Table of Contents

[1 Overview of Oracle Database What is data? 3](#_Toc183188834)

[1.1 What is a Database? 3](#_Toc183188835)

[1.2 Types of databases: 3](#_Toc183188836)

[2 ARCHITECTURE 5](#_Toc183188837)

[3 RAM: 7](#_Toc183188838)

[3.1 SGA :( SHARED GLOBAL AREA/SYSTEM GLOBAL AREA) 7](#_Toc183188839)

[3.1.1 LIBRARY CACHE: 7](#_Toc183188840)

[3.1.2 DATA DICTIONARY CACHE: 7](#_Toc183188841)

[3.1.3 RESULT SET CACHE: 7](#_Toc183188842)

[3.1.4 CONTROL STRUCTURE: 7](#_Toc183188843)

[3.1.5 UGA (USER GLOBAL AREA): 7](#_Toc183188844)

[3.1.6 DATABASE BUFFER CACHE: 7](#_Toc183188845)

[3.1.7 STATES OF DATABASE BLOCKS IN DBBC: 9](#_Toc183188846)

[3.1.8 REDO BUFFER CACHE: 9](#_Toc183188847)

[3.2 PGA (PROGRAM GLOBAL AREAA): 9](#_Toc183188848)

[3.3 BACKGROUND PROCESS 9](#_Toc183188849)

[3.3.1 DATABASE WRITER (DBWR): 9](#_Toc183188850)

[3.3.2 LOG WRITER (LGWR) 10](#_Toc183188851)

[3.3.3 SMON (SYSTEM MONITOR): 10](#_Toc183188852)

[3.3.4 PMON (PROCESS MONITOR): 11](#_Toc183188853)

[3.3.5 MMON (MEMORY MONITOR): 11](#_Toc183188854)

[3.3.6 Check Point: 11](#_Toc183188855)

[3.3.7 ARCH: 11](#_Toc183188856)

[3.4 STORAGE 11](#_Toc183188857)

[3.4.1 CONTROL FILES: 11](#_Toc183188858)

[3.4.2 REDO LOG FILES: 12](#_Toc183188859)

[3.4.3 DATA FILES: 12](#_Toc183188860)

[4 DATABASE STARTUP PROCESS 14](#_Toc183188861)

[4.1 NO MOUNT STATE: 14](#_Toc183188862)

[4.2 MOUNT STATE: 14](#_Toc183188863)

[4.3 OPEN STATE: 15](#_Toc183188864)

[5 DATABASE SHUTDOWN PROCESS 16](#_Toc183188865)

[5.1 SHUTDOWN NORMAL: 16](#_Toc183188866)

[5.2 SHUTDOWN TRANSACTION: 16](#_Toc183188867)

[5.3 SHUT IMMEDIATE: 16](#_Toc183188868)

[5.4 SHUTDOWN ABORT: 16](#_Toc183188869)

[6 MULTIPLEXING OF CONTROL FILE 17](#_Toc183188870)

[7 TABLESPACE MANAGEMENT 18](#_Toc183188871)

[8 USER MANAGEMENT 19](#_Toc183188872)

[8.1 PROFILE: (view dba\_profiles) 20](#_Toc183188873)

[8.1.1 Create a new profile: 21](#_Toc183188874)

[8.1.2 Alter a profile: 21](#_Toc183188875)

[8.1.3 Change profile of an user: 21](#_Toc183188876)

[8.1.4 How to make a user non-expiry: 21](#_Toc183188877)

[8.2 PRIVILEGES: 22](#_Toc183188878)

[8.3 ROLE: 23](#_Toc183188879)

[9 BACKUP AND RECOVERY 25](#_Toc183188880)

[9.1 What is a Backup? 27](#_Toc183188881)

[9.2 RESTORE: 27](#_Toc183188882)

[9.3 RECOVERY: 27](#_Toc183188883)

[9.4 PHYSICAL BACKUP: 27](#_Toc183188884)

[10 SELECT PROCESS 28](#_Toc183188885)

[11 Update Process 29](#_Toc183188886)

# Overview of Oracle Database What is data?

* Data can be facts related to any object. ex: name,age,height,weight etc.
* A picture,image,file,pdf etc can also be considered as data.

## What is a Database?

A database is a systematic or organized collection of related information that is stored in such a way that it can be easily accessed,retrieved,managed and updated.

