Lab 4 (2%)

Stored Procedures

topics

* Stored Procedures: <https://docs.microsoft.com/en-us/sql/relational-databases/stored-procedures/create-a-stored-procedure?view=sql-server-ver15>
* variables: <https://docs.microsoft.com/en-us/sql/t-sql/language-elements/variables-transact-sql?view=sql-server-ver15>
* sql case when statement: <https://docs.microsoft.com/en-us/sql/t-sql/language-elements/case-transact-sql?view=sql-server-ver15>
* if else: <https://docs.microsoft.com/en-us/sql/t-sql/language-elements/if-else-transact-sql?view=sql-server-ver15>
* Control of flow: <https://docs.microsoft.com/en-us/sql/t-sql/language-elements/control-of-flow?view=sql-server-ver15>
* Stored procedures, return : <https://docs.microsoft.com/en-us/sql/relational-databases/stored-procedures/return-data-from-a-stored-procedure?view=sql-server-ver15>

Group work acknowledgment

We, Mansoor Ahmad Zafar, declare that the attached assignment is our own work in accordance with the Seneca Academic Policy. No part of this assignment has been copied manually or electronically from any other source (including web sites) **or distributed to other students.**

Specify below what each member has done towards the completion of this work:

Name Task(s)

1- Mansoor Ahmad Zafar Everything

2-

3-

Before you start

You are to create a new database named “AviaCo” and run the sql script you are given to create the tables in the database.

Instructions

Answer each of the following questions and show the result set underneath each part.

1. Write a stored procedure named displayCustomers\_grpX that shows customers with outstanding balances (balance > 0).

Testing and result set:

create procedure displayCustomers\_grp11

as

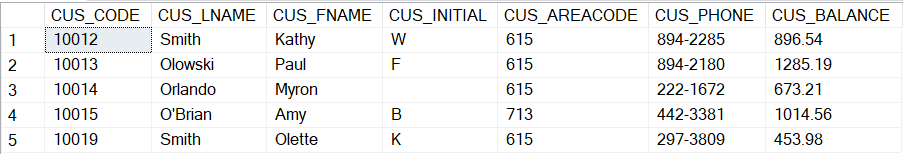
select \*

from customer c

where c.cus\_balance > 0

go

exec displayCustomers\_grp11;



1. Write a stored procedure named displayCustomers\_areacode\_grpX that takes the area code as parameter and shows the customers in the given area code that have outstanding balance.

Testing and result set:

alter procedure displayCustomers\_areacode\_grp11

@area\_code varchar(3)

as

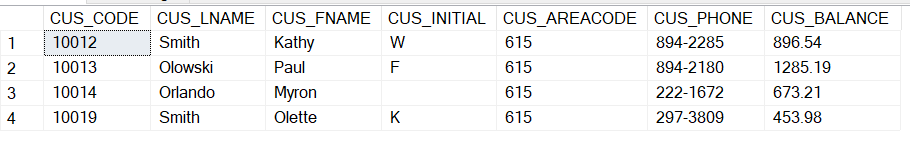
select \*

from customer c

where CUS\_AREACODE = @area\_code and c.cus\_balance > 0

go

exec displayCustomers\_areacode\_grp11 '615';



1. Write a stored procedure named displayPilots\_grpX that takes a rating code and year as input and shows the pilot emp num, Lname, fname, rtg\_code, earnrtg\_date who earned the rating given in input at the specified year given in input.

Testing and result set:

alter procedure displayPilots\_grp11

@rCode varchar(6), @year datetime

as

select p.EMP\_NUM, e.EMP\_FNAME, e.EMP\_LNAME, er.RTG\_CODE, er.EARNRTG\_DATE

from PILOT p

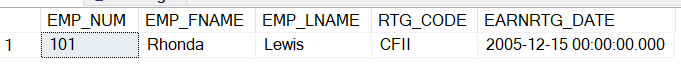
join EMPLOYEE e on e.EMP\_NUM = p.EMP\_NUM

join EARNEDRATING eR on er.EMP\_NUM = p.EMP\_NUM

where year(er.EARNRTG\_DATE) = @year and er.RTG\_CODE = @rCode

go

exec displayPilots\_grp11 'CFII', 2005;



1. Write a stored procedure named tagCharters\_grpX that takes the number of hours flown @nbhours as input and label each of the charters with 'short' if the number of hours flown is less than @nbhours and 'long' otherwise.

Write the statement to execute the stored procedure by passing 5 as argument to @nbhours.

Testing and result set: top 5 of 18 rows

create procedure tagCharters\_grp11

@nbhours int

as

select top 5 c.CHAR\_TRIP, 'flag' =

case

when @nbhours > c.CHAR\_HOURS\_FLOWN then 'short'

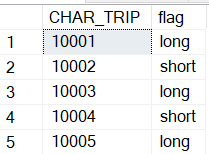
else 'long'

end

from charter c

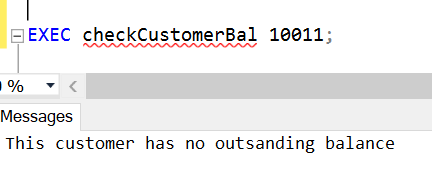
go

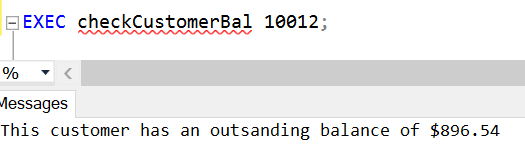
EXEC tagCharters\_grp11 @nbhours=5;



1. Write a stored procedure named checkCustomerBal\_grpX that takes a customer code @cus\_code as input and prints a message (not a table) of whether the customer has an outstanding balance along with the amount or no outstanding balance.  
   Write the statement to execute the stored procedure by testing with 10011 and 10012 as argument.

