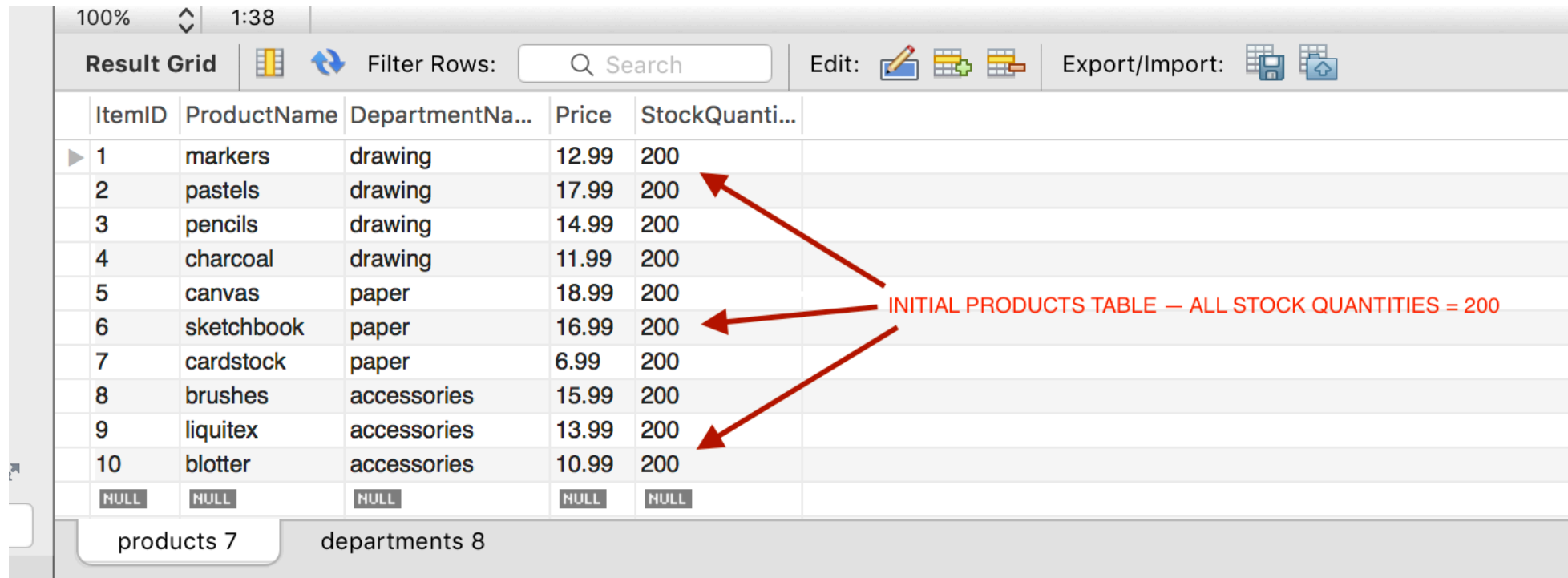


Bamazon

bamazonCustomer.js

MySQL database – shows initial “products” table where all quantities = 200



100% 1:38

Result Grid Filter Rows: Search Edit: Export/Import:

ItemID	ProductName	DepartmentNa...	Price	StockQuanti...
1	markers	drawing	12.99	200
2	pastels	drawing	17.99	200
3	pencils	drawing	14.99	200
4	charcoal	drawing	11.99	200
5	canvas	paper	18.99	200
6	sketchbook	paper	16.99	200
7	cardstock	paper	6.99	200
8	brushes	accessories	15.99	200
9	liquitex	accessories	13.99	200
10	blotter	accessories	10.99	200
NULL	NULL	NULL	NULL	NULL

products 7 departments 8

INITIAL PRODUCTS TABLE — ALL STOCK QUANTITIES = 200

MySQL database – shows initial “departments” table (but I didn’t get to code “Level 3” of the HW exercise in JS.

