

1. Write a query to Display the product details (product_class_code, product_id, product_desc, product_price,) as per the following criteria and sort them in descending order of category: a. If the category is 2050, increase the price by 2000 b. If the category is 2051, increase the price by 500 c. If the category is 2052, increase the price by 600. Hint: Use case statement. no permanent change in table required. (60 ROWS) [NOTE: PRODUCT TABLE]

```

1  SELECT
2      product_class_code,
3      product_id,
4      product_desc,
5      product_price,
6  CASE
7      WHEN product_class_code = 2050 THEN product_price + 2000
8      WHEN product_class_code = 2051 THEN product_price + 500
9      WHEN product_class_code = 2052 THEN product_price + 600
10     ELSE product_price = 0
11 END AS IncreasedValue
12 FROM
13     product;

```

100% 1:1

Result Grid Filter Rows: Search Export:

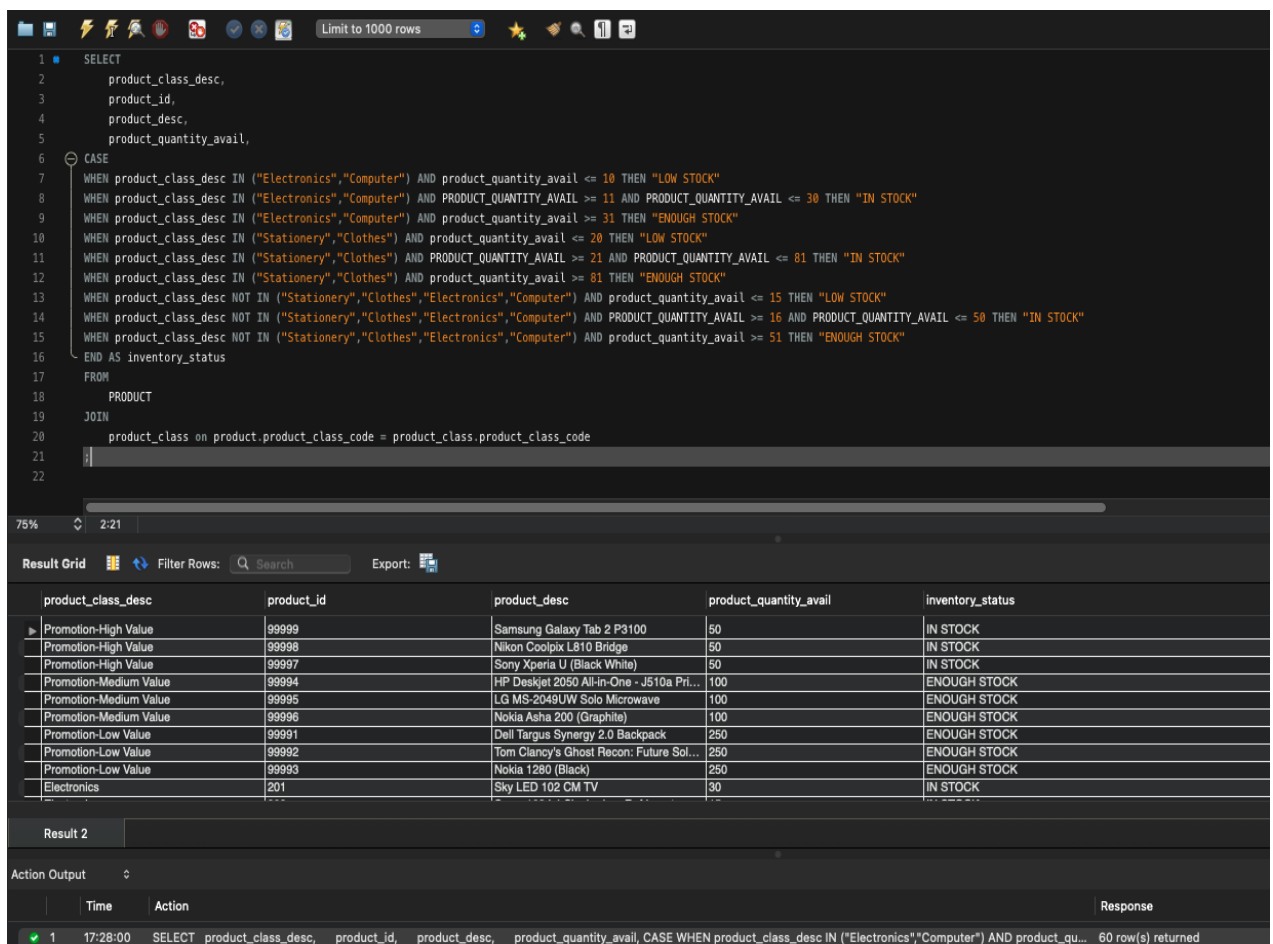
product_class_co...	product_id	product_desc	product_pri...	IncreasedValue
3000	99999	Samsung Galaxy Tab 2 P3100	19300.00	0
3000	99998	Nikon Coolpix L810 Bridge	14987.00	0
3000	99997	Sony Xperia U (Black White)	16499.00	0
3001	99994	HP Deskjet 2050 All-in-One - J510a Pri...	3749.00	0
3001	99995	LG MS-2049UJW Solo Microwave	4800.00	0
3001	99996	Nokia Asha 200 (Graphite)	4070.00	0
3002	99991	Dell Targus Synergy 2.0 Backpack	999.00	0
3002	99992	Tom Clancy's Ghost Recon: Future Sol...	999.00	0
3002	99993	Nokia 1280 (Black)	999.00	0
2050	201	Sky LED 102 CM TV	35000.00	37000.00

