*COMS2002***Database Fundamentals II - Assignment**

<NAME> (<STUDENT NUMBER>)

15/03/2013

Table of Contents

[Question 3 2](#_Toc351118585)

[Question 4 4](#_Toc351118586)

[Question 5 7](#_Toc351118587)

[Query 7](#_Toc351118588)

[Description 7](#_Toc351118589)

[Question 6 7](#_Toc351118590)

[Query 7](#_Toc351118591)

[Description 7](#_Toc351118592)

[Question 7 8](#_Toc351118593)

[Query 8](#_Toc351118594)

[Description 8](#_Toc351118595)

[Question 8 9](#_Toc351118596)

[Query 9](#_Toc351118597)

[Description 9](#_Toc351118598)

[Question 9 9](#_Toc351118599)

[Query 9](#_Toc351118600)

[Description 10](#_Toc351118601)

[Question 10 10](#_Toc351118602)

[Query 10](#_Toc351118603)

[Description 10](#_Toc351118604)

[Question 11 11](#_Toc351118605)

[Query 11](#_Toc351118606)

[Description 11](#_Toc351118607)

[Question 12 12](#_Toc351118608)

[Query 12](#_Toc351118609)

[Description 12](#_Toc351118610)

[Question 13 12](#_Toc351118611)

[Query 12](#_Toc351118612)

[Description 13](#_Toc351118613)

[Question 14 13](#_Toc351118614)

[Query 13](#_Toc351118615)

[Description 14](#_Toc351118616)

[Question 15 14](#_Toc351118617)

[Query 14](#_Toc351118618)

[Description 14](#_Toc351118619)

[Question 16 15](#_Toc351118620)

[Query 15](#_Toc351118621)

[Description 15](#_Toc351118622)

[Question 17 15](#_Toc351118623)

[Query 15](#_Toc351118624)

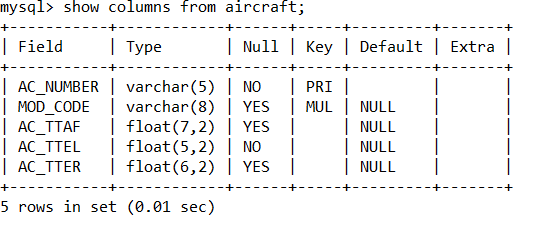
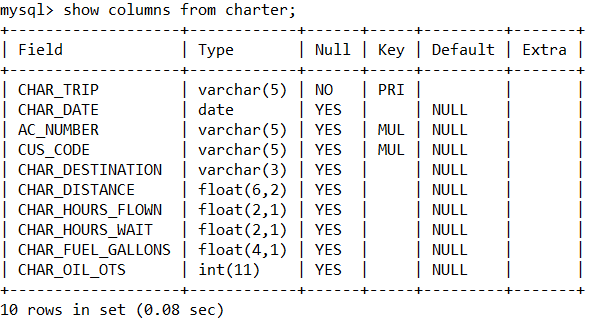
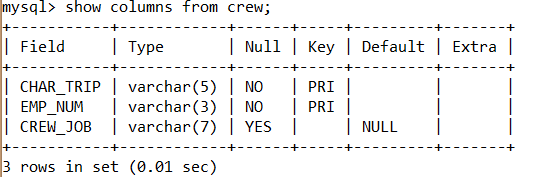
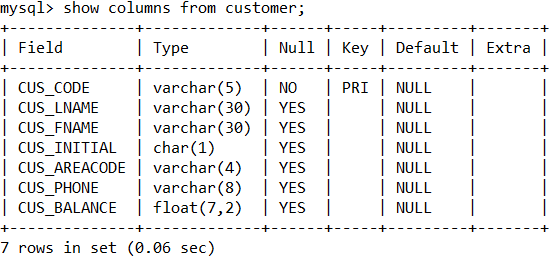
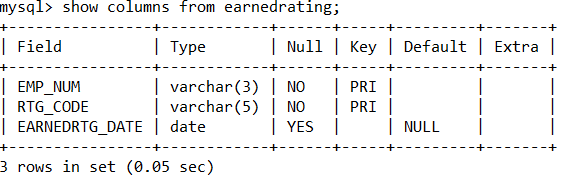
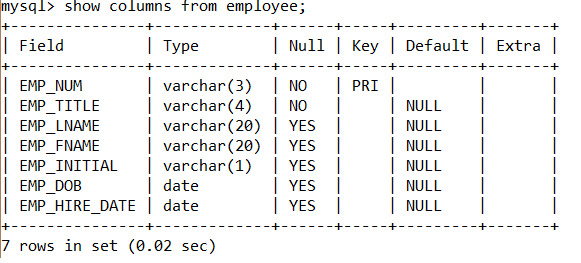
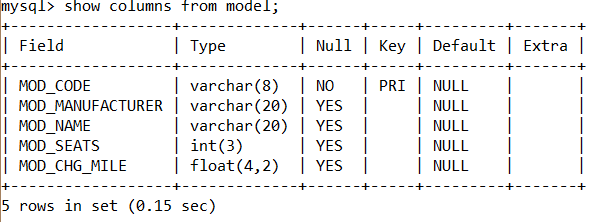
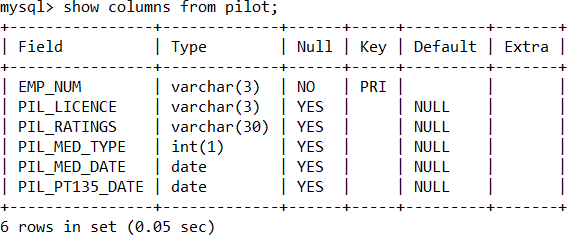
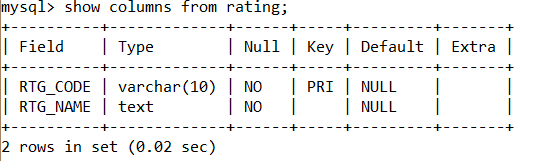
[Description 16](#_Toc351118625)