* In a database we can organize the data in rows and columns in the form of a table.
* MYSQL, SQL server, MongoDB, Oracle Database , Postgrace SQL and others are all different types of databases commonly used today.
* These modern databases are managed by Database Management System(DBMS). To interact with and manipulate the data stored in these databases a widely used language called Structured Query Language(SQL) is utilized.

## Types of databases:

Here are some popular types of databases.

1. Distributed databases:

A distributed database is a type of database that has contributions from the common database and

information captured by local computers. In this type of database system, the data is not in one place

and is distributed at various organizations.

1. Relational databases:

This type of database defines database relationships in the form of tables. It is also called Relational

DBMS, which is the most popular DBMS type in the market. Database examples of the RDBMS

system include MySQL, Oracle, and Microsoft SQL Server database.

1. Object-oriented databases:

This type of computers database supports the storage of all data types. The data is stored in the form of

objects. The data is stored in the form of objects. The objects to be held in the database have attributes

and methods that define what to do with the data. PostgreSQL is an example of an object-oriented

relational DBMS.

1. Cloud database:

A cloud database is a database which is optimized or built for such a virtualized environment. There are

so many advantages of a cloud database, some of which can pay for storage capacity and bandwidth. It

also offers scalability on-demand, along with high availability.

# ARCHITECTURE

|  |  |  |
| --- | --- | --- |
| **RAM** | **INSTANCE** | **SGA** |
| **PROCESSOR** | **BACKGROUND PROCESSES** |
| **STORAGE** | **DATABASE** | **CONTROL FILES, REDOLOG FILES,**  **DATA FILES** |

**REDO BUFFER CACHE**

**DATA DICTIONARY CACHE**

**RAM**

|  |
| --- |
| LIBRARY CACHE |
| DATA  DICTIONARY CACHE |
|  |
| RESULT SET CACHE |
| CONTROL STRUCTURE |
| USER GLOBALAREA |

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |

**SHARED POOL**

**BACK GROUND PROCESSES**

**SMON**

**PMON MMON**

**DBWR LGWR CKPT**

**ARCH**

**SERVER PROCESS**

**STORAGE**

**SP FILE**

**DATA FILES**

**REDOLOG FILES**

**P FILE**

CONTROL FILES

**PASSWORD**

**FILE**

# RAM:

* ORACLE architecture contains INSTANCE AND DATABASE.
* The instance is categorized in SGA and BACKGROUND PROCESSES.
* SGA is the combination of Shared pool, database buffer cache and redo buffer cache.
* Whereas smon , pmon , mmon , lgwr, dbwr, ckpt, arch are the mandatory background processes.
* Whereas the database contains control files, redo log files, data files and some more files like spfile , pfile, password file.

## SGA :( SHARED GLOBAL AREA/SYSTEM GLOBAL AREA)

SGA is sub categorized into library cache, data dictionary cache, result set cache, control structure, user

global area.

### LIBRARY CACHE:

LIBRARY CACHE consist of previously executed SQL statements history.

### DATA DICTIONARY CACHE:

DATA DICTIONARY CACHE consist of meta of objects like tables and views.

### RESULT SET CACHE:

RESULT SET CACHE is an area of memory that stores the result of db query for reuse purpose.

### CONTROL STRUCTURE:

CONTROL STRUCTURE consist of table level and row level lock information.

If any user tries to update the data in a table, the other user can’t update the data until

commit/rollback.

### UGA (USER GLOBAL AREA):

UGA is a memory in the place that is associated with user session.

The UGA normally comes from the session time, session login time.

### DATABASE BUFFER CACHE:

* DATABASE BUFFER CACHE is the mandatory component of SGA, and it is the largest component of SGA.
* DATABASE BUFFER CACHE consist of data blocks in which it stores the data.
* In case of retrival the data is copied from datafiles to database buffer cache by a process called server process.

**DATABASEBUFFER CACHE**

SERVER PROCESS

**DATA FILES**

* IN case of adding new data to database “database writer process” will copy data from dbbc to data files.

**DATABASE**

**BUFFER CACHE**

**DBWR**

**DATA FILES**

* When a user needs data from the database then oracle first check to see if the required block is already in the database buffer cache or not .

