02.01.2014

EGE ÜNİVERSİTESİ BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

Automotive Sales Automation Database

***DATABASE MANAGEMENT HOMEWORK***

***GRUP ÜYELERİ:***

***OMER SALAJ 05110000798***

***A.AMMAR KARCIOĞLU 05100000052***

***AHMET ERCAN ÜNAL 05130001212***

*İÇİNDEKİLER*

[DIAGRAMS OF DATABASE MODEL 3](#_Toc344319528)

[DIAGRAM: 3](#_Toc344319529)

GELİŞTİRİLMİŞ MODELDE TABLOLAR VE ÖZELLİKLERİ ………………………………………………………………………4

AOUTOMOTİVE SALES AUTOMATION DATABASE……………………………………………………………………………….8

FIRST STEP OF DATABASE MODEL………………………………………………………………………………………………..8

TABLES

CAR&OPTIONS TABLES……………………………………………………………………………………………………………8

SALE&SALESPERSONS TABLES………………………………………………………………………………………………..9

CREATE TABLE SQLs OF FİRST STEP MODEL………………………..…………………………………………………10

INSERTS-UPDATES-DELETES OF FİRST STEP MODEL………………………………..…………………………….11

QUERİES OF FİRST STEP MODEL…………………………………………………………………………………………….12

SECOND STEP OF DATABASE……………………………………………………………………………………………………..14

TABLES

CAR&COSTUMERS&MODELS&OPTIONS TABLES…………………………………………………………………..14

PACKAGE&SALE&SALESPERSONS TABLES…………………………………………………………………………….15

SELECTEDOPTIONS&SELECTEDPACKAGES TABLES………………………………………………………………..16

CREATE TABLE SQLs OF SECOND STEP MODEL……………………………………………………………………..17

TRIGGERS OF SECOND STEP MODEL……………………………………………………………………………………..19

INSERTS-UPDATES-DELETES OF FİRST STEP MODEL………………………………………………………………23

QUERİES OF FİRST STEP MODEL……………………………………………………………………………………………26

DOĞAL DİL CÜMLELERİYLE GELİŞTİRİLMİŞ MODEL……………………………………………………………………………27

**GÖREV DAĞILIMI:**

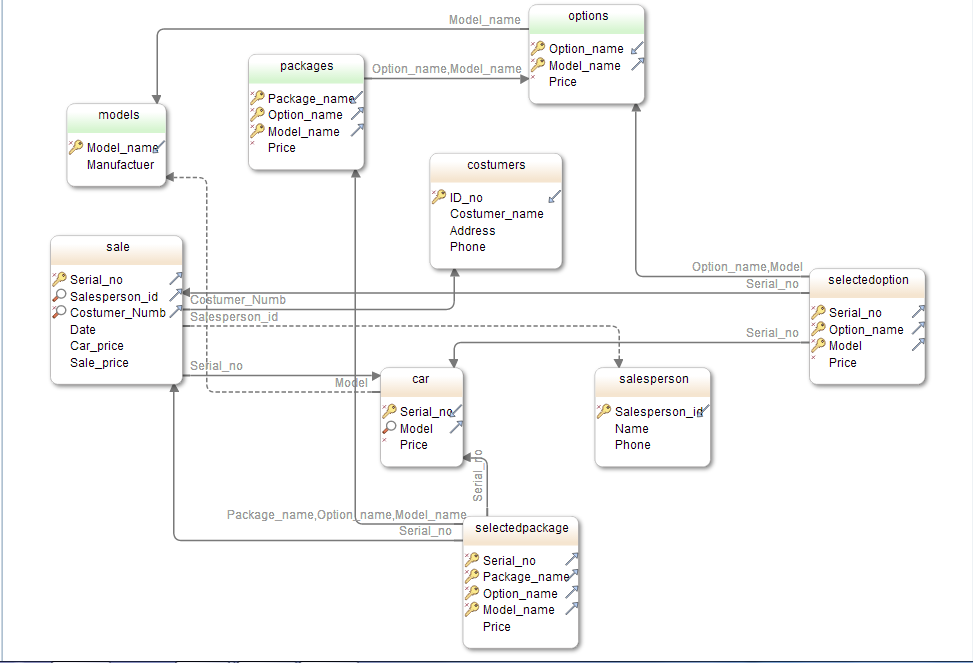
**MODEL TASARIMI**: A.AMMAR KARCIOĞLU-OMER SALAJ-AHMET ERCAN ÜNAL

**FİRST STEP OF MODEL & CONTROLS**: OMER SALAJ

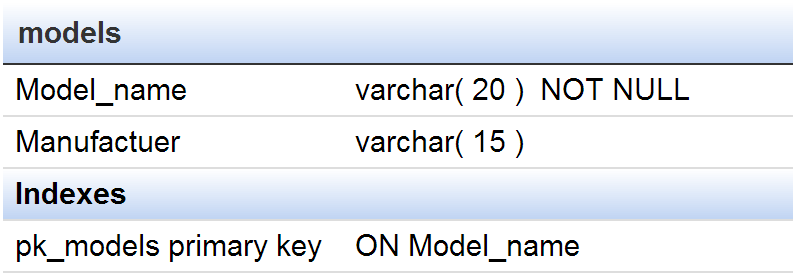
**QUERİES & REPORT**: A.AMMAR KARCIOĞLU

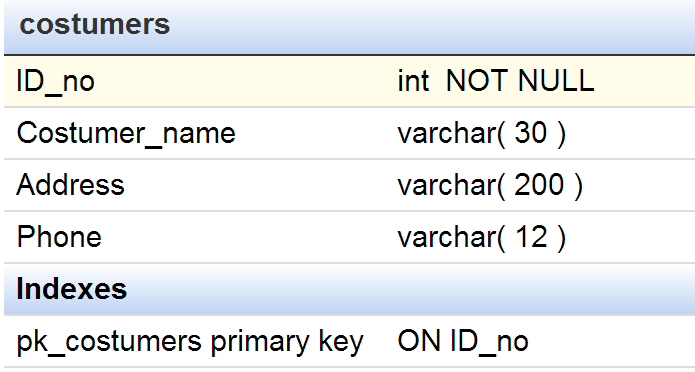
**SECOND STEP OF MODEL SQLs & TRIGGERS**: AHMET ERCAN ÜNAL

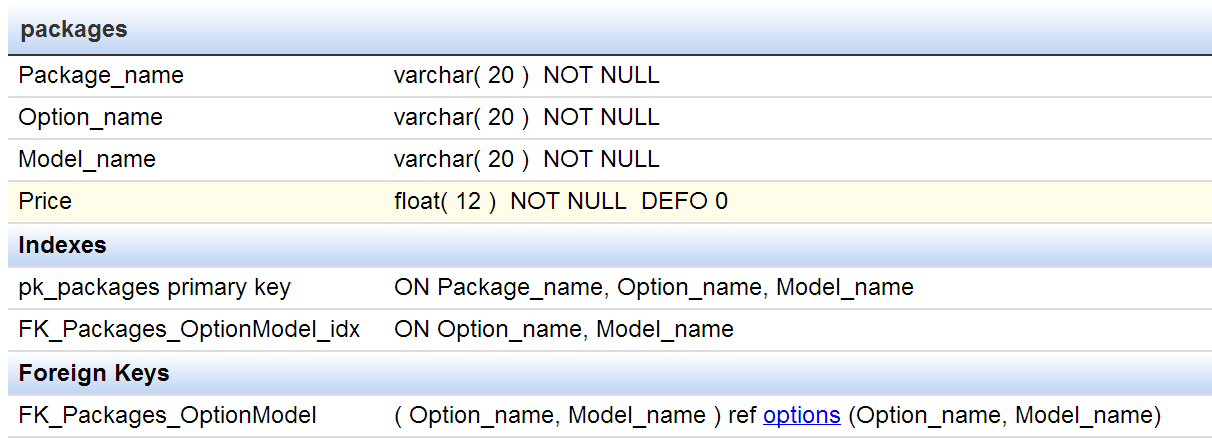
***DİAGARAMS OF DATABASE MODEL***

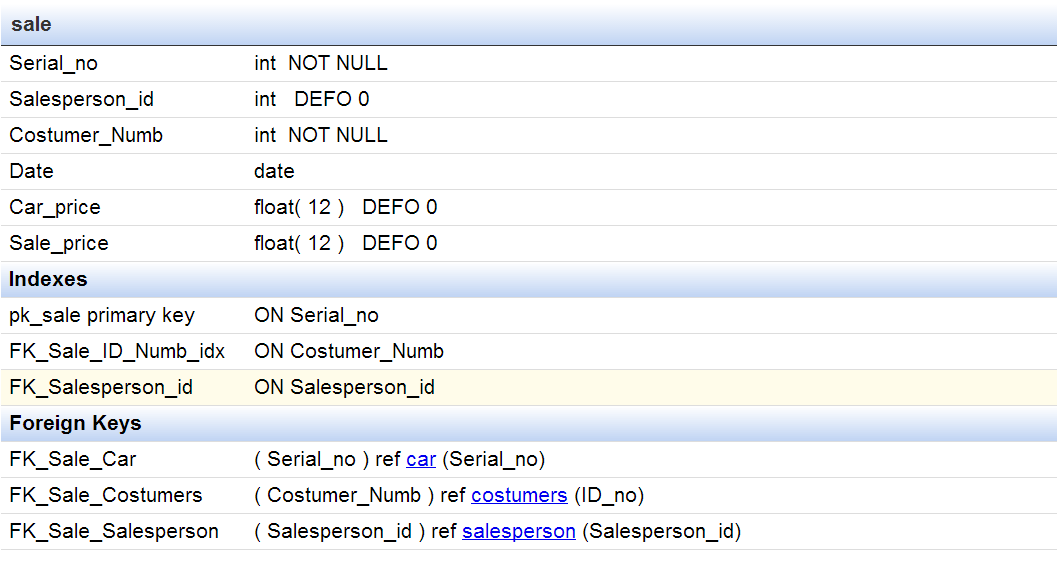


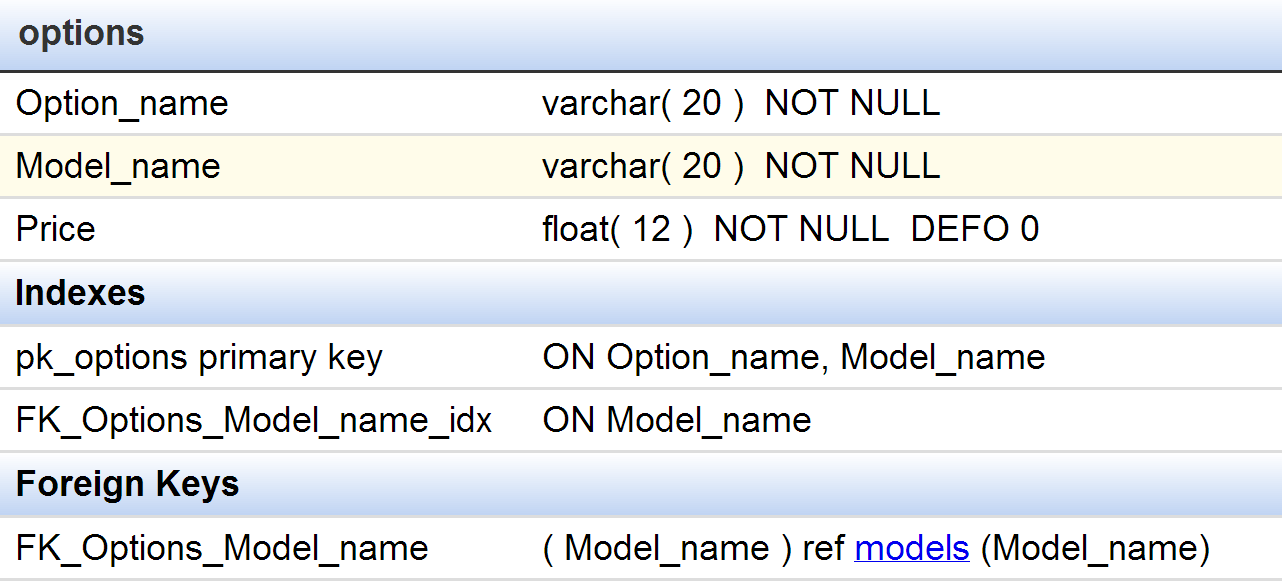
***GELİŞTİRİLMİŞ MODELDE TABLOLAR VE ÖZELLİKLERİ***

