DATABASE SYSTEMS

CSE355

PROJECT REPORT

UFUK GÜRBÜZ – 150113058

MUSTAFA YEMURAL – 150113053

PROCEDURES

1) PROJECT EXPLANATION

Project Name

" Robotics Products Trading (E-Commerce Area) "

Scope of The Project

The project is about robotics activities and products. We decided to work in e-commerce area and we chose robotics area for working. We will make a e-commerce website about robotics products.

Description

We will work onto e-commerce area about robotics products. We live in technology age. The world is developing with artifical intelligence, industrial activities and robotics breakthroughs. As a result of these, the new generation cares these areas. So, we thought about these subjects and decided to build a website about robotics products. The project will contain detailed table structures, multiple relationships and built-in tasks for database part("Ms Sql"). We will make website interface part using "Php", "Html", "Css", "JavaScript", "JQuery", "Bootstrap", "Photoshop" Technologies (and many assistant tools).

Project Steps

1) We arranged a meeting with a "e-commerce company". We will meet with them for obtaining information about "database (table structures, multiple relationships and built-in tasks)" and "website(Php,Html,Css,Js,JQuery,Bootstrap)" parts. So, we can make a more beautiful website.

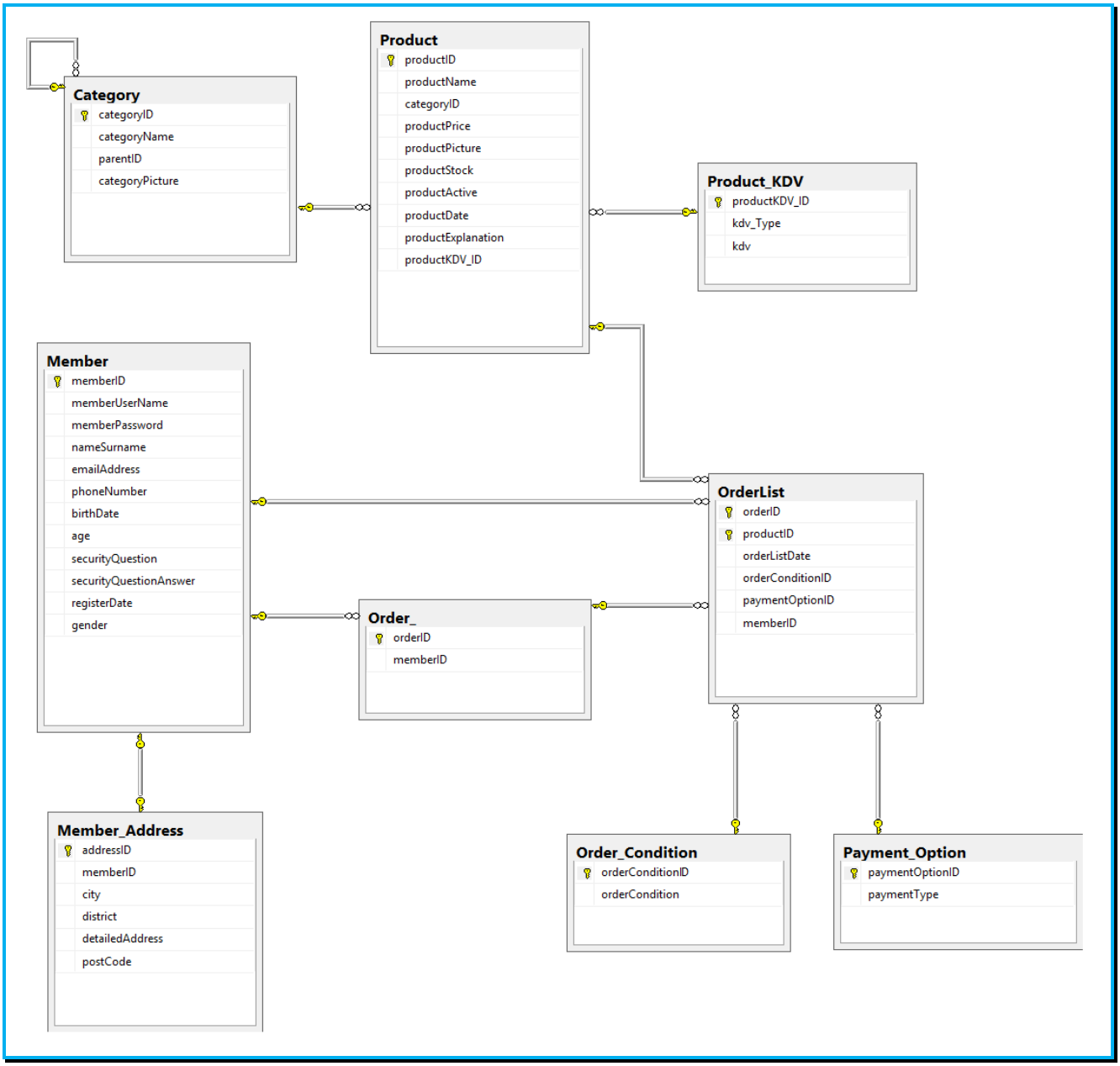
2) We will create the data tables according to king of products. Then, we will define between tables. After all, we will create many built-in tasks (predefined queries) for doing easier than table operations.

