

## Practical No. 1

Aim: Overview, features and functionality. Application Development in MS - Access.

(i) Creating a new database and tables for the Student application.

### • Opening MS Access

When you first start Access, you have the option of either opening an existing database or creating a new one. Access provides many ready-made databases for you to use and also several wizards to help you to quickly create a database.

### • Creating a new database

i) Select the Blank Database button and press OK. A window opens, asking you to select a folder and name for your database.

ii) Click the create button, the Student; Database window opens. Every Access database has a database window. The window has buttons for each database object: Table, Queries, Forms, Reports, Pages, Macros and Modules.

Table is the currently selected option.

iii) Select Create table in Design view option and click at New option.

iv) Select the Design view in the new window and click OK. The table design

window appears. Enter the field names and their data types in the table.

v) (Defining the Primary Key)

With the cursor in the row ROLL NO, right click the mouse button and menu appears.

### Saving the Table Structure:

vi) Save the table structure by pressing Save button or by selecting File > Save As from the menu bar. Enter the name of table as Student and click OK button.

vii) Click the Close item (X) in the top right corner to close the window. You will be returned to database window.

viii) In the database window, you will see that new table Student is now listed.

### (ii) Create a query involving one table.

i) In the database window, click Queries and select Create Query in Design View and then click New.

ii) Double-click Design View to open the Show Table dialog box.

iii) With the table selected, click Add and then Close. A query window appears.

### Selecting fields:

iv) The next step is to select the fields that you want to see in the query. We have

selected ROLLNO, NAME and MARKS fields.

The fields can be placed on the grid in a number of different ways:

- > Double click the field name.
  - > Drag the field name from the table onto the top line of the query grid.
  - > Click the down arrow in the field cell in the query grid to display a list of field names from which you can select.
- v) Double click ROLLNO, NAME and Marks in turn to place them in the query grid.
- vi) Click the Descending option in MARKS column to sort the student in descending order.

Running the Query :

- vii) Click the Run button on the toolbar or by clicking Query > Run option. The query result appears.

## Practical No. 2

Aim: Exercises on creation of Tables.

Sol. Tables are created using CREATE TABLE Command. This command is a part of DDL of SQL. This column names must be specified along the data types. Each table must have atleast one column. Tables are divided into rows and columns. Each row represents one piece of data and each column can be thought of representing a component of that piece of data. The syntax of CREATE TABLE command is

```
CREATE TABLE <Tablename> (<column name> <datatype> [<size>],<column name> <datatype> [<size>];<column name> <datatype> [<size>]....);
```

Inserting data into tables :

The Insert operation :

- Creates a new row (empty) in the database file.
- Loads the values passed (by the SQL Insert) into the column field.

The syntax is

```
INSERT INTO <Tablename>  
VALUES (<List of Values>);
```

The word VALUES must precede the list of data to be inserted.

INSERT INTO Emp

VALUES (10001, 'A. Williams', 'Production',  
'264 Staff Colony', 'Ahmedabad', 'M', 16046);

While inserting data into tables, following points should be taken care of

- i) Character data should be enclosed within quotes.
- ii) Column values for data types of column is provided with single quotes.
- iii) NULL values are given as NULL, without any quotes.
- iv) If no data is available for all the columns, then the column list must be included, following the table name.

To insert the employee Code, Name, Address, City, Sex and Salary, the following command is used.

INSERT INTO Emp (Ecode, Ename, Dept, Address,  
City, Sex, Salary)

VALUES (10003, 'Ishita Sharma', 'R&D',  
'44B Vikas Puri', 'New Delhi', 'F', 21565);

## Practical No. 3

Aim: Exercises on different forms of Select statement, altering and dropping of tables.

Sol:

### 1. SELECT Statement

A SELECT command retrieves information from the database. Using a SELECT command, you can do the following:

- Selection : You can use the selection capability in SQL to choose the rows in a table that you want returned by a query. You can use criteria to selectively restrict the rows that you see.

The syntax of SELECT statement is

```
SELECT [DISTINCT] {*, Column [alias], ... }  
FROM Table;
```

- SELECT identifies the columns.

- FROM identifies which Table.

A SELECT statement must include the following:

- A SELECT clause, which specifies the columns to be displayed.

• A FROM clause, which specifies the table containing the columns listed in the SELECT clause.

- Selecting All Columns -

When data from all rows and columns from the table are to be viewed then the syntax of SELECT

statement is

`SELECT * FROM <tablename>;`

where asterisk (\*) means all columns in table.

- Selecting Specific Columns -

You can use the SELECT statement to display specific columns of the table by specifying the column names, separated by commas.

e.g. `SELECT Ecode, Ename, Salary.`

- Column Heading Default -

Character column headings and its data as well as data column headings and its data are left justified within a column with width. Number headings and its data are right justified.

e.g. `SELECT Ename, Salary  
from Emp;`

- Using Arithmetic Operators

You may need to modify the way in which data is displayed and want to perform calculations. This can be done with arithmetic expressions. An arithmetic expression may contain column names, constant numeric and arithmetic operators like +, -, \*, /.

This is shown as below:

`SELECT Ename, Salary, Salary + 1000  
from Emp;`

- Operator Precedence

Multiplication and division have higher priority over addition and subtraction. If an arithmetic expression contains more than one

operator, multiplication and division are evaluated first.

You can use parentheses to force the expression within parentheses to be evaluated first.

e.g. `SELECT Ename, Salary, 12 * Salary + 500,  
 FROM Emp;`

#### • Defining a Column Alias

While displaying the result of a query, SQL normally uses the name of the selected column as the column headings. In many cases, this heading may not be descriptive and hence is difficult to understand. You can change a column heading by using a Column Alias.