[Question 18 17](#_Toc351118626)

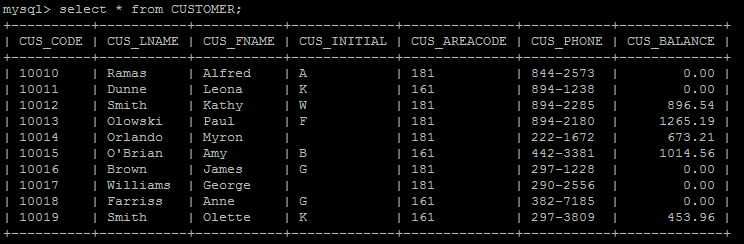
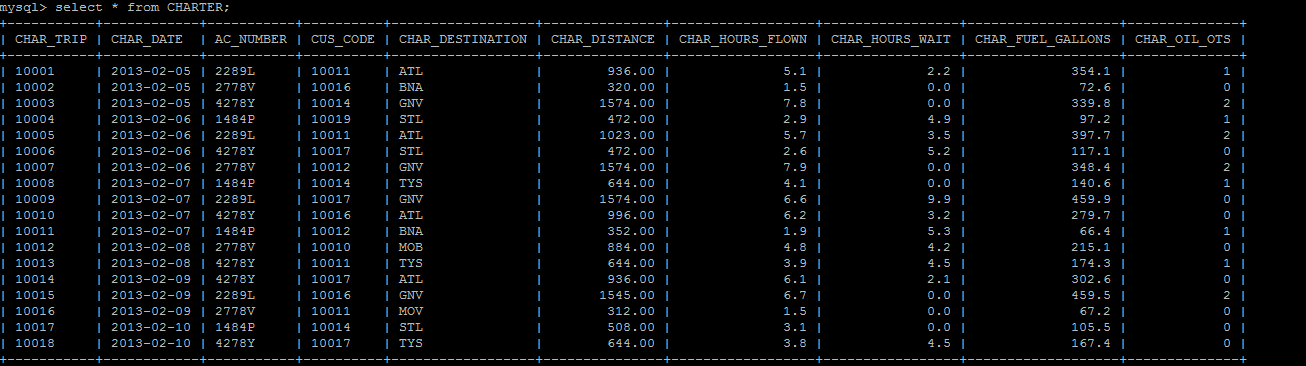
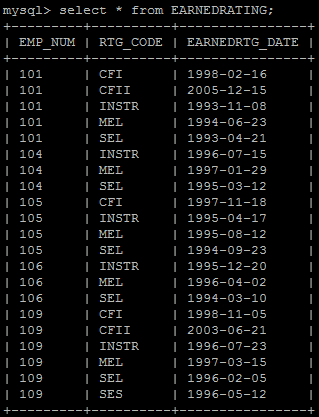
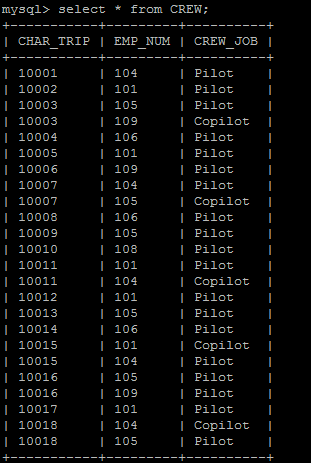
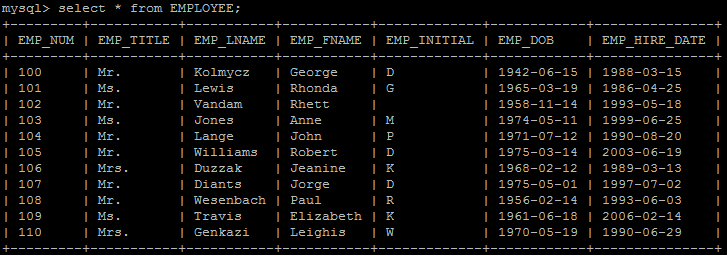
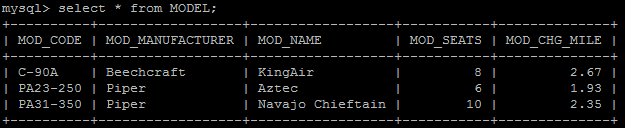
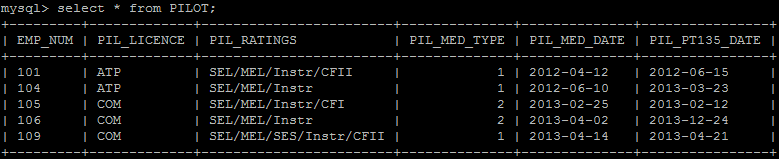
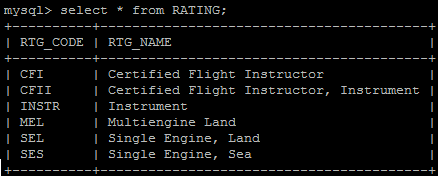
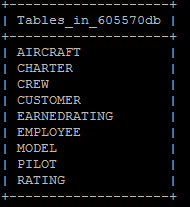
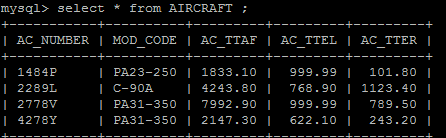
[Query 17](#_Toc351118627)

[Description 17](#_Toc351118628)

