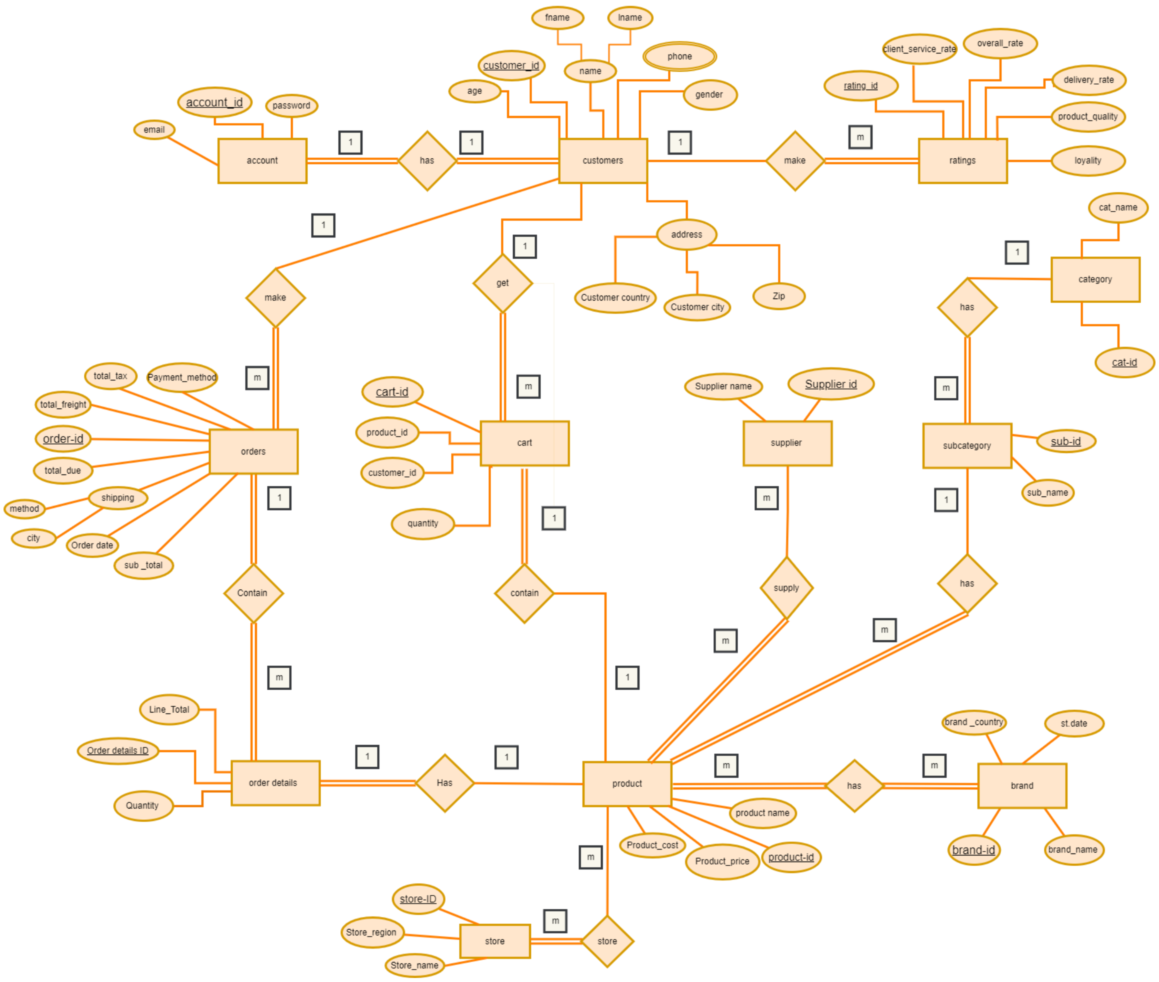
. Documentation

E-Commerce

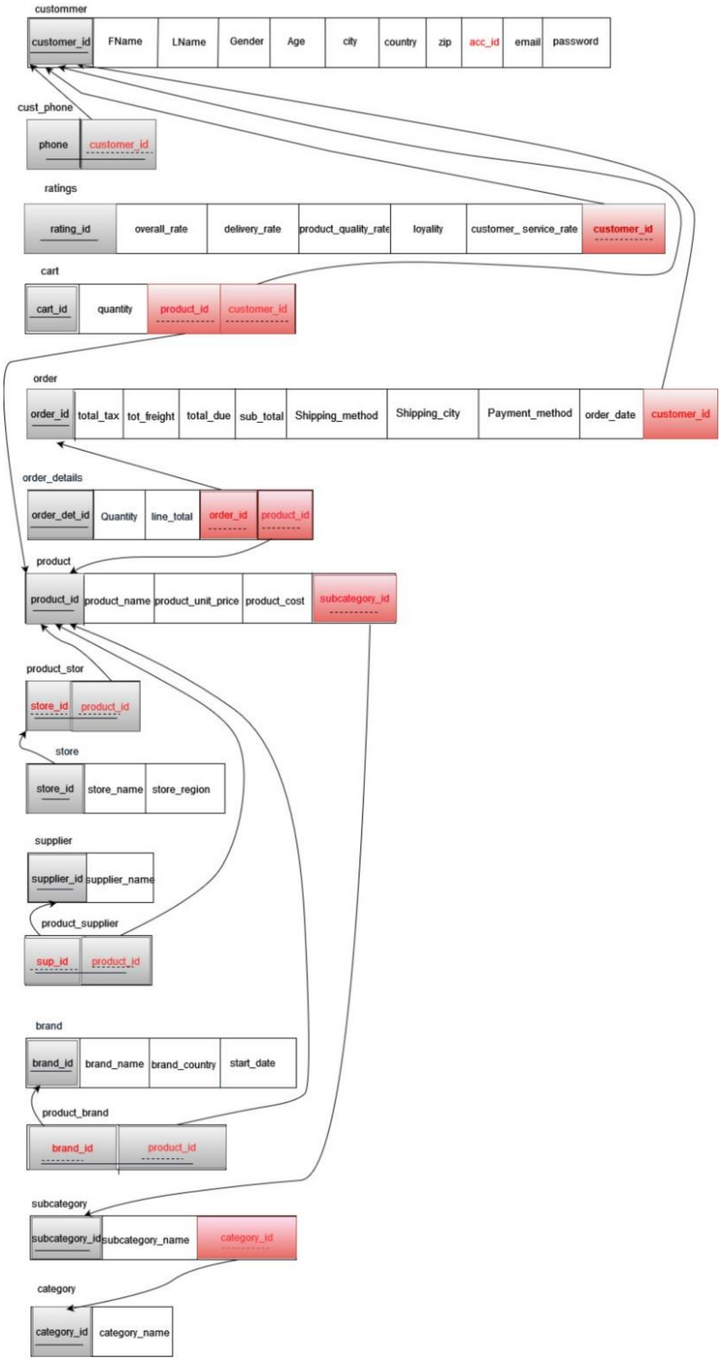
Table of Contents

. ERD

E -Commerce ERD



. Mapping



custommer

customer_id	FName	LName	Gender	Age	city	country	zip	acc_id	email	password

cust_phone

phone	customer_id

ratings

rating_id	overall_rate	delivery_rate	product_quality_rate	loyality	customer_ service_rate	customer_id

cart

cart_id	quantity	product_id	customer_id

order

order_id	total_tax	tot_freight	total_due	sub_total	Shipping_method	Shipping_city	Payment_method	order_date	customer_id

order_details

order_det_id	Quantity	line_total	order_id	product_id

product

product_id	product_name	product_unit_price	product_cost	subcategory_id

product_stor

store_id	product_id

store

store_id	store_name	store_region

supplier

supplier_id	supplier_name

product_supplier

sup_id	product_id

brand

brand_id	brand_name	brand_country	start_date

product_brand

brand_id	product_id

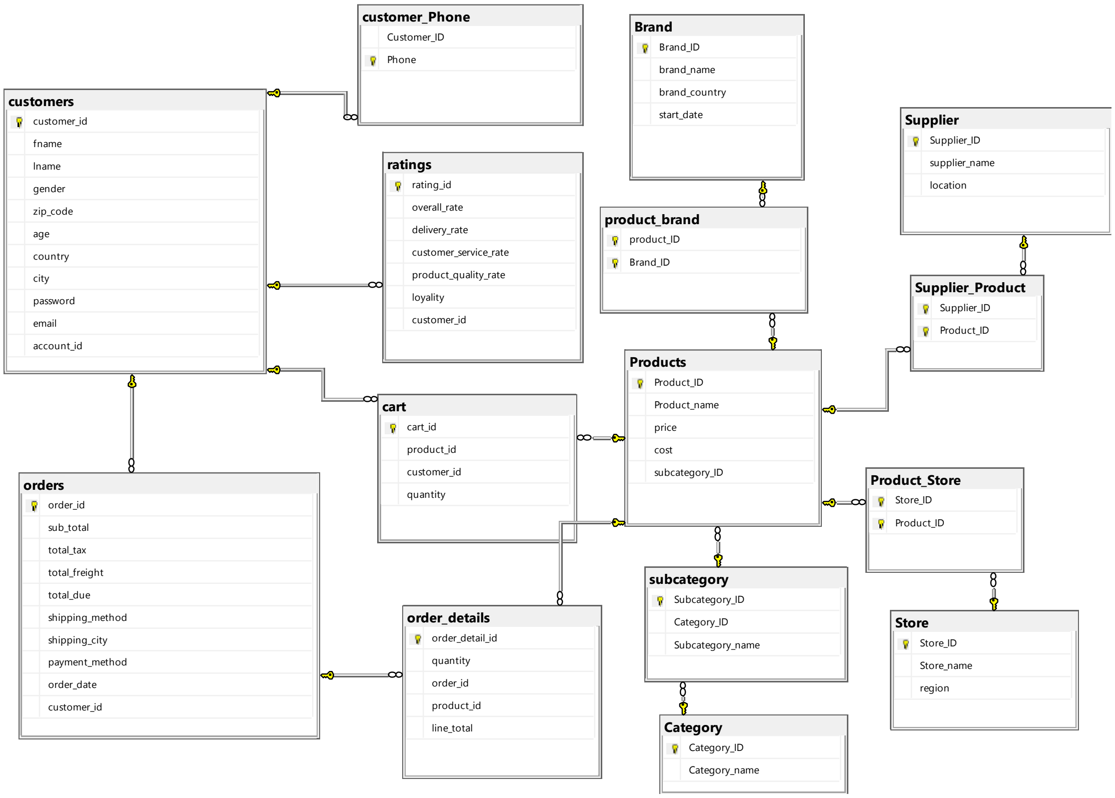
subcategory

subcategory_id	subcategory_name	category_id

category

category_id	category_name

🗀 Diagram


customer_Phone
Customer_ID
Phone
Brand
Brand_ID
brand_name
brand_country
start_date
customers
customer_id
fname
lname
gender
zip_code
age
country
city
password
email
account_id
ratings
rating_id
overall_rate
delivery_rate
customer_service_rate
product_quality_rate
loyality
customer_id
Supplier
Supplier_ID
supplier_name
location
product_brand
product_ID
Brand_ID
Supplier_Product
Supplier_ID
Product_ID
Products
Product_ID
Product_name
price
cost
subcategory_ID
cart
cart_id
product_id
customer_id
quantity
orders
order_id
sub_total
total_tax
total_freight
total_due
shipping_method
shipping_city
payment_method
order_date
customer_id
Product_Store
Store_ID
Product_ID
subcategory
Subcategory_ID
Category_ID
Subcategory_name
order_details
order_detail_id
quantity
order_id
product_id
line_total
Store
Store_ID
Store_name
region
Category
Category_ID
Category_name

Tables

Objects

Name
dbo.Brand *Contain Product brand information*
dbo.cart *Cart Shopping*
dbo.Category *Category information*
dbo.customer_Phone *Customer Phone Numbers*
dbo.customers *Customers information*
dbo.order_details *Order details information*
dbo.orders *Orders information*
dbo.product_brand
dbo.Product_Store
dbo.Products *Products information*
dbo.ratings *Customer Rating of our services*
dbo.Store *Store information*
dbo.subcategory *Subcategory information*
dbo.Supplier *Supplier information*
dbo.Supplier_Product

[dbo].[Brand]

MS_Description

Contain Product brand information

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability
PK C	Brand_ID *Primary key for Brand records.*	int	4	NOT NULL
	brand_name *Brand Name*	varchar(50)	50	NULL allowed
	brand_country *Country of brand Origin*	varchar(50)	50	NULL allowed
	start_date *The start date of launching the brand on the market*	date	3	NULL allowed

Indexes

Key	Name	Key Columns	Unique
PK C	PK_Brand	Brand_ID	True

SQL Script

```
CREATE TABLE [dbo].[Brand]
(
[Brand_ID] [int] NOT NULL,
[brand_name] [varchar] (50) COLLATE Arabic_CI_AS NULL,
[brand_country] [varchar] (50) COLLATE Arabic_CI_AS NULL,
[start_date] [date] NULL
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[Brand] ADD CONSTRAINT [PK_Brand] PRIMARY KEY CLUSTERED ([Brand_ID]) ON
[PRIMARY]
GO
EXEC sp_addextendedproperty N'MS_Description', N'Contain Product brand information',
'SCHEMA', N'dbo', 'TABLE', N'Brand', NULL, NULL
GO
EXEC sp_addextendedproperty N'MS_Description', N'Country of brand Origin', 'SCHEMA',
N'dbo', 'TABLE', N'Brand', 'COLUMN', N'brand_country'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key for Brand records.',
'SCHEMA', N'dbo', 'TABLE', N'Brand', 'COLUMN', N'Brand_ID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Brand Name', 'SCHEMA', N'dbo', 'TABLE',
N'Brand', 'COLUMN', N'brand_name'
```

```
GO
EXEC sp_addextendedproperty N'MS_Description', N'The start date of launching the brand on
the market ', 'SCHEMA', N'dbo', 'TABLE', N'Brand', 'COLUMN', N'start_date'
GO
```

Used By

[dbo].[product_brand]

Linked From

Table	Join	Title / Name / Description
Product_brand	**Brand.Brand_ID**= Product_brand.Brand_ID	FK_Product_brand_Brand_ID Foreign key constraint referencing **Brand.Brand_ID** .

