Database Administration

Dadhi Ram Ghimire

Unit 1: Database Architecture

Oracle is a Relational database management system, which uses Relational Data Model to store its database and SQL (Structured Query Language) to process the stored data. Oracle DBMS is best explained with the term Oracle Instance.

An Oracle Instance is a complex set of memory structures and operating system processes. It manages all database activities, such as transaction processing, database recovery, form generation and so on. The instance structure is loosely styled after UNIX's implementation of multitasking OS.

Discrete processes perform specialized task within the RDMBS that work together to accomplish the goals of the instance. Each process has a separate memory block that it uses to store local and private variables, address stacks and other runtime information. These processes are known as Oracle Background processes. The processes use a common shared memory area for processing data concurrently. This memory area is known as System Global Area (SGA).

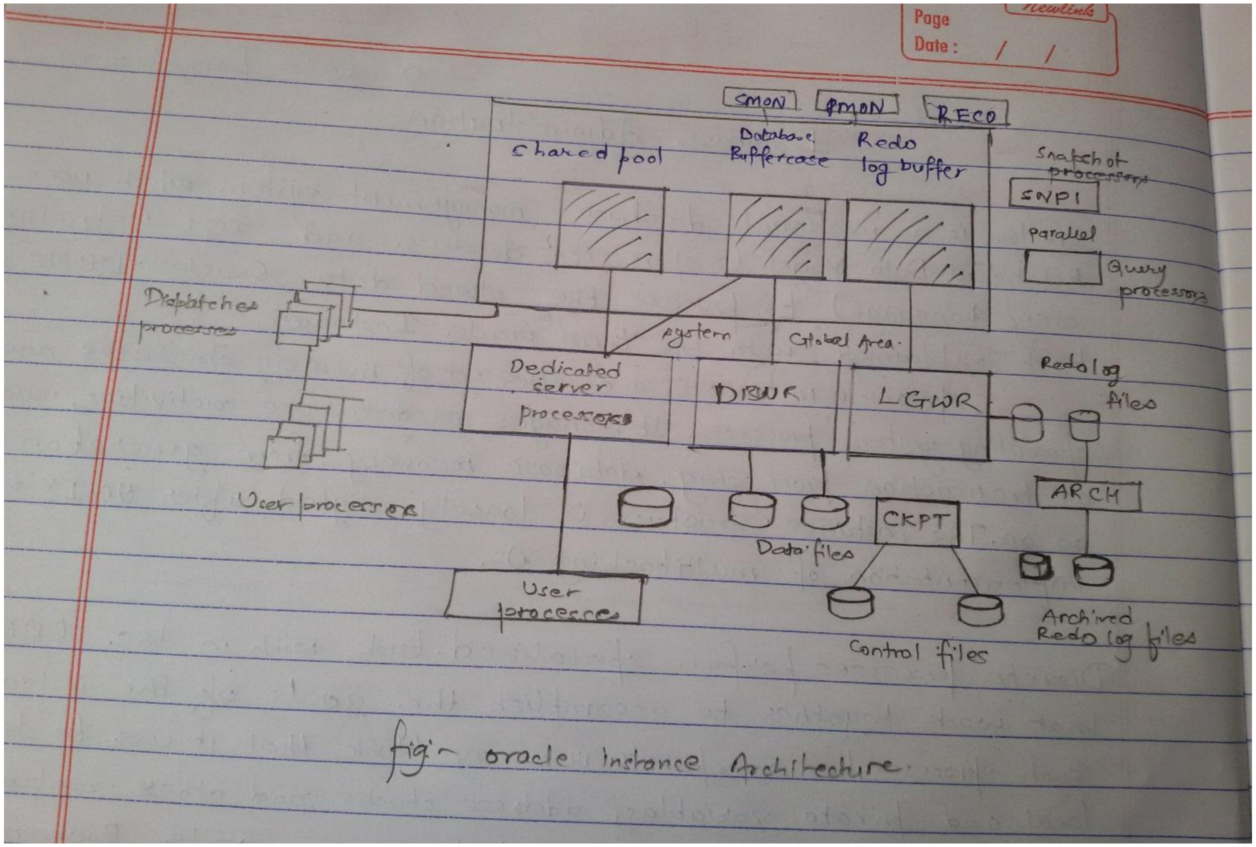


fig:- oracle instance Architecture.

System Global Area:

The SGA is the primary component of the instance. It provides memory structure necessary for data manipulation, SQL statement parsing and re-do caching. Multiple processes can access and modify the data contained in it in a synchronized manned. The SGA consists of the following components:

1. Shared Pool
2. Database buffer cache
3. Redo log buffer
4. Multithread Server structure

1) Shared pool:

Shared pool contains the memory that stores the text, parse tree of SQL statements. In this pool, SQL statements are executed and changes made during a transaction is written to the blocks in database buffer cache. If same SQL statements are used repeatedly, parse tree is reused.

2) Database Buffer Cache:

The buffer cache is composed of memory blocks. All data manipulated by oracle server is first loaded into the buffer cache before being used. All data updates are performed in the buffer blocks. The data movement (swapping and loading) between buffer and disk is done using LRU (Least Recently Used) algorithm. Buffer blocks that has been modified are called dirty blocks and are placed in the dirty list. The list keeps track of all the data modification. The disk writer process updates the disk based on the entry in the dirty list.

3) The Redo Log Buffer:

The redo log buffer is used to store redo information in the memory before it is stored in redo log files in the disk. It is circular buffer.

The ORACLE Background Process:

The oracle server processes the transactions concurrently. To accomplish the task concurrently the server divides the entire workload between a numbers of programs each of which operates largely independently. The Oracle background processes are:

a) DBWR(Database Writer):

Database writer process is responsible for writing the dirty blocks from the database buffer cache to the data files on the disk.

b) LGWR:

Log Writer is the process that writes redo log entries from the redo log buffer in the SGA to the online log files. Log Writer performs this write when a commit occurs.

c) CKPT:

The Check Point process is an optional background process that performs the check point task that Log Writer would normally perform. This process reduces the amount of workload on Log Writer when there are frequent checkpoints occurring.

d) **ARCH:**

The archiver process is responsible for copying full online redo logs to the archive redo log files while the archiver is copying the redo log, no other process can write to the log.

e) **Dispatcher Process:**

The Dispatcher process passes the user request to the SGA request queue and returns the server response back to the corresponding user.

f) **SMON:**

The System Monitor performs automatic instance recovery. If the last database shutdown was not clean SMON automatically rolls forward the operations that were complete but couldn't be installed in the databases and rolls back unfinished transactions.

g) **PMON:**

Process Monitor is responsible for cleaning up terminated or failed processes, rolling back uncommitted transactions, releasing the logs held by disconnected processes and freeing SGA resources held by failed processes.

