

**TIS1101 – Database Fundamentals**

**Group Project: Part 2**

**Airline Reservation System**

Group Members (TC01 | TT02):

1. Samuel Wong (1151103687)
2. Wayile Jialade (1151102347)

Contents

[Business Rules: 3](#_Toc494322466)

[Entity Relationship Diagram: 4](#_Toc494322467)

[Data Dictionary 5](#_Toc494322468)

[Data Definition Language (DDL): 7](#_Toc494322469)

[Creating the database 7](#_Toc494322470)

[Connecting to the database 7](#_Toc494322471)

[Create tables 7](#_Toc494322472)

[LIST TABLES 9](#_Toc494322473)

[Data Insertion: 10](#_Toc494322474)

[Contact Table 10](#_Toc494322475)

[Aircraft Table 12](#_Toc494322476)

[Fare Table 14](#_Toc494322477)

[Passenger Table 16](#_Toc494322478)

[Flight Table 17](#_Toc494322479)

[Book Table 19](#_Toc494322480)

[Data Manipulation Language (DML): 21](#_Toc494322481)

[Triggers: 21](#_Toc494322482)

[Stored Procedure: 24](#_Toc494322483)

[Aggregate function : 28](#_Toc494322484)

[Query with a Group by and Having clauses: 30](#_Toc494322485)

[Query with a Subquery: 31](#_Toc494322486)

[Queries that are not covered in lecture: 32](#_Toc494322487)

# Business Rules:

1. All passengers have to submit their names, contact details, gender and passport no.

2. All passengers can access their flight details such as flight date, departure time, arrival time, starting location, arrival location, duration of flight and aircraft details.

3. All passengers have to pay for their flight tickets which depend on the original price and the discount rate if there is any.

4. All passengers have to provide their emails, cell phone numbers and telephone numbers as their contact details.

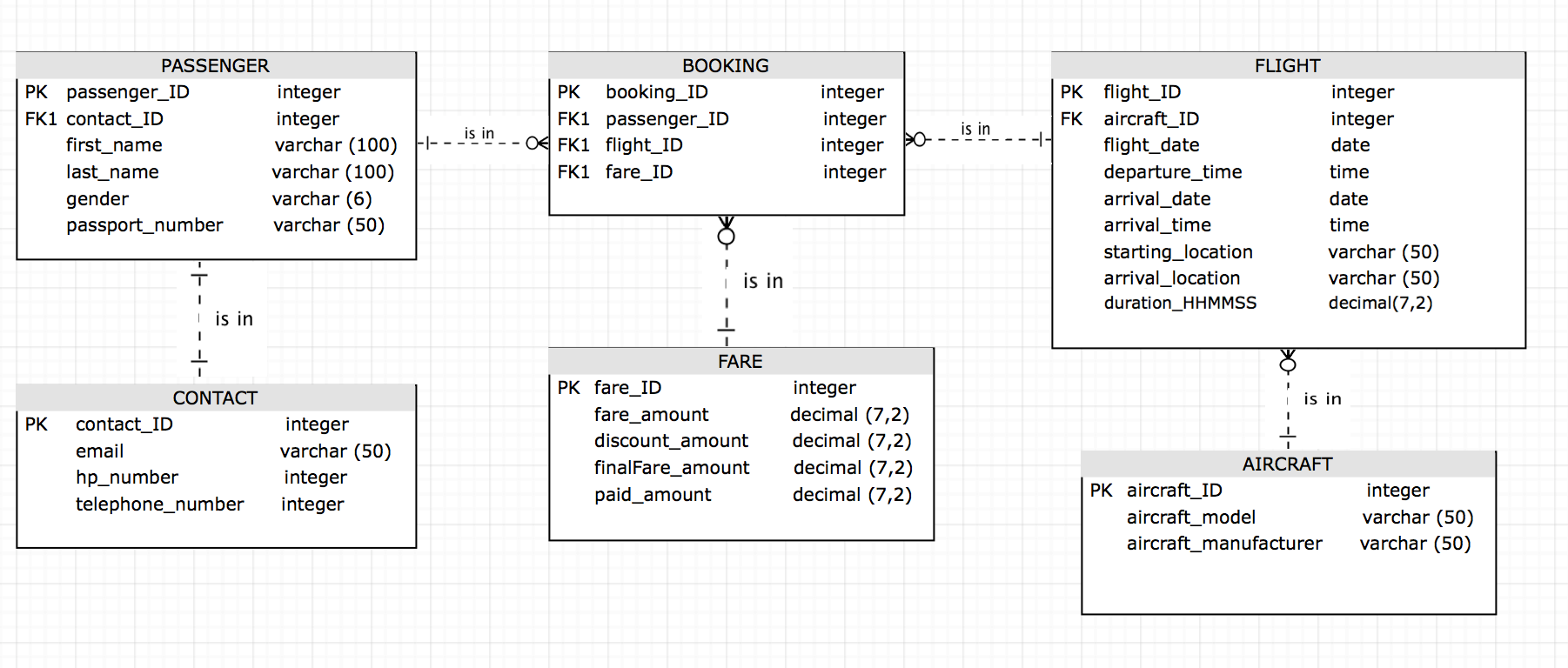
5. Flight details will include aircraft details such as aircraft model and aircraft manufacturer.

6. All details for the booking of a flight will be displayed on the ticket and will include all information about the passenger, flight and fares.

7. Arrival times for all available flights will be before 11:59pm of any given day and departure times for all available flights will be after 12:01am of any given day.

