Practical 16

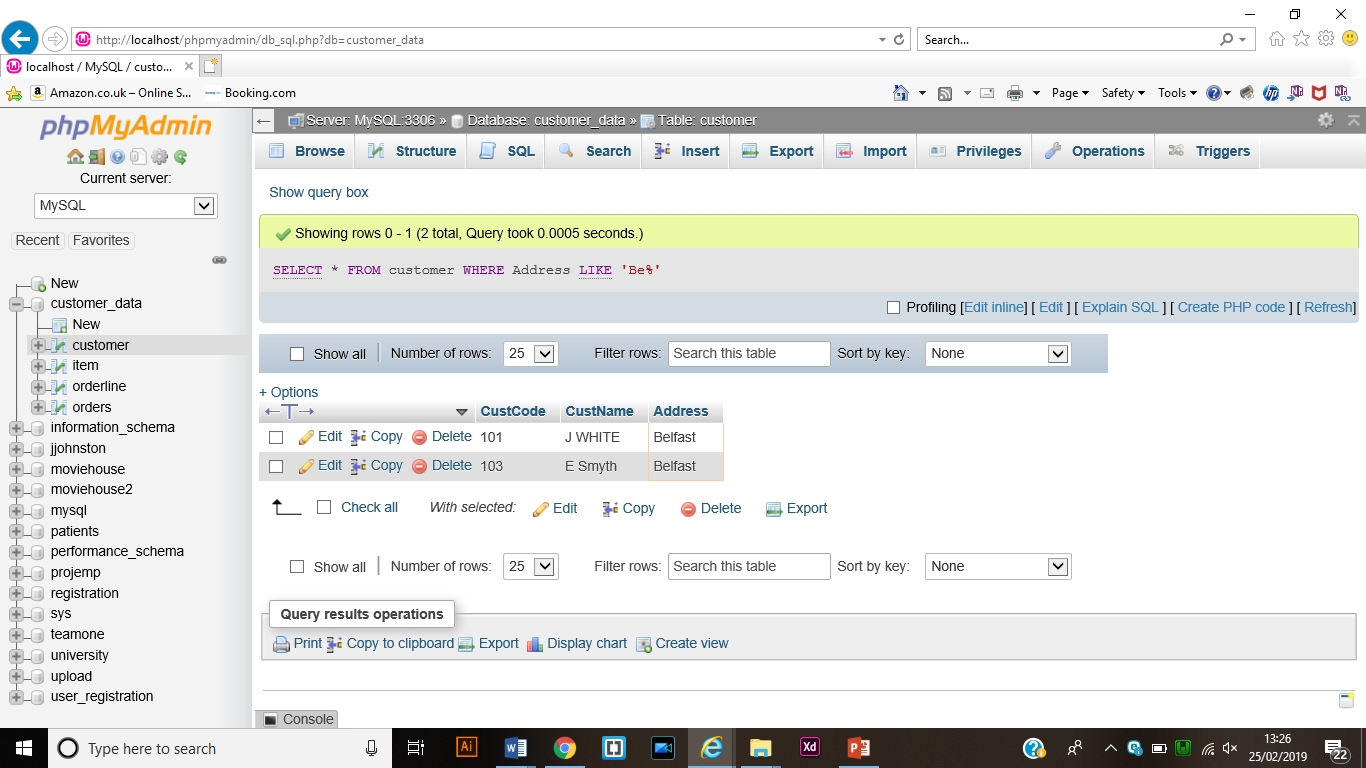
1. Get a list of all customers with their address starting with BE.

Here is the coding written into the SQL window within PHP my Admin.

SELECT \* FROM customer

WHERE Address LIKE 'Be%';

Results of the Query within php my Admin

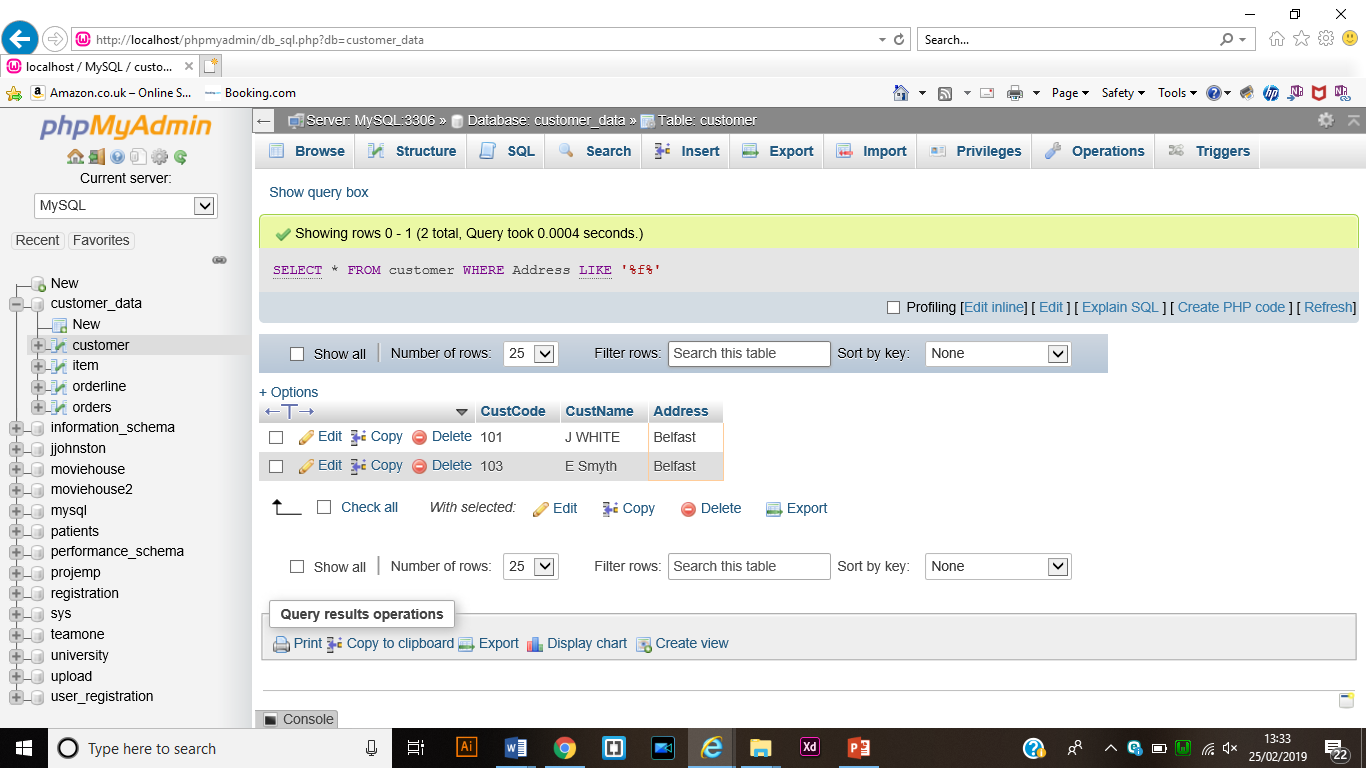


1. Get a list of all customers with their address within any characters followed by f and then by any characters.

This shows the SQL Query source code in PHPMyAdmin

SELECT \* FROM customer

WHERE Address LIKE '%f%';

Result from the SQL Query.

1. Create your own scenarios that you can use Like operator and Wildcards.

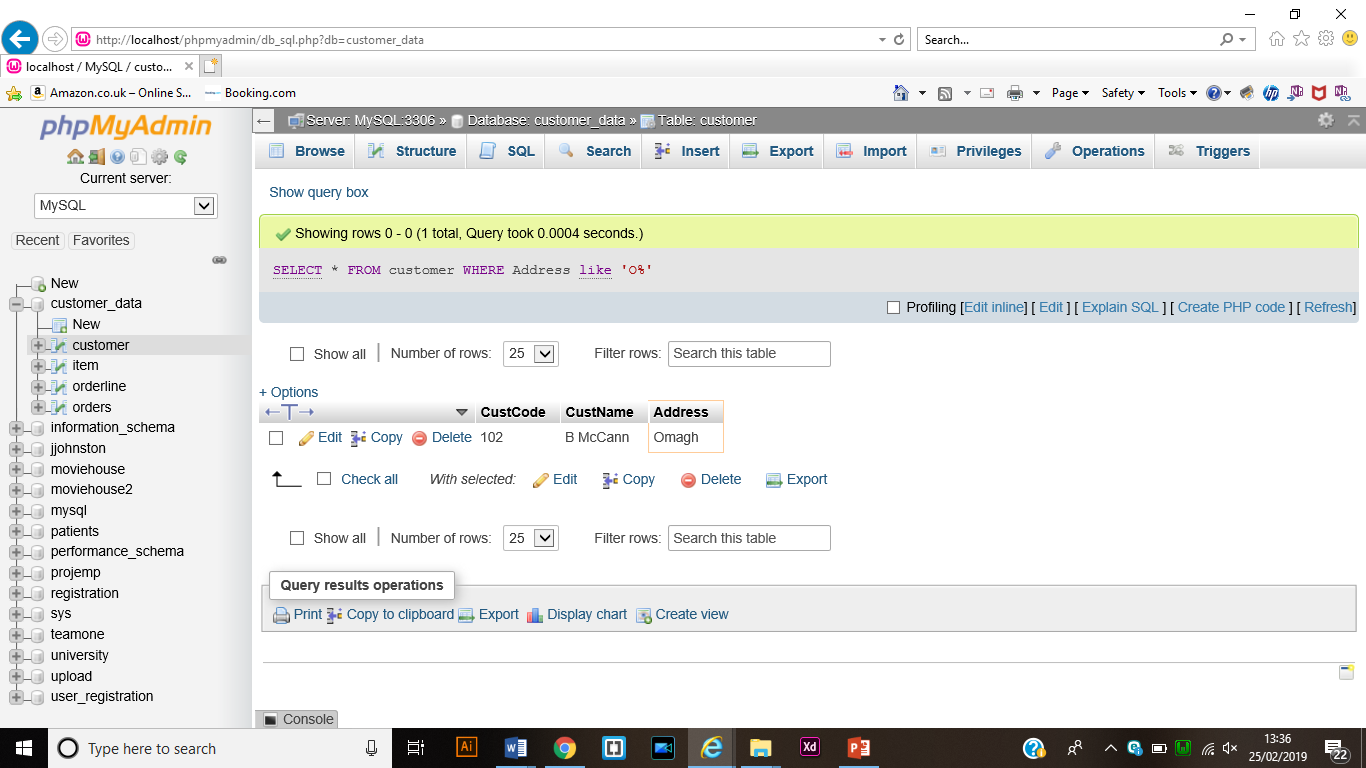
Return customer information for customers whos address begins with O

SQL Sourcecode.

