SE-COMP(B)	Roll number :
Experiment no. : 4	Date of Implementation :
Aim : To implement simple SQL commands	
Tool Used : PostgreSQL/MySql	
Related Course outcome : At the end of the course, Students will be able to Use SQL : Standard language of relational database	

Rubrics for assessment of Experiment:

Indicator	Poor	Average	Good
Timeliness • Maintains assignment deadline (3)	Assignment not done (0)	One or More than One week late (1-2)	Maintains deadline (3)
Completeness and neatness • Complete all parts of QUERY assignment(3)	N/A	< 80% complete (1-2)	100% complete (3)
Originality • Extent of plagiarism(2)	Copied it from someone else(0)	At least few questions have been done without copying(1)	Assignment has been solved completely without copying (2)
Knowledge • In depth knowledge of the QUERY assignment(2)	Unable to answer 2 questions(0)	Unable to answer 1 question (1)	Able to answer 2 questions (2)

Assessment Marks :

Timeliness	
Completeness and neatness	
Originality	
Knowledge	
Total	

Total : (Out of 10)

Teacher's Sign :

EXPERIMENT 4	Basic SQL Commands
Aim	To implement simple SQL commands
Tools	PostgreSQL/MySql
Theory	**SELECT:** SELECT statement returns a result set of records from one or more tables. The select statement has optional clauses: • WHERE specifies which rows to retrieve • GROUP BY groups rows sharing a property so that an aggregate function can be applied to each group having group. • HAVING selects among the groups defined by the GROUP BY clause. • ORDER BY specifies an order in which to return the rows. Syntax: SELECT<attribute list> FROM<table list> WHERE<condition> Where • Attribute list is a list of attribute name whose values to be retrieved by the query. • Table list is a list of table name required to process query. • Condition is a Boolean expression that identifies the tuples to be retrieved by query. **SQL Aggregate Functions** SQL aggregate functions return a single value, calculated from values in a column. Useful aggregate functions: • AVG() - Returns the average value • COUNT() - Returns the number of rows • FIRST() - Returns the first value • LAST() - Returns the last value • MAX() - Returns the largest value • MIN() - Returns the smallest value • SUM() - Returns the sum **The SQL ORDER BY Keyword** The ORDER BY keyword is used to sort the result-set by one or more columns. The ORDER BY keyword sorts the records in ascending order by default. To sort the records in a descending order, you can use the DESC keyword. **SQL ORDER BY Syntax** SELECT column_name, column_name FROM table_name ORDER BY column_name ASC|DESC, column_name ASC|DESC;

Procedure

TASK 1:

1. Create following table:

Table name : **sales_order**

Column Name	Data type	Size	Constraint
order_no	varchar	6	Primary Key
Order_date	date		NOT NULL
Client_no	varchar	6	NOT NULL
Dely_addr	varchar	25	
Salesman_no	varchar	6	
Dely_type	char	1	
Billed_yn	char	1	
Dely_date	Date		
Order_status	varchar	10	

2. Insert 5-6 records in table.
3. Find the names of all clients having ‘a’ as the second letter in their names.
4. Find out the clients who stay in a city whose second letter is ‘a’
5. Find the list of all clients who stay in ‘mumbai’ ordered by their names
6. Print the list of clients whose bal_due is greater than value 10000
7. Print the information from sales_order table for orders placed in the month of January
8. Display the order information for client_no C001 and C002
9. Find the products whose selling price is greater than 2000 and less than or equal to 5000
10. Find the products whose selling price is more than 1500. Calculate new selling price as original selling price * 1.5. Rename the new column in the above query as new_price
11. Count the total number of orders
12. Calculate the average price of all the product
13. Determine minimum and maximum product prices
14. count the number of products having price greater than or equal to 1500
15. Display the order number and day on which clients placed their order
16. Display the order_date in the format ‘dd-month-yy’
17. Display the month (in alphabets) and date when the order must be delivered
18. Find the date, 15 days after today’s date
19. Find the no. of days elapsed between today’s date and the delivery date of orders placed by the clients.

Task2: Use select with where statement with SQL aggregate functions for the tables created in Expt. no. 3

Post Lab Questions:

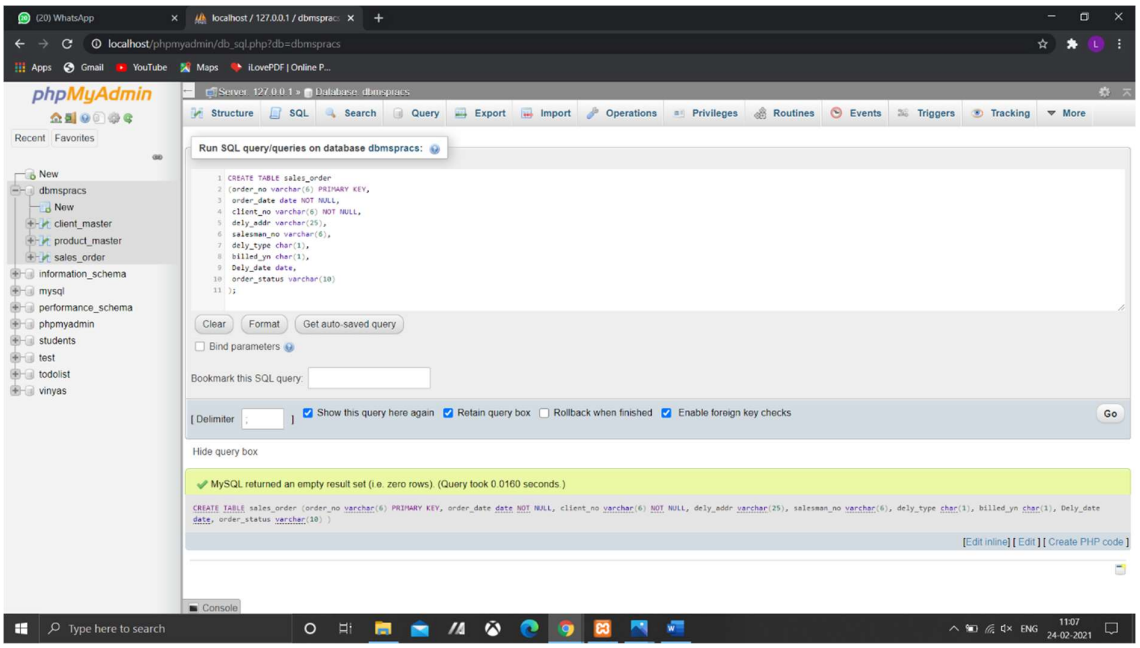
1. Write a short note on DBA.
2. Write different date functions and date formats.
3. Differentiate between group by and having with example.
4. Give different string functions.

Name: Brendan Lucas, Div: SE COMP B, Roll No: 8953

DBMS Practical Implementation, Lab 4

To Create Table:

```
CREATE TABLE sales_order
(order_no varchar(6) PRIMARY KEY,
 order_date date NOT NULL,
 client_no varchar(6) NOT NULL,
 dely_addr varchar(25),
 salesman_no varchar(6),
 dely_type char(1),
 billed_yn char(1),
 Dely_date date,
 order_status varchar(10)
);
```



Inserting Values

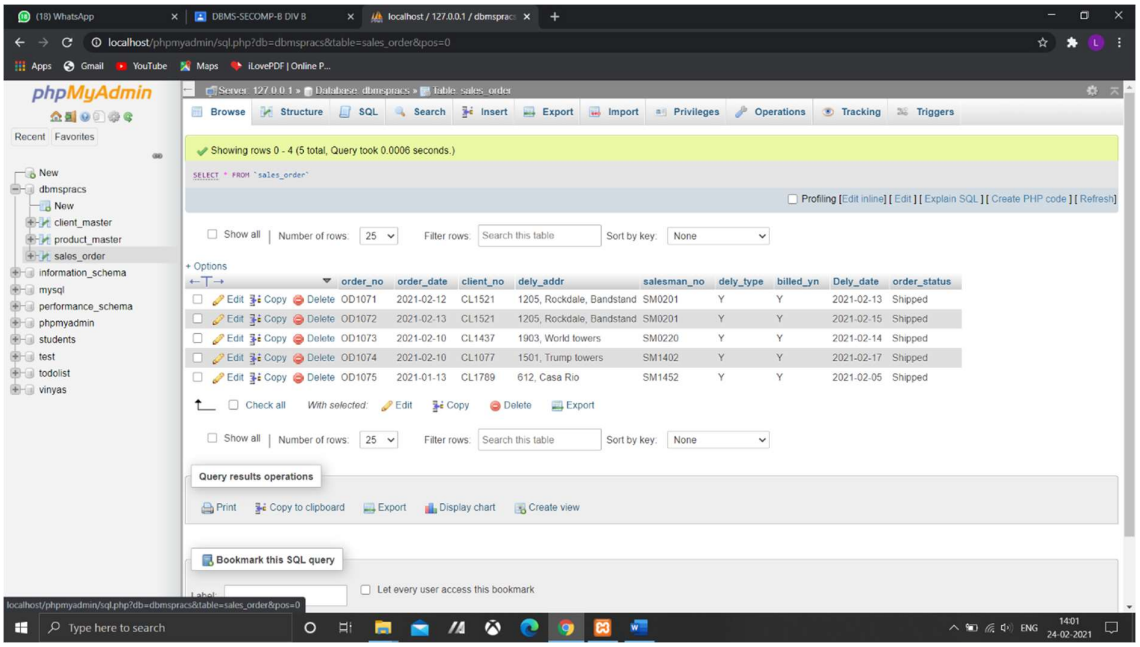
INSERT INTO `sales_order` VALUES ('OD1071', '2021-02-12', 'CL1521', '1205, Rockdale, Bandstand', 'SM0201', 'Y', 'Y', '2021-02-13', 'Shipped');

INSERT INTO `sales_order` VALUES ('OD1072', '2021-02-13', 'CL1521', '1205, Rockdale, Bandstand', 'SM0201', 'Y', 'Y', '2021-02-15', 'Shipped');

INSERT INTO `sales_order` VALUES ('OD1073', '2021-02-10', 'CL1437', '1903, World towers', 'SM0220', 'Y', 'Y', '2021-02-14', 'Shipped');

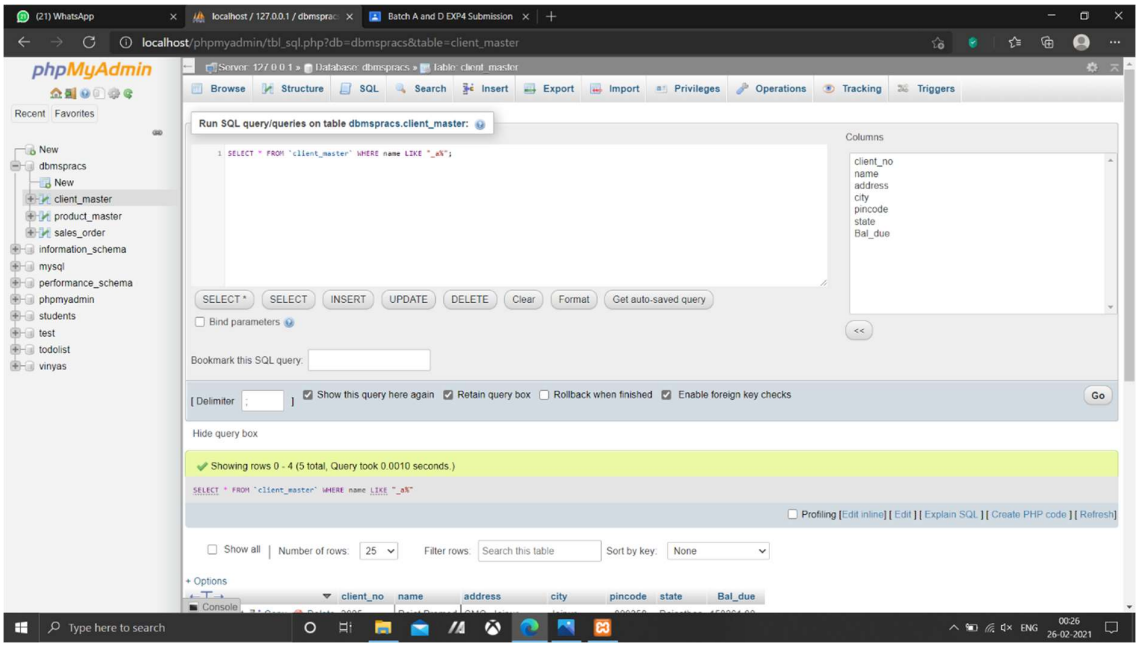
INSERT INTO `sales_order` VALUES ('OD1074', '2021-02-10', 'CL1077', '1501, Trump towers', 'SM14022', 'Y', 'Y', '2021-02-17', 'Shipped);