3) We finished database part. So, we can start website part. Firstly, we will decide design of website. Then, we will design front-end part of website using "Html", "Css", "Js", "JQuery", "Bootstrap". Finaly, we will connect website to database using "Php". So, we can take product information from database and we can show these information onto website.

2) UPDATED E/R DIAGRAMS

MAIN E/R DIAGRAM

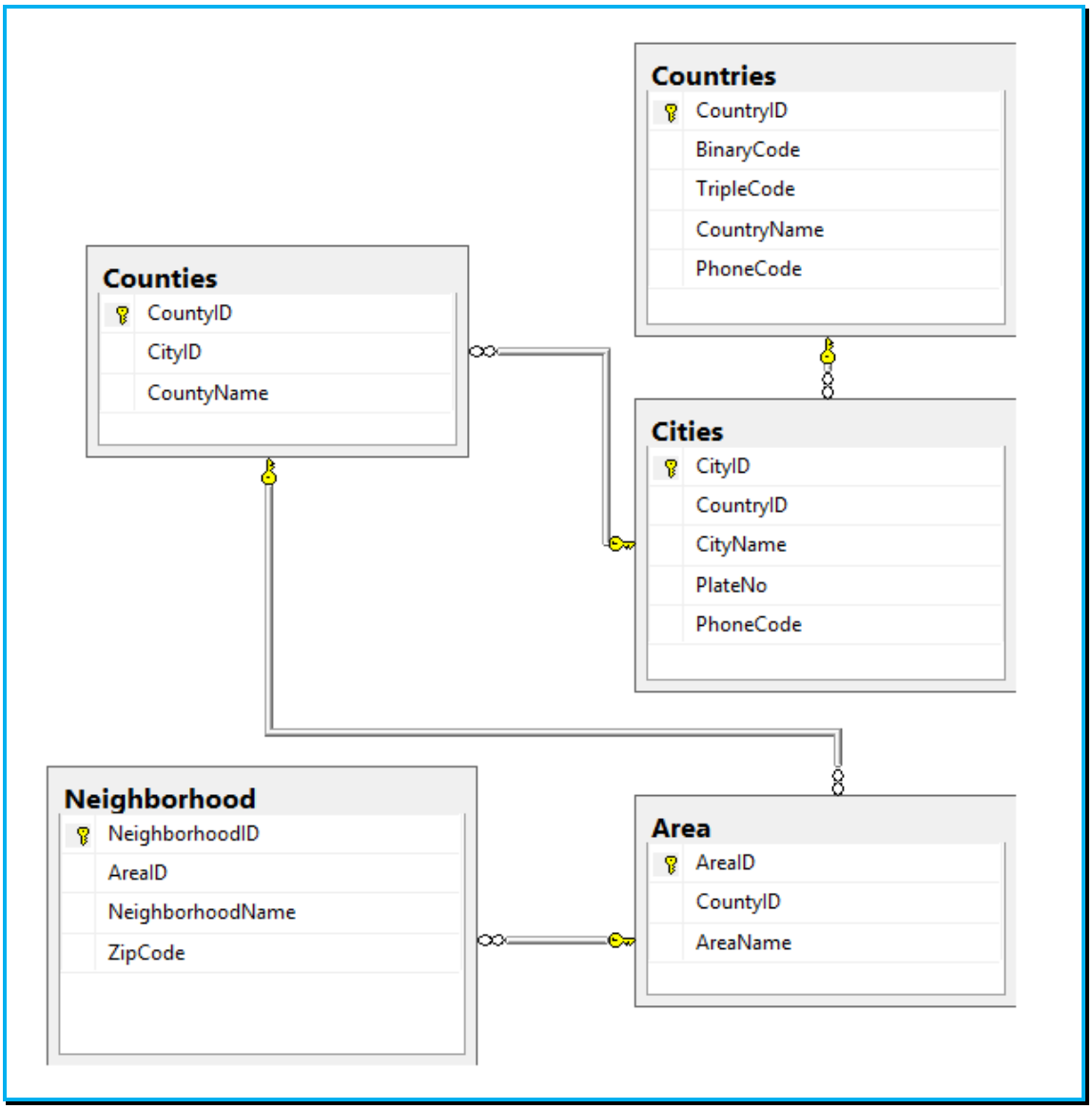
We updated our E/R diagram of e-commerce database. This diagram is main diagram which it contains all main tables.