SELECT \* FROM customer

WHERE Address like 'O%';

Result when query is run



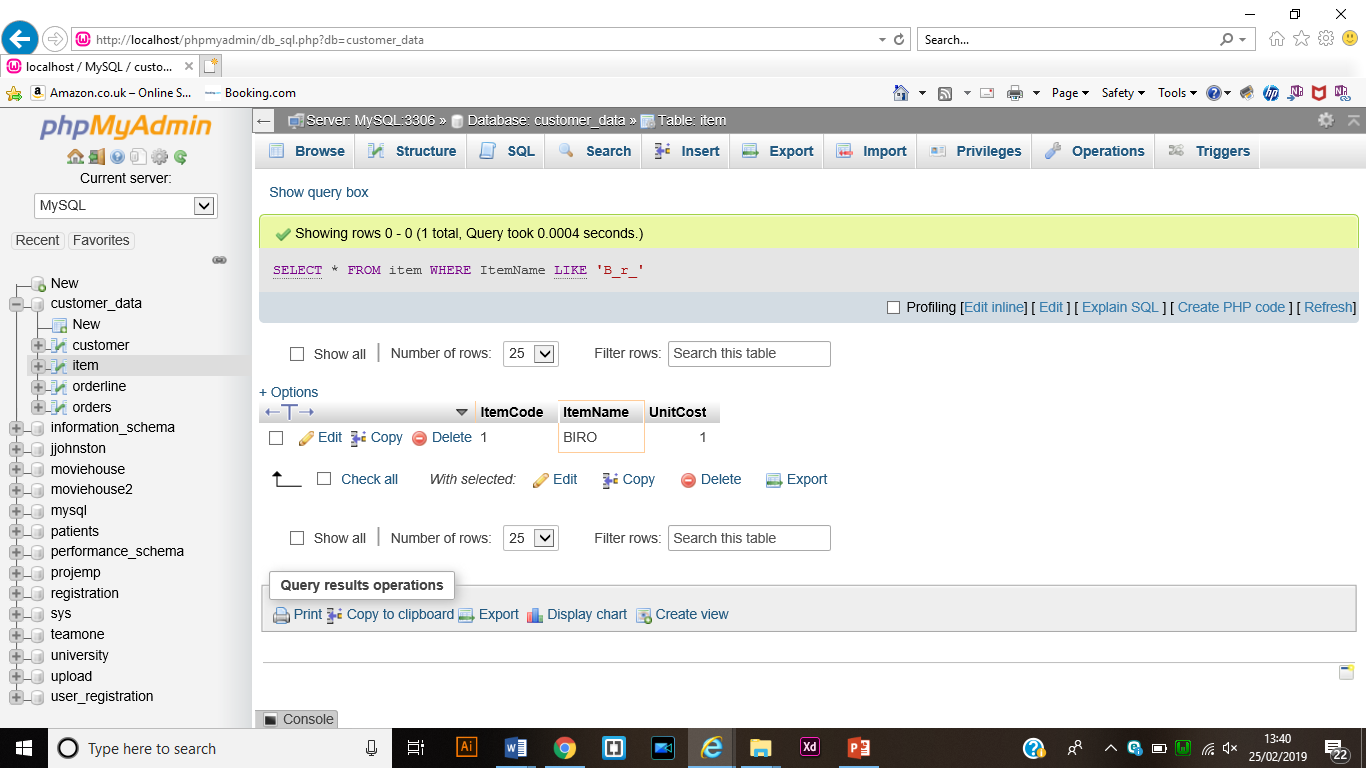
Return all from Item Table where itemname is b\_r\_

SQL Source Code

SELECT \* FROM item

WHERE ItemName LIKE 'B\_r\_';

Result from the query being run



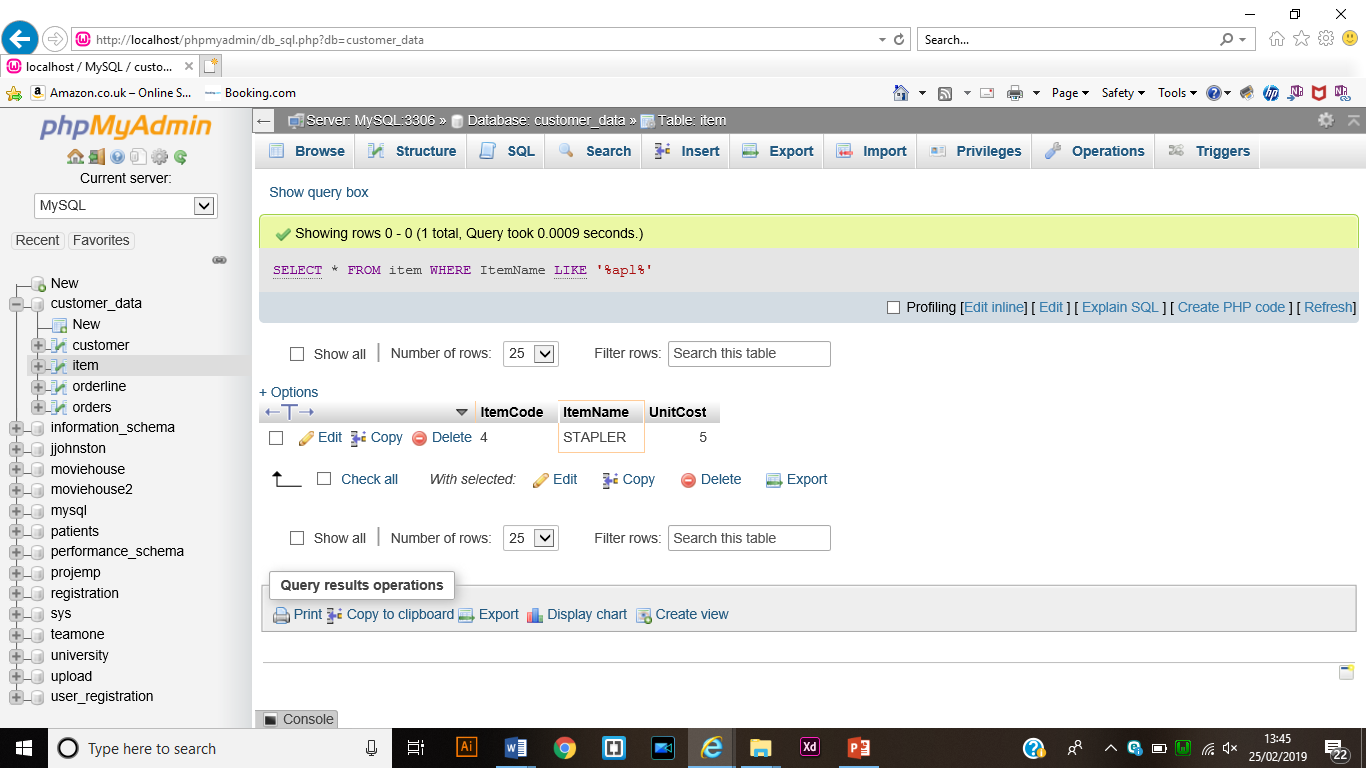
Return item detail where ItemName contains APL

SQL Source Code

SELECT \* FROM item

WHERE ItemName LIKE '%apl%';