[dbo].[cart]

MS_Description

Cart Shopping

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability	Identity
FK C	cart_id *Primary key for Cart records.* *Identity / Auto increment column*	int	4	NOT NULL	1 - 1
FK	product_id *Unique product identification number.* *Foreign key to Products.product_id .*	int	4	NULL allowed	
FK	customer_id *Unique customer identification number.* *Foreign key referring to Customers.customer_id .*	int	4	NULL allowed	
	quantity *Quantity of products in cart*	int	4	NULL allowed	

Indexes

Key	Name	Key Columns	Unique
FK C	PK__cart__2EF52A27CC68CCC3	cart_id	True

Foreign Keys

Name	Columns
FK_cart_customers	customer_id->[dbo].[customers].[customer_id]
FK_cart_Products	product_id->[dbo].[Products].[Product_ID]

SQL Script

```
CREATE TABLE [dbo].[cart]
(
[cart_id] [int] NOT NULL IDENTITY(1, 1),
[product_id] [int] NULL,
[customer_id] [int] NULL,
[quantity] [int] NULL
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[cart] ADD CONSTRAINT [PK__cart__2EF52A27CC68CCC3] PRIMARY KEY
```

```
CLUSTERED ([cart_id]) ON [PRIMARY]
GO
ALTER TABLE [dbo].[cart] ADD CONSTRAINT [FK_cart_customers] FOREIGN KEY ([customer_id])
REFERENCES [dbo].[customers] ([customer_id])
GO
ALTER TABLE [dbo].[cart] ADD CONSTRAINT [FK_cart_Products] FOREIGN KEY ([product_id])
REFERENCES [dbo].[Products] ([Product_ID])
GO
EXEC sp_addextendedproperty N'MS_Description', N'Cart Shopping', 'SCHEMA', N'dbo',
'TABLE', N'cart', NULL, NULL
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key for Cart records.
Identity / Auto increment column', 'SCHEMA', N'dbo', 'TABLE', N'cart', 'COLUMN',
N'cart_id'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Unique customer identification number.
Foreign key referring to Customers.customer_id .', 'SCHEMA', N'dbo', 'TABLE', N'cart',
'COLUMN', N'customer_id'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Unique product identification number.
Foreign key to Products.product_id .', 'SCHEMA', N'dbo', 'TABLE', N'cart', 'COLUMN',
N'product_id'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Quantity of products in cart ',
'SCHEMA', N'dbo', 'TABLE', N'cart', 'COLUMN', N'quantity'
GO
```

Uses

[dbo].[customers]
[dbo].[Products]

Used By

[dbo].[add_product_cart_procedure]
[dbo].[AddToCart]
[dbo].[cart_product_details_procedure]
[dbo].[cart_products_update]
[dbo].[change_quantity_of_certain_product]
[dbo].[choose_order_products_from_cart]
[dbo].[empty_cart_from_products_procedure]
[dbo].[MakeOrder]
[dbo].[total_price_of_products_in_cart]

Links To

Table	Join	Title / Name / Description
Products	**Cart.Product_ID**=Product.Product_ID	FK_Cart_Product_ID Foreign key constraint referencing Product.Product_ID
Customers	**Cart.Customer_ID=** Customer.Customer_ID	FK_Cart_Customer_ID Foreign key constraint referencing Customers.Customer_ID

[dbo].[Category]

MS_Description

Category information

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability
PK C	Category_ID *Primary key for Category records.*	int	4	NOT NULL
	Category_name *Category Name*	varchar(50)	50	NULL allowed

Indexes

Key	Name	Key Columns	Unique
PK C	PK_Category	Category_ID	True

SQL Script

```
CREATE TABLE [dbo].[Category]
(
[Category_ID] [int] NOT NULL,
[Category_name] [varchar] (50) COLLATE Arabic_CI_AS NULL
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[Category] ADD CONSTRAINT [PK_Category] PRIMARY KEY CLUSTERED
([Category_ID]) ON [PRIMARY]
GO
EXEC sp_addextendedproperty N'MS_Description', N'Category information ', 'SCHEMA',
N'dbo', 'TABLE', N'Category', NULL, NULL
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key for Category records.',
'SCHEMA', N'dbo', 'TABLE', N'Category', 'COLUMN', N'Category_ID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Category Name', 'SCHEMA', N'dbo',
'TABLE', N'Category', 'COLUMN', N'Category_name'
GO
```

Used By

[dbo].[subcategory]
[dbo].[all_products_in_our_commerce]
[dbo].[get_quantity]
[dbo].[products_in_e_commerce]

[dbo].[search_about_product]
[dbo].[search_by_product]
[dbo].[view_all_products]

Linked From

Table	**Join**	Title / Name / Description
Subcategory	**Category_ID**=Subcategory.Category_ID	FK_Subcategory_Category_ID Foreign key constraint referencing Category.Category_ID

[dbo].[customer_Phone]

MS_Description

Customer Phone Numbers

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability
PK C FK	Customer_ID *Customer identification number.* *Foreign key to Customers.Customer_ID.* *Composite PK with customer_Phone.Phone .*	int	4	NOT NULL
PK C	Phone *Customer Phone numbers .* *Composite PK with customer_Phone.Customer_ID.*	varchar(50)	50	NOT NULL

Indexes

Key	Name	Key Columns	Unique
PK C	PK_customer_Phone	Customer_ID, Phone	True

Foreign Keys

Name	Columns
FK_customer_Phone_customers	Customer_ID->[dbo].[customers].[customer_id]

SQL Script

```
CREATE TABLE [dbo].[customer_Phone]
(
[Customer_ID] [int] NOT NULL,
[Phone] [varchar] (50) COLLATE Arabic_CI_AS NOT NULL
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[customer_Phone] ADD CONSTRAINT [PK_customer_Phone] PRIMARY KEY
CLUSTERED ([Customer_ID], [Phone]) ON [PRIMARY]
GO
ALTER TABLE [dbo].[customer_Phone] ADD CONSTRAINT [FK_customer_Phone_customers] FOREIGN
KEY ([Customer_ID]) REFERENCES [dbo].[customers] ([customer_id])
GO
EXEC sp_addextendedproperty N'MS_Description', N'Customer Phone Numbers ', 'SCHEMA',
N'dbo', 'TABLE', N'customer_Phone', NULL, NULL
GO
EXEC sp_addextendedproperty N'MS_Description', N'Customer identification number.
```

```
Foreign key to Customers.Customer_ID.
Composite PK with customer_Phone.Phone .', 'SCHEMA', N'dbo', 'TABLE', N'customer_Phone',
'COLUMN', N'Customer_ID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Customer Phone numbers .
Composite PK with customer_Phone.Customer_ID.', 'SCHEMA', N'dbo', 'TABLE', N'customer_-
Phone', 'COLUMN', N'Phone'
GO
```

Uses

[dbo].[customers]

Used By

[dbo].[customer_register]
[dbo].[registration_proc]

Links To

Table	Join	Title / Name / Description
Customers	**Customer.Customer_ID=** Customer_Phone.Customer_ID	FK_Customer_Phone_Customer_ID Foreign key constraint referencing Customers.Customer_ID

[dbo].[customers]

MS_Description

Customers information

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability	Identity
PK C	customer_id *Primary key for Customers records.* *Unique customer identification number.* *Identity / Auto increment column*	int	4	NOT NULL	1 - 1
	fname *Customer first name*	varchar(50)	50	NULL allowed	
	lname *Customer last name*	varchar(50)	50	NULL allowed	
▦	gender *Customer gender.*	varchar(50)	50	NULL allowed	
	zip_code *Zip code of city*	varchar(50)	50	NULL allowed	
	age *Customer age*	int	4	NULL allowed	
	country *Customer country*	varchar(50)	50	NULL allowed	
	city *Customer city*	varchar(50)	50	NULL allowed	
	password *Customer account password*	varchar(50)	50	NULL allowed	
	email *Customer account email*	varchar(50)	50	NULL allowed	
	account_id *Unique customer identification account number.*	int	4	NULL allowed	

Indexes

Key	Name	Key Columns	Unique
PK C	PK__customer__CD65CB850D9A9941	customer_id	True

Check Constraints

Name	On	Constraint

	Column	
CK__customers__gende__531856C7 *To check values of gender between only ('m'/'f') where :'m' refer to male and 'f' refer to female*	gender	([gender]='m' OR [gender]='f')

SQL Script

```
CREATE TABLE [dbo].[customers]
(
[customer_id] [int] NOT NULL IDENTITY(1, 1),
[fname] [varchar] (50) COLLATE Arabic_CI_AS NULL,
[lname] [varchar] (50) COLLATE Arabic_CI_AS NULL,
[gender] [varchar] (50) COLLATE Arabic_CI_AS NULL,
[zip_code] [varchar] (50) COLLATE Arabic_CI_AS NULL,
[age] [int] NULL,
[country] [varchar] (50) COLLATE Arabic_CI_AS NULL,
[city] [varchar] (50) COLLATE Arabic_CI_AS NULL,
[password] [varchar] (50) COLLATE Arabic_CI_AS NULL,
[email] [varchar] (50) COLLATE Arabic_CI_AS NULL,
[account_id] [int] NULL
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[customers] ADD CONSTRAINT [CK__customers__gende__531856C7] CHECK
(([gender]='m' OR [gender]='f'))
GO
ALTER TABLE [dbo].[customers] ADD CONSTRAINT [PK__customer__CD65CB850D9A9941] PRIMARY KEY
CLUSTERED ([customer_id]) ON [PRIMARY]
GO
EXEC sp_addextendedproperty N'MS_Description', N'Customers information', 'SCHEMA',
N'dbo', 'TABLE', N'customers', NULL, NULL
GO
EXEC sp_addextendedproperty N'MS_Description', N'Unique customer identification account
number. ', 'SCHEMA', N'dbo', 'TABLE', N'customers', 'COLUMN', N'account_id'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Customer age', 'SCHEMA', N'dbo',
'TABLE', N'customers', 'COLUMN', N'age'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Customer city', 'SCHEMA', N'dbo',
'TABLE', N'customers', 'COLUMN', N'city'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Customer country', 'SCHEMA', N'dbo',
'TABLE', N'customers', 'COLUMN', N'country'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key for Customers records.
Unique customer identification number.
Identity / Auto increment column', 'SCHEMA', N'dbo', 'TABLE', N'customers', 'COLUMN',
N'customer_id'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Customer  account email', 'SCHEMA',
N'dbo', 'TABLE', N'customers', 'COLUMN', N'email'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Customer first name', 'SCHEMA', N'dbo',
'TABLE', N'customers', 'COLUMN', N'fname'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Customer gender.', 'SCHEMA', N'dbo',
'TABLE', N'customers', 'COLUMN', N'gender'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Customer last name', 'SCHEMA', N'dbo',
'TABLE', N'customers', 'COLUMN', N'lname'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Customer  account password ', 'SCHEMA',
N'dbo', 'TABLE', N'customers', 'COLUMN', N'password'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Zip code of city', 'SCHEMA', N'dbo',
'TABLE', N'customers', 'COLUMN', N'zip_code'
GO
EXEC sp_addextendedproperty N'MS_Description', N'To check values of gender between only
(''m''/''f'') where :''m'' refer to male  and ''f'' refer to female ', 'SCHEMA', N'dbo',
'TABLE', N'customers', 'CONSTRAINT', N'CK__customers__gende__531856C7'
GO
```

Used By

[dbo].[cart]
[dbo].[customer_Phone]
[dbo].[orders]
[dbo].[ratings]
[dbo].[customer_register]
[dbo].[get_sales_by_age_category]
[dbo].[get_totalsales_by_age_category]
[dbo].[get_totalsales_by_city]
[dbo].[get_totalsales_by_gender]
[dbo].[Login_Proc]
[dbo].[registration_proc]
[dbo].[sales_per_age_category]
[dbo].[sales_per_gender]

Linked From

Table	**Join**	Title / Name / Description
Orders	**Customers.Customer_ID=**Orders.Customer_ID	FK_Orders_Customer_ID Foreign key constraint referencing **Customers.CustomerID** .
Ratings	**Customers.Customer_ID=**Ratings.Customer_ID	FK_Ratings_Customer_ID Foreign key constraint referencing **Customers.CustomerID** .
Cart	**Customers.Customer_ID=**Cart.Customer_ID	FK_Cart_Customer_ID Foreign key constraint referencing **Customers.CustomerID** .
Customer_Phone	**Customers.Customer_ID=**Customer_Phone.Customer_ID	FK_Customer_Phone_Customer_ID Foreign key constraint referencing **Customers.CustomerID** .

