Ryan Carey

Ruc230

4/16/2019

Organization of Data

Assignment 6

1. Create a VIEW named vCustomerOrders and submit your SQL query for creating this view. (10 points)

create or replace view vCustomerOrders

(Customer, OrderDate, OrderID)

as

select c.cust\_first\_name || ' ' || c.cust\_last\_name as "Customer Name", o.order\_timestamp as "OrderDate", o.order\_id as "OrderID"

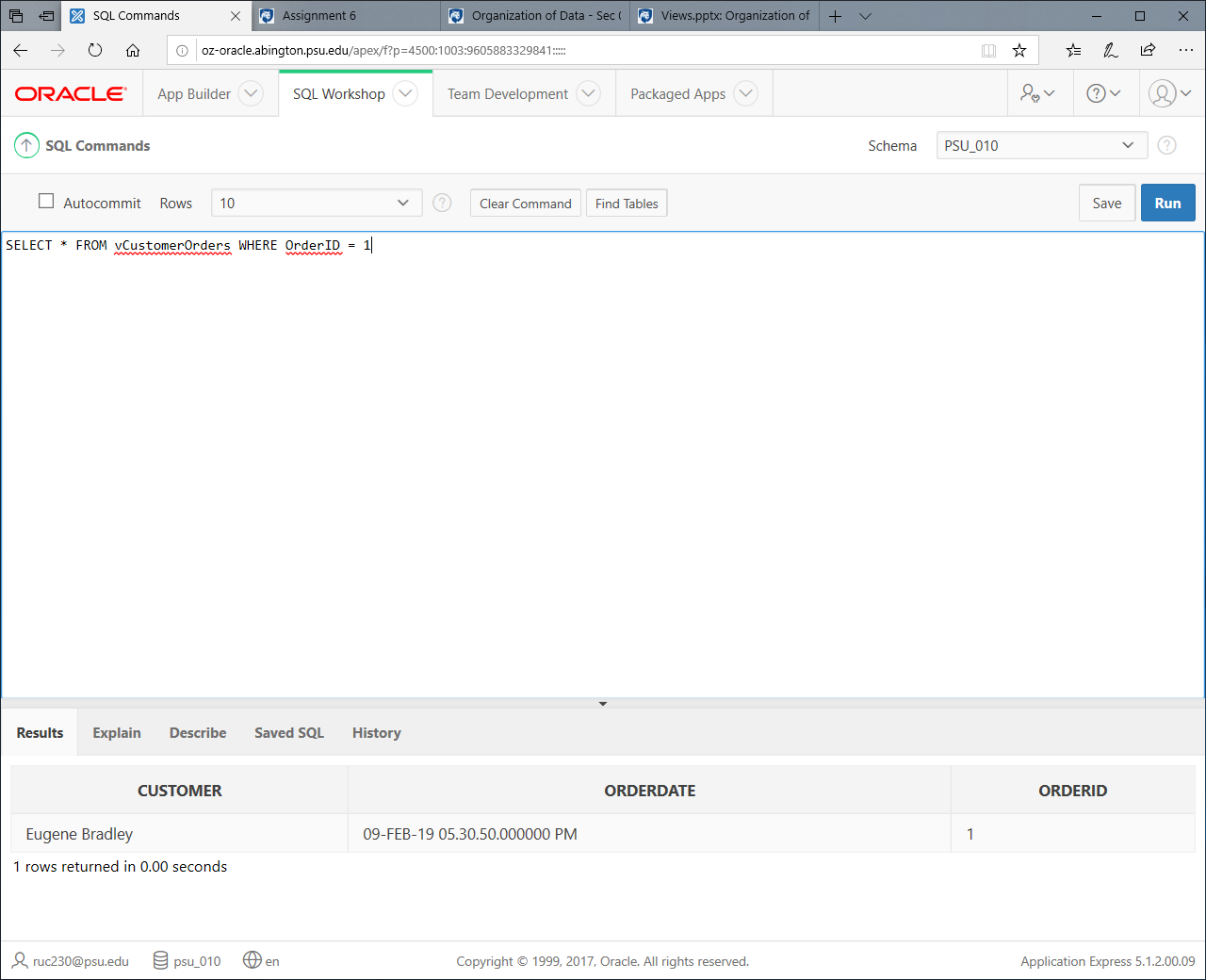
from demo\_orders o

join demo\_customers c on (c.customer\_id = o.customer\_id)