Result



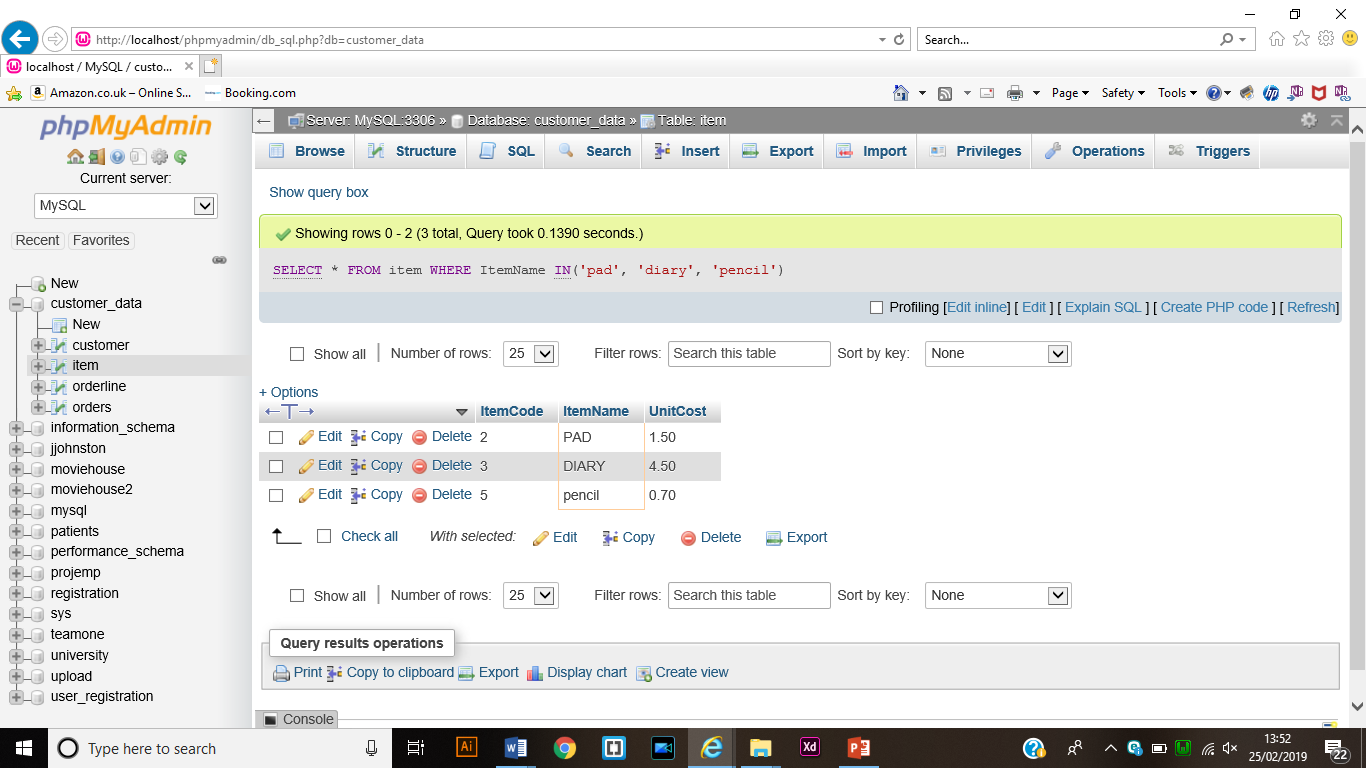
1. Get All items information for item pad, diary or pencil

SQL Source Code

SELECT \* FROM item

WHERE ItemName IN('pad', 'diary', 'pencil');

Results of the SQL Query



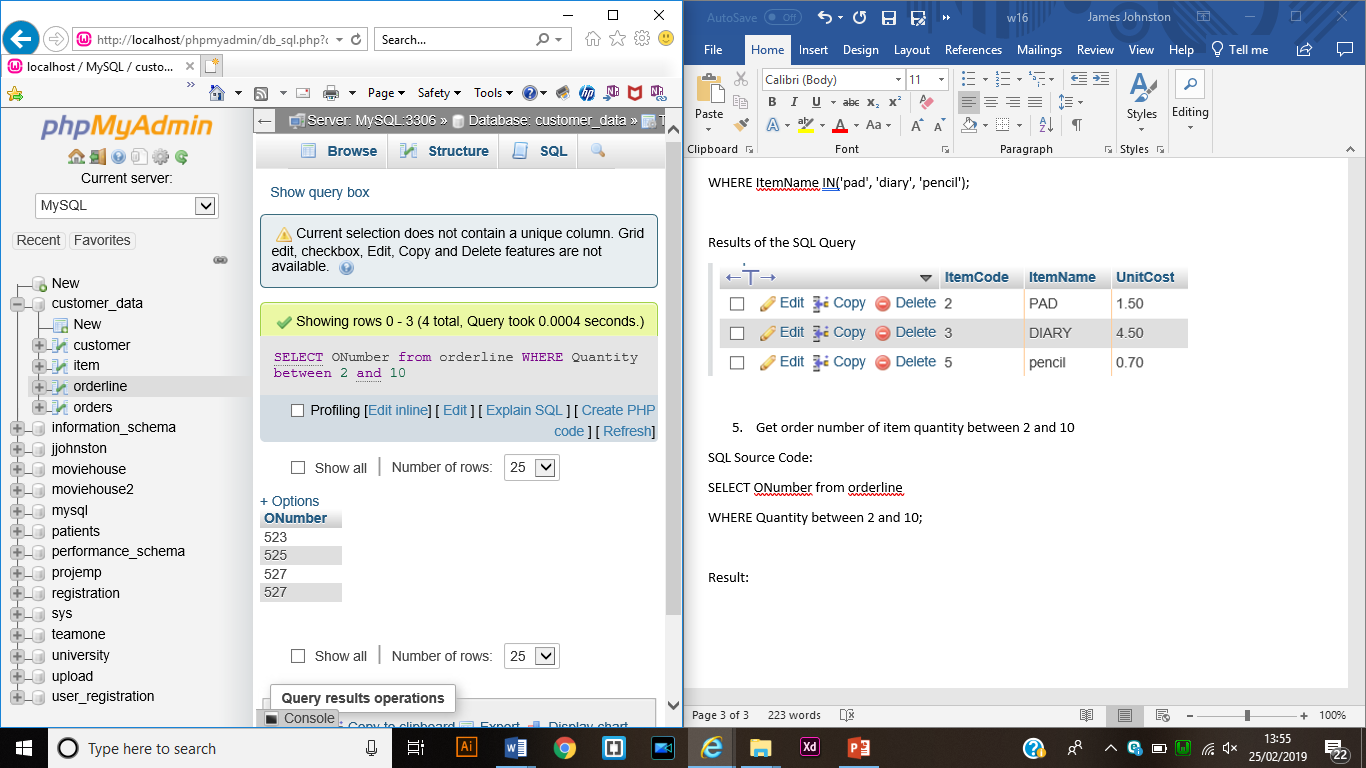
1. Get order number of item quantity between 2 and 10

SQL Source Code:

SELECT ONumber from orderline

WHERE Quantity between 2 and 10;

Result:



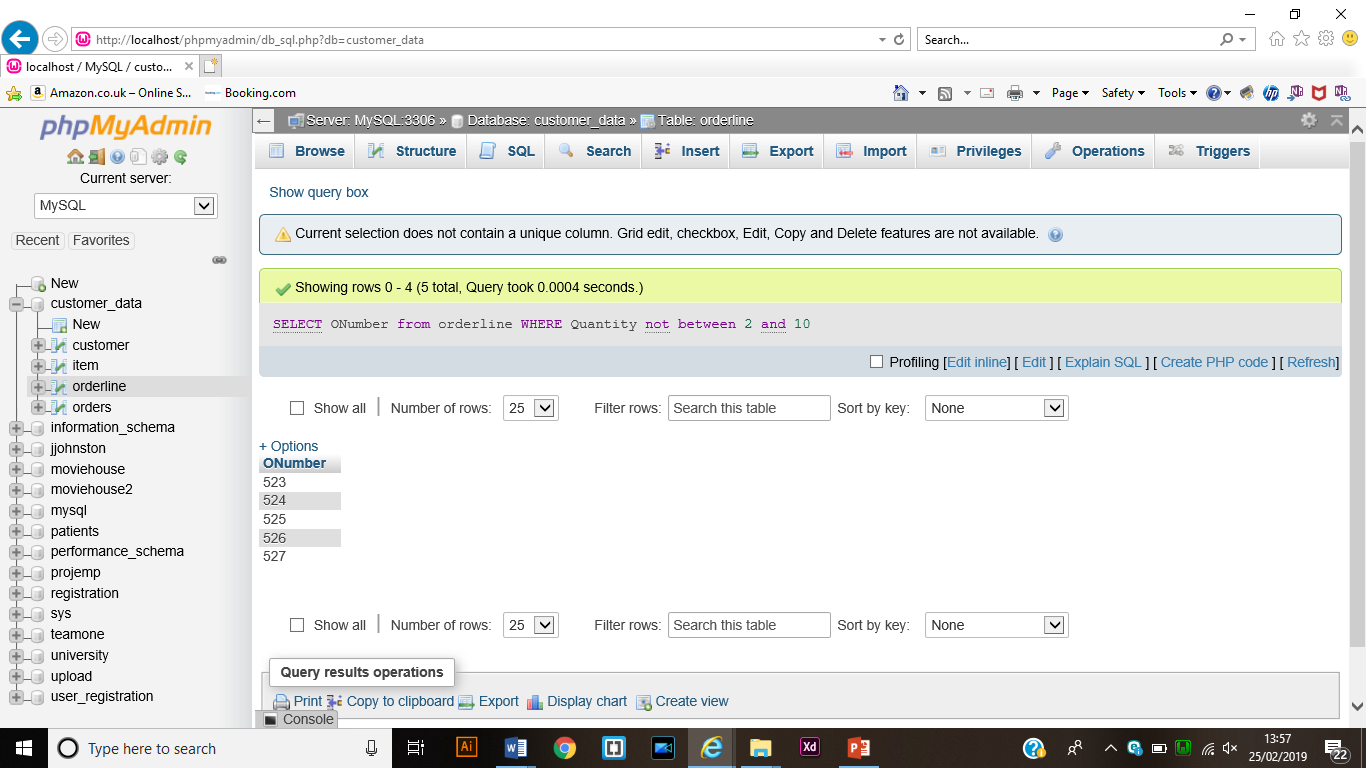
1. Get Order number of item quantity not between 2 and 10

