Lead Training

2nd Batch 2025

Date: 17/07/2025

DOYENSYS
VALUE INNOVATION

DIL
Doyensys Institute of Learning

Table of contents

01 Linux Basic commands

02 Cronjobs and shell scripts basics

03 Introduction to database – Oracle Architecture

04 Getting started with oracle server

05 Managing an oracle instance

06 Database creation

07 Dictionary & Dynamic Performance views

2

Introduction to database

Database Administrator

- Introduction
- Who is a DBA?
- Database editions
- DBA roles and Responsibilities
- Day to Day activities
- Oracle database architecture overview
- Instance
- SGA and PGA

Database

A database is a structured collection of data that is organized and stored in a way that allows for efficient retrieval, management, and manipulation of information. Databases are fundamental to storing and organizing vast amounts of data in a structured and accessible manner. They serve as the foundation for various applications and are crucial in many aspects of information technology.

Key Components of a Database:

- **Data:**
  Databases store data, which can be in various forms such as text, numbers, images, etc.
- **Tables:**
  Data in a database is organized into tables, which consist of rows and columns. Each row represents a record, while each column represents a field or attribute.
- **Schema:**
  The structure or design of the database is defined by its schema. It outlines the tables, fields, relationships, and constraints.
- **Queries:**
  Queries are used to retrieve, insert, update, or delete data in the database. SQL (Structured Query Language) is a common language used to interact with relational databases.

Database

- **Indexes:**
  Indexes are used to optimize data retrieval operations. They enhance the speed of data retrieval by providing quick access to specific rows based on the values in certain columns.

- **Relationships:**
  Relationships define how tables are connected to each other. Primary keys and foreign keys establish relationships between tables.

- **Normalization:**
  Normalization is the process of organizing data to eliminate redundancy and dependency. It involves dividing large tables into smaller, related tables.

Types of Databases:

- **Relational Databases:**
  Organize data into tables with predefined relationships between them. Examples include MySQL, PostgreSQL, Oracle Database, and Microsoft SQL Server.

- **NoSQL Databases:**
  Designed for handling large volumes of unstructured or semi-structured data. Types include document-oriented (MongoDB), key-value stores (Redis), and column-family stores (Cassandra).

- **Graph Databases:**
  Focus on representing and querying relationships between entities. Examples include Neo4j and Amazon Neptune.

- **In-Memory Databases:**
  Store data in the system's main memory (RAM) for faster data retrieval. Examples include Redis and SAP HANA.

Advantages of Databases:

- **Data Integrity:**
  Databases enforce rules and constraints to ensure data accuracy and consistency.
- **Data Security:**
  Access to data can be restricted based on user roles and permissions.
- **Efficient Retrieval:**
  Databases use indexes and optimization techniques to retrieve data quickly.
- **Scalability:**
  Databases can handle large amounts of data and scale horizontally or vertically to accommodate growth.
- **Data Redundancy Reduction:**
  Normalization techniques reduce data redundancy, minimizing storage space.
- **Concurrency Control:**
  Databases manage multiple users accessing the data simultaneously, ensuring data consistency.
- **Data Recovery:**
  Regular backups and recovery mechanisms protect against data loss.

DBA roles and Responsibilities (Day to Day Activities)

- Database Design and Architecture
- Database Installation and Configuration
- Backup and Recovery
- Security Management
- Performance Monitoring and Tuning
- Capacity Planning
- Database Upgrades and Patching
- Data Migration
- Documentation
- Collaboration with Development Teams
- Troubleshooting and Support
- Compliance and Auditing
- Disaster Recovery Planning

Oracle Database Editions

Oracle Database Standard Edition 2

Oracle Database Standard Edition 2 delivers unprecedented ease of use, power, and performance for workgroup, department-level, and Web applications.
Note: Oracle Database Standard Edition 2 is available starting with Oracle Database 12c Release 1 (12.1.0.2). For 12.1.0.1, Oracle Database Standard Edition One and Oracle Database Standard Edition are available.

Oracle Database Enterprise Edition

Oracle Database Enterprise Edition provides the performance, availability, scalability, and security required for mission-critical applications such as high-volume online transaction processing (OLTP) applications, query-intensive data warehouses, and demanding Internet applications. Oracle Database Enterprise Edition contains all of the components of Oracle Database.

