Session 1
Introduction to Database management Systems

- Jayendra Khatod

Objectives

- **Introduction to Database**
- **Introduction to Database Management System (DBMS)**
- **Understanding Types of DBMS**
- **Internal Architecture of Oracle Relational DBMS**

What is a database?

A database is a storage space for content / information (data).

But what is data? And where is it stored?

Data is factual information about objects and concepts, such as: measurements, statistics etc..

You can find it in:

- **filing cabinets, spreadsheets, folders**
- **Ledgers, lists, piles of papers on your desk**
- **colleagues' memories etc..**

What does “managing information” mean?

- **Making information *work* for us**
- **Making information useful**
- **Avoiding "accidental disorganisation”**
- **Making information easily accessible and integrated with the rest of our work**

Managing as *re-organising*

We often need to access and re-sort data for various uses. These may include:

- **Creating mailing lists**
- **Writing management reports**
- **Generating lists of selected news stories**
- **Identifying various client needs**

Can you add to the list?

Managing as *re-processing*

The processing power of a database allows it to:

- **Sort**
- **Match**
- **Link**
- **Aggregate**
- **Skip fields**
- **Calculate**
- **Arrange**

Databases everywhere!

Because of the versatility of databases, we find them powering all sorts of projects:

- **A web site that is capturing registered users**
- **A medical record system for a health care facility**
- **Your personal address book in your e-mail client**
- **A system that issues airline reservations**
- **Google, Facebook, eBay, Yahoomail…everything around us requires a Database..**
- **It is the brain/memory of all the systems..**

What Is a DBMS?

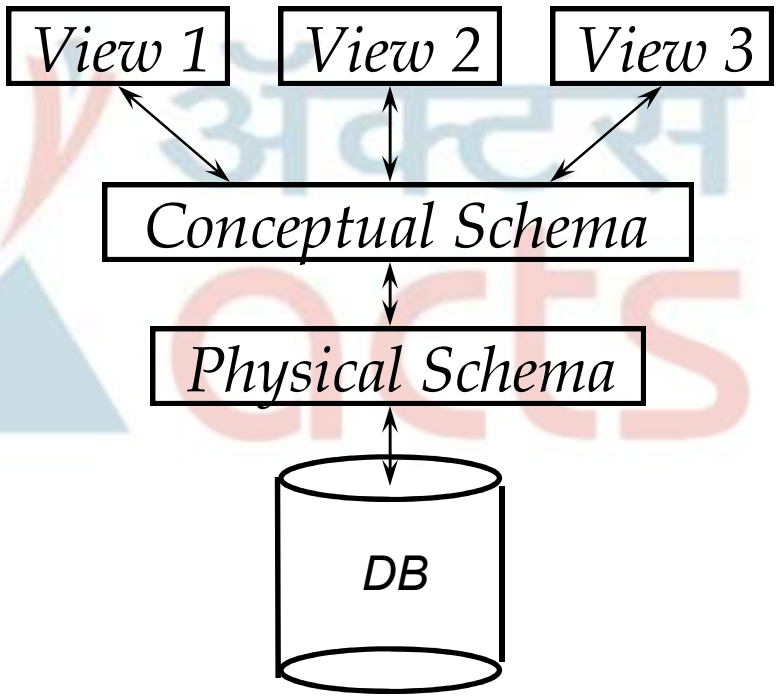
- **A *very large, integrated* and *organized* collection of data.**
- **Models real-world enterprise.**
  - **Entities (e.g., students, courses)**
  - **Relationships (e.g., Madonna is taking ITCS6160)**
- **A *Database Management System (DBMS)* is a software package designed to maintain and utilize databases.**

Why Use a DBMS?

