      ,8,'Drop'
                      ,26,'Lock Table'
                      ,44,'Commit'
                      ,45,'Rollback'
                      ,47,'PL/SQL Execute'
                      ,'Other') command
FROM V$SESSION v
```

6.4 Get output of a query to a text file from SQL Plus

```
set pagesize 5000
set linesize 5000
spool F:\TEMP\OUTPUT.TXT
set colsep "|"
select * from MKM.PRODUCT;
set colsep " "
spool off
set pagesize 50
set linesize 200
```

6.5 To see size of your tables and indexes

```
select SEGMENT_NAME, SEGMENT_TYPE, BYTES from USER_SEGMENTS
where segment_type IN ('TABLE','INDEX')
```

6.6 To see free spaces in table spaces

```
SELECT b.tablespace_name, b."Total MB", ROUND(b."Total MB" -
a."Free MB",2) "Used MB",
       a."Free MB", ROUND(((b."Total MB"- a."Free
MB")/b."Total MB")*100,2) "% used"
```

```
FROM (
 (SELECT f.tablespace_name, ROUND(SUM(f.bytes/(1024*1024)),2)
"Free MB" FROM dba_free_space f
 GROUP BY f.tablespace_name) a
 RIGHT JOIN
 (SELECT d.tablespace_name, ROUND(SUM(d.bytes/(1024*1024)),2)
"Total MB" FROM dba_data_files d
 GROUP BY d.tablespace_name) b
 ON a.tablespace_name = b.tablespace_name
 )
 ORDER BY 1
```

7 New features of 10/11g

Which are not covered in this book so far. Any feature which is introduced in 10g, is also available in 11g with usually more options.

Flashback query with changing data

You already have flashback query option in Oracle 9i where you can query snapshot of old data from undo table space. But in 10g, you can examine how values changed between two time points.

You can query yesterday's version of table data as:

```
SELECT * FROM PRODUCT AS OF TIMESTAMP SYSDATE-1
```

Rollback monitoring

In 9i, you can't tell easily how much time it is going to take to rollback a large transaction (until it is complete, the locks acquired by the transaction won't be released). In 10g you can simply get this vital information from V$SESSION_LONGOPS view.

Improved table space management

10g automatically creates a SYSAUX table space during database creation. It is used to hold data for several of Oracle's support schemas like DBSNMP, ODM etc.

Flashback table

In 9i, to restore a dropped table you need to use import from back up or do incomplete recovery. In 10g, when a table is dropped, it goes to Oracle's Recycle Bin and you can easily restore it from there! You can run following command from SQL Plus.

```
SELECT * FROM DBA_RECYCLEBIN
```

To restore the table use – `FLASHBACK TABLE` *`schema.table_name`* `TO BEFORE DROP`

Hey that's easy!

SQL Plus improvements

Several enhancement has been made to SQL Plus like prompting, improved file manipulations, fast DUAL optimization plan etc.

Automatic Storage Management (ASM)

In 9i, disk storage consideration (like RAID, stripping, logical grouping etc.) was done manually. From 10g it can be done automatically with the help of ASM.

RMAN improvement

Several enhancements have been made for incremental backup methods.

Auditing improvements

Now auditing can be performed at very detail level (FGA – fine grained audit). In 10g, auditing will show you what user made exactly what changes to data.

Wait interface

Data dictionary views now show more information regarding on user wait events, which will help to diagnose wait problems better.

Materialized view improvement

Oracle introduces materialized view advisor feature, which helps tuning materialized views. Now Oracle server can decide whether to rewrite the materialized view or not dynamically.

Model clause

Using MODEL clause in SQL, you can treat multidimensional data as an array in memory where you can apply spreadsheet like calculations!

Enterprise Manager (EM) changes

Its architecture has been changed in 10g. It's now installed as an HTTP server (instead of as a client tool in 9i). So, you will use it inside browser! Moreover lots of new features have been added. See section 1.6 for more information.

Virtual Private Database improvements

VPD was also in 8i and 9i. But in 10g, it has improved to support a variety of requirements, such as masking columns selectively based on the policy and applying the policy only when certain columns are accessed. VPD has not been discussed so far in this book. It is a method to implement security at finer grain. For example, if you want to implement that logged on user will only see his own records in database (e.g. seeing only his salary information from salary table), VPD can be set up in such a way that "where username = USER" will be automatically added to SQLs.

