STEP 3 REPORT

**a) Project description: explain what your database project is about.**

We wanted to create Fritolay’s database. There are 3 main employee kind. RegionalManager> SalesChief>Salesman. This sorting is about management. Also there are products. We have taken the list of products from the person we talked to (They are using oracle). Actually we wanted to do a database of stock exchange but it was really hard. So we kept it simple. Customers make a list of products that they want and Salesmans make a list of products that they want. There are also warehouses where the warehouse employees work.

**b) Scope: what is included/exclude? Which processes are supported, which ones are not?**

We have deleted some of the entities from the previous step. Because they were just a bunch of useless data that we cant to anything with them. Actually they would be useful in real life to keep track of everything simultaneously. It would take weeks to do the real project i guess. To sum up we have kept what we can use and we deleted what is useless for the project.

**c) Tables**

i) Name of the fields/columns.

ii) Definition of the table.

iii) Data types of the fields.

iv) Information about indexes, primary key, foreign key.

v) Information about uniques, identity, check constraints, defaults, computed columns, if any.

vi) Information about triggers, if any.

**Customer Table**

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

Customers have these attributes. They have also a salesman who is providing chips to them. Age column is computed from birthdate. Phone number is unique and salesmanId is foreign key here. We have also indexed surname ascending in this table. Primary key is identity it increase as you enter new data.

**CustomerProductList Table**

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

This table is basically a ladder. It connects salesmanID and customerID to listCust table via customerProductListId. Nothing special here. We had to create this table in order to table associative entity (product list).

**ListCust Table**

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

This table keeps data of the products in a product list. There are 2 primary key(productID and customerProductListId) here because of associative entity. Also there are 2 foreign key one of them connects productID to product table and the other one connects customerProductListID to customerProductList table.

**Product Table**

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

In this table we keep data of products. Such as doritos(product) taco (kind) Süper (size) 107 (gr) 8 (price). We have taken all these from real person. There is also a constraint which is not so sensible but a products price can not be higher than 15 liras. Primary key is identity it increase as you enter new data.

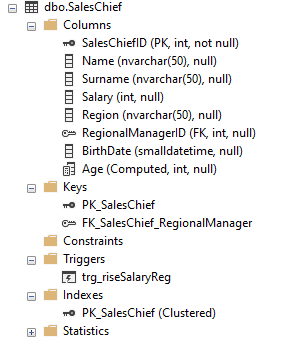
**RegionalManager Table**

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

Regional manager manages saleschief. There are some attributes above. Age is computed from birthdate. Primary key is identity it increase as you enter new data.

**SalesChief Table**



Sales chief is managed by regional manager and also sales chief manages salesmans. Age is computed. RegionalManagerID is foreign key which connects to regionalManager table. There is one trigger here which raises the salary of regionalManager after the insertion of saleschief. Primary key is identity it increase as you enter new data.

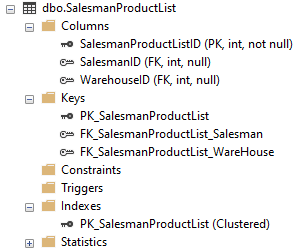
**Salesman Table**

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

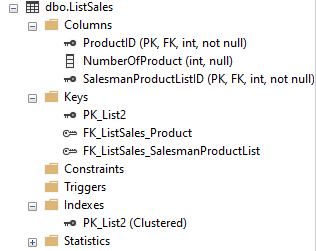
Salesman is managed by salesChief so there is foreign key about that. Primary key is identity it increase as you enter new data.

**SalesmanProductList**



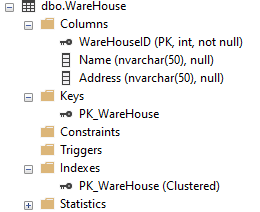
This table is similar to CustomerProductList but we have warehouse here. Salesman gets the products from warehouse. There are two foreign key which connects to related tables. Primary key is identity it increase as you enter new data.

**ListSales Table**



This table is created to keep track of products taken from warehouses by salesmans. There are 2 primary key because of associative entity.

**WareHouse Table**



This table keeps data of warehouses around Turkey. Primary key is identity it increase as you enter new data.

**WareHouseWorker Table**

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

These employees have shifthours. They have salary of asgari ücret. As they work more shifhour increases and so the salary. We have set default salary to 4250. And additionaly shifthours\*10 is added to their salary. Foreign key connects to warehouse table. Primary key is identity it increase as you enter new data.

