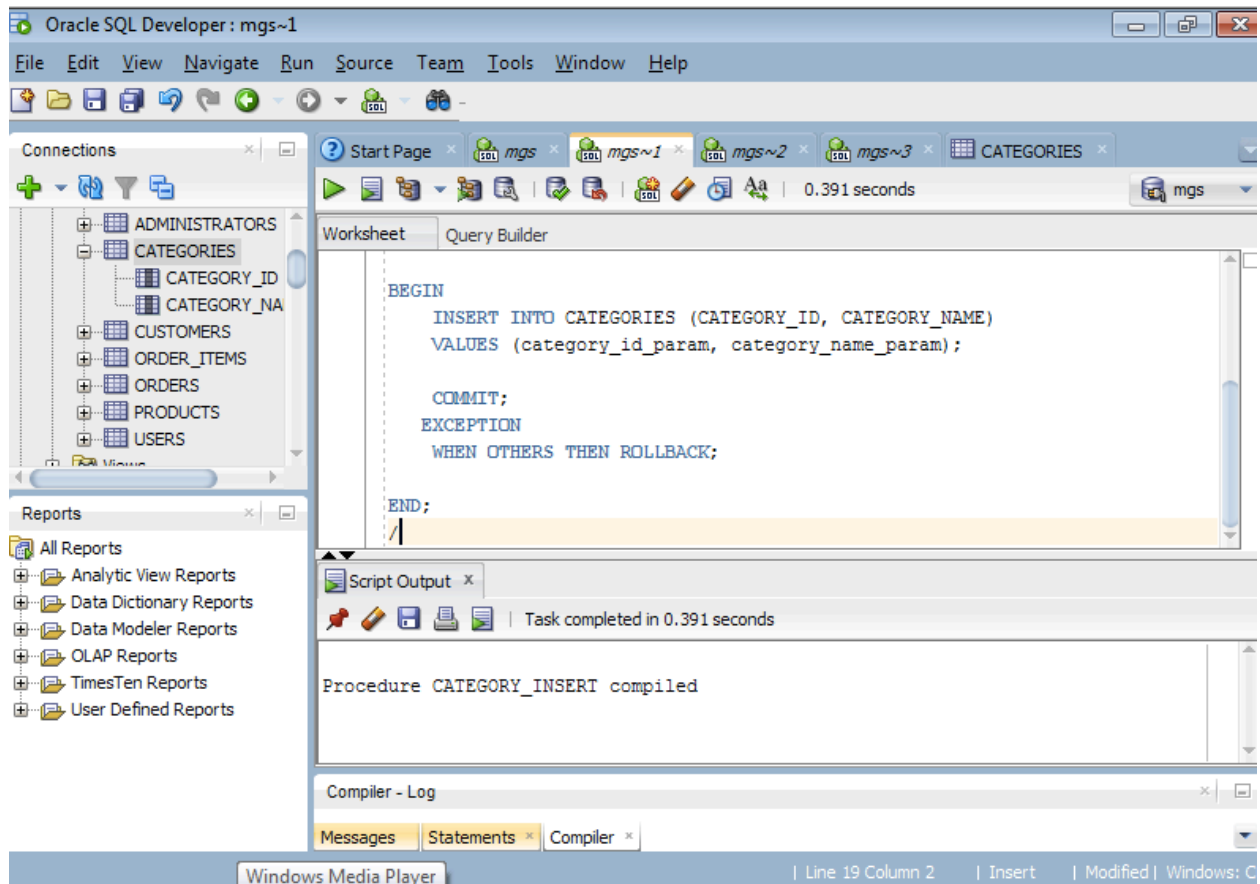


Domas Budrys Assignment 7 – CSCI 4430

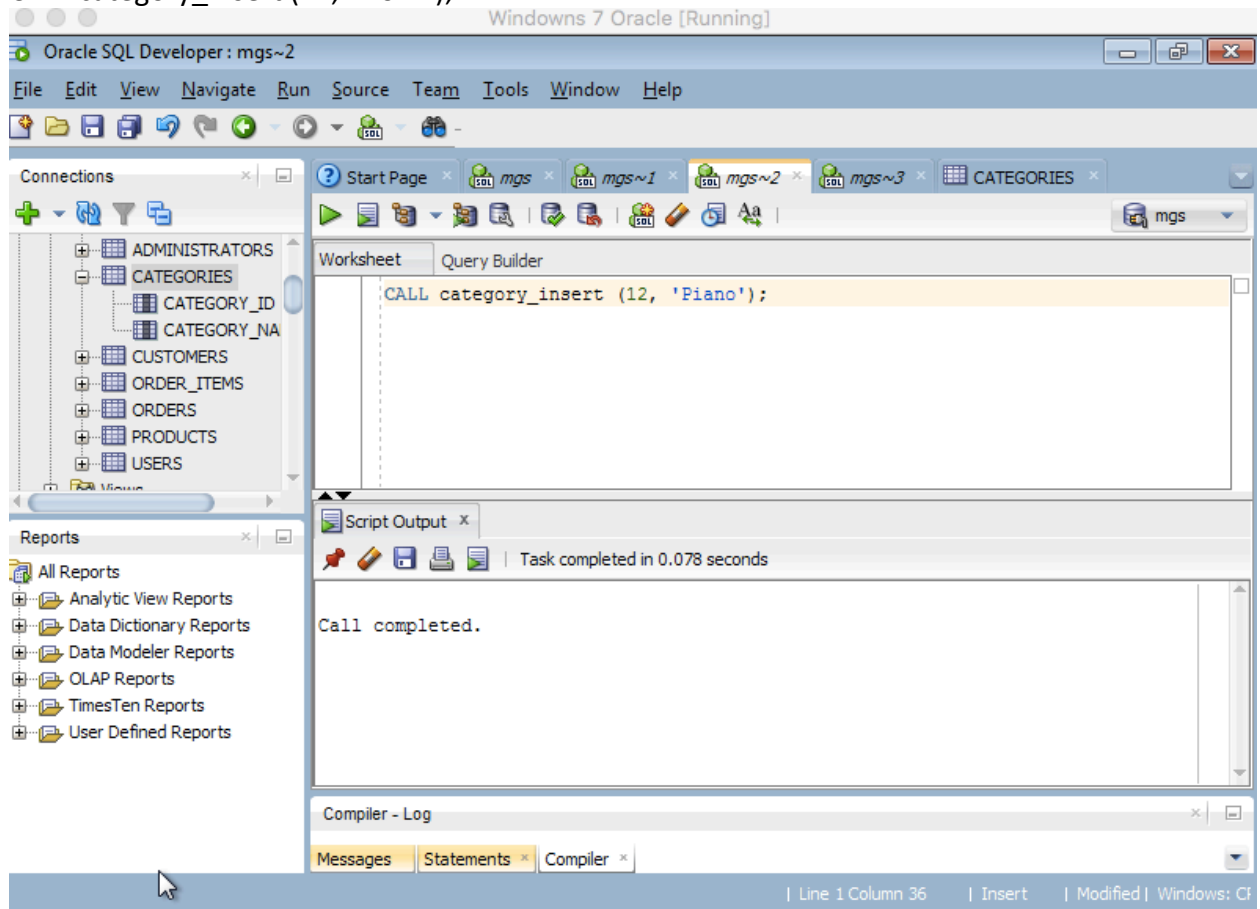
1.

```
CREATE OR REPLACE PROCEDURE category_insert
(
    category_id_param  NUMBER,
    category_name_param VARCHAR2
