



İleri Veritabanı

Mehmet Kocaman

H5180010

İçindekiler

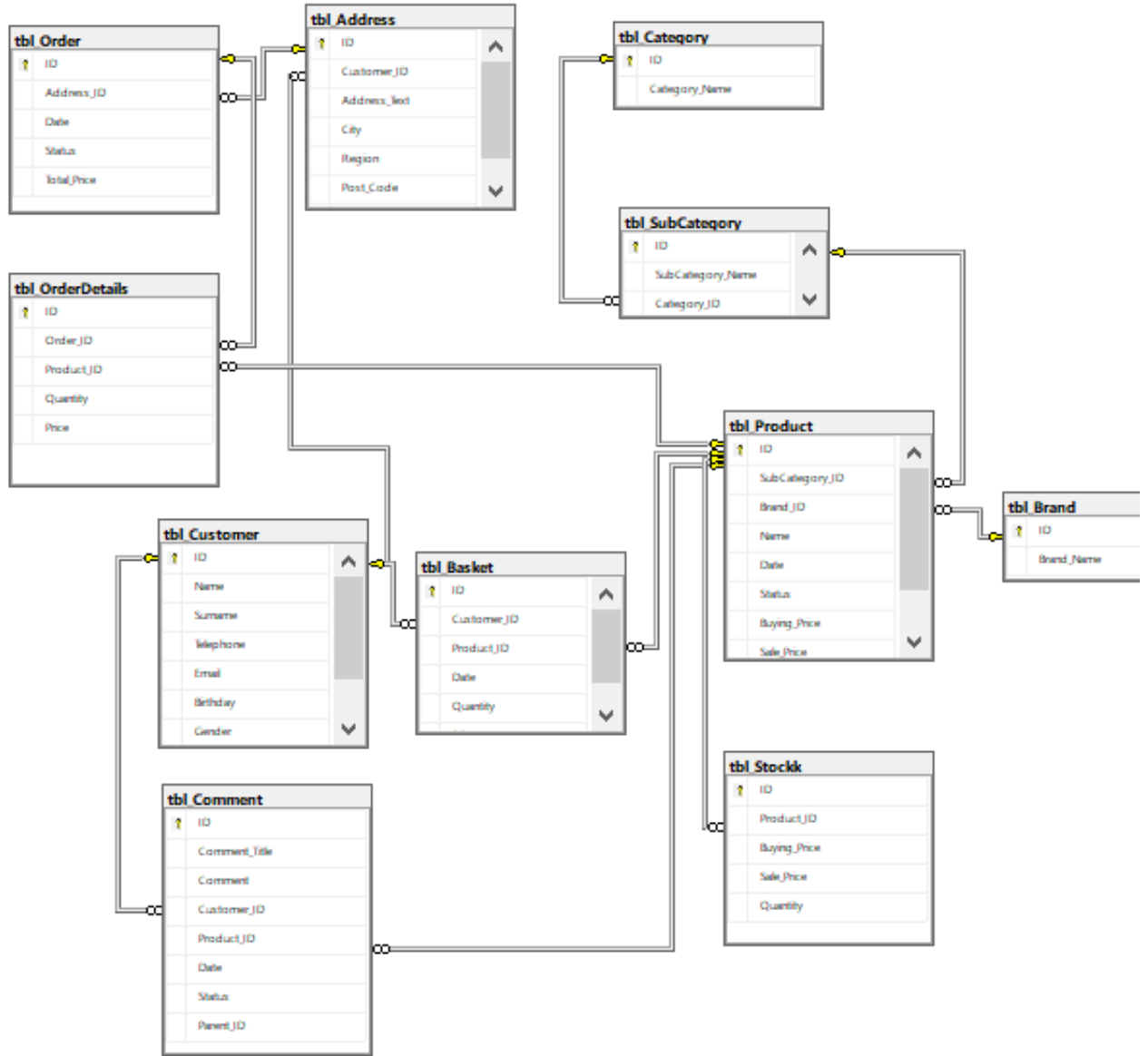
İçindekiler	2
Proje Amaç.....	3
Proje Diagram	4
Proje Tablolar.....	5
Customer Tablosu.....	5
Address Tablosu.....	5
Product Tablosu	6
Brand Tablosu.....	6
Category Tablosu	7
SubCategory Tablosu	7
Stockk Tablosu	7
Comment Tablosu.....	8
Basket Tablosu.....	8
Order Tablosu	9
Order Details Tablosu	9
Tablo Procedure İşlemleri	10
Customer Insert	10
Address Insert.....	10
Product Insert	11
Brand Insert	11
Category Insert	12
SubCategory Insert	12
Stockk Insert	12
Comment Insert.....	13
Basket Insert	13
Order Insert	14
Order Details Insert	14
Tablo View İşlemleri	15
Customer Address View	16
Product View	16
Category ve SubCategoryView	16
Comment View	16
Order Order Details View	16
Fonksiyon Kar Hesabı	17
Rastgele Müşteri Kayıt sss	18