- **Data independence and efficient access.**
- **Reduced application development time.**
- **Data integrity and security.**
- **Uniform data administration.**
- **Concurrent access, recovery from crashes.**

Levels of Abstraction in a DBMS

- **Many *views*, single *conceptual (logical) schema* and *physical schema*.**
  - **Views describe how users see the data.**
  - **Conceptual schema defines logical structure**
  - **Physical schema describes the files and indexes used.**



Example: University Database

- **Conceptual schema:**
  - ***Students(sid: string, name: string, login: string, age: integer, gpa:real)***
  - ***Courses(cid: string, cname:string, credits:integer)***
  - ***Enrolled(sid:string, cid:string, grade:string)***
- **Physical schema:**
  - **Relations stored as unordered files.**
  - **Index on first column of Students.**
- **External Schema (View):**
  - ***Course_info(cid:string,enrollment:integer)***

Data Independence

- **Data independence is the type of data transparency that matters for a centralized DBMS. It refers to the immunity of user applications to make changes in the definition and organization of data.**

- **Data independence has two types:**
  1. **Physical Independence and**
  2. **Logical Independence**

✉ ***It is one of the most important benefits of using a DBMS!***

Physical Independence

- **The logical schema stays unchanged even though the storage space or type of some data is changed for reasons of optimization or reorganization.**
  [The ability to change the physical schema without changing the logical schema is called as Physical Data Independence.]

- **For example, a change to the internal schema, such as using different file organization, storage devices, or indexing strategy, should be possible without having to change the conceptual or external schemas**

Logical Independence

- **The external schema may stay unchanged for most changes of the logical schema. This is especially desirable as the application software does not need to be modified or newly translated.**

- **[The ability to change the logical schema without changing the external schema or application programs is called as Logical Data Independence.]**

- **For example, the addition or removal of new entities, attributes, or relationships to the conceptual schema should be possible without having to rewrite existing application programs.**

Concurrency Control

- **Concurrent execution of user programs is essential for good DBMS performance.**
  - **Because disk accesses are frequent, and relatively slow, it is important to keep the cpu humming by working on several user programs concurrently.**
- **Interleaving actions of different user programs can lead to inconsistency: e.g., check is cleared while account balance is being computed.**
- **DBMS ensures such problems don't arise: users can pretend they are using a single-user system**

Transaction: An Execution of a DB Program

- **Key concept is transaction, which is an atomic sequence of database actions (reads/writes).**
- **Each transaction, executed completely, must leave the DB in a consistent state.**
- **A good DBMS always ensures the Automicity, Consistency, Isolation and Durability of the Transactions.**
- **These are also referred to as the ACID properties of a Transaction.**

ACID Properties

- **Atomicity requires that database modifications must follow an "all or nothing" rule. Each transaction is said to be atomic. If one part of the transaction fails, the entire transaction fails and the database state is left unchanged.**

- **This guarantees that 'an incomplete transaction' cannot exist**

ACID Properties

- **Consistency property ensures that any transaction the database performs will take it from one consistent state to another.**

- **Consistency states that only consistent (valid according to all the rules defined) data will be written to the database.**

ACID Properties

- **Isolation refers to the requirement that no transaction should be able to interfere with another transaction at all.**

- **In other words, it should not be possible that two transactions affect the same rows concurrently, as the outcome would be unpredicted and the system thus unreliable.**

ACID Properties

- **Durability means that once a transaction has been committed, it will remain so.**

- **In other words, every committed transaction is protected against power loss/crash/errors and cannot be lost by the system and can thus be guaranteed to be completed.**

Data Models

- **A data model is a collection of concepts for describing**
  - **data**
  - **data relationships**
  - **data semantics**
  - **data constraints**
- **A schema is a description of a particular collection of data, using a given data model.**

Types of DBMS based on Data models

- **Based on various Data Models there are different types of the Database management systems as follows:**
  - **OODBMS (Object Oriented Database Management System)**
  - **NDBMS (Networked Database Management System)**
  - **HDBMS (Hierarchical Database Management System)**
  - **RDBMS (Relational Database Management System)**

Out of these DBMS systems the RDBMS is most widely used system.

Object Oriented Database Management System

- **Based on collection of objects.**
- **Objects contains values stored in instance variables.**
- **Object can also contain bodies of code called methods.**
- **Objects that contain same type of value and same methods are grouped together into classes.**
- **A class may be viewed as type definition for objects.**
- **This is similar to ADT of programming languages.**
- **The only way in which one object can access the data of another object is by invoking a method of another object.**

Networked Database Management System

- **Data is represented by collection of records.**
- **Relationships among data are represented by links (pointers).**
- **The records in the database are organized as arbitrary graphs.**

