Oracle Database Architecture: End-to-End Flow (Physical & Logical)

https://digitalksystems.com
https://www.youtube.com/channel/UCCGTnI9vvF_ETMhGUXGdFWw

Overview

Oracle Database architecture consists of two main layers:

- **Logical Architecture** : How data is logically stored and organized.
- **Physical Architecture** : How data is physically stored on disk.

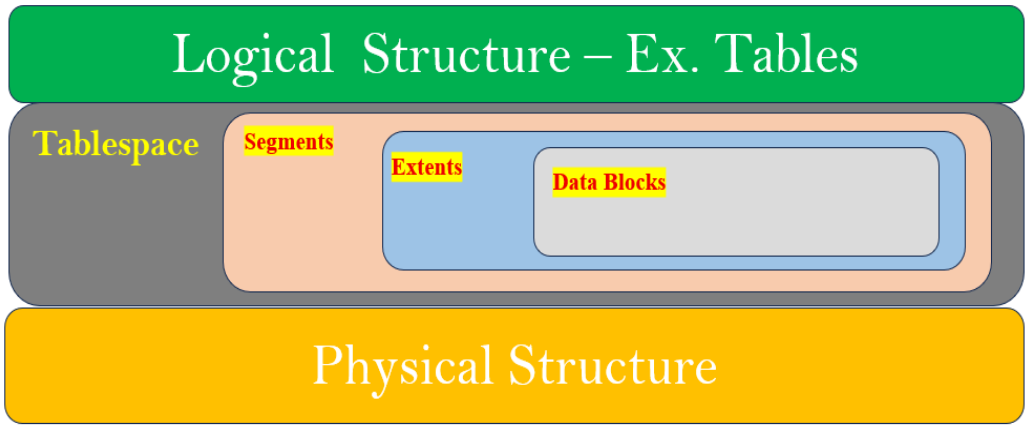
*On top of these, Oracle has an **Instance**, which is the combination of **memory structures** and **background processes** that manage database operations such as insert, update, and delete.*

I will explain the end-to-end flow considering these key components and how they interact during DML operations.

Logical Architecture

Components:

- **Tablespaces** : Logical storage units that group related data files.
- **Segments** : Each database object (like a table or index) has one or more segments.
- **Extents** : Contiguous blocks of storage allocated to a segment.
- **Blocks** : The smallest unit of I/O in Oracle; represents the actual data storage.



Hierarchy:

Database → Tablespace → Segment → Extent → Block

Example:

- A table resides in a segment.
- A segment resides in a tablespace.
- A tablespace is composed of datafiles.

Physical Architecture

Main Files in Oracle:

- **Datafiles** : Store actual table and index data (rows, blocks, etc.).
- **Redo Log Files** : Record all changes made to the database (for recovery).
- **Control Files** : Store metadata about the database (structure, file names, log history).
- **Parameter Files (PFILE/SPFILE)** : Store initialization parameters for instance startup.
- **Archived Redo Logs** : Archived copies of redo logs for backup/recovery.
- **Password File** : Used for authentication of privileged users.

Oracle Instance Architecture

An **instance** consists of:

- **Memory structures** (SGA + PGA)
- **Background processes** (DBWR, LGWR, SMON, PMON, CKPT, etc.)

A. Memory Structures

1. System Global Area (SGA) – Shared memory used by all sessions

Component	Description
Database Buffer Cache	Stores copies of data blocks read from datafiles. DML operations modify blocks here first.
Shared Pool	Stores parsed SQL statements and data dictionary information.
Redo Log Buffer	Stores redo entries describing changes made to data blocks.
Large Pool	Used for large memory allocations like RMAN backup, parallel execution.
Java Pool	For Java code execution within the database.
Streams Pool	Used for Oracle Streams replication.

2. Program Global Area (PGA) – Private memory for each user session

Component	Description
Sort Area	Used for ORDER BY, GROUP BY operations.
Session Memory	Holds session variables and cursor states.