Basic Terminologies:

Control Files:

Oracle’s control files are binary files containing information about the assortment of the files that comes together to support Oracle database. They contain the information that describe the names, locations and sizes of database file. In general, there is only one control file but database administrator create more than one control file.

The Online Redo Logs (OR Redo Log):

As sessions interact with oracle database 11g, the details of their activities are recorded in online Redo logs. Redo logs are transaction logs. A transaction is a unit of work, passed to the database for processing.

Tablespace:

A table space is fancy oracle name where the data resides. Oracle stores data logically in tablespace and physically in data files. A database administrator can create a new tablespace, add and remove data files from tablespace and change various parameters of a tablespace.

SYSTEM Tablespace:

When Oracle is installed, a system tablespace is created that contains oracle’s data dictionary. System tablespace contains following information:

I. Datatypes
II. Field Size
III. Viewing and manipulation rights

SYSAUX (System Auxiliary) Tablespace:

Many tools and options that supports the oracle database 11g activities store their objects in the SYSAUX tablespace.

Data Files:

Data Files are operating system files where the data resides. Data files can be created, expanded and deleted as per the requirement. Data files contains table, index, schemas, etc. A tablespace contains many data files bit a data file is associated with only one tablespace.

Roles and Responsibilities of Database Administrator:

1) **Installation and Configuration:**

DBA must install and customize the database software and any assorted programs that run alongside and access the database.

2) **Create data files and tablespace:**

DBA should create datafiles and tablespace as per the requirement.

3) Create and manage accounts:

DBA must create user accounts and grant specified privileges to them.

4) Performance Tuning:

DBA tweaks the database to increase efficiency of database.

5) Configure Backup

6) Work with Developers

7) Stay Current

8) Recovery

9) Install Patches

Unit 2

Control Files and Redo Log Files

Multiplexing Control Files:

Multiplexing is defined as keeping a copy of same control file in different location. Copying the control file to multiple locations and changing the CONTROL_FILES initialization parameter to include all control files name specifies the multiplexing of control file. (We can multiplex single control file up to 8 places or create 8 files).

By storing the control file on multiple location, we avoid the risk of single point of failure when control files are multiplexed, update to control files takes little more time.

When multiplexing control files, Oracle updates all the control files at the same time, but uses only the first control file in the default location.

--- Procedure of multiplexing the control file:

1) Shutdown the database
2) Copy the content files to more locations by using operating system command
3) Change initialization parameter file to include the new control file names in the parameter CONTROL_FILES

Example:

```
SQL> show parameter spfile;
NAME                          TYPE       VALUE

------------------------------------ ----------- ------------------------------

spfile                        string     C:\APP\SHORYUKANE\PRODUCT\11.1

                                         .0\DB_1\DATABASE\SPFILEORCL.OR

                                         A

SQL> select * from v$controlfile;
```

```
STATUS

-------

NAME

--------------------------------------------------------------------------------

IS_ BLOCK_SIZE FILE_SIZE_BLKS

--- ---------- --------------

C:\APP\SHORYUKANE\ORADATA\ORCL\CONTROL01.CTL

NO       16384            594

SQL> shutdown immediate

Database closed.

Database dismounted.

ORACLE instance shut down.

SQL> $copy c:\app\shoryukane\oradata\orcl\control01.ctl h:\backuporcl\control02.ctl

SQL> startup

ORACLE instance started.

Total System Global Area 1071333376 bytes

Fixed Size                  1334380 bytes

Variable Size             327156628 bytes

Database Buffers          738197504 bytes

Redo Buffers                4644864 bytes
```

```
Database mounted.

Database opened.
```

Redo Log Files/ Online Redo Log File:

Redo Logs records all changes to the database. The redo log buffer in SGA (System Global Area) is written to the redo log file periodically by LGWR process. The redo log files are accessed and are open during normal database operation; hence they are called online Redo log files. A Log Writer writes Redo information from redo log buffer to the online redo log files under a variety of circumstances.

1) A user commits a transaction, even if this is the only transaction in the log buffer.
2) The redo log buffer becomes one third full.
3) When there is approximately 1 MB of change record in the log buffer

LGWR always writes in Redo Logs in cyclic order; if there are 3 Redo Log files, LGWR first writes in first file first. If first file is full, it writes to second file and to the third file. When all files are full, it writes back to first file. LGWR always writes the content of log buffer into Redo Log files before DBWR writes new or modified data blocks to datafiles from database buffer cache.

Server Parameter Files:

Oracle Server Parameter files for oracle instance are called pfile and spfile. It is the text file that store the oracle Server instance parameters and can be changed by text editor.

Avoids the resource and I/O

Managing and maintaining Online Redo Logs:

Redo Threads:

In Oracle Real Application cluster environment more than one oracle instance are running at the same time and each instance has its own Redo Log groups. A separate set of Redo Log groups for each instance is called a Redo Thread and this mechanism avoids the resource and I/O contention.

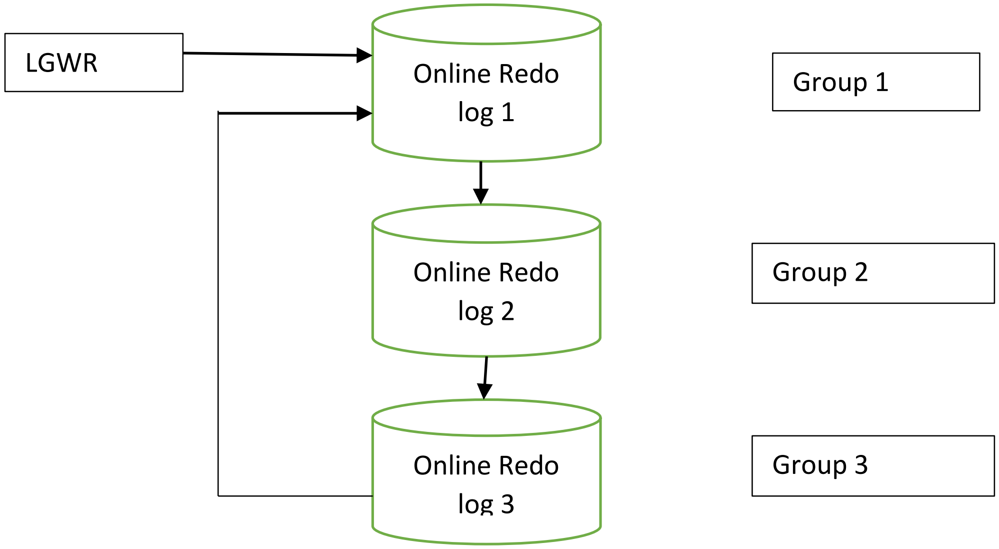
How Oracle Database writes to Redo Log?