Segment management

Segment spaces can be managed more efficiently. Wastage of space in segments is minimized. DBAs will have more control over space allocation.

Transportable table space

In 9i, transportable table spaces can be plugged in to other databases only if they run on same platform. In 10g, they can be plugged to database even running on different platforms. This is very helpful for data movement across different systems. Another improvement is that, data can be unloaded from tables in very fast manner (in non-text format, though) but the unloaded file, again, can be used across platforms.

Automatic shared memory management

The space allocation for DB buffer cache, shared pool, large pool and Java pool can be set to manage automatically and dynamically as and when required (i.e. depending on the database workload)!

Scheduler

Usually dbms_job can be used only to run PL/SQL programs. However, the newly introduced dbms_scheduler can even run OS commands at specified time/interval.

Database replay

Starting from 11g, you now have option of capturing work load from a production database and replay that on test database to simulate live conditions.

7.1 What is g in Oracle's grid computing?

Oracle defines grid computing like this: with grid computing, groups of independent, modular hardware and software components can be connected and rejoined on demand to meet the changing needs of businesses.

What does it exactly mean?

We already know what is parallel computing. A complex task is divided into smaller parts and each part is processed independently by a computer. Then the outputs are combined to get final result.

How does grid computing differ from this parallel computing?

Grid computing is an abstract (or virtualization, as Oracle says) concept. A grid can be an infrastructure grid, an application grid, an information grid and so on. What – getting more confused? Ok, please read on.

In early days (~1980s) of database, the relational database management was starting to gain popularity. In RDBMS terms, a customer buys products and a transaction is generated. All these info like customer, product, transaction etc. are stored in RDBMS. They are linked together (by foreign keys, in 3rd normal form). So from a high level view, customer, product, transaction are all part of RDBMS.

Now come to the present days. We are gathering data like never before! Besides RDBMS, we now have lots of different stuffs like OLAP (Business Intelligence/ Data Warehouse etc.), OLTP (transactional data), BPEL (Business Process Execution Language), Web services (using XML), OWB (Oracle Warehouse Builder) etc.

The grid is collection of all these - i.e. different applications and data which speak with one another.

Oracle has product of almost everything nowadays – which is wrapped around a fuzzy name called Fusion Middleware. These components are designed so that one component (say OWB) can interact with another (say core Oracle database). This constitutes an architectural grid.

The relational data on your RDBMS can be termed as information grid.

There's another aspect of grid. Let us take Oracle Real Application Cluster or RAC. Here multiple instances of database are inter-connected to safeguard single point of failure. This is an example of infrastructure grid. Everything is part of grid. It is a concept. It could have been told as Matrix computing too. (Did you enjoy Matrix series of movies?)

Now you know what the buzzword Grid computing mean!

7.2 What are Oracle Fusion Middleware (OFM) and Service Oriented Architecture (SOA)?

If you are not sure about what are these things and try to have a look at Oracle's website, there is a good chance that you might find yourself confused.

That's quite expected. Oracle's website is for marketing their product. It's not their interest to describe their products in a way that people think it is there is no magic about it!

So, here I try to explain the things in non-geeky terms.

The Fusion Middleware consists of Oracle's non core-database products – some of which are not actually middleware!

This includes Oracle's Developer suite (Forms & Reports), Java related tools, web logic server, content management etc. OFM depends on open standards such as BPEL, SOAP, XML and JMS.

Oracle SOA is a part of OFM.

What is SOA?

The main essence of SOA is that applications will talk with each other in a language (i.e. data format, process steps) which is understood by all others applications communicating with.

SOA is about reuse.

SOA helps business to move, change, partner and re-invent itself with ease and grace.

SOA extends idea of reuse not only to web services but also with business services.

SOA components are loosely coupled.

SOA can contain web service, BPEL etc.

Web service

Example, you throw a postcode to Yahoo Geo-coder and it gives you back latitude, longitude of that post code.

You ask for price of particular item to a website, it supplies you the price.

Usually, web service results are returned in XML format to ensure universal compatibility.

In SOA segment, you will often hear the term *Orchestration*. What does it mean?

If you seen an orchestra, you know that conductor just draws some invisible drawings in mid air by moving his magic wand from one side to another. However, musicians can decipher his rhythm and plays their instruments so that every one plays same tune at same pace.

Orchestration in SOA has similar meaning. It ensures that all applications under SOA, know how to be in sync with other applications in the group.

But how orchestration is implemented?

It is done via BPEL or Business Process Execution Language.