**d) Views**

**e) Triggers**

**i) Name of the trigger.**

trg\_riseSalaryReg

**ii) Definition, and when/how it works.**

This trigger works when a new saleschief is inserted. We basically increased the salary of regional manager. If there is more saleschief to be managed by regional managers than they should earn more money.

**iii) Screenshot of the code of each trigger, and screenshots of before and after states of the table data that each**

**trigger works on.**

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

Before



After the insertion of a new salesChief

tablo içeren bir resim

Açıklama otomatik olarak oluşturuldu

**f) Stored procedures**

**i) Name of the store procedure.**

sp\_averageAge

**ii) Definition.**

This stored procedure calculates the average age of the salesmans grouped by salesChiefs. If saleschief has only one salesman than procedure does not calculate avg of age.

**iii) Screenshots of the code of each stored procedure, and screenshots of before and after states of the data that each stored procedure works on.**

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

tablo içeren bir resim

Açıklama otomatik olarak oluşturuldu

**i) Name of the store procedure.**

sp\_CalcListTotal

**ii) Definition.**

This sp calculates the total price of the product list that customer has with given listID

**iii) Screenshots of the code of each stored procedure, and screenshots of before and after states of the data that each stored procedure works on.**

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu



**i) Name of the store procedure.**

sp\_CalcTotalPayment

**ii) Definition.**

This sp calculates the total price of the all lists that customer has with the given customerID

**iii) Screenshots of the code of each stored procedure, and screenshots of before and after states of the data that each stored procedure works on.**

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu



**i) Name of the store procedure.**

sp\_deleteProduct

**ii) Definition.**

This sp deletes a product group from list with the given produc name.

**iii) Screenshots of the code of each stored procedure, and screenshots of before and after states of the data that each stored procedure works on.**

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

**Before the deletion**

tablo içeren bir resim

Açıklama otomatik olarak oluşturuldu

**after the deletion of cheetos**

tablo içeren bir resim

Açıklama otomatik olarak oluşturuldu

**i) Name of the store procedure.**

sp\_empOfMonth

**ii) Definition.**

This sp shows top 3 employee who worked the most around warehouses workers calculated by shifthours with the given warehouseid.

**iii) Screenshots of the code of each stored procedure, and screenshots of before and after states of the data that each stored procedure works on.**

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

tablo içeren bir resim

Açıklama otomatik olarak oluşturuldu

**i) Name of the store procedure.**

sp\_insertNewSalesman

**ii) Definition.**

This sp inserts a new Salesman to table with the given parameters.

**iii) Screenshots of the code of each stored procedure, and screenshots of before and after states of the data that each stored procedure works on.**

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

tablo içeren bir resim

Açıklama otomatik olarak oluşturuldu tablo içeren bir resim

Açıklama otomatik olarak oluşturuldu

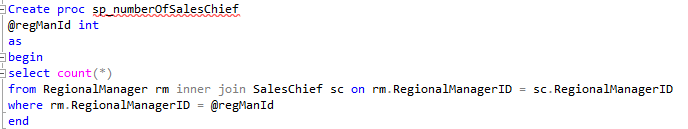
**i) Name of the store procedure.**

sp\_numberOfSalesChief

**ii) Definition.**

This sp calculates the number of salesChiefs that a regional manager manages with the given regionalManagerID input.

**iii) Screenshots of the code of each stored procedure, and screenshots of before and after states of the data that each stored procedure works on.**





**i) Name of the store procedure.**

sp\_riseSalarySalesman

**ii) Definition.**

This sp increases the salary of a salesman with the if else statements with the amount of percentages. SalesmanId is input parameter.

**iii) Screenshots of the code of each stored procedure, and screenshots of before and after states of the data that each stored procedure works on.**

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

metin, dolap, ekran görüntüsü içeren bir resim

Açıklama otomatik olarak oluşturuldu

metin, dolap, ekran görüntüsü içeren bir resim

Açıklama otomatik olarak oluşturuldu

**i) Name of the store procedure.**

sp\_updateNoOfProductInList

**ii) Definition.**

This sp updates the number of a product in a list with the input parameters.

**iii) Screenshots of the code of each stored procedure, and screenshots of before and after states of the data that each stored procedure works on.**

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

**i) Name of the store procedure.**

sp\_updateProductPrice

**ii) Definition.**

This sp updates the price of a product with the given percentage.

**iii) Screenshots of the code of each stored procedure, and screenshots of before and after states of the data that each stored procedure works on.**

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

**DIAGRAM OF WHOLE DATABASE**