[dbo].[order_details]

MS_Description

Order details information

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability	Identity
PK C	order_detail_id *Primary key for order_detail records.* *Unique order detail identification number.* *Identity / Auto increment column*	int	4	NOT NULL	1 - 1
	quantity *Quantity of products in the order*	int	4	NULL allowed	
FK	order_id *Unique order identification number.* *Foreign key constraint referencing orders.order_id*	int	4	NULL allowed	
FK	product_id *Unique product identification number.* *Foreign key constraint referencing Products.product_id*	int	4	NULL allowed	
	line_total *Line total of the order*	int	4	NULL allowed	

Indexes

Key	Name	Key Columns	Unique
PK C	PK__order_de__3C5A40803C609D65	order_detail_id	True

Triggers

Name	ANSI Nulls On	Quoted Identifier On	On
lineTotal *Trigger to calculate line total from: line_total = quantity*price*	True	True	After Insert
tr_Calculate_SubTotal00 *Trigger to calculate sub total from: sub_total = sum of line_total*	True	True	After Insert Update

Foreign Keys

Name	Columns
FK_order_detai_orders	order_id->[dbo].[orders].[order_id]

FK_order_detai_Products	product_id->[dbo].[Products].[Product_ID]

SQL Script

```
CREATE TABLE [dbo].[order_details]
(
[order_detail_id] [int] NOT NULL IDENTITY(1, 1),
[quantity] [int] NULL,
[order_id] [int] NULL,
[product_id] [int] NULL,
[line_total] [int] NULL
) ON [PRIMARY]
GO
create trigger [dbo].[lineTotal]
on [dbo].[order_details]
after insert
as
   update Order_details
   set line_total = quantity*price
   from Products p inner join Order_details od
   on p.Product_ID = od.Product_ID
GO
CREATE TRIGGER [dbo].[tr_Calculate_SubTotal00]
ON [dbo].[order_details]
AFTER INSERT, UPDATE
AS
BEGIN
    SET NOCOUNT ON;
    UPDATE Orders
    SET sub_total = (SELECT SUM(line_total) FROM Order_details WHERE Order_ID =
Orders.Order_ID)
    FROM Orders
    JOIN inserted ON Orders.Order_ID = inserted.Order_ID;
END;
GO
ALTER TABLE [dbo].[order_details] ADD CONSTRAINT [PK__order_de__3C5A40803C609D65] PRIMARY
KEY CLUSTERED ([order_detail_id]) ON [PRIMARY]
GO
ALTER TABLE [dbo].[order_details] ADD CONSTRAINT [FK_order_detai_orders] FOREIGN KEY
([order_id]) REFERENCES [dbo].[orders] ([order_id])
GO
ALTER TABLE [dbo].[order_details] ADD CONSTRAINT [FK_order_detai_Products] FOREIGN KEY
([product_id]) REFERENCES [dbo].[Products] ([Product_ID])
GO
EXEC sp_addextendedproperty N'MS_Description', N'Order details information ', 'SCHEMA',
N'dbo', 'TABLE', N'order_details', NULL, NULL
GO
EXEC sp_addextendedproperty N'MS_Description', N'Line total of the order ', 'SCHEMA',
N'dbo', 'TABLE', N'order_details', 'COLUMN', N'line_total'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key for order_detail records.
Unique order detail identification number.
Identity / Auto increment column', 'SCHEMA', N'dbo', 'TABLE', N'order_details', 'COLUMN',
N'order_detail_id'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Unique order identification number.
Foreign key constraint referencing orders.order_id', 'SCHEMA', N'dbo', 'TABLE',
N'order_details', 'COLUMN', N'order_id'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Unique product identification number.
Foreign key constraint referencing Products.product_id', 'SCHEMA', N'dbo', 'TABLE',
N'order_details', 'COLUMN', N'product_id'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Quantity of products in the order',
'SCHEMA', N'dbo', 'TABLE', N'order_details', 'COLUMN', N'quantity'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Trigger to calculate line total from:
line_total = quantity*price', 'SCHEMA', N'dbo', 'TABLE', N'order_details', 'TRIGGER',
N'lineTotal'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Trigger to calculate  sub total
from: sub_total =  sum of line_total', 'SCHEMA', N'dbo', 'TABLE', N'order_details',
'TRIGGER', N'tr_Calculate_SubTotal00'
GO
```

Uses

[dbo].[orders]
[dbo].[Products]

Used By

[dbo].[choose_order_products_from_cart]
[dbo].[get_quantity]
[dbo].[LOWER_prod_Name_totalsales]
[dbo].[lowest_ten_products]
[dbo].[MakeOrder]
[dbo].[product_history]
[dbo].[products_sales_per_months]
[dbo].[profit]
[dbo].[top_prod_Name_totalsales]
[dbo].[top_ten_products]
[dbo].[totalsales_each_product_by_MONTH]

Links To

Table	Join	Title / Name / Description
Orders	**Order_details.Order_ID=**Orders.Order_ID	FK_Orders_details_Order_ID Foreign key constraint referencing Orders.Order_ID
Products	**Order_details.Order_ID=**Products.Product_ID	FK_Orders_details_Product_ID Foreign key constraint referencing Products.Product_ID .

[dbo].[orders]

MS_Description

Orders information

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability	Identity
PK C	order_id *Primary key for order records.* *Unique order identification number.* *Identity / Auto increment column*	int	4	NOT NULL	1 - 1
	sub_total *Sub total :Total Due with no tax or freight*	int	4	NULL allowed	
	total_tax *Order tax value*	int	4	NULL allowed	
	total_freight *Order freight value*	int	4	NULL allowed	
	total_due *Order total due value*	int	4	NULL allowed	
	shipping_method *Order shipping method*	varchar(50)	50	NULL allowed	
	shipping_city *Order shipping city*	varchar(50)	50	NULL allowed	
	payment_method *Order payment method*	varchar(50)	50	NULL allowed	
	order_date *Order date*	date	3	NULL allowed	
FK	customer_id *Unique customer identification number.* *Foreign key referring to Customers.customer_id* *.*	int	4	NULL allowed	

Indexes

Key	Name	Key Columns	Unique
PK C	PK__orders02__46596229A551B6A1	order_id	True

Triggers

Name	ANSI Nulls On	Quoted Identifier On	On
order_data *Trigger to calculate : total tax , total freight and total due*	True	True	After Update

OrderDate *Trigger to make order date is today date*	True	True	After Insert
shipping_method_select *Trigger to select shipping method based on shipping city*	True	True	After Insert

Foreign Keys

Name	Columns
FK_orders_customers	customer_id->[dbo].[customers].[customer_id]

SQL Script

```
CREATE TABLE [dbo].[orders]
(
[order_id] [int] NOT NULL IDENTITY(1, 1),
[sub_total] [int] NULL,
[total_tax] [int] NULL,
[total_freight] [int] NULL,
[total_due] [int] NULL,
[shipping_method] [varchar] (50) COLLATE Arabic_CI_AS NULL,
[shipping_city] [varchar] (50) COLLATE Arabic_CI_AS NULL,
[payment_method] [varchar] (50) COLLATE Arabic_CI_AS NULL,
[order_date] [date] NULL,
[customer_id] [int] NULL
) ON [PRIMARY]
GO
create trigger [dbo].[order_data]
on [dbo].[orders]
after update
as
update orders
set total_tax = sub_total*0.14, total_freight = sub_total*0.05, total_due=
sub_total+total_freight+total_tax
GO
create trigger [dbo].[OrderDate]
on [dbo].[orders]
after insert
as

   UPDATE Orders
   set Order_date= GETDATE() from orders inner join inserted on
orders.order_id=inserted.order_id




GO
CREATE TRIGGER [dbo].[shipping_method_select]
ON [dbo].[orders]
AFTER INSERT
AS
BEGIN
```

```
    DECLARE @shipping_method VARCHAR(50);
    DECLARE @shipping_city VARCHAR(50);

    SELECT @shipping_city = shipping_city FROM inserted;

    IF @shipping_city in ('cairo' , 'Alexandria','Aswan','Ismailia','Qena')
        SET @shipping_method = 'car';
    ELSE
        SET @shipping_method = 'train';

    UPDATE orders SET shipping_method = @shipping_method WHERE Order_ID = (SELECT Order_-
ID FROM inserted);
END;
GO
ALTER TABLE [dbo].[orders] ADD CONSTRAINT [PK__orders02__46596229A551B6A1] PRIMARY KEY
CLUSTERED ([order_id]) ON [PRIMARY]
GO
ALTER TABLE [dbo].[orders] ADD CONSTRAINT [FK_orders_customers] FOREIGN KEY
([customer_id]) REFERENCES [dbo].[customers] ([customer_id])
GO
EXEC sp_addextendedproperty N'MS_Description', N'Orders information ', 'SCHEMA', N'dbo',
'TABLE', N'orders', NULL, NULL
GO
EXEC sp_addextendedproperty N'MS_Description', N'Unique customer identification number.
Foreign key referring to Customers.customer_id .', 'SCHEMA', N'dbo', 'TABLE', N'orders',
'COLUMN', N'customer_id'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Order date', 'SCHEMA', N'dbo', 'TABLE',
N'orders', 'COLUMN', N'order_date'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key for order records.
Unique order identification number.
Identity / Auto increment column', 'SCHEMA', N'dbo', 'TABLE', N'orders', 'COLUMN',
N'order_id'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Order payment method', 'SCHEMA', N'dbo',
'TABLE', N'orders', 'COLUMN', N'payment_method'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Order shipping city', 'SCHEMA', N'dbo',
'TABLE', N'orders', 'COLUMN', N'shipping_city'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Order shipping method', 'SCHEMA',
N'dbo', 'TABLE', N'orders', 'COLUMN', N'shipping_method'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Sub total :Total Due with no tax or
freight ', 'SCHEMA', N'dbo', 'TABLE', N'orders', 'COLUMN', N'sub_total'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Order total due value', 'SCHEMA',
N'dbo', 'TABLE', N'orders', 'COLUMN', N'total_due'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Order freight value', 'SCHEMA', N'dbo',
'TABLE', N'orders', 'COLUMN', N'total_freight'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Order tax value ', 'SCHEMA', N'dbo',
'TABLE', N'orders', 'COLUMN', N'total_tax'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Trigger to calculate : total tax , total
```

```
freight and total due ', 'SCHEMA', N'dbo', 'TABLE', N'orders', 'TRIGGER', N'order_data'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Trigger to make order date is today
date ', 'SCHEMA', N'dbo', 'TABLE', N'orders', 'TRIGGER', N'OrderDate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Trigger to select shipping method based
on shipping city', 'SCHEMA', N'dbo', 'TABLE', N'orders', 'TRIGGER',
N'shipping_method_select'
GO
```

Uses

[dbo].[customers]

