SE COMP - B	Roll number :
Experiment no. : 7	Date of Implementation :
Aim : To implement Nested Sub-queries in SQL	
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 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 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 7	Nested subqueries in SQL
Aim	To implement nested sub-queries in SQL
Tools	PostgreSQL/Mysql
Procedure	Use the tables created in the previous experiments and Perform the following queries using nested sub-queries. **Client_master** (client_no, name, address, city, pincode, state, bal_due) **Product_master** (product_no, description, profit_percentage, unit_measure, qty_on_hand, reorder_level, sell_price, cost_price) **Sales_order**(order_no, order_date, client_no, dely_Addr, salesman_no, dely_type, billed_yn, dely_date, order_status) **Sales_order_details**(order_no, product_no, qty_ordered, qty_disp, product_rate) 1. Find the product no. and description of non-moving products i.e. products not being sold. 2. Find the customer name, address for the client who has placed order no 'O191' 3. Find the clients names who have placed orders before the month of May'96 4. Find out if the product '1.44 Drive' has been ordered by any client and print the client_no, name to whom it was sold 5. Find the names of clients who have placed orders worth Rs. 10000 or more 6. Retrieve all the orders placed by a client named 'Rahul Desai' from the sales_order table. 7. Retrieve name, address, city of all the clients who have placed an order through salesman no 's001'.
Post Lab Questions:	1. What is incremental Update? 2. Explain is use of **on delete cascade** and **on update cascade** with suitable example?

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

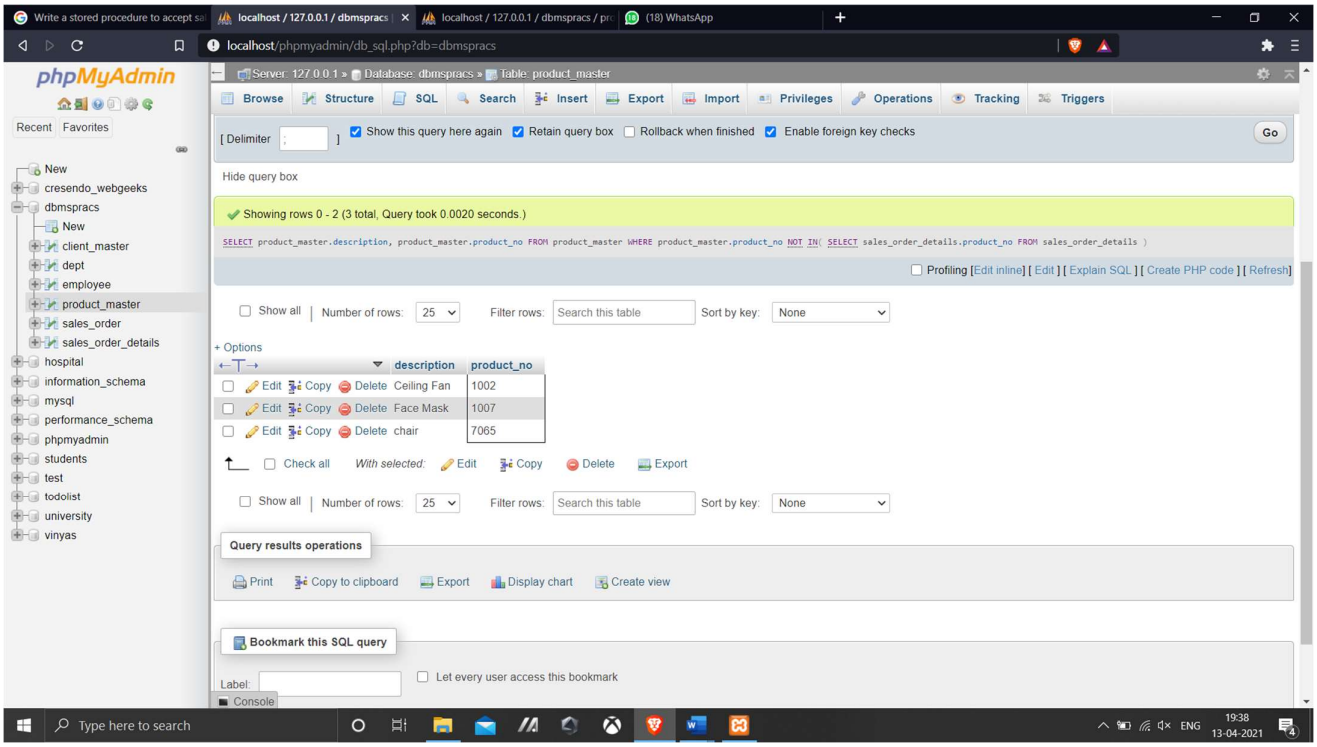
DBMS Practical Implementation, Lab 7.

