|  |
| --- |
| Close-up image showing the leaf-sides of two oversized books side-by-side on a bookshelf, with additional books in soft focus background |
| GMCA 5– Day 1 MySQL Tasks  Rafal Szymanski |
| |  |  |  | | --- | --- | --- | | Rafal Szymanski | 4/17/23 | GMCA 5 | |

Table of Contents

[Query 1: List of Customers in ‘sql\_store’ database 2](#_Toc132707721)

[Query 2: Customers by first and last name with points 2](#_Toc132707722)

[Task 1: Change of Points 3](#_Toc132707723)

[X10 3](#_Toc132707724)

[+100 3](#_Toc132707725)

[(points+ 10) \*100 3](#_Toc132707726)

[Task 2: Products by name, unit price and new\_price +10% 4](#_Toc132707727)

[Task 3: Customers with a birth date of > '1990-01-01' 4](#_Toc132707728)

[Task 4: Product with the most quantity in ‘sql\_inventory’ database 5](#_Toc132707729)

[1. Name of the product with the most quantity in stock (order by + limit) 5](#_Toc132707730)

[2. Name of the product with the most quantity in stock (nested SELECT) 5](#_Toc132707731)

[Task 5: The most expensive product in ‘sql\_inventory’ database 6](#_Toc132707732)

[1. Name of the most expensive product (order by + limit) 6](#_Toc132707733)

[2. Name of the most expensive product (nested SELECT) 6](#_Toc132707734)

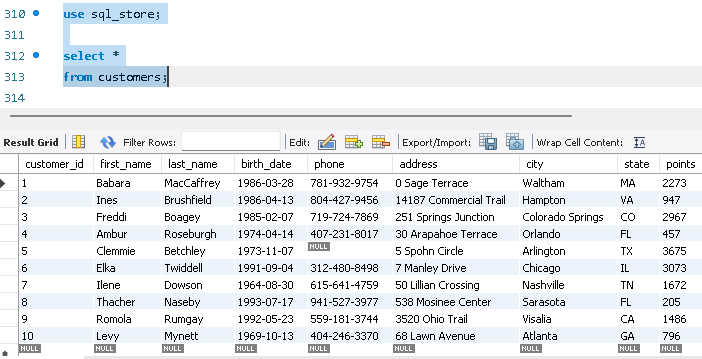
[Task 6: The oldest customer in ‘sql\_store’ database 7](#_Toc132707735)

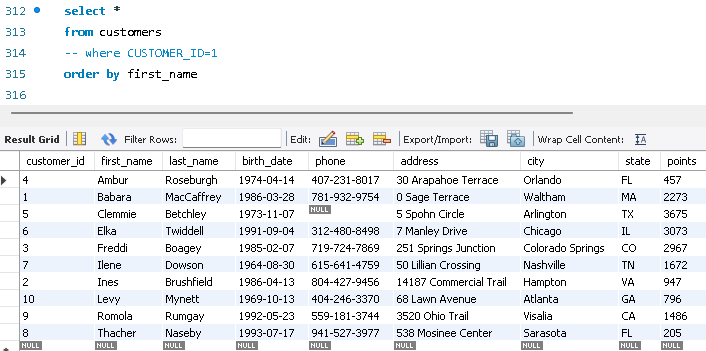
[1. Details of the oldest customer (order by + limit) 7](#_Toc132707736)

[2. Details of the oldest customer (nested SELECT) 7](#_Toc132707737)

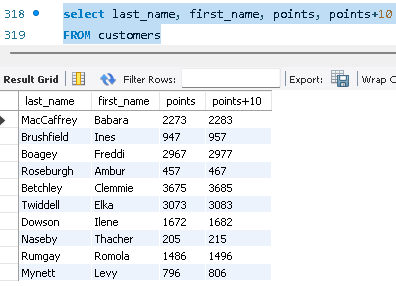
[Creating an EER Diagram 8](#_Toc132707738)

# Query 1: List of Customers in ‘sql\_store’ database

First query was to start using sql\_store database and to select all the details from customers’ table:

Then we needed to list all the customers in order of their first name:

# Query 2: Customers by first and last name with points

Here we had to list customers last and first name with their points and also add 10 to their points in a separate column:

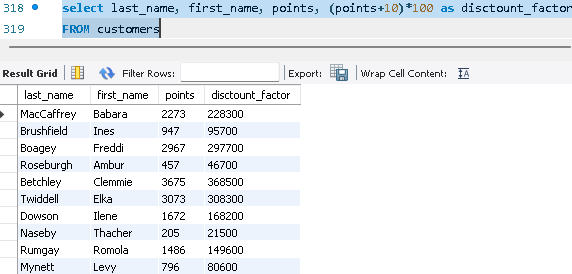
# Task 1: Change of Points

Changing the points to read times by 10 and plus 100 and to create a discount factor so the table now shows a discount header and changing the (point + 10) \*100