Hierarchical Database Management System

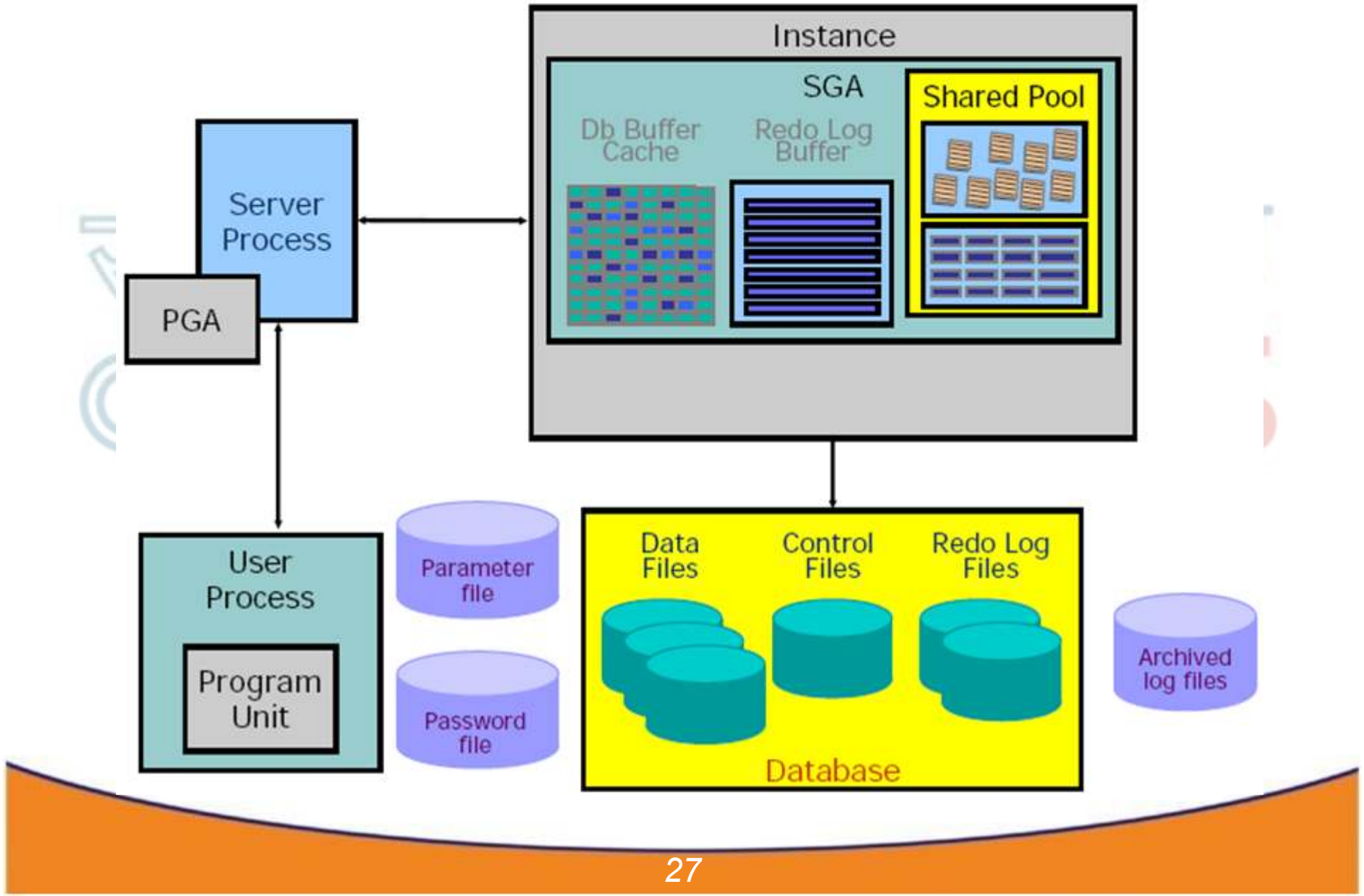
- **It is similar to the network model in the sense that data is represented by collection of records and relationships among data are represented by links respectively.**
- **It differs from the network model in that the records are organized as collections of trees rather than arbitrary graphs.**