The redo log of a database consists of two or more redo log files. The database repairs a minimum of two files to guarantee that one is always available for writing while other being archived(if the database is in ARCHIVELOG MODE).When current log file fills, LGWR begins writing in next available log file(LOG SWITCH). Filled redo log files are available to LGWR for reuse depending upon whether archiving is enabled.

Log File States:

1) **Current:** The file that is being used by log writer.

2) **Active:** These are full online Redo log files required for instance recovery. Archiver processes is busy on the active Redo Log File.
3) **Inactive:** These are also full online redo log files but they are no longer required for instance Recovery. Contents of these files are already archived. LGWR can overwrite these files.



Log Sequence number:

Oracle assign each redo log files a log sequence number every time a log switch occurs and LGWR begins writing to it. When the database archive redo log files, the archived log retains its log sequence number.

Multiplexing Redo Log Files:

To protect a failure involving the redo log itself, oracle database allows multiplexed redo log i.e. two or more identical copies of the Redo log can be automatically maintained in separate locations.

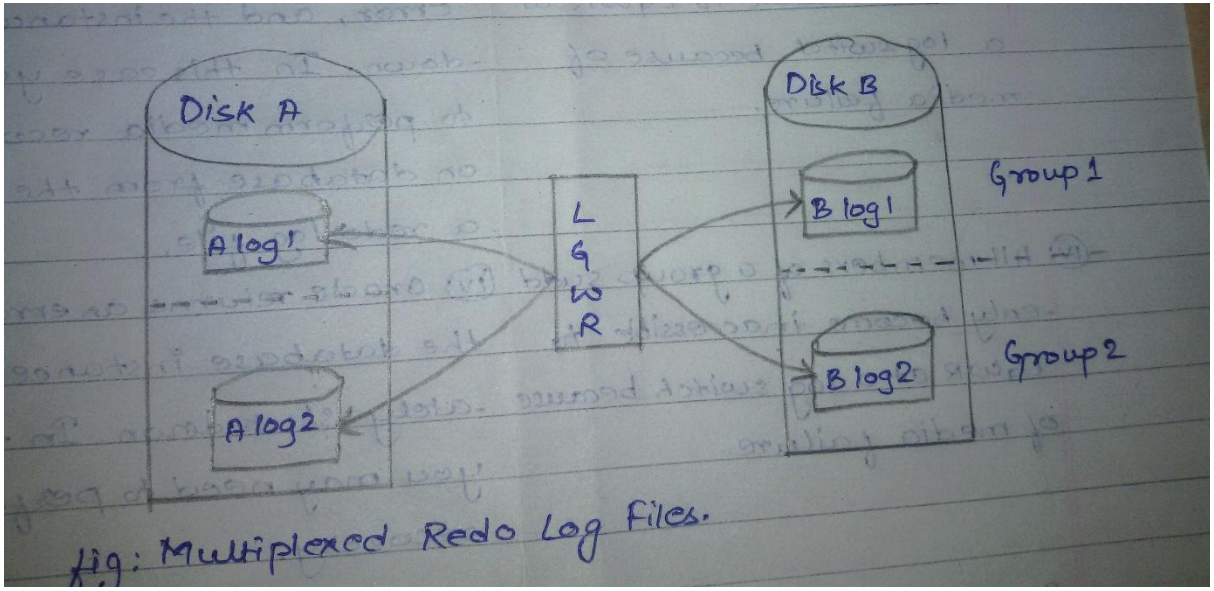


fig: Multiplexed Redo Log Files.

Responding to Redo Log Failure:

Whenever LGWR cannot write to a member of a group, the database marks that member invalid and writes an error message to the LGWR trace file and to the database alert log.

Condition	**LGWR Action**
1) LGWR can successfully write to at least one member in a group.	Writing proceeds as normal LGWR writes to the available members and ignores the unavailable members.
2) LGWR cannot access next group at a log switch because the group needs to be archived.	Database operation temporarily halts until the group becomes available or until group is archived.
3) All members of next group are inaccessible to LGWR at a log. Switch because of media failure.	Oracle database returns an error, and the instance shutdown. In this case you need to perform media recovery on database from the loss of a redo log file.
4) All members of a group suddenly became inaccessible to LGWR at a log switch because of media failure.	Oracle returns and error and the database instance immediately shuts down. In this case you may need to perform media recovery.

Archive log mode and Non Archive log mode:

Database instance can run in both archive mode and non-archive mode. If database is running in non-archive mode, the filled redo log files are not archived/written in archived redo log files. This mode allows to recover a database from an instance failure but not media failure. Filled log files after a log switch are immediately available for writing.

If database is running in archived mode, database can be recovered from a media failure and database can be backed up during running of an instance. Filled log files are used by ARCH process for archiving after a log switch.

Unit 3:

User, Roles, Security, Privileges:

Creating new user:

Create user newuser identified by newpassword

Default tablespace users

```
Temporary tablespace temp

Quota unlimited on users

Quota unlimited on Temp

Profile default;
```

Granting Privileges To User:

```
Grant connect to newuser;          /System Privileges

Grant resource to newuser;         /System Privileges


Grant Select on emp to newuser;         /Object Privilege

Grant insert on emp to newuse;          / Object Privilege


Grant Select on emp to newuser with Grant option;
```

Role:

Role is a database object created to manage privileges to a user. In general Role is a set of system and object privileges that can be granted to a user, Role or group of users.