SQL SOURCE CODE:

SELECT ONumber from orderline

WHERE Quantity not between 2 and 10;

Results



1. Get all items information if the items have been ordered

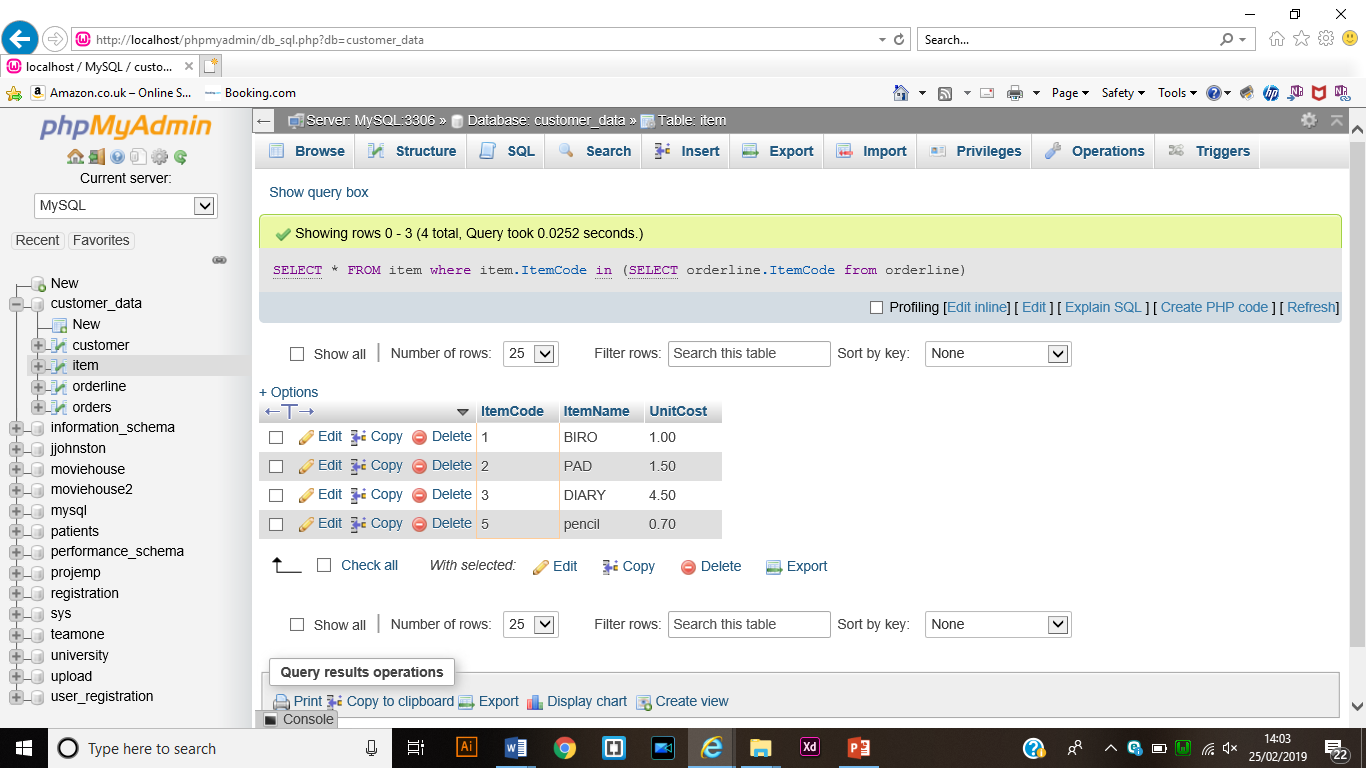
SQL Source Code:

SELECT \* FROM item

where item.ItemCode in

(SELECT orderline.ItemCode from orderline);

Results



1. Get all item’s information if the items have not been ordered

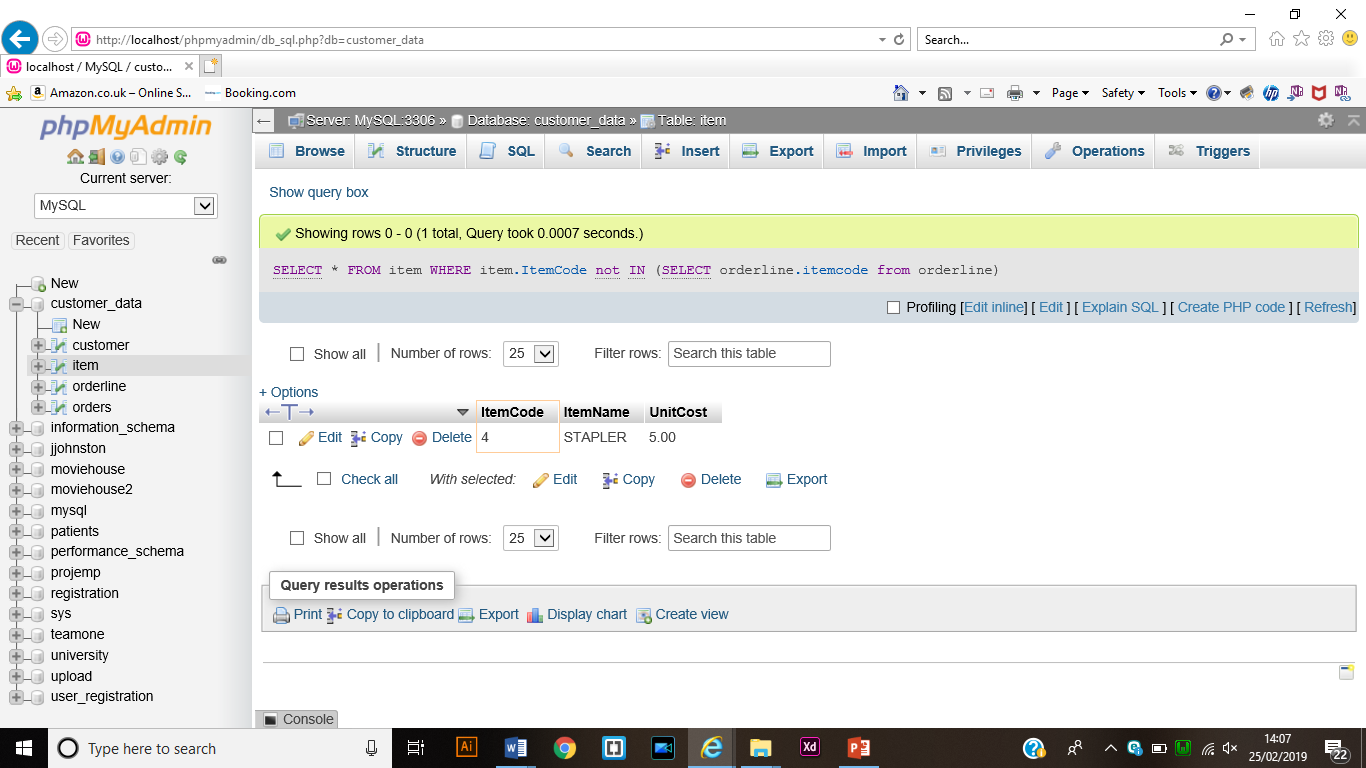
SQL Source Code

SELECT \* FROM item

WHERE item.ItemCode not IN

(SELECT orderline.itemcode from orderline);