BPEL is a tool, using which you can draw how data moves from one application to another. For those who have not used it, it is like a Visio flow chart diagram editor. But, when you draw objects in BPEL, you tell them what to do. You instruct them where to read data from, how to process it and where to send output after processing finished. So, basically it is a graphical tool to define business process. Without BPEL, the whole process will look like thousands of lines of PL/SQL (or Java or C++ or whatever) codes!

Behind the scene BPEL still writes codes - but it just make simpler (!) for any business user to understand and define the process. Now whether that is good or bad is debatable – I am just outlining the concept here.

So, BPEL works to integrate several applications. BPEL usually follows XML standard as interface to several components.

Let us take a bigger example. Your supplier sends you a file which contains all the products, quantities and unit price you ordered for. You need to update your inventory accordingly. Now assume that your supplier quoted the price in € but you need to put that in £ in your database. Using absolute minimum technology, you need to write a small program to convert € to £ while loading that data to your system. But if you are using BPEL, you can visually draw the program!

8 DBA interview questions

Answers are not provided. Some questions are basic and straightforward. Some questions are open ended. If you cover OCP curriculum, you will be able to answer most questions. These questions are for DBAs with 3 – 8 years of experience.

Please note that in most DBA interviews, besides core DBA questions, candidates are usually asked some questions about the operating system (say Unix) as well.

Concentrate on questions depending on the role actually required in the position.

8.1 Open-ended questions

1. What DBA activities did you to do everyday?
2. What is your typical day like?
3. What other parts of your organization do you interact with and how?
4. Do you consider yourself a development DBA or a production DBA and why?
5. What database and overall architecture would you suggest for testing new middleware without impacting production?
6. How do you assess my database's health?
7. How would you approach a performance problem with a three-tier application?
8. How do you test your backup/recovery procedures?
9. How would you support the upgrade process for multiple applications, with different application rollout cycles, in the same instance?
10. What ER tool (& version) you used?
11. What should consist of DW Team?

8.2 Technical questions

8.2.1 Set 1

1. Explain the difference between a hot backup and a cold backup and the benefits associated with each.
2. You have just had to restore from backup and do not have any control files. How would you go about bringing up this database?

3. How do you switch from an init.ora file to a spfile?
4. Explain the difference between a data block, an extent and a segment.
5. Give two examples of how you might determine the structure of the table DEPT.
6. Where would you look for errors from the database engine?
7. Compare and contrast TRUNCATE and DELETE for a table.
8. Give the reasoning behind using an index.
9. Give the two types of tables involved in producing a star schema and the type of data they hold.
10. What type of index should you use on a fact table?
11. Give two examples of referential integrity constraints.
12. A table is classified as a parent table and you want to drop and re-create it. How would you do this without affecting the children tables?
13. Explain the difference between ARCHIVELOG mode and NOARCHIVELOG mode and the benefits and disadvantages to each.
14. What command would you use to create a backup control file?
15. Give the stages of instance startup to a usable state where normal users may access it.
16. What column differentiates the V$ views to the GV$ views and how?
17. How would you go about generating an EXPLAIN plan?
18. How would you go about increasing the buffer cache hit ratio?
19. Explain an ORA-01555 (Snapshot too old)
20. Explain the difference between $ORACLE_HOME and $ORACLE_BASE.
21. How would you determine the time zone under which a database was operating?
22. Explain the use of setting GLOBAL_NAMES equal to TRUE.
23. What command would you use to encrypt a PL/SQL application?
24. Explain the difference between a FUNCTION, PROCEDURE and PACKAGE.
25. Explain the use of table functions.
26. Name three advisory statistics you can collect.
27. Where in the Oracle directory tree structure are audit traces placed?
28. Explain materialized views and how they are used.
29. When a user process fails, what background process cleans up after it?
30. What background process refreshes materialized views?
31. How would you determine what sessions are connected and what resources they are waiting for?
32. Describe what redo logs are.
33. How would you force a log switch?
34. Give two methods you could use to determine what DDL changes have been made.
35. What does coalescing a table space do?

36. What is the difference between a TEMPORARY table space and a PERMANENT table space?
37. Name a table space automatically created when you create a database.
38. When creating a user, what permissions must you grant to allow them to connect to the database?
39. How do you add a data file to a table space?
40. What view would you use to look at the size of a data file?
41. What view would you use to determine free space in a table space?
42. How can you rebuild an index?
43. You have just compiled a PL/SQL package but got errors, how would you view the errors?
44. How can you gather statistics on a table?
45. How can you enable a trace for a session?
46. What is the difference between the SQL*Loader and IMPORT utilities?
47. Name two files used for network connection to a database.

8.2.2 Set 2

1. A user has connect and resource privileges allocated by the DBA. He is been allocated quota on three table spaces by the DBA, - Default Table space Quota 500 M, Table space 1 Quota 100 M, Table space 2 Quota 200 M. Now the user tries to make a table, which uses 300 M in table space 2. Will he be successful?
