Database Systems Lab - 14CS2012

REGISTER NO : UR14CS228
DATE : 22-08-16
EXPERIMENT NO : 3

Video URL : https://youtu.be/nqQuIvhjIFY

AIM:

To use Aggregate and Built – in Functions, Group by and Order by functions in an database table.

DESCRIPTION:

GROUP BY: Optional section of SELECT statement used to group data based on distinct values of
specified columns.Creates a data set, containing several groups based on a condition.

ORDER BY: Used with select statement only. to view data from a table in sorted order.Rows
retrieved from the table can be sorted in either ascending or descending order.Sorting is based on
columns specified in select statement.Default sort order is ascending order

AGGREGATE FUNCTION: Functions that take a collection / set of values & returns a single value.
Used as expressions in the select statement to return summary data.

SYNTAX:

1. Display the number of customers.

```
SELECT COUNT(customer_id) FROM Customer
```

COUNT(customer_id)
7

2.What is the maximum quantity that has been ordered?

```
SELECT MAX(ordered_quantity) FROM Orderline
```

MAX(ordered_quantity)
40100

3.Display the minimum, maximum and average prices of the product.

SELECT MAX(price) AS max, min(price) AS min, avg(price) AS average FROM Product

max	min	average
50	10	27.7778

4.Display the average price of each product.

SELECT avg(price),product_description FROM Product GROUP BY product_description

avg(price)	product_description
30.0000	Book Shelf
30.0000	Duplex Boo
40.0000	Duplex Tab
50.0000	Keyboard
15.0000	Manager's
20.0000	Office Cha
10.0000	Office Des
40.0000	Table Lamp

5.Find the total quantity ordered for each product.

SELECT sum(ordered_quantity), product_id FROM Orderline GROUP BY product_id

sum(ordered_quantity)	product_id
56560	2000
20500	3000
30005	3001
10000	4000
41100	4001

6.Find out the no. of orders placed on each date.

```
          SELECT Orders.order_date, sum(Orderline.ordered_quantity) FROM
Orders,Orderline WHERE Orders.order_id=Orderline.order_id GROUP BY
Orders.order_date
```

order_date	sum(Orderquantity.ordered_quantity)
01-OCT-14	30000
02-OCT-14	34560
03-OCT-14	1000
10-OCT-14	92605

7.Find out how many products are ordered in each order.

```
          SELECT order_id, sum(ordered_quantity) FROM Orderline GROUP BY order_id
```

order_id	sum(ordered_quantity)
100	10000
101	20000
102	34560
103	1000
104	2000
105	30005
106	20500
107	40100

8.Sort the order table based on the order date.

```
          SELECT * FROM Orders GROUP BY order_date
```

order_id	order_date	customer_id
100	01-OCT-14	1
102	02-OCT-14	3
103	03-OCT-14	2
104	10-OCT-14	1

9.Display the products and their quantity and sort the result on the sum of quantity ordered in descending order

```
          SELECT product_description, sum(product_quantity)As quantity FROM Product
GROUP BY product_description ORDER BY quantity DESC;
```

product_description	quantity
Table Lamp	50
Duplex Tab	40
Manager's	30
Keyboard	20
Office Cha	20
Book Shelf	20
Duplex Boo	10
Office Des	10

10. Display the supplier details sorted by their ratings in descending order.

```
SELECT supplier_details FROM Suppliers ORDER BY supplier_rating DESC;
```

Wood supplier
Duplex Supplier
Human resource
Keyboard Supplier
Office item Supplier

11. Display the names of all customers in uppercase.

```
SELECT UPPER(customer_name) FROM Customer
```

UPPER(customer_name)
RICHARD JOHNS
MARY JOHNS
JOSEPH JOHNS
CATHY COOK
RICHARD NEWMAN
MARY SMITH
JOHN DOE

12. Find the number of months between ordered date of order_id=103 and current date

```
SELECT TIMESTAMPDIFF(MONTH,order_date,curdate()) FROM Orders WHERE order_id ='103'
```

TIMESTAMPDIFF(MONTH,order_date,curdate())
8

13. Prefix the postal code with two zeroes

UPDATE Customer SET pstal_code=concat('00',pstal_code)

pstal_code
0010059
0010039
0010056
0085719
0092010
0094032
0010059

14. Display the substring “Newman” from “Richard Newman”.

SELECT SUBSTRING_INDEX(customer_name," ",-1) FROM Customer WHERE customer_name='Richard johns'

SUBSTRING_INDEX(customer_name," ",-1)
Johns

15. Display the number of characters in the name of customers which ends with ‘e’.

SELECT customer_name, char_length(customer_name) AS length FROM Customer WHERE customer_name LIKE '%e'

customer_name	length
John Doe	8

Result:

SQL queries using DDL and DCL commands are successfully executed.