Result



1. Try the examples from page 11 to 22

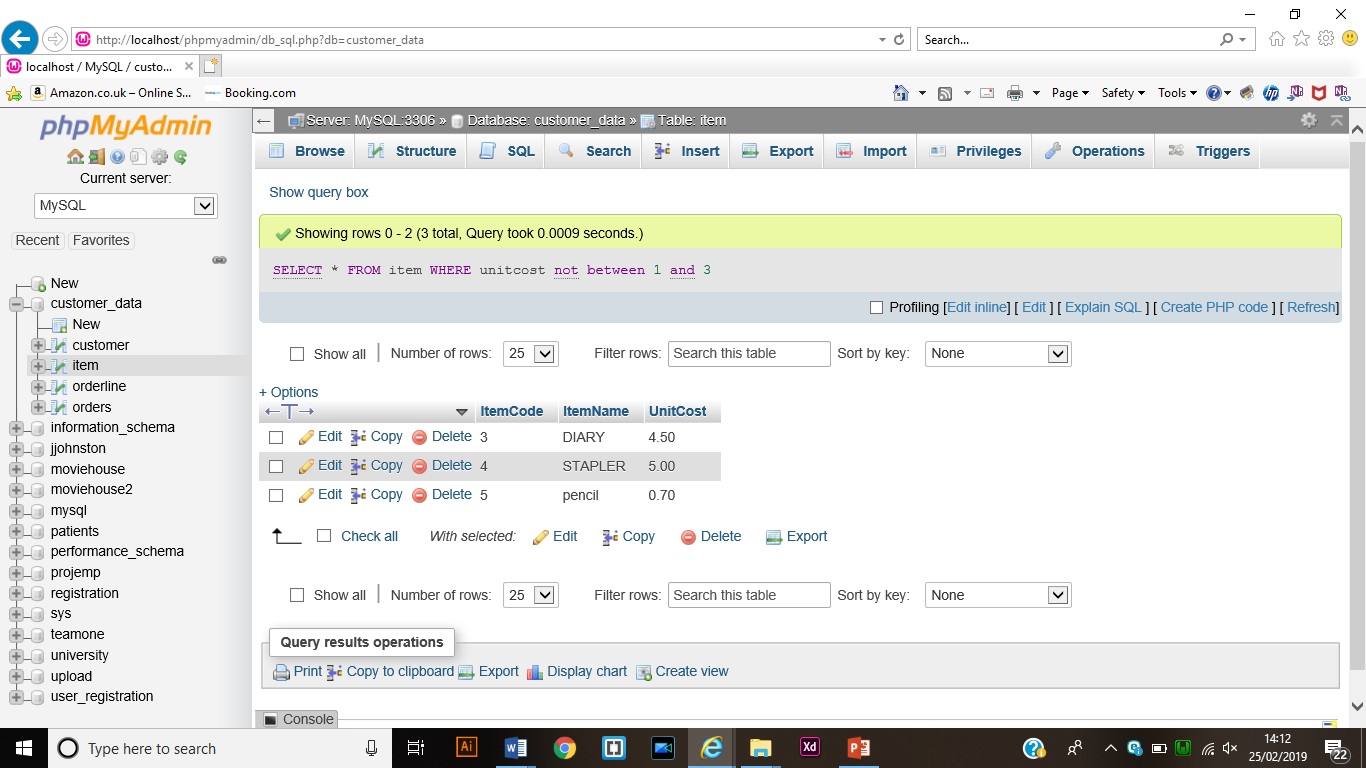
Page 11

Source Code

SELECT \* FROM item

WHERE unitcost not between 1 and 3;

Result



Page 12

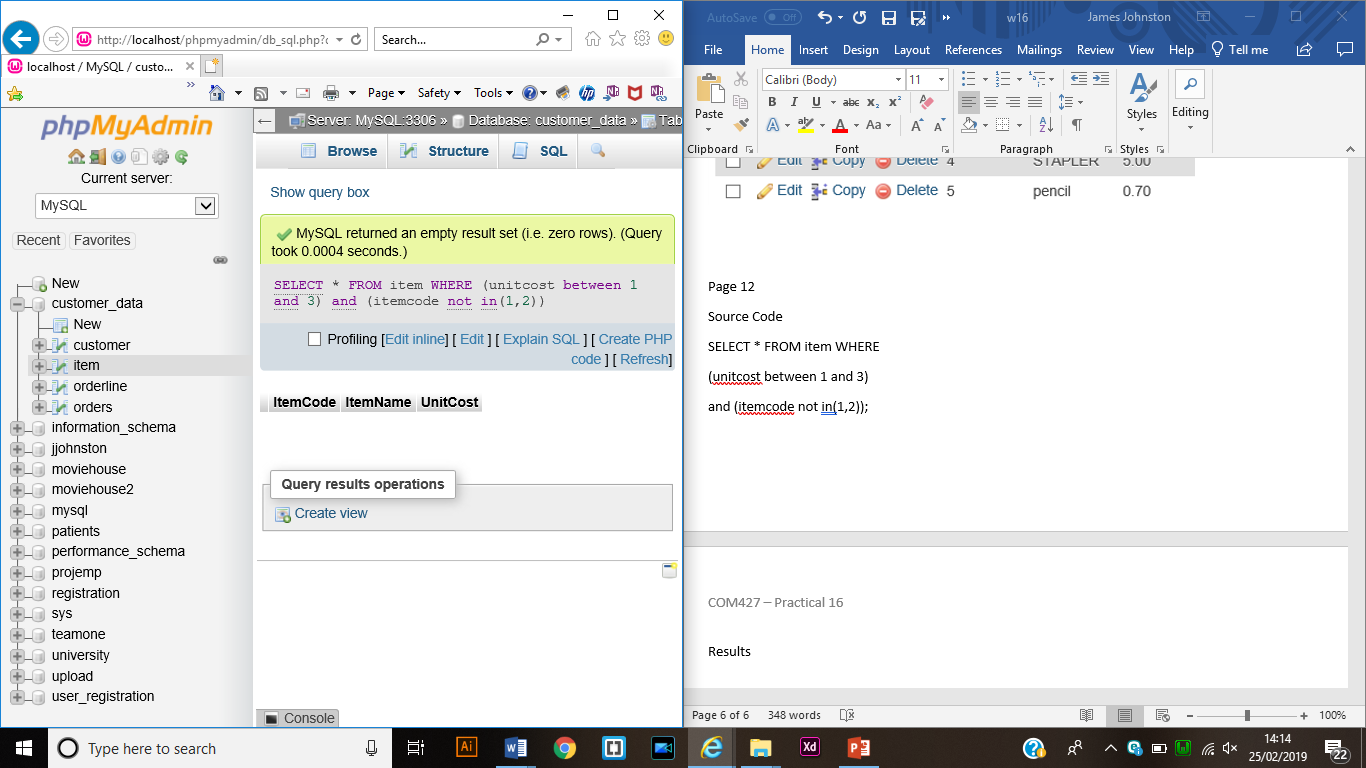
Source Code

SELECT \* FROM item WHERE

(unitcost between 1 and 3)

and (itemcode not in(1,2));

Results



Page 13

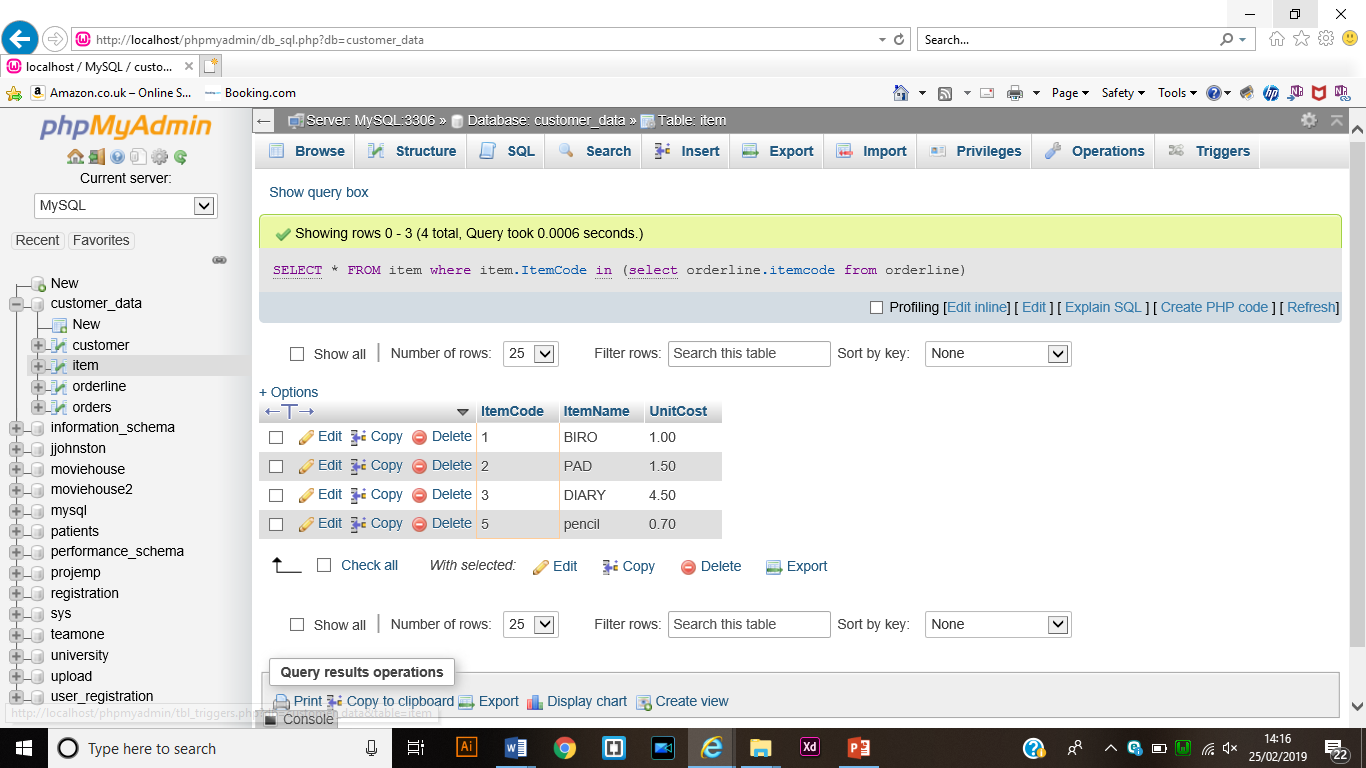
Source Code

SELECT \* FROM item

where item.ItemCode in

(select orderline.itemcode from orderline);