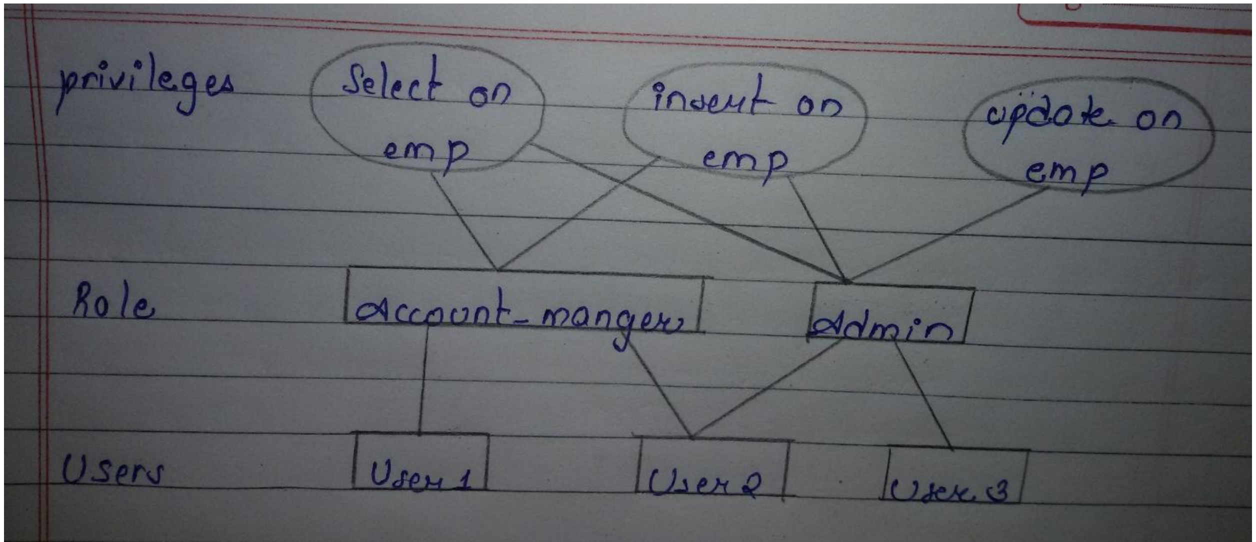


Fig:- Roles management

Benefits of Roles:

1) Easier privilege management
2) Dynamic privilege management
3) Selective privilege management
4) Privilege can be granted through operating system.

Some System Privileges:

Role	Description
1) Connect	User can connect to database and start a session.
2) Resource	User can create various database objects CREATE TABLE, CREATE TRIGGER, CREATE VIEW, etc.
3) DBA	Gives all system privileges to user WITH ADMIN OPTION

Operating modes of Oracle database 11G:

1) **No mount Phase:** In this mode database read the initial parameter file spfile or init.ora file and starts up the Oracle memory structures as well Oracle background process. Here,

instance is started and the database is not associated with the current instance. This mode is used to recreate control file.

Startup nomount

2) **Mount Phase:** This phase is used to associate the currently started instance with the database. In this phase data files and Redo log files are not opened so that end users cannot access the database. This phase is used to carry out some maintenance works.
3) **Open Phase:** In this phase datafiles and Redo log files are opened making database accessible to users.

Startup open;

Alter database open;

Some alternate ways to open oracle database:

1) Read only mode: If this mode is enabled, changes to the database cannot be made. Insert, Delete, Update and alter operations cannot be performed on database objects.

   Alter database open read only;

2) Restrict mode: This mode is mainly used for maintenance works and only privileged system users can access the database users with the DBA privileges have access to database.

   Startup restrict;

3) Quiesce mode: In this mode only sys and system users can query the database without stopping the database without stopping the database and performing a subsequent restrict. The activities of other users continue until they end their session.

```
Alter system quiesce restrict;
```

4) Force startup: It is used in rare instance where database has shutdown properly but the database is not opening up properly. This command forces a shutdown abort followed by startup command.

```
Startup force;
```

Database shutdown modes: When shutting down an instance, these actions are performed

1) Close the database (including data file and redo logs).
2) Unmount the database so that instance and background process runs without an associated database.
3) Shutdown the instance to close the control files.

1) Normal Shutdown: It is the perfect way to shutdown oracle database, this mode waits for all users to disconnect and all transactions to complete before shutdown occurs. When this command is issued new users are not allowed in database.

```
Shutdown;
```

Shutdown normal;

2) Immediate Shutdown: This mode rollbacks all current transactions and users are disconnected. No new transactions are allowed in system. However, this shutdown puts database in a consistent state.

   Shutdown immediate;

3) Transactional Shutdown: It is similar to immediate mode expect the current transactions are allowed to complete.

   Shutdown transactional;

4) Abort Shutdown: When this option is used, all transactions are ended immediately without waiting for back or commit. This mode puts database into inconsistent state.

   Shutdown Abort;

Procedural Language/ Structured Query Language (PL/SQL):

PL/SQL is a combination of SQL along with procedural features of programming Language. It was developed by Oracle Corporation in the early 90's to enhance the capabilities of SQL. Oracle use a PL/SQL engine to process PL/SQL statements. A PL/SQL code can be stored in the client system or in database server side.

Each PL/SQL program consists of SQL and PL/SQL statements which form a PL/SQL block.

PL/SQL Block consists of three section.

1) The declaration section(Optional)
2) The Execution Section(Compulsory)

3) The Exception handling section(Optional)

Declaration Section:

The declaration section of PL/SQL Block start with reserved keyword DECLARE. This section is optional and used to declare any placeholders like variables, constants, cursors and records. These placeholders are used in execution section.

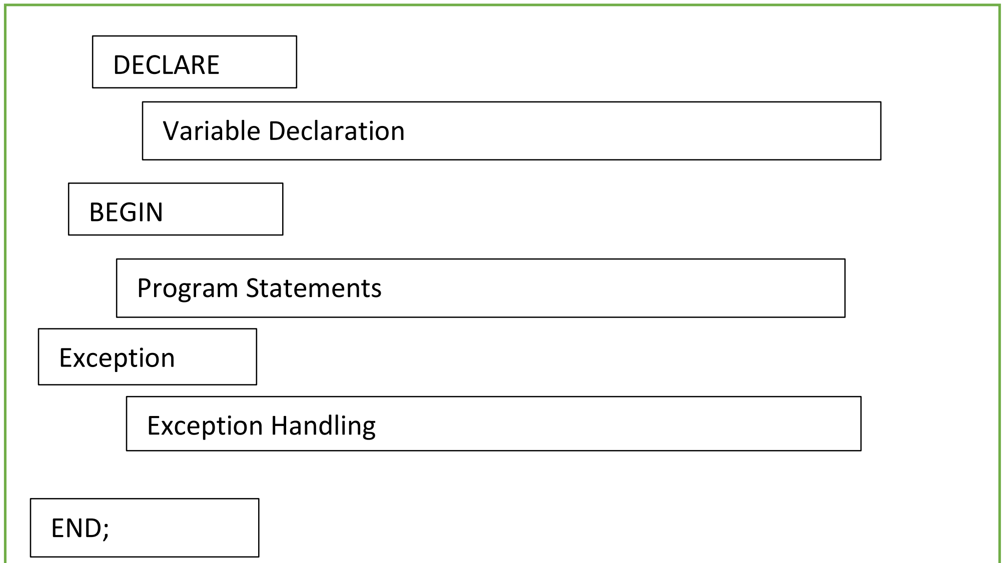
Execution Section:

This section starts with reserved keyword BEGIN and ends with END. This is a mandatory section and where program logic is written to perform certain task. The programming constructs like loops, conditional statements and SQL statements form the execution section.

Exception Section:

This section starts with reserved keyword EXCEPTION. This section is optional and it handles the error so that program terminates normally.

Error messages can be thrown to user using this section.



