Author: Chunli Yu [422888242][cyu22@syr.edu]

Project 2

Contents

1. Design adjustments

- *use my own model*

(1) Add tables PlaneModels, Service to track workBeingDone, Gates, Reservations and TicketsInTrips
(2) Break down table Entertainments into AirplainEntertainments and EntertainmentOptions;
Break down table Refreshments into FlightRefreshments and RefreshmentOptions;
Break down table AirportFees into AirportFees and FeeTypes.
(3) Separate AirportLocations from table Airports
Separate FlightGates, FlightsTimes from table ScheduledFlights.
(4) Incorporate table CustomerTransactions to Trips.
Incorporate table Refunds to Feetypes.
(5) Adjust Payments to enable one trip paid by many times so that extra luggage fee can be paid when the customer has already bought the ticked.
(6) Abandon tables LandingStatus, ScheduledFlightStatus which can be tracked by checking ScheduledFlightTimes.

2. Table creation

```
/*
 * CSE.581.M004 SPRING 15.Intro D/Base Mngmt Syst.
 * Created by Chunli Yu [cyu22@syr.edu]
 * Last modification date: 2015-04-18 02:52:59.963
 * tables Creation
 */

-- Table: Personals
IF OBJECT_ID('Personals', 'U') IS NOT NULL DROP TABLE Personals
CREATE TABLE Personals (
    PersonId VARCHAR(10) PRIMARY KEY,
    FirstName VARCHAR(30)  NOT NULL,
    MiddleName VARCHAR(30),
    LastName VARCHAR(30)  NOT NULL,
    PhoneNumber VARCHAR(15)  NOT NULL,
);

-- Table: Addresses
IF OBJECT_ID('Addresses', 'U') IS NOT NULL DROP TABLE Addresses
CREATE TABLE Addresses (
    PersonId VARCHAR(10)  PRIMARY KEY REFERENCES Personals(PersonId),
    Street1 VARCHAR(100)  NOT NULL,
    Street2 VARCHAR(100),
    City VARCHAR(15)  NOT NULL,
    USState VARCHAR(2)  NOT NULL,
    Zipcode VARCHAR(5)   NOT NULL,
);

-- Table: Jobs
IF OBJECT_ID('Jobs', 'U') IS NOT NULL DROP TABLE Jobs
CREATE TABLE Jobs (
    JobId INT PRIMARY KEY IDENTITY(1, 1),
    JobTitle VARCHAR(30)  NOT NULL,
    JobDescription TEXT,
);

-- Table: Customers
IF OBJECT_ID('Customers', 'U') IS NOT NULL DROP TABLE Customers
CREATE TABLE Customers (
```

```
    CustomerId VARCHAR(10)  PRIMARY KEY REFERENCES Personals(PersonId),
    FrequentFlyerNumber VARCHAR(15),
);

-- Table: CreditCardTypes
IF OBJECT_ID('CreditCardTypes', 'U') IS NOT NULL DROP TABLE CreditCardTypes
CREATE TABLE CreditCardTypes (
    CardTypeId INT PRIMARY KEY IDENTITY(1, 1),
    CardTypeName VARCHAR(30)  NOT NULL,
);

-- Table: CreditCardDetails
IF OBJECT_ID('CreditCardDetails', 'U') IS NOT NULL DROP TABLE CreditCardDetails
CREATE TABLE CreditCardDetails (
    CardNumber VARCHAR(20)  PRIMARY KEY,
    Holder VARCHAR(10)  NOT NULL REFERENCES Customers(CustomerId),
    CardType INT  NOT NULL REFERENCES CreditCardTypes(CardTypeId),
    ExpirationMonth INT  NOT NULL,
    ExpirationYear INT  NOT NULL,
);

-- Table: Employees
IF OBJECT_ID('Employees', 'U') IS NOT NULL DROP TABLE Employees
CREATE TABLE Employees (
    EmployeeId VARCHAR(10) PRIMARY KEY REFERENCES Personals(PersonId),
    JobId INT NOT NULL REFERENCES Jobs(JobId),
    DirectSupervisor VARCHAR(10) REFERENCES Employees(EmployeeId),
    SSN VARCHAR(20)  NOT NULL,
    Salary DECIMAL(10,2)  NOT NULL,
);

-- Table: PropulsionMethods
IF OBJECT_ID('PropulsionMethods', 'U') IS NOT NULL DROP TABLE PropulsionMethods
CREATE TABLE PropulsionMethods (
    PropulsionId INT PRIMARY KEY IDENTITY(1, 1),
    PropulsionMethod VARCHAR(30)  NOT NULL,
);

-- Table: Manufacturers
IF OBJECT_ID('Manufacturers', 'U') IS NOT NULL DROP TABLE Manufacturers
CREATE TABLE Manufacturers (
    ManufacturerId VARCHAR(10) PRIMARY KEY,
    ManufacturerName VARCHAR(50)  NOT NULL,
);

-- Table: PlaneModels
IF OBJECT_ID('PlaneModels', 'U') IS NOT NULL DROP TABLE PlaneModels
CREATE TABLE PlaneModels (
    ManufacturerId VARCHAR(10) REFERENCES Manufacturers(ManufacturerId),
    ModelNumber VARCHAR(10)  NOT NULL,
    PropulsionMethod INT NOT NULL REFERENCES PropulsionMethods(PropulsionId),
    NumberOfPilots INT  NOT NULL,
    NumberOfAttendants INT  NOT NULL,
    FlyRange INT  NOT NULL,
       PRIMARY KEY(ManufacturerId,ModelNumber)
);

-- Table: EntertainmentOptions
```