### STATES OF DATABASE BLOCKS IN DBBC:

* **UNUSED:** when the instance is started the blocks are in unused state.
* **PINNED:** the blocks which are reserved for a particular query.
* **DIRTY BLOCKS:** the blocks which are filled with data.
* **CLEAR:** when the data in the blocks is written to the datafiles then the blocks go to cleared state.

### REDO BUFFER CACHE:

* REDO BUFFER CACHE consists of change vectors happening on the database.
* Change vectors are generated in the case of create, drop, alter, update, insert commands are executed.
* The data in redo buffer cache is copied to redo log files by a background process called log writer (lgwr).

## PGA (PROGRAM GLOBAL AREAA):

* For Complex queries like group by, order by a big portion of memory is required. In those cases, operations will be done in PGA.

Redo log File 1

* If memory in PGA is not sufficient to perform such operations then the operation simultaneously takes place in temp data file BACKGROUND PROCESSES

## BACKGROUND PROCESS

### DATABASE WRITER (DBWR):

DATABASE WRITER is responsible for writing data from database buffer cache to data

files.

* Whenever a server process changes or modifies a data block, it becomes a dirty block. Once a server process makes changes to the data block the user may commit transactions or transactions may not be committed for some time. In either case, the dirty block is not immediately written back to disk.
* Writing dirty blocks to disk takes place under the following two conditions:

1. When a server process cannot find a clean, reusable buffer alter scanning a threshold number of buffers, then the database writer process writes the dirty blocks to disk.
2. When the check point takes place, the database writer process writes the dirty blocks to the disk.

### LOG WRITER (LGWR)

LGWR is responsible for writing change vectors from redo buffer cache to redo log files.

**LGWR writes change vectors to redo log files in the following scenarios:1**

1. For every 3 seconds
2. Whenever a commit is issued.
3. Before dbwr writes data into data files.
4. When 1MB of data is filled in redo buffer cache.
5. When 1/3 of redo buffer cache is filled.

### SMON (SYSTEM MONITOR):

SMON is a background which is responsible for:

1. Instance recovery
2. Merge two free extents and create a big extent.

Instance Recovery:

Instance recovery means rolling forward and rolling backward the changes on the

database.

When an instance is crashed changes made to the SGA may not be written to the

respective data files ,

when we restart the instance smon background process will automatically perform

instance recovery by performing.

1. Rolling Forward
2. Rolling Backward
3. ROLLING FORWARD:

Transactions which are committed but not written into the data files have to be

written to data files.

Changes which are made to the online redo log files but not the database, so their

changes are written to data files by rolling forward changes from online redo logs to

the data files.

1. ROLLING BACKWARD:

Dirty buffers which are written into data files but not committed, have to be rolled

back from undo table space.

### PMON (PROCESS MONITOR):

* PMON is going to copy metadata from system files to data dictionary cache.
* PMON is going to release orphan locks.
* When there is a new connection PMON is going to authenticate the login details.

### MMON (MEMORY MONITOR):

MMON collects memory statistics from SGA periodically and stores them as snapshot and writes

these statistics to sysaux data file.

### Check Point:

* CHECKPOINT monitors dirty buffers on database buffer cache and signal DBWR to write dirty buffers into data files.
* CHECKPOINT process update control file, redo log files, data files with latest SCN.

### ARCH:

ARCH will turn redo log files into archive log files.

ARCH will copy the content of redo log files to archive log files, whenever a log switch occurs.

## STORAGE

### CONTROL FILES:

A control file is a small binary file that records the physical structure of a database. Control files

consist of two parts:

1. Non-reusable section:

It consists of

1. Database name
2. The time stamp of database creation.
3. Names and locations of associated data files and redo log files.
4. Current log sequence number.
5. Check point information.
6. Reusable section:

* The reusable section of the control file will contain RMAN backup information.
* Without control file the database cannot be mounted, and recovery is difficult.
* We can have a maximum of 8 control files.

### REDO LOG FILES:

* REDO LOG FILES stores all the change vectors happening on the database.
* If archiving is disabled, a filled redo log file is available after the changes recorded in it have been written to the data files.
* If archiving is enabled, a filled redo log file is available to LGWR after the changes recorded in it have been written to the data files and the file has been archived.

STATUS OF REDO LOG FILE:

1. Current:The redo log file to which the log writer is currently writing.
2. Active: The redo log files which are needed for instance recovery.
3. Inactive: The redo log files which are not required, for instance recovery.
4. Unused:The redo log files which are not used yet.

