Veritabanı yönetim sistemleri dersinin projesi

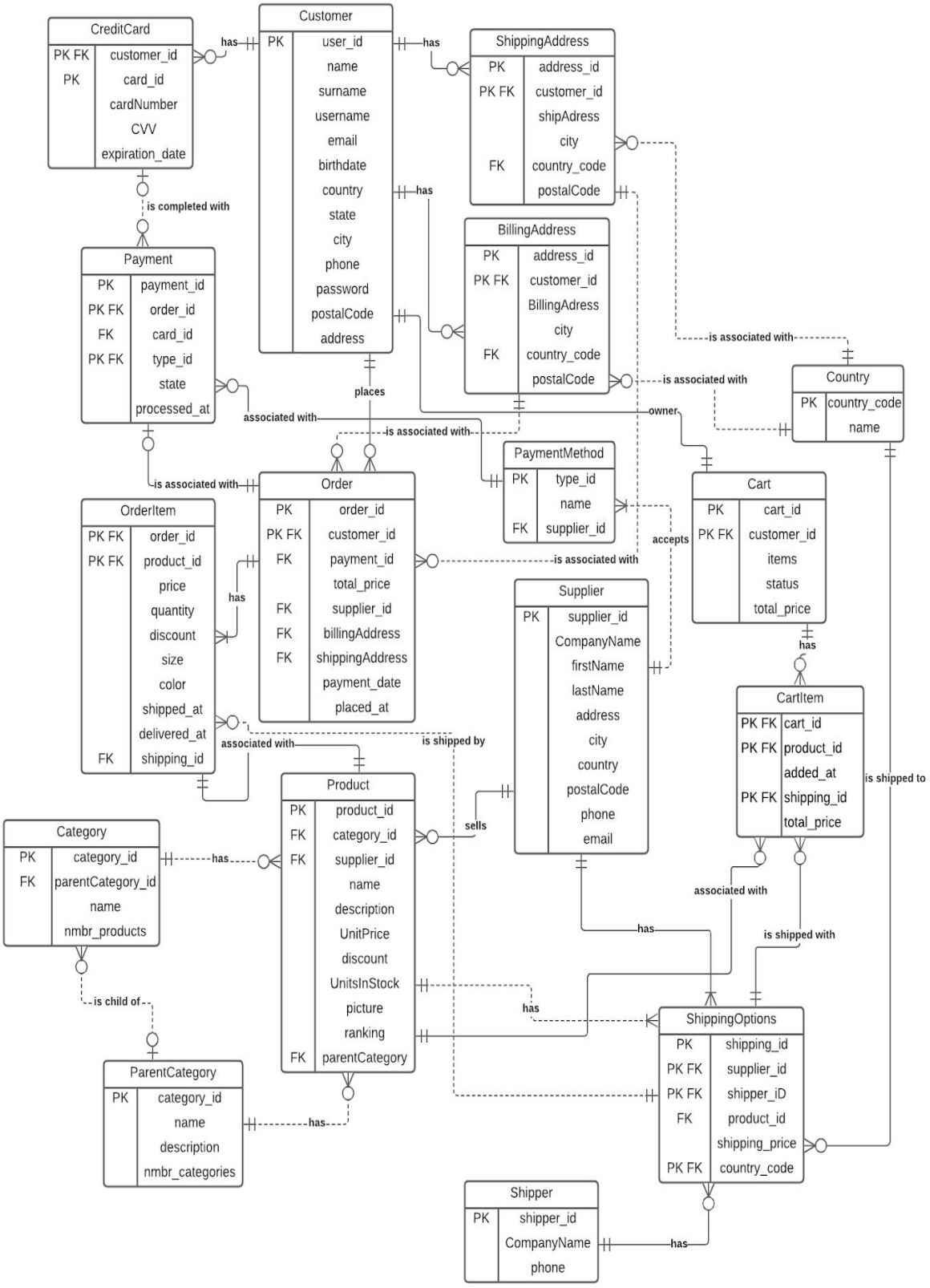
Ders Adı **: Veritabanı yönetim sistemleri**

Hazırlayan: hajer gafsi

Öğrenci numara : B181210562

ŞUBE : 1C

E-mail: Hajer.gafsi@ogr.sakarya.edu.tr


CreditCard
PK FK customer_id
PK card_id
cardNumber
CVV
expiration_date
has
Customer
PK user_id
name
surname
username
email
birthdate
country
state
city
phone
password
postalCode
address
has
ShippingAddress
PK address_id
PK FK customer_id
shipAdress
city
FK country_code
postalCode
is completed with
Payment
PK payment_id
PK FK order_id
FK card_id
PK FK type_id
state
processed_at
has
BillingAddress
PK address_id
PK FK customer_id
BillingAdress
city
FK country_code
postalCode
is associated with
Country
PK country_code
name
places
associated with
is associated with
owner
is associated with
PaymentMethod
PK type_id
name
FK supplier_id
is associated with
Order
PK order_id
PK FK customer_id
FK payment_id
total_price
FK supplier_id
FK billingAddress
FK shippingAddress
payment_date
placed_at
accepts
Cart
PK cart_id
PK FK customer_id
items
status
total_price
OrderItem
PK FK order_id
PK FK product_id
price
quantity
discount
size
color
shipped_at
delivered_at
FK shipping_id
has
Supplier
PK supplier_id
CompanyName
firstName
lastName
address
city
country
postalCode
phone
email
has
CartItem
PK FK cart_id
PK FK product_id
added_at
PK FK shipping_id
total_price
is shipped to
associated with
is shipped by
Product
PK product_id
FK category_id
FK supplier_id
name
description
UnitPrice
discount
UnitsInStock
picture
ranking
FK parentCategory
sells
Category
PK category_id
FK parentCategory_id
name
nmbr_products
has
associated with
has
is shipped with
is child of
has
ShippingOptions
PK shipping_id
PK FK supplier_id
PK FK shipper_iD
FK product_id
shipping_price
PK FK country_code
ParentCategory
PK category_id
name
description
nmbr_categories
has
Shipper
PK shipper_id
CompanyName
phone
has

Projenin tanitimi

Bir E-ticaret sitesinin siparis sisteminin veritabani modellenmesi ve yönetim sistemi

Iş kuralları :

- ✔ **Bir müşteri(customer)'nin id, name, surname, username, email ,birthdate, country , state, city, phone, password, postalcode, ve address bilgileri vardir.**
- ✔ **Bir Adress'in (Billing veya shipping Adresi) address_id, address, customer_id, city, country_id ve postal_code bilgileri mevcuttur**
- ✔ **Bir sipariş'(order)in order_id, customer_id, payment_id, total_price, supplier_id billingAdress, shippingAdress ve payment_date bilgileri vardir**
- ✔ **Bir ürün'ün (product) product_id, category_id, supplier_id, name,description , unit_price, discount, unitsInStock, picture, ranking ve parentCategory bilgileri vardir**
- ✔ **Bir saticinin (supplier) supplier_id, CompanyName, firstName, lastName, address, email , country, city ve phone bilgileri mevcuttur.**
- ✔ **Bir müşteri'nin birden fazla shipping adresi olabilir**
- ✔ **Bir müşteri'nin birden fazla billing adresi olabilir**
- ✔ **Bir müşteri birden fazla sipariş verebilir**
- ✔ **Bir müşteri'nin birden fazla kredi karti olabilir**
- ✔ **Bir shipping veya billing Address sadece bir müşteri'ye ilişkili olabilir**
- ✔ **Bir shipping veya billing Address sadece bir 'ye (country) ilişkili olabilir**
- ✔ **Bir ülke (country) birden fazla shipping ve billing adreslerinde yer alabilir**
- ✔ **Bir ülke (country) birden fazla kargo seçenegi (shipping option) da yer alabilir**
- ✔ **Bir shipping veya billing address birden fazla sipariş'te yer alabilir**
- ✔ **Bir sipariş sadece bir müşteri'ye ait olabilir**
- ✔ **Bir sipariş'in sadece bir billing adresi ve bir shipping adresi olabilir**
- ✔ **Bir müşteri sadece bir sepete(cart) sahip olabilir**
- ✔ **Bir sepet(cart) sadece bir müşteri'ye ait olabilir**
- ✔ **Bir sepet(cart) 'in birden fazla CartItem olabilir**

- ✔ **Bir sipariş en az bir orderItem içerebilir**
- ✔ **Bir OrderItem sadece bir sipariş'a ait olabilir**
- ✔ **Bir ödeme (payment) sadece bir sipariş'a ait olabilir**
- ✔ **Bir ödeme en fazla bir kredi kart ile tamamlanir**
- ✔ **Bir ödeme sadece bir ödeme türü(paymentMethod)dan oluşur**
- ✔ **Bir Kredi Kart birden fazla ödeme tamamlayabilir**
- ✔ **Bir Kredi Kart sadece bir müşteri 'ye ait olabilir**
- ✔ **Bir ParentCategory birden fazla ürünu olabilir**
- ✔ **Bir ParentCategory birden fazla category ye sahip olabilir**
- ✔ **Bir Category'nin sadece bir tane ParentCategory'nin çocugu olabilir**
- ✔ **Bir Category'nin birden fazla ürün'ü olabilir**
- ✔ **Bir ürün sadece bir category'ye ait olabilir**
- ✔ **Bir ürün sadece bir parentCategory'ye ait olabilir**
- ✔ **Bir ürün sadece bir satici(supplier) tarafindan satilir**
- ✔ **Bir ürün sadece bir OrderItem'e ilişkili olabilir**
- ✔ **Bir ürün'un en az bir ödeme seçenegi olur**
- ✔ **Bir ürün birden fazla cartItem'e ilişkili olabilir**
- ✔ **Bir satici (supplier) en az bir ödeme türü(paymentMethod) kabul eder**
- ✔ **Bir satici (supplier) en az bir kargo seçenegi (shipping Option) kullanir**
- ✔ **Bir satici birden fazla ürün satar**
- ✔ **Bir ödeme türü(paymentMethod) sadece bir satici'ye ilişkili olabilir**
- ✔ **Bir ödeme türü(paymentMethod) birden fazla ödemede(payment) yer alabilir**
- ✔ **Bir CartItem sadece bir sepetin (cart) item'i olabilir**
- ✔ **Bir CartItem sadece bir kargo seçenegi (shipping Option) ile gonderilebilir**
- ✔ **Bir CartItem sadece bir urun(product) ile ilişkili olabilir**
- ✔ **Bir Gonderici (shipper) birden fazla kargo seçenekleri (shippingOptions) olur**
- ✔ **Bir kargo seçenegi (shippingOption) sadece bir Gonderici'ye ilişkili olur**