Result set:





create procedure checkCustomerBal\_grp11

@cus\_code int

as

declare @bal float

set @bal= (select c.CUS\_BALANCE from CUSTOMER c where c.CUS\_CODE = @cus\_code)

if @bal <= 0

print 'This customer has no outstanding balance'

else

print concat('This customer has an outstanding balance of ', @bal)

go

exec checkCustomerBal\_grp11 10011

exec checkCustomerBal\_grp11 10012

1. Write a stored procedure named getnbChartersCus\_grpX that takes a customer code as input and returns the number of charters the customer has booked.

Use the return statement so the stored procedure returns a value.

If the customer code is not existing in the customer table, print a message 'invalid customer code' and return -1.

For testing, run the code below. @count gets assigned the return value out of the stored procedure.

create procedure getnbChartersCus\_grp11

@cusCode int

as

declare @count int

select @count=count(\*)

from charter c

where c.CUS\_CODE = @cusCode

if @cusCode in (select CUS\_CODE from CHARTER)

RETURN @count

ELSE

print 'invalid customer code'

return -1

go

DECLARE @count int;

exec @count= getnbChartersCus\_grp11 10011;

print @count;

Result set:



1. Write a stored procedure named checkConsumptionv1\_grpX that takes no input and show the charter trip, destination, fuel consumption (CHAR\_FUEL\_GALLONS) ,

the average fuel consumption of charters of the same destination, and an additional column named 'status' with the values:

'consumed more than the average fuel consumption for that destination'

or 'consumed less than the average fuel consumption for that destination'.

Testing and result set: top 5 out of 18 rows

create procedure checkConsumptionv1\_grp11

as

select top 5 c.CHAR\_TRIP, c.CHAR\_DESTINATION, c.CHAR\_FUEL\_GALLONS,

(select avg(CHAR\_FUEL\_GALLONS) from charter where CHAR\_DESTINATION = c.CHAR\_DESTINATION) as avgFuelConsDes,

'status' =

case

when

(select avg(CHAR\_FUEL\_GALLONS) from charter where CHAR\_DESTINATION = c.CHAR\_DESTINATION)

< c.CHAR\_FUEL\_GALLONS

then 'This charter has consumed more than the average fuel consumption for that destination'

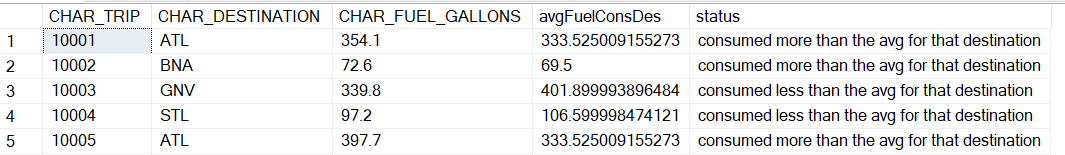
else 'This charter has consumed less than the average fuel consumption for that destination'

end

from charter c

go

exec checkConsumptionv1\_grp11;

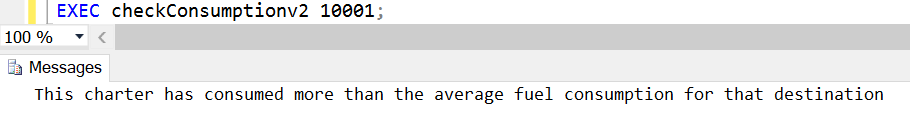


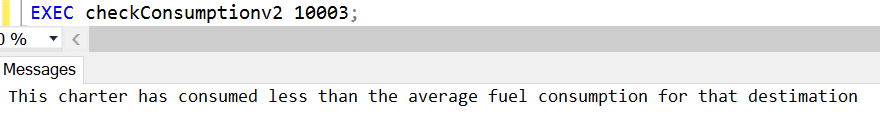
1. Write a stored procedure named checkConsumptionv2\_grpX that takes a charter trip code as input, and output a message saying :

"This charter has consumed more than the average fuel consumption compared to other charters flying to the same destination" or less otherwise.

Result set:

Test the procedure as per the following commands:





create procedure checkConsumptionv2\_grp11

@ctrip varchar(10)

as

declare @avgFuelConsp float; -- to hold the average for each destination

declare @fuelConsp float; -- to hold the amount we used

--Define avgFuelConsp for charters in same destination

select @avgFuelConsp=avg(c.CHAR\_FUEL\_GALLONS)

from charter c

where c.CHAR\_DESTINATION = (select CHAR\_DESTINATION from charter where CHAR\_TRIP = @ctrip)

select @fuelConsp=avg(c.CHAR\_FUEL\_GALLONS)

from charter c

where c.CHAR\_TRIP = @ctrip

IF @fuelConsp >= @avgFuelConsp

print 'This charter has consumed more than the average fuel consumption for that destination'

ELSE

print 'This charter has consumed less than the average fuel consumption for that destination'

go

exec checkConsumptionv2\_grp11 10001

exec checkConsumptionv2\_grp11 10003

SUBMISSION

The following files should be submitted on BB:

* Lab4\_GroupX.doc file. Replace X with your group number.
* Lab4\_GroupX.sql: this file should contain
  + the question number and the question text enclosed with /\*..\*/ taken from the lab file.
  + your answer sql code underneath each question,
  + the sql statement(s) for executing the procedure with the supplied arguments as per the question specifications.

Note: If a student does NOT contribute to the work, do not list his/her name(s) under the group section in the lab file and will get 0.

Grading rubrics

Each question is worth 5pts. Total is 40 pts.

If the output is shown without the code, the answer is worth 0.