#### SCN (SYSTEM CHANGE NUMBER):

Whenever a transaction is committed, LGWR writes the transaction from redo buffer

cache of SGA to a redo log file, and SCN to identify the redo records for each committed

transaction.

1. **LOG SWITCH**:

A log switch is the point at which log writer stops writing to one redo log file and

starts writing to another. Normally, a log switch occurs when the current redo log file

is completely filled, and writing must continue to the next redo log file.

1. **LOG SEQUENCE NUMBER:**

Oracle database assigns each redo log file a new log sequence number every time a

log switch occurs and LGWR begins writing to it.

### DATA FILES:

#### SYSTEM:

* This table space contains the data dictionary, which is the central set of tables and views that contain administrative information about the database.
* It also contains meta data.
* These are all contained in the sys schema and can be accessed only sys schema or other administrative user with required privilege.

#### SYSAUX:

* Sysaux table space contains the statistical data that is collected by MMON background process.

#### TEMP

* This table space is useful when the memory in PGA is not sufficient to carry out the sorting operations like group by, order by.

#### UNDO:

* When a transaction modifies data, Oracle database copies the original data before modifying it.
* The original copy of the modified data is called undo data and is stored in undo table space.

# DATABASE STARTUP PROCESS

ORACLE database startup process has 3 steps

A diagram of a state

Description automatically generated

## NO MOUNT STATE:

* In order to go to no mount state oracle will scan either pfile or spfile in $ORACLE\_HOME and based on initialization parameters oracle will create an instance.
* If both are existing priority will be given to spfile.
* Spfile consist of the location of control file.

## MOUNT STATE:

* In order to go to the mount state from no mount state oracle will scan inside the control file for physical existence of data files and redo log files.
* Mounting is nothing but associating an instance with database.

## OPEN STATE:

* For a database to go from mount state to open, oracle will check for the physical locations of redo log and data files mentioned in control file.
* If all the files exist then oracle will check whether control, redo and data files are in sync with scn or not.
* If they are not in sync, then smon will perform instance recovery. After that the database will be opened.

# DATABASE SHUTDOWN PROCESS

## SHUTDOWN NORMAL:

* This is a graceful shutdown.
* No new connections are allowed.
* ORACLE will wait for all the users to disconnect from their sessions.
* Once all the sessions are disconnected oracle will first dismount the database.And, oracle will release the memory occupied by the instance in ram.

## SHUTDOWN TRANSACTION:

* This is a graceful shutdown.
* NO new connections are allowed.
* Oracle will wait for all the ongoing transactions are completed, either by commit (or) rollback.

## SHUT IMMEDIATE:

* This is also a graceful shutdown.
* No new connections are allowed.
* Oracle will rollback all the ongoing transactions and disconnect all the user sessions and then dismounts the database and releases the memory occupied by an instance in ram close the database.

## SHUTDOWN ABORT:

* This is not a graceful shutdown.
* Next startup requires an instance recovery.
* When we use shut abort oracle will terminate the instance immediately without writing database cache dirty buffers to data files.

# MULTIPLEXING OF CONTROL FILE

1. **What is multiplexing of control file?**

Multiplexing is the process of maintaining a copy of the same control files on different disk drivers. To multiplex your control files, we copy the control file to multiple locations.

1. **Why to do Multiplexing?**

If you have multiple control files and one of them gets damaged, then you can use the other control files to get your database operational in no time. That is why multiplexing of control files is important.

**Step1: Set the control files parameter.**

SQL> Alter System set CONTROL\_FILES = 'u02/oradata/ord/control01.ctl', 'u03/oradata/ord/control01.ctl', 'u04/oradata/ord/control01.ctl', 'u05/oradata/ord/control01.ctl' scope = spfile;

**Step2: Shut down the instance**

SQL>shutdown;

**Step3: Copy the control file to more locations using operating system command.**

$ cp u02/app/oradata/ord/control01.ctl u05/app/oradata/ord/control02.ctl

**Step4: Start the Database**

SQL> startup;

# TABLESPACE MANAGEMENT

1. **Create a NEW table space:**

SQL> create table space DATA datafile ‘/u01/app/oracle/oradata/data01.dbf’ size 10G autoextend on next 500M;

**Note:** 500M of extent will be added to the datafile automatically, when the space is required.

1. **Adding a datafile:**

SQL> alter tablespace DATA add datafile ‘/u01/app/oracle/oradata/data02.dbf’ size 2G;

1. **Resize a datafile: (To increase or decrease the datafile):**

SQL> alter database datafile ‘/u01/app/oracle/oradata/data02.dbf’ resize 3G;

1. **Change default tablespace:**

To know the default permanent tablespace.

