

## LAB03: Subquery

---

### Submission:

- Submit a lab file named “int205\_**lab03**\_xxxxxxxxxxx.docx/.pdf” into the LEB2 system. xxxxxxxxxxxx = your student id

### Due Date & Time:

- Lecturer will inform the LAB02 due date and time in lab class.
- 

### Guideline for writing sub-query

- The subquery (inner query) executes once before the main query (outer query).
- The result of the subquery is used by the main query.
- Enclose subqueries in parentheses.
- Place subqueries on the right side of the comparison condition.
- Use single-row operators (>, <, =, <=, >=, <>) with single-row subqueries (return ONLY one value), and use multiple-row operators (IN, >ANY, >ALL, <ANY, <ALL) with multiple-row subqueries (return one or more values).
- A common problem with subqueries occurs when no rows are returned by the subquery. So, the main query also returns no rows.
- The ORDER BY clause in the subquery is NOT needed unless you are performing Top-N analysis.

### Type 1 – Nested Subquery

- Database evaluates the whole query in two steps:
  - First, execute the subquery (inner query).
  - Second, use the result of the subquery in the parent statement (outer query).

```
select customername
from customers
where creditlimit > ( select creditlimit
                     from customers
                     where customername = 'Land of Toys Inc.' )
```

### Type 2 - Correlated Subquery

- Database evaluated once for each row processed by the parent statement.
  - This operation is used when a subquery refers to a column from a table in an outer query.
  - The unqualified columns in the subquery are resolved by looking in the tables named in the inner query and then in the tables named in the outer query.

```
select lastname
from employees e
where exists (select *
              from offices
              where officecode = e.officecode)
```

### The Syntax of SELECT statement:

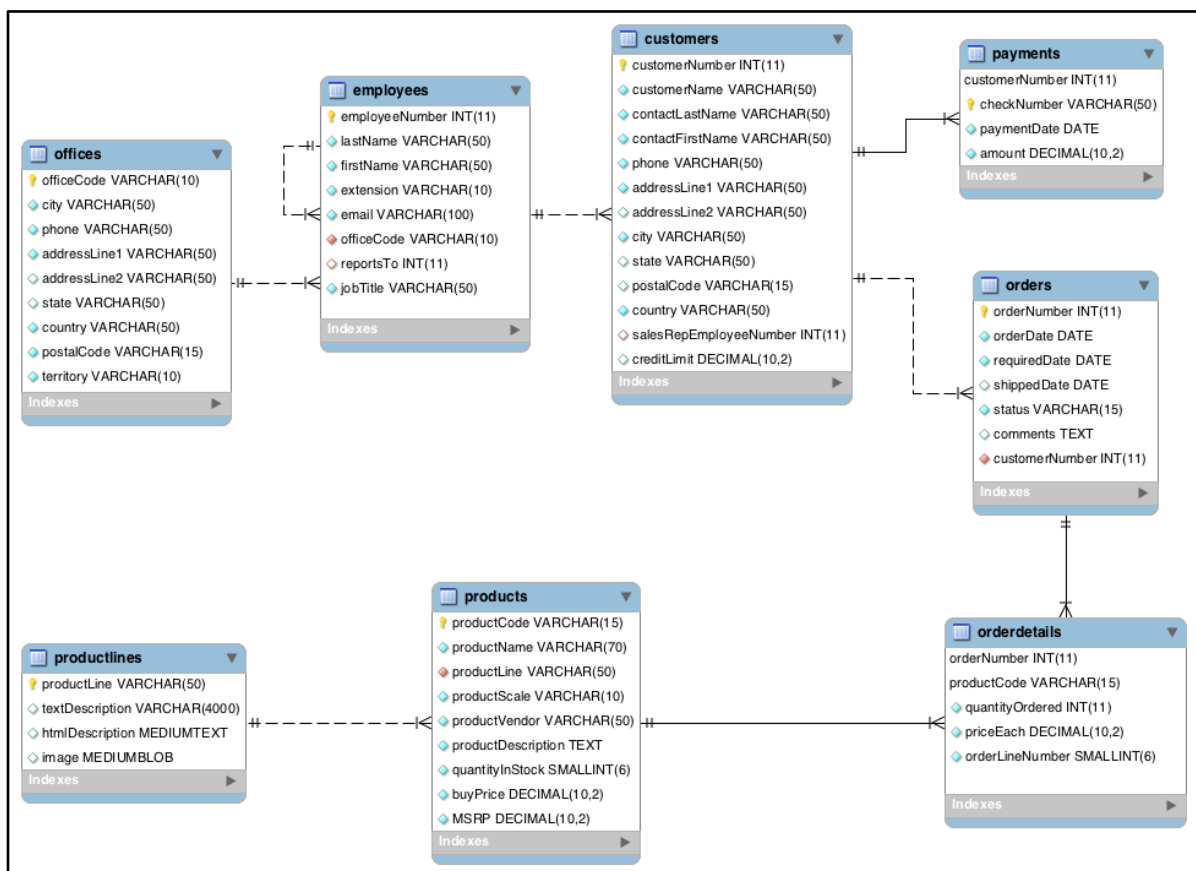
Documentation: <https://dev.mysql.com/doc/refman/8.0/en/select.html>

