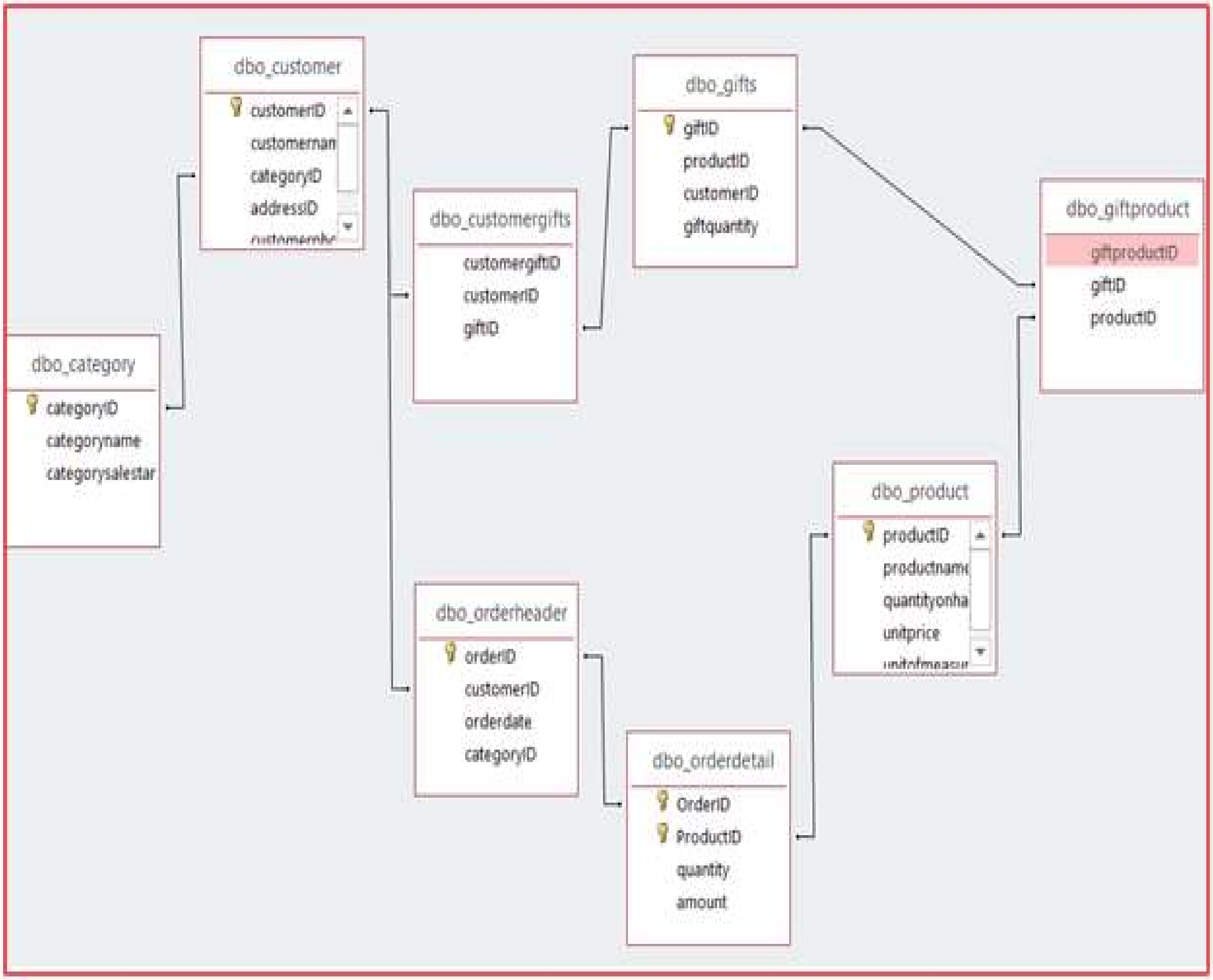
WEEK 10

PART 2 (*Continuation to Lab 04 - Project Part1*)