Used By

[dbo].[order_details]
[dbo].[choose_order_products_from_cart]
[dbo].[get_sales_by_age_category]
[dbo].[get_totalsales_by_age_category]
[dbo].[get_totalsales_by_city]
[dbo].[get_totalsales_by_DAy]
[dbo].[get_totalsales_by_gender]
[dbo].[get_totalsales_by_MONTH]
[dbo].[get_totalsales_by_Quarter]
[dbo].[get_totalsales_by_year]
[dbo].[LOWER_prod_Name_totalsales]
[dbo].[MakeOrder]
[dbo].[order_history]
[dbo].[overall_total_sales]
[dbo].[product_history]
[dbo].[products_sales_per_months]
[dbo].[qvelall_total_sales]
[dbo].[sales_per_age_category]
[dbo].[sales_per_city]
[dbo].[sales_per_gender]
[dbo].[top_prod_Name_totalsales]
[dbo].[total_sale_details]
[dbo].[totalsales_each_product_by_MONTH]

Linked from

Table	Join	Title / Name / Description
Order_details	**Orders.Order_ID=**Order_details.Order_ID	FK_Orders_details_Order_ID Foreign key constraint referencing Orders.Order_ID

Links To

Table	Join	Title / Name / Description
Customers	**Orders.Order_ID=**Customers.Order_ID	FK_Customers_Order_ID Foreign key constraint referencing Orders.Order_ID

[dbo].[product_brand]

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability
PK C FK	product_ID *Product identification number.* *Foreign key referring to Products.Product_ID.* *Composite PK with product_brand.Brand_ID.*	int	4	NOT NULL
PK C FK	Brand_ID *Brand identification number.* *Foreign key referring to Brand.Brand_ID.* *Composite PK with product_brand.Product_ID..*	int	4	NOT NULL

Indexes

Key	Name	Key Columns	Unique
PK C	PK_product_brand	product_ID, Brand_ID	True

Foreign Keys

Name	Columns
FK_product_brand_Brand	Brand_ID->[dbo].[Brand].[Brand_ID]
FK_product_brand_Products	product_ID->[dbo].[Products].[Product_ID]

SQL Script

```
CREATE TABLE [dbo].[product_brand]
(
[product_ID] [int] NOT NULL,
[Brand_ID] [int] NOT NULL
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[product_brand] ADD CONSTRAINT [PK_product_brand] PRIMARY KEY CLUSTERED
([product_ID], [Brand_ID]) ON [PRIMARY]
GO
ALTER TABLE [dbo].[product_brand] ADD CONSTRAINT [FK_product_brand_Brand] FOREIGN KEY
([Brand_ID]) REFERENCES [dbo].[Brand] ([Brand_ID])
GO
ALTER TABLE [dbo].[product_brand] ADD CONSTRAINT [FK_product_brand_Products] FOREIGN KEY
([product_ID]) REFERENCES [dbo].[Products] ([Product_ID])
GO
EXEC sp_addextendedproperty N'MS_Description', N'Brand identification number.
Foreign key referring to Brand.Brand_ID.
Composite PK with product_brand.Product_ID..', 'SCHEMA', N'dbo', 'TABLE',
N'product_brand', 'COLUMN', N'Brand_ID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Product identification number.
Foreign key referring to Products.Product_ID.
Composite PK with product_brand.Brand_ID.', 'SCHEMA', N'dbo', 'TABLE', N'product_brand',
'COLUMN', N'product_ID'
GO
```

Uses

[dbo].[Brand]
[dbo].[Products]

Links To

Table	Join	Title / Name / Description
Products	**Product_brand.Product_ID=**Products.Product_ID	FK_Produc_brand_Product_ID Foreign key constraint referencing Products.Product_ID
Brand	**Product_brand.Brand_ID=**Brand.Brand_ID	FK_Produc_brand_Brand_ID Foreign key constraint referencing Brand.Brand_ID

[dbo].[Product_Store]

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability
PK FK	Store_ID *Store identification number.* *Foreign key referring to Store.Store_ID* *Composite PK with product_Store.Product_ID.*	int	4	NOT NULL
PK FK	Product_ID *Product identification number.* *Foreign key referring to Products.Product_ID.* *Composite PK with product_store.Store_ID.*	int	4	NOT NULL

Indexes

Key	Name	Key Columns	Unique
PK	PK_Product_Store	Store_ID, Product_ID	True

Foreign Keys

Name	Columns
FK_Product_Store_Products	Product_ID->[dbo].[Products].[Product_ID]
FK_Product_Store_Store	Store_ID->[dbo].[Store].[Store_ID]

SQL Script

```
CREATE TABLE [dbo].[Product_Store]
(
[Store_ID] [int] NOT NULL,
[Product_ID] [int] NOT NULL
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[Product_Store] ADD CONSTRAINT [PK_Product_Store] PRIMARY KEY CLUSTERED
([Store_ID], [Product_ID]) ON [PRIMARY]
GO
ALTER TABLE [dbo].[Product_Store] ADD CONSTRAINT [FK_Product_Store_Products] FOREIGN KEY
([Product_ID]) REFERENCES [dbo].[Products] ([Product_ID])
GO
ALTER TABLE [dbo].[Product_Store] ADD CONSTRAINT [FK_Product_Store_Store] FOREIGN KEY
([Store_ID]) REFERENCES [dbo].[Store] ([Store_ID])
GO
EXEC sp_addextendedproperty N'MS_Description', N'Product identification number.
Foreign key referring to Products.Product_ID.
Composite PK with product_store.Store_ID.', 'SCHEMA', N'dbo', 'TABLE', N'Product_Store',
'COLUMN', N'Product_ID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Store identification number.
Foreign key referring to Store.Store_ID
Composite PK with product_Store.Product_ID.', 'SCHEMA', N'dbo', 'TABLE', N'Product_-
Store', 'COLUMN', N'Store_ID'
GO
```

Uses

[dbo].[Products]
[dbo].[Store]

Links To

Table	Join	Title / Name / Description
Products	**Product_Store.Product_ID=**Products.Product_ID	FK_Product_Store_Product_ID Foreign key constraint referencing Products.Product_ID
Store	**Product_Store.Store_ID=**Store.Store_ID	FK_Product_Store_Store_ID Foreign key constraint referencing Store.Store_ID

[dbo].[Products]

MS_Description

Products information

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability
PK C	Product_ID *Primary key for Products records.* *Product identification number.*	int	4	NOT NULL
	Product_name *Product Name*	varchar(50)	50	NULL allowed
	price *Product price in the website*	float	8	NULL allowed
	cost *Product cost from supplier*	float	8	NULL allowed
FK	subcategory_ID *Subcategory identification number.* *Foreign key referring to Subcategory.subcategory_ID.*	int	4	NULL allowed

Indexes

Key	Name	Key Columns	Unique
PK C	PK_Products	Product_ID	True

Foreign Keys

Name	Columns
FK_Products_subcategory	subcategory_ID->[dbo].[subcategory].[Subcategory_ID]

SQL Script

```
CREATE TABLE [dbo].[Products]
(
[Product_ID] [int] NOT NULL,
[Product_name] [varchar] (50) COLLATE Arabic_CI_AS NULL,
[price] [float] NULL,
[cost] [float] NULL,
[subcategory_ID] [int] NULL
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[Products] ADD CONSTRAINT [PK_Products] PRIMARY KEY CLUSTERED
([Product_ID]) ON [PRIMARY]
```

```
GO
ALTER TABLE [dbo].[Products] ADD CONSTRAINT [FK_Products_subcategory] FOREIGN KEY
([subcategory_ID]) REFERENCES [dbo].[subcategory] ([Subcategory_ID])
GO
EXEC sp_addextendedproperty N'MS_Description', N'Products information', 'SCHEMA', N'dbo',
'TABLE', N'Products', NULL, NULL
GO
EXEC sp_addextendedproperty N'MS_Description', N'Product cost from supplier ', 'SCHEMA',
N'dbo', 'TABLE', N'Products', 'COLUMN', N'cost'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Product price in the website', 'SCHEMA',
N'dbo', 'TABLE', N'Products', 'COLUMN', N'price'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key for Products records.
Product identification number. ', 'SCHEMA', N'dbo', 'TABLE', N'Products', 'COLUMN',
N'Product_ID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Product Name', 'SCHEMA', N'dbo',
'TABLE', N'Products', 'COLUMN', N'Product_name'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Subcategory identification number.
 Foreign key referring to Subcategory.subcategory_ID.', 'SCHEMA', N'dbo', 'TABLE',
N'Products', 'COLUMN', N'subcategory_ID'
GO
```

Uses

[dbo].[subcategory]

Used By

[dbo].[cart]
[dbo].[order_details]
[dbo].[product_brand]
[dbo].[Product_Store]
[dbo].[Supplier_Product]
[dbo].[all_products_in_our_commerce]
[dbo].[cart_product_details_procedure]
[dbo].[get_quantity]
[dbo].[LOWER_prod_Name_totalsales]
[dbo].[lowest_ten_products]
[dbo].[product_history]
[dbo].[products_in_e_commerce]
[dbo].[products_sales_per_months]
[dbo].[profit]
[dbo].[search_about_product]
[dbo].[search_by_product]
[dbo].[search_products]
[dbo].[top_prod_Name_totalsales]
[dbo].[top_ten_products]
[dbo].[total_price_of_products_in_cart]
[dbo].[totalsales_each_product_by_MONTH]
[dbo].[view_all_products]

Linked From

Table	**Join**	Title / Name / Description
Order_details	**Products.Product_ID=** Order_details.Product_ID	FK_Order_details _Product_ID Foreign key constraint referencing Products.Product_ID
Cart	**Products.Product_ID=** Cart.Product_ID	FK_Cart _Product_ID Foreign key constraint referencing Products.Product_ID
Product_brand	**Products.Product_ID=** Product_brand .Product_ID	FK_ Product_brand _Product_ID Foreign key constraint referencing Products.Product_ID
Supplier_Product	**Products.Product_ID=** Supplier_Product.Product_ID	FK_ Supplier_Product _Product_ID Foreign key constraint referencing Products.Product_ID
Product_Store	**Products.Product_ID=** Product_Store .Product_ID	FK_ Product_Store _Product_ID Foreign key constraint referencing Products.Product_ID

Links To

Table	**Join**	Title / Name / Description
Subcategory	**Products.SubcategoryID=**Subcategory.Subcategory_ID	FK_Products_Subcategory _ID Foreign key constraint referencing Subcategory.Subcategory_ID

▤ [dbo].[ratings]

MS_Description

Customer Rating of our services

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability	Identity
PK C	rating_id *Primary key for Ratings records.* *Rating identification number.* *Identity / Auto increment column*	int	4	NOT NULL	1 - 1
▤	overall_rate *Overall rate of service (rate from 5)*	int	4	NULL allowed	
▤	delivery_rate *Rate of delivery service (rate from 5)*	int	4	NULL allowed	
▤	customer_service_rate *Rate of customer service (rate from 5)*	int	4	NULL allowed	
▤	product_quality_rate *Rate of product quality (rate from 5)*	int	4	NULL allowed	
▤	loyality *Rate of loyalty (If customer will make order again or not rate from [yes-No])*	varchar(50)	50	NULL allowed	
FK	customer_id *Unique customer identification number.* *Foreign key referring to Customers.customer_id .*	int	4	NULL allowed	

Indexes

Key	Name	Key Columns	Unique
PK C	PK__ratings__D35B278B87CF6DB9	rating_id	True

Check Constraints

Name	On Column	Constraint
CK__ratings__custome__6FB49575 *To check values between 1:5 only*	customer_service_rate	([customer_service_rate]<=(5))
CK__ratings__deliver__6EC0713C *To check values between 1:5 only*	delivery_rate	([delivery_rate]<=(5))
CK__ratings__loyalit__719CDDE7 *To check answers between "Yes"/"No" only*	loyality	([loyality]='no' OR [loyality]='yes')
CK__ratings__overall__6DCC4D03 *To check values between 1:5 only*	overall_rate	([overall_rate]<=(5))
CK__ratings__product__70A8B9AE	product_quality_rate	([product_quality_rate]<=(5))

To check values between 1:5 only		

Foreign Keys

Name	Columns
FK_ratings_customers	customer_id->[dbo].[customers].[customer_id]