)
AS
BEGIN
    INSERT INTO CATEGORIES (CATEGORY_ID, CATEGORY_NAME)
    VALUES (category_id_param, category_name_param);

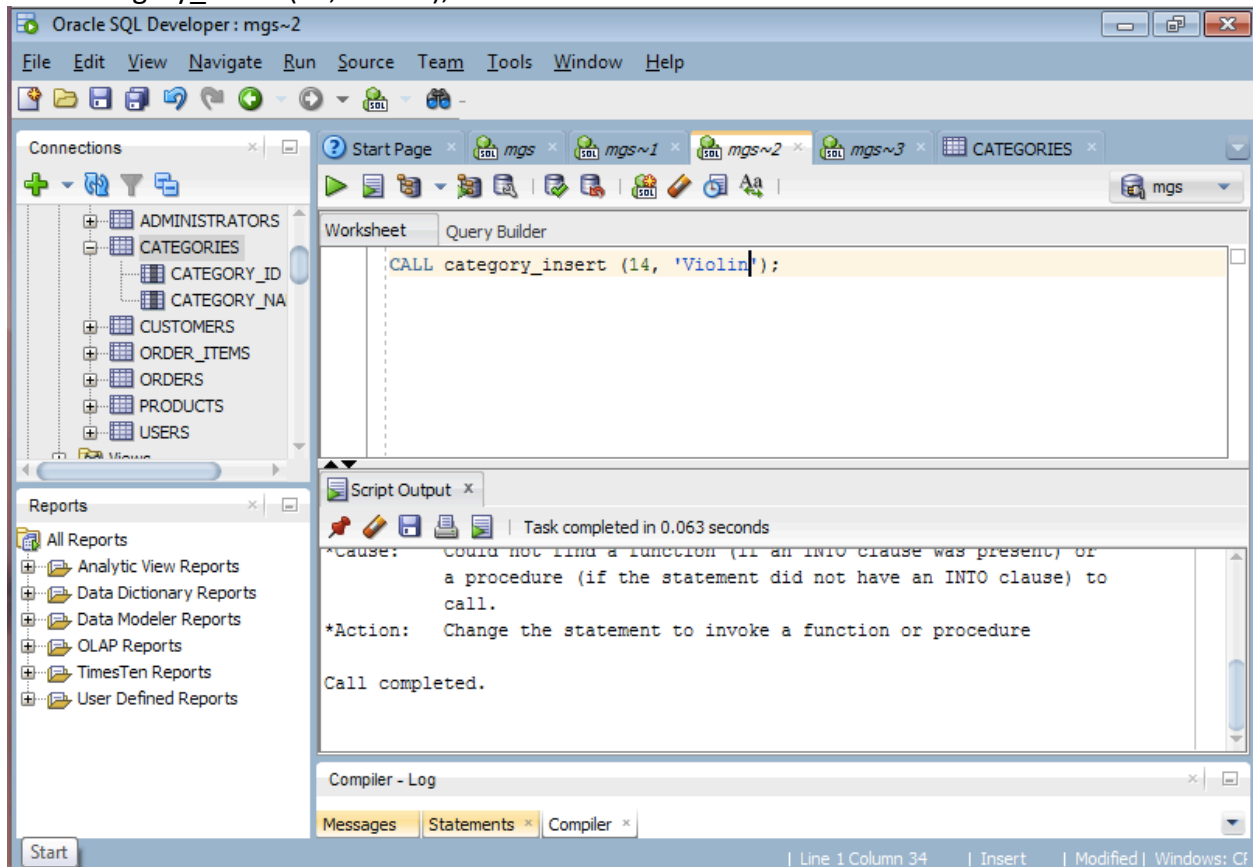
    COMMIT;
EXCEPTION
    WHEN OTHERS THEN ROLLBACK;
END;
/
```