0. Inserting Necessary Values:

```
INSERT INTO `client_master` (`client_no`, `name`, `address`, `city`, `pincode`, `state`, `Bal_due`) VALUES ('C0004', 'Rahul Desai', 'The Height,Indra Nagar', 'Mumbai', '400080', 'Maharashtra', '2000');

INSERT INTO `product_master` (`product_no`, `description`, `profit_percent`, `unit_measure`, `qty_on_hand`, `reorder_level`, `cost_price`, `new_price`) VALUES ('1014', '1.44 Drive', '12', '1', '250', '500', '2500', '3000');

INSERT INTO `sales_order` (`order_no`, `order_date`, `client_no`, `dely_addr`, `salesman_no`, `dely_type`, `billed_yn`, `Dely_date`, `order_status`) VALUES ('O191', '1995-04-01', 'C0004', 'The height, Indra Nagar', 'S001', 'Y', 'Y', '1995-04-14', 'Shipped');

INSERT INTO `sales_order_details` (`order_no`, `product_no`, `qty_ordered`, `qty_disp`, `product_rate`) VALUES ('O191', '1014', '25', '25', '3000');
```

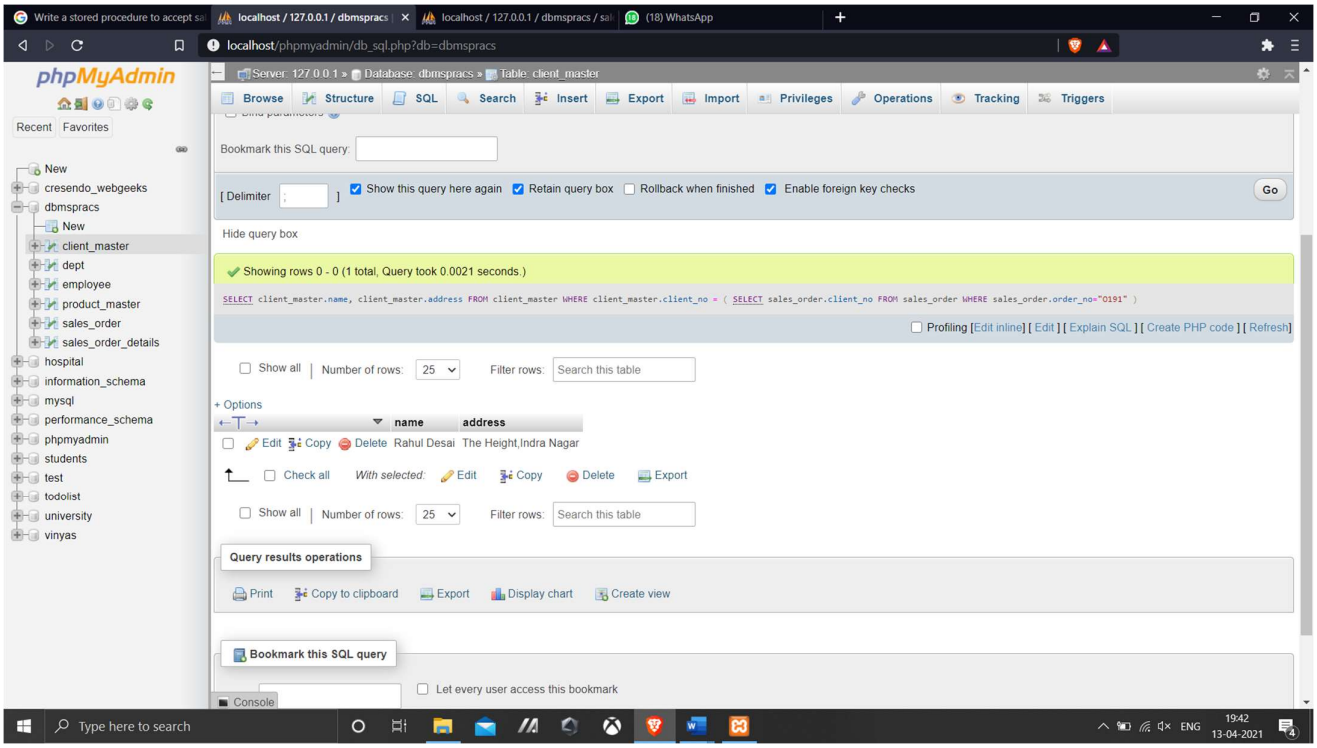
1. Find the product no. and description of non-moving products i.e. products not being sold.

```
SELECT product_master.description, product_master.product_no
FROM product_master
WHERE product_master.product_no NOT IN(
SELECT sales_order_details.product_no FROM sales_order_details
);
```