SQL Script

```
CREATE TABLE [dbo].[ratings]
(
[rating_id] [int] NOT NULL IDENTITY(1, 1),
[overall_rate] [int] NULL,
[delivery_rate] [int] NULL,
[customer_service_rate] [int] NULL,
[product_quality_rate] [int] NULL,
[loyality] [varchar] (50) COLLATE Arabic_CI_AS NULL,
[customer_id] [int] NULL
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[ratings] ADD CONSTRAINT [CK__ratings__custome__6FB49575] CHECK
(([customer_service_rate]<=(5)))
GO
ALTER TABLE [dbo].[ratings] ADD CONSTRAINT [CK__ratings__deliver__6EC0713C] CHECK
(([delivery_rate]<=(5)))
GO
ALTER TABLE [dbo].[ratings] ADD CONSTRAINT [CK__ratings__loyalit__719CDDE7] CHECK
(([loyality]='no' OR [loyality]='yes'))
GO
ALTER TABLE [dbo].[ratings] ADD CONSTRAINT [CK__ratings__overall__6DCC4D03] CHECK
(([overall_rate]<=(5)))
GO
ALTER TABLE [dbo].[ratings] ADD CONSTRAINT [CK__ratings__product__70A8B9AE] CHECK
(([product_quality_rate]<=(5)))
GO
ALTER TABLE [dbo].[ratings] ADD CONSTRAINT [PK__ratings__D35B278B87CF6DB9] PRIMARY KEY
CLUSTERED ([rating_id]) ON [PRIMARY]
GO
ALTER TABLE [dbo].[ratings] ADD CONSTRAINT [FK_ratings_customers] FOREIGN KEY
([customer_id]) REFERENCES [dbo].[customers] ([customer_id])
GO
EXEC sp_addextendedproperty N'MS_Description', N'Customer Rating of our services ',
'SCHEMA', N'dbo', 'TABLE', N'ratings', NULL, NULL
GO
EXEC sp_addextendedproperty N'MS_Description', N'Unique customer identification number.
Foreign key referring to Customers.customer_id .', 'SCHEMA', N'dbo', 'TABLE', N'ratings',
'COLUMN', N'customer_id'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Rate of customer service (rate from 5)',
'SCHEMA', N'dbo', 'TABLE', N'ratings', 'COLUMN', N'customer_service_rate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Rate of delivery service (rate from 5)',
'SCHEMA', N'dbo', 'TABLE', N'ratings', 'COLUMN', N'delivery_rate'
```

```
GO
EXEC sp_addextendedproperty N'MS_Description', N'Rate of loyalty (If customer will make
order again or not rate from [yes-No])', 'SCHEMA', N'dbo', 'TABLE', N'ratings', 'COLUMN',
N'loyality'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Overall rate of service (rate from 5)  ',
'SCHEMA', N'dbo', 'TABLE', N'ratings', 'COLUMN', N'overall_rate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Rate of product quality ( rate from 5)',
'SCHEMA', N'dbo', 'TABLE', N'ratings', 'COLUMN', N'product_quality_rate'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key for Ratings records.
Rating identification number.
Identity / Auto increment column', 'SCHEMA', N'dbo', 'TABLE', N'ratings', 'COLUMN',
N'rating_id'
GO
EXEC sp_addextendedproperty N'MS_Description', N'To check values between 1:5 only',
'SCHEMA', N'dbo', 'TABLE', N'ratings', 'CONSTRAINT', N'CK__ratings__custome__6FB49575'
GO
EXEC sp_addextendedproperty N'MS_Description', N'To check values between 1:5 only',
'SCHEMA', N'dbo', 'TABLE', N'ratings', 'CONSTRAINT', N'CK__ratings__deliver__6EC0713C'
GO
EXEC sp_addextendedproperty N'MS_Description', N'To check answers  between "Yes"/"No"
only', 'SCHEMA', N'dbo', 'TABLE', N'ratings', 'CONSTRAINT',
N'CK__ratings__loyalit__719CDDE7'
GO
EXEC sp_addextendedproperty N'MS_Description', N'To check values between 1:5 only',
'SCHEMA', N'dbo', 'TABLE', N'ratings', 'CONSTRAINT', N'CK__ratings__overall__6DCC4D03'
GO
EXEC sp_addextendedproperty N'MS_Description', N'To check values between 1:5 only',
'SCHEMA', N'dbo', 'TABLE', N'ratings', 'CONSTRAINT', N'CK__ratings__product__70A8B9AE'
GO
```

Uses

[dbo].[customers]

Used By

[dbo].[Rating_survey]

Links To

Table	Join	Title / Name / Description
Customers	**Raitings.Customer_ID=**Customers.Customer_ID	FK_Ratings_Customer_ID Foreign key constraint referencing Customers.Customer._ID

[dbo].[Store]

MS_Description

Store information

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability
PK C	Store_ID *Primary key for Store records.* *Store unique identification number.*	int	4	NOT NULL
	Store_name *Store name*	varchar(50)	50	NULL allowed
	region *Store region*	varchar(50)	50	NULL allowed

Indexes

Key	Name	Key Columns	Unique
PK C	PK_Store	Store_ID	True

SQL Script

```
CREATE TABLE [dbo].[Store]
(
[Store_ID] [int] NOT NULL,
[Store_name] [varchar] (50) COLLATE Arabic_CI_AS NULL,
[region] [varchar] (50) COLLATE Arabic_CI_AS NULL
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[Store] ADD CONSTRAINT [PK_Store] PRIMARY KEY CLUSTERED ([Store_ID]) ON
[PRIMARY]
GO
EXEC sp_addextendedproperty N'MS_Description', N'Store information', 'SCHEMA', N'dbo',
'TABLE', N'Store', NULL, NULL
GO
EXEC sp_addextendedproperty N'MS_Description', N'Store region', 'SCHEMA', N'dbo',
'TABLE', N'Store', 'COLUMN', N'region'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key for Store records.
Store unique identification number. ', 'SCHEMA', N'dbo', 'TABLE', N'Store', 'COLUMN',
N'Store_ID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Store name', 'SCHEMA', N'dbo', 'TABLE',
N'Store', 'COLUMN', N'Store_name'
GO
```

Used By

[dbo].[Product_Store]

Linked From

Table	Join	Title / Name / Description
Product_Store	**Store.Store_ID=** Product_Store.Store_ID	FK_ Product_Store.Store_ID Foreign key constraint referencing Store.Store_ID

[dbo].[subcategory]

MS_Description

Subcategory information

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability
PK C	Subcategory_ID *Primary key for subcategory records.* *Subcategory unique identification number.*	int	4	NOT NULL
FK	Category_ID *Primary key for Category records.* *Foreign key referring to Category.Category.ID*	int	4	NULL allowed
	Subcategory_name *Subcategory name*	varchar(50)	50	NULL allowed

Indexes

Key	Name	Key Columns	Unique
PK C	PK_subcategory	Subcategory_ID	True

Foreign Keys

Name	Columns
FK_subcategory_Category	Category_ID->[dbo].[Category].[Category_ID]

SQL Script

```sql
CREATE TABLE [dbo].[subcategory]
(
[Subcategory_ID] [int] NOT NULL,
[Category_ID] [int] NULL,
[Subcategory_name] [varchar] (50) COLLATE Arabic_CI_AS NULL
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[subcategory] ADD CONSTRAINT [PK_subcategory] PRIMARY KEY CLUSTERED
([Subcategory_ID]) ON [PRIMARY]
GO
ALTER TABLE [dbo].[subcategory] ADD CONSTRAINT [FK_subcategory_Category] FOREIGN KEY
([Category_ID]) REFERENCES [dbo].[Category] ([Category_ID])
GO
EXEC sp_addextendedproperty N'MS_Description', N'Subcategory information', 'SCHEMA',
N'dbo', 'TABLE', N'subcategory', NULL, NULL
```

```
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key for Category records.
Foreign key referring to Category.Category.ID', 'SCHEMA', N'dbo', 'TABLE',
N'subcategory', 'COLUMN', N'Category_ID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key for subcategory records.
Subcategory unique identification number. ', 'SCHEMA', N'dbo', 'TABLE', N'subcategory',
'COLUMN', N'Subcategory_ID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Subcategory name', 'SCHEMA', N'dbo',
'TABLE', N'subcategory', 'COLUMN', N'Subcategory_name'
GO
```

Uses

[dbo].[Category]

Used By

[dbo].[Products]

[dbo].[all_products_in_our_commerce]

[dbo].[get_quantity]

[dbo].[products_in_e_commerce]

[dbo].[search_about_product]

[dbo].[search_by_product]

[dbo].[view_all_products]

Linked From

Table	**Join**	Title / Name / Description
Products	**Subcategory.Subcategory_ID=** Products.Subcategory_ID	FK_ Products_.Subcategory_ID Foreign key constraint referencing Subcategory.Subcategory_ID

Links To

Table	**Join**	Title / Name / Description
Category	**Subcategory.Category_ID=** Category.Category_ID	FK_ Subcategory_Category_ID Foreign key constraint referencing Category.Category_ID

[dbo].[Supplier]

MS_Description

Supplier information

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability
PK C	Supplier_ID *Primary key for Supplier records.* *Supplier unique identification number.*	int	4	NOT NULL
	supplier_name *Supplier name*	varchar(50)	50	NULL allowed
	location *Supplier location*	varchar(50)	50	NULL allowed

Indexes

Key	Name	Key Columns	Unique
PK C	PK_Supplier	Supplier_ID	True

SQL Script

```
CREATE TABLE [dbo].[Supplier]
(
[Supplier_ID] [int] NOT NULL,
[supplier_name] [varchar] (50) COLLATE Arabic_CI_AS NULL,
[location] [varchar] (50) COLLATE Arabic_CI_AS NULL
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[Supplier] ADD CONSTRAINT [PK_Supplier] PRIMARY KEY CLUSTERED
([Supplier_ID]) ON [PRIMARY]
GO
EXEC sp_addextendedproperty N'MS_Description', N'Supplier information', 'SCHEMA', N'dbo',
'TABLE', N'Supplier', NULL, NULL
GO
EXEC sp_addextendedproperty N'MS_Description', N'Supplier location', 'SCHEMA', N'dbo',
'TABLE', N'Supplier', 'COLUMN', N'location'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Primary key for Supplier records.
Supplier unique identification number. ', 'SCHEMA', N'dbo', 'TABLE', N'Supplier',
'COLUMN', N'Supplier_ID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Supplier name', 'SCHEMA', N'dbo',
'TABLE', N'Supplier', 'COLUMN', N'supplier_name'
GO
```

Used By

[dbo].[Supplier_Product]

Linked From

Table	Join	Title / Name / Description
Supplier_Product	**Supplier.Supplier_ID=** Supplier_Product.Supplier_ID	FK _Supplier_Product_Supplier_ID Foreign key constraint referencing Supplier.Supplier_ID

[dbo].[Supplier_Product]

Columns

Key	Name	Data Type	Max Length (Bytes)	Nullability
PK C FK	Supplier_ID *Supplier identification number.* *Foreign key referring to Supplier.Supplier_ID* *Composite PK with Supplier_product.Product_ID.*	int	4	NOT NULL
PK C FK	Product_ID *Product identification number.* *Foreign key referring to Products.Product_ID.* *Composite PK with Supplier_product.Supplier_ID.*	int	4	NOT NULL

Indexes

Key	Name	Key Columns	Unique
PK C	PK_Supplier_Product	Supplier_ID, Product_ID	True

Foreign Keys

Name	Columns
FK_Supplier_Product_Products	Product_ID->[dbo].[Products].[Product_ID]
FK_Supplier_Product_Supplier	Supplier_ID->[dbo].[Supplier].[Supplier_ID]

SQL Script

```
CREATE TABLE [dbo].[Supplier_Product]
(
[Supplier_ID] [int] NOT NULL,
[Product_ID] [int] NOT NULL
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[Supplier_Product] ADD CONSTRAINT [PK_Supplier_Product] PRIMARY KEY
CLUSTERED ([Supplier_ID], [Product_ID]) ON [PRIMARY]
GO
ALTER TABLE [dbo].[Supplier_Product] ADD CONSTRAINT [FK_Supplier_Product_Products]
FOREIGN KEY ([Product_ID]) REFERENCES [dbo].[Products] ([Product_ID])
GO
ALTER TABLE [dbo].[Supplier_Product] ADD CONSTRAINT [FK_Supplier_Product_Supplier]
FOREIGN KEY ([Supplier_ID]) REFERENCES [dbo].[Supplier] ([Supplier_ID])
GO
EXEC sp_addextendedproperty N'MS_Description', N'Product identification number.
Foreign key referring to Products.Product_ID.
Composite PK with Supplier_product.Supplier_ID.', 'SCHEMA', N'dbo', 'TABLE', N'Supplier_-
```

```
Product', 'COLUMN', N'Product_ID'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Supplier identification number.
Foreign key referring to Supplier.Supplier_ID
Composite PK with Supplier_product.Product_ID.', 'SCHEMA', N'dbo', 'TABLE', N'Supplier_-
Product', 'COLUMN', N'Supplier_ID'
GO
```