DECLARE

Variable Declaration

BEGIN

Program Statements

Exception

Exception Handling

END;

Fig: PL/SQL Block

Advantages of PL/SQL:

1) Block Structure
2) Procedural Language Capability
3) Better Performance
4) Error Handling

PL/SQL placeholders:

Placeholders are temporary storage area. PL/SQL placeholders can be variables, constant, cursors and records.

PL/SQL variables:

These are the placeholders that store the vales that can change during program execution.

Syntax: variable_name datatype[NOT NULL:=value];

Each variable declaration is a separate statement and must be terminated by semicolon.

Example:

DECLARE

Salary number (6);

DECLARE

Salary number (4);

Dept varchar2 (10) NOTNULL:=”HR DEPT;” //If user does not insert anything in dept field then by default it inserts HR DEPT

We can directly assign vales to variables. Variable_name :=value;

We can assign values to variables directly from Table’s Columns by using SELECT

SELECT Column.name INTO variable_name from table_name [where condition];

Example:

DECLARE

var_full_name varchar2(20);

```
        var-address varchar2(20);
        var_emp_id number(6):=1;
BEGIN
        SELECT full_name, address
        INTO var_full_name , var_address FROM emp
        WHERE emp_id=var_emp_id;
        dbms_output.put_line(var_full_name);
        dbms_output.put_line( ‘The employee ’||var_emp_id||’ has name ’|| var_full_name||’
                                                has address ’||var_address);
END;
```

Using Loop in PL/SQL:

```
Declare
I number(2);
Begin
        I:=1;
```

```
    Loop
    Dbms_output.put_line(I);
    I:=I+1;
IF I>5 THEN
    EXIT;
END IF;
END LOOP;
END;
```

Cursors:

A cursor is a temporary memory area created when a SQL statement is executed. A cursor can hold more than one row, but can process one row at a time. Cursor is mainly used in SQL SELECT Statements. In other words, cursor acts as pointer to SELECT query result. Disconnected database architecture is achieved with the help of cursors.

Types of Cursor:

1) Implicit: They are created by default when DML statements like INSERT, UPDATE and DELETE are executed. It is also created when SELECT query returns only one row.

2) Explicit: Explicit cursor must be created when a SELECT statement returns more than one row. Even though cursor stores multiple rows, only one record is accessible at a time. We can fetch row one at a time from cursor until data is found in the cursor.

   Example:

```
DECLARE
P product%ROWTYPE;
Cursor producer IS select product_name, product_price from products;
Begin
Open producer;
Loop
        Fetch producer INTO p;
        Exit when producer/NOTFOUND;
        Dbms_output.put_line(p.product_name||’ ‘||p.price);
End Loop;
Close producer;
End;
```

Stored Procedure:

A stored procedure is a name, persistent PL/SQL block which performs one or more specific task. A stored procedure has a header and body. The header consists of the name of procedure and parameters or variables passed to the procedure.

Syntax:

```
CREATE [OR REPLACE] PROCEDURE proc_name[List of parameters]

IS

Declaration Section

BEGIN

        Execution Section

Exception

        Exception Section

END;
```

Parameters Types:

1) IN: This type of parameter is passed to the stored procedure.
2) OUT: This is return type parameter i.e. value returned after procedure execution.
3) IN OUT: This parameter performs dual functions i.e. it is used to pass and return values.

Example:

```
Create or Replace PROCEDURE product_list(id IN Number)
```

```
IS
Pname products.product_name%TYPE;
BEGIN
SELECT product_name INTO pname WHERE PID=id;
Dbms_output.put_line(‘The rewuired product is ’||pname);
End product_list;
```

Executing Procedure:

```
    Exec product_list(1);
```

Trigger:

A trigger is a PL/SQL block structure which is fired when a DML statements like Insert, Delete, Update is executed on a database table. A trigger is triggered automatically when an associated DML Statement is executed.

Syntax:

```
CREATE OR [REPLACE] TRIGGER_NAME
```

```
{BEFORE| AFTER| INSTEAD OF}
{INSERT[OR]| DELETE[OR]| UPDATE}
[OF Col_name]
ON Table_name
[FOR EACH ROW]
WHEN(condition)
BEGIN
        Sql Statements
END;
```

PL/SQL Function:

A function is a named PL/SQL block which is similar to a procedure. The major difference between procedure and a function is that a function must return a value.

Syntax:

```
Create [Or Replace] Function function_name[parameters]
Return return_datatype;
IS
Declaration section
Begin
Execution section
Return return_variable;
Exception
Exception section
END;
/
```

Example:

```
Create Or Replace Function
Return Varchar2(20);
Is
```

```
Emp_name varchar2(20);
Begin
Select first_name INTO emp_name from emo where emp_id=100;
Return emp_name;
END;
```

View:

View is a virtual table that is extracted from one or more tables using SQL Select statements. Views are created to limit the access of users to database tables as DML queries like INSERT, DELETE and UPDATE cannot be performed on view.

Syntax:

```
Create [or replace] view view_name
As select columns
From table
Where condition;
```

Example:

```
Create or replace view emp_name AS Select emp_name from emp where dept_id=100;
```

Package:

A package is a collection of PL/SQL elements that are “packaged” or grouped together within a special BEGIN_END syntax. A package may include following elements:

1) Cursors
2) Variables
3) Exceptions and programs
4) Procedure and functions

It consists of two ports:

1) Package specification
2) Package Body.

1) Package Specification:

   ```
   Create [or replace] package package_name
   Begin
   Procedure p1();
   Function f1();
   End;
   ```

Database Auditing:

Auditing is monitoring and recording of selected user database actions. It can be based upon individual actions, such as type of SQL statements executed or on combinations of factors that can include username, application, time and so on. Security policies can trigger auditing when specified elements in oracle database are accessed or altered, including the content within a specified object.

Auditing is typically used to:

1) Enable future accountability for actions taken in particular schema, table or row, or affecting certain content.
2) Deter users from inappropriate actions.
3) Investigate suspicious activities.
4) Notify an audition that an unauthorized user is manipulating or deleting data and the user have more privileges than the specified ones.
5) Monitor and gather data about specific database activities.
6) Detect problems with an authorization or access control implementation.

Auditing Types and Records:

Oracle allows audit options to audit the following actions.

1) Successful statement executions, unsuccessful statement executions, or both.

2) Statement executions once in each user session or once every time statement is executed.
3) Activities of all users of a specific user.

Database Backup and Recovery:

Database backup is the process of storing database files in a separate disk or tape. Database parameter files, control files, data files and archive log files are stored for this purpose. These backup files can be restored into their original location and database recovery can take place in case of instance failure, media failure and other disasters.

