Experiment No: 1

Title: Implementation of DDL commands of SQL with suitable examples

- Create table
- Alter table
- Drop Table

Objective:

- ✓ To understand the different issues involved in the design and implementation of a database system
- ✓ To understand and use data definition language to write query for a database

Theory:

Oracle has many tools such as SQL * PLUS, Oracle Forms, Oracle Report Writer, Oracle Graphics etc.

- ❖ **SQL * PLUS**: The SQL * PLUS tool is made up of two distinct parts. These are
    - **Interactive SQL:** Interactive SQL is designed for create, access and manipulate data structures like tables and indexes.
    - **PL/SQL:** PL/SQL can be used to developed programs for different applications.

- ❖ **Oracle Forms:** This tool allows you to create a data entry screen along with the suitable menu objects. Thus it is the oracle forms tool that handles data gathering and data validation in a commercial application.

- ❖ **Report Writer:** Report writer allows programmers to prepare innovative reports using data from the oracle structures like tables, views etc. It is the report writer tool that handles the reporting section of commercial application.

- ❖ **Oracle Graphics:** Some of the data can be better represented in the form of pictures. The oracle graphics tool allows programmers to prepare graphs using data from oracle structures like tables, views etc.

SQL (Structured Query Language):

Structured Query Language is a database computer language designed for managing data in relational database management systems (RDBMS), and originally based upon Relational Algebra. Its scope includes data query and update, schema creation and modification, and data access control.

SQL was one of the first languages for Edgar F. Codd's relational model and became the most widely used language for relational databases.

- IBM developed SQL in mid of 1970's.
- Oracle incorporated in the year 1979.
- SQL used by IBM/DB2 and DS Database Systems.
- SQL adopted as standard language for RDBS by ASNI in 1989.

DATA TYPES:

1. **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. The maximum number of character is 255 characters.

2. **VARCHAR (Size) / VARCHAR2 (Size)**: This data type is used to store variable length alphanumeric data. The maximum character can hold is 2000 character.

3. **NUMBER (P, S):** The NUMBER data type is used to store number (fixed or floating point). Number of virtually any magnitude may be stored up to 38 digits of precision. Number as large as $9.99 * 10^{124}$. The precision (p) determines the number of places to the right of the decimal. If scale is omitted then the default is zero. If precision is omitted, values are stored with their original precision up to the maximum of 38 digits.

4. **DATE:** This data type is used to represent date and time. The standard format is DD-MM-YY as in 17-SEP-2009. To enter dates other than the standard format, use the

appropriate functions. Date time stores date in the 24-Hours 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 the current month.

5. **LONG:** This data type is used to store variable length character strings containing up to 2GB. Long data can be used to store arrays of binary data in ASCII format. LONG values cannot be indexed, and the normal character functions such as SUBSTR cannot be applied.

6. **RAW:** The RAW data type is used to store binary data, such as digitized picture or image. Data loaded into columns of these data types are stored without any further conversion. RAW data type can have a maximum length of 255 bytes. LONG RAW data type can contain up to 2GB.

SQL language is sub-divided into several language elements, including:

- *Clauses*, which are in some cases optional, constituent components of statements and queries.
- *Expressions,* which can produce either scalar values or tables consisting of columns and rows of data.
- *Predicates* which specify conditions that can be evaluated to SQL three-valued logic (3VL) Boolean truth values and which are used to limit the effects of statements and queries, or to change program flow.
- *Queries* which retrieve data based on specific criteria.
- *Statements* which may have a persistent effect on schemas and data, or which may control transactions, program flow, connections, sessions, or diagnostics.
- SQL statements also include the semicolon (";") statement terminator. Though not required on every platform, it is defined as a standard part of the SQL grammar.

- *Insignificant white space* is generally ignored in SQL statements and queries, making it easier to format SQL code for readability.

