Himadri Sekhar Tewari

IST 659

Project Deliverable 1 & Project Deliverable 2

**Database for an Online Pizza Company**

**Project Summary:**

I want to build a database for an online pizza company. It start with a guest customer entry details or new customer register first time in the webpage by providing the name, address, email and phone number. After successful registration, customer can select pizzas of different sizes and toppings limited to 10 toppings. Customer can select other products present from the product page and can increase or decrease the quantity or remove the item. Only one coupon can be added to an order. A coupon can be either order type or item type. After selecting one or more products and adding in cart, customer can choose the order destination as either carryout or delivery. After selecting the destination, customer can choose the order ready date as current date or future date or future time. After selecting the date/time, customer need to select payment methods (one or more payments are applicable). The different type of payment types is Cash or Credit Card. When selecting cash, customer need to enter only the amount. For credit cards, customer has to enter the card number, expiration date, CVV, name in the card, billing address. Once the card number is entered, card type will be auto selected and the correct card name will be displayed. Customer has the option to save a maximum of 3 cards in the customer account. After selecting the payments, customer has the option to review the order before confirmation. click on Confirm Order button and an order confirmation number is generated in the webpage and an email is sent to the customer in the registered/given email.

Question 1:

Use the database to find out the most popular product

Question 2:

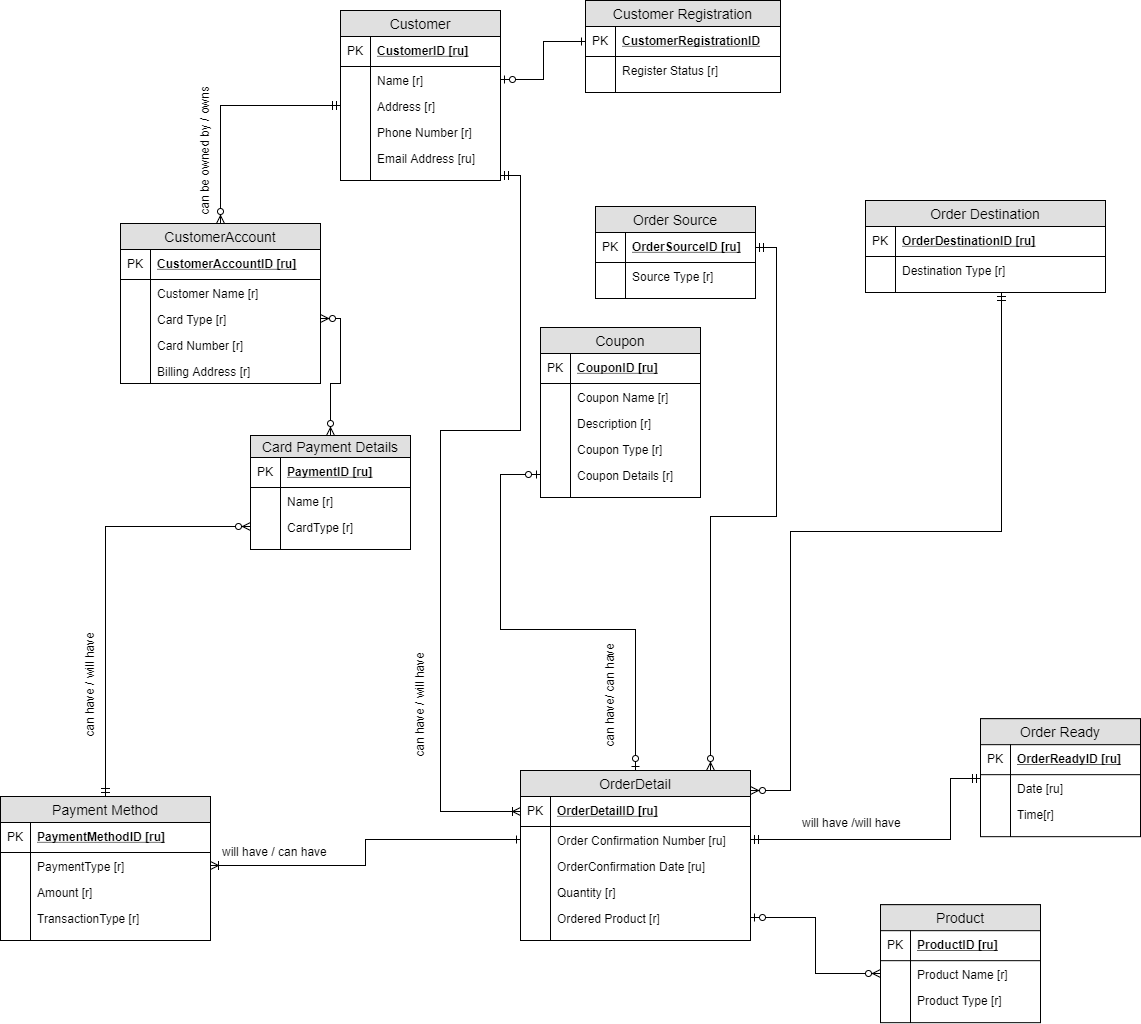
Use the database to compare the sale of current year to last year

Question 3:

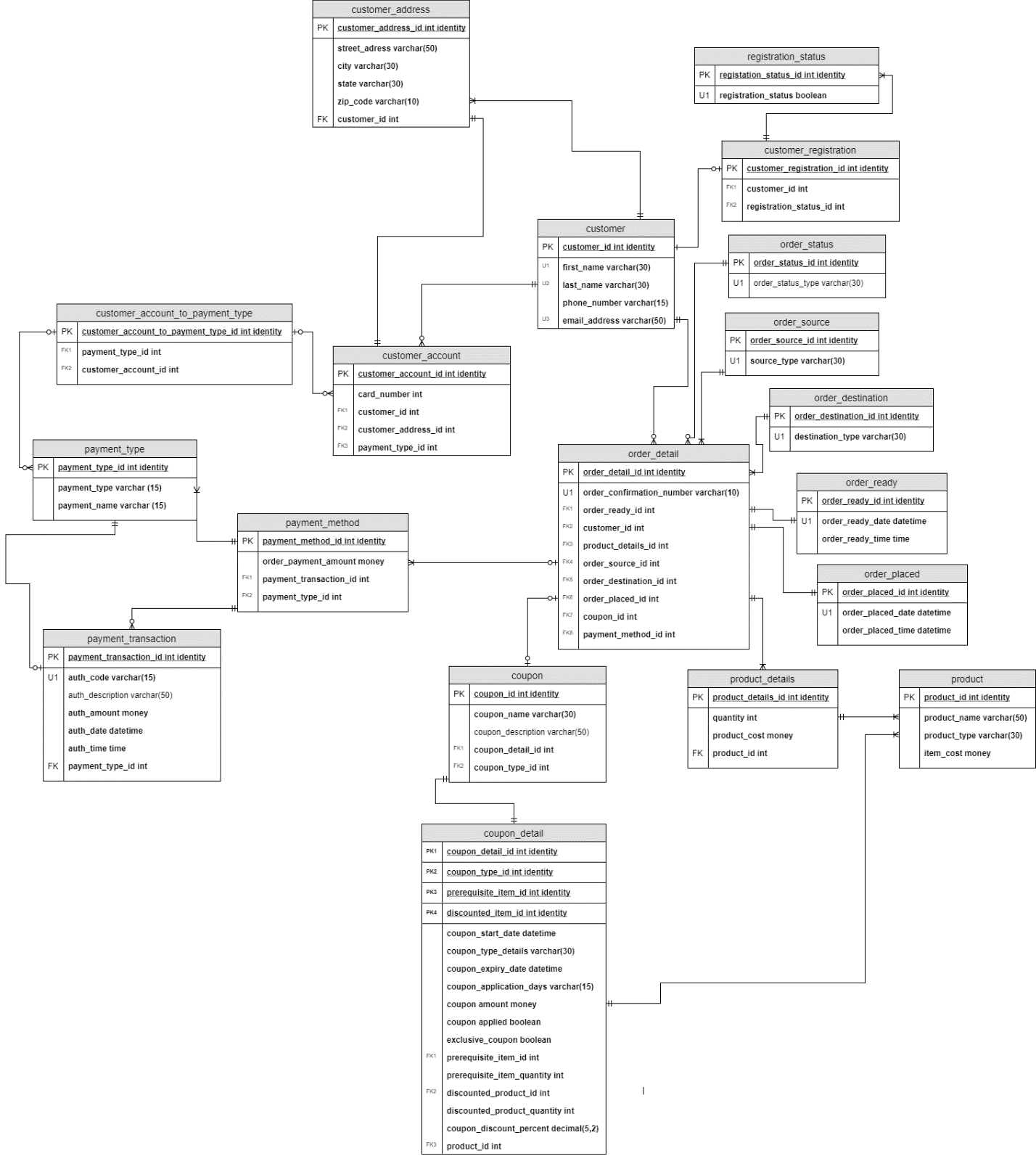
Comparison between guest and registered customer

