Booking database system

[Type the abstract of the document here. The abstract is typically a short summary of the contents of the document. Type the abstract of the document here. The abstract is typically a short summary of the contents of the document.]

[Type the document subtitle]

Contents

[Request of a booking database for an airline company 2](#_Toc293249639)

[Use the Web to investigate the online booking system of a popular airline 4](#_Toc293249640)

[Entity\_Relationship Diagram 6](#_Toc293249641)

[The DDL statements required to create your relations 7](#_Toc293249643)

[The DDL statements required to create the constrains on the database relations 7](#_Toc293249644)

[INSERT statements to insert the test data into the database relations 8](#_Toc293249645)

[Queries 10](#_Toc293249646)

[A - Display the number of passengers stored in the system 10](#_Toc293249647)

[B - List all the aircraft which occur in scheduled flights. 10](#_Toc293249649)

[C - List all the flights from a specific departure airport 10](#_Toc293249659)

[D - List all the flights to a specific destination 11](#_Toc293249670)

[E - List all the flights which have seats available from a specific departure airport to a specific destination airport 12](#_Toc293249681)

[F - List all the flights which have seats available on a specific date from a specific departure airport to a specific destination airport 13](#_Toc293249693)

[G - Create a booking for a specific passenger for a specific flight 14](#_Toc293249705)

[H - Update the seat availability for a specific flight 15](#_Toc293249720)

[I - List all the bookings for a specific passenger 15](#_Toc293249721)

[J - Calculate the total cost for a customer for all the flights which she has booked 16](#_Toc293249722)

[DDL to create tables and set up constraints 16](#_Toc293249723)

[DML to load data into the database 22](#_Toc293249724)

[Java Program which updates the flight costs by 10% for all flights departing after December 31st, 2005 24](#_Toc293249725)

# Request of a booking database for an airline company

A booking database is required for an airline company. The company stores information on the following entities:

Airline Company, Passenger, Airport, Aircraft, Flight, Flight Availability and Booking

1. Use the Web to investigate the online booking system of a popular airline.
2. Using the information you have observed, choose attributes which are associated with each of the entities listed above, identifying a new entities which form part of the system.
3. Create an Entity\_Relationship Diagram, showing the entities, the relationships, and the constraints which characterize the system.
4. Check this against your chosen airline system.
5. Convert the ER diagram into a set of relations.
6. Generate the DDL statements required to create your relations.
7. Generate the DDL statements required to create the constrains on your relations.
8. Using MySQL, create your database and set up the constraints.
9. Choose representative test data for each relation.
10. Create INSERT statements to insert the test data into your relations and insert the test data.
11. Create queries to satisfy the following requirements:
    1. Display the number of passengers stored in the system.
    2. List all the aircraft which occur in scheduled flights.
    3. List all the flights from a specific departure airport.
    4. List all the flights to a specific destination.
    5. List all the flights which have seats available from a specific departure airport to a specific destination airport.
    6. List all the flights which have seats available on a specific date from a specific departure airport to a specific destination airport.
    7. Create a booking for a specific passenger for a specific flight.
    8. Update the seat availability for a specific flight.
    9. List all the bookings for a specific passenger.
    10. Calculate the total cost for a customer for all the flights which she has booked.
12. Run EXPLAIN on queries, i, and j, listed in sub task 11.
13. Create indices on relations, Passenger, Bookings, Flight, and FlightAvailability.
14. Rerun EXPLAIN on queries, i, and j, listed in sub task 11.
15. Compare the results obtained from completing sub task 12 and sub task 14, and write a short report, explaining the differences.
16. Write a Java program, using JDBC and transaction management, which updates the flight costs by 10% for all flights departing after December 31st, 2005.

This topic is concerned with database systems. It is important that you schedule your time, so that work is of a high quality and is submitted on time.

It is also important to ensure the quality of the database design by careful attention to detail and the development and execution of effective test strategies.

