What Is Oracle Database

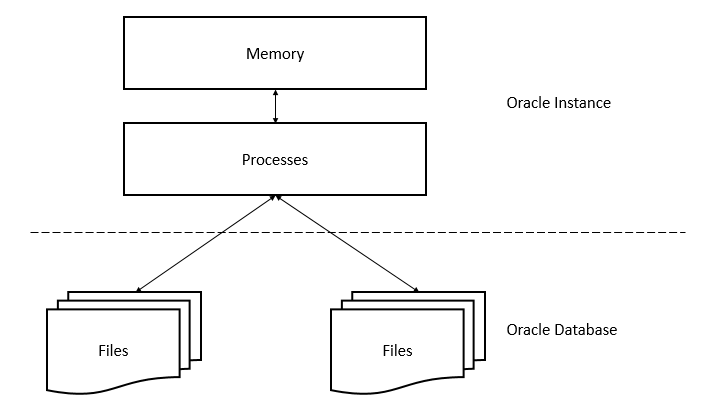
Oracle Database is an object-relational database management system developed and marketed by Oracle Corporation. Oracle Database is commonly referred to as Oracle RDBMS or simply Oracle.

## Database and Instance

An **Oracle Database** server consists of a **database** and at least one **database instance**. A database is a set of files that store data while a database instance is a set of memory structures which manages database files. The database instance also consists of background processes.

A database and an instance are closely connected therefore the term **Oracle Database** is often used to refer to both instance and database.

The following picture illustrates the Oracle Database server architecture:



In this architecture, an Oracle database server includes two main parts: files (Oracle Database) and memory (Oracle Instance).

## Oracle Database

One of the essential tasks of the Oracle Database is to store data. The following section briefly describes the physical and logical storage structure of an Oracle Database.

### Physical storage structures

The physical storage structures are simple files that store data. When you execute a CREATE DATABASEstatement to create a new database, the following files are created:

* Data Files: data files contain real data, e.g., sales orders and customers. The data of logical database structures such as tables and indexes are physically stored in the data files.
* Control files: every Oracle database has a control file that contains metadata. The metadata describes the physical structure of the database including database name and locations of data files.
* Online redo log files: every Oracle Database has an online redo log that consists of two or more online redo log files. An online redo log is made up of redo entries that record all changes made to the data.

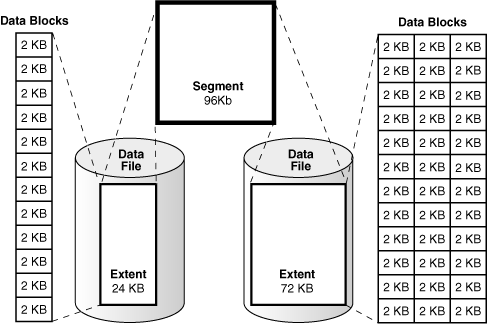
Besides these files, an Oracle database includes other important files such as parameter files, network files, backup files, and archived redo log files for backup and recovery.

### Logical Storage Structures

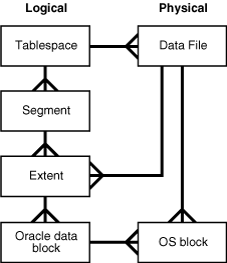
Oracle Database uses logical storage structure for a fine-grained control of disk space usage. The following are logical storage structures in an Oracle Database:

* Data blocks: a data block corresponds to a number of bytes on the disk. Oracle stores data in data blocks. Data blocks are also referred to as logical blocks, Oracle blocks or pages.
* Extents: an extent is a specific number of logically contiguous data blocks used to store the particular type of information.
* Segments: a segment is a set of extents allocated for storing user objects, e.g., a table or an index.
* Tablespaces: a database is divided into logical storage units called tablespaces. A tablespace is a logical container for a segment. Each tablespace consists of at least one data file.