Relational Database Management System

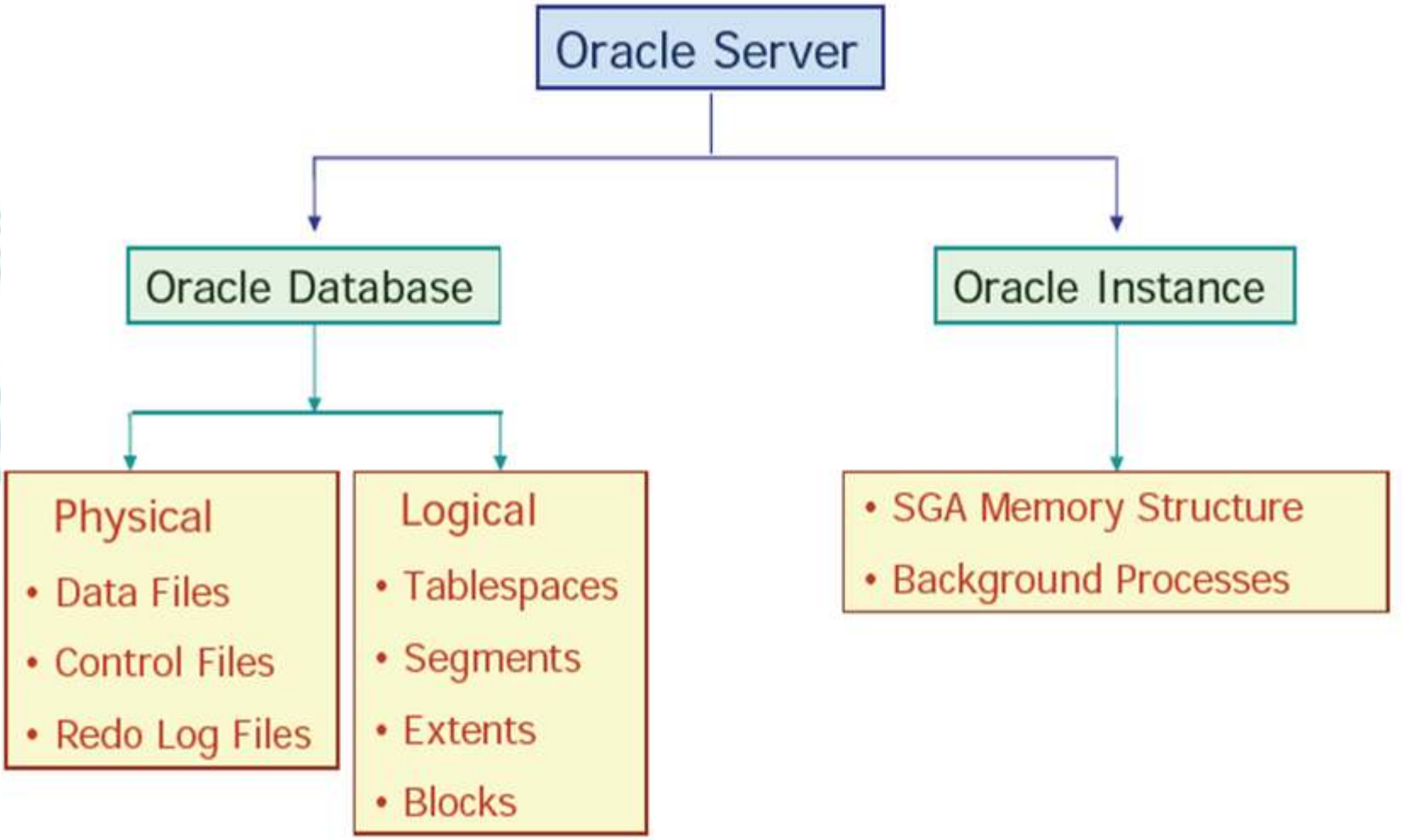
- **In a Relational Database Management System, Data is represented as Entities and their Relationships**
- **Entities (For ex. Car)**
  - **Attributes (For ex. Model, color etc..)**
- **Relationships**
  - **One-to-One**
  - **One-to-Many**
  - **Many-to-One**
  - **Many-to-many**

Oracle Database is a RDBMS.

