Database Systems Project Step 3

Kesgin Kuafor Database

Members:

150116836 – Enes Nadir SANLI as Representative
150116844 – Abdurrahman ŞAKAR
150113852 – Doğukan DENİZ

Content

Project Description

Kesgin Kuafor is a barber store. There are a lot of people that come here. But the problem here is that they didn't keep any data of their customers up to now. Reservation situations are made by phone and they are written on the papers. Also, they sell care products like wax, hair spray and shampoo etc. for increasing their incomes. All these operations are kept on the papers or receipt so they can't trace monthly customer count, income or data at a specific month easily. We have a project for them. We will implement a database for all their records. In our database project we have Customer, Staff, Owner, Barber, Salary, Receipt, CategoryChoice, Category, ProductChoice, Reservation, Storage, InChargeOfStorage and Product tables. After the implemented database we will present this project to the client within a web application. Web application will have a useful and simple interface so our client can use on all their work instead of paper.

Scope

While we were planning our database with its requirements and data, we determined most of the required field. We have included what we planned but during the process of completing our project we had some additional requirements. Thinking about the project from the beginning, we were supported about the project by the owner of company during the processes. But for some parts like salaries of owners and barbers we created data different from the real one.

Data Requirements Analysis

We provided unreal data for some relations like Reservation, Receipt, Customer and Staff tables. Because of the business rule we also couldn't provide real Salary data for the table. Current state of the database can be enough for demos with client.

Tables with Their Definitions

There is a file named "DatabaseTables.jpg" in the project file, It has tables and information about attributes, Primary/Foreign Key, datatypes, unique and etc. Here we add definitions of tables.

Staff: This table keeps information about the people of the company and has two types one is Owner and other one is Barber.

Salary: This table keeps owner's determined salaries for the barber and also himself.

Customer: This table keeps information about customers that come to the barber store.

Reservation: This table keeps information about reservation of customers done.

Receipt: This table keeps information about the prices of bought products and done reservations.

Storage: This table keeps information about the storage that products kept.

InChargeOfStorage: Each storage has to have a responsible person, and this table keeps information about that person.

Product: This table holds information about product to be sold.

Category: This table holds information about categories that the customer can choose for their needs.

ProductChoice: This table keeps the products that written a specific receipt.

CategoryChoice: This table keeps the categories that written a specific receipt.

Views

1- **Name:** view_ProductTypeShampoo

Def: This view lists only the product type shampoo.

SQLQuery4.sql - TB...01-1292\enadi (55))*

```sql
Create View view_ProductTypeShampoo As
Select p.productID, p.productName
From Product p
Where p.productType='Şampuan'

Select * From view_ProductTypeShampoo
```

125 %

Results | Messages

	productID	productName
1	513	Kuaf Keratin Şampuan
2	514	S Düzleştirici Şampuan
3	515	H&S S Dökülmesine Karşı
4	516	Sebamed S Dökülmesi
5	517	Gliss Şampuan
6	518	Gliss Besleyici Şampuan
7	519	Kuaf Tuzsuz Şamp.
8	520	Yüz Yıkama Köpüğü

2- **Name:** view_GetTodayReservation

Def: This view lists the reservations of the day

SQLQuery4.sql - TB...01-1292\enadi (55))*

```sql
Create View view_GetTodayReservation As
Select c.firstName + ' ' + c.lastName FullName, r.reservationTime
From Reservation r Inner Join Receipt rc
    On r.receiptID=rc.receiptID
    Inner Join Customer c
        On rc.customerID=c.customerID
Where Cast(r.reservationTime as date)=Cast(GETDATE() as date)

Select * From view_GetTodayReservation
```

125 %

Results | Messages

	FullName	reservationTime
1	Süleyman Yozgat	2021-01-01 22:30:...
2	Ersin Öztekin	2021-01-01 10:10:...
3	Erman Sarı	2021-01-01 10:30:...