8. The company will provide free gifts for the first 3 passengers of all flights.

# Entity Relationship Diagram:

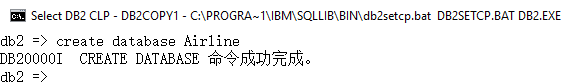


# Data Dictionary

# Data Definition Language (DDL):

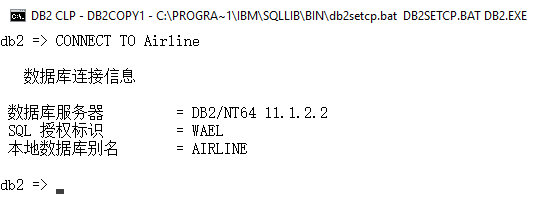
## Creating the database

CREATE DATABASE Airline



## Connecting to the database

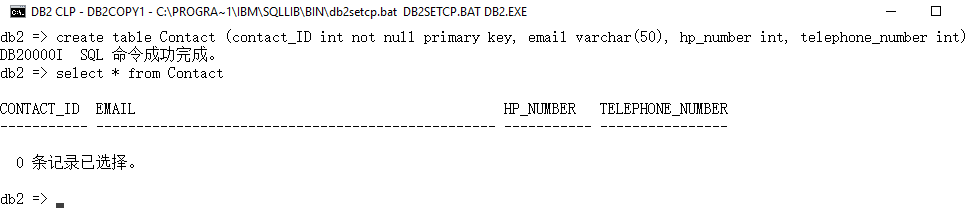
CONNECT TO Airline



## Create tables

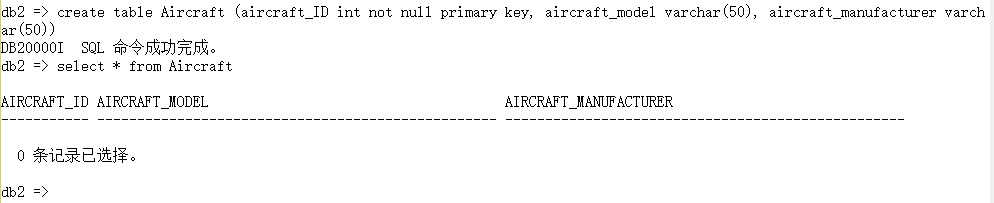
create table Contact (contact\_ID int not null primary key, email varchar(50), hp\_number int, telephone\_number int) ;

select \* from Contact



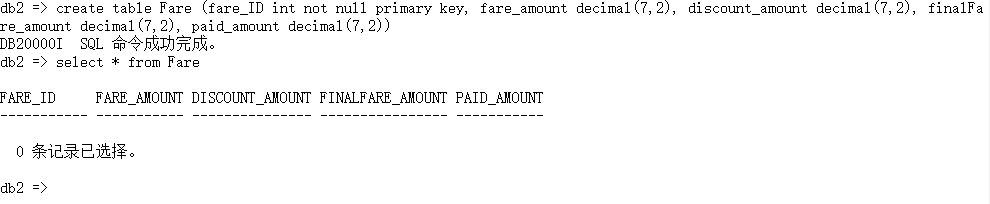
create table Aircraft (aircraft\_ID int not null primary key, aircraft\_model varchar(50), aircraft\_manufacturer varchar(50)) ;

select \* from Aircraft



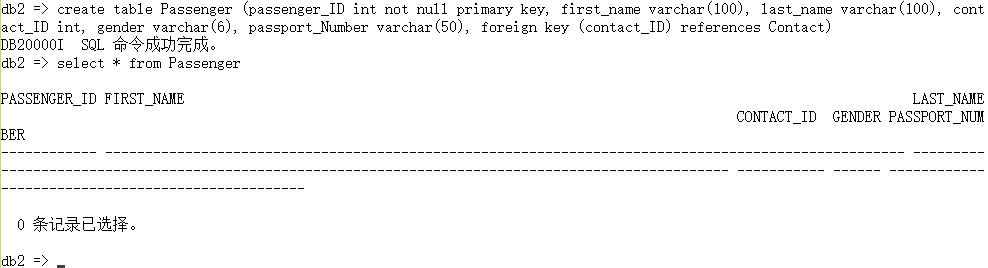
create table Fare (fare\_ID int not null primary key, fare\_amount decimal(7,2), discount\_amount decimal(7,2), finalFare\_amount decimal(7,2), paid\_amount decimal(7,2)) ;

select \* from Fare



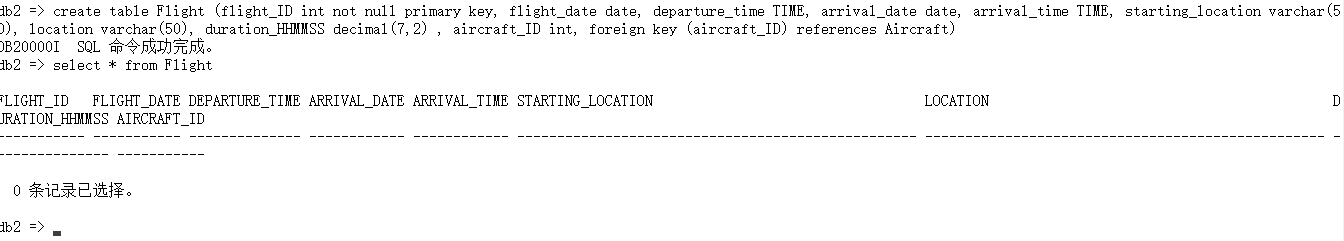
create table Passenger (passenger\_ID int not null primary key, first\_name varchar(100), last\_name varchar(100), contact\_ID int, gender varchar(6), passport\_Number varchar(50), foreign key (contact\_ID) references Contact) ;

select \* from Passenger



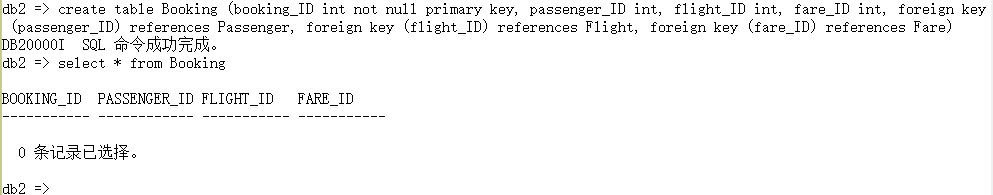
create table Flight (flight\_ID int not null primary key, flight\_date date, departure\_time TIME, arrival\_date date, arrival\_time TIME, starting\_location varchar(50), location varchar(50), duration\_HHMMSS decimal(7,2) , aircraft\_ID int, foreign key (aircraft\_ID) references Aircraft) ;

select \* from Flight

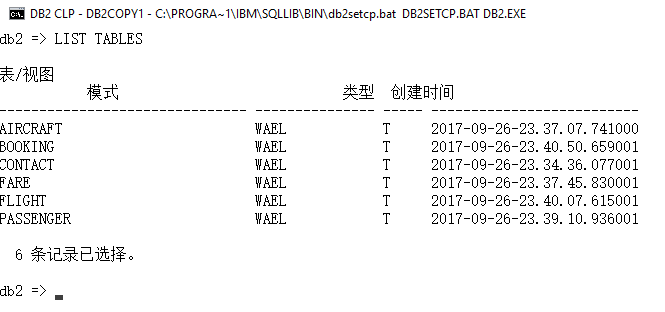


create table Booking (booking\_ID int not null primary key, passenger\_ID int, flight\_ID int, fare\_ID int, foreign key (passenger\_ID) references Passenger, foreign key (flight\_ID) references Flight, foreign key (fare\_ID) references Fare) ;

select \* from Booking



## LIST TABLES



# Data Insertion:

## Contact Table

insert into Contact values (001, 'theguy007@gmail.com', 0123345599, 083454992)

insert into Contact values (002, 'batmanfly@gmail.com', 0155240279, 083773977)

insert into Contact values (003, 'geargod@gmail.com', 0111045319, 083599954)

insert into Contact values (004, 'chessit87@gmail.com', 0111045319, 083599954)

insert into Contact values (005, 'whatsupbro@gmail.com', 0141065619, 076589364)

insert into Contact values (006, 'wrty1290y@gmail.com', 013457890, 090657487)

insert into Contact values (007, 'uighurbty9@gmail.com', 0141065619, 076589364)