|  |  |
| --- | --- |
| **Entity** | **Attribute** |
| Customer | CustomerID [ru]  Name [r]  Address[r]  Phone Number [r]  Email address [ru] |
| Product | ProductID [ru]  Product Name [r]  Product Type [r] |
| Coupon | CouponID [ru]  Coupon Name [r]  Description [r]  Coupon Type [r]  Coupon Details [r] |
| Order Source | OrderSourceID [ru]  Source Type [r] |
| Order Destination | OrderDestinationID [ru]  Destination Type [r] |
| Order Ready | OrderReadyID [ru]  Date [ru]  Time [r] |
| Payment Method | PaymentMethodID [ru]  PaymentType [r]  Amount [r]  TransactionType [r] |
| Card Payment Details | PaymentID [ru]  Name [r]  Card Type [r] |
| Customer Account | CustomerAccountID [ru]  Customer Name [r]  Card Type [r]  Card Number [r]  Billing Address [r] |
| Order Detail | OrderDetailID [ru]  Order Confirmation Number [ru]  Order Confirmation Date [ru]  Quantity [r]  Ordered Product [r] |

**Conceptual Model:**



**Normalized Logical Model:**

**Project Deliverable 2:**

**SQL DDL Commands**

-- Creating customer table

create table customer(

customer\_ID int identity

,first\_name varchar(30) not null

, last\_name varchar(30) not null

, phone\_number varchar(15) not null

, email\_address varchar(50) not null

-- Constraint on the customer table

constraint PK\_customer primary key (customer\_ID),

constraint U1\_customer unique (first\_name),

constraint U2\_customer unique (last\_name),

constraint U3\_customer unique (email\_address)

)

-- End creating customer table

-- Creating customer address table

create table customer\_address(

-- Columns for customer address table

customer\_address\_ID int identity

,street\_address varchar(50) not null

, city varchar(30) not null

, state varchar(30) not null

, zip\_code varchar(10) not null

, customer\_ID int not null

-- Constraints on the customer address table

constraint PK\_customer\_address primary key (customer\_address\_ID),

constraint FK1\_customer\_address foreign key (customer\_ID) references customer(customer\_ID)

)

-- End creating customer address table

-- Creating registration status table

create table registration\_status(

-- Columns for registration status table

registration\_status\_ID int identity

, registration\_status bit

-- Constraints on registration status table

constraint PK\_registration\_status primary key (registration\_status\_ID),

constraint U1\_registration\_status unique (registration\_status)

)

-- End creating registration\_status table

-- Creating customer registration table

create table customer\_registration(

-- Columns for customer registration table

customer\_registration\_ID int identity

, customer\_ID int not null

, registration\_status\_ID int not null

-- Constraints on customer registration table

constraint PK\_customer\_registration primary key (customer\_registration\_ID),

constraint FK1\_customer\_registration foreign key (customer\_ID)

references customer(customer\_ID),

constraint FK2\_customer\_registration foreign key (registration\_status\_ID)

references registration\_status(registration\_status\_ID)

)

-- End creating customer registration table

-- Creating order source table

create table order\_source(

order\_source\_ID int identity

, source\_type varchar(30) not null

--constraints on order\_source table

constraint PK\_order\_source primary key (order\_source\_ID),

constraint U1\_order\_source unique (source\_type)

)

-- End creating order source table

-- Creating order status table

create table order\_status(

order\_status\_ID int identity

, order\_status\_type varchar(30) not null

--constraints on order status table

constraint PK\_order\_status primary key (order\_status\_ID),

constraint U1\_order\_status unique (order\_status\_type)

)

-- End creating order status table

-- Creating order destination table

create table order\_destination(

order\_destination\_ID int identity

, destination\_type varchar(30) not null

--constraints on order destination table

constraint PK\_order\_destination primary key (order\_destination\_ID),

constraint U1\_order\_destination unique (destination\_type)

)

-- End creating order destination table

-- Creating order ready table

create table order\_ready(

order\_ready\_ID int identity

, order\_ready\_date datetime

, order\_ready\_time time

--constraints on order ready table

constraint PK\_order\_ready primary key (order\_ready\_ID),

constraint U1\_order\_ready unique (order\_ready\_date)

)

-- End creating order ready table

-- Creating order placed table

create table order\_placed(

order\_placed\_ID int identity

, order\_placed\_date datetime

, order\_placed\_time time

--constraints on order placed table

constraint PK\_order\_placed primary key (order\_placed\_ID),

constraint U1\_order\_placed unique (order\_placed\_date)

)

-- End creating order placed table

-- Creating product table

create table product(

product\_ID int identity

, product\_name varchar(50) not null

, product\_type varchar(30) not null

, item\_cost money

--constraints on product table

constraint PK\_product primary key (product\_ID),

)

-- End creating product table

-- Creating product details table

create table product\_details(

product\_details\_ID int identity

, quantity int not null

, product\_cost money

, product\_ID int

--constraints on product details table

constraint PK\_product\_details primary key (product\_details\_ID),

constraint FK\_product\_details foreign key (product\_ID) references product(product\_ID)

)

-- End creating product details table

-- Creating payment type table

create table payment\_type(

payment\_type\_ID int identity

, payment\_type varchar(15) not null

, payment\_name varchar(15) not null

--constraints on payment type table

constraint PK\_payment\_type primary key (payment\_type\_ID),

)

-- End creating payment type table

-- Creating payment transaction table

create table payment\_transaction(

payment\_transaction\_ID int identity

, auth\_code varchar(15)

, auth\_description varchar(50)

, auth\_amount money

, auth\_date datetime

, auth\_time time

, payment\_type\_ID int

--constraints on payment transaction table

constraint PK\_payment\_transaction primary key (payment\_transaction\_ID),

constraint U1\_payment\_transaction unique (auth\_code),

constraint FK\_payment\_transaction foreign key (payment\_type\_ID) references payment\_type(payment\_type\_ID)

)

-- End creating payment transaction table

-- Creating payment method table

create table payment\_method(

payment\_method\_ID int identity

, order\_payment\_amount money

, payment\_transaction\_ID int

, payment\_type\_ID int

--constraints on payment method table

constraint PK\_payment\_method primary key (payment\_method\_ID),

constraint FK1\_payment\_method foreign key (payment\_transaction\_ID)

references payment\_transaction(payment\_transaction\_ID),

constraint FK2\_payment\_method foreign key (payment\_type\_ID)

references payment\_type(payment\_type\_ID)