Proje Ama

Bir E – Ticaret sistemi iin veritabanı sistemi oluřturuldu. Mřterilerin rahatlıkla rn sepete ekleyip kolaylıkla sipariř verebilmesini saėlamaktadır.

Mřterilerin rnlerini sepete ekleyip sipariř verme sistemine sahip bir veritabanı sistemi oluřturulmuřtur. Mřterilerin birden fazla adresi olma zelliėine sahiptir. Mřteriler birden fazla rn alabilirler ve rnleri birden fazla mřteri seėebilir. Satıcı aısından rn kar hesabı saėlanmaktadır. rnler kategorilerine ve bir alt kategorilerine gre listelenmektedir.

Proje Diagram



E-Ticaret veritabanı diyagramı bu şekildedir. Tabloların ilişki column' ları gösterilmektedir.

Tablolar;

- tbl_Customer
- tbl_Address
- tbl_Product
- tbl_Category
- tbl_SubCategory
- tbl_Brand
- tbl_Stockk
- tbl_Basket
- tbl_Order
- tbl_OrderDetails
- tbl_Comment

Proje Tabloları

Customer Tablosu

Column Name	Data Type	Allow Nulls
ID	int	<input type="checkbox"/>
Name	nvarchar(50)	<input checked="" type="checkbox"/>
Surname	nvarchar(50)	<input checked="" type="checkbox"/>
Telephone	char(11)	<input checked="" type="checkbox"/>
Email	nvarchar(50)	<input checked="" type="checkbox"/>
Birthday	date	<input checked="" type="checkbox"/>
Gender	char(1)	<input checked="" type="checkbox"/>
Status	char(1)	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

- ID Primary Key dir.
- Name ve Surname alanı non-clustured index' dir.
- Müşterilerin bilgilerinin tutulduğu tablo yapısıdır.
- Status alanı eğer kullanıcı Aktif ise durumu 'A' değilse durumu 'P' olarak tasarlanmıştır.

Address Tablosu

Column Name	Data Type	Allow Nulls
ID	int	<input type="checkbox"/>
Customer_ID	int	<input checked="" type="checkbox"/>
Address_Text	nvarchar(MAX)	<input checked="" type="checkbox"/>
City	nvarchar(50)	<input checked="" type="checkbox"/>
Region	nvarchar(50)	<input checked="" type="checkbox"/>
Post_Code	nvarchar(50)	<input checked="" type="checkbox"/>
Country	nvarchar(50)	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

- ID alanı Primary Key' dir.
- Customer_ID alanı Foreign Key' dir.
- Customer_ID, City ve Region alanı non-clustured index' dir.
- Müşterilerin bu tablo sayesinde birden fazla adrese sahip olmasın sağlar.
- Müşterilerin adres bilgilerinin tutulduğu tablodur.