100% 1:38

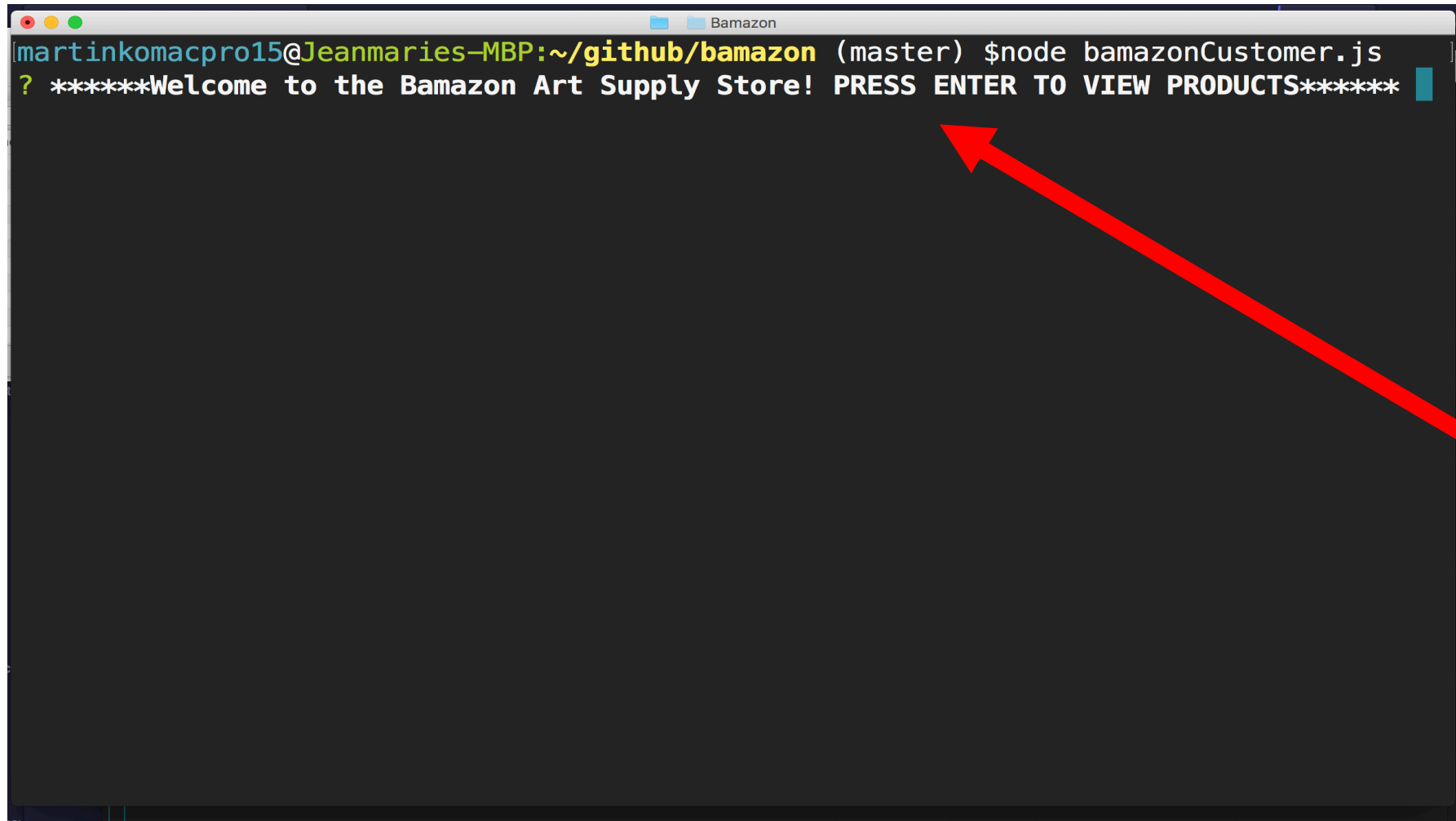
Result Grid Filter Rows: Search Edit: Export/Import:

DepartmentID	DepartmentNa...	OverheadCosts	TotalSales
1	drawing	999.99	8600.99
2	paper	1500.00	9900.00
3	accessories	799.86	1536.00
NULL	NULL	NULL	NULL

CREATED “DEPARTMENTS” TABLE IN MYSQL

products 7 departments 8

Starting bamazonCustomer.js. . .