)

-- End creating payment method table

-- Creating customer account table

create table customer\_account(

customer\_account\_ID int identity

, customer\_ID int

, customer\_address\_ID int

, payment\_type\_ID int

--constraints on customer account table

constraint PK\_customer\_account primary key (customer\_account\_ID),

constraint FK1\_customer\_account foreign key (customer\_ID)

references customer(customer\_ID),

constraint FK2\_customer\_account foreign key (customer\_address\_ID)

references customer\_address(customer\_address\_ID),

constraint FK3\_customer\_account foreign key (payment\_type\_ID)

references payment\_type(payment\_type\_ID)

)

-- End creating customer account table

-- Creating customer account to payment type table

create table customer\_account\_to\_payment\_type(

customer\_account\_to\_payment\_type\_ID int identity

, payment\_type\_ID int

, customer\_account\_ID int

--constraints on customer account to payment type table

constraint PK\_customer\_account\_to\_payment\_type primary key (customer\_account\_to\_payment\_type\_ID),

constraint FK1\_customer\_account\_to\_payment\_type foreign key (customer\_account\_ID)

references customer\_account(customer\_account\_ID),

constraint FK2\_customer\_account\_to\_payment\_type foreign key (payment\_type\_ID)

references payment\_type(payment\_type\_ID)

)

-- End creating customer account to payment type table

-- Creating order detail table

create table order\_detail(

order\_detail\_ID int identity

, order\_confirmation\_number varchar(10) not null

, order\_ready\_ID int

, customer\_ID int

, product\_details\_ID int

, order\_source\_ID int

, order\_destination\_ID int

, order\_placed\_ID int

, coupon\_ID int

, payment\_method\_ID int

--constraints on order detail table

constraint PK\_order\_detail primary key (order\_detail\_ID),

constraint U1\_order\_detail unique (order\_confirmation\_number),

constraint FK1\_order\_detail foreign key (order\_ready\_ID)

references order\_ready(order\_ready\_ID),

constraint FK2\_order\_detail foreign key (customer\_ID)

references customer(customer\_ID),

constraint FK3\_order\_detail foreign key (product\_details\_ID)

references product\_details(product\_details\_ID),

constraint FK4\_order\_detail foreign key (order\_source\_ID)

references order\_source(order\_source\_ID),

constraint FK5\_order\_detail foreign key (order\_destination\_ID)

references order\_destination(order\_destination\_ID),

constraint FK6\_order\_detail foreign key (order\_placed\_ID)

references order\_placed(order\_placed\_ID),

constraint FK7\_order\_detail foreign key (coupon\_ID)

references coupon(coupon\_ID),

constraint FK8\_order\_detail foreign key (payment\_method\_ID)

references payment\_method(payment\_method\_ID)

)

-- End creating order detail table

-- Creating coupon detail table

create table coupon\_detail(

coupon\_detail\_ID int identity

, coupon\_type\_ID int

, discounted\_item\_ID int

, coupon\_start\_date datetime

, coupon\_type\_details varchar(30) not null

, coupon\_expiry\_date datetime

, coupon\_application\_days varchar(15)

, coupon\_amount money

, coupon\_applied bit

, exclusive\_coupon bit

, discounted\_product\_quantity int

, coupon\_discount\_percent decimal(5,2)

, product\_ID int

--constraints on coupon detail table

constraint PK1\_coupon\_detail primary key (coupon\_detail\_ID),

constraint FK1\_coupon\_detail foreign key (product\_ID)

references product(product\_ID),

)

-- End creating coupon detail table

-- Creating coupon detail table

create table coupon\_detail(

coupon\_detail\_ID int identity

, coupon\_type\_ID int

, discounted\_item\_ID int

, coupon\_start\_date datetime

, coupon\_type\_details varchar(30) not null

, coupon\_expiry\_date datetime

, coupon\_application\_days varchar(15)

, coupon\_amount money

, coupon\_applied bit

, exclusive\_coupon bit

, discounted\_product\_quantity int

, coupon\_discount\_percent decimal(5,2)

, product\_ID int

--constraints on coupon detail table

constraint PK1\_coupon\_detail primary key (coupon\_detail\_ID),

constraint FK1\_coupon\_detail foreign key (product\_ID)

references product(product\_ID),

)

-- End creating coupon detail table

-- Creating coupon table

create table coupon(

coupon\_ID int identity

, coupon\_name varchar(30) not null

, coupon\_description varchar(50) not null

, coupon\_detail\_ID int

--constraints on coupon table

constraint PK\_coupon primary key (coupon\_ID),

constraint FK1\_coupon foreign key (coupon\_detail\_ID)

references coupon\_detail(coupon\_detail\_ID),

)

-- End creating coupon table

**More SQL DDL**

go

-- Stored Procedure to create new customer

create procedure new\_customer(

@firstname varchar(30),

@lastname varchar(30),

@phonenumber varchar(15),

@emailaddress varchar(50)

) as

begin

declare @customerID int

-- Get the customer ID from the customer table and store it in @customerID

select @customerID = customer\_ID from customer

where customer\_ID = @customerID

-- Insert new customer record in customer table

insert into customer(first\_name, last\_name, phone\_number,email\_address)

values (@firstname, @lastname, @phonenumber, @emailaddress)

-- increment customerID using @@identity

return @@identity

end

exec new\_customer 'Katrina','Adams','7876650091','kadams@syr.edu'

go

-- Stored Procedure to create customer address

alter procedure customeraddress(

@streetaddress varchar(50),

@city varchar(30),

@state varchar(30),

@zipcode varchar(10),

@customerID int

) as

begin

declare @customeraddressID int

/\*

Get the customeraddress ID from the customer adress table

by joining the customerID of customer

table with customer address table

and store it in @customerID

\*/

select @customeraddressID = customer\_address\_ID

from customer\_address

join customer

on customer.customer\_ID = customer\_address.customer\_ID

-- Insert new customer adress

insert into customer\_address(street\_address, city, state,zip\_code, customer\_ID)

values (@streetaddress, @city, @state, @zipcode, @customerID)

-- increment customerID using @@identity

return @@identity

end

go

-- Create view to see customer address and registration status

create view customer\_registration\_details as(

select customer.first\_name

, customer.last\_name

, customer\_address.street\_address

, customer\_address.city

, customer\_address.state

, customer\_address.zip\_code

, registration\_status.registration\_status

from customer

join customer\_address

on customer.customer\_ID = customer\_address.customer\_ID

join customer\_registration

on customer\_registration.customer\_ID = customer.customer\_ID

join registration\_status

on registration\_status.registration\_status\_ID = customer\_registration.registration\_status\_ID

)

go

-- Create View to see product sold most

create view product\_sold\_details as(

select product.product\_name

, product.product\_type

, sum(product\_details.product\_cost) as product\_cost

from product

inner join product\_details

on product.product\_ID = product\_details.product\_ID

group by product.product\_name, product.product\_type

)

select \* from dbo.product\_sold\_details order by product\_cost desc

select \* from dbo.customer\_registration\_details order by registration\_status desc

go

-- Create function to multiply quantity with product price

create function product\_price(@quantity int, @itemcost money)

returns money as

begin

declare @returnvalue money

set @returnvalue = @quantity \* @itemcost

return @returnvalue

end

go

-- Stored Procedure to create new payment method

create procedure new\_payment\_method(@paymenttransactionID int) as

begin

select @paymenttransactionID = payment\_transaction\_ID from payment\_transaction

where payment\_transaction\_ID = @paymenttransactionID

-- Insert new data in payment method table

insert into payment\_method(order\_payment\_amount, payment\_transaction\_ID, payment\_type\_ID)

values ((select auth\_amount from payment\_transaction

where payment\_transaction\_ID = @paymenttransactionID),

(select payment\_transaction\_ID from payment\_transaction

where payment\_transaction\_ID = @paymenttransactionID),

(select payment\_type\_ID from payment\_transaction

where payment\_transaction\_ID = @paymenttransactionID)