- ✔ Bir kargo seçenegi (shippingOption) sadece bir ulkeye kargonun gondermesine saglar
- ✔ Bir kargo seçenegi birden fazla Cart item ile baglantili olur
- ✔ Bir kargo seçenegi sadece bir satici tarafindan kullanilir
- ✔ Bir kargo seçenegi sadece bir ürüne ilişkilidir
- ✔ Bir kargo seçenegi birden fazla OrderItem 'in gonderilmesini saglayabilir
- ✔ Bir OrderItem sadece bir sipariş (order) ile baglantili olur
- ✔ Bir OrderItem sadece kargo seçenegi ile gonderilir
- ✔ Bir OrderItem sadece bir ürün ile baglantili olur

İlişkisel Şema (Metinsel Gösterim) :

- ➢ Customer(user_id: serial, name: varchar, surname: varchar, username: varchar, email: varchar, birthdate: date, country: varchar, state: varchar, city: varchar, phone: varchar, password: varchar, postalCode: char(5), address: varchar)
- ➢ Order(order_id: serial, customer_id: serial, payment_id: serial, total_price: Numeric, supplier_id: serial, placed_at: date, billingAdress: serial, shippingAdress: serial, payment_date: date)
- ➢ Product(product_id: serial, category_id: serial, supplier_id: serial, name: varchar, description: varchar, UntPrice: numeric, discount: numeric, UnitsInStock: int, picture: varchar, ranking: numeric, parentCategory: serial)
- ➢ Supplier(supplier_id: serial, CompanyName: varchar, firstName: varchar, lastname: varchar, email: varchar, birthdate: date, country: varchar, city: varchar, phone: varchar, postalCode: char(5), address: varchar)
- ➢ OrderItem (order_id: serial, product_id: serial, price: numeric, quantity: int, discount: numeric, size: char, color: varchar, shipped_at: date, delivered_at: date, shipping_id: serial)
- ➢ nmbr_products: int)
- ➢ ParentCategory(category_id: serial, name: varchar, description: varchar, nmbr_categories: int)
- ➢ Shipper(shipper_id: serial, CompanyName: varchar, phone: varchar)
- ➢ ShippingOptions(shipping_id: serial, supplier_id: serial, shipper_id: serial, country_code: char(3), product_id: serial, shipping_price: numeric)

- CartItem(cart_id: serial, product_id: serial, shipping_id: serial, added_at: date, total_price: numeric)
- Cart(cart_id: serial, customer_id: serial, items: int, status: varchar, total_price: numeric)
- Countries(country_code: char(3), name: varchar)
- BillingAddress(address_id: serial, customer_id: serial, billingAddress: varchar, city: varchar, postalCode: char(5), country_code: char(3))
- ShippingAddress(address_id: serial, customer_id: serial, shipAddress: varchar, postalCode: char(5), city: varchar, country_code: char(3))
- PaymentMethod(type_id: serial, name: varchar, supplier_id: serial)
- Payment(payment_id: serial, order_id: serial, type_id: serial, card_id: serial, state: varchar, processed_at: date)
- CreditCard(card_id: serial, customer_id: serial, cardNumber: char(16), CVV: char(3), expiration_date: date)
- Category(category_id: serial, parentCategory_id: serial, name: varchar,