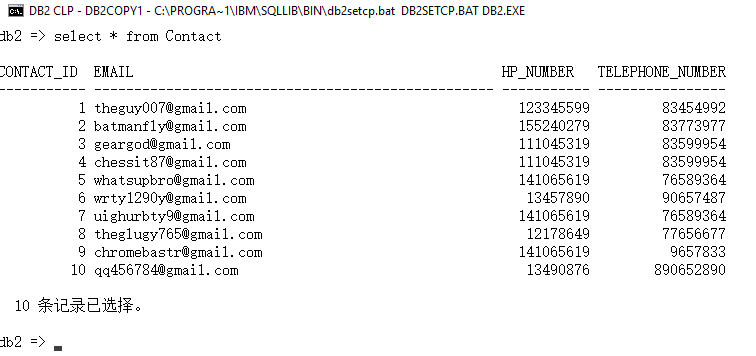
insert into Contact values (008, 'theglugy765@gmail.com', 012178649, 077656677)

insert into Contact values (009, 'chromebastr@gmail.com', 0141065619, 09657833)

insert into Contact values (010, 'qq456784@gmail.com', 013490876, 0890652890)



select \* from Contact



## Aircraft Table

insert into Aircraft values (1001, 'A380F', 'Airbus')

insert into Aircraft values (1002, '747-100', 'Boeing')

insert into Aircraft values (1003, 'C105', 'Dot')

insert into Aircraft values (1004, 'A380F', 'Airbus')

insert into Aircraft values (1005, '737-100', 'Boeing')

insert into Aircraft values (1006, 'C105', 'Dot')

insert into Aircraft values (1007, ' A380F ', 'Airbus')

insert into Aircraft values (1008, '747-100', 'Boeing')

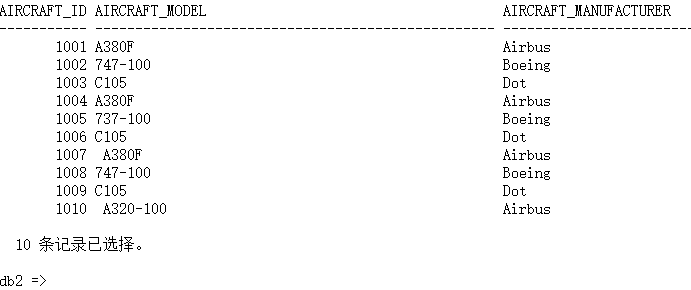
insert into Aircraft values (1009, 'C105', 'Dot')

insert into Aircraft values (1010, ' A320-100 ', 'Airbus')





select \* from Aircraft



## Fare Table

insert into Fare values (12001, 350.00, 15.00, 335.00, 335.00)

insert into Fare values ( 12002, 750.00, 35.00, 715.00, 715.00)

insert into Fare values ( 12003, 800.00, 35.00, 765.00, 765.00)

insert into Fare values ( 12004, 850.00, 400.00, 435.00, 435.00)

insert into Fare values ( 12005, 900.00, 135.00, 765.00, 765.00)

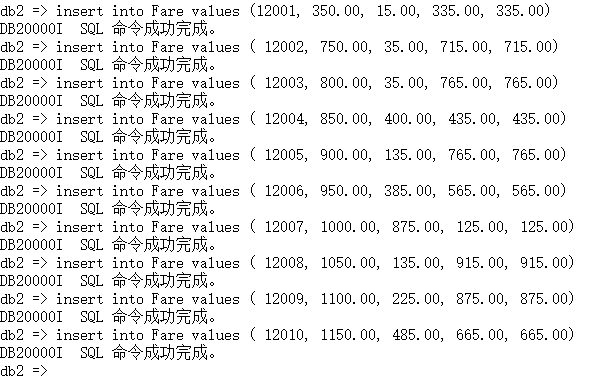
insert into Fare values ( 12006, 950.00, 385.00, 565.00, 565.00)

insert into Fare values ( 12007, 1000.00, 875.00, 125.00, 125.00)

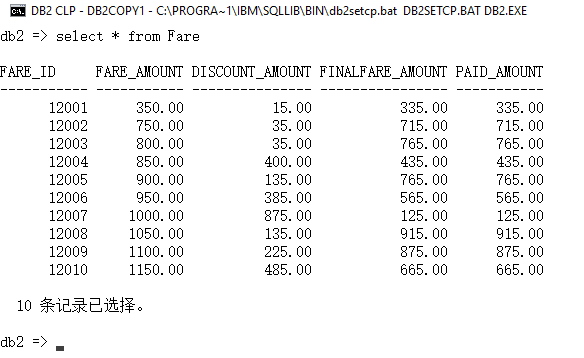
insert into Fare values ( 12008, 1050.00, 135.00, 915.00, 915.00)

insert into Fare values ( 12009, 1100.00, 225.00, 875.00, 875.00)

insert into Fare values ( 12010, 1150.00, 485.00, 665.00, 665.00)



select \* from Fare



## Passenger Table

insert into Passenger values (17082101, 'John', 'Harry', 001, 'Male', 'A0075551202')

insert into Passenger values (17082102, 'Mary', 'Swift', 002, 'Female', 'A0062351230')

insert into Passenger values (17082103, 'Timothy', 'Jonathan', 003, 'Male', 'A0062923745')

insert into Passenger values (17082104, 'Tom', 'Galydo', 004, 'Male', 'A0062675432')

insert into Passenger values (17082105, 'Lucy', 'Hong', 005, 'Female', 'A006251389')