Internal Architecture of Oracle RDBMS

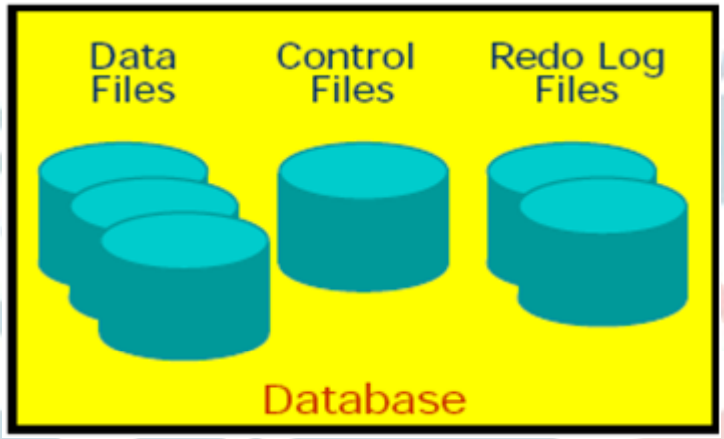


Oracle Server Components



Oracle Database

- **A collection of data that is treated as a unit, which stores and retrieves related information.**



- **Three types of files:**

  Control files:

  - **Information required to maintain and verify integrity of the database.**
  - **A database needs at least one control file.**

Oracle Database (Continued..)

Data Files:

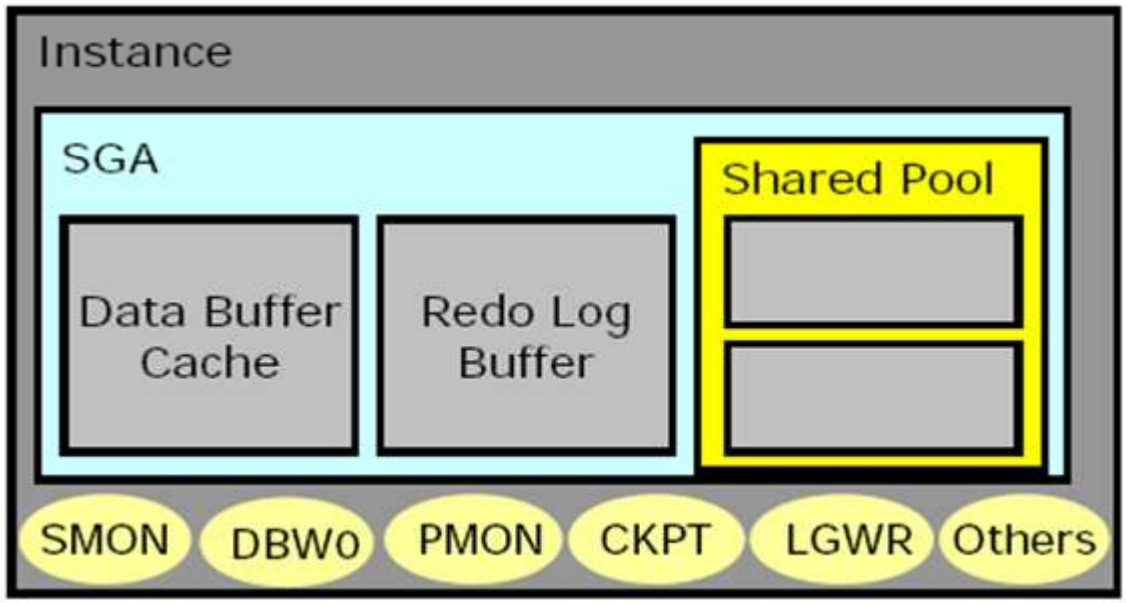
- **At least one data file for each tablespace.**
- **Can belong to only one tablespace.**
- **Contain the data in the database, including tables, indexes, temp segments etc..**

Redo Logs Files

- **Maintain the logs of the changes made to the database to enable recovery.**
- **At least two redo log files are required for each database.**