Tabloları oluşturmaya sağlayan SQL komutları

```
-- CREATE TABLE "Customer" ------------------------------------
CREATE TABLE "public"."Customer" (
    "user_id" Serial NOT NULL,
    "name " Character Varying NOT NULL,
    "surname" Character Varying NOT NULL,
    "username" Character Varying NOT NULL,
    "email" Character Varying NOT NULL,
    "birthdate" Date,
    "country" Character Varying,
    "state" Character Varying,
    "city" Character Varying,
    "phone" Character Varying,
    "password" Character Varying NOT NULL,
    "postalCode" Character Varying,
    "address" Character Varying,
```

```
        PRIMARY KEY ( "user_id" ),
        CONSTRAINT "unique_Customer_user_id" UNIQUE( "user_id" ),
        CONSTRAINT "unique_Customer_username" UNIQUE( "username" ),
        CONSTRAINT "unique_Customer_email" UNIQUE( "email" ) );
;
-- ------------------------------------------------------------
-- CREATE TABLE "Country " ----------------------------------
CREATE TABLE "public"."Country " (
        "country_code" Character( 3 ) NOT NULL,
        "name" Character Varying NOT NULL,
        PRIMARY KEY ( "country_code" ) );
;
-- ------------------------------------------------------------
-- CREATE TABLE "BillingAddress" ----------------------------
CREATE TABLE "public"."BillingAddress" (
        "address_id" Serial NOT NULL,
        "customer_id" Serial NOT NULL,
        "billingAddress" Character Varying NOT NULL,
        "city" Character Varying NOT NULL,
        "country_code" Character( 3 ) NOT NULL,
        "postalCode" Character( 5 ) NOT NULL,
        PRIMARY KEY ( "address_id", "customer_id" ),
        CONSTRAINT "unique_BillingAddress_address_id" UNIQUE( "address_id" ) );
 -- CREATE LINK "public.BillingAddress.country_fk" -------------
ALTER TABLE "public"."BillingAddress"
        ADD CONSTRAINT "country_fk" FOREIGN KEY ( "country_code" )
        REFERENCES "public"."Country " ( "country_code" ) MATCH FULL
        ON DELETE Cascade
        ON UPDATE Cascade;
```

```
-- ------------------------------------------------------

-- ------------------------------------------------------

-- CREATE LINK "customer_idFK" -------------------------------
ALTER TABLE "public"."BillingAddress"
        ADD CONSTRAINT "customer_idFK" FOREIGN KEY ( "customer_id" )
        REFERENCES "public"."Customer" ( "user_id" ) MATCH FULL
        ON DELETE Cascade
        ON UPDATE Cascade;

-- ------------------------------------------------------

-- CREATE TABLE "Supplier" ----------------------------------
CREATE TABLE "public"."Supplier" (
        "supplier_id" Serial NOT NULL,
        "CompanyName" Character Varying,
        "firstName" Character Varying NOT NULL,
        "lastName" Character Varying NOT NULL,
        "address" Character Varying,
        "city" Character Varying,
        "country" Character Varying,
        "postalCode" Character Varying,
        "phone" Character Varying,
        "email" Character Varying NOT NULL,
        PRIMARY KEY ( "supplier_id" ) );
;

-- ------------------------------------------------------

-- CREATE TABLE "ShippingAddress" -----------------------------
CREATE TABLE "public"."ShippingAddress" (
        "address_id" Serial DEFAULT
nextval('"ShippingAddress_address_id_seq"'::regclass) NOT NULL,
        "customer_id " Serial NOT NULL,
```

```
        "shipAddress" Character Varying NOT NULL,
        "city" Character Varying NOT NULL,
        "country_code" Character( 3 ) NOT NULL,
        "postalCode" Character( 5 ) NOT NULL,
        PRIMARY KEY ( "address_id", "customer_id " ),
        CONSTRAINT "unique_ShippingAddress_address_id" UNIQUE( "address_id" )
);
 ;
-- ------------------------------------------------------------
-- CREATE LINK "public.ShippingAddress.customerFK" -------------
ALTER TABLE "public"."ShippingAddress"
        ADD CONSTRAINT "customerFK" FOREIGN KEY ( "customer_id " )
        REFERENCES "public"."Customer" ( "user_id" ) MATCH FULL
        ON DELETE Cascade
        ON UPDATE Cascade;
-- ------------------------------------------------------------
-- CREATE LINK "public.Country .countryFK" ----------------------
ALTER TABLE "public"."Country "
        ADD CONSTRAINT "countryFK" FOREIGN KEY ( "country_code" )
        REFERENCES "public"."Country " ( "country_code" ) MATCH FULL
        ON DELETE Cascade
        ON UPDATE Cascade;
-- ------------------------------------------------------------



COMMIT;
-- CREATE TABLE "PaymentMethod" ------------------------------
CREATE TABLE "public"."PaymentMethod" (
        "type_id" Serial NOT NULL,
```

```
        "name" Character Varying NOT NULL,
        "supplier_id" Serial NOT NULL,
        PRIMARY KEY ( "type_id" ) );
;
-- ------------------------------------------------------------


-- CREATE LINK "public.PaymentMethod.supplierFK" ---------------
ALTER TABLE "public"."PaymentMethod"
        ADD CONSTRAINT "supplierFK" FOREIGN KEY ( "supplier_id" )
        REFERENCES "public"."Supplier" ( "supplier_id" ) MATCH FULL
        ON DELETE Cascade
        ON UPDATE Cascade;
-- ------------------------------------------------------------
-- CREATE TABLE "ParentCategory" -----------------------------
CREATE TABLE "public"."ParentCategory" (
        "category_id" Serial DEFAULT
nextval('"ParentCategory_category_id_seq"'::regclass) NOT NULL,
        "name" Character Varying NOT NULL,
        "description" Character Varying NOT NULL,
        "nmbr_categories" Integer NOT NULL,
        PRIMARY KEY ( "category_id" ) );
;
-- ------------------------------------------------------------
-- CREATE TABLE "Category" -----------------------------------
CREATE TABLE "public"."Category" (
        "category_id" Serial DEFAULT
nextval('"Category_category_id_seq"'::regclass) NOT NULL,
        "parentCategory_id" Serial DEFAULT
nextval('"Category_parentCategory_id_seq"'::regclass) NOT NULL,
        "name " Character Varying NOT NULL,
```

```
        "nmbr_products" Integer NOT NULL,
        PRIMARY KEY ( "category_id" ) );
;
-- ------------------------------------------------------------
-- CREATE LINK "public.Category." ----------------------------
ALTER TABLE "public"."Category"
        ADD CONSTRAINT  FOREIGN KEY ( "parentCategory_id" )
        REFERENCES "public"."ParentCategory" ( "category_id" ) MATCH FULL
        ON DELETE Cascade
        ON UPDATE Cascade;
-- ------------------------------------------------------------
-- CREATE TABLE "Product" ------------------------------------
CREATE TABLE "public"."Product" (
        "product_id" Serial NOT NULL,
        "category_id" Serial NOT NULL,
        "supplier_id" Serial NOT NULL,
        "name" Character Varying NOT NULL,
        "description" Character Varying,
        "UnitPrice" Numeric NOT NULL,
        "discount" Numeric,
        "UnitsInStock" Integer NOT NULL,
        "picture" Character Varying,
        "ranking" Numeric,
        "parentCategory" Serial,
        PRIMARY KEY ( "product_id" ) );
;
-- ------------------------------------------------------------
-- CREATE LINK "public.Order.customerProdFK" -----------------
ALTER TABLE "public"."Order"
```

```
        ADD CONSTRAINT "customerProdFK" FOREIGN KEY ( "customer_id" )
        REFERENCES "public"."Customer" ( "user_id" ) MATCH FULL
        ON DELETE Cascade
        ON UPDATE Cascade;
-- ------------------------------------------------------
-- CREATE LINK "public.Order.billAddressFK" ----------------
ALTER TABLE "public"."Order"
        ADD CONSTRAINT "billAddressFK" FOREIGN KEY ( "billingAddress" )
        REFERENCES "public"."BillingAddress" ( "address_id" ) MATCH FULL
        ON DELETE Cascade
        ON UPDATE Cascade;
-- ------------------------------------------------------
-- CREATE LINK "public.Order.shipAddress" ------------------
ALTER TABLE "public"."Order"
        ADD CONSTRAINT "shipAddress" FOREIGN KEY ( "shippingAddress" )
        REFERENCES "public"."ShippingAddress" ( "address_id" ) MATCH FULL
        ON DELETE Cascade
        ON UPDATE Cascade;
-- ------------------------------------------------------
-- CREATE LINK "public.Order.orderSupplierAddress" ---------------
ALTER TABLE "public"."Order"
        ADD CONSTRAINT "orderSupplierFK" FOREIGN KEY ( "supplier_id " )
        REFERENCES "public"."Supplier" ( "supplier_id" ) MATCH FULL
        ON DELETE Cascade
        ON UPDATE Cascade;
-- ------------------------------------------------------
-- CREATE TABLE "OrderItem" ---------------------------------
CREATE TABLE "public"."OrderItem" (
        "order_id" Serial NOT NULL,
```

```
        "product_id" Serial NOT NULL,
        "price" Numeric NOT NULL,
        "quantity" Integer DEFAULT 1 NOT NULL,
        "discount" Numeric,
        "size" Numeric NOT NULL,
        "color" Character Varying NOT NULL,
        "shipped_at" Date,
        "delivered_at" Date,
        "placed_at" Date NOT NULL,
        "shipping_id" Serial NOT NULL,
        PRIMARY KEY ( "order_id", "product_id" ) );
;
-- ------------------------------------------------------------
-- CREATE LINK "public.OrderItem.orderFK" ----------------------
ALTER TABLE "public"."OrderItem"
        ADD CONSTRAINT "orderFK" FOREIGN KEY ( "order_id" )
        REFERENCES "public"."Order" ( "order_id" ) MATCH FULL
        ON DELETE Cascade
        ON UPDATE Cascade;
-- ------------------------------------------------------------
-- CREATE LINK "public.OrderItem.productOrderFK" ---------------
ALTER TABLE "public"."OrderItem"
        ADD CONSTRAINT "productOrderFK" FOREIGN KEY ( "product_id" )
        REFERENCES "public"."Product" ( "product_id" ) MATCH FULL
        ON DELETE Cascade
        ON UPDATE Cascade;
-- ------------------------------------------------------------
-- CREATE TABLE "CreditCard" ---------------------------------
CREATE TABLE "public"."CreditCard" (
```

```
        "card_id" Serial NOT NULL,
        "customer_id" Serial NOT NULL,
        "cardNumber" Character( 16 ) NOT NULL,
        "CVV" Character( 3 ) NOT NULL,
        "expiration_date" Date NOT NULL,
        PRIMARY KEY ( "card_id", "customer_id" ) );
CONSTRAINT "CreditCard_card_id_key" UNIQUE( "card_id" ) );
;
-- ------------------------------------------------------------
-- CREATE LINK "public.CreditCard.custFK" ----------------------
ALTER TABLE "public"."CreditCard"
        ADD CONSTRAINT "custFK" FOREIGN KEY ( "customer_id" )
        REFERENCES "public"."Customer" ( "user_id" ) MATCH FULL
        ON DELETE Cascade
        ON UPDATE Cascade;
-- ------------------------------------------------------------
-- CREATE TABLE "Payment" ------------------------------------
CREATE TABLE "public"."Payment" (
        "payment_id" Serial NOT NULL,
        "order_id" Serial NOT NULL,
        "card_id" Serial NOT NULL,
        "type_id" Serial NOT NULL,
        "state" Character Varying DEFAULT '"not processed"' NOT NULL,
        "processed_at" Date,
PRIMARY KEY( "payment_id", "order_id", "type_id" ) );
;
-- CREATE LINK "public.Payment.cardFK" ------------------------
ALTER TABLE "public"."Payment"
        ADD CONSTRAINT "cardFK" FOREIGN KEY ( "card_id" )
```

```
        REFERENCES "public"."CreditCard" ( "card_id" ) MATCH FULL
        ON DELETE Cascade
        ON UPDATE Cascade;
-- ------------------------------------------------------------
-- CREATE LINK "public.Payment.orderPaymentFK" ----------------
ALTER TABLE "public"."Payment"
        ADD CONSTRAINT "orderPaymentFK" FOREIGN KEY ( "order_id" )
        REFERENCES "public"."Order" ( "order_id" ) MATCH FULL
        ON DELETE Cascade
        ON UPDATE Cascade;
-- ------------------------------------------------------------
-- CREATE LINK "public.Payment.paymentTypeFK" -----------------
ALTER TABLE "public"."Payment"
        ADD CONSTRAINT "paymentTypeFK" FOREIGN KEY ( "type_id" )
        REFERENCES "public"."PaymentMethod" ( "type_id" ) MATCH FULL
        ON DELETE Cascade
        ON UPDATE Cascade;
-- ------------------------------------------------------------
-- CREATE TABLE "Cart" ----------------------------------------
CREATE TABLE "public"."Cart" (
        "cart_id" Serial NOT NULL,
        "customer_id" Serial NOT NULL,
        "items" Integer DEFAULT 0 NOT NULL,
        "status" Character Varying NOT NULL,
        "total_price" Numeric DEFAULT 0,
        PRIMARY KEY ( "cart_id", "customer_id" ),
        CONSTRAINT "unique_Cart_cart_id" UNIQUE( "cart_id" ) );
;
-- ------------------------------------------------------------
```

```
-- CREATE LINK "public.Cart.CustFK" ---------------------------
ALTER TABLE "public"."Cart"
        ADD CONSTRAINT "CustFK" FOREIGN KEY ( "customer_id" )
        REFERENCES "public"."Customer" ( "user_id" ) MATCH FULL
        ON DELETE Cascade
        ON UPDATE Cascade;
-- -----------------------------------------------------------
-- CREATE TABLE "CartItem" ----------------------------------
CREATE TABLE "public"."CartItem" (
        "cart_id" Serial NOT NULL,
        "product_id" Serial NOT NULL,
        "added_at" Date NOT NULL,
        "shipping_id " Serial NOT NULL,
        "total_price " Numeric NOT NULL,
        PRIMARY KEY ( "cart_id", "product_id" ) );
;
-- -----------------------------------------------------------
-- CREATE LINK "public.CartItem.cartFK" ----------------------
ALTER TABLE "public"."CartItem"
        ADD CONSTRAINT "cartFK" FOREIGN KEY ( "cart_id" )
        REFERENCES "public"."Cart" ( "cart_id" ) MATCH FULL
        ON DELETE Cascade
        ON UPDATE Cascade;
-- -----------------------------------------------------------
-- CREATE LINK "public.CartItem.productCartFK" ----------------
ALTER TABLE "public"."CartItem"
        ADD CONSTRAINT "productCartFK" FOREIGN KEY ( "product_id" )
        REFERENCES "public"."Product" ( "product_id" ) MATCH FULL
        ON DELETE Cascade
```

```
        ON UPDATE Cascade;
-- ------------------------------------------------------------
-- CREATE TABLE "ShippingOptions" -----------------------------
CREATE TABLE "public"."ShippingOptions" (
        "shipping_id " Serial NOT NULL,
        "supplier_id" Serial NOT NULL,
        "shipper_id" Serial NOT NULL,
        "product_id " Serial NOT NULL,
        "shipping_price" Numeric NOT NULL,
        "country_code" Character( 3 ) NOT NULL,
        PRIMARY KEY ( "shipping_id ", "supplier_id", "shipper_id", "country_code" ) );
;
-- ------------------------------------------------------------
-- CREATE LINK "public.ShippingOptions.cntryFK" ----------------
ALTER TABLE "public"."ShippingOptions"
        ADD CONSTRAINT "cntryFK" FOREIGN KEY ( "country_code" )
        REFERENCES "public"."Country " ( "country_code" ) MATCH FULL
        ON DELETE Cascade
        ON UPDATE Cascade;
-- ------------------------------------------------------------
-- CREATE LINK "public.ShippingOptions.prdFK" ------------------
ALTER TABLE "public"."ShippingOptions"
        ADD CONSTRAINT "prdFK" FOREIGN KEY ( "product_id " )
        REFERENCES "public"."Product" ( "product_id" ) MATCH FULL
        ON DELETE Cascade
        ON UPDATE Cascade;
-- ------------------------------------------------------------
-- CREATE LINK "public.ShippingOptions.suppFK" -----------------
ALTER TABLE "public"."ShippingOptions"
```

```
        ADD CONSTRAINT "suppFK" FOREIGN KEY ( "supplier_id" )
        REFERENCES "public"."Supplier" ( "supplier_id" ) MATCH FULL
        ON DELETE Cascade
        ON UPDATE Cascade;
-- ------------------------------------------------------------
-- CREATE LINK "public.ShippingOptions.shipperFK" --------------
ALTER TABLE "public"."ShippingOptions"
        ADD CONSTRAINT "shipperFK" FOREIGN KEY ( "shipper_id" )
        REFERENCES "public"."Shipper" ( "shipper_id" ) MATCH FULL
        ON DELETE Cascade
        ON UPDATE Cascade;
-- ------------------------------------------------------------
-- CREATE TABLE "Shipper" ----------------------------------
CREATE TABLE "public"."Shipper" (
        "shipper_id" Serial NOT NULL,
        "CompanyName" Character Varying NOT NULL,
        "phone" Character Varying NOT NULL,
        PRIMARY KEY ( "shipper_id" ) );
;
-- ------------------------------------------------------------
```