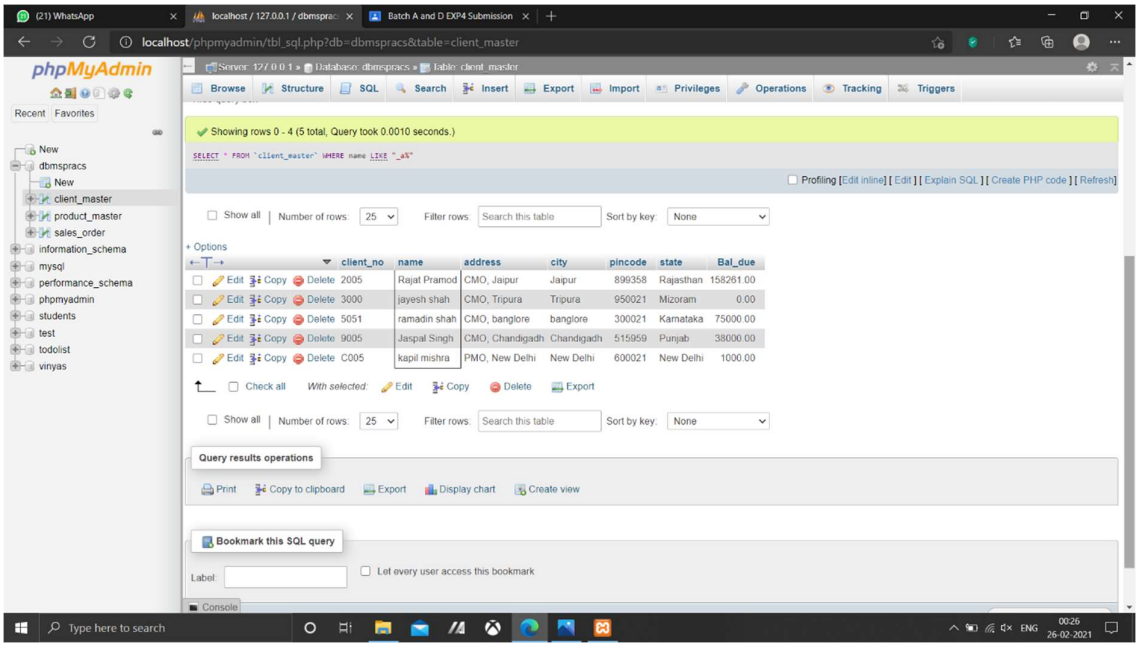
INSERT INTO `sales_order` VALUES ('OD1075', '2021-01-13', 'CL1789', '612, Casa Rio', 'SM1452', 'Y', 'Y', '2021-02-05', 'Shipped');



3. Find the names of all clients having ‘a’ as the second letter in their names.

Query:- SELECT * FROM `client_master` WHERE name LIKE "_a%";

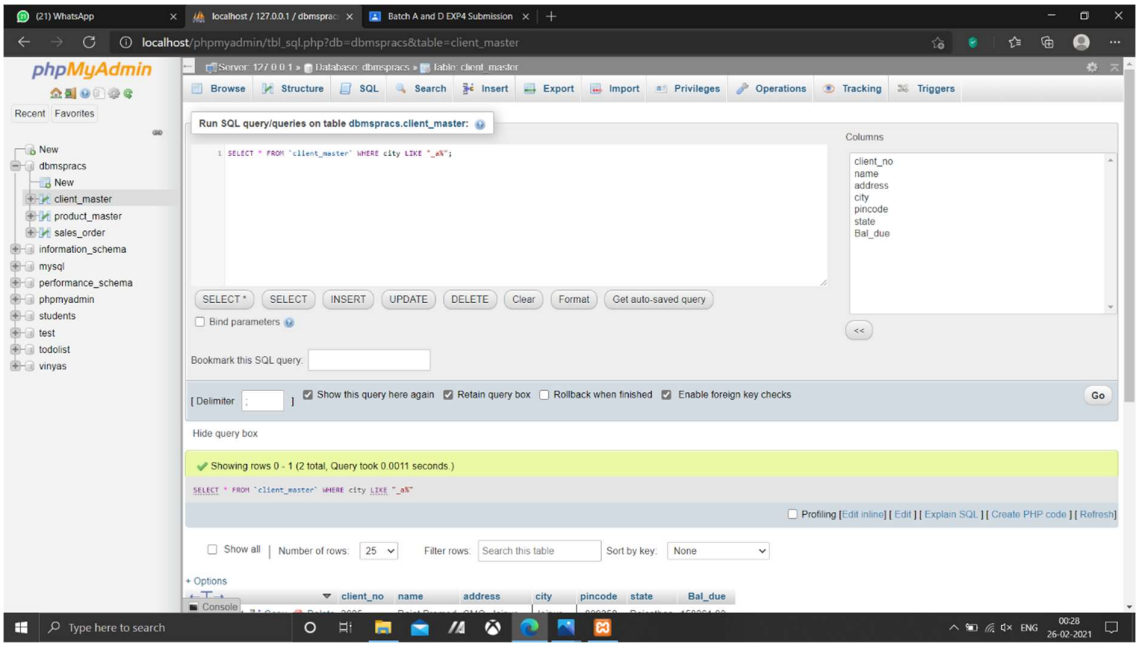


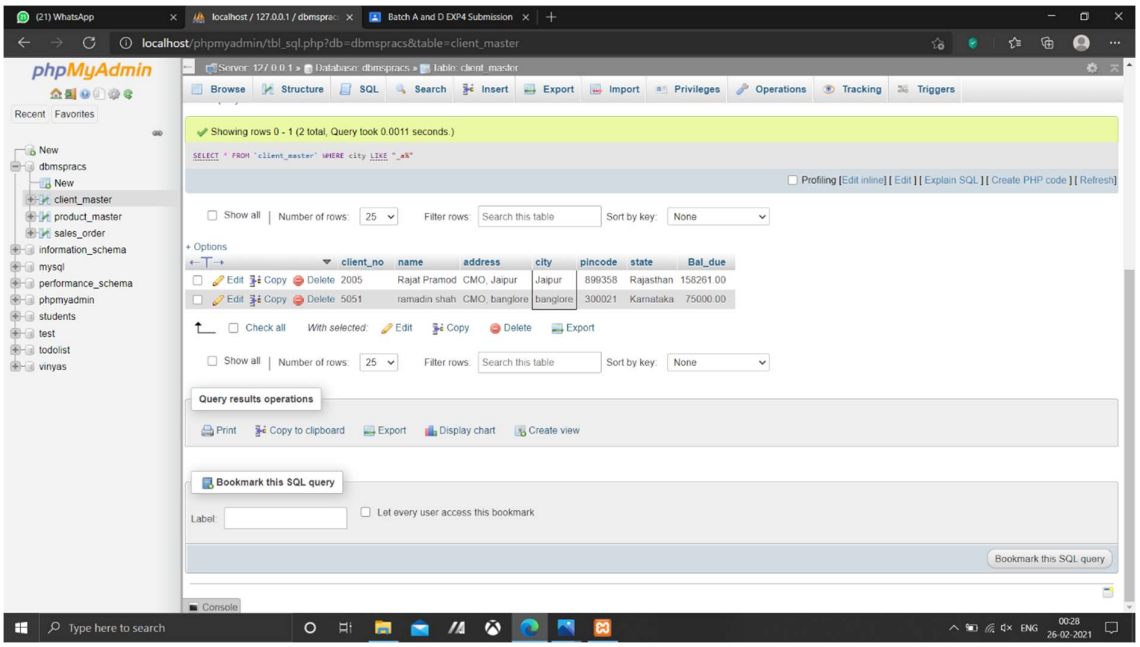


4. Find out the clients who stay in a city whose second letter is ‘a’

Query:- SELECT * FROM `client_master` WHERE city LIKE "_a%";

Screenshots:

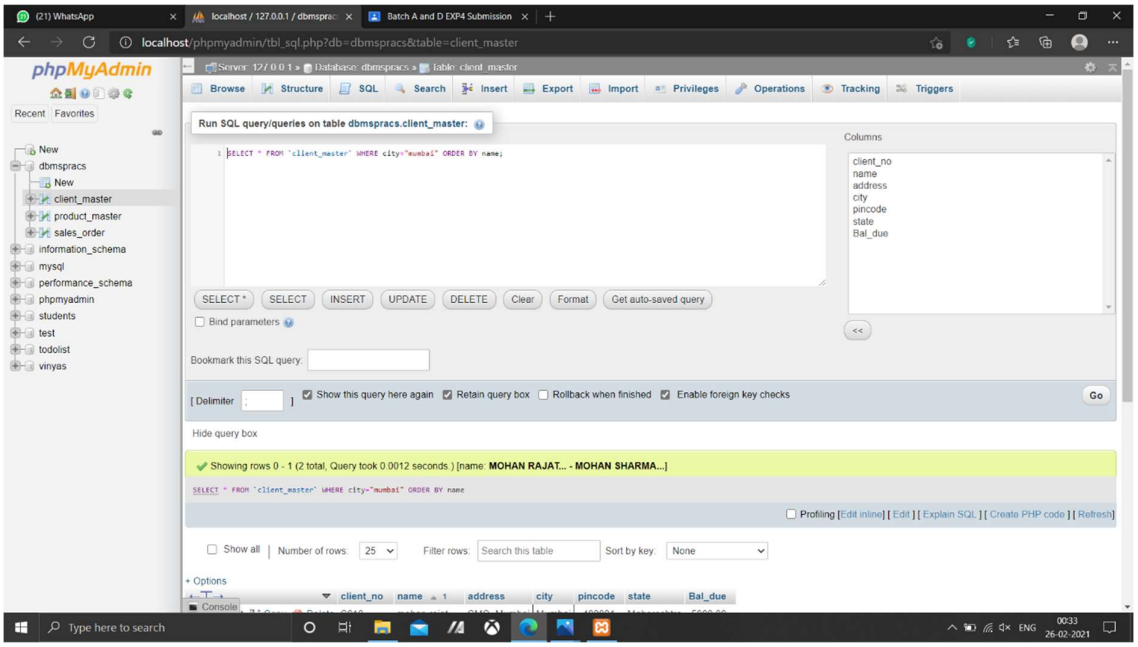


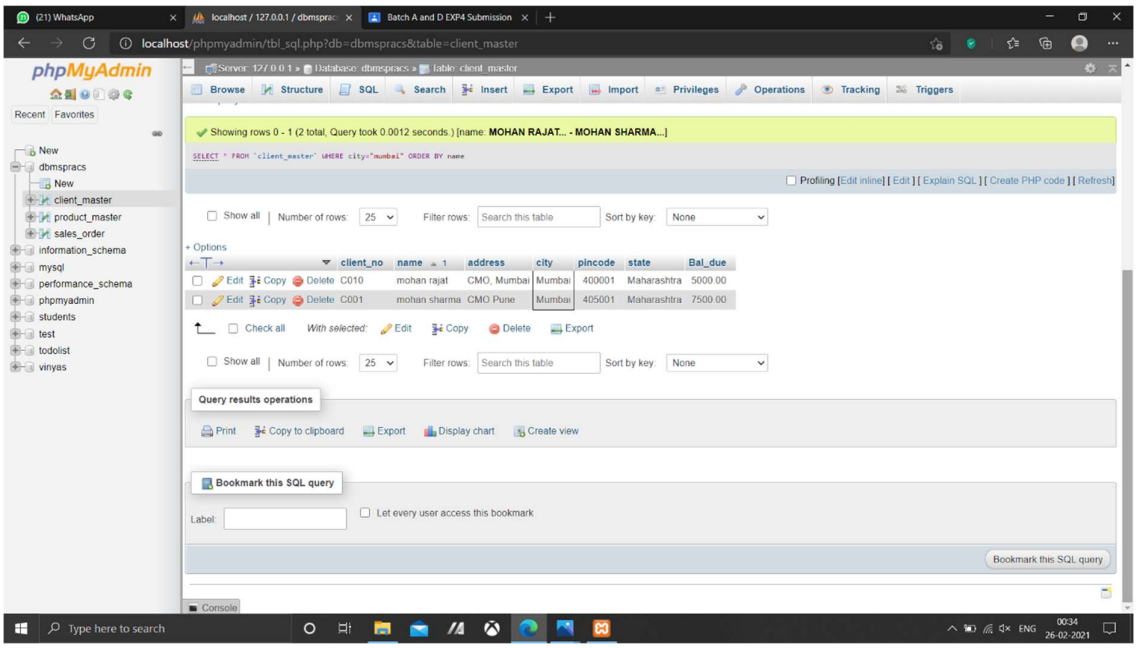


5. Find the list of all clients who stay in ‘mumbai’ ordered by their names

Query:- SELECT * FROM `client_master` WHERE city="mumbai" ORDER BY name;

Screenshots:

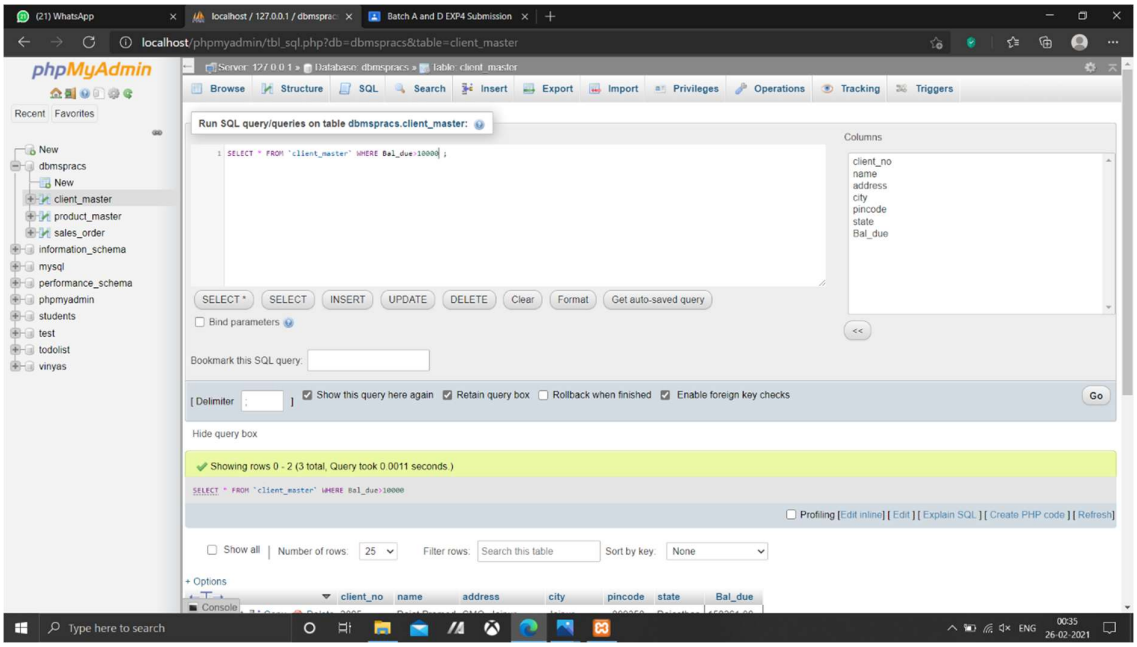


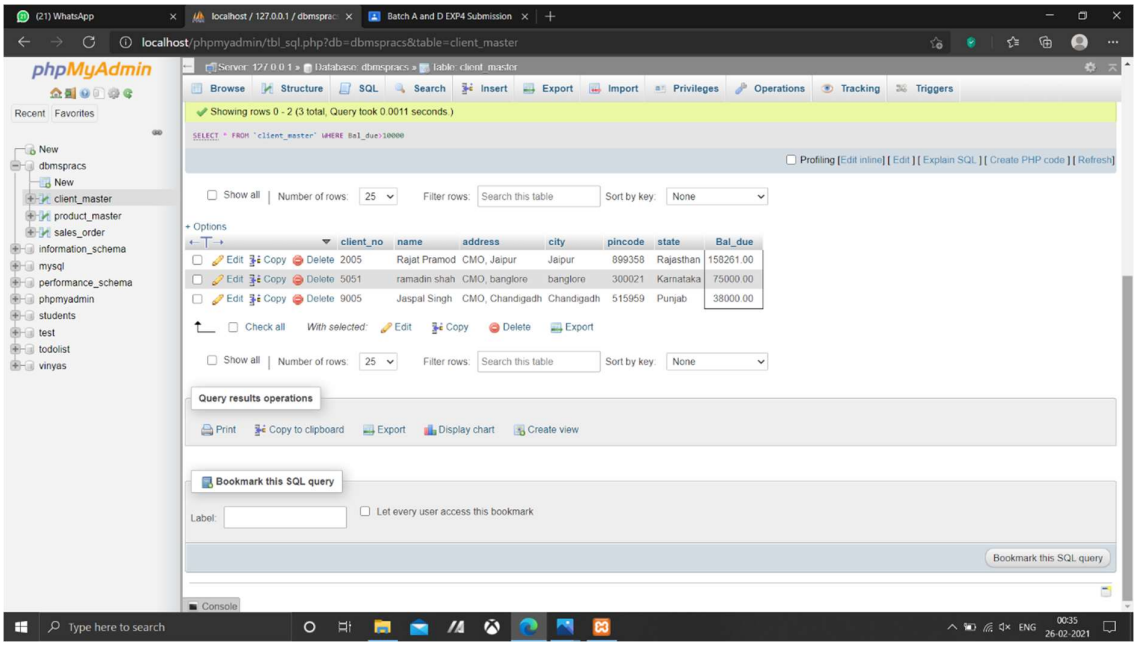


6. Print the list of clients whose bal_due is greater than value 10000

Query :- SELECT * FROM `client_master` WHERE Bal_due>10000 ;

Screenshots:

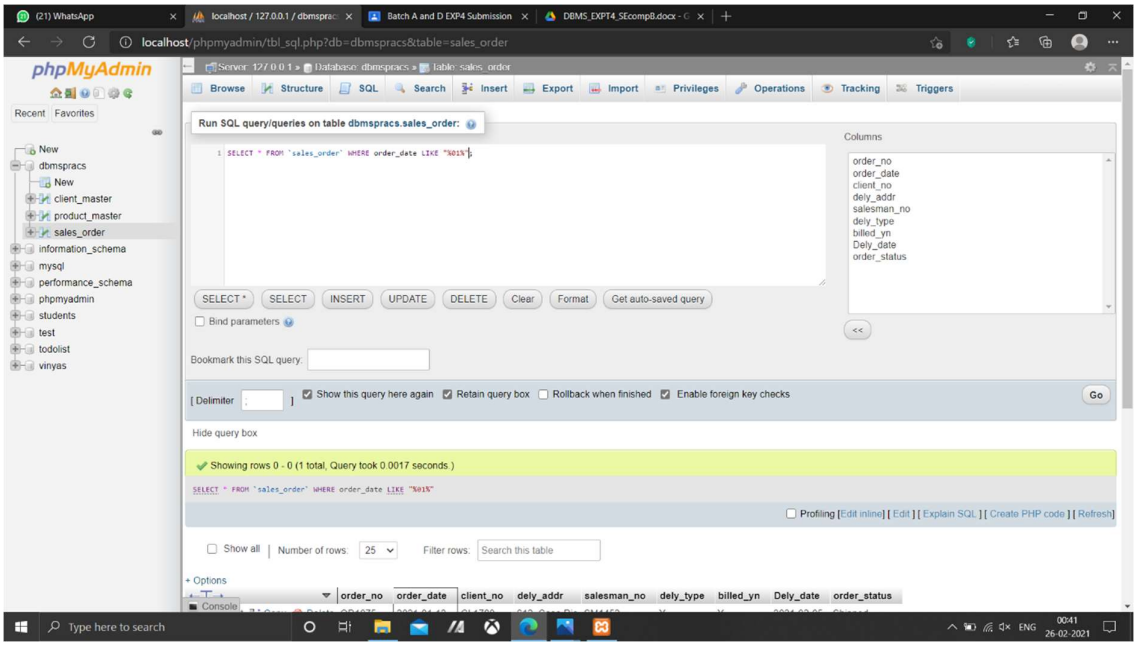


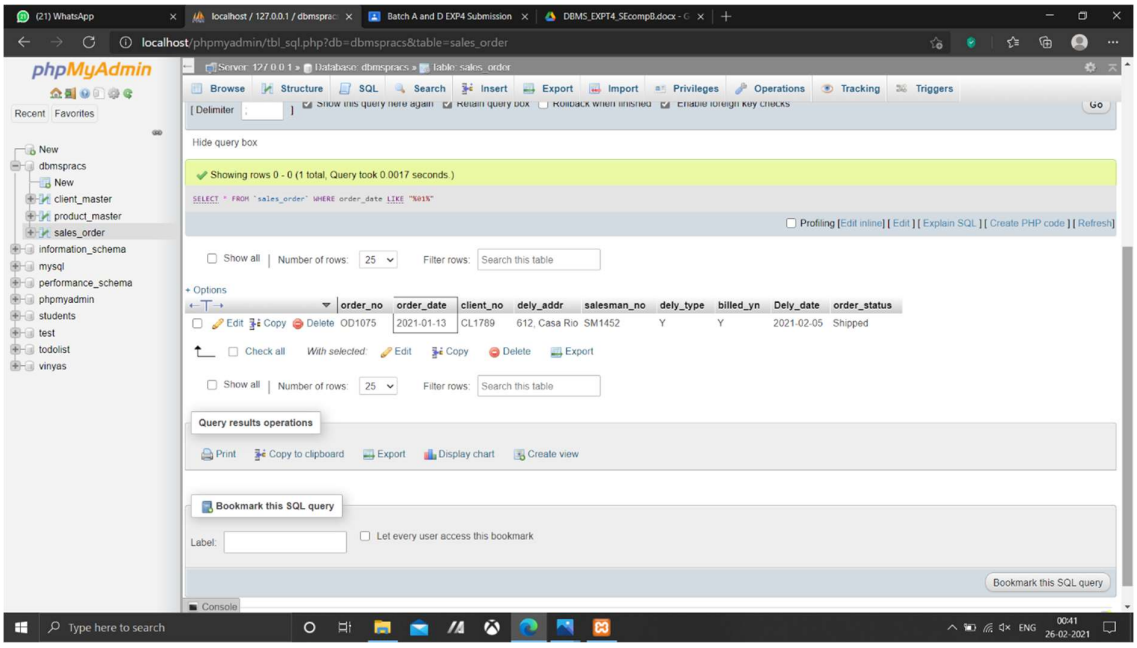


7. Print the information from sales_order table for orders placed in the month of January

Query:- SELECT * FROM `sales_order` WHERE order_date LIKE "%01%";

Screenshots:

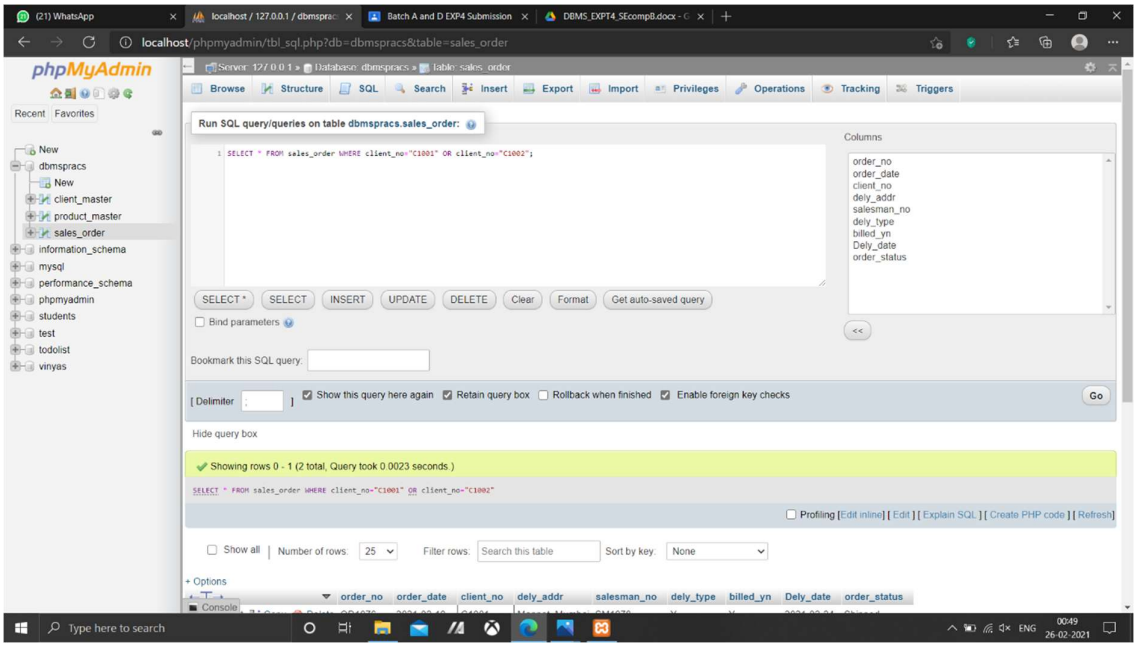


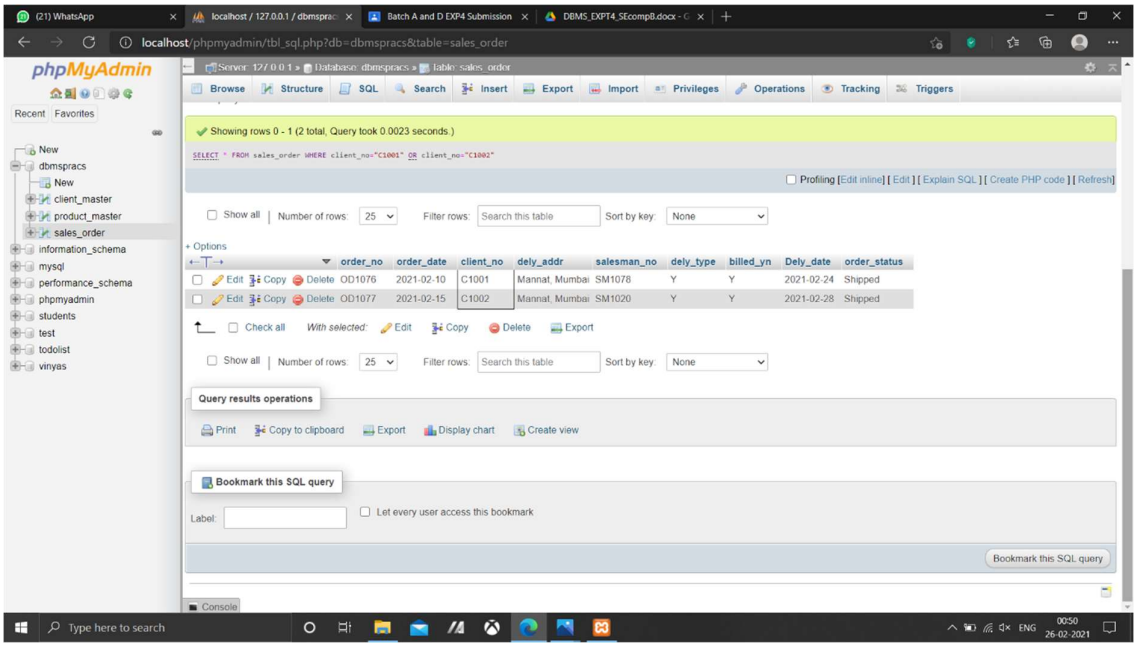


8. Display the order information for client_no C001 and C002

Query :- SELECT * FROM sales_order WHERE client_no="C001" OR client_no="C002";

Screenshots:

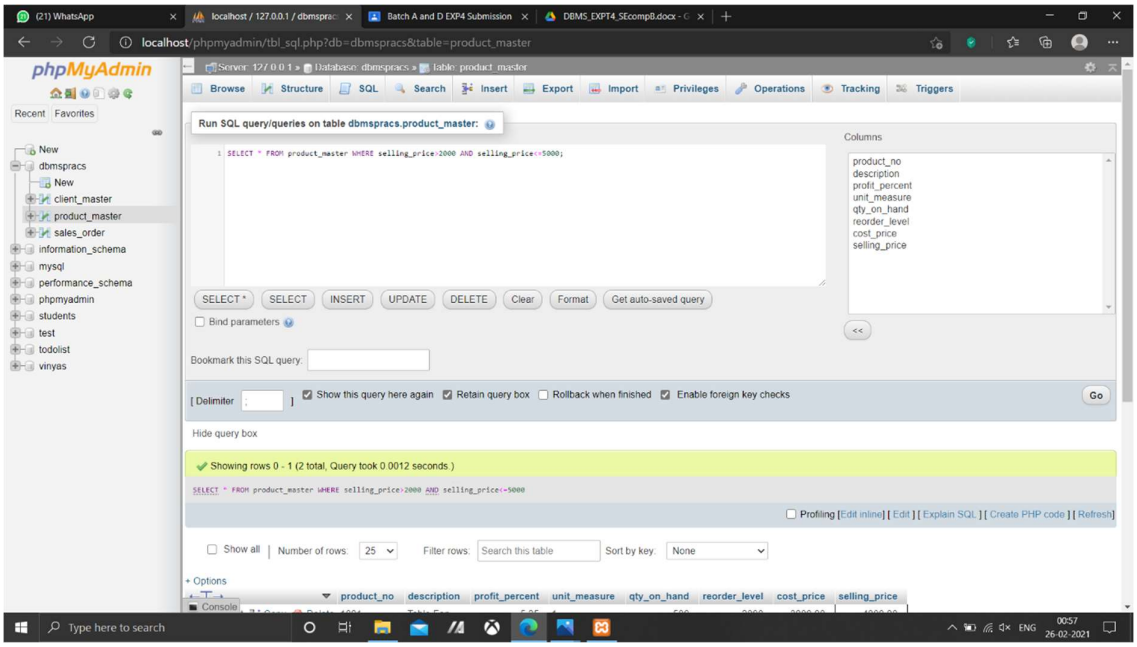


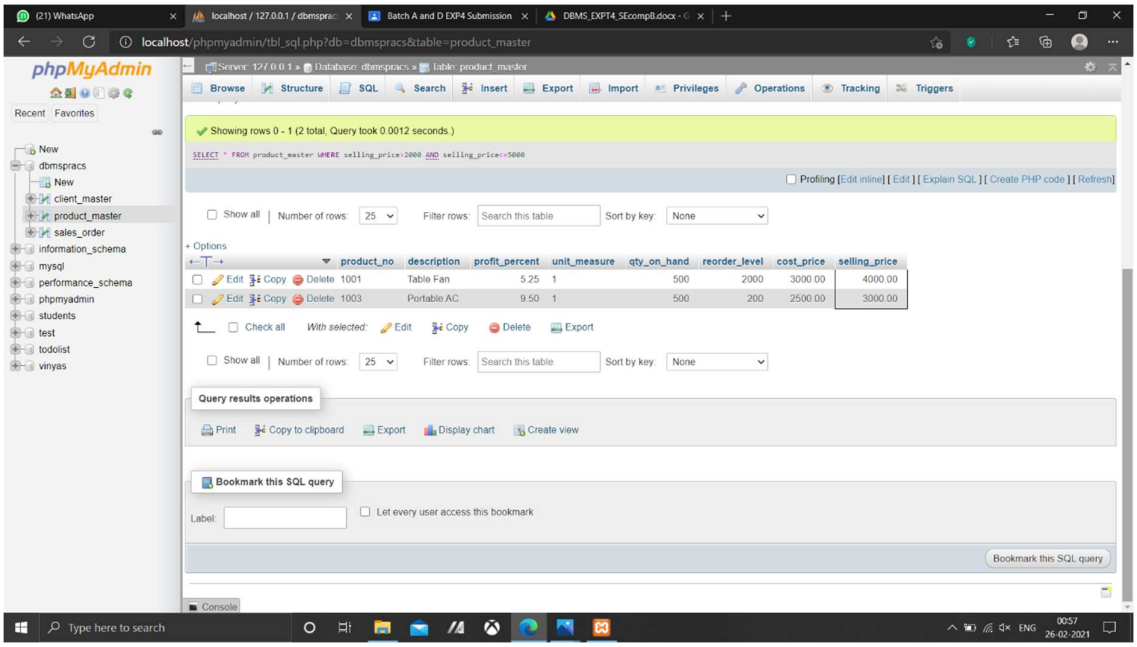


9. Find the products whose selling price is greater than 2000 and less than or equal to 5000

Query :- SELECT * FROM product_master WHERE selling_price>2000 AND selling_price<=5000;

Screenshots:





10. Find the products whose selling price is more than 1500. Calculate new selling price as original selling price * 1.5. Rename the new column in the above query as new_price

Query:-

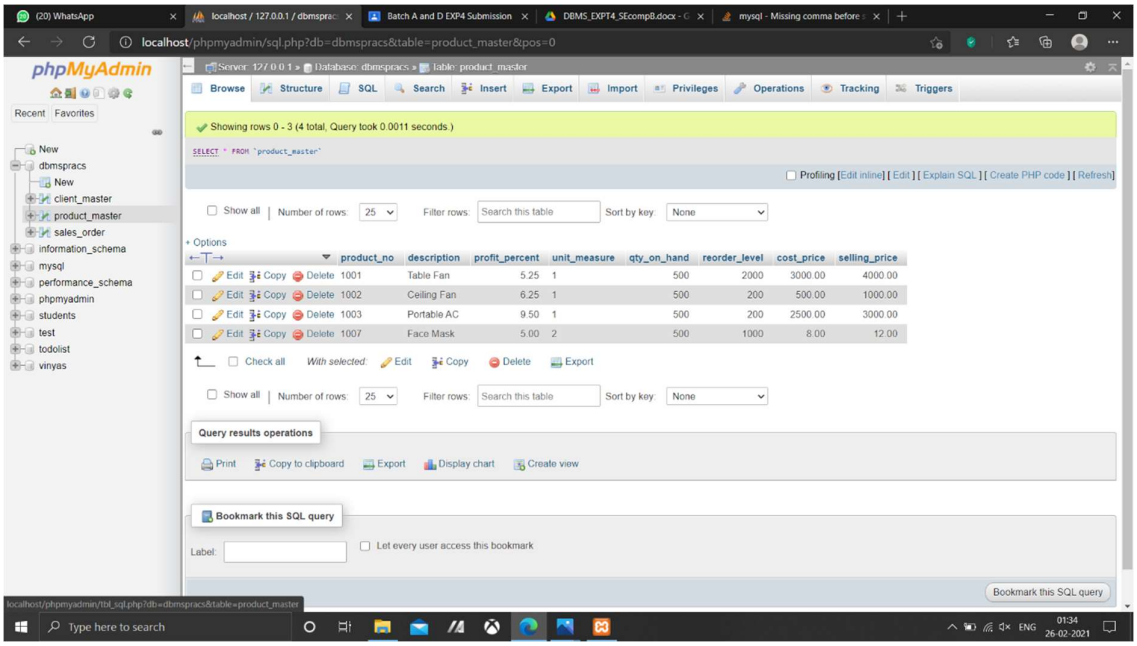
SELECT * FROM product_master WHERE selling_price>=1500;

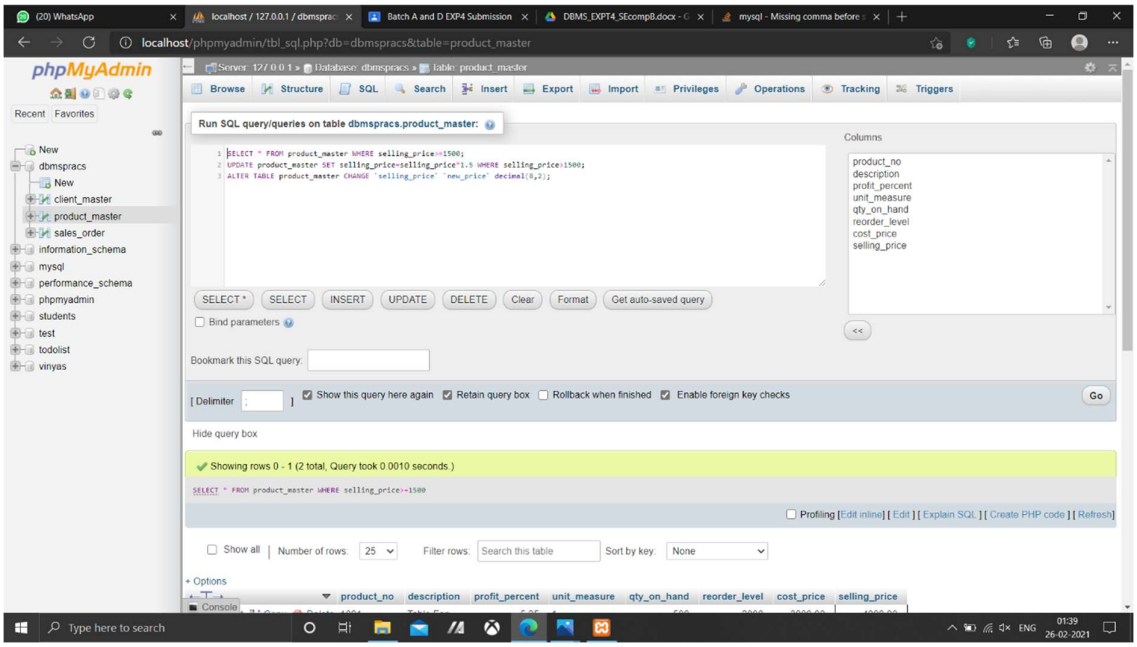
UPDATE product_master SET selling_price=selling_price*1.5 WHERE selling_price>1500;

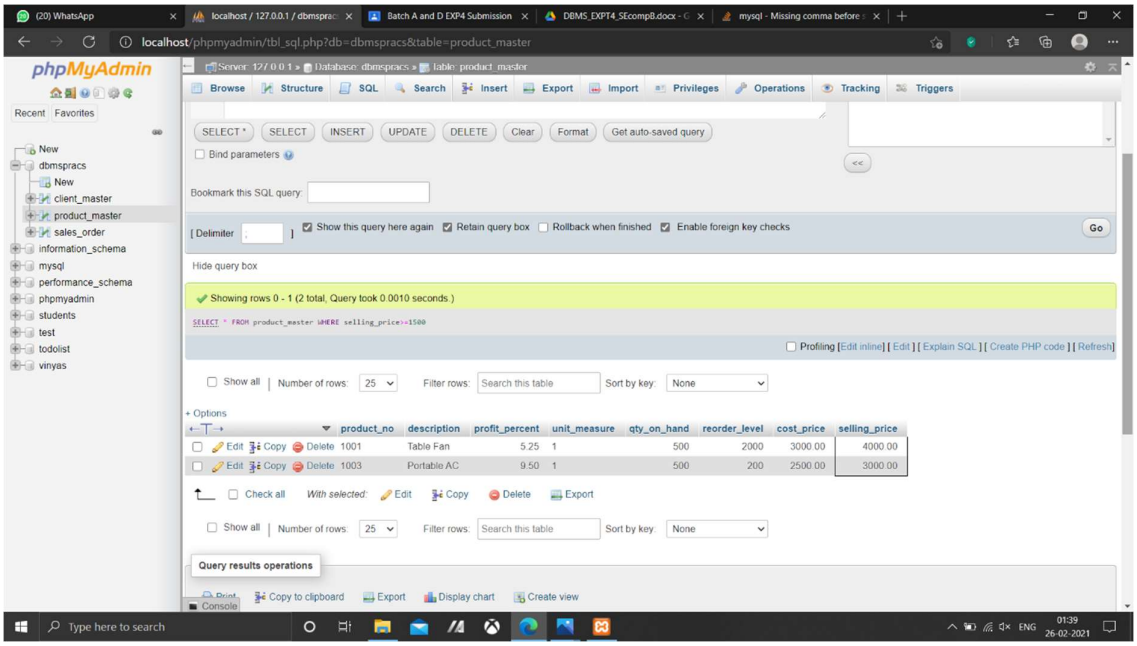
ALTER TABLE product_master CHANGE `selling_price` `new_price` decimal(8,2);

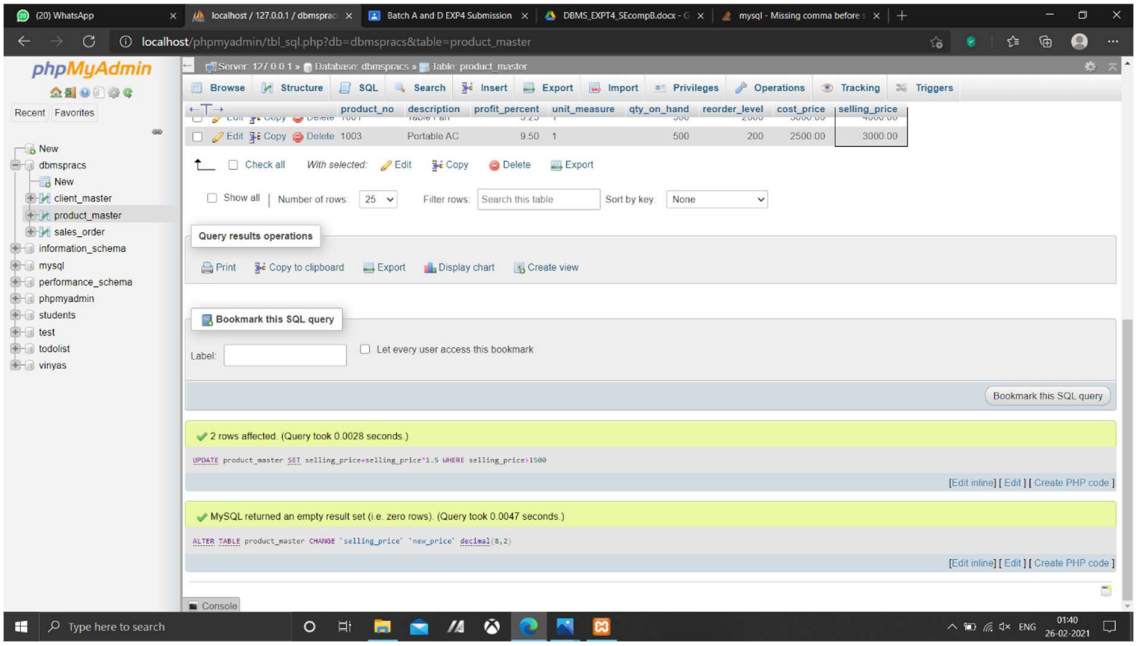
Screenshots:

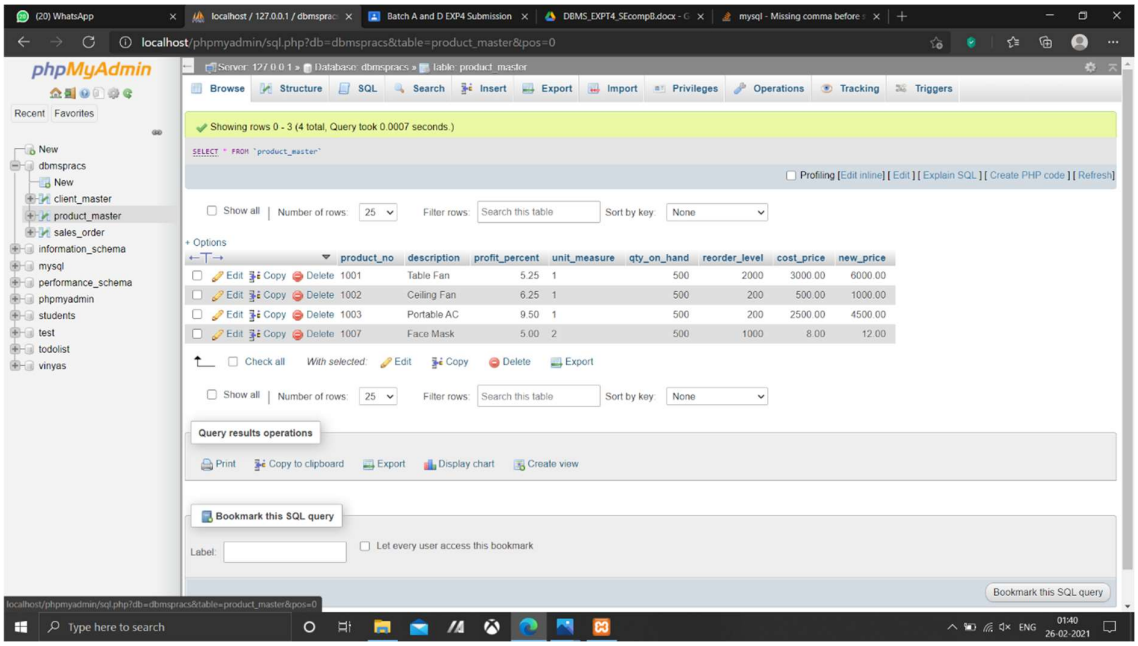
Before:






localhost/phpmyadmin/tbl_sql.php?db=dbmspracs&table=product_master
Server: 127.0.0.1 » Database: dbmspracs » Table: product_master
Browse Structure SQL Search Insert Export Import Privileges Operations Tracking Triggers
SELECT * FROM product_master WHERE selling_price>1500
Showing rows 0 - 1 (2 total, Query took 0.0010 seconds.)
product_no description profit_percent unit_measure qty_on_hand reorder_level cost_price selling_price
1001 Table Fan 5.25 1 500 2000 3000.00 4000.00
1003 Portable AC 9.50 1 500 200 2500.00 3000.00
Query results operations
Print Copy to clipboard Export Display chart Create view

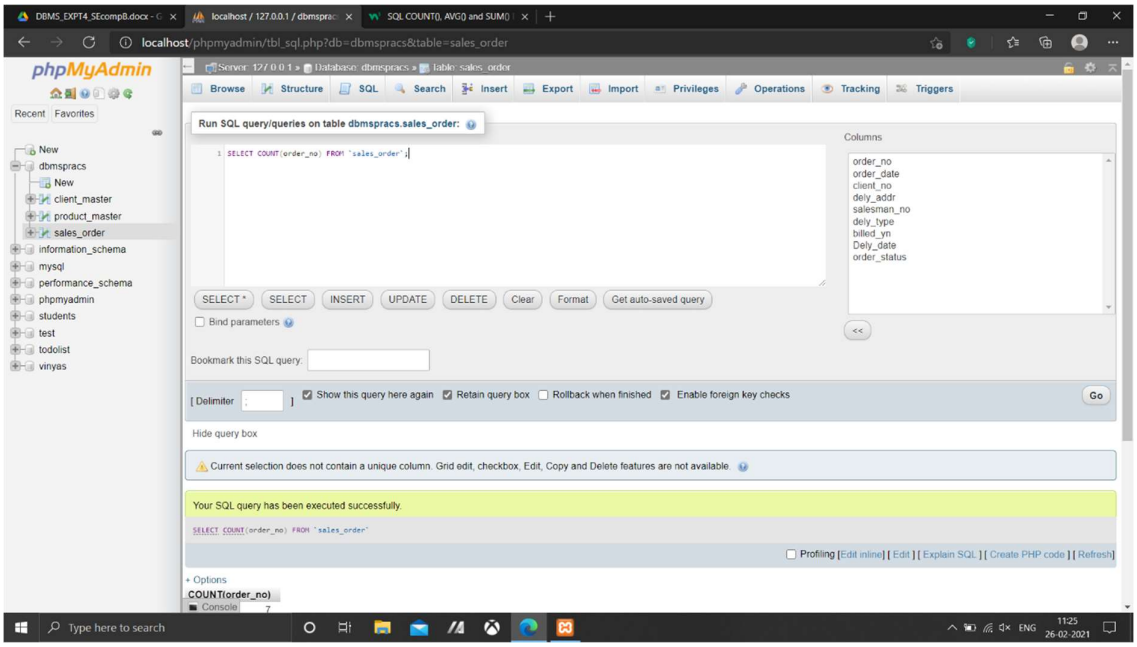

localhost/phpmyadmin/tbl_sql.php?db=dbmspracs&table=product_master
Server: 127.0.0.1 » Database: dbmspracs » Table: product_master
Browse Structure SQL Search Insert Export Import Privileges Operations Tracking Triggers
product_no description profit_percent unit_measure qty_on_hand reorder_level cost_price selling_price
1003 Portable AC 9.50 1 500 200 2500.00 3000.00
Query results operations
Print Copy to clipboard Export Display chart Create view
Bookmark this SQL query
Label:
Let every user access this bookmark
Bookmark this SQL query
2 rows affected. (Query took 0.0028 seconds.)
UPDATE product_master SET selling_price=selling_price*1.5 WHERE selling_price>1500
MySQL returned an empty result set (i.e. zero rows). (Query took 0.0047 seconds.)
ALTER TABLE product_master CHANGE `selling_price` `new_price` decimal(8,2)

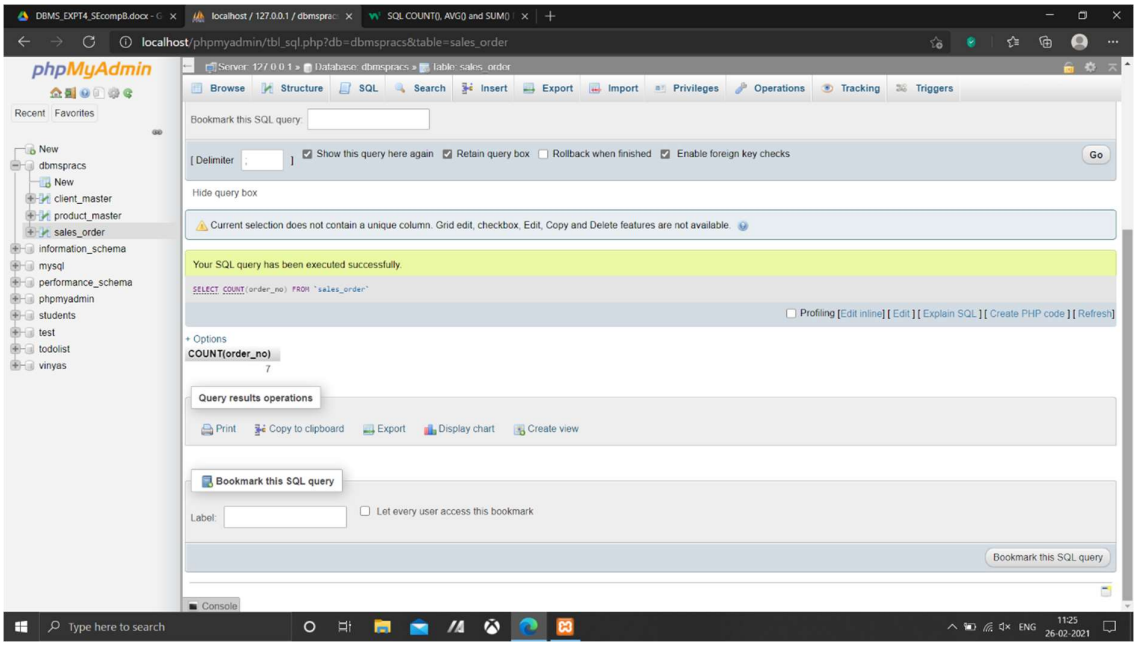

localhost/phpmyadmin/sql.php?db=dbmspracs&table=product_master&pos=0
Server: 127.0.0.1 » Database: dbmspracs » Table: product_master
Browse Structure SQL Search Insert Export Import Privileges Operations Tracking Triggers
SELECT * FROM `product_master`
Showing rows 0 - 3 (4 total, Query took 0.0007 seconds.)
product_no description profit_percent unit_measure qty_on_hand reorder_level cost_price new_price
1001 Table Fan 5.25 1 500 2000 3000.00 6000.00
1002 Ceiling Fan 6.25 1 500 200 500.00 1000.00
1003 Portable AC 9.50 1 500 200 2500.00 4500.00
1007 Face Mask 5.00 2 500 1000 8.00 12.00
Query results operations
Print Copy to clipboard Export Display chart Create view
Bookmark this SQL query
Label:
Let every user access this bookmark
Bookmark this SQL query

11. Count the total number of orders

Query:- SELECT COUNT(order_no) FROM `sales_order`;

Screenshots:





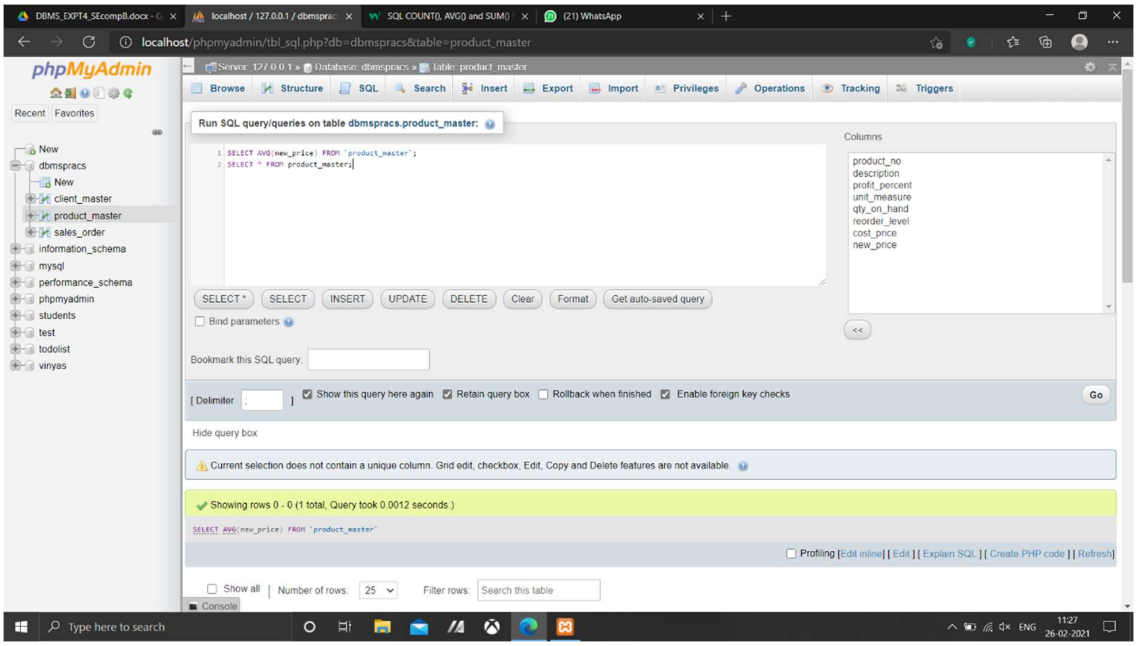
12. Calculate the average price of all the product

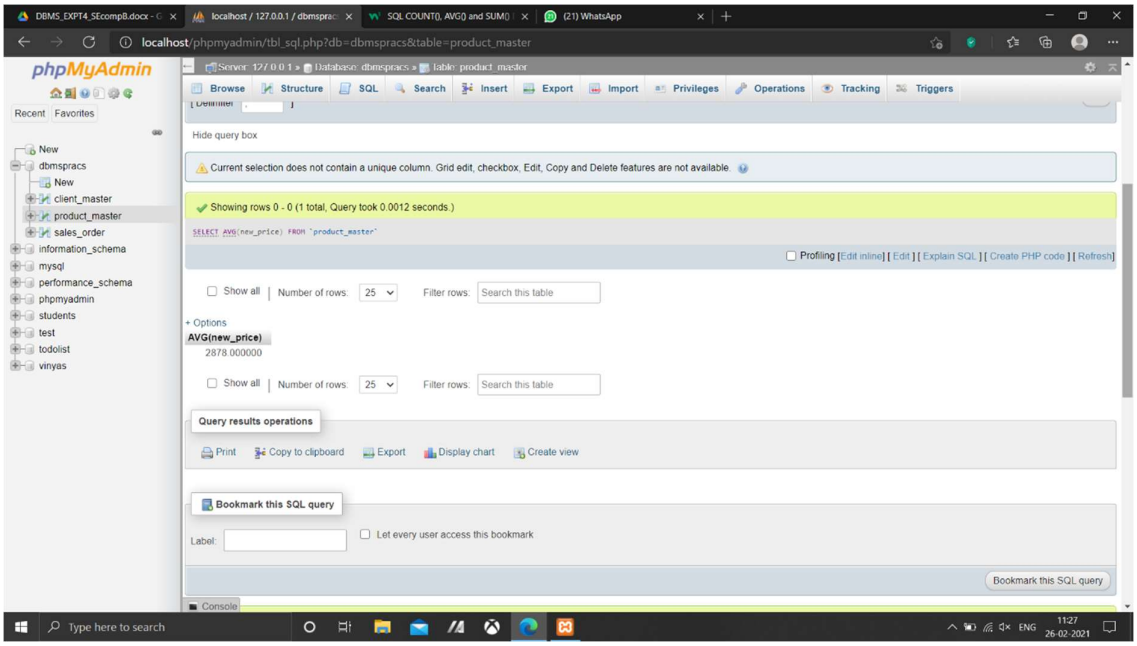
Query:-

SELECT AVG(new_price) FROM `product_master`;