1) Physical database Design –



2) DDL - Scripts to Build FOURSEASONS DATABASE TABLES

```
--FOURSEASONS - TABLE CREATION (TSQL SCRIPTS)

-- Customer

CREATE TABLE [dbo].[customer](
       [customerID] [int] IDENTITY(1,1) NOT NULL,
       [customername] [varchar](50) NULL,
       [categoryID] [int] NOT NULL,
       [addressID] [int] NOT NULL,
       [customerphone] [varchar](15) NULL,
       [customermailID] [varchar](50) NULL,
 CONSTRAINT [PK_Customer] PRIMARY KEY CLUSTERED
(
       [customerID] ASC
)WITH (STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]


ALTER TABLE [dbo].[customer]  WITH CHECK ADD FOREIGN KEY([addressID])
REFERENCES [dbo].[address] ([addressID])


ALTER TABLE [dbo].[customer]  WITH CHECK ADD FOREIGN KEY([categoryID])
REFERENCES [dbo].[category] ([categoryID])


-- Category Table  CREATE/ALTER

CREATE TABLE [dbo].[category](
       [categoryID] [int] IDENTITY(1,1) NOT NULL,
       [categoryname] [varbinary](50) NOT NULL,
       [categorysalestarget] [money] NOT NULL,
 CONSTRAINT [PK_Category] PRIMARY KEY CLUSTERED
(
       [categoryID] ASC
)WITH (STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]


ALTER TABLE [dbo].[category] ADD  CONSTRAINT [DF_CategoryID_categorysalestarget]  DEFAULT
((0)) FOR [categorysalestarget]


ALTER TABLE [dbo].[category]  WITH CHECK ADD  CONSTRAINT [FK_Category_Category] FOREIGN
KEY([categoryID])
REFERENCES [dbo].[category] ([categoryID])


ALTER TABLE [dbo].[category] CHECK CONSTRAINT [FK_Category_Category]


-- PRODUCT Table  create/Alter
/****** Object:  Table [dbo].[product]    Script Date: 10/1/2018 3:31:37 AM ******/
CREATE TABLE [dbo].[product](
       [productID] [int] IDENTITY(1,1) NOT NULL,
       [productname] [varchar](40) NOT NULL,
       [quantityonhand] [int] NOT NULL,
       [unitprice] [money] NOT NULL,
       [unitofmeasure] [char](10) NULL,
 CONSTRAINT [PK_Product] PRIMARY KEY CLUSTERED
(
       [productID] ASC
)WITH (STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]


ALTER TABLE [dbo].[product] ADD  CONSTRAINT [DF_Product_quantityonhand]  DEFAULT ((0))
FOR [quantityonhand]


ALTER TABLE [dbo].[product] ADD  CONSTRAINT [DF_Product_unitprice]  DEFAULT ((0.00)) FOR
[unitprice]


-- GIFT TABLE CREATE/ALTER

CREATE TABLE [dbo].[gifts](
       [giftID] [int] IDENTITY(1,1) NOT NULL,
       [productID] [int] NOT NULL,
       [customerID] [int] NOT NULL,
       [giftquantity] [int] NOT NULL,
 CONSTRAINT [PK_gifts] PRIMARY KEY CLUSTERED
(
       [giftID] ASC
)WITH (STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]


ALTER TABLE [dbo].[gifts] ADD  CONSTRAINT [DF_gifts_giftquantity]  DEFAULT ((0)) FOR
[giftquantity]


-- CUSTOMERGIFT INTERMEDIATORY ( LINKS TO CUSTOMER & GIFTS )

CREATE TABLE [dbo].[customergifts](
       [customergiftID] [int] IDENTITY(1,1) NOT NULL,
       [customerID] [int] NOT NULL,
       [giftID] [int] NOT NULL
) ON [PRIMARY]


ALTER TABLE [dbo].[customergifts]  WITH CHECK ADD FOREIGN KEY([giftID])
REFERENCES [dbo].[gifts] ([giftID])


-- PRODUCTGIFT INTERMEDIATORY (LINKS TO PRODUCT & GIFTS)
CREATE TABLE [dbo].[giftproduct](
       [giftproductID] [int] IDENTITY(1,1) NOT NULL,
       [giftID] [int] NOT NULL,
       [productID] [int] NOT NULL
) ON [PRIMARY]
```

```sql
ALTER TABLE [dbo].[giftproduct]  WITH CHECK ADD FOREIGN KEY([giftID])
REFERENCES [dbo].[gifts] ([giftID])


ALTER TABLE [dbo].[giftproduct]  WITH CHECK ADD FOREIGN KEY([productID])
REFERENCES [dbo].[product] ([productID])

-- ORDER HEADER



CREATE TABLE [dbo].[orderheader](
       [orderID] [int] IDENTITY(1,1) NOT NULL,
       [customerID] [int] NOT NULL,
       [orderdate] [datetime] NOT NULL,
       [categoryID] [int] NOT NULL,
 CONSTRAINT [PK_orderheader] PRIMARY KEY CLUSTERED
(
       [orderID] ASC
)WITH (STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]


-- ORDER DETAIL (LINEITEMS)
/****** Object:  Table [dbo].[orderdetail]    Script Date: 10/1/2018 3:37:56 AM ******/
CREATE TABLE [dbo].[orderdetail](
       [OrderID] [int] NOT NULL,
       [ProductID] [int] NOT NULL,
       [quantity] [int] NOT NULL,
       [amount] [money] NOT NULL,
 CONSTRAINT [PK_orderdetail] PRIMARY KEY CLUSTERED
(
       [OrderID] ASC,
       [ProductID] ASC
)WITH (STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
GO

ALTER TABLE [dbo].[orderdetail] ADD  CONSTRAINT [DF_orderdetail_quantity]  DEFAULT ((1))
FOR [quantity]
GO

ALTER TABLE [dbo].[orderdetail]  WITH CHECK ADD FOREIGN KEY([OrderID])
REFERENCES [dbo].[orderheader] ([orderID])
GO

ALTER TABLE [dbo].[orderdetail]  WITH CHECK ADD FOREIGN KEY([ProductID])
REFERENCES [dbo].[product] ([productID])
GO
```

3) INSERTS (Just adding simple insert for example)

```sql
INSERT INTO [dbo].[customer]
           ([customername]
           ,[categoryID]
           ,[addressID]
           ,[customerphone]
           ,[customermailID])
     VALUES
           ( 'Handyman Quilts and accessories' ,
                 3,
                 6,
                 '972-812-9122',
                 'info@hqa.com'
               )
GO
```