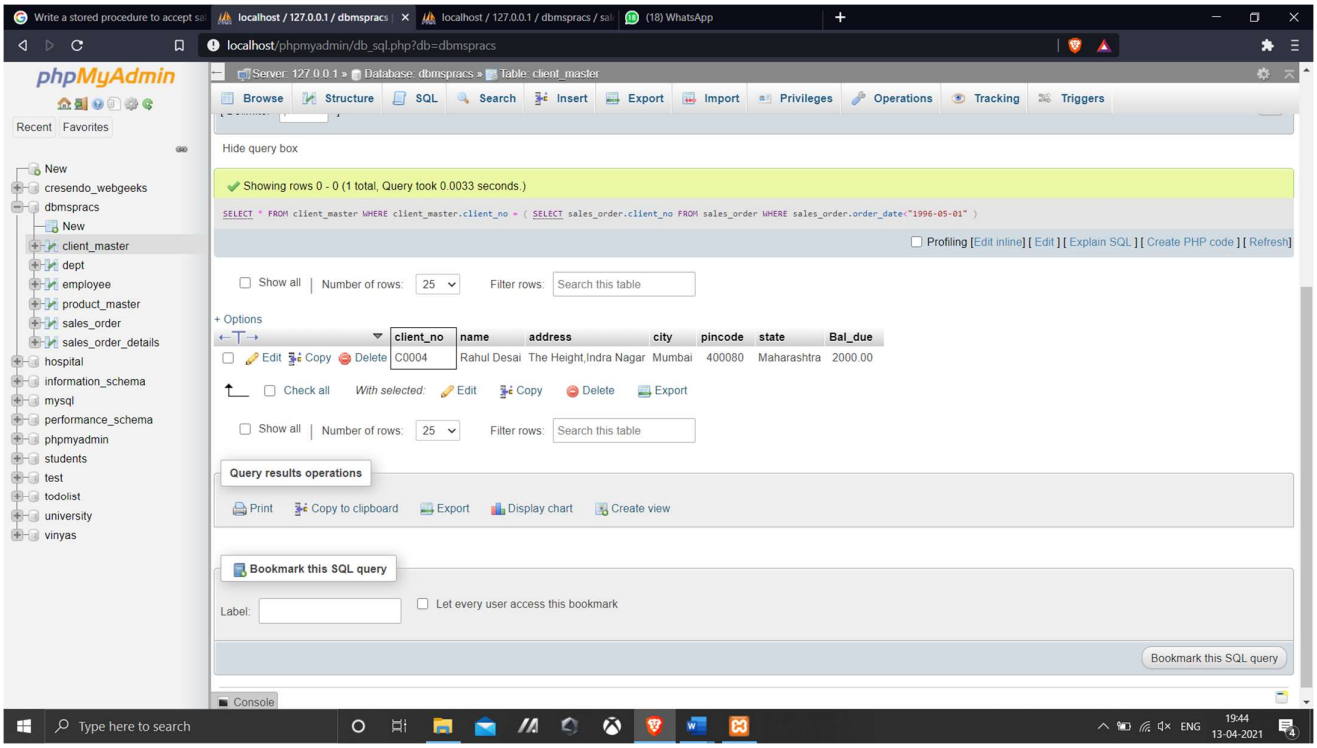
2. Find the customer name, address for the client who has placed order no ‘O191’

```
SELECT client_master.name, client_master.address
FROM client_master
WHERE client_master.client_no = (
SELECT sales_order.client_no
    FROM sales_order
    WHERE sales_order.order_no="O191"
);
```



