Module/lab Session-2

The Select statement:

The syntax of the select statement with most of the options is given below

Select [All/Distinct] scalar-expression(s)

For table name

[Where conditional- expression]

[Group by column (s)]

[Having conditional-expression]

[Order by column(s)];

In the above syntax, only the select statement and from clause are required. All the other four clauses-

Where, group by, having and order by-are options

* You include them in the select statement, only when you required the functions they provides.
* The from clause specifies that table or tables from where the data has to be retrieved
* The where clauses tell SQL to include only certain rows of data in the result set
* It is in the where clause that you specify the search criteria. For example, you might code something like where price >20.
* The group by clause specifies a summary query. These are usually used with aggregate function like sum, avg, max, min, etc.
* The having clause tells SQL to include only certain groups produced by the group by clause in the query result.
* The Order by clause sorts or orders the results based on the data in one or more columns in the ascending or descending order.

If noting is specified, the result set will be sorted in ascending order,

Example: order by price desc.

[In this session we will use a table called book for example]

Create table book with the following contains like

ISBN, Title, Author Name, Publisher Name, Date of publication, Price. [Data to be entered using

Query: Get the title, author name and publisher of all books published in 1997 and the price is greater than $25]

Sql\_command:

select title, author, publisher from book where year =1997 and price >25;

<Screen short the output will be placed here>

You can use all the comparison operator’s =, < >, <=, > and > = in the where clause

< > Is not equal in standard SQL, in SQL/DS it is written as ‘~=’ and in MS-SQL server it is written as ‘!=’. We can use AND, OR, NOT in where clause.

Eliminating Duplicates: select using Distinct

When you give a select statement, the RDBMS does not eliminate the duplicates.

Select publisher from book; // lets see the result

Select distinct publisher from book; // lets see the result and compare.

Select using IN: “if we want to get the rows which contain certain values, the best way to do it is to use **IN** conditional expression.

Query: You want to get the title, author and publisher name of all the books. Published in year 1993,1996 and 1998.

Sql\_command:

Select title, author, publisher, year

From book

Where year in (1993,1996,1998);

Or

Select title, author, publisher, year

From book

Where year = 1993

Or year = 1996

Or year = 1998;

Try using NOT IN. Query your self and see the result.

Select using between: Between can be used to get those items that fall within a range

Query: Get all the details of the books whose price is in the range of 10 and 20 inclusive.

Sql\_Command: [without using between]

select title, author, publisher, price

from book

where price >= 10 and price <= 25;

Sql\_Command: [Using between]

select title, author, publisher, price

from book

where price between 10 and 25;

Try using NOT BETWEEN

select title, author, publisher, price

from book

where price NOT BETWEEN 10 and 60;

Computed Values:

In Sql, statement can be used for retrieving the computed values without any problem.

Consider the following query.

For all books get the price in Indian rupees (the price given is in Dollar)

1 dollar = 68 rupee.

The SQL, statements achieves

select title, author, price\*40 as price\_rp from book where price >40;

Ordering while selecting:

For getting the results in a particular order we use the ORDER BY clause.

Query: “get all the distributor details in descending order of discount”

Sql\_Command: select title, author, price

from book

order by price desc;

Try the following Sql\_Command

select title, author, price

from book

where price>40

order by 3;

What did you observer? [Let discuss]

One important thing to remember here is that the column(s) on which the order by is specified, should be a part of the result table. It is possible to identify columns in the order by clause by column number instead of column name, that is by the ordering left-to-right.

Insert, Update and Delete Operations

Update:

update table name

set column =scalar expression

[, column = scalar expression] …

[where condition]

1. Change the price of the book ID = 21 to 60.

update book

set price =60

where ID =’21’;

2. Increase the price of all books, which are published before 1997 by 20%

update book

set price = price \* 1.2

where year < 1997;

3. Change the publisher name for all ‘Microsoft press’ to ‘MP’ and increase the price by 10%

update book

set publisher = ‘MP’, price = price \*1.10 where publisher = ‘Microsoft press’;

Delete statement: The delete statement is used to delete an already existing row or rows from a table.

Delete from table name

[where conditional-expression]

All rows in the table, which satisfies the condition will be deleted. If the where clause is omitted all row’s will be deleted.

1. Delete all the books published by McGraw-Hill

delete from book

where publisher = ‘McGraw-Hill’;

2. Try without where condition and share the output.