4) VIEWS

```
-- CUSTOMER CONTACT - CUSTOMER AND ADDRESS TABLE JOINED TO CREATE A VIRTUAL VIEW.

CREATE VIEW [dbo].[customercontact]
AS
SELECT        dbo.customer.customerID,  dbo.customer.customername,
                  address.address1, address.address2, address.city, address.state,
category.categoryname
FROM            dbo.customer
                  INNER JOIN
                         dbo.address ON customer.addressID = address.addressID
                           INNER JOIN
                         dbo.category ON customer.categoryID = category.categoryID
GO

-- TO VIEW CATEGORY WISE ORDERS
CREATE VIEW [dbo].[categorysales]
AS
SELECT         dbo.category.categoryID, dbo.category.categoryname,
SUM(dbo.orderdetail.amount) AS Sales$
FROM           dbo.orderdetail
             INNER JOIN
               dbo.orderheader ON dbo.orderdetail.OrderID = dbo.orderheader.orderID
                      RIGHT OUTER JOIN
               dbo.category ON dbo.orderheader.categoryID = dbo.category.categoryID
GROUP BY dbo.category.categoryID, dbo.category.categoryname
GO


-- CUSTOMER ORDERS W/CUSTOMER & PRODUCT DETAILS
-- View will generate list of all customer orders with customer details, product details
and sales amount
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE VIEW [dbo].[customerorder]
AS
SELECT        dbo.orderheader.orderID, dbo.orderheader.orderdate,
              dbo.customer.customername, dbo.customer.categoryID,
dbo.customer.customerphone, dbo.customer.customermailID,
                       dbo.product.productname,
              dbo.orderdetail.quantity, dbo.orderdetail.amount
```

```
FROM            dbo.orderdetail
                INNER JOIN
                         dbo.orderheader ON dbo.orderdetail.OrderID = 
dbo.orderheader.orderID
                            INNER JOIN
                         dbo.product ON dbo.orderdetail.ProductID = dbo.product.productID
                            INNER JOIN
                         dbo.customer ON dbo.customer.customerID = 
dbo.orderheader.customerID
GO
```

5) STORED PROCEDURES

```
-- =============================================
-- Author:      <Subbu KANDHASWAMY>
-- Create Date: <09/29/2018>
-- Description: This Stored procedure will help update the product inventory by accepting 
ProductID and the Quantity received fresh to the store.
Update will be performed by adding the new quantity to the existing quantity on Hand.
-- =============================================
CREATE PROCEDURE UPDATEPRODQTY (@productID int, @addqty int)
AS
BEGIN
       DECLARE @updateqty Int
       SET @updateqty = @addqty*1
    -- Insert statements for procedure here
    Update dbo.product set quantityonhand = quantityonhand+@updateqty
       IF @@ERROR <> 0
       BEGIN
              ROLLBACK
              RETURN
       END
END
GO

-- =============================================
-- Author:      <Author, Subbu Kadhaswamy>
-- Create Date: <09/29/2018>
-- Description: <Stored Procedure will generate Sales made by each customer using the 
Customer Table, Order header and Order Detail table links>
-- =============================================
ALTER PROCEDURE SALESBYCUSTOMER
AS
BEGIN
       SELECT         dbo.customer.customername, dbo.customer.customerphone, 
SUM(dbo.orderdetail.amount) AS SALES_$
FROM            dbo.orderdetail INNER JOIN
                         dbo.orderheader ON dbo.orderdetail.OrderID = 
dbo.orderheader.orderID INNER JOIN
                         dbo.product ON dbo.orderdetail.ProductID = dbo.product.productID 
INNER JOIN
                         dbo.customer ON dbo.customer.customerID = 
dbo.orderheader.customerID
            GROUP BY dbo.customer.customername, dbo.customer.customerphone
END
GO
```

6) FUNCTION

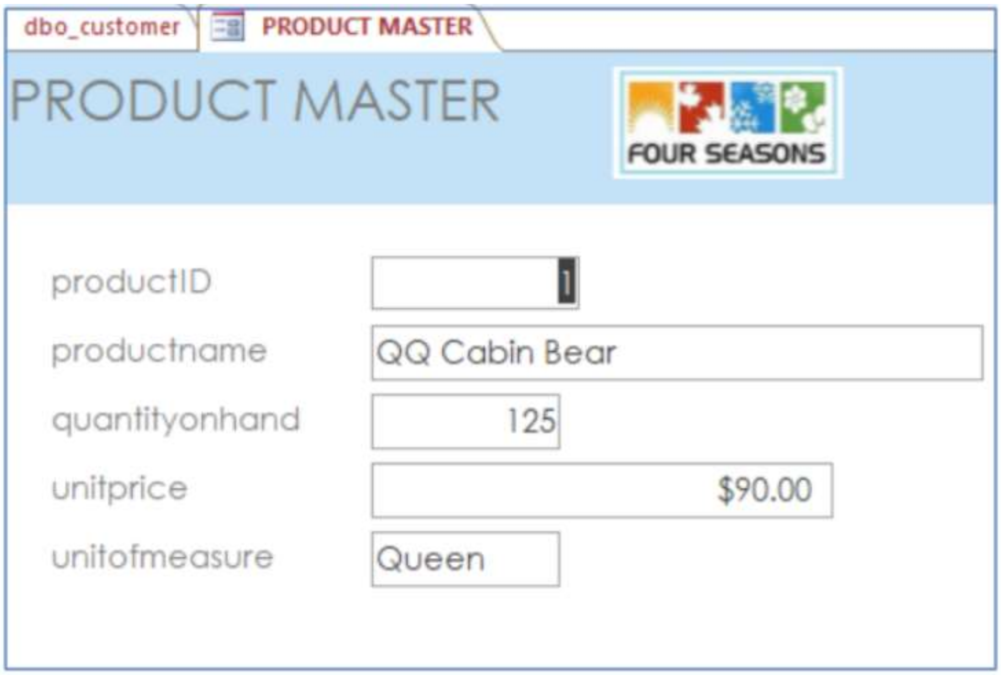
```
-- =============================================
-- Author:      <Subbu KANDHASWAMY>
-- Create Date: <09/29/2018>
-- Description:  <To run week based sales and other week based tasks, this function will
return the week number of any date. >

CREATE FUNCTION dbo.FSISalesWeekNumber (@DATE datetime)
RETURNS int
WITH EXECUTE AS CALLER
AS
BEGIN
       DECLARE @FSISalesWeekNumber int;
       SET @FSISalesWeekNumber= DATEPART(wk,@DATE)+1
       -DATEPART(wk,CAST(DATEPART(yy,@DATE) as CHAR(4))+'0104');
--Special cases: Jan 1-3 may belong to the previous year
       IF (@FSISalesWeekNumber=0)
              SET @FSISalesWeekNumber=dbo.FSISalesWeekNumber(CAST(DATEPART(yy,@DATE)-1
              AS CHAR(4))+'12'+ CAST(24+DATEPART(DAY,@DATE) AS CHAR(2)))+1;
--Special case: Dec 29-31 may belong to the next year
       IF ((DATEPART(mm,@DATE)=12) AND
              ((DATEPART(dd,@DATE)-DATEPART(dw,@DATE))>= 28))
              SET @FSISalesWeekNumber=1;
RETURN(@FSISalesWeekNumber);
END;
GO
```

