

# DBA Tasks

- User management:  
**CREATE USER, ALTER USER, GRANT/REVOKE privileges or roles**
- Storage management:  
**CREATE/DROP TABLESPACE, add datafiles**
- Instance control:  
**STARTUP, SHUTDOWN, ALTER SYSTEM**
- Backup/recovery:  
**RMAN configuration, data pump exports/imports**



# DBA Privileges

- **SYSDBA:** Full control (create/drop/startup/shutdown, backup/recovery)
- **SYSOPER:** Operational tasks (startup/shutdown, backup/recovery, but no data dictionary access)
- **DBA role:** Grants common administrative privileges (create user, tablespace, grant roles)
- **Performance tuning:** Gathering statistics, monitoring wait events



# Managing Database Memory

## Monitoring & Advisors:

- Query views like **V\$MEMORY\_TARGET\_ADVICE**, **V\$PGA\_TARGET\_ADVICE**, and **V\$SGA\_DYNAMIC\_COMPONENTS** to see how memory changes affect performance.
- Use **Automatic Memory Advisors** in Enterprise Manager or SQL Developer for tuning recommendations.

## Dynamic Changes:

- With ASMM or AMM, you can **ALTER SYSTEM SET ... SCOPE=BOTH** to adjust targets without a restart (within **MEMORY\_MAX\_TARGET**).



# Managing Database Memory

## Best Practice:

- Start with AMM for simplicity in development/test.
- In production, consider ASMM or manual sizing after baseline performance analysis.



# Creating a Database

**As a DBA, you will need to manage the database and its components. As an organisation grows, and business requirements change, there will always be a need to create new databases.**



# **Why is memory and process management important in a multi-user database environment?**

**In a multi-user database environment, multiple users perform simultaneous operations such as queries, updates, and transactions.**

**Effective memory and process management is essential for the following reasons:**

- ◆ **1. Performance Optimization**

**Proper memory allocation ensures quick data access and execution of queries. If memory is insufficient or misallocated, processes can slow down, affecting user experience.**



# Why is memory and process management important in a multi-user database environment?

- ◆ **2. Resource Sharing**

**Fair distribution of system resources avoids bottlenecks and ensures all users get adequate performance.**

- ◆ **3. System Stability**

**If server processes are not managed correctly, they may consume excessive memory, causing database crashes or degraded performance.**

- ◆ **4. Scalability**

**Proper management allows the database to handle a growing number of users without reconfiguration.**



# Memory areas used by an Oracle database

- ◆ **SGA (System Global Area)**

The SGA is a shared memory region used by all Oracle server processes. It holds data and control information for the entire instance.

- ◆ **PGA (Program Global Area)**

The PGA is a non-shared (private) memory region allocated per user process. It stores data and control info for a single user session.



# Automatic Memory Management in Oracle

Oracle provides Automatic Memory Management (AMM) to simplify the task of managing memory. It uses two key initialisation parameters:

- ◆ **MEMORY\_MAX\_TARGET**

This sets the maximum amount of memory that Oracle can use for both the SGA and PGA combined.

Acts as the upper boundary and cannot be exceeded unless changed manually.

# Automatic Memory Management in Oracle

- ◆ **MEMORY\_TARGET**

This sets the current total memory allocation for Oracle to use across the SGA and PGA.

Oracle internally adjusts memory between the SGA and PGA dynamically based on workload.

 Example:

```
ALTER SYSTEM SET MEMORY_MAX_TARGET = 1G SCOPE = SPFILE;  
ALTER SYSTEM SET MEMORY_TARGET = 750M SCOPE = SPFILE;
```



# User (Client) Process vs. Oracle (Server) Process

## ◆ User (Client) Process

This is the program initiated by a user's application.

- It sends SQL requests to the database and receives results.
- It runs on the client machine or application middleware.

# User (Client) Process vs. Oracle (Server) Process

## ◆ Oracle (Server) Process

This process runs on the database server.

It:

- Parses and executes SQL sent from the user process
- Retrieves and manipulates data
- Returns results to the user process

 Every time a user interacts with the database, both a user process and a server process are involved.

# Dedicated vs. Shared Server Processes

## Dedicated Server Process

- A one-to-one relationship:
  - One server process per user session
- The server process remains active for the entire duration of the user's connection.

### Advantages:

- Predictable performance for each user.
- Easier to debug or isolate issues per session.

### Disadvantages:

- High memory usage: Each user consumes their own server process.
- Not scalable for systems with thousands of users.

# Dedicated vs. Shared Server Processes

## Shared Server Process

- A many-to-few relationship:
  - Multiple user sessions share a small pool of server processes
- Requests are queued in a dispatcher and processed as resources free up.

### Advantages:

- Efficient memory usage.
- Better scalability – can support many users with fewer server processes.

### Disadvantages:

- Slightly more complex setup.
- Users may experience slight delays under heavy load due to queuing.

# Dedicated vs. Shared Server Processes

## Performance Comparison Under High Load:

Scenario	Dedicated Server	Shared Server
<b>Heavy number of users</b>	High memory/resource strain	Efficient memory use
<b>CPU usage</b>	More processes = more CPU	Controlled CPU via shared pool
<b>Latency for simple queries</b>	Low	Slight queuing delay possible
<b>System memory availability</b>	May become a bottleneck	Managed more conservatively

# Table Spaces

## What is a Tablespace and Why It Matters?

- ◆ Definition

A **tablespace** is a logical storage unit in an Oracle database.

It groups related physical data files and defines where and how database objects (like tables and indexes) are stored on disk.

Think of a **tablespace** as a container for data files, and inside those containers live your tables, indexes, etc.

# Table Spaces

Why would we use a table space?

Improves Performance

- By spreading data across multiple disks/tablespaces, Oracle can balance I/O and speed up access.

Enables Space Management

- You can allocate and control storage for different applications, users, or data types.

Supports Backup & Recovery

- Tablespaces can be backed up or restored individually.

Simplifies Organization

- Separating data into logical groups (e.g., by department or function) helps in maintenance.

# Table Space Types:

Tablespace Type	Purpose	Example Use
Permanent	Stores <b>user-created</b> objects such as tables, indexes, and views.	Core application data
Temporary	Stores <b>intermediate</b> results (e.g., for sorting large queries).	ORDER BY, GROUP BY operations
Undo	Stores <b>undo data</b> for transactions to support rollback and read consistency.	Cancel/rollback transactions

# Table Space Types:

Tablespace Type	Purpose	Example Use
Permanent	Stores <b>user-created</b> objects such as tables, indexes, and views.	Core application data
Temporary	Stores <b>intermediate</b> results (e.g., for sorting large queries).	ORDER BY, GROUP BY operations
Undo	Stores <b>undo data</b> for transactions to support rollback and read consistency.	Cancel/rollback transactions