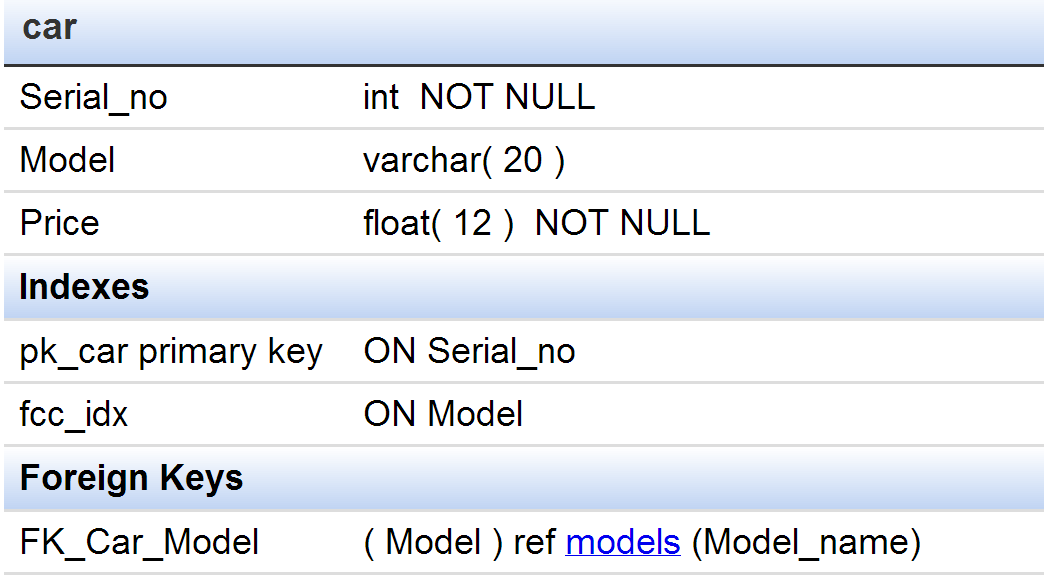


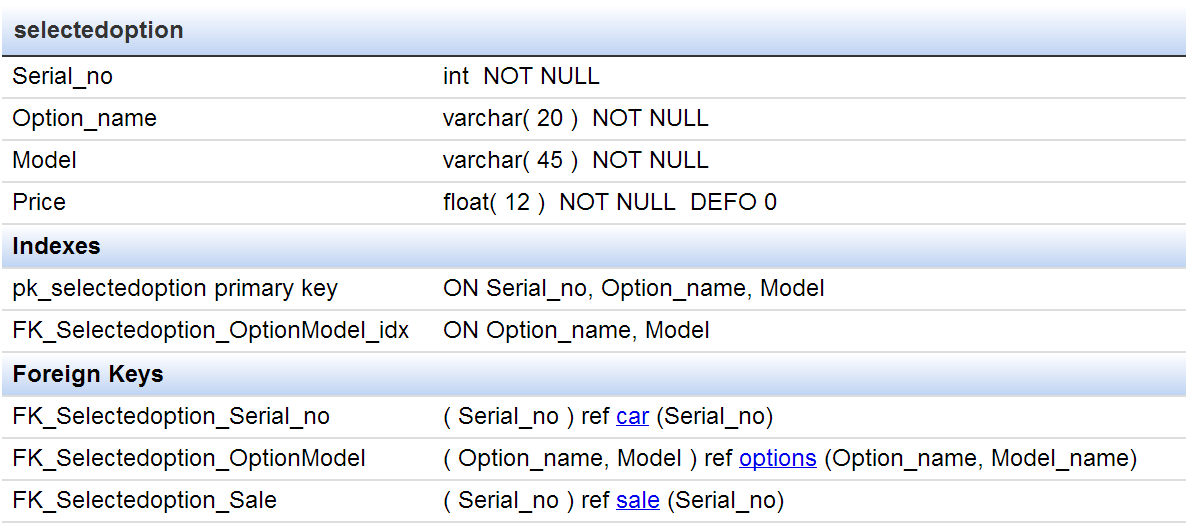


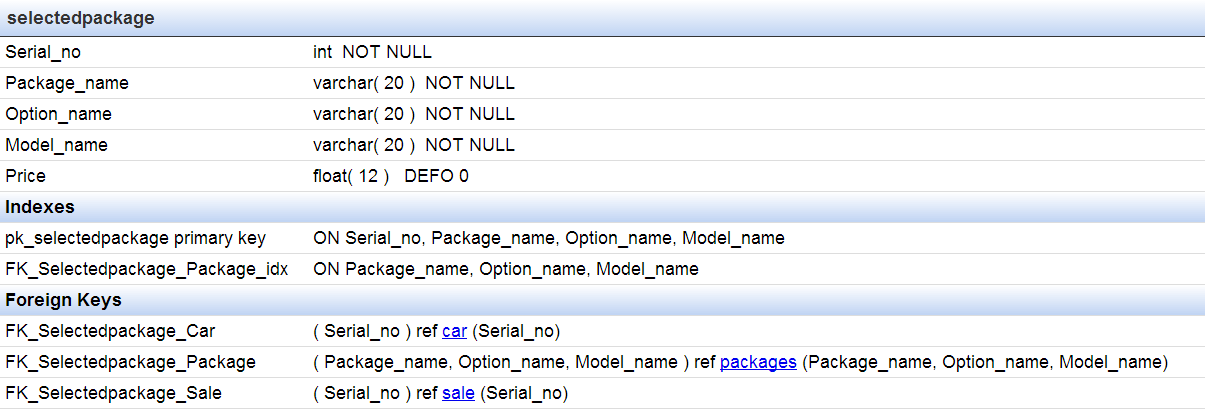


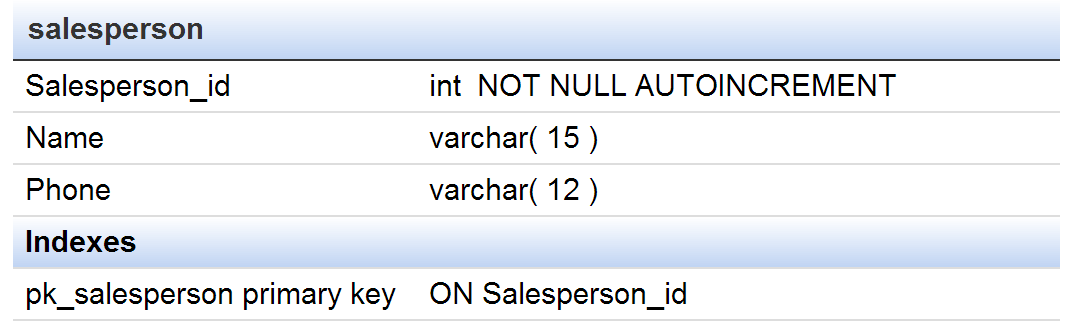












# Automotive Sales Automation Database

Projemiz araba satış sisteminin kontrolü ve satılan bu arabaların müşterilerle olan ilişkilerinin denetimini sağlayan bir veri tabanı modelidir. Bu projede modellerimizi 2 adımda gerçekleştireceğiz… Birincisi bize verilen 4 tabloya ait veri tabanı modelini kurmak, ikincisi ise bizden istenen kısıtlar doğrultusunda geliştireceğimiz yeni veri tabanı modelidir. Bu kapsamda modelimizin ilk içeriğinde arabalar, arabaya ait opsiyonlar, arabanın satışı ve arabanın satıcısı şeklinde dört farklı tablo ile bunları ilişkilendireceğiz. İkinci aşama olan gelişmiş model kısmında ise bunlara ek olarak müşteriler, araba modelleri, uygulanan paketler, seçilen özellikler ve seçilen paketler şeklinde ek tablolar ekleyip projemizi daha etkin hale getirmiş olacağız. Böylelikle satışları ve bu satışların müşterilerle olan ilişkilerini kontrol altına almış olacağız.