Oracle Database Editions

Oracle Database Express Edition

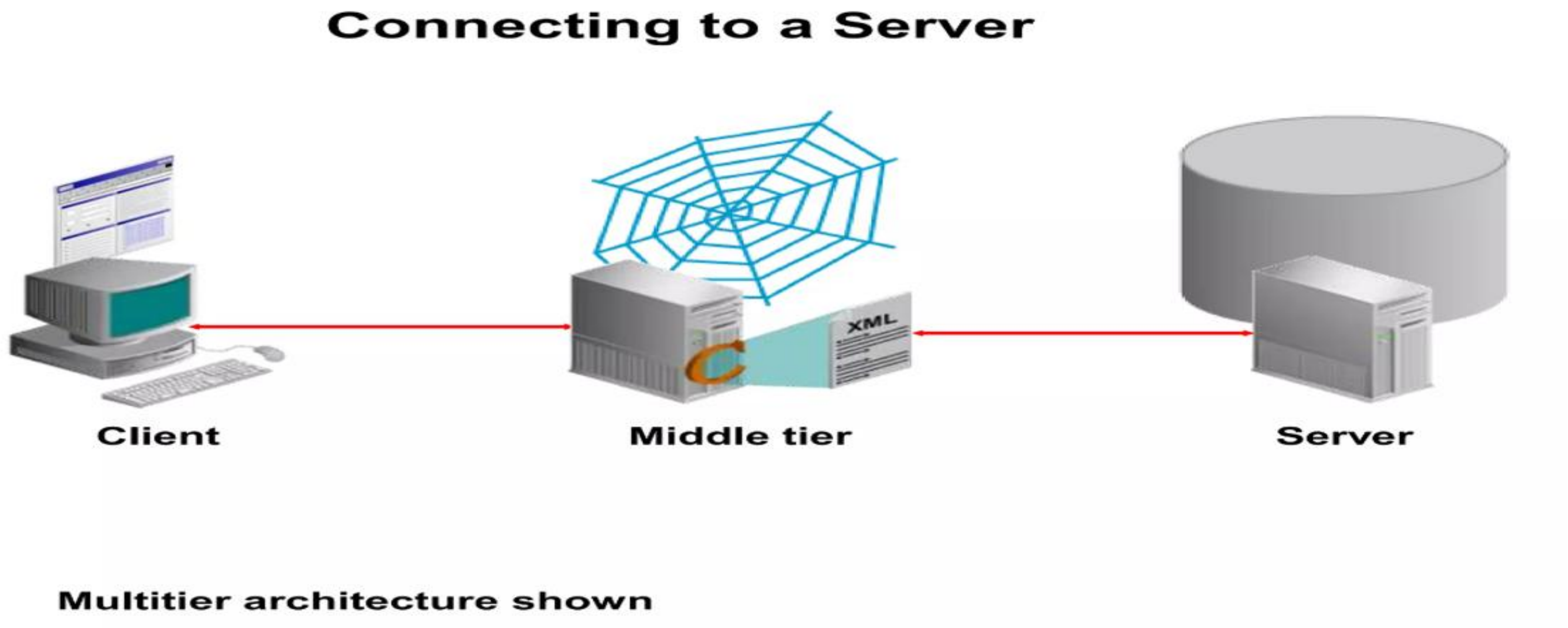
Oracle Database Express Edition (Oracle Database XE) is an entry-level edition of Oracle Database that is quick to download, simple to install and manage, and is free to develop, deploy, and distribute. Oracle Database XE makes it easy to upgrade to the other editions of Oracle without costly and complex migrations. Oracle Database XE can be installed on any size machine with any number of CPUs, stores up to 11 GB of user data, using up to 1 GB of memory, and using only one CPU on the host machine. Support is provided by an online forum. The current version of Oracle Database XE is Oracle Database 11*g* Release 2.

Oracle Database Personal Edition

Oracle Database Personal Edition supports single-user development and deployment environments that require full compatibility with Oracle Database Standard Edition One, Oracle Database Standard Edition, and Oracle Database Enterprise Edition. Personal Edition includes all of the components that are included with Enterprise Edition, as well as all of the options that are available with Enterprise Edition, with the exception of the Oracle Real Application Clusters option, which cannot be used with Personal Edition. Personal Edition is available on Windows and Linux platforms only. The Management Packs are not included in Personal Edition.

Oracle database architecture overview

Oracle Database follows a multi-tiered architecture that provides a highly scalable and flexible platform for managing and processing data. The architecture of Oracle Database can be broadly categorized into the following components.



Oracle database architecture overview

Client Tier:

- Client Applications: These are the end-user applications that interact with the Oracle Database. Examples include web applications, desktop applications, or command-line tools.
- Oracle Net Services: Handles communication between the client applications and the database server. It includes protocols such as TCP/IP and Oracle's SQL*Net.

Middle Tier:

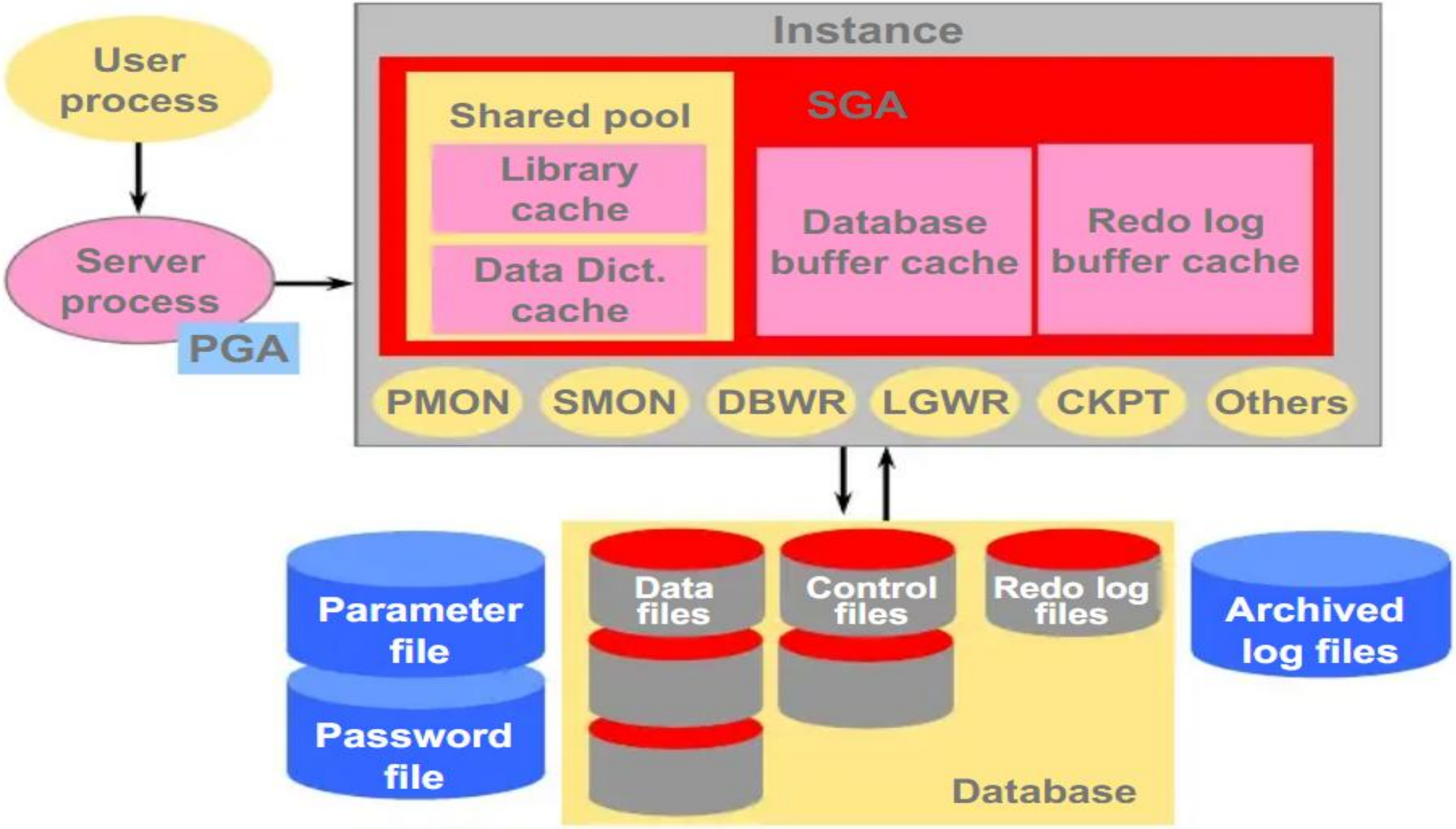
- Application Server: In some architectures, there is an intermediate application server layer that hosts middleware components, which mediate communication between client applications and the database. Oracle provides products like Oracle WebLogic Server for this purpose.

Oracle database architecture overview

Database Tier:

- Oracle Database Instance: The database instance is a set of memory structures and background processes that manage database operations. It includes components such as the System Global Area (SGA) and the background processes (e.g., processes for managing cache, processes for managing transactions).
- Data Files: Physical files on disk that store the actual data and metadata of the database. These include data files, control files, and redo log files.
- Tablespaces: Logical storage containers within the database that organize and group related data files.

Oracle database architecture



Database instance and Memory structure

An Oracle instance refers to the combination of memory structures and background processes that collectively manage and provide access to an Oracle Database. When a user connects to an Oracle Database, they are interacting with a specific instance. Each running Oracle Database is associated with one and only one instance.

The key components of Oracle database instance.

System Global Area (SGA):

- The SGA is a shared memory region that is allocated when an Oracle instance starts. It contains data and control information that is shared among all the connected user sessions and background processes.
- Key components of the SGA include the Database Buffer Cache, Shared Pool, Redo Log Buffer, and other structures.

Program Global Area (PGA):

- The PGA is a memory region that is allocated for each individual user session or background process. It contains data and control information specific to that session or process.
- Each user connecting to the database has their own PGA.

SGA and PGA in oracle instance

The SGA is divided into several components, each serving a specific purpose. Some of the key components include:

Buffer Cache (Database Buffer Cache):
This component stores copies of data blocks read from data files. It helps in reducing physical I/O by caching frequently accessed data in memory.

Buffer States:
A buffer in the Database Buffer Cache can be in one of several states, including:
Free: The buffer is not currently being used.
Pinned: The buffer is in use by a process.
Dirty: The buffer has been modified and needs to be written back to the data file.
Touch: The buffer has been accessed but not modified.

Shared Pool:
It contains the library cache, which stores parsed SQL statements and execution plans, and the data dictionary cache, which stores information about the database's structure.

SGA and PGA in oracle instance

The SGA is divided into several components, each serving a specific purpose. Some of the key components include:

Java Pool:
The Java Pool, part of the SGA, is used for managing Java execution within the database.

Large Pool:
This is an optional area used for large memory allocations, such as backup and restore operations.

Redo Log Buffer:
It holds information about changes made to the database, which is necessary for recovery in the event of a system failure.