Use: `SELECT *, dbo.FSISalesWeekNumber(orderdate) AS ‘Sale Order Week' from orderheader`

Application – MS-Access 2016

7) FORMS

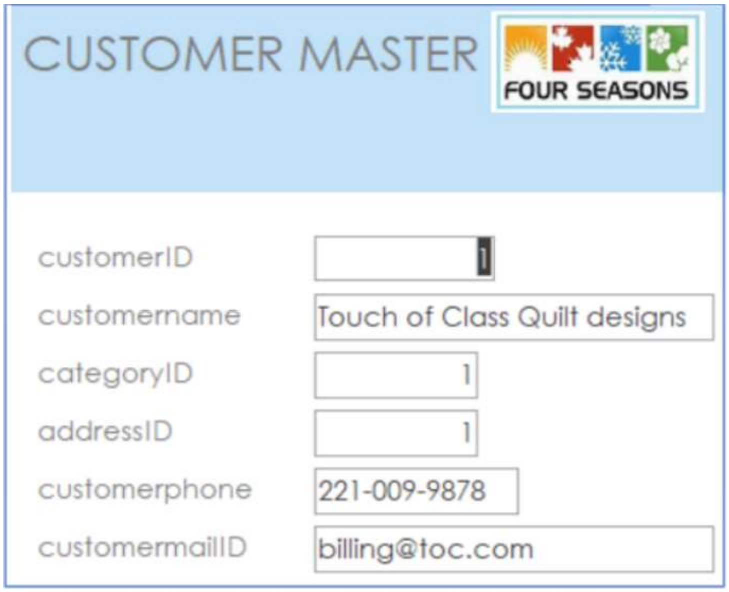
Product



PRODUCT MASTER

productID	1
productname	QQ Cabin Bear
quantityonhand	125
unitprice	$90.00
unitofmeasure	Queen