Result



Page 14

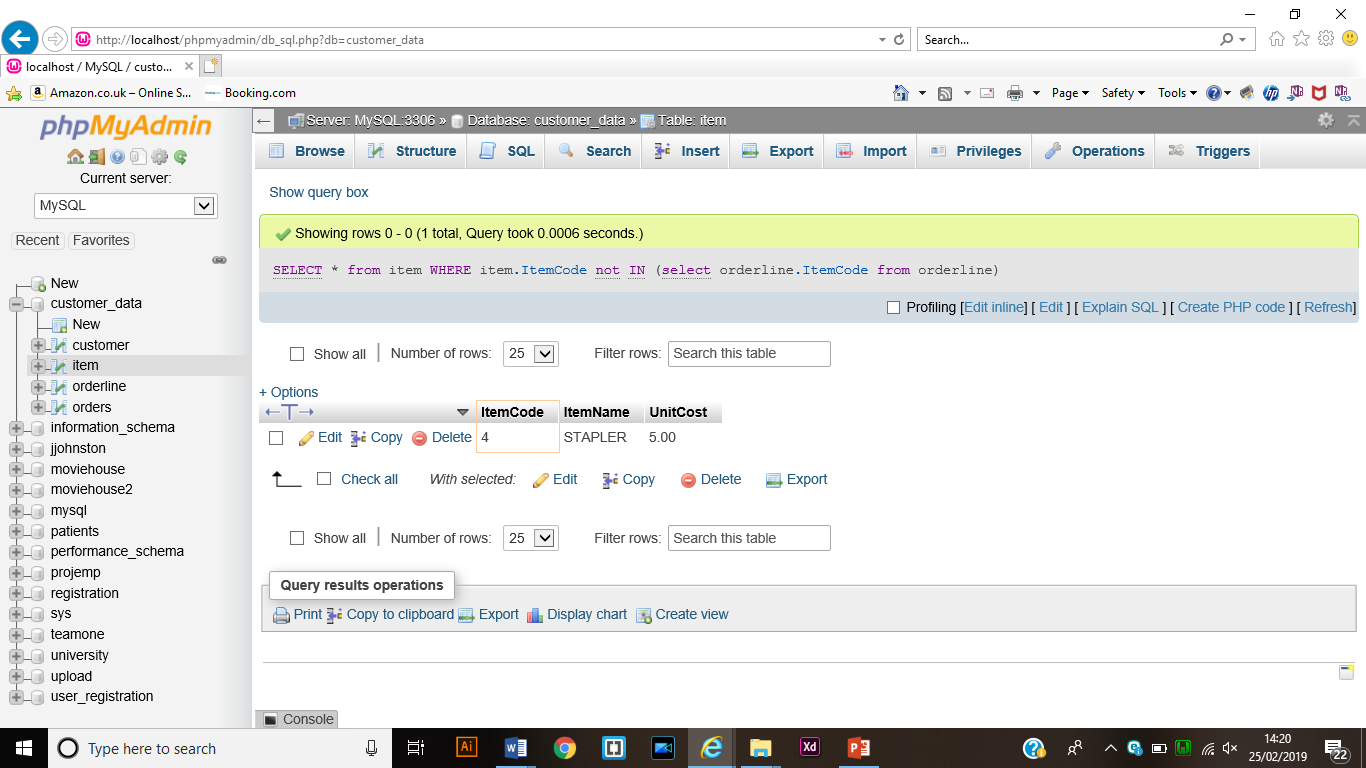
Source Code

SELECT \* from item

WHERE item.ItemCode not IN

(select orderline.ItemCode from orderline);

Result



Page 15

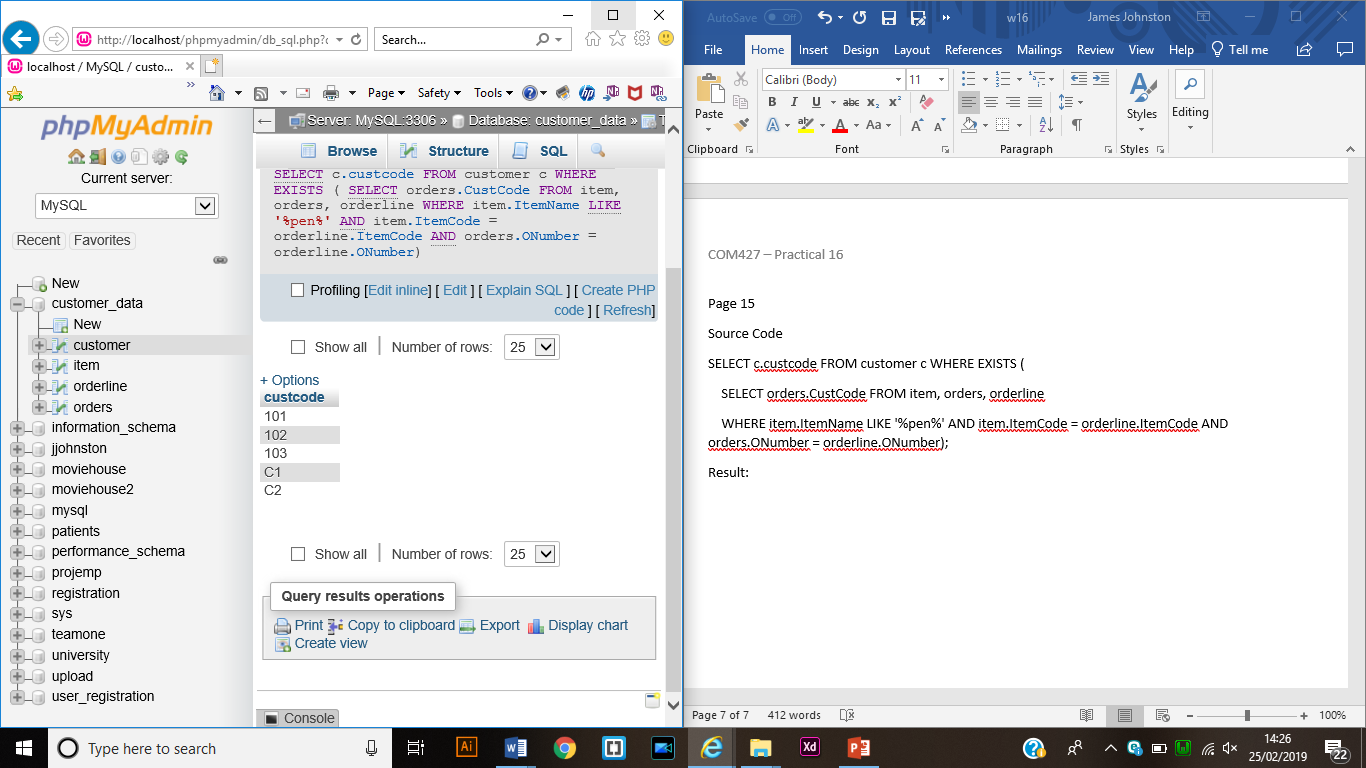
Source Code

SELECT c.custcode FROM customer c WHERE EXISTS (

SELECT orders.CustCode FROM item, orders, orderline

WHERE item.ItemName LIKE '%pen%' AND item.ItemCode = orderline.ItemCode AND orders.ONumber = orderline.ONumber);

Result:



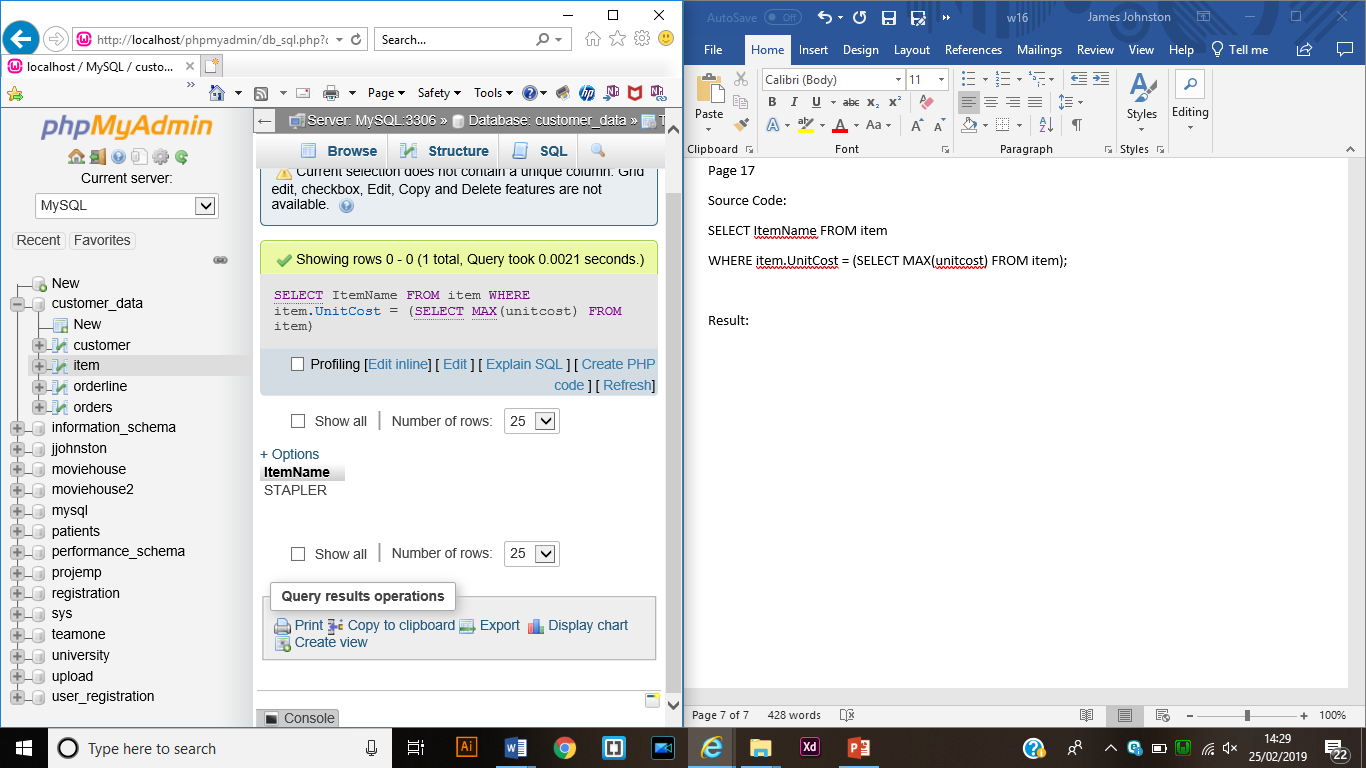
Page 17

Source Code:

SELECT ItemName FROM item

WHERE item.UnitCost = (SELECT MAX(unitcost) FROM item);

Result:



Page 18

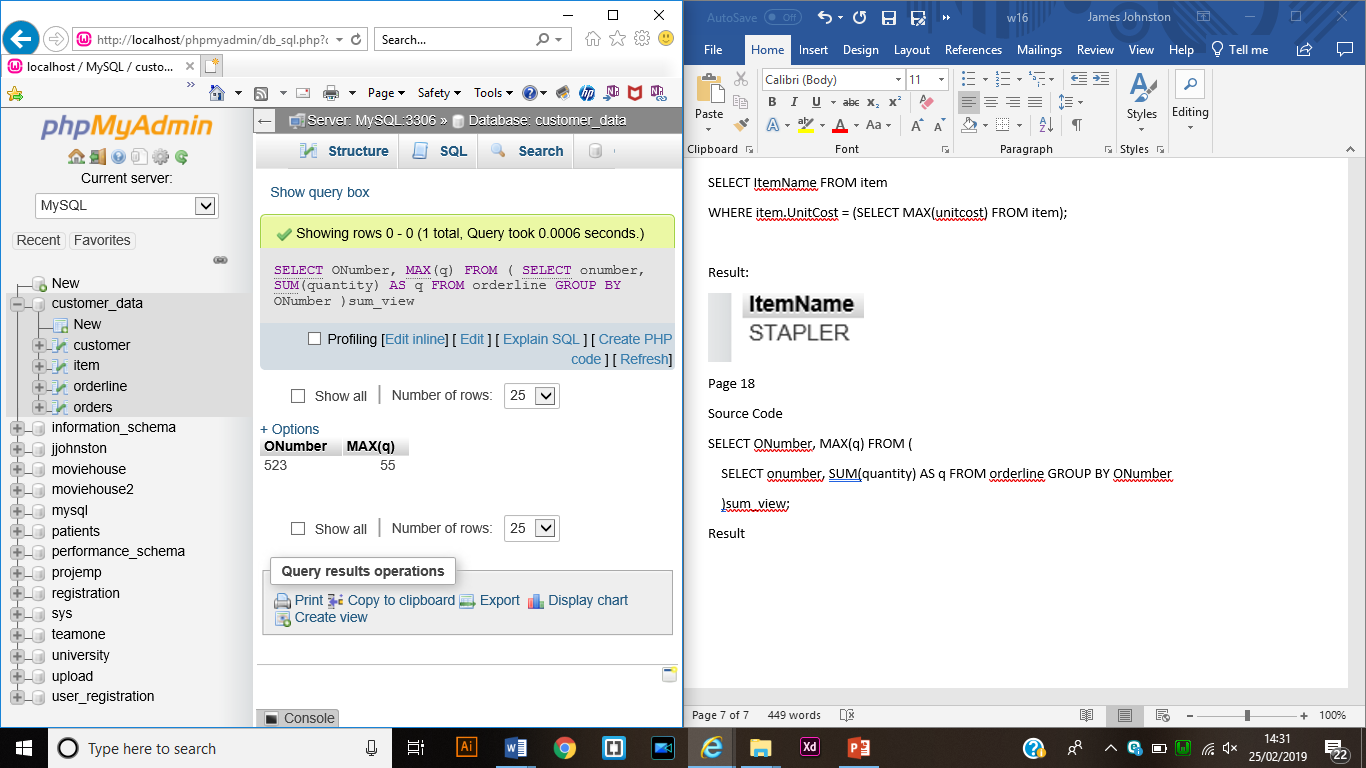
Source Code

SELECT ONumber, MAX(q) FROM (

SELECT onumber, SUM(quantity) AS q FROM orderline GROUP BY ONumber

)sum\_view;

Result



Page 21

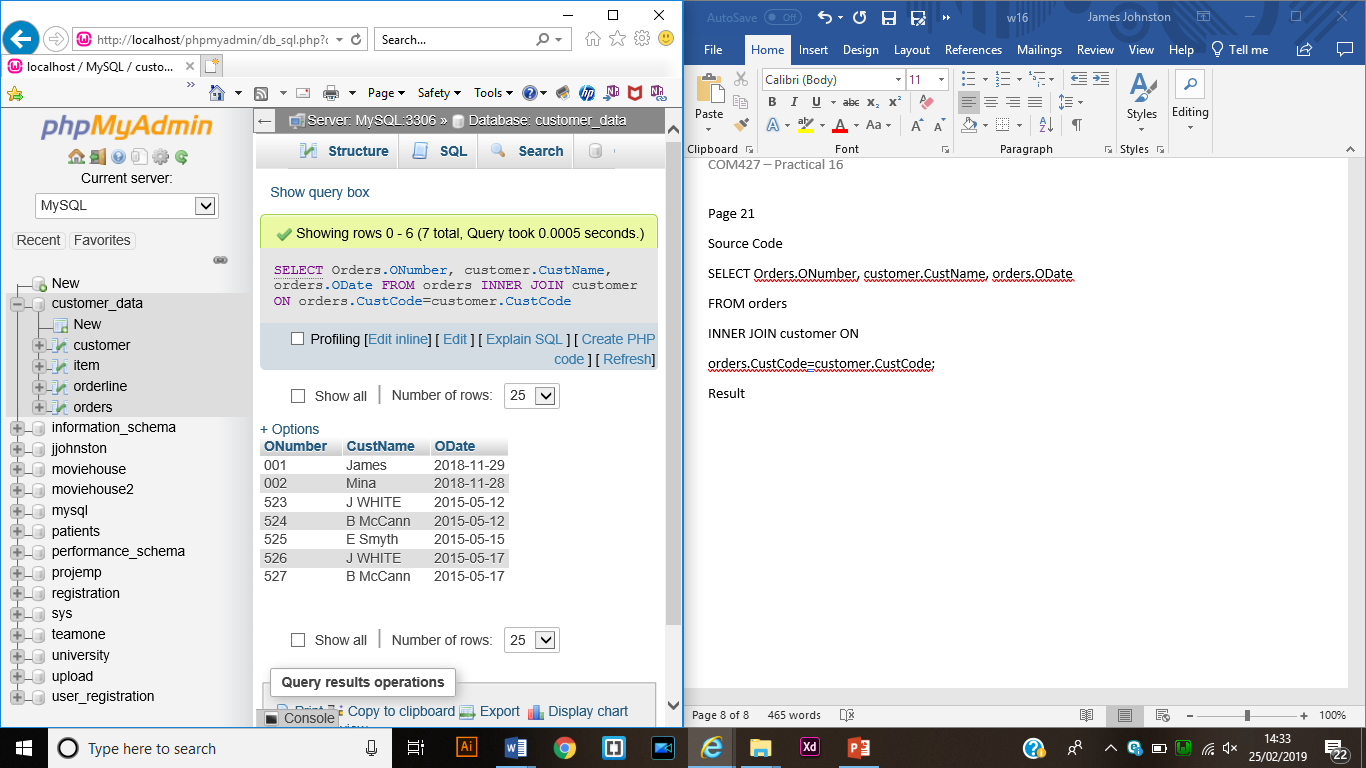
Source Code

SELECT Orders.ONumber, customer.CustName, orders.ODate

FROM orders

INNER JOIN customer ON

orders.CustCode=customer.CustCode;