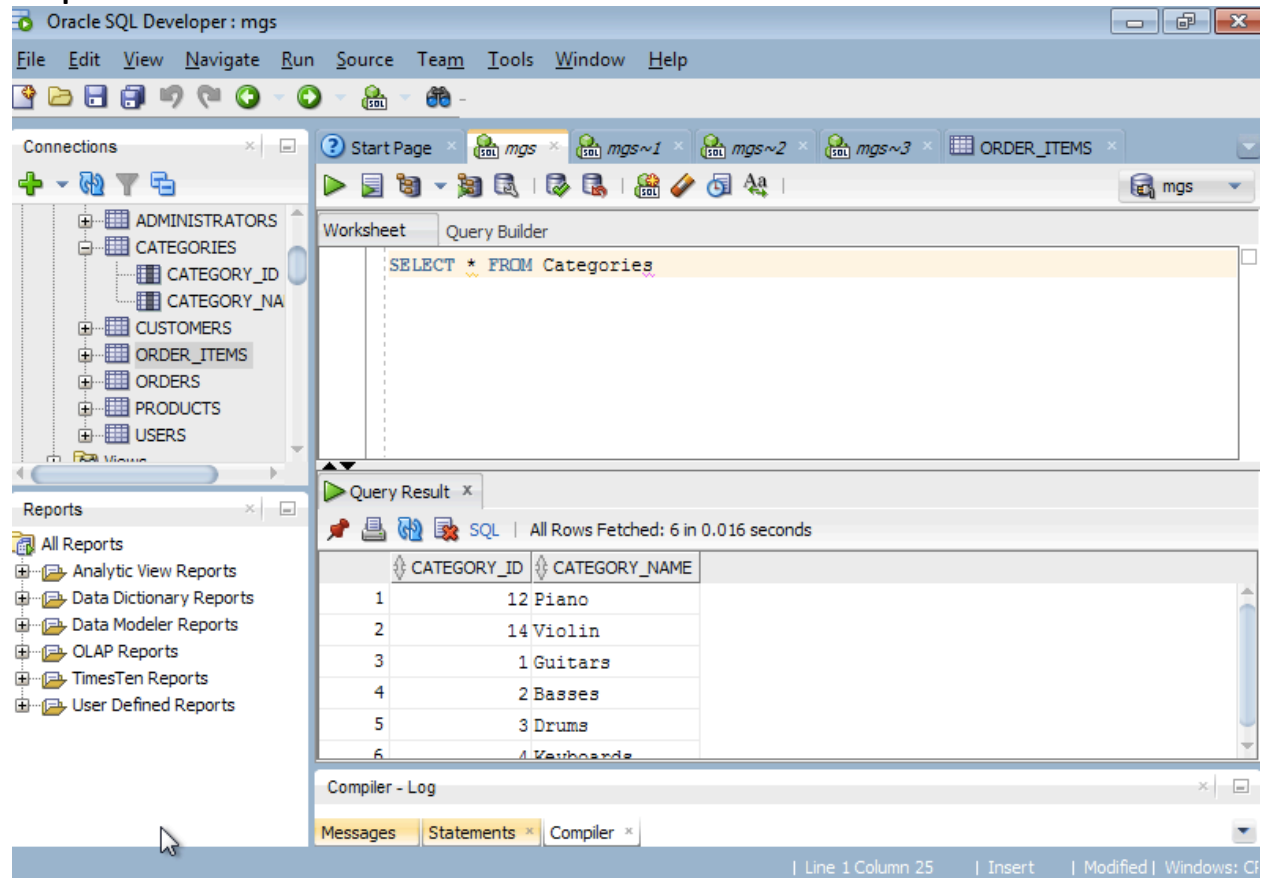
CALL category_insert (14, 'Violin');



