

Day 10 Assignment SQL

1. Function: Total Stock Value for a Given Category

```
CREATE OR REPLACE FUNCTION get_stock_value_by_category(p_category_id
INT)
RETURNS DECIMAL(10,2) AS $$
DECLARE
    v_stock_value DECIMAL(10,2);
BEGIN
    SELECT ROUND(SUM(unit_price * units_in_stock)::DECIMAL, 2)
    INTO v_stock_value
    FROM products
    WHERE category_id = p_category_id;

    RETURN COALESCE(v_stock_value, 0.00); -- Return 0 if no products found
END;
$$ LANGUAGE plpgsql;
```

[Query](#) [Query History](#)

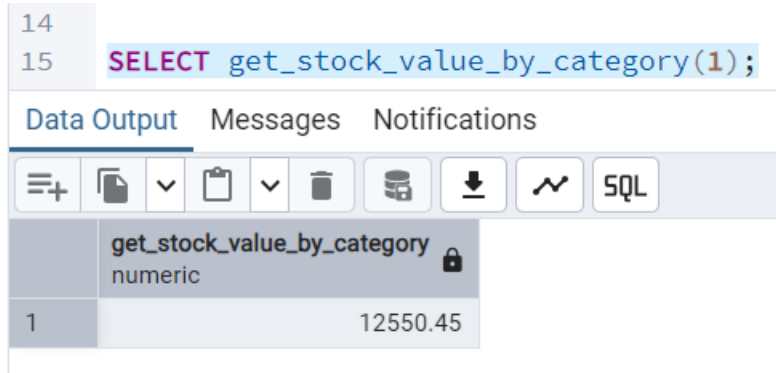
```
1 CREATE OR REPLACE FUNCTION get_stock_value_by_category(p_category_id INT)
2 RETURNS DECIMAL(10,2) AS $$
3 DECLARE
4     v_stock_value DECIMAL(10,2);
5 BEGIN
6     SELECT ROUND(SUM(unit_price * units_in_stock)::DECIMAL, 2)
7     INTO v_stock_value
8     FROM products
9     WHERE category_id = p_category_id;
10
11     RETURN COALESCE(v_stock_value, 0.00); -- Return 0 if no products found
12 END;
13 $$ LANGUAGE plpgsql;
```

[Data Output](#) [Messages](#) [Notifications](#)

```
CREATE FUNCTION
```

```
Query returned successfully in 76 msec.
```

```
SELECT get_stock_value_by_category(1);
```



The screenshot shows a SQL IDE interface. At the top, a query editor displays two lines of code: line 14 is empty, and line 15 contains the SQL query `SELECT get_stock_value_by_category(1);`. Below the editor is a toolbar with icons for file operations, execution, and SQL formatting. The 'Data Output' tab is selected, showing a table with one row. The table has two columns: the first column is labeled '1' and the second column is labeled '12550.45'. The table header row shows the column name 'get_stock_value_by_category' and its data type 'numeric'.

	get_stock_value_by_category numeric
1	12550.45

2. Cursor Query Example (Read Products One by One)

```
DO $$
```

```
DECLARE
```

```
    cur_products CURSOR FOR
```

```
        SELECT product_id, product_name, unit_price FROM products;
```

```
    rec_product RECORD;
```

```
BEGIN
```

```
    OPEN cur_products;
```

```
LOOP
```

```
    FETCH cur_products INTO rec_product;
```

```
    EXIT WHEN NOT FOUND;
```

```
        RAISE NOTICE 'Product ID: %, Name: %, Price: %',
```

```
            rec_product.product_id, rec_product.product_name,  
rec_product.unit_price;
```

```
    END LOOP;
```

```
    CLOSE cur_products;
```

```
END;
```

```
$$;
```

```

16
17  DO $$
18  DECLARE
19      cur_products CURSOR FOR
20          SELECT product_id, product_name, unit_price FROM products;
21
22      rec_product RECORD;
23  BEGIN
24      OPEN cur_products;
25
26      LOOP
27          FETCH cur_products INTO rec_product;
28          EXIT WHEN NOT FOUND;
29
30          RAISE NOTICE 'Product ID: %, Name: %, Price: %',
31              rec_product.product_id, rec_product.product_name, rec_product

```

Data Output Messages Notifications

```

NOTICE: Product ID: 2, Name: Chang, Price: 19
NOTICE: Product ID: 3, Name: Aniseed Syrup, Price: 10
NOTICE: Product ID: 4, Name: Chef Anton's Cajun Seasoning, Price: 22
NOTICE: Product ID: 5, Name: Chef Anton's Gumbo Mix, Price: 21.35
NOTICE: Product ID: 6, Name: Grandma's Boysenberry Spread, Price: 25
NOTICE: Product ID: 7, Name: Uncle Bob's Organic Dried Pears, Price: 30
NOTICE: Product ID: 8, Name: Northwoods Cranberry Sauce, Price: 40
NOTICE: Product ID: 9, Name: Mishi Kobe Niku, Price: 97
NOTICE: Product ID: 10, Name: Ikura, Price: 31
NOTICE: Product ID: 11, Name: Queso Cabrales, Price: 21
NOTICE: Product ID: 12, Name: Queso Manchego La Pastora, Price: 38
NOTICE: Product ID: 13, Name: Konbu, Price: 6
NOTICE: Product ID: 14, Name: Tofu, Price: 23.25
NOTICE: Product ID: 15, Name: Genen Shouvu, Price: 13

```