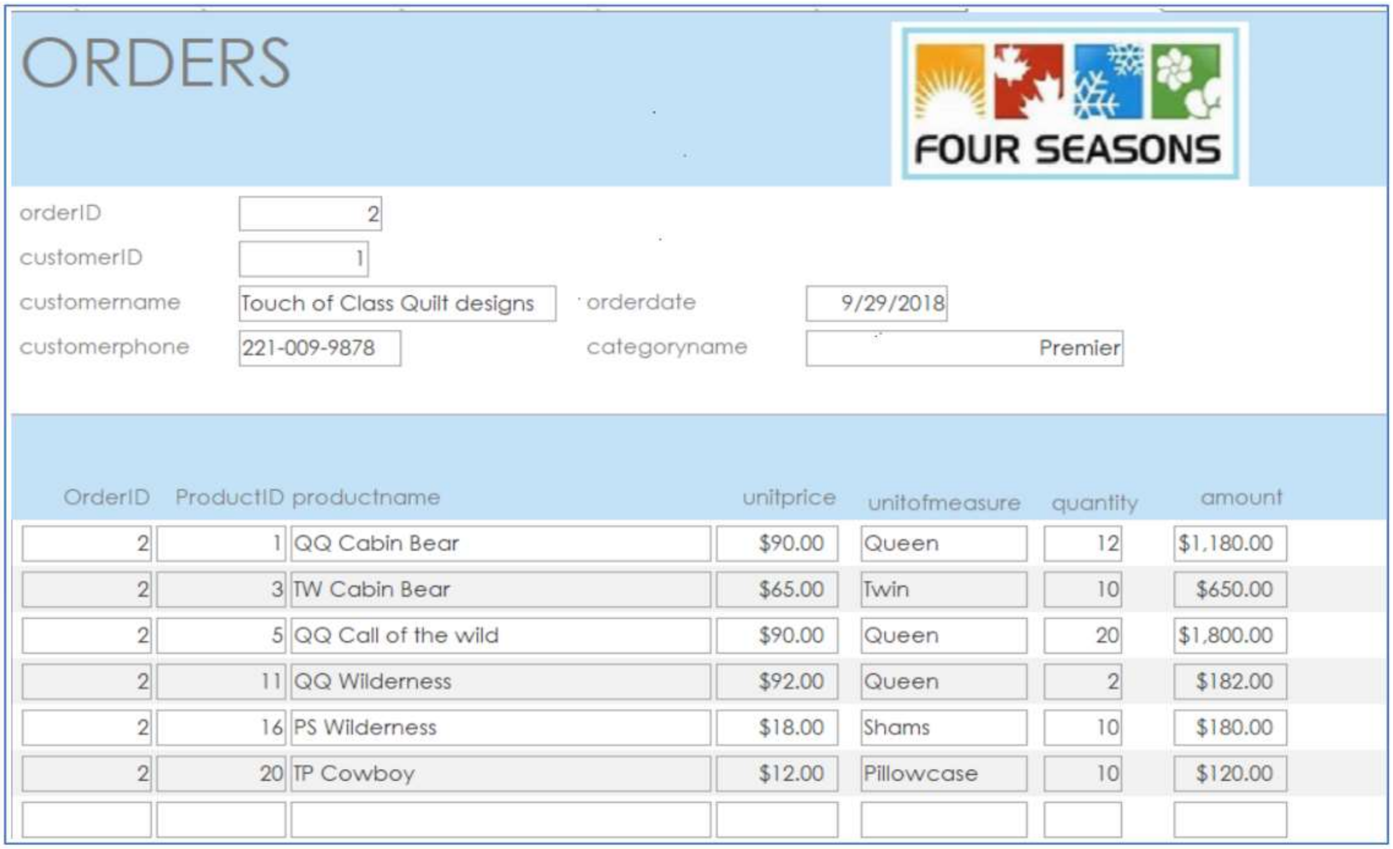
Customer



CUSTOMER MASTER

customerID	1
customername	Touch of Class Quilt designs
categoryID	1
addressID	1
customerphone	221-009-9878
customermailID	billing@toc.com

Orders Entry Form :



ORDERS

orderID 2

customerID 1

customername Touch of Class Quilt designs

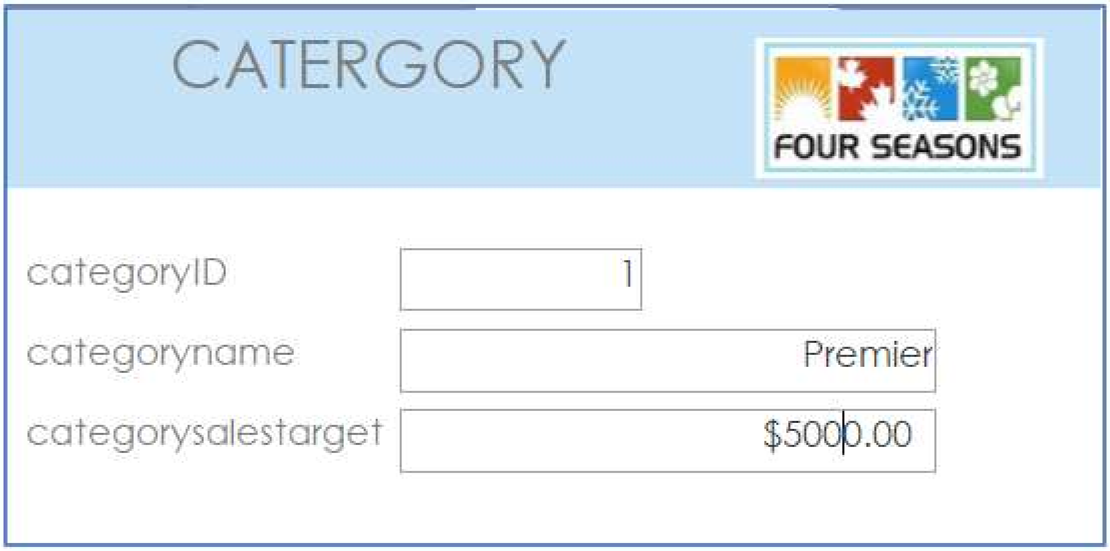
orderdate 9/29/2018

customerphone 221-009-9878

categoryname Premier

OrderID	ProductID	productname	unitprice	unitofmeasure	quantity	amount
2	1	QQ Cabin Bear	$90.00	Queen	12	$1,180.00
2	3	TW Cabin Bear	$65.00	Twin	10	$650.00
2	5	QQ Call of the wild	$90.00	Queen	20	$1,800.00
2	11	QQ Wilderness	$92.00	Queen	2	$182.00
2	16	PS Wilderness	$18.00	Shams	10	$180.00
2	20	TP Cowboy	$12.00	Pillowcase	10	$120.00