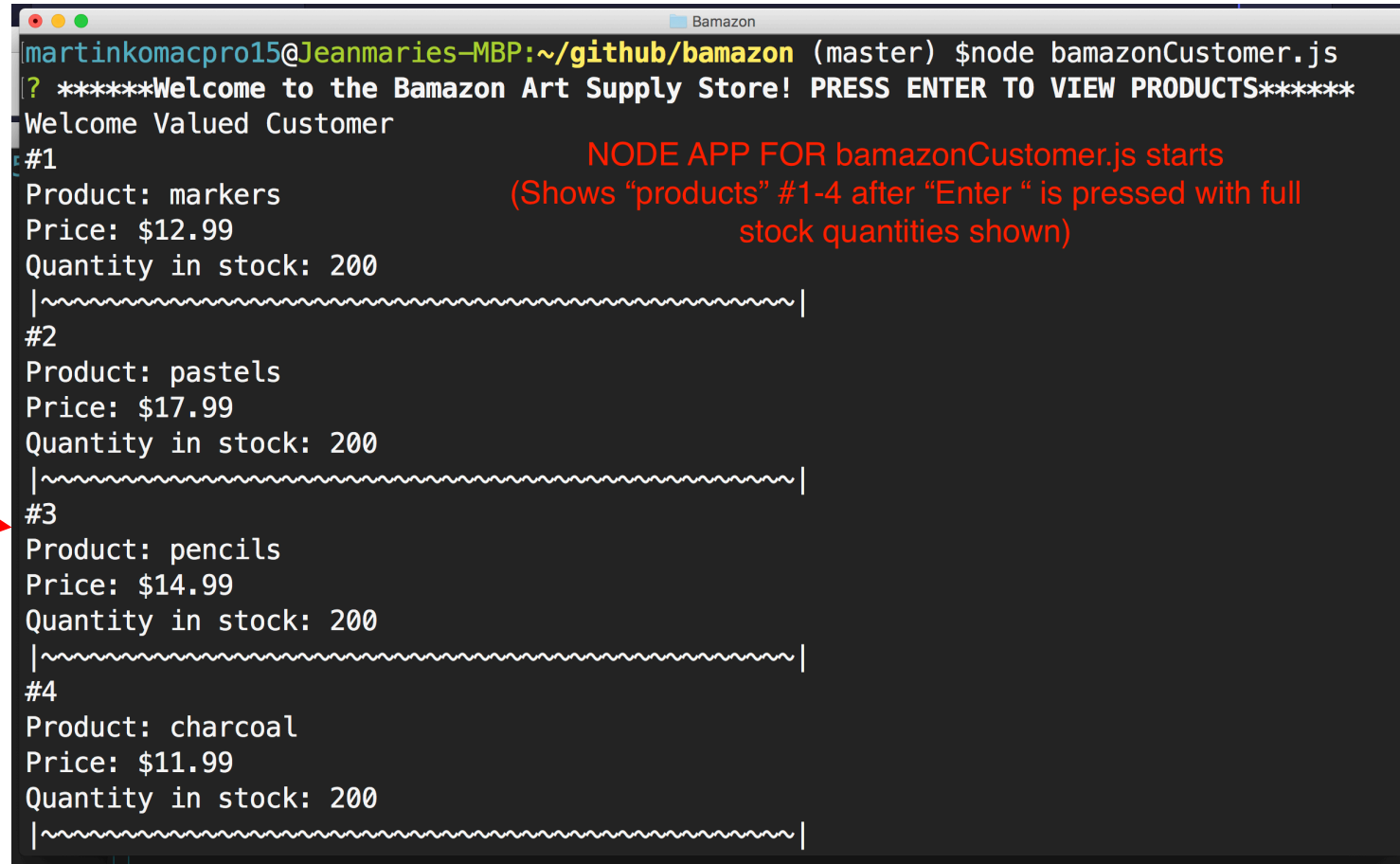


A terminal window titled "Bamazon" is shown. The prompt is `martinkomacpro15@Jeanmaries-MBP:~/github/bamazon (master)`. The command `$node bamazonCustomer.js` has been executed. The output is `? *****Welcome to the Bamazon Art Supply Store! PRESS ENTER TO VIEW PRODUCTS*****`. A red arrow points from the right side of the terminal towards the output text.

```
martinkomacpro15@Jeanmaries-MBP:~/github/bamazon (master) $node bamazonCustomer.js
? *****Welcome to the Bamazon Art Supply Store! PRESS ENTER TO VIEW PRODUCTS*****
```

Shows Products #1-4 after “Enter” is pressed with quantity=200 for all products (200 initial stock). . .