Fonksiyonları oluşturmaya sağlayan SQL komutları

```
CREATE OR REPLACE FUNCTION public."setToZero"()
 RETURNS trigger
 LANGUAGE plpgsql
AS $function$
        BEGIN
          NEW."nmbr_products" := 0 ;
```

```
        RETURN NEW ;
        END;
        $function$
-------------------------------------------------------
CREATE OR REPLACE FUNCTION public."setChildCatsToZero"()
 RETURNS trigger
 LANGUAGE plpgsql
AS $function$
        BEGIN
                NEW."nmbr_categories" := 0 ;
                RETURN NEW ;
        END;
        $function$
-------------------------------------------------------
CREATE OR REPLACE FUNCTION public."setChildCatsToZero"()
 RETURNS trigger
 LANGUAGE plpgsql
AS $function$
        BEGIN
                NEW."nmbr_categories" := 0 ;
                RETURN NEW ;
        END;
        $function$
-------------------------------------------------------
CREATE OR REPLACE FUNCTION "public"."setPayment"( )
RETURNS Trigger
        AS $function$
        DECLARE
        ordId INT ;
```

```
            BEGIN
            ordId := ( SELECT "order_id" FROM "public"."Order" WHERE
"order_id"=NEW."order_id" )AS payId;
            UPDATE "public"."Order" SET "payment_id" = NEW."payment_id" WHERE
"order_id" = ordId ;
             UPDATE "public"."Order" SET "payment_date" = NEW."processed_at"
WHERE "order_id" = ordId ;
            RETURN NEW ;
           END;
           $function$
LANGUAGE plpgsql;
-------------------------------------------------------------
CREATE OR REPLACE FUNCTION public."calculateOrderPrice"()
 RETURNS trigger
 LANGUAGE plpgsql
AS $function$
            DECLARE
            prc INTEGER ;
   BEGIN
      prc = (SELECT SM FROM (SELECT SUM(oi."price") as SM, o."order_id" ord_id
FROM "public"."OrderItem" oi JOIN "public"."Order" o ON  o."order_id" =
oi."order_id" GROUP BY o."order_id") as TB  WHERE "ord_id" = NEW."order_id") AS
prc ;
      UPDATE "public"."Order" SET "total_price" = prc WHERE "order_id" =
NEW."order_id" ;
              RETURN NEW ;
           END;
           $function$
-------------------------------------------------------------
CREATE OR REPLACE FUNCTION "public"."calculatePriceItem"()
RETURNS Trigger
```

```
        AS $function$
         DECLARE
    BEGIN
NEW."price" := NEW."quantity" * (SELECT "UnitPrice" FROM "public"."Product"
WHERE "product_id" = NEW."product_id") - NEW."discount" + (SELECT
"shipping_price" FROM "public"."ShippingOptions" WHERE "shipping_id" =
NEW."shipping_id") ;
         RETURN NEW ;
        END;
        $function$
LANGUAGE plpgsql;
-----------------------------------------------------------

CREATE OR REPLACE FUNCTION "public"."setDate"( )
RETURNS Trigger
        AS $function$
         DECLARE BEGIN
         NEW."placed_at" := CURRENT_DATE ;
         RETURN NEW ;
        END;
        $function$
LANGUAGE plpgsql;
-----------------------------------------------------------

CREATE OR REPLACE FUNCTION "public"."setDatePayment"( )
RETURNS Trigger
        AS $function$
         DECLARE BEGIN
         NEW."processed_at" := CURRENT_DATE ;
         RETURN NEW ;
        END;
        $function$
```

LANGUAGE plpgsql;

Tetkileyicileri oluşturmaya sağlayan SQL komutları

CREATE TRIGGER "triggerPlacementDate" BEFORE INSERT ON public."Payment" FOR EACH ROW EXECUTE PROCEDURE "setDatePayment"()

CREATE TRIGGER "triggerPlacementDate" BEFORE INSERT ON public."Order" FOR EACH ROW EXECUTE PROCEDURE "setDate"()

CREATE TRIGGER "triggerPriceCalculate" BEFORE INSERT OR UPDATE ON "public"."OrderItem" FOR EACH ROW EXECUTE PROCEDURE "calculatePriceItem"()

CREATE TRIGGER "triggerOrder" AFTER INSERT OR DELETE OR UPDATE ON public."OrderItem" FOR EACH ROW EXECUTE PROCEDURE "calculateOrderPrice"()

CREATE TRIGGER "triggerPayment" AFTER INSERT ON public."Payment" FOR EACH ROW EXECUTE PROCEDURE "setPayment"()

CREATE TRIGGER "triggerProd" BEFORE INSERT ON public."Product" FOR EACH ROW EXECUTE PROCEDURE "setParentCategory"()

CREATE TRIGGER "triggerParent" BEFORE INSERT ON public."ParentCategory" FOR EACH ROW EXECUTE PROCEDURE "setChildCatsToZero"()

Tablolara veri girmeye sağlayan bazı SQL komutları

INSERT INTO "public"."Customer" ("name", "surname", "username", "email", "birthdate", "country", "state", "city", "phone", "password", "postalCode", "address")

VALUES (
'loujey','zinedine','loulou79','loujeyedinn68@gmail.com','15/02/1983','america','california','new york','+242844554','123456779loui',5234, 'street of potato '),

(
'lama','tejdin','lamou389','lamatjdin48@gmail.com','08/12/1963','america','florida','miami','+2152564764','ka3bamlewi',3529, 'street of cringe '),

(
'karim','hedi','agha2585','kimohedi50@gmail.com','08/01/2001','Azerbaijan','baku','baku','+213572564','zoro250kilo',3209, 'street of go '),

(

'sandy','aghayev','aghandy85','sandyaghayev0@gmail.com','28/11/1975','Azerbaijan','baku','baku','+9941556469','luffy5naruto',5209, 'street of washington '),

(
'fredi','sefi','fredis85','fredok90@gmail.com','26/12/1985','america','florida','miami','+3215564869','ka3bamlewi',10089, 'city of flowers'),

(
'salim','sta','salim24','salimsta60@gmail.com','24/02/1989','america','california','san diego','(555) 555-1234','sahfalablebi25',10001, '132, My street');

INSERT INTO "public"."PaymentMethod" ("name", "supplier_id")

VALUES ('CreditCard', 1),

('Paypal', 2),

('CreditCard', 3),

('CreditCard', 7);

INSERT INTO "public"."Shipper" ("CompanyName", "phone")

VALUES

('chinaPost', '+8741522145'),

('SingapourExpress' , '+8584152145');

INSERT INTO "public"."BillingAddress" ("customer_id", "billingAddress", "city", "country_code", "postalCode")

VALUES (11, '07 rue des jammins', 'Paris', 'FRA', '54321'),

(13, 'street of potato ', 'Chicago', 'USA', '98741'),

(14, 'street of cringe ', 'Washington', 'USA', '20145'),

(15, 'street of go ', 'Baku', 'AZE', '68541'),

(16, 'street of Washington ', 'Tunis', 'TUN', '7000');

INSERT INTO "public"."ShippingAddress"("customer_id", "shipAddress", "city", "country_code", "postalCode")

VALUES (11, '07 rue des jammins', 'Paris', 'FRA', '54321'),

(1, '07 rue des jammins', 'Paris', 'FRA', '54321'),

(13, 'street of potato ', 'Chicago', 'USA', '98741'),

(14, 'street of cringe ', 'Washington', 'USA', '20145'),

(15, 'street of go ', 'Baku', 'AZE', '68541'),

(16, 'street of Washington ', 'Tunis', 'TUN', '7000');

INSERT INTO "public"."Order" ("customer_id","supplier_id ", "billingAddress", "shippingAddress")

VALUES (1,2, 1, 2),

(1,2, 1, 2),

(11,2, 2, 1),

(13,3, 3, 3),

(14,2, 4, 4),

(15,7, 5, 5),

(13,3, 3, 3),

(16,3, 6, 6);

INSERT INTO "public"."ShippingOptions" ("supplier_id", "shipper_id", "product_id ", "shipping_price", "country_code")

VALUES (2, 1, 2, 14.5, 'USA'),

(2, 2, 2, 144.5, 'FRA'),

(2, 1, 4, 14.5, 'TUN'),

(3, 1, 5, 142.5, 'USA'),

(2, 2, 7, 14.5, 'AZE'),

(1, 1, 17, 14.5, 'FRA');

INSERT INTO "public"."OrderItem" ("order_id", "product_id", "price", "quantity", "size", "color", "shipping_id")

VALUES (9, 7, 250, 100, 10.2, 'red', 1),

(9, 4, 241, 100, 10.2, 'red', 1),

(10, 15, 28, 104, 12.2, 'blue', 1),

(10, 5, 158, 170, 10.2, 'green', 1),

(11, 6, 985, 170, 104, 'red', 1),

(11, 7, 145, 54, 1.2, 'grey', 1),

(12, 2, 298, 140, 12.2, 'blue', 1);

INSERT INTO "public"."OrderItem" ("order_id", "product_id", "price", "quantity", "size", "color", "shipping_id")

VALUES (10, 2, 14.5, 1, 14.5, 'purple', 1),

(11, 2, 14.5, 1, 14.5, 'red', 1),

(13, 4, 14.5, 1, 14.5, 'red', 3),

(14, 5, 14.5, 1, 14.5, 'purple', 4),

(15, 7, 14.5, 1, 14.5, 'purple', 5),

(16, 17, 14.5, 1, 14.5, 'green', 6),

(17, 2, 14.5, 1, 14.5, 'white', 2),

(18, 2, 14.5, 1, 14.5, 'white', 1),

(19, 2, 14.5, 1, 14.5, 'black', 1),

(20, 4, 14.5, 1, 14.5, 'black', 3),

(21, 7, 14.5, 1, 14.5, 'grey', 5),

(22, 17, 14.5, 1, 14.5, 'pink', 6),

(23, 2, 14.5, 1, 14.5, 'blue', 2),

(24, 2, 14.5, 1, 14.5, 'green', 1);

INSERT INTO "public"."CreditCard" ("customer_id", "cardNumber", "CVV", "expiration_date")

VALUES (1,'5131530665913726', '513', '2023-10-13'),

(11, '4024007104653315', '422', '2023-12-27'),

(12, '4532799623021991', '508', '2028-02-07'),

(13, '4024007120712467', '366', '2031-05-21'),

(14, '5388946431631667', '938', '2031-08-07'),

(15, '5339450679758878', '145', '2023-08-01'),

(16, '5113031906904634', '120', '2031-08-14');

INSERT INTO "public"."Payment" ("order_id", "card_id", "type_id", "state")

VALUES (9, 1, 1, 'processed'),

(10, 2, 1, 'processed'),

(11, 3, 1, 'processed'),

(13, 3, 1, 'processed');

INSERT INTO "public"."ParentCategory" ("name", "description", "nmbr_categories")

VALUES ('Woman', 'women articles', 4);

INSERT INTO "public"."ParentCategory" ("name", "description", "nmbr_categories")

VALUES ('Men', 'women articles', 4);

INSERT INTO "public"."ParentCategory" ("name", "description", "nmbr_categories")

VALUES ('House and living', 'House and living articles', 3);

INSERT INTO "public"."ParentCategory" ("name", "description", "nmbr_categories")

VALUES ('Electronics', 'Electronic articles', 5);

INSERT INTO "public"."ParentCategory" ("name", "description", "nmbr_categories")

