# In-Class Practice – Day 18 (Procedures and Functions)

1. **Create a procedure called *insert\_vendor\_contact* that will INSERT a new vendor contact into *vendor\_contacts***
   1. The procedure should be passed 3 input parameters for:
      1. vendor\_id
      2. last\_name
      3. first\_name
   2. The main body of this procedure should run an insert statement to insert a new row into the vendor\_contacts table and then commit that inserted row.
2. **Write a CALL to insert 2 new contacts** for vendor 1, 2 [from script below – notice the differences between the two]

--Call 1

CALL insert\_vendor\_contact(1,'Thorton','Bob');

--Call 2

BEGIN

insert\_vendor\_contact(2,'Williams','Selma');

END;

/

1. **Call the created procedure (by name)**
   1. Try to call procedure and insert a new contact but in the following order: vendor\_id, First Name, Last Name

---------------------------------------------------------------------------------

1. **Create a function called invoice\_count that returns the number of invoices based on a given vendor\_id.**
2. **Call the created function in different methods**
   1. Use the following SQL to test your function

SELECT vendor\_id, vendor\_name, invoice\_count(vendor\_id)

FROM vendors

WHERE invoice\_count(vendor\_id) > 0;

---------------------------------------------------------------------------------

## **If you get this far, go back to the practice on the previous day’s module and work on Dynamic SQL, Bulk Collect, Cursor, and Exceptions.**

---------------------------------------------------------------------------------

## Extra Practice on using a default in your procedure

1. Using default values in Procedure.

* Update ***insert\_vendor\_contact*** procedure to default first\_name to ‘?’ if not passed into the procedure.
* Test a call without providing an first\_name
* Test a call with providing an first\_name