SGA and PGA in oracle instance

Background Processes and process structure

These are dedicated processes that run in the background and perform various tasks to manage and maintain the database. Important background processes include:

Process Monitor Process (PMON)

The process monitor (PMON) monitors the other background processes and performs process recovery when a server or dispatcher process terminates abnormally. PMON is responsible for cleaning up the database buffer cache and freeing resources that the client process was using. For example, PMON resets the status of the active transaction table, releases locks that are no longer required, and removes the process ID from the list of active processes.
PMON also registers information about the instance and dispatcher processes with the Oracle Net listener . When an instance starts, PMON polls the listener to determine whether it is running. If the listener is running, then PMON passes it relevant parameters. If it is not running, then PMON periodically attempts to contact it.

Background Process

System Monitor Process (SMON):

The system monitor process (SMON) is in charge of a variety of system-level cleanup duties. The duties assigned to SMON include:

Performing instance recovery, if necessary, at instance startup. In an Oracle RAC database, the SMON process of one database instance can perform instance recovery for a failed instance.

Recovering terminated transactions that were skipped during instance recovery because of file-read or tablespace offline errors. SMON recovers the transactions when the tablespace or file is brought back online.

Cleaning up unused temporary segments. For example, Oracle Database allocates extents when creating an index. If the operation fails, then SMON cleans up the temporary space.

Coalescing contiguous free extents within dictionary-managed tablespaces.

SMON checks regularly to see whether it is needed. Other processes can call SMON if they detect a need for it.

Background Process

Database Writer Process (DBW*n*)
The database writer process (DBWn) writes the contents of database buffers to data files. DBWn processes write modified buffers in the database buffer cache to disk.

Although one database writer process (DBW0) is adequate for most systems, you can configure additional processes—DBW1 through DBW9 and DBWa through DBWj—to improve write performance if your system modifies data heavily. These additional DBWn processes are not useful on uniprocessor systems.

The DBWn process writes dirty buffers to disk under the following conditions:

- When a server process cannot find a clean reusable buffer after scanning a threshold number of buffers, it signals DBWn to write. DBWn writes dirty buffers to disk asynchronously if possible while performing other processing.

- DBWn periodically writes buffers to advance the checkpoint, which is the position in the redo thread from which instance recovery begins. The log position of the checkpoint is determined by the oldest dirty buffer in the buffer cache.

In many cases the blocks that DBWn writes are scattered throughout the disk. Thus, the writes tend to be slower than the sequential writes performed by LGWR. DBWn performs multiblock writes when possible, to improve efficiency. The number of blocks written in a multiblock write varies by operating system.

Background Process

Log Writer Process (LGWR)

The **log writer process (LGWR)** manages the redo log buffer. LGWR writes one contiguous portion of the buffer to the online redo log.
By separating the tasks of modifying database buffers, performing scattered writes of dirty buffers to disk, and performing fast sequential writes of redo to disk, the database improves performance.

In the following circumstances, LGWR writes all redo entries that have been copied into the buffer since the last time it wrote:

- A user commits a transaction.
- An online redo log switch occurs.
- Three seconds have passed since LGWR last wrote.
- The redo log buffer is one-third full or contains 1 MB of buffered data.
- DBW*n* must write modified buffers to disk.

Before DBW*n* can write a dirty buffer, redo records associated with changes to the buffer must be written to disk (the write-ahead protocol).
If DBW*n* finds that some redo records have not been written, it signals LGWR to write the records to disk and waits for LGWR to complete before writing the data buffers to disk.

Background Process

LGWR and Commits:

Oracle Database uses a **fast commit** mechanism to improve performance for committed transactions. When a user issues a COMMIT statement, the transaction is assigned a system change number (SCN). LGWR puts a commit record in the redo log buffer and writes it to disk immediately, along with the commit SCN and transaction's redo entries.

The redo log buffer is circular. When LGWR writes redo entries from the redo log buffer to an online redo log file, server processes can copy new entries over the entries in the redo log buffer that have been written to disk.

LGWR normally writes fast enough to ensure that space is always available in the buffer for new entries, even when access to the online redo log is heavy.

The atomic write of the redo entry containing the transaction's commit record is the single event that determines the transaction has committed.

Oracle Database returns a success code to the committing transaction although the data buffers have not yet been written to disk.

The corresponding changes to data blocks are deferred until it is efficient for DBW*n* to write them to the data files.