Category Master



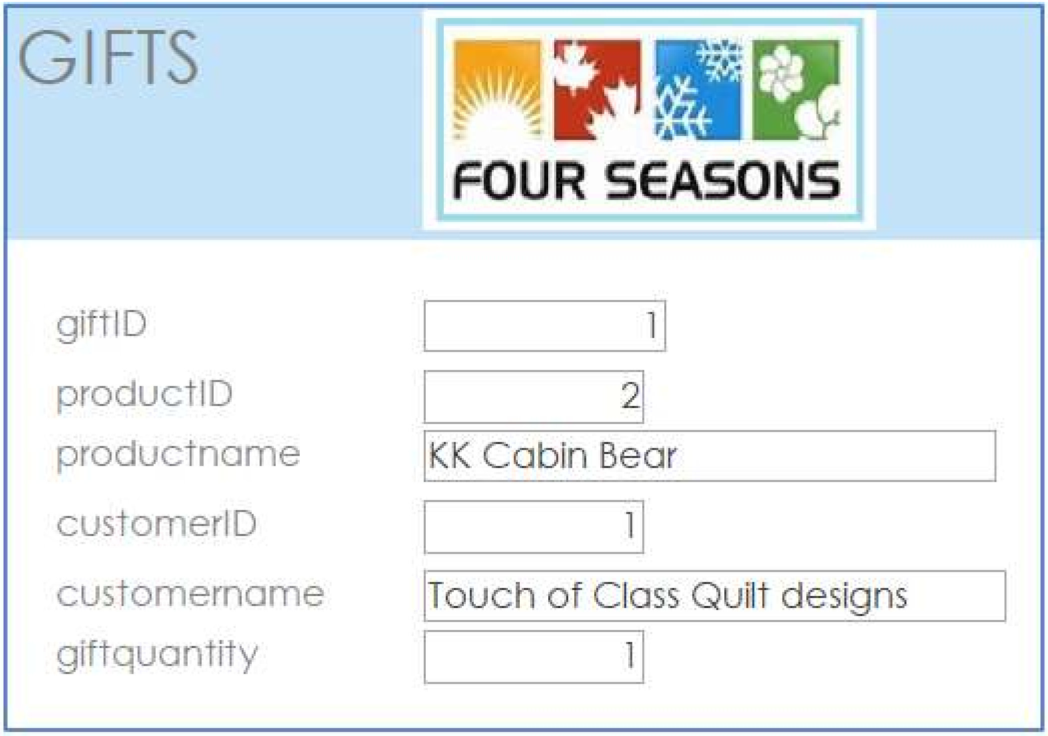
CATERGORY

categoryID 1

categoryname Premier

categorysalestarget $5000.00

Gifts

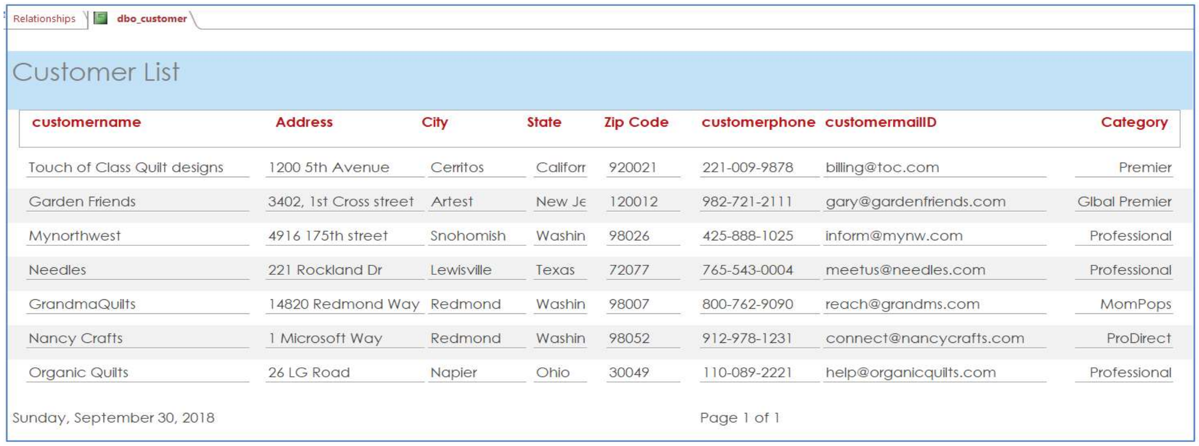


GIFTS

giftID	1
productID	2
productname	KK Cabin Bear
customerID	1
customername	Touch of Class Quilt designs
giftquantity	1

8) REPORTS :

List of all Customers with Customer Category (Link – Customer, Address, & Category Master tables.



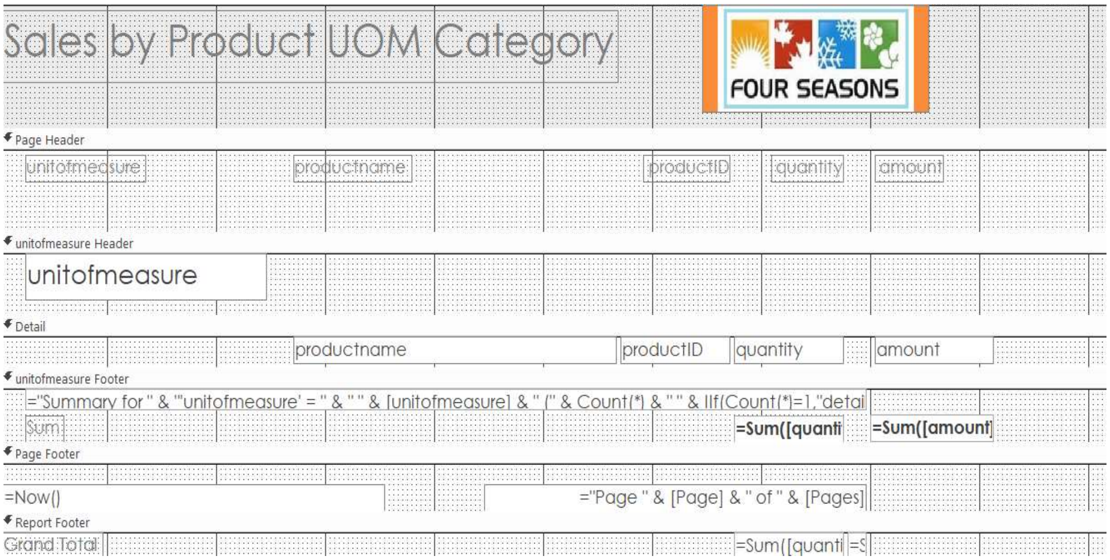
Customer List

customername	Address	City	State	Zip Code	customerphone	customermailID	Category
Touch of Class Quilt designs	1200 5th Avenue	Cerritos	Califorr	920021	221-009-9878	billing@toc.com	Premier
Garden Friends	3402, 1st Cross street	Artest	New Je	120012	982-721-2111	gary@gardenfriends.com	Glbal Premier
Mynorthwest	4916 175th street	Snohomish	Washin	98026	425-888-1025	inform@mynw.com	Professional
Needles	221 Rockland Dr	Lewisville	Texas	72077	765-543-0004	meetus@needles.com	Professional
GrandmaQuilts	14820 Redmond Way	Redmond	Washin	98007	800-762-9090	reach@grandms.com	MomPops
Nancy Crafts	1 Microsoft Way	Redmond	Washin	98052	912-978-1231	connect@nancycrafts.com	ProDirect
Organic Quilts	26 LG Road	Napier	Ohio	30049	110-089-2221	help@organicquilts.com	Professional