Basic components of oracle backup:

1) Parameter Files(init.ora/spfile)
2) Control Files
3) Data files
4) Archive log files(Optional)

Types of backup:

1) Physical backup: Physical backup is also known as full database backup in which essential database files are stored in alternative location or tapes. Database can be restored and recovered when oracle binaries are available.
2) Logical Backup: In logical backup, selected database objects are backed up for future recovery. Tables, index, etc. be back up can be done using Oracle import/export utility known as data pump.\

Types of physical database backup:

1) Cold Backup: Cold backup is done when database is not online. Database is shut down before essential database files are copied to a backup location or backup tape. Database may be in archive log or non-archive log mode during cold backup as changes are already committed or rolled back during database shutdown. It is also called consistent backup.
2) Hot Backup: Database backup done during online state of database is known as Hot backup. Database must be in archive log mode to perform hot backup as uncommitted transactions are recorded there when database is restored form hot backup, subsequent recovery is done using archive log files to bring database to consistent state.

Consistent and Inconsistent Backup:

- In technical term, consistent Backup is referred as the uniformity SCN(System Change Number) in back up control files and SCN in control file of default location.
- Inconsistent backup is referred to as the dissimilarity on SCN of backup control files and SCN of control files of original location.

Recovery from cold back up:

Recovery from cold backup is a straight forward process in which backup files are restored to their default location. Since, database backup is in consistent state no subsequent recovery is needed.

Recovery from Hot Backup:

Recovery from hot backup can be done in two ways:

1) Complete recovery: In this process database is restored and then recovered forward using all available archive logs.
2) Incomplete Recovery: It is also known as point-in-time recovery. Here, database is restored are then recovered to a predetermined point.
   a. Cancel based recovery

      SQL>recover database until cancel;

   b. Timestamp recovery
      SQL>recover database until time '2015-03-15:14:01:00';
   c. Change based recovery
      SQL> recover database until change 1234;

Parallel Recovery:

Database Recovery process can be accelerated using parallel recovery method, which is applicable to both instance recovery and media recovery. To achieve parallel recovery multiple slave processors are used. Parallel recovery process is most beneficial when several datafiles on different disks are being recovered.

Recovery requires that the changes be applied in datafiles inexactly the same order in which they occurred.

In serial recovery scenario, the SMON(System Monitor) background process both reads the redo log files and applies change to the data files. This may take a considerably long time. However, in parallel recovery, the SMON process is only responsible for reading the redo log files. The changes are applied to database by multiple slave processors. SMON reads the redo log files and serializes the change before dispatching them to parallel slave processors. The parallel slave processor then apply those changes to the data files in the proper order.

RECOVERY_PARALLELISM Parameter:

The RECOVERY_PARALLELISM initialization parameters control the degree of parallelism to use for a recovery i.e. it specifies the number of slave processors to participate in a recovery process. A value of 1 indicates serial recovery while the parameter setting cannot exceed the PARALLEL_MAX_SERVERS setting.

ASM (Automatic Storage Management):

Oracle ASM is a volume manager and file system for oracle database file that supports single instance oracle database and Oracle Real Application Clusters (Oracle RAC) configurations. Oracle

ASM uses disk groups to store data files; an oracle ASM disk group is a collection of disk that oracle ASM manages as a unit. Within a disk group, Oracle ASM creates a file system interface for oracle database files. The content of files that are stored in a disk group is evenly distributed to provide uniform performance.

Oracle RMAN (Recovery manager):

Recovery manager is an oracle database client that backup and recovery of database and automates administration of backup strategies. It greatly simplifies backing up, restoring and recovering database files. The RMAN environment consists of utilities and database that play a role in backing up data. The RMAN environment must include following components.

1. **A target database:**

   It is the database to which RMAN is connect using TARGET keyboard. RMAN performs backup and recovery operation on target database.
2. **A fast / flash recovery area:**

   A disk location can store and manages files related to backup and recovery. We can set fast recovery area and size using DB_RECOVERY_FILE_DEST and DB_RECOVERY_FILE_DEST_SIZE parameters respectively.
3. **RMAN client:**

It is an oracle database executable that interprets commands, directs language commands), and recorded its activity in target database control file. The RMAN executable is pore-installed with the installation of oracle**.**

4. **A media manager:**

   RMAN needs to interact / access sequential access devices during backup and recovery process. A media manager controls these devices during backup and recovery operation.

5. **Recovery catalog:**

   Recovery catalog is a separate database schema that store RMAN activities related to target databases. Recovery catalog is used for database recovery if the control files of target database is lost.

Starting RMAN and connecting to database:

RMAN client is started by issuing RMAN command at the command prompt of operating system.

```
%rman

RMAN>
RMAN>connect TARGET SYS@orcl
Target database password:
```

Logical backup – Import/export and data pump:

Logical backup in oracle database can be done through import/export command using oracle data pump utility. In logical backup selected schemas, tables, indexes can be backed up rather than whole database backup.

Expo Sys /password@database_name as sysdba (In os command pmpt)

Most command parameter are:

FILE-output file

DIRECT-direct path

OWNER-list of owners

TABLES-list of table names

PARFILE-parameter file name

TABLESPACES-list of tablespace to export

Similarly Database import is run as:

IMP Sys/password@database_name as sysdba

File-input file

IGNORE-Ignore object created error

ROWS-import data rows

Oracle data pump:

Form oracle log, oracle introduced new utilities EXPDP and IMPDP for fast data loading (import/export). It is significantly faster than import/export utility.

Creates data_pump_dir in $oracle_home.

Example:

Expdp system/password DIRECTORY = data_pump_dir

DUMPFILE = backup.dmp schemas =HR

Unit 6

Performance Tuning:

One of the biggest responsibilities of a DBA is to ensure that have the oracle database is tuned properly. The oracle RDBMS is highly tunable and allows the database to be monitored and adjusted to increase its performance. Performance tuning is done for following reasons:

1. The speed of computing might be wasting valuable human time(user waiting for response)
2. To enable system to keep up with speed of business conducted.
3. To optimize hardware usage to save money.

To perform performance turning, the bottleneck of performance is identified during performance testing. These bottleneck are rectified during performance tuning.

Performance bottlenecks:

1. **Slow physical I/o:**
   This is caused by poorly configured disks. It may also be caused by unnecessary physical I/O requests by poorly tuned SQL statements.
2. **Excessive CPU usage:**
   Excessive CPU usage means that there is little idle CPU on the system. It is caused by inadequately sized systems and un-tuned SQL statements.

3. **Memory contention:**
   This is caused by poorly allocated memory during installation memory allocated to SGA and PGA may be inadequate which result in memory contention.