Background Process

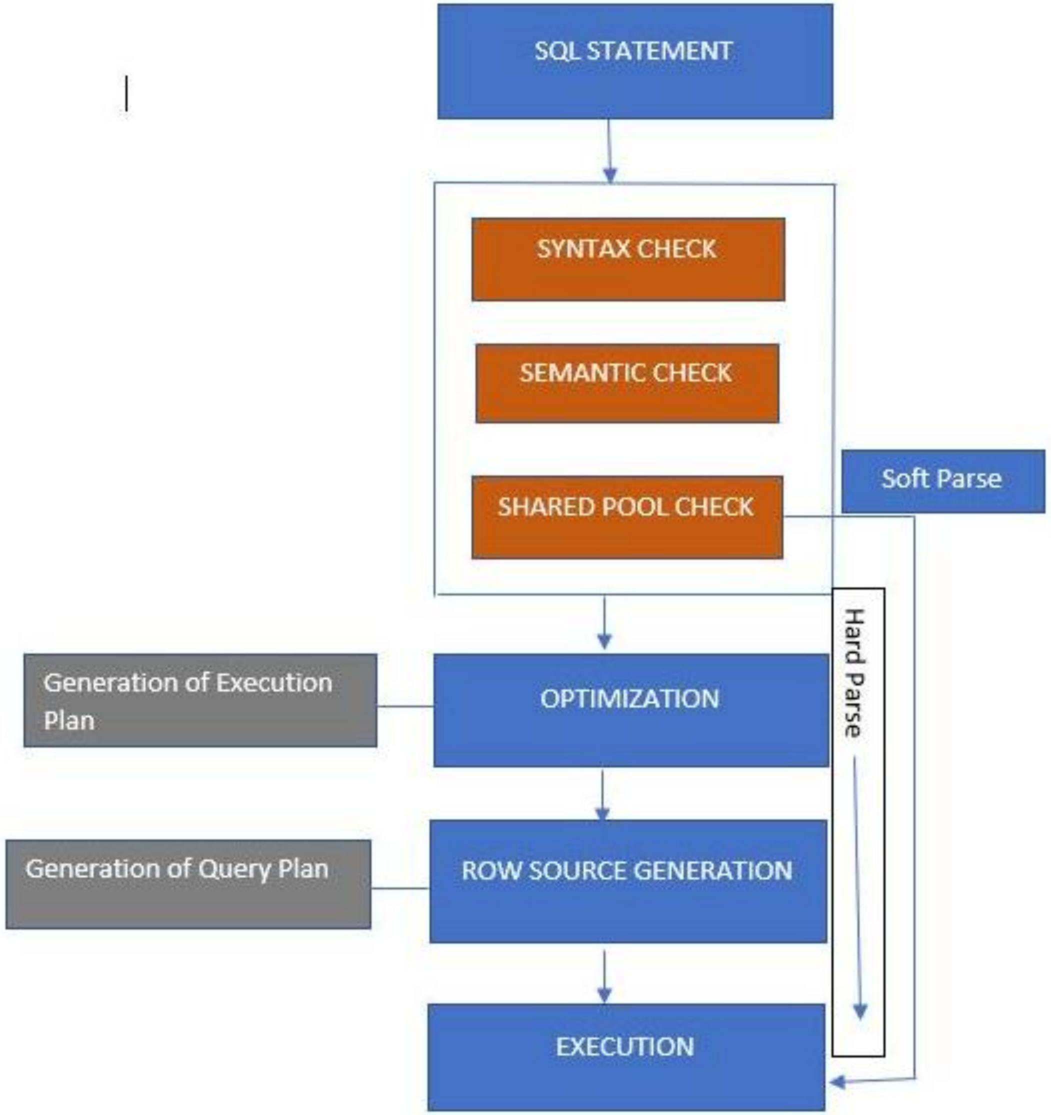
Checkpoint Process (CKPT)

The checkpoint process (CKPT) updates the control file and data file headers with checkpoint information and signals DBW*n* to write blocks to disk. Checkpoint information includes the checkpoint position, SCN, location in online redo log to begin recovery, and so on. CKPT does not write data blocks to data files or redo blocks to online redo log files.

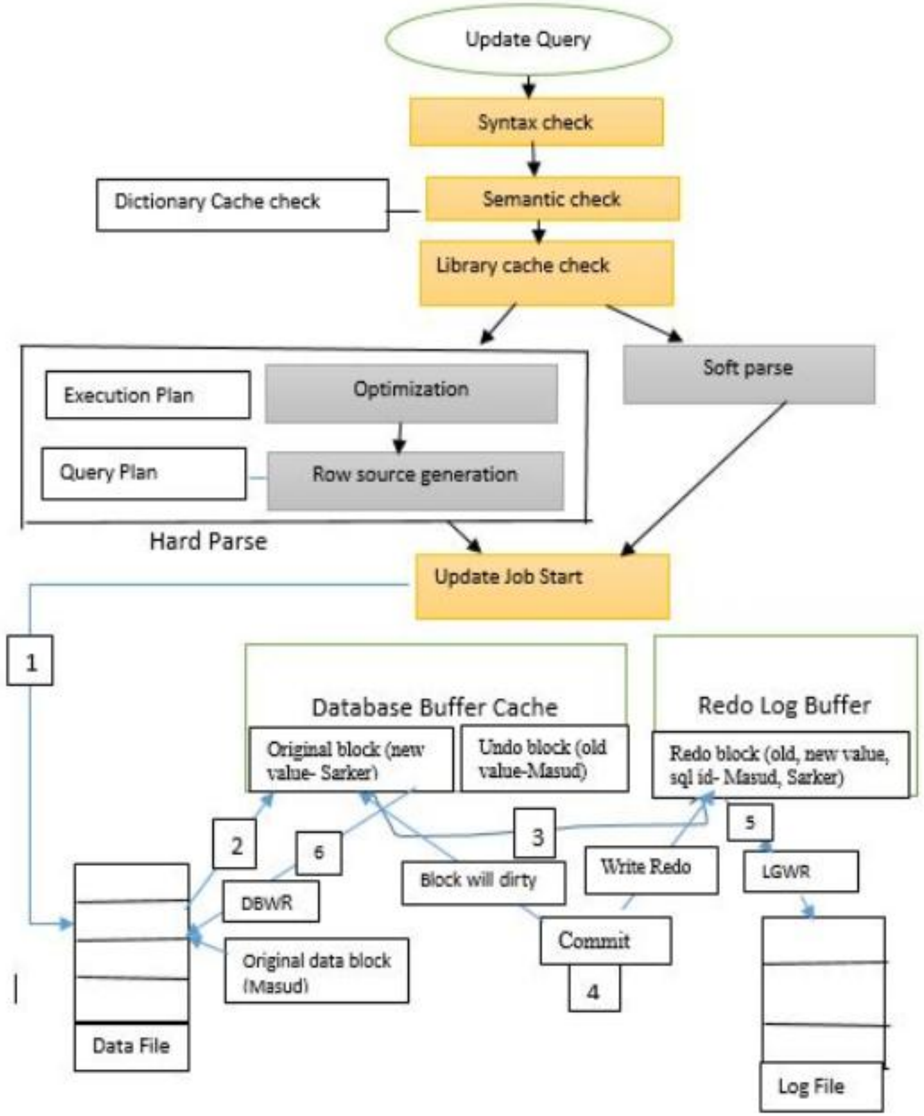
Archiver Processes (ARC*n*)