Result

Page 22

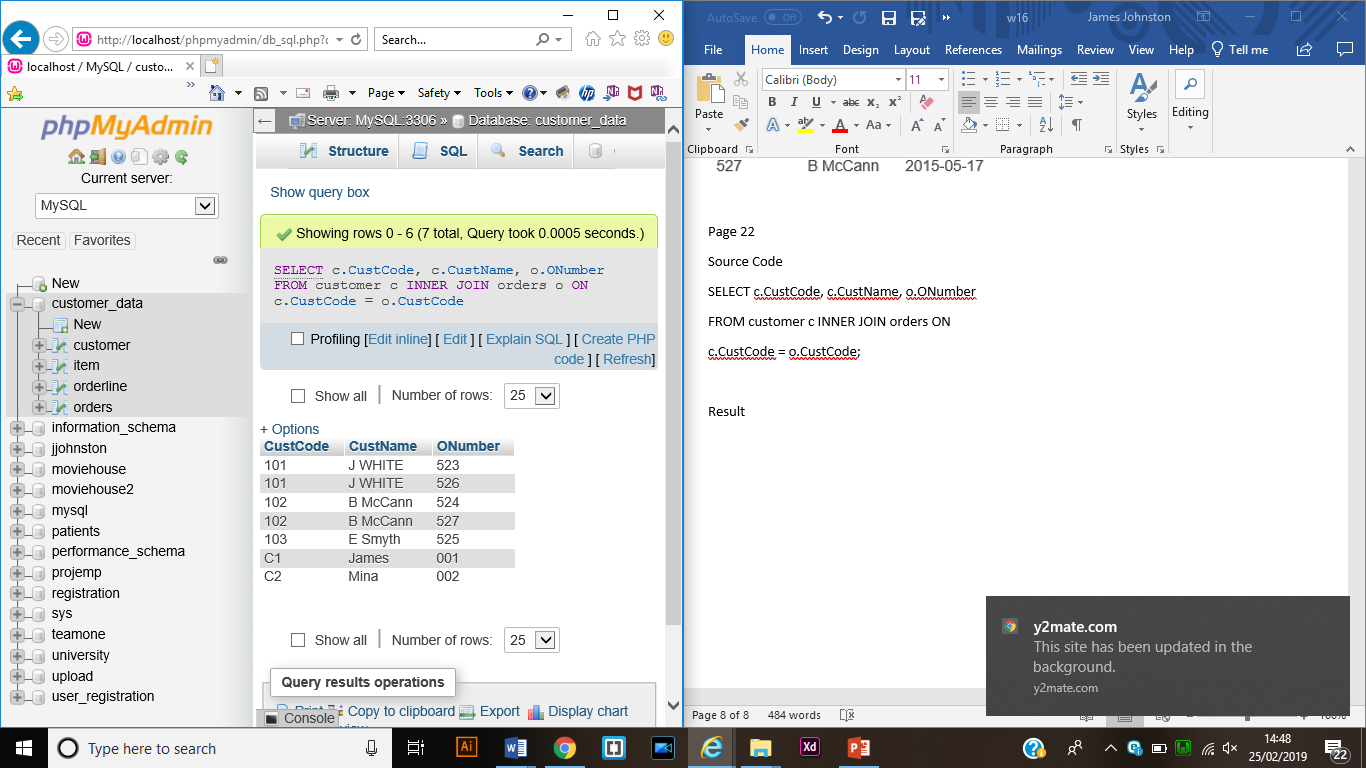
Source Code

SELECT c.CustCode, c.CustName, o.ONumber

FROM customer c INNER JOIN orders ON

c.CustCode = o.CustCode;

Result



1. Get order number and their price (Quantity \* UnitCost)

Source Code

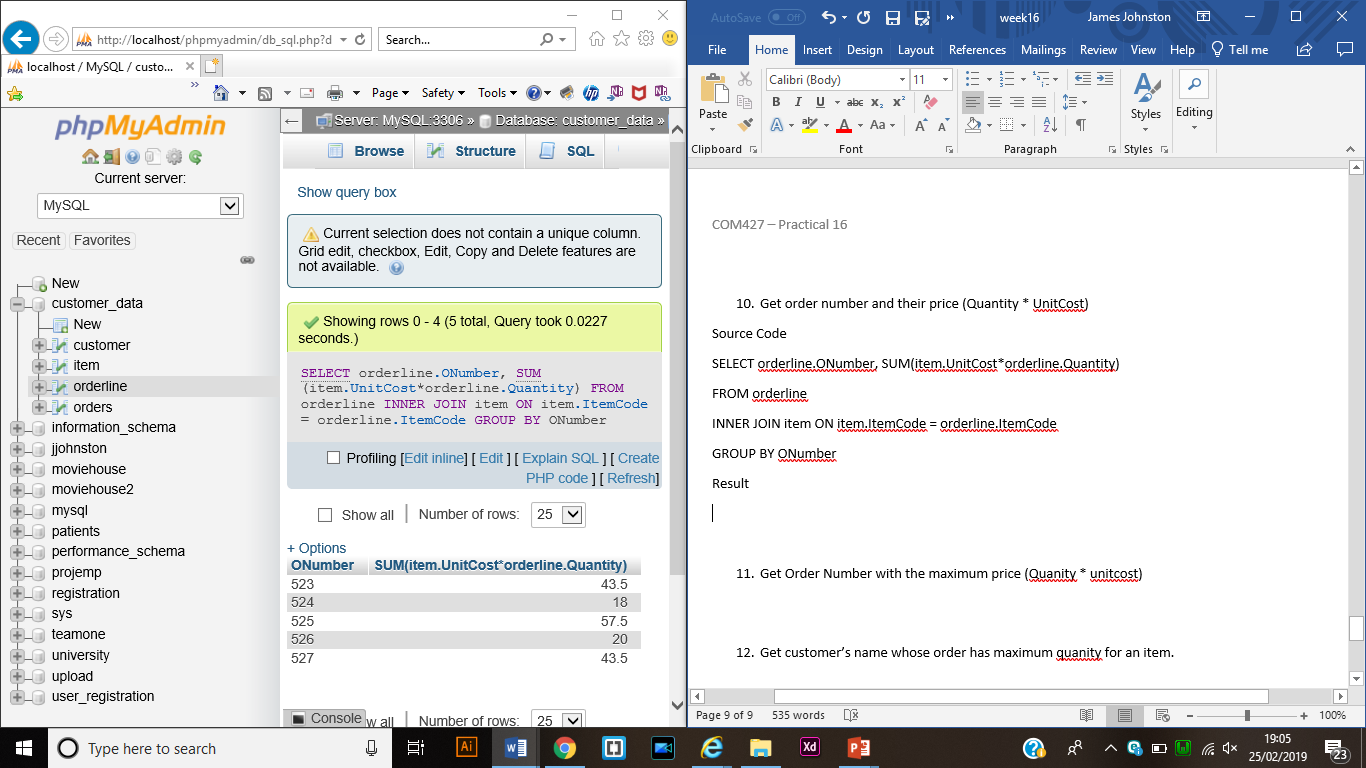
SELECT orderline.ONumber, SUM(item.UnitCost\*orderline.Quantity)

FROM orderline

INNER JOIN item ON item.ItemCode = orderline.ItemCode

GROUP BY ONumber

Result



1. Get Order Number with the maximum price (Quanity \* unitcost)

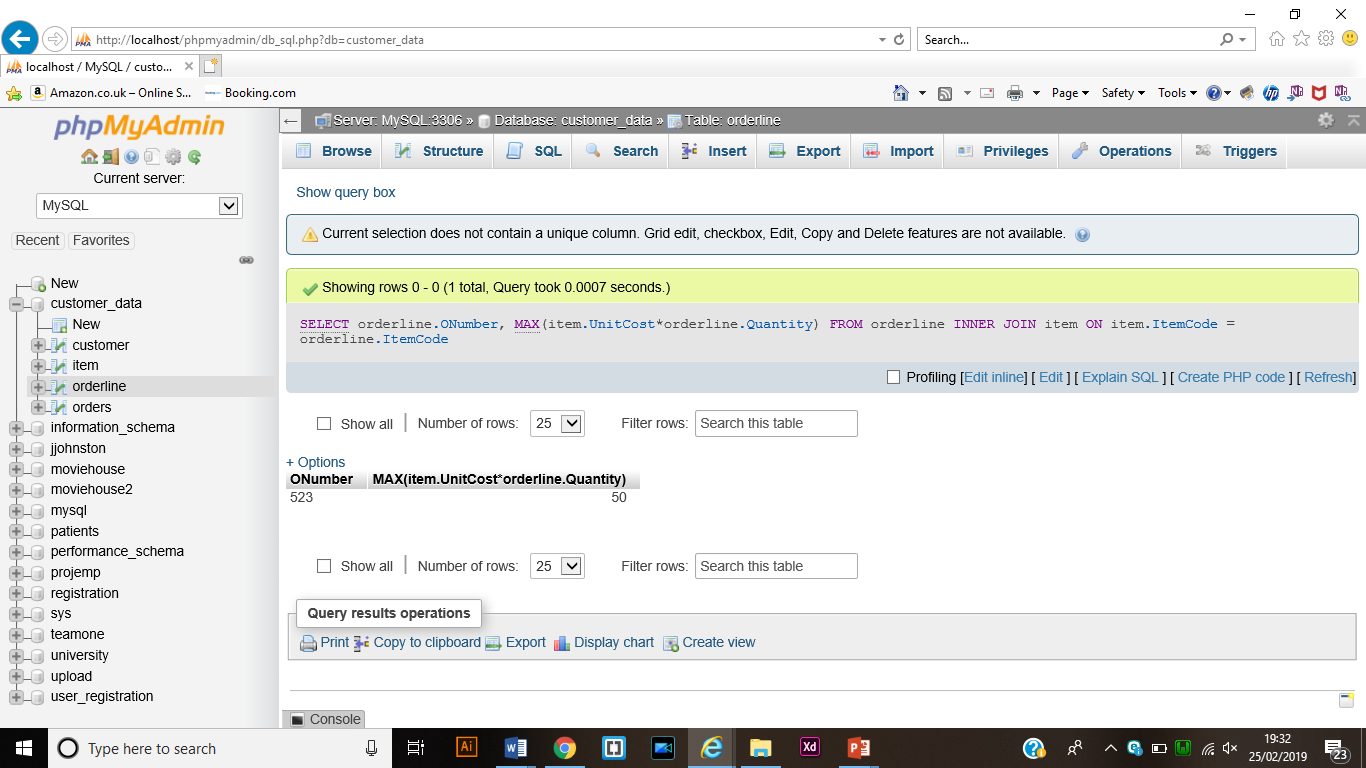
SQL Source Code

SELECT orderline.ONumber, MAX(item.UnitCost\*orderline.Quantity)

FROM orderline

INNER JOIN item ON item.ItemCode = orderline.ItemCode;

Result



1. Get customer’s name whose order has maximum quanity for an item.

SQL source Code

SELECT DISTINCT customer.CustName, SUM(orderline.Quantity)

FROM customer

INNER JOIN orders ON orders.CustCode = customer.CustCode

INNER JOIN orderline ON

orders.ONumber = orderline.ONumber

WHERE orderline.Quantity = (

SELECT MAX(Quantity) FROM orderline);

Result