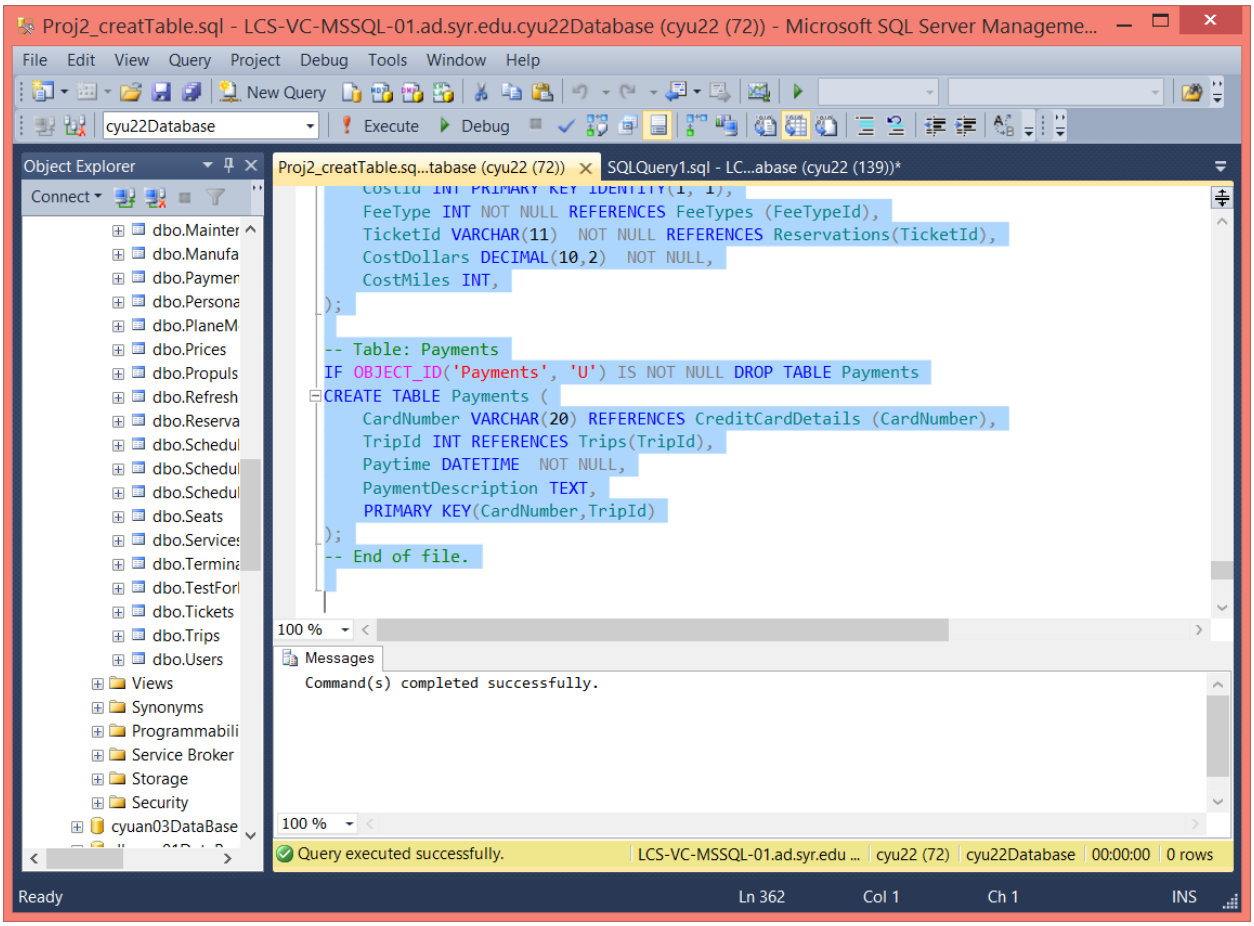
```
IF OBJECT_ID('EntertainmentOptions', 'U') IS NOT NULL DROP TABLE EntertainmentOptions
CREATE TABLE EntertainmentOptions (
    EntertainmentOptionId  INT  PRIMARY KEY IDENTITY(1, 1),
    EntertainmentName VARCHAR(30)   NOT NULL,
);

-- Table: AirplaneEntertainments
IF OBJECT_ID('AirplaneEntertainments', 'U') IS NOT NULL DROP TABLE AirplaneEntertainments
CREATE TABLE AirplaneEntertainments (
    ManufacturerId VARCHAR(10),
    ModelNumber VARCHAR(10),
    EntertainmentOptionId  INT REFERENCES EntertainmentOptions (EntertainmentOptionId),
    PRIMARY KEY (ManufacturerId,ModelNumber, EntertainmentOptionId),
    FOREIGN KEY  (ManufacturerId, ModelNumber) REFERENCES PlaneModels (ManufacturerId,
ModelNumber)
);

-- Table: CanWork
IF OBJECT_ID('CanWork', 'U') IS NOT NULL DROP TABLE CanWork
CREATE TABLE CanWork (
    EmployeeId VARCHAR(10) REFERENCES Employees(EmployeeId),
    ManufacturerId VARCHAR(10),
    ModelNumber VARCHAR(10),
    PRIMARY KEY(EmployeeId, ManufacturerId,ModelNumber),
    FOREIGN KEY  (ManufacturerId, ModelNumber) REFERENCES PlaneModels (ManufacturerId,
ModelNumber)
);

-- Table: Airports
IF OBJECT_ID('Airports', 'U') IS NOT NULL DROP TABLE Airports
CREATE TABLE Airports (
    AirportId VARCHAR(5) PRIMARY KEY,
    AirportName VARCHAR(50)  NOT NULL,
    HangarCapacity INT NOT NULL DEFAULT 0,
);

-- Table: AirportLocations
IF OBJECT_ID('AirportLocations', 'U') IS NOT NULL DROP TABLE AirportLocations
CREATE TABLE AirportLocations (
    AirportId VARCHAR(5) PRIMARY KEY REFERENCES Airports(AirportId),
    City VARCHAR(15)   NOT NULL,
    USState VARCHAR(2)   NOT NULL
);

-- Table: FeeTypes
IF OBJECT_ID('FeeTypes', 'U') IS NOT NULL DROP TABLE FeeTypes
CREATE TABLE FeeTypes (
    FeeTypeId INT PRIMARY KEY IDENTITY(1, 1),
    FeeTypeName VARCHAR(30)  NOT NULL,
);

-- Table: AirportFees
IF OBJECT_ID('AirportFees', 'U') IS NOT NULL DROP TABLE AirportFees
CREATE TABLE AirportFees (
    AirportId VARCHAR(5) REFERENCES Airports(AirportId),
    FeeTypeId INT REFERENCES FeeTypes(FeeTypeId),
    Fees DECIMAL(8,2),
    PRIMARY KEY(AirportId,FeeTypeId)
```

```
);

-- Table: CanLand
IF OBJECT_ID('CanLand', 'U') IS NOT NULL DROP TABLE CanLand
CREATE TABLE CanLand (
    AirportId VARCHAR(5) REFERENCES Airports(AirportId),
    ManufacturerId VARCHAR(10),
    ModelNumber VARCHAR(10),
    PRIMARY KEY(AirportId,ManufacturerId,ModelNumber),
    FOREIGN KEY  (ManufacturerId, ModelNumber) REFERENCES PlaneModels (ManufacturerId,
ModelNumber)
);

-- Table: Availabilities
IF OBJECT_ID('Availabilities', 'U') IS NOT NULL DROP TABLE Availabilities
CREATE TABLE Availabilities (
    AvailabilityId INT  PRIMARY KEY IDENTITY(1, 1),
    AvailabilityStatus VARCHAR(30)  NOT NULL,
);

-- Table: Airplanes
IF OBJECT_ID('Airplanes', 'U') IS NOT NULL DROP TABLE Airplanes
CREATE TABLE Airplanes (
    PlaneId VARCHAR(6)  PRIMARY KEY,
    ManufactureId VARCHAR(10)  NOT NULL,
    ModelNumber VARCHAR(10)  NOT NULL,
    ManufactureDate DATE  NOT NULL,
    AvailabilityId INT NOT NULL,
    CurrentLocation VARCHAR(5) REFERENCES Airports(AirportId),
);

-- Table: Services
IF OBJECT_ID('Services', 'U') IS NOT NULL DROP TABLE Services
CREATE TABLE Services (
    ServiceId INT PRIMARY KEY IDENTITY(1, 1),
    ServiceName VARCHAR(100)  NOT NULL,
    ServiceDescription TEXT,
);

-- Table: MaintenanceRecords
IF OBJECT_ID('MaintenanceRecords', 'U') IS NOT NULL DROP TABLE MaintenanceRecords
CREATE TABLE MaintenanceRecords (
    MaintenanceId INT  PRIMARY KEY IDENTITY(1, 1),
    PlaneId VARCHAR(6)  NOT NULL REFERENCES Airplanes(PlaneId),
    ExpectedDate DATE  NOT NULL,
    MantenancedDate DATE,
    WorkBeingDone INT  NOT NULL REFERENCES Services(ServiceId),
    MantenanceMan VARCHAR(10)  NOT NULL REFERENCES Employees(EmployeeId),
    MaintenanceDescription TEXT,
);

-- Table: FlightRoutes
IF OBJECT_ID('FlightRoutes', 'U') IS NOT NULL DROP TABLE FlightRoutes
CREATE TABLE FlightRoutes (
    FlightRoutId INT  PRIMARY KEY IDENTITY(1, 1),
    DepartureAirpport VARCHAR(5)  NOT NULL REFERENCES Airports(AirportId),
    ArrivalAirport VARCHAR(5)  NOT NULL REFERENCES Airports(AirportId),
    FlightDuration TIME(0)  NOT NULL,
```

```
    FlightDistance DECIMAL(10,2) NOT NULL,
);

-- Table: RefreshmentOptions
IF OBJECT_ID('RefreshmentOptions', 'U') IS NOT NULL DROP TABLE RefreshmentOptions
CREATE TABLE RefreshmentOptions (
    RefreshmentOptionId INT PRIMARY KEY IDENTITY(1, 1),
    RefreshmentName VARCHAR(30)  NOT NULL,
);

-- Table: FlightRefreshments
IF OBJECT_ID('FlightRefreshments', 'U') IS NOT NULL DROP TABLE FlightRefreshments
CREATE TABLE FlightRefreshments (
    FlightRoutId INT REFERENCES FlightRoutes(FlightRoutId),
    RefreshmentId INT REFERENCES RefreshmentOptions(RefreshmentOptionId),
    Cost INT,
    PRIMARY KEY(FlightRoutId,RefreshmentId)
);

-- Table: ScheduledFlights
IF OBJECT_ID('ScheduledFlights', 'U') IS NOT NULL DROP TABLE ScheduledFlights
CREATE TABLE ScheduledFlights (
    ScheduledFlightId INT PRIMARY KEY IDENTITY(1, 1),
    FlightRouteId INT NOT NULL REFERENCES FlightRoutes(FlightRoutId),
    FlightNumber VARCHAR(15)  NOT NULL,
    AvailableSeats INT  NOT NULL DEFAULT 0,
    PlaneId VARCHAR(6)  NOT NULL REFERENCES Airplanes(PlaneId),
);

-- Table: Crew
IF OBJECT_ID('Crew', 'U') IS NOT NULL DROP TABLE Crew
CREATE TABLE Crew (
    EmployeeId VARCHAR(10)   REFERENCES Employees(EmployeeId),
    ScheduledFlightId INT  REFERENCES ScheduledFlights(ScheduledFlightId),
    PRIMARY KEY(EmployeeId,ScheduledFlightId)
);

-- Table: Terminals
IF OBJECT_ID('Terminals', 'U') IS NOT NULL DROP TABLE Terminals
CREATE TABLE Terminals (
    TerminalId INT PRIMARY KEY IDENTITY(1, 1),
    TerminalName VARCHAR(10)  NOT NULL,
    AirportId VARCHAR(5)  NOT NULL REFERENCES Airports(AirportId),
    TerminalDescription TEXT,
);

-- Table: Gates
IF OBJECT_ID('Gates', 'U') IS NOT NULL DROP TABLE Gates
CREATE TABLE Gates (
    GateId INT PRIMARY KEY IDENTITY(1, 1),
    GateName VARCHAR(2)  NOT NULL,
    TerminalId INT NOT NULL REFERENCES Terminals(TerminalId),
);

-- Table: ScheduledFlightGates
IF OBJECT_ID('ScheduledFlightGates', 'U') IS NOT NULL DROP TABLE ScheduledFlightGates
CREATE TABLE ScheduledFlightGates (
    ScheduledFlightId INT  PRIMARY KEY REFERENCES ScheduledFlights(ScheduledFlightId),
```

```
    ScheduledDepartureGate INT  NOT NULL REFERENCES Gates(GateId),
    ScheduledArrivalGate INT  NOT NULL REFERENCES Gates(GateId),
    ActuralDepartureGate INT REFERENCES Gates(GateId),
    ActuralArrivalGate INT REFERENCES Gates(GateId),
);

-- Table: ScheduledFlightTimes
IF OBJECT_ID('ScheduledFlightTimes', 'U') IS NOT NULL DROP TABLE ScheduledFlightTimes
CREATE TABLE ScheduledFlightTimes (
    ScheduledFlightId INT  PRIMARY KEY REFERENCES ScheduledFlights(ScheduledFlightId),
    ScheduledDepartureTime DATETIME  NOT NULL,
    ScheduledArrivalTime DATETIME  NOT NULL,
    ProjectedDepartureTime DATETIME,
    ProjectedArrivalTime DATETIME,
    ActuralDepartureTime DATETIME,
    ActuralArrivalTime DATETIME,
);

-- Table: Classes
IF OBJECT_ID('Classes', 'U') IS NOT NULL DROP TABLE Classes
CREATE TABLE Classes (
    ClassId INT PRIMARY KEY IDENTITY(1, 1),
    ClassName VARCHAR(15)  NOT NULL,
);

-- Table: Seats
IF OBJECT_ID('Seats', 'U') IS NOT NULL DROP TABLE Seats
CREATE TABLE Seats (
    SeatId INT PRIMARY KEY IDENTITY(1, 1),
    ClassId INT NOT NULL REFERENCES Classes(ClassId),
    RowNumber INT  NOT NULL,
    ColumnLetter CHAR(1)  NOT NULL,
    ManufacturerId VARCHAR(10),
    ModelNumber VARCHAR(10),
    FOREIGN KEY  (ManufacturerId, ModelNumber) REFERENCES PlaneModels (ManufacturerId,
ModelNumber)
);

-- Table: Prices
IF OBJECT_ID('Prices', 'U') IS NOT NULL DROP TABLE Prices
CREATE TABLE Prices (
    ScheduledFlightId INT REFERENCES ScheduledFlights(ScheduledFlightId),
    ClassId INT REFERENCES Classes(ClassId),
    Price DECIMAL(10,2)  NOT NULL,
    PRIMARY KEY(ScheduledFlightId,ClassId)
);

-- Table: Trips
IF OBJECT_ID('Trips', 'U') IS NOT NULL DROP TABLE Trips
CREATE TABLE Trips (
    TripId INT PRIMARY KEY IDENTITY(1, 1),
    DateBooked DATETIME  NOT NULL,
    CustomerId VARCHAR(10)  NOT NULL REFERENCES Customers(CustomerId),
    CheckInTime DATETIME,
);

-- Table: Tickets
IF OBJECT_ID('Tickets', 'U') IS NOT NULL DROP TABLE Tickets
```

```sql
CREATE TABLE Tickets (
    TicketId VARCHAR(11) PRIMARY KEY,
    ScheduledFlightId INT  NOT NULL REFERENCES ScheduledFlights(ScheduledFlightId),
    CheckInTime DATETIME  NOT NULL,
);

-- Table: Reservations
IF OBJECT_ID('Reservations', 'U') IS NOT NULL DROP TABLE Reservations
CREATE TABLE Reservations (
    TicketId VARCHAR(11) PRIMARY KEY,
    SeatId INT REFERENCES Seats(SeatId),
    TripId INT,
    ReservationDescription TEXT
);

-- Table: Costs
IF OBJECT_ID('Costs', 'U') IS NOT NULL DROP TABLE Costs
CREATE TABLE Costs (
    CostId INT PRIMARY KEY IDENTITY(1, 1),
    FeeType INT NOT NULL REFERENCES FeeTypes (FeeTypeId),
    TicketId VARCHAR(11)  NOT NULL REFERENCES Reservations(TicketId),
    CostDollars DECIMAL(10,2)  NOT NULL,
    CostMiles INT,
);

-- Table: Payments
IF OBJECT_ID('Payments', 'U') IS NOT NULL DROP TABLE Payments
CREATE TABLE Payments (
    CardNumber VARCHAR(20) REFERENCES CreditCardDetails (CardNumber),
    TripId INT REFERENCES Trips(TripId),
    Paytime DATETIME  NOT NULL,
    PaymentDescription TEXT,
    PRIMARY KEY(CardNumber,TripId)
);
-- End of file.
```


