Sr.No.	Experiment	Page No		Date	Marks	Signature
		From	To			
1.	A) Introduction of Database management systems, Oracle concepts and Create a table. B) How to insert data in a table using insert and display the records in a table.					
2.	A) Update or Delete records of a table and modifying structure of a table using Alter and Drop command. B) Study of character functions for manipulation of data items.					

EXPERIMENT NO: 1 **DATE:**

A) **TITLE**: Introduction of Database Management Systems, SQL Concepts, Oracle concepts and Create a table.

OBJECTIVES: On completion of this experiment student will able to…

- know the concept of database management system.
- know the concept of Oracle.
- create a table in database.

THEORY:

❖ Introduction of Database Management Systems:

DBMS is a collection of interrelated data and a set of programs to access those data.
Primary goal of DBMS is to provide a way to store and retrieve database information that is both convenient and efficient.
Examples of DBMS are Banking System, Universities, Airlines, etc.

❖ Introduction of Oracle:

The relational model, sponsored by IBM (in June 1970), then came to accepted as the definitive model for RDBMS. The language developed by IBM to manipulate the data stored within model (Dr. E.F.Codd model) was originally called Structured English Query Language (SEQUEL) with the word English later dropped in favor Structured Query Language(SQL).
In 1979 a company called Relational Software, Inc. released the first commercially available implementation of SQL. Relational Software later come to be known as Oracle Corporation. Oracle Corporation is a company that produces the most widely used, Server based, Multi user RDBMS named Oracle.

❖ Oracle Tools:

The Oracle product is primarily divided into

Oracle Server tools: Oracle Server Product is either called Oracle Workgroup Server or Oracle Enterprise Server. Oracle Workgroup Server or Oracle Enterprise Server is used for data storage.

Oracle Client tools: The client roll most commonly used for Commercial Application Development is called Oracle Developer 2000. Oracle Developer 2000, Oracle's tool box which consists of Oracle Forms, Oracle Reports and Oracle Graphics. This suite of tools is used to capture, validate and display data according to user and system needs.

❖ SQL (Structured Query Language):

SQL (Structured Query Language) is a database sublanguage for querying and modifying relational databases. It was developed by IBM Research in the mid 1970 and standardized by ANSI in 1986.

❖ Components of SQL:

1) DDL (Data Definition Language):

Is a language, which includes the commands, which are used dynamically to set up, change and remove any data structure e.g. tables, views and indexes. The examples are CREATE, ALTER & DROP.

2) DML (Data Manipulation Language):

Is a language, which includes the commands, which are used to enter new rows, change existing rows and remove unwanted rows from the tables in database. The examples are INSERT, UPDATE & DELETE.

3) DCL (Data Control Language):
Is a language, which includes the commands, which are used to give or remove access rights to both the Oracle database and the structures within it. The examples are GRANT & REVOKE.

4) DQL (Data Query Language):
It is the component of SQL statement that allows getting data from the database and imposing ordering upon it. In includes the SELECT statement. It allows getting the data out of the database perform operations with it.

❖ **The CREATE TABLE command:** The CREATE TABLE command defines each column of the table uniquely. Each column has a minimum of three attributes, a name, datatype and size (i.e. column width).

Rules for creating Tables:
1. A name can have maximum upto 30 characters.
2. Alphabets from A-Z, a-z and numbers from 0-9 are allowed.
3. A name should begin with an alphabet.
4. The use of the special character like _ is allowed and alsorecommended (Special characters like $, # are allowed only in Oracle).
5. SQL reserved words not allowed. For example: create, select and so on.

SYNTAX	**CREATE TABLE <tablename>
	(<ColumnName1> <DataType>(<size>),
	<ColumnName2> <DataType>(<size>), ……);**

EXAMPLE	**Create table** client_master (c_no varchar2(5), name varchar2(10), address varchar2(20), pincode number(6), bal_due number(10,2));

EXCERCISE:

1) Create a table "emp" with the following fields:

EMPNO ENAME JOB HIREDATE SAL COMM DEPTNO MGR

2) Create a table "dept" with the following fields: DEPTNO DNAME LOCATION

3) Create a table "stud_master" with the following fields: REG_NO S_NAME BRANCH

4) Create a table "stud_detail" with the following fields:

REG_NO COURSE_CODE COURSE_NAME MARKS SEM

EXPERIMENT-1 cont- **DATE:**

B) **TITLE**: How to insert data in a table using insert command and display the records in a table.

OBJECTIVES: On completion of this experiment student will able to…

- insert records into a table.
- display records from a table.