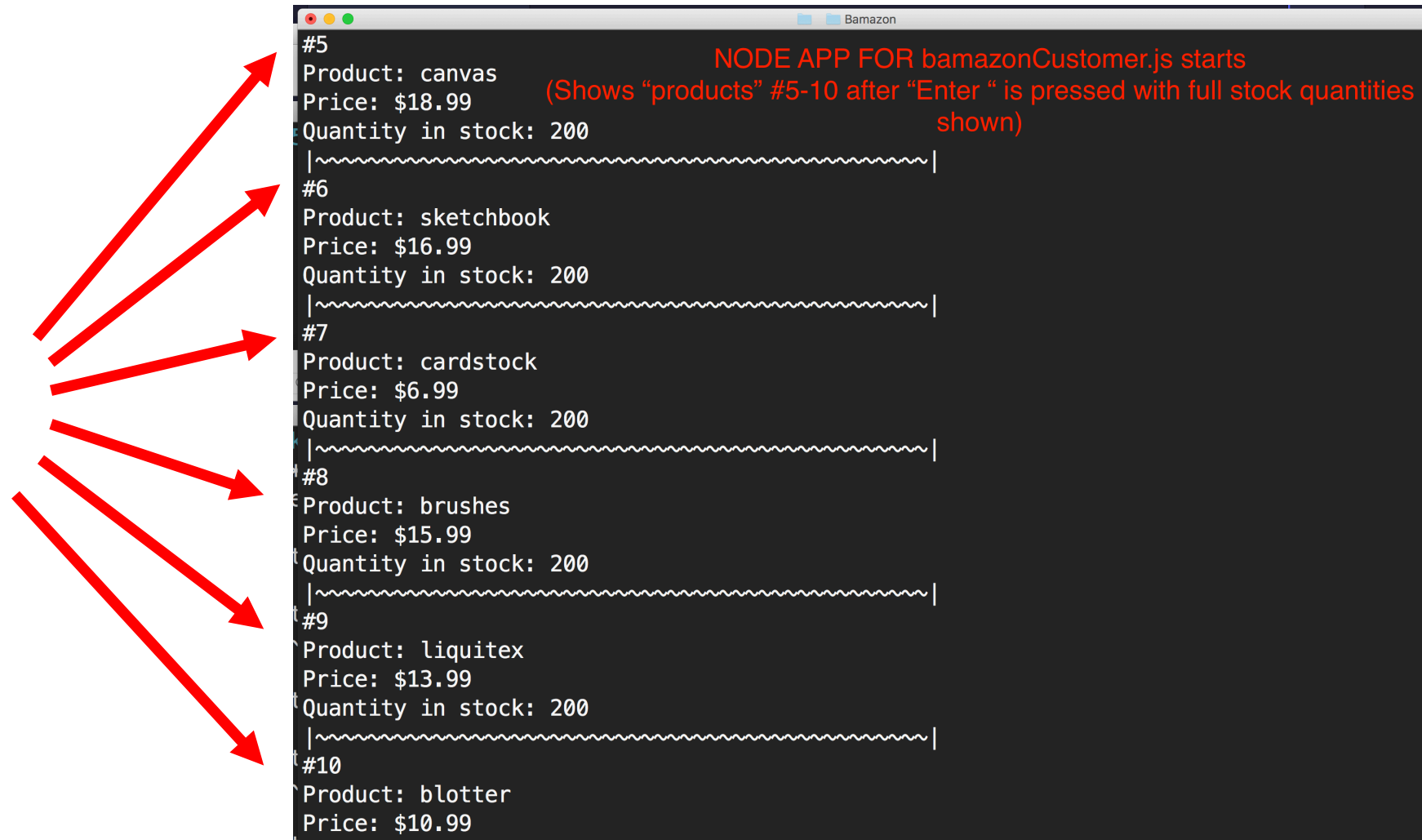
. . .



```
martinkomacpro15@Jeanmaries-MBP:~/github/bamazon (master) $node bamazonCustomer.js
? *****Welcome to the Bamazon Art Supply Store! PRESS ENTER TO VIEW PRODUCTS*****
Welcome Valued Customer
#1
Product: markers
Price: $12.99
Quantity in stock: 200
|~~~~~|
#2
Product: pastels
Price: $17.99
Quantity in stock: 200
|~~~~~|
#3
Product: pencils
Price: $14.99
Quantity in stock: 200
|~~~~~|
#4
Product: charcoal
Price: $11.99
Quantity in stock: 200
|~~~~~|
```

NODE APP FOR bamazonCustomer.js starts
(Shows “products” #1-4 after “Enter “ is pressed with full
stock quantities shown)

Shows Products #5-10 after “Enter” is pressed with quantity=200 for all products.