The following picture illustrates segments, extents and data blocks within a tablespace:

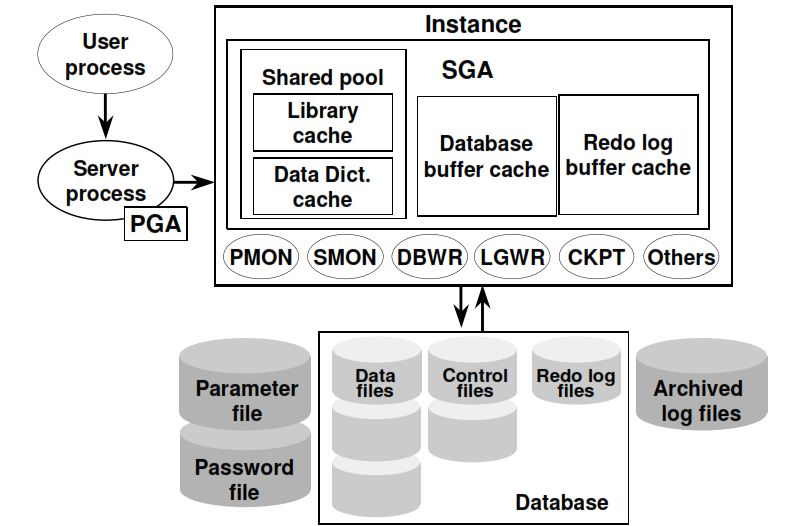


And the next figure shows the relationship between logical and physical storage structures:



## Oracle instance

Oracle instance is an interface between client applications (users) and database. An Oracle instance consists of three main parts: System Global Area (SGA), Program Global Area (PGA), and background processes.



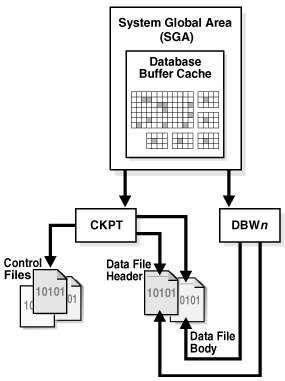
The SGA is a shared memory structure allocated when the instance started up and released when it is shut down. The SGA is a group of shared memory structures that contain data and control information for one database instance.

Different from the SGA, which is available to all processes, PGA is a private memory area allocated to each session when the session started and released when the session ends.

### Major Oracle Database’s background processes

The following are the major background processes of an Oracle instance:

* PMON is the process monitor that regulates all other processes. PMON cleans up abnormally connected database connections and automatically registers a database instance with the listener process. PMON is a single most process that must be alive in an Oracle database.
* SMON is the system monitor process that performs system level clean-up operation. It has two primary responsibilities including automatically instance recovery in the event of a failed instance, e.g., power failure and cleaning up of temporary files.
* DBWn is the database writer. Oracle performs every operation in memory instead of the disk because processing in memory is faster than on disk. The DBWn process reads data from disk and writes it back to the disk. An Oracle instance has many database writers DBW0, DBW1, DBW2, and so on.
* CKPT is the checkpoint process. In Oracle, data that is on disk is called block and the data which in memory is called buffer. When the block is written to the buffer and changed, the buffer becomes dirty, and it needs to be written down to the disk. The CKPT process updates the control and data file headers with checkpoint information and signals writing of dirty buffers to disk. Note that Oracle 12c allows both full and incremental checkpoints.



* LGWR is the log writer process which is the key to the recoverability architecture. Every change occurs in the database is written out to a file called redo log for recovery purposes. And these changes are written and logged by LGWR process. The LGWR process first writes the changes to memory and then disk as redo logs which then can be used for recovery.
* ARCn is the archiver process that copies the content of redo logs to archive redo log files. The archiver process can have multiple processes such as ARC0, ARC1, and ARC3, which allow the archiver to write to various destinations such D: drive, E drive or other storage.
* MMON is the manageability monitor process that gathers performance metrics.
* MMAN is the memory manager that automatically manages memory in an Oracle database.
* LREG is the listener registration process that registers information on the database instance and dispatcher processes with the Oracle Net Listener.

# Oracle Basics

## Section 1. Querying data

This section helps you learn how to query data from the Oracle Database. We will start with a simple query that allows you to retrieve data from a single table.

* [SELECT](http://www.oracletutorial.com/oracle-basics/oracle-select/) – show you how to query data from a single table.

## Section 2. Sorting data

* [ORDER BY](http://www.oracletutorial.com/oracle-basics/oracle-order-by/) – sort the result set of a query in ascending or descending order.