Result 2

Action Output

	Time	Action	Response
1	17:25:36	SELECT product_class_code, product_id, product_desc, product_price, CASE WHEN product_class_code = 2050 THEN product_price + 2000 WHEN product_cla...	60 row(s) returned

2. Write a query to display (product_class_desc, product_id, product_desc, product_quantity_avail) and Show inventory status of products as below as per their available quantity: a. For Electronics and Computer categories, if available quantity is <= 10, show 'Low stock', 11 <= qty <= 30, show 'In stock', >= 31, show 'Enough stock' b. For Stationery and Clothes categories, if qty <= 20, show 'Low stock', 21 <= qty <= 80, show 'In stock', >= 81, show 'Enough stock' c. Rest of the categories, if qty <= 15 – 'Low Stock', 16 <= qty <= 50 – 'In Stock', >= 51 – 'Enough stock' For all categories, if available quantity is 0, show 'Out of stock'. Hint: Use case statement. (60 ROWS) [NOTE: TABLES TO BE USED – product, product_class]



The screenshot shows a SQL IDE interface with a query editor at the top and a results grid at the bottom. The query is a SELECT statement with a CASE statement to determine the inventory status based on product class and quantity.

```

1 SELECT
2     product_class_desc,
3     product_id,
4     product_desc,
5     product_quantity_avail,
6     CASE