## FİRST STEP OF DATABASE

* Create Tables
* [Inserts](C:\\Users\\Fırat\\Desktop\\Rapor\\BookStore\\BookStore_Insert.sql)
* [Updates](C:\\Users\\Fırat\\Desktop\\Rapor\\BookStore\\BOOKSTORE_Update.sql)
* [Deletes](file:///C:\Users\Fırat\Desktop\Rapor\BookStore\BookStore_Delete.sql)
* [Queries](C:\\Users\\Fırat\\Desktop\\Rapor\\BookStore\\BookStore_Sorgu.sql)

### CAR

**Bir araca ait araç bilgilerinin ve fiyatının tutulduğu tablo**

Serial\_no int(11) NOT NULL, **Her aracın seri no’sunu 1er artıtır.**

Model varchar(20) DEFAULT NULL, **Arabanın model bilgisini tutar**

Manufactuer varchar(15) DEFAULT NULL, **Üretici firmanın bilgisini tutar.**

Price float NOT NULL, **Araba fiyatının bilgisini tutar.**

PRIMARY KEY (`Serial\_no`) **Entity integrity constraint. PK Null olamaz.**

**OPTIONS**

**Bir araca ait ‘options’ı listeleyen tablo.**

Serial\_no int(11) NOT NULL, **Aracın seri numarasını tutar**

Option\_name varchar(20) NOT NULL, **Arabaya ait özelliğin ismidir.**

Price float NOT NULL, **Bu opsiyonun fiyatını tutar.**

PRIMARY KEY (`Serial\_no`,`Option\_name`), **Entity integrity constraint. PK Null olamaz.**

FOREIGN KEY (`Serial\_no`) REFERENCES `car` (`Serial\_no`), **Referential integrity constraint.**

### SALE

**Bir aracın satış bilgileri listeleyen tablo**

Salesperson\_id int(11) NOT NULL, **Satıcının numarasını artan şekilde tutar**

Serial\_no int(11) NOT NULL, **Araca ait seri numarayı tutar**

Date` date DEFAULT NULL, **Satışın hangi tarihte gerçekleştiğini tutar**

Sale\_price float NOT NULL, **Satış fiyatını tutar.**

PRIMARY KEY (Salesperson\_id,Serial\_no), **Entity integrity constraint. PK Null olamaz.**

FOREIGN KEY (Salesperson\_id) REFERENCES salesperson(Salesperson\_id) **Ref. integrity constraint.**

FOREIGN KEY (`Serial\_no`) REFERENCES `car` (`Serial\_no`) **Referential integrity constraint.**

### SALESPERSON

**Satıcının bilgilerini listeleyen tablo**

Salesperson\_id int(11) NOT NULL, **Satıcının numarasını artan şekilde tutar**

Name varchar(15) DEFAULT NULL, **Satıcının ismini tutar**

Phone varchar(12) DEFAULT NULL, **Satıcının numarasını tutar**

PRIMARY KEY (`Salesperson\_id`), **Entity integrity constraint. PK Null olamaz.**

## CREATE TABLE SQLs OF FİRST STEP MODEL

CREATE TABLE `car` (

`Serial\_no` int(11) NOT NULL,

`Model` varchar(20) DEFAULT NULL,

`Manufactuer` varchar(15) DEFAULT NULL,

`Price` float NOT NULL,

PRIMARY KEY (`Serial\_no`)

)

CREATE TABLE `options` (

`Serial\_no` int(11) NOT NULL,

`Option\_name` varchar(20) NOT NULL,

`Price` float NOT NULL,

PRIMARY KEY (`Serial\_no`,`Option\_name`),

CONSTRAINT `FK\_Option\_Serial\_no` FOREIGN KEY (`Serial\_no`) REFERENCES `car` (`Serial\_no`) ON DELETE NO ACTION ON UPDATE CASCADE

)

CREATE TABLE `sale` (

`Salesperson\_id` int(11) NOT NULL,

`Serial\_no` int(11) NOT NULL,

`Date` date DEFAULT NULL,

`Sale\_price` float NOT NULL,

PRIMARY KEY (`Salesperson\_id`,`Serial\_no`),

KEY `FK\_Sale\_Serial\_no` (`Serial\_no`),

CONSTRAINT `FK\_Sale\_Salesperson\_id` FOREIGN KEY (`Salesperson\_id`) REFERENCES `salesperson` (`Salesperson\_id`) ON DELETE CASCADE ON UPDATE CASCADE,

CONSTRAINT `FK\_Sale\_Serial\_no` FOREIGN KEY (`Serial\_no`) REFERENCES `car` (`Serial\_no`) ON DELETE CASCADE ON UPDATE CASCADE

)

CREATE TABLE `salesperson` (

`Salesperson\_id` int(11) NOT NULL AUTO\_INCREMENT,

`Name` varchar(15) DEFAULT NULL,

`Phone` varchar(12) DEFAULT NULL,

PRIMARY KEY (`Salesperson\_id`)

)

**(INSERTS – UPDATES – DELETES) OF FİRST STEP MODEL**

INSERT INTO `automotivesalesautomation`.`car` (`Serial\_no`, `Model`, `Manufactuer`, `Price`) VALUES (111222333, 'Zafira', 'OPEL', 30000);

INSERT INTO `automotivesalesautomation`.`car` (`Serial\_no`, `Model`, `Manufactuer`, `Price`) VALUES (123123123, 'Meriva', 'OPEL', 38000);

INSERT INTO `automotivesalesautomation`.`car` (`Serial\_no`, `Model`, `Manufactuer`, `Price`) VALUES (454545454, 'Focus', 'FORD', 35000);

INSERT INTO `automotivesalesautomation`.`car` (`Serial\_no`, `Model`, `Manufactuer`, `Price`) VALUES (565656566, 'Mustang', 'FORD', 13000);

INSERT INTO `automotivesalesautomation`.`car` (`Serial\_no`, `Model`, `Manufactuer`, `Price`) VALUES (789789789, 'Veneno', 'LAMBORGHINI', 800000);

INSERT INTO `automotivesalesautomation`.`car` (`Serial\_no`, `Model`, `Manufactuer`, `Price`) VALUES (987987987, 'Gallardo', 'LAMBORGHINI', 450000);

INSERT INTO `automotivesalesautomation`.`car` (`Serial\_no`, `Model`, `Manufactuer`, `Price`) VALUES (123456789, 'Boxter', 'PORSCHE', 120000);

INSERT INTO `automotivesalesautomation`.`car` (`Serial\_no`, `Model`, `Manufactuer`, `Price`) VALUES (987654321, 'Cayenne', 'PORSCHE', 500000);

INSERT INTO `automotivesalesautomation`.`car` (`Serial\_no`, `Model`, `Manufactuer`, `Price`) VALUES (135135135, 'Passat', 'VOLKSWAGEN', 50000);

INSERT INTO `automotivesalesautomation`.`car` (`Serial\_no`, `Model`, `Manufactuer`, `Price`) VALUES (579579579, 'Scirocco', 'VOLKSWAGEN', 32000);

INSERT INTO `automotivesalesautomation`.`car` (`Serial\_no`, `Price`) VALUES (999999999, 85000);

INSERT INTO `automotivesalesautomation`.`car` (`Serial\_no`, `Price`) VALUES (999999988, 90000);

INSERT INTO `automotivesalesautomation`.`options` (`Serial\_no`, `Option\_name`, `Price`) VALUES (789789789, 'TURBO', 30000);

INSERT INTO `automotivesalesautomation`.`options` (`Serial\_no`, `Option\_name`, `Price`) VALUES (987654321, 'TURBO', 35000);

INSERT INTO `automotivesalesautomation`.`options` (`Serial\_no`, `Option\_name`, `Price`) VALUES (123456789, 'SPARKO', 10000);

INSERT INTO `automotivesalesautomation`.`options` (`Serial\_no`, `Option\_name`, `Price`) VALUES (111222333, 'SUROUND', 5000);

INSERT INTO `automotivesalesautomation`.`salesperson` (`Salesperson\_id`, `Name`, `Phone`) VALUES (1,'James', '444-555-4544');

INSERT INTO `automotivesalesautomation`.`salesperson` (`Salesperson\_id`, `Name`, `Phone`) VALUES (2,'Robert', '222-333-2323');

INSERT INTO `automotivesalesautomation`.`salesperson` (`Salesperson\_id`, `Name`, `Phone`) VALUES (3,'John', '121-211-3123');

INSERT INTO `automotivesalesautomation`.`salesperson` (`Salesperson\_id`,`Name`, `Phone`) VALUES (4,'Richard', '686-766-7777');

INSERT INTO `automotivesalesautomation`.`sale` (`Salesperson\_id`, `Serial\_no`, `Date`, `Sale\_price`) VALUES (1, 987654321, '2013-10-08.', 535000);

INSERT INTO `automotivesalesautomation`.`sale` (`Salesperson\_id`, `Serial\_no`, `Date`, `Sale\_price`) VALUES (2, 789789789, '2013-10-28', 830000);

INSERT INTO `automotivesalesautomation`.`sale` (`Salesperson\_id`, `Serial\_no`, `Date`, `Sale\_price`) VALUES (2, 123456789, '2012-12-12.', 130000);

INSERT INTO `automotivesalesautomation`.`sale` (`Salesperson\_id`, `Serial\_no`, `Date`, `Sale\_price`) VALUES (3, 579579579, '2013-06-11.', 32000)

INSERT INTO `automotivesalesautomation`.`sale` (`Salesperson\_id`, `Serial\_no`, `Date`, `Sale\_price`) VALUES (1, 111222333, '2013-01-02.', 35000);

INSERT INTO `automotivesalesautomation`.`sale` (`Salesperson\_id`, `Serial\_no`, `Date`, `Sale\_price`) VALUES (1, 123123123, '2013-01-02.', 30000);

UPDATE `automotivesalesautomation`.`car` SET `Price`=`Price`\*1.1 WHERE `Manufactuer`='FORD';

UPDATE `automotivesalesautomation`.`options` SET `Price`=5500 WHERE `Serial\_no`=111222333 and `Option\_name`='SUROUND';

UPDATE `automotivesalesautomation`.`salesperson` SET `Phone`='661-364-9110' WHERE `Salesperson\_id`=17;

DELETE FROM options;

DELETE FROM car;

DELETE FROM salesperson;

DELETE FROM sale;

DELETE FROM `automotivesalesautomation`.`car` WHERE `Serial\_no`=32323132;

DELETE FROM `automotivesalesautomation`.`options` WHERE `Serial\_no`=987654321 and `Option\_name`='TURBO';

DELETE FROM `automotivesalesautomation`.`salesperson` WHERE `Salesperson\_id`=17;

**QUERIES OF FİRST STEP MODEL**

/\* Options tablosundaki optionun ismi ve fiyatini listeler \*/

SELECT `Option\_name` , `Price`

FROM options;

/\* Satis fiyati 50000 'den buyuk olan satislari listeler \*/

SELECT `Serial\_no` , `Date` , `Sale\_price`

FROM sale

WHERE `Sale\_price` > 50000;

/\* Model ve manufactuer bilgilerini tekrarsiz listeler \*/

SELECT DISTINCT`Model` , `Manufactuer`

FROM car

WHERE `Manufactuer` IS NOT NULL

ORDER BY `Manufactuer` ASC;

/\* Arabalarin options ozelliklerini tekrarsiz listeler \*/

SELECT DISTINCT `Option\_name` , `Model` , `Manufactuer`

FROM car , options

WHERE `car`.`Serial\_no` = `options`.`Serial\_no`;

/\* Hangi satici hangi tarihte ve fiyatlarda araba satti\*/

SELECT Name , Date , Sale\_price

FROM sale , salesperson

WHERE `sale`.`Salesperson\_id` = `salesperson`.`Salesperson\_id` AND Name IS NOT NULL;

/\* Opel'in sattigi arabalarin maksimum, minimum ve ortalama satis fiyati gosterir\*/

SELECT MAX(`Sale\_price`) , MIN(`Sale\_price`) , AVG(`Sale\_price`)

FROM car , sale

WHERE `car`.`Serial\_no` = `sale`.`Serial\_no` AND `Manufactuer` = 'OPEL'

/\* Satis fiyati 50000 'den buyuk olan araclarin bilgilerini fiyatina gore artan sirada listeler\*/