# Question 3



# Question 4



# Question 5

## Query

SELECT CHAR\_TRIP,

CHAR\_DATE,

AC\_NUMBER,

CHAR\_DESTINATION,

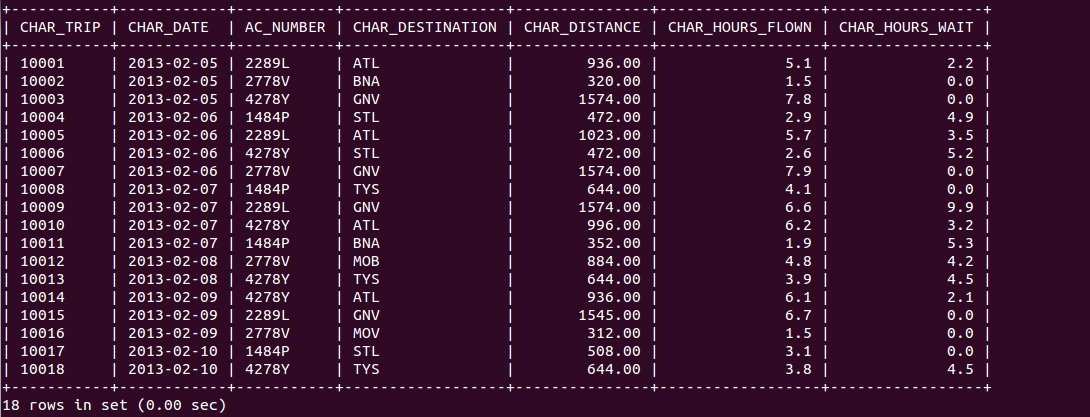
CHAR\_DISTANCE,

CHAR\_HOURS\_FLOWN,

CHAR\_HOURS\_WAIT

FROM CHARTER;

## Description



This SQL query will return the first seven attributes from the table know as CHARTER, and it will display all know data from the table Charter for the seven attributes.

# Question 6

## Query

SELECT Date\_format(CHAR\_DATE, '%d-%b-%y') AS CHAR\_DATE,

AC\_NUMBER,

CHAR\_DESTINATION,

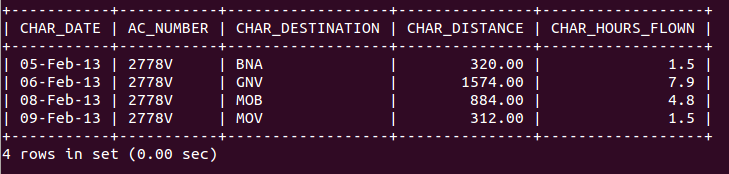
CHAR\_DISTANCE,

CHAR\_HOURS\_FLOWN

FROM CHARTER

WHERE AC\_NUMBER = '2778V';

## Description



This simple query displays the requested fields from a single table (Charter). The conditions for the query are that AC\_NUMBER=2778V. The WHERE clause is used to specify this.

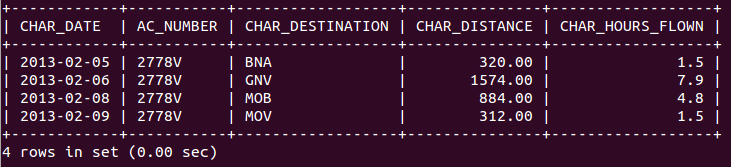
# Question 7

## Query

CREATE VIEW AC\_2778V AS SELECT CHAR\_DATE, AC\_NUMBER, CHAR\_DESTINATION,

CHAR\_DISTANCE, CHAR\_HOURS\_FLOWN FROM CHARTER WHERE AC\_NUMBER = '2778V';

## Description



This SQL query will create a view, a view is a virtual table, it is a data object which does not contain any data. The difference between a view and a table is that views are definitions built on top of other tables. If data is changed in the underlying table, the same change is reflected in the view. This view is created with the name 'AC\_2778V' and contains the specified requested attributes from the table named CHARTER providing the condition AC\_NUMBER = '2778V' is met.

# Question 8

## Query

SELECT CHAR\_DATE,

CHARTER.AC\_NUMBER,

CHAR\_DESTINATION,

CUS\_LNAME,

CUS\_AREACODE,

CUS\_PHONE

FROM CHARTER

INNER JOIN CUSTOMER

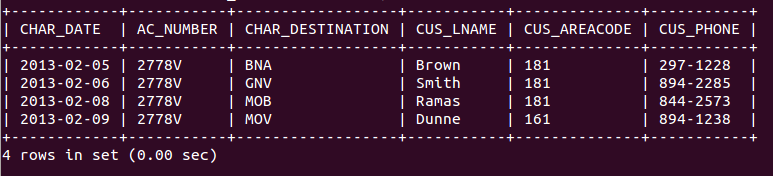
ON CHARTER.CUS\_CODE = CUSTOMER.CUS\_CODE

INNER JOIN AIRCRAFT

ON CHARTER.AC\_NUMBER = AIRCRAFT.AC\_NUMBER

WHERE AIRCRAFT.AC\_NUMBER = '2778V';

## Description



This SQL query selects and displays the specific attributes as requested for all the records that have an Aircraft Number of “2778V” and manages relations using the inner join function.