7         WHEN product_class_desc IN ("Electronics","Computer") AND product_quantity_avail <= 10 THEN "LOW STOCK"
8         WHEN product_class_desc IN ("Electronics","Computer") AND PRODUCT_QUANTITY_AVAIL >= 11 AND PRODUCT_QUANTITY_AVAIL <= 30 THEN "IN STOCK"
9         WHEN product_class_desc IN ("Electronics","Computer") AND product_quantity_avail >= 31 THEN "ENOUGH STOCK"
10        WHEN product_class_desc IN ("Stationery","Clothes") AND product_quantity_avail <= 20 THEN "LOW STOCK"
11        WHEN product_class_desc IN ("Stationery","Clothes") AND PRODUCT_QUANTITY_AVAIL >= 21 AND PRODUCT_QUANTITY_AVAIL <= 81 THEN "IN STOCK"
12        WHEN product_class_desc IN ("Stationery","Clothes") AND product_quantity_avail >= 81 THEN "ENOUGH STOCK"
13        WHEN product_class_desc NOT IN ("Stationery","Clothes","Electronics","Computer") AND product_quantity_avail <= 15 THEN "LOW STOCK"
14        WHEN product_class_desc NOT IN ("Stationery","Clothes","Electronics","Computer") AND PRODUCT_QUANTITY_AVAIL >= 16 AND PRODUCT_QUANTITY_AVAIL <= 50 THEN "IN STOCK"
15        WHEN product_class_desc NOT IN ("Stationery","Clothes","Electronics","Computer") AND product_quantity_avail >= 51 THEN "ENOUGH STOCK"
16    END AS inventory_status
17 FROM
18     PRODUCT
19 JOIN
20     product_class on product.product_class_code = product_class.product_class_code
21
22

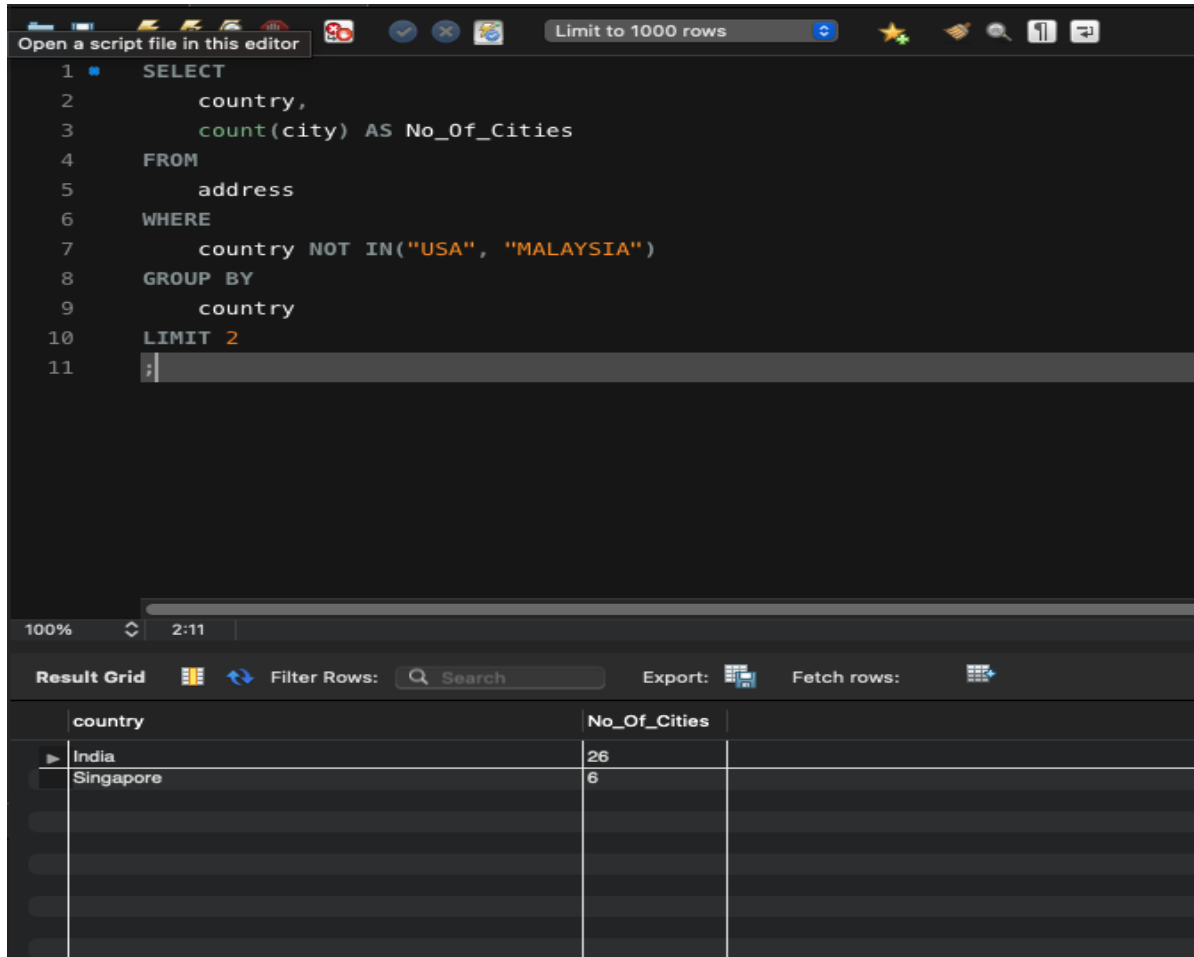
```