Uses

[dbo].[Products]
[dbo].[Supplier]

Links To

Table	**Join**	Title / Name / Description
Products	**Supplier_Product.Product_ID=**Products.Product_ID	FK _Supplier_Product_Product_ID Foreign key constraint referencing Products.Product_ID
Supplier	**Supplier_Product.Supplier_ID=**Supplier.Supplier_ID	FK _Supplier_Product_Supplier_ID Foreign key constraint referencing Supplier.Supplier_ID

Stored Procedures

Objects

Name
[dbo].[Brand_products]
dbo.AddToCart *Use product_id ,customer_id ,quantity to fill cart*
dbo.all_products_in_our_commerce
dbo.cart_product_details_procedure
dbo.cart_products_update
dbo.change_quantity_of_certain_product *Use product_id ,customer_id ,quantity to modify the quantity on the cart*
dbo.choose_order_products_from_cart
dbo.create_rating_survey
dbo.customer_rates
dbo.customer_rates_based_on_loyality
dbo.customer_register *Register to the website by using name, password, Email, gender, age , city ,phone and zip cod of city*
dbo.empty_cart_from_products_procedure
dbo.get_quantity
dbo.get_sales_by_age_category *Get sales by category*
dbo.get_totalsales_by_age_category
dbo.get_totalsales_by_city
dbo.get_totalsales_by_DAy
dbo.get_totalsales_by_gender
dbo.get_totalsales_by_MONTH
dbo.get_totalsales_by_Quarter
dbo.get_totalsales_by_year
dbo.Login_Proc

Login_procedure that takes two parameters email and password
dbo.LOWER_prod_Name_totalsales
dbo.lowest_ten_products *Show lowest 10 products*
dbo.loyal_customer_rates
dbo.MakeOrder *Use Customers.customer_id ,Orders.payment_method,Orders.shipping_city to make order*
dbo.nonloyal_customers_rates *Show the non-loyal customer rate*
dbo.order_history *Use Customers.customer_id to show the previous order of him/her*
dbo.overall_total_sales *Show total sales by sum orders.total_due*
dbo.product_history *Use customers.customer_id to show the product history of him/her*
dbo.products_in_e_commerce
dbo.products_sales_per_months *Use Orders.order_date to show product sales per month*
dbo.profit
dbo.qvelall_total_sales
dbo.Rating_survey *Use Customers.Customer_ID , Ratings.Overall_rate, Ratings.delivery_rate , Ratings.customer_service_rate,Ratings.Product_quality_rate and Ratings.Loyality*
dbo.registration_proc
dbo.sales_per_age_category
dbo.sales_per_city *Sales per city*
dbo.sales_per_gender *Sales per gender*
dbo.search_about_product
dbo.search_by_product *Use keyword var to search the product name in the website by any letters of its name*
dbo.search_products
dbo.top_prod_Name_totalsales
dbo.top_ten_products
dbo.total_price_of_products_in_cart
dbo.total_sale_details
dbo.totalsales_each_product_by_MONTH

dbo.view_all_products
Show all products

[dbo].[Brand_products]

Parameters

Name	Data Type	Max Length (Bytes)
@Brand	varchar(30)	30

SQL Script

```
CREATE proc [dbo].[Brand_products] @Brand varchar(30)
AS
select  Distinct pr.Product_name,
pr.price,pr.Cost,b.brand_country,sub.Subcategory_name,c.Category_name
from brand b join product_brand pd on b.Brand_ID =pd.Brand_ID
join Products pr on pr.Product_ID=pd.product_ID
join subcategory sub on sub.Subcategory_ID=pr.subcategory_ID
join Category c on c.Category_ID=sub.Category_ID
where b.brand_name=@Brand
GO
```

Uses

[dbo].[Brand]
[dbo].[Category]
[dbo].[product_brand]
[dbo].[Products]
[dbo].[subcategory]

[dbo].[AddToCart]

MS_Description

Use product_id ,customer_id ,quantity to fill cart

Parameters

Name	Data Type	Max Length (Bytes)
@product_id	varchar(50)	50
@customer_id	varchar(50)	50
@quantity	varchar(50)	50

SQL Script

```
CREATE proc [dbo].[AddToCart]
@product_id varchar(50),
@customer_id varchar(50),
@quantity varchar(50)
AS
declare @i int = 1
while @i <= count(@product_id)
begin
  insert into cart(product_id, customer_id, quantity)
  values(@product_id,@customer_id,@quantity)
  set @i = @i + 1
end
GO
EXEC sp_addextendedproperty N'MS_Description', N'
 Use product_id ,customer_id ,quantity to fill cart', 'SCHEMA', N'dbo', 'PROCEDURE',
N'AddToCart', NULL, NULL
GO
```

Uses

[dbo].[cart]

Project > . > User databases > E-Commerce > Programmability > Stored Procedures > dbo.all_products_in_our_commerce

[dbo].[all_products_in_our_commerce]

SQL Script

```
CREATE PROC [dbo].[all_products_in_our_commerce]
AS
    SELECT Product_name  , price , Subcategory_name , Category_name
    FROM Products AS  p INNER JOIN  subcategory AS sc
    ON p.subcategory_ID = sc.Subcategory_ID
    INNER JOIN     Category c
    ON sc.Category_ID = c.Category_ID
GO
```

Uses

[dbo].[Category]

[dbo].[Products]

[dbo].[subcategory]

Project > . > User databases > E-Commerce > Programmability > Stored Procedures > dbo.cart_product_details_procedure

[dbo].[cart_product_details_procedure]

Parameters

Name	Data Type	Max Length (Bytes)
@customer_id	int	4

SQL Script

```
CREATE PROC [dbo].[cart_product_details_procedure]
    @customer_id INT
AS
    SELECT customer_id , c.product_id , p.price*c.quantity AS total_pay
    FROM cart AS c INNER JOIN Products AS p
    ON c.product_id = p.Product_ID
    WHERE c.customer_id = @customer_id
GO
```

Uses

[dbo].[cart]
[dbo].[Products]

[dbo].[cart_products_update]

Parameters

Name	Data Type	Max Length (Bytes)
@newproduct_id	int	4
@customer_id	int	4
@newquantity	int	4
@add_sub	varchar	1

SQL Script

```
CREATE PROC [dbo].[cart_products_update]
    @newproduct_id INT,
    @customer_id INT,
    @newquantity INT,
    @add_sub varchar(1)

AS
                IF(EXISTS(SELECT * FROM cart WHERE  product_id =  @newproduct_id ))
                    BEGIN
                        IF @add_sub = '+'
                            BEGIN
                                UPDATE cart
                                SET quantity = quantity + @newquantity
                                WHERE customer_id = @customer_id and product_id =
@newproduct_id
                            END
                        IF @add_sub = '-'
                            BEGIN
                                UPDATE cart
                                SET quantity = quantity - @newquantity
                                WHERE customer_id = @customer_id and product_id =
@newproduct_id
                            END
                    END
                ELSE
                    BEGIN
                        INSERT INTO cart(product_id,customer_id,quantity)
                        VALUES(@newproduct_id,@customer_id,@newquantity)
                    END
GO
```

Uses

[dbo].[cart]

[dbo].[change_quantity_of_certain_product]

MS_Description

Use product_id ,customer_id ,quantity to modify the quantity on the cart

Parameters

Name	Data Type	Max Length (Bytes)
@customer_id	int	4
@product_id	int	4
@new_quantity	int	4

SQL Script

```sql
create proc [dbo].[change_quantity_of_certain_product]
@customer_id int,
@product_id int,
@new_quantity int
as
update cart
set quantity = @new_quantity
where customer_id = @customer_id and product_id = @product_id
GO
EXEC sp_addextendedproperty N'MS_Description', N' Use product_id ,customer_id ,quantity
to modify the quantity on the cart ', 'SCHEMA', N'dbo', 'PROCEDURE',
N'change_quantity_of_certain_product', NULL, NULL
GO
```

Uses

[dbo].[cart]

[dbo].[choose_order_products_from_cart]

Parameters

Name	Data Type	Max Length (Bytes)
@customer_id	int	4
@payment_method	varchar(50)	50
@shipping_city	varchar(50)	50

SQL Script

```
CREATE PROC [dbo].[choose_order_products_from_cart]
    @customer_id INT,
    @payment_method VARCHAR(50),
    @shipping_city VARCHAR(50)
AS
    DECLARE @order_id INT
    INSERT INTO orders(customer_id,payment_method,shipping_city)
    VALUES(@customer_id,@payment_method,@shipping_city)
    SET @order_id = SCOPE_IDENTITY()

    DECLARE @product_id INT , @quantit INT
    INSERT INTO order_details(quantity,product_id,order_id)
    SELECT product_id, quantity , @order_id
    FROM cart
    WHERE customer_id = @customer_id

    DELETE FROM cart
    WHERE customer_id = @customer_id
GO
```

Uses

[dbo].[cart]
[dbo].[order_details]
[dbo].[orders]

[dbo].[create_rating_survey]

Parameters

Name	Data Type	Max Length (Bytes)
@survey_id	int	4
@overall_rate	int	4
@delivery_rate	int	4
@customer_service_rate	int	4
@loyality	varchar(3)	3
@product_quality_rate	int	4
@customer_id	int	4

SQL Script

```
CREATE PROC [dbo].[create_rating_survey]
    @survey_id INT,
    @overall_rate INT,
    @delivery_rate INT,
    @customer_service_rate INT,
    @loyality VARCHAR(3),
    @product_quality_rate INT,
    @customer_id INT
AS
    INSERT INTO
rating(survey_id,overall_rate,delivery_rate,customer_service_rate,loyality,product_quality_rate,customer_id)

VALUES(@survey_id,@overall_rate,@delivery_rate,@customer_service_rate,@loyality,@product_quality_rate,@custom
GO
```

[dbo].[customer_rates]

SQL Script

```
CREATE PROC [dbo].[customer_rates]
AS
    SELECT  AVG(overall_rate) AS Average_Over_all_rate,
           AVG(delivery_rate) AS Average_delivery_rate,
           AVG(customer_service_rate) AS Average_customer_service_rate,
           AVG(product_quality_rate) AS Average_product_quality_rate
    FROM rating
GO
```

[dbo].[customer_rates_based_on_loyality]

SQL Script

```
CREATE PROC [dbo].[customer_rates_based_on_loyality]
AS
    SELECT loyality, AVG(overall_rate) AS Average_Over_all_rate,
           AVG(delivery_rate) AS Average_delivery_rate,
           AVG(customer_service_rate) AS Average_customer_service_rate,
           AVG(product_quality_rate) AS Average_product_quality_rate
    FROM rating
    GROUP BY loyality
GO
```

[dbo].[customer_register]

MS_Description

Register to the website by using name, password, Email, gender, age , city ,phone and zip cod of city

Parameters

Name	Data Type	Max Length (Bytes)
@fname	varchar(50)	50
@lname	varchar(50)	50
@gender	varchar(50)	50
@zip	varchar(50)	50
@Age	int	4
@country	varchar(50)	50
@city	varchar(50)	50
@password	varchar(50)	50
@email	varchar(50)	50
@phone	varchar(11)	11

SQL Script

```
CREATE proc [dbo].[customer_register]
@fname varchar(50),
@lname varchar(50),
@gender varchar(50),
@zip varchar(50),
@Age int,
@country varchar(50),
@city varchar(50),
@password varchar(50),
@email varchar(50),
@phone varchar(11)
AS
declare @lastcustomerid int
if (exists (select * from customers where email = @email))
begin
select 'this email is already used' as error_massege
end
else
begin
insert into customers(fname, lname, gender, zip_code,age, country,city, password, email)
values(@fname,@lname,@gender, @zip,@age, @country,@city, @password, @email)
set @lastcustomerid = scope_identity()
insert into customer_phone values (@lastcustomerid, @phone)
```

```
end
GO
EXEC sp_addextendedproperty N'MS_Description', N'Register to the website by using name,
password, Email, gender, age , city ,phone and zip cod of city', 'SCHEMA', N'dbo',
'PROCEDURE', N'customer_register', NULL, NULL
GO
```

Uses

[dbo].[customer_Phone]

[dbo].[customers]

[dbo].[empty_cart_from_products_procedure]

Parameters

Name	Data Type	Max Length (Bytes)
@customer_id	int	4

SQL Script

```
CREATE PROC [dbo].[empty_cart_from_products_procedure]
    @customer_id INT

AS
    DELETE FROM cart
    WHERE customer_id = @customer_id
GO
```

Uses

[dbo].[cart]

[dbo].[get_quantity]

Parameters

Name	Data Type	Max Length (Bytes)
@category_name	varchar(50)	50
@Subcategory_name	varchar(50)	50