There are five types of SQL statements. They are:

1. DATA DEFINITION LANGUAGE (DDL)
2. DATA MANIPULATION LANGUAGE (DML)
3. DATA RETRIEVAL LANGUAGE (DRL)
4. TRANSATIONAL CONTROL LANGUAGE (TCL)
5. DATA CONTROL LANGUAGE (DCL)

1. **DATA DEFINITION LANGUAGE (DDL):** The Data Definition Language (DDL) is used to create and destroy databases and database objects. These commands will primarily be used by database administrators during the setup and removal phases of a database project. Let's take a look at the structure and usage of four basic DDL commands:

1. CREATE 2. ALTER 3. DROP 4. RENAME

1. CREATE:

(a)CREATE TABLE: This is used to create a new relation (table)

Syntax: CREATE TABLE <relation_name/table_name >
(field_1 data_type(size),field_2 data_type(size), .. .);

Example:

```
SQL> CREATE TABLE Student (sno NUMBER (3), sname CHAR (10), class CHAR (5));
```

2. ALTER:

(a) ALTER TABLE ...ADD...: This is used to add some extra fields into existing relation.

Syntax: ALTER TABLE relation_name ADD (new field_1 data_type(size), new field_2 data_type(size),..);

Example: SQL>ALTER TABLE std ADD (Address CHAR(10));

(b) ALTER TABLE...MODIFY...: This is used to change the width as well as data type of fields of existing relations.

Syntax: ALTER TABLE relation_name MODIFY (field_1 newdata_type(Size), field_2 newdata_type(Size),… field_newdata_type(Size));

Example:SQL>ALTER TABLE student MODIFY(sname VARCHAR(10),class VARCHAR(5));

c) ALTER TABLE..DROP..... This is used to remove any field of existing relations.

Syntax: ALTER TABLE relation_name DROP COLUMN (field_name);

Example:SQL>ALTER TABLE student DROP column (sname);

d)ALTER TABLE..RENAME...: This is used to change the name of fields in existing relations.

Syntax: ALTER TABLE relation_name RENAME COLUMN (OLD field_name) to (NEW field_name);

Example: SQL>ALTER TABLE student RENAME COLUMN sname to stu_name;

3. DROP TABLE: This is used to delete the structure of a relation. It permanently deletes the records in the table.

Syntax: DROP TABLE relation_name;

Example: SQL>DROP TABLE std;

4. RENAME: It is used to modify the name of the existing database object.

Syntax: RENAME TABLE old_relation_name TO new_relation_name;

Example: SQL>RENAME TABLE std TO std1;

<u>LAB PRACTICE ASSIGNMENT:</u>

1. Create a table EMPLOYEE with following schema:

(Emp_no, E_name, E_address, E_ph_no, Dept_no, Dept_name,Job_id , Salary)

2. Add a new column; HIREDATE to the existing relation.

3. Change the datatype of JOB_ID from char to varchar2.

4. Change the name of column/field Emp_no to E_no.

5. Modify the column width of the job field of emp table

****

Experiment No:2

Title : Implementation of DML commands of SQL with suitable examples

- Insert table
- Update table
- Delete Table

Objective :

- ✓ To understand the different issues involved in the design and implementation of a database system
- ✓ To understand and use data manipulation language to query, update, and manage a database

Theory :

DATA MANIPULATION LANGUAGE (DML): The Data Manipulation Language (DML) is used to retrieve, insert and modify database information. These commands will be used by all database users during the routine operation of the database. Let's take a brief look at the basic DML commands:

1. INSERT 2. UPDATE 3. DELETE

1. INSERT INTO: This is used to add records into a relation. These are three type of INSERT INTO queries which are as

a) Inserting a single record

Syntax: INSERT INTO < relation/table name> (field_1,field_2……field_n)VALUES (data_1,data_2,........ data_n);

Example: SQL>INSERT INTO student(sno,sname,class,address)VALUES (1,’Ravi’,’M.Tech’,’Palakol’);

b) Inserting a single record

Syntax: INSERT INTO < relation/table name>VALUES (data_1,data_2,. data_n);