B. Background Processes

Process	Description
DBWR (Database Writer)	Writes dirty buffers (modified data blocks) from Buffer Cache to Datafiles.
LGWR (Log Writer)	Writes redo entries from Redo Log Buffer to Redo Log Files.
CKPT (Checkpoint)	Signals DBWR to write all modified blocks to disk; updates control files and datafile headers.
SMON (System Monitor)	Performs recovery after instance failure.
PMON (Process Monitor)	Cleans up failed user processes and releases resources.
ARCn (Archiver)	Copies full redo logs to archive destinations after they are filled.
MMON/MMNL	Monitor and manage system statistics and metrics.

End-to-End Flow: Insert, Update, Delete Operations

Let's take an example to understand what happens **step-by-step**.

Example:

- INSERT INTO employees VALUES (101, 'John', 'HR');
- UPDATE employees SET dept = 'Finance' WHERE emp_id = 101;
- DELETE FROM employees WHERE emp_id = 101;

I'll break down what happens internally:

Step 1: User Connection & SQL Parsing

- The user connects to the **Oracle Instance**.
- The SQL is sent to the **Server Process**.
- **Shared Pool**:
  - SQL is parsed.
  - Execution plan is created (if not cached).
  - Data dictionary info is accessed.

Step 2: Execution Begins (Buffer Cache & PGA)

- The server process checks whether the required data block is in the **Database Buffer Cache**.
- If **not found**, it reads the block from the **datafile** into the **Buffer Cache**.

Step 3: Performing DML (Insert / Update / Delete)

INSERT:

- A new row is inserted into the block in the **Buffer Cache**.
- A corresponding **redo record** (describing this change) is generated and stored in the **Redo Log Buffer**.
- The block is now marked as **dirty** (modified but not yet written to disk).

UPDATE:

- The old data block is read into the Buffer Cache (if not already there).
- Oracle records the old value in the **Undo Tablespace** (for rollback/recovery).
- The new value replaces the old one in the Buffer Cache.
- A redo entry is generated in the Redo Log Buffer.

DELETE:

- Oracle marks the row as deleted in the Buffer Cache.
- Before deleting, the old image of the data is written to the **Undo Tablespace**.
- Redo entries are created for both the delete and the undo changes.

Step 4: Redo & Undo Recording

Action	Where Stored	Purpose
Undo	Undo Tablespace	Allows rollback, read consistency, recovery.
Redo	Redo Log Buffer → Redo Log Files	Ensures durability (replay after crash).

Step 5: Commit

When COMMIT is issued:

- **LGWR** writes all redo entries from the **Redo Log Buffer** to the **Online Redo Log Files**.
- Once LGWR confirms successful write, the transaction is considered **committed**.
- The **Undo** data is marked as expired (no longer needed for rollback).
- **At this point:**
  - Changes are still in memory (Buffer Cache), not yet in datafiles.
  - But the transaction is safe because redo is on disk.

Step 6: Checkpoint & DBWR Activity

- Periodically or at checkpoint:
  - **DBWR** writes dirty blocks from **Buffer Cache** to **Datafiles**.
  - **CKPT** updates headers of Datafiles and Control Files with latest checkpoint info.

This ensures database consistency between memory and disk.

Step 7: Instance Recovery (If Failure Occurs)

- If the instance crashes before DBWR writes dirty blocks:
  - On restart, **SMON** uses **Redo Log Files** to reapply committed changes (ROLL FORWARD).
  - Uses **Undo Tablespace** to rollback uncommitted transactions (ROLL BACK).

This guarantees **Atomicity** and **Durability** (ACID properties).

End-to-End Flow Summary

Step	Action	Component Involved
1	SQL parsed and optimized	Shared Pool
2	Data block fetched	Buffer Cache / Datafile
3	DML changes applied	Buffer Cache + Undo Tablespace
4	Redo generated	Redo Log Buffer
5	Commit	LGWR writes redo to Redo Logs
6	Checkpoint	DBWR + CKPT write blocks and metadata
7	Recovery (if needed)	SMON + Undo + Redo

Conclusion

- **SGA** handles caching, parsing, and buffering for performance.
- **PGA** handles per-session memory needs.
- **Redo Logs** guarantee recoverability.
- **Undo Tablespace** ensures consistency.
- **DBWR, LGWR, CKPT, SMON, PMON** coordinate writing and recovery.

Together, they make Oracle ACID-compliant and highly reliable for transactional systems.