**SQL Assignment Part 1 – Results**

1. **Task 1**

* Using the Query 2 you created change the points to reads times by 10 and plus 100.

|  |  |  |  |
| --- | --- | --- | --- |
| **last\_name** | **first\_name** | **points** | **(points \* 10) + 100** |
| MacCaffrey | Babara | 2,273 | 22,830 |
| Brushfield | Ines | 947 | 9,570 |
| Boagey | Freddi | 2,967 | 29,770 |
| Roseburgh | Ambur | 457 | 4,670 |
| Betchley | Clemmie | 3,675 | 36,850 |
| Twiddell | Elka | 3,073 | 30,830 |
| Dowson | Ilene | 1,672 | 16,820 |
| Naseby | Thacher | 205 | 2,150 |
| Rumgay | Romola | 1,486 | 14,960 |
| Mynett | Levy | 796 | 8,060 |

* Change the Query 2 code to create a discount factor so the table now shows a discount header and changing the (point + 10) \*100

|  |  |  |  |
| --- | --- | --- | --- |
| **last\_name** | **first\_name** | **points** | **discount\_factor** |
| MacCaffrey | Babara | 2,273 | 228,300 |
| Brushfield | Ines | 947 | 95,700 |
| Boagey | Freddi | 2,967 | 297,700 |
| Roseburgh | Ambur | 457 | 46,700 |
| Betchley | Clemmie | 3,675 | 368,500 |
| Twiddell | Elka | 3,073 | 308,300 |
| Dowson | Ilene | 1,672 | 168,200 |
| Naseby | Thacher | 205 | 21,500 |
| Rumgay | Romola | 1,486 | 149,600 |
| Mynett | Levy | 796 | 80,600 |

1. **Task 2**

* Write a SQL query to return all the products in our database in the result set. We have to show columns; name, unit price, and new column called new price which is based on this expression, (unit price \* 1.1 ).
* So what you are doing is increasing the product price of each by 10%.
* So with the query we want all the products the original price and the new price.

|  |  |  |
| --- | --- | --- |
| **name** | **unit\_price** | **new\_price** |
| Foam Dinner Plate | 1.21 | 1.331 |
| Pork - Bacon,back Peameal | 4.65 | 5.115 |
| Lettuce - Romaine, Heart | 3.35 | 3.685 |
| Brocolinni - Gaylan, Chinese | 4.53 | 4.983 |
| Sauce - Ranch Dressing | 1.63 | 1.793 |
| Petit Baguette | 2.39 | 2.629 |
| Sweet Pea Sprouts | 3.29 | 3.619 |
| Island Oasis - Raspberry | 0.74 | 0.814 |
| Longan | 2.26 | 2.486 |
| Broom - Push | 1.09 | 1.199 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **customer\_id** | **first\_name** | **last\_name** | **birth\_date** | **phone** | **address** | **city** | **state** | **points** |
| 6 | Elka | Twiddell | 04/09/1991 | 312-480-8498 | 7 Manley Drive | Chicago | IL | 3073 |
| 8 | Thacher | Naseby | 17/07/1993 | 941-527-3977 | 538 Mosinee Center | Sarasota | FL | 205 |
| 9 | Romola | Rumgay | 23/05/1992 | 559-181-3744 | 3520 Ohio Trail | Visalia | CA | 1486 |

1. **Task 3**

* In this task create a new query to find all the customers with a birth date of > '1990-01-01'

1. **Task 4**

* Select sql\_inventory.
* Write a query to find out the name of the product with most amount in stock:

|  |  |
| --- | --- |
| **name** | **quantity\_in\_stock** |
| Sweet Pea Sprouts | 98 |

1. **Task 5**

* Select sql\_inventory.
* Write a query to find out the name of the most expensive product.