Sunday, September 30, 2018 Page 1 of 1

Product Sales by UOM

<Design & Output>



Sales by Product UOM Category

FOUR SEASONS

Page Header

unitofmeasure productname productID quantity amount

unitofmeasure Header

unitofmeasure

Detail

productname productID quantity amount

unitofmeasure Footer

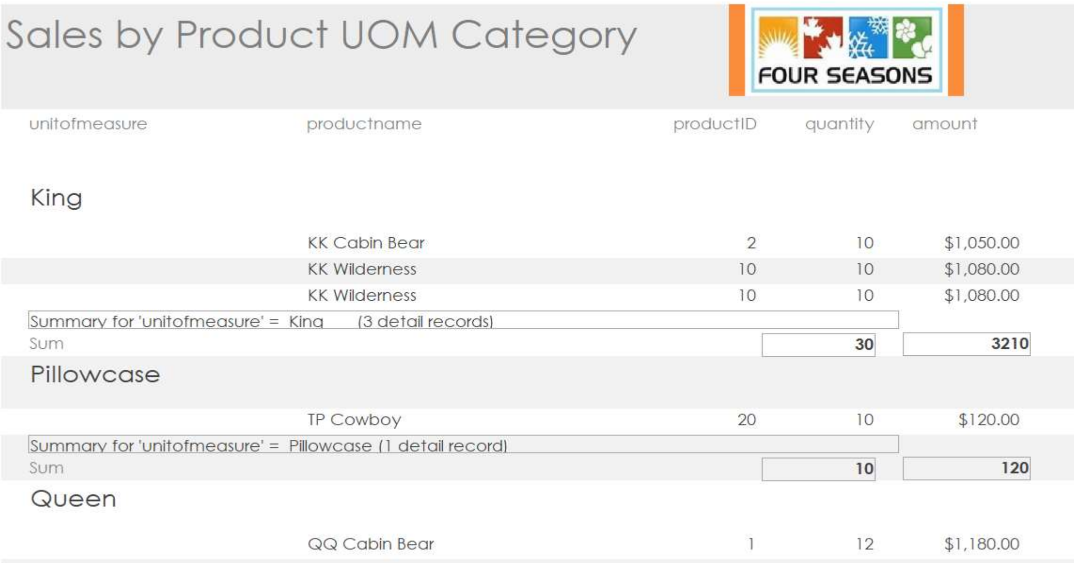
```
="Summary for " & "'unitofmeasure' = " & " " & [unitofmeasure] & " (" & Count(*) & " " & IIf(Count(*)=1,"detai
Sum    =Sum([quanti    =Sum([amount]
```

Page Footer

```
=Now()    ="Page " & [Page] & " of " & [Pages]
```

Report Footer

```
Grand Total    =Sum([quanti    =S
```



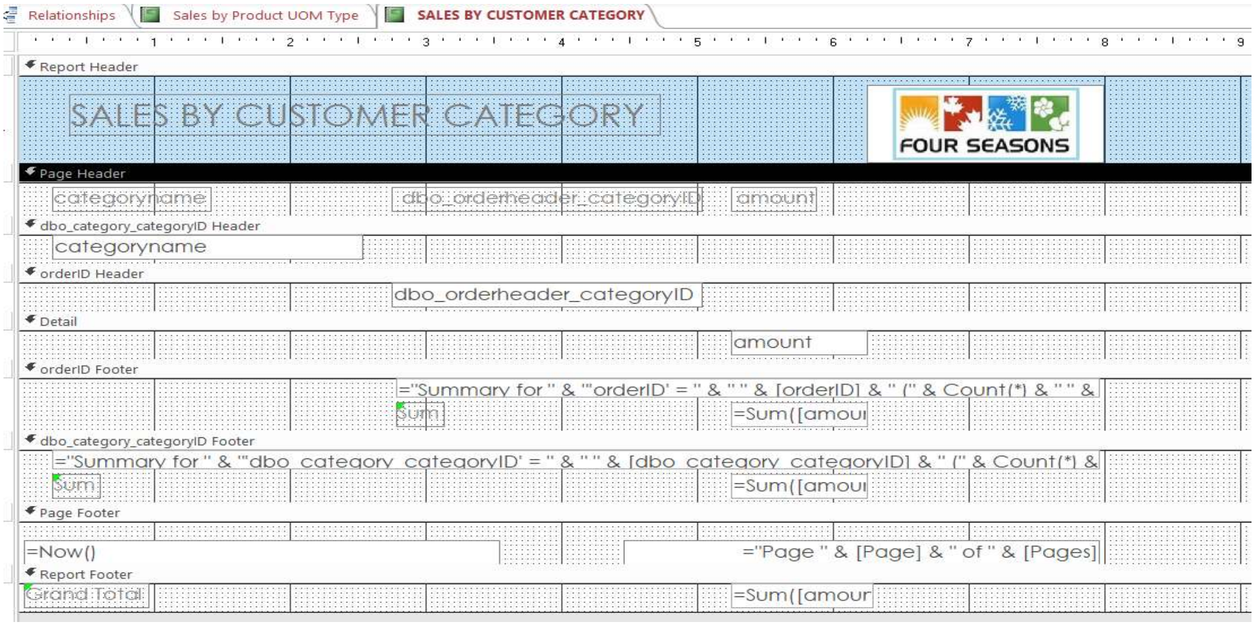
Sales by Product UOM Category

FOUR SEASONS

unitofmeasure	productname	productID	quantity	amount
King				
	KK Cabin Bear	2	10	$1,050.00
	KK Wilderness	10	10	$1,080.00
	KK Wilderness	10	10	$1,080.00
Summary for 'unitofmeasure' = King (3 detail records)				
Sum			30	3210
Pillowcase				
	TP Cowboy	20	10	$120.00
Summary for 'unitofmeasure' = Pillowcase (1 detail record)				
Sum			10	120
Queen				
	QQ Cabin Bear	1	12	$1,180.00