)

end

**SQL DML INSERT Statements**

--Insert status in registration status table

insert into registration\_status(registration\_status)

values (1)

-- Insert record in customer registation table

insert into customer\_registration (customer\_ID, registration\_status\_ID)

values(1, 2)

insert into customer\_registration (customer\_ID, registration\_status\_ID)

values(2, 2)

insert into customer\_registration (customer\_ID, registration\_status\_ID)

values(3, 2)

insert into customer\_registration (customer\_ID, registration\_status\_ID)

values(4, 1)

insert into customer\_registration (customer\_ID, registration\_status\_ID)

values(5, 2)

insert into customer\_registration (customer\_ID, registration\_status\_ID)

values(6, 1)

insert into customer\_registration (customer\_ID, registration\_status\_ID)

values(7, 2)

insert into customer\_registration (customer\_ID, registration\_status\_ID)

values(8, 2)

insert into customer\_registration (customer\_ID, registration\_status\_ID)

values(9, 1)

insert into customer\_registration (customer\_ID, registration\_status\_ID)

values(10, 2)

-- Insert record in order status table

insert into order\_status (order\_status\_type)

values ('In Progress')

insert into order\_status (order\_status\_type)

values ('In Oven')

insert into order\_status (order\_status\_type)

values ('Ready')insert into order\_status (order\_status\_type)

values ('Completed')

-- Insert order source table

insert into order\_source (source\_type)

values ('Online')

-- Insert records in order destination table

insert into order\_destination(destination\_type)

values('Delivery')

insert into order\_destination(destination\_type)

values('Carryout')

-- Insert data in product table

insert into product (product\_name, product\_type, item\_cost)

values ('Medium Pizza', 'Pizza', $15.43)

insert into product (product\_name, product\_type, item\_cost)

values ('Small Pizza', 'Pizza', $12.43)

insert into product (product\_name, product\_type, item\_cost)

values ('Large Pizza', 'Pizza', $17.43)

insert into product (product\_name, product\_type, item\_cost)

values ('Extra Large Pizza', 'Pizza', $20.43)

insert into product (product\_name, product\_type, item\_cost)

values ('6 pcs Chicken Nuggets', 'Chicken', $6.99)

insert into product (product\_name, product\_type, item\_cost)

values ('10 pcs Chicken Nuggets', 'Chicken', $8.99)

insert into product (product\_name, product\_type, item\_cost)

values ('12 pcs Chicken Nuggets', 'Chicken', $10.99)

insert into product (product\_name, product\_type, item\_cost)

values ('Tomato', 'Sides', $0.99)

insert into product (product\_name, product\_type, item\_cost)

values ('Chicken', 'Sides', $0.99)

insert into product (product\_name, product\_type, item\_cost)

values ('Jalapeno', 'Sides', $0.99)

insert into product (product\_name, product\_type, item\_cost)

values ('Black Olives', 'Sides', $0.99)

insert into product (product\_name, product\_type, item\_cost)

values ('Green Olives', 'Sides', $0.99)

insert into product (product\_name, product\_type, item\_cost)

values ('Cheese', 'Sides', $0.99)

insert into product (product\_name, product\_type, item\_cost)

values ('Pepperoni', 'Sides', $0.99)

insert into product (product\_name, product\_type, item\_cost)

values ('Italian sausage', 'Sides', $0.99)

insert into product (product\_name, product\_type, item\_cost)

values ('Bacon', 'Sides', $0.99)

insert into product (product\_name, product\_type, item\_cost)

values ('Pineapple', 'Sides', $0.99)

insert into product (product\_name, product\_type, item\_cost)

values ('Peppers', 'Sides', $0.99)

insert into product (product\_name, product\_type, item\_cost)

values ('Beef', 'Sides', $0.99)

insert into product (product\_name, product\_type, item\_cost)

values ('Mushrooms', 'Sides', $0.99)

insert into product (product\_name, product\_type, item\_cost)

values ('Onions', 'Sides', $0.99)

insert into product (product\_name, product\_type, item\_cost)

values ('Green bell peppers', 'Sides', $0.99)

-- Inserting data in product details table

insert into product\_details (quantity, product\_cost, product\_ID)

values (3, dbo.product\_price(3,12.43) , 2)

insert into product\_details (quantity, product\_cost, product\_ID)

values (1, dbo.product\_price(1,17.43), 3)

insert into product\_details (quantity, product\_cost, product\_ID)

values (4, dbo.product\_price(4,17.43), 3)

insert into product\_details (quantity, product\_cost, product\_ID)

values (1, dbo.product\_price(1,17.43), 3)

insert into product\_details (quantity, product\_cost, product\_ID)

values (2, dbo.product\_price(2,17.43), 3)

insert into product\_details (quantity, product\_cost, product\_ID)

values (1, dbo.product\_price(1,10.99), 7)

insert into product\_details (quantity, product\_cost, product\_ID)

values (2, dbo.product\_price(2,6.99), 5)

insert into product\_details (quantity, product\_cost, product\_ID)

values (1, dbo.product\_price(1,17.43), 3)

insert into product\_details (quantity, product\_cost, product\_ID)

values (2, dbo.product\_price(2,20.43), 4)

insert into product\_details (quantity, product\_cost, product\_ID)

values (1, dbo.product\_price(1,12.43), 2)

insert into product\_details (quantity, product\_cost, product\_ID)

values (2, dbo.product\_price(2,15.43), 1)

insert into product\_details (quantity, product\_cost, product\_ID)

values (3, dbo.product\_price(3,17.43), 3)

insert into product\_details (quantity, product\_cost, product\_ID)

values (1, dbo.product\_price(1,17.43), 3)

insert into product\_details (quantity, product\_cost, product\_ID)

values (1, dbo.product\_price(1,12.43), 2)

insert into product\_details (quantity, product\_cost, product\_ID)

values (3, dbo.product\_price(3,10.99), 7)

insert into product\_details (quantity, product\_cost, product\_ID)

values (5, dbo.product\_price(5,12.43), 2)

insert into product\_details (quantity, product\_cost, product\_ID)

values (1, dbo.product\_price(1,17.43), 3)

insert into product\_details (quantity, product\_cost, product\_ID)

values (1, dbo.product\_price(1,17.43), 3)

insert into product\_details (quantity, product\_cost, product\_ID)

values (1, dbo.product\_price(1,17.43), 3)

-- Inserting record in payment type table

insert into payment\_type(payment\_type, payment\_name)

values ('Cash', 'Cash')

insert into payment\_type(payment\_type, payment\_name)

values ('Card', 'MasterCard')

insert into payment\_type(payment\_type, payment\_name)

values ('Card', 'Visa')

insert into payment\_type(payment\_type, payment\_name)

values ('Card', 'Discover')

insert into payment\_type(payment\_type, payment\_name)

values ('Card', 'American Express')