Note: The MySQL error code 1064 is a syntax error. This means the reason there's a problem is because MySQL doesn't understand what you're asking it to do.

#### Switch to SQL Editor

- You should specify the classicmodels database before writing SQL statements using the following command:  
USE db\_name;

The USE statement tells MySQL to use the named database as the default (current) database for subsequent statements. This statement requires some privilege for the database or some object within it.

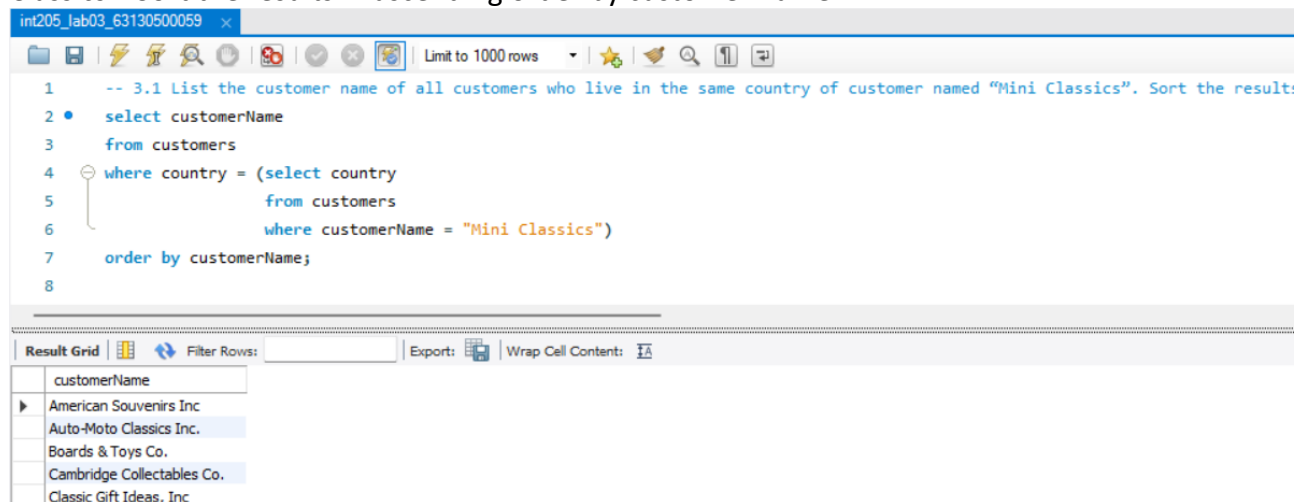


### The ER diagram for the classicmodels.

Note: The MSRP is "Manufacturer's suggested retail price" (ราคาขายปลีกแนะนำของผู้ผลิต).

**Task 1: Using the “classicmodels” database and write SQL statements to answer the following questions.**

3.1 List the customer name of all customers who live in the same country of customer named “Mini Classics”. Sort the results in ascending order by customer name.