Proj2_creatTable.sql - LCS-VC-MSSQL-01.ad.syr.edu.cyu22Database (cyu22 (72)) - Microsoft SQL Server Manageme...
File Edit View Query Project Debug Tools Window Help
New Query
cyu22Database
Execute
Debug
Object Explorer
Connect
Proj2_creatTable.sq...tabase (cyu22 (72))
SQLQuery1.sql - LC...abase (cyu22 (139))*
FeeType INT NOT NULL REFERENCES FeeTypes (FeeTypeId),
TicketId VARCHAR(11) NOT NULL REFERENCES Reservations(TicketId),
CostDollars DECIMAL(10,2) NOT NULL,
CostMiles INT,
);
-- Table: Payments
IF OBJECT_ID('Payments', 'U') IS NOT NULL DROP TABLE Payments
CREATE TABLE Payments (
CardNumber VARCHAR(20) REFERENCES CreditCardDetails (CardNumber),
TripId INT REFERENCES Trips(TripId),
Paytime DATETIME NOT NULL,
PaymentDescription TEXT,
PRIMARY KEY(CardNumber,TripId)
);
-- End of file.
100 %
Messages
Command(s) completed successfully.
Query executed successfully.
LCS-VC-MSSQL-01.ad.syr.edu ...
cyu22 (72)
cyu22Database
00:00:00
0 rows
Ready
Ln 362
Col 1
Ch 1
INS

- cyu22DataBase
  - Database Diagrams
  - Tables
    - System Tables
    - FileTables
    - dbo.Addresses
    - dbo.AirplaneEntertainments
    - dbo.Airplanes
    - dbo.AirportFees
    - dbo.AirportLocations
    - dbo.Airports
    - dbo.Availabilities
    - dbo.CanLand
    - dbo.CanWork
    - dbo.Classes
    - dbo.Costs
    - dbo.CourseEnrollment
    - dbo.CourseGrade
    - dbo.Courses
    - dbo.CreditCardDetails
    - dbo.CreditCardTypes
    - dbo.Crew
    - dbo.Customers
    - dbo.Employees
    - dbo.EntertainmentOptions
    - dbo.FeeTypes
    - dbo.FlightRefreshments
    - dbo.FlightRoutes
    - dbo.Gates
    - dbo.Jobs
    - dbo.LetterGrades
    - dbo.MaintenanceRecords
    - dbo.Manufacturers
    - dbo.Payments
    - dbo.Personals

- dbo.MaintenanceRecords
- dbo.Manufacturers
- dbo.Payments
- dbo.Personals
- dbo.PlaneModels
- dbo.Prices
- dbo.PropulsionMethods
- dbo.RefreshmentOptions
- dbo.Reservations
- dbo.ScheduledFlightGates
- dbo.ScheduledFlights
- dbo.ScheduledFlightTimes
- dbo.Seats
- dbo.Services
- dbo.Terminals
- dbo.TestForIndex
- dbo.Tickets
- dbo.Trips
- dbo.Users

2. Data load

```
/*
 * CSE.581.M004 SPRING 15.Intro D/Base Mngmt Syst.
 * Created by Chunli Yu [cyu22@syr.edu]
 * Last modification date: 2015-04-18 02:52:59.963
 * Data load – load test data into tables.
 * Most of the tables should have between 5 – 10 records, with an average of ~6 records
per table
 */

 --load test data into table Personals
INSERT INTO Personals (PersonId, FirstName,MiddleName,LastName,PhoneNumber)
              VALUES        ('100126789', 'Sam',NULL,'Slade','315-513-7715'),
                            ('200134789', 'George','Angus','Jungle','205-432-4356'),
                            ('300126789', 'Sue',NULL,'Smith','205-232-1400'),
                            ('400196189', 'Mike','Michael','Jones','205-542-6500'),
                            ('500125729', 'Sara',NULL,'Smart','205-612-9023'),
                            ('600126489', 'Roy','Van','Rojers','334-554-3022'),
                            ('700123281', 'Zee','Hantin','Top','256-476-6545'),
                            ('800126688', 'Gene',NULL,'Julia','315-373-2012'),
                            ('900129764', 'Chris',NULL,'Niswandee','541-754-3010'),
                            ('100126661', 'Jane',NULL,'Roe','167-598-4412'),
                            ('110132348', 'John',NULL,'Smith','112-569-8775'),
                            ('120156093', 'Lee','Harvey','Osword','205-625-1234');

--load test data into table Addresses
INSERT INTO Addresses (PersonId,Street1,Street2,City,USState,Zipcode)
```

```
              VALUES
('100126789', '1600 Pennsylvania Ave','1 Main St','Washington','DC','20500'),
('200134789', 'The Alamo','2 Front St','San Antonio','TX','78210'),
('300126789', '1 Microsoft Way',NULL,'Redmond','WA','98052'),
('400196189', '821 Zimbabwe Ave',NULL,'Washington','DC','20500'),
('500125729', '84 Bigboned Way',NULL,'Washington','DC','25732'),
('600126489', '1376 Airline Drive','303 Harbor Dr','North Pole','AK','99705'),
('700123281', '2225 E 5th Ave',NULL,'Anchorage','AK','99501'),
('800126688', '2021 Copper River Hwy','703 Hamilton Dr','Cordova','AK','99574'),
('900129764', '200 E MAIN ST',NULL,'PHOENIX','AZ','85123'),
('100126661', '300 BOYLSTON AVE E',NULL,'SEATTLE','WA','98102'),
('110132348', '795 E DRAGRAM','SMALLSYS INC','TUCSON','AZ','85705'),
('120156093', '123 Stanford Ave',NULL,'Stanford','CA','94305');

--load test data into table Jobs
INSERT INTO Jobs (JobTitle,JobDescription)
              VALUES
('Aviation Safety Inspector', 'These government officials have the authority to enforce
CFRs and FARs in addition to inspecting aircraft operator operations, manuals, and
training records.'),
('Pilot in Command', 'responsible for the operation and safety of an aircraft during
flight time.'),
('Flight Attendant', 'The main responsibility of a flight attendant is to make sure
passengers are safe. Next, the must provide great customer service.'),
('Avionics Technicians', 'Avionics technicians specialize in working on the electronics
systems of aircraft. Avionics technician jobs involve troubleshooting, repairing,
replacing, and installing avionic equipment. Calibration of the equipment may also be
required.'),
('Aviation Meteorologist', 'Aviation meteorologists provide weather information to
airline flight dispatchers and pilots. They must determine current and forecasted weather
conditions for all altitudes, including the direction and speed of wind, cloud cover, and
precipitation.'),
('Flight Deck Officer', 'Airline crew member who has successfully completed training with
the federal government and has been authorized by TSA to carry a weapon on board an
aircraft with certain restrictions.'),
('Security Inspector', 'Member of TSA whose responsibility is to ensure aircraft
operators adhere to the directives set forth in the AOSSP.'),
('Airline Flight Instructor', 'An airline flight instructor provides recurrent training
for the pilots. Airline flight instructors may be senior pilots who fly for the
airline.');

--load test data into table Customers
INSERT INTO Customers (CustomerId,FrequentFlyerNumber)
              VALUES        ('100126789', 'LH1236699999'),
                            ('200134789', 'AC1543786238'),
                            ('300126789', NULL),
                            ('400196189', NULL),
                            ('500125729', NULL);

--load test data into table CreditCardTypes
INSERT INTO CreditCardTypes (CardTypeName)
              VALUES        ('Discover'),
                            ('MasterCard'),
                            ('Chase Freedom'),
                            ('UnionPay'),
                            ('Bank of America'),
                            ('PLUS'),
                            ('Ebay'),
```

```
                            ('American Express'),
                            ('Paypal');

--load test data into table CreditCardDetails
INSERT INTO CreditCardDetails (CardNumber,Holder,CardType,ExpirationMonth,ExpirationYear)
             VALUES         ('5529420350615465', '100126789',2,12,14),
                            ('2937816292739723', '200134789',7,02,18),
                            ('1234567898765432', '300126789',1,12,19),
                            ('4000123456789123', '400196189',5,11,21),
                            ('5529420322806242', '500125729',7,06,16),
                            ('5000001234567899', '100126789',3,05,17),
                            ('5412235678901234', '100126789',4,09,22);

--load test data into table Employees
INSERT INTO Employees (EmployeeId,JobId,DirectSupervisor,SSN,Salary)
             VALUES         ('600126489', 1,'110132348','433-54-3937',3500.99),
                            ('700123281', 2,'100126661','451-43-0201',4567.00),
                            ('800126688', 3,'100126661','468-47-2983',3456.00),
                            ('900129764', 4,'110132348','501-50-6728',4999.00),
                            ('100126661', 8,'110132348','518-51-2341',5100.00),
                            ('110132348', 7,'110132348','526-52-3932',4300.00),
                            ('120156093', 3,'100126661','400-40-7638',7500.00);
                            ('110132348', 7,'110132348','526-52-3932',4300.00),
                            ('120156093', 3,'100126661','400-40-7638',7500.00);

--load test data into table PropulsionMethods
INSERT INTO PropulsionMethods (PropulsionMethod)
             VALUES         ('jet'),
                            ('propeller'),
                            ('hypersonic');

--load test data into table Manufacturers
INSERT INTO Manufacturers (ManufacturerId,ManufacturerName)
             VALUES         ('00H00H74H', 'Aachen Flugzeugbau'),
                            ('00H00H75H', 'Abrams Air Craft Corporation'),
                            ('00H00H78H', 'ABS Aircraft AG'),
                            ('00H01H1EH', 'Ace Aircraft Manufacturing and Supply'),
                            ('00H01H12H', 'Aces High Light Aircraft Ltd'),
                            ('00H01H22H', 'Allied Aviation'),
                            ('00H01H2FH', 'AviPro Aircraft Ltd.'),
                            ('00H01H36H', 'Avid Aircraft Inc.'),
                            ('00H01H39H', 'Aviation Scotland Ltd.');

--load test data into table PlaneModels
INSERT INTO PlaneModels (ManufacturerId,
ModelNumber,PropulsionMethod,NumberOfPilots,NumberOfAttendants,FlyRange)
             VALUES         ('00H00H74H', '737',1,2,7,4146746),
                            ('00H00H75H', '717',2,4,10,4021214),
                            ('00H00H78H', '757',3,3,8,2016878),
                            ('00H01H1EH', '777X',1,1,7,180836),
                            ('00H01H12H', '717',2,2,10,127396),
                            ('00H01H22H', '377',3,3,15,11651011),
                            ('00H01H2FH', '747',1,2,9,115204),
                            ('00H01H36H', '777',2,2,8,101485),
                            ('00H01H39H', '787',3,3,12,802944);

--load test data into table EntertainmentOptions
INSERT INTO EntertainmentOptions (EntertainmentName)
```

```
              VALUES        ('wifi'),
                            ('tv'),
                            ('movies'),
                            ('magazine'),
                            ('radio');

--load test data into table AirplaneEntertainments
INSERT INTO AirplaneEntertainments (ManufacturerId,ModelNumber,EntertainmentOptionId)
              VALUES        ('00H00H74H', '737',1),
                            ('00H00H74H', '737',3),
                            ('00H00H74H', '737',4),
                            ('00H00H75H', '717',2),
                            ('00H00H75H', '717',3),
                            ('00H00H78H', '757',3),
                            ('00H00H78H', '757',4),
                            ('00H00H78H', '757',5),
                            ('00H01H1EH', '777X',1),
                            ('00H01H12H', '717',2),
                            ('00H01H22H', '377',3),
                            ('00H01H2FH', '747',4),
                            ('00H01H36H', '777',3),
                            ('00H01H39H', '787',5);

--load test data into table CanWork
INSERT INTO CanWork (EmployeeId,ManufacturerId,ModelNumber)
              VALUES        ('600126489', '00H00H74H', '737'),
                            ('600126489', '00H01H12H', '717'),
                            ('700123281', '00H01H39H', '787'),
                            ('800126688', '00H00H74H', '737'),
                            ('900129764', '00H01H1EH', '777X'),
                            ('100126661', '00H01H22H', '377'),
                            ('110132348', '00H00H74H', '737'),
                            ('120156093', '00H01H39H', '787'),
                            ('120156093', '00H00H78H', '757'),
                            ('120156093', '00H01H2FH', '747');

--load test data into table Airports
INSERT INTO Airports (AirportId,AirportName,HangarCapacity)
              VALUES        ('LGA', 'LaGuardia Airport',300),
                            ('JFK', 'John F. Kennedy International Airport', 400),
                            ('IAD', 'Washington Dulles International Airport',200),
                            ('DCA', 'Reagan National Airport 6',150),
                            ('BUU', 'BURLINGTON MUNI',200),
                            ('JVL', 'SOUTHERN WISCONSIN RGNL',150),
                            ('C29', 'MIDDLETON MUNI - MOREY FIELD',400),
                            ('MKE', 'GENERAL MITCHELL INTL',350),
                            ('C35', 'REEDSBURG MUNI',210);

--load test data into table AirportLocations
INSERT INTO AirportLocations (AirportId,City,USState)
              VALUES        ('LGA', 'New York City', 'NY'),
                            ('JFK', 'New York City', 'NY'),
                            ('IAD', 'Washington','DC'),
                            ('DCA', 'Washington','DC'),
                            ('BUU', 'Burlington', 'WI'),
                            ('JVL', 'Janesville', 'WI'),
                            ('C29', 'Middleton', 'WI'),
                            ('MKE', 'Milwaukee', 'WI'),
```

```
                            ('C35', 'Reedsburg', 'WI');

--load test data into table FeeTypes
INSERT INTO FeeTypes (FeeTypeName)
              VALUES        ('Airport Construction Fee'),
                            ('Airport Maintenance Fee'),
                            ('City Tax'),
                            ('State Tax'),
                            ('Ticket Price'),
                            ('Luggage Charge'),
                            ('Refund'),
                            ('Others');

--load test data into table AirportFees
INSERT INTO AirportFees (AirportId,FeeTypeId,Fees)
              VALUES        ('LGA', 1,50),
                            ('LGA', 2,20),
                            ('LGA', 3,0.08),
                            ('LGA', 4,0.13),
                            ('JFK', 1,60),
                            ('JFK', 2,43),
                            ('JFK', 3,0.08),
                            ('JFK', 4,0.10),
                            ('IAD', 1,80),
                            ('IAD', 2,45),
                            ('IAD', 3,0.10),
                            ('IAD', 4,0.05),
                            ('DCA', 1,90),
                            ('DCA', 2,50),
                            ('DCA', 3,0.15),
                            ('BUU', 1,100),
                            ('BUU', 2,60),
                            ('BUU', 3,0.13),
                            ('JVL', 1,75),
                            ('JVL', 2,21),
                            ('C29', 1,60),
                            ('C29', 2,15),
                            ('MKE', 1,40),
                            ('C35', 1,45);

--load test data into table CanLand
INSERT INTO CanLand (AirportId,ManufacturerId,ModelNumber)
              VALUES        ('LGA', '00H00H74H', '737'),
                            ('LGA', '00H00H75H', '717'),
                            ('LGA', '00H01H1EH', '777X'),
                            ('JFK', '00H01H1EH', '777X'),
                            ('JFK', '00H00H75H', '717'),
                            ('JFK', '00H00H78H', '757'),
                            ('JFK', '00H01H2FH', '747'),
                            ('IAD', '00H01H1EH', '777X'),
                            ('DCA', '00H00H75H', '717'),
                            ('BUU', '00H01H39H', '787'),
                            ('JVL', '00H01H22H', '377'),
                            ('C29', '00H00H78H', '757'),
                            ('MKE', '00H01H39H', '787'),
                            ('C35', '00H01H39H', '787');

--load test data into table Availabilities
```