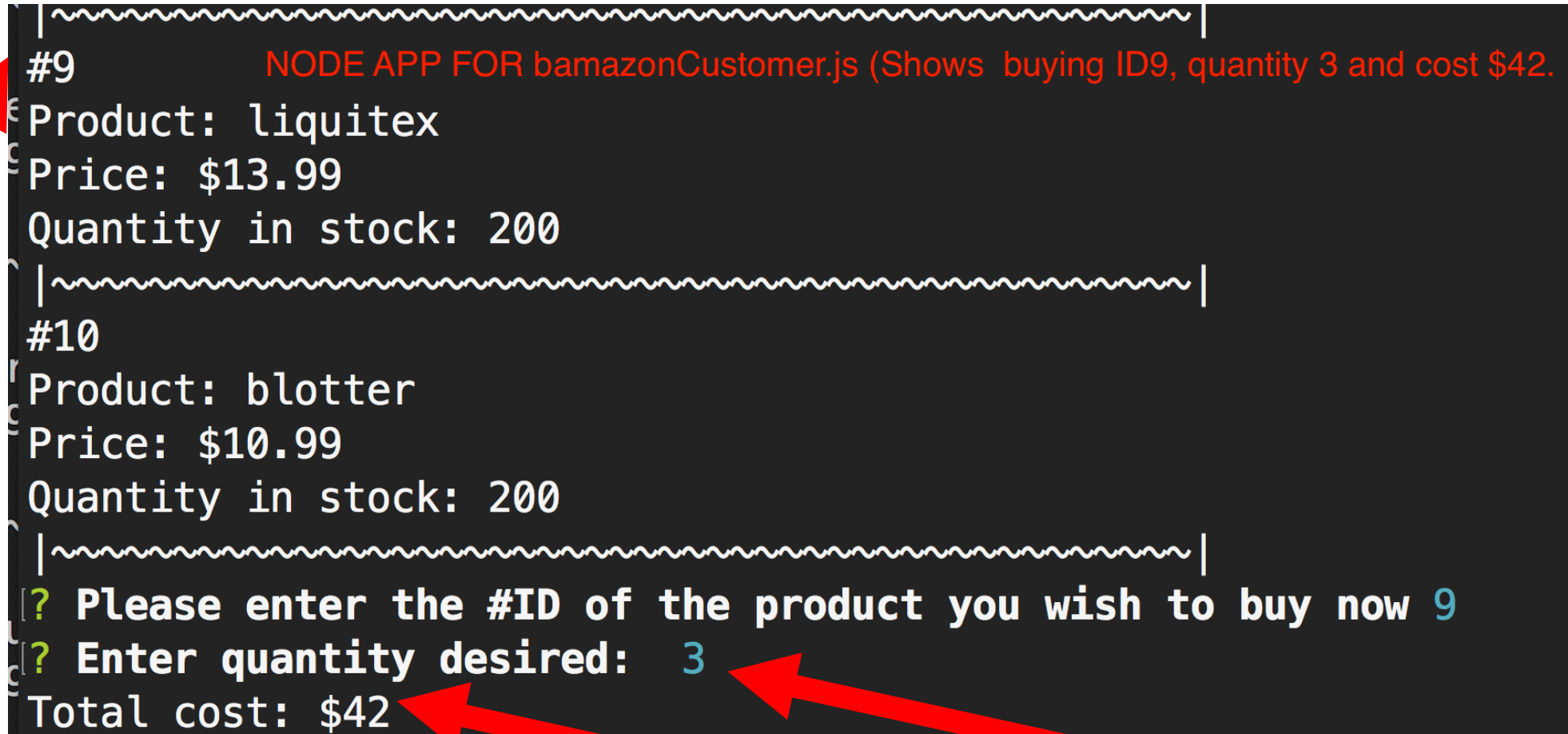


A terminal window titled "Bamazon" displays a list of products. Red arrows point from the left to each product entry. Red text at the top right explains the context: "NODE APP FOR bamazonCustomer.js starts (Shows 'products' #5-10 after 'Enter' is pressed with full stock quantities shown)".

```
#5
Product: canvas
Price: $18.99
Quantity in stock: 200
|~~~~~|
#6
Product: sketchbook
Price: $16.99
Quantity in stock: 200
|~~~~~|
#7
Product: cardstock
Price: $6.99
Quantity in stock: 200
|~~~~~|
#8
Product: brushes
Price: $15.99
Quantity in stock: 200
|~~~~~|
#9
Product: liquitex
Price: $13.99
Quantity in stock: 200
|~~~~~|
#10
Product: blotter
Price: $10.99
```

NODE APP FOR bamazonCustomer.js starts
(Shows "products" #5-10 after "Enter" is pressed with full stock quantities shown)

...Prompts and responses for item to be purchased (#9 Liquitex), quantity (3) and calculation of total cost (\$45)...