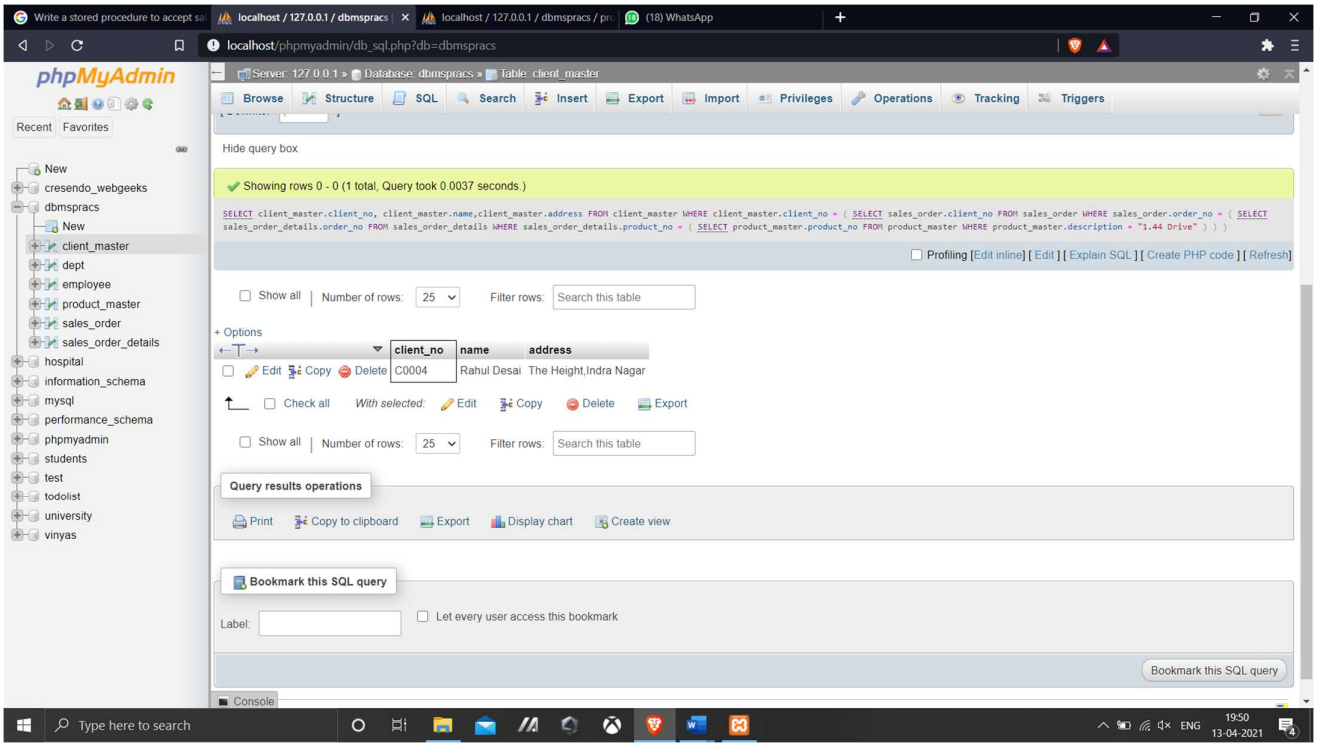
3. Find the clients names who have placed orders before the month of May’96

```
SELECT *
FROM client_master
WHERE client_master.client_no = (
SELECT sales_order.client_no
    FROM sales_order
    WHERE sales_order.order_date<"1996-05-01"
);
```