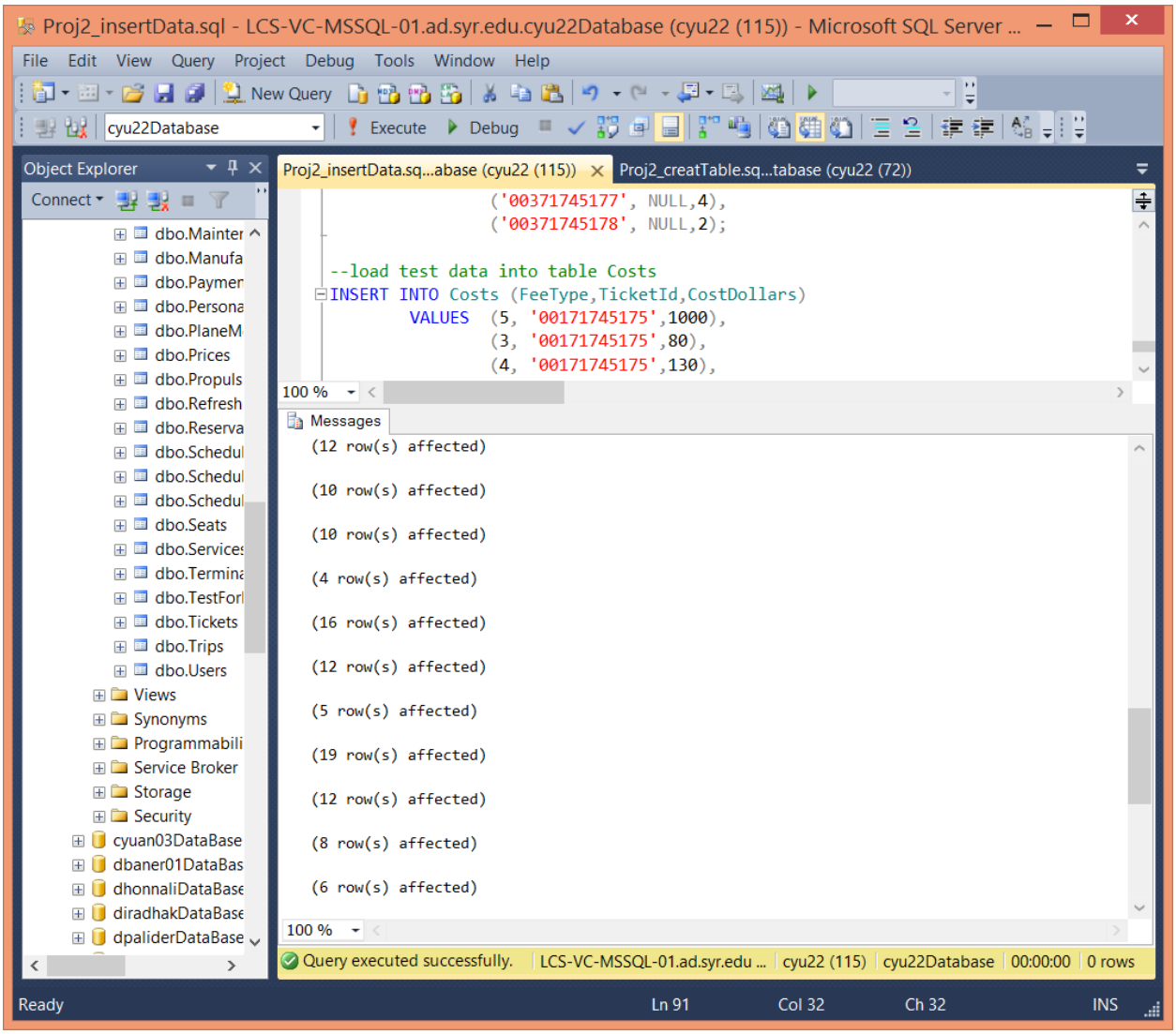
```
INSERT INTO Availabilities (AvailabilityStatus)
              VALUES        ('available'),
                            ('retired'),
                            ('requires maintenance');

--load test data into table Airplanes
INSERT INTO Airplanes
(PlaneId,ManufactureId,ModelNumber,ManufactureDate,AvailabilityId,CurrentLocation)
              VALUES        ('N323', '00H00H74H', '737','2012-06-18',1,'JFK'),
                            ('N389', '00H00H74H', '737','2005-03-01',1,'LGA'),
                            ('N234', '00H01H12H', '717','2001-02-11',1,'JFK'),
                            ('N221', '00H01H22H', '377','1991-09-17',2,'IAD'),
                            ('N100', '00H01H39H', '787','1987-08-15',3,'JVL'),
                            ('N194', '00H01H36H', '777','1997-07-11',1,'JFK'),
                            ('N504', '00H01H36H', '777','2008-07-13',1,'JFK'),
                            ('N519', '00H01H1EH', '777X','2013-01-14',1,'C35'),
                            ('N666', '00H00H78H', '757','2001-02-19',1,'BUU'),
                            ('N777', '00H01H2FH', '747','2002-04-21',1,'JFK');

--load test data into table Services
INSERT INTO Services (ServiceName,ServiceDescription)
              VALUES
('Turbine engine compressor and hot section repair', 'equipment is maintained before
break down occurs'),
('Honeycomb and composite structure repair and alteration', NULL),
('Main and tail rotor blade repairs', NULL),
('Avionics maintenance', NULL),
('Sheet metal fabrication', NULL),
('Scheduled inspections and flight testing of aircraft', NULL);

--load test data into table MaintenanceRecords
INSERT INTO MaintenanceRecords
(PlaneId,ExpectedDate,MantenancedDate,WorkBeingDone,MantenanceMan,MaintenanceDescription)
              VALUES
('N323', '2012-01-01', '2012-01-10',1,'600126489',NULL),
('N389', '2013-03-02', '2013-03-02',2,'900129764',NULL),
('N234', '2014-03-04', '2014-07-04',3,'600126489',NULL),
('N221', '2015-04-04', '2015-04-01',1,'900129764',NULL),
('N100', '2016-05-05', '2016-05-05',4,'900129764',NULL),
('N323', '2012-06-06', '2012-06-09',5,'900129764','Stress failures due to rapid and
repetitive expansion and contraction with extreme temperature swings'),
('N323', '2013-07-07', '2013-07-23',6,'600126489','An in-depth exhaust system inspection
should be done as part of the aircraft''s annual inspection.');

--load test data into table FlightRoutes
INSERT INTO FlightRoutes (DepartureAirpport,ArrivalAirport,FlightDuration,FlightDistance)
              VALUES        ('LGA', 'IAD', '7:23',3553.3),
                            ('JFK', 'IAD', '2:36',1159.9),
                            ('IAD', 'JFK','2:19',1040.1),
                            ('DCA', 'BUU','20:23',9814.5),
                            ('BUU', 'JVL', '1:30',625.2),
                            ('JVL', 'C29', '8:44',4206.8),
                            ('C29', 'JFK', '8:29',4086.2),
                            ('MKE', 'C35', '5:17',4077.3),
                            ('C35', 'MKE', '4:39',6879.9);

--load test data into table RefreshmentOptions
INSERT INTO RefreshmentOptions (RefreshmentName)
```

```
            VALUES        ('snacks'),
                          ('full meals'),
                          ('drinks');

--load test data into table FlightRefreshments
INSERT INTO FlightRefreshments (FlightRoutId,RefreshmentId,Cost)
            VALUES        (1, 1,8),
                          (1, 3,2),
                          (2, 2,15),
                          (2, 3,2),
                          (3, 1,7),
                          (3, 2,12),
                          (3, 3,3),
                          (4, 1,5),
                          (5, 2,8),
                          (6, 3,1);

--load test data into table ScheduledFlights
INSERT INTO ScheduledFlights (FlightRouteId,FlightNumber,AvailableSeats,PlaneId)
            VALUES        (1,'AA6594', 55,'N323'),
                          (2,'GF5222', 75,'N323'),
                          (3,'LH6639', 90,'N389'),
                          (4,'CX7121', 180,'N234'),
                          (5,'BD193', 150,'N221'),
                          (6,'FG910', 75,'N100'),
                          (7,'JQ35', 70,'N100'),
                          (8,'AA7363', 120,'N504'),
                          (9,'NZ718', 100,'N519'),
                          (5,'BA872', 150,'N666');

--load test data into table Crew
INSERT INTO Crew (EmployeeId,ScheduledFlightId)
            VALUES        ('700123281', 1),
                          ('800126688', 1),
                          ('120156093', 1),
                          ('700123281', 2),
                          ('800126688', 2),
                          ('700123281', 3),
                          ('120156093', 3);

--load test data into table Terminals
INSERT INTO Terminals (TerminalName, AirportId,TerminalDescription)
            VALUES        ('Terminal 1', 'LGA','On the 1st floor'),
                          ('Terminal 2', 'LGA','On the 1st floor'),
                          ('Terminal 3', 'LGA','On the 2nd floor'),
                          ('Terminal 4', 'LGA','On the 2nd floor'),
                          ('Terminal 1', 'JFK', NULL),
                          ('Terminal 2', 'JFK', NULL),
                          ('Terminal 3', 'JFK', NULL),
                          ('Terminal 4', 'JFK', NULL),
                          ('Terminal 1', 'DCA', NULL),
                          ('Terminal 2', 'DCA', NULL),
                          ('Terminal 2', 'IAD', NULL);

--load test data into table Gates
INSERT INTO Gates (GateName,TerminalId)
            VALUES  ('A1', 1),
                    ('A2', 1),
```

```
                        ('B1', 1),
                        ('B2', 1),
                        ('C1', 1),
                        ('C2', 1),
                        ('A1', 2),
                        ('A2', 2),
                        ('B1', 2),
                        ('B2', 2),
                        ('C1', 3),
                        ('C2', 3);

--load test data into table ScheduledFlightGates
INSERT INTO ScheduledFlightGates
(ScheduledFlightId,ScheduledDepartureGate,ScheduledArrivalGate,ActuralDepartureGate,Actur
alArrivalGate)
              VALUES        (1, 1,7,1,8),
                            (2, 3,8,3,8),
                            (3, 1,9,5,8),
                            (4, 9,11,9,2),
                            (5, 2,7,4,9),
                            (6, 5,7,5,8),
                            (7, 4,11,4,12),
                            (8, 1,10,1,9),
                            (9, 1,11,NULL,NULL),
                            (10, 2,12,NULL,NULL);

--load test data into table ScheduledFlightTimes
INSERT INTO ScheduledFlightTimes (ScheduledFlightId,
                                   ScheduledDepartureTime, ScheduledArrivalTime,
                                   ProjectedDepartureTime, ProjectedArrivalTime,
                                  ActuralDepartureTime,ActuralArrivalTime)
              VALUES
(1, '2015-06-01 01:24:00','2015-06-01 08:47:09','2015-06-01 01:34:00','2015-06-01
08:57:09','2015-06-01 01:38:00','2015-06-01 09:06:09'),
(2, '2015-05-02 02:34:02','2015-05-02 05:10:09','2015-05-02 02:44:00','2015-05-02
05:15:09','2015-05-02 02:50:02','2015-05-02 05:17:09'),
(3, '2015-07-03 03:54:03','2015-07-03 06:13:09','2015-07-03 03:54:00','2015-07-03
06:13:09','2015-07-03 03:54:03','2015-07-03 06:13:09'),
(4, '2015-06-04 04:11:04','2015-06-05 00:34:09','2015-06-04 04:19:00','2015-06-05
00:42:09','2015-06-04 04:20:04','2015-06-05 00:34:09'),
(5, '2015-08-05 05:12:05','2015-08-05 06:42:09','2015-08-05 05:17:00','2015-08-05
06:47:09','2015-08-05 05:20:05','2015-08-05 06:55:09'),
(6, '2015-09-07 06:13:06','2015-09-07 14:57:09','2015-09-07 06:16:00','2015-09-07
15:07:09','2015-09-07 06:25:06','2015-09-07 15:35:09'),
(7, '2015-10-08 07:14:07','2015-10-08 15:43:09','2015-10-08 07:19:07','2015-10-08
15:49:09','2015-10-08 07:19:07','2015-10-08 15:49:09'),
(8, '2015-11-09 08:15:08','2015-11-09 13:32:09','2015-11-09 08:15:08','2015-11-09
13:32:09','2015-11-09 08:23:08','2015-11-09 13:37:09'),
(9, '2015-12-10 09:16:09','2015-12-10 13:55:09',NULL,NULL,NULL,NULL),
(10, '2015-06-11 10:17:10','2015-06-11 11:47:09','2015-06-11 10:17:10','2015-06-11
11:47:09',NULL,NULL);

--load test data into table Classes
INSERT INTO Classes (ClassName)
              VALUES        ('first class'),
                            ('business class'),
                            ('economy plus'),
                            ('economy');
```