# Question 9

## Query

SELECT Date\_format(CHAR\_DATE, '%d-%b-%y') AS CHAR\_DATE,

CHAR\_DESTINATION,

CHARTER.AC\_NUMBER,

MOD\_NAME,

MOD\_CHG\_MILE

FROM CHARTER

INNER JOIN AIRCRAFT

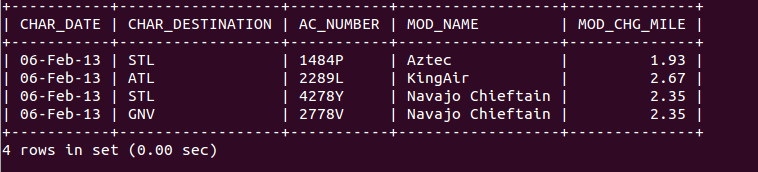
ON CHARTER.AC\_NUMBER = AIRCRAFT.AC\_NUMBER

INNER JOIN MODEL

ON AIRCRAFT.MOD\_CODE = MODEL.MOD\_CODE

WHERE CHAR\_DATE = '2013-02-06';

## Description



This SQL query returns the required fields from the CHARTER and MODEL tables whilst using the relationship between CHARTER, AIRCRAFT and MODEL tables; this is done so using an INNER JOIN statement matching AC\_NUMBER from CHARTER to AIRCRAFT and MOD\_CODE from AIRCRAFT to MODEL.

# Question 10

## Query

SELECT Date\_format(CHAR\_DATE, '%d-%b-%y') AS CHAR\_DATE,

CHAR\_DESTINATION,

CHARTER.AC\_NUMBER,

MOD\_NAME,

MOD\_CHG\_MILE,

CUS\_LNAME

FROM CHARTER

INNER JOIN CUSTOMER

ON CHARTER.CUS\_CODE = CUSTOMER.CUS\_CODE

INNER JOIN AIRCRAFT

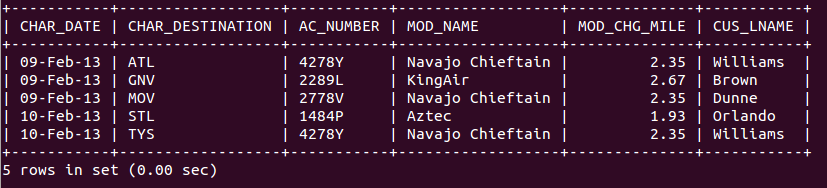
ON CHARTER.AC\_NUMBER = AIRCRAFT.AC\_NUMBER

INNER JOIN MODEL

ON AIRCRAFT.MOD\_CODE = MODEL.MOD\_CODE

WHERE CHAR\_DATE >= '2013-02-09';

## Description



This query does the same as Question 9, however it restricts records to those created since 9 February by making sure that CHAR\_DATE is greater than or equal to this date. It also requires an additional inner join to relate the customer codes in the CHARTER table to those found in the CUSTOMER table using the CUS\_CODE field.

# Question 11

## Query

SELECT Date\_format(CHAR\_DATE, '%d-%b-%y') AS CHAR\_DATE,

CHAR\_DESTINATION,

CHARTER.AC\_NUMBER,

MOD\_CHG\_MILE,

CHAR\_DISTANCE,

EMPLOYEE.EMP\_NUM,

CREW\_JOB,

EMP\_LNAME

FROM CHARTER

INNER JOIN CREW

ON CHARTER.CHAR\_TRIP = CREW.CHAR\_TRIP

INNER JOIN EMPLOYEE

ON CREW.EMP\_NUM = EMPLOYEE.EMP\_NUM

INNER JOIN AIRCRAFT

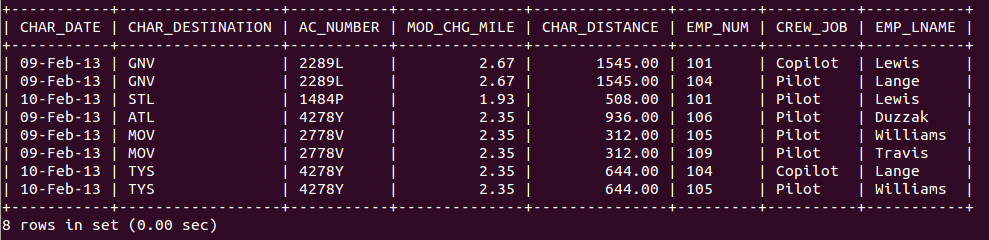
ON CHARTER.AC\_NUMBER = AIRCRAFT.AC\_NUMBER