The results grid shows the following data:

product_class_desc	product_id	product_desc	product_quantity_avail	inventory_status
Promotion-High Value	99999	Samsung Galaxy Tab 2 P3100	50	IN STOCK
Promotion-High Value	99998	Nikon Coolpix L810 Bridge	50	IN STOCK
Promotion-High Value	99997	Sony Xperia U (Black White)	50	IN STOCK
Promotion-Medium Value	99994	HP Deskjet 2050 All-in-One - J510a Pri...	100	ENOUGH STOCK
Promotion-Medium Value	99995	LG MS-2049UW Solo Microwave	100	ENOUGH STOCK
Promotion-Medium Value	99996	Nokia Asha 200 (Graphite)	100	ENOUGH STOCK
Promotion-Low Value	99991	Deil Targus Synergy 2.0 Backpack	250	ENOUGH STOCK
Promotion-Low Value	99992	Tom Clancy's Ghost Recon: Future Sol...	250	ENOUGH STOCK
Promotion-Low Value	99993	Nokia 1280 (Black)	250	ENOUGH STOCK
Electronics	201	Sky LED 102 CM TV	30	IN STOCK

The Action Output section shows the execution of the query, indicating that 60 rows were returned.

3. Write a query to Show the count of cities in all countries other than USA & MALAYSIA, with more than 1 city, in the descending order of CITIES. (2 rows) [NOTE: ADDRESS TABLE, Do not use Distinct]



The screenshot shows a SQL editor interface with a query written in a dark theme. The query is as follows:

```
1 SELECT
2     country,
3     count(city) AS No_Of_Cities
4 FROM
5     address
6 WHERE
7     country NOT IN("USA", "MALAYSIA")
8 GROUP BY
9     country
10 LIMIT 2
11
```

Below the editor, the result grid is displayed, showing the output of the query. The grid has two columns: 'country' and 'No_Of_Cities'. The results are sorted in descending order of the number of cities.

country	No_Of_Cities
India	26
Singapore	6

4. Write a query to display the customer_id, customer full name, city, pincode, and order details (order id, product class desc, product desc, subtotal(product_quantity * product_price)) for orders shipped to cities whose pin codes do not have any 0s in them. Sort the output on customer name and subtotal. (52 ROWS) [NOTE: TABLE TO BE USED - online_customer, address, order_header, order_items, product, product_class]

```

1 SELECT
2     online_customer.customer_id,
3     CONCAT(customer_fname, customer_lname) AS customer_full_name,
4     city,
5     pincode,
6     order_header.order_id,
7     order_date,
8     product_class.product_class_desc,
9     product_desc,
10    product_quantity * product_price AS Subtotal
11 FROM
12     online_customer
13 JOIN
14     address ON online_customer.address_id = address.address_id
15 JOIN
16     order_header ON online_customer.customer_id = order_header.customer_id
17 JOIN
18     order_items ON order_header.order_id = order_items.order_id
19 JOIN
20     product ON order_items.product_id = product.product_id
21 LEFT JOIN
22     product_class ON product.product_class_code = product_class.product_class_code
23 WHERE
24     order_status = "shipped" AND pincode NOT LIKE "0%"
25 ORDER BY
26     customer_full_name asc, Subtotal asc, order_date
27
28

```

50% 2:27

Result Grid Filter Rows: Search Export:

	customer_id	customer_full_na...	city	pincode	order_id	ord...	product_class_desc	product_desc	Subtotal
▶	30	AnitaKohli	Amherst	14228	10059	NULL	Electronics	Cybershot DWC-W325 Camera	5300.00
	19	BhartiSubhash	Dharmapuri	635897	10054	NULL	Clothes	Infant Sleepwear Blue	500.00
	19	BhartiSubhash	Dharmapuri	635897	10034	NULL	Bags	Women Hand Bag	1600.00
	19	BhartiSubhash	Dharmapuri	635897	10034	NULL	Kitchen Items	Philis Wah Collection Juicer JM12	2029.00
	19	BhartiSubhash	Dharmapuri	635897	10054	NULL	Bags	HP ODC Laptop Bag 15.5	3390.00
	10	BidhanC.Roy	Hosur	635235	10070	NULL	Stationery	4M Post It Pad 3.5	70.00
	10	BidhanC.Roy	Hosur	635235	10015	NULL	Stationery	4M Post It Pad 3.5	175.00
	10	BidhanC.Roy	Hosur	635235	10070	NULL	Stationery	Kasyo DJ-2100 Desktop Calculator	338.00
	10	BidhanC.Roy	Hosur	635235	10015	NULL	Watches	Adidas Analog Watch	2295.00
	10	BidhanC.Roy	Hosur	635235	10015	NULL	Furnitures	Supreme Fusion Cupboard 02TB	3000.00