NOTE: You can find high resolution picture in 'zip' file of project.

ADDRESS E/R DIAGRAM

We added a second E/R diagram into e-commerce database for keeping all address(city, county, area, neighborhood) in Turkey. This diagram is main diagram which it contains all main tables.



NOTE: You can find high resolution picture in 'zip' file of project.

3) THE TABLE DESCRIPTIONS AND REQUIREMENTS ANALYSIS

Member

It contains members that register the website. Every member can trade from this web site. If you give a order, you must register the site. Columns, data types and keys:

COLUMN NAME	DATA TYPE	DESCRIPTION
MemberID (PK)	bigint	It defines unique user ID.
memberUserName	nvarchar(15)	It defines username.
memberPassword	nvarchar(15)	It defines user password.
nameSurname	nvarchar(60)	It defines user name and surname
emailAddress	nvarchar(50)	It defines user e-mail address.
phoneNumber	float	It defines user's phone number.
birthDate	date	It define user's birthdate.
age		It defines user age and this is a calculated value.
securityQuestion	nvarchar(40)	It defines security question for user that keeps security.
sequrityQuestionAnswer	nvarchar(40)	It defines security answer for user that keeps security.
registerDate	date	It defines register date of the user to the website.
gender	varchar(7)	It defines user's gender.

Constraint:

- "memberID" column constrainted by 'PRIMARY KEY'.

- "memberUserName" and "emailAddress" columns constrainted by 'UNIQUE'.

- "registerDate" column constrainted by 'DEFAULT(GetDate())'.

- "gender" column constrainted by 'CHECK(gender IN ('M','F'))'.

Indice:

- "memberID"

Member_Address

This table keeps users' address information.

COLUMN NAME	DATA TYPE	DESCRIPTION
addressID (PK)	int	It defines unique address ID.
memberID (FK)	bigint	It defines unique member ID.
city	nvarchar(30)	It defines member city.
district	nvarchar(30)	It defines member district.
detailedAddress	nvarchar(100)	It defines member detailed address information.
postCode	nvarchar(20)	It defines post code of the district.

Constraint:

- “addressID” column constrainted by ‘PRIMARY KEY’.

- “memberID” column constrainted by ‘FOREIGN KEY’.

- “city” column constrainted by ‘DEFAULT('İstanbul')’.

- “postCode” column constrainted by ‘DEFAULT('34000')’.

Indice:

- “addressID” - “memberID”

Order

It bounds “Member” and “OrderList” tables.

COLUMN NAME	DATA TYPE	DESCRIPTION
orderID (PK)	İnt	It defines unique order ID.
memberID (FK)	bigint	It defines unique member ID.

Constraint:

- “orderID” column constrainted by ‘PRIMARY KEY’.

- “memberID” column constrainted by ‘FOREIGN KEY’.

