APPENDIX

Description of Commonly Used Functions in Access

There are three types of functions that are used to set the Control Source property of calculated controls and/or to form part of calculated field expression in SQL statement. A brief description of the commonly used functions is below :

A-1. Domain Aggregate Functions

These functions are used to perform calculations based on values in a field of a table or query. Criteria to select the set of records in the table or query that is desired to be used for calculations may also be specified. The criteria, if not specified, imply that all the records of the table or query specific to the field are used for computation. All the domain aggregate functions use the same syntax as is given hereunder :

DFunction ("FldName", "TblName" or "QryName", "SrchCond")

Wherein DFunction refers to a named domain aggregate function. A brief description of its input arguments is given below:

FldName : It refers to the name of field that is to be searched in a table or query, which is specified as an argument.

TblName (or QueryName) : It refers to the name of a table or query that contains the field specified as second input argument.

SrchCond : It refers to the search condition on the basis of which the relevant record is searched.

Some of the important domain aggregate functions have been described as below :

(a) DLookup : This function is meant to look up information that is stored in a table or query, which is not the underlying source of Access Form or Report. It is used to set the Control Source property of a calculated control to display data from other table or query. Consider the following example:

DLookup ("Name", "Accounts", "Code = '110001'")

In the above example, this function has been applied to search the name of account (in Accounts table) whose code is '110001'.

(b) DMax and DMin : These functions are used to retrieve respectively the maximum and minimum values in the specified field. Consider the following example :

DMin ("Amount", "Vouchers", "Debit = '711001'")

Dmax ("Amount", "Vouchers", "Debit = '711001'")

In the above examples, the amount of minimum purchase transaction and maximum purchase transaction is retrieved and reported. It may also be noted that '711001' is the code of Purchase account in Accounts table

(c) DSum : This function computes and returns the sum of the values in the specified field or expression. For Example, in a table : ***Sales*** that contains

© NCERT not to be republished

ItemCode, Price and Quantity as fields, the total amount of *sales* may be computed by using the DSum () function as follows :
DSum ("Price*Quantity", "Sales")
However, if the total sales is to computed for a particular item coded as 1678, the DSum () function shall be applied as follows :
DSum ("Price*Quantity", "Sales", "ItemCode = 1678")

(d) DFirst and DLast : These functions are used to retrieve respectively the values in the specified field from first and last physical records.
Consider the following application examples :
DFirst ("Name", "Accounts")
DLast ("Name", "Accounts")
In the above examples, the name first and last account that physically exists in Accounts table is retrieved and reported.

(e) DCount : This function is meant to compute the number of records with non-null values in the specified field. Consider the following application example :
DCount ("*", "Accounts")
In the above example, The number of records in accounts table are counted and reported by DCount () function.

A-2. SQL Aggregate Functions

The SQL aggregate functions have the functionality similar to that of domain aggregate function. However, unlike domain aggregate functions, these functions cannot be called directly into controls used in Forms and Reports of Access. These functions are used in SQL statements that provide the underlying record source of Forms and Reports. All these functions, when used require the GROUP BY clause in SQL statement :

(a) Sum : This function is used to compute and return the sum of a set of values. For Example, consider the following SQL statement that has been used in Chapter-V to prepare the underlying information source of Trial Balance (Model-I.).

```
SELECT Debit As Code, SUM (Amount) AS Total
FROM VOUCHERS
GROUP BY Debit ;
```

In the above SQL statement, Sum () has been used to compute the total amount by which the transacted accounts have beeen debited.

(b) Min and Max : These functions are used to retrieve respectively the minimum and maximum of value set with respect to field or query expression. For Example, the following SQL statement is capable of returning the amount of minimum and maximum sales transaction in Model-I :

```
SELECT Min (Amount) As MinSales, Max (Amount) As MaxSales
FROM Vouchers
WHERE Credit = '811001' ;
```

© NCERT not to be republished

It may be noted that the sales account that is coded as '811001' is credited as and when a sales transaction is recorded.

(c) Count : This function counts the number of records returned by a query. The number of times a sales transaction has occurred and recorded in books of accounts can be known by executing the following SQL statement.

SQL statement.
SELECT count (*)
FROM Vouchers
WHERE Credit = '811001'

In the above SQL statement, the Credit field stores the account code of sales when a sales transaction occurs. The WHERE clause restricts the number of records returned by the above SQL to those in which credit field has the account code of sales. Accordingly, the count () function returns the count value of records returned by the above SQL statement.

(d) First and Last : These functions are meant to retrieve the first and last record of a value set pertaining to a field or query expression.

A-3. Other Functions

(a) IIF : The purpose of this function is to provide a value to the field from a mutually exclusive set of values. Its syntax is as given below :
IIIF (<Condition>, Value-1, Value-2)
Wherein <Condition> refers to any logical expression in which a comparison is made by using following comparison operators :
= equal to
<less than
>greater than
<= less than or equal to
>= greater than or equal to
The condition formed by the above comparison operators is evaluated to result into TRUE or FALSE.
<Value-1> This value is returned by IIF() function to the field, if the condition turns out to be TRUE
<Value-2> This value is returned by IIF() function to the field, if the condition turns out to be FALSE
Example : Suppose a field Type is to return the string of characters "Debit" when its value is 0 and "Credit" when its value is 1, IIF() function is used as shown below :
IIF (Type = 0, "Debit", "Credit")

(b) Abs : The purpose of this function is to return absolute value, This function receives a numeric value as its input argument and returns an absolute value. Consider the following examples on use of Abs ( ) function :
When – 84 is given as input argument to Abs(– 84), it returns 84

© NCERT not to be republished

When 84 is given as input argument to Abs(84), it returns 84

(c) Val : The purpose of this function is to return the numbers contained in a string as a numeric value of appropriate type. Its Syntax is **Val**(string)
The string argument of the above Val( ) function is any valid string expression. The Val( ) function stops reading the string at the first character that cannot be recognised as number. For example, Val("12431") returns the value 12431 by converting the enclosed string of numerals into value. However, Val ("12,431") returns the numeric value 12 because comma after 12 in the enclosed string of characters in Val ( ) function is not recognised as number.

© NCERT not to be republished

NOTE

© NCERT not to be republished