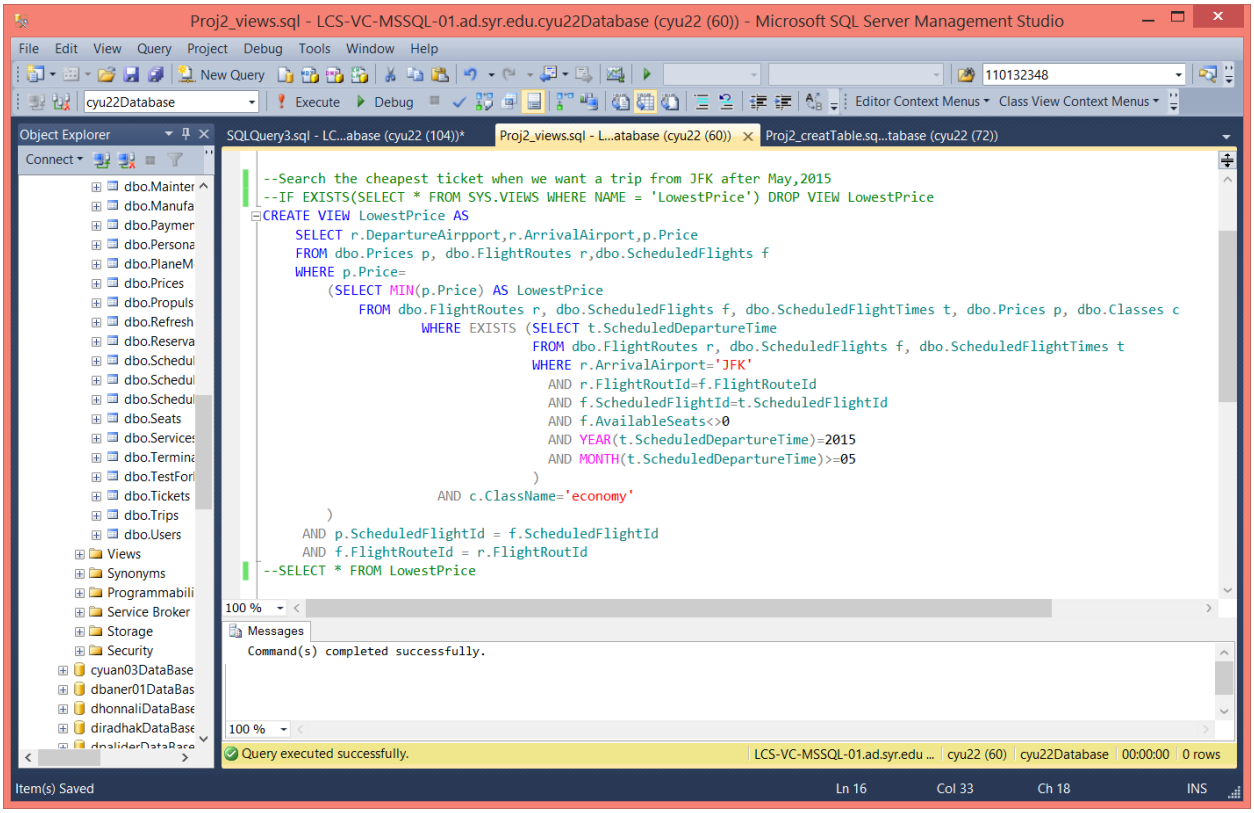
```sql
--load test data into table Seats
INSERT INTO Seats (ClassId,RowNumber,ColumnLetter,ManufacturerId,ModelNumber)
              VALUES        (1, 1,'A','00H00H74H', '737'),
                            (1, 1,'B','00H00H74H', '737'),
                            (2, 1,'A','00H00H74H', '737'),
                            (2, 1,'B','00H00H74H', '737'),
                            (3, 1,'C','00H00H74H', '737'),
                            (3, 1,'D','00H00H74H', '737'),
                            (4, 2,'A','00H00H74H', '737'),
                            (4, 2,'B','00H00H74H', '737'),
                            (1, 1,'A','00H01H1EH', '777X'),
                            (1, 1,'B','00H01H1EH', '777X'),
                            (2, 1,'A','00H01H1EH', '777X'),
                            (2, 1,'B','00H01H1EH', '777X'),
                            (3, 1,'C','00H01H1EH', '777X'),
                            (3, 1,'D','00H01H1EH', '777X'),
                            (4, 2,'A','00H01H1EH', '777X'),
                            (4, 2,'B','00H01H1EH', '777X');

--load test data into table Prices
INSERT INTO Prices (ScheduledFlightId, ClassId,Price)
              VALUES        (1, 1,1000.00),
                            (1, 2,800.00),
                            (1, 3,650.00),
                            (1, 4,550.00),
                            (2, 1,580.00),
                            (2, 2,440.00),
                            (2, 3,350.00),
                            (2, 4,300.00),
                            (3, 1,900.00),
                            (3, 2,800.00),
                            (3, 3,700.00),
                            (3, 4,600.00);

--load test data into table Trips
INSERT INTO Trips (DateBooked,CustomerId,CheckInTime)
              VALUES        ('2015-04-01 00:44:00', '100126789','2015-06-01 00:44:00'),
                            ('2015-05-01 00:44:00', '100126789','2015-05-02 01:34:02'),
                            ('2015-05-25 00:44:00', '200134789','2015-06-01 00:44:00'),
                            ('2015-04-27 00:44:00', '300126789','2015-05-02 01:34:02'),
                            ('2015-05-03 00:44:00', '400196189','2015-05-02 01:34:02');

--load test data into table Tickets
INSERT INTO Tickets (TicketId,ScheduledFlightId,CheckInTime)
              VALUES        ('00171745175', 1,'2015-06-01 00:44:00'),
                            ('00171745176', 1,'2015-06-01 00:44:00'),
                            ('00171745177', 1,'2015-06-01 00:44:00'),
                            ('00171745178', 1,'2015-06-01 00:44:00'),
                            ('00271745175', 2,'2015-05-02 01:34:02'),
                            ('00271745176', 2,'2015-05-02 01:34:02'),
                            ('00271745177', 2,'2015-05-02 01:34:02'),
                            ('00271745178', 2,'2015-05-02 01:34:02'),
                            ('00371745175', 3,'2015-07-03 03:00:03'),
                            ('00371745176', 3,'2015-07-03 03:00:03'),
                            ('00371745177', 3,'2015-07-03 03:00:03'),
                            ('00371745178', 3,'2015-07-03 03:00:03'),
```