Indice:

- “orderID” - “memberID”

OrderList

It keeps detailed order infos.

COLUMN NAME	DATA TYPE	DESCRIPTION
OrderID (PK)	int	It defines unique order ID
ProductID (PK)	int	It defines product.
orderListDate	date	It defines order date.
OrderConditionID (FK)	int	It defines order condition.
PaymentOptionID (FK)	int	It defines payment option.
MemberID (FK)	bignit	It defines related member.

Constraint:

- “orderID”, “productID” columns constrainted by ‘PRIMARY KEY’.

- “productID”, “orderConditionID”, “paymentOptionID”, “memberID” columns constrainted by ‘FOREIGN KEY’.

Indice:

- “orderID”, - “productID”, - “orderConditionID”, - “paymentOptionID”, - “memberID”

Order_Condition

It keeps order condition options.

COLUMN NAME	DATA TYPE	DESCRIPTION
orderConditionID (PK)	int	It defines unique identifier.
orderCondition	nvarchar(30)	It defines which order condition

Constraint:

- "orderConditionID" column constrainted by 'PRIMARY KEY'.

- "orderCondition" column constrainted by 'UNIQUE'.

Indice:

- "orderConditionID"

Payment_Option

It keeps payment options.

COLUMN NAME	DATA TYPE	DESCRIPTION
paymentOptionID (PK)	int	It defines unique identifier.
paymentType	nvarchar(20)	It defines which payment type.

Constraint:

- "paymentOptionID" column constrainted by 'PRIMARY KEY'.

- "paymentOption" column constrainted by 'UNIQUE'.

Indice:

- "paymentOptionID"

Product

It keeps detailed product informations. Products are heart of the e-commerce web site.

COLUMN NAME	DATA TYPE	DESCRIPTION
productID (PK)	int	It defines unique identifier.
productName	nvarchar(50)	It defines name of the product.
categoryID (FK)	int	It defines category of the product which belongs.
productPrice	nvarchar(10)	It defines price of the product.
productPicture	nvarchar(60)	It defines picture name of the product.
productStock	int	It defines number of stock of the product.
productActive	bit	It defines whether product is active or not.
productDate	date	It defines product addition date.
productExplanation	ntext	It defines product description.
productKDV_ID (FK)	int	It defines product KDV option.

Constraint:

- “productID” column constrainted by ‘PRIMARY KEY’.

- “categoryID”, “productKDV_ID” columns constrainted by ‘FOREIGN KEY’.

- “productStrock” column constrainted by ‘CHECK(productStock>=0)’.

-“productActive” column constrainted by ‘CHECK(productActive IN (0,1))’.

- “productDate” column constrainted by ‘DEFAULT(GetDate())’.

Indice:

- “productID”, - “categoryID”, - “productKDV_ID”

Product_KDV

It keeps KDV options.

COLUMN NAME	DATA TYPE	DESCRIPTION
productKDV_ID (PK)	int	It defines unique identifier.
kdv_Type	nvarchar(15)	It defines which KDV type.
kdv	int	It defines KDV value of the type.

Constraint:

- “productKDV_ID” column constrainted by ‘PRIMARY KEY’.

- “kdv_Type” column constrainted by ‘UNIQUE’.

Indice:

- “productKDV_ID”

Category

It keeps category types. Categories help user to find products easily.

COLUMN NAME	DATA TYPE	DESCRIPTION
categoryID (PK)	int	It defines unique identifier.
categoryName	nvarchar(40)	It defines category name.
parentID (FK)	int	It defines parent ID which specify to category ID.
categoryPicture	nvarchar(60)	It defines category picture name.

Constraint:

- “categoryID” column constrainted by ‘PRIMARY KEY’.

-“parentID” column constrainted by ‘FOREIGN KEY’, DEFAULT(0), CHECK(parentID>=0).