CALL category_insert (12, 'Piano');



Output:



The screenshot displays the Oracle SQL Developer interface. The 'Connections' pane on the left shows a tree view of the database schema, including tables like ADMINISTRATORS, CATEGORIES, CUSTOMERS, ORDER_ITEMS, ORDERS, PRODUCTS, and USERS. The 'Query Builder' tab is active, showing a SQL query: `SELECT * FROM Categories`. The 'Query Result' pane below the query shows the output of the query, displaying 6 rows of data. The status bar at the bottom indicates 'All Rows Fetched: 6 in 0.016 seconds'.

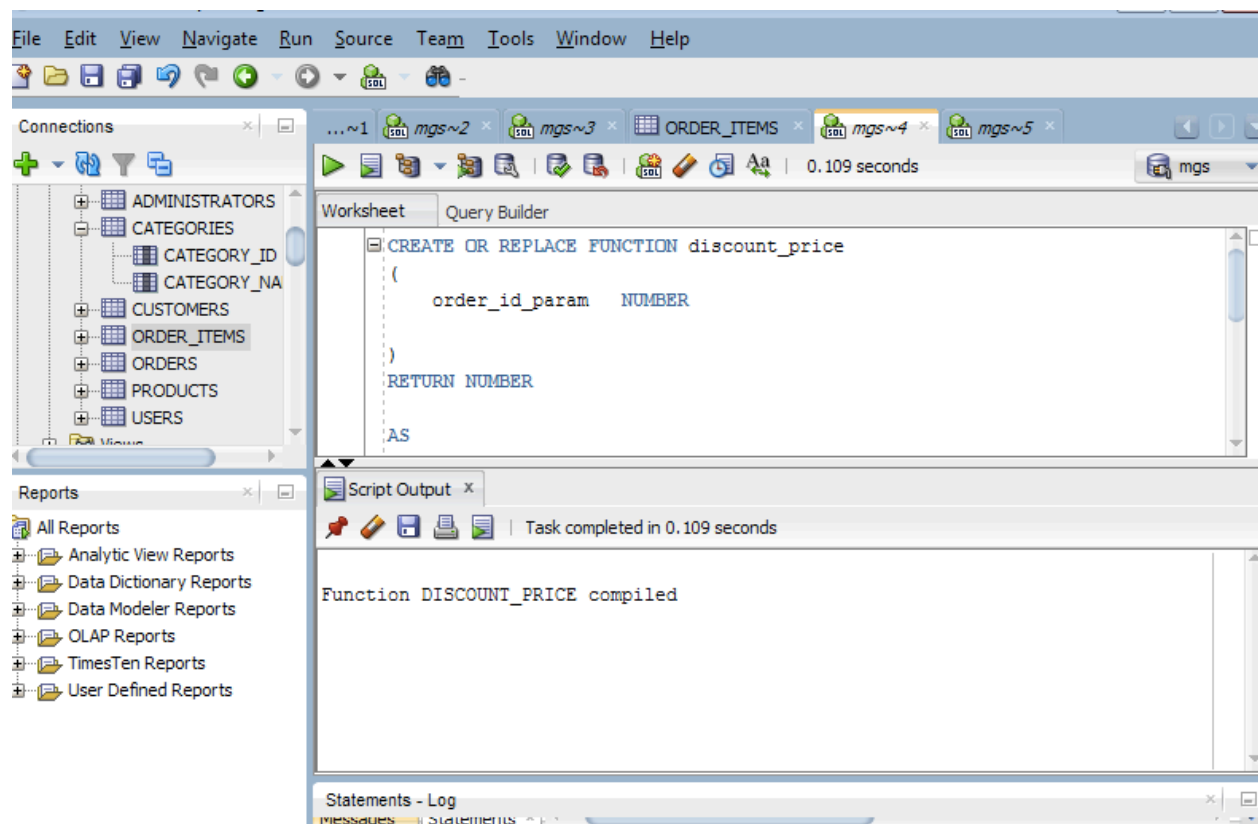
| | CATEGORY_ID | CATEGORY_NAME |
|---|-------------|---------------|
| 1 | 12 | Piano |
| 2 | 14 | Violin |
| 3 | 1 | Guitars |
| 4 | 2 | Basses |
| 5 | 3 | Drums |
| 6 | 4 | Keyboards |