INNER JOIN MODEL

ON AIRCRAFT.MOD\_CODE = MODEL.MOD\_CODE

WHERE CHAR\_DATE >= '2013-02-09';

## Description



This SQL query displays the requested attributes from the different tables that were generated after 9 February 2013 and manages relations using the inner join function.

# Question 12

## Query

SELECT Date\_format(CHAR\_DATE, '%d-%b-%y') AS CHAR\_DATE,

CHARTER.AC\_NUMBER,

MOD\_NAME,

CHAR\_HOURS\_FLOWN,

CHAR\_FUEL\_GALLONS,

( CHAR\_FUEL\_GALLONS / CHAR\_HOURS\_FLOWN ) AS FuelperGallon

FROM CHARTER

INNER JOIN AIRCRAFT

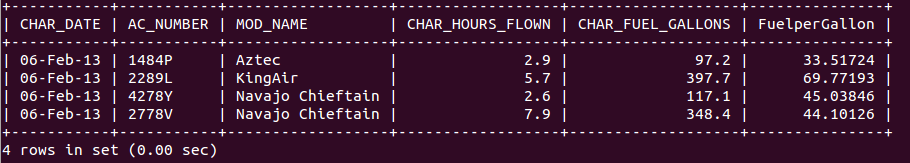
ON CHARTER.AC\_NUMBER = AIRCRAFT.AC\_NUMBER

INNER JOIN MODEL

ON AIRCRAFT.MOD\_CODE = MODEL.MOD\_CODE

WHERE CHAR\_DATE = '2013-02-06';

## Description



This query builds on much from Question 9, including constraints but has been slightly modified to return a calculated field, 'fuel per hour' (FuelperGallon) by dividing CHAR\_FUEL\_GALLONS by CHAR\_HOURS\_FLOWN. Contrary to Question 9, MOD\_CHG\_MILE has not been returned, whereas CHAR\_FUEL\_GALLONS and CHAR\_HOURS\_FLOWN has been returned.

# Question 13

## Query

SELECT CHAR\_DATE,

CUS\_LNAME,

CHAR\_DISTANCE,

MOD\_CHG\_MILE,

( MOD\_CHG\_MILE \* CHAR\_DISTANCE ) AS MileageCharge

FROM CUSTOMER,

MODEL,

CHARTER,

AIRCRAFT

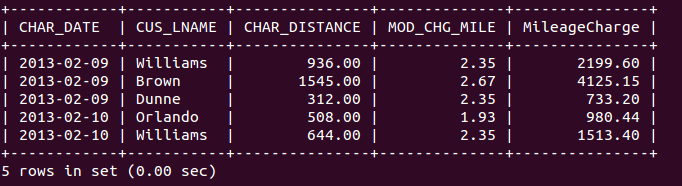
WHERE CHARTER.AC\_NUMBER = AIRCRAFT.AC\_NUMBER

AND AIRCRAFT.MOD\_CODE = MODEL.MOD\_CODE

AND CHARTER.CUS\_CODE = CUSTOMER.CUS\_CODE

AND CHAR\_DATE >= '2013-02-09';

## Description



This SQL query selects 4 fields from the tables CHARTER and MODEL with the constraint that CHAR\_DATE be after the 9th of Feb. It also calculates a field, MileageCharge, which is the product of MOD\_CHG\_MILE and CHAR\_DISTANCE and joins tables AIRCRAFT, MODEL, CHARTER and CUSTOMER by equating primary keys to foreign keys.