The **archiver processes (ARCn)** copy online redo log files to offline storage after a redo log switch occurs. These processes can also collect transaction redo data and transmit it to standby database destinations. ARC*n* processes exist *only* when the database is in ARCHIVELOG mode and automatic archiving is enabled.

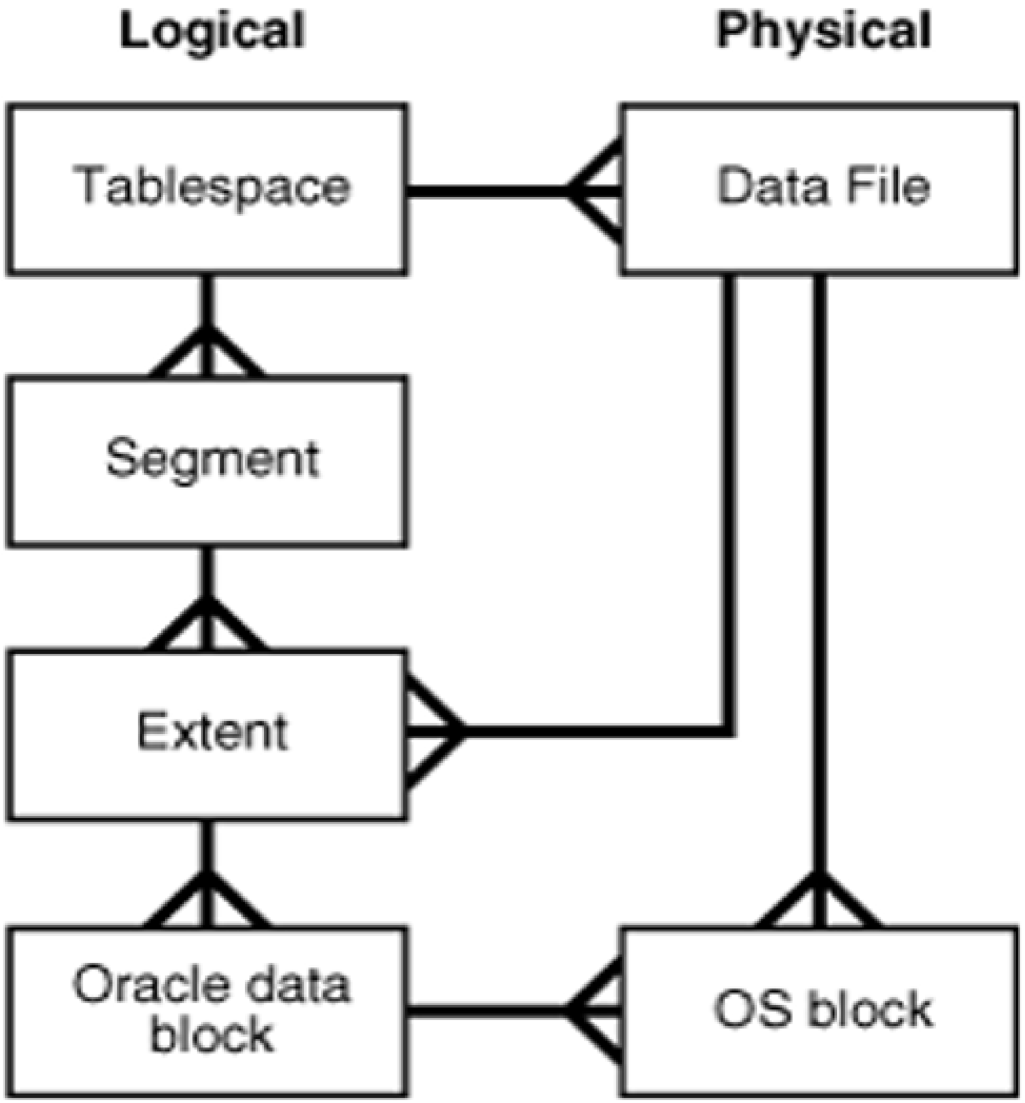
How select statement works?