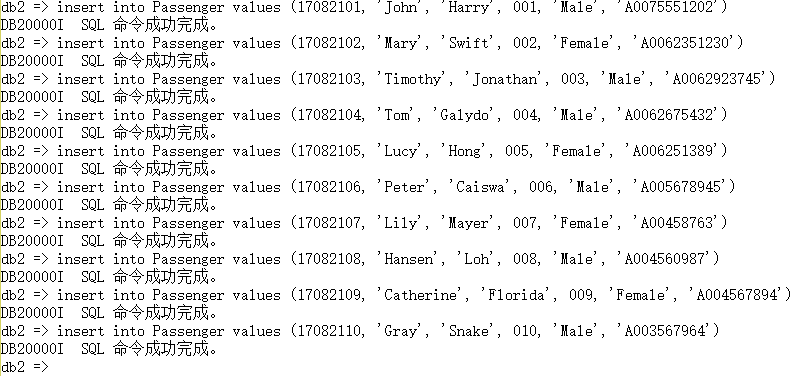
insert into Passenger values (17082106, 'Peter', 'Caiswa', 006, 'Male', 'A005678945')

insert into Passenger values (17082107, 'Lily', 'Mayer', 007, 'Female', 'A00458763')

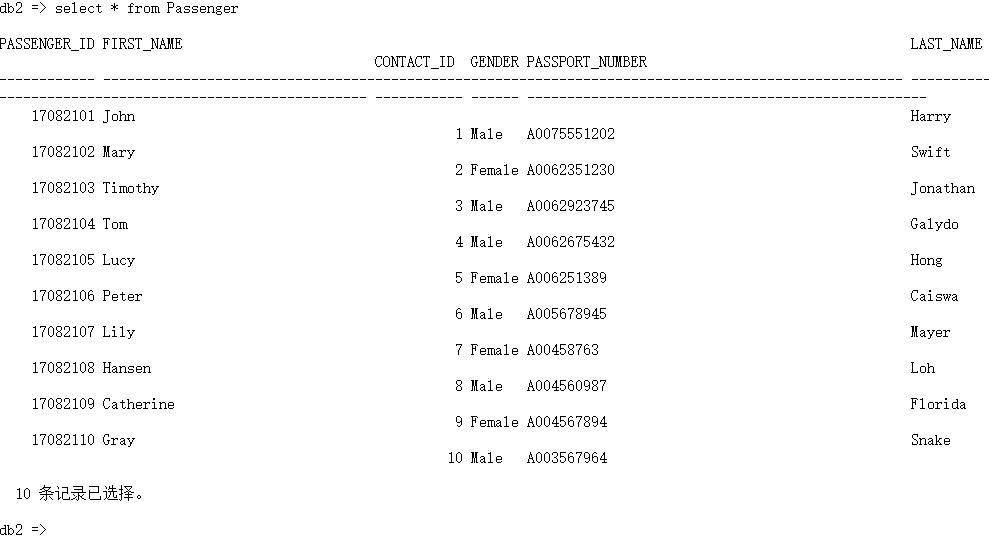
insert into Passenger values (17082108, 'Hansen', 'Loh', 008, 'Male', 'A004560987')

insert into Passenger values (17082109, 'Catherine', 'Florida', 009, 'Female', 'A004567894')

insert into Passenger values (17082110, 'Gray', 'Snake', 010, 'Male', 'A003567964')



select \* from Passenger



## Flight Table

insert into Flight values (35001, '2017-02-15','09:30:25',’2017-02-15’,'11:30:25', 'Kuala Lumpur', 'Kuching', 0.00, 1001 )

insert into Flight values (35002, '2017-02-17','10:35:27',’2017-02-27’,'13:35:25', 'Kota Kinabalu', 'Kuala Lumpur', 0.00, 1002 )

insert into Flight values (35003, '2017-02-19','07:40:45',’2017-02-19’,'9:30:25', 'Bintulu', 'Alor Setar', 0.00, 1003 )

insert into Flight values (35004, '2017-02-21','06:30:25',’2017-02-15’,'11:30:25', 'Melaka', 'Kuching', 0.00, 1004 )

insert into Flight values (35005, '2017-02-23','05:45:25',’2017-02-23’,'11:45:25', 'Johor Bahru', 'Pulau Pinang', 0.00, 1005 )

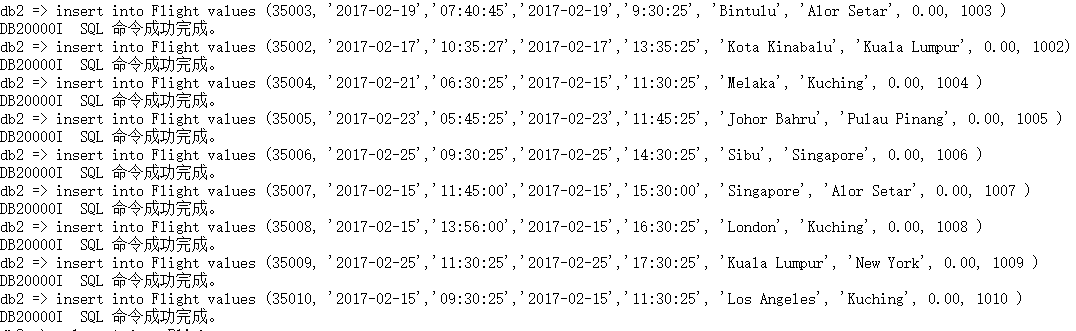
insert into Flight values (35006, '2017-02-25','09:30:25',’2017-02-25’,'14:30:25', 'Sibu', 'Singapore', 0.00, 1006 )

insert into Flight values (35007, '2017-02-15','11:45:00',’2017-02-15’,'15:30:00', 'Singapore', 'Alor Setar', 0.00, 1007 )

insert into Flight values (35008, '2017-02-15','13:56:00',’2017-02-15’,'16:30:25', 'London', 'Kuching', 0.00, 1008 )

insert into Flight values (35009, '2017-02-25','11:30:25',’2017-02-25’,'17:30:25', 'Kuala Lumpur', 'New York', 0.00, 1009 )

insert into Flight values (35010, '2017-02-15','09:30:25',’2017-02-15’,'11:30:25', 'Los Angeles', 'Kuching', 0.00, 1010 )



select \* from Flight



## Book Table

insert into Booking values (18001010, 17082101, 35001, 12001)

insert into Booking values (18001011, 17082102, 35002, 12002)

insert into Booking values (18001012, 17082103, 35003, 12003)

insert into Booking values (18001013, 17082104, 35004, 12004)

insert into Booking values (18001014, 17082105, 35005, 12005)

