Chapter 15

How to create stored procedures and functions

Exercises

1. Write a script that creates and calls a stored procedure named insert_category. First, code a statement that creates a procedure that adds a new row to the Categories table. To do that, this procedure should have one parameter for the category name.

   Code at least two CALL statements that test this procedure. (Note that this table doesn't allow duplicate category names.)

2. Write a script that creates and calls a stored function named discount_price that calculates the discount price of an item in the Order_Items table (discount amount subtracted from item price). To do that, this function should accept one parameter for the item ID, and it should return the value of the discount price for that item.

3. Write a script that creates and calls a stored function named item_total that calculates the total amount of an item in the Order_Items table (discount price multiplied by quantity). To do that, this function should accept one parameter for the item ID, it should use the discount_price function that you created in exercise 2, and it should return the value of the total for that item.

4. Write a script that creates and calls a stored procedure named insert_products that inserts a row into the Products table. This stored procedure should accept five parameters, one for each of these columns: category_id, product_code, product_name, list_price, and discount_percent.

   This stored procedure should set the description column to an empty string, and it should set the date_added column to the current date.

   If the value for the list_price column is a negative number, the stored procedure should raise an error that indicates that this column doesn't accept negative numbers. Similarly, the procedure should raise an error if the value for the discount_percent column is a negative number.

   Code at least two CALL statements that test this procedure.

5. Write a script that creates and calls a stored procedure named update_product_discount that updates the discount_percent column in the Products table. This procedure should have one parameter for the product ID and another for the discount percent.

   If the value for the discount_percent column is a negative number, the stored procedure should raise an error that the value for this column must be a positive number.

   Code at least two CALL statements that test this procedure.