# Question 14

## Query

SELECT CHAR\_DATE,

CUS\_LNAME,

( MOD\_CHG\_MILE \* CHAR\_DISTANCE ) AS

MileageCharge,

( CHAR\_HOURS\_WAIT \* 50 ) AS

WaitingCharge,

( ( MOD\_CHG\_MILE \* CHAR\_DISTANCE ) + ( CHAR\_HOURS\_WAIT \* 50 ) ) AS

TotalCharge

FROM CUSTOMER,

MODEL,

CHARTER,

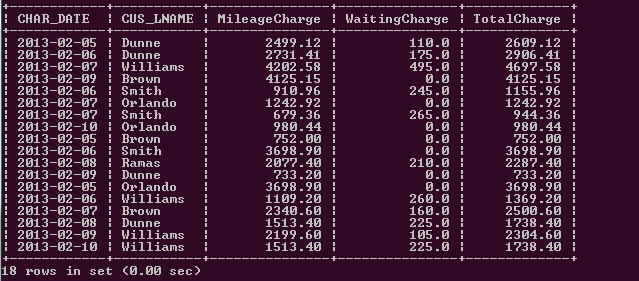
AIRCRAFT

WHERE CHARTER.AC\_NUMBER = AIRCRAFT.AC\_NUMBER

AND AIRCRAFT.MOD\_CODE = MODEL.MOD\_CODE

AND CHARTER.CUS\_CODE = CUSTOMER.CUS\_CODE;

## Description



This SQL query selects 2 fields from tables CHARTER and CUSTOMER. It also calculates 3 fields, MileageCharge, WaitingCharge and TotalCharge where MileageCharge is the product of MOD\_CHG\_MILE and CHAR\_DISTANCE, WaitingCharge is the CHAR\_HOURS\_WAIT multiplied by 50 and TotalCharge is the sum of MileageCharge and WaitingCharge. This query also joins tables CHARTER, AIRCRAFT, MODEL and CUSTOMER by equating primary keys to foreign keys.

# Question 15

## Query

SELECT CUS\_LNAME,

CUS\_FNAME,

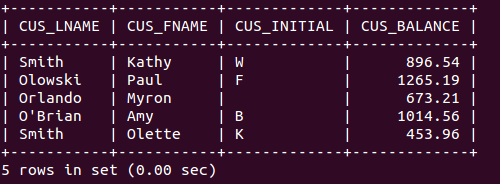
CUS\_INITIAL,

CUS\_BALANCE

FROM CUSTOMER

WHERE CUS\_BALANCE > 0;

## Description



This simple query displays the requested field values from the CUSTOMER table if the balance of the customer (CUS\_BALANCE) is not settled i.e. CUS\_BALANCE>0.

# Question 16

## Query

SELECT Min(CUS\_BALANCE) AS MIN\_BALANCE,

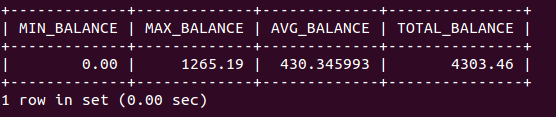
Max(CUS\_BALANCE) AS MAX\_BALANCE,

Avg(CUS\_BALANCE) AS AVG\_BALANCE,

Sum(CUS\_BALANCE) AS TOTAL\_BALANCE

FROM CUSTOMER;

## Description



This SQL query displays the maximum, minimum, average and total balance from the CUSTOMER table using the following commands MAX(), MIN(), AVG() and SUM() respectively; and gives them their own heading using the “AS” command.