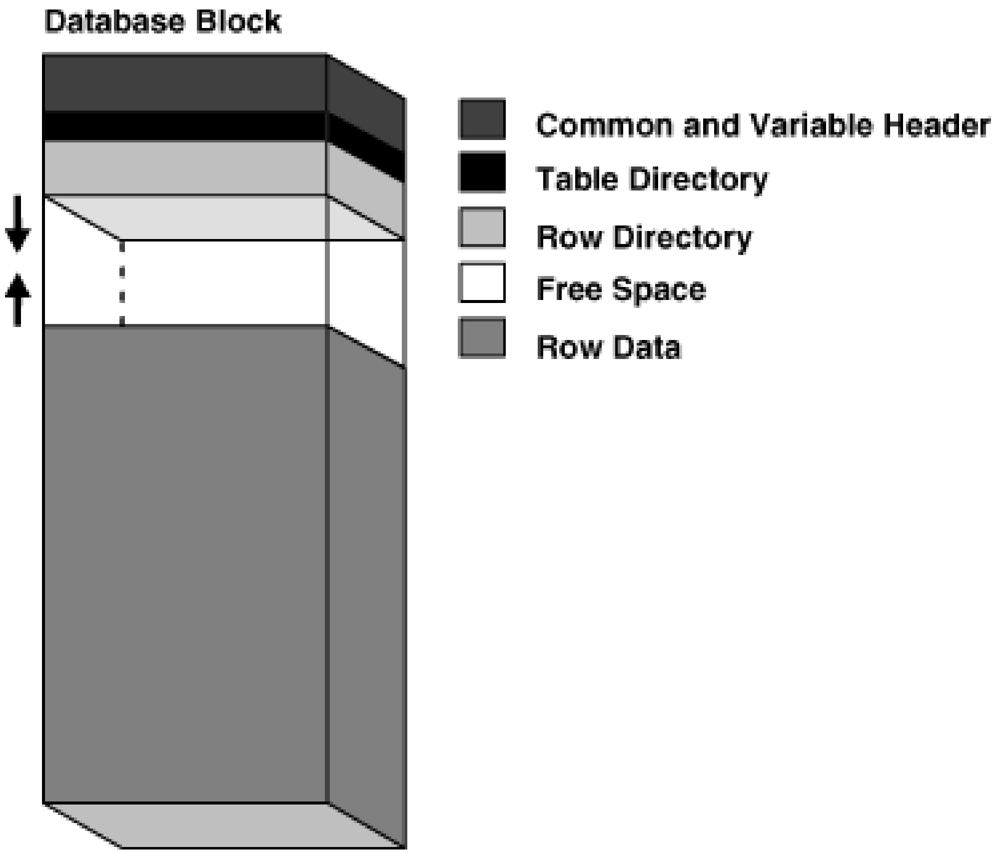
How update statement works?



BEST


Logical
Physical
Tablespace
Data File
Segment
Extent
Oracle data block
OS block

Data block structure



Data blocks :

- It is the smallest logical unit of data storage in Oracle Database.
- One logical data block corresponds to a specific number of bytes of physical disk space, for example, 2 KB.
- Data blocks are the smallest units of storage that Oracle Database can use or allocate.