3- **Name:** view_GetCategories

Def: This view lists all categories from category table

SQLQuery4.sql - TB...01-1292\enadi (55))*

```sql
Create View view_GetCategories As
Select *
From Category c

Select * From view_GetCategories
```

125 %

Results | Messages

	categoryID	categoryName	price
1	100	Saç	25
2	101	Saç-Yıkama	30
3	102	Sakal	15
4	103	Saç-Sakal	35
5	104	Saç-Sakal-Yıka...	40
6	105	Ense Düzeltme	10
7	106	Çocuk Saç	15
8	107	Saç Yıkama	10
9	108	Masaj	15
10	109	Saç Boyama	100
11	110	Bıyık Boyama	25

4- **Name:** view_GetMonthlyIncome

Def: This view lists income of each month

SQLQuery4.sql - TB...01-1292\enadi (55))*

```sql
Create View view_GetMonthlyIncome As
Select Datename(Month, r.date) _Month, sum(r.totalPrice) Income
From Receipt r
Group By Datename(Month, r.date)

Select * From view_GetMonthlyIncome
```

125 %

Results | Messages

	_Month	Income
1	Decem...	1655
2	January	315

5- **Name:** view_GetProducts

Def: This view list products with their id, name and price

SQLQuery4.sql - TB...01-1292\enadi (55))*

```sql
Create View view_GetProducts As
Select p.productID, p.productName, p.price
From Product p

Select * From view_GetProducts
```

125 %

Results | Messages

	productID	productName	price
1	500	Hobby Jöle(300 ml)	30
2	501	Egos Jöle(300 ml)	27
3	502	Hobby Jöle(500 ml)	48
4	503	L'Oreal Paris Jöle(150)	40
5	504	Saç Şekillendirici	44
6	505	Saç Bakım Köpüğü	25
7	506	Bitkisel Şekillendirici	45
8	507	Oxiplexx Protein Sütü	65
9	508	Keratin S Bakım Sütü	55

Triggers

1- Name: TrgSetReceiptDate

Def: This trigger work after insertion to receipt table and sets the date of receipt table.

SQLQuery4.sql - TB...01-1292\enadi (55))*

```sql
Create Trigger TrgSetReceiptDate
    On  Receipt
    After Insert
As
Begin
     Set Nocount On;
     update r
     set date=Cast(GETDATE() as smalldatetime)
     From Receipt r inner join inserted i
         on r.receiptID=i.receiptID
End
```

2- Name: TrgDecreaseSoldProductCount

Def: When there is an insertion or deletion on the productChoice table this trigger adds or decreases the quantity from product table.

SQLQuery4.sql - TB...01-1292\enadi (55))*

```sql
Create Trigger TrgDecreaseSoldProductCount
    On  ProductChoice
    After Insert, Delete, Update
As
Begin
     Set Nocount On;
     Update p
     Set quantity= p.quantity-i.quantity
     From Product p Inner Join inserted i
         On p.productID=i.productID

     Update p
     Set quantity= p.quantity+d.quantity
     From Product p Inner Join deleted d
         On p.productID=d.productID
End
```

3- Name: TrgSetStaffSubtypeContent

Def: This trigger inserts the new staff record to the Owner or Barber table according to its staffType.

SQLQuery4.sql - TB...01-1292\enadi (55))*

```sql
Create Trigger TrgSetStaffSubtypeContent
    On  Staff
    After Insert
 As
Begin
     Declare @newSsn smallint, @staffType char(1)
     Set Nocount On;
     If Exists(Select * From inserted)
     Begin
         set @newSsn = (Select i.SSN From inserted i)
         set @staffType = (Select i.staffType From inserted i)

         If @staffType='o'
         Begin
             Insert Into Ownerr(OSSN) values(@newSsn)
         End
         Else If @staffType='b'
         Begin
             Insert Into Barber(BSSN) values(@newSsn)
         End
     End
 End
```

Stored Procedures

1- Name: Sp_CreateReceipt

Def: This SP creates receipt with given customerID as parameter.

Before:

35	39	1000	2021-01-01 14:00:...	115
36	40	1001	2021-01-01 14:24:...	200

Query executed successfully.

After:

36	40	1001	2021-01-01 14:24:...	200
37	41	1001	2021-01-01 22:16:...	0

Query executed successfully.