2. After a table is defined, can columns be removed?
3. Can we add columns to a table, which is already defined?
4. Can you explain recovery of in a RDBMS?
5. Describe the procedure for moving a data file from one disk to another. Show actual SQL used for this procedure.
6. Difference & when to use FIRST_ROWS & ALL_ROWS parameter, while using an optimizer
7. Explain B-Tree index?
8. Explain composite index?
9. Explain different kind of locking in RDBMS?
10. Explain the definition of a "star schema."
11. Explain the difference between a "hot" and a "cold" backup. What is involved with executing a "hot" backup?
12. Explain the differences between logical and physical database backups. What are the benefits and penalties of each?
13. Explain the processing of a program using cursor?
14. Explain third normalized form?
15. Explain two phase commit?

16. Explain what a bitmapped index is, what its applications would be, and why it's useful.
17. Explain what an Oracle snapshot is.
18. Have you heard about Tuxedo, Encina, CICS?
19. How can one implicitly disconnect a user from the database, who is idle for a long time? (Using Profiles)
20. Is it possible that there could be performance degradation due to so many indexes? When and how?
21. Questions were asked about parameters in export, like: consistent = y, (what is the purpose of it)?
22. There are 100 data files, numbered from 1 to 100. File number 10 is deleted and it has 500 MB of data. The database is working in No-Archive log mode. How can the database be recovered?
23. There are about a 10,000 records in a table. User 1 runs a query, which is doing a full-table scan. While doing the full table scan, user 2 changes some value x to y using a DML statement in record. What does the 1st user see (x or y)? If x, then where is the value stored? (Rollback Segments).
24. What RDBMS objects are required before you can create a table?
25. What are the contents of a Control File?
26. What are the different data structures used for indexes?
27. What are the types of recovery one can perform?
28. What can you do with an alias that you cannot do with a synonym?
29. What is Normalization?
30. What is a correlation name and explain the usage?
31. What is a cursor?
32. What is a foreign key?
33. What is a join? Explain outer join with an example?
34. What is a page in RDBMS?
35. What is a partitioned table?
36. What is a stored procedure? Why there are required?
37. What is a sub query?
38. What is a synonym?
39. What is a view?
40. What is commit and rollback?
41. What is distributed transaction processing?
42. What is logging in a RDBMS?
43. What is primary key?
44. What is query execution plan?
45. What is query optimizer?
46. What is referential integrity?
47. What is the SQL statement necessary to delete a user named "Joe" and everything that he owns in a database?

48. What is the difference between a searched update and a positioned update?
49. What is the difference between static SQL and dynamic SQL?
50. What is the importance of redo log files?
51. What is the save point in a transaction?
52. What is the use of LIKE?
53. What is the use of the parameter "MAXTRANS”. Is it used at block level, extent level or segment level?
54. What is transition monitor?
55. What is transaction? Explain atomic property of a transaction?
56. What variables in the "INIT.ORA" file affect the amount of RAM needed for the SGA?
57. What's the default password for the Oracle user "Scott"?
58. Where and when can we use the HAVING clause?
59. Which SQL error will you get when you try to fetch data from a cursor after it has run out of data?
60. Which parameter can be used to enable check pointing at regular intervals (FAST_START_MTTR_TARGET)?
61. Which parameter do you use to read multiple blocks in one go (DB_MULTIBLOCK_READ_COUNT)
62. Why a stored procedure supposes to give better performance?
63. Why indexes are so important?
64. Why is the difference between Direct Path Export and Conventional Path Export?
65. Why it is bad to use the SELECT * in static SQL?
66. Why row level locking is desired?
67. Why would you use UNION instead of a JOIN?

8.2.3 Set 3

1. What data warehouse features are there in Oracle 9i?
2. What is the difference between B-Tree & Bitmap index? Explain.
3. What is dimension table?
4. What is surrogate key?
5. What is 3rd normalization?
6. What is ref cursor?
7. What is materialized view?
8. What is query rewrite?
9. What is inline view?
10. What is conformed dimension?
11. What is slowly changing dimension?

12. How ER diagram differs from DW dimension? (Answer – OLTP schema vs. Star schema)
13. What is type1, type2, type3 related to dimension?
14. What is the difference between top down and bottom up approach?
15. When is full-table scan beneficial?
16. What performance tunings you have done?
17. Update Emp set salary=5000 where id=100 – explain all steps that happens internally.
18. State differences between 8i and 9i?
19. State differences between 9i and 10g?
20. What parameters of database you can't change without recreating one? (Answer – block, buffer, and character set etc.)