SELECT `Model` , `Manufactuer` , `Date` , `Sale\_price`

FROM car JOIN sale ON `car`.`Serial\_no` = `sale`.`Serial\_no`

WHERE `Sale\_price` > 50000

ORDER BY `Sale\_price`;

/\* 2 numarali saticinin 01 ocak 2013 tarihinden sonra sattigi arabalarin options isimleri ve fiyatlari\*/

Select Option\_name , Price

from salesperson , sale , options

where salesperson.Salesperson\_id = sale.Salesperson\_id

and sale.Serial\_no = options.Serial\_no

and `Date` > '2013-01-01'

and salesperson.Salesperson\_id = 2;

/\* Option fiyati 5000 'den buyuk olan araclari satan kisinin isme gore siralanmis bilgileri ve option bilgileri \*/

SELECT `salesperson`.`Salesperson\_id` , Name , Option\_name , Date , Price

FROM sale JOIN options ON `options`.`Serial\_no` = `sale`.`Serial\_no`

JOIN salesperson ON `sale`.`Salesperson\_id` = `salesperson`.`Salesperson\_id`

WHERE Price > 5000

ORDER BY Name;

/\* Iki ve ikiden daha fazla arac satan saticinin ismi ve yaptigi satis sayisi\*/

select `salesperson`.`Salesperson\_id` , `salesperson`.`Name` , count(\*)

from sale , salesperson

where `salesperson`.`Salesperson\_id` = `sale`.`Salesperson\_id`

group by Salesperson\_id

having count(\*)>=2;

## SECOND STEP OF DATABASE

* [Create Tables](C:\\Users\\Fırat\\Desktop\\Rapor\\BookStore\\BookStore_create.sql)
* [Triggers](C:\\Users\\Fırat\\Desktop\\Rapor\\BookStore\\BookStore_Trigger.sql)
* [Inserts](C:\\Users\\Fırat\\Desktop\\Rapor\\BookStore\\BookStore_Insert.sql)
* [Updates](C:\\Users\\Fırat\\Desktop\\Rapor\\BookStore\\BOOKSTORE_Update.sql)
* [Deletes](file:///C:\Users\Fırat\Desktop\Rapor\BookStore\BookStore_Delete.sql)
* [Queries](C:\\Users\\Fırat\\Desktop\\Rapor\\BookStore\\BookStore_Sorgu.sql)

### CAR

**Bir araca ait araç bilgilerinin ve fiyatının tutulduğu tablo**

Serial\_no int(11) NOT NULL, **Her aracın seri no’sunu 1er artıtır.**

Model varchar(20) DEFAULT NULL, **Arabanın model bilgisini tutar**

Price float NOT NULL, **Araba fiyatının bilgisini tutar.**

PRIMARY KEY (`Serial\_no`) **Entity integrity constraint. PK Null olamaz.**

FOREIGN KEY (`Model`) REFERENCES `models` (`Model\_name`) **Referential Integrity Constraint**

### COSTUMERS

**Müşterilere ait bilgilerin tutulduğu tablo**

ID\_no int(11) NOT NULL, **Her müşterinin ıd no’sunu tutar.**

`Costumer\_name` varchar(30) DEFAULT NULL, **Müşterinin isim bilgisini tutar**

`Address` varchar(200) DEFAULT NULL, **Müşterinin adres bilgisini tutar.**

`Phone` varchar(12) DEFAULT NULL, **Müşterinin telefon numara bilgisini tutar**

PRIMARY KEY (`ID\_no`) **Entity integrity constraint. PK Null olamaz.**

### MODELS

**Arabalara ait modelleri ve bilgilerini tutan tablo**

Model\_name varchar(20) NOT NULL, **Model isminin bilgilerini tutar.**

Manufactuer varchar(15) DEFAULT NULL, **Üretici firmanın bilgisini tutar**

PRIMARY KEY (`Model\_name`) **Entity integrity constraint. PK Null olamaz.**

### OPTIONS

**Arabalara ait opsiyon bilgilerini tutan tablo**

Option\_name varchar(20) NOT NULL, **Opsiyon isminin bilgilerini tutar.**

Model\_name varchar(20) NOT NULL, **Model isminin bilgisini tutar**

Price float NOT NULL DEFAULT '0', **Opsiyonun fiyat bilgisini tutar.**

PRIMARY KEY (`Option\_name`,`Model\_name`) **Entity integrity constraint. PK Null olamaz.**

FOREIGN KEY (`Model\_name`) REFERENCES `models` (`Model\_name`) **Referential Integrity Constraint**

### PACKAGE

**Araca uygulan paket özelliklerini ve bilgilerini tutan tablo**

Package\_name varchar(20) NOT NULL, **Uygulanan paket isminin bilgilerini tutar.**

Option\_name varchar(20) NOT NULL, **Opsiyon isminin bilgisini tutar**

Model\_name varchar(20) NOT NULL, **Model isminin bilgisini tutar**

Price float NOT NULL DEFAULT '0', **Paket fiyat bilgisini tutar.**

PRIMARY KEY (`Package\_name`,`Option\_name`,`Model\_name`) **Entity integrity constraint. PK Null olamaz.**

FOREIGN KEY (`Option\_name`, `Model\_name`) REFERENCES `options` (`Option\_name`, `Model\_name`) **Referential Integrity Constraint**

### SALE

**Araba satışına ait tüm bilgilerin tutulduğu tablo**

Salesperson\_id int(11) DEFAULT '0', **Satıcının ıd\_no sunun bilgilerini tutar.**

Serial\_no int(11) NOT NULL, **Satılan aracın seri numarasını tutar**

Costumer\_Numb int(11) NOT NULL, **Müşteri numarasının bilgisini tutar.**

Date date DEFAULT NULL, **Satışın tarih bilgisini tutar**

Car\_price float DEFAULT '0', **Satılan aracın fiyat bilgisini tutar**

Sale\_price float DEFAULT '0', **Satılan aracın toplam fiyat bilgisini tutar**

PRIMARY KEY (`Serial\_no`), **Entity integrity constraint. PK Null olamaz.**

FOREIGN KEY (`Costumer\_Numb`) REFERENCES `costumers` (`ID\_no`) **Referential Integrity Constraint**

FOREIGN KEY (`Salesperson\_id`) REFERENCES `salesperson` (`Salesperson\_id`) **Referential Integrity Constraint**

FOREIGN KEY (`Serial\_no`) REFERENCES `car` (`Serial\_no`) **Referential Integrity Constraint**

### SALESPERSON

**Satıcının bilgilerin tutulduğu tablo**

Salesperson\_id int(11) NOT NULL AUTO\_INCREMENT, **Satıcının ıd\_no sunu artan sırada bilgilerini tutar.**

Name varchar(15) DEFAULT NULL, **Satıcının isim bilgilerini tutar**

Phone varchar(12) DEFAULT NULL, **Satıcının telefon numarasının bilgisini tutar.**

PRIMARY KEY (`Salesperson\_id`) **Entity integrity constraint. PK Null olamaz.**

### SELECTEDOPTION

**Seçilen opsiyon’ların bilgilerinin tutulduğu tablo**

Serial\_no int(11) NOT NULL, **Opsiyon eklenecek aracın seri no bilgilerini tutar.**

Option\_name varchar(20) NOT NULL, **Seçilen opsiyonun isim bilgilerini tutar**

Model varchar(45) NOT NULL, **Opsiyonun hangi modele eklenme bilgisini tutar.**

Price float NOT NULL DEFAULT '0',

PRIMARY KEY (`Serial\_no`,`Option\_name`,`Model`), **Entity integrity constraint. PK Null olamaz.**

FOREIGN KEY (`Option\_name`, `Model`) REFERENCES `options` (`Option\_name`, `Model\_name`) **Referential Integrity Constraint**

FOREIGN KEY (`Serial\_no`) REFERENCES `sale` (`Serial\_no`) **Referential Integrity Constraint**

FOREIGN KEY (`Serial\_no`) REFERENCES `car` (`Serial\_no`) **Referential Integrity Constraint**

### SELECTEDPACKAGE

**Seçilen paketin ve ona ait bilgilerin tutulduğu tablo**

`Serial\_no` int(11) NOT NULL, **Paketin eklenecek aracın seri no bilgilerini tutar**

`Package\_name` varchar(20) NOT NULL, **Seçilen paketin isim bilgilerini tutar**

`Option\_name` varchar(20) NOT NULL, **Seçilen paketin option isim bilgilerini tutar** `Model\_name` varchar(20) NOT NULL, **Seçilen paketin hangi modele eklendiği bilgisi verir**

`Price` float DEFAULT '0', **Seçilen paketin fiyat bilgisini tutar**

PRIMARY KEY (`Serial\_no`,`Package\_name`,`Option\_name`,`Model\_name`), **Entity integrity constraint. PK Null olamaz**

FOREIGN KEY (`Package\_name`, `Option\_name`, `Model\_name`) REFERENCES `packages` (`Package\_name`, `Option\_name`, `Model\_name`) **Referential Integrity Constraint**

FOREIGN KEY (`Serial\_no`) REFERENCES `sale` (`Serial\_no`) **Referential Integrity Constraint**

FOREIGN KEY (`Serial\_no`) REFERENCES `car` (`Serial\_no`) **Referential Integrity Constraint**

CREATE TABLE SQLs OF SECOND STEP MODEL

CREATE TABLE `car` (

`Serial\_no` int(11) NOT NULL,

`Model` varchar(20) DEFAULT NULL,

`Price` float NOT NULL,

PRIMARY KEY (`Serial\_no`),

KEY `fcc\_idx` (`Model`),

CONSTRAINT `FK\_Car\_Model` FOREIGN KEY (`Model`) REFERENCES `models` (`Model\_name`) ON DELETE CASCADE ON UPDATE CASCADE

)

CREATE TABLE `costumers` (

`ID\_no` int(11) NOT NULL,

`Costumer\_name` varchar(30) DEFAULT NULL,

`Address` varchar(200) DEFAULT NULL,

`Phone` varchar(12) DEFAULT NULL,

PRIMARY KEY (`ID\_no`)

)

CREATE TABLE `models` (

`Model\_name` varchar(20) NOT NULL,

`Manufactuer` varchar(15) DEFAULT NULL,

PRIMARY KEY (`Model\_name`)

)

CREATE TABLE `options` (

`Option\_name` varchar(20) NOT NULL,

`Model\_name` varchar(20) NOT NULL,

`Price` float NOT NULL DEFAULT '0',

PRIMARY KEY (`Option\_name`,`Model\_name`),

KEY `FK\_Options\_Model\_name\_idx` (`Model\_name`),

CONSTRAINT `FK\_Options\_Model\_name` FOREIGN KEY (`Model\_name`) REFERENCES `models` (`Model\_name`) ON DELETE CASCADE ON UPDATE CASCADE

)