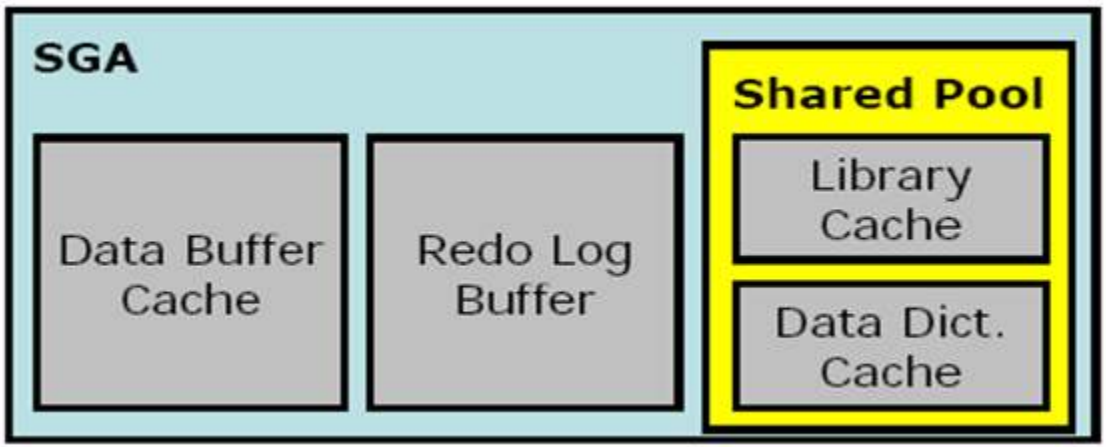
Oracle Instance

- **A means to access Oracle database**
- **Can open and use one database at a time**
- **Consists of SGA and background processes**



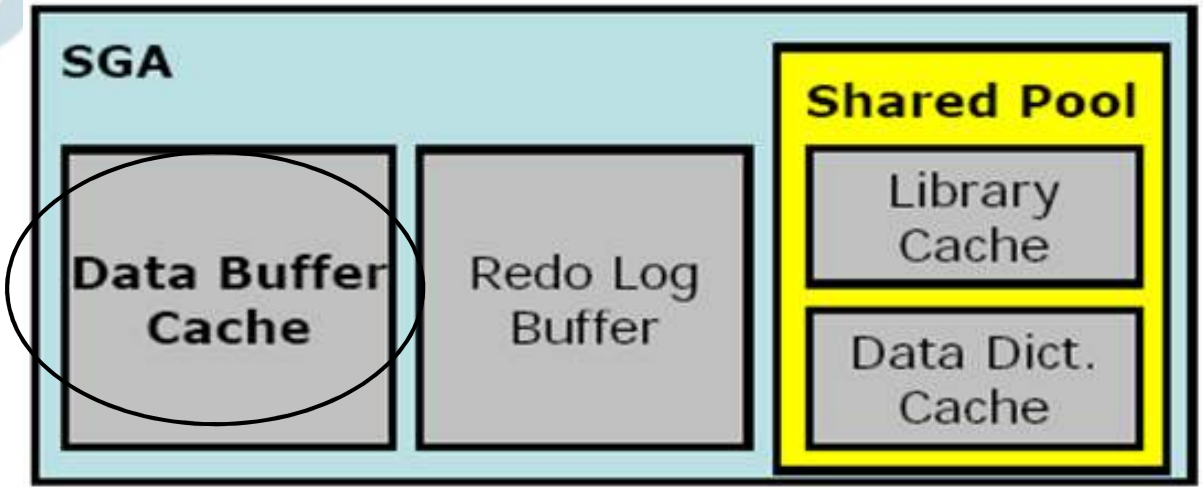
System Global area

- **Stores database information**
- **Consists of several memory structures:**
  - **Database Buffer Cache**
  - **Redo Log Buffer**
  - **Shared Pool**



