Oracle System Architecture

Seema Sirpal
Delhi University Computer Centre

Oracle Database Structure

- A database is collection of data files plus programs that manipulate those data files.
- Two types of information are stored in an Oracle database
    - **User data, relevant to a particular application and**
    - **System data, that the database needs to manage itself**
- Components
    - **Database files – contain all database data**
    - **Control files – contain info to access and manipulate the data**
    - **Redo Logs – record all the transactions on the database**
    - **Tablespaces – logical divisions of a database**
    - **Segments – set of database blocks**
    - **Extents – allocation of contiguous space on disk exp in bytes.**

Oracle Database Structure

Database files

- An Oracle database consists of one or more database files.
- These files contain data that can be accessed with the help of SQL.
- A file can be associated with one and only one tablespace.
- The database when created initially, has only one physical file.
- The maximum number of files that can be opened at a time are 32 by default.
- One or more physical files form a logical unit called tablespace.

Oracle Database Structure

Control files

- A database's overall physical architecture is maintained by its control files.
- These record control information about all the files within the database.
- A control file is associated with a single database
- Control files are automatically modified by Oracle; a user cannot edit them.
- They are used to maintain internal consistency and guide recovery operations

Oracle Database Structure

Redo Log files

- A Redo Log contains all the transactions that have occurred against the database.
- No activity in the database can take place without being recorded in Redo Logs.
- The Redo Logs are necessary to protect against data loss.
- Redo Logs are also called transaction logs.
- Every database requires a minimum of two Redo Logs.
- These are used in recovery operations to restore lost or damaged files.

Oracle Database Structure

Tablespaces

- A database is divided into logical divisions called tablespaces
- A database may have one or more tablespaces
- Each logical tablespace corresponds to one or more physical database files.

•The typical tablespace present in an Oracle database are :

- **SYSTEM – stores all information needed to manage itself**
- **TEMP – stores all temporary tables**
- **TOOLS – stores database objects needed to support different tools**
- **USER – information about users is stored**
- **DATA & INDEX – holds the actual data and the indexes**
- **ROLLBACK – where all undo information is stored**

Oracle Database Structure

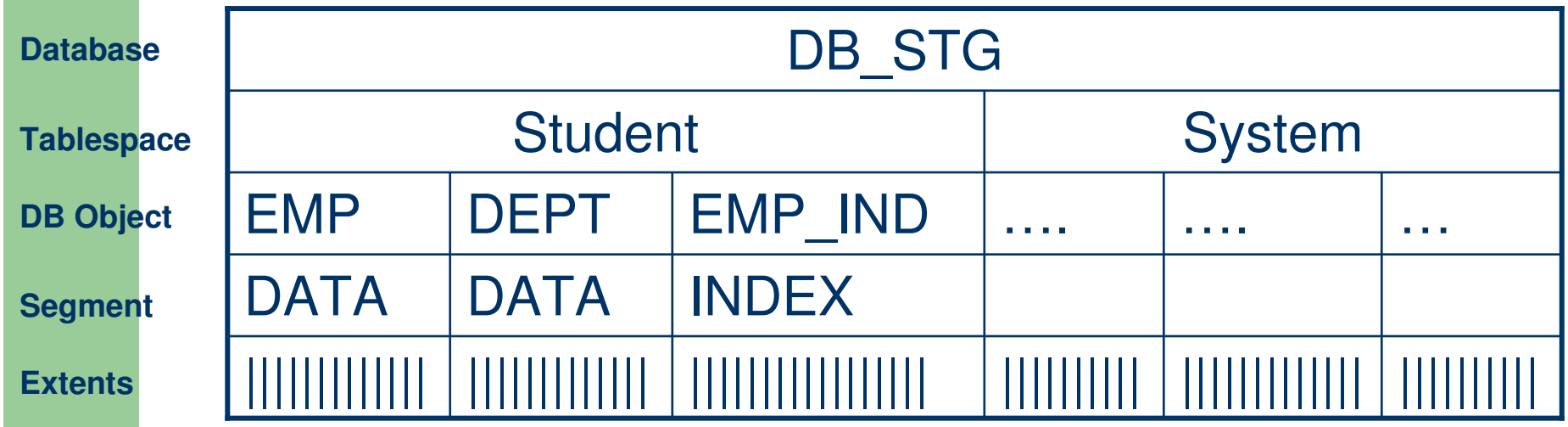
Segments & Extents

- All data in tablespace are stored in allocations of database space called segments.
- For each table there is a table segment. For indexes so-called index segments are allocated. The segment associated with a database object belongs to exactly one tablespace.
- An Oracle database requires upto five types of segments, viz:

  Data segments, Index segments, Rollback segments, Temporary segments, Bootstrap segments

- An extent is an allocation of contiguous database space expressed in bytes

Logical Database Structures



Storage Management and Processes

The ORACLE DBMS server is based on a so-called *Multi-Server Architecture*.

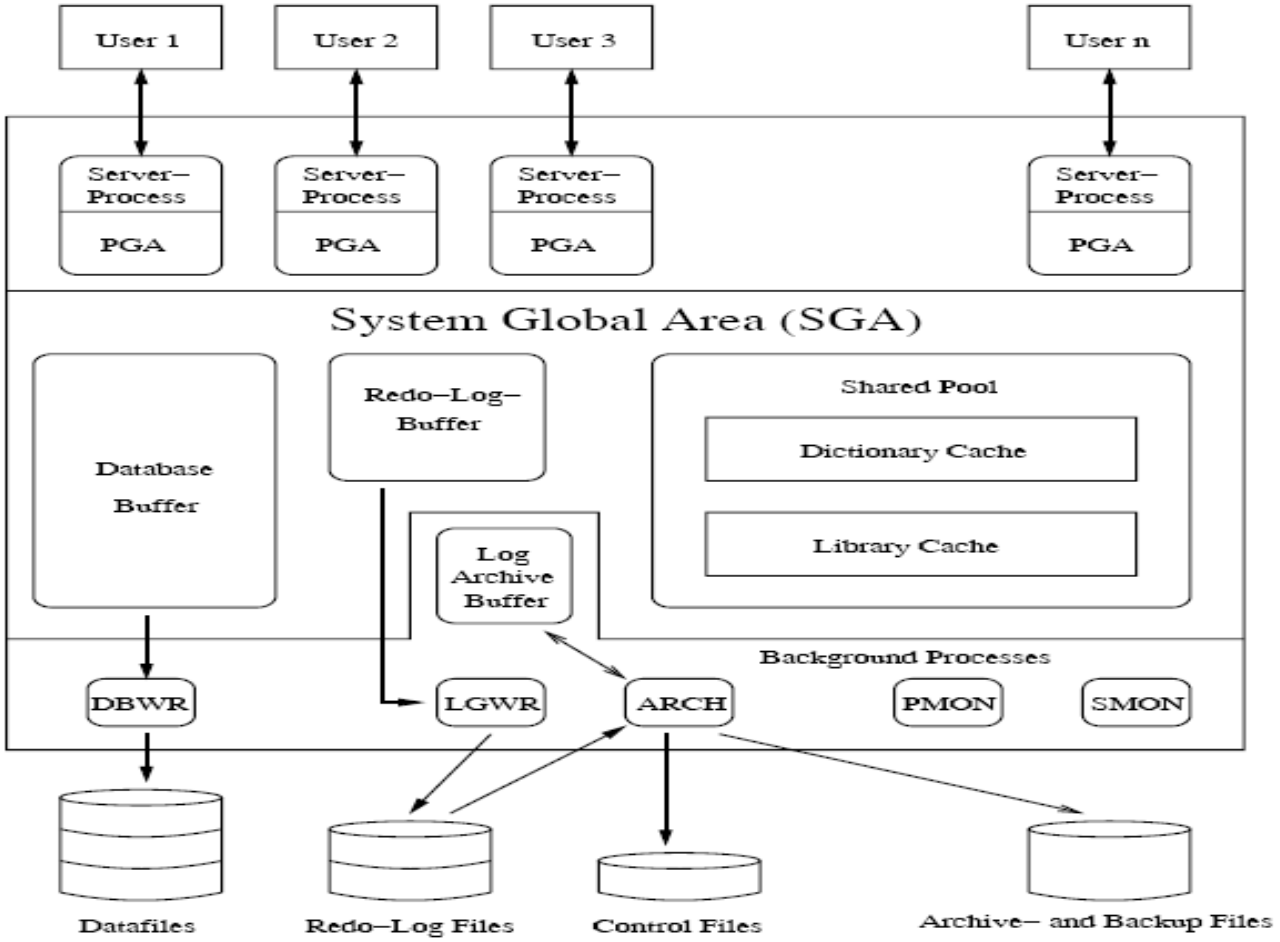


Figure 4: Oracle System Architecture

Storage Management and Processes

System Global Area (SGA)
The SGA consists of the *shared pool*, the *database buffer*, and the *redo-log buffer*. Furthermore, several background processes are started. The combination of SGA and processes is called *database instance*.

Program Global Area (PGA) is the area in the memory that is used by a single Oracle user process. It contains the user's context area (variables etc.), as well as process information. The memory in the PGA is not sharable.

For each database instance, there is a set of processes. These processes maintain and enforce the relationships between the database's physical structures and memory structures. Oracle processes are typically background processes that perform I/O operations at database run-time.

Storage Management and Processes

DBWR writes modified data blocks to the data files.

LGWR manages writing the contents of the redo-log-buffer to the redo-log files.

SMON When a database instance is started, the system monitor process performs instance recovery as needed. It cleans up the database from aborted transactions involved.

PMON The process monitor process cleans up behind failed user processes and it also cleans up the resources used by these processes.

USER The task of this process is to communicate with other processes started by application programs such as SQL*Plus. It is responsible for sending respective operations and requests to the SGA or PGA.

Oracle Enterprise Manager

Features

- Administer, diagnose, and tune multiple databases.
- Distribute software to multiple servers and clients.
- Schedule jobs on multiple nodes at varying time intervals.
- Monitor objects and events throughout the network.
- Administer Oracle Parallel Servers.
- Integrate participating Oracle or third-party tools.

DBA Tools

DBA Tools	Task
Instance Manager	Manages instance and sessions
Schema Manager	Manages schema objects
Security Manager	Controls security
Storage Manager	Manages storage
SQL Worksheet	Enters and executes DBA commands, SQL statements, and PL/SQL commands
Backup Manager	Manages database backup and recovery environment