SQL Script

```
CREATE proc [dbo].[get_quantity] @category_name varchar(50) , @Subcategory_name
varchar(50)
as

select Product_name , SUM(quantity) as Total_quantity from Products p
join order_details od on od.product_id = p.Product_ID
join subcategory sb on p.subcategory_ID = sb.Subcategory_ID
join Category c on c.Category_ID = sb.Category_ID
where Category_name = @category_name and Subcategory_name = @Subcategory_name
group by Product_name
GO
```

Uses

[dbo].[Category]
[dbo].[order_details]
[dbo].[Products]
[dbo].[subcategory]

[dbo].[get_sales_by_age_category]

MS_Description

Get sales by category

SQL Script

```
CREATE PROCEDURE [dbo].[get_sales_by_age_category]
AS
declare @categories table (age_category varchar(50), total_sales int)
insert into @categories (age_category, total_sales)
SELECT
        CASE
            WHEN age <= 20 THEN 'Under 20'
            WHEN age BETWEEN 21 AND 30 THEN '21-30'
            WHEN age BETWEEN 31 AND 40 THEN '31-40'
            WHEN age BETWEEN 41 AND 50 THEN '41-50'
            WHEN age > 50 THEN 'Over 50'
        END,
        sum(total_due)
    FROM orders JOIN customers
    ON orders.customer_id = customers.customer_id
    GROUP BY age;
select sum(total_sales), age_category
from @categories group by age_category

GO
EXEC sp_addextendedproperty N'MS_Description', N'Get sales by category', 'SCHEMA',
N'dbo', 'PROCEDURE', N'get_sales_by_age_category', NULL, NULL
GO
```

Uses

[dbo].[customers]
[dbo].[orders]

Project > . > User databases > E-Commerce > Programmability > Stored Procedures > dbo.get_totalsales_by_age_category

[dbo].[get_totalsales_by_age_category]

SQL Script

```
CREATE PROC [dbo].[get_totalsales_by_age_category]
AS
    DECLARE @categories TABLE(age_category VARCHAR(20) , total_sales INT)
    INSERT INTO @categories(age_category,total_sales)
    SELECT
      age = CASE
                WHEN age < 25 THEN  'Under 25'
                WHEN age BETWEEN 25 AND 35 THEN  '25-35'
                WHEN age BETWEEN 35 AND 45 THEN  '35-45'
                WHEN age BETWEEN 45 AND 55 THEN  '45-55'
                WHEN age >55 THEN  'over 55'
             END,
      SUM(total_due)
    FROM orders AS o INNER JOIN customers AS c
    ON o.customer_id = c.customer_id
    GROUP BY age
    SELECT SUM(total_sales) AS total_sales , age_category
    FROM @categories
    GROUP BY age_category
GO
```

Uses

[dbo].[customers]
[dbo].[orders]

[dbo].[get_totalsales_by_city]

SQL Script

```sql
CREATE PROC [dbo].[get_totalsales_by_city]
AS
    SELECT city , SUM(total_due) AS total_sales
    FROM customers AS c INNER JOIN orders AS o
    ON c.customer_id = o.customer_id
    GROUP BY city
GO
```

Uses

[dbo].[customers]
[dbo].[orders]

[dbo].[get_totalsales_by_DAy]

SQL Script

```
CREATE PROC [dbo].[get_totalsales_by_DAy]
AS
    SELECT TOP 10 SUM(total_due) AS total_sales ,
            DATEPART(DAY, order_date) AS DAY_of_order ,
            DATENAME(WEEKDAY, order_date) AS DAY_of_order
    FROM orders
    GROUP BY DATEPART(DAY, order_date),DATENAME(WEEKDAY, order_date)
    ORDER BY SUM(total_due) DESC
GO
```

Uses

[dbo].[orders]

[dbo].[get_totalsales_by_gender]

SQL Script

```sql
CREATE PROC [dbo].[get_totalsales_by_gender]
AS
    SELECT gender , SUM(total_due) AS total_sales
    FROM customers AS c INNER JOIN orders AS o
    ON c.customer_id = o.customer_id
    GROUP BY gender
GO
```

Uses

[dbo].[customers]
[dbo].[orders]

[dbo].[get_totalsales_by_MONTH]

SQL Script

```
CREATE PROC [dbo].[get_totalsales_by_MONTH]
AS
    SELECT TOP 5 SUM(total_due) AS total_sales ,
            DATEPART(MONTH, order_date) AS Month_of_order ,
            DATENAME(MONTH, order_date) AS Month_of_order
    FROM  orders
    GROUP BY DATEPART(MONTH, order_date),
             DATENAME(MONTH, order_date)
    ORDER BY SUM(total_due) DESC
GO
```

Uses

[dbo].[orders]

Project > . > User databases > E-Commerce > Programmability > Stored Procedures > dbo.get_totalsales_by_Quarter

[dbo].[get_totalsales_by_Quarter]

SQL Script

```
CREATE PROC [dbo].[get_totalsales_by_Quarter]
AS
    SELECT SUM(total_due) AS total_sales ,
           DATEPART(QUARTER, order_date) AS Quarter_of_order
    FROM orders
    GROUP BY DATEPART(QUARTER, order_date)
GO
```

Uses

[dbo].[orders]

[dbo].[get_totalsales_by_year]

SQL Script

```
CREATE PROC [dbo].[get_totalsales_by_year]
AS
    SELECT SUM(total_due) AS total_sales ,
            DATEPART(YEAR, order_date) AS Year_of_order
    FROM orders
    GROUP BY DATEPART(YEAR, order_date)
GO
```

Uses

[dbo].[orders]

[dbo].[Login_Proc]

MS_Description

Login_procedure that takes two parameters email and password

Parameters

Name	Data Type	Max Length (Bytes)
@email	varchar(50)	50
@password	varchar(50)	50

SQL Script

```
CREATE PROC [dbo].[Login_Proc]
    @email VARCHAR(50),
    @password VARCHAR(50)
AS
    ----check if the email and password is exists
    IF(EXISTS(select * FROM customers WHERE email = @email AND password = @password))
        BEGIN
            SELECT 'successful login ' AS verify_message
        END
    ELSE
        BEGIN
            SELECT 'email or password not correct' AS login_error
        END
GO
EXEC sp_addextendedproperty N'MS_Description', N'Login_procedure that takes two
parameters email and password', 'SCHEMA', N'dbo', 'PROCEDURE', N'Login_Proc', NULL, NULL
GO
```

Uses

[dbo].[customers]

[dbo].[LOWER_prod_Name_totalsales]

SQL Script

```sql
CREATE PROC [dbo].[LOWER_prod_Name_totalsales]
AS
    SELECT TOP 10 SUM(total_due) AS Lower_total_sales,Product_name
    FROM  orders AS o INNER JOIN order_details od
    ON o.order_id = od.order_id
    INNER JOIN Products AS p
    ON od.product_id = p.Product_ID
    GROUP BY Product_name
    ORDER BY SUM(total_due) ASC
GO
```

Uses

[dbo].[order_details]
[dbo].[orders]
[dbo].[Products]

[dbo].[lowest_ten_products]

MS_Description

Show lowest 10 products

SQL Script

```
create proc [dbo].[lowest_ten_products]
as
select top 10 sum(line_total), product_name from order_details od inner join Products p
on od.product_id = p.Product_ID
group by Product_name
order by sum(line_total) asc
GO
EXEC sp_addextendedproperty N'MS_Description', N'Show lowest 10 products', 'SCHEMA',
N'dbo', 'PROCEDURE', N'lowest_ten_products', NULL, NULL
GO
```

Uses

[dbo].[order_details]
[dbo].[Products]

[dbo].[loyal_customer_rates]

SQL Script

```
create proc [dbo].[loyal_customer_rates]
as
select avg(overall_rate) as over_all_rate, avg(delivery_rate) as deliver_rate
, avg(product_quality_rate) as product_quality_rate,
avg(customer_service_rate) as customer_service_rate
from survey_question
where loyality = 'yes'
GO
```

[dbo].[MakeOrder]

MS_Description

Use Customers.customer_id ,Orders.payment_method,Orders.shipping_city to make order

Parameters

Name	Data Type	Max Length (Bytes)
@customer_id	int	4
@payment_method	varchar(50)	50
@shipping_city	varchar(50)	50

SQL Script

```sql
CREATE proc [dbo].[MakeOrder]
@customer_id int,
@payment_method varchar(50),
@shipping_city varchar(50)
AS
declare @order_id int
insert into orders(shipping_city,payment_method,customer_id, order_date)
values(@shipping_city,@payment_method,@customer_id, GETDATE())

set @order_id = SCOPE_IDENTITY()

declare @product_id int, @quantity int

insert into Order_details(quantity,Product_ID,Order_ID)
select quantity,product_id,@order_id from cart where customer_id = @customer_id

delete from cart where customer_id = @customer_id
GO
EXEC sp_addextendedproperty N'MS_Description', N'Use Customers.customer_id
,Orders.payment_method,Orders.shipping_city to make order', 'SCHEMA', N'dbo',
'PROCEDURE', N'MakeOrder', NULL, NULL
GO
```

Uses

[dbo].[cart]
[dbo].[order_details]
[dbo].[orders]

[dbo].[nonloyal_customers_rates]

MS_Description

Show the non-loyal customer rate

SQL Script

```
create proc [dbo].[nonloyal_customers_rates]
as
select avg(overall_rate) as over_all_rate, avg(delivery_rate) as deliver_rate
, avg(product_quality_rate) as product_quality_rate,
avg(customer_service_rate) as customer_service_rate
from survey_question
where loyality = 'no'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Show the non-loyal customer rate',
'SCHEMA', N'dbo', 'PROCEDURE', N'nonloyal_customers_rates', NULL, NULL
GO
```

[dbo].[order_history]

MS_Description

Use Customers.customer_id to show the previous order of him/her

Parameters

Name	Data Type	Max Length (Bytes)
@customer_id	varchar(50)	50

SQL Script

```
CREATE proc [dbo].[order_history]
@customer_id varchar(50)
as
select * from orders where customer_id = @customer_id
GO
EXEC sp_addextendedproperty N'MS_Description', N'Use Customers.customer_id to show the previous order of him/her', 'SCHEMA', N'dbo', 'PROCEDURE', N'order_history', NULL, NULL
GO
```

Uses

[dbo].[orders]

[dbo].[overall_total_sales]

MS_Description

Show total sales by sum orders.total_due

SQL Script

```
create proc [dbo].[overall_total_sales]
as
select total_sales = sum(total_due),total_taxes = sum(total_tax),
total_freight = sum(total_freight) , total_subtotal = sum(sub_total)
from orders
GO
EXEC sp_addextendedproperty N'MS_Description', N'Show total sales by sum
orders.total_due', 'SCHEMA', N'dbo', 'PROCEDURE', N'overall_total_sales', NULL, NULL
GO
```

Uses

[dbo].[orders]

[dbo].[product_history]

MS_Description

Use customers.customer_id to show the product history of him/her

Parameters

Name	Data Type	Max Length (Bytes)
@customer_id	int	4

SQL Script

```
create proc [dbo].[product_history]
@customer_id int
as
select product_name, price, quantity, order_date from
products p inner join order_details od on p.Product_ID = od.product_id
inner join orders o on od.order_id = o.order_id
where o.customer_id = @customer_id
GO
EXEC sp_addextendedproperty N'MS_Description', N'Use customers.customer_id to show the
product_history of him/her', 'SCHEMA', N'dbo', 'PROCEDURE', N'product_history', NULL,
NULL
GO
```

Uses

[dbo].[order_details]
[dbo].[orders]
[dbo].[Products]

[dbo].[products_in_e_commerce]

SQL Script

```
CREATE PROC [dbo].[products_in_e_commerce]
AS
    SELECT Category_name , Subcategory_name , Product_name , price
    FROM Category AS c INNER JOIN  subcategory AS sc
    ON c.Category_ID = sc.Category_ID
    INNER JOIN    Products AS p
    ON p.subcategory_ID = sc.Subcategory_ID
GO
```

Uses

[dbo].[Category]
[dbo].[Products]
[dbo].[subcategory]

[dbo].[products_sales_per_months]