Gathering performance statistics:

1. **Baseline:**
   Baseline contains specific performance data from a time period that is preserved for comparison with other similar workload periods when performance problems occurs. Most DBA know their system i.e. they know the high peak periods. They gather the statistics during high peak periods and save it for future comparison.

   Baseline data gather should include following:
   1. Application statistics (transaction volume, response time)

    2. Database statistics (memory, process)
    3. Operating system statistics
    4. Disk I/O statistics
    5. Network statistics

2. **Snapshots:**
   Snapshots are set of historical data for specific periods that are used for performance comparison. By default oracle database takes snapshots every hour and these snapshots remain in system for 8 days.

There are basically two kinds of performance tuning:

1. **Instance tuning:**
   Instance tuning is concerned with initial design of database to avoid bottlenecks that could lead to performance problem. During instance tuning following options must be considered.
    a. Allocating memory to database structures
    b. Determining I/O requirements to different parts of database
    c. Tuning operating system for optional performance of database
2. **SQL tuning:**
   Many application programs consider SQL as a messaging services i.e. queries are issued and data is returned from database. However, many client tools generate inefficient SQL statements. SQL tuning is concerned with quick response time to a SQL query i.e. query that operates on small number of rows, extents etc. Indexing is also used during SQL tuning.

Performance tuning can also be classified according to actions performance on database:

1. **Proactive monitoring:**
   Proactive monitoring usually occurs on a regularly schedule interval, several performance statistics are examined to identified whether the system behavior and resource usage has changed. It is also called proactive tuning. It does not result in configuration changes to the system, unless the monitoring expose a serious problem that is developing.
2. **Bottleneck Elimination:**
   The purpose of tuning operation focuses on reducing resource consumption and reducing the response time of an operation. The main goal is to improve the efficient use of a particular resources. Most bottleneck elimination activities are as follows:
   a. Change in the way the application is used
   b. Change in oracle
   c. Change in host hardware configuration

AWR (Automatic Workload Repository):

Automatic workload repository collects processes and maintains performance statistics for problem detection and self-tuning process. This data is location in memory as well as stored in the database. The gather data can be displayed as view and reports.

The data collected and processed by AWR include:

1. Object statistical that determine both access and usage statistics of database segments.
2. Time model statistics based on time usage for activities.
3. Session statistics
4. SQL statement that are producing highest load in the system.
5. ASH (Active Session History)
   Statistics, representing the history of recent session activity.

Gathering database statistics using AWR is enabled default and is controlled by the STATISTICS_LEVEL initialization parameter. This parameter should be set to TYPICAL or ALL to enable statistics gathering by AWR.

Managing the Automatic Workload Repository):

1. Managing snapshots:
   By default oracle creates snapshots every one hour and retain the statistics in the workload Repository for 8 days. The primary interface for managing snapshots is Oracle Enterprise Manager (OEM). If OEM is unavailable it can be managed by using SQLPLUS by using DBMS_WORKLOAD_REPOSITORY packages following activities are performed during snapshots management:
   a. Creating snapshots
   b. Dropping snapshots
   c. Modification snapshot setting

2. **Managing Baselines:**
   The primary interfaces for managing baselines is OEM. If OEM is unavailable, it is manages from SQLPLUS using DBMS_WORKLOAD_REPOSITORY package. Managing baseline includes:
   a. Creating baseline
   b. Dropping baseline
   c. Renaming baseline
   d. Displaying the baseline metrics
   e. Modifying window size of the default moving window baseline

ADDM (Automatic Database Diagnostics Monitor)

When problem occur with system, it is important to perform accurate and timely diagnosis of the problem before making any changes to the system. In oracle, the statistical data needed for accurate diagnosis of a problem is stored in AWR and the ADDM performs following actions:

1. Analyzes the AWR data on a regular basis
2. Diagnosis the root causes of performance problems
3. Provides recommendations for correcting any problems
4. Identified non-perform areas of the system

AWR is repository of historical performance data, ADDM can analyze performance issues after the event. It saves times and resources that may be used to reproduce a problem.

Benefits of ADDM:

1. Automatic performance diagnostics report every hour by default.
2. Problems diagnosis
3. Time based quantification of problem impact and recommendations.
4. Identification of root cause, not symptom.
5. Recommendation for treating root cause of problem
6. Minimal overhead on system

ADDM Analysis:

An ADDM analysis can be performed on o pair of AWR snapshots and a set of instances from the same database. The pair of AWR snapshots define the time period for analysis, and the set of instance define the target for analysis.

In oracle RAC (Real Application Cluster), ADDM has three modes:

1. Database:
   In this modes, ADDM analyzes all instance of database.
2. Instance:
   In this modes, ADDM analyzes a particular instance of database.

3. Partial:

ADDM analyze a subset of all database instances.

The types of problems that ADDM considers are:

1. CPU Bottlenecks
2. Undersized memory structure
3. I/O capacity issues
4. High load SQL statements
5. Oracle RAC specifies issues
6. Sub-optimal used of oracle database by the application
7. Database configuration issues
8. Concurrency issues

SQL tuning advisor:

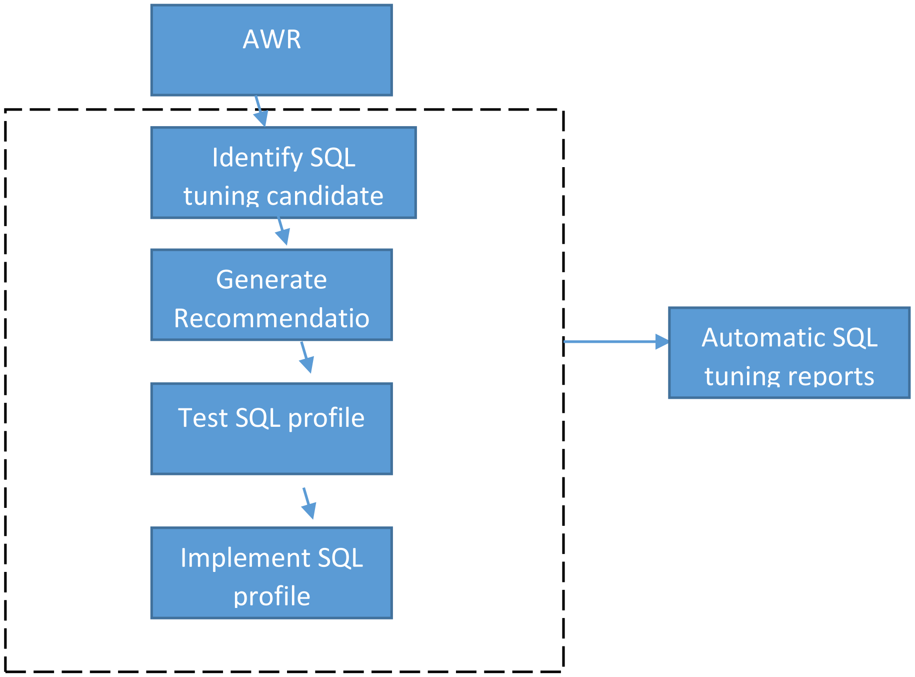
The SQL tuning advisor takes one or more SQL statement as input and invokes the automatic tuning optimizer to perform SQL tuning on the statements. The output of the SQL tuning advisor is in the form of advice or recommendation along its expected benefit. The recommendation relates to collection of statistics on objects, creation of new indexes, restructuring of the SQL statements.