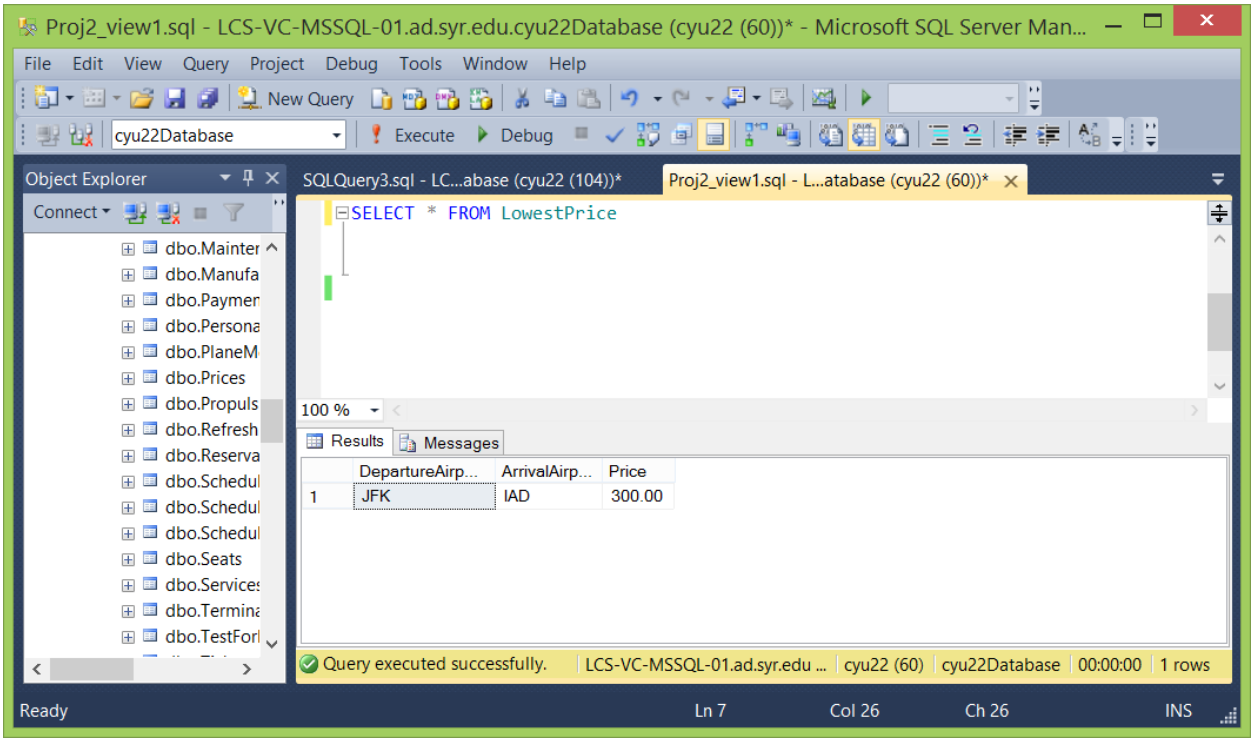
```
                            ('00571745171', 1,'2015-06-01 00:44:00'),
                            ('00571745172', 1,'2015-06-01 00:44:00'),
                            ('00571745173', 3,'2015-07-03 03:00:03'),
                            ('00571745174', 3,'2015-07-03 03:00:03'),
                            ('00571745175', 3,'2015-07-03 03:00:03'),
                            ('00571745176', 3,'2015-07-03 03:00:03'),
                            ('00571745177', 3,'2015-07-03 03:00:03');

--load test data into table Reservations
INSERT INTO Reservations (TicketId,SeatId,TripId)
             VALUES         ('00171745175', 1, 1),
                            ('00171745176', 2,2),
                            ('00171745177', 3,3),
                            ('00171745178', 4,4),
                            ('00271745175', 5,5),
                            ('00271745176', NULL,1),
                            ('00271745177', 7,2),
                            ('00271745178', 8,4),
                            ('00371745175', 1,1),
                            ('00371745176', 2,3),
                            ('00371745177', NULL,4),
                            ('00371745178', NULL,2);

--load test data into table Costs
INSERT INTO Costs (FeeType,TicketId,CostDollars)
             VALUES         (5, '00171745175',1000),
                            (3, '00171745175',80),
                            (4, '00171745175',130),
                            (6, '00171745175',300),
                            (5, '00171745176',1000),
                            (5, '00171745177',800),
                            (3, '00171745177',64),
                            (4, '00171745177',104);

--load test data into table Payments
INSERT INTO Payments (CardNumber,TripId,Paytime,PaymentDescription)
             VALUES
('5529420350615465', 1,'2015-04-01 00:55:00','Bank Transfer'),
('5412235678901234', 1,'2015-04-01 00:57:00','Receipt Requred'),
('5412235678901234', 2,'2015-05-01 00:44:00','Request a security token'),
('1234567898765432', 3,'2015-05-25 00:44:00',NULL),
('4000123456789123', 4,'2015-04-27 00:44:00',NULL),
('5529420322806242', 5,'2015-05-03 00:44:00',NULL);
-- End of file.
```


Proj2_insertData.sql - LCS-VC-MSSQL-01.ad.syr.edu.cyu22Database (cyu22 (115)) - Microsoft SQL Server ...
File Edit View Query Project Debug Tools Window Help
New Query
cyu22Database
Execute
Debug
Object Explorer
Connect
Proj2_insertData.sq...abase (cyu22 (115))
Proj2_creatTable.sq...tabase (cyu22 (72))
('00371745177', NULL,4),
('00371745178', NULL,2);
--load test data into table Costs
INSERT INTO Costs (FeeType,TicketId,CostDollars)
VALUES (5, '00171745175',1000),
(3, '00171745175',80),
(4, '00171745175',130),
100 %
Messages
(12 row(s) affected)
(10 row(s) affected)
(10 row(s) affected)
(4 row(s) affected)
(16 row(s) affected)
(12 row(s) affected)
(5 row(s) affected)
(19 row(s) affected)
(12 row(s) affected)
(8 row(s) affected)
(6 row(s) affected)
100 %
Query executed successfully.
LCS-VC-MSSQL-01.ad.syr.edu ...
cyu22 (115)
cyu22Database
00:00:00
0 rows
Ready
Ln 91
Col 32
Ch 32
INS
dbo.Mainter
dbo.Manufa
dbo.Paymer
dbo.Persona
dbo.PlaneM
dbo.Prices
dbo.Propuls
dbo.Refresh
dbo.Reserva
dbo.Schedul
dbo.Schedul
dbo.Schedul
dbo.Seats
dbo.Services
dbo.Termina
dbo.TestForl
dbo.Tickets
dbo.Trips
dbo.Users
Views
Synonyms
Programmabili
Service Broker
Storage
Security
cyuan03DataBase
dbaner01DataBas
dhonnaliDataBase
diradhakDataBase
dpaliderDataBase