A terminal window showing a Node.js application for a Amazon customer. The application displays product information for two items: #9 (Liquitex) and #10 (blotter). The user is prompted to enter a product ID and a quantity. The user enters '9' for the product ID and '3' for the quantity. The application calculates the total cost as \$42. Red arrows point from the text above to the values '9', '3', and '\$42' in the terminal output.

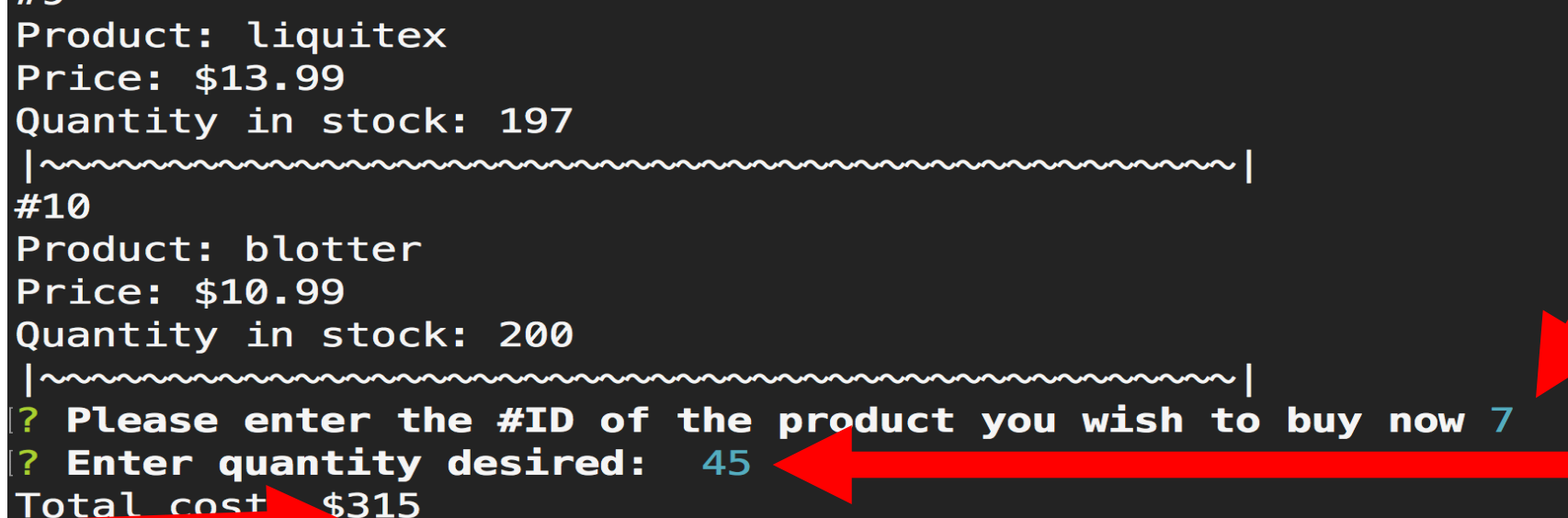
```
| ~~~~~ |
#9      NODE APP FOR bamazonCustomer.js (Shows buying ID9, quantity 3 and cost $42.
Product: liquitex
Price: $13.99
Quantity in stock: 200
| ~~~~~ |
#10
Product: blotter
Price: $10.99
Quantity in stock: 200
| ~~~~~ |
? Please enter the #ID of the product you wish to buy now 9
? Enter quantity desired: 3
Total cost: $42
```

... The quantity of available stock is then updated to reflect that 3 units of #9 (Liquitex) were purchased....

```
|~|
#7
Product: cardstock
Price: $6.99
Quantity in stock: 155
|~|
#8
Product: brushes
Price: $15.99
Quantity in stock: 200
|~|
#9
Product: liquitex
Price: $13.99
Quantity in stock: 197
|~|
#10
Product: blotter
Price: $10.99
Quantity in stock: 200
|~|
? Please enter the #ID of the product you wish to buy now
```


...Next...Prompts and responses for item to be purchased (#7 cardstock), quantity (45) and calculation of total cost (\$315)...