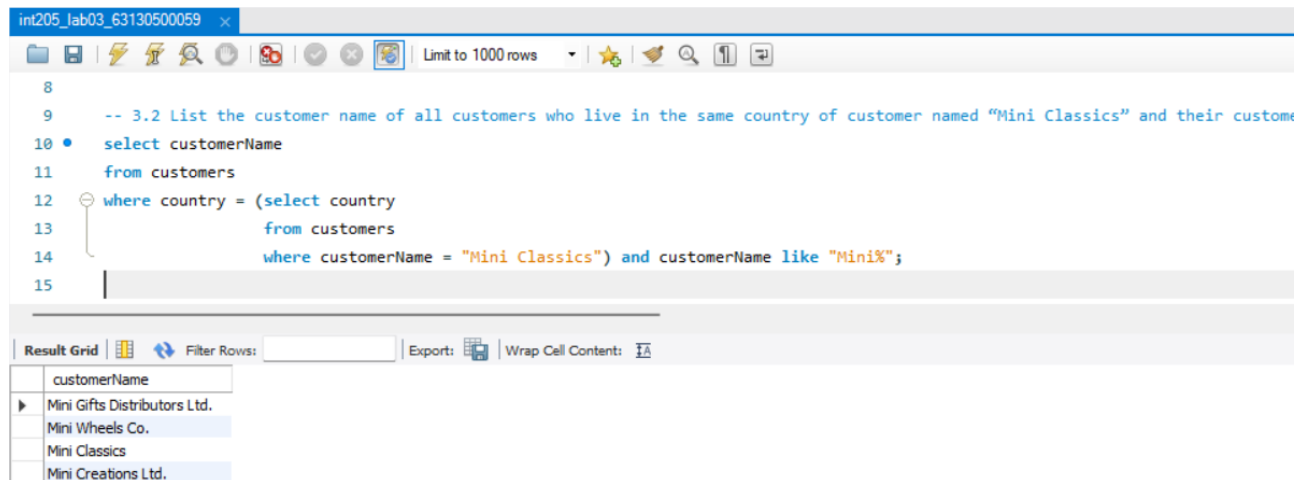
The screenshot shows a SQL IDE window with the following SQL query:

```
1 -- 3.1 List the customer name of all customers who live in the same country of customer named "Mini Classics". Sort the results
2 • select customerName
3   from customers
4  where country = (select country
5                  from customers
6                  where customerName = "Mini Classics")
7  order by customerName;
8
```

The result grid shows the following data:

customerName
American Souvenirs Inc
Auto-Moto Classics Inc.
Boards & Toys Co.
Cambridge Collectables Co.
Classic Gift Ideas, Inc

3.2 List the customer name of all customers who live in the same country of customer named “Mini Classics” and their customer names start with “Mini”.



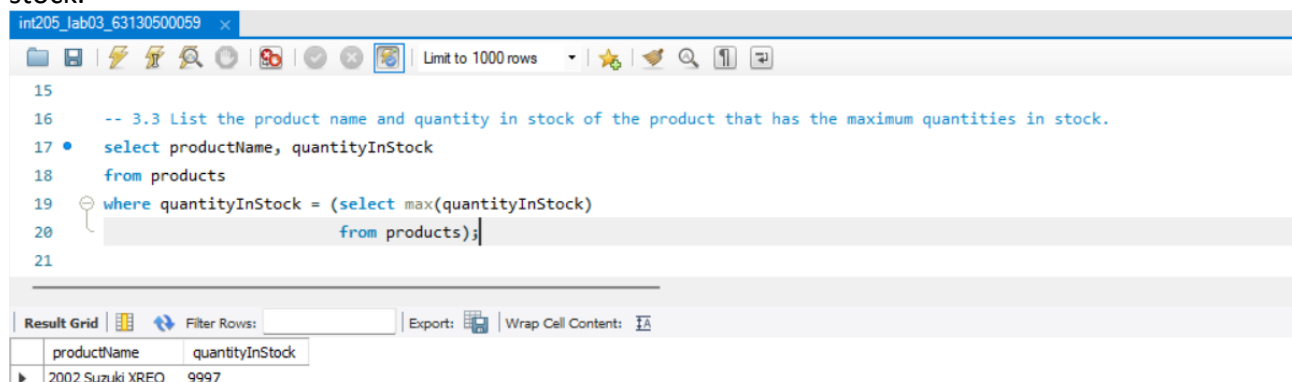
The screenshot shows a SQL IDE window with the following SQL query:

```
8
9 -- 3.2 List the customer name of all customers who live in the same country of customer named "Mini Classics" and their customer
10 • select customerName
11   from customers
12  where country = (select country
13                  from customers
14                  where customerName = "Mini Classics") and customerName like "Mini%";
15
```

The result grid shows the following data:

customerName
Mini Gifts Distributors Ltd.
Mini Wheels Co.
Mini Classics
Mini Creations Ltd.