21. What is asynchronous PL/SQL block?
22. What Oracle books you read?
23. What is autonomous transaction?
24. What are various types of transactions?
25. What extra Oracle features you like to see?
26. Explain B Tree & Bitmap index.
27. Explain clearly what will happen if I index a gender column using B tree instead of Bitmap.
28. What is basically stored in index?
29. If a table has no rows, does it take any space?
30. What largest database you have handled? How many tables were there?
31. What is two phase commit?
32. What is distributed database?
33. What is RAC & what is its advantage?
34. How do you handle exception in PL/SQL procedure?
35. What is the advantage or package over procedure?
36. What is pragma-restricted function?
37. What are Coyd's golden rules (not about normalization)?
38. How do you suggest backup strategy of production database?
39. What database challenges you faced in your last project?
40. What is 2nd normalization?
41. When do you de-normalize data?
42. What database tuning you have done?
43. What is the length of rowid data?
44. What Oracle analytic functions you have used?
45. Can you specify sequence in SQL loader?
46. Explain direct/indirect load?
47. What are different components of SGA? What are different background processes? What are functions of each of them?
48. What is Reco process?
49. How do you see all columns of all tables?

50. Explain implicit & explicit cursor.
51. How do you declare record type variable?
52. What is PL/SQL table? Define it's syntax.
53. How do you select duplicate rows from a table?
54. How do you avoid fragmentation?
55. Explain deadlock with an example.
56. Why people sometimes don't run database in archive log mode?
57. What does DBWR & LGWR do?
58. Who write data in database buffer cache & redo log buffer?
59. Can you get back sequence values already generated?
60. Explain shared server concept.
61. When a new session starts, how do you see extra processes created from data dictionary tables?
62. Where do you find all database objects?
63. Where the source of function/procedures is stored?
64. What do you know about Standby database & Replication?
65. What are different types of backups?
66. Explain all steps of complete & incomplete recovery.
67. What are various types of recoveries? Explain.
68. What database tuning you have done?
69. What tuning tools you have used?
70. What database design you have performed?
71. What backup method you suggest for a 24x7 production database?
72. Explain some tuning methods.
73. Explain various components of SGA.
74. Where does init.ora reside?
75. Did you install Oracle server?
76. When do you go for import/export instead of backup?

8.2.4 Set 4 (SQL)

1. What is wrong with following query: SELECT column1, column2, AVG(column3), column4 FROM table1 GROUP BY column1, column2
2. What is wrong with this query: SELECT AVG(COUNT(employee_id) FROM employee GROUP BY dept
3. How do you delete duplicate rows from a table using SQL? [Ans – delete from dup_test d1 where rowid not in (select max(rowid) from dup_test d2 where d1.num = d2.num)]
4. How do you identify duplicate rows in a table?
5. A table has following columns – deptartment, employee_name and salary. How do you list average salary of all departments?