The submitted documentation folder for this assessment submission must contain the following:

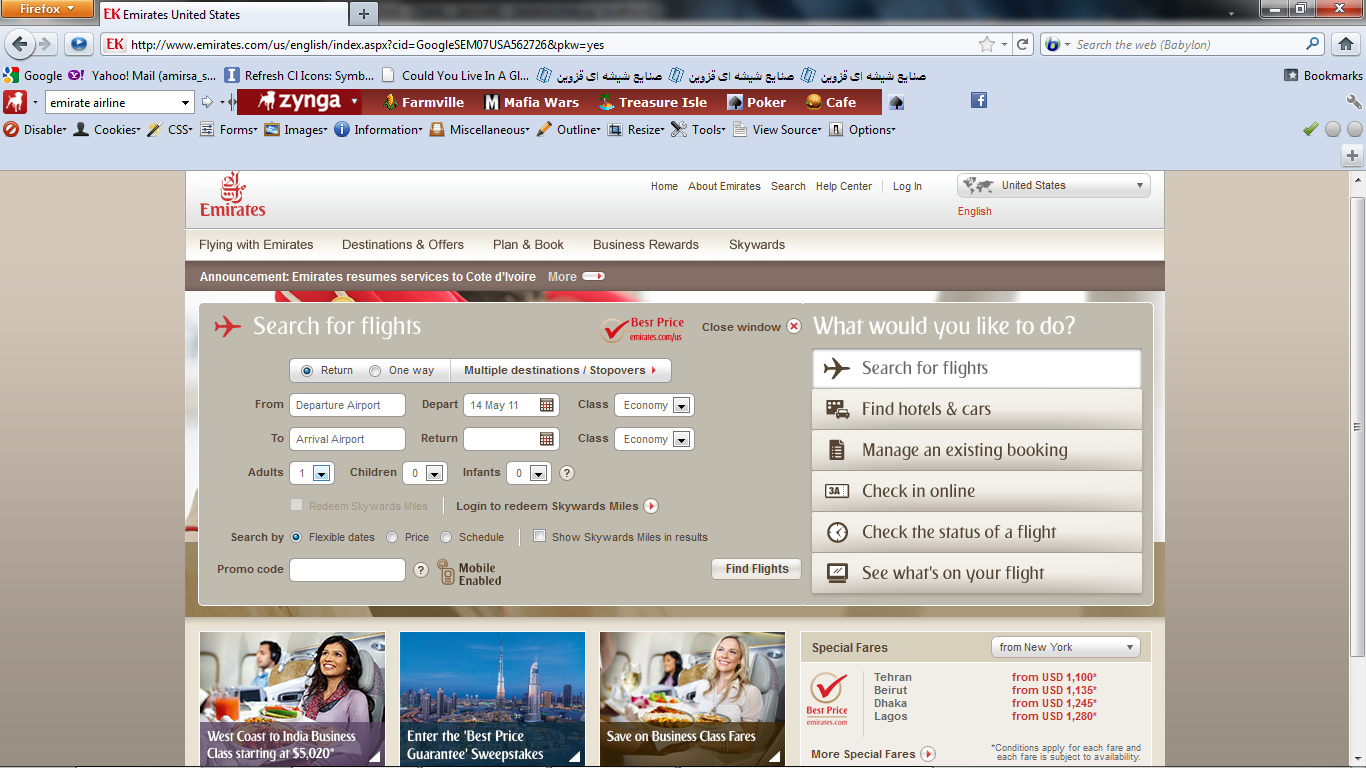
* A statement of the problem, identifying the airline on which your database is based
* Conceptual Design
  + Entity Relationship Diagram(s) showing entities, relationships, and constraints
* Logical Design
  + Diagrammatic representation of each relation derived from the ER diagram showing all attributes, keys, and foreign keys
  + Normalization of data
  + Updated representation of the relations
  + Domain specification for each attribute
  + Relation constraints
* Physical Design , indicating indexes and any changes to normalized relations, justifying the inclusion of each index and/or change to the normalized relations
  + DDL to create tables
  + DDL to set up constraints
  + DML to load data into the database
* Tabular representation of the data in each relation, indicating the keys and foreign keys
* List of information requirements, the query that was generated in response to each requirement, followed by a representation the table generated in response to each query

