**Airline Reservation System\_Database Implementation**

**1.Create Database and Table  
  
USE** [ARS]

**CREATE** **TABLE** Customer(

CustomerID **int** **IDENTITY**(1,1) **PRIMARY** **KEY**,

Lastname **varchar**(20),

Firstname **varchar**(20),

Country **varchar**(20),

State **varchar**(20),

City **varchar**(20),

Street **varchar**(20),

Apartment **varchar**(20),

Zipcode **int**,

CustomerPhoneNo **int**,

CustomerEmail **varchar**(20),

CustomerGender **varchar**(20),

CustomerPassword **varchar**(20),

CustomerCardInfo **varchar**(20),

);

**INSERT** **INTO** Customer (Lastname , Firstname , Country, State, City, Street, Apartment, Zipcode,

CustomerPhoneNo, CustomerEmail, CustomerGender, CustomerPassword, CustomerCardInfo ) **values** ('snow', 'jon', 'US', 'MA', 'Boston', 'jamaica', 'Apt 5', 02122, 123456, 'jsnow@12',

'M', 1234, 'Amex');

**INSERT** **INTO** Customer (Lastname , Firstname , Country, State, City, Street, Apartment, Zipcode,

CustomerPhoneNo, CustomerEmail, CustomerGender, CustomerPassword, CustomerCardInfo ) **values** ('sn', 'on', 'USA', 'MAW', 'Bton', 'jamaica', 'Apt 4', 02125, 129456, 'jow@12',

'F', 1634, 'Mex');

**CREATE** **TABLE** Luggage(

LuggageID **int** **IDENTITY**(1,1) **PRIMARY** **KEY**,

CustomerID **int** **not** **null**,

LuggageWeight **int** **not** **null**

);

**ALTER** **TABLE** Luggage

**ADD** **FOREIGN** **KEY** (CustomerID) **REFERENCES** Customer(CustomerID);

**INSERT** **into** Luggage (CustomerID, LuggageWeight) **values** (1, 23);

**INSERT** **into** Luggage (CustomerID, LuggageWeight) **values** (2, 33);

**CREATE** **TABLE** [Order] (

OrderID **int** **IDENTITY**(1,1) **PRIMARY** **KEY**,

CustomerID **int** **not** **null**,

OrderTime **Datetime**,

OrderPrice **int** **not** **null**,

OrderDetail **varchar**(50),

OrderQuantity **int**

)

**ALTER** **TABLE** [Order]

**ADD** **FOREIGN** **KEY** (CustomerID) **REFERENCES** Customer(CustomerID);

**INSERT** **into** [Order] (CustomerID, OrderTime, OrderPrice, OrderDetail, OrderQuantity) **values** (1, '01/02/2013', 21, 'Fivetickets', 22);

**INSERT** **into** [Order] (CustomerID, OrderTime, OrderPrice, OrderDetail, OrderQuantity) **values** (2, '03/01/2013', 31, 'Fivetickets', 32);

**CREATE** **TABLE** Ticket(

TicketID **int** **IDENTITY**(1,1) **PRIMARY** **KEY**,

TicketPrice **int** **not** **null**,

PriceAfterDiscount **int** **not** **null**,

Discount **int** **not** **null**,

Class **varchar**(20)

)

**INSERT** **into** Ticket (TicketPrice, PriceAfterDiscount, Discount, Class) **values** (250, 150, 100, 'First');

**INSERT** **into** Ticket (TicketPrice, PriceAfterDiscount, Discount, Class) **values** (350, 250, 100, 'Economy');

**CREATE** **TABLE** Airline(

AirlineID **int** **IDENTITY**(1,1) **PRIMARY** **KEY**,

AirlineName **varchar**(25),

AirlineAddress **varchar**(50),

IATACode **char**(3),

AirlineAlliance **varchar**(25),

AirlineWebsite **varchar**(50),

)

**insert** **into** Airline (AirlineName, AirlineAddress, IATACode, AirlineAlliance, AirlineWebsite) **values**

('Jetblue', 'logan', 'AAA', 'AZ', 'jetblue.com')

**insert** **into** Airline (AirlineName, AirlineAddress, IATACode, AirlineAlliance, AirlineWebsite) **values**

('blue', 'log', 'AA', 'AU', 'blue.com')

**CREATE** **TABLE** Aircraft(

AircraftID **int** **IDENTITY**(1,1) **PRIMARY** **KEY**,

AircraftType **varchar**(20),

YearManufactured **int**,

AircraftAge **int**,

MaximumCapacity **int**,

)

**insert** **into** Aircraft (AircraftType, YearManufactured, AircraftAge, MaximumCapacity) **values**

('Airbus 320', 2001, 5, 250);

**insert** **into** Aircraft (AircraftType, YearManufactured, AircraftAge, MaximumCapacity) **values**

('boeing 320', 2003, 7, 150);

**CREATE** **TABLE** Flight (

FlightID **int** **IDENTITY**(1,1) **PRIMARY** **KEY**,

AirlineID **int** **not** **null**,

AircraftID **int** **not** **null**,

DepartureAirport **varchar**(50),

DepartureCity **varchar**(50),

DepartureTime **Datetime**,

ArrivalAirport **varchar**(50),

ArrivalCity **varchar**(50),

ArrivalTime **Datetime**,

Duration **int** **not** **null**,

DelayTime **int** **not** **null**,

PassengerNo **int**,

FlightType **varchar**(50)

)

**ALTER** **TABLE** Flight

**ADD** **FOREIGN** **KEY** (AirlineID)**REFERENCES** Airline(AirlineID);

**ALTER** **TABLE** Flight

**ADD** **FOREIGN** **KEY** (AircraftID) **REFERENCES** Aircraft(AircraftID);

**insert** **into** Flight (AirlineID , AircraftID, DepartureAirport, DepartureCity, DepartureTime, ArrivalAirport, ArrivalCity,

ArrivalTime, Duration, DelayTime, PassengerNo, FlightType) **values** (1,1,'Logan','boston','02/02/2002','canada','ontario',

'02/03/2002', 5, 2, 234, 'Passenger');

**insert** **into** Flight (AirlineID, AircraftID, DepartureAirport, DepartureCity, DepartureTime, ArrivalAirport, ArrivalCity,

ArrivalTime, Duration, DelayTime, PassengerNo, FlightType) **values** (2, 2, 'Logan', 'boston', '02/02/2006', 'canana', 'montreal',

'02/03/2006', 6, 5, 274, 'Passenger')

**SELECT** \* **FROM** Flight ;

**CREATE** **TABLE** International\_Flight(

FlightID **int** **IDENTITY**(1,1) **PRIMARY** **KEY**,

ForeignLanguage **nvarchar**(25),

PassportVerification **varchar**(20)

);

**ALTER** **TABLE** International\_Flight

**ADD** **FOREIGN** **KEY** (FlightID) **REFERENCES** Flight(FlightID);

**Insert** **into** International\_Flight (ForeignLanguage, PassportVerification) **values** ('English', 'driverlicense');

**Insert** **into** International\_Flight (ForeignLanguage, PassportVerification) **values** ('Eng', 'dlicense');

**CREATE** **TABLE** Domestic\_Flight(

FlightID **int** **IDENTITY**(1,1) **PRIMARY** **KEY**,

IDVerification **varchar**(20),

);

**ALTER** **TABLE** Domestic\_Flight

**ADD** **FOREIGN** **KEY** (FlightID) **REFERENCES** Flight(FlightID);

**Insert** **into** Domestic\_Flight (IDVerification) **values** ('dlicense');

**Insert** **into** Domestic\_Flight (IDVerification) **values** ('drivinglicense');

**CREATE** **TABLE** Payment

(

PaymentNo **int** **IDENTITY**(1,1) **PRIMARY** **KEY**,

OrderID **int** **not** **null**,

PaymentStatus **varchar**(10),

TotalAmount **int** **not** **null**,

SubmitTime **datetime**,

PaymentType **varchar**(13)