SELECT * FROM product_master;

Screenshots:





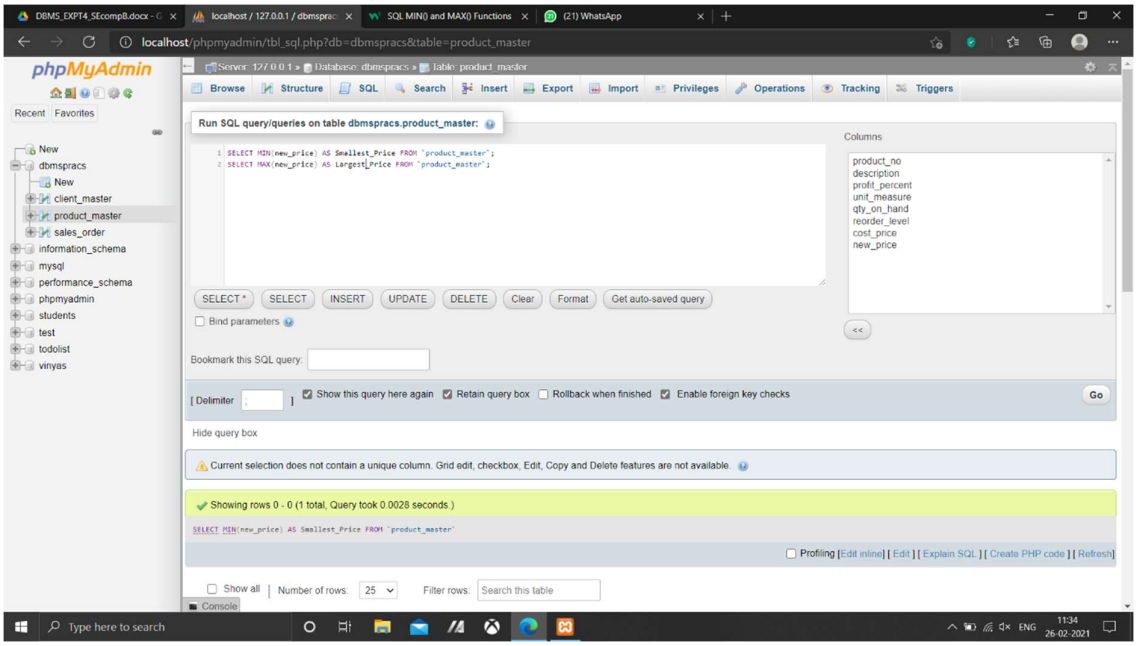
13. Determine minimum and maximum product prices

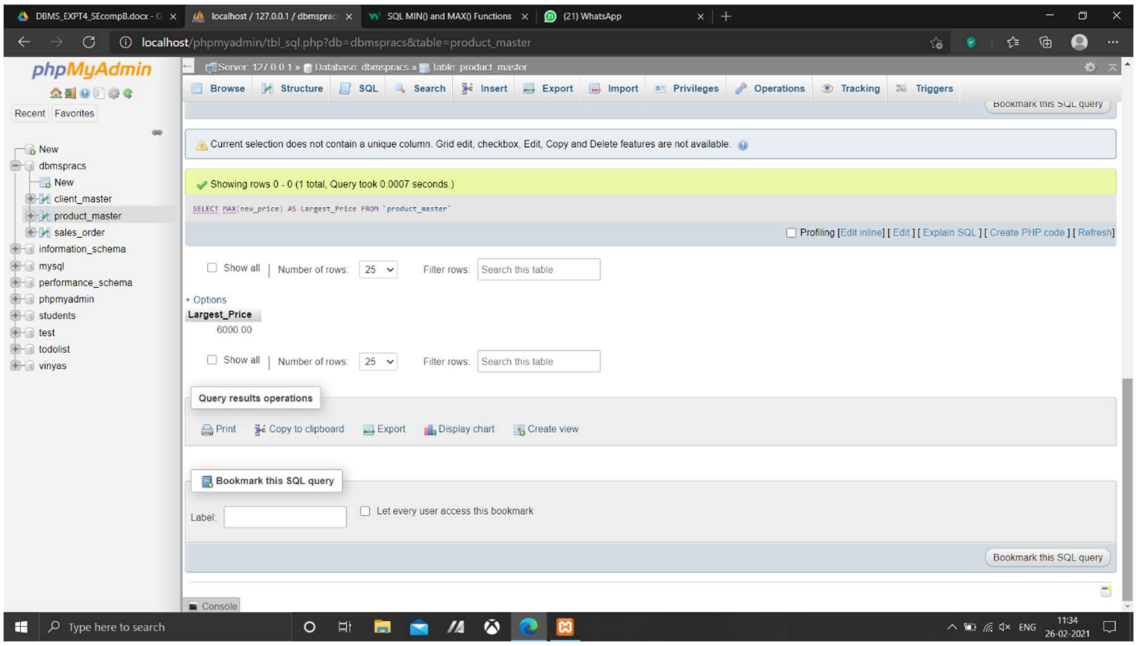
Query:-

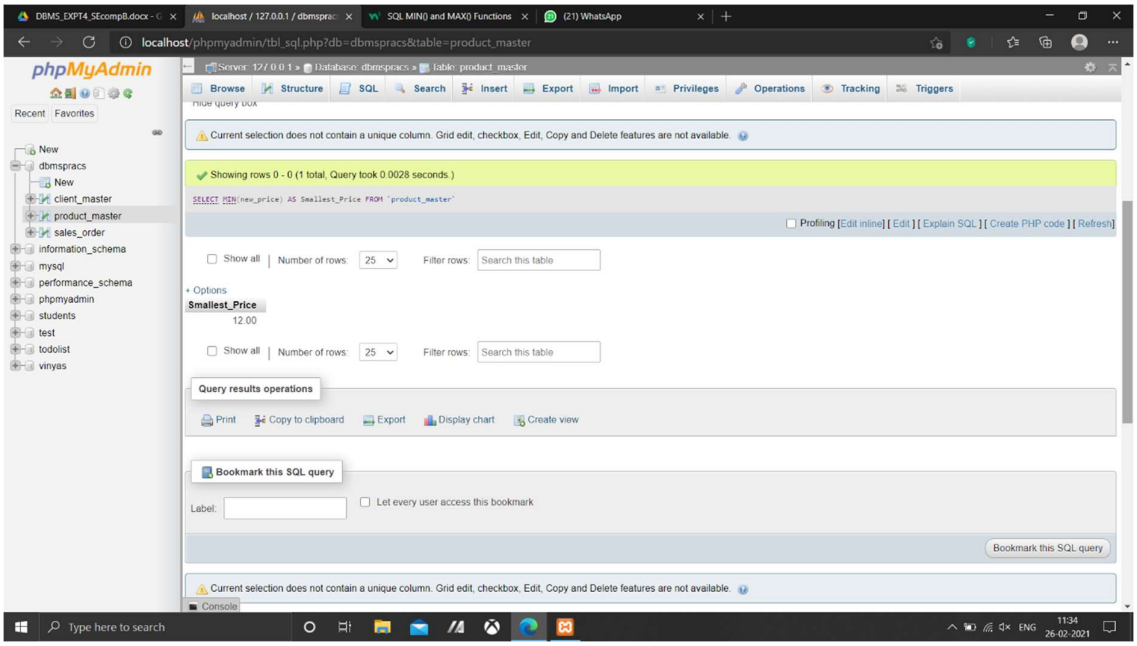
SELECT MIN(new_price) AS Smallest_Price FROM `product_master`;

SELECT MAX(new_price) AS Largest_Price FROM `product_master`;

Screenshots:






DBMS_EXPT4_SEcompB.docx - C
localhost / 127.0.0.1 / dbmsprac
SQL MIN() and MAX() Functions
(21) WhatsApp
localhost/phpmyadmin/tbl_sql.php?db=dbmspracs&table=product_master
phpMyAdmin
Recent
Favorites
New
dbmspracs
New
client_master
product_master
sales_order
information_schema
mysql
performance_schema
phpmyadmin
students
test
todolist
vinyas
Server: 127.0.0.1 » Database: dbmspracs » Table: product_master
Browse
Structure
SQL
Search
Insert
Export
Import
Privileges
Operations
Tracking
Triggers
Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available.
Showing rows 0 - 0 (1 total, Query took 0.0028 seconds.)
SELECT MIN(new_price) AS Smallest_Price FROM `product_master`
Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
Show all
Number of rows:
25
Filter rows:
Search this table
Options
Smallest_Price
12.00
Query results operations
Print
Copy to clipboard
Export
Display chart
Create view
Bookmark this SQL query
Label:
Let every user access this bookmark
Bookmark this SQL query
Console
Type here to search
ENG
11:34
26-02-2021

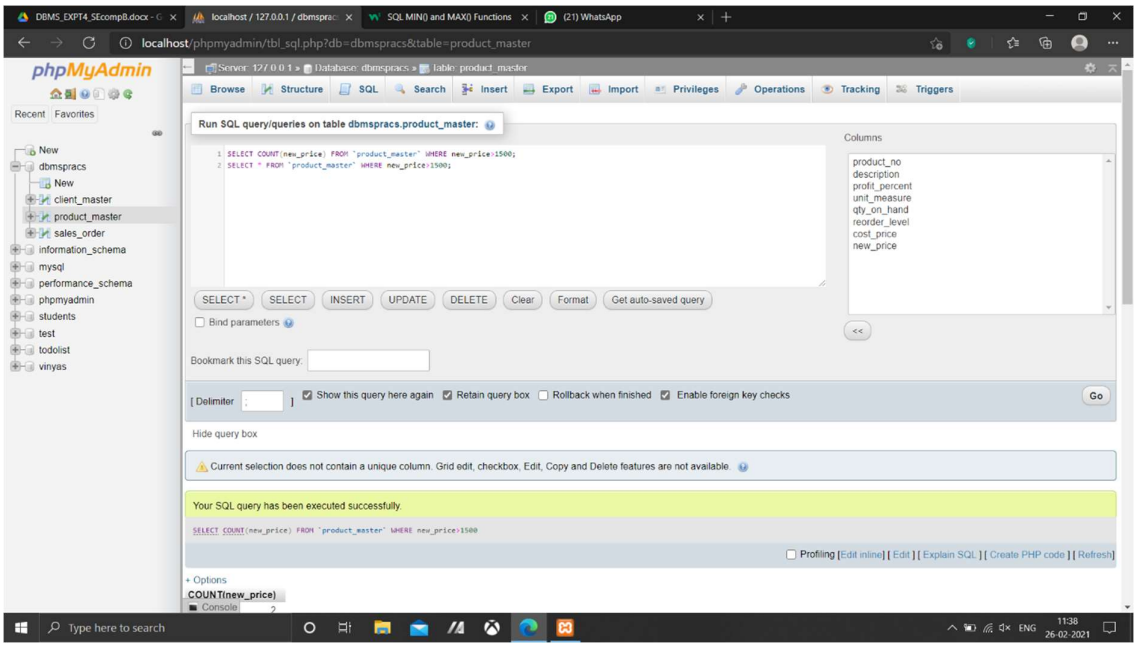
14. count the number of products having price greater than or equal to 1500

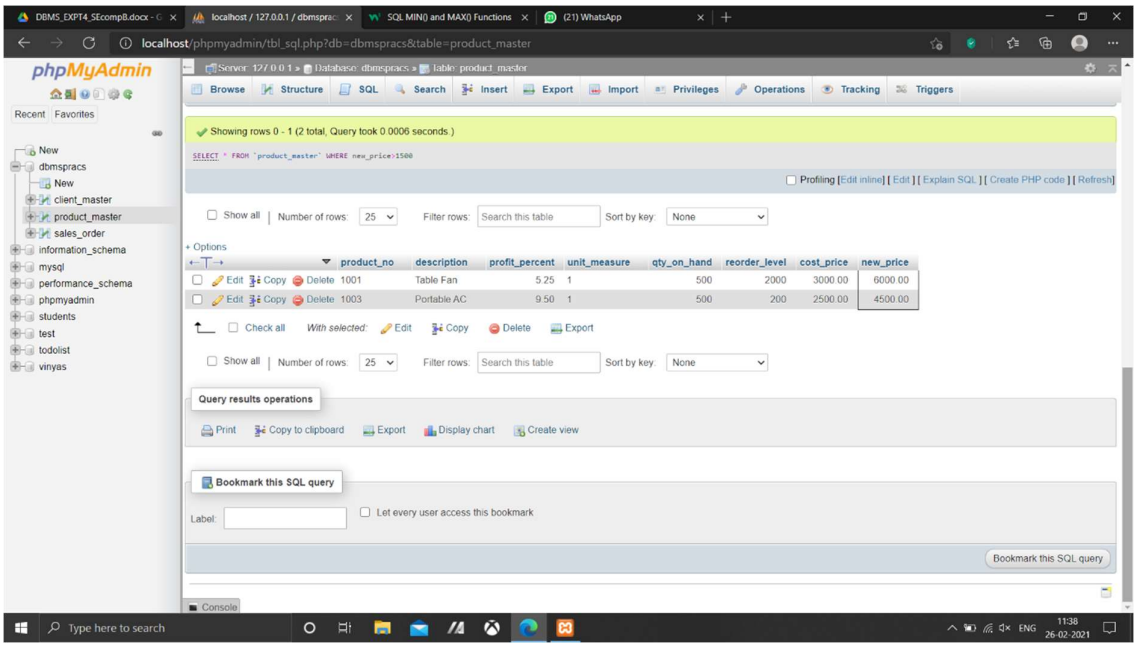
Query:-

SELECT COUNT(new_price) FROM `product_master` WHERE new_price>1500;