SQLQuery13.sql - T...01-1292\enadi (56))* | SQLQuery4.sql - TB...01-1292\enadi (55))*

```sql
CREATE PROCEDURE Sp_CreateReceipt
    @customerID smallint,
    @receiptID smallint Output
AS
BEGIN
    Set Nocount On
    Insert Into Receipt(customerID, totalPrice) Values(@customerID, 0)
    Select @receiptID=@@Identity
    Return
END
GO
Declare @receiptID smallint
Exec Sp_CreateReceipt 1001, @receiptID = @receiptID Output;
```

2- Name: Sp_CreateProductReceipt

Def: This SP adds products to ProductChoice table and updates receipt totalPrice of given receiptID

Before:

8	39	505	1	1	25
9	39	511	1	1	30

After:

36	40	1001	2021-01-01 14:24:...	200
37	41	1001	2021-01-01 22:16:...	22

SQLQuery13.sql - T...01-1292\enadi (56))* SQLQuery4.sql - TB...01-1292\enadi (55))*

```sql
Create Proc Sp_CreateProductReceipt
    @receiptID smallint, @newProductID smallint, @newStorageID smallint, @price smallint
As
Begin
    Set Nocount On
    If Exists(Select * From ProductChoice pc
              where pc.receiptID=@receiptID and pc.productID=@newProductID and pc.storageID=@newStorageID)
    Begin
        Update pc
        Set pc.quantity = pc.quantity + 1
        From ProductChoice pc
        Where pc.receiptID=@receiptID and pc.productID=@newProductID and pc.storageID=@newStorageID
    End
    Else
    Begin
        Insert Into ProductChoice Values(@receiptID, @newProductID, @newStorageID, 1, @price)
    End
    Update r
        Set r.totalPrice = r.totalPrice + @price
        From Receipt r
        Where r.receiptID=@receiptID
End
GO

Exec Sp_CreateProductReceipt 41, 500, 1, 22
Select * From Receipt
```

3- Name: Sp_CreateReservationReceipt

Def: This SP adds categories to CategoryChoice table and updates receipt totalPrice of given receiptID

Before:

36	40	1001	2021-01-01 14:24:...	200
37	41	1001	2021-01-01 22:16:...	22

After:

36	40	1001	2021-01-01 14:24:...	200
37	41	1001	2021-01-01 22:16:...	44

SQLQuery13.sql - T...01-1292\enadi (56))* | SQLQuery4.sql - TB...01-1292\enadi (55))*

```sql
Create Proc Sp_CreateReservationReceipt
    @receiptID smallint, @newCategoryID smallint, @price smallint
As
Begin
    Set Nocount On

    If Exists(Select * From CategoryChoice cc
              where cc.receiptID=@receiptID and cc.categoryID=@newCategoryID)
    Begin
        Update cc
        Set cc.quantity = cc.quantity + 1
        From CategoryChoice cc
        Where cc.receiptID=@receiptID and cc.categoryID=@newCategoryID
    End
    Else
    Begin
        Insert Into CategoryChoice Values(@receiptID, @newCategoryID, 1, @price)
    End

    Update r
        Set r.totalPrice = r.totalPrice + @price
        From Receipt r
        Where r.receiptID=@receiptID
End
GO
Exec Sp_CreateReservationReceipt 41, 109, 22
Select * From Receipt
```

4- Name: Sp_GetCustomerID

Def: This SP returns Existing Customer or Creates new one's customerID.

Before/After:

125 %

Results | Messages

	customerID
1	NULL

	customerID
1	1003

SQLQuery13.sql - T...01-1292\enadi (56))* | SQLQuery4.sql - TB...01-1292\enadi (55))*

```sql
Create Proc Sp_GetCustomerID
    @customerName nvarchar(25), @customerLastName nvarchar(25),
    @customerID smallint Output
As
Begin
    Set Nocount On

    If Exists(Select * From Customer c
              Where c.firstName like @customerName and c.lastName like @customerLastName)
    Begin
        Select @customerID = c.customerID
        From Customer c
        Where c.firstName like @customerName and c.lastName like @customerLastName
        Return
    End
    Else
    Begin
        Insert Into Customer(firstName, lastName) Values(@customerName, @customerLastName)
        Select @customerID=@@Identity
        Return
    End
End
GO
Declare @customerID smallint
select @customerID customerID
Exec Sp_GetCustomerID 'hakan', 'sarı', @customerID = @customerID Output;
select @customerID customerID
```

5- Name: Sp_GetReservationOfTheDay

Def: This SP returns Reservation of the given day

SQLQuery13.sql - T...01-1292\enadi (56))* | SQLQuery4.sql - TB...01-1292\enadi (55))*

```sql
Create Proc Sp_GetReservationOfTheDay
    @givenDay date
As
Begin
    Set Nocount On

    SELECT c.firstName + c.lastName AS FullName, r.reservationTime
    FROM dbo.Reservation AS r INNER JOIN
                        dbo.Receipt AS rc ON r.receiptID = rc.receiptID INNER JOIN
                        dbo.Customer AS c ON rc.customerID = c.customerID
    WHERE (CAST(r.reservationTime AS date) = @givenDay)
End
GO

Exec Sp_GetReservationOfTheDay '2021-01-01'
```

125 %

Results | Messages

	FullName	reservationTime
1	SüleymanYoz...	2021-01-01 22:30:...
2	ErsinÖztekin	2021-01-01 10:10:...
3	ErmanSan	2021-01-01 10:30:...