);

**ALTER** **TABLE** Payment

**ADD** **FOREIGN** **KEY** (OrderID)**REFERENCES** [Order](OrderID);

**insert** **into** Payment (OrderID, paymentStatus, TotalAmount, SubmitTime, PaymentType) **values** (3, 'Paid', 250, '01/02/2001', 'Card');

**insert** **into** Payment (OrderID, paymentStatus, TotalAmount, SubmitTime, PaymentType) **values** (4, 'Paid', 250, '01/02/2001', 'OnlinePayment');

**select** \* **from** Payment

**CREATE** **TABLE** Card(

PaymentNo **int** **IDENTITY**(1,1) **PRIMARY** **KEY**,

CardNo **int** **not** **null**,

NameOnCard **varchar**(25),

ExpDate **date**,

SecurityCode **int**

);

**ALTER** **TABLE** Card

**ADD** **FOREIGN** **KEY** (PaymentNo)**REFERENCES** Payment(PaymentNo);

**insert** **into** Card(CardNo, NameOnCard, ExpDate, SecurityCode) **values** (1234, 'Bofa', '01/02/2003', 2345);

**insert** **into** Card(CardNo, NameOnCard, ExpDate, SecurityCode) **values** (16234, 'Bof', '01/03/2003', 2346);

**CREATE** **TABLE** OnlinePayment(

PaymentNo **int** **IDENTITY**(1,1) **PRIMARY** **KEY**,

OnlinePaymentAccount **varchar**(25),

OnlinePaymentMethod **varchar**(25),

);

**Alter** **table** OnlinePayment

**ADD** **FOREIGN** **KEY** (PaymentNo) **REFERENCES** Payment(PaymentNo);

**insert** **into** OnlinePayment (OnlinePaymentAccount, OnlinePaymentMethod) **values** ('Debit', 'Card');

**insert** **into** OnlinePayment (OnlinePaymentAccount, OnlinePaymentMethod) **values** ('Credit', 'Card');

**select** \* **from** Customer

**select** \* **from** [Order]

**select** \* **from** Luggage

**CREATE** **PROCEDURE** sp\_CustomerOrderLuggage

**AS** **BEGIN**

**Select** **DISTINCT** c.Firstname+' '+c.Lastname **as** FullName,o.OrderDetail,o.OrderQuantity,l.LuggageWeight

**FROM** Customer c

**inner** **join** [Order] o

**on** c.CustomerID = o.CustomerID

**inner** **join** Luggage l

**on** c.CustomerID = l.CustomerID

**END**

**EXEC** sp\_CustomerOrderLuggage

**select** \* **from** Customer

**select** \* **from** [Order]

**select** \* **from** Payment

**select** \* **from** OnlinePayment

**select** \* **from** Card

**CREATE** **PROCEDURE** sp\_CustomerOrderPayment

**AS** **BEGIN**

**SELECT** c.Lastname ,c.Firstname,c.CustomerEmail, o.OrderTime,o.OrderQuantity,p.PaymentStatus,p.PaymentType,op.OnlinePaymentMethod,

cn.CardNo

**FROM** Customer c

**inner** **join** [Order] o

**on** c.CustomerID = o.CustomerID

**inner** **join** Payment p

**on** o.OrderID = p.OrderID

**inner** **join** OnlinePayment op

**on** p.PaymentNo = op.PaymentNo

**inner** **join** Card cn

**on** p.PaymentNo = cn.PaymentNo

**END**

**EXEC** sp\_CustomerOrderPayment

**ALTER** **PROCEDURE** sp\_flight

@inputdate **DateTime**

**AS**

**BEGIN**

**IF** **EXISTS** (**SELECT** \* **FROM** Flight **WHERE** DepartureTime = @inputdate)

**BEGIN**

**SELECT** \* **FROM** Flight **WHERE** DepartureTime = @inputdate

**END**

**ELSE**

**BEGIN**

**Select** 'No flights available' **as** FlightStatus

**END**

**END**

**EXEC** sp\_flight '2002-02-02 00:00:00';

**2. Insert Data**

--Insert Table--

USE AirlineBookingSys

INSERT INTO Customer VALUES (1,'Smith', 'John', 'USA', 'MA', 'Boston', '137 Park Dr', '#1', 02215, 8572034041, 'asjdad@skc.com', 'Male', 'Ghds172gW');

INSERT INTO Customer VALUES (2,'Jones', 'David', 'USA', 'MA', 'Boston', '65 Brainerd Rd', '#12', 02134, 5672446296, 'adkfh@skc.net', 'Male', 'HGDjkd9730hg');

INSERT INTO Customer VALUES (3,'Johnson', 'Michael', 'USA', 'MA', 'Boston', '126 Jerset St', '#5', 02215, 8573947635, 'dsfjslaj@dkjs.cnd', 'Female', 'HSgdjs19jsh');

INSERT INTO Customer VALUES (4,'Lee', 'Chris', 'USA', 'MA', 'Boston', '221 Massachusetts Ave', '#28', 02115, 0923874562, 'asdkfjklsf@kjd.sdf', 'Female', 'sdkjflksjdfFG18237F');

INSERT INTO Customer VALUES (5,'Lee', 'Mike', 'USA', 'MA', 'Boston', '101 S Huntington Ave', '#13', 02130, 1287456372, 'asdfjaslfj@lksjd.ckj', 'Male', 'djHDK287@f');

INSERT INTO Customer VALUES (6,'Smith', 'Michael', 'USA', 'NY', 'New York', '261 Hudson St', '#19', 10013, 9836762392, 'lkjflj@jkdj.ckid', 'Female', 'sdkjf328Fn12');

INSERT INTO Customer VALUES (7,'Smith', 'David', 'USA', 'NY', 'New York', '225 Schermerhorn St', '#112', 11201, 1099827462, 'asdjflkj@id.cdid', 'Male', 'JGDj@kldk123');

INSERT INTO Customer VALUES (8,'Johnson', 'Mike', 'USA', 'NY', 'New York', '100 W 31st', '#123', 10001, 8503685943, 'asjdad@skc.com', 'Male', 'Ghds172gW');

INSERT INTO Customer VALUES (9,'Smith', 'John', 'USA', 'NY', 'New York', '100 W 31st', '#12', 10001, 981729762, 'asdkjf@kjds.cssd', 'Male', 'adskjG32Kdjbd');

INSERT INTO Customer VALUES (10,'Jones', 'Chris', 'USA', 'MA', 'Boston', '126 Jerset St', '#512', 02215, 8336239723, 'alsdkh@lksdm.cs', 'Female', 'kdf@sdkFHGH111');

INSERT INTO Customer VALUES (11,'Brown', 'Maria', 'USA', 'NY', 'New York', '225 Schermerhorn St', '#23', 11201,1928374312, 'sdkfj@ksjd.csd', 'Female', '111GGGdjdd');

INSERT INTO Customer VALUES (12,'Lopez', 'Emma', 'USA', 'CA', 'Los Angeles', '5555 W 6th St', '#21', 90036,1827398261, 'asdjlfj@kjd.csd', 'Female', 'jjGGG88229');

INSERT INTO Customer VALUES (13,'Lopez', 'Ava', 'UK', 'London', 'London', '17 Coulson St', '#32', 099232,9873234122, 'asdkjfh@d.c', 'Female', 'sdfkj###kdsWWW111');

INSERT INTO Customer VALUES (14,'Brown', 'Ava', 'UK', 'London', 'London', '20 Hertford St', '#23', 022312,192840432, 'fuscsn@lsdi.sc', 'Female', 'sdfHHHHh1212');

INSERT INTO Customer VALUES (15,'Zhang', 'Aiguo', 'China', 'BJ', 'Beijing', '18 Guangqumenwai St', '#192', 100022,12938481231, 'sadfkjs@sc.sd', 'Female', '111jhdhGGGG');

