*Please note: as per the assignment’s instructions, query result tables longer than 5 lines were truncated at the 5th row.*

Question #1:

/\* a \*/

insert into customer\_T (customerid, customername, customeraddress, customercity, customerstate, customerpostalcode)

values (27, 'Tyler DiNapoli-Chiappelli', '140 West 62nd Street', 'New York', 'NY', '10023');

/\* b \*/

insert into doesbusinessin\_T (customerid, territoryid)

values (27, 3);

/\* c \*/

insert into customershipaddress\_T (shipaddressid, customerid, territoryid, shipaddress, shipcity, shipstate, shipzip)

values (35, 27, 3, '0000 Broadway', 'New York', 'NY', '10025');

/\* d \*/

insert into order\_T (orderid, customerid, orderdate, fulfillmentdate, salespersonid)

values (93, 27, to\_date('09/23/2020','mm/dd/yyyy') ,null, 3);

/\* e \*/

insert into orderline\_T (orderlineid, orderid, productid, orderedquantity)

values (100, 93, 3, 2);

Question #2:

**5-57 query:**

alter table Product\_T

add Qty\_On\_Hand int

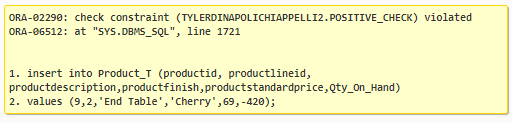
add constraint length\_check check (length(Qty\_On\_Hand)=5)

add constraint positive\_check check (Qty\_On\_Hand > 0);

**5-58 query and “failure” result:**

insert into Product\_T (productid, productlineid, productdescription,productfinish,productstandardprice,Qty\_On\_Hand)

values (9,2,'End Table','Cherry',69,-420);



**Query to select all columns from Product\_T table:**

select \*

from product\_t;

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **PRODUCTID** | **PRODUCTLINEID** | **PRODUCTDESCRIPTION** | **PRODUCTFINISH** | **PRODUCTSTANDARDPRICE** | **PRODUCTONHAND** |
| 1 | 1 | Cherry End Table | Cherry | 175 | 0 |
| 2 | 1 | Birch Coffee Tables | Birch | 200 | 0 |
| 3 | 1 | Oak Computer Desk | Oak | 750 | 0 |
| 4 | 1 | Entertainment Center | Cherry | 1650 | 0 |
| 5 | 2 | Writer's Desk | Oak | 325 | 0 |

/\* Question #3 \*/

/\* a: Please note, per the book's instructions, this view creation query and following table query were copied and pasted, with only minor edits, directly from page 270 \*/

CREATE VIEW TSales AS

SELECT SalespersonName, ProductDescription, SUM(OrderedQuantity) AS Totorders

FROM Salesperson\_T, Orderline\_T, Product\_T, Order\_T

WHERE Salesperson\_T.SalespersonID = Order\_T.SalespersonID

AND Order\_T.OrderID = Orderline\_T.OrderID

AND Orderline\_T.ProductID = Product\_T.ProductID

GROUP BY SalespersonName, ProductDescription;

---------------------------------------------------------------

select salespersonname, productdescription, totorders

from TSales A

where totorders = (

select max(totorders)

from TSales B

where A.salespersonname = B.salespersonname);

|  |  |  |
| --- | --- | --- |
| **SALESPERSONNAME** | **PRODUCTDESCRIPTION** | **TOTORDERS** |
| Julie Dawson | Writer's Desk | 20 |
| William Strong | Birch Coffee Tables | 26 |
| Jacob Winslow | Writer's Desk | 3 |
| Jacob Winslow | 8-Drawer Dresser | 3 |
| Fred Flinstone | Nightstand | 2 |

/\* b: Please note, though the book's instructions on page 271 stated to use their table creation query, it was riddled with errors and it was more efficient to re-write my own \*/

create view TopTerritory as

select sale.salespersonname, dbi.territoryid, prod.productdescription, sum(ordlin.orderedquantity) as End\_Table\_Quantity\_by\_Territory

from salesperson\_T sale

left outer join order\_T ord

on sale.salespersonid = ord.salespersonid

left outer join orderline\_T ordlin

on ord.orderid = ordlin.orderid

left outer join product\_T prod

on ordlin.productid = prod.productid

left outer join doesbusinessin\_T dbi

on ord.customerid = dbi.customerid

where prod.productdescription like '%End Table%'

group by dbi.territoryid, prod.productdescription, sale.salespersonname;

----------------------------------------------------------

select \*

from topterritory

where End\_Table\_Quantity\_by\_Territory = (

select max(End\_Table\_Quantity\_by\_Territory)

from topterritory);

|  |  |  |  |
| --- | --- | --- | --- |
| **SALESPERSONNAME** | **TERRITORYID** | **PRODUCTDESCRIPTION** | **END\_TABLE\_QUANTITY\_BY\_TERRITORY** |
| William Strong | 4 | Cherry End Table | 9 |

4a:

*/\* Please note: a manual look into the Territory\_T table showed that "Northeast" is territoryid 3 \*/*

select ord.orderid, ord.fulfillmentdate, custship.territoryid

from order\_T ord

left outer join customershipaddress\_T custship

on ord.customerid = custship.customerid

where custship.territoryid = 3 and ord.fulfillmentdate is null;

|  |  |  |
| --- | --- | --- |
| **ORDERID** | **FULFILLMENTDATE** | **TERRITORYID** |
| 93 | - | 3 |

4b:

**5-63:**

select customername, customerstate, customerpostalcode

from customer\_T

where customerstate in ('CA','WA')

order by customerpostalcode desc;

|  |  |  |
| --- | --- | --- |
| **CUSTOMERNAME** | **CUSTOMERSTATE** | **CUSTOMERPOSTALCODE** |
| Impressions | CA | 94206-4056 |

**5-67:**

select productlineid, round(avg(productstandardprice),2) as "Average\_Price"

from product\_T

group by productlineid;

|  |  |
| --- | --- |
| **PRODUCTLINEID** | **Average\_Price** |
| 1 | 581.22 |
| 2 | 608.33 |
| 4 | 995 |
| 5 | 165 |
| 3 | 222.4 |

**5-68:**

select productlineid, round(avg(productstandardprice),2) as "Average\_Price"

from product\_T

where productstandardprice > 200

group by productlineid

having avg(productstandardprice) >= 500;

|  |  |
| --- | --- |
| **PRODUCTLINEID** | **Average\_Price** |
| 1 | 784.33 |
| 2 | 608.33 |
| 4 | 995 |

**5-77:**

select cust.customerid, count(ord.orderid) as "Total\_Orders"

from customer\_T cust

left outer join order\_T ord

on cust.customerid = ord.customerid

group by cust.customerid

having count(ord.orderid) > 2;

|  |  |
| --- | --- |
| **CUSTOMERID** | **Total\_Orders** |
| 6 | 3 |
| 1 | 8 |
| 15 | 3 |
| 12 | 5 |
| 4 | 28 |

**6-46:**

select count(prod.productid) as Products\_in\_Line, sub.productlinename, sub.Average\_Product\_Price\_Per\_Line

from

(select prod.productlineid, pl.productlinename, round(avg(prod.productstandardprice), 2) as Average\_Product\_Price\_Per\_Line

from productline\_T pl

left outer join product\_T prod

on pl.productlineid = prod.productlineid

group by pl.productlinename, prod.productlineid) sub

left outer join product\_T prod

on prod.productlineid = sub.productlineid

group by productlinename, Average\_Product\_Price\_Per\_Line;

|  |  |  |
| --- | --- | --- |
| **PRODUCTS\_IN\_LINE** | **PRODUCTLINENAME** | **AVERAGE\_PRODUCT\_PRICE\_PER\_LINE** |
| 2 | Futuristic | 165 |
| 2 | Classical | 995 |
| 9 | Basic | 581.22 |
| 0 | Spanish Style | - |
| 5 | Modern | 222.4 |

**6-59:**

select distinct ord.customerid, prodline.productlinename

from order\_T ord

left outer join orderline\_T ordlin

on ord.orderid = ordlin.orderid

left outer join product\_T prod

on ordlin.productid = prod.productid

left outer join productline\_T prodline

on prod.productlineid = prodline.productlineid

where prodline.productlinename = 'Basic' and extract (month from ord.orderdate) = 3 and extract (year from ord.orderdate) = 2018;

|  |  |
| --- | --- |
| **CUSTOMERID** | **PRODUCTLINENAME** |
| 16 | Basic |
| 13 | Basic |
| 1 | Basic |
| 4 | Basic |
| 9 | Basic |

**6-64:**

select work.workcenterid, count(prod\_in.productid) as Total\_Products

from producedin\_T prod\_in

right outer join workcenter\_T work

on prod\_in.workcenterid = work.workcenterid

group by work.workcenterid;

(query output below page break)

|  |  |
| --- | --- |
| **WORKCENTERID** | **TOTAL\_PRODUCTS** |
| SM1 | 0 |
| Tampa1 | 14 |
| WR1 | 0 |

**6-69:**

select cust.customerid, cust.customername, ord.orderid

from customer\_T cust

left outer join order\_T ord

on cust.customerid = ord.customerid;

|  |  |  |
| --- | --- | --- |
| **CUSTOMERID** | **CUSTOMERNAME** | **ORDERID** |
| 27 | Tyler DiNapoli-Chiappelli | 93 |
| 4 | Eastern Furniture | 1 |
| 3 | Home Furnishings | 2 |
| 1 | Contemporary Casuals | 3 |
| 6 | Furniture Gallery | 4 |

**6-78:**

select prod.productid, prod.productlineid, prod.productdescription, prod.productstandardprice, sub.Average\_Price\_By\_Product\_Line

from product\_T prod

left outer join (

select productlineid, round(avg(productstandardprice), 2) as Average\_Price\_By\_Product\_Line

from product\_T

group by productlineid) sub

on prod.productlineid = sub.productlineid

where prod.productstandardprice < sub.Average\_Price\_By\_Product\_Line;

(query output below page break)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **PRODUCTID** | **PRODUCTLINEID** | **PRODUCTDESCRIPTION** | **PRODUCTSTANDARDPRICE** | **AVERAGE\_PRICE\_BY\_PRODUCT\_LINE** |
| 1 | 1 | Cherry End Table | 175 | 581.22 |
| 2 | 1 | Birch Coffee Tables | 200 | 581.22 |
| 11 | 1 | 4-Drawer Dresser | 500 | 581.22 |
| 13 | 1 | Nightstand | 150 | 581.22 |
| 21 | 1 | Pine End Table | 256 | 581.22 |