VALUES ('supermarket', 'supermarket articles', 6);

INSERT INTO "public"."ParentCategory" ("name", "description", "nmbr_categories")

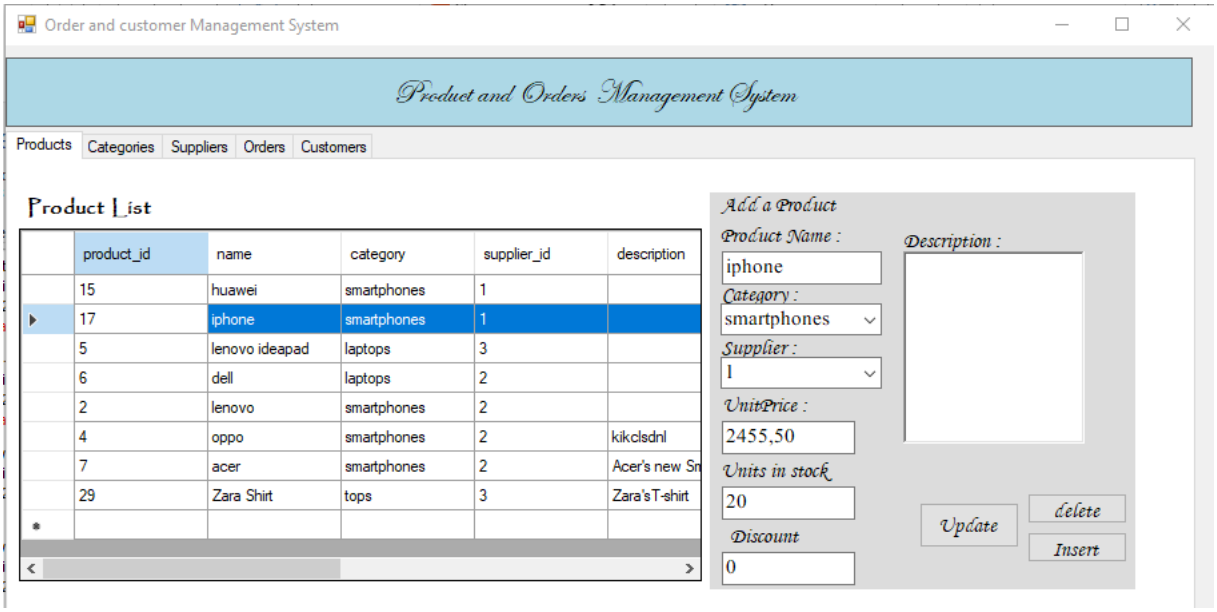
VALUES ('shoes and bags', 'shoes and bags', 2);

Veritabanını Yönetmeye sağlayan Uygulamanin çalışmasi

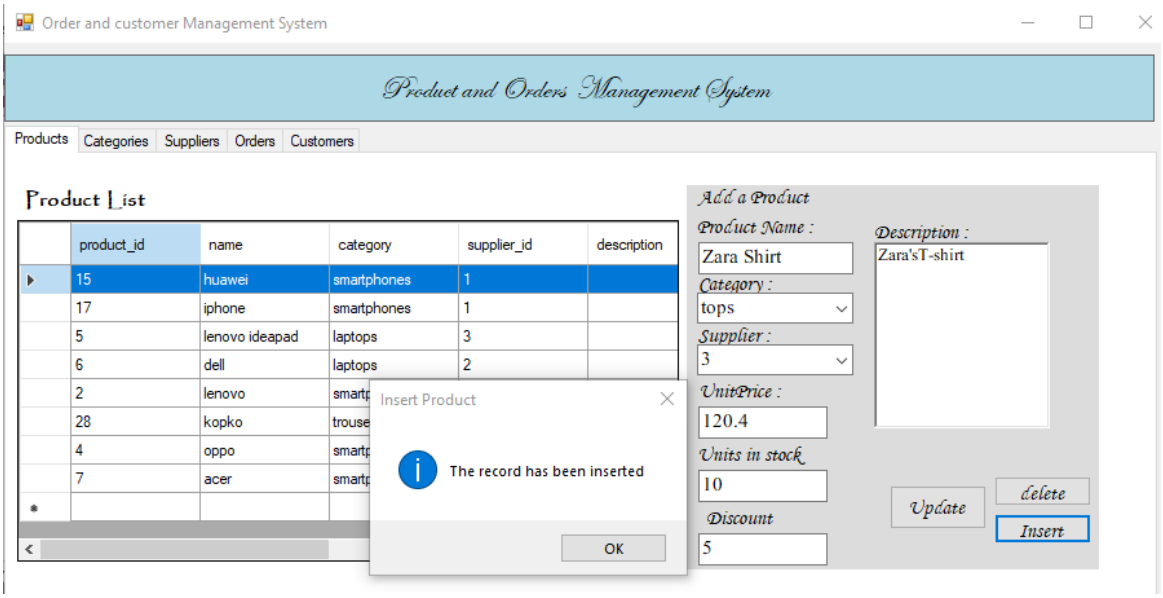
Tanıtımı : C# dilinde geliştirilen, Products, Categories, Sub-Categories, Suppliers, Orders ve Customers Yoneten 5 Pencereden oluşan bir Windows Forms CRUD Uygulamasi.

Şıklar (Pencereler / Tabs) :

1. **Produts :**



- SELECT işlemi : Bir urunu tikladiginizda sagdaki formunda seçtiginiz urunun bilgileri gorunmektedir



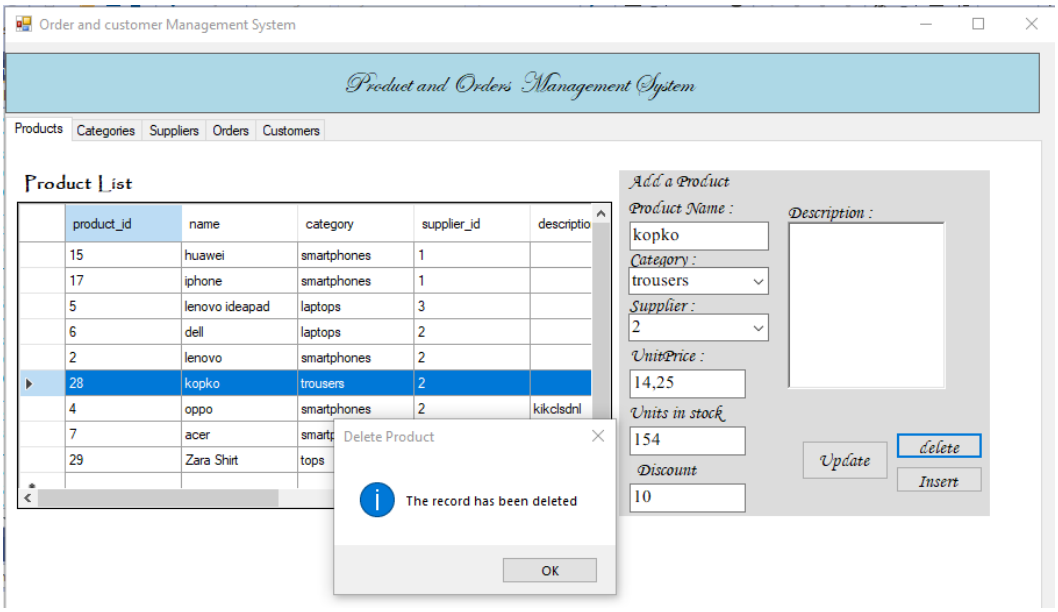
- INSERT işlemi : Formunu doldurtuktan sonra Insert tikladiginizda eklmek istediginiz urunu Product Listeye eklenir aşagidaki resimde

goruldugu gibi :

Product List

product_id	name	category	supplier_id	descriptio
15	huawei	smartphones	1	
17	iphone	smartphones	1	
5	lenovo ideapad	laptops	3	
6	dell	laptops	2	
2	lenovo	smartphones	2	
28	kopko	trousers	2	
4	oppo	smartphones	2	kikclsdnl
7	acer	smartphones	2	Acer's new
29	Zara Shirt	tops	3	Zara'sT-sh

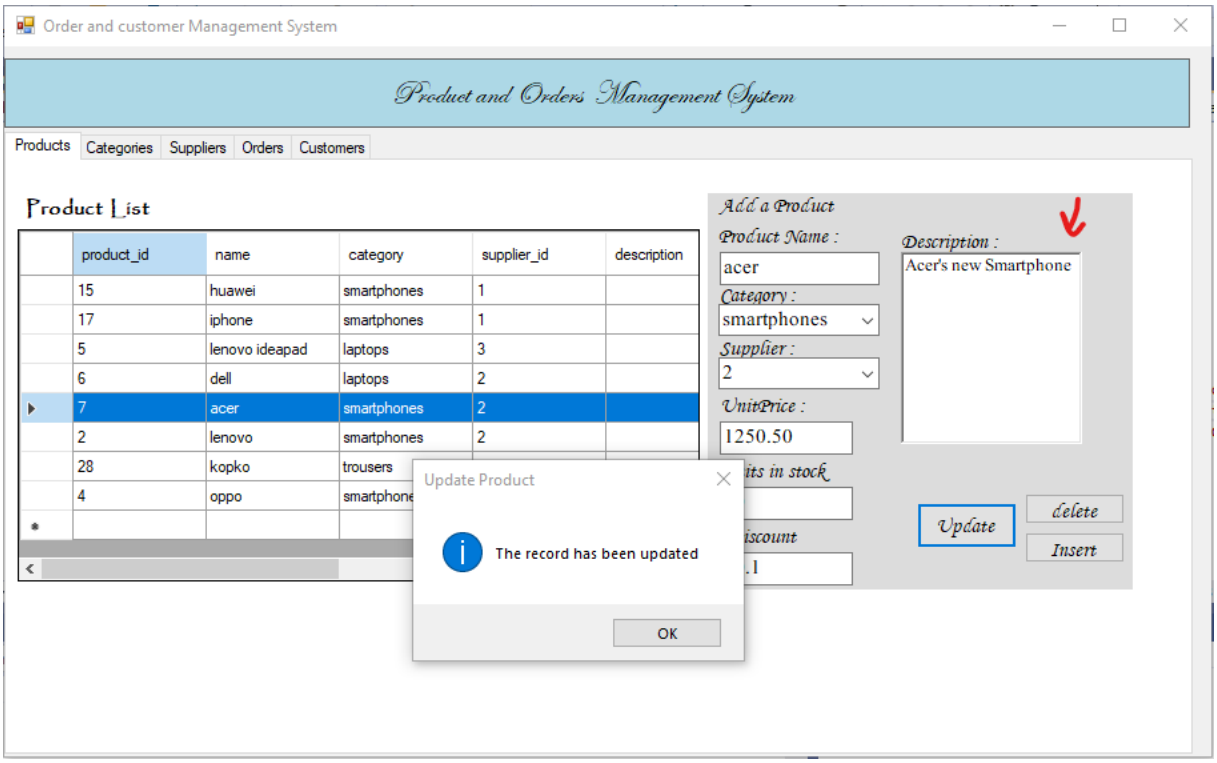
- DELETE işlemi : Listeden bir urunu seçip delete butona tikladiginizda urun silinir (Alttaki resimde goruldugu gibi sildikten sonra eleman listed gorumuyor)