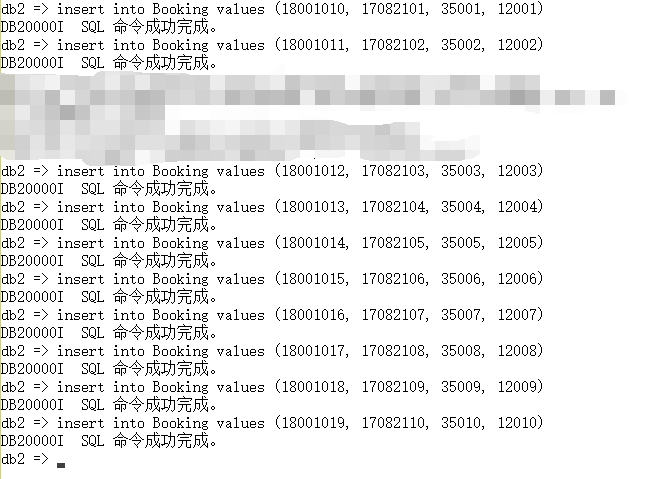
insert into Booking values (18001015, 17082106, 35006, 12006)

insert into Booking values (18001016, 17082107, 35007, 12007)

insert into Booking values (18001017, 17082108, 35008, 12008)

insert into Booking values (18001018, 17082109, 35009, 12009)

insert into Booking values (18001019, 17082110, 35010, 12010)



select \* from Booking

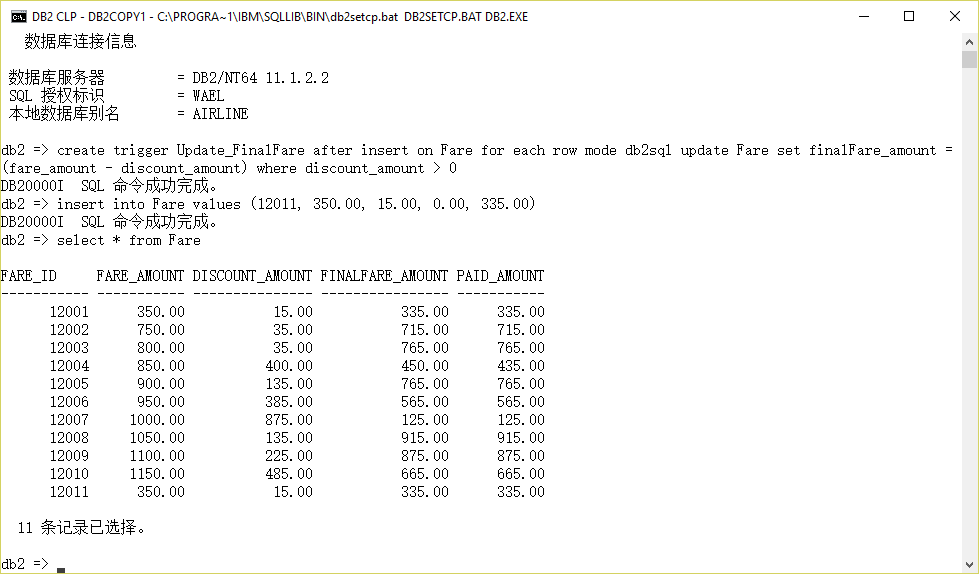


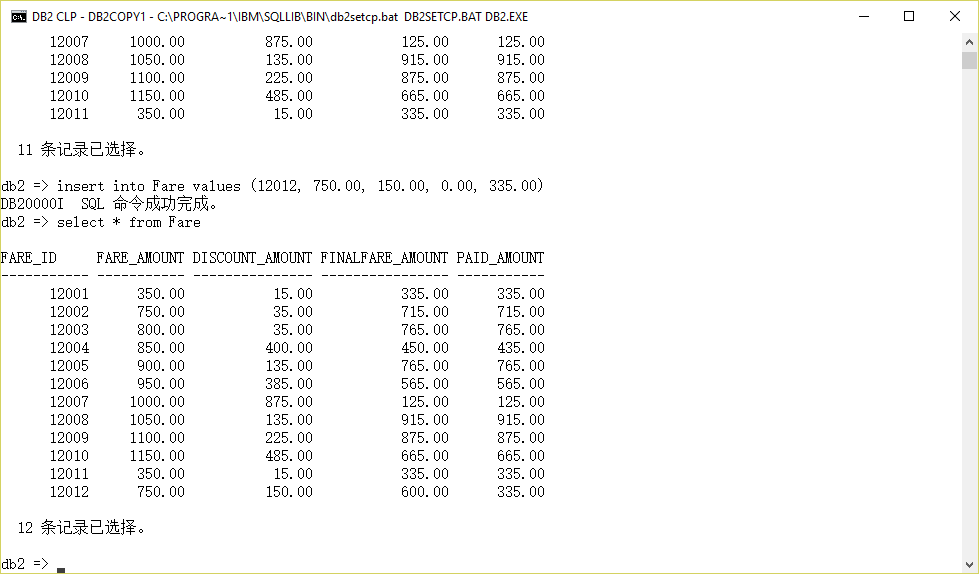
# Data Manipulation Language (DML):

## Triggers:

In the database, the Fare table includes fare\_amount, discount\_amount and finalFare\_amount. A trigger is used to automatically update all finalFare\_amount in the table according to amount and amount after every insertion of the Fare table.

create trigger Update\_FinalFare after insert on Fare for each row mode db2sql update Fare set finalFare\_amount = (fare\_amount - discount\_amount) where discount\_amount > 0

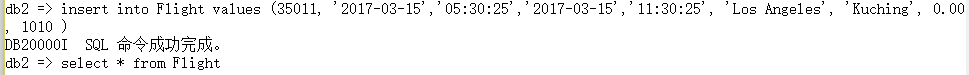


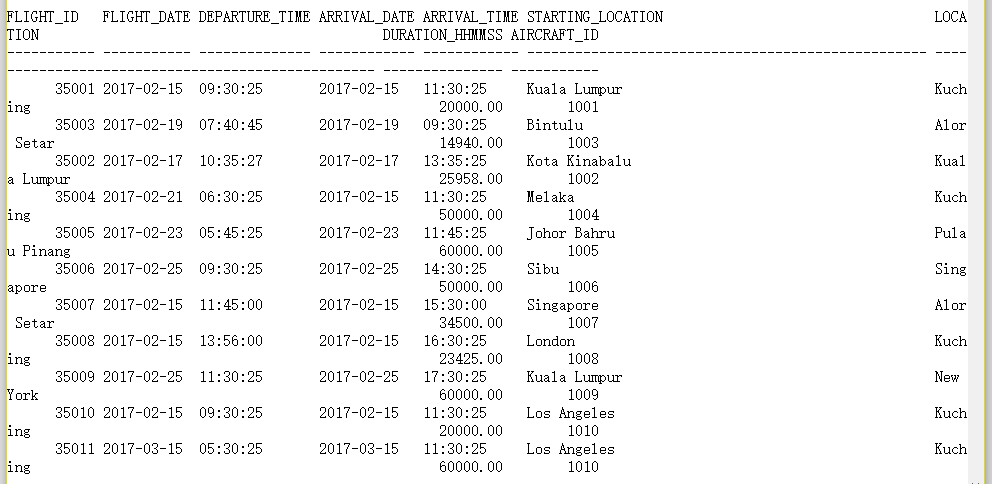


In the database, the Flight table includes departure\_time, arrival\_time and duration\_HHMMSS. A trigger is created to automatically calculate duration\_HHMMSS according to the departure\_time and arrival\_time every time an insertion is done on the Flight table.