CREATE TABLE `packages` (

`Package\_name` varchar(20) NOT NULL,

`Option\_name` varchar(20) NOT NULL,

`Model\_name` varchar(20) NOT NULL,

`Price` float NOT NULL DEFAULT '0',

PRIMARY KEY (`Package\_name`,`Option\_name`,`Model\_name`),

KEY `FK\_Packages\_OptionModel\_idx` (`Option\_name`,`Model\_name`),

CONSTRAINT `FK\_Packages\_OptionModel` FOREIGN KEY (`Option\_name`, `Model\_name`) REFERENCES `options` (`Option\_name`, `Model\_name`) ON DELETE CASCADE ON UPDATE CASCADE

)

CREATE TABLE `sale` (

`Salesperson\_id` int(11) DEFAULT '0',

`Serial\_no` int(11) NOT NULL,

`Costumer\_Numb` int(11) NOT NULL,

`Date` date DEFAULT NULL,

`Car\_price` float DEFAULT '0',

`Sale\_price` float DEFAULT '0',

PRIMARY KEY (`Serial\_no`),

KEY `FK\_Sale\_Serial\_no` (`Serial\_no`),

KEY `FK\_Sale\_ID\_Numb\_idx` (`Costumer\_Numb`),

KEY `FK\_Salesperson\_id` (`Salesperson\_id`),

CONSTRAINT `FK\_Sale\_ID\_Numb` FOREIGN KEY (`Costumer\_Numb`) REFERENCES `costumers` (`ID\_no`) ON DELETE CASCADE ON UPDATE CASCADE,

CONSTRAINT `FK\_Salesperson\_id` FOREIGN KEY (`Salesperson\_id`) REFERENCES `salesperson` (`Salesperson\_id`) ON DELETE SET NULL ON UPDATE CASCADE,

CONSTRAINT `FK\_Sale\_Serial\_no` FOREIGN KEY (`Serial\_no`) REFERENCES `car` (`Serial\_no`) ON DELETE CASCADE ON UPDATE CASCADE

)

CREATE TABLE `salesperson` (

`Salesperson\_id` int(11) NOT NULL AUTO\_INCREMENT,

`Name` varchar(15) DEFAULT NULL,

`Phone` varchar(12) DEFAULT NULL,

PRIMARY KEY (`Salesperson\_id`)

)

CREATE TABLE `selectedoption` (

`Serial\_no` int(11) NOT NULL,

`Option\_name` varchar(20) NOT NULL,

`Model` varchar(45) NOT NULL,

`Price` float NOT NULL DEFAULT '0',

PRIMARY KEY (`Serial\_no`,`Option\_name`,`Model`),

KEY `FK\_Selectedoption\_OptionModel\_idx` (`Option\_name`,`Model`),

CONSTRAINT `FK\_Selectedoption\_OptionModel` FOREIGN KEY (`Option\_name`, `Model`) REFERENCES `options` (`Option\_name`, `Model\_name`) ON DELETE CASCADE ON UPDATE CASCADE,

CONSTRAINT `FK\_Selectedoption\_Sale` FOREIGN KEY (`Serial\_no`) REFERENCES `sale` (`Serial\_no`) ON DELETE CASCADE ON UPDATE CASCADE,

CONSTRAINT `FK\_Selectedoption\_Serial\_no` FOREIGN KEY (`Serial\_no`) REFERENCES `car` (`Serial\_no`) ON DELETE CASCADE ON UPDATE CASCADE

)

CREATE TABLE `selectedpackage` (

`Serial\_no` int(11) NOT NULL,

`Package\_name` varchar(20) NOT NULL,

`Option\_name` varchar(20) NOT NULL,

`Model\_name` varchar(20) NOT NULL,

`Price` float DEFAULT '0',

PRIMARY KEY (`Serial\_no`,`Package\_name`,`Option\_name`,`Model\_name`)

KEY `FK\_Selectedpackage\_Package\_idx` (`Package\_name`,`Option\_name`,`Model\_name`),

CONSTRAINT `FK\_Selectedpackage\_Package` FOREIGN KEY (`Package\_name`, `Option\_name`, `Model\_name`) REFERENCES `packages` (`Package\_name`, `Option\_name`, `Model\_name`) ON DELETE CASCADE ON UPDATE CASCADE,

CONSTRAINT `FK\_Selectedpackage\_Sale` FOREIGN KEY (`Serial\_no`) REFERENCES `sale` (`Serial\_no`) ON DELETE CASCADE ON UPDATE CASCADE,

CONSTRAINT `FK\_Selectedpackage\_Car` FOREIGN KEY (`Serial\_no`) REFERENCES `car` (`Serial\_no`) ON DELETE CASCADE ON UPDATE CASCADE

)

***TRIGGERS OF SECOND STEP MODEL***

**Costumers Table:**

**(Before delete)**

USE `automotivesalesautomation`;

DELIMITER $$

CREATE DEFINER=`root`@`localhost` TRIGGER `costumers\_BDEL` BEFORE DELETE ON `costumers` FOR EACH ROW

begin

declare sayi int default 0;

**/\*Costumer'a Herhangi Bir Satış Yapıldıysa İlgili Alıcının Silinmemesi Gerekiyor\*/**

set @sayi = (select count(\*) from sale where Costumer\_Numb=OLD.ID\_no);

if(@sayi>0) then

SIGNAL SQLSTATE '45000' SET MESSAGE\_TEXT = *"Silmek istediğiniz kişinin aktif satışı bulunduğundan bu kişiyi silemezsiniz";*

end if;

end

**sale Table:**

**(after delete)**

USE `automotivesalesautomation`;

DELIMITER $$

CREATE DEFINER=`root`@`localhost` TRIGGER `sale\_ADEL` AFTER DELETE ON `sale` FOR EACH ROW

begin

**/\*Satışı Sildiğimizde onunla ilgili seçilmiş olan option ve packaga'larında silinmesi gerekir.\*/**

Delete from selectedoption Where Serial\_no=OLD.Serial\_no;

Delete from selectedpackage Where Serial\_no=OLD.Serial\_no;

End

**(Before Insert)**

USE `automotivesalesautomation`;

DELIMITER $$

CREATE DEFINER=`root`@`localhost` TRIGGER `sale\_BINS` BEFORE INSERT ON `sale` FOR EACH ROW

begin

**/\*Yeni Eklenen Kaydın Price Bilgisini Örnek Olarak Girdiğimizden Dolayı Fiyat Girmiyoruz**

**Onun Yerine Arabanın Fiyatını Direk İlgili Car tablosunda ki Araba Fiyatından Çekiyoruz**

**Başlangıçta Option ve Package Olmadığından Sale\_price Bilgisine Araba Fiyatını Gönderiyoruz.**

**Selected Option ve Package Eklendiğinde, Güncellendiğinde ve Silindiğinde Sale\_price Tablosu Güncellenir\*/**

Set NEW.Car\_price := (Select Price From car Where Serial\_no=NEW.Serial\_no);

Set New.Sale\_price :=(Select Price From car Where Serial\_no=NEW.Serial\_no);

end

**salespersons Table:**

**(Before Delete)**

USE `automotivesalesautomation`;

DELIMITER $$

CREATE DEFINER=`root`@`localhost` TRIGGER `salesperson\_BDEL` BEFORE DELETE ON `salesperson` FOR EACH ROW

begin

declare sayi int default 0;

**/\*Satıcı Herhangi Bir Araba Sattıysa Bu Satıcının Bilgileri Silinmemesi Lazım\*/**

set @sayi = (select count(\*) from sale where Salesperson\_id=OLD.Salesperson\_id);

if(@sayi>0) then

SIGNAL SQLSTATE '45000' SET MESSAGE\_TEXT = *"Silmek istediğiniz kişinin aktif satışı bulunduğundan bu kişiyi silemezsiniz";*

end if;

end

**salespersons Table:**

**(Before Delete)**

USE `automotivesalesautomation`;

DELIMITER $$

CREATE DEFINER=`root`@`localhost` TRIGGER `salesperson\_BDEL` BEFORE DELETE ON `salesperson` FOR EACH ROW

begin

declare sayi int default 0;

**/\*Satıcı Herhangi Bir Araba Sattıysa Bu Satıcının Bilgileri Silinmemesi Lazım\*/**

set @sayi = (select count(\*) from sale where Salesperson\_id=OLD.Salesperson\_id);

if(@sayi>0) then

SIGNAL SQLSTATE '45000' SET MESSAGE\_TEXT = *"Silmek istediğiniz kişinin aktif satışı bulunduğundan bu kişiyi silemezsiniz";*

end if;

end

**selectedoptions Table:**

**(After Delete)**

USE `automotivesalesautomation`;

DELIMITER $$

CREATE DEFINER=`root`@`localhost` TRIGGER `selectedoption\_ADEL` AFTER DELETE ON `selectedoption` FOR EACH ROW

begin

**#Option Silindiğinde Satış Fiyatı Güncellenmelidir.**

Update sale set Sale\_price=Sale\_Price - OLD.Price Where Serial\_no=OLD.Serial\_no;

End

**(After Insert)**

USE `automotivesalesautomation`;

DELIMITER $$

CREATE DEFINER=`root`@`localhost` TRIGGER `selectedoption\_AINS` AFTER INSERT ON `selectedoption` FOR EACH ROW

begin

**/\* İlgili Arabaya Eklediğimiz Option Satış Fiyatını Güncellemelidir. \*/**

update sale Set Sale\_price=Sale\_price + NEW.Price Where Serial\_no=NEW.Serial\_no;

end

**selectedoptions Table-devam:**

**(After Update)**

USE `automotivesalesautomation`;

DELIMITER $$

CREATE DEFINER=`root`@`localhost` TRIGGER `selectedoption\_AUPD` AFTER UPDATE ON `selectedoption` FOR EACH ROW

begin

**/\* Selectedoption'da ki Price Değerim Güncellendiğinde Satış Fiyatına Yansımalıdır. \*/**

Update sale set Sale\_price= Sale\_price + NEW.Price - OLD.Price Where Serial\_no=NEW.Serial\_no;

end

**(Before Insert)**

USE `automotivesalesautomation`;

DELIMITER $$

CREATE DEFINER=`root`@`localhost` TRIGGER `selectedoption\_BINS` BEFORE INSERT ON `selectedoption` FOR EACH ROW

begin

declare sayi int default 0;

declare say2 int default 0;

**/\* Arabanin satilip satilmadigini kontrol ediyoruz\*/**

Set @say2 = (Select count(\*) from sale s where s.Serial\_no=NEW.Serial\_no);

if(@say2 <1) then

SIGNAL SQLSTATE '45000' SET MESSAGE\_TEXT = *"Bu araba hala satilmamistir !!!";*

end if;