Example: SQL>INSERT INTO student VALUES (1,’Ravi’,’M.Tech’,’Palakol’);

c) Inserting all records from another relation

Syntax: INSERT INTO relation_name_1 SELECT Field_1,field_2,field_n FROM relation_name_2 WHERE field_x=data;

Example: SQL>INSERT INTO std SELECT sno,sname FROM student

WHERE name = ‘Ramu‘;

d) Inserting multiple records

Syntax: INSERT INTO relation_name field_1,field_2,…. field_n) VALUES

(&data_1,&data_2,....... &data_n);

Example: SQL>INSERT INTO student (sno, sname, class,address)

VALUES (&sno,’&sname’,’&class’,’&address’);

```
Enter value for sno: 101
Enter value for name: Ravi
Enter value for class: M.Tech
Enter value for name: Palakol
```

2. UPDATE-SET-WHERE: This is used to update the content of a record in a relation.

Syntax: SQL>UPDATE relation name SET Field_name1=data,field_name2=data,

WHERE field_name=data;

Example: SQL>UPDATE student SET sname = ‘kumar’ WHERE sno=1;

3. DELETE-FROM: This is used to delete all the records of a relation but it will retain the structure of that relation.

a) DELETE-FROM: This is used to delete all the records of relation.

Syntax: SQL>DELETE FROM relation_name;

Example: SQL>DELETE FROM std;

b) DELETE -FROM-WHERE: This is used to delete a selected record from a relation.

Syntax: SQL>DELETE FROM relation_name WHERE condition;

Example: SQL>DELETE FROM student WHERE sno = 2;

5. TRUNCATE: This command will remove the data permanently. But structure will not be removed.

Difference between Truncate & Delete:-

- ✓ By using truncate command data will be removed permanently & will not get back where as by using delete command data will be removed temporally & get back by using roll back command.
- ✓ By using delete command data will be removed based on the condition where as by using truncate command there is no condition.
- ✓ Truncate is a DDL command & delete is a DML command.

Syntax: TRUNCATE TABLE <Table name>

Example TRUNCATE TABLE student;

- **To Retrieve data from one or more tables.**

1. SELECT FROM: To display all fields for all records.

Syntax : SELECT * FROM relation_name;

Example : SQL> select * from dept;

```
DEPTNO      DNAME           LOC
 --------   -----------     ----------
10          ACCOUNTING      NEW YORK
20          RESEARCH        DALLAS
30          SALES           CHICAGO
40          OPERATIONS      BOSTON
```

2. SELECT FROM: To display a set of fields for all records of relation.

Syntax: SELECT a set of fields FROM relation_name;

Example: SQL> select deptno, dname from dept;

```
DEPTNO      DNAME
-------     ----------
10          ACCOUNTING
 20         RESEARCH
 30         SALES
```

3. SELECT - FROM -WHERE: This query is used to display a selected set of fields for a selected set of records of a relation.

Syntax: SELECT a set of fields FROM relation_name WHERE condition;

Example: SQL> select * FROM dept WHERE deptno<=20;

```
DEPTNO      DNAME            LOC
------      -----------      ------------
10          ACCOUNTING       NEW YORK
        20        RESEARCH       DALLAS
```

LAB PRACTICE ASSIGNMENT:

Create a table EMPLOYEE with following schema:

(Emp_no, E_name, E_address, E_ph_no, Dept_no, Dept_name,Job_id , Salary)

Write SQL queries for following question:

1. Insert aleast 5 rows in the table.
2. Display all the information of EMP table.
3. Display the record of each employee who works in department D10.
4. Update the city of Emp_no-12 with current city as Nagpur.
5. Display the details of Employee who works in department MECH.
6. Delete the email_id of employee James.
7. Display the complete record of employees working in SALES Department.

****

Experiment No: 3

Title: Implementation of different types of functions with suitable examples.

