LAB # 2

RETRIEVING DATA FROM DATABASE

OBJECTIVE

Retrieve data from your database using SELECT Statement

THEORY

Introduction to SQL

SQL (Structured Query Language) is the standard language used for creating, updating and querying relational databases. It is supported by all modern database management systems (e.g. Oracle, IBM DB2, Microsoft Access, Microsoft SQL Server, PostgreeSQL, MySQL, etc.).

SQL is a declarative language (as opposed to a procedural language like C, Perl, etc.). This means that the language is used by the programmer to directly describe what end result is required (e.g. what data is required from a query operation), as opposed to explicitly describing all the steps required to retrieve the required data from storage). Thus, SQL is high-level, quite easy to learn, allowing complex tasks on a database to be specified simply.

SQL has a defined syntax that specifies how standard keywords can be combined to form valid SQL statements. SQL statements can be divided into three categories, according to their function:

1. Data Manipulation: Statements that retrieve, insert, update, edit or delete data stored in database tables. These statements begin with the keywords: SELECT, INSERT, UPDATE or DELETE.

2. Data Definition: Statements that create, modify, or destroy database objects (such as tables). These statements begin with the keywords: CREATE, ALTER or DROP.

3. Data Control: Statements that authorise certain users to view, change or delete data. These statements begin with the keywords: GRANT or REVOKE. We do not consider data control in the lab exercises.

Using SELECT Statement:

The SELECT statement retrieves rows and columns from one or more tables and presents data in the result set i.e. output of SELECT statement in a grid.

1. Choose All Columns or Fields

The asterisk (*) sign is used for retrieving all columns from a table.

SYNTAX

```
SELECT * FROM table_name
```

Examples:

```
select * from bonus
select * from employee
select * from department
select * from salary
```

2. Choose Selected Columns or Fields

For selecting specific columns, we mention the columns by listing their names.

SYNTAX:

```
SELECT column1, column2, … FROM table_name
```

Examples:

```
1.  SELECT ename, job, sal FROM Employee
2.  SELECT dno, dname from department
```

3. Using the Column Alias with the AS Clause

```
SELECT DName AS 'Department Name' FROM Department
```

We have use AS clause to as the heading of 'Department Name' instead of Dname from the Department table as shown below.

4. Joining Constant Characters in the SQL Statement

Join column Emp_Name and Emp_Job column of Employee table having the same data type by this query:

```
SELECT EName + ' does the work of ' + Job
AS 'Employee Work' FROM Employee
```

This will work only for the columns of the same data type; otherwise SQL Server will generate an error as shown below:

```
SELECT ENo + ' is the employee number of ' + EName AS EMP FROM
Employee
```

5. Computing Values in SELECT Statements

To obtain an increase of 25 percent in the salary of employees in the employees table can be calculated like this:

```
SELECT ENo, EName, Job, Sal AS 'Previous Salary', Sal * 0.25 AS increment,
Sal + Sal * 0.25 AS 'Current Salary' FROM Employee
```

Single quote ‘ ‘ is used in the second AS-Clause because it contains space between two words, otherwise it is not necessary.

6. Using WHERE-Clause in SELECT statement

You can restrict the rows returned from the query by using the WHERE clause. A WHERE-clause has a condition that must be true and should directly follow from the FROM clause.

SYNTAX:

```
SELECT column_name FROM table_name WHERE condition(s)
```

Comparison operators are used in conditions that compare one expression to another. Following is a list of comparison operators:

OPERATOR	MEANING
=	Equal to
>	Greater than
>=	Greater than or equal to
<	Less than
<=	Less than or equal to
<>	Not equal to
!	Not

Following point must be kept in mind when using the WHERE clause:

- Character strings and dates are enclosed in single quotation marks.
- Character values are not case sensitive and date values are format sensitive.
- The default date format is MM/DD/YYYY

Other comparison operators used with the WHERE clause are:

OPERATOR	MEANING
BETWEEN .. AND …	Between two values
IN	Match any of a list of values
LIKE	Match a character pattern
IS NULL	Is a null value

7. Limiting Records By WHERE Clause

Using the WHERE clause, following query will retrieve details of all those employees whose Emp_head number is 125:

```
SELECT * FROM Employee WHERE Head = 125
```

Try these as well:

```
select * from employee where hiredate='1991-09-06'
                                              or
select * from employee where hiredate='9/6/1991'
```

8. Using DISTINCT keyword

Using DISTINCT keyword with our query, repeating records can be eliminated.

Type in the following query in the Editor Pane.

```
SELECT Head FROM Employee
```

Head column from the Employee table will be retrieved and as used here without DISTINCT keyword, all records will be displayed.

Now edit the same query with the DISTINCT keyword.

```
SELECT distinct Head FROM Employee
```

Repeating records will be eliminated.

9. Using TOP-Clause

The TOP-Clause limits the rows retrieved from the query result.

SYNTAX:

SELECT TOP (*n*)   column_name_list   FROM   table_name

10. Limiting Records by Using ‘TOP’ and ‘PERCENT’ In SELECT Statement

The following query will retrieve first five rows from the employee table using the TOP-clause:

```
SELECT TOP 5 * FROM Employee
```

Asterisk (*) sign again provides all the columns of the table, and column names can be provided to restrict the number of columns retrieved as:

```
    SELECT TOP 5 eno,ename FROM Employee
```

Similarly a percent of records can be retrieved from a table. For example

```
SELECT    TOP 60 PERCENT * FROM  Salary
```