Product Tablosu

DESKTOP-8F8HA90\S...- dbo.tbl_Product			
	Column Name	Data Type	Allow Nulls
PK	ID	int	<input type="checkbox"/>
	SubCategory_ID	int	<input checked="" type="checkbox"/>
	Brand_ID	int	<input checked="" type="checkbox"/>
	Name	nvarchar(50)	<input checked="" type="checkbox"/>
	Date	date	<input checked="" type="checkbox"/>
	Status	char(1)	<input checked="" type="checkbox"/>
	Buying_Price	money	<input checked="" type="checkbox"/>
	Sale_Price	money	<input checked="" type="checkbox"/>
	Quantity	int	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

- ID alanı Primary Key' dir.
- SubCategory_ID ve Brand_ID alanı Foreign Key' dir.s
- Category_ID, Brand_ID ve Name alanı non-clustured index' dir.
- Müşterilerin satın alacağı ürünlerin kategorilerine ve markalarına göre listelendiği tablo yapısıdır.
- Status alanı eğer ürün aktif ise 'A' ürün pasif ise 'P' olarak tasarlanmıştır.
- Brand_ID ile tbl_Brand tablosundaki ID ilişkilidir.
- SubCategory_ID ile tbl_SubCategory tablosundaki ID ilişkilidir.

Brand Tablosu

DESKTOP-8F8HA90\S...et - dbo.tbl_Brand			
	Column Name	Data Type	Allow Nulls
PK	ID	int	<input type="checkbox"/>
	Brand_Name	nvarchar(50)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

- ID alanı Primary Key' dir.
- Brand_Name alanı non-clustured index' dir.
- Ürünlerin markaları bu tabloda listelenir.

Category Tablosu

DESKTOP-8F8HA90\S...dbo.tbl_Category - X DESKTOP-8F8HA90\S...et -			
	Column Name	Data Type	Allow Nulls
PK	ID	int	<input type="checkbox"/>
	Category_Name	nvarchar(50)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

- ID alanı Primary Key' dir.
- Category_Name alanı non-clustured index' dir.
- Ürünlerin kategorilerinin listelendiği tablo yapısıdır.

SubCategory Tablosu

DESKTOP-8F8HA90\S...o.tbl_SubCategory - X DESKTOP-8F8HA90\S...t -			
	Column Name	Data Type	Allow Nulls
PK	ID	int	<input type="checkbox"/>
	SubCategory_Name	nvarchar(50)	<input checked="" type="checkbox"/>
	Category_ID	int	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

- ID alanı Primary Key' dir.
- Customer_ID alanı Foreign Key' dir.
- SubCategory_Name ve Category_ID alanı non-clustured index' dir.
- Category_ID ile tbl_Category tablosundaki ID alanı ilişkilidir.

Stockk Tablosu

DESKTOP-8F8HA90\S...t - dbo.tbl_Stockk - X DESKTOP-8F8HA90\S...- d			
	Column Name	Data Type	Allow Nulls
PK	ID	int	<input type="checkbox"/>
	Product_ID	int	<input checked="" type="checkbox"/>
	Buying_Price	money	<input checked="" type="checkbox"/>
	Sale_Price	money	<input checked="" type="checkbox"/>
	Quantity	int	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

- ID alanı Primary Key' dir.
- Product_ID alanı non-clustured index ve Foreign Key alanıdır.
- Product_ID ile tbl_Product tablosundaki ID alanı ilişkilidir.
- Ürünlerin stok bilgisinin tutulduğu kar oranının yapıldığı tablo yapısıdır.

Comment Tablosu

DESKTOP-8F8HA90\...dbo.tbl_Comment - X DESKTOP-8F8HA90\S...ic			
	Column Name	Data Type	Allow Nulls
PK	ID	int	<input type="checkbox"/>
	Comment_Title	nvarchar(50)	<input checked="" type="checkbox"/>
	Comment	nvarchar(50)	<input checked="" type="checkbox"/>
	Customer_ID	int	<input checked="" type="checkbox"/>
	Product_ID	int	<input checked="" type="checkbox"/>
	Date	datetime	<input checked="" type="checkbox"/>
	Status	char(1)	<input checked="" type="checkbox"/>
	Parent_ID	int	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

- ID alanı Primary Key' dir.
- Comment_Title, Comment, Customer_ID ve Product_ID alanı non-clustured index' dir.
- Customer_ID ve Product_ID alanı Foreign Key' dir.
- Product_ID ile tbl_Product tablosundaki ID alanı ilişkilidir.
- Customer_ID ile tbl_Customer tablosundaki ID alanı ilişkilidir.
- Müşterilerin ürünlere yorum yapmasını sağlayan tablo yapısıdır.
- Parent_ID '0' ise ana yorum olarak algılanır.
- Parent_ID farklı bir değer aldığı anda ana yorum yapılan bir alt yorum olarak algılanır.

