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, Divnoor Bhandohal, Sophia Huynh, 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- Divnoor Bhandohal 1-5

2- Sophia Huynh 6-8

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:

exec displayCustomers;

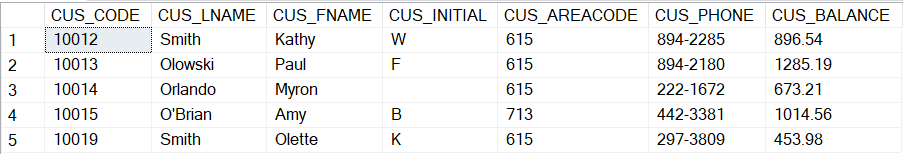
CREATE PROCEDURE displayCustomers

AS

select \* from CUSTOMER

where CUS\_BALANCE > 0

Go



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:

CREATE PROCEDURE displayCustomers\_areacode

@c\_code varchar(3)

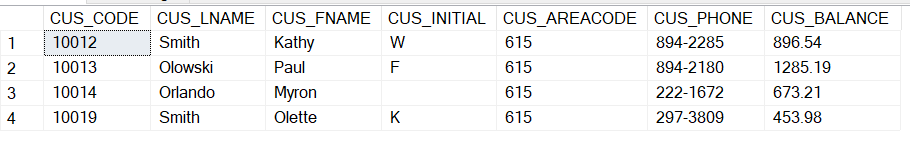
AS

select \* from CUSTOMER

where CUS\_BALANCE > 0 AND CUS\_AREACODE = @c\_code

Go

exec displayCustomers\_areacode '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 who earned the rating given in input at the specified year in input.

Testing and result set:

create procedure displayPilots

@rtg\_code varchar(5), @r\_date int

As

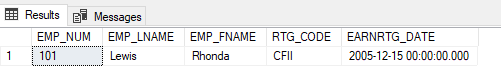
Select emp.EMP\_NUM, EMP\_LNAME, EMP\_FNAME , RTG\_CODE, EARNRTG\_DATE from EMPLOYEE emp

join EARNEDRATING e ON e.EMP\_NUM = emp.EMP\_NUM

Where RTG\_CODE = @rtg\_code AND DATEPART(yyyy,EARNRTG\_DATE) = @r\_date

go

exec displayPilots '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

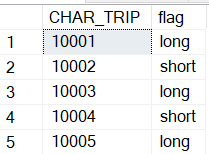
@nbhours float(8)

AS

Select CHAR\_TRIP , IIF(CHAR\_HOURS\_FLOWN < @nbhours, 'short', 'long') from CHARTER

Go

EXEC tagCharters @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.

CREATE PROCEDURE checkCustomerBal

@c\_code INT

AS

DECLARE @c\_balance FLOAT(8);

SET @c\_balance = (SELECT CUS\_BALANCE FROM CUSTOMER WHERE CUS\_CODE = @c\_code)

IF @c\_balance = 0

PRINT 'This customer has no outstanding balance'

Else

PRINT concat ('This customer has an outstanding balance of ', @c\_balance)

GO

Text

Description automatically generated

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 0.

CREATE PROCEDURE getnbChartersCus\_grp2

@c\_code INT

AS

DECLARE @c\_count INT;

SET @c\_count = (SELECT COUNT(\*) FROM CHARTER WHERE CUS\_CODE = @c\_code GROUP BY CUS\_CODE)

IF @c\_code <> 0

RETURN @c\_count

ELSE

PRINT 'invalid customer code'

RETURN 0

GO



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'.

CREATE PROCEDURE checkConsumptionv1\_grp2

AS

SELECT CHAR\_TRIP, CHAR\_DESTINATION, CHAR\_FUEL\_GALLONS,

(SELECT AVG(CHAR\_FUEL\_GALLONS) FROM CHARTER WHERE CHAR\_DESTINATION = C.CHAR\_DESTINATION) AS avgFuelConsDes,

'status' = CASE

WHEN CHAR\_FUEL\_GALLONS > (SELECT AVG(CHAR\_FUEL\_GALLONS) FROM CHARTER WHERE CHAR\_DESTINATION = C.CHAR\_DESTINATION)

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

ELSE 'consumed less than the average fuel consumption for that destination'

END

FROM CHARTER C

GO

Table

Description automatically generated

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.

CREATE PROCEDURE checkConsumptionv2\_grp2

@trip\_code INT

AS

DECLARE @avg\_Fuel FLOAT(8);

SELECT @avg\_Fuel = (SELECT AVG(CHAR\_FUEL\_GALLONS) FROM CHARTER WHERE CHAR\_DESTINATION = CHARTER.CHAR\_DESTINATION) FROM CHARTER WHERE CHAR\_TRIP = @trip\_code

DECLARE @char\_Fuel FLOAT(8);

SELECT @char\_Fuel = CHAR\_FUEL\_GALLONS FROM CHARTER WHERE CHAR\_TRIP = @trip\_code

BEGIN

IF @char\_Fuel > @avg\_Fuel

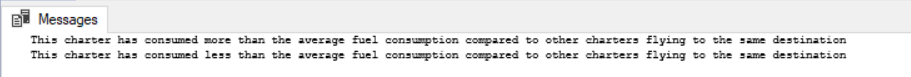
PRINT 'This charter has consumed more than the average fuel consumption compared to other charters flying to the same destination'

ELSE

PRINT 'This charter has consumed less than the average fuel consumption compared to other charters flying to the same destination'

END

GO



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.