**Objective**: Learning about Oracle 9i and learn and implement the basic syntax of SQL like create table, insert data into table etc.

**What is Oracle 9i?** : **Oracle 9i** is a version of the **Oracle** Database. The ‘i’ stands for "Internet" to indicate that **9i** is "Internet ready".

**Data types in Oracle 9i**:

|  |  |  |  |
| --- | --- | --- | --- |
| Data type | Description | Size | Memory |
| CHAR(size) | This data type is used to store character strings values of fixed length. The size in brackets determines the number of characters the cell can hold | 255 Characters | Up to 32767 Bytes in PLSQL.  Up to 4000 Bytes in Oracle 9i. |
| VARCHAR(size)/ VARCHAR2(size) | This data type is used to store variable length alphanumeric data. It is a more flexible form of the CHAR data type. | 4000 characters | Up to 32767 Bytes in PLSQL.  Up to 4000 Bytes in Oracle 9i. |
| DATE | This data type is used to represent date and time. The standard format is DD-MON-YY as in 214UN-04. To enter dates other than the standard format, use the appropriate functions. Date Time stores date in the 24-hour format. By default, the time in a date field is 12:00:00 am, if no time portion is specified. The default date for a date field is the first day of the current month. |  |  |
| NUMBER(size) | The NUMBER data type is used to store numbers (fixed or floating point). Numbers of (P, S) virtually any magnitude maybe stored up to 38 digits of precision. Valid values are 0, and positive and negative numbers with magnitude 1.0E430 to 9.9E125. | Maximum 38 digits |  |
|  |  |  |  |
| LONG | This data type is used to store variable length character strings containing up to 2 PB. LONG data can be used to store arrays of binary data in ASCII format. Only one LONG value can be defined per table. |  | Up to 2 gigabytes |
| RAW/ LONG RAW | The RAW /LONG RAW data types are used to store binary data, such as digitized LONG picture or image. Data loaded into columns of these data types are stored without any RA W further conversion. |  | Up to 2 gigabytes |

**What is SQL?** : Elaboration of SQL is Structure Query Language. SQL is a language that provides an interface to relational database systems. SQL was develop by IBM in the 1970s for use in System R.

**Feature of SQL:**

1. SQL can be used by a range of users, including those with little or no programming experiences.

2. It is a non-procedural language

3. It reduces the amount of time required for creating and maintaining

4. It is an English-like language.

**DDL:** It is a set of SQL commands to create, modify and delete database structures but not data. It includes CREATE, ALTER, DROP, TRUNCATE, COMMENT, GRANT, REVOKE statements.

**DML:** It is the area of SQL that allows changing data within the database. It includes INSERT, UPDATE, DELETE, CALL, EXPLAIN PLAN, LOCK statements.

**DCL:** It is the component of SQL statements that control access to data and to the database. It contains COMMIT, SAVEPOINT, ROLLBACK, SET TRANSACTION, GRANT, REVOKE statements.

**DQL:** It is the component of SQL statement that allows getting data from the database and imposing ordering on it. It contains SELECT statement.

**Some SQL Operation and Syntax:**

**Create Table:** CREATE TABLE <table name> (<ColumnName> <DataType>(<size>), <ColumnName2> <DataType>(<size>));

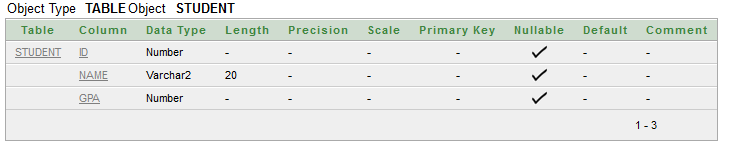
Ex: CREATE TABLE student (id NUMBER, name VARCHAR2(20), gpa NUMBER);

**Output:** Table created.

**Show the structure of table:** DESCRIBE <TableName>;

Ex: DSCRIBE student;

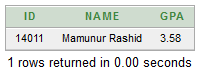
**Output:**



**Insert data into table:** INSERT INTO <TableName> (<ColumnName>, <ColumnName2>, ColumnName3>) VALUES(<expression>, <expression2>, <expression3>);

**Ex:** INSERT INTO student(id, name, gpa)VALUES(14011,'Mamunur Rashid', 3.58);

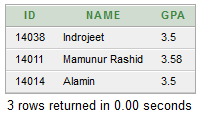
**Output:** 1 row(s) inserted.