-- Inserting record in payment transaction table

insert into payment\_transaction (auth\_code, auth\_description, auth\_amount, auth\_date,auth\_time, payment\_type\_ID)

values ('01ABD', 'credit card transaction', 69.72, 09-20-2018, CURRENT\_TIMESTAMP, 2)-- 5

insert into payment\_transaction (auth\_code, auth\_description, auth\_amount, auth\_date,auth\_time, payment\_type\_ID)

values ('01AB2', 'credit card transaction', 37.29, 08-19-2018, CURRENT\_TIMESTAMP, 3)-- 2

insert into payment\_transaction (auth\_code, auth\_description, auth\_amount, auth\_date,auth\_time, payment\_type\_ID)

values (null, null, 17.43, 01-01-2018, CURRENT\_TIMESTAMP, 1)--1

insert into payment\_transaction (auth\_code, auth\_description, auth\_amount, auth\_date,auth\_time, payment\_type\_ID)

values ('01BGF', 'credit card transaction', (select product\_cost from product\_details

where product\_details\_ID = 4), 04-20-2017, CURRENT\_TIMESTAMP, (select payment\_type\_ID from payment\_type

where payment\_type\_ID = 4))--4

insert into payment\_transaction (auth\_code, auth\_description, auth\_amount, auth\_date,auth\_time, payment\_type\_ID)

values ('01B3F', 'credit card transaction', (select product\_cost from product\_details

where product\_details\_ID = 6), 01-20-2018, CURRENT\_TIMESTAMP, (select payment\_type\_ID from payment\_type

where payment\_type\_ID =2))

insert into payment\_transaction (auth\_code, auth\_description, auth\_amount, auth\_date,auth\_time, payment\_type\_ID)

values ('5BYI', 'credit card transaction', (select product\_cost from product\_details

where product\_details\_ID = 7), 09-20-2018, CURRENT\_TIMESTAMP, (select payment\_type\_ID from payment\_type

where payment\_type\_ID = 3)) -- RUN THIS CODE

insert into payment\_transaction (auth\_code, auth\_description, auth\_amount, auth\_date,auth\_time, payment\_type\_ID)

values ('01BGF1', 'credit card transaction', (select product\_cost from product\_details

where product\_details\_ID = 8), 12-20-2017, CURRENT\_TIMESTAMP, (select payment\_type\_ID from payment\_type

where payment\_type\_ID = 3))

insert into payment\_transaction (auth\_code, auth\_description, auth\_amount, auth\_date,auth\_time, payment\_type\_ID)

values ('01BGJ', 'credit card transaction', (select product\_cost from product\_details

where product\_details\_ID = 9), 03-20-2018, CURRENT\_TIMESTAMP, (select payment\_type\_ID from payment\_type

where payment\_type\_ID = 3))

insert into payment\_transaction (auth\_code, auth\_description, auth\_amount, auth\_date,auth\_time, payment\_type\_ID)

values ('01BGFS', 'credit card transaction', (select product\_cost from product\_details

where product\_details\_ID = 10), 02-20-2017, CURRENT\_TIMESTAMP, (select payment\_type\_ID from payment\_type

where payment\_type\_ID = 4))

insert into payment\_transaction (auth\_code, auth\_description, auth\_amount, auth\_date,auth\_time, payment\_type\_ID)

values ('98BYT', 'credit card transaction', (select product\_cost from product\_details

where product\_details\_ID = 11), 06-20-2018, CURRENT\_TIMESTAMP, (select payment\_type\_ID from payment\_type

where payment\_type\_ID = 2))

insert into payment\_transaction (auth\_code, auth\_description, auth\_amount, auth\_date,auth\_time, payment\_type\_ID)

values ('4BER', 'credit card transaction', (select product\_cost from product\_details

where product\_details\_ID = 12), 04-20-2017, CURRENT\_TIMESTAMP, (select payment\_type\_ID from payment\_type

where payment\_type\_ID = 3)) -- RUN THIS

insert into payment\_transaction (auth\_code, auth\_description, auth\_amount, auth\_date,auth\_time, payment\_type\_ID)

values ('01DKU', 'credit card transaction', (select product\_cost from product\_details

where product\_details\_ID = 13), 04-23-2018, CURRENT\_TIMESTAMP, (select payment\_type\_ID from payment\_type

where payment\_type\_ID = 3))

insert into payment\_transaction (auth\_code, auth\_description, auth\_amount, auth\_date,auth\_time, payment\_type\_ID)

values ('04YTB', 'credit card transaction', (select product\_cost from product\_details

where product\_details\_ID = 14), 04-05-2018, CURRENT\_TIMESTAMP, (select payment\_type\_ID from payment\_type

where payment\_type\_ID = 3))

insert into payment\_transaction (auth\_code, auth\_description, auth\_amount, auth\_date,auth\_time, payment\_type\_ID)

values ('01VGF', 'credit card transaction', (select product\_cost from product\_details

where product\_details\_ID = 15), 01-20-2017, CURRENT\_TIMESTAMP, (select payment\_type\_ID from payment\_type

where payment\_type\_ID = 4))

insert into payment\_transaction (auth\_code, auth\_description, auth\_amount, auth\_date,auth\_time, payment\_type\_ID)

values ('4GTF3', 'credit card transaction', (select product\_cost from product\_details

where product\_details\_ID = 16), 04-20-2017, CURRENT\_TIMESTAMP, (select payment\_type\_ID from payment\_type

where payment\_type\_ID = 2))--RUN THIS

insert into payment\_transaction (auth\_code, auth\_description, auth\_amount, auth\_date,auth\_time, payment\_type\_ID)

values ('5BWER', 'credit card transaction', (select product\_cost from product\_details

where product\_details\_ID = 17), 04-20-2017, CURRENT\_TIMESTAMP, (select payment\_type\_ID from payment\_type

where payment\_type\_ID = 4)) -- RUN THIS

insert into payment\_transaction (auth\_code, auth\_description, auth\_amount, auth\_date,auth\_time, payment\_type\_ID)

values ('9LKI', 'credit card transaction', (select product\_cost from product\_details

where product\_details\_ID = 18), 04-20-2017, CURRENT\_TIMESTAMP, (select payment\_type\_ID from payment\_type

where payment\_type\_ID = 2)) -- RUN THIS

insert into payment\_transaction (auth\_code, auth\_description, auth\_amount, auth\_date,auth\_time, payment\_type\_ID)

values ('1NYE', 'credit card transaction', (select product\_cost from product\_details

where product\_details\_ID = 19), 01-01-2018, CURRENT\_TIMESTAMP, (select payment\_type\_ID from payment\_type

where payment\_type\_ID = 2))

insert into payment\_transaction (auth\_code, auth\_description, auth\_amount, auth\_date,auth\_time, payment\_type\_ID)

values ('1TGE', 'credit card transaction', (select product\_cost from product\_details

where product\_details\_ID = 20), 02-20-2018, CURRENT\_TIMESTAMP, (select payment\_type\_ID from payment\_type

where payment\_type\_ID = 3))

insert into payment\_transaction (auth\_code, auth\_description, auth\_amount, auth\_date,auth\_time, payment\_type\_ID)

values ('5NYR', 'credit card transaction', (select product\_cost from product\_details

where product\_details\_ID = 21), 06-28-2018, CURRENT\_TIMESTAMP, (select payment\_type\_ID from payment\_type

where payment\_type\_ID = 2))

insert into payment\_transaction (auth\_code, auth\_description, auth\_amount, auth\_date,auth\_time, payment\_type\_ID)

values ('4BYH', 'credit card transaction', (select product\_cost from product\_details

where product\_details\_ID = 22), 01-12-2018, CURRENT\_TIMESTAMP, (select payment\_type\_ID from payment\_type

where payment\_type\_ID = 2))

insert into payment\_transaction (auth\_code, auth\_description, auth\_amount, auth\_date,auth\_time, payment\_type\_ID)

values ('6NYC', 'credit card transaction', (select product\_cost from product\_details

where product\_details\_ID = 23), 09-16-2018, CURRENT\_TIMESTAMP, (select payment\_type\_ID from payment\_type

where payment\_type\_ID = 4))

insert into payment\_transaction (auth\_code, auth\_description, auth\_amount, auth\_date,auth\_time, payment\_type\_ID)