MS_Description

Use Orders.order_date to show product sales per month

Parameters

Name	Data Type	Max Length (Bytes)
@year	int	4

SQL Script

```
CREATE proc [dbo].[products_sales_per_months]
@year int
as
select ProductName,
       ISNULL([1],0) as Jan ,
       ISNULL([2],0) as Feb,
       ISNULL([3],0) as Mar,
       ISNULL([4],0) as Apr ,
       ISNULL([5],0) as May,
       ISNULL([6],0) as Jun,
       ISNULL([7],0) as Jul,
       ISNULL([8],0) as Aug,
       ISNULL([9],0) as Sep,
       ISNULL([10],0) as Oct,
       ISNULL([11],0) +
       ISNULL([12],0)+
        ISNULL([1],0) +
       ISNULL([2],0)+
       ISNULL([3],0) +
       ISNULL([4],0) +
       ISNULL([5],0) +
       ISNULL([6],0) +
       ISNULL([7],0)+
       ISNULL([8],0) +
       ISNULL([9],0) +
       ISNULL([10],0) +
       ISNULL([11],0) +
       ISNULL([12],0) as [Total sales]

 from  ( SELECT MONTH(o.order_date) AS orderDate, p.product_name AS ProductName,
    SUM(isnull(od.line_total,0)) AS LineTotal
    FROM orders o
    INNER JOIN order_details od ON o.order_id = od.order_id
    INNER JOIN products p ON od.product_id = p.Product_ID
    WHERE YEAR(o.order_date) = @year
    GROUP BY MONTH(o.order_date), p.product_name
) as total_product_sales

PIVOT (
SUM (LineTotal)
FOR orderDate IN ([1],[2],[3],[4],[5],[6],[7],[8],[9],[10],[11],[12]))
AS pivot_productSales
GO
EXEC sp_addextendedproperty N'MS_Description', N'Use Orders.order_date to show product
sales per month ', 'SCHEMA', N'dbo', 'PROCEDURE', N'products_sales_per_months', NULL,
NULL
GO
```

Uses

[dbo].[order_details]
[dbo].[orders]
[dbo].[Products]

[dbo].[profit]

SQL Script

```sql
CREATE PROC [dbo].[profit]
AS

    SELECT Product_name ,
          SUM(quantity) AS total_product_sold_qty,
          SUM(od.quantity*p.price - od.quantity*p.cost) AS total_profit_from_each_product
    FROM Products p inner join  order_details od
    ON od.product_id=p.Product_ID
    group by Product_name

GO
```

Uses

[dbo].[order_details]
[dbo].[Products]

[dbo].[qvelall_total_sales]

SQL Script

```
create proc [dbo].[qvelall_total_sales]
as
select total_sales = sum(total_due),total_taxes = sum(total_tax),
total_freight = sum(total_freight) , total_subtotal = sum(sub_total)
from orders
GO
```

Uses

[dbo].[orders]

[dbo].[Rating_survey]

MS_Description

Use Customers.Customer_ID , Ratings.Overall_rate, Ratings.delivery_rate , Ratings.customer_service_rate,Ratings.Product_quality_rate and Ratings.Loyality

Parameters

Name	Data Type	Max Length (Bytes)
@customer_id	int	4
@overall_rate	int	4
@delivery_rate	int	4
@customer_service_rate	int	4
@loyality	varchar(50)	50
@product_quality_rate	int	4

SQL Script

```
CREATE proc [dbo].[Rating_survey]
@customer_id int,
@overall_rate int,
@delivery_rate int,
@customer_service_rate int,
@loyality varchar(50),
@product_quality_rate int
as
insert into ratings (overall_rate, delivery_rate, customer_service_rate,
product_quality_rate,loyality, customer_id)
values (@overall_rate, @delivery_rate, @customer_service_rate,@product_quality_rate,
@loyality, @customer_id)
GO
EXEC sp_addextendedproperty N'MS_Description', N'Use Customers.Customer_ID ,
Ratings.Overall_rate, Ratings.delivery_rate ,
Ratings.customer_service_rate,Ratings.Product_quality_rate and Ratings.Loyality ',
'SCHEMA', N'dbo', 'PROCEDURE', N'Rating_survey', NULL, NULL
GO
```

Uses

[dbo].[ratings]

[dbo].[registration_proc]

Parameters

Name	Data Type	Max Length (Bytes)
@fname	varchar(50)	50
@lname	varchar(50)	50
@gender	varchar(50)	50
@zip	varchar(50)	50
@country	varchar(50)	50
@city	varchar(50)	50
@age	int	4
@phone	varchar(11)	11
@email	varchar(50)	50
@password	varchar(50)	50

SQL Script

```
CREATE PROC [dbo].[registration_proc]
    @fname VARCHAR(50),
    @lname VARCHAR(50),
    @gender VARCHAR(50),
    @zip VARCHAR(50),
    @country VARCHAR(50),
    @city VARCHAR(50),
    @age INT,
    @phone VARCHAR(11),
    @email VARCHAR(50),
    @password VARCHAR(50)

AS
    declare @lastcutomerid int ---- declare variable that the last id from
(scope_identity)

    ---- check if this customer has an account or new customer
    IF(EXISTS(SELECT *  FROM customers WHERE email = @email))
        BEGIN
            SELECT 'this email is already used' AS error_messagee
        END
    ---- if the customer is new will enter all info. that are required from him
    ELSE
        BEGIN
            INSERT INTO
customers(fname,lname,age,gender,zip_code,country,city,email,password)
            VALUES(@fname,@lname,@age,@gender,@zip,@country,@city,@email,@password)
            SET @lastcutomerid = scope_identity()  --returns the last identity value in
```

```
the same scope(table that we work on)
            INSERT INTO customer_Phone(Customer_ID , Phone)
            VALUES(@lastcutomerid , @phone)
        END
GO
```

Uses

[dbo].[customer_Phone]
[dbo].[customers]

[dbo].[sales_per_age_category]

SQL Script

```
create proc [dbo].[sales_per_age_category]
as
select
case
        when age <= 20 then 'Under 20'
        when age between 21 and 30 then '21 to 30'
        when age between 31 and 40 then '31 to 40'
        when age between 41 and 50 then '41 to 50'
        when age >50 then 'over 50'
        end,
sum(total_due)
from orders o inner join customers c on c.customer_id = o.customer_id
group by c.age
GO
```

Uses

[dbo].[customers]
[dbo].[orders]

[dbo].[sales_per_city]

MS_Description

Sales per city

Parameters

Name	Data Type	Max Length (Bytes)
@city	varchar(50)	50

SQL Script

```
CREATE proc [dbo].[sales_per_city]
@city varchar(50)
as
select shipping_city, sum(total_due) as total_sales
from orders
group by shipping_city having shipping_city like '%'+@city+'%'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Sales per city', 'SCHEMA', N'dbo',
'PROCEDURE', N'sales_per_city', NULL, NULL
GO
```

Uses

[dbo].[orders]

[dbo].[sales_per_gender]

MS_Description

Sales per gender

SQL Script

```
create proc [dbo].[sales_per_gender]
as
select gender , sum(total_due)
from orders o inner join customers c on o.customer_id=c.customer_id
group by gender
GO
EXEC sp_addextendedproperty N'MS_Description', N'Sales per gender', 'SCHEMA', N'dbo',
'PROCEDURE', N'sales_per_gender', NULL, NULL
GO
```

Uses

[dbo].[customers]
[dbo].[orders]

[dbo].[search_about_product]

Parameters

Name	Data Type	Max Length (Bytes)
@product_letters	varchar(50)	50

SQL Script

```
CREATE PROC [dbo].[search_about_product]
    @product_letters VARCHAR(50)
AS
    SELECT Product_name  , price , Subcategory_name , Category_name
    FROM Products AS  p INNER JOIN  subcategory AS sc
    ON p.subcategory_ID = sc.Subcategory_ID
    INNER JOIN     Category c
    ON sc.Category_ID = c.Category_ID
    WHERE Product_name LIKE '%'+ @product_letters +'%'


select product_name from Products
GO
```

Uses

[dbo].[Category]
[dbo].[Products]
[dbo].[subcategory]

[dbo].[search_by_product]

MS_Description

Use keyword var to search the product name in the website by any letters of its name

Parameters

Name	Data Type	Max Length (Bytes)
@product_letters	varchar(50)	50

SQL Script

```
CREATE proc [dbo].[search_by_product]
@product_letters varchar(50)
as
select product_id, Product_name, price, category_name, subcategory_name
from Products p inner join subcategory s on p.subcategory_ID = s.Subcategory_ID
inner join Category c on s.Category_ID =c.Category_ID
where product_name like '%'+ @product_letters +'%'
GO
EXEC sp_addextendedproperty N'MS_Description', N'Use keyword var to search the product
name in the website by any letters of its name', 'SCHEMA', N'dbo', 'PROCEDURE',
N'search_by_product', NULL, NULL
GO
```

Uses

[dbo].[Category]
[dbo].[Products]
[dbo].[subcategory]

[dbo].[search_products]

Parameters

Name	Data Type	Max Length (Bytes)
@keyword	varchar(50)	50

SQL Script

```
CREATE proc [dbo].[search_products]
@keyword varchar(50)
as
select product_name, price from products where product_name like '%'+@keyword+'%'
GO
```

Uses

[dbo].[Products]

[dbo].[top_prod_Name_totalsales]

SQL Script

```
CREATE PROC [dbo].[top_prod_Name_totalsales]
AS
    SELECT TOP 10 SUM(total_due) AS top_total_sales,SUM(total_due) AS
Lower_total_sales,Product_name
    FROM  orders AS o INNER JOIN order_details od
    ON o.order_id = od.order_id
    INNER JOIN Products AS p
    ON od.product_id = p.Product_ID
    GROUP BY Product_name
    ORDER BY SUM(total_due) DESC
GO
```

Uses

[dbo].[order_details]
[dbo].[orders]
[dbo].[Products]

Project > . > User databases > E-Commerce > Programmability > Stored Procedures > dbo.top_ten_products

[dbo].[top_ten_products]

SQL Script

```
create proc [dbo].[top_ten_products]
as
select top 10 sum(line_total), product_name from order_details od inner join Products p
on od.product_id = p.Product_ID
group by Product_name
order by sum(line_total) desc
GO
```

Uses

[dbo].[order_details]
[dbo].[Products]

Project > . > User databases > E-Commerce > Programmability > Stored Procedures > dbo.total_price_of_products_in_cart

[dbo].[total_price_of_products_in_cart]

Parameters

Name	Data Type	Max Length (Bytes)
@customer_id	int	4

SQL Script

```
CREATE PROC [dbo].[total_price_of_products_in_cart]
    @customer_id INT
AS
    SELECT
        SUM(c.quantity*p.price) AS cart_subtotal ,
        SUM(c.quantity*p.price)*0.14 AS tax ,
        SUM(c.quantity*p.price)*0.07 AS freight,
        SUM(c.quantity*p.price)*1.21 AS total_pay
    FROM products  AS p INNER JOIN cart AS c
    ON c.product_id = p.Product_ID
    WHERE c.customer_id = @customer_id
GO
```

Uses

[dbo].[cart]
[dbo].[Products]

[dbo].[total_sale_details]

SQL Script

```
CREATE PROC [dbo].[total_sale_details]
AS
    SELECT  total_sales = SUM(total_due),
            total_taxes = SUM(total_tax),
            total_freight = SUM(total_freight),
            total_subtotal = SUM(sub_total)
    FROM orders
GO
```

Uses

[dbo].[orders]

Project > . > User databases > E-Commerce > Programmability > Stored Procedures > dbo.totalsales_each_product_by_MONTH

[dbo].[totalsales_each_product_by_MONTH]

SQL Script

```
CREATE PROC [dbo].[totalsales_each_product_by_MONTH]
AS
    SELECT  SUM(total_due) AS total_sales ,
            DATEPART(MONTH, order_date) AS Month_of_order ,
            DATENAME(MONTH, order_date) AS Month_of_order,Product_name

    FROM  orders AS o INNER JOIN order_details od
    ON o.order_id = od.order_id
    INNER JOIN Products AS p
    ON od.product_id = p.Product_ID
    GROUP BY DATEPART(MONTH, order_date),
             DATENAME(MONTH, order_date),
             Product_name
    ORDER BY SUM(total_due) DESC
GO
```

Uses

[dbo].[order_details]
[dbo].[orders]
[dbo].[Products]

[dbo].[view_all_products]

MS_Description

Show all products

SQL Script

```sql
CREATE proc [dbo].[view_all_products]
AS
select product_id, Product_name, price, category_name, subcategory_name, category_name
from Products p inner join subcategory s on p.subcategory_ID = s.Subcategory_ID
inner join Category c on s.Category_ID =c.Category_ID
GO
EXEC sp_addextendedproperty N'MS_Description', N'Show all products ', 'SCHEMA', N'dbo',
'PROCEDURE', N'view_all_products', NULL, NULL
GO
```

Uses

[dbo].[Category]
[dbo].[Products]
[dbo].[subcategory]