Product List

product_id	name	category	supplier_id	description
15	huawei	smartphones	1	
17	iphone	smartphones	1	
5	lenovo ideapad	laptops	3	
6	dell	laptops	2	
2	lenovo	smartphones	2	
4	oppo	smartphones	2	kikclsdnl
7	acer	smartphones	2	Acer's new Sn
29	Zara Shirt	tops	3	Zara'sT-shirt

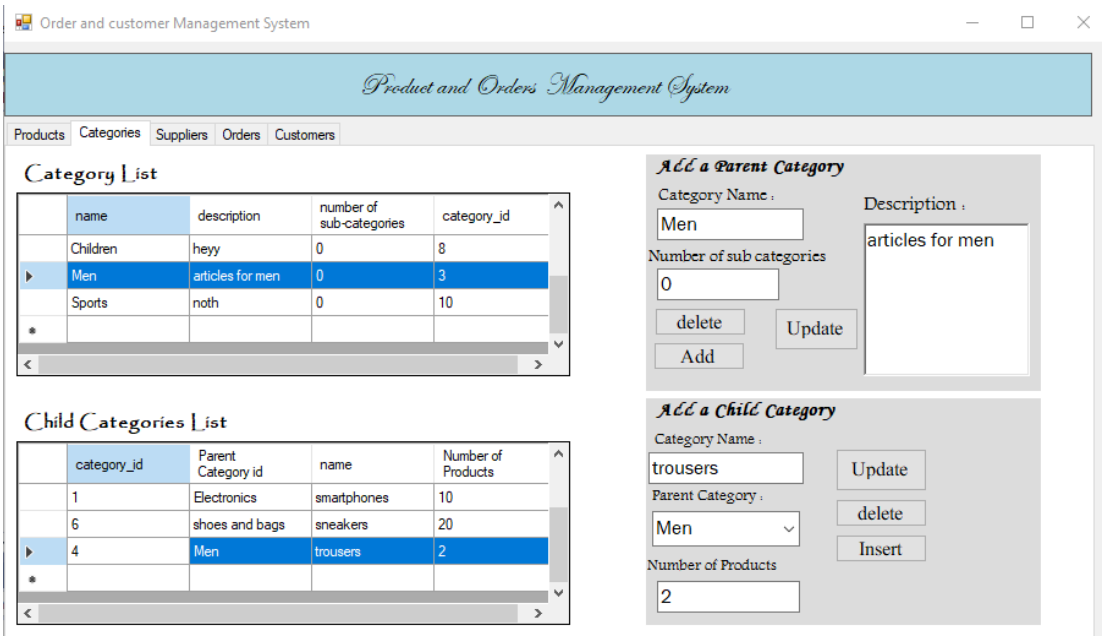
- UPDATE Işlemi : Listeden bir urunu seçip sagdaki formunda degistirmek istediginiz bilgileri girip Update tikladiginizda, guncelleme işlemi gerçekleştirebilirsiniz



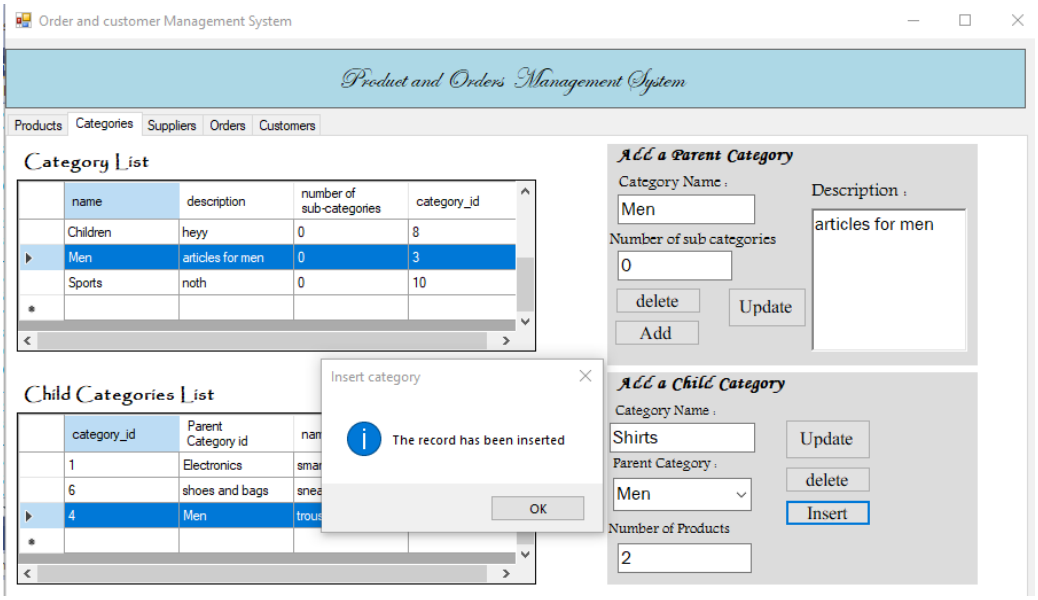
2. Categories :

Bu pencerenin altinda Categories ve Parent Categories Gorunmektektedir ve CRUD islemleri aynen Products penceresinde yapildigi gibi yapilabilmektedir

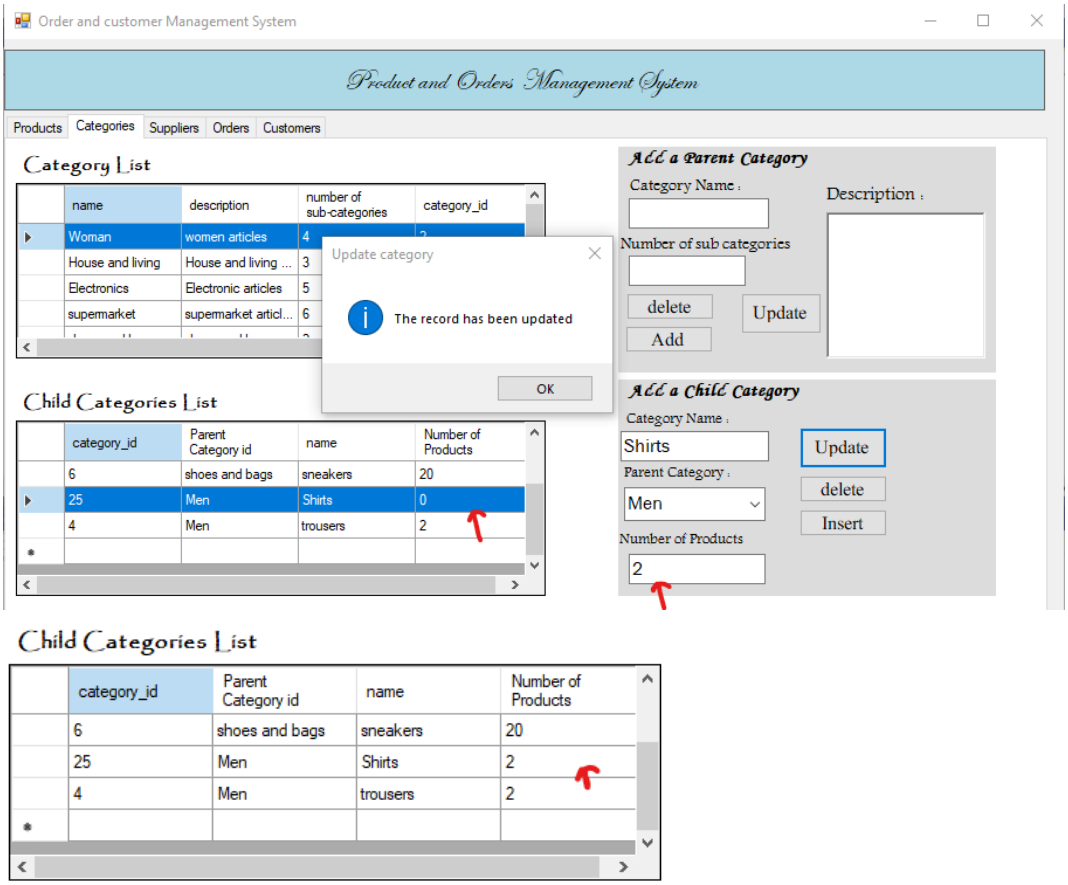
- SELECT işlemi :



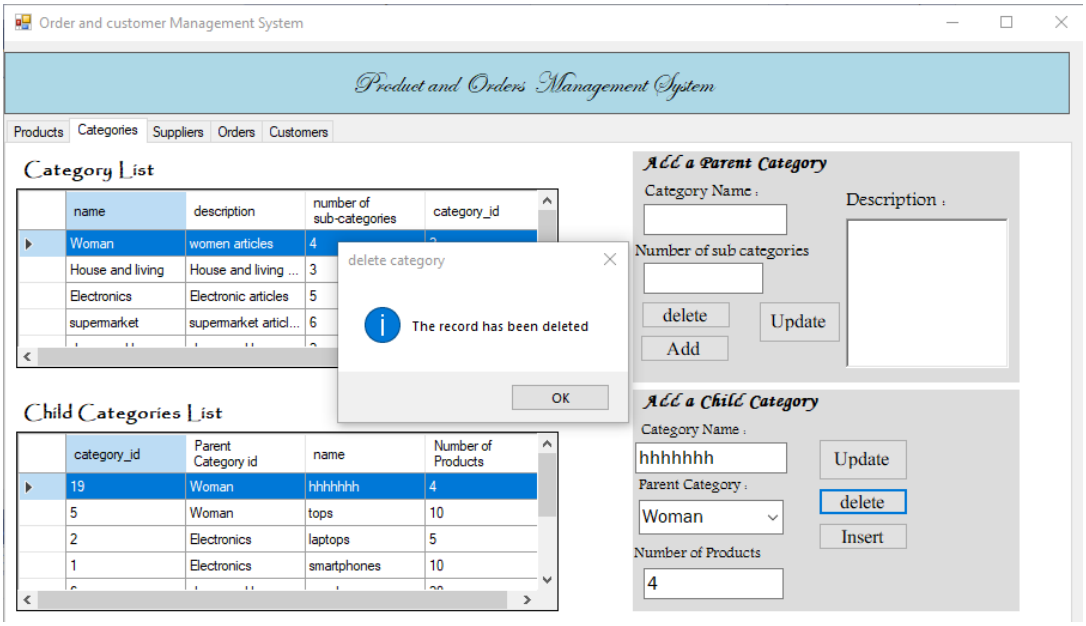
- INSERT işlemi : (Not : yeni bir Category veya Parent Category eklenilginde Child Sayisi(number of sub Categories ve number of Products 0 olarak tanimalnir)