values ('7UNO', 'credit card transaction', (select product\_cost from product\_details

where product\_details\_ID = 24), 07-22-2018, CURRENT\_TIMESTAMP, (select payment\_type\_ID from payment\_type

where payment\_type\_ID = 2)) -- RUN THIS

insert into payment\_transaction (auth\_code, auth\_description, auth\_amount, auth\_date,auth\_time, payment\_type\_ID)

values ('8GRT', 'credit card transaction', (select product\_cost from product\_details

where product\_details\_ID = 25), 03-02-2018, CURRENT\_TIMESTAMP, (select payment\_type\_ID from payment\_type

where payment\_type\_ID = 4))

insert into payment\_transaction (auth\_code, auth\_description, auth\_amount, auth\_date,auth\_time, payment\_type\_ID)

values ('5BUM', 'credit card transaction', (select product\_cost from product\_details

where product\_details\_ID = 26), 07-08-2017, CURRENT\_TIMESTAMP, (select payment\_type\_ID from payment\_type

where payment\_type\_ID = 2)) -- RUN THIS

insert into payment\_transaction (auth\_code, auth\_description, auth\_amount, auth\_date,auth\_time, payment\_type\_ID)

values ('9NTH', 'credit card transaction', (select product\_cost from product\_details

where product\_details\_ID = 27), 01-12-2018, CURRENT\_TIMESTAMP, (select payment\_type\_ID from payment\_type

where payment\_type\_ID = 2))

insert into payment\_transaction (auth\_code, auth\_description, auth\_amount, auth\_date,auth\_time, payment\_type\_ID)

values ('0LOK', 'credit card transaction', (select product\_cost from product\_details

where product\_details\_ID = 29), 01-01-2018, CURRENT\_TIMESTAMP, (select payment\_type\_ID from payment\_type

where payment\_type\_ID = 3)) -- RUN THIS

insert into payment\_transaction (auth\_code, auth\_description, auth\_amount, auth\_date,auth\_time, payment\_type\_ID)

values ('8MNY', 'credit card transaction', (select product\_cost from product\_details

where product\_details\_ID = 30), 02-16-2018, CURRENT\_TIMESTAMP, (select payment\_type\_ID from payment\_type

where payment\_type\_ID = 3)) -- RUN THIS

insert into payment\_transaction (auth\_code, auth\_description, auth\_amount, auth\_date,auth\_time, payment\_type\_ID)

values ('5BTG', 'credit card transaction', (select product\_cost from product\_details

where product\_details\_ID = 31), 04-16-2017, CURRENT\_TIMESTAMP, (select payment\_type\_ID from payment\_type

where payment\_type\_ID = 2)) -- RUN THIS

insert into payment\_transaction (auth\_code, auth\_description, auth\_amount, auth\_date,auth\_time, payment\_type\_ID)

values ('8NTM', 'credit card transaction', (select product\_cost from product\_details

where product\_details\_ID = 32), 05-02-2018, CURRENT\_TIMESTAMP, (select payment\_type\_ID from payment\_type

where payment\_type\_ID = 2)) -- RUN THIS

insert into payment\_transaction (auth\_code, auth\_description, auth\_amount, auth\_date,auth\_time, payment\_type\_ID)

values ('7BTE', 'credit card transaction', (select product\_cost from product\_details

where product\_details\_ID = 33), 05-19-2018, CURRENT\_TIMESTAMP, (select payment\_type\_ID from payment\_type

where payment\_type\_ID = 2))

insert into payment\_transaction (auth\_code, auth\_description, auth\_amount, auth\_date,auth\_time, payment\_type\_ID)

values ('0L7M', 'credit card transaction', (select product\_cost from product\_details

where product\_details\_ID = 34), 05-21-2017, CURRENT\_TIMESTAMP, (select payment\_type\_ID from payment\_type

where payment\_type\_ID = 4)) -- RUN THIS

insert into payment\_transaction (auth\_code, auth\_description, auth\_amount, auth\_date,auth\_time, payment\_type\_ID)

values ('9MUN', 'credit card transaction', (select product\_cost from product\_details

where product\_details\_ID = 35), 05-02-2017, CURRENT\_TIMESTAMP, (select payment\_type\_ID from payment\_type

where payment\_type\_ID = 3))

insert into payment\_transaction (auth\_code, auth\_description, auth\_amount, auth\_date,auth\_time, payment\_type\_ID)

values ('4BYUR', 'credit card transaction', (select product\_cost from product\_details

where product\_details\_ID = 36), 03-03-2017, CURRENT\_TIMESTAMP, (select payment\_type\_ID from payment\_type

where payment\_type\_ID = 2)) -- RUN THIS

insert into payment\_transaction (auth\_code, auth\_description, auth\_amount, auth\_date,auth\_time, payment\_type\_ID)

values ('4BUUI', 'credit card transaction', (select product\_cost from product\_details

where product\_details\_ID = 37), 04-20-2018, CURRENT\_TIMESTAMP, (select payment\_type\_ID from payment\_type

where payment\_type\_ID = 2)) -- RUN THIS

insert into payment\_transaction (auth\_code, auth\_description, auth\_amount, auth\_date,auth\_time, payment\_type\_ID)

values ('7MYT', 'credit card transaction', (select product\_cost from product\_details

where product\_details\_ID = 38), 02-07-2018, CURRENT\_TIMESTAMP, (select payment\_type\_ID from payment\_type

where payment\_type\_ID = 2))

insert into payment\_transaction (auth\_code, auth\_description, auth\_amount, auth\_date,auth\_time, payment\_type\_ID)

values ('7M6G', 'credit card transaction', (select product\_cost from product\_details

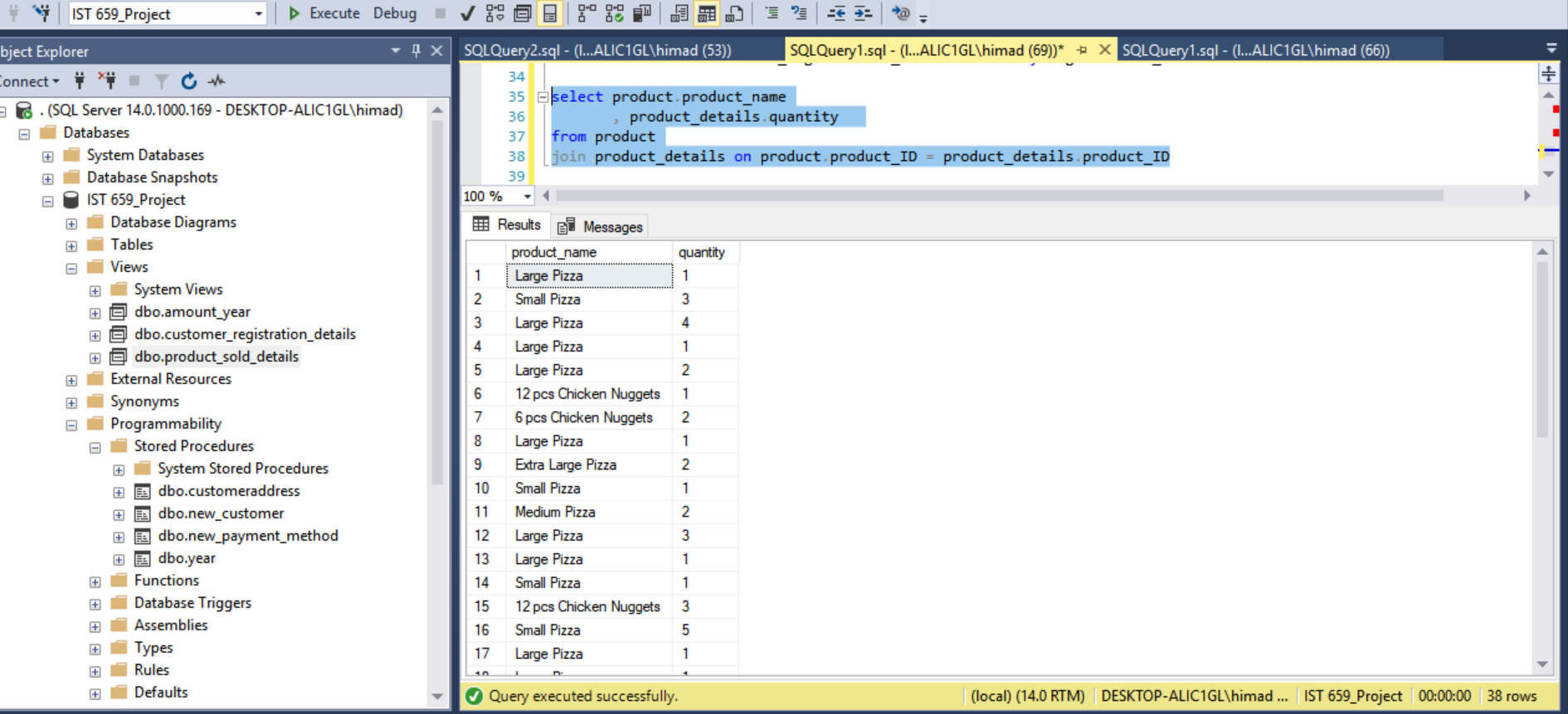
where product\_details\_ID = 38), 06-17-2018, CURRENT\_TIMESTAMP, (select payment\_type\_ID from payment\_type