## Section 3. Filtering data

* [DISTINCT](http://www.oracletutorial.com/oracle-basics/oracle-select-distinct/)  – introduce you how to eliminate duplicate rows from the output of a query.
* [WHERE](http://www.oracletutorial.com/oracle-basics/oracle-where/) – learn how to specify a condition for rows in the result set returned by a query.
* [AND](http://www.oracletutorial.com/oracle-basics/oracle-and/) – combine two or more Boolean expressions and return true if all expressions are true.
* [OR](http://www.oracletutorial.com/oracle-basics/oracle-or/)–  combine two or more Boolean expressions and return true if one of the expressions is true.
* [FETCH](http://www.oracletutorial.com/oracle-basics/oracle-fetch/) – show you how to limit rows returned by a query using the row limiting clause.
* [IN](http://www.oracletutorial.com/oracle-basics/oracle-in/) – determine if a value matches any value in a list or a subquery.
* [BETWEEN](http://www.oracletutorial.com/oracle-basics/oracle-between/) – filter data based on a range of values.
* [LIKE](http://www.oracletutorial.com/oracle-basics/oracle-like/)  – perform matching based on specific patterns.

## Section 4. Joining tables

* [INNER JOIN](http://www.oracletutorial.com/oracle-basics/oracle-inner-join/) – show you how to query rows from a table that have matching rows from another table.
* [LEFT JOIN](http://www.oracletutorial.com/oracle-basics/oracle-left-join/) – introduce you to the left-join concept and learn how to use it to select rows from the left table that have or don’t have the matching rows in right table.
* [RIGHT JOIN](http://www.oracletutorial.com/oracle-basics/oracle-right-join/) – explain the right-join concept and show you how to apply it to query rows from the right table that have or don’t have the matching rows in the left table.
* [CROSS JOIN](http://www.oracletutorial.com/oracle-basics/oracle-cross-join/) – cover how to make a Cartesian product from multiple tables.
* [Self join](http://www.oracletutorial.com/oracle-basics/oracle-self-join/) – show you how to join a table to itself to query hierarchical data or compare rows within the same table.

## Section 5. Grouping data

* [GROUP BY](http://www.oracletutorial.com/oracle-basics/oracle-group-by/)– teach you how to group rows into subgroups and apply an aggregate function for each group
* [HAVING](http://www.oracletutorial.com/oracle-basics/oracle-having/) – show you how to filter group of rows.

## Section 6. Subquery

* [Subquery](http://www.oracletutorial.com/oracle-basics/oracle-subquery/)– introduce the concept of subquery and how to use the subqueries to perform advanced data selection techniques.
* [Correlated Subquery](http://www.oracletutorial.com/oracle-basics/oracle-correlated-subquery/) – learn about the correlated subquery which is a subquery that depends on the values returned by the outer query.
* [EXISTS](http://www.oracletutorial.com/oracle-basics/oracle-exists/) and [NOT EXISTS](http://www.oracletutorial.com/oracle-basics/oracle-not-exists/) – check for the existence of rows returned by a subquery.
* [ANY](http://www.oracletutorial.com/oracle-basics/oracle-any/), [SOME](http://www.oracletutorial.com/oracle-basics/oracle-any/), and [ALL](http://www.oracletutorial.com/oracle-basics/oracle-all/) – compare a value to a list or subquery.

## Section 7. Set Operators

This section walks you the steps of using the set operators to combine result sets of two or more independent queries.

* [UNION](http://www.oracletutorial.com/oracle-basics/oracle-union/) – show you how to combine the results of two queries into a single result.
* [INTERSECT](http://www.oracletutorial.com/oracle-basics/oracle-intersect/) – teach you how to make an intersection of the results of two independent queries.
* [MINUS](http://www.oracletutorial.com/oracle-basics/oracle-minus/) – learn how to subtract a result from another.

## Section 8. Modifying data

In this section, you’ll learn how to change the contents of an Oracle database. The SQL commands for modifying data are referred to as Data Manipulation Language (DML).

* [INSERT](http://www.oracletutorial.com/oracle-basics/oracle-insert/) – learn how to insert a row into a table.
* [INSERT INTO SELECT](http://www.oracletutorial.com/oracle-basics/oracle-insert-into-select/) – insert data into a table from the result of a query.
* [INSERT ALL](http://www.oracletutorial.com/oracle-basics/oracle-insert-all/) – discuss multitable insert statement to insert multiple rows into a table or multiple tables.
* [UPDATE](http://www.oracletutorial.com/oracle-basics/oracle-update/) – teach you how to change the existing values of a table.
* [DELETE](http://www.oracletutorial.com/oracle-basics/oracle-delete/) – show you how to delete one or more row from a table.
* [MERGE](http://www.oracletutorial.com/oracle-basics/oracle-merge/) – walk you through the steps of performing a mixture of insertion, update, and deletion using a single statement.