- Number Function
- Aggregate Function
- Character Function
- Conversion Function
- Date Function

Objective:

- ✓ *To understand and implement various types of function in SQL.*

NUMBER FUNCTION:

Abs(n) :Select abs(-15) from dual;

Exp(n): Select exp(4) from dual;

Power(m,n): Select power(4,2) from dual;

Mod(m,n): Select mod(10,3) from dual;

Round(m,n): Select round(100.256,2) from dual;

Trunc(m,n): ;Select trunc(100.256,2) from dual;

Sqrt(m,n);Select sqrt(16) from dual;

Develop aggregate plan strategies to assist with summarization of several data entries.

Aggregative operators: In addition to simply retrieving data, we often want to perform some computation or summarization. SQL allows the use of arithmetic expressions. We now consider a powerful class of constructs for computing aggregate values such as MIN and SUM.

1. Count: COUNT following by a column name returns the count of tuple in that column. If DISTINCT keyword is used then it will return only the count of unique tuple in the column. Otherwise, it will return count of all the tuples (including duplicates) count (*) indicates all the tuples of the column.

Syntax: COUNT (Column name)

Example: SELECT COUNT (Sal) FROM emp;

2. SUM: SUM followed by a column name returns the sum of all the values in that column.

Syntax: SUM (Column name)

Example: SELECT SUM (Sal) From emp;

3. AVG: AVG followed by a column name returns the average value of that column values.

Syntax: AVG (n1, n2...)

Example: Select AVG (10, 15, 30) FROM DUAL;

4. MAX: MAX followed by a column name returns the maximum value of that column.

Syntax: MAX (Column name)

Example: SELECT MAX (Sal) FROM emp;

```
SQL> select deptno, max(sal) from emp group by deptno;

    DEPTNO      MAX (SAL)
    ------   --------
    10        5000
    20        3000
    30        2850

SQL> select deptno, max (sal) from emp group by deptno having max(sal)<3000;

  DEPTNO     MAX(SAL)
  -----      --------
     30      2850
```

5. MIN: MIN followed by column name returns the minimum value of that column.

Syntax: MIN (Column name)

Example: SELECT MIN (Sal) FROM emp;

```
SQL>select deptno,min(sal) from emp group by deptno having min(sal)>1000;

  DEPTNO MIN (SAL)
  -----    --------
    10    1300
```

CHARACTER FUNCTION:

initcap(char) : select initcap(“hello”) from dual;

lower (char): select lower (‘HELLO’) from dual;

upper (char) :select upper (‘hello’) from dual;

ltrim (char,[set]): select ltrim (‘cseit’, ‘cse’) from dual;

rtrim (char,[set]): select rtrim (‘cseit’, ‘it’) from dual;

replace (char,search): select replace(‘jack and jue’,‘j’,‘bl’) from dual;

CONVERSION FUNCTIONS:

To_char: TO_CHAR (number) converts n to a value of VARCHAR2 data type, using the optional number format fmt. The value n can be of type NUMBER, BINARY_FLOAT, or BINARY_DOUBLE.

```
SQL>select to_char(65,'RN')from dual;

LXV
```

To_number : TO_NUMBER converts expr to a value of NUMBER data type.

```
SQL>Select to_number ('1234.64') from Dual;
1234.64
```

To_date:TO_DATE converts char of CHAR, VARCHAR2, NCHAR, or NVARCHAR2 data type to a value of DATE data type.

```
SQL>SELECT TO_DATE('January 15, 1989, 11:00 A.M.')FROM DUAL;

TO_DATE
---------
15-JAN-89
```

STRING FUNCTIONS:

Concat: CONCAT returns char1 concatenated with char2. Both char1 and char2 can be any of the datatypes

SQL>SELECT CONCAT('ORACLE','CORPORATION')FROM DUAL;

    ORACLECORPORATION

Lpad: LPAD returns expr1, left-padded to length n characters with the sequence of characters in expr2.