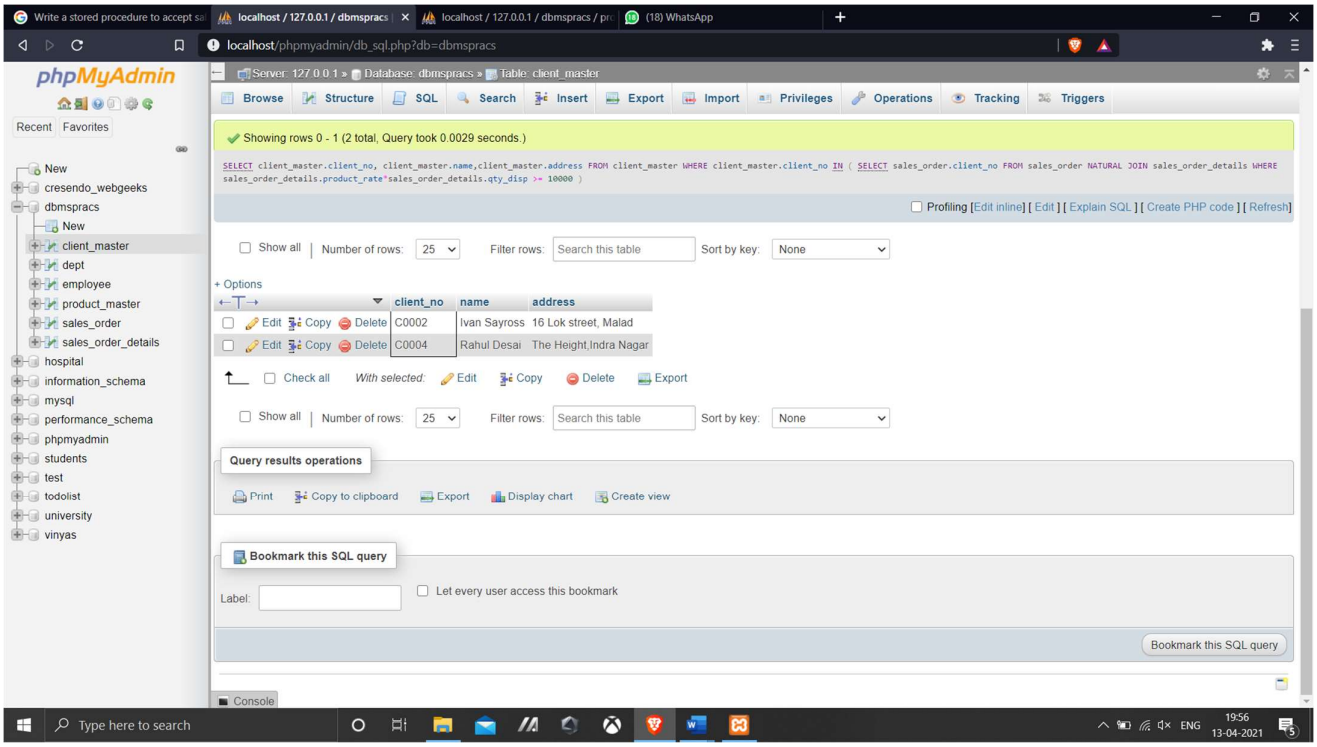
4. Find out if the product ‘1.44 Drive’ has been ordered by any client and print the client_no, name to whom it was sold

```
SELECT client_master.client_no, client_master.name,client_master.address
FROM client_master WHERE client_master.client_no = (
        SELECT sales_order.client_no FROM sales_order WHERE
sales_order.order_no = (
        SELECT sales_order_details.order_no FROM sales_order_details WHERE
sales_order_details.product_no = (
                SELECT product_master.product_no FROM product_master
WHERE product_master.description = "1.44 Drive"
        )
    )
);
```