SELECT * FROM `product_master` WHERE new_price>1500;

Screenshots:

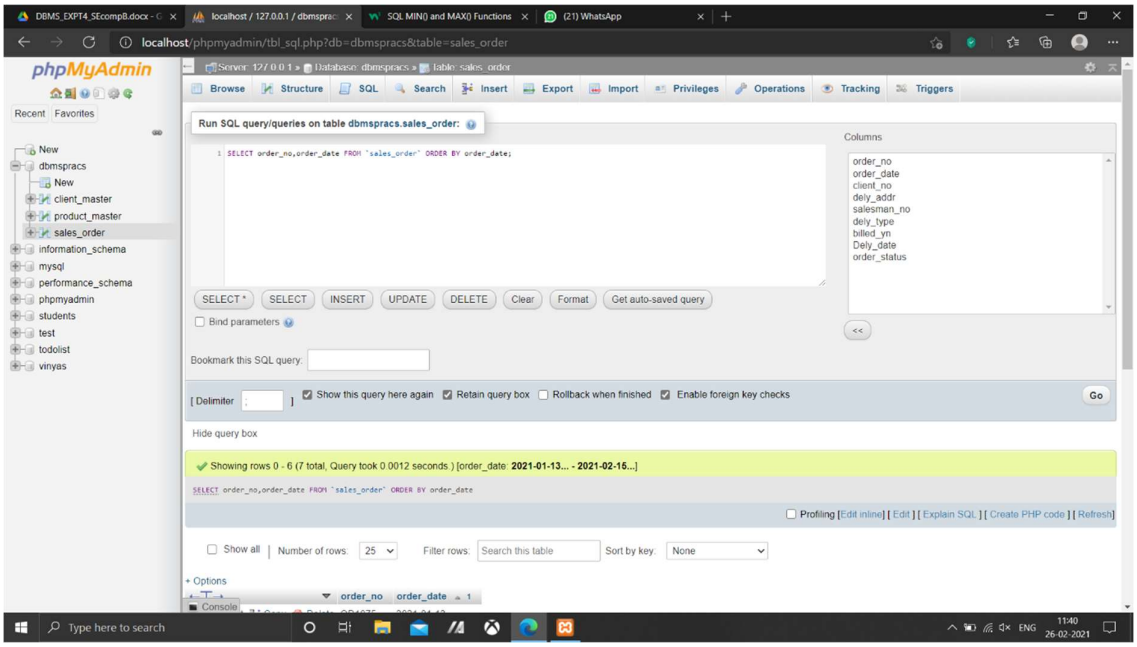


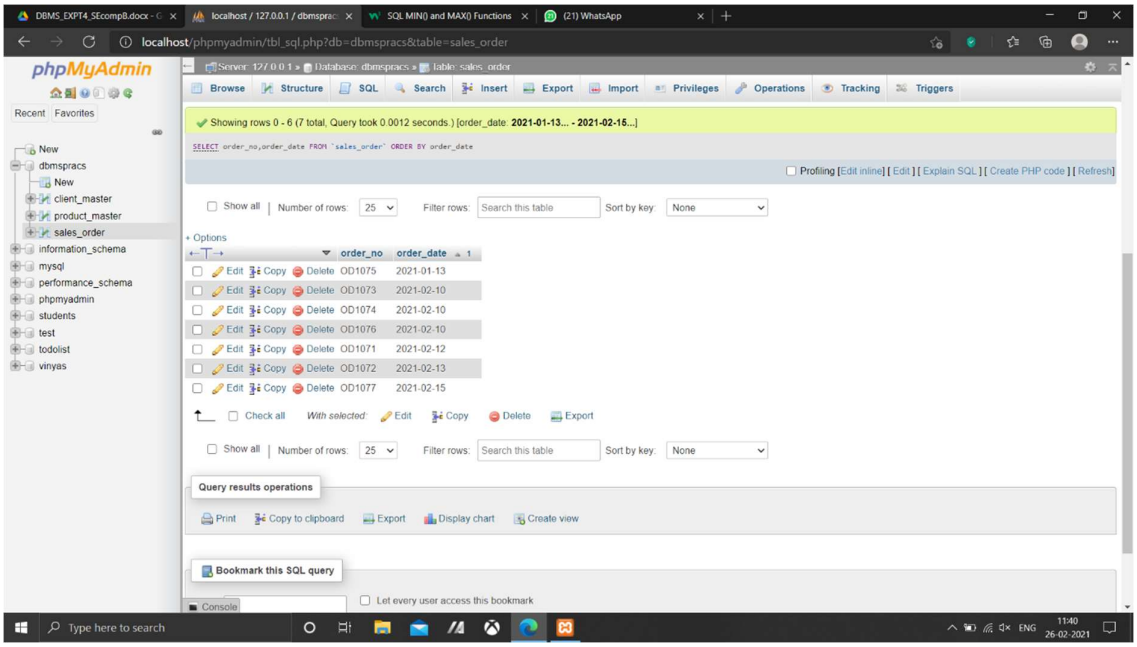


15. Display the order number and day on which clients placed their order

Query:- SELECT order_no,order_date FROM `sales_order` ORDER BY order_date;

Screnshots:



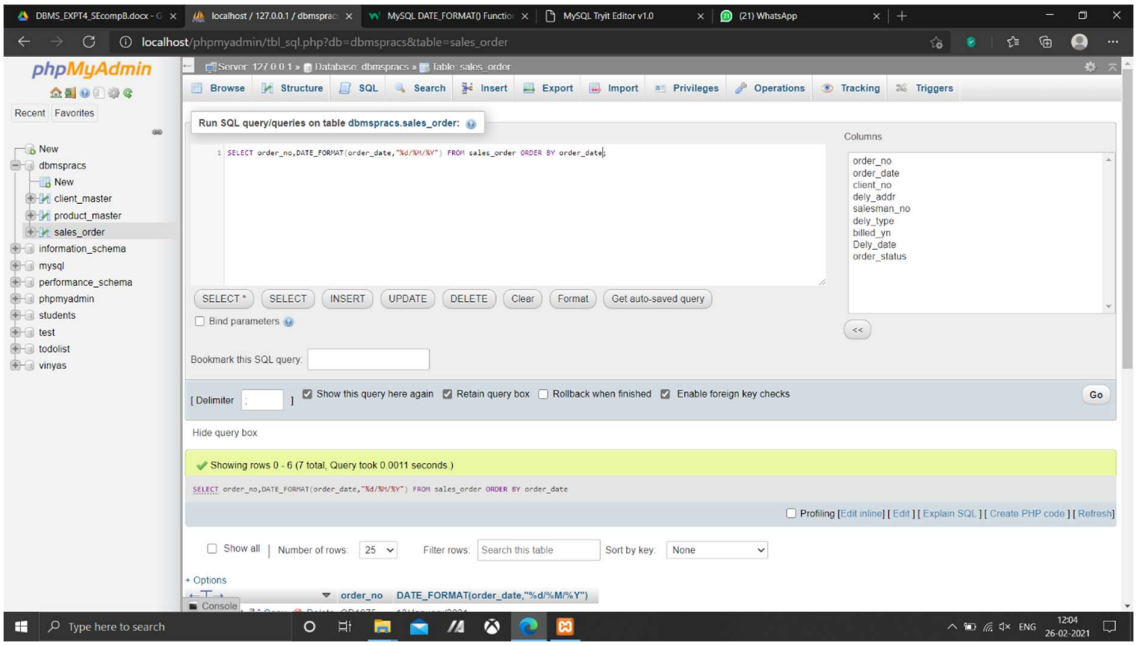


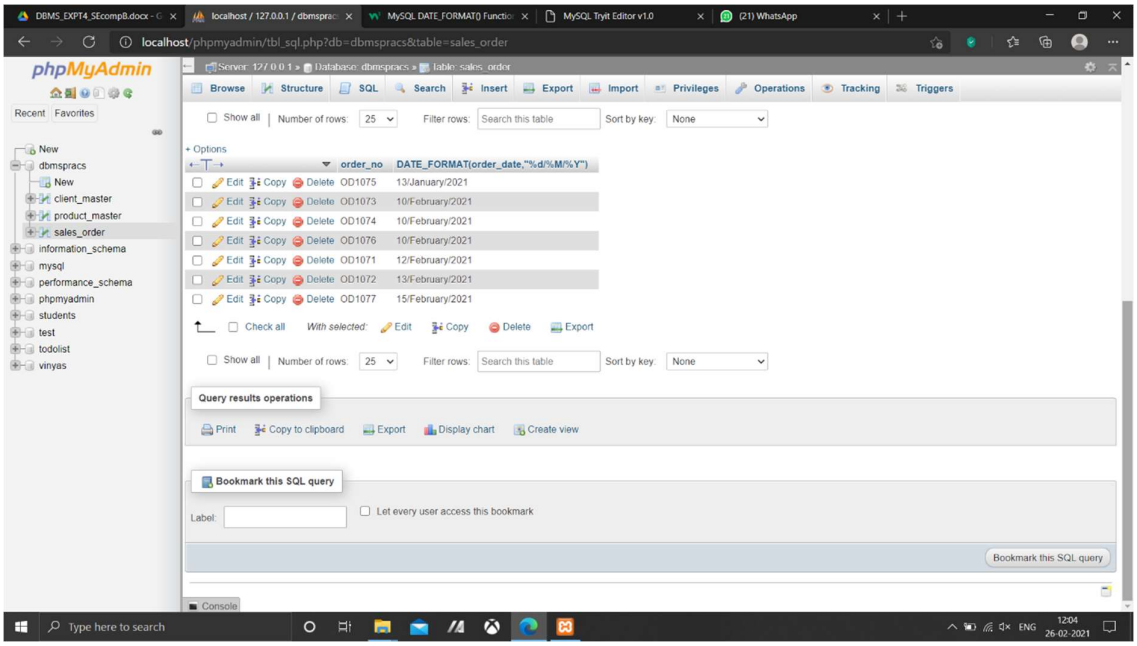
16. Display the order_date in the format ‘dd-month-yy’

Query:-

SELECT order_no,DATE_FORMAT(order_date,"%d/%M/%Y") FROM sales_order ORDER BY order_date;

Screenshots

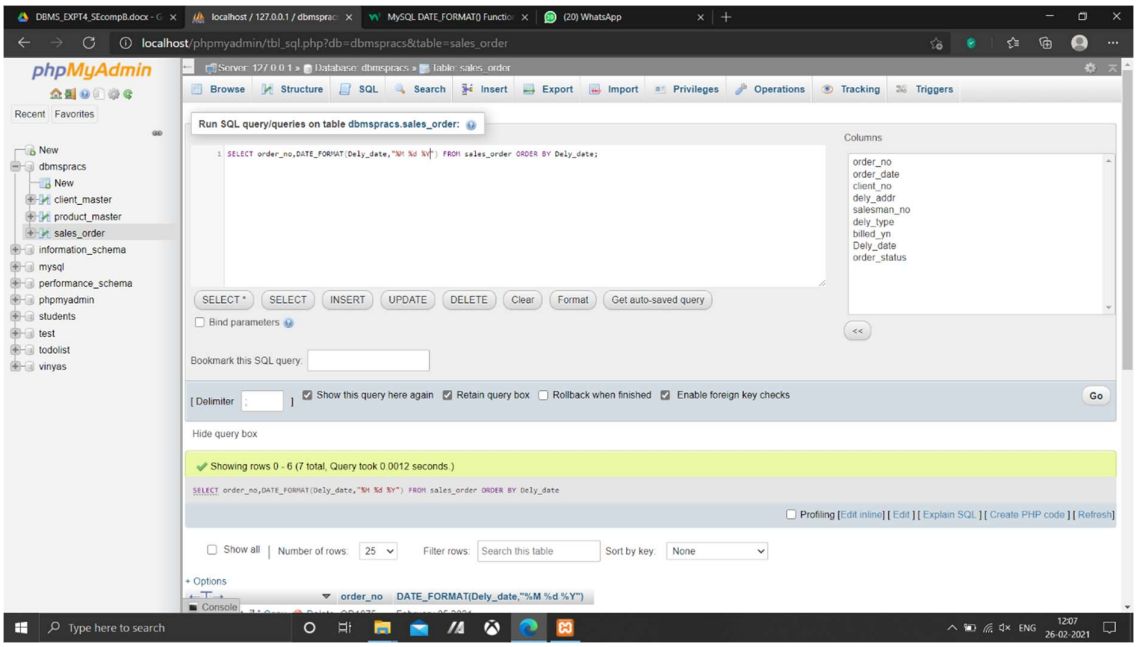


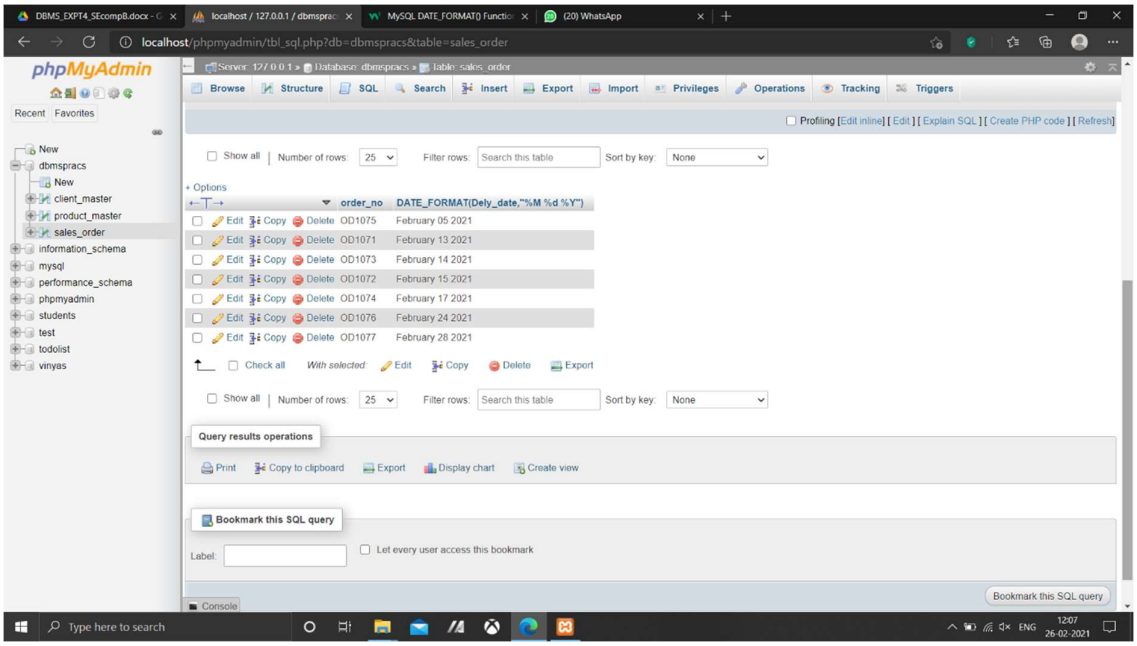


17. Display the month (in alphabets) and date when the order must be delivered

Query:- SELECT order_no,DATE_FORMAT(Dely_date,"%M %d %Y") FROM sales_order ORDER BY Dely_date;

