ASSIGNMENT 1

DBL-1

Aim:

DBMS using connections(Client-Data sever, two tier) Oracle/MySQL (ODBC/JDBC), SQL prompt to create data base tables insert, update data values, delete table, use table, select queries with/without where clause. ,demonstrate use of stored procedure / function (create procedure at the data side and make use of it on the client side).

Theory:

A) CREATE Table

The CREATE TABLE statement is used to create a table in a database.Tables are organized into rows and columns; and each table must have a name.

Syntax:

CREATE TABLE *table_name*
(
column_name1 data_type(*size*),
column_name2 data_type(*size*),
column_name3 data_type(*size*),

);

The column_name parameters specify the names of the columns of the table.

The data_type parameter specifies what type of data the column can hold (e.g. varchar, integer, decimal, date, etc.).

The size parameter specifies the maximum length of the column of the table.

B) ALTER Table

The ALTER TABLE statement is used to add, delete, or modify columns in an existing table.

Syntax:

ALTER TABLE table_name
MODIFY COLUMN column_name datatype

1)To add a column in a table, use the following syntax:

ALTER TABLE table_name
ADD column_name datatype

2) To delete a column in a table, use the following syntax (notice that some database systems don't allow deleting a column):

```
ALTER TABLE table_name
DROP COLUMN column_name
```

3) Now we want to change the data type of the column named "DateOfBirth" in the "Persons" table.

We use the following SQL statement:

```
ALTER TABLE Persons
ALTER COLUMN DateOfBirth year
```

4) Next, we want to delete the column named "DateOfBirth" in the "Persons" table. We

use the following SQL statement:

```
ALTER TABLE Persons
DROP COLUMN DateOfBirth
```

C) View

In SQL, a view is a virtual table based on the result-set of an SQL statement.

A view contains rows and columns, just like a real table. The fields in a view are fields from one or more real tables in the database.

You can add SQL functions, WHERE, and JOIN statements to a view and present the data as if the data were coming from one single table.

1)SQL CREATE VIEW Syntax

```
CREATE VIEW view_name AS
SELECT column_name(s)
FROM table_name
WHERE condition
```

2) We can query the view above as follows:

```
SELECT * FROM [ViewName]
```

3) You can update a view by using the following syntax:

SQL CREATE OR REPLACE VIEW Syntax

```
CREATE OR REPLACE VIEW view_name AS
SELECT column_name(s)
FROM table_name
WHERE condition
```

4) You can delete a view with the DROP VIEW command.

SQL DROP VIEW Syntax

```
DROP VIEW view_name
```

D) JOINS

SQL joins are used to combine rows from two or more tables.

An SQL JOIN clause is used to combine rows from two or more tables, based on a common field between them.

The most common type of join is: **SQL INNER JOIN (simple join)**. An SQL INNER JOIN return all rows from multiple tables where the join condition is met.

Different SQL JOINs

Before we continue with examples, we will list the types of the different SQL JOINs you can use:

- **INNER JOIN**: Returns all rows when there is at least one match in BOTH tables
- **LEFT JOIN**: Return all rows from the left table, and the matched rows from the right table
- **RIGHT JOIN**: Return all rows from the right table, and the matched rows from the left table
- **FULL JOIN**: Return all rows when there is a match in ONE of the tables

E) INDEX

An index can be created in a table to find data more quickly and efficiently.

The users cannot see the indexes, they are just used to speed up searches/queries.

Note: Updating a table with indexes takes more time than updating a table without (because the indexes also need an update). So you should only create indexes on columns (and tables) that will be frequently searched against.

1) SQL CREATE INDEX Syntax

Creates an index on a table. Duplicate values are allowed:

```
CREATE INDEX index_name
ON table_name (column_name)
```

2) SQL CREATE UNIQUE INDEX Syntax

Creates a unique index on a table. Duplicate values are not allowed:

```
CREATE UNIQUE INDEX index_name
ON table_name (column_name)
```

3)DROP INDEX

```
ALTER TABLE table_name DROP INDEX index_name
```

CONCLUSION :

Thus we have studied Syntax and Use of DDL Statements.

*Try to solve as per steps given below………**All students T.E.Comp – Div II***

Step 1. Create a Category table based on the table instance given below. Confirm the table is created.

Attributes	**Datatype**	**Length**
CategoryID	Char	3
Category	Char	20
Description	Varchar2	100

Step2. Create a Toybrand table based on the table instance given below. Confirm the table is created.

Attributes	**Datatype**	**Length**
BrandID	Char	3
BrandName	Varchar2	20

Step3. Create a Toys table based on the table instance given below. Confirm the table is created

Attributes	**Datatype**	**Length**
ToyID	Char	6
ToyName	Varchar2	10
ToyDesc	Varchar2	25
CategoryID	Char	3
ToyRate	Number	5,2
BrandID	Char	3
ToyQoh	Number	5
LowerAge	Number	3
UpperAge	Number	3
ToyWeight	Number	6

Step4. Enforce the following Integrity rules while creating the Category Table.

a. CategoryID should be the primary key
b. Category must be unique but not primary key.

c. Description of categories can allow storage of Null values.

Step5. Enforce the following Integrity rules while creating the ToyBrand Table.

a. The BrandID must be the primary key

b. BrandName must be unique but not primary key.

Step6. Enforce the following Integrity rules while creating the Toys Table.

a. The ToyID must be the primary key
b. The QOH of the toys must be between 0 and 200
c. The Toy Name and Description should not allow NULL values.
d. The lower age of toys must be 1 by default.
e. The Values of CategoryID should be present in Category table.

Step7. Modify the toys table to enforce the following Data Integrity rules.

a. The upper age for toys should be 1 by default.
b. The price of the toys should be greater than zero.
c. The weight of the toys should be 1 by default.