INSERT INTO Customer VALUES (16,'Zhang', 'Yi', 'China', 'SH', 'Shanghai', '17 renming St', '#12', 100004,1928323422, 'lknkj@ww.sss', 'Male', 'sdjfjdkkFF444');

INSERT INTO Customer VALUES (17,'Li', 'Aiguo', 'China', 'JS', 'Suzhou', '88 Suhui St', '#12', 215600,19283721302, 'skufsh@ldkl.cs', 'Male', '209238Gh');

INSERT INTO Customer VALUES (18,'Smith', 'Mia', 'USA', 'WA', 'Seattle', '50 University St', '#1', 98101, 938372232, 'sdklfj@lkdj.see', 'Female', 'sdfkljs@k9999KKK');

INSERT INTO Customer VALUES (19,'smith', 'Jane', 'USA', 'WA', 'Seattle', '50 University St', '#21', 98101,2938232323, 'slkduf@ldk.cls', 'Female', 'KKKdhhh!!11');

INSERT INTO Customer VALUES (20,'Zhang', 'Aiguo', 'China', 'JS', 'Suzhou', '88 Suhui St', '#17', 215600,13948393822, 'lkjlkj@lkjd.csm', 'Male', 'aa##2322HHH');

INSERT INTO [Order] VALUES (1,1,DEFAULT, 5023.12,'Sold by AATT Company', 1);

INSERT INTO [Order] VALUES (2,6,'2020-02-06 16:53:27.063', 15832.23,'Sold by GSSD Company, VIP User, Mobile Phone Platform', 1);

INSERT INTO [Order] VALUES (3,1,'2018-10-24 10:23:06.063', 4320,'Sold by AATT Company', 1);

INSERT INTO [Order] VALUES (4,6,DEFAULT, 24002.43,'Sold by GSSD Company, VIP User, Mobile Phone Platform',1);

INSERT INTO [Order] VALUES (5,4,'2019-06-13 20:17:03.023', 9820,'Sold by TGP Company, Desktop Platform',1);

INSERT INTO [Order] VALUES (6,17,'2019-10-14 23:53:09.023', 50123.76,'Sold by LSK Company, Super VIP User',1);

INSERT INTO [Order] VALUES (7,20,DEFAULT, 2012.74,'Sold by AATT Company, Special Price',1);

INSERT INTO [Order] VALUES (8,17,'2020-01-20 09:43:24.023', 8920,'Sold by GAK Company, User Fisrt Purchase',1);

INSERT INTO [Order] VALUES (9,3,'2019-08-13 21:34:57.000', 7932,'Sold by GAK Company',1);

INSERT INTO [Order] VALUES (10,9,'2019-08-13 21:34:57.000', 2034.79,'Sold by AATT Company',1);

INSERT INTO [Order] VALUES (11,11,'2017-10-24 19:53:31.146', 6702,'Sold by GDK Company, User Fisrt Purchase, No Refund Fee',1);

INSERT INTO [Order] VALUES (12,13,'2017-06-13 18:34:00.000', 2454.32,'Sold by Platform, User Fisrt Purchase',1);

INSERT INTO [Order] VALUES (13,1,'2019-12-02 10:03:42.000',234.34, 'Sold by Platform, Lowest Price, No Refund Fee',1);

INSERT INTO [Order] Values (14,8,'2018-07-30 21:17:06.000', 2234.78, 'Sold by Platform, Employee Discount, Refund Fee 15%',1);

INSERT INTO [Order] Values (15,15,'2020-03-31 15:47:58.003', 29311, 'Sold by ACG Company, Refund fee 5%',1);

INSERT INTO [Order] Values (16,15,'2019-02-28 15:47:58.003', 29311, 'Sold by ACG Company, Refund fee 5%',4);

INSERT INTO [Order] Values (17,15,'2011-04-01 15:47:58.003', 29311, 'Sold by ACG Company, Refund fee 5%',1);

INSERT INTO [Order] Values (18,15,'2019-03-30 15:47:58.003', 29311, 'Sold by ACG Company, Refund fee 5%',2);

INSERT INTO [Order] Values (19,15,'2016-12-31 15:47:58.003', 29311, 'Sold by ACG Company, Refund fee 5%',1);

INSERT INTO [Order] Values (20,15,'2012-11-29 15:47:58.003', 29311, 'Sold by ACG Company, Refund fee 5%',2);

INSERT INTO Luggage VALUES (1,1,13.1);

INSERT INTO Luggage VALUES (2,13,23);

INSERT INTO Luggage VALUES (3,17,22);

INSERT INTO Luggage VALUES (4,6,03);

INSERT INTO Luggage VALUES (5,7,12);

INSERT INTO Luggage VALUES (6,2,17);

INSERT INTO Luggage VALUES (7,1,13.5);

INSERT INTO Luggage VALUES (8,11,22);

INSERT INTO Luggage VALUES (9,19,22);

INSERT INTO Luggage VALUES (10,9,12);

INSERT INTO Luggage VALUES (11,10,6);

INSERT INTO Luggage VALUES (12,14,9.4);

INSERT INTO Luggage VALUES (13,17,11);

INSERT INTO Luggage VALUES (14,6,15);

INSERT INTO Luggage VALUES (15,11,16);

INSERT INTO Ticket VALUES (1,300,25,'Economic');

INSERT INTO Ticket VALUES (2,506.2,0,'Economic');

INSERT INTO Ticket VALUES (3,908.7,0,'Business');

INSERT INTO Ticket VALUES (4,409.5,10,'Economic');

INSERT INTO Ticket VALUES (5,809.6, 50,'Business');

INSERT INTO Ticket VALUES (6,1460,0,'First');

INSERT INTO Ticket VALUES (7,908.3,20,'Business');

INSERT INTO Ticket VALUES (8,417,0,'Economic');

INSERT INTO Ticket VALUES (9,900,5,'Business');

INSERT INTO Ticket VALUES (10,2000,70,'First');

INSERT INTO Ticket VALUES (11,500.4,0,'Economic');

INSERT INTO Ticket VALUES (12, 200, 90,'Economic');

INSERT INTO Ticket VALUES (13,496,0,'Economic');

INSERT INTO Ticket VALUES (14,1762.5,0,'First');

INSERT INTO Ticket VALUES (15,980.2,40,'Business');

INSERT INTO Payment VALUES (1,1,'Confirmed',5023.12,'2020-02-04 17:23:02.023','Card');

INSERT INTO Payment VALUES (2,2,'Pending',15832.23,'2020-02-06 17:23:02.023','Card');

INSERT INTO Payment VALUES (3,3,'Confirmed',4320,'2019-04-25 12:23:02.023','Card');

INSERT INTO Payment VALUES (4,4,'Confirmed',24002.43,'2020-04-01 12:23:02.023','Card');

INSERT INTO Payment VALUES (5,5,'Cancealed',9820,'2016-04-14 12:23:02.000','Card');

INSERT INTO Payment VALUES (6,6,'Cancealed',9372.5,'2019-10-14 15:23:02.000','Card');

INSERT INTO Payment VALUES (7,7,'Pending',10002,'2020-03-20 11:03:42.000','Card');

INSERT INTO Payment VALUES (8,8,'Confirmed',6290,'2019-11-20 20:37:09.000','Card');

INSERT INTO Payment VALUES (9,9,'Cancealed',7347,'2020-04-01 09:17:05.000','OnlinePayment');

INSERT INTO Payment VALUES (10,10,'Pending',2748,'2018-12-23 19:06:47.000','OnlinePayment');

INSERT INTO Payment VALUES (11,11,'Confirmed',209122,'2019-06-09 19:06:47.000','OnlinePayment');

INSERT INTO Payment VALUES (12,12,'Confirmed',2918.2,'2020-03-20 10:03:42.000','OnlinePayment');

INSERT INTO Payment VALUES (13,13,'Pending',281.3, '2019-12-02 10:03:42.000','OnlinePayment');