2.

```
CREATE OR REPLACE FUNCTION discount_price
(
    order_id_param  NUMBER
)
RETURN NUMBER
AS
    order_id_var  NUMBER;

BEGIN
    SELECT ITEM_PRICE - DISCOUNT_AMOUNT AS "TotalSome "
    INTO order_id_var
    FROM ORDER_ITEMS
    WHERE ORDER_ID = order_id_param;

RETURN order_id_var;
END;
```



```
SELECT DISCOUNT_PRICE(ORDER_ID) AS "Discount Total"
FROM ORDER_ITEMS
WHERE ORDER_ID = 2;
```

The screenshot displays the Oracle SQL Developer interface. The top menu bar includes File, Edit, View, Navigate, Run, Source, Team, Tools, Window, and Help. The left sidebar contains a 'Connections' pane with a tree view of database objects (ADMINISTRATORS, CATEGORIES, CATEGORY_ID, CATEGORY_NAME, CUSTOMERS, ORDER_ITEMS, ORDERS, PRODUCTS, USERS) and a 'Reports' pane with categories like All Reports, Analytic View Reports, Data Dictionary Reports, Data Modeler Reports, OLAP Reports, TimesTen Reports, and User Defined Reports. The main workspace is divided into a 'Worksheet' and a 'Query Builder' tab. The 'Worksheet' tab shows the following SQL query:

```
SELECT DISCOUNT_PRICE(ORDER_ID) AS "Discount Total"
FROM ORDER_ITEMS
WHERE ORDER_ID = 2;
```

Below the query, the 'Query Result' pane shows the execution status: 'All Rows Fetched: 1 in 0 seconds'. The results are displayed in a table with one column, 'Discount Total', and one row with the value 303.79.

| Discount Total |
|----------------|
| 303.79 |

At the bottom, the 'Statements - Log' pane is visible, showing a list of statements.

3.

CREATE OR REPLACE FUNCTION item_total

```
(  
    order_id_param  NUMBER  
)
```