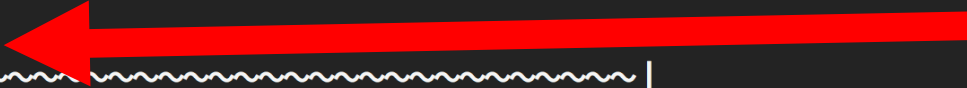
```
| ~~~~~ |  
#7  
Product: cardstock  
Price: $6.99  
Quantity in stock: 200  
| ~~~~~ |  
#8  
Product: brushes  
Price: $15.99  
Quantity in stock: 200  
| ~~~~~ |  
#9  
Product: liquitex  
Price: $13.99  
Quantity in stock: 197  
| ~~~~~ |  
#10  
Product: blotter  
Price: $10.99  
Quantity in stock: 200  
| ~~~~~ |  
? Please enter the #ID of the product you wish to buy now 7  
? Enter quantity desired: 45  
Total cost $315
```

Three red arrows are overlaid on the terminal output. One arrow points from the right side of the screen to the number '7' in the prompt '? Please enter the #ID of the product you wish to buy now 7'. A second arrow points from the bottom right to the number '45' in the prompt '? Enter quantity desired: 45'. A third arrow points from the bottom left to the text '\$315' in the line 'Total cost \$315'.

... The quantity of available stock is then updated to reflect that 45 units of #7 (cardstock) were purchased. .

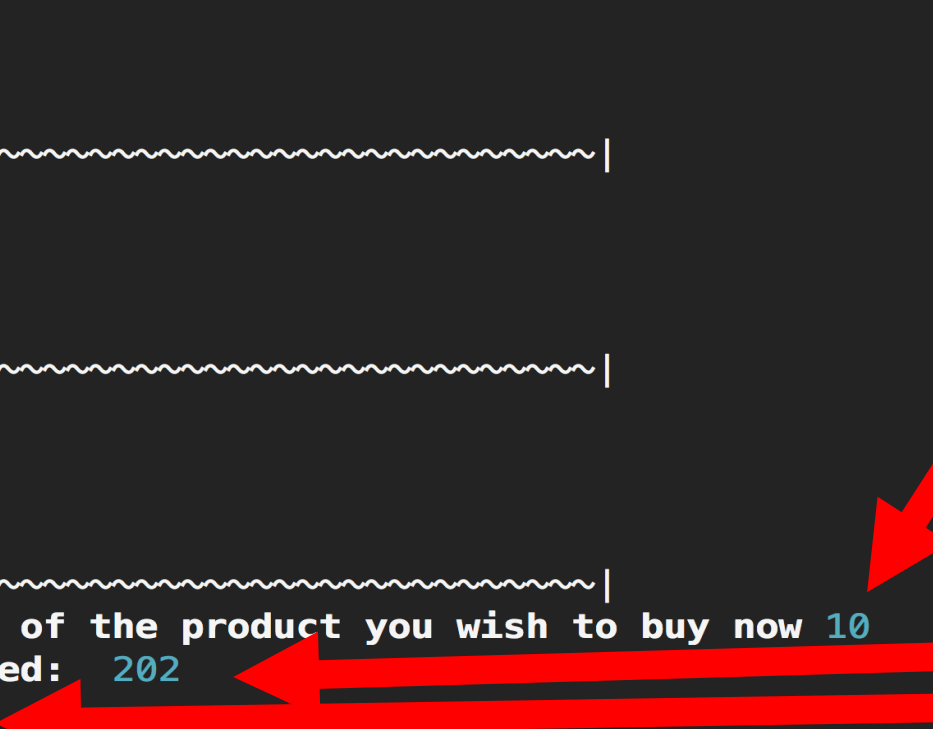
..

```
| ~~~~~ |  
#7  
Product: cardstock  
Price: $6.99  
Quantity in stock: 155  
| ~~~~~ |  
#8  
Product: brushes  
Price: $15.99  
Quantity in stock: 200  
| ~~~~~ |  
#9  
Product: liquitex  
Price: $13.99  
Quantity in stock: 197  
| ~~~~~ |  
#10  
Product: blotter  
Price: $10.99  
Quantity in stock: 200  
| ~~~~~ |  
? Please enter the #ID of the product you wish to buy now
```