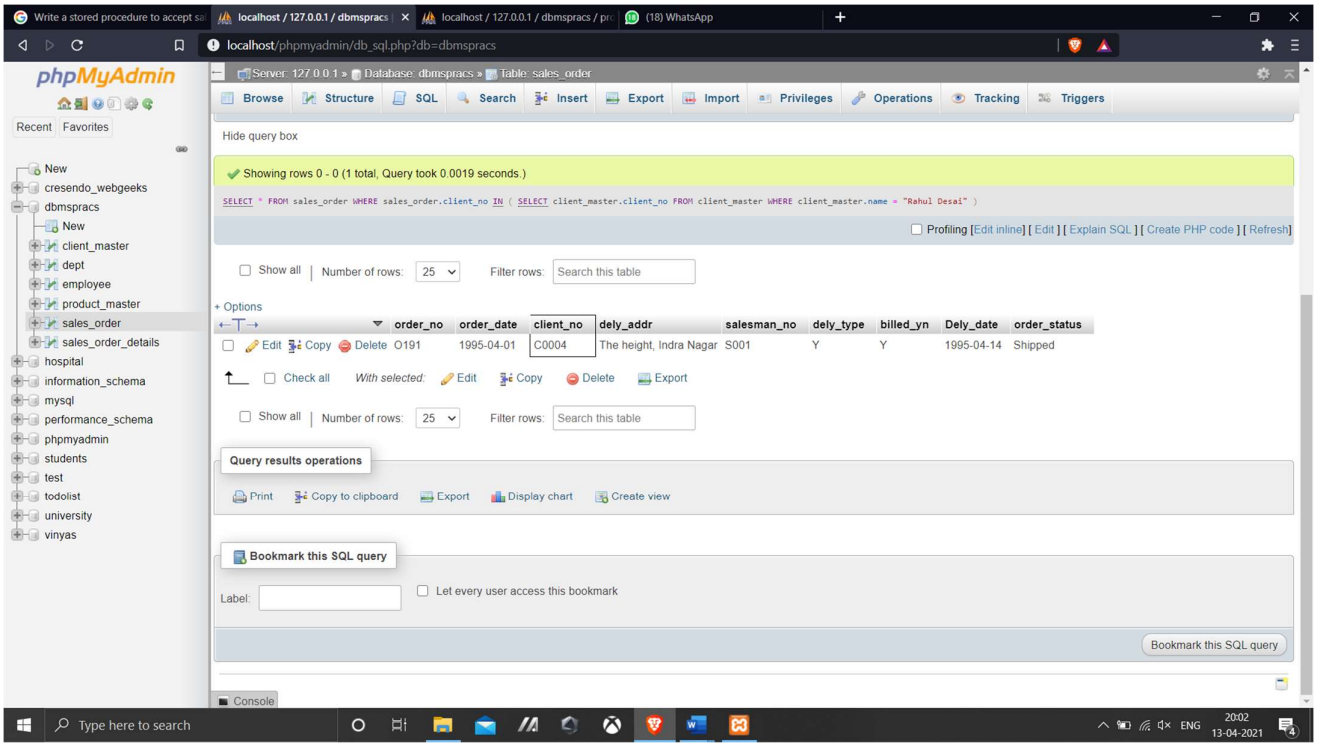
5. Find the names of clients who have placed orders worth Rs. 10000 or more

```
SELECT client_master.client_no, client_master.name,client_master.address
FROM client_master
WHERE client_master.client_no IN  (
        SELECT sales_order.client_no
    FROM sales_order NATURAL JOIN sales_order_details
    WHERE sales_order_details.product_rate*sales_order_details.qty_disp >=
10000
);
```