Basket Tablosu

DESKTOP-8F8HA90\S...t - dbo.tbl_Basket - X DESKTOP-8F8HA90\S...ica			
	Column Name	Data Type	Allow Nulls
PK	ID	int	<input type="checkbox"/>
	Customer_ID	int	<input checked="" type="checkbox"/>
	Product_ID	int	<input checked="" type="checkbox"/>
	Date	date	<input checked="" type="checkbox"/>
	Quantity	int	<input checked="" type="checkbox"/>
	Price	money	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

- ID alanı Primary Key' dir.
- Customer_ID ve Product_ID alanı non-clustured index' dir ve Foreign Key alanlarıdır.
- Product_ID ile tbl_Product tablosundaki ID alanı ilişkilidir.
- Customer_ID ile tbl_Customer tablosundaki ID alanı ilişkilidir.
- Müşterinin ürünlerini sepete eklemesini sağlayan tablo yapısıdır.

Order Tablosu

DESKTOP-8F8HA90\S...et - dbo.tbl_Order			
	Column Name	Data Type	Allow Nulls
PK	ID	int	<input type="checkbox"/>
	Address_ID	int	<input checked="" type="checkbox"/>
	Date	date	<input checked="" type="checkbox"/>
	Status	char(1)	<input checked="" type="checkbox"/>
	Total_Price	money	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

- ID alanı Primary Key' dir.
- Address_ID Foreign Key' dir.
- Address_ID alanı non-clustured index' dir.
- Address_ID ile tbl_Address tablosundaki ID alanı ilişkilidir.
- Müşterinin ürünlerini sepete ekledikten sonra siparişini gerçekleştirmesine yardımcı olan tablodur.

Order Details Tablosu

DESKTOP-8F8HA90\S...o.tbl_OrderDetails			
	Column Name	Data Type	Allow Nulls
PK	ID	int	<input type="checkbox"/>
	Order_ID	int	<input checked="" type="checkbox"/>
	Product_ID	int	<input checked="" type="checkbox"/>
	Quantity	int	<input checked="" type="checkbox"/>
	Price	money	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

- ID alanı Primary Key' dir.
- Order_ID ve Product_ID Foreign Key' dir.
- Order_ID ve Product_ID alanı non-clustured index' dir.
- Order_ID ile tbl_Order tablosundaki ID alanı ilişkilidir.
- Product_ID ile tbl_Product tablosundaki ID alanı ilişkilidir.
- Müşterinin ürünlerini sepete ekleyip sonra siparişini gerçekleştirdikten sonra sipariş sonrası satın aldığı ürünleri görmek için yapılan tablodur.

Tablo Procedure İşlemleri

Customer Insert Procedure

```
:create procedure usp_CustomerInsert
@Name nvarchar(50),
@Surname nvarchar(50),
@Telephone char(11),
@email nvarchar(50),
@Birthday date,
@Gender char(1),
@Status char(1)
as
:begin

:    INSERT INTO tbl_Customer(Name,Surname,Telephone,Email,Birthday,Gender,Status)
:    VALUES (@Name,@Surname,@Telephone,@Email,@Birthday,@Gender,@Status)
:
:
:    end
```

Customer tablosuna kayıt atmamızı sağlayan Stored Procedure örnekte ki gibidir. Kayıt işleminde bütün columnları kayıt atıyoruz. Birthday alanı kayıt esnasında unutulduğunda getdate() olarak ayarlanmıştır. Status otomatik olarak 'A' aktif olarak çağırılıyor.