Select all Orders and their Customers. Include OrderID, OrderDate, CustomerName (Customer First and Last Name as a single attribute).

Run SQL query below and paste screenshot of your result. (10 points)

SELECT \* FROM vCustomerOrders WHERE OrderID = 1



2. Create a View named vTotalQuantitiesOrderedByProduct and submit your SQL query for creating this view. (10 points)

create view vTotalQuantitiesOrderedByProduct (TotalQuantity, Product\_Name)

as

select sum(quantity), pi.product\_name

from demo\_order\_items oi

join demo\_product\_info pi on (oi.product\_id = pi.product\_id)

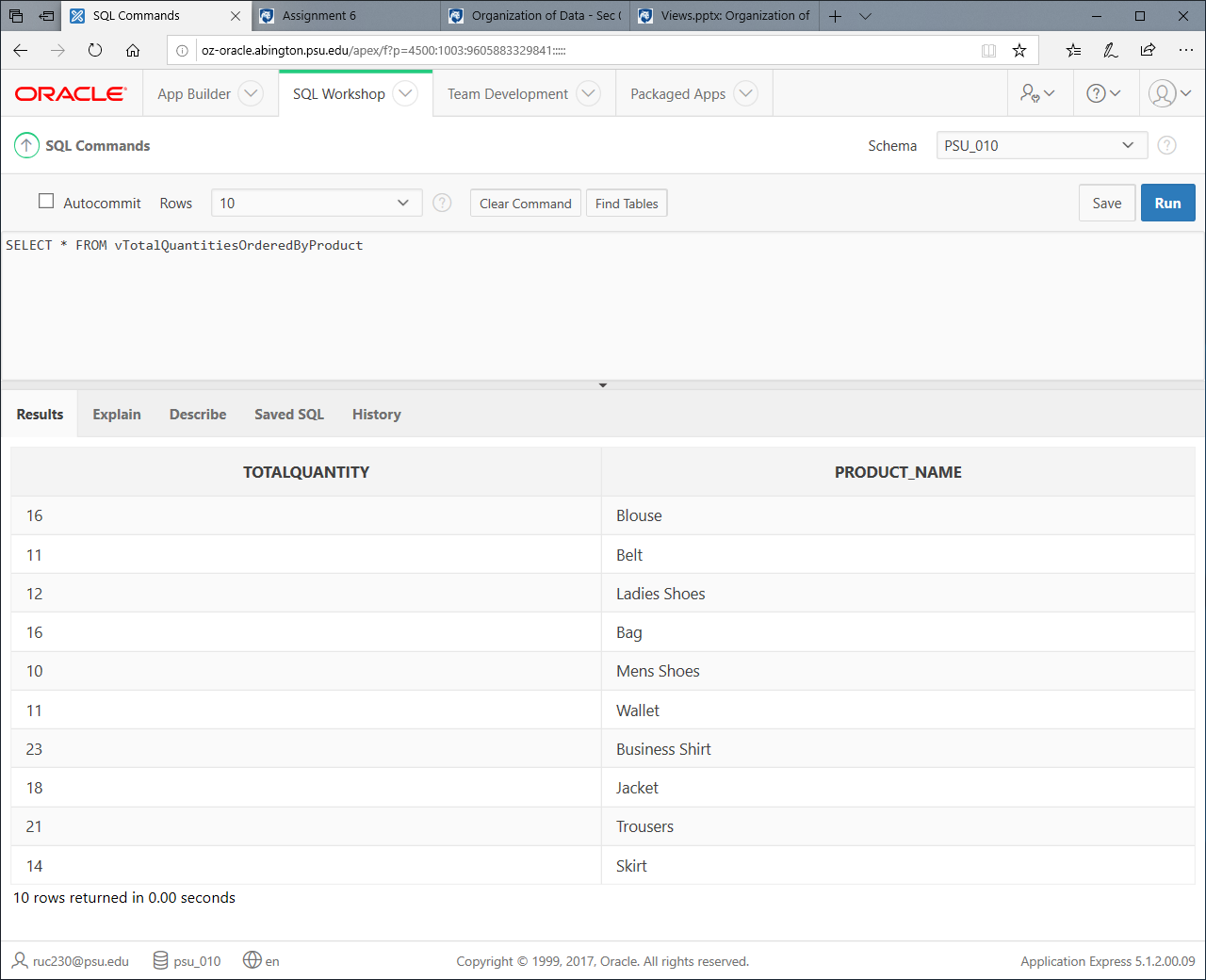
group by pi.product\_name

This view should perform the following query:

Select count of all products grouped by product name. Include SUM(Quantity) as TotalQuantity and Product Name. (Perform INNER JOIN on DEMO\_PRODUCT\_INFO and DEMO\_ORDER\_ITEMS and group it by ProductName)

Run SQL query below and paste screenshot of your result. (10 points)

SELECT \* FROM vTotalQuantitiesOrderedByProduct



3. Create a Stored Procedure Named spGetCustomerNameforOrder and submit your SQL query for creating this SP. (10 points)

create or replace procedure spGetCustomerNameforOrder(OID in demo\_ORDERS.order\_ID%TYPE)

IS

l\_cursor SYS\_REFCURSOR;

thisCustomer demo\_customers.cust\_first\_name%TYPE;

begin

OPEN l\_cursor FOR

select Customer

FROM vCustomerOrders

WHERE OrderID = OID;

fetch l\_cursor

INTO thisCustomer;

DBMS\_OUTPUT.PUT\_LINE(thisCustomer);

CLOSE l\_cursor;

end;