## Section 9. Data definition

This section shows you how to manage the most important database objects including databases and tables.

* [CREATE TABLE](http://www.oracletutorial.com/oracle-basics/oracle-create-table/) – walk you through the steps of creating new tables in the database.
* [Identity Column](http://www.oracletutorial.com/oracle-basics/oracle-identity-column/) – learn how to use the identity clause to define the identity column for a table.
* [ALTER TABLE](http://www.oracletutorial.com/oracle-basics/oracle-alter-table/) – teach you how to change the structure of existing tables.
* [ALTER TABLE ADD column](http://www.oracletutorial.com/oracle-basics/oracle-alter-table-add-column/) – show you how to add one or more columns to an existing table
* [ALTER TABLE MODIFY column](http://www.oracletutorial.com/oracle-basics/oracle-alter-table-modify-column/) – show you how to change the definition of existing columns in a table.
* [Drop columns](http://www.oracletutorial.com/oracle-basics/oracle-drop-column/) – learn how to use various statements to drop one or more columns from a table.
* [DROP TABLE](http://www.oracletutorial.com/oracle-basics/oracle-drop-table/) – show you how to delete tables from the database.
* [TRUNCATE TABLE](http://www.oracletutorial.com/oracle-basics/oracle-truncate-table/) – delete all data from a table faster and more efficiently.
* [RENAME table](http://www.oracletutorial.com/oracle-basics/oracle-rename-table/) –  walk you through the process of renaming a table and handling its dependent objects.

## Section 10. Oracle data types

* [Oracle data types](http://www.oracletutorial.com/oracle-basics/oracle-data-types/) – give you an overview of the built-in Oracle data types.
* [NUMBER](http://www.oracletutorial.com/oracle-basics/oracle-number-data-type/) – introduces you to the numeric data type and show you how to use it to define numeric columns for a table.
* [FLOAT](http://www.oracletutorial.com/oracle-basics/oracle-float/) – demystify float data type in Oracle by practical examples.
* [CHAR](http://www.oracletutorial.com/oracle-basics/oracle-char/) – learn about fixed-length character string.
* [NCHAR](http://www.oracletutorial.com/oracle-basics/oracle-nchar/) –  show you how to store fixed-length Unicode character data and explain the differences between CHAR and NCHAR data types
* [VARCHAR2](http://www.oracletutorial.com/oracle-basics/oracle-varchar2/) – introduce you to the variable-length character and show you how to define variable-length character columns in a table.
* [NVARCHAR2](http://www.oracletutorial.com/oracle-basics/oracle-nvarchar2/) – learn how to store variable-length Unicode characters in the database.
* [DATE](http://www.oracletutorial.com/oracle-basics/oracle-date/) – discuss the date and time data type and show you how to handle date time data effectively.
* [TIMESTAMP](http://www.oracletutorial.com/oracle-basics/oracle-timestamp/) – introduce you how to store date and time with the fractional seconds precision.
* [INTERVAL](http://www.oracletutorial.com/oracle-basics/oracle-interval/)– focus on the interval data types to store periods of time.
* [TIMESTAMP WITH TIME ZONE](http://www.oracletutorial.com/oracle-basics/oracle-timestamp-with-time-zone/) – learn how to store datetime with timezone data.

## Section 11. Constraints

* [Primary key](http://www.oracletutorial.com/oracle-basics/oracle-primary-key/)  – explain you to the primary key concept and show you how to use primary key constraint to manage a primary key of a table.
* [Foreign key](http://www.oracletutorial.com/oracle-basics/oracle-foreign-key/) – introduce you to the foreign key concept and show you use foreign key constraint to enforce the relationship between tables.
* [NOT NULL constraint](http://www.oracletutorial.com/oracle-basics/oracle-not-null/) – show you how to ensure a column not to accept null values.
* [UNIQUE constraint](http://www.oracletutorial.com/oracle-basics/oracle-unique-constraint/) – discuss how to ensure data stored in a column or a group of columns is unique among rows within the whole table.
* [CHECK constraint](http://www.oracletutorial.com/oracle-basics/oracle-check-constraint/) – walk you through the process of adding logic for checking data before storing them in tables.