SQL> select property value from database properties Where property name=’default\_permanent\_tablespace’;

To change the default tablespace

SQL> alter database default tablespace DATA;

# USER MANAGEMENT

1. **Create a user:**

SQL>create user <> identified by <password> PROFILE DEFAULT

DEFAULT TABLESPACE USERS TEMPORARY TABLESPACE TEMP;

* Minimum privilege required to connect to a database is a create session.

SQL>grant create session to <username>

1. **Change password of a user:**

SQL>alter user <username>identified by <new password>;

1. **Lock/unlock a user:**

SQL>alter user <username> account lock;

SQL>alter user <username> account unlock;

1. **Make a user password expiry:**

* When we make a user id expiry, then when the user does login, it will prompt him to set a new password.

SQL>alter user <username> account expire;

1. **Changing** **default tablespace of a user:**

SQL>select username, default\_tablespace from dba\_users where username='<username>';

SQL>alter user <username>default tablespace <tablespace name>;

SQL>select username, default\_tablespace from dba\_users where username='<username>';

## PROFILE: (view dba\_profiles)

A profile enforces set of password security rules and resource usage limit. While creating a user if no profile is mentioned, then DEFAULT profile will be assigned.

SQL> desc dba\_profiles;

SQL>col limit for a12 col profile for a14

set lines 200

set pagesize 200

select profile,resource\_name, RESOURCE\_TYPE, limit from dba\_profiles where profile='DEFAULT';

**PARAMETERS :-**

\*SESSION\_PER\_USER – No. of allowed concurrent sessions for a user

\*CPU\_PER\_SESSION – CPU time limit for a session, expressed in hundredth of

seconds.

\*CPU\_PER\_CALL – Specify the CPU time limit for a call (a parse, execute, or fetch), expressed in hundredths of seconds.

\*CONNECT\_TIME – Specify the total elapsed time limit for a session, expressed in minutes.

\*IDLE\_TIME – Specify the permitted periods of continuous inactive time during a session, expressed in minutes.

\*LOGICAL\_READS\_PER\_SESSION – Specify the permitted number of data blocks read in a session, including blocks read from memory and disk

\*LOGICAL\_READS\_PER\_CALL –permitted number of data blocks read for a call to process a SQL statement (a parse, execute, or fetch).

\*PRIVATE\_SGA – SGA a session can allocate in the shared pool of the system global area (SGA), expressed in bytes.

\*FAILED\_LOGIN\_ATTEMPTS – No. of failed attempts to log in to the user account before the account is locked

\*PASSWORD\_LIFE\_TIME: No. of days the account will be open. after that it will expiry.

\*PASSWORD\_REUSE\_TIME: number of days before which a password cannot be reused

\*PASSWORD\_REUSE\_MAX: number of days before which a password can be reused

\*PASSWORD\_LOCK\_TIME: Number of days the user account remains locked after failed login

\*PASSWORD\_GRACE\_TIME: Number of grace days for user to change password

\*PASSWORD\_VERIFY\_FUNCTION: PL/SQL that can be used for password verification.

### Create a new profile:

CREATE PROFILE "<profile name>" LIMIT

COMPOSITE\_LIMIT UNLIMITED SESSIONS\_PER\_USER UNLIMITED CPU\_PER\_SESSION UNLIMITED CPU\_PER\_CALL UNLIMITED LOGICAL\_READS\_PER\_SESSION UNLIMITED LOGICAL\_READS\_PER\_CALL UNLIMITED IDLE\_TIME 90

CONNECT\_TIME UNLIMITED PRIVATE\_SGA UNLIMITED FAILED\_LOGIN\_ATTEMPTS 10

PASSWORD\_LIFE\_TIME 180 PASSWORD\_REUSE\_TIME UNLIMITED PASSWORD\_REUSE\_MAX UNLIMITED PASSWORD\_VERIFY\_FUNCTION NULL PASSWORD\_LOCK\_TIME UNLIMITED PASSWORD\_GRACE\_TIME UNLIMITED;