6- Name: Sp_GetReservationOfTheCustomer

Def: This SP returns Reservation of the given customer

SQLQuery13.sql - T...01-1292\enadi (56))* | SQLQuery4.sql - TB...01-1292\enadi (55))*

```sql
Create Proc Sp_GetReservationOfTheCustomer
    @customerName nvarchar(25), @customerLastname nvarchar(25)
As
Begin
    Set Nocount On
    If Exists(Select * From Customer c Inner Join Receipt rc
              On c.customerID=rc.customerID Inner Join Reservation r
                On rc.receiptID=r.receiptID
              Where c.firstName like @customerName and c.lastName like @customerLastname)
    Begin
    SELECT c.firstName + ' ' + c.lastName AS FullName, r.reservationTime
    FROM Reservation r Inner Join Receipt rc
         ON r.receiptID = rc.receiptID Inner Join Customer c
            ON rc.customerID = c.customerID
    WHERE (c.firstName like @customerName and c.lastName like @customerLastname)
    End
    Else
    Begin
        Select @customerName + ' ' + @customerLastname FullName, NULL as ReservationTime
    End
End
GO
Exec Sp_GetReservationOfTheCustomer 'hakan', 'sarı'
```

125 %

Results | Messages

	FullName	reservationTime
1	Hakan San	2020-12-26 10:30:...

7- Name: Sp_GetTotalSalaryOfGivenStaffType

Def: This SP returns the salary of given staffType

SQLQuery13.sql - T...01-1292\enadi (56))* | SQLQuery4.sql - TB...01-1292\enadi (55))*

```sql
Create Proc Sp_GetTotalSalaryOfGivenStaffType
    @staffType char(1),
    @totalSalary int Output
As
Begin
    Set Nocount On
    Select @totalSalary = sum(sl.amount)
        From Staff st Inner Join Salary sl
            On st.salaryID=sl.salaryID
        Where st.staffType like @staffType

    Return
End
GO
Declare @totalSalary int
select @totalSalary totalSalary
Exec Sp_GetTotalSalaryOfGivenStaffType 'b', @totalSalary=@totalSalary Output;
select @totalSalary totalSalary
```

125 %

Results | Messages

	totalSalary
1	NULL

	totalSalary
1	10040

8- Name: Sp_GetIncomeOfGivenMonth

Def: This SP returns the income of given month

SQLQuery13.sql - T...01-1292\enadi (56))* | SQLQuery4.sql - TB...01-1292\enadi (55))*

SQLQuery13.sql - TB701-1292\MYMSSQLSERVER.KESGIN_KUAFOR (TB701-1292\enadi (56))*

```sql
    @totalIncome int Output
As
Begin
    Set Nocount On
    Select @totalIncome = sum(r.totalPrice)
    From Receipt r
    Where DATENAME(MONTH, Cast(r.date as datetime)) like @month

    Return
End
GO
Declare @totalIncome int
select @totalIncome totalIncome
Exec Sp_GetIncomeOfGivenMonth 'January', @totalIncome=@totalIncome Output;
select @totalIncome totalIncome
```

125 %

Results | Messages

	totalIncome
1	NULL

	totalIncome
1	359

9- Name: Sp_ProductQuantity

Def: This SP returns if the given product is enough or not

SQLQuery1.sql - DE...TC50\asakar11 (54))* DESKTOP-DGOTC50\...FOR - dbo.Product

```sql
create procedure Sp_ProductQuantity(@productID  smallint)
as
declare @Quantity smallint

select @Quantity = p.quantity
from Product p
where productID = @productID

if (@Quantity < 10)
begin
print 'Yetersiz miktarda ürün.'
end
else
begin
print 'Yeterli miktarda ürün.'
end

exec Sp_ProductQuantity 518
```

130 %

Messages

```
Yeterli miktarda ürün.
```