RETURN NUMBER

AS

```
    total_item_var  NUMBER;
```

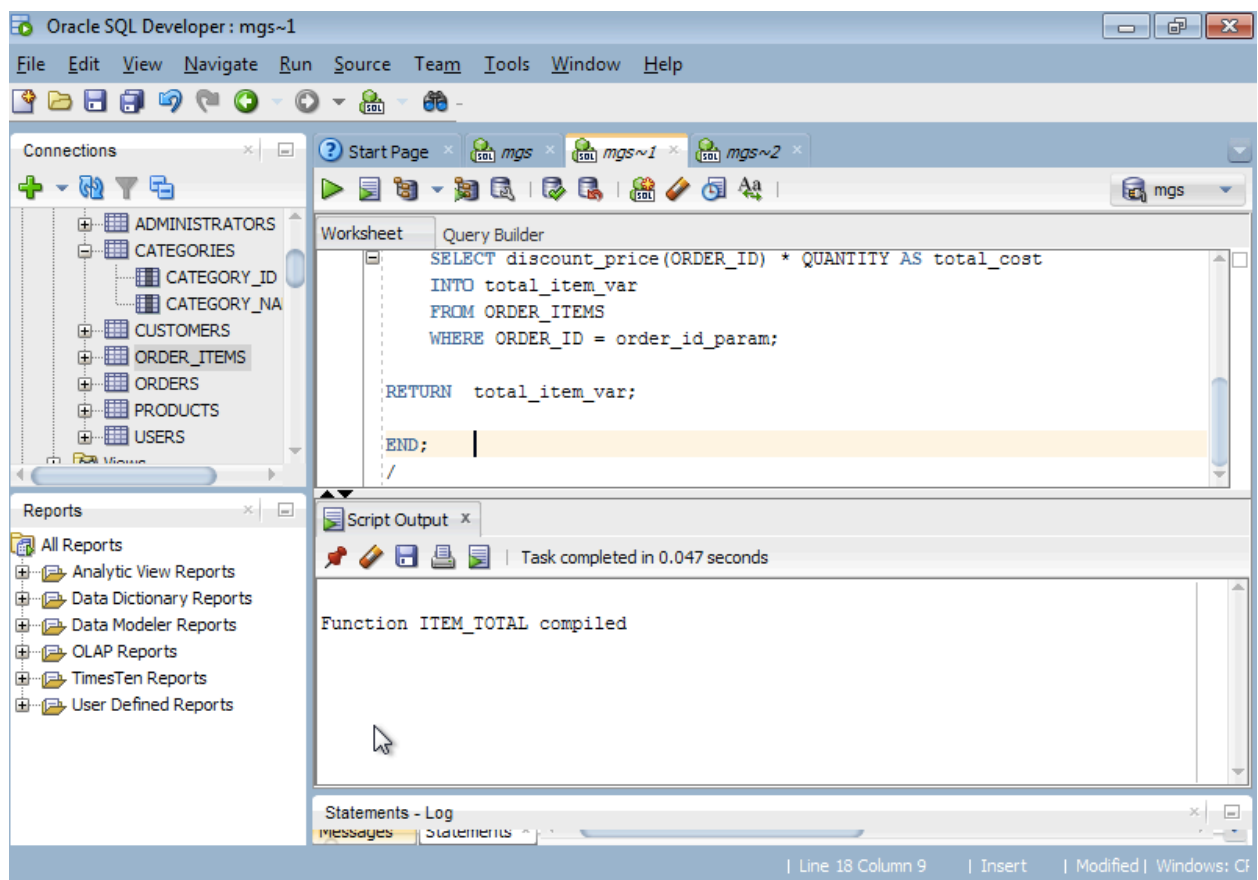
BEGIN

```
    SELECT discount_price(ORDER_ID) * QUANTITY AS total_cost  
    INTO total_item_var  
    FROM ORDER_ITEMS  
    WHERE ORDER_ID = order_id_param;
```

RETURN total_item_var;

END;

/



```
SELECT ORDER_ID, ITEM_TOTAL(order_id) AS "Total Cost"
FROM ORDER_ITEMS
WHERE ORDER_ID = 1;
```

The screenshot displays the Oracle SQL Developer interface. The main window is titled 'Oracle SQL Developer : mgs~2'. The 'Connections' pane on the left shows a tree view of database objects, including 'ADMINISTRATORS', 'CATEGORIES', 'CATEGORY_ID', 'CATEGORY_NAME', 'CUSTOMERS', 'ORDER_ITEMS', 'ORDERS', 'PRODUCTS', and 'USERS'. The 'Reports' pane below it lists various report types. The central 'Worksheet' area contains the following SQL query:

```
SELECT ORDER_ID, ITEM_TOTAL(order_id) AS "Total Cost"
FROM ORDER_ITEMS
WHERE ORDER_ID = 1;
```

Below the query, the 'Query Result' pane shows the output of the query. It indicates 'All Rows Fetched: 1 in 0 seconds'. The results are displayed in a table with two columns: 'ORDER_ID' and 'Total Cost'.

| ORDER_ID | Total Cost |
|----------|------------|
| 1 | 839.3 |

4.

CREATE OR REPLACE PROCEDURE insert_products

```
(  
    product_id_param      NUMBER,  
    category_id_param     NUMBER,  
    product_code_param    VARCHAR2,  
    product