**/\*Eklemeye Çalıştığımız Option Özelliği Daha Öncesinde Bu Araba İçin Seçilen Package'da Var Mı?\*/**

Set @sayi = (Select count(\*) from selectedpackage sp

where sp.Serial\_no=NEW.Serial\_no and sp.Option\_name=NEW.Option\_name and sp.Model\_name=NEW.Model);

if(@sayi >0) then

SIGNAL SQLSTATE '45000' SET MESSAGE\_TEXT = *"Daha Öncesinde Package'a Kayıt Eklendiğinden Bu Option'ı Ekliyemezsiniz";*

end if;

end

**selectedpackages Table:**

**(After Delete)**

USE `automotivesalesautomation`;

DELIMITER $$

CREATE DEFINER=`root`@`localhost` TRIGGER `selectedpackage\_ADEL` AFTER DELETE ON `selectedpackage` FOR EACH ROW

begin

**#Package Option'ı Silindiğinde Satış Fiyatı Güncellenmelidir.**

Update sale set Sale\_price=Sale\_Price - OLD.Price Where Serial\_no=OLD.Serial\_no;

end

**(After Insert)**

USE `automotivesalesautomation`;

DELIMITER $$

CREATE DEFINER=`root`@`localhost` TRIGGER `selectedpackage\_AINS` AFTER INSERT ON `selectedpackage` FOR EACH ROW

Begin

**/\* Package Eklendiğinde Sale\_Price Güncellenmelidir \*/**

Update sale Set Sale\_price = Sale\_price + NEW.Price Where Serial\_no=NEW.Serial\_no;

end

**selectedpackages Table-devam:**

**(After Update)**

USE `automotivesalesautomation`;

DELIMITER $$

CREATE DEFINER=`root`@`localhost` TRIGGER `selectedpackage\_AUPD` AFTER UPDATE ON `selectedpackage` FOR EACH ROW

Begin

**/\* Selectedpackage'da ki Price Değerim Güncellendiğinde Satış Fiyatına Yansımalıdır. \*/**

Update sale set Sale\_price= Sale\_price + NEW.Price - OLD.Price Where Serial\_no=NEW.Serial\_no;

end

**(Before Insert)**

USE `automotivesalesautomation`;

DELIMITER $$

CREATE DEFINER=`root`@`localhost` TRIGGER `selectedpackage\_BINS` BEFORE INSERT ON `selectedpackage` FOR EACH ROW

begin

declare sayi int default 0;

declare sayi2 int default 0;

**/\*Eğer Ekleyeceğimiz Yeni Package'da ki Herhangi Bir Option Seçilmiş Olan Optionlarda Varsa Package İptal Edilmelidir\*/**

Set @sayi = (Select count(\*) from selectedoption Where Serial\_no=NEW.Serial\_no and

Option\_name=NEW.Option\_name and Model=NEW.Model\_name);

if(@sayi >0) then

Delete from selectedpackage where Package\_Name=NEW.Package\_Name and Serial\_no=NEW.Serial\_no;

SIGNAL SQLSTATE '45000' SET MESSAGE\_TEXT = *"Daha Öncesinde SelectedOption'a Kayıt Eklenmiş";*

*end if;*

**/\* Birden fazla package eklediğimiz zaman bir çakışma varsa sonradan eklediğimiz paketin ekleme işlemini iptal etmemiz gerekir. \*/**

set @sayi2=(Select count(\*) from selectedpackage where Serial\_no=NEW.Serial\_no and Option\_name=NEW.Option\_name

and Model\_name=New.Model\_name);

if(@sayi2>0) then

Delete from selectedpackage where Package\_Name=NEW.Package\_Name and Serial\_no=NEW.Serial\_no;

SIGNAL SQLSTATE '45000' SET MESSAGE\_TEXT = *"Bu package daha önceden eklenen bir package'ın option'ı ile çakıştığından ekleme iptal edildi";*

end if;

end

**(INSERTS – UPDATES – DELETES) OF SECOND STEP MODEL**

INSERT INTO `automotivesalesautomation`.`models` (`Model\_name`, `Manufactuer`) VALUES ('Clio', 'RENO');

INSERT INTO `automotivesalesautomation`.`models` (`Model\_name`, `Manufactuer`) VALUES ('Corsa', 'OPEL');

INSERT INTO `automotivesalesautomation`.`models` (`Model\_name`, `Manufactuer`) VALUES ('Zafira', 'OPEL');

INSERT INTO `automotivesalesautomation`.`models` (`Model\_name`, `Manufactuer`) VALUES ('Meriva', 'OPEL');

INSERT INTO `automotivesalesautomation`.`models` (`Model\_name`, `Manufactuer`) VALUES ('Punto', 'FIAT');

INSERT INTO `automotivesalesautomation`.`models` (`Model\_name`, `Manufactuer`) VALUES ('Volante', 'ASTONMARTIN');

INSERT INTO `automotivesalesautomation`.`models` (`Model\_name`, `Manufactuer`) VALUES ('Roadster', 'ASTONMARTIN');

INSERT INTO `automotivesalesautomation`.`models` (`Model\_name`, `Manufactuer`) VALUES ('Z4', 'BMW');

INSERT INTO `automotivesalesautomation`.`models` (`Model\_name`, `Manufactuer`) VALUES ('Z8', 'BMW');

INSERT INTO `automotivesalesautomation`.`models` (`Model\_name`, `Manufactuer`) VALUES ('Gallardo', 'LAMBORGHINI');

INSERT INTO `automotivesalesautomation`.`models` (`Model\_name`, `Manufactuer`) VALUES ('Veneno', 'LAMBORGHINI');

INSERT INTO `automotivesalesautomation`.`models` (`Model\_name`, `Manufactuer`) VALUES ('Aventador', 'LAMBORGHINI');

INSERT INTO `automotivesalesautomation`.`models` (`Model\_name`, `Manufactuer`) VALUES ('Focus', 'FORD');

INSERT INTO `automotivesalesautomation`.`models` (`Model\_name`, `Manufactuer`) VALUES ('Escort', 'FORD');

INSERT INTO `automotivesalesautomation`.`models` (`Model\_name`, `Manufactuer`) VALUES ('Golf', 'VOLKSWAGEN');

INSERT INTO `automotivesalesautomation`.`models` (`Model\_name`, `Manufactuer`) VALUES ('Civic', 'HONDA');

INSERT INTO `automotivesalesautomation`.`models` (`Model\_name`, `Manufactuer`) VALUES ('Corolla', 'TOYOTA');

INSERT INTO `automotivesalesautomation`.`models` (`Model\_name`, `Manufactuer`) VALUES ('Integra', 'HONDA');

INSERT INTO `automotivesalesautomation`.`models` (`Model\_name`, `Manufactuer`) VALUES ('X7', 'BMW');

INSERT INTO `automotivesalesautomation`.`models` (`Model\_name`, `Manufactuer`) VALUES ('X8', 'BMW');

INSERT INTO `automotivesalesautomation`.`car` (`Serial\_no`, `Model`, `Price`) VALUES ('1000', 'Corolla', '40000');

INSERT INTO `automotivesalesautomation`.`car` (`Serial\_no`, `Model`, `Price`) VALUES ('1001', 'Gallardo', '500000');

INSERT INTO `automotivesalesautomation`.`car` (`Serial\_no`, `Model`, `Price`) VALUES ('1002', 'Punto', '15000');

INSERT INTO `automotivesalesautomation`.`car` (`Serial\_no`, `Model`, `Price`) VALUES ('1003', 'Punto', '15000');

INSERT INTO `automotivesalesautomation`.`car` (`Serial\_no`, `Model`, `Price`) VALUES ('1004', 'Punto', '15000');

INSERT INTO `automotivesalesautomation`.`car` (`Serial\_no`, `Model`, `Price`) VALUES ('1005', 'Punto', '15000');

INSERT INTO `automotivesalesautomation`.`car` (`Serial\_no`, `Model`, `Price`) VALUES ('1006', 'Zafira', '50000');

INSERT INTO `automotivesalesautomation`.`car` (`Serial\_no`, `Model`, `Price`) VALUES ('1007', 'X8', '110000');

INSERT INTO `automotivesalesautomation`.`car` (`Serial\_no`, `Model`, `Price`) VALUES ('1008', 'Roadster', '250000');

INSERT INTO `automotivesalesautomation`.`car` (`Serial\_no`, `Model`, `Price`) VALUES ('1009', 'Punto', '10000');

INSERT INTO `automotivesalesautomation`.`car` (`Serial\_no`, `Model`, `Price`) VALUES ('1010', 'Punto', '10000');

INSERT INTO `automotivesalesautomation`.`car` (`Serial\_no`, `Model`, `Price`) VALUES ('1011', 'Punto', '10000');

INSERT INTO `automotivesalesautomation`.`car` (`Serial\_no`, `Model`, `Price`) VALUES ('1012', 'X8', '110000');

INSERT INTO `automotivesalesautomation`.`costumers` (`ID\_no`, `Costumer\_name`, `Address`, `Phone`) VALUES ('99123123', 'John', '11 Puddledock, London EC4V 3PD, UK', '07700 900138');

INSERT INTO `automotivesalesautomation`.`costumers` (`ID\_no`, `Costumer\_name`, `Address`, `Phone`) VALUES ('99321321', 'Smith', '147 Kirby Road, Walton on the Naze, Essex CO14 8RL, UK', '07700 900967');

INSERT INTO `automotivesalesautomation`.`costumers` (`ID\_no`, `Costumer\_name`, `Address`, `Phone`) VALUES ('99123333', 'Carter', '16 Cragview Road, Belford, Northumberland NE70 7NT, UK', '07700 900395 ');

INSERT INTO `automotivesalesautomation`.`costumers` (`ID\_no`, `Costumer\_name`, `Address`, `Phone`) VALUES ('99334534', 'Bill', '62A Gowrie Road, London SW11 5NR, UK', '07700 900337');

INSERT INTO `automotivesalesautomation`.`costumers` (`ID\_no`, `Costumer\_name`, `Address`, `Phone`) VALUES ('99545345', 'Wayne', '85 Epsom Road, Guildford, Surrey GU1 3PA, UK', '07700 900864');

INSERT INTO `automotivesalesautomation`.`costumers` (`ID\_no`, `Costumer\_name`, `Address`, `Phone`) VALUES ('99997778', 'Michael', '58 Durban Road, Plymouth PL3 4LG, UK', '07700 900916');

INSERT INTO `automotivesalesautomation`.`costumers` (`ID\_no`, `Costumer\_name`, `Address`, `Phone`) VALUES ('99909090', 'Marck', 'Cheltenham, Gloucestershire GL50 9DJ, UK', '07700 900910');

INSERT INTO `automotivesalesautomation`.`costumers` (`ID\_no`, `Costumer\_name`, `Address`, `Phone`) VALUES ('99629919', 'Justin', '1 Planefield Road, Inverness, Highland IV3 5DN, UK', '07700 900956

INSERT INTO `automotivesalesautomation`.`options` (`Option\_name`, `Model\_name`, `Price`) VALUES ('SURROUND', 'Punto', '5000');