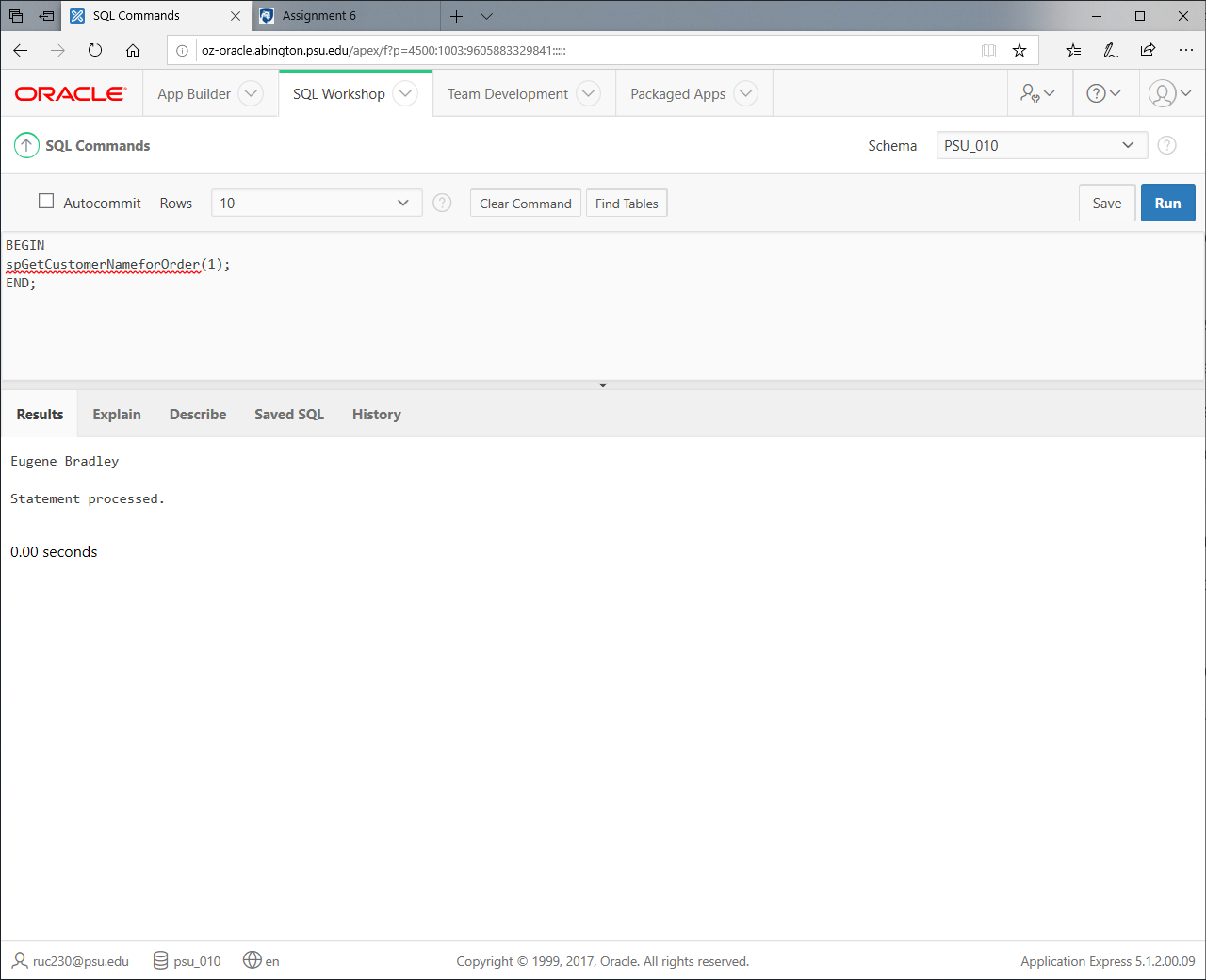
This SP should accept OrderID as a parameter and print the name of the customer on screen (Full Name). The procedure MUST make use of the view created in Question 1 and SELECT MUST be done on this View.

Run SQL query below and paste screenshot of your result. (10 points)

BEGIN

sp\_GetCustomerNameforOrder(1);

END;



4. Create a Stored Procedure Named sp\_CreateNewCustomer and submit your SQL query for creating this SP. (10 points)

create or replace procedure spCreateNewCustomer(firstn in demo\_customers.cust\_first\_name%TYPE, lastn in demo\_customers.cust\_last\_name%type)

IS

begin

insert into demo\_customers (cust\_first\_name, cust\_last\_name)

values (firstn, lastn);

end;

This SP should accept required Customer attributes (Customer First Name, Customer Last Name).