# Use the Web to investigate the online booking system of a popular airline

“http://www.emirates.com/us/english/index.aspx?cid=GoogleSEM07USA562726&pkw=yes “

Emirate Airline is one of the most popular airlines in the Middle East. In the web site of the company to facilitate the booking process, it creates a link for searching flights. The necessary information which the form needs to find that specific flight listed below:

* + 1. First of all there is radio button to choose one way or two way flight
    2. Multiple destination / stopovers button redirect customers to a page which contains a form of multiple “Search for flights” form. Customers can add many flights to the list.
    3. The departure specification which contains name of an airport, a date and a seat type or class.
    4. In the fourth step, they get the destination information which is its airport name. The return date and return seat class only are visible in a return flight not in one way flights.
    5. Number and type of passengers, the airline has specific price for children and infants. In this step number of passengers will get.
    6. Finally the customers by clicking on find flights button submit the form and get the information about that specific flight



# Entity\_Relationship Diagram

# C:\Users\saleh\Desktop\Drawing1.jpg

# The DDL statements required to create your relations

|  |
| --- |
| ALTER TABLE *[table name]*  ADD CONSTRAINT `*[RealationName\_FK]*` FOREIGN KEY (`*[ForeignkeyColumnName]*`) REFERENCES `*[ReferenceTableName]*` (`*[ReferenceForeignKeyColumnName]*`) ON DELETE CASCADE ON UPDATE CASCADE; |

# The DDL statements required to create the constrains on the database relations

|  |
| --- |
| ALTER TABLE `flight`  CONSTRAINT `AircraftID\_FK` FOREIGN KEY (`AircraftID`) REFERENCES `aircraft` (`AircraftID`) ON DELETE CASCADE ON UPDATE CASCADE;  ALTER TABLE `flight`  CONSTRAINT `AirlineID\_FK` FOREIGN KEY (`AirlineID`) REFERENCES `airline` (`AirlineID`) ON DELETE CASCADE ON UPDATE CASCADE;  ALTER TABLE `flight`  CONSTRAINT `DepartureAirportID\_FK` FOREIGN KEY (`DepartureAirportID`) REFERENCES `airport` (`AirportID`) ON DELETE CASCADE ON UPDATE CASCADE;  ALTER TABLE `flight`  CONSTRAINT `DestinationAirportID\_FK` FOREIGN KEY (`DestinationAirportID`) REFERENCES `airport` (`AirportID`) ON DELETE CASCADE ON UPDATE CASCADE;  ALTER TABLE `flight`  CONSTRAINT `FlightID\_FK` FOREIGN KEY (`FlightID`) REFERENCES `flightavailability` (`FlightAvailabilityID`) ON DELETE CASCADE ON UPDATE CASCADE;  ALTER TABLE `booking`  CONSTRAINT `BookingFlightAvailabilityID\_FK` FOREIGN KEY (`FlightAvailabilityID`) REFERENCES `flightavailability` (`FlightAvailabilityID`) ON DELETE CASCADE ON UPDATE CASCADE;  ALTER TABLE `booking`  CONSTRAINT `PassengerID\_FK` FOREIGN KEY (`PassengerID`) REFERENCES `passenger` (`PassengerID`) ON DELETE CASCADE ON UPDATE CASCADE;  ALTER TABLE `booking`  CONSTRAINT `SeatClassID\_FK` FOREIGN KEY (`SeatClassID`) REFERENCES `seatclass` (`SeatClassID`) ON DELETE CASCADE ON UPDATE CASCADE; |