SQL>SELECT LPAD('ORACLE',15,'*')FROM DUAL;

    *********ORACLE

Rpad: RPAD returns expr1, right-padded to length n characters with expr2, replicated as many times as necessary.

SQL>SELECT RPAD ('ORACLE',15,'*')FROM DUAL;

    ORACLE*********

Ltrim: Returns a character expression after removing leading blanks.

SQL>SELECT LTRIM('SSMITHSS','S')FROM DUAL;

    MITHSS

Rtrim: Returns a character string after truncating all trailing blanks

SQL>SELECT RTRIM('SSMITHSS','S')FROM DUAL;

    SSMITH

Lower: Returns a character expression after converting uppercase character data to lowercase.

SQL>SELECT LOWER('DBMS')FROM DUAL;

    dbms

Upper: Returns a character expression with lowercase character data converted to uppercase

SQL>SELECT UPPER('dbms')FROM DUAL;

    DBMS

Length: Returns the number of characters, rather than the number of bytes, of the given string expression, excluding trailing blanks.

SQL>SELECT LENGTH('DATABASE')FROM DUAL;

    8

Substr: Returns part of a character, binary, text, or image expression.

```
SQL>SELECT SUBSTR(‘ABCDEFGHIJ’3,4)FROM DUAL;
        CDEF
```

Instr: The INSTR functions search string for substring. The function returns an integer indicating the position of the character in string that is the first character of this occurrence.

```
SQL>SELECT INSTR('CORPORATE FLOOR','OR',3,2)FROM DUAL;
        14
```

DATE FUNCTIONS:

Sysdate:

```
SQL>SELECT SYSDATE FROM DUAL;
        29-DEC-08
```

next_day:

```
SQL>SELECT NEXT_DAY(SYSDATE,’WED’)FROM DUAL;
        05-JAN-09
```

add_months:

```
SQL>SELECT ADD_MONTHS(SYSDATE,2)FROM DUAL;
        28-FEB-09
```

last_day:

```
SQL>SELECT LAST_DAY(SYSDATE)FROM DUAL;
        31-DEC-08
```

months_between:

```
SQL>SELECT MONTHS_BETWEEN(SYSDATE,HIREDATE)FROM EMP;
 4
```

Least:

```
SQL>SELECT LEAST('10-JAN-07','12-OCT-07')FROM DUAL;
        10-JAN-07
```

Greatest:

```
SQL>SELECT GREATEST('10-JAN-07','12-OCT-07')FROM DUAL;
        10-JAN-07
```

Trunc:

```
SQL>SELECT TRUNC(SYSDATE,'DAY')FROM DUAL;
        28-DEC-08
```

Round:

```
SQL>SELECT ROUND(SYSDATE,'DAY')FROM DUAL;
        28-DEC-08
```

to_char:

```
SQL> select to_char(sysdate, "dd\mm\yy") from dual;
        24-mar-05.
```

to_date:

```
SQL> select to date (sysdate, "dd\mm\yy") from dual;
        24-mar-o5.
```

<u>LAB PRACTICE ASSIGNMENT:</u>

Create a table EMPLOYEE with following schema:

(Emp_no, E_name, E_address, E_ph_no, Dept_no, Dept_name,Job_id, Designation , Salary)

Write SQL statements for the following query.

1. List the E_no, E_name, Salary of all employees working for MANAGER.
2. Display all the details of the employee whose salary is more than the Sal of any IT PROFF..
3. List the employees in the ascending order of Designations of those joined after 1981.
4. List the employees along with their Experience and Daily Salary.
5. List the employees who are either ‘CLERK’ or ‘ANALYST’ .
6. List the employees who joined on 1-MAY-81, 3-DEC-81, 17-DEC-81,19-JAN-80 .
7. List the employees who are working for the Deptno 10 or20.
8. List the Enames those are starting with ‘S’ .
9. Dislay the name as well as the first five characters of name(s) starting with ‘H’
10. List all the emps except ‘PRESIDENT’ & ‘MGR” in asc order of Salaries.

****