**Viewing Data in the tables:** SELECT \*FROM <TableName>;

**Ex:** SELECT \*FROM student;

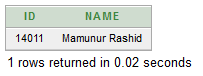
**Output:**



**View selected Column and all Rows:** SELECT <ColumnName1>, <ColumnName2> FROM <TableName>;

**Ex:** SELECT id, name FROM student;

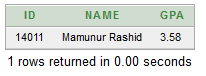
**Output:**



**View selected Rows and all Column:** SELECT \*FROM <TableName> WHERE <Condition>;

**Ex:** SELECT \*FROM student WHERE id=14011;

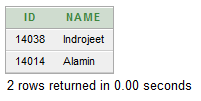
**Output:**



**View selected Columns and selected Rows:** SELECT <ColumnName1>, <ColumnName2> FROM <TableName> WHERE <Condition>;

**Ex:** SELECT id, name FROM student WHERE gpa=3.50;

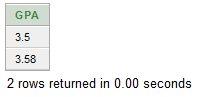
**Output:**



**View distinct rows:** SELECT DISTINCT <ColumnName1>, <ColumnName2> FROM <TableName>;

**Ex:** SELECT DISTINCT gpa FROM student;

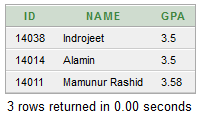
**Output:**



**Sorting data in a table:** SELECT \*FROM <TableName> ORDER BY <ColumnName1>, <ColumnName2> <[Sort Order]>;

**Ex:** SELECT \*FROM student ORDER BY gpa;

**Output:**

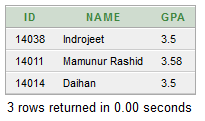


**Update data in table:** UPDATE <TableName> SET <ColumnName> = <Value> WHERE <Condition>;

**Ex:** UPDATE student SET name='Daihan' WHERE id=14014;

**Output:**

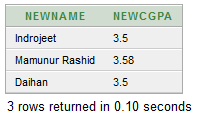
1 row(s) updated.



**Creating table from a table:** CREATE TABLE <TableName> (<ColumnName1>, <ColumnName2>) AS SELECT <ColumnName>, <ColumnName> FROM <TableName>;

**Ex:** CREATE TABLE studentInfo (newName, newCgpa) AS SELECT name, gpa FROM student;

**Output:** Table created.



**Delete all rows from a table:** DELETE FROM <TableName>;

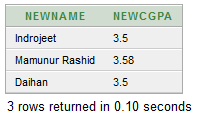
**Ex:** DELETE FROM studentInfo;

**Output:** 3 row(s) deleted.

**Insert data into ta table from another table:** INSERT INTO <TableName> SELECT <ColumnName1>, <ColumnName2> FROM <TableName>

**Ex:** INSERT INTO studentInfo SELECT name, gpa FROM student;

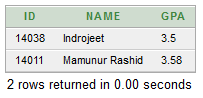
**Output:** 3 row(s) inserted.



**Delete specified row:** DELETE FROM <TableName>; WHERE <Condition>;

**Ex:** DELETE FROM student WHERE id=14014;

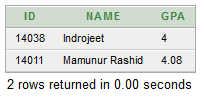
**Output:** 1 row(s) deleted.



**Update all rows:** UPDATE <TableName> SET <ColumnName> = expression;

**Ex:** UPDATE student SET gpa = gpa + 0.5;

**Output:** 2 row(s) updated.



**Adding new column:** ALTER TABLE <TableName> ADD <ColumnName>(<DataType(size));

**Ex:** ALTER TABLE student ADD (address VARCHAR2(20));

**Output:** Table altered.



**Modify existing column:** ALTER TABLE <TableName> MODIFY <ColumnName>(<DataType(size));

**Ex:** ALTER TABLE student MODIFY (Address VARCHAR2(15));

**Output:** Table altered.

**Changing Table name:** RENAME <PresentName> TO <NewName>;

**Ex:** RENAME student TO NewStudentInformation;

**Output:** Statement processed.

**Drop a table:** DROP TABLE <TabaleName>;

**Ex:** DROP TABLE NewStudentInformation;

**Output:** Table dropped.