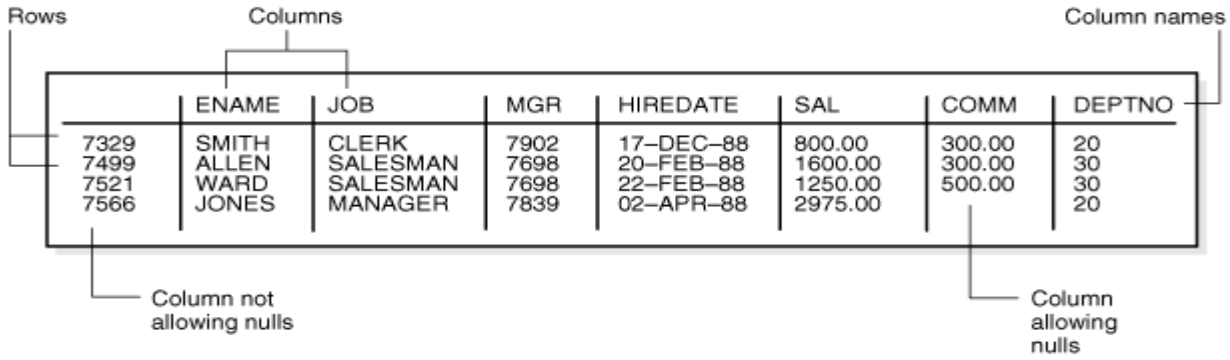
# EXPERIMENT NO.1

**Aim- To study about Oracle and SQL.**

## ORACLE

Oracle database (Oracle DB) is a relational database management system (RDBMS) from the Oracle Corporation. The system is built around a relational database framework in which data objects may be directly accessed by users (or an application front end) through structured query language (SQL). Oracle is a fully scalable relational database architecture and is often used by global enterprises, which manage and process data across wide and local area networks. The Oracle database has its own network component to allow communications across networks.

A table is the basic unit of data organization in an Oracle database. A table describes an **entity**, which is something of significance about which information must be recorded. For example, an employee could be an entity.




**Datatypes**

A datatype associates a fixed set of properties with the values that can be used in a column of a table or in an argument of a procedure or function. These properties cause Oracle to treat values of one datatype differently from values of another datatype. For example, Oracle can add values of NUMBER datatype, but not values of RAW datatype.

Oracle supplies the following built-in datatypes:

**Character datatypes**

CHAR

NCHAR

VARCHAR2 and VARCHAR

NVARCHAR2

CLOB

NCLOB

LONG

**Numeric datatypes**

NUMBER

**Time and date datatypes:**

DATE

INTERVAL DAY TO SECOND

INTERVAL YEAR TO MONTH

TIMESTAMP

TIMESTAMP WITH TIME ZONE

TIMESTAMP WITH LOCAL TIME ZONE

**Binary datatypes**

BLOB

BFILE

RAW

LONG RAW

Another datatype, ROWID, is used for values in the ROWID pseudo column, which represents the unique address of each row in a table.

| **Datatype** | **Description** | **Column Length and Default** |
|---|---|---|
| CHAR (size) | Fixed-length character data of length size bytes. | Fixed for every row in the table (with trailing blanks); maximum size is 2000 bytes per row, default size is 1 byte per row. Consider the character set (one-byte or multibyte) before setting size. |
| VARCHAR2 (size) | Variable-length character data. | Variable for each row, up to 4000 bytes per row. Consider the character set (one-byte or multibyte) before setting size. A maximum size must be specified. |
| NCHAR(size) | Fixed-length character data of length size characters or bytes, depending on the national character set. | Fixed for every row in the table (with trailing blanks). Column size is the number of characters for a fixed-width national character set or the number of bytes for a varying-width national character set. Maximum size is determined by the number of bytes required to store one character, with an upper limit of 2000 bytes per row. Default is 1 character or 1 byte, depending on the |

| | | |
|---|---|---|
| | | character set. |
| NVARCHAR2 (size) | Variable-length character data of length size characters or bytes, depending on national character set. A maximum size must be specified. | Variable for each row. Column size is the number of characters for a fixed-width national character set or the number of bytes for a varying-width national character set. Maximum size is determined by the number of bytes required to store one character, with an upper limit of 4000 bytes per row. Default is 1 character or 1 byte, depending on the character set. |
| CLOB | Single-byte character data. | Up to 2^32 - 1 bytes, or 4 gigabytes. |
| NCLOB | Single-byte or fixed-length multibyte national character set (NCHAR) data. | Up to 2^32 - 1 bytes, or 4 gigabytes. |
| LONG | Variable-length character data. | Variable for each row in the table, up to 2^31 - 1 bytes, or 2 gigabytes, per row. Provided for backward compatibility. |
| NUMBER (p, s) | Variable-length numeric data. Maximum precision p and/or scale s is 38. | Variable for each row. The maximum space required for a given column is 21 bytes per row. |
| DATE | Fixed-length date and time data, ranging from Jan. 1, 4712 B.C.E. to Dec. 31, 4712 C.E. | Fixed at 7 bytes for each row in the table. Default format is a string (such as DD-MON-YY) specified by NLS_DATE_FORMAT parameter. |
| BLOB | Unstructured binary data. | Up to 2^32 - 1 bytes, or 4 gigabytes. |
| BFILE | Binary data stored in an external file. | Up to 2^32 - 1 bytes, or 4 gigabytes. |
| RAW (size) | Variable-length raw binary data. | Variable for each row in the table, up to 2000 bytes per row. A maximum size must be specified. Provided for backward compatibility. |