THEORY:

❖ Inserting Data into Tables using INSERT INTO command:

Once a table is created, most natural thing to do is load this table with data to be manipulated later.
When inserting a single raw of data into the table, insert operation:
Creates a new raw (empty) in the databasetable.
Loads the values passed (by the SQL insert) into the columns specified.

SYNTAX	**INSERT INTO** <tablename> **[(**<ColumnName1>, <ColumnName2>, …… **)] VALUES**(<value1>,< value2>, ……)**;**

EXAMPLE	**INSERT INTO client_master** (c_no, name, address, pincode, bal_due) **VALUES** ('C001', 'Obaid', 'A-5, Lucknow', 384120, 500)**;**

Note: Character value (expression) placed within the INSERT INTO statement must be enclosed in single quotes (').

❖ Display / Viewing data in the Tables using SELECT command:

Once data has been inserted into a table, the next most logical operation would be to view what has been inserted. The SELECT SQL verb is used to achieve this.

The SELECT command is used to retrieve rows selected from one or more tables.

The SELECT statement can be used to Display some or all the columns from a specified table.
Display some or all of the rows from a specified table. Display calculated values from the table.
Display statistical information from the tables, like averages or sums of column values.
Combine information from two or more tables

In order to view global table data the syntax is:

SELECT <ColumnName 1> TO <ColumnName N> from TableName;

Note: Here**, ColumnName 1 to ColumnName N** represents table column names and they **separated by ','.**

<u>All Rows and All Columns:</u> When data from all rows and columns from the table are to be viewed the syntax of the SELECT statement will be used. The syntax is.

SYNTAX	**SELECT * FROM** <TableName>**;**

EXAMPLE	**SELECT * FROM** client_master**;**

- **Oracle allows the use of the Meta character asterisk (*), this is expanded by Oracle to mean all rows and all columns in the table**.

Displaying Some Columns from a Table:

SYNTAX	**SELECT** <ColumnName 1>,<ColumnName 2>, …,
	<ColumnName N>
	FROM <TableName>**;**

EXAMPLE	**SELECT** c_no, name **FROM** client_master;

❖ Displaying Some Specified Rows from the Table:

If you want conditional retrieval of rows i.e. only those rows which satisfy certain condition. You can use WHERE clause in the SELECT statement

SYNTAX	**SELECT** <ColumnName 1>,<ColumnName 2>, …,
	<ColumnName N>
	FROM <TableName>
	WHERE <Condition>**;**

Here, **<Condtion>** is always quantified as **<ColumnName = Value>**.

EXAMPLE	**SELECT** c_no, name **FROM** client_master **WHERE** bal_due>500;

Elimination of duplicates from the select statement:

A table could hold duplicate rows. In such a case, to see only unique rows, you have

to use DISTINCT clause. The **DISTINCT** clause allows removing duplicates from the result set. The DISTINCT clause can be only be used with SELECT statements

SYNTAX	**SELECT DISTINCT** <ColumnName 1>, …,
	<ColumnName N>
	FROM <TableName>**;**

EXAMPLE	**SELECT DISTINCT** job **FROM** emp;

The **SELECT DISTINCT *** SQL syntax scans through **entire rows**, and eliminates rows that have exactly the same contents in each column

SYNTAX	**SELECT DISTINCT * FROM <TableName>;**

EXAMPLE	**SELECT DISTINCT * FROM** client_master**;**

Sorting data in a Table:

Oracle allows data from a table to be viewed in a sorted order. The rows retrieved from the table will be sorted in either **ascending** or **descending** order depending on the condition specified in the SELECTsentence.

SYNTAX	**SELECT** <ColumnName 1>, … <ColumnName N>**FROM** <TableName> **ORDER BY** <ColumnName 1>, …, <ColumnName N> **<[Sort Order]>;**

EXAMPLE	**a. SELECT * FROM** client_master **ORDER BY** Name**;** (In ascending order) **b. SELECT * FROM** emp **ORDER BY**Job **DESC;** (In descending order)

The format to display the records -
SELECT[DISTINCT]{*, column[alias],…} FROM table
WHERE condition(s) Group by column(s)
HAVING group of row condition(s) ORDER BY {column. Expr} [ASC/DESC];

EXERCISES:

1) Insert records into **emp** table.

2) Insert records into **dept** table.

3) Insert records into **stud_master** table.

4) Insert records into **stud_detail** table.

5) Select all information from **emp** table.

6) List all the employees who have salary between 1000 and 2000.

7) List names and jobs of all clerks in department 20.

8) Display all the different job types.

9) List department numbers and names in department name order.

10) Select all information from **stud_master** table.

11) Display Registration number and name of students whose department is "computer engineering".