6. How do you find database’s time zone? [select dbtimezone from dual]

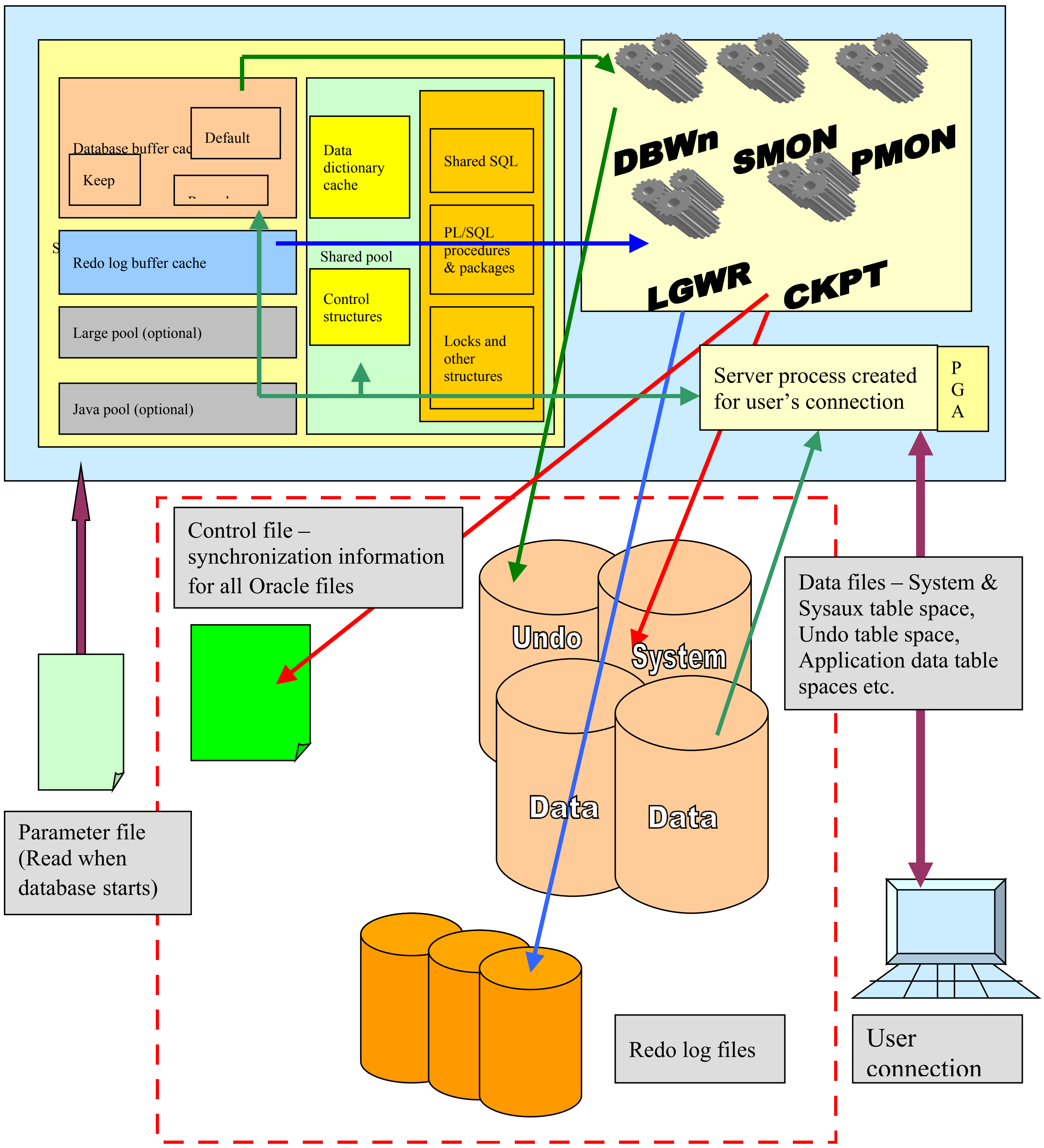
8.3 Questions you should ask

What processes do you follow while implementing changes in production?
Beside the DBAs and system administrators, who has access to the "Oracle" operating system account?
How often is the oracle operating system account password changed?
Are the DBAs co-located with the teams they support?
How is capacity planning performed?
Is there adequate capacity already in place to support the expected growth over the next year?
Is there a formal job definition for the DBA role?
Is there a defined technical career path?
How is IT aligned with the business areas?
How many employees report to more than one manager?
How do you determine if a DBA has been successful?
How are the application DBAs and production control DBAs organized?

9 References

- Oracle classroom course presentations
- OCA/OCP Oracle books (DBA I, DBA II, Performance Tuning, SQL) from BPB publications.
- Oracle manuals
- Oracle Press DBA and PL/SQL books
- Classroom course at SQL Star International, Calcutta.
- http://download.oracle.com/docs/cd/B28359_01/rac.111/b28252.pdf
- http://www.oracle.com/technology/pub/articles/chan_sing2rac_install.html
- http://www.orafaq.com/wiki/RAC_FAQ

10 Oracle server's diagrammatic overview


Database buffer cac
Default
Keep
Redo log buffer cache
Large pool (optional)
Java pool (optional)
Data dictionary cache
Shared pool
Control structures
Shared SQL
PL/SQL procedures & packages
Locks and other structures
DBWn
SMON
PMON
LGWR
CKPT
Server process created for user's connection
P G A
Control file – synchronization information for all Oracle files
Undo
System
Data
Data
Data files – System & Sysaux table space, Undo table space, Application data table spaces etc.
Parameter file (Read when database starts)
Redo log files
User connection