INSERT INTO Payment VALUES (14,14,'Cancealed',22032,'2019-04-17 13:29:56.000','OnlinePayment');

INSERT INTO Payment VALUES (15,15,'Pending',573.3,'2019-10-10 07:09:08.000','OnlinePayment');

INSERT INTO Payment VALUES (16,15,'Pending',573.3,'2019-02-28 15:47:58.003','Card');

INSERT INTO Payment VALUES (17,15,'Pending',573.3,'2011-04-01 15:47:58.003','Card');

INSERT INTO Payment VALUES (18,15,'Pending',573.3,'2019-03-30 15:47:58.003','OnlinePayment');

INSERT INTO Payment VALUES (19,15,'Pending',573.3,'2016-12-31 15:47:58.003','OnlinePayment');

INSERT INTO Payment VALUES (20,15,'Pending',573.3,'2012-11-29 15:47:58.003','OnlinePayment');

INSERT INTO [Card] VALUES (1, 'Card',123823823,'Emma', '09/24', 281,'Credit');

INSERT INTO [Card] VALUES (2, 'Card',298190831,'David', '01/28', 287,'Credit');

INSERT INTO [Card] VALUES (3, 'Card',488823232,'John', '03/20', 198,'Credit');

INSERT INTO [Card] VALUES (4, 'Card',123998232,'Helen', '02/24', 837,'Credit');

INSERT INTO [Card] VALUES (5, 'Card',923131232,'Amy', '04/24', 723,'Debit');

INSERT INTO [Card] VALUES (6, 'Card',123881923,'Mike', '09/27', 723,'Credit');

INSERT INTO [Card] VALUES (7, 'Card',923131232,'Amy', '04/24', 923,'Credit');

INSERT INTO [Card] VALUES (8, 'Card',288191132,'Bob', '11/22', 393,'Debit');

INSERT INTO [Card] VALUES (16, 'Card',288191432,'Bob', '11/24', 373,'Debit');

INSERT INTO [Card] VALUES (17, 'Card',288181232,'Bob', '09/25', 293,'Debit');

INSERT INTO OnlinePayment VALUES (9, 'OnlinePayment', 'gyh18393', 'ApplePay');

INSERT INTO OnlinePayment VALUES (10, 'OnlinePayment','sjd128384', 'ApplePay');

INSERT INTO OnlinePayment VALUES (11, 'OnlinePayment','818ndad', 'ApplePay');

INSERT INTO OnlinePayment VALUES (12, 'OnlinePayment','9983hdd', 'PayPal');

INSERT INTO OnlinePayment VALUES (13, 'OnlinePayment','8273jshed', 'GooglePay');

INSERT INTO OnlinePayment VALUES (14, 'OnlinePayment','hgh2823', 'ApplePay');

INSERT INTO OnlinePayment VALUES (15, 'OnlinePayment','88823jjds', 'PayPal');

INSERT INTO OnlinePayment VALUES (18, 'OnlinePayment','00023jjds', 'PayPal');

INSERT INTO OnlinePayment VALUES (19, 'OnlinePayment','99923jjds', 'PayPal');

INSERT INTO OnlinePayment VALUES (20, 'OnlinePayment','77723jjds', 'PayPal');

INSERT INTO Airline VALUES (1, 'Alaska Airlines', 'Seattle', 'AS','Star Alliance', 'WWW.sdsfs.com');

INSERT INTO Airline VALUES (2, 'Allegiant Air', 'Los Vegas', 'G4','Star Alliance', 'WWW.jhse.com');

INSERT INTO Airline VALUES (3, 'American Airlines', 'Dallas', 'AA','Star Alliance', 'WWW.sdfdsfs.com');

INSERT INTO Airline VALUES (4, 'Delta Airlines', 'Atlanta', 'DL','Star Alliance', 'WWW.sdfdsf.com');

INSERT INTO Airline VALUES (5, 'Frontier Airlines', 'Denver', 'F9','Star Alliance', 'WWW.jdnfsf.com');

INSERT INTO Airline VALUES (6, 'Hawaiian Airlines', 'Honolulu', 'HA','SkyTeam', 'WWW.kljhkhk.com');

INSERT INTO Airline VALUES (7, 'JetBlue Airways', 'New York', 'B6','SkyTeam', 'WWW.sdhfkjsdhf.com');

INSERT INTO Airline VALUES (8, 'Southwest Airlines', 'Dallas', 'WN', 'SkyTeam','WWW.lksdhfkjsdhf.com');

INSERT INTO Airline VALUES (9, 'Spirit Airlines', 'Atlantic', 'NK', 'SkyTeam','WWW.nsdfsf.com');

INSERT INTO Airline VALUES (10, 'SUn Coiuntry Airlines', 'Minneapolis', 'SY','SkyTeam', 'WWW.sdfsf.com');

INSERT INTO Airline VALUES (11, 'United Airlines', 'Chciago', 'UA', 'OneWorld','WWW.lksdfhsd.com');

INSERT INTO Airline VALUES (12, 'Air China', 'Beijing', 'CA', 'OneWorld','WWW.sdfsfs.com');

INSERT INTO Airline VALUES (13, 'Hainan Airlines', 'Hainan', 'HU', 'OneWorld','WWW.klshdfksf.com');

INSERT INTO Airline VALUES (14, 'Shanghai Airlines', 'Shanghai', 'FM', 'OneWorld','WWW.slfjslf.com');

INSERT INTO Airline VALUES (15, 'Shenzhen Airline', 'Shenzhen', 'ZH', 'OneWorld','WWW.sdfsdf.com');

INSERT INTO Aircraft VALUES (1,'A380',1990,400);

INSERT INTO Aircraft VALUES (2,'A370',1995,200);

INSERT INTO Aircraft VALUES (3,'A300',1980,300);

INSERT INTO Aircraft VALUES (4,'A390',2015,400);

INSERT INTO Aircraft VALUES (5,'A385',2010,370);

INSERT INTO Aircraft VALUES (6,'A375',2005,350);

INSERT INTO Aircraft VALUES (7,'A380',1990,400);

INSERT INTO Aircraft VALUES (8,'A400',2018,430);

INSERT INTO Aircraft VALUES (9,'A365',1985,270);

INSERT INTO Aircraft VALUES (10,'A350',1970,200);

INSERT INTO Flight VALUES (1,2,1,'BOS','2019-04-25 12:23:02.023','PVG','2019-04-26 12:23:02.023', 10, 0, 200, 'International Flight');

INSERT INTO Flight VALUES (2,1,9,'DSA','2019-06-06 12:23:02.023','PVG','2019-06-09 12:23:02.023', 15, 0, 300, 'International Flight');

INSERT INTO Flight VALUES (3,4,3,'LFD','2019-06-06 12:23:02.023','BOS','2019-06-07 12:23:02.023', 8, 3, 150, 'Domestic Flight');

INSERT INTO Flight VALUES (4,5,4,'PVG','2019-10-07 12:23:02.023','KDF','2019-10-09 12:23:02.023', 3, 1, 394, 'Domestic Flight');

INSERT INTO Flight VALUES (5,9,3,'PVG','2019-10-12 12:23:02.023','KDF','2019-10-13 12:23:02.023', 3, 3,293, 'Domestic Flight');

INSERT INTO Flight VALUES (6,9,7,'PVG','2019-11-07 12:23:02.000','KDF','2019-11-09 12:23:02.000', 4, 0, 673, 'Domestic Flight');

INSERT INTO Flight VALUES (7,11,5,'CSV','2018-03-08 20:23:02.023','PVG','2018-03-09 20:24:02.023', 7, 5,150, 'International Flight');

INSERT INTO Flight VALUES (8,11,10,'CSV','2018-03-09 20:23:02.023','PVG','2018-03-10 20:24:02.023', 7, 10,250, 'International Flight');