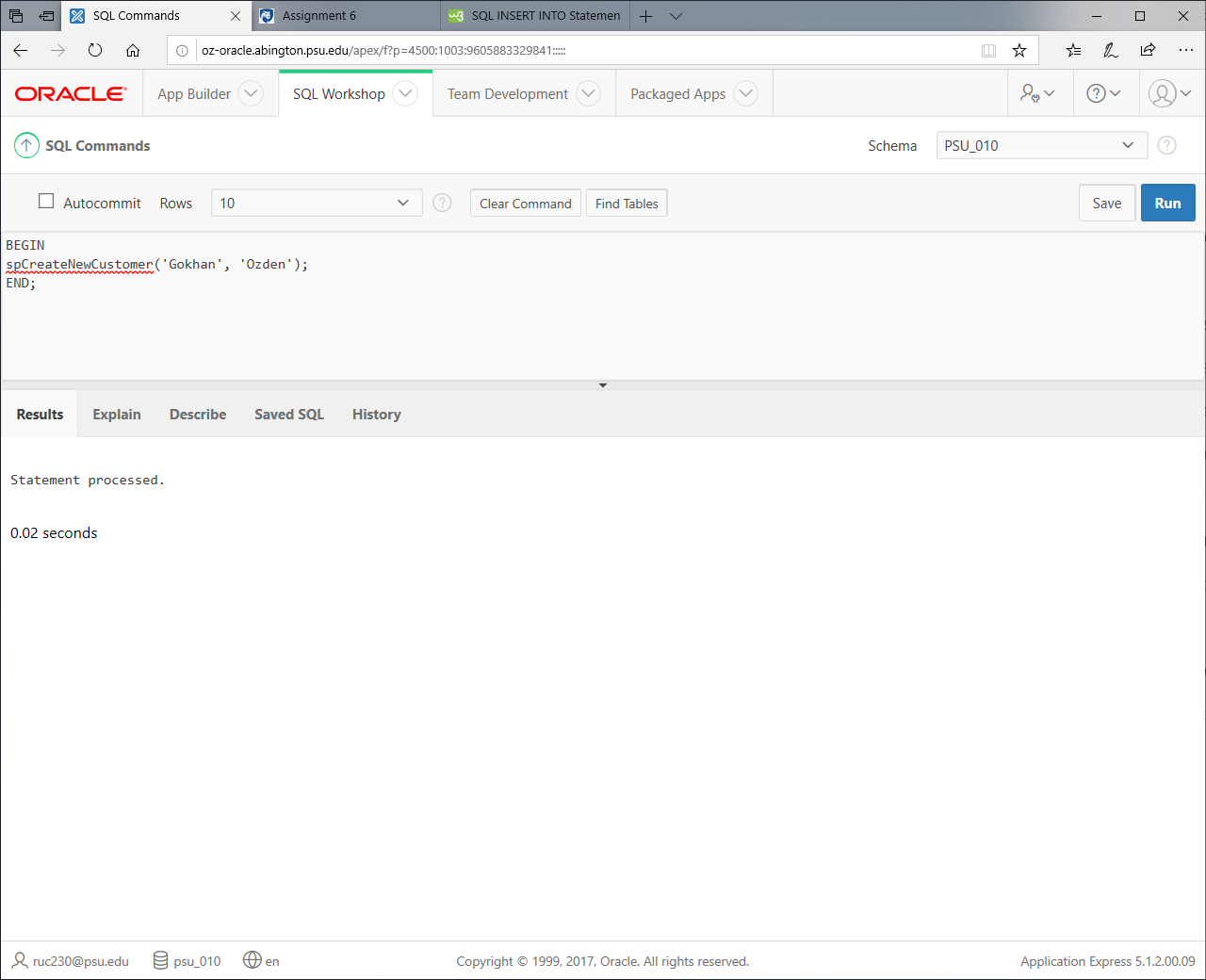
In this stored procedure you will be expected to do an INSERT into DEMO\_CUSTOMERS table.

Run these two SQL queries below and paste screenshots of your results (two screenshots one for each). (10 points)