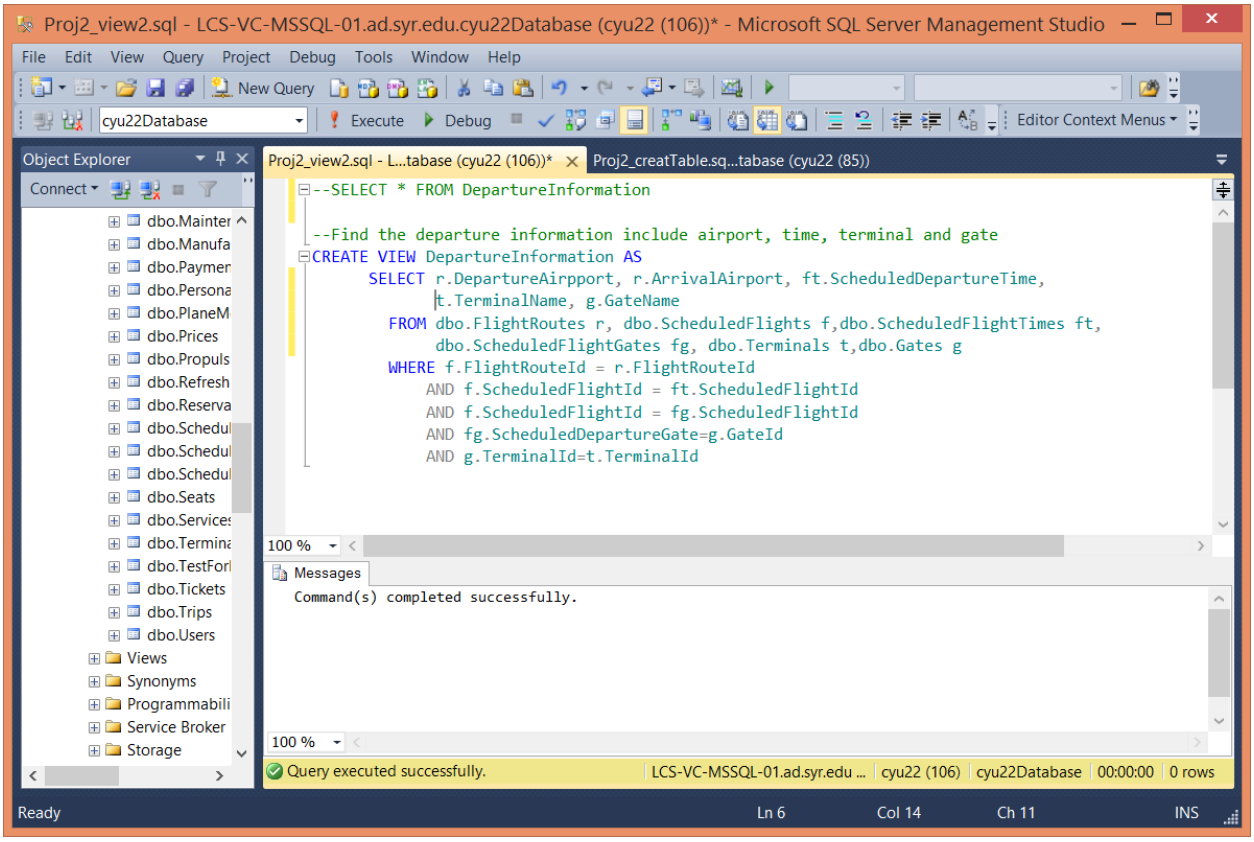
3. Views

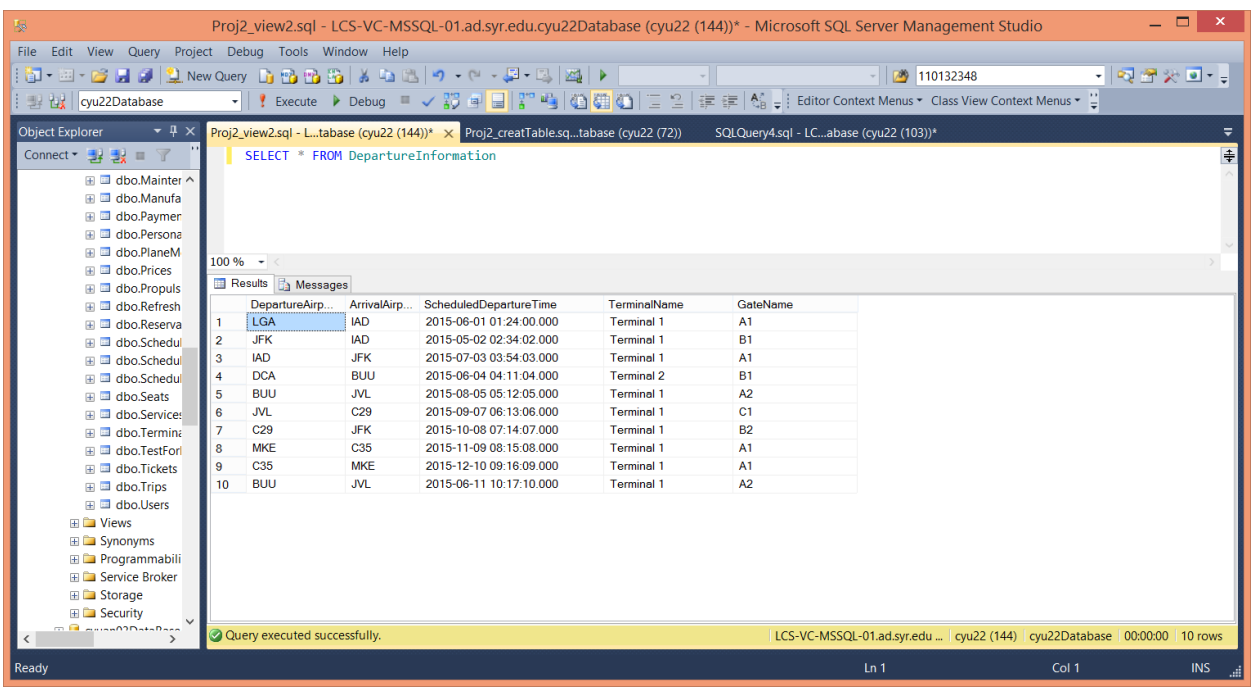
(1) LowestPrice



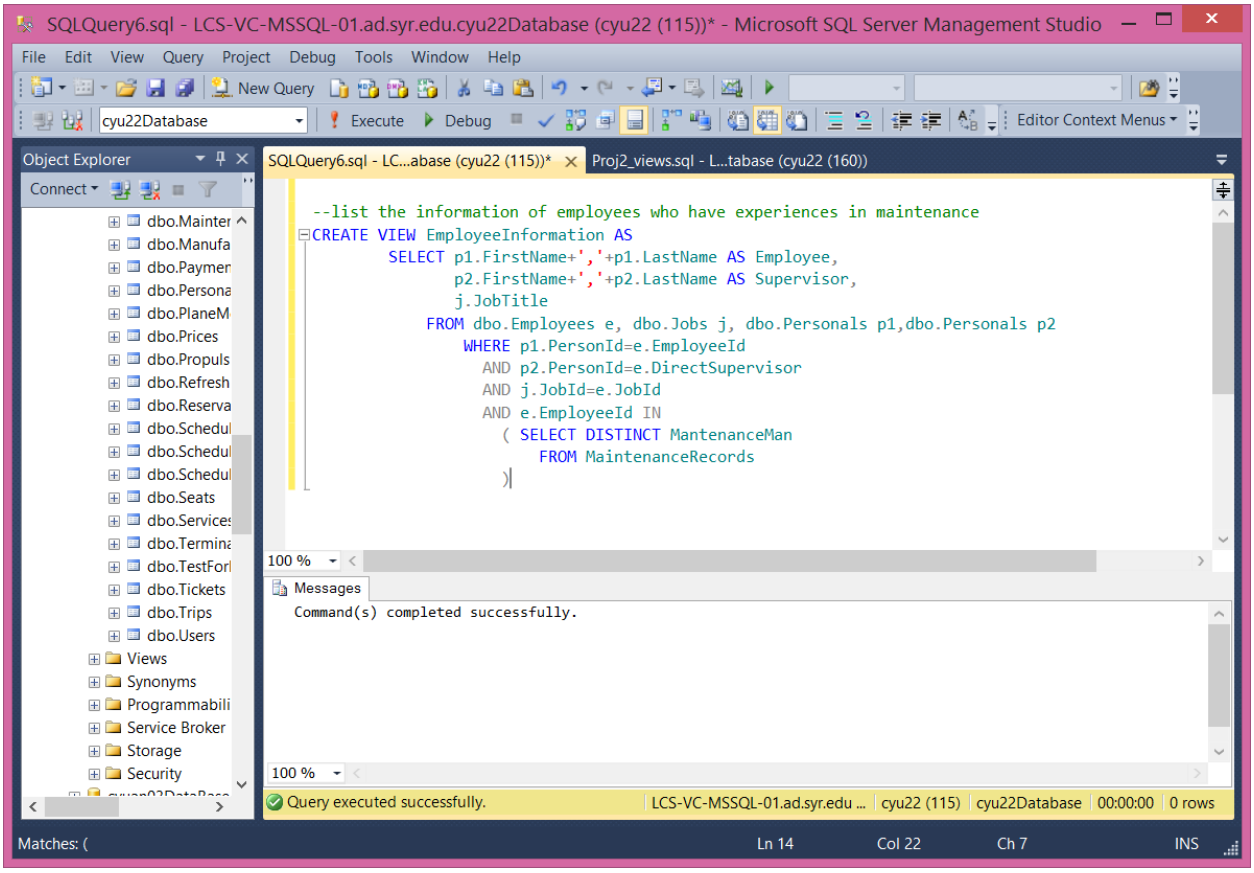


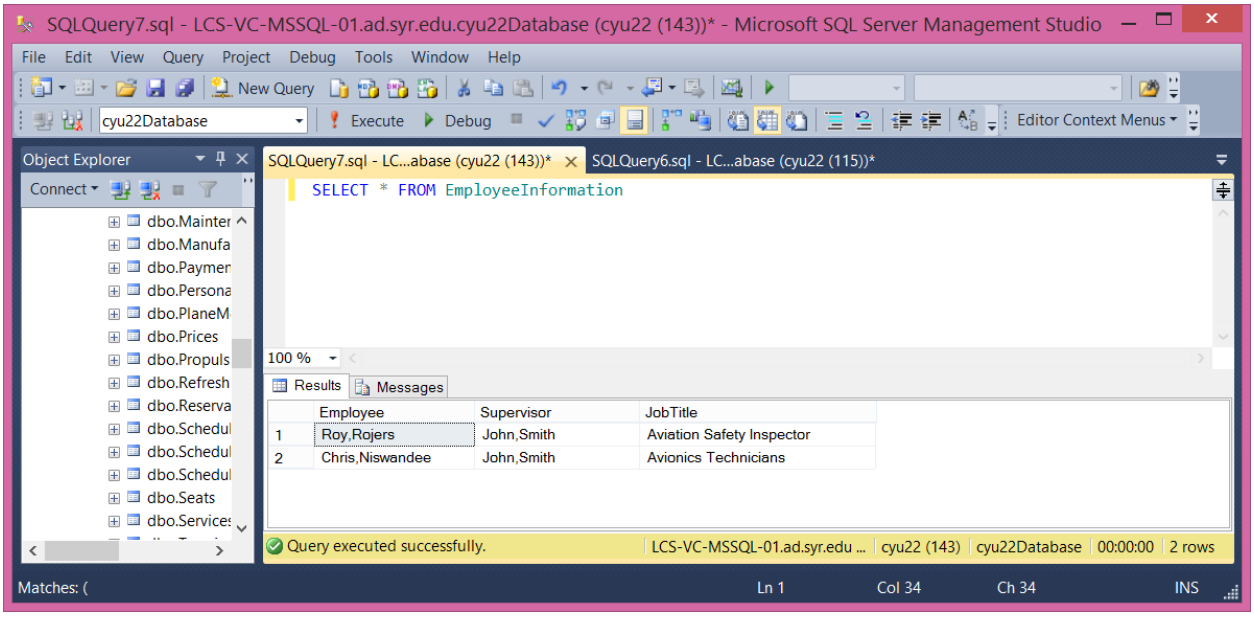
(2) DepartureInformation



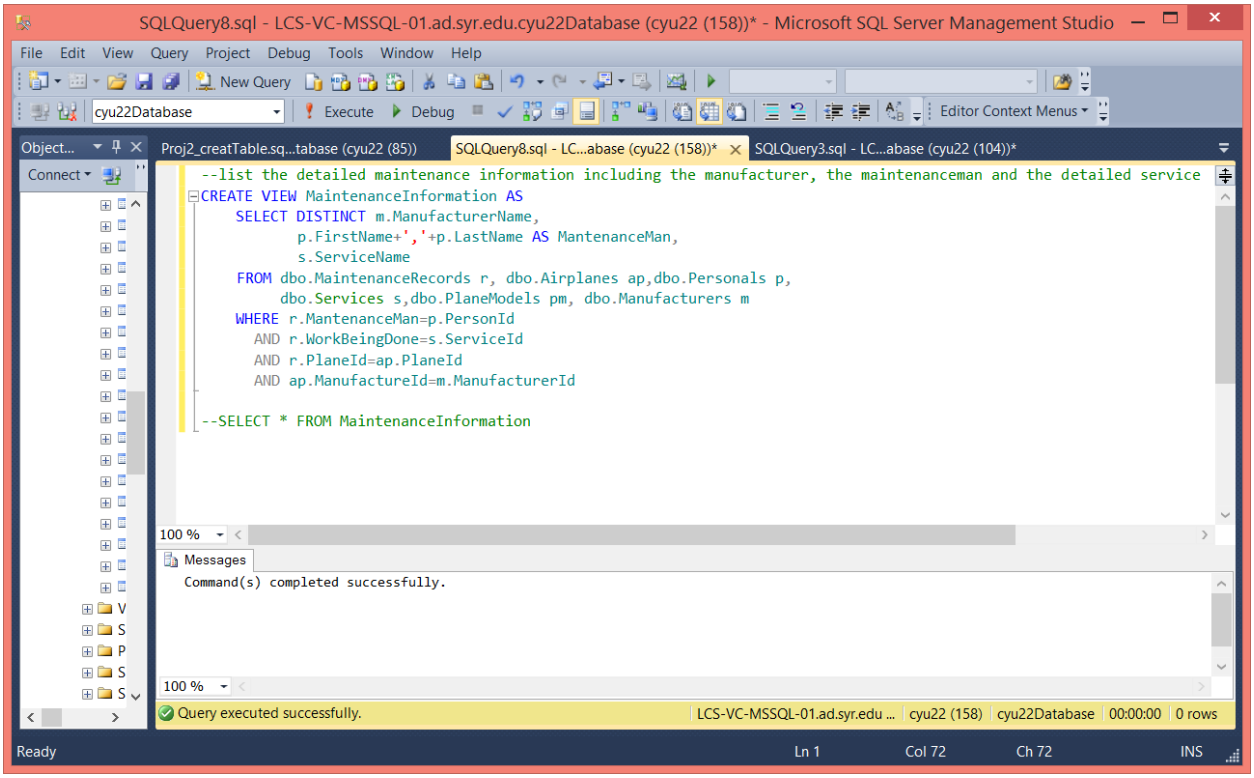


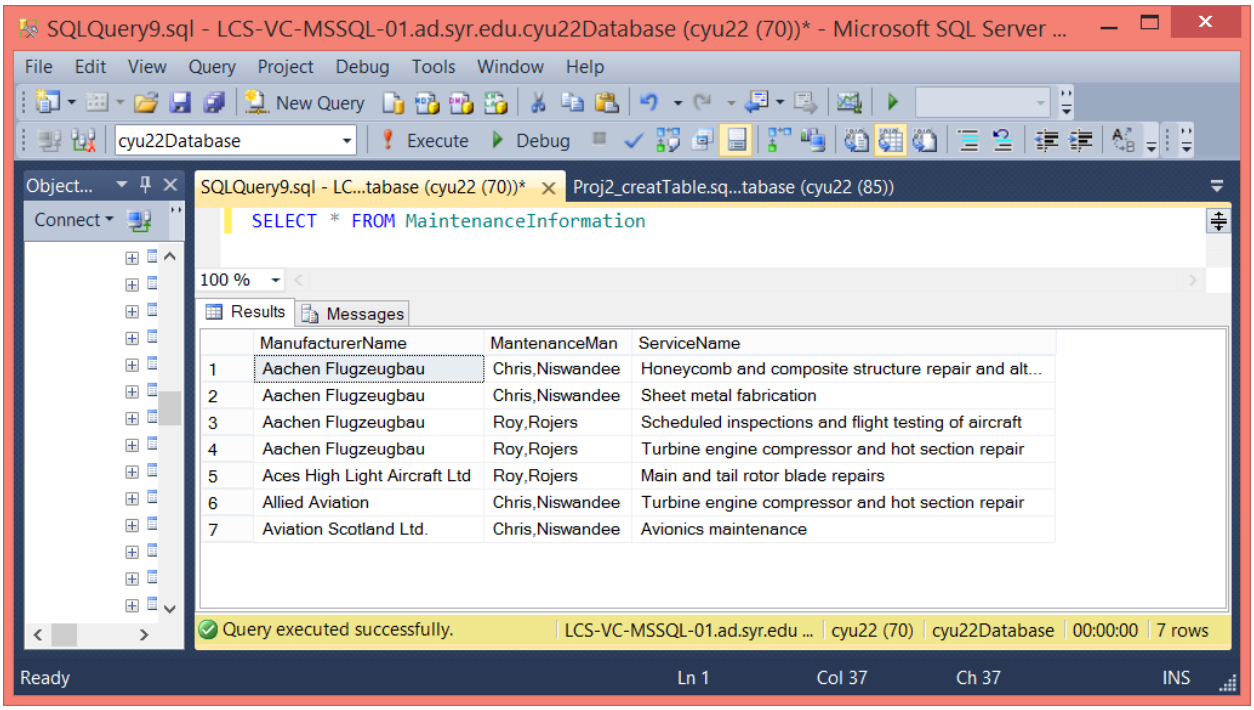
(3) EmployeeInformation





(4) MaintenanceInformation





4. Stored Procedures

(1) SP1: ReserveTicket

```
-- SP1 ReserveTicket:Reserve a ticket of a desired scheduled flight
CREATE PROCEDURE dbo.ReserveTicket(@ScheduledFlightId AS INT,@SetId INT,@TripId AS INT)
AS
	DECLARE @ErrorOccured AS BIT
	SET @ErrorOccured = 0

	BEGIN TRAN
	DECLARE @AvailableSeats AS INT
	DECLARE @TicketReserved AS VARCHAR(11)
	DECLARE @TicketId AS VARCHAR(11)

	SELECT @AvailableSeats = AvailableSeats
		FROM ScheduledFlights
		WHERE ScheduledFlightId = @ScheduledFlightId
	SELECT @TicketReserved = TicketId
		FROM Reservations
		WHERE SeatId = @SeatId
		  AND SeatId IS NOT NULL

	IF @AvailableSeats = 0
		BEGIN
			PRINT 'Sorry, the tickets have sold out.'
			SET @ErrorOccured = 1
		END
	ELSE IF @TicketReserved IS NOT NULL
		BEGIN
			PRINT 'The seat has been reserved!'
			SET @errorOccured = 1
		END
	ELSE
		BEGIN
		    SELECT @TicketId =
		        (SELECT TOP 1 TicketId FROM Tickets
		           WHERE ScheduledFlightId = @ScheduledFlightId
		                AND TicketId NOT IN (
		                    SELECT TicketId FROM Reservations
		                           WHERE ScheduledFlightId = @ScheduledFlightId
		                           )
		)
			INSERT INTO Reservations (TicketId, SetId, TripId)
				VALUES (@TicketId, @SetId, @TripId)
			PRINT 'Congratuation! Ticket Reserved Successfully. '

			BEGIN
				PRINT 'Update Available Seats'
		  UPDATE ScheduledFlights
		                    SET AvailableSeats = @AvailableSeats - 1
		                    WHERE ScheduledFlightId = @ScheduledFlightId
				END
		END

	IF @errorOccured = 1
		BEGIN
			PRINT 'Rollback'
```

```
                        ROLLBACK TRAN
                END
        ELSE
                BEGIN
                        PRINT 'Commit'
                        COMMIT TRAN
                END;
```

(2) RefundTicket

```
-- SP2 RefundTicket:Refund a ticket based on the date
CREATE PROCEDURE dbo.RefundTicket(@TicketId AS INT) AS
        DECLARE @ErrorOccured AS BIT
        SET @ErrorOccured = 0

        BEGIN TRAN

        DECLARE @CheckInTime AS DATETIME
        DECLARE @TicketPaid AS DECIMAL(10,2)
        DECLARE @FeeTypeId AS INT
        DECLARE @TicketRefund AS DECIMAL(10,2)
        DECLARE @TicketRefundRatio AS DECIMAL(10,2)

        SELECT @CheckInTime = CheckInTime
                FROM Tickets
                WHERE TicketId = @TicketId
        SELECT @TicketPaid = SUM(Costs.CostDollars)
                FROM Costs
                WHERE TicketId = @TicketId
        SELECT @FeeTypeId = FeeTypeId
                FROM FeeTypes
                WHERE FeeTypeName = 'Refund'

        IF @CheckInTime < GETEDATE()
                BEGIN
                        PRINT 'Sorry, You cannot be refunded.'
                        SET @ErrorOccured = 1
                END
        ELSE IF DATEDIFF(MINUTE, GETEDATE(), @CheckInTime)/60.0>24
                BEGIN
                        PRINT '10% Service Charge Required!'
                        SELECT @TicketRefundRatio=0.9
                END
        ELSE IF DATEDIFF(MINUTE, GETEDATE(), @CheckInTime)/60.0>2
                BEGIN
                        PRINT '30% Service Charge Required!'
                        SELECT @TicketRefundRatio=0.7
                END
        ELSE
                BEGIN
                    PRINT '50% Service Charge Required!'
                    SELECT @TicketRefundRatio=0.7
                END

        IF @TicketPaid > 0
            BEGIN
             SELECT @TicketRefund= -@TicketRefundRatio*@TicketPaid
                INSERT INTO Costs (FeeTypeId,TicketId,CostDollars)
```

```
                                VALUES (@FeeTypeId,@TicketId,@TicketRefund)
            INSERT INTO Reservations (ReservationDescription)
                                VALUES ('Ticket Refunded.')

               DECLARE @ScheduledFlightId AS INT
               SELECT @ScheduledFlightId = ScheduledFlightId
                         FROM Tickets
                         WHERE TicketId = @TicketId

               PRINT 'Update Available Seats'
            UPDATE ScheduledFlights
                                SET AvailableSeats = @AvailableSeats + 1
                                WHERE ScheduledFlightId = @ScheduledFlightId

               PRINT 'Congratuation! Ticket Refunded Successfully. '

      IF @errorOccured = 1
             BEGIN
                    PRINT 'Rollback'
                    ROLLBACK TRAN
             END
      ELSE
             BEGIN