Address Insert Procedure

```
create procedure usp_AddressInsert
@Address_Text nvarchar(MAX),
@Customer_ID int,
@City nvarchar(50),
@Region nvarchar(50),
@Post_Code char(5),
@Country nvarchar(50)
as
begin

    INSERT INTO tbl_Address(Address_Text,Customer_ID,City,Region,Post_Code,Country)
    VALUES (@Address_Text,@Customer_ID,@City,@Region,@Post_Code,@Country)

end
```

Address tablosuna kayıt atmamızı sağlayan Stored Procedure örnekte ki gibidir. Procedure sayesinde bir Müşterinin birden fazla Adresi olabilmektedir.

Product Insert

```
CREATE PROCEDURE usp_ProductInsert
@SubCategory_ID int,
@Brand_ID int,
@Name nvarchar(50),
@Buying_Price money,
@Sale_Price money,
@Quantity int
as
begin
    insert into tbl_Product(SubCategory_ID,Brand_ID,Name,Buying_Price,Sale_Price,Quantity)
    values(@SubCategory_ID,@Brand_ID,@Name,@Buying_Price,@Sale_Price,@Quantity)
end
```

Product tablosuna kayıt yapmamızı sağlayan Stored Procedure örnekte ki gibidir. Ürünlerin kaydını tutmamıza yarayan Product tablosuna kayıt yapmamızı kolaylaştırıyor. Date alanı otomatik getdate() olarak geliyor.

Brand Insert

```
create procedure usp_BrandInsert
@Brand_Name nvarchar(50)
as
begin
    INSERT INTO tbl_Brand(Brand_Name)
    VALUES (@Brand_Name)
end
```

Brand tablosuna kayıt yapmamızı sağlayan Stored Procedure örnekte ki gibidir. Ürünlerin markaları olmaktadır. Bu markalar ayrı bir tabloya kayıt yapılması ve listelenmesi gerekir. Ürünlerin bilgilerinde yer alan marka isimleri bu tablodan çekilir.

Category Insert

```
create procedure usp_CategoryInsert
@CategoryName nvarchar(50)
as
begin

    insert into tbl_Category(Category_Name)
    values (@CategoryName)

end
```

Category tablosuna kayıt yapmamızı sağlayan Stored Procedure örnekte ki gibidir. Ürünler bir kategoriden gelmektedir. Bu kategoriler ürünlerin kolay bulunması sağlar. Insert işlemi bu şekilde gerçekleşir.

SubCategory Insert

```
create procedure usp_SubCategoryInsert
@SubCategory_Name nvarchar(50),
@Category_ID int
as
begin

    insert into tbl_SubCategory(SubCategory_Name,Category_ID)
    values (@SubCategory_Name,@Category_ID)

end
```

SubCategory tablosuna kayıt yapmamızı sağlayan Stored Procedure örnekte ki gibidir. Ürünler bir kategoriden gelmektedir. Bu kategorilerinde bir alt kategorisi olmak zorundadır. Insert işlemi bu şekilde gerçekleşir.

Stockk Insert

```
CREATE PROCEDURE usp_StockkInsert
@Product_ID int,
@Buying_Price money,
@Sale_Price money,
@Quantity int
as
begin
    insert into tbl_Stockk(Product_ID,Buying_Price,Sale_Price,Quantity)
    values(@Product_ID,@Buying_Price,@Sale_Price,@Quantity)

end
```

Stockk tablosuna kayıt yapmamızı sağlayan Stored Procedure örnekte ki gibidir. Ürünler stok bilgilerinin tutulmasını sağlayan bir tablo yapısıdır. Bu tabloya kayıt örnekte görüldüğü gibi yapılmaktadır.

Comment Insert

```
create procedure usp_CommentInsert
@Comment_Title nvarchar(50),
@Comment nvarchar(50),
@Customer_ID int,
@Product_ID int,
@Parent_ID int
as
begin

    INSERT INTO tbl_Comment(Comment_Title,Comment,Customer_ID,Product_ID,Parent_ID)
    VALUES (@Comment_Title,@Comment,@Customer_ID,@Product_ID,@Parent_ID)

end
```

Comment tablosuna kayıt yapmamızı sağlayan Stored Procedure örnekte ki gibidir. Ürünlere müşterilerin yorum yapmasını sağlayan tablo kaydı yukarıdaki gibidir. Date alanı getdate() olarak gelmektedir.