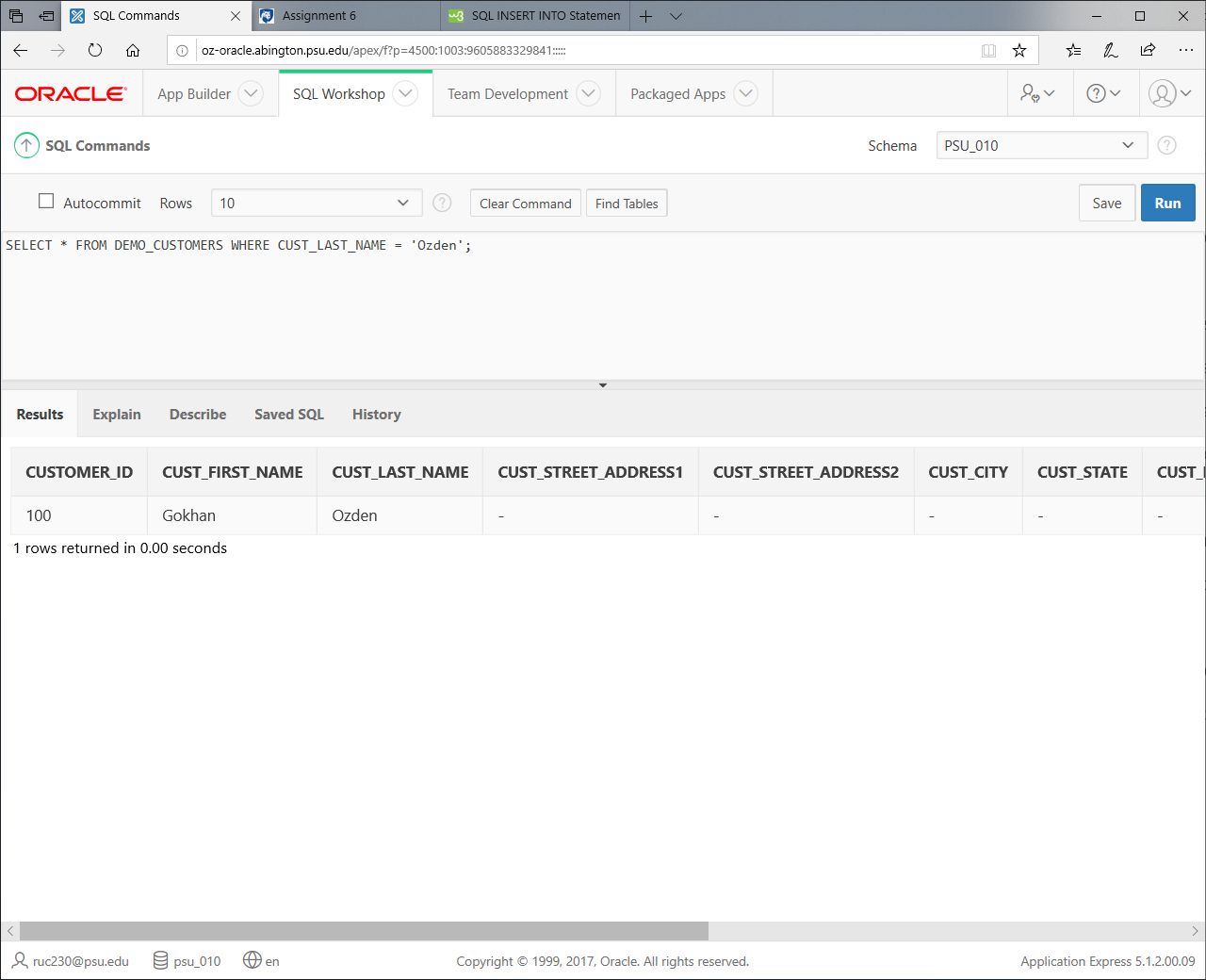
BEGIN

sp\_CreateNewCustomer('Gokhan', 'Ozden');

END;



SELECT \* FROM DEMO\_CUSTOMERS WHERE CUST\_LAST\_NAME = 'Ozden';



5. Write a function named "F\_StateName" that takes the CUST\_STATE as an input and returns the state full name from the DEMO\_STATES table. (10 points) Show me the results of the below query: (10 points

create or replace function FStateName

(

cst in varchar2

)

return varchar2

is

retstate varchar2(50);

begin

SELECT demo\_states.state\_name

INTO retstate

FROM demo\_states

WHERE demo\_states.st = cst;

return retstate;

end;

SELECT Cust\_First\_Name, Cust\_Last\_Name, F\_StateName(Cust\_State) FROM Demo\_Customers