# INSERT statements to insert the test data into the database relations

|  |
| --- |
| INSERT INTO `aircraft` (`AircraftID`, `ModelNumber`, `EnginePower`, `Company`) VALUES  (1,747,40000,'boeing'),  (2,777,55000,'boeing'),  (3,545,38000,'foker');  COMMIT;  INSERT INTO `airline` (`AirlineID`, `Name`, `Telephone`, `WebAddress`, `Address`) VALUES  (1,'Lufthansa','85 453 36 37','www.lufthansa.com','Berlin - Germany'),  (2,'KLM','76 876 87 90','www.klm.com','Amesterdam - Poland'),  (3,'Emirate','97 152 56 78','www.emirate.com','dubai - emirate');  COMMIT;  INSERT INTO `airport` (`AirportID`, `AirportName`, `City`, `Country`) VALUES  (1,'London ','London','UK'),  (2,'Paris','paris','france'),  (3,'madrid','Madrid','Spain');  COMMIT;  INSERT INTO `passenger` (`PassengerID`, `Name`, `FamilyName`, `Gender`, `Address`, `Telephone`, `Email`) VALUES  (1,'Jack','jonson','0','Otawa - Canada','921437 94 94','email@email.com'),  (2,'jenna','cornney','1','Dubai - united arabian of emirate','971 543 73 48','jenna@email.com'),  (3,'Talieh','Dastmalchi','1','Tehran - Iran','98 912 113 5665','Talieh@email.com');  COMMIT;  INSERT INTO `seatclass` (`SeatClassID`, `Class`) VALUES  (1,'economy'),  (2,'business'),  (3,'first class');  COMMIT; |

# Queries

|  |
| --- |
| A - Display the number of passengers stored in the system |
| SELECT COUNT( `passenger`.`PassengerID` ) FROM `passenger` |

|  |
| --- |
| B- List all the aircraft which occur in scheduled flights. |
| SELECTaircraft.Company,aircraft.ModelNumber,flight.FlightNumber,flight.`Date`,flight.`Time`FROMflightINNER JOIN aircraft ON (flight.AircraftID = aircraft.AircraftID) |

|  |
| --- |
| C - List all the flights from a specific departure airport |
| SELECT`flight`.`Date`,`flight`.`Time`,`flight`.`FlightNumber`,`airport`.`AirportName` as 'Departure airport'FROMflightINNER JOIN airport ON (flight.DepartureAirportID = airport.AirportID)WHEREairport.AirportName LIKE 'madrid' |

|  |
| --- |
| D - List all the flights to a specific destination |
| SELECTflight.`Date`,flight.`Time`,flight.FlightNumber,airport.AirportName AS `Destination airport`FROMflightINNER JOIN airport ON (flight.DestinationAirportID = airport.AirportID)WHEREairport.AirportName LIKE 'Paris' |

|  |
| --- |
| E - List all the flights which have seats available from a specific departure airport to a specific destination airport |
| SELECTflight.`Date`,flight.`Time`,flight.FlightNumberFROMflightINNER JOIN flightavailability ON (flight.FlightID = flightavailability.FlightID)WHEREflight.DepartureAirportID = (SELECT airport.AirportID FROM airport WHERE airport.City LIKE 'London') ANDflight.DestinationAirportID = (SELECT airport.AirportID FROM airport WHERE airport.City LIKE 'Paris') ANDflightavailability.AvailableSeats > 0 |

|  |
| --- |
| F - List all the flights which have seats available on a specific date from a specific departure airport to a specific destination airport |
| SELECTflight.`Date`,flight.`Time`,flight.FlightNumberFROMflightINNER JOIN flightavailability ON (flight.FlightID = flightavailability.FlightID)WHEREflight.DepartureAirportID = (SELECT airport.AirportID FROM airport WHERE airport.City LIKE 'London') ANDflight.DestinationAirportID = (SELECT airport.AirportID FROM airport WHERE airport.City LIKE 'Paris') ANDflightavailability.AvailableSeats > 0 AND flight.`date` = ' 2011-05-06' |