Result 2

Action Output

	Time	Action	Response
✓ 1	17:30:24	SELECT online_customer.customer_id, CONCAT(customer_fname, customer_lname) AS customer_full_name, city, pincode, order_header.order_id, order_date,...	52 row(s) returned

5. Write a Query to display product id,product description,totalquantity(sum(product quantity)) for an item which has been bought maximum no. of times (Quantity Wise) along with product id 201. (USE SUB-QUERY) (1 ROW) [NOTE: ORDER_ITEMS TABLE, PRODUCT TABLE]

```

1 SELECT
2     product.product_id,
3     product_desc,
4     sum(product_quantity) AS totalquantity
5 FROM
6     product JOIN order_items ON product.product_id = order_items.product_id
7 WHERE
8     product.product_id = 201
9

```

product_id	product_desc	totalquant...
201	Sky LED 102 CM TV	6

Result 2

Action Output

	Time	Action	Response
1	17:31:31	SELECT product.product_id, product_desc, sum(product_quantity) AS totalquantity FROM product JOIN order_items ON product.product_id = order_items.product_id...	1 row(s) returned

6. Write a query to display the customer_id,customer name, email and order details (order id, product desc,product qty, subtotal(product_quantity * product_price)) for all customers even if they have not ordered any item.(225 ROWS) [NOTE: TABLE TO BE USED - online_customer, order_header, order_items, product]

```

1 SELECT
2     online_customer.customer_id,
3     CONCAT(customer_fname, customer_lname) AS customer_name,
4     customer_email,
5     order_header.order_id,
6     product_desc,
7     product_quantity,
8     (product_quantity*product_price) AS subtotal
9 FROM
10    online_customer
11 LEFT JOIN
12    order_header on online_customer.customer_id = order_header.customer_id
13 LEFT JOIN
14    order_items on order_header.order_id = order_items.order_id
15 LEFT JOIN
16    product on order_items.product_id = product.product_id
17
18
19

```

customer_id	customer_name	customer_email	order_id	product_desc	product_quantity	subtotal
1	JenniferWilson	jen_w@gmail.com	10050	Cricket Set for Boys	1	4500.00
1	JenniferWilson	jen_w@gmail.com	10050	Sams 21L Microwave Oven	1	6880.00
1	JenniferWilson	jen_w@gmail.com	10050	HP ODC School Bag 2.5'	2	1598.00
1	JenniferWilson	jen_w@gmail.com	10011	OnePlus 6 Smart Phone	1	32500.00
1	JenniferWilson	jen_w@gmail.com	10011	Samsung Galaxy On6	2	28000.00
1	JenniferWilson	jen_w@gmail.com	10011	KitchAart Siphon Coffee Maker 500 ml	1	1790.00
1	JenniferWilson	jen_w@gmail.com	10001	Sky LED 102 CM TV	1	35000.00
1	JenniferWilson	jen_w@gmail.com	10001	Infant Sleepwear Blue	3	750.00
1	JenniferWilson	jen_w@gmail.com	10001	Samsung Galaxy On6	1	14000.00
1	JenniferWilson	jen_w@gmail.com	10001	Foldable Premium Chair	1	4000.00

Result 1

Action Output

	Time	Action	Response
1	17:32:07	SELECT online_customer.customer_id, CONCAT(customer_fname, customer_lname) AS customer_name, customer_email, order_header.order_id, product_desc,...	225 row(s) returned

7. Write a query to display carton id, (len*width*height) as carton_vol and identify the optimum carton (carton with the least volume whose volume is greater than the total volume of all items (len * width * height * product_quantity)) for a given order whose order id is 10006, Assume all items of an order are packed into one single carton (box). (1 ROW) [NOTE: CARTON TABLE]

```

1 SELECT
2     carton_id,
3     (len*width*height) AS carton_volume
4 FROM
5     carton
6 WHERE
7     (len*width*height) > ( select sum(len*width*height*product_quantity) from product join order_items on product.product_id = order_items.product_id WHERE order_id = 10006)
8 limit 1
9
10

```

Result Grid

carton_id	carton_volume
40	1215000000

Action Output

Time	Action	Response	Duration / Fetch Time
17:33:18	SELECT carton_id, (len*width*height) AS carton_volume FROM carton WHERE (len*width*height) > (select sum(len*width*height*product_quantity) from product join...	1 row(s) returned	0.00043 sec / 0.000...