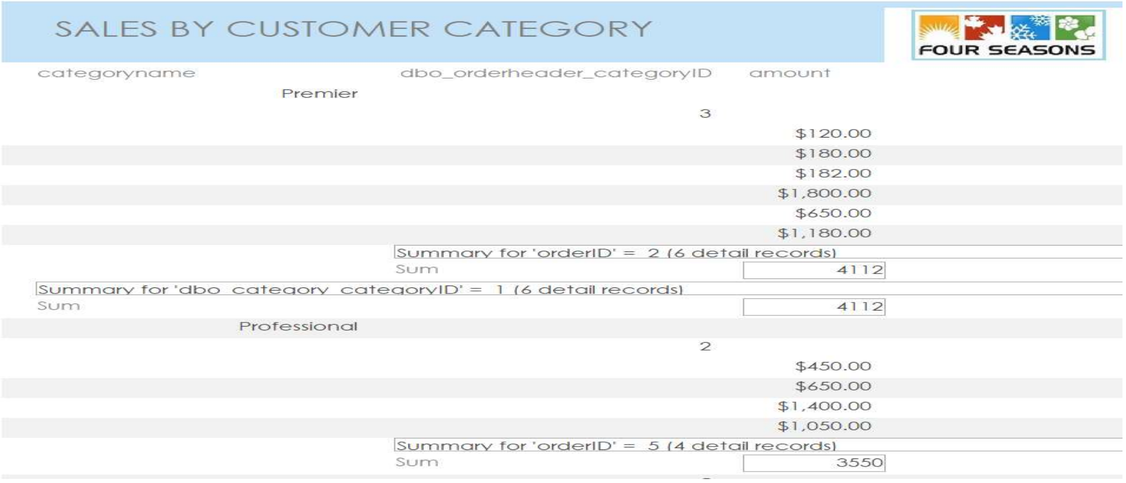
Sales by Customer Category (Premier /Professional/Pro-Direct/Mom-Pops/Global Premier

<Design>



```
Report Header
    SALES BY CUSTOMER CATEGORY
Page Header
    categoryname    dbo_orderheader_categoryID    amount
dbo_category_categoryID Header
    categoryname
orderID Header
    dbo_orderheader_categoryID
Detail
    amount
orderID Footer
    ="Summary for " & "'orderID' = " & " " & [orderID] & " (" & Count(*) & " " &
    Sum    =Sum([amou
dbo_category_categoryID Footer
    ="Summary for " & "'dbo category categoryID' = " & " " & [dbo category categoryID] & " (" & Count(*) &
    Sum    =Sum([amou
Page Footer
    =Now()    ="Page " & [Page] & " of " & [Pages]
Report Footer
    Grand Total    =Sum([amour
```

<Ouput with first set of sample records>



```
SALES BY CUSTOMER CATEGORY
categoryname              dbo_orderheader_categoryID   amount
                 Premier
                                                3
                                                          $120.00
                                                          $180.00
                                                          $182.00
                                                        $1,800.00
                                                          $650.00
                                                        $1,180.00
                          Summary for 'orderID' =  2 (6 detail records)
                          Sum                                4112
Summary for 'dbo category categoryID' =  1 (6 detail records)
Sum                                                          4112
               Professional
                                                2
                                                          $450.00
                                                          $650.00
                                                        $1,400.00
                                                        $1,050.00
                          Summary for 'orderID' =  5 (4 detail records)
                          Sum                                3550
```

9) Other Reports (Didn't add screenshots – considering the # of increasing pages)

- Top & Low Selling Products (Improve Awareness)
- Top & Low Performing Customers (Improve Sales Tactics)
- Most appreciated customer (Recognize by Gifting)
- Sales by Region (For Targetting Trade-Shows)

10) Summary :

Attached Physical database Design in continuation to Normalized structure in Lab 04 – Project.

Creation of Tables using DDL and Modification of records (DML) are done using TSQL. TSQL Scripts are attached for the Creation.

Scripts used to Create Views/ Function /Stored Procedures are attached.

For Implementation, I used Ms-Access as Application as front-end.

Attached forms screenshots for - Manage Customer/Category / Products / Gifts Master tables . And Order Entry (Order/Line items) for customers.

Also, Quality reports were built and generated using MS-Access Reports – Attached screen shot of some of the reports built for this project.

11) Conclusion :

What assumptions did you have at the start of your project that changed by the end? Think in terms of both your own problem domain as well as your knowledge of the process.

- **Initial assumptions on the project was bit complex and the intention was to help S&D process for a real-time business, I had to simplify considering time crunch and deadline.**
- **Also my initial plan was to use Visual Studio as application front end and PowerBI OR SQL Server Reporting Service for Reports. But settled for Ms-Access which is extremely simple to build forms , reports and to manage the database via ODBC connection.**

The next time you do this, what will be different?

- **Target for a real time business requirement, and design /develop using latest technologies.**

Regardless of whether you go through these steps again, how do you think it will inform your approach to data as an information professional?

- **The exercise to define and extract information from the raw data available change for each requirement, and it's a constant evolution of process. Each time, the approach will vary but still following some golden rules of modeling.**

-END OF PROJECT-