                    PRINT 'Commit'
                    COMMIT TRAN
             END;
```

(3) FireEmployee

```
-- SP3 FireEmployee and update the coresponding information
CREATE PROCEDURE dbo.FireEmployee( @EmployeeId AS VARCHAR(10)) AS
       DECLARE @Temp AS VARCHAR(10)

       DECLARE SubordinateCursor CURSOR FOR
                    SELECT EmployeeId
                           FROM Employees
                           WHERE DirectSupervisor = @EmployId

       OPEN SubordinateCursor
       FETCH NEXT FROM SubordinateCursor INTO @Temp

       WHILE @@FETCH_STATUS = 0
       BEGIN
              PRINT 'Sorry, Your boss has to be changed.'
              UPDATE Employees
                 SET DirectSupervisor=
                        (SELECT TOP 1 EmployeeId FROM Employees
                                 WHERE JobId =
                                                (SELECT JobId
                                                        FROM Employees
                                                        WHERE EmployeeId=@EmployId
                                                )
                                           AND EmployeeId <> @EmployId
                  )
                    FETCH NEXT FROM MyCursor INTO @temp
       END
```

```
        CLOSE MyCursor
        DEALLOCATE MyCursor

    UPDATE Employees
          SET EmployeeDescription='Already fired !!!'
          WHERE EmployeeId = @EmployeeId

    DELETE FROM Crew
          WEHRE EmployeeId = @EmployeeId

        DELETE FROM CanWork
          WEHRE EmployeeId = @EmployeeId
```

5. Functions

(1) FnGetTripInformation

```
--Get the trip information including ticketId, DepartureAirpport
ArrivalAirport,CheckInTime returned as a table
CREATE FUNCTION FnGetTripInformation(@CustomerId INT)
        RETURNS @TripInformationTable TABLE
               (
                CustomerId  VARCHAR(10),
                TicketId VARCHAR(11),
                DepartureAirpport VARCHAR(5),
                ArrivalAirport VARCHAR(5),
                CheckInTime DATETIME,
                TripId INT
               )
AS

BEGIN
    DECLARE @TripId VARCHAR(10)

    DECLARE GetTripIdCursor CURSOR FOR
                    SELECT TripId
                           FROM Trips
                                  WHERE CustomerId=@CustomerId

        OPEN GetTripIdCursor

        FETCH NEXT FROM GetTripIdCursor INTO @TripId

        WHILE @@FETCH_STATUS = 0
                    BEGIN
                            INSERT INTO @TripInformationTable (
                                           TicketId,
                                           CustomerId,

                                           DepartureAirpport,
                                                      ArrivalAirport,
                                                      CheckInTime,
                                                      TripId
                                                      )
                         SELECT DISTINCT Reservations.TicketId,
                                  Trips.CustomerId,
                                  FlightRoutes.DepartureAirpport,
```

```
                                        FlightRoutes.ArrivalAirport,
                                        Tickets.CheckInTime,
                                        Reservations.TripId
                          FROM Trips,
Reservations,Tickets,Personals,ScheduledFlights,FlightRoutes,Seats
                          WHERE Trips.TripId=@TripId
                               AND Reservations.TripId=@TripId
                               AND Personals.PersonId=@CustomerId
                               AND Tickets.TicketId=Reservations.TicketId
                               AND
Tickets.ScheduledFlightId=ScheduledFlights.ScheduledFlightId
                               AND
ScheduledFlights.FlightRouteId=FlightRoutes.FlightRoutId

                             FETCH NEXT FROM GetTripIdCursor INTO @TripId
                     END

       CLOSE GetTripIdCursor
       DEALLOCATE GetTripIdCursor

       RETURN
END;

drop function FnGetTripInformation

SELECT * FROM FnGetTripInformation('100126789')
```

(2) GetScheduledDepartureGate

```
--Funtion2 GetScheduledDepartureGate: Given the ticketId, to find out which gate should
the customer depart
CREATE FUNCTION dbo.GetScheduledDepartureGate( @TicketId AS VARCHAR(11))
    RETURNS VARCHAR(2) AS
       BEGIN
              DECLARE @GateName VARCHAR(2)

              SELECT @GateName= g.GateName
                     FROM dbo.Reservations r, dbo.ScheduledFlightGates f, dbo.Gates g
                     WHERE r.ScheduledFlightId=f.ScheduledFlightId
                     AND f.ActuralDepartureGate=g.GateId

        RETURN @GateName
    END;

SELECT dbo.GetScheduledDepartureGate('00171745175') AS DepartureGate;
```