### Alter a profile:

SQL>ALTER PROFILE <profile name> LIMIT FAILED\_LOGIN\_ATTEMPS UNLIMITED;

### Change profile of an user:

SQL> select username, profile from dba\_users where username='<username>';

SQL>ALTER USER <username> PROFILE <profile name>;

### How to make a user non-expiry:

SQL> select username, profile, EXPIRY\_DATE from dba\_users where username='<username>';

SQL> ALTER PROFILE <profile name>LIMIT PASSWORD\_LIFE\_TIME UNLIMITED;

SQL> select username, profile, EXPIRY\_DATE from dba\_users where username='<username>';

## PRIVILEGES:

A privilege is a permission to execute either a particular type of sql statements or to perform particular action on database objects.

Two type of privilege:

1. SYSTEM PRIVILEGE
2. OBJECT PRIVILEGE
   1. **SYSTEM PRIVILEGE**

A system privilege is the right to perform a particular action or to perform an action on any object of a particular type.

**List of all system privileges:**

SQL>select distinct privilege from dba\_sys\_privs;

1. Grant a System Privilege to a User:

SQL> select privilege, grantee from dba\_sys\_privs where grantee='<username>';

1. Revoke a system privilege from a user:

SQL>REVOKE create any table from <username>;

* 1. **OBJECT PRIVILEGE:**

An object privilege is the right to perform a particular action on an object or to access another user’s object.

1. List of object privileges:

SQL> select distinct privilege from DBA\_TAB\_PRIVS;

1. Grant objects privilege:

SQL>grant insert, update, delete on <table name> to <username>;

**View the granted object privilege:**

SQL>select grantee, owner, table\_name, privilege from dba\_tab\_privs where grantee='<username>';

1. Revoke object privilege:

SQL>revoke update on <table name> from <username>;

## ROLE:

A role is a collection of privileges. It allows easier management of privileges.

1. **Create a role:**

SQL>create role <role name>;

1. **Grant privileges to a role:**

SQL>grant create session to <role name>;

SQL>grant select any table to;

SQL>grant insert on siebel.test2 to dev\_role;

1. **List of SYSTEM privileges granted to a ROLE:**

SQL>select role,privilege from role\_sys\_privs where role='<user name>';

1. **List of OBJECT privileges granted to ROLE**

SQL> select role, owner, table\_name, privilege from role\_tab\_privs where role='<user name>';

1. **Grant role to a User:**

SQL>grant <role name>to <user name>;

1. **List of the user and granted role:**

SQL> select grantee,GRANTED\_ROLE from dba\_role\_privs where granted\_role='<role name>';

1. **Drop a user:**

Dropping a user will drop all the objects it owns.

SQL>drop user <user name> cascade;

**Note:** CASCADE is used to drop all objects in a user's schema before dropping the user.

1. **Drop a Role:**

SQL> drop role <role name>;