- UPDATE Işlemi :



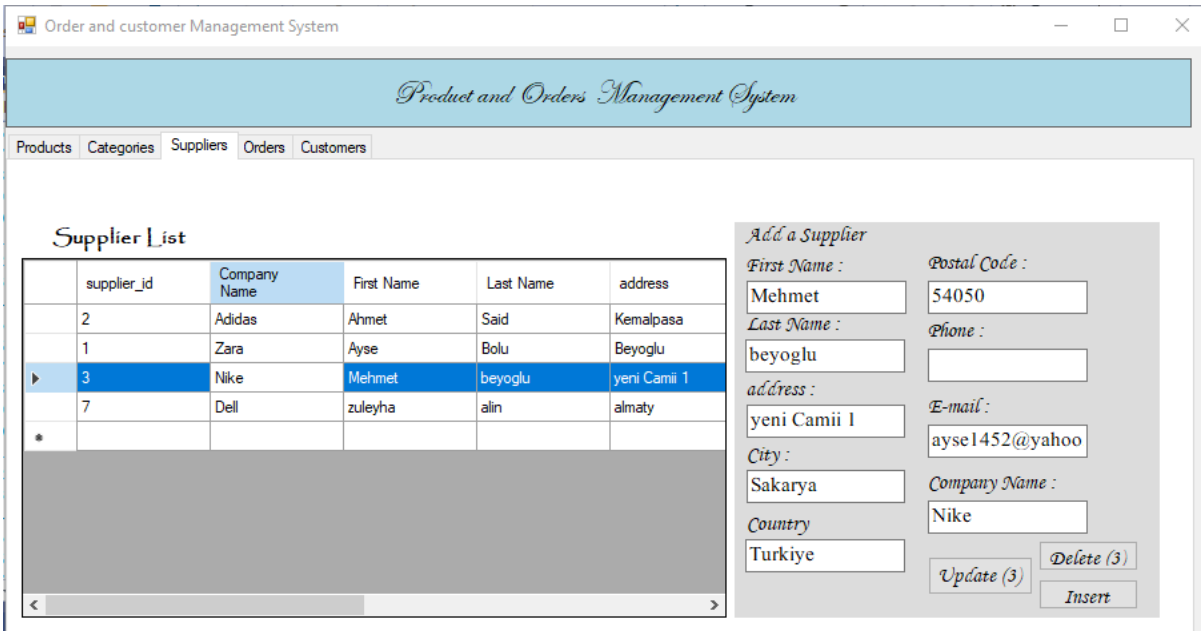
- DELETE işlemi :



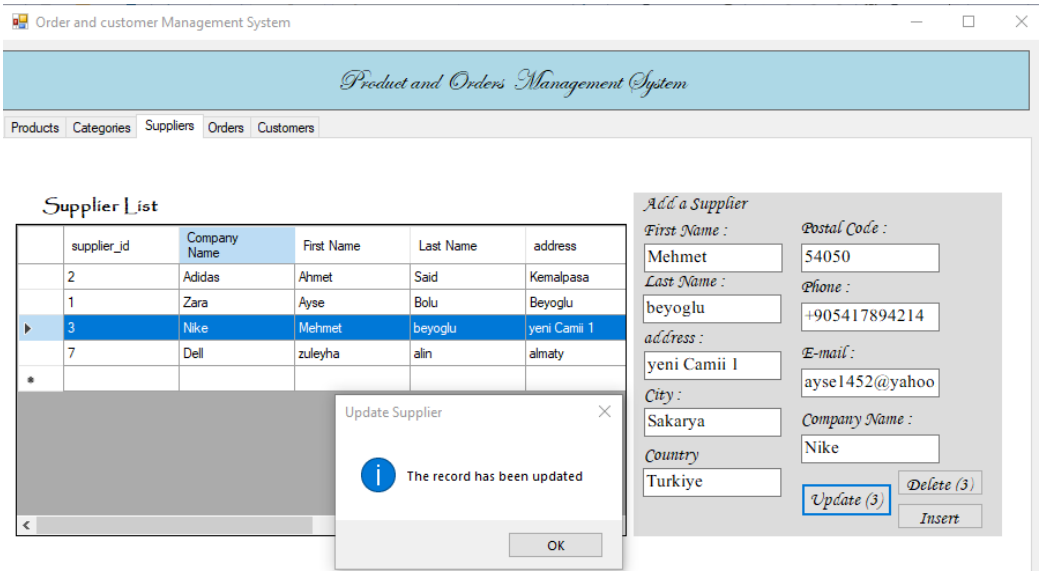
3. **Suppliers :**

Bu pencerenin altinda Suppliers Gorunmektektedir ve CRUD islemleri aynen Products penceresinde yapildigi gibi yapilabilmektedir

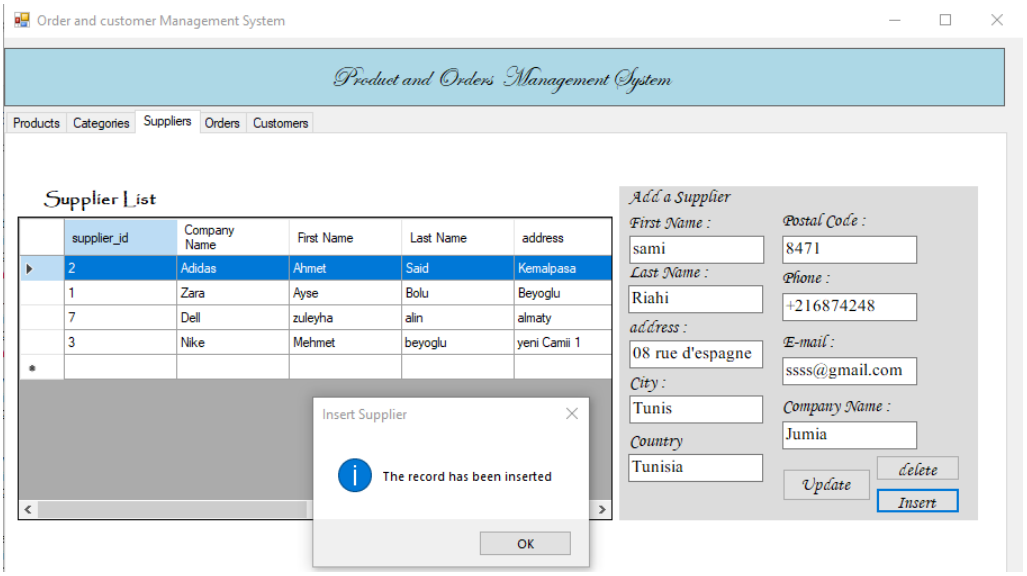
- SELECT işlemi :



- UPDATE Işlemi :

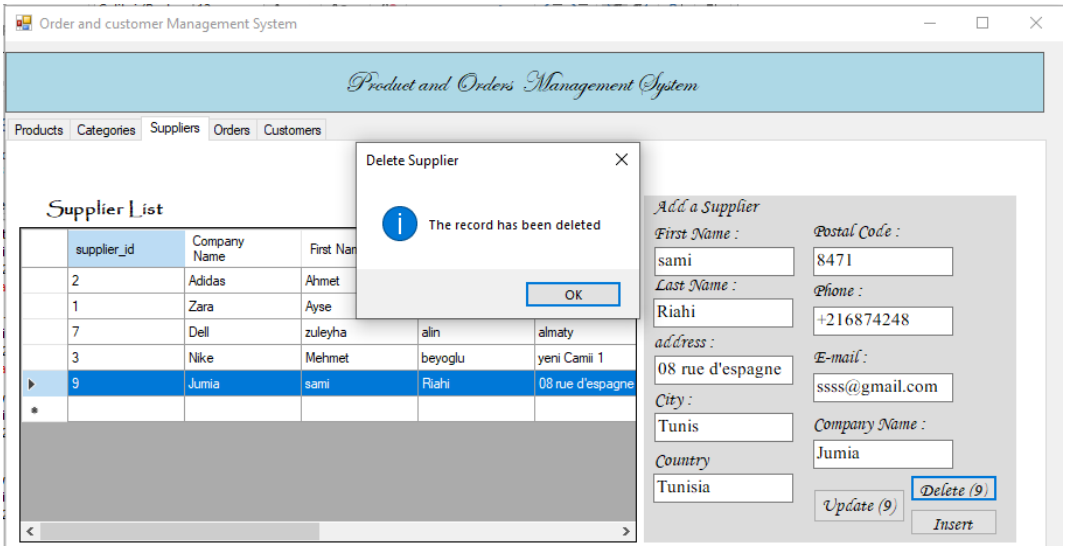


- INSERT işlemi :



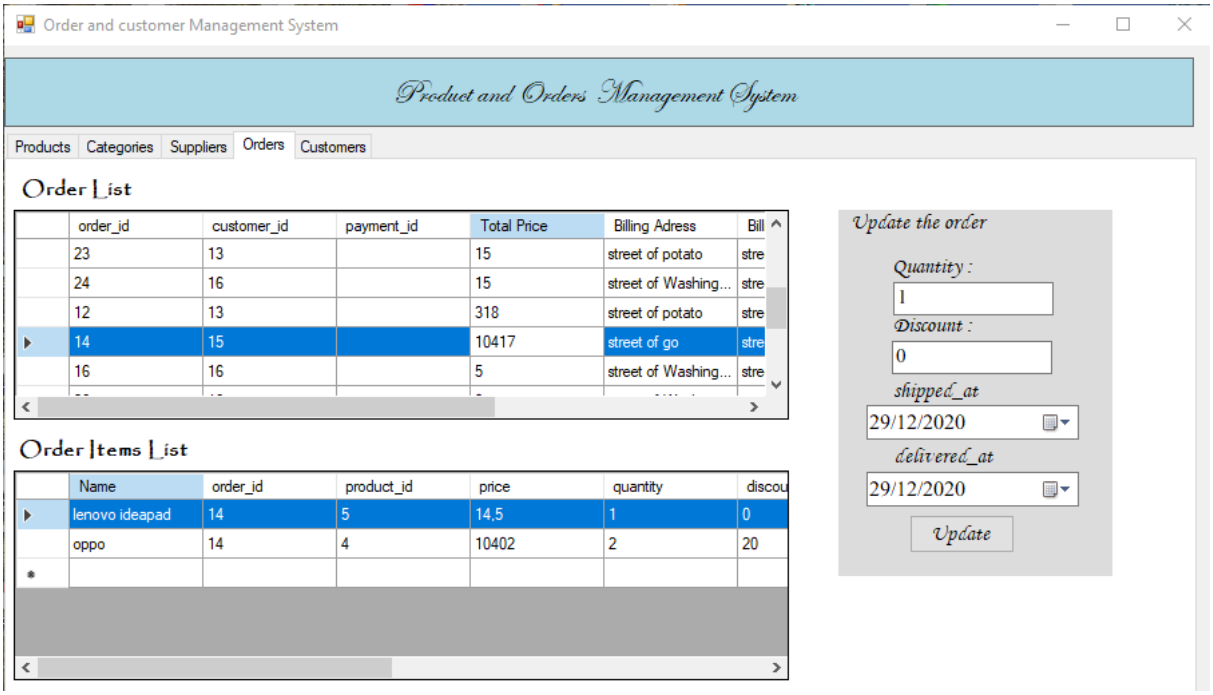
(Aşagidaki resimde yeni eklenen kayit gorunuyor)