INSERT INTO Flight VALUES (9,3,6,'LFD','2020-04-03 12:23:02.023','BOS','2020-04-07 12:23:02.023', 4, 11, 160, 'Domestic Flight');

INSERT INTO Flight VALUES (10,3,7,'LFD','2020-06-09 12:23:02.023','CSV','2020-06-10 12:23:02.023', 10, 19, 193, 'Domestic Flight');

INSERT INTO Flight VALUES (11,6,1,'BOS','2018-03-17 12:23:02.023','PVG','2018-03-19 12:23:02.023', 23, 4, 301, 'International Flight');

INSERT INTO Flight VALUES (12,7,8,'SCD','2020-04-01 12:23:02.023','PVG','2019-04-01 12:23:02.023', 2, 0, 204, 'Domestic Flight');

INSERT INTO Flight VALUES (13,8,9,'CSV','2019-09-24 12:23:02.023','FGB','2019-09-26 12:23:02.023', 6, 13, 293, 'International Flight');

INSERT INTO Flight VALUES (14,9,7,'PVG','2018-11-11 12:23:02.023','DFD','2018-11-17 12:23:02.023', 3, 6, 200, 'Domestic Flight');

INSERT INTO Flight VALUES (15,14,6,'DSA','2019-04-25 12:23:02.023','PVG','2019-04-25 12:23:02.023', 12, 2, 200, 'International Flight');

INSERT INTO Flight VALUES (16,14,6,'DSA','2020-04-25 12:23:02.023','BOS','2020-04-25 12:23:02.023', 12, 2, 100, 'International Flight');

INSERT INTO Flight VALUES (17,14,6,'LFD','2018-04-25 12:23:02.023','BOS','2018-04-25 12:23:02.023', 12, 2, 50, 'Domestic Flight');

INSERT INTO Flight VALUES (18,14,6,'DSA','2017-03-25 12:23:02.023','DFD','2017-03-25 12:23:02.023', 12, 2, 600, 'International Flight');

INSERT INTO Flight VALUES (19,14,6,'DSA','2015-02-25 12:23:02.023','PVG','2015-02-25 12:23:02.023', 12, 2, 150, 'International Flight');

INSERT INTO Flight VALUES (20,14,6,'BOS','2020-04-25 12:23:02.023','PVG','2020-04-25 12:23:02.023', 12, 2, 200, 'Domestic Flight');

INSERT INTO InternationalFlight VALUES (1,'International Flight','Spanish', 'HC123123');

INSERT INTO InternationalFlight VALUES (2,'International Flight','English', 'HH831231');

INSERT INTO InternationalFlight VALUES (7,'International Flight','English', 'HA123921');

INSERT INTO InternationalFlight VALUES (8,'International Flight','Spanish', 'HC827313');

INSERT INTO InternationalFlight VALUES (11,'International Flight','English', 'HD282831');

INSERT INTO InternationalFlight VALUES (13,'International Flight','Chinese', 'HE231231');

INSERT INTO InternationalFlight VALUES (15,'International Flight','English', 'HD934343');

INSERT INTO InternationalFlight VALUES (16,'International Flight','English', 'HH933343');

INSERT INTO InternationalFlight VALUES (18,'International Flight','English', 'HC564343');

INSERT INTO InternationalFlight VALUES (19,'International Flight','English', 'HD784343');

INSERT INTO DomecticFlight VALUES (3,'Domestic Flight', 131231231);

INSERT INTO DomecticFlight VALUES (4,'Domestic Flight', 131283901);

INSERT INTO DomecticFlight VALUES (5,'Domestic Flight', 982424242);

INSERT INTO DomecticFlight VALUES (6,'Domestic Flight', 123881923);

INSERT INTO DomecticFlight VALUES (9,'Domestic Flight', 398138120);

INSERT INTO DomecticFlight VALUES (10,'Domestic Flight', 398902383);

INSERT INTO DomecticFlight VALUES (12,'Domestic Flight', 398103813);

INSERT INTO DomecticFlight VALUES (14,'Domestic Flight', 198237917);

INSERT INTO DomecticFlight VALUES (17,'Domestic Flight', 193567917);

INSERT INTO DomecticFlight VALUES (20,'Domestic Flight', 346797917);

**3.Stored Procedures**

**SP1:** Renew password policy

RightCustomerPassword which takes @CustomerID and @new\_CustomerPassword as input parameters and returns @message as an output parameter. The stored procedure performs the following business rules:

* 1. If @CustomerID is not found, it returns the message “Invalid Customer ID”
  2. If @new\_CustomerPassword is not the same as the current customer password, it must update the customer password and return the message “Customer password has updated”
  3. If @new\_CustomerPassword is the same as the customer password, it does nothing and return the message “The new password cannot be the same as the old one”

CREATE PROCEDURE RightCustomerPassword (@CustomerID INT, @new\_CustomerPassword VARCHAR(20), @message varchar (60) output)

AS

IF @CustomerID not in (SELECT CustomerID FROM Customer)

BEGIN

SET @message= 'Invalid Customer ID'

PRINT @message

END

IF @new\_CustomerPassword =(SELECT CustomerPassword FROM Customer WHERE CustomerID =@CustomerID)

BEGIN

UPDATE Customer SET CustomerPassword = @new\_CustomerPassword WHERE CustomerID =@CustomerID

SET @message= 'The new password cannot be the same as the old one'

PRINT @message

END

IF @new\_CustomerPassword != (SELECT CustomerPassword FROM Customer WHERE CustomerID =@CustomerID)

BEGIN

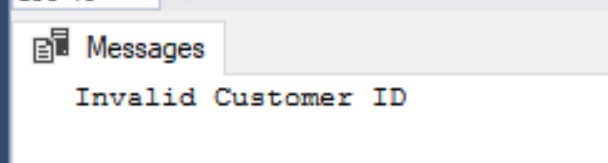
SET @message= 'Customer password has updated'

PRINT @message

END

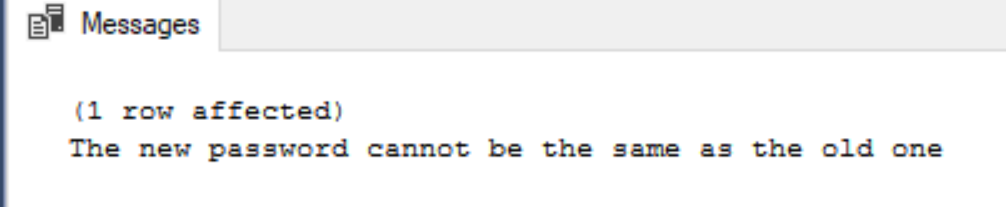
DECLARE @message varchar(60)

EXECUTE RightCustomerPassword @CustomerID ='200', @new\_CustomerPassword ='Ghds172GW', @message='a'



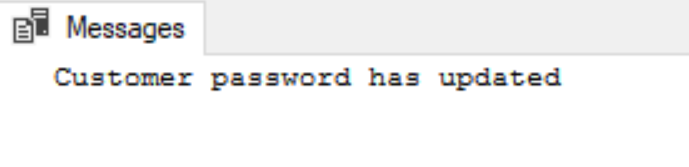
DECLARE @message varchar(60)

EXECUTE RightCustomerPassword @CustomerID = 1, @new\_CustomerPassword ='Ghds172gW', @message='a'



DECLARE @message varchar(60)

EXECUTE RightCustomerPassword @CustomerID = 1, @new\_CustomerPassword ='ghds666GW', @message='a'



**SP2:** Show customer’s address information

CustomerAddressInfo which takes @CustomerID as an input parameter, and outputs a result set which includes the following fields: CustomerID, Country, State, City, Street, Apartment.