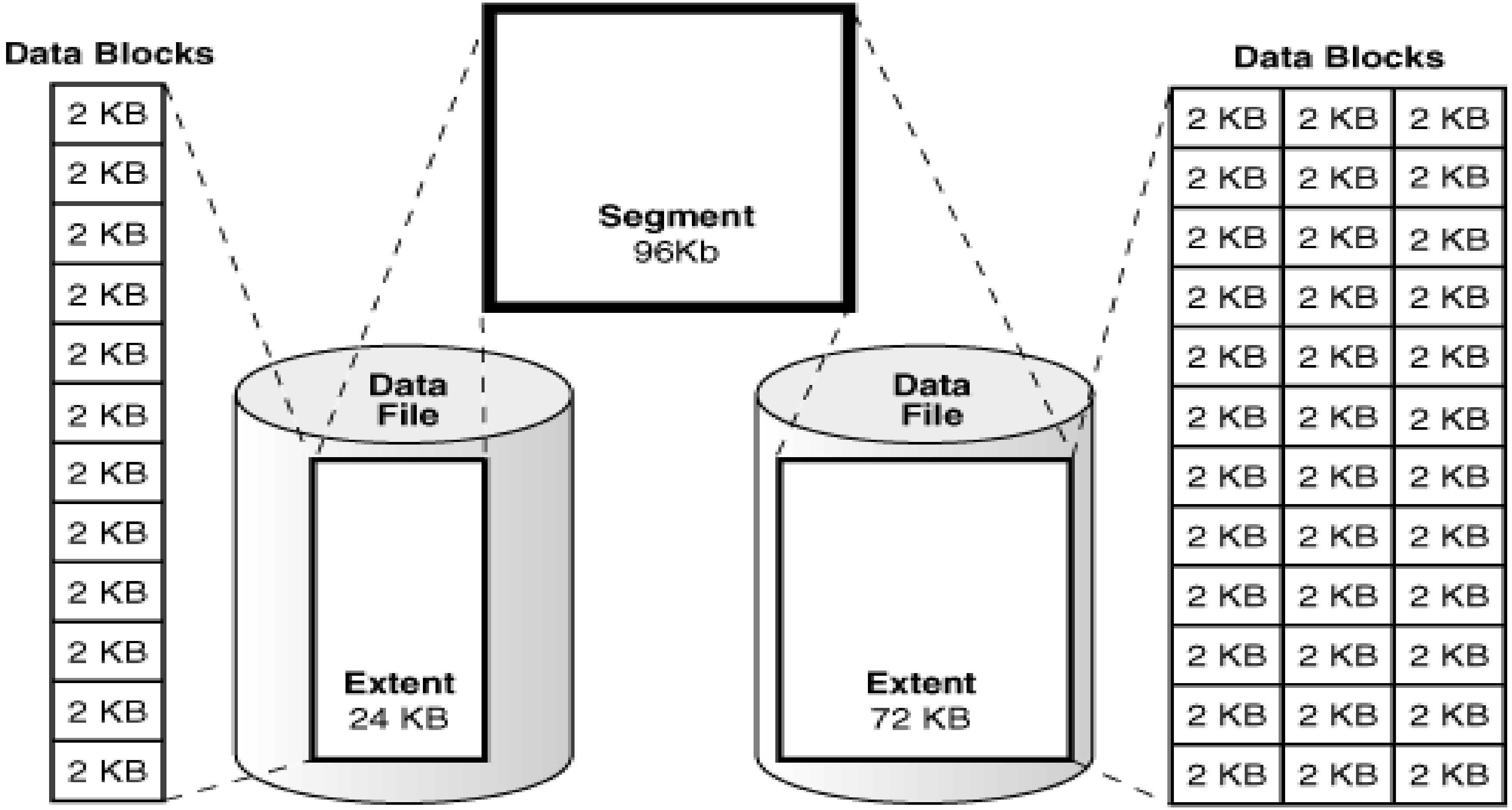
BEST

Extent
It is a set of logically contiguous data blocks allocated for storing a specific type of information
In the preceding graphic, the 24 KB extent has 12 data blocks, while the 72 KB extent has 36 data blocks.

Segment
It is a set of extents allocated for a specific database object, such as a table.
For example, the data for the employees table is stored in its own data segment, whereas each index for employees is stored in its own index segment. Every database object that consumes storage consists of a single segment.

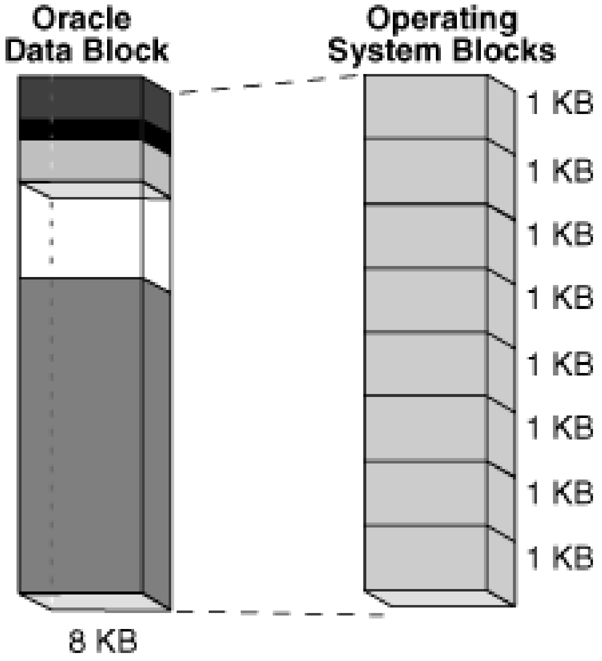
Tablespace
It is a database storage unit that contains one or more segments.
Each segment belongs to one and only one tablespace. Thus, all extents for a segment are stored in the same tablespace. Within a tablespace, a segment can include extents from multiple data files, as shown in the preceding graphic. For example, one extent for a segment may be stored in users01.dbf, while another is stored in users02.dbf. A single extent can never span data files.

BEST



Data Blocks
2 KB
2 KB
2 KB
2 KB
2 KB
2 KB
2 KB
2 KB
2 KB
2 KB
2 KB
2 KB
Segment
96Kb
Data File
Extent
24 KB
Data File
Extent
72 KB
Data Blocks
2 KB 2 KB 2 KB
2 KB 2 KB 2 KB
2 KB 2 KB 2 KB
2 KB 2 KB 2 KB
2 KB 2 KB 2 KB
2 KB 2 KB 2 KB
2 KB 2 KB 2 KB
2 KB 2 KB 2 KB
2 KB 2 KB 2 KB
2 KB 2 KB 2 KB
2 KB 2 KB 2 KB
2 KB 2 KB 2 KB

Physical storage structure



Operating System Blocks

- At the physical level, database data is stored in disk files made up of operating system blocks.
- An operating system block is the minimum unit of data that the operating system can read or write. In contrast, an Oracle block is a logical storage structure whose size and structure are not known to the operating system.
- The below shows that operating system blocks may differ in size from data blocks. The database requests data in multiples of data blocks, not operating system blocks.

Physical storage structure

Data files and temp files

A data file is a physical file in persistent storage that was created by Oracle Database and contains data structures such as tables and indexes.
A temp file is a data file that belongs to a temporary tablespace. The database writes data to these files in an Oracle proprietary format that cannot be read by other programs.

Control files

A control file is a root file that tracks the physical components of the CDB. PDBs do not have their own separate control files.

Online redo log files

The online redo log is a set of files containing records of changes made to data within the database.