# Question 17

## Query

SELECT CHARTER.CHAR\_TRIP,

Date\_format(CHAR\_DATE, '%d-%b-%y'),

CHARTER.AC\_NUMBER,

MOD\_NAME,

CHAR\_HOURS\_FLOWN,

EMP\_LNAME,

CREW\_JOB

FROM CHARTER

INNER JOIN AIRCRAFT

ON CHARTER.AC\_NUMBER = AIRCRAFT.AC\_NUMBER

INNER JOIN MODEL

ON AIRCRAFT.MOD\_CODE = MODEL.MOD\_CODE

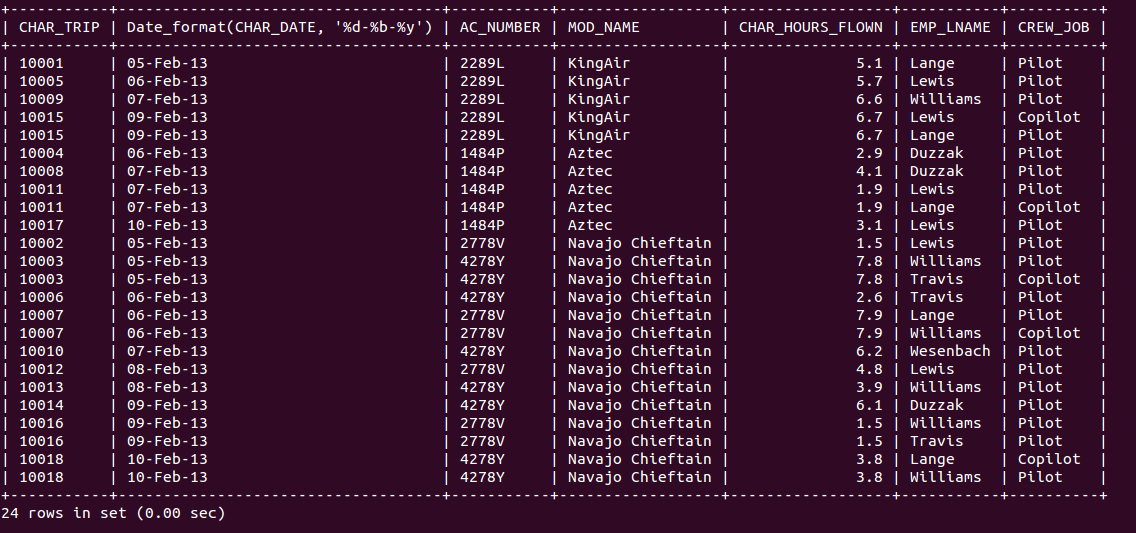
INNER JOIN CREW

ON CHARTER.CHAR\_TRIP = CREW.CHAR\_TRIP

INNER JOIN EMPLOYEE

ON CREW.EMP\_NUM = EMPLOYEE.EMP\_NUM;

## Description



This SQL query will return information from 3 tables, namely: CHARTER, MODEL and EMPLOYEE by linking this tables using primary/foreign keys within those tables as well as from tables: AIRCRAFT and CREW. There are no specified constraints on the resultant data, so no 'WHERE' statement was used.

# Question 18

## Query

SELECT EMP\_NUM,

EMP\_LNAME,

EMP\_FNAME,

EMP\_HIRE\_DATE,

EMP\_DOB,

Curdate() AS

QueryDate,

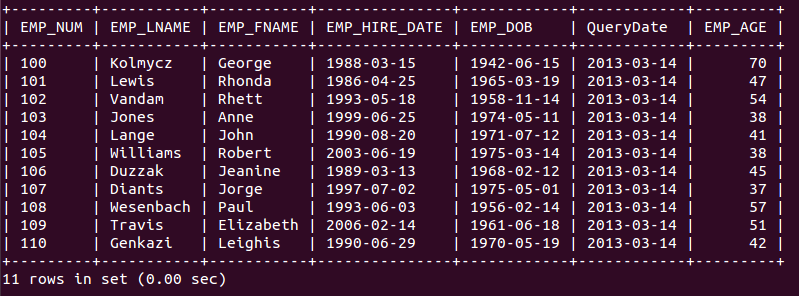
Date\_format(Now(), '%Y') - Date\_format(EMP\_DOB, '%Y') - (

Date\_format(Now(), '00-%m-%d') < Date\_format(EMP\_DOB, '00-%m-%d') ) AS

EMP\_AGE

FROM EMPLOYEE;

## Description



This SQL query selects and displays the specific attributes as requested, but adds onto those attributes the attribute QueryDate, which is just the date that the query was run on. Also added to these attributes is an attribute name EMP\_AGE, which calculates the age of each employee by subtracting the current year from the given employee date of birth, and then checks if the current month and day is less than the birth month and day, and if this condition is true, it subtracts one from the initial age calculations regarding year because it hasn’t been that persons birthday in the current year yet.