8. Write a query to display details (customer id,customer fullname,order id,product quantity) of customers who bought more than ten (i.e. total order qty) products per shipped order. (11 ROWS) [NOTE: TABLES TO BE USED - online_customer, order_header, order_items,]

```

1 SELECT
2     online_customer.customer_id,
3     CONCAT(customer_fname, customer_lname) AS customer_fullname,
4     order_header.order_id,
5     sum(product_quantity) AS total_quantity
6 FROM
7     online_customer
8 JOIN order_header ON online_customer.customer_id = order_header.customer_id
9 JOIN order_items ON order_header.order_id = order_items.order_id
10 WHERE
11     order_header.order_status = "Shipped"
12 GROUP BY
13     order_header.order_id
14 HAVING sum(product_quantity) > 10
15

```

Result Grid

customer_id	customer_fullname	order_id	total_quantity
2	JacksonDavis	10002	16
5	RamyaRavinder	10003	12
6	AnitaGoswami	10006	15
3	KomalChoudhary	10007	53
7	AshwathiBhatt	10008	25

Action Output

Time	Action	Response
17:34:02	SELECT online_customer.customer_id, CONCAT(customer_fname, customer_lname) AS customer_fullname, order_header.order_id, sum(product_quantity) AS total...	11 row(s) returned

9. Write a query to display the order_id, customer id and customer full name of customers along with (product_quantity) as total quantity of products shipped for order ids > 10060. (6 ROWS)
[NOTE: TABLES TO BE USED - online_customer, order_header, order_items]

```

1 SELECT
2     order_header.order_id,
3     order_header.customer_id,
4     CONCAT(online_customer.customer_fname, online_customer.customer_lname) AS customer_fullname,
5     sum(product_quantity) AS total_quantity
6 FROM
7     order_header
8 JOIN online_customer ON order_header.customer_id = online_customer.customer_id
9 JOIN order_items ON order_header.order_id = order_items.order_id
10 WHERE
11     order_status = "Shipped" AND order_header.order_id > 10060
12 GROUP BY
13     order_header.order_id
14 ;
15

```

100% 1:15

Result Grid Filter Rows: Search Export:

order_id	customer_id	customer_fullname	total_quantity
10061	37	JamesNewton Howard	1
10063	18	SureshBabu	8
10064	35	ThomasNewman	3
10068	51	Ahmad Bin GhAzail	3
10069	23	AnnaPinnock	4

Result 2

Action Output

Time	Action	Response
17:35:14	SELECT order_header.order_id, order_header.customer_id, CONCAT(online_customer.customer_fname, online_customer.customer_lname) AS customer_fullname, su...	6 row(s) returned

10. Write a query to display product class description ,total quantity (sum(product_quantity)),Total value (product_quantity * product price) and show which class of products have been shipped highest(Quantity) to countries outside India other than USA? Also show the total value of those items. (1 ROWS)[NOTE:PRODUCT TABLE,ADDRESS TABLE,ONLINE_CUSTOMER TABLE,ORDER_HEADER TABLE,ORDER_ITEMS TABLE,PRODUCT_CLASS TABLE]

```

1 SELECT
2     country,
3     product_class_desc,
4     sum(product_quantity) AS total_value,
5     (product_quantity * product_price) AS total_quantity
6 FROM
7     address
8 JOIN online_customer ON address.address_id = online_customer.address_id
9 JOIN order_header ON online_customer.customer_id = order_header.customer_id
10 JOIN order_items ON order_header.order_id = order_items.order_id
11 JOIN product ON order_items.product_id = product.product_id
12 JOIN product_class ON product.product_class_code = product_class.product_class_code
13 WHERE
14     order_header.order_status = "Shipped" And country NOT IN ("India","USA")
15 GROUP BY
16     order_header.order_id, product.product_id, product_class.product_class_code
17 order by
18     count((product_quantity * product_price))desc limit 1;
19

```

100% 1:19

Result Grid Filter Rows: Search Export: Fetch rows:

country	product_class_desc	total_value	total_quantity
Sri Lanka	Bags	2	1598.00

Result 1

Action Output

Time	Action	Response
17:36:41	SELECT country, product_class_desc, sum(product_quantity) AS total_value, (product_quantity * product_price) AS total_quantity FROM address JOIN onlinecus...	1 row(s) returned