

**Information System Management Lab
BCOM 307**

Assignment #28

Submitted by:

Name: YASH JAIN
Enrollment No: 03914788818
Semester: B.Com(H) 5TH Semester
Class: B.COM(H)
Section: B.Com 5A
Date of Submission: 14/11/2021

Submitted to:

Praveen Kumar Singh
Assistant Professor, MAIMS



Department of Commerce
Maharaja Agrasen Institute of Management Studies
Affiliated to Guru Gobind Singh Indraprastha University, Delhi
Sector -22, Rohini, Delhi -110086, India; www.maims.ac.in



Maharaja Agrasen Institute of Management Studies

Affiliated to GGS IP University; Recognized u/s 2(f) of UGC

Recognized by Bar Council of India; ISO 9001: 2015

Certified Institution Sector 22, Rohini, Delhi -110086, India;

www.maims.ac.in

Department of Commerce

Academic Year: 2020-21

Semester: Vth

Assignment No. 28

Unit No:

Course/Subject Code: BCOM 307

Issue Date

Subject Title: Information System Management Lab

Last Date of Submission:

Instructions for Students:

1. **All Questions are Compulsory.**
2. The student should attach proper cover page for each assignment clearly mentioning the Assignment No.
3. Each assignment should be prepared by the student individually with proper explanation and screenshots.
4. A4 size ruled sheets should be used for the assignment.
5. Assignment pages should be serially numbered at the bottom of page.

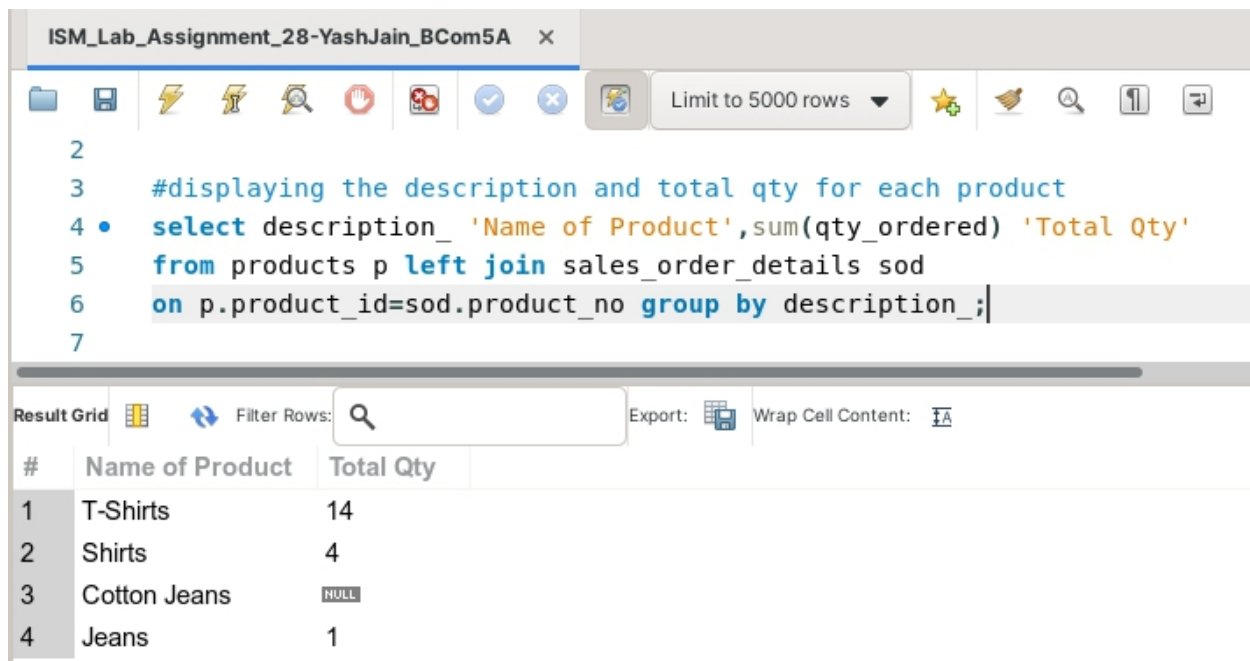
During online education mode, upload scanned copy of the complete assignment including cover page latest by due date.