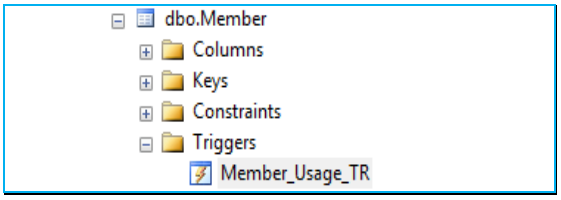
Indice:

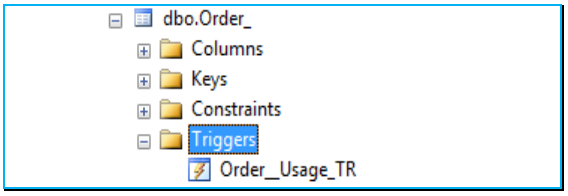
- “categoryID”, - “categoryName”, - “parentID”

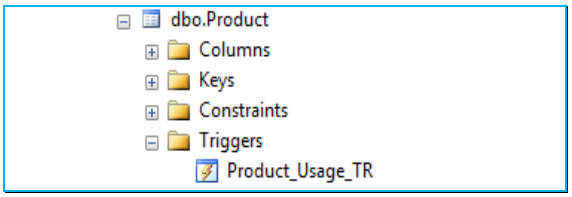
4) TRIGGER DESCRIPTIONS

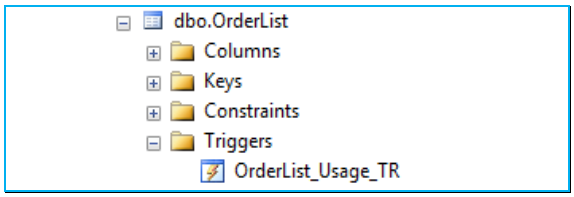
We created a "_DatabaseLog" table for keeping all transactions like 'INSERT', 'DELETE', 'UPDATE'. For this, we created triggers for each tables. When you did any 'insert', 'delete', 'update' operations, the triggers saves all operations into "_DatabaseLog" table.

Sample Triggers:







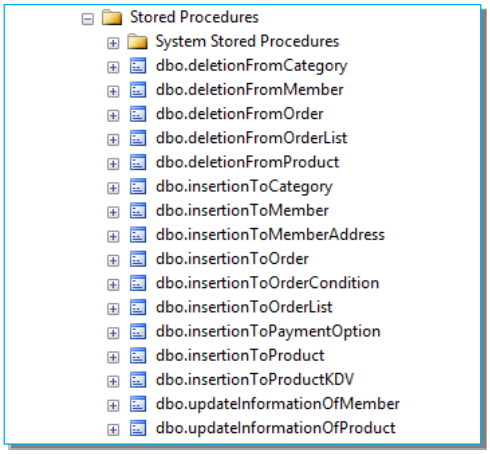


_DatabaseLog Table Logs:

	Log Date	Log Time	Affected Table Name	Activity
▶	2016-12-22	04:12:22	Area	DELETE
	2016-12-22	04:12:39	Area	INSERT
	2016-12-22	04:12:57	Area	UPDATE
	2016-12-22	04:12:57	Area	UPDATE
	2016-12-22	04:12:57	Area	UPDATE
	2016-12-22	04:12:57	Area	UPDATE
*	*NULL*	*NULL*	*NULL*	*NULL*

5) STORED PROCEDURE DESCRIPTIONS

- We created '16 stored procedures'. These procedures makes "insertion, deletion and update" operations for main important tables. So, we can did these operations easily. You can examine ".sql queries in zip file" as more detailed