`SELECT Ename AS Name, Dept as Department  
 from Emp;`

The result of the query displays the column headings in uppercase by default column headings appear in uppercase.

`SELECT Ename "Name",  
 Dept "Department"  
 from Emp;`

The above query would generate the same output but the headings will not be in the uppercase and will be in the form as shown within quotations.

## Practical No. 4

**Aim:** Exercise on insertion of data into tables.

**Sol:** New rows can be added to a table using the **INSERT** statement. Only one row is inserted at a time. The syntax is

```
INSERT INTO <Tablename>  
VALUES (<List-of-Values>);
```

### i) Inserting new rows

**Note:** Enclose character and date values within single quotation marks. Do not enclose numeric values within single quotation marks.

To insert limited values in the row, mention the field names also as shown below:

To insert only the employee number, name and salary, the following command is used

```
INSERT INTO Emp(Ecode, Ename, Salary)  
VALUES (10010, 'Danish Arona', 25750);
```

The

### ii) Inserting rows with NULL Values

You can insert null values in a row by two ways (a) Implicit method (b) Explicit method.

In the Implicit method, omit the column from the column list.

```
e.g., INSERT INTO Emp(Ecode, Ename, Salary)  
VALUES (10011, 'Kanav', 25400);
```

In the Explicit method, specify the NULL keyword.

e.g. `INSERT INTO Emp`

`VALUES (10012, 'Pallavi Kathpalia', null, null, null, null, null, null, null);`

Any column that is not listed explicitly obtains a null value in the new row.

### iii) Inserting Data Values

If a date must be entered in a format other than the default format e.g. another century or a specific time; you must use the TO\_DATE function.

e.g. `INSERT INTO Emp`

`VALUES (10013, 'Gaurav Kathpalia', 'Production', '555 Model Town', 'Karnal', 'M', 25650, TO_DATE ('Feb 14, 2001', 'MON DD, YYYY'), 1500);`

### iv) Use of Substitution Variables to Insert Values

You can use substitution variables to insert values in a table. The '&' symbol is used as an substitution operator. When a substitution operator is used, SQL prompts for the value of the variable, accepts it and then substitutes it in place of variable.

## Practical No. 5

Aim: Exercise on deletion of data using different conditions.

You can remove existing rows from a table by using the DELETE command. The entire row is deleted from the table and specific columns cannot be deleted from the table.

A set of rows can also be deleted from the table by specifying the condition.

- Specific rows are deleted when you specify the WHERE clause.

- All rows in the table are deleted if you omit the WHERE clause.

The syntax is

```
DELETE      FROM <tablename>
[WHERE      <Condition>];
```

In the syntax

'tablename' is the name of table

'condition' identifies the rows to be deleted and is composed of column names, expressions, constants, subqueries and comparison operators.

\* Delete all the records of RND Department

```
DELETE      FROM Emp
WHERE      Dept = 'RND';
```

\* Delete all records of employees having salary less than 15000.

```
DELETE      FROM Emp
```

WHERE DOJIN > TO\_DATE ('30.06.2002',  
'DD.MM.YYYY');

\* Deleting rows based on another table:

You can use subqueries to rows from a table based on values from another table.

The following query search from Emp table to find the Ecode corresponding to A. Williams and use it in INCR table, which deletes rows of data based on that Ecode.

```
DELETE FROM INCR  
WHERE Ecode =  
(SELECT Ecode  
FROM Emp  
WHERE Ename = 'A. Williams');
```

## Practical No. 6

Aim: Exercises on UPDATE Statement.

Columns in table are updated using the UPDATE command. Values of a single column or group of columns can be updated. Updating can be carried out for all the rows in a table or selected rows. Expression can also be used with UPDATE. The syntax is

UPDATE                    <Tablename>  
SET                        <Colname> = <Value> [, Colname =  
                            Value, ...]

[WHERE Condition];

In the above syntax

Tablename : is the name of table

Colname : is the name of column in the table

Value : is the corresponding value

Condition : identifies the rows to be updated and is composed of column names, expressions, constant, subqueries and comparison operators.

\* Increase every employees salary by 10%

UPDATE                    Emp

SET                        Salary = Salary + (Salary \* 0.1);

\* Change the department of Ishita Sharma  
to Servicing

```
UPDATE Emp
SET Dept = 'Servicing'
WHERE Ename = 'Ishita Sharma';
```

### (i) Updating rows in a table

The UPDATE Command modifies specific rows, if the WHERE clause is included. All rows in the table are modified if you omit WHERE clause :

```
UPDATE Emp
SET Salary = 35700
WHERE Ename = 'Suresh Baroth';
```

The salary of only Suresh Baroth will be updated i.e. that specific row is updated. If you omit WHERE clause from above query, Salary of all the rows will be updated.

### Guidelines :

- Columns in a table are updated using the UPDATE command.
- Values of a single column or group of columns can be updated.
- Updating can be carried out for all the rows in a table or selected rows.
- Expression can also be used with update.

- Subqueries can also be used with UPDATE command.

(ii) Updating rows based on another table

You can use subqueries in UPDATE statements to update rows in a table based on values from another table.

The following query updates the Incr table based on the values from the Emp table. It changes the Ecode of employees with 10007 code and Date of increment of employees with Date of joining of Employees with Ecode 1000P -

UPDATE Incr

SET

Ecode = (SELECT Ecode  
FROM Emp  
WHERE Ecode = 10007)

WHERE DateIncr = (SELECT Dojoin

FROM Emp

WHERE Ecode = 10007);