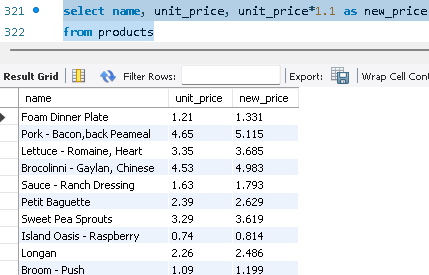
## X10

## +100

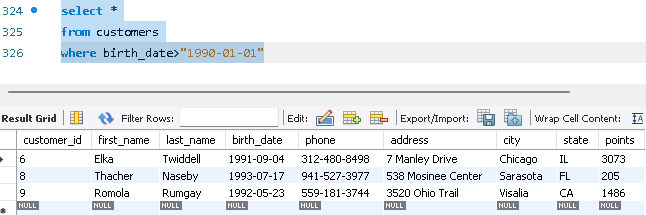
## (points+ 10) \*100



# Task 2: Products by name, unit price and new\_price +10%

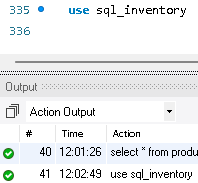
In this task we had to write a query to return all the products in our database to show columns; name, unit price, and new column called new price which is based on this expression, (unit price \* 1.1 )

# Task 3: Customers with a birth date of > '1990-01-01'

A new query to find all the customers with a birth date of > '1990-01-01':

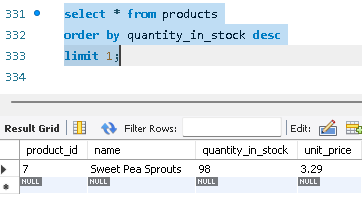
# Task 4: Product with the most quantity in ‘sql\_inventory’ database

Using sql\_inventory database:

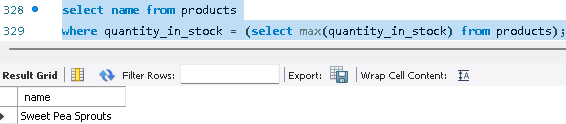


And finding out the name of the product with the most quantity in stock can be done in two ways. First showing all the details of the product, and second way using nested SELECT query displaying only the name of the product.

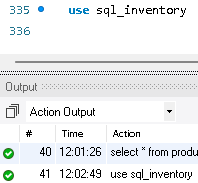
## Name of the product with the most quantity in stock (order by + limit)



## Name of the product with the most quantity in stock (nested SELECT)

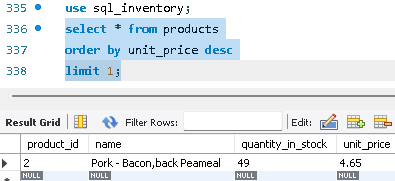


# Task 5: The most expensive product in ‘sql\_inventory’ database

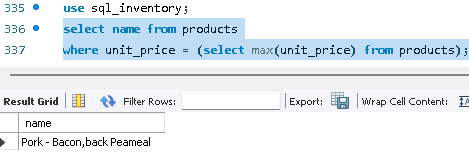
Using sql\_inventory database:

Similarly, to Task 4, to find out the name of the most expensive product we can do it in two ways, the second with a nested SELECT command.

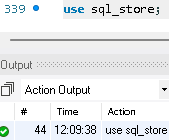
## Name of the most expensive product (order by + limit)



## Name of the most expensive product (nested SELECT)

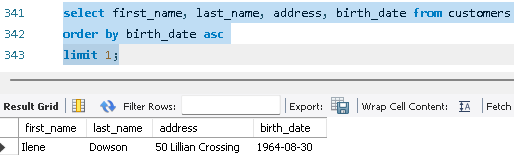


# Task 6: The oldest customer in ‘sql\_store’ database

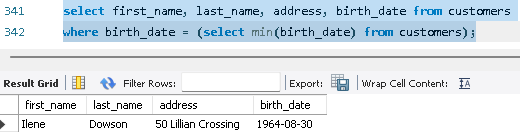
Using sql\_store database:

Again, finding out the details of the oldest customer can be done in two ways, using order by and nested SELECT command:

## Details of the oldest customer (order by + limit)



## Details of the oldest customer (nested SELECT)



# Creating an EER Diagram