- DELETE işlemi :

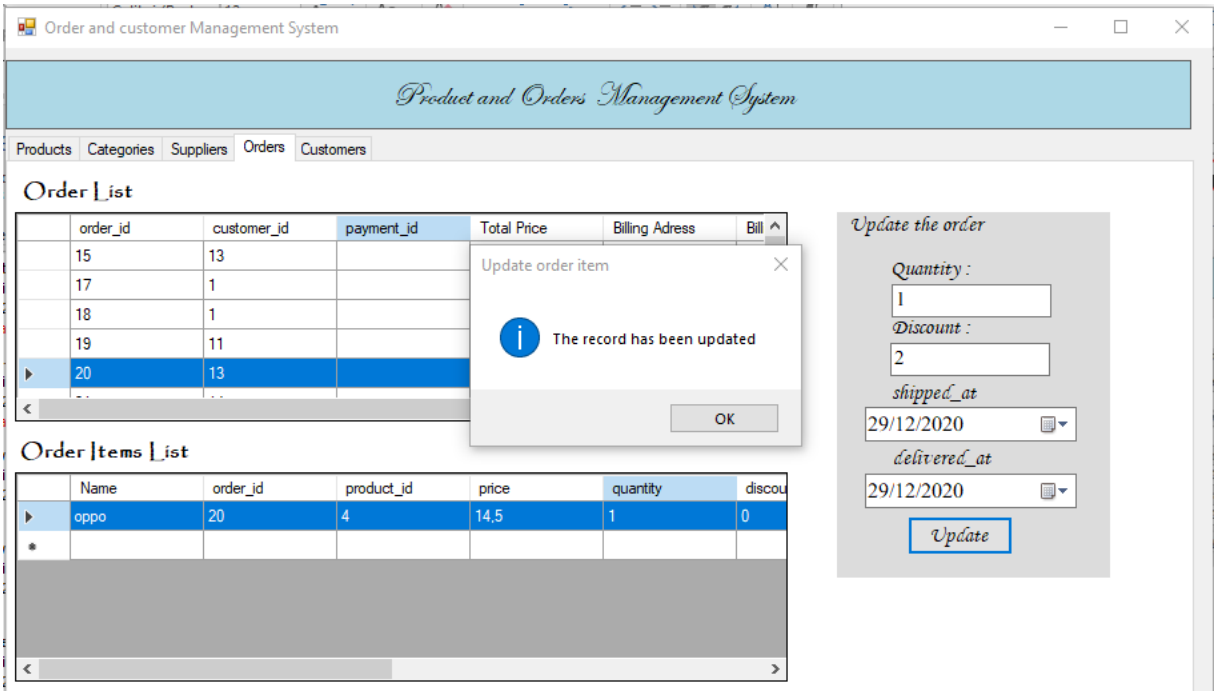


4. Orders

Bu pencerenin altinda Orders Gorunmektektedir ve her Order tikladiginizda Order'u olusturan butun itemleri yazdirilir ve bu itemlerin Quantity, Discount, shipped_at ve delivered_at bilgileri UPDATE yapilabilmektedir .



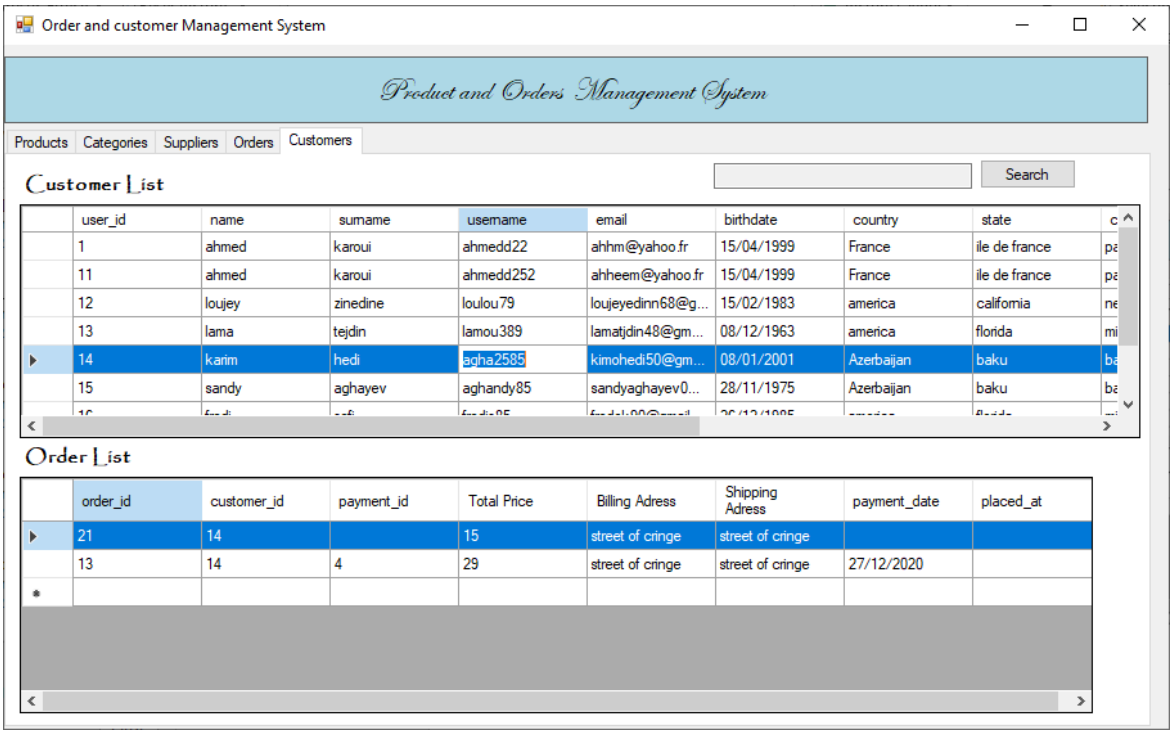
- UPDATE işlemi : Order Items Listesinden bir itemi seçip sagdaki formunda yeni bilgileri girip update tikladiginizda guncelleme islemi geçleştirebilirsiniz. **Guncelleme işlemi sirasindan price ve Total Price degerleri otomatik olarak guncellenir.**



5. Customers :

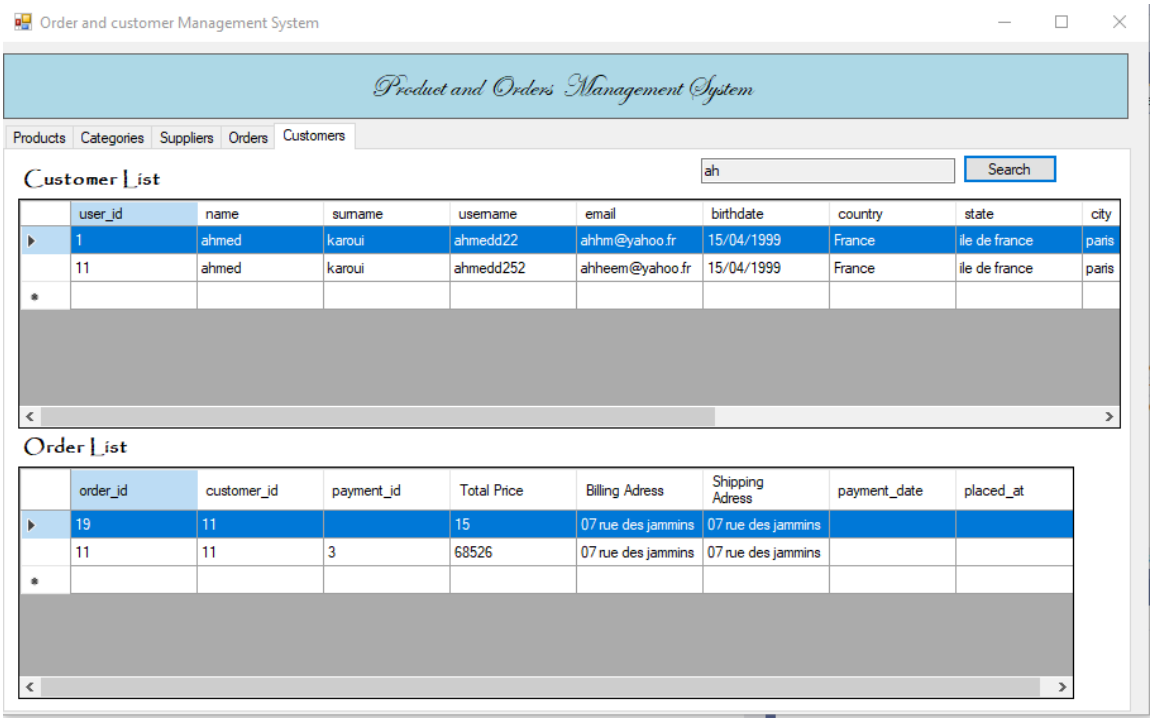
- SELECT işlemi :

  Bir customer'e tikladiginizda ona ait olan butun order'leri goruntulebilir



- Search Bar kullanarak SELECT işlemi :

  Search bar'a istediginiz isme sahip veya girdiginiz karakterlerle başlayan isimlere sahip Customers goruntulebilirsiniz (filtreyebilirsiniz)



Projenin Github Linki : https://github.com/hajergafsi/E-commerce-database-Mangement-System-with-windows-forms.git
Youtube Videonun Linki : https://youtu.be/7R_l2cU3XXc

Not: Yazdigim butun SQL komutlari Bir dump File da Github Repo'sunda bulabilirsiniz !

Ilginiz için Tesekkur Ederim ! ☺