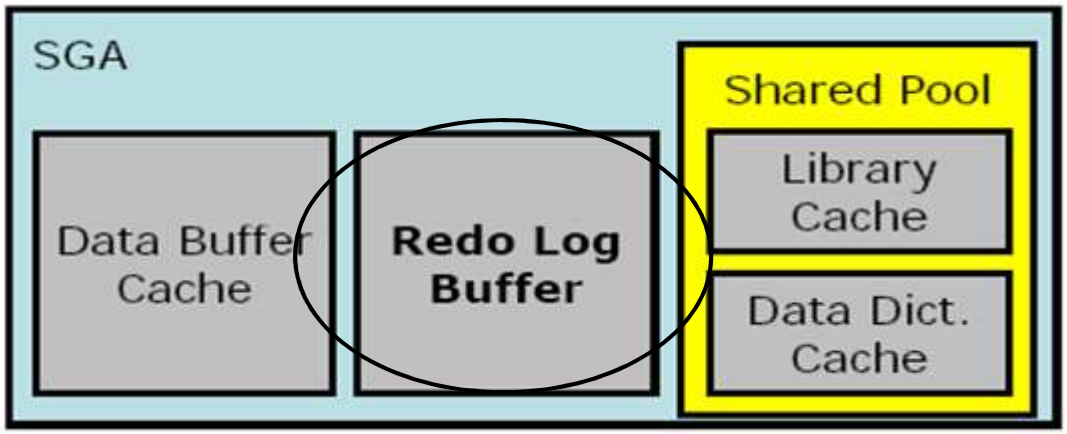
Database Buffer Cache

- **Stores the most recently used data**
- **The data is read from, and written to data files**
- **Size of each buffer in the buffer cache is equal to the size of an Oracle Block.**
- **Size of buffer is thus based on the DB_BLOCK_SIZE parameter in init.ora**



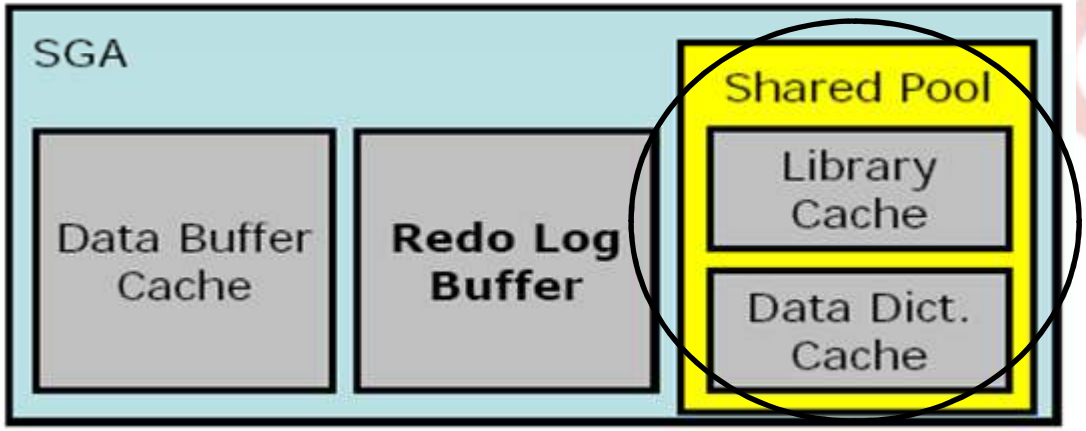
Redo log Buffer

- **Records changes made to a database**
- **The server process records changes in the redo log buffer**
- **It recoords the block that is changed, the location of the change and the new value**
- **The buffer is reused after it is filled, once all the old redo entries are recorded in the redo log files.**



Shared Pool

- **Stores the most recently executed SQL statements & used data from the data dictionary**
- **The shared pool is sized by SHARED_POOL_SIZE parameter of init.ora**



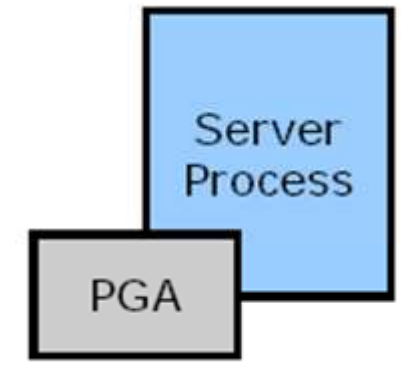
Shared Pool (Continued..)

- **Has two components:**
- **Library Cache: Stores information about the most recently used SQL statements.**
- **The Library Cache contains:**
  - **The text of SQL statement**
  - **The Parse Tree: complied version of a statement.**
  - **The Execution Plan.**
- **Data dictionary Cache:**

  A collection of the most recently used definitions in the database.

Program Global Area (PGA)

- **PGA is a memory area that contains:**
  - **Session information**
  - **Cursor information**
  - **SQL execution work areas**
  - **Sort area**
  - **Hash_join area**
  - **Bitmap merge area**
  - **Bitmap create area**



The PGA work area size influences the SQL performance

Summary

In this lesson, you should have learned :

- **What a Database is**
- **What a DBMS is**
- **What a Data Model is**
- **ACID Properties of a Transaction**
- **Types of a DBMS**
- **The internal Architecture of Oracle RDBMS**


Thank You !