INSERT INTO `automotivesalesautomation`.`options` (`Option\_name`, `Model\_name`, `Price`) VALUES ('TURBO', 'Gallardo', '50000');

INSERT INTO `automotivesalesautomation`.`options` (`Option\_name`, `Model\_name`, `Price`) VALUES ('AIRCONDITION', 'Gallardo', '20000');

INSERT INTO `automotivesalesautomation`.`options` (`Option\_name`, `Model\_name`, `Price`) VALUES ('TURBO', 'Roadster', '40000');

INSERT INTO `automotivesalesautomation`.`options` (`Option\_name`, `Model\_name`, `Price`) VALUES ('SPARKO', 'Gallardo', '20000');

INSERT INTO `automotivesalesautomation`.`options` (`Option\_name`, `Model\_name`, `Price`) VALUES ('SPARKO\_CHILD', 'Punto', '500');

INSERT INTO `automotivesalesautomation`.`options` (`Option\_name`, `Model\_name`, `Price`) VALUES ('NITRO', 'Roadster', '60000');

INSERT INTO `automotivesalesautomation`.`options` (`Option\_name`, `Model\_name`, `Price`) VALUES ('RIM', 'Zafira', '1500');

INSERT INTO `automotivesalesautomation`.`options` (`Option\_name`, `Model\_name`, `Price`) VALUES ('AIRCONDITION', 'Zafira', '2200');

INSERT INTO `automotivesalesautomation`.`options` (`Option\_name`, `Model\_name`, `Price`) VALUES ('ALARM', 'Zafira', '300');

INSERT INTO `automotivesalesautomation`.`options` (`Option\_name`, `Model\_name`, `Price`) VALUES ('GPS', 'Gallardo', '5000');

INSERT INTO `automotivesalesautomation`.`options` (`Option\_name`, `Model\_name`, `Price`) VALUES ('GPS', 'Roadster', '6000');

INSERT INTO `automotivesalesautomation`.`options` (`Option\_name`, `Model\_name`, `Price`) VALUES ('ALARM', 'Punto', '250');

INSERT INTO `automotivesalesautomation`.`packages` (`Package\_name`, `Option\_name`, `Model\_name`, `Price`) VALUES ('STANDARD','AIRCONDITION','Gallardo','20000');

INSERT INTO `automotivesalesautomation`.`packages` (`Package\_name`, `Option\_name`, `Model\_name`, `Price`) VALUES ('STANDARD','GPS','Gallardo','5000');

INSERT INTO `automotivesalesautomation`.`packages` (`Package\_name`, `Option\_name`, `Model\_name`, `Price`) VALUES ('FULL', 'SPARKO', 'Gallardo', '19000');

INSERT INTO `automotivesalesautomation`.`packages` (`Package\_name`, `Option\_name`, `Model\_name`, `Price`) VALUES ('FULL', 'TURBO', 'Gallardo', '58000');

INSERT INTO `automotivesalesautomation`.`packages` (`Package\_name`, `Option\_name`, `Model\_name`, `Price`) VALUES ('FULL', 'AIRCONDITION', 'Gallardo', '19000');

INSERT INTO `automotivesalesautomation`.`packages` (`Package\_name`, `Option\_name`, `Model\_name`, `Price`) VALUES ('FULL', 'GPS', 'Gallardo', '4500');

INSERT INTO `automotivesalesautomation`.`packages` (`Package\_name`, `Option\_name`, `Model\_name`, `Price`) VALUES ('FULL', 'ALARM', 'PUNTO', '200');

INSERT INTO `automotivesalesautomation`.`packages` (`Package\_name`, `Option\_name`, `Model\_name`, `Price`) VALUES ('FULL', 'SURROUND', 'PUNTO', '1500');

INSERT INTO `automotivesalesautomation`.`packages` (`Package\_name`, `Option\_name`, `Model\_name`, `Price`) VALUES ('FULL', 'SPARKO\_CHILD', 'PUNTO', '500');

INSERT INTO `automotivesalesautomation`.`packages` (`Package\_name`, `Option\_name`, `Model\_name`, `Price`) VALUES ('CHILD', 'SPARKO\_CHILD', 'PUNTO', '450');

INSERT INTO `automotivesalesautomation`.`packages` (`Package\_name`, `Option\_name`, `Model\_name`, `Price`) VALUES ('CHILD', 'ALARM', 'PUNTO', '200');

INSERT INTO `automotivesalesautomation`.`salesperson` (`Salesperson\_id`, `Name`, `Phone`) VALUES ('11123123', 'Oliver', '07700 900932');

INSERT INTO `automotivesalesautomation`.`salesperson` (`Salesperson\_id`, `Name`, `Phone`) VALUES ('11321321', 'Emily', '07700 900812');

INSERT INTO `automotivesalesautomation`.`salesperson` (`Salesperson\_id`, `Name`, `Phone`) VALUES ('11445566', 'Charlie', '07700 900817');

INSERT INTO `automotivesalesautomation`.`salesperson` (`Salesperson\_id`, `Name`, `Phone`) VALUES ('11321233', 'Sophia', '07700 90077

INSERT INTO `automotivesalesautomation`.`selectedpackage` (`Serial\_no`, `Package\_name`, `Option\_name`, `Model\_name`, `Price`) VALUES ('1001', 'FULL', 'AIRCONDITION', 'Gallardo', '19000');

INSERT INTO `automotivesalesautomation`.`selectedpackage` (`Serial\_no`, `Package\_name`, `Option\_name`, `Model\_name`, `Price`) VALUES ('1001', 'FULL', 'GPS', 'Gallardo', '4500');

INSERT INTO `automotivesalesautomation`.`selectedpackage` (`Serial\_no`, `Package\_name`, `Option\_name`, `Model\_name`, `Price`) VALUES ('1001', 'FULL', 'SPARKO', 'Gallardo', '19000');

INSERT INTO `automotivesalesautomation`.`selectedpackage` (`Serial\_no`, `Package\_name`, `Option\_name`, `Model\_name`, `Price`) VALUES ('1001', 'FULL', 'TURBO', 'Gallardo', '58000');

INSERT INTO `automotivesalesautomation`.`selectedpackage` (`Serial\_no`, `Package\_name`, `Option\_name`, `Model\_name`, `Price`) VALUES ('1003', 'CHILD', 'ALARM', 'PUNTO', '200');

INSERT INTO `automotivesalesautomation`.`selectedpackage` (`Serial\_no`, `Package\_name`, `Option\_name`, `Model\_name`, `Price`) VALUES ('1003', 'CHILD', 'SPARKO\_CHILD', 'PUNTO', '450');

INSERT INTO `automotivesalesautomation`.`selectedoption` (`Serial\_no`, `Option\_name`, `Model`, `Price`) VALUES ('1002', 'ALARM', 'Punto', '250');

INSERT INTO `automotivesalesautomation`.`selectedoption` (`Serial\_no`, `Option\_name`, `Model`, `Price`) VALUES ('1003', 'SURROUND', 'Punto', '5000');

INSERT INTO `automotivesalesautomation`.`selectedoption` (`Serial\_no`, `Option\_name`, `Model`, `Price`) VALUES ('1006', 'RIM', 'Zafira', '1500');

INSERT INTO `automotivesalesautomation`.`sale` (`Salesperson\_id`, `Serial\_no`, `Costumer\_Numb`, `Date`) VALUES ('11123123', '1002', '99629919', '2013-11-11');

INSERT INTO `automotivesalesautomation`.`sale` (`Salesperson\_id`, `Serial\_no`, `Costumer\_Numb`, `Date`) VALUES ('11123123', '1001', '99629919', '2013-11-11');

INSERT INTO `automotivesalesautomation`.`sale` (`Salesperson\_id`, `Serial\_no`, `Costumer\_Numb`, `Date`) VALUES ('11123123', '1003', '99545345', '2013-11-25');

INSERT INTO `automotivesalesautomation`.`sale` (`Salesperson\_id`, `Serial\_no`, `Costumer\_Numb`, `Date`) VALUES ('11445566', '1005', '99123123', '2012-10-10');

INSERT INTO `automotivesalesautomation`.`sale` (`Salesperson\_id`, `Serial\_no`, `Costumer\_Numb`, `Date`) VALUES ('11321321', '1006', '99321321', '2013-02-11');

UPDATE `automotivesalesautomation`.`car` SET `Price`= `Price`\*1.1 WHERE `Model`='X8';

UPDATE `automotivesalesautomation`.`costumers` SET `Phone`='07700 900889' WHERE `ID\_no`='99442441';

UPDATE `automotivesalesautomation`.`models` SET `Model\_name`='X5' WHERE `Model\_name`='X6';

DELETE FROM selectedoption;

DELETE FROM selectedpackage;

DELETE FROM packages;

DELETE FROM models;

DELETE FROM costumers;

DELETE FROM options;

DELETE FROM car;

DELETE FROM salesperson;

DELETE FROM sale;

DELETE FROM `automotivesalesautomation`.`car` WHERE `Serial\_no`='1012';

DELETE FROM `automotivesalesautomation`.`costumers` WHERE `ID\_no`='99442441';

DELETE FROM `automotivesalesautomation`.`models` WHERE `Model\_name`='X6';

**QUERIES OF FİRST STEP MODEL**

SELECT costumers.Costumer\_name, costumers.Address, costumers.Phone FROM costumers;

/\*Arabaları Listeler\*/

Select Serial\_no From car;

/\*Package'ların Toplam Fiyatını Listeler\*/

Select packages.Package\_name,packages.Model\_name, Sum(packages.Price) From packages GROUP BY(Package\_name);

/\*1\_Arabaların Modellerini ve Manufactuerlarını Gösteririyor\*/

Select c.Serial\_no,c.Model,m.Manufactuer,c.Price From car c, models m Where c.Model=m.Model\_name;

/\*2\_Modellerin Sahip Olduğu Packagesları Gösterir\*/

Select p.Package\_name,m.Model\_name,m.Manufactuer From models m, packages p where m.Model\_name=p.Model\_name GROUP BY p.Package\_name;

/\*3\_Bütün Satılan Arabaların Bütün Option'larını Gösterir ve Model ve Option İsimlerine Göre Sıralar\*/

Select \* From ((Select so.Option\_name,so.Model,so.Price From selectedoption so)

UNION

(Select sp.Option\_name,sp.Model\_name,sp.Price From selectedpackage sp)) as Sonuclar order by Model,Option\_name;

/\*4\_Package'da Indirim Yapılan Optionlar\*/

Select p.Option\_name,p.Model\_name,p.Price as Package\_Price,s.Price as Option\_Price

From packages p,`options` as s Where p.Option\_name=s.Option\_name and p.Model\_name=s.Model\_name and p.Price <s.Price

/\*1\_Hangi Satıcı Kaç Tane Satış Yaptığını Listeler, Satıcının İsmine Göre Sıralar\*/