Oracle database can automatically tune SQL statements by identifying problematic SQL statement and implementing tuning recommendation using tuning advisor during maintenance windows.

Automatic SQL Tuning Advisor:

Oracle database automatically runs the SQL tuning advisor on selected high-load SQL statement from Automatic Workload Repository (AWR) that quality as tuning candidates. It runs in the default maintenance windows on a nightly basis. Once SQL tuning begins, which by default runs for at most one hour during a maintenance windows, the following steps are performed:

1. Identify SQL candidates in the AWR for tuning.
2. Tune each statements individually by calling the SQL tuning advisor.
3. Test SQL profile by executing the SQL statements.
4. Optionally implement the SQL profile provided they meet the criteria of threefold performance improvement.



To enable automatic SQL tuning use the ENABLE procedure in DBMS_AUTO_TASK_ADMIN package.

```
BEGIN
DBMS_AUTO_TASK_ADMIN.ENABLE (
Client-name=> ‘sql tuning advisor’,
Operation => NULL,
Window-name => NULL);
END;
/
```

Configuring automatic SQL tuning:

The behavior of the automatic SQL tuning task can be configured using DBMS_SQLTUNE package. The DBMS_SQLTUNE packages enables DBA to configure automatic SQL tuning by specifying the task parameter using SQL_TUNING_TASK_PARAMETER produce.

Parameter	Description
ACCEPT_SQL_PROFILE	Specifies whether to accept SQL profile automatically
MAX_SQL_PROFILES_PER_EXEC	Specifies maximum number of profiles per SQL tuning task
MAX_AUTO_SQL_PROFILES	Specifies the limit of SQL profiles that are accepted in total
EXECUTION_DAYS_TO_EXPIRE	Specifies the number of days for which to save the task history

Virtual private database:

Virtual private database is a security feature of oracle database which allows user to see only privileged data. It enhances security of database by blocking access to unprivileged data within a database.

Virtual private database works with the help of security policies and context to ensure fine grained control on database objects. Virtual private database policy dynamically appends the WHERE clause in select queries and DML operations on the target table or views.

Example:

Employees table

Emp id	Emp Name	Job	Location
1	John	IT	AUS
2	Smith	IT	AUS
3	David	Account	NEP
4	Ram	Management	USA

If following select query is issued Australia

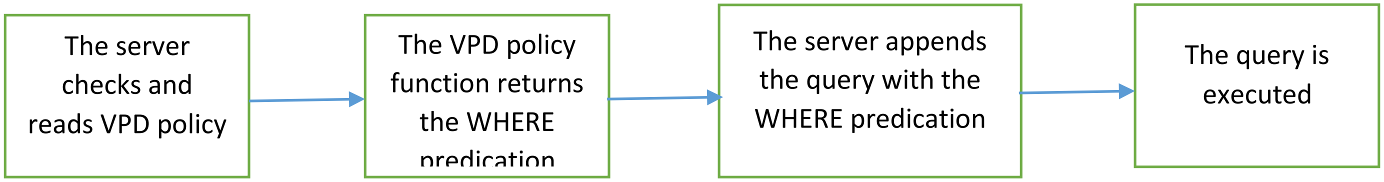
Select * from employee;

The VPD policy applied on target database automatically appends WHERE clause as

Select * from employee where location=”AUS”;

Working of VPD:

Whenever user tries to access a table, the VPD policy on table is activated.



The server checks and reads VPD policy

The VPD policy function returns the WHERE predication

The server appends the query with the WHERE predication

The query is executed

VPD implementation:

Implementation of VPD invokes two steps:

1. Creation of a policy function
2. Attachment of policy function to the hole

Creating a policy function:

A policy is a condition to filter the target data, which appears as virtual database to authorized user. During its evaluation, the server prepares the WHIEH clause which is then attached to user query.

Types of policy:

1. Static
2. Dynamic

Static policies are invoked once and store in the SGA (System Global Area), the same WHERE predicate is appended to the same user query. Whereas Dynamic policy executes every times when a user issues a query.

Example

Employees

Emp Name	Emp Job	Location
John	Dev	GER
Kate	Adm	GER
Miller	HR	AUS
Andy	Dev	AUS
Mac	Dev	AUS
Dan	Adm	RUS
Philip	HR	RUS

User SCOTT creates a policy function, which returns a condition to filter location by AUS

```
CREATE or REPLACE FUNCTION F_LOC_POLICY (P_schema varchar2, P_obj varchar2);

RETURN VARCHAR2

IS

BEGIN

RETURN ‘location=”AUS”’;

END;

/
```

Now, creating EMPLOCATION policy using DBMS_RLS package

```
BEGIN

Dbms_rls.add_policy( object_schema=&gt; ‘SCOTT’,object_name=&gt; ‘EMPLOYEES’, policy_name=&gt;
‘EMPLOCATION’, function_schema=&gt;’SCOTT’,policy_schema=&gt;’F_LOC_POLICY’,
statement_types=&gt;’select, insert’,update_check=&gt;’true’);

END;

/
```

Revision

Database corruption

Database corruption is a problem that causes some data stored in the database unreadable or corrupt. It is caused by hardware failure, software error or sometime user error, when it does occur, we should properly identify the affected objects and reuse or recover usable data.

Identifying Corruption

The typical method is to use dbv utility to scan the data file for errors

```
C:/> dbv file = c:\oracle\product\10.2.0\oradata\orcl\datafile.dbf blocksize = 8192;
```

Using RMAN

```
RMAN> Backup validate check logical database fileperset = 10;
```

Once validate is complete the check corruption using

V$Database_Block_Corruption

Lookup for data block address:

```
Select
            Dbms_utility.data_block_address_file_block() from dual;
```

Once data block is identified, the usable data is extracted using operating system utility bit by bit.

Database security:

➔Privileges

1. Object privilege
2. System privilege

➔ Role

➔ Authentication

1. OS authentication
2. Oracle password

VDP policy type:

1. Static policy
2. Dynamic policy

Column masking:

1. VDP
2. Specified column is display in the data base.