3.3 List the product name and quantity in stock of the product that has the maximum quantities in stock.



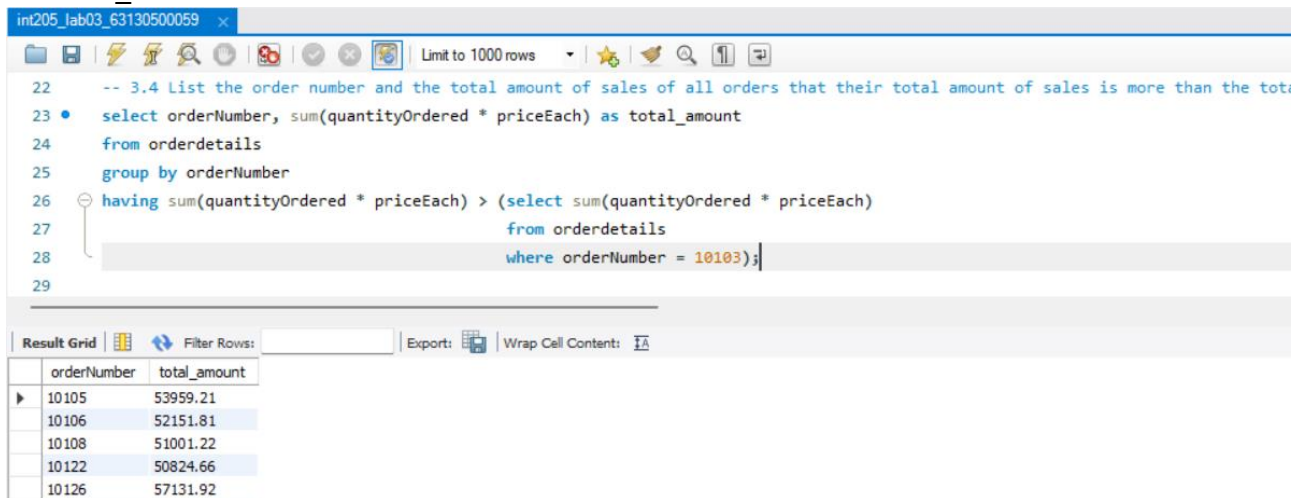
The screenshot shows a SQL IDE window with the following SQL query:

```
15
16 -- 3.3 List the product name and quantity in stock of the product that has the maximum quantities in stock.
17 • select productName, quantityInStock
18   from products
19  where quantityInStock = (select max(quantityInStock)
20                          from products);
21
```

The result grid shows the following data:

productName	quantityInStock
2002 Suzuki XREO	9997

3.4 List the order number and the total amount of sales of all orders that their total amount of sales is more than the total amount of sales order number 10103. Name the total amount of sales column to "total\_amount".



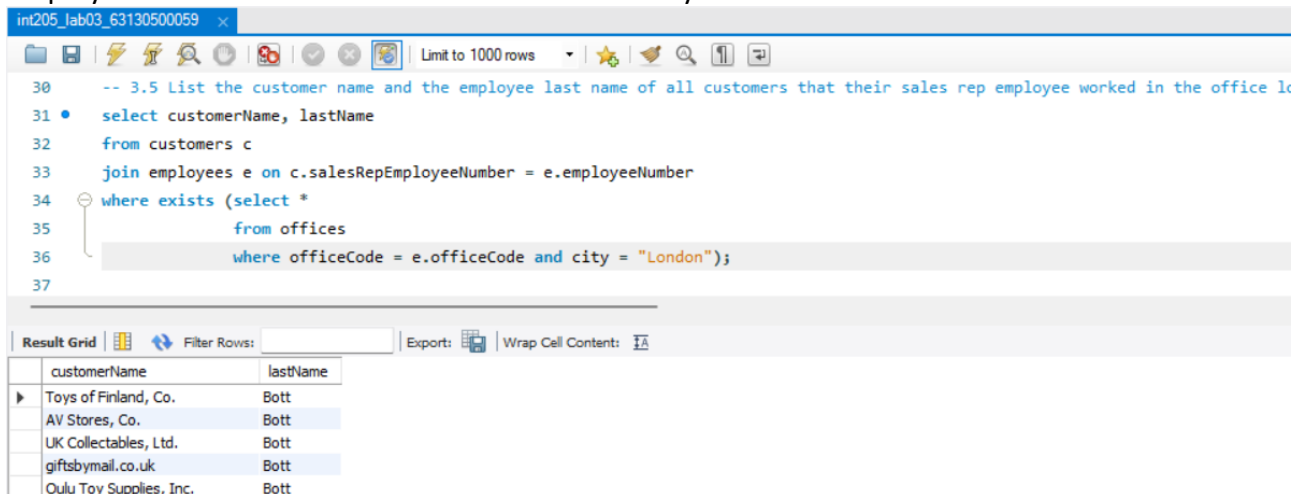
The screenshot shows a SQL IDE window with a query editor and a result grid. The query is as follows:

```
-- 3.4 List the order number and the total amount of sales of all orders that their total amount of sales is more than the total amount of sales order number 10103. Name the total amount of sales column to "total_amount".
select orderNumber, sum(quantityOrdered * priceEach) as total_amount
from orderdetails
group by orderNumber
having sum(quantityOrdered * priceEach) > (select sum(quantityOrdered * priceEach)
from orderdetails
where orderNumber = 10103);
```