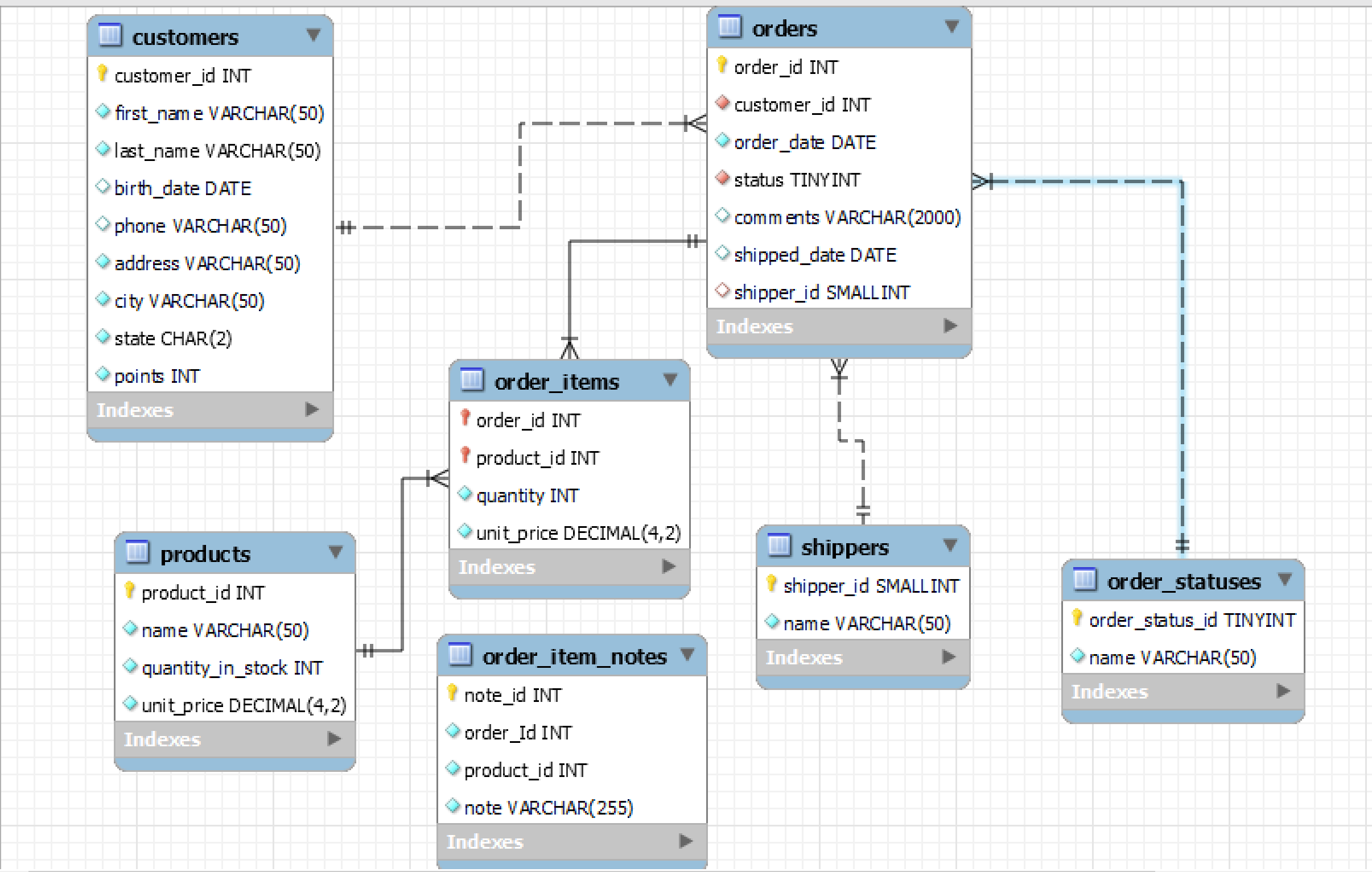
|  |  |
| --- | --- |
| **name** | **unit\_price** |
| Pork - Bacon,back Peameal | 4.65 |

1. **Task 6**

* Select sql\_store.
* Write a query to find out the first name, last name, address and the birthdate of the oldest customer.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **first\_name** | **last\_name** | **birth\_date** | **age** | **address** |
| Ilene | Dowson | 30/08/1964 | 59 | 50 Lillian Crossing |

1. **Task 7:** Entity Relation Diagram (ERD) of sql-store database

****

**Tables and their primary keys:**

|  |  |
| --- | --- |
| **Table name** | **Primary key(s)** |
| customers | customer\_id |
| order | order\_id |
| products | product\_id |
| order\_items | order\_id, product\_id |
| shippers | shipper\_id |
| order\_statuses | order\_status\_id |
| order\_item\_notes | note\_id |

**Relationships between Tables:**

1. customers and orders: The relationship is one to many. The customer\_id is a foreign key in the orders table.
2. orders and order\_item: The relationship is one to many. The order\_id is a foreign key in the order\_items table.
3. products and order\_items: The relationship is one to many. The product\_key is a foreign key in the order\_items table.
4. shippers and orders: The relationship is one to many. The shipper\_id is a foreign key in the orders table.
5. order\_statuses and orders: The relationship is one to many. The order\_status\_id is a foreign key in the orders table.