create trigger Update\_FlightDuration after insert on Flight for each row mode db2sql update Flight set duration\_minutes = arrival\_time - departure\_time where exists (select arrival\_time, departure\_time from Flight)





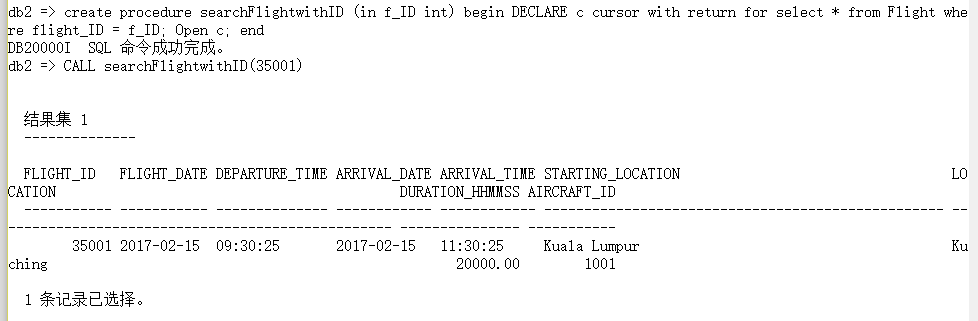


## Stored Procedure:

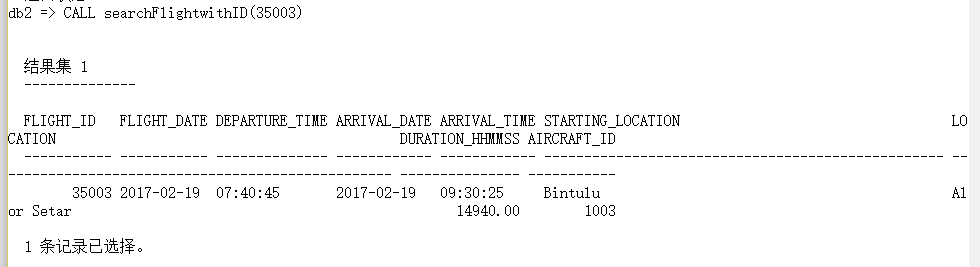
A stored procedure named searchFlightwithID is created to search the database on Flight table for a relevant Flight according to the provided Flight ID.

create procedure searchFlightwithID (in f\_ID int) begin DECLARE c cursor with return for select \* from Flight where flight\_ID = f\_ID; Open c; end

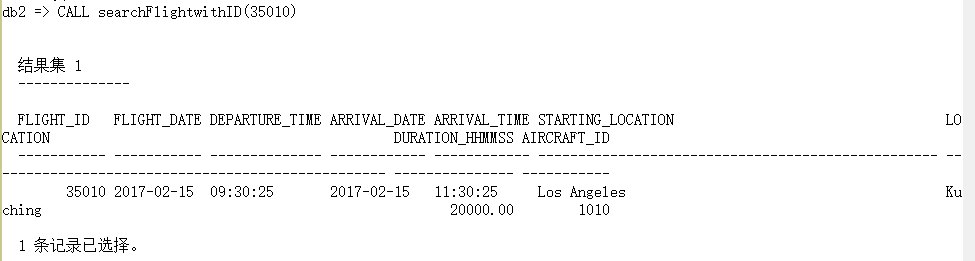
CALL searchFlightwithID(35001)



CALL searchFlightwithID(35003)



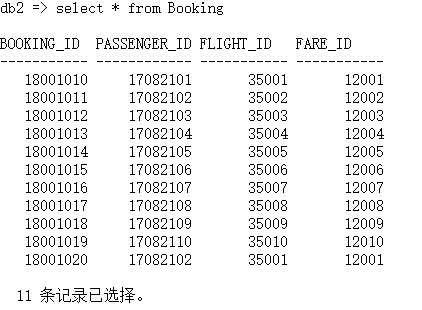
CALL searchFlightwithID(35010)

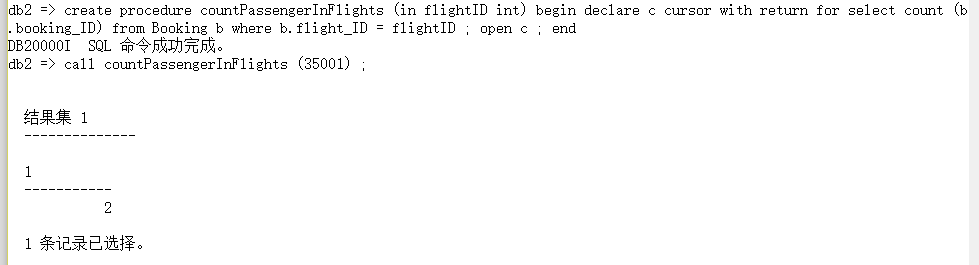


A stored procedure named countPassengerInFlights is created to count the number of passenger in a flight.

create procedure countPassengerInFlights (in flightID int) begin declare c cursor with return for select count (b.booking\_ID) from Booking b where b.flight\_ID = flightID ; open c ; end

call countPassengerInFlights (35001) ;

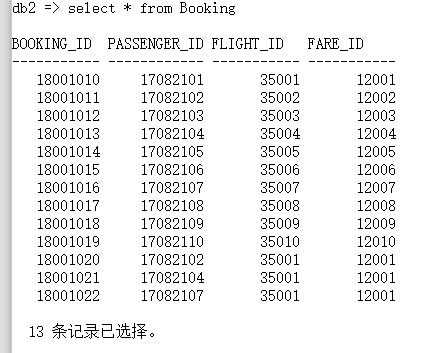


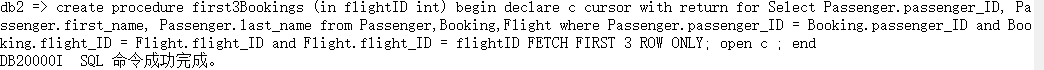


A stored procedure named first3Passengers is created to select the first 3 bookings made by passengers in a flight. This procedure can be useful when free gifts are provided for the first 3 passengers in a flight.