|  |
| --- |
| G - Create a booking for a specific passenger for a specific flight |
| INSERT INTO `booking` ( `booking`.`FlightAvailabilityID` , `booking`.`PassengerID` , `booking`.`TotalPrice`, `booking`.`SeatClassID` )VALUES (( SELECT `flightavailability`.`FlightAvailabilityID`FROM `flightavailability`WHERE `flightavailability`.`FlightID` =1), (SELECT `passenger` .`PassengerID`FROM `passenger`WHERE `passenger`.`Name` LIKE 'Talieh'AND `passenger`.`FamilyName` LIKE 'Dastmalchi'), 1800,(SELECT `seatclass`.`SeatClassID`FROM`seatclass`WHERE `seatclass`.`Class` LIKE 'business')); |

|  |
| --- |
| H - Update the seat availability for a specific flight |
| UPDATE `flightavailability` SET `flightavailability`.`AvailableSeats` = 15  WHERE  `flightavailability`.`FlightID` = ( SELECT `flight`.`FlightID`  FROM flight  WHERE `flight`.`DepartureAirportID` = (  SELECT `airport`.`AirportID`  FROM `airport`  WHERE city = 'London' ) ) |

|  |
| --- |
| I - List all the bookings for a specific passenger |
| SELECT  `passenger`.`Name`,  `passenger`.`FamilyName`,  `passenger`.`Address`,  `passenger`.`Telephone`,  `flight`.`FlightNumber`,  `flight`.`Date`,  `flight`.`Time`,  `booking`.`TotalPrice`  FROM  `flight`,  `booking`,  `passenger`,  `flightavailability`  WHERE  `passenger`.`Name` = 'Talieh' AND  `passenger`.`FamilyName` = 'Dastmalchi' AND  `passenger`.`PassengerID` = `booking`.`PassengerID` AND  `booking`.`FlightAvailabilityID` = `flightavailability`.`FlightAvailabilityID` AND  `flightavailability`.`FlightID` = `flight`.`FlightID` |

|  |
| --- |
| J - Calculate the total cost for a customer for all the flights which she has booked |
| SELECT  SUM(`booking`.`TotalPrice`)  FROM  `booking`,  `passenger`  WHERE  `booking`.PassengerID = `passenger`.`PassengerID` AND  `passenger`.`Name` = 'Talieh' AND  `passenger`.`FamilyName` = 'Dastmalchi' |