where payment\_type\_ID = 2))

**SQL DML SELECT Statements**

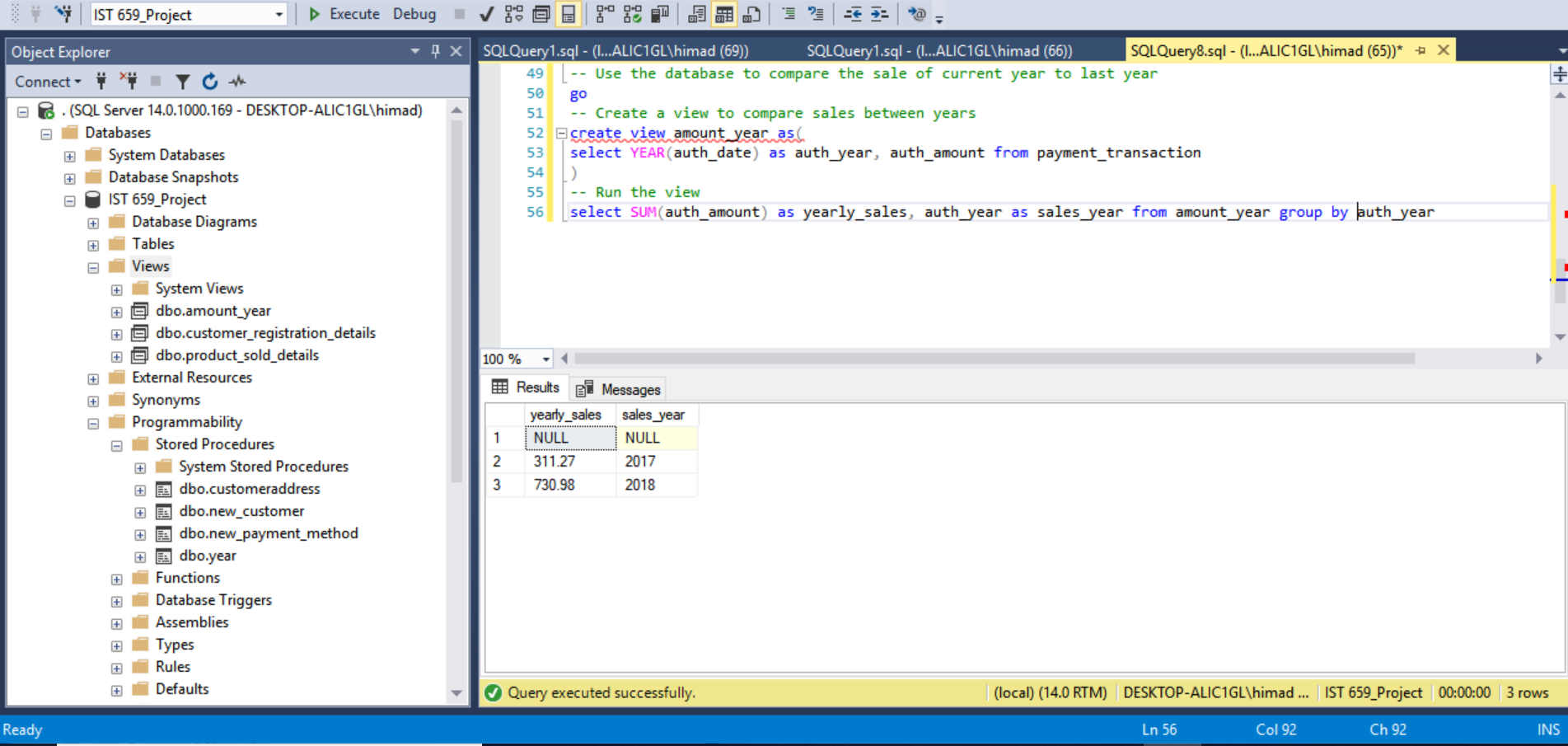
Question 1:

Use the database to find out the most popular product



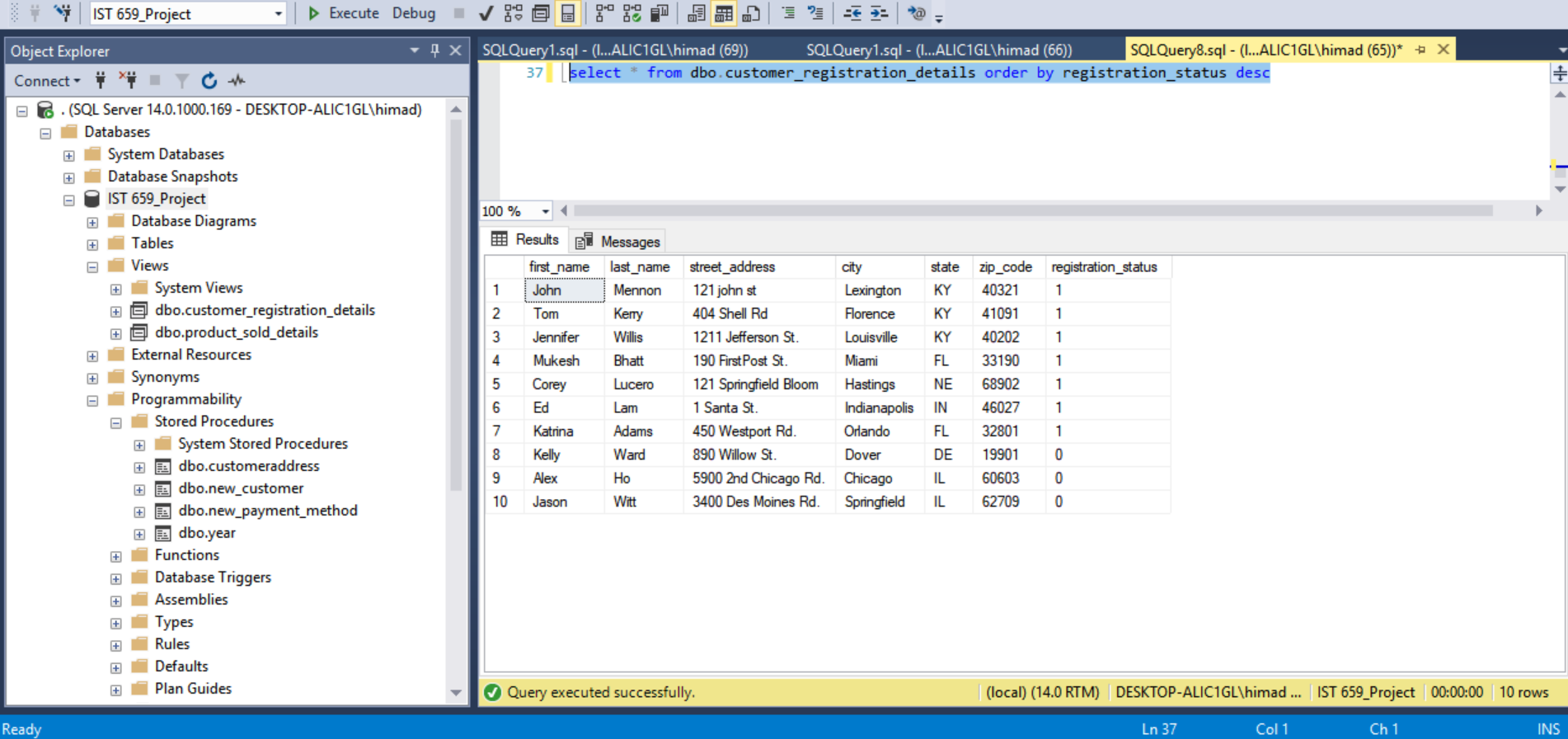
Question 2:

Use the database to compare the sale of current year to last year

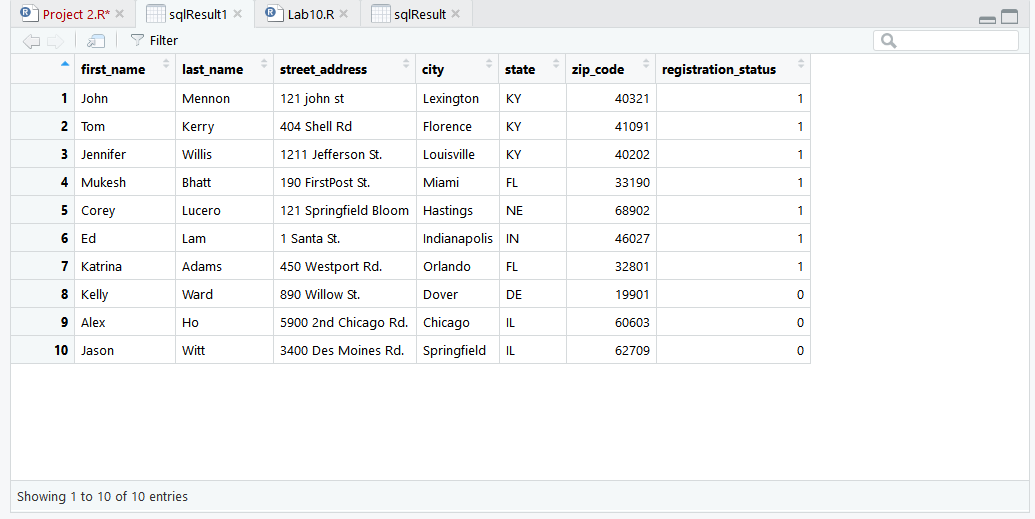


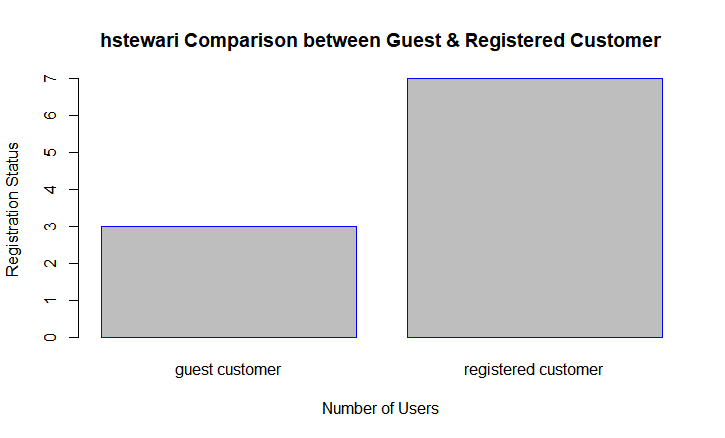
Question 3:

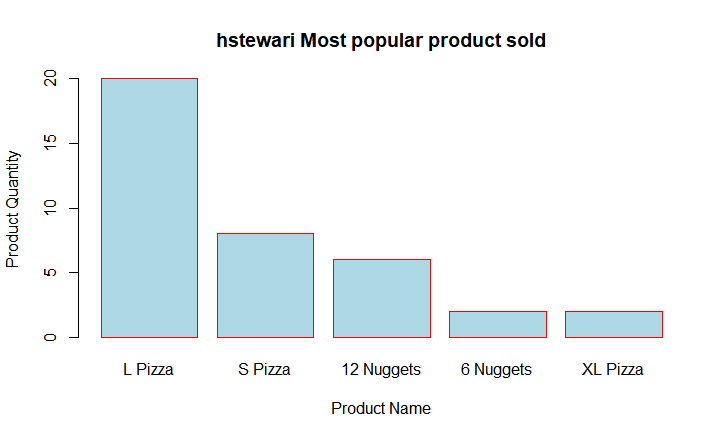
Comparison between guest and registered customer



**GUI Prototype**



****

****

**# Ready RODBC for use in this script**

**require(RODBC)**

**#create a connection to SQL Server using 64-bit DSN**

**myconn <- odbcConnect("IST 659\_Project")**

**#SQL statement to see customer address and registration status**

**sqlSelectStatement <-**

**"select \***

**from customer\_registration\_details**

**order by registration\_status desc**

**"**

**# Send the request to the server and store the result in a variable**

**sqlResult <- sqlQuery(myconn, sqlSelectStatement)**

**# SQL query to compare between guest and registered customer**

**sqlSelectStatement1 <-**

**"**

**select \***

**from customer\_registration\_details**

**order by registration\_status desc**

**"**

**# Send the request to the server and store the result in a variable**

**sqlResult1 <- sqlQuery(myconn, sqlSelectStatement1)**

**#Plot a bar chart of comparison between guest and registered customer**

**registrationCount <- table(sqlResult1$registration\_status)**

**barplot(registrationCount,**

**main = "hstewari Comparison between Guest & Registered Customer",**

**ylab = "Registration Status",**

**xlab = "Number of Users",**

**border = "blue",**

**names.arg = registrationCount**

**)**

**# SQL query to to find out the most popular product sold**

**sqlSelectStatement2 <-**

**"**

**select**

**product.product\_name**

**, product\_details.quantity**

**from product**

**join product\_details on product.product\_ID = product\_details.product\_ID**

**"**

**# Send the request to the server and store the result in a variable**

**sqlResult2 <- sqlQuery(myconn, sqlSelectStatement2)**

**# Create a list of products**

**products <- c("L Pizza", "S Pizza", "12 Nuggets", "6 Nuggets",**

**"XL Pizza")**

**#Plot a bar chart to find out the most popular product sold**

**productQty <- table(sqlResult2$quantity)**

**barplot(productQty,**

**main = "hstewari Most popular product sold",**

**ylab = "Product Quantity",**

**xlab = "Product Name",**

**border = "red",**

**col = "light blue",**

**names.arg = products**

**)**

**# Close all connections**

**odbcCloseAll()**

**#Fin**