Select sp.`Name`, COUNT(\*) From sale s,salesperson sp,costumers c

Where s.Salesperson\_id=sp.Salesperson\_id and s.Costumer\_Numb=c.ID\_no

GROUP BY s.Salesperson\_id,sp.`Name` order by sp.`Name`;

/\*2\_Bir Müşteriye Birden Fazla Satış Yapan Satıcının İsmi\*/

Select \* From(Select sp.`Name`, COUNT(\*) as TotalCount From sale s,salesperson sp,costumers c

Where s.Salesperson\_id=sp.Salesperson\_id and s.Costumer\_Numb=c.ID\_no

GROUP BY c.ID\_No) as Table1 Where TotalCount > 1;

/\*3\_2013 Tarihinden Önce Hangi Satıcının Hangi Müşteriye Hangi Arabayı Hangi Tarihde Sattığını Listeler\*/

Select sp.`Name`,c.Costumer\_name,s.Serial\_no,s.Date From sale s,salesperson sp,costumers c

Where s.Salesperson\_id=sp.Salesperson\_id and s.Costumer\_Numb=c.ID\_no and s.Date < "2013-01-01";

/\*List the names of Models and their Manufacturers with more than 3 OPTIONs.\*/

Select m.Model\_name,m.Manufactuer,count(\*) From models m, `options` os Where m.Model\_name = os.Model\_name

GROUP BY os.Model\_name

Having Count(\*)>2;

/\*List the names of CUSTOMERs and the related SALEPERSONs for each SALE and group them by SALESPERSON name.\*/

Select sp.`Name`,c.Costumer\_name From sale s,salesperson sp,costumers c

Where s.Salesperson\_id=sp.Salesperson\_id and s.Costumer\_Numb=c.ID\_no

GROUP BY sp.`Name`

/\*List the names of SALESPERSONs who sold more than 2 CARs.\*/

Select \* From(Select sp.`Name`, COUNT(\*) as TotalCount From sale s,salesperson sp

Where s.Salesperson\_id=sp.Salesperson\_id GROUP BY `Name` ) as Table1 Where TotalCount

/\*List the name of the OPTION which is the most repeated one in all of the PACKAGEs\*

Select Option\_name,Sayi From

(Select Option\_name,Count(\*) as Sayi FROM packages GROUP BY Option\_name ORDER by Count(\*) DESC) as Tablom

GROUP BY Option\_name

Having Sayi=(Select Max(sayi) From (Select Count(\*) as sayi From packages GROUP BY Option\_name) as deneme);

**DOĞAL DİL CÜMLELERİYLE GELİŞTİRİLMİŞ MODEL**

***MAIN ENTITIES*: car, costumers, models, options, packages, sale, salesperson, selectedoption, selectedpackages**

**car TABLE:** Bir araca ait araç bilgilerinin ve araç fiyatının tutlduğu tablodur. Attributes: serial no, model ve price dır. ‘Serial no’da araçların seri numaraları artan bir şekilde tutulur. ‘Model’ ise seri numarası belli olan aracın modelini gösterir. Ayrıca ‘price’ ise aracın özellik ve paket eklenmemiş kendi fiyatını tutar. Arabanın seri numarası PK özellik gösterip null değer alamazken(entity integrity constrain), bu tablodaki araba modeli başka tablo olan ’models‘ tablosundaki model isimlerini referans eder(referential integrity constraint).

**costumers TABLE:** Müşterilere ait bilgilerin tutulduğu tablodur.

Attributes: ID\_no, costumer\_name, Address, Phone. Bu tabloda müşterilere ait isim,adres ve telefon numarası gibi şahsi bilgiler tutulmaktadır. ID\_no attribute’ ise müşterinin seri numarasını verir ve PK özellik göstermektedir bu tabloda. Hiçbir zaman null değer atanamaz (entity integrity constrain) bu tablo için.

Müşteriye herhangi bir satış yapıldığı zaman da alıcının silinmemesi lazım bu kıstı göz önünde bulunduralım.

**models TABLE:** Arabalar ait modelleri ve bu modellerin özelliklerini tutar.

Attributes: Model\_name, Manifactuer… Özelliklerden de anlaşılacağı üzere tabloda arabanın model ismi ve üretici firmanın bilgileri tutulmaktadır. Burada model ismi PK olup null değer içermez(entity integrity constraint).

**options TABLE:** Araçlara ait opsiyon bilgelerinin tutulduğu tablodur.

Attributes: Option\_name, Model\_name, Price… Tabloda opsiyonun ismi, aracın model ismi ve opsiyon fiyatı bilgileri bulunmaktadır. Burada Option\_name ve Model\_name beraber PK olmaktadır (entity integrity constraint). Ayrıca tablodaki model ismi ‘models’ tablosundaki model ismini referans etmektedir (referential integrity constraint). Buradaki fiyat toplam fiyatla karıştırılmamalı. Bu fiyat en son satışta ki toplam araç fiyatına eklenecektir.

**packages TABLE:** Bu tabloda araçlar için uygulanan paketler ve bilgileri vardır

Attributes: Package\_name, Option\_name, Model\_name, Price… Package\_name, Option\_name, Model\_name beraber PK olup null değer alamazlar(entity integrity constraint). Option\_name ve Model\_name ise beraber ‘options’ tablosundaki bu Option\_name ve Model\_name’i referans etmektedirler (referential integrity constraint). Buradaki fiyat ise paket için uygulanan fiyatı vermektedir. PK ların birden çok olması, içeriğinde pakete ait opsiyonların ve bunların hangi araca uygulanacakları bilgileri bulunduğundandır. Ve bu tablo bize arabanın modelini girdiğimizde arabaya ait uygulanan paketleri ve opsiyonların ismilerini verecektir. Ayrıca buradaki paketlerde bulunan opsiyonların toplam fiyatı(yani paket fiyatı) her birinin tek tek toplamından küçük eşittir ve paket için indirim yapıldığını gösterir (semantic constraint)

**sale TABLE:** Araba satışına ait bilgilerin tutulduğu tablodur.

Attributes: Salesperson\_id, Serial\_no, Costumer\_Numb, Date, Car\_price, Sale\_price… Arabanın seri numarası PK olup (entity integrity constraint) bu değer girildiğinde aracın satış bilgileri listelenecektir. Bu bilgiler satıcının id’si, aracın seri numarası, müşteri numarası, satış tarihi, aracın ve aracın toplam satış fiyatıdır. Costumer\_Numb özelliği ‘costumers’ tablosundaki ID\_no yu, Salesperson\_id ‘salesperson’ tablosundaki Salesperson\_id’yi, Serial\_no ise ‘car’ tablosundaki Serial\_no’yu referans eder(referential integrity constraint). Kim hangi aracı kime hangi tarihte ne kadara sattı bilgisini bize vermiş olur bu tablo.

Sale tablomuz için ek kısıtlardan bahsedecek olursak; satış silindiği zaman bu satışla ilişkili option ve package’ların da silinmesi gerekir ve araç silindiğinde araca ait satışlarda silinecektir.Ayrıca araca herhangi bir özellik eklemediğimiz zaman fiyatını direkt ‘car’ tablosundan çekiyoruz. Ama option ve package eklediğimiz zaman satış fiyatımızı da güncellememiz lazım.

**salesperson TABLE:** Satıcının bilgilerinin tutulduğu tablodur.

Attributes: Salesperson\_id, Name, Phone… Tabloda bulunan Salesperson\_id numarası PK olup (entity integrity constraint) satıcının sıra numarasını vermektedir. Ayrıca ‘sales’ tablosu tarafından referans edilmekteydi. Biz satış işlemleri sırasında bu satışı gerçekleştirenin bilgilerine (sıra nosu, isim, telefon numarası) burdan ulaşabiliriz. Ek kısıt olarak; satıcı herhangi bir araba sattıysa bu satıcının bilgilerinin silinmemesi gerekir diyebiliriz.

**selectedoptionTABLE**: Seçilen opsiyon bilgilerinin tutulduğu tablodur.

Attributes: Serial\_no, Option\_name, Model, Price… Burada : Serial\_no, Option\_name, Model üçü beraber PK dir(entity integrity constraint). Tabloda araç için alıcının eklemek istediği ek opsiyonlar bulunmaktadır ama burdaki özellikler seçilmiş olma durumundadırlar. Option\_name, Model ise beraber ‘options’ tablosundaki Option\_name, Model i ve ayrıca Serial\_no ‘car’ tablosundaki Serial\_no yu referans etmektedirler(referential integrity constraint). Ek olarak buradaki Serial\_no ‘sale’ tablosundaki Serial\_no yu referans etmektedir(referential integrity constraint).Buradaki Price ise seçilen opsiyonun fiyatını vermektedir ve sonunda sale tablosundaki Price a eklenip toplam fiyatı bulmakta kullanılacaktır. Hangi seri numaralı aracın hangi modeline eklenen seçilmiş opsiyonun fiyatı bilgilerinin hepsini listeleyecektir. Araca option ekleyip-çıkardığımız zaman ve o option ın fiyatı da değişiyorsa aracın fiyatı güncellenmeli. Özellikle option ekleyeceksek araca, bu araçta önceden seçilmiş pakette bu option’ın olup olmadığını kontrol etmeliyiz.

**selectedpackageTABLE:** Seçilen paketin ve pakete ait bilgilerin tutulduğu tablo

Attributes: Serial\_no, Package\_name, Option\_name, Model\_name, Price…

Serial\_no, Package\_name, Option\_name, Model\_name dördü birlikte PK olup null değer içermezler(entity integrity constraint). Burada araç için seçilen paket bilgisi bulunur. Package\_name, Option\_name, Model\_name beraber ‘packages’ tablosundaki kendi attribulerini ve Serial\_no ise ‘car’ tablosundaki Serial\_no yu referans etmektedirler(referential integrity constraint). ). Ek olarak buradaki Serial\_no ‘sale’ tablosundaki Serial\_no yu referans etmektedir(referential integrity constraint).Bu tabloda hangi seri numaraya ait modele eklenen paket ismini ve içeriğinde bulunan opsiyonları ve bu paketin toplam fiyat bilgilerine ulaşabilir. Ayrıca burdaki Price, içeriğinde bulunan her bir opsiyonun toplam fiyatından küçük eşittir(semantic constraint) bize paket seçildiğinde bir nevi indirim yapıldığını gösterir. Bu tablo için kısıt olarak ise şunları söyleyebiliriz; seçilen paketin fiyatı değiştirildiği zaman aracın fiyatı da bu doğrultuda güncellenmeli. Önemli olan kısım ise, eğer araca paket ekleyeceksek, bu paketin içinde araçta var olan bir option olma durumudur. Eğer araçta bu pakette var olan bir option varsa seçilen paket iptal edilecektir. Ayrıca araca birden fazla paket ekleme işlemi yapılıyorsa ve bu eklenen paketlerde çakışma varsa seçim önceliği önce seçilene verilmesi lazım ve sonradan seçtiğimiz paketi iptal edilmesi gerekir.