The result grid displays the following data:

orderNumber	total_amount
10105	53959.21
10106	52151.81
10108	51001.22
10122	50824.66
10126	57131.92

3.5 List the customer name and the employee last name of all customers that their sales rep employee worked in the office located in London city.



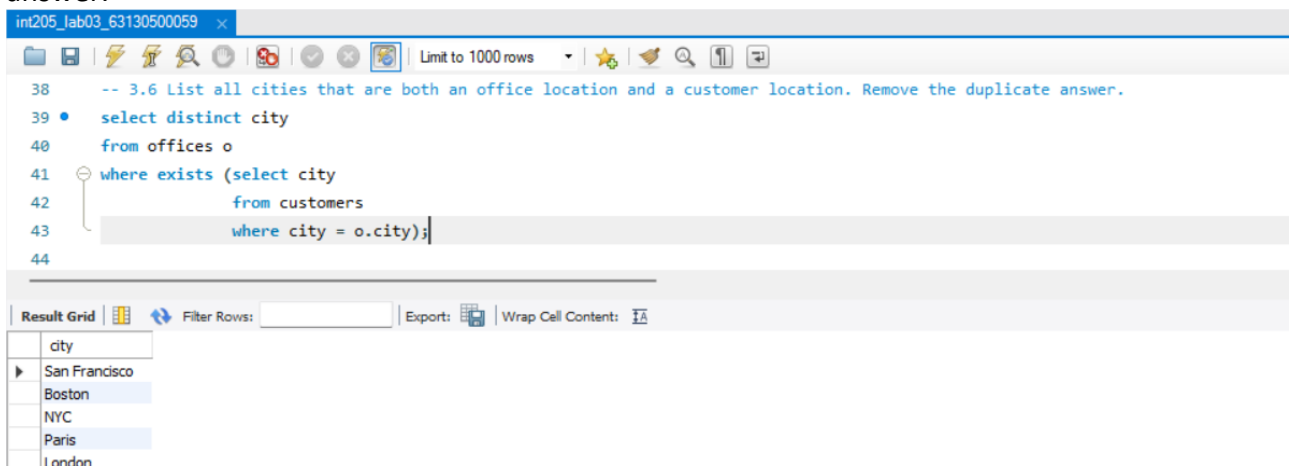
The screenshot shows a SQL IDE window with a query editor and a result grid. The query is as follows:

```
-- 3.5 List the customer name and the employee last name of all customers that their sales rep employee worked in the office located in London city.
select customerName, lastName
from customers c
join employees e on c.salesRepEmployeeNumber = e.employeeNumber
where exists (select *
from offices
where officeCode = e.officeCode and city = "London");
```

The result grid displays the following data:

customerName	lastName
Toys of Finland, Co.	Bott
AV Stores, Co.	Bott
UK Collectables, Ltd.	Bott
giftsbymail.co.uk	Bott
Oulu Toy Supplies, Inc.	Bott

3.6 List all cities that are both an office location and a customer location. Remove the duplicate answer.



The screenshot shows a SQL IDE window with a query editor and a result grid. The query is as follows:

```
-- 3.6 List all cities that are both an office location and a customer location. Remove the duplicate answer.
select distinct city
from offices o
where exists (select city
from customers
where city = o.city);
```

The result grid displays the following data:

city
San Francisco
Boston
NYC
Paris
London

3.7 List all cities where customers who do not live in the same city of their sales rep employee's office city. Remove the duplicate answer.