create procedure first3Bookings (in flightID int) begin declare c cursor with return for Select Passenger.passenger\_ID, Passenger.first\_name, Passenger.last\_name from Passenger,Booking,Flight where Passenger.passenger\_ID = Booking.passenger\_ID and Booking.flight\_ID = Flight.flight\_ID and Flight.flight\_ID = flightID FETCH FIRST 3 ROW ONLY; open c ; end

call first3Bookings (35001);



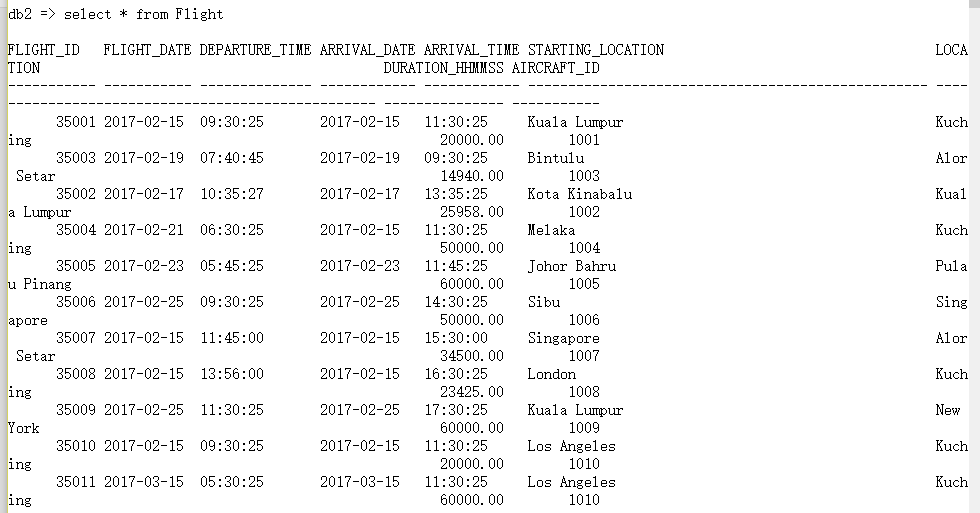


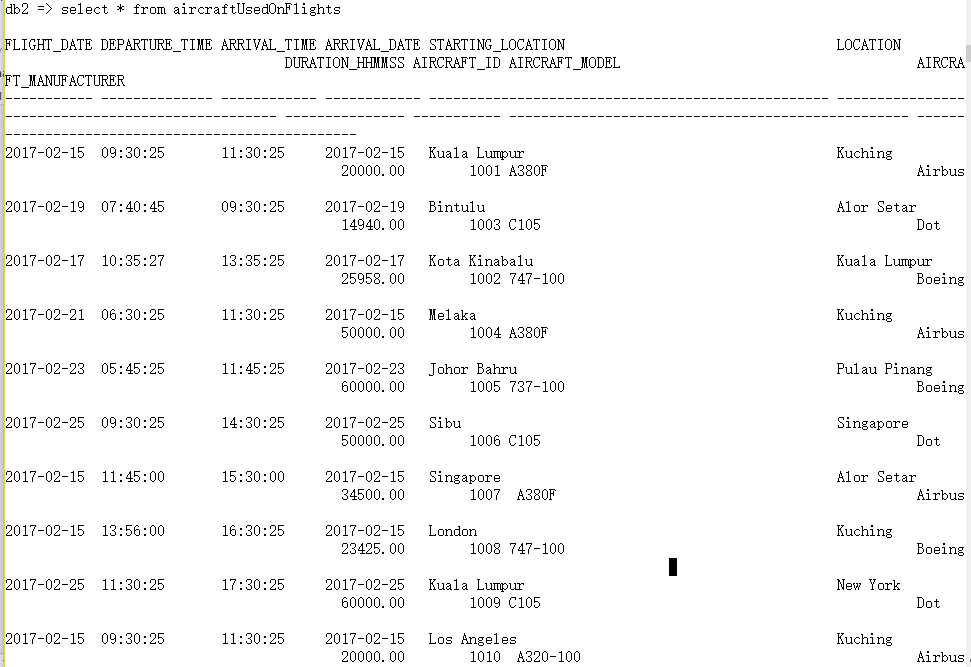


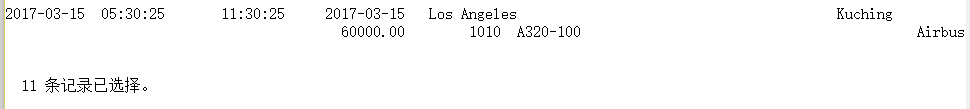
A view function named aircraftUsedOnFlights is created to display the aircraft used and other related details of all flights.

create view aircraftUsedOnFlights as select f.flight\_date, f.departure\_time, f.arrival\_time, f.arrival\_date, f.starting\_location, f.location, f.duration\_minutes, f.aircraft\_ID, a.aircraft\_model, a.aircraft\_manufacturer from Flight f, Aircraft a where f.aircraft\_ID = a.aircraft\_ID ;

select \* from aircraftUsedOnFlights ;







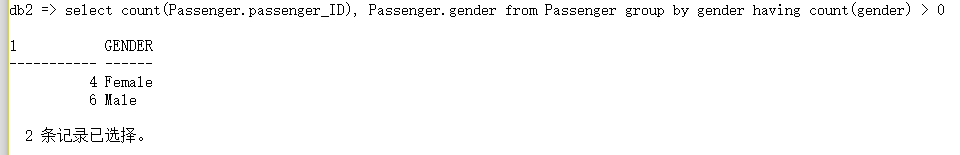
## Aggregate function :

For analysis purpose, a query is created to display the total number of passenger categorized by gender.

select count(Passenger.passenger\_ID), Passenger.gender from Passenger group by gender having count(gender) > 0



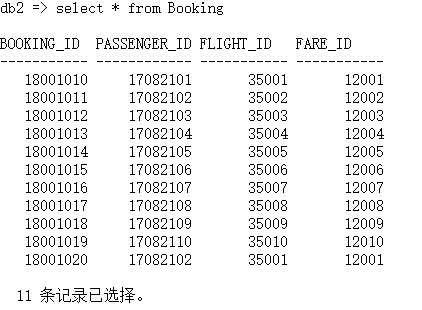


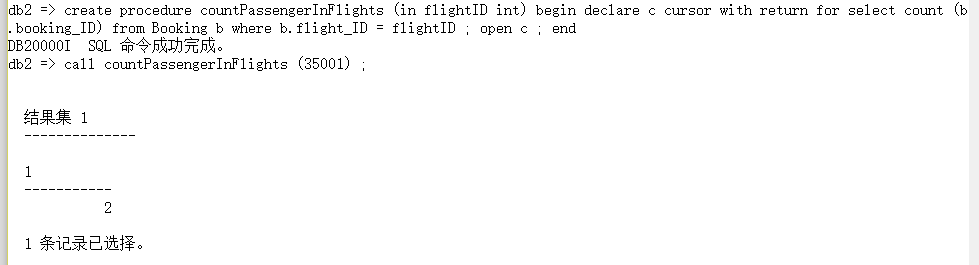


A Count function is used in the stored procedure, countPassengerInFlights to display the total number of passenger in a flight.