CREATE PROCEDURE CustomerAddressInfo (@CustomerID INT)

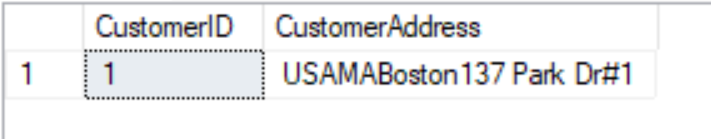
AS

SELECT CustomerID, Country + [State] + City + Street + Apartment AS CustomerAddress

FROM Customer

WHERE CustomerID =@CustomerID

EXECUTE CustomerAddressInfo @CustomerID = 1



**SP3:** Find the customers who have many orders

**Input:** Number of orders  
**Output:** (data set) customer

CREATE PROCEDURE RepeatedCustomers @Number int

AS

DECLARE @CustomerID INT

BEGIN

SELECT [Customer].[CustomerID], Count([OrderID]) NumberOrder

FROM [dbo].[Customer] LEFT JOIN [dbo].[Order]

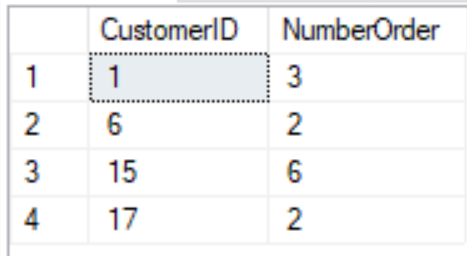
ON [Customer].[CustomerID] = [Order].[CustomerID]

GROUP BY [Customer].[CustomerID]

HAVING Count([OrderID])>= @Number

END;

Execute RepeatedCustomers 2



**SP4:** Check whether a customer saved his/her payment method.

**Input:** CustomerID  
**Output:** PaymentMethod

CREATE PROCEDURE Checksave @CustomerID INT

AS

BEGIN

SELECT Customer.CustomerID, Payment.PaymentType

FROM Customer

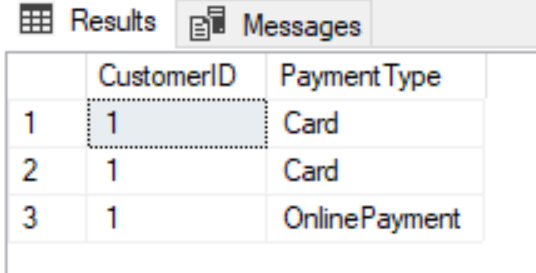
Left Join [Order] on Customer.CustomerID = [Order].CustomerID

Join Payment on [Order].OrderID = Payment.OrderID

WHERE Customer.CUstomerID = @CUstomerID

END;

EXECUTE Checksave @CustomerID = 1



**SP5:** To retrieve customer details along with fullname, order detail., Luggageweight

CREATE PROCEDURE sp\_CustomerOrderLuggage

AS BEGIN

Select DISTINCT c.Firstname+' '+c.Lastname as FullName,o.OrderDetail,o.OrderQuantity,l.LuggageWeight

FROM Customer c

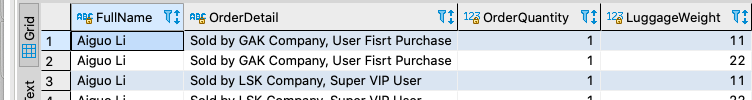
inner join [Order] o

on c.CustomerID = o.CustomerID

inner join Luggage l

on c.CustomerID = l.CustomerID

END

EXEC sp\_CustomerOrderLuggage****

**SP6:** Details of a customers payment details along with names, email, order time, order quantity, payment status, payment type, online payment methoda and card no.

CREATE PROCEDURE sp\_CustomerOrderPayment

AS BEGIN

SELECT c.Lastname ,c.Firstname,c.CustomerEmail, o.OrderTime,o.OrderQuantity,p.PaymentStatus,p.PaymentType,op.OnlinePaymentMethod,

cn.CardNo

FROM Customer c

inner join [Order] o

on c.CustomerID = o.CustomerID

inner join Payment p

on o.OrderID = p.OrderID

inner join OnlinePayment op

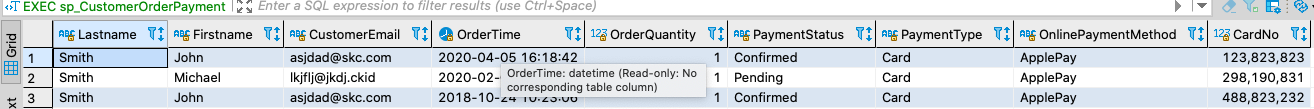
on p.PaymentNo = op.OPaymentNo

inner join Card cn

on p.PaymentNo = cn.CPaymentNo

END

EXEC sp\_CustomerOrderPayment

****

**SP7:** To check whether the flight is available during that departure time or not

CREATE PROCEDURE sp\_flightdetail

@inputdate DateTime

AS

BEGIN

IF EXISTS (SELECT \* FROM Flight WHERE DepartureTime = @inputdate)

BEGIN

SELECT \* FROM Flight WHERE DepartureTime = @inputdate

END

ELSE

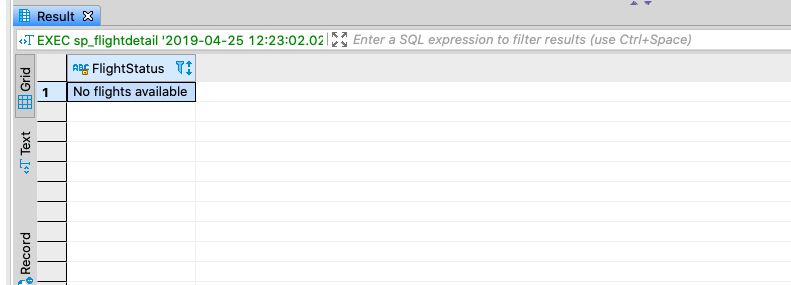
BEGIN

Select 'No flights available' as FlightStatus

END

END

EXEC sp\_flightdetail '2019-04-25 12:23:02.025';

****

**SP8:**

Stored procedure named SearchFlight which takes : @DepartureAirport @ArrivalAirport @DepartureMonth @DepartureDay as input, allows users to search available flights from one airport to another airport in one specific day. All these unknown variables are from Input. If there is any flight satisfying the requirement, other data about AirlineID, AirlineName, AircraftID and AircraftType are also listed. If there is no flight satisfying the requirement, it will tell ‘No Flight Available’.

INPUT: @DepartureAirport @ArrivalAirport @DepartureMonth @DepartureDay

CREATE PROC SearchFlight

@DepartureAirport VARCHAR(10),

@ArrivalAirport VARCHAR(10),

@DepartureMonth DECIMAL(2,0),

@DepartureDay DECIMAL(2,0)

AS

BEGIN

IF @DepartureAirport IN (SELECT DepartureAirport FROM Flight)

AND @ArrivalAirport IN (SELECT ArrivalAirport FROM Flight)

AND @DepartureMonth IN (SELECT MONTH(DepartureTime) FROM Flight)

AND @DepartureDay IN (SELECT DAY(DepartureTime) FROM Flight)

SELECT f.AirlineID, AirlineName, DepartureAirport, ArrivalAirport, DepartureTime, ArrivalTime, DuartionByHrs, f.AircraftID, AircraftType

FROM Flight f JOIN Airline al ON f.AirlineID = al.AirlineID

JOIN Aircraft af ON f.AircraftID = af.AircraftID

WHERE DepartureAirport = @DepartureAirport

AND ArrivalAirport = @ArrivalAirport

AND MONTH(DepartureTime) = @DepartureMonth

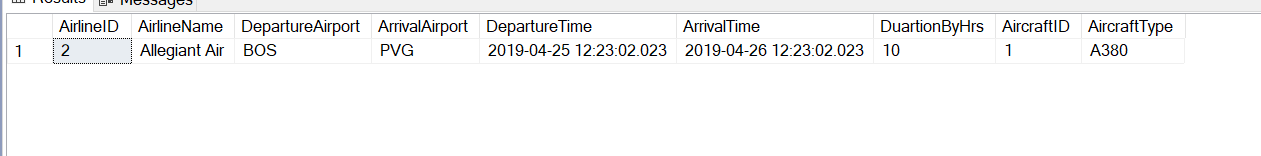
AND DAY(DepartureTime) = @DepartureDay

ELSE

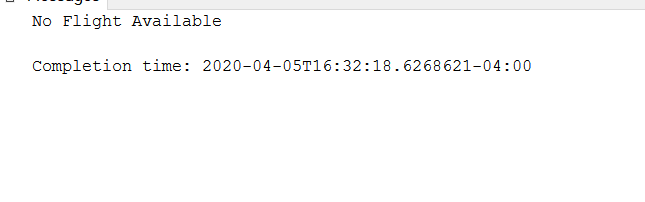
PRINT 'No Flight Available'