# BACKUP AND RECOVERY

**BACKUP AND RECOVERY**

Physical

Backup

Logical Backup

RMAN

Backup

DATA

Pump

**FULL BACKUP**

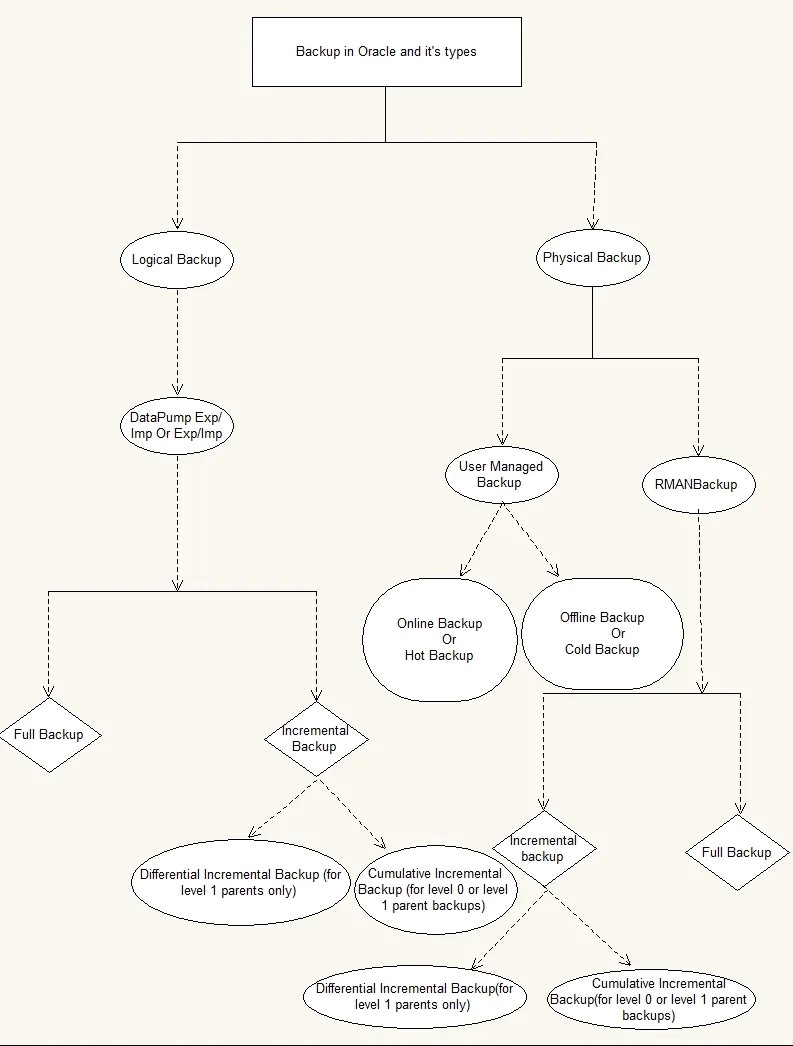
**INCREMENTAL BACKUP**

**CUMULATIVE BACKUP**

**EXPDP**

**IMPDP**

Expdp Impdp



## What is a Backup?

A backup is a copy of data. This copy can include important parts of the database, such as the control file and datafiles. A backup is a safeguard against unexpected data loss and application errors. If you lose the original data, then you can reconstruct it by using a backup.

Backups are divided into physical backups and logical backups. Physical backups, which are the primary concern in a backup and recovery strategy, are copies of physical database files. You can make physical backups with either the Recovery Manager (RMAN) utility or operating system utilities. In contrast, logical backups contain logical data (for example, tables and stored procedures) extracted with an Oracle utility and stored in a binary file. You can use logical backups to supplement physical backups.

## RESTORE:

Data restore is the process of copying backup data from secondary storage and restoring it to its original location or a new location.

## RECOVERY:

To recover a restored datafile is to update it by applying archived redo logs and online redo logs, that is, records of changes made to the database after the backup was taken.

## PHYSICAL BACKUP:

Physical backups are backups of physical database files: datafiles and control files. If you run the database in ARCHIVELOG mode, then the database also generates archived redo logs. You can back up the datafiles, control files, and archived redo logs.

# SELECT PROCESS

The moment user gives select, Oracle will check in library cache, if the query is existed there.

If the query is already existing in library cache, then as per the existing execution plan, data will be fetched by server process and handover to user process.

If The query not there in library cache Oracle will perform syntactical checks and semantical checks by using metadata existing in data dictionary cache.

1. syntactical checks means --checking if the syntax is correct or not.
2. semantical checks means --checking If the tables and columns in that query are existing and checking if the user who is querying the query has privileges on the table.

once parsing is done optimiser will generate a best execution plan based on the statistics available.

Now based on the generated plan server process will copy the resulting data from data files to database buffer cache and also writes a record entry in library cache take this query executed for the result of queried again.

# UPDATE PROCESS

When a user gives update statement this request will be taken by server process and initially Oracle will check if this query is executed recently library cache.

If this query is not executed recently in library cache, then Oracle will perform parsing using the metadata existing in data dictionary cache.

Parsing is nothing but syntactical checks and symantical checks optimizers will generate execution plan based on the statistics, execution plan generated by optimizer.

Update the table

Library cache

Parsing

Optimizer execution

Fetching

DBBC and

Undo table space

**Rollback**

**Commit**

**Datafiles**

Server process will fetch the data to be updated from database buffer cache and copy past image of data is kept undo tablespace. Now the data kept in database buffer cache is updated to new value.

Now user left with two options either commit or rollback. If the user is giving commit the data which is in database buffer cache will be made permanent to the database.

If the user gives rollback the past image from undo will be copied back to the data files