Screenshots:

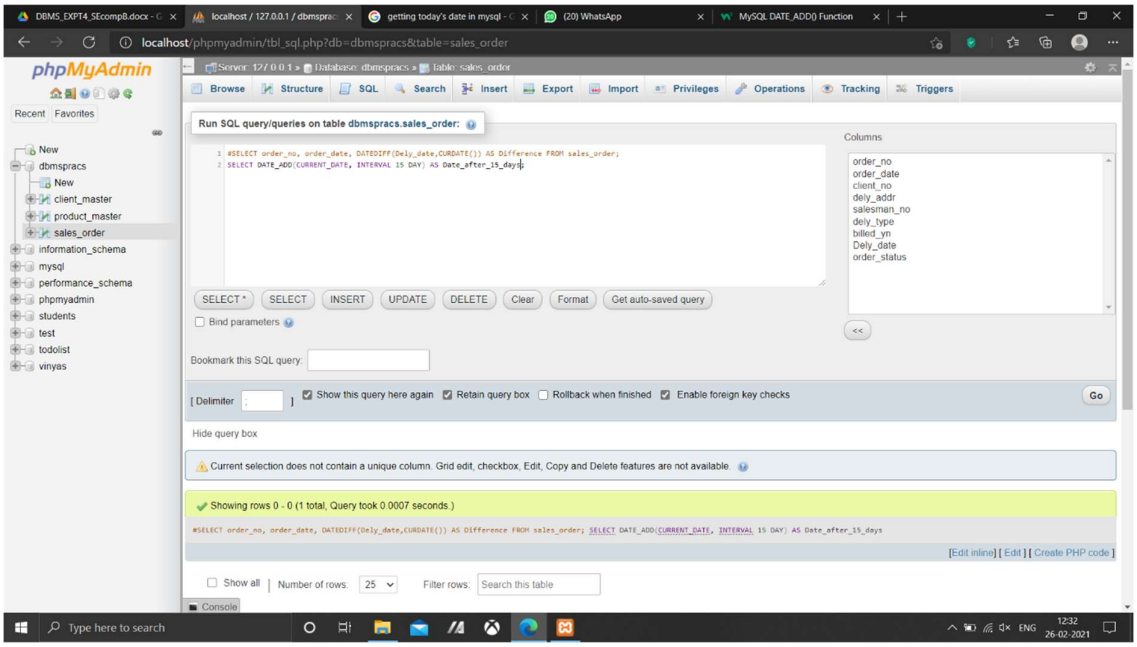


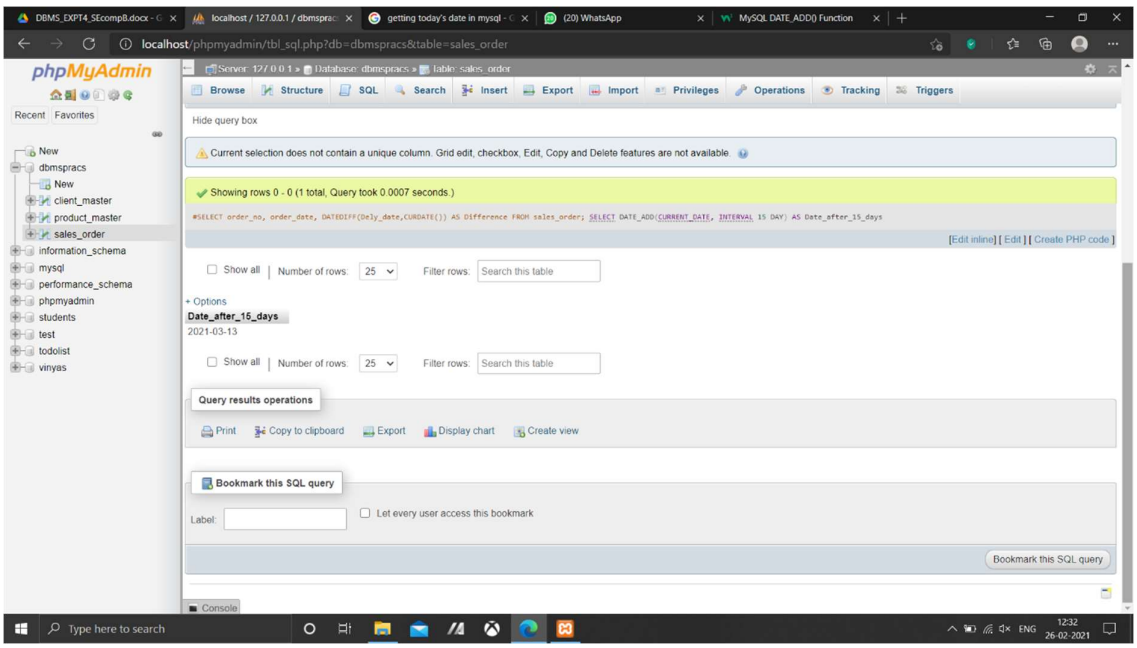


18. Find the date, 15 days after today's date

Query:- SELECT DATE_ADD(CURRENT_DATE, INTERVAL 15 DAY) AS Date_after_15_days;

Screenshots:

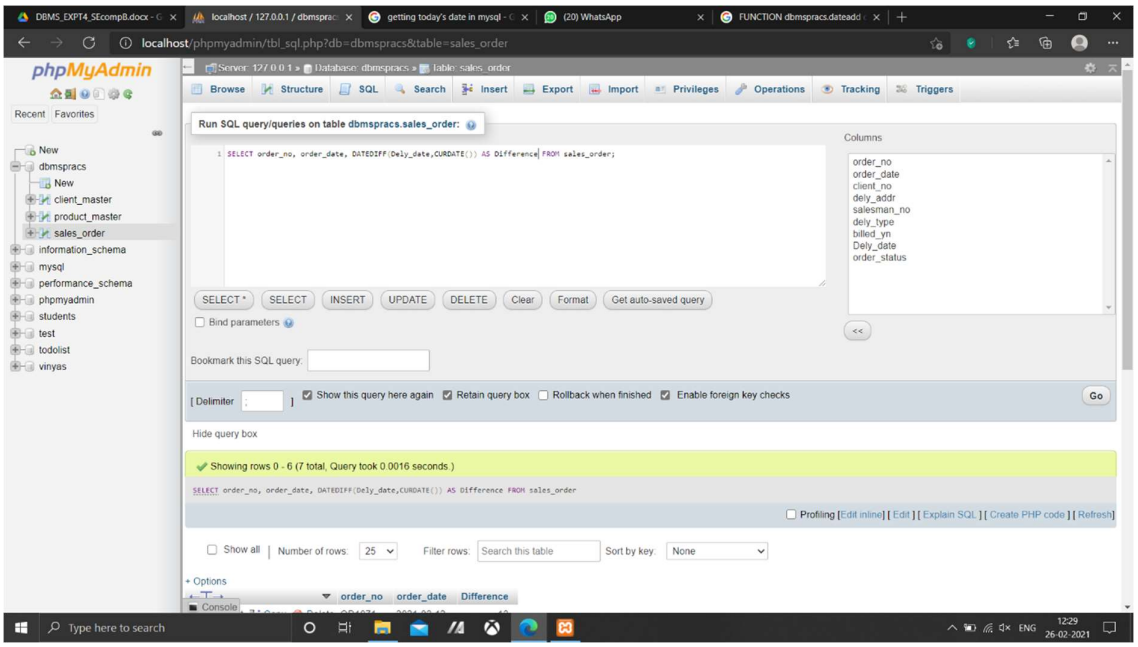


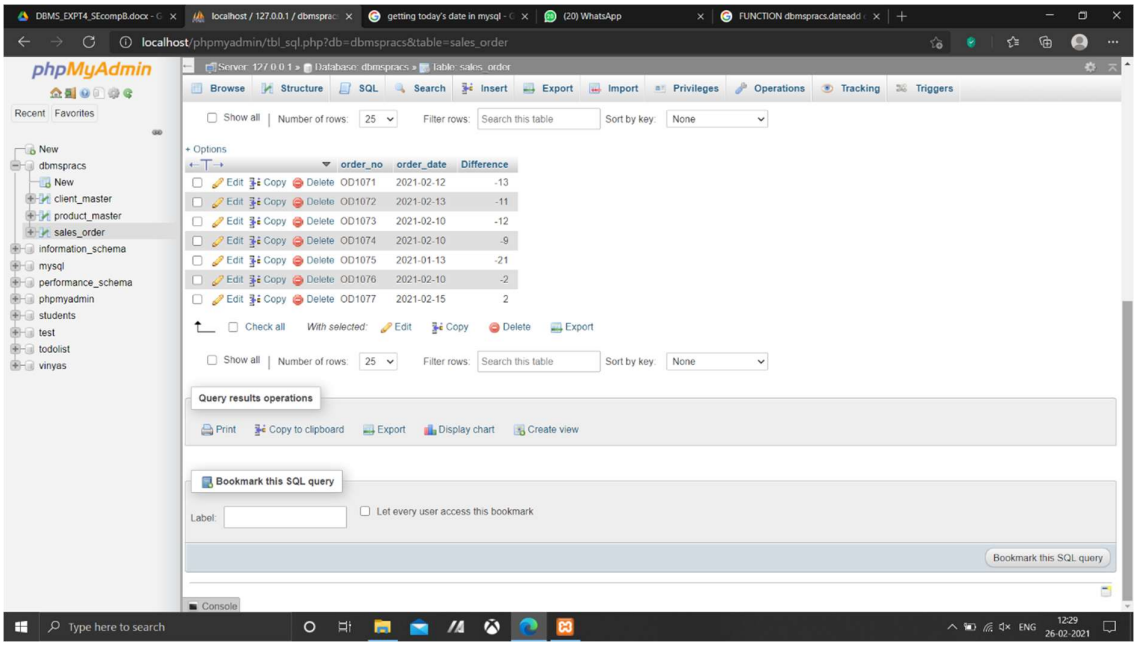


19. Find the no. of days elapsed between today's date and the delivery date of orders placed by the clients.

Query:- SELECT order_no, order_date, DATEDIFF(Dely_date,CURDATE()) AS Difference FROM sales_order;

Screenshots:





Q1 Write a short note on DBA.

Ans (i) Database Administrator manages and controls three levels of database like internal level, conceptual level, and external level of Database management system architecture and in discussion with comprehensive users community gives definition of world view of database. It then provides external view of different users and applications.

(ii) Database Administrator ensures held responsible to maintain integrity and security of database restricting of unauthorized users. It grants permission to users of database and contains profile of each and every user in database.

(iii) Database Administrator also held accountable that database is protected and secured and that any chance of database keeps at minimum

Q2 Write different date functions & Date formats.

Ans

Date Functions.

(i) Date functions are used to manipulate date and data in date format.

(ii) They may or may not take date formats as arguments.

(iii) They come built in with mySQL and can be used from finding current date and time to converting to date & till finding difference between days.

eg:- DATEDIFF(), Date_Format(), DateADD(), etc

Date Formats :-

(i) Date Formats are used to ~~specify~~ specify the format of the date to be returned.

(ii) The default format for MySQL is 'yyyy-mm-dd' In this format data is stored inside the database.

(iii) They are passed as arguments within a function call. They are usually denoted by a percent sign (%) proceeded by a alphabet

eg: %d, %Y, %m, %a etc.

Page No.
Date

Q3 Differentiate between the group by & having clauses

Ans The group by clause:-

(i) The group by clause is used to group the data according to particulars column or row.

(ii) Group by can be used without having clause with the select statement.

(iii) It cannot contain aggregate functions

(iv) It groups the output on basis of some rows & columns.

The Having clause:-

(i) It is used for applying some extra conditions to the query

(ii) Having cannot be used without group by clause

(iii) The having clause can contain aggregate functions

(iv) It restricts the query output by using some condition.

Q4 Give the name of some string functions.

Ans

(i) ASCII()
Returns the ASCII value for the specific character.

(ii) CHAR_LENGTH()
Returns the length of a string (In characters)

(iii) CONCAT()
Adds two or more expressions together

(iv) FIND_IN_SET()
Returns the position of a string within a list of strings

(v) INSERT()
Inserts a string within a string at the specified position and for a certain number of characters.

(vi) LCASE();
Converts a string to to lower-case.

(vii) LENGTH();
Returns the length of a string (in bytes)

(viii) Reverse();
Reverse a string and ~~return~~ returns the result;

(ix) STRCMP();
Compares two strings