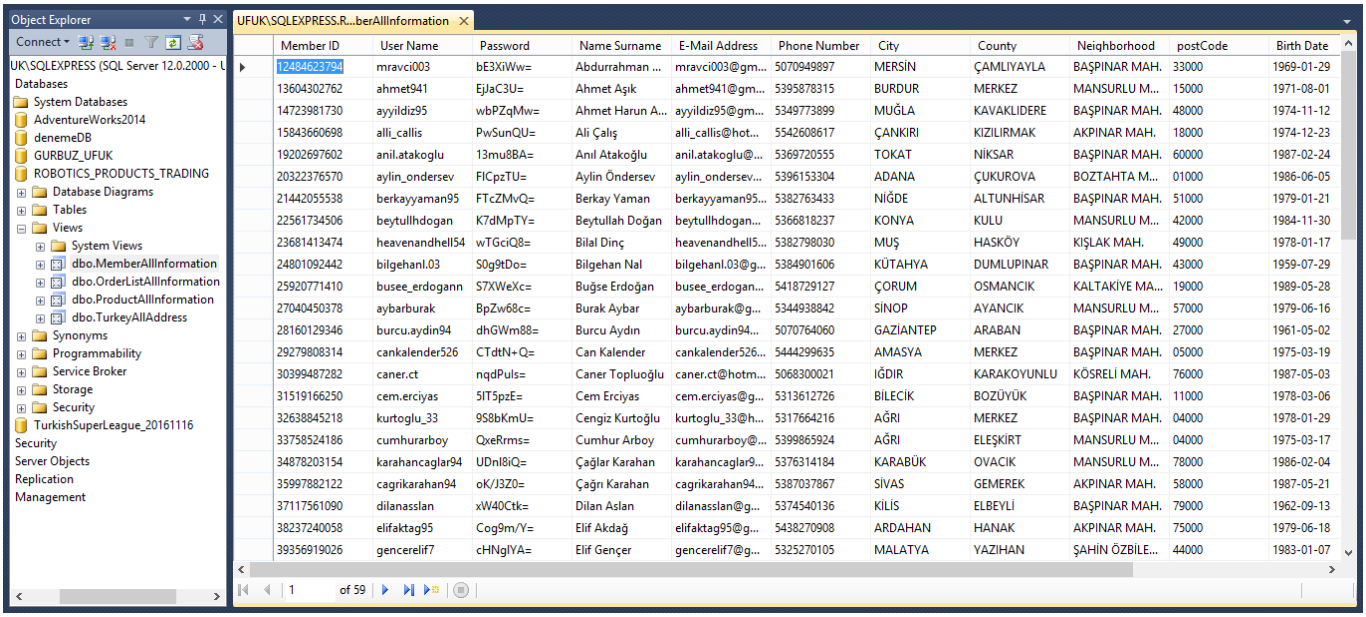


6) VIEW DESCRIPTIONS

- We created ‘4 views’. We combined some important tables using inner joins. So, we can take datas from website side. You can examine “.sql queries in zip file” as more detailed.