The screenshot shows a SQL IDE window with a query editor and a result grid. The query editor contains the following SQL code:

```
45 -- 3.7 List all cities where customers who do not live in the same city of their sales rep employee's office city. Remove the
46 • select distinct city
47   from customers c
48  where not exists (select city
49                   from employees e
50                  join offices o on e.officeCode = o.officeCode
51                  where employeeNumber = c.salesRepEmployeeNumber);
52
```

The result grid shows the following data:

city
Warszawa
Lisboa
Singapore
Cunewalde
Madrid

3.8 List the customer name of all customers who have a credit limit greater than all average credit limits of customers in each city.

The screenshot shows a SQL IDE window with a query editor and a result grid. The query editor contains the following SQL code:

```
53 -- 3.8 List the customer name of all customers who have a credit limit greater than all average credit limits of customers in e
54 • select customerName
55   from customers
56  where creditLimit >all (select avg(creditLimit)
57                        from customers
58                        group by city);
59
```

The result grid shows the following data:

customerName
Euro+ Shopping Channel

3.9 List the customer name of all customers who have a credit limit greater than their average credit limits of customers in their cities.

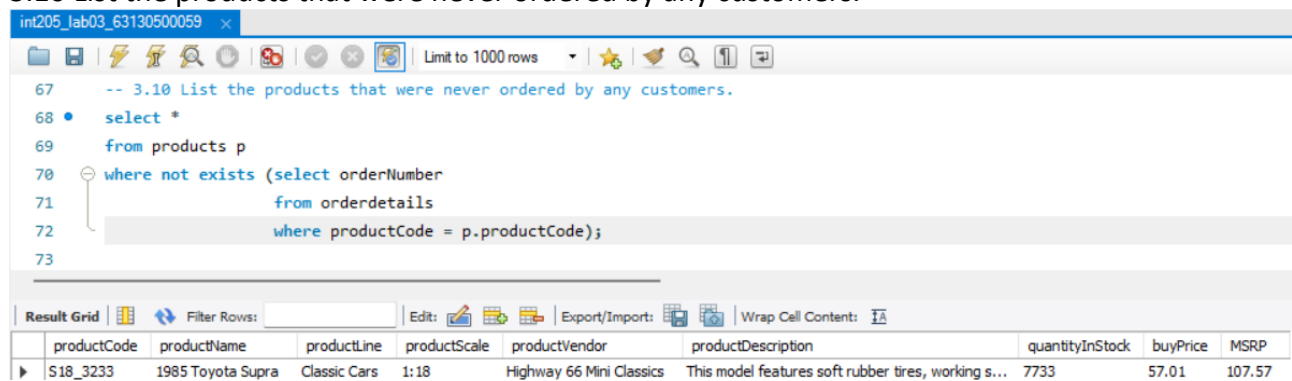
The screenshot shows a SQL IDE window with a query editor and a result grid. The query editor contains the following SQL code:

```
60 -- 3.9 List the customer name of all customers who have a credit limit greater than their average credit limits of customers in
61 • select customerName
62   from customers c
63  where creditLimit > (select avg(creditLimit)
64                    from customers
65                    where city = c.city);
66
```

The result grid shows the following data:

customerName
La Rochelle Gifts
Blauer See Auto, Co.
Land of Toys Inc.
Euro+ Shopping Channel
Dragon Souvenirs, Ltd.

3.10 List the products that were never ordered by any customers.



The screenshot shows a database management tool interface. At the top, there's a tab labeled 'int205\_Jab03\_63130500059'. Below the tab is a toolbar with various icons. The main area displays a SQL query:

```
67 -- 3.10 List the products that were never ordered by any customers.  
68 • select *  
69 from products p  
70 where not exists (select orderNumber  
71                  from orderdetails  
72                  where productCode = p.productCode);  
73
```

Below the query editor is a 'Result Grid' section. It includes a 'Filter Rows:' input field and buttons for 'Edit', 'Export/Import', and 'Wrap Cell Content:'. The grid displays the following data:

productCode	productName	productLine	productScale	productVendor	productDescription	quantityInStock	buyPrice	MSRP
S18_3233	1985 Toyota Supra	Classic Cars	1:18	Highway 66 Mini Classics	This model features soft rubber tires, working s...	7733	57.01	107.57