# DDL to create tables and set up constraints

|  |
| --- |
| SET FOREIGN\_KEY\_CHECKS=0;  CREATE DATABASE `taliehairline`  CHARACTER SET 'latin1'  COLLATE 'latin1\_swedish\_ci';  USE `taliehairline`;  CREATE TABLE `aircraft` (  `AircraftID` int(20) NOT NULL AUTO\_INCREMENT,  `ModelNumber` int(20) NOT NULL,  `EnginePower` int(11) DEFAULT NULL,  `Company` varchar(20) DEFAULT NULL,  PRIMARY KEY (`AircraftID`),  UNIQUE KEY `AircraftID` (`AircraftID`)  ) ENGINE=InnoDB AUTO\_INCREMENT=4 DEFAULT CHARSET=latin1 COMMENT='InnoDB free: 3072 kB';  CREATE TABLE `airline` (  `AirlineID` int(20) NOT NULL AUTO\_INCREMENT,  `Name` varchar(20) NOT NULL,  `Telephone` varchar(20) NOT NULL,  `WebAddress` varchar(60) NOT NULL,  `Address` text,  PRIMARY KEY (`AirlineID`),  UNIQUE KEY `AirlineID` (`AirlineID`)  ) ENGINE=InnoDB AUTO\_INCREMENT=4 DEFAULT CHARSET=latin1 COMMENT='InnoDB free: 3072 kB';  CREATE TABLE `airport` (  `AirportID` int(20) NOT NULL AUTO\_INCREMENT,  `AirportName` varchar(20) NOT NULL DEFAULT '',  `City` varchar(20) NOT NULL,  `Country` varchar(20) DEFAULT NULL,  PRIMARY KEY (`AirportID`),  UNIQUE KEY `AirportID` (`AirportID`)  ) ENGINE=InnoDB AUTO\_INCREMENT=4 DEFAULT CHARSET=latin1 COMMENT='InnoDB free: 3072 kB';  CREATE TABLE `flightavailability` (  `FlightAvailabilityID` int(20) NOT NULL AUTO\_INCREMENT,  `AvailableSeats` int(20) NOT NULL,  `FlightID` int(20) NOT NULL,  PRIMARY KEY (`FlightAvailabilityID`),  UNIQUE KEY `FlightAvailabilityID` (`FlightAvailabilityID`)  ) ENGINE=InnoDB AUTO\_INCREMENT=3 DEFAULT CHARSET=latin1 COMMENT='InnoDB free: 3072 kB; (`flightid`) REFER `tannazflight/flight`(`flightId`)';  CREATE TABLE `passenger` (  `PassengerID` int(20) NOT NULL AUTO\_INCREMENT,  `Name` varchar(20) NOT NULL,  `FamilyName` varchar(20) NOT NULL,  `Gender` binary(1) NOT NULL,  `Address` text,  `Telephone` varchar(20) DEFAULT NULL,  `Email` varchar(100) DEFAULT NULL,  PRIMARY KEY (`PassengerID`),  UNIQUE KEY `PassengerID` (`PassengerID`)  ) ENGINE=InnoDB AUTO\_INCREMENT=4 DEFAULT CHARSET=latin1 COMMENT='InnoDB free: 3072 kB';  CREATE TABLE `seatclass` (  `SeatClassID` int(20) NOT NULL AUTO\_INCREMENT,  `Class` varchar(20) NOT NULL,  PRIMARY KEY (`SeatClassID`),  UNIQUE KEY `SeatClassID` (`SeatClassID`)  ) ENGINE=InnoDB AUTO\_INCREMENT=4 DEFAULT CHARSET=latin1 COMMENT='InnoDB free: 3072 kB';  CREATE TABLE `booking` (  `BookingID` int(20) NOT NULL AUTO\_INCREMENT,  `TotalPrice` int(20) NOT NULL,  `FlightAvailabilityID` int(20) NOT NULL,  `PassengerID` int(20) NOT NULL,  `SeatClassID` int(11) DEFAULT NULL,  PRIMARY KEY (`BookingID`),  UNIQUE KEY `BookingID` (`BookingID`),  KEY `FlightAvailabilityID` (`FlightAvailabilityID`),  KEY `PassengerID` (`PassengerID`),  KEY `SeatClassID` (`SeatClassID`),  CONSTRAINT `BookingFlightAvailabilityID\_FK` FOREIGN KEY (`FlightAvailabilityID`) REFERENCES `flightavailability` (`FlightAvailabilityID`) ON DELETE CASCADE ON UPDATE CASCADE,  CONSTRAINT `PassengerID\_FK` FOREIGN KEY (`PassengerID`) REFERENCES `passenger` (`PassengerID`) ON DELETE CASCADE ON UPDATE CASCADE,  CONSTRAINT `SeatClassID\_FK` FOREIGN KEY (`SeatClassID`) REFERENCES `seatclass` (`SeatClassID`) ON DELETE CASCADE ON UPDATE CASCADE  ) ENGINE=InnoDB AUTO\_INCREMENT=4 DEFAULT CHARSET=latin1 COMMENT='InnoDB