Basket Insert

```
create procedure usp_BasketInsert
@Customer_ID int,
@Product_ID int,
@Quantity int,
@Price money
as
begin
    INSERT INTO tbl_Basket(Customer_ID,Product_ID,Quantity,Price)
    VALUES (@Customer_ID,@Product_ID,@Quantity,@Price)

end
```

Basket tablosuna kayıt yapmamızı sağlayan Stored Procedure örnekte ki gibidir. Ürünlerin sepete eklenmesini sağlayan tablo yapısına kayıt işlemi Procedure sayesinde gerçekleşiyor.

Order Insert

```
CREATE PROCEDURE usp_OrderInsert
@Address_ID int,
@TotalPrice money
as
begin

INSERT INTO tbl_Order(Address_ID,Total_Price)
values (@Address_ID,@TotalPrice)

end
```

Order tablosuna kayıt yapmamızı sağlayan Stored Procedure örnekte ki gibidir. Ürünleri sepete ekleyen müşteri onayladığı zaman sipariş kısmına kayıt gerçekleşir. Procedure burada kullanılıyor. Date alanı getdate() olarak verilmiştir. Siparişi gerçekleştiren kullanıcının Status alanı 'A' aktif olmaktadır. Gerçekleşmez ise Status alanı 'P' pasif olmaktadır.

Order Details Insert

```
CREATE PROCEDURE usp_OrderDetailsInsert
@Order_ID int,
@Product_ID int,
@Quantity int,
@Price money
as
begin
INSERT INTO tbl_OrderDetails(Order_Id,Product_ID,Quantity,Price)
values (@Order_Id,@Product_ID,@Quantity,@Price)

end
```

Order Details tablosuna kayıt yapmamızı sağlayan Stored Procedure örnekte ki gibidir. Ürünleri sepete ekleyen müşteri onayladığı zaman sipariş kısmına kayıt gerçekleşir. Sipariş onaylandığında Sipariş detayları bu tabloya kayıt gerçekleştirir. Siparişte ki ürün bilgileri bu tabloda listelenir.

Tablo View İşlemleri

Customer Address View

```
create view V_CustomerAddress
as

SELECT c.Name+' '+c.Surname as NameSurname,
       a.Address_Text+' '+a.Region+' / '+a.City as Address,
       c.Telephone,
       c.Email,
       c.Birthday,
       Status

FROM tbl_Customer c
     INNER JOIN tbl_Address a ON c.ID=a.Customer_ID

SELECT * FROM V_CustomerAddress
```

Bir müşterinin birden fazla adresi vardır. Müşteri ve Adres tablosunu birleştirerek yoğunluğu azaltmak için bu şekilde bir view yazılmıştır.

Product View

```
create View V_ProductINFO
as
select c.Category_Name,
       s.SubCategory_Name,
       p.Name,
       p.Date,
       p.Buying_Price,
       p.Sale_Price,
       p.Quantity,
       p.Status
from tbl_Product p
     INNER JOIN tbl_SubCategory s on p.SubCategory_ID=s.ID
     INNER JOIN tbl_Category c on s.Category_ID=c.ID
```

Ürünler listelenirken görünüşü kolaylaştırmak için yapılmıştır.

Category ve SubCategory View

```
create view V_CategoryandSub
as
SELECT c.Category_Name,
       s.SubCategory_Name
FROM tbl_Category c
     INNER JOIN tbl_SubCategory s on c.ID=s.Category_ID

select * from V_CategoryandSub
```

Ürünler bir kategori ve bir alt kategoride listelenir. Görünüşü kolaylaştırmak için Category ve SubCategory tabloları birleştirilmiştir.