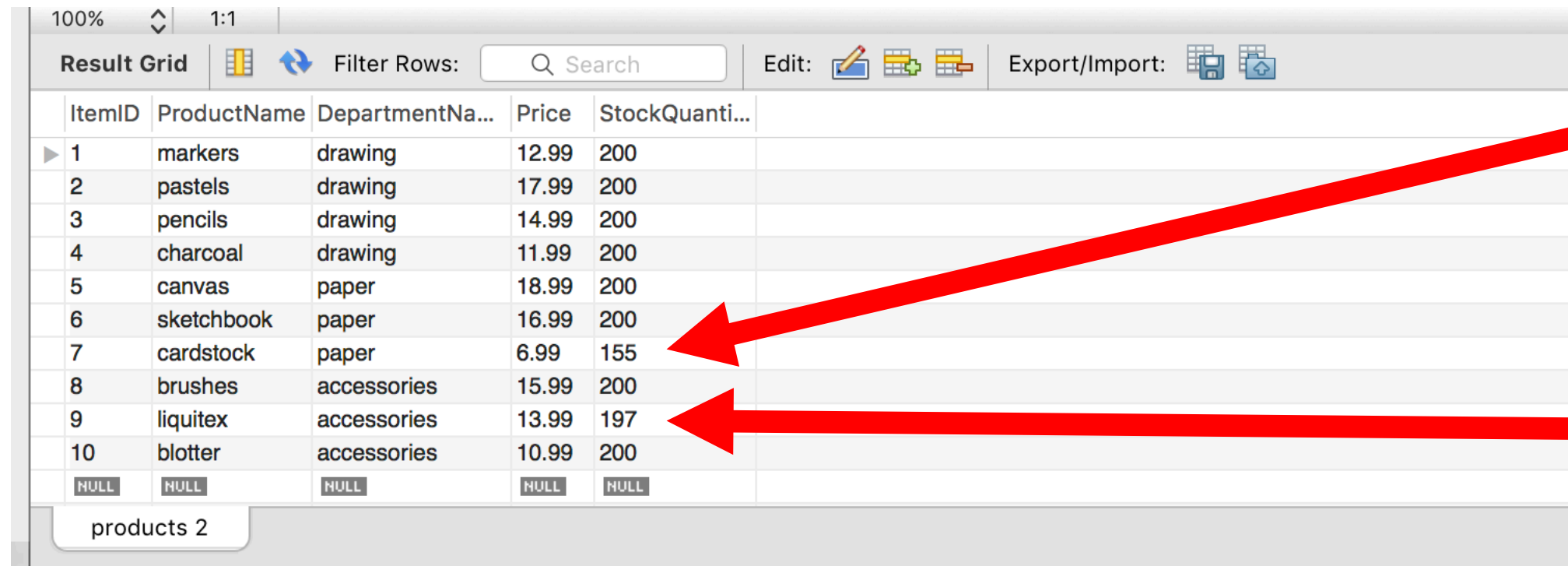


...Next...Prompts and responses for item to be purchased (#10 blotter), quantity (202) and response of “Insufficient quantity” is returned since there were only 200 in stock

```
| ~~~~~|  
#7  
Product: cardstock  
Price: $6.99  
Quantity in stock: 155  
| ~~~~~|  
#8  
Product: brushes  
Price: $15.99  
Quantity in stock: 200  
| ~~~~~|  
#9  
Product: liquitex  
Price: $13.99  
Quantity in stock: 197  
| ~~~~~|  
#10  
Product: blotter  
Price: $10.99  
Quantity in stock: 200  
| ~~~~~|  
? Please enter the #ID of the product you wish to buy now 10  
? Enter quantity desired: 202  
Insufficient quantity
```



MySQL database – shows “products” table after purchases were made and quantity of stock modified



100% 1:1

Result Grid Filter Rows: Search Edit: Export/Import:

ItemID	ProductName	DepartmentNa...	Price	StockQuanti...
1	markers	drawing	12.99	200
2	pastels	drawing	17.99	200
3	pencils	drawing	14.99	200
4	charcoal	drawing	11.99	200
5	canvas	paper	18.99	200
6	sketchbook	paper	16.99	200
7	cardstock	paper	6.99	155
8	brushes	accessories	15.99	200
9	liquitex	accessories	13.99	197
10	blotter	accessories	10.99	200
NULL	NULL	NULL	NULL	NULL

products 2

The End

bamazonCustomer.js