Data Manager	Exports, imports, and loads data in & out of an Oracle database

Query Execution

- Assume a user (working with SQL*Plus) issues an update statement on the table TAB such that more than one tuple is affected by the update. The statement is passed to the server by the USER process. Then the server (or rather the query processor) checks whether this statement is already contained in the library cache such that the corresponding information (parse tree, execution plan) can be used. If the statement can not be found, it is parsed and after verifying the statement (user privileges, affected tables and columns) using data from the dictionary cache, a query execution plan is generated by the query optimizer. Together with the parse tree, this plan is stored in the library cache.
- For the objects affected by the statement (here the table TAB) it is checked, whether the corresponding data blocks already exist in the database buffer. If not, the USER process reads the data blocks into the database buffer. If there is not enough space in the buffer, the least recently used blocks of other objects are written back to the disk by the DBWR process.
- The modifications of the tuples affected by the update occurs in the database buffer. Before the data blocks are modified, the “before image” of the tuples is written to the rollback segments by the DBWR process.

Query Execution

- While the redo-log buffer is filled during the data block modifications, the LGWR process writes entries from the redo-log buffer to the redo-log files.
- After all tuples (or rather the corresponding data blocks) have been modified in the database buffer, the modifications can be committed by the user using the **commit** command.
- As long as no **commit** has been issued by the user, modifications can be undone using the **rollback** statement. In this case, the modified data blocks in the database buffer are overwritten by the original blocks stored in the rollback segments.
- If the user issues a **commit**, the space allocated for the blocks in the rollback segments is deallocated and can be used by other transactions. Furthermore, the modified blocks in the database buffer are unlocked such that other users now can read the modified blocks.
- The end of the transaction (more precisely the **commit**) is recorded in the redo-log files. The modified blocks are only written to the disk by the DBWR process if the space allocated for the blocks is needed for other blocks.