Question No.	Question	CO No.
1	Print the description and total quantity sold for each product.	CO2, CO3, CO4, CO5
2	Find the non-moving products , i.e. products not sold.	
3	Find the name and complete address for the customer who have placed order no. O19001.	
4	Add a new column product rate in the table sales_order and update the values.	
5	Find the names of clients who have placed order worth rs. 1000 or more.	

ASSIGNMENT 28 - REVISION OF CONCEPTS II

Task 1 : Print the description and total quantity sold for each product.

This task can be completed using the **LEFT JOIN** and the **GROUP BY** Clause.



The screenshot shows a SQL IDE window titled "ISM_Lab_Assignment_28-YashJain_BCom5A". The query editor contains the following SQL code:

```

2
3  #displaying the description and total qty for each product
4 • select description_ 'Name of Product',sum(qty_ordered) 'Total Qty'
5    from products p left join sales_order_details sod
6    on p.product_id=sod.product_no group by description_;
7

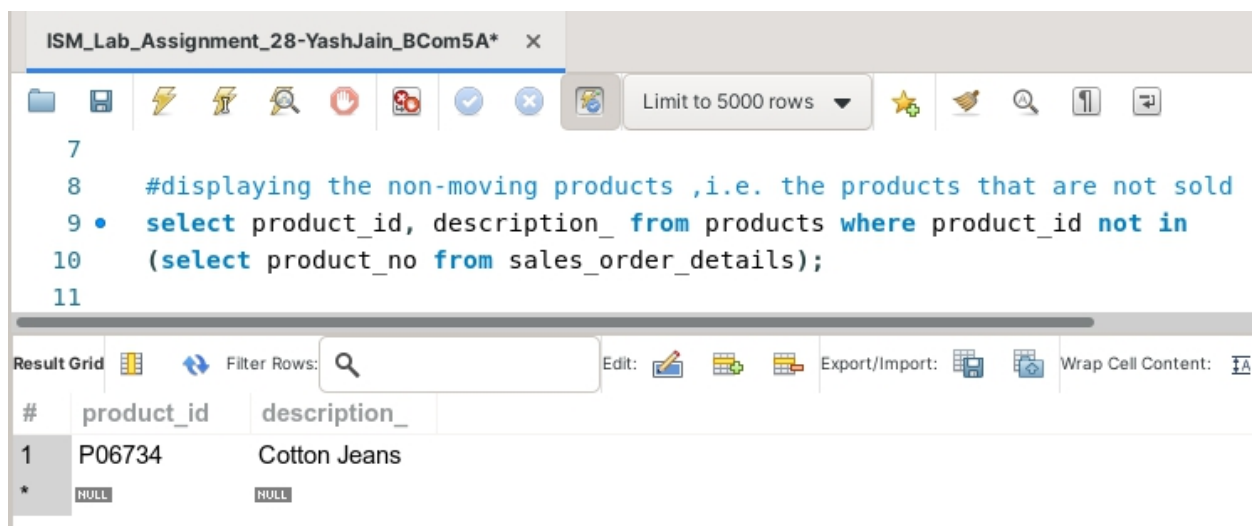
```

Below the query editor is the "Result Grid" showing the output of the query:

#	Name of Product	Total Qty
1	T-Shirts	14
2	Shirts	4
3	Cotton Jeans	NULL
4	Jeans	1

Task 2: Find the non-moving products , i.e. products not sold.

This task can be completed using the **NOT IN** Predicate, alongside **SQL Subquery**.



The screenshot shows a SQL IDE window titled "ISM_Lab_Assignment_28-YashJain_BCom5A*". The query editor contains the following SQL code:

```

7
8  #displaying the non-moving products ,i.e. the products that are not sold
9 • select product_id, description_ from products where product_id not in
10    (select product_no from sales_order_details);
11

```

Below the query editor is the "Result Grid" showing the output of the query:

#	product_id	description_
1	P06734	Cotton Jeans
*	NULL	NULL

Task 3: Find the name and complete address for the customer who have placed order no. O19001.

This task can be completed using the **INNER JOIN** Clause, along with the **CONCAT()** Function, which combines three columns of the table. The syntax for this is :

```
CONCAT ( string 1, separator, string 2, separator.....);
```

The screenshot shows a database IDE window titled "ISM_Lab_Assignment_28-YashJain_BCom5A". The SQL editor contains the following code:

```
11
12 #displaying client details for clients with order no 019001
13 • select so.order_no,client_id,client_name, concat(city,', ',state,' - ',pincode) 'Address'
14   from clients c inner join sales_order so on c.client_id=so.client_no where so.order_no='019001';
15
```

The "Result Grid" at the bottom shows the output of the query:

#	order_no	client_id	client_name	Address
1	O19001	C0001	Ivan Bayross	Bangalore, Maharashtra - 400054

Task 4: Add a new column product rate in the table sales_order and update the values.