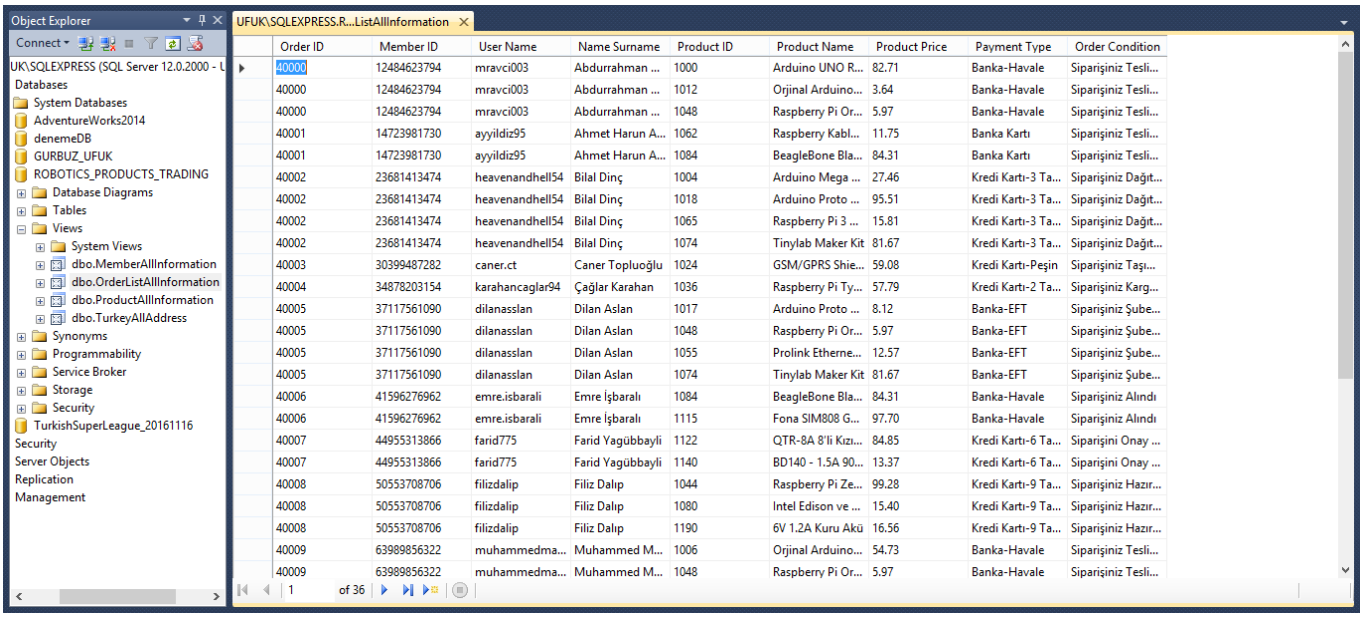
MemberAllInformation View

We combined “Member” and “Member_Address” tables for this view. So, we can see information of a Member as a single table.



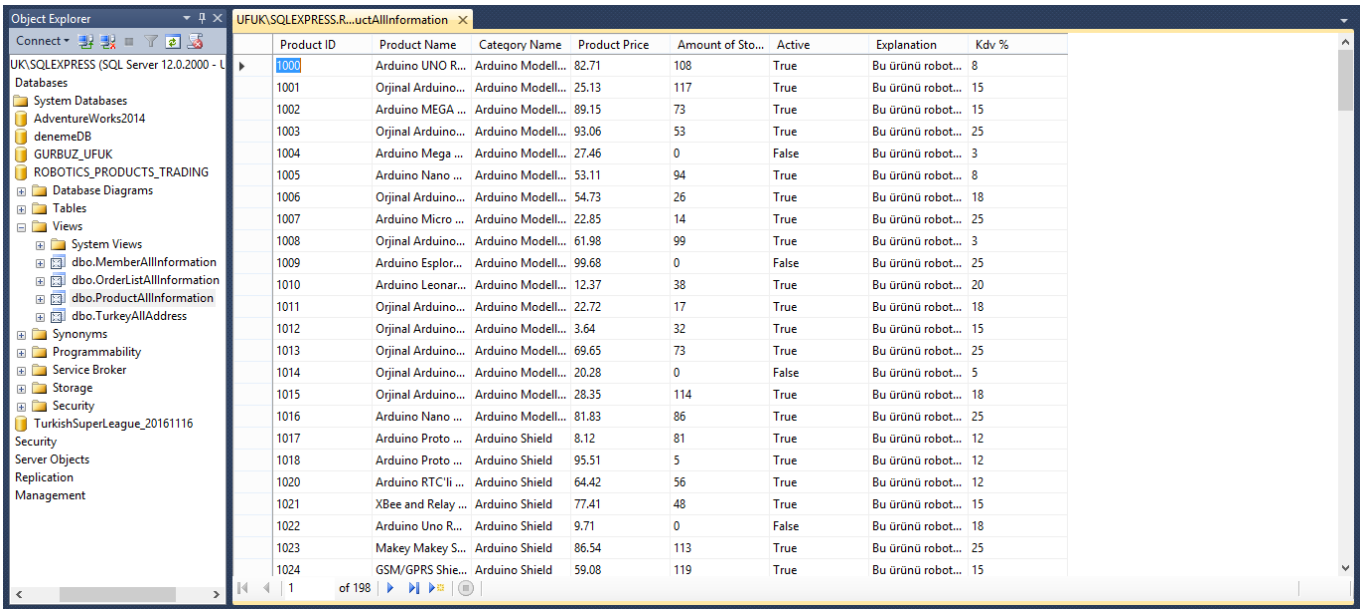
OrderListAllInformation View

We combined “OrderList”, “OrderCondition”, “Payment_Option”, “Product” and “Member” tables for this view. So, we can see all orders as a single table.



ProductAllInformation View

We combined “Product”, “Product_KDV” and “Category” tables for this view. So, we can see all information of products as a single table.



TurkeyAllAddress View

We combined “Cities”, “Counties”, “Area” and “Neighborhood” tables for this view. So, we can see all addresses in Turkey as a single table.