Comment View

```
create view V_Comment
as
select cu.Name+' '+cu.Surname as NameSurname,
       p.Name,
       c.Comment_Title,
       c.Comment,
       c.Date,
       c.Status,
       c.Parent_ID
from tbl_Comment c
INNER JOIN tbl_Customer cu ON c.Customer_ID=cu.ID
INNER JOIN tbl_Product p ON p.ID=c.Product_ID
```

Comment tablosunda listelenen verilerin görünürlüğünü kolaylaştırmak adına yazılmıştır. İlişkili tablolar birleştirilip yazılmıştır.

Order Order Details View

```
create view V_Order_OrderDetails_Product
as
select c.Name+' '+c.Surname as NameSurname,
       a.Address_Text+' '+a.Region+' / '+a.City as Address,
       p.Name as ProductName,
       o.Date,
       o.Status,
       od.Quantity,
       od.Price,
       o.Total_Price
from tbl_Order o
INNER JOIN tbl_OrderDetails od on o.ID=od.Order_ID
INNER JOIN tbl_Address a on a.ID=o.Address_ID
INNER JOIN tbl_Customer c on a.Customer_ID=c.ID
INNER JOIN tbl_Product p on p.ID=od.Product_ID
```

Sipariş, Müşteri, Sipariş Detay ,Adres ve Ürün tablolarını ilişkilendirip listeleme sağlanmıştır.

Fonksiyon Kar Hesabı

```
create FUNCTION fn_ProfitAccount
(
@Product_ID int
)
returns decimal
as
begin
    set @Product_ID=(select top 1 Product_ID from tbl_Stockk Where Product_ID=@Product_ID)

    declare @TotalBuyingQuantity decimal
    set @TotalBuyingQuantity=(select SUM(Buying_Price*Quantity)
                             from tbl_Stockk
                             where Product_ID=@Product_ID )

    declare @FinalSalePrice decimal
    set @FinalSalePrice=(select top 1 Sale_Price from tbl_Stockk
                         where Product_ID=@Product_ID
                         order by ID desc)

    declare @TotalProductQuantity decimal
    set @TotalProductQuantity=(select SUM(Quantity)
                               from tbl_Stockk
                               where Product_ID=@Product_ID)

    declare @SalePriceQuantity decimal
    set @SalePriceQuantity=@FinalSalePrice*@TotalProductQuantity

    declare @ProfitRate decimal
    set @ProfitRate= (@SalePriceQuantity - @TotalBuyingQuantity)*100/@TotalBuyingQuantity

    return @ProfitRate
end
```

Kar hesabı yüzde oranına göre yazılmaktadır. Ürünün alış fiyatı ile adeti çarpılır. Birden fazla aynı ürün stoku varsa column toplanır. Son satış fiyatı alınır ve toplam ürün adetine çarpılır. Toplam Alış Fiyat ve Toplam Satış Fiyat ortaya çıkar.

$(TSF-TAF)*100/TAF$

Sonuç yüzde oranında yazdırılır.

Rastgele Müşteri Kayıt

```
declare @Name nvarchar(100)
declare @Surname nvarchar(100)
declare @Telephone char(11)
declare @Email nvarchar(100)
declare @Gender char(1)

declare @sayac int
set @sayac = 0

while (@sayac<100)
begin

    declare @RId int
    set @RId = RAND() * 12923 +1
    set @Name = (select Ad from isimler where Id = @RId)
    set @Gender = ( select Cinsiyet from isimler where Id = @RId)

    set @RId = RAND() *1603+1
    set @Surname = (select Soyadi from Soyisimler where Id = @RId)

    set @Email = @Name + '.'+@Surname+ '@medipol.edu.tr'

    declare @RTelephone2 int
    set @RTelephone2=RAND()*38+32

    declare @RTelephone3 int
    set @RTelephone3=RAND()*999+101

    declare @RTelephone4 int
    set @RTelephone4=RAND()*9999+1000

    set @Telephone='05392925592'

    insert into tbl_Customer(Name,Surname,Telephone,Email,Gender)
    values(@Name,@Surname,@Telephone,@Email,@Gender)

    set @sayac = @sayac+1
end
```