This task can be completed using the **ALTER TABLE** and the **UPDATE** Commands.

The screenshot shows the same database IDE window. The SQL editor contains the following code:

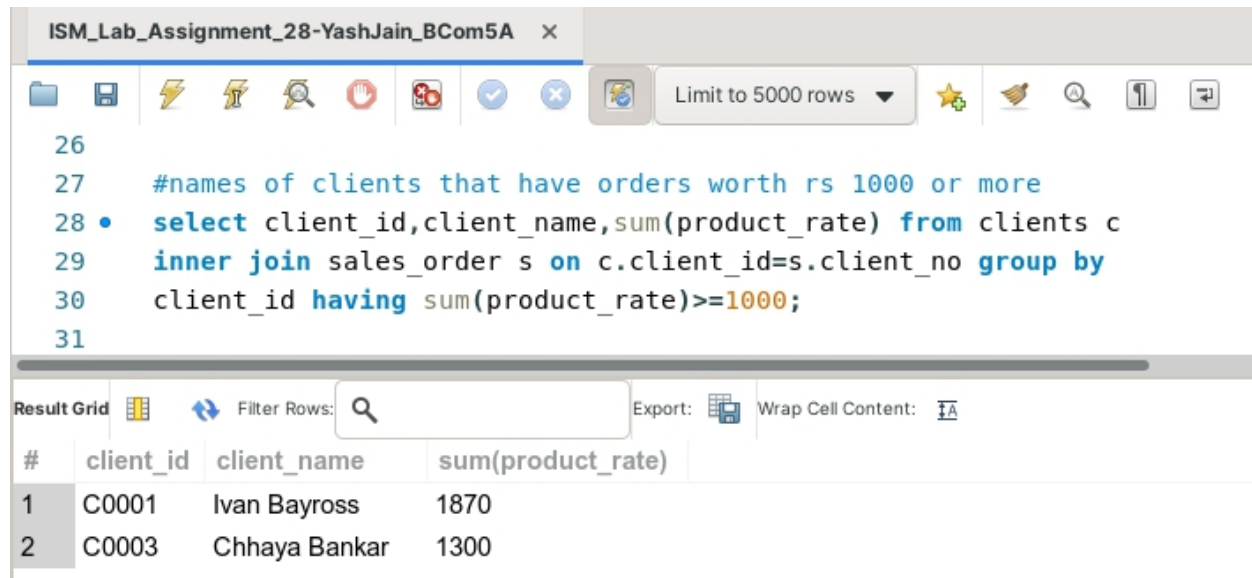
```
15
16 #adding new column product_rate
17 • alter table sales_order add column product_rate int default 100;
18
19 #updating values
20 • update sales_order set product_rate=1500 where order_no='019001';
21 • update sales_order set product_rate=200 where order_no='019002';
22 • update sales_order set product_rate=370 where order_no='019003';
23 • update sales_order set product_rate=1300 where order_no='046865';
24
25 • select order_no,product_rate from sales_order;
```

The "Result Grid" at the bottom shows the output of the final SELECT query:

#	order_no	product_rate
1	O19001	1500
2	O19002	200
3	O19003	370
4	O46865	1300
*	NULL	NULL

Task 5: Find the names of clients who have placed order worth rs. 1000 or more.

This task can be completed using the **INNER JOIN** Clause, along with the **GROUP BY** and the **HAVING** Clause.



The screenshot shows a database query editor window titled "ISM_Lab_Assignment_28-YashJain_BCom5A". The query is as follows:

```
26  
27 #names of clients that have orders worth rs 1000 or more  
28 • select client_id,client_name,sum(product_rate) from clients c  
29 inner join sales_order s on c.client_id=s.client_no group by  
30 client_id having sum(product_rate)>=1000;  
31
```

Below the query editor, the "Result Grid" displays the results of the query. It includes a search bar, an "Export" button, and a "Wrap Cell Content" checkbox. The results are shown in a table with 4 columns: #, client_id, client_name, and sum(product_rate).

#	client_id	client_name	sum(product_rate)
1	C0001	Ivan Bayross	1870
2	C0003	Chhaya Bankar	1300