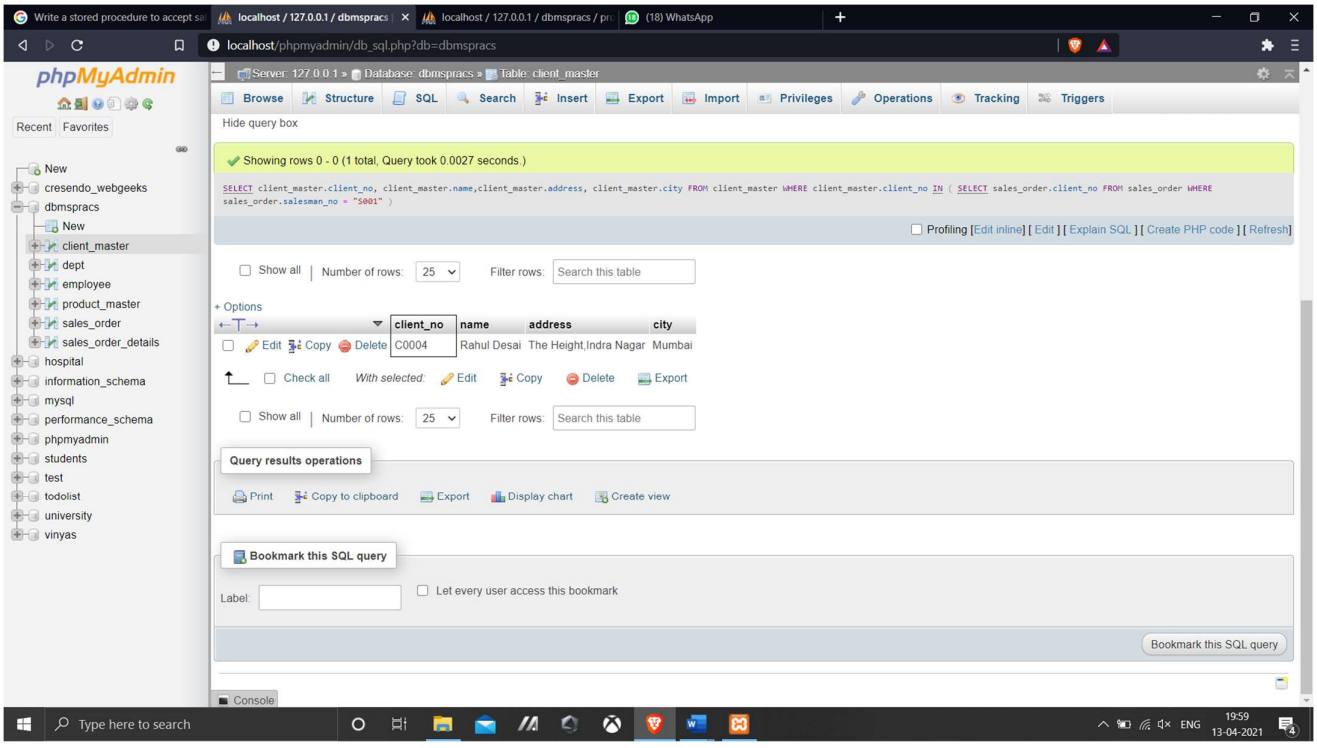
6. Retrieve all the orders placed by a client named ‘Rahul Desai’ from the sales_order table.

```
SELECT *
FROM sales_order
WHERE sales_order.client_no IN  (
        SELECT client_master.client_no
        FROM client_master
        WHERE client_master.name = "Rahul Desai"
);
```



7. Retrieve name, address, city of all the clients who have placed an order through salesman no ‘s001’.

```
SELECT client_master.client_no, client_master.name,client_master.address,
client_master.city
FROM client_master
WHERE client_master.client_no IN  (
        SELECT sales_order.client_no
    FROM sales_order
    WHERE sales_order.salesman_no = "S001"
);
```



Brendon Lucas
8463
SE Comps

Exp-7 - Portlet

Q 1) What is Incremental Updates?

Ans

An incremental update adds new records to a project data set from Hive table

It performs a partial update of a project data set by selecting adding new and modified records. The data set should be a project data set.

The Incremental update operation fetches a subset of the records in the Source Hive table. The subset is determined by using a filtering predicate that specifies the Hive table columns that holds the records and value of the records to fetch. The records in the subset batch are ingested as follows:

- If a record is brand new (does not exist in the data set), it is added to the data set.

- If a record already exists in the data set but its content has been changed, it replaces the record in the data set.

(2) What is UPDATE Cascade & DELETE Cascade with suitable example?

Ans

DELETE Cascade:- When we create a foreign key using this option, it deletes the referencing rows in the child table when the referenced row is deleted in the parent table which has a primary key.

UPDATE Cascade:- When we create a foreign key using UPDATE CASCADE the referencing rows are updated in the child table when the referenced row is updated in the parent table which has a primary key.

Example -

Suppose that we have two tables:- buildings and rooms. In this database model, each buildings has one or many rooms. However each room belongs to one only building. A room would not exist without a building.

When you delete a row from the buildings table, you also want to delete all rows in the rooms table that references to the rows in the buildings table. For example, when we delete a row with building no. 1 in the building table, we also want the rows in the rooms table that refers to building no 2. will be also removed.

Syntax :-

```
CREATE TABLE rooms (
    room_no INT PRIMARY AUTO_INCREMENT,
    room_name VARCHAR(255) NOT NULL,
    building_no INT NOT NULL,

    FOREIGN KEY (building_no)
    REFERENCES buildings (building_no)
    ON DELETE CASCADE
);
```