| | | |
|---|---|---|
| LONG RAW | Variable-length raw binary data. | Variable for each row in the table, up to 2^31 - 1 bytes, or 2 gigabytes, per row. Provided for backward compatibility. |
| ROWID | Binary data representing row addresses. | Fixed at 10 bytes (extended ROWID) or 6 bytes (restricted ROWID) for each row in the table. |

## SQL

**SQL**, which is an abbreviation for **Structured Query Language**, is a language to request data from a database, to add, update, or remove data within a database, or to manipulate the metadata of the database.

SQL is a declarative language in which the expected result or operation is given without the specific details about how to accomplish the task. The steps required to execute SQL statements are handled transparently by the SQL database. Sometimes SQL is characterized as non-procedural because procedural languages generally require the details of the operations to be specified, such as opening and closing tables, loading and searching indexes, or flushing buffers and writing data to file systems. Therefore, SQL is considered to be designed at a higher conceptual level of operation than procedural languages because the lower level logical and physical operations are not specified and are determined by the SQL engine or server process that executes it.

Instructions are given in the form of statements, consisting of a specific SQL statement and additional parameters and operands that apply to that statement. SQL statements and their modifiers are based upon official SQL standards and certain extensions to that each database provider implements. Commonly used statements are grouped into the following categories:

**A) Data Definition Language (DDL)-** It is used to define the database structure or schema. DDL is also used to specify additional properties/constraints of the data.

1. **CREATE** – It is used to create a new table, a view of a table, or other object in database.

   **SYNTAX:**

   CREATE TABLE table_name (
   column1 datatype,
   column2 datatype,
   column3 datatype,
   .....
   columnN datatype,
   PRIMARY KEY( one or more columns ) );

2. **SELECT** – It is used to retrieve certain records from one or more tables.

   **SYNTAX:**

SELECT column1, column2....columnN

FROM table_name;

3. **ALTER** – It is used to modify an existing database object, such as a table.

   **SYNTAX:**

   ALTER TABLE table_name {ADD|DROP|MODIFY} column_name {data_ype};

   FOR RENAME: ALTER TABLE table_name RENAME TO new_table_name;

4. **DROP** – It is used to delete an entire table, a view of a table or other object in the database.

   **SYNTAX:**

   DROP TABLE table_name;

**B) Data Manipulation Language (DML) -** It allows us to modify the database instance by inserting, modifying, and deleting its data. It is responsible for performing all types of data modification in a database.

1. **INSERT** – It is used to create a record.

**SYNTAX:**

INSERT INTO table_name( column1, column2....columnN)

VALUES ( value1, value2....valueN);

2. **UPDATE** – It is used to change certain records.

**SYNTAX:**

UPDATE table_name

SET column1 = value1, column2 = value2....columnN=valueN

[ WHERE CONDITION ];

3. **DELETE** – It is used to delete certain records.

**SYNTAX:**

DELETE FROM table_name

WHERE {CONDITION};

**C) Data Control Language (DCL)-** It is used to grant and take back authority from any database user.

**1.GRANT** – It is used to give a privilege to someone.

**SYNTAX:**

GRANT privilege_name

ON object_name

TO {user_name |PUBLIC |role_name}

[WITH GRANT OPTION];

**2.REVOKE** - Used to take back privileges granted to someone.

**SYNTAX:**

REVOKE privilege_name

ON object_name

FROM {user_name |PUBLIC |role_name}

**D) Transaction Control Language (TCL)-** It is used to manage transactions in the database. These are used to manage the changes made by DML-statements. It also allows statements to be grouped together into logical transactions.

**1. COMMIT-** It is used to save all the transactions to the database.

**SYNTAX:**

COMMIT;

**2. ROLLBACK-** It is used to undo transactions that have not already been saved to the database.

**SYNTAX:**

ROLLBACK;

**3.SAVEPOINT-** It is used to roll the transaction back to a certain point without rolling back the entire transaction.

**SYNTAX:**

SAVEPOINT SAVEPOINT_NAME;