END

EXEC SearchFLight 'BOS', 'PVG', 04, 25



EXEC SearchFLight 'BOS', 'PVG', 04, 29



**4.TRIGGER**

This trigger query updates the Total amount of 200$ as per to the luggage weight > 20

ALTER TRIGGER trgAfterUpdate ON [dbo].[Luggage]

FOR UPDATE

AS

Begin

declare @PaymentNo int

select \* FROM inserted

if exists (SELECT \* FROM inserted where LuggageWeight > 20)

BEGIN

SET @PaymentNo = (SELECT PaymentNo from Payment p

inner join [Order]o

on p.orderID = o.OrderID

inner join Luggage l

on o.CustomerID = l.CustomerID

where l.CustomerID = (select CustomerID from inserted))

update payment set TotalAmount = TotalAmount+200 where paymentNo = @PaymentNo

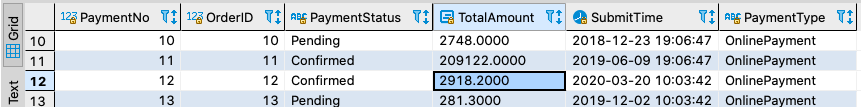
END

end

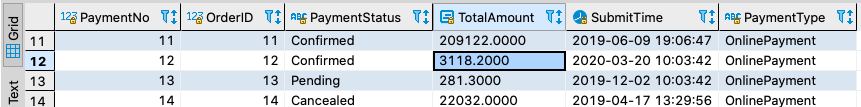
GO

UPDATE luggage SET LuggageWeight = 22 where LuggageID = 2

**Before trigger**

****

**After trigger**

****

**5.Views**

**View1**: the amount of order group by customer’s gender

Create VIEW AmountofGender as

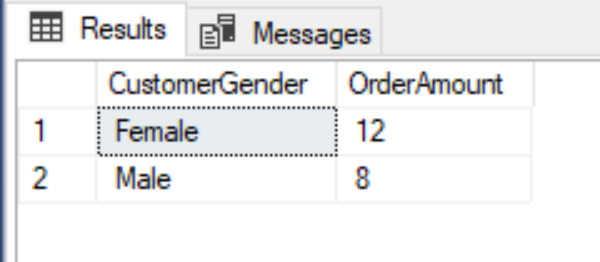
select c. CustomerGender, count(o.CustomerID) OrderAmount

from [order] o, Customer c

where o.CustomerID = c.CustomerID

group by c. CustomerGender;

select \* from AmountofGender

****

**View2**: the amount of flight group by aircraft’s type

Create VIEW AmountofAircraft as

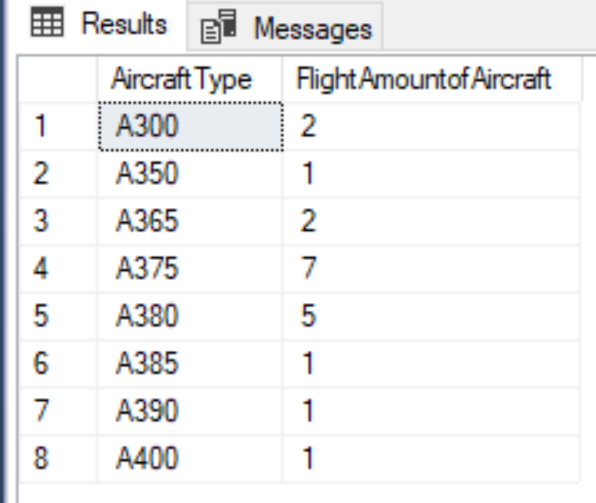
select Aircraft.AircraftType, count(Flight.AircraftID) FlightAmountofAircraft

from Aircraft, Flight

where Aircraft.AircraftID = Flight.AircraftID

group by Aircraft.AircraftType;

select \* from AmountofAircraft



**View3**: the amount of flight group by airline’s name

Create VIEW AmountofAirline as

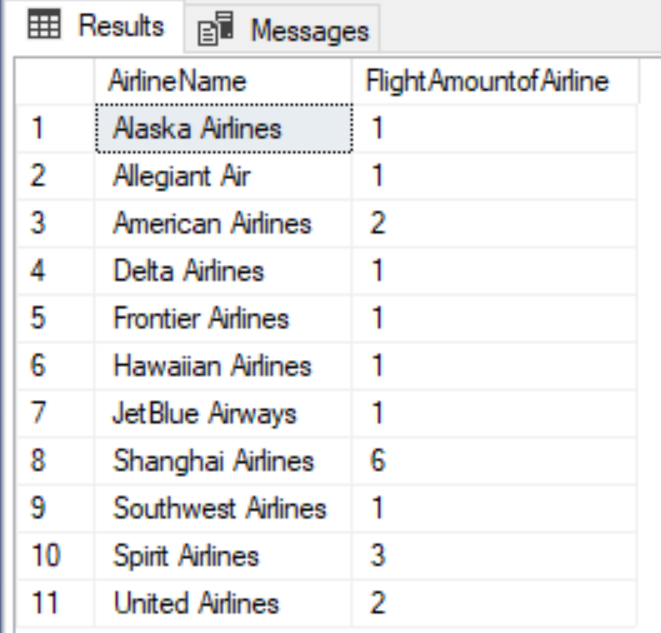
select Airline.AirlineName, count(Flight.AirlineID) FlightAmountofAirline

from Airline, Flight

where Airline.AirlineID = Flight.AirlineID

group by Airline.AirlineName;

select \* from AmountofAirline



**View4**: the amount of order group by payment types

Create VIEW AmountofPaymentType as

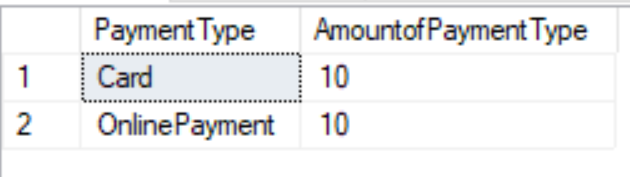
select Payment.PaymentType, count(Payment.OrderID) AmountofPaymentType

from Payment, [Order]

where Payment.OrderID = [Order].OrderID

group by Payment.PaymentType;

select \* from AmountofPaymentType



**6.Table-level CHECK Constraints**

This check verifies that the departure airport cannot be the same as arrival airport, otherwise the operation returns a table-level check constraint violation.

Alter Table Flight

Add Constraint CityCheck CHECK (DepartureAirport<>ArrivalAirport)

--test violation

insert into Flight VALUES (30,2,1,'BOS','2019-04-25 12:23:02.023','BOS','2019-04-26 12:23:02.023', 10, 0, 200, 'Domestic Flight');

Msg 547, Level 16, State 0, Line 5

The INSERT statement conflicted with the CHECK constraint "CityCheck". The conflict occurred in database "AirlineBookingSys", table "dbo.Flight".

The statement has been terminated.

--test without violation

insert into Flight VALUES (30,2,1,'BOS','2019-04-25 12:23:02.023','Los','2019-04-26 12:23:02.023', 10, 0, 200, 'Domestic Flight');

(1 row affected)

**7.Computed Columns based on a function**

**CC1:** Compute the aircraft age using DATEDIFF and YEAR function.

Create Procedure AircraftAge @CheckDate date

AS

Begin

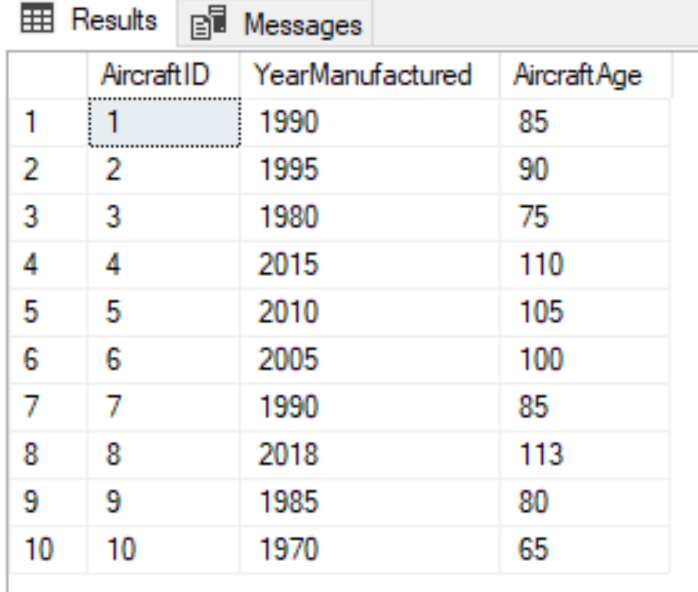
Select AircraftID, YearManufactured, DATEDIFF(year, Year(@CheckDate), YearManufactured) as AircraftAge

From Aircraft

where DATEDIFF(year, Year(@CheckDate), YearManufactured) >= 0

End;

Exec AircraftAge @CheckDate='04/04/2020'



**8.Column Data Encryption**Column Data Encryption: in Card table, Encrypt the value of Security Code since they are associated with card account security.

--create database master key  
CREATE MASTER KEY ENCRYPTION BY PASSWORD ='Pa$$word'  
--CREATE CERTIFICATE  
CREATE CERTIFICATE CertTest  
with SUBJECT = 'Test Certificate'  
GO  
--ENCRYPT SYMMETRIC KEY, BY CERTIFICATE  
CREATE SYMMETRIC KEY SymmetricByCert  
WITH ALGORITHM = AES\_256  
ENCRYPTION BY CERTIFICATE CertTest;  
GO  
--Change column of SecurityCode into varbinary type

SELECT CPaymentNo,

PaymentType,

CardNo,

NameOnCard,

ExpDate,

CardType,

SecurityCode\_Encrypt = CONVERT(varbinary(500), SecurityCode)

INTO Card\_Encrypt

FROM [Card]

--OPEN SYMMETRIC KEY, DECRYPTION BY CERTIFICATE

OPEN SYMMETRIC KEY SymmetricByCert

DECRYPTION BY CERTIFICATE CertTest

--Encrypt SecurityCode\_Encrypt By SYMMETRIC Key and insert into [Card]\_Encrypt

update Card\_Encrypt

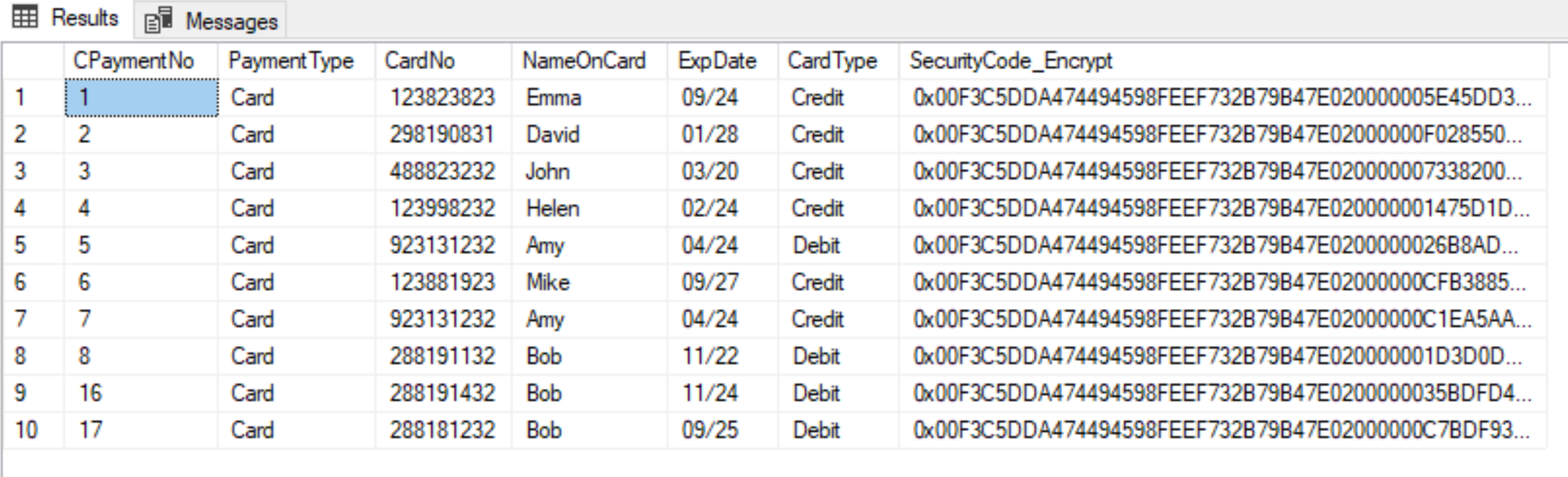
set SecurityCode\_Encrypt = EncryptByKey(KEY\_GUID('SymmetricByCert'), SecurityCode\_Encrypt)

select convert (varchar(500), SecurityCode\_Encrypt)

from Card\_Encrypt

--Now we cannot see the value which is encrypted

select \* from Card\_Encrypt



--decipher the data with DecryptByKey function

OPEN SYMMETRIC KEY SymmetricByCert

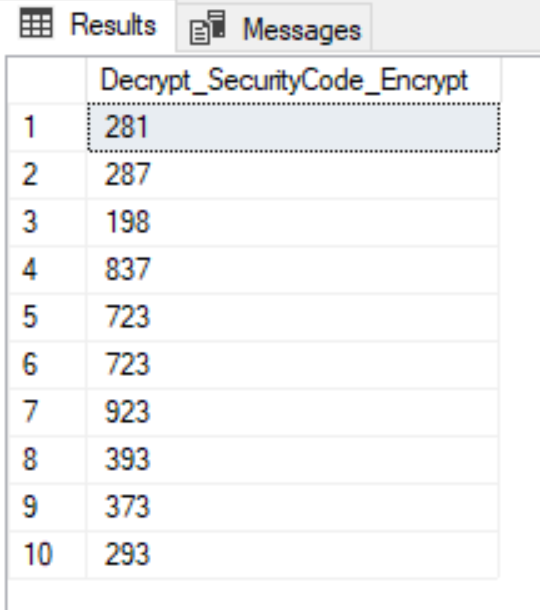
DECRYPTION BY CERTIFICATE CertTest

select convert(varchar(25), DecryptByKey(SecurityCode\_Encrypt)) Decrypt\_SecurityCode\_Encrypt

from Card\_Encrypt

CLOSE SYMMETRIC KEY SymmetricByCert;

--Now we can see the value back



**9.Non-clustered Indexes**Set the column of CustomerFirstName in the table of Customer as Non-clustered Indexes to speed up  
the search of customer’s information by First Name.

CREATE NONCLUSTERED INDEX [NCindex\_Customer] ON [dbo].[Customer]

(

[FirstName] ASC

)WITH (SORT\_IN\_TEMPDB = OFF, DROP\_EXISTING = OFF, IGNORE\_DUP\_KEY = OFF, ONLINE = OFF)

ON [PRIMARY]

use AirlineBookingSys

EXEC Sp\_helpindex Customer