free: 3072 kB; (`customerID`) REFER `tannazflight/customer`(`customerID`)';  CREATE TABLE `flight` (  `FlightID` int(20) NOT NULL AUTO\_INCREMENT,  `DepartureAirportID` int(20) NOT NULL,  `DestinationAirportID` int(20) NOT NULL,  `Time` time NOT NULL,  `Date` date NOT NULL,  `AirlineID` int(20) NOT NULL,  `AircraftID` int(20) NOT NULL,  `FlightNumber` int(11) DEFAULT NULL,  PRIMARY KEY (`FlightID`),  UNIQUE KEY `FlightID` (`FlightID`),  KEY `DepartureAirportID` (`DepartureAirportID`,`DestinationAirportID`),  KEY `DestinationAirportID` (`DestinationAirportID`,`AirlineID`,`AircraftID`),  KEY `AirlineID` (`AirlineID`),  KEY `AircraftID` (`AircraftID`),  KEY `DepartureAirportID\_2` (`DepartureAirportID`),  CONSTRAINT `AircraftID\_FK` FOREIGN KEY (`AircraftID`) REFERENCES `aircraft` (`AircraftID`) ON DELETE CASCADE ON UPDATE CASCADE,  CONSTRAINT `AirlineID\_FK` FOREIGN KEY (`AirlineID`) REFERENCES `airline` (`AirlineID`) ON DELETE CASCADE ON UPDATE CASCADE,  CONSTRAINT `DepartureAirportID\_FK` FOREIGN KEY (`DepartureAirportID`) REFERENCES `airport` (`AirportID`) ON DELETE CASCADE ON UPDATE CASCADE,  CONSTRAINT `DestinationAirportID\_FK` FOREIGN KEY (`DestinationAirportID`) REFERENCES `airport` (`AirportID`) ON DELETE CASCADE ON UPDATE CASCADE,  CONSTRAINT `FlightID\_FK` FOREIGN KEY (`FlightID`) REFERENCES `flightavailability` (`FlightAvailabilityID`) ON DELETE CASCADE ON UPDATE CASCADE  ) ENGINE=InnoDB AUTO\_INCREMENT=3 DEFAULT CHARSET=latin1 COMMENT='InnoDB free: 3072 kB; (`aircraftId`) REFER `tannazflight/aircraft`(`aircraftId`)';  INSERT INTO `aircraft` (`AircraftID`, `ModelNumber`, `EnginePower`, `Company`) VALUES  (1,747,40000,'boeing'),  (2,777,55000,'boeing'),  (3,545,38000,'foker');  COMMIT;  INSERT INTO `airline` (`AirlineID`, `Name`, `Telephone`, `WebAddress`, `Address`) VALUES  (1,'Lufthansa','85 453 36 37','www.lufthansa.com','Berlin - Germany'),  (2,'KLM','76 876 87 90','www.klm.com','Amesterdam - Poland'),  (3,'Emirate','97 152 56 78','www.emirate.com','dubai - emirate');  COMMIT;  INSERT INTO `airport` (`AirportID`, `AirportName`, `City`, `Country`) VALUES  (1,'London ','London','UK'),  (2,'Paris','paris','france'),  (3,'madrid','Madrid','Spain');  COMMIT;  INSERT INTO `flightavailability` (`FlightAvailabilityID`, `AvailableSeats`, `FlightID`) VALUES  (1,200,1),  (2,500,2);  COMMIT;  INSERT INTO `passenger` (`PassengerID`, `Name`, `FamilyName`, `Gender`, `Address`, `Telephone`, `Email`) VALUES  (1,'Jack','jonson','0','Otawa - Canada','921437 94 94','email@email.com'),  (2,'jenna','cornney','1','Dubai - united arabian of emirate','971 543 73 48','jenna@email.com'),  (3,'Talieh','Dastmalchi','1','Tehran - Iran','98 912 113 5665','Talieh@email.com');  COMMIT;  INSERT INTO `seatclass` (`SeatClassID`, `Class`) VALUES  (1,'economy'),  (2,'business'),  (3,'first class');  COMMIT;  INSERT INTO `booking` (`BookingID`, `TotalPrice`, `FlightAvailabilityID`, `PassengerID`, `SeatClassID`) VALUES  (1,1800,1,3,2),  (2,4000,1,1,1),  (3,2000,2,3,1);  COMMIT;  INSERT INTO `flight` (`FlightID`, `DepartureAirportID`, `DestinationAirportID`, `Time`, `Date`, `AirlineID`, `AircraftID`, `FlightNumber`) VALUES  (1,1,2,'14:25:43','2011-05-06',1,2,322),  (2,3,1,'15:45:00','2011-05-07',2,1,323);  COMMIT; |

# DML to load data into the database

|  |
| --- |
| LOAD DATA **LOCAL** INFILE '/taliehBooking.txt'  INTO TABLE booking  FIELDS TERMINATED BY ','  LINES TERMINATED BY '\n'  (BookingID,TotalPrice, FlightAvailabilityID, PassengerID,SeatClassID);  LOAD DATA **LOCAL** INFILE '/taliehPassenger.txt'  INTO TABLE passenger  FIELDS TERMINATED BY ','  LINES TERMINATED BY '\n'  (PassengerID,Name, FamilyName, Gender,Address,Telephone,Email);  LOAD DATA **LOCAL** INFILE '/taliehFlight.txt'  INTO TABLE flight  FIELDS TERMINATED BY ','  LINES TERMINATED BY '\n'  (FlightID,Name, DepartureAirportID, DestinationAirportID, Time, Date, AirlineID, AircraftID, FlightNumber);  LOAD DATA **LOCAL** INFILE '/taliehAircraft.txt'  INTO TABLE aircraft  FIELDS TERMINATED BY ','  LINES TERMINATED BY '\n'  (AircraftID, ModelNumber, EnginePower, Company);  LOAD DATA **LOCAL** INFILE '/taliehAirline.txt'  INTO TABLE airline  FIELDS TERMINATED BY ','  LINES TERMINATED BY '\n'  (AirlineID, Name, Telephone, WebAddress, Address);  LOAD DATA **LOCAL** INFILE '/taliehSeatClass.txt'  INTO TABLE seatclass  FIELDS TERMINATED BY ','  LINES TERMINATED BY '\n'  (SeatClassID,Class);  LOAD DATA **LOCAL** INFILE '/taliehFlightAvailability.txt'  INTO TABLE flightavailability  FIELDS TERMINATED BY ','  LINES TERMINATED BY '\n'  (FlightAvailabilityID, AvailableSeats, FlightID); |

|  |
| --- |
| Java Program which updates the flight costs by 10% for all flights departing after December 31st, 2005 |
| import java.sql.Connection;  import java.sql.DriverManager;  import java.sql.PreparedStatement;  import java.sql.ResultSet;  import java.sql.Statement;  public class UpdatePriceTest {  public void updateBooking() {  try{  Class.*forName*("com.mysql.jdbc.Driver");  Connection con=DriverManager.*getConnection*("jdbc:mysql://localhost:3306/taliehairline", "root", "root");  String sql= "SELECT flight.`Date`, flight.`Time`, flight.FlightNumber, `booking`.`BookingID`,`booking`.`TotalPrice`"+  "FROM flight INNER JOIN flightavailability ON (flight.FlightID = flightavailability.FlightID), `booking`" +  "WHERE `booking`.`FlightAvailabilityID` = `flightavailability`.`FlightAvailabilityID` AND flight.`date` > ' 2005-12-31'";  Statement stmt=con.createStatement();  PreparedStatement ps = null;  ResultSet rs = stmt.executeQuery(sql);  while(rs.next())  {  sql="UPDATE `booking` SET TotalPrice = ? WHERE BookingID = ?";  int price = rs.getInt("TotalPrice");  price = price - (price \* 10/100);  ps = con.prepareStatement(sql);  ps.setInt(1, price);  ps.setInt(2, rs.getInt("BookingID"));  ps.execute();  }      ps.close();  con.close();    }  catch (Exception EX){  EX.printStackTrace();  }  }  public static void main(String[] args) {  UpdatePriceTest priceTest = new UpdatePriceTest();  priceTest.updateBooking();  }  } |