create procedure countPassengerInFlights (in flightID int) begin declare c cursor with return for select count (b.booking\_ID) from Booking b where b.flight\_ID = flightID ; open c ; end

call countPassengerInFlights (35001) ;





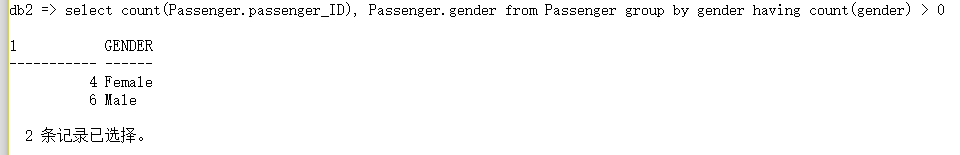
## Query with a Group by and Having clauses:

For analysis purpose, a query is created to display the total number of passenger categorized by gender.

select count(Passenger.passenger\_ID), Passenger.gender from Passenger group by gender having count(gender) > 0





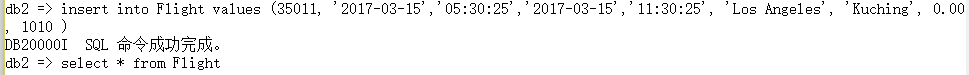


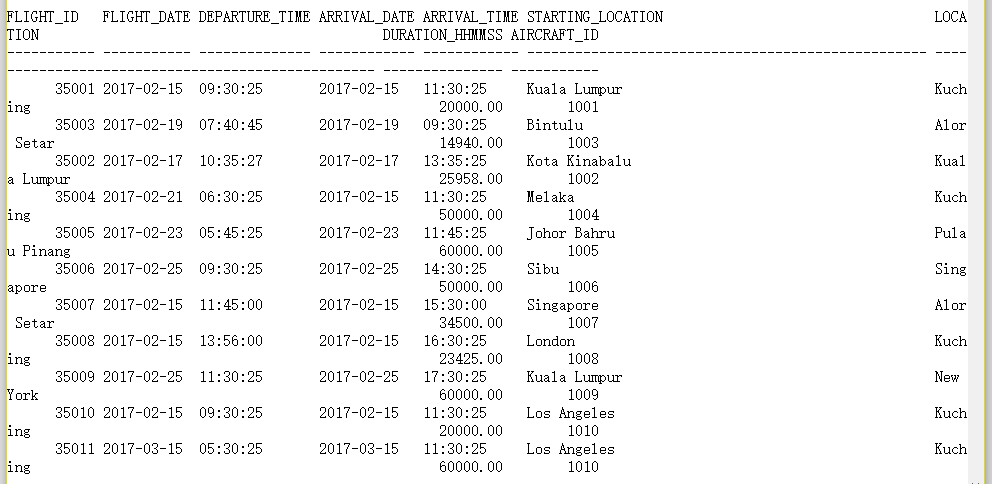
## Query with a Subquery:

In the database, the Flight table includes departure\_time, arrival\_time and duration\_HHMMSS. A trigger is created to automatically calculate duration\_HHMMSS according to the departure\_time and arrival\_time every time an insertion is done on the Flight table.

create trigger Update\_FlightDuration after insert on Flight for each row mode db2sql update Flight set duration\_minutes = arrival\_time - departure\_time where exists (select arrival\_time, departure\_time from Flight)



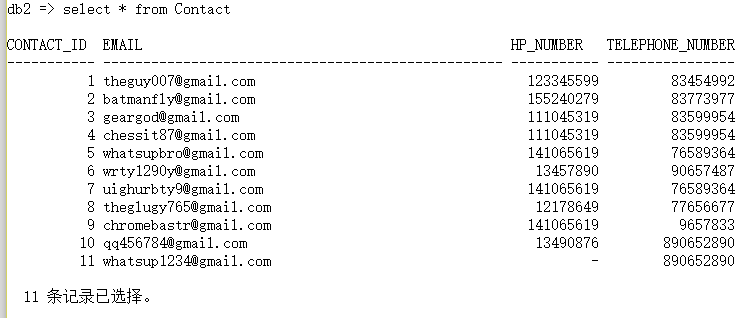


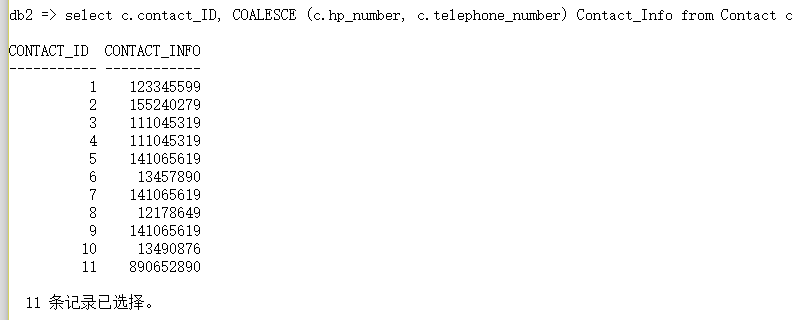


## Queries that are not covered in lecture:

A query to display the best contact method or information of all entries in the Contact table. For example, if mobile number is not available or null, telephone number is displayed and if both mobile number and telephone number are available, mobile number will be displayed.

select c.contact\_ID, COALESCE (c.hp\_number, c.telephone\_number) Contact\_Info from Contact c





A stored procedure named first3Passengers is created to select the first 3 bookings made by passengers in a flight. This procedure can be useful when free gifts are provided for the first 3 passengers in a flight.

create procedure first3Passengers (in flightID int) begin declare c cursor with return for Select Passenger.passenger\_ID, Passenger.first\_name, Passenger.last\_name from Passenger,Booking,Flight where Passenger.passenger\_ID = Booking.passenger\_ID and Booking.flight\_ID = Flight.flight\_ID and Flight.flight\_ID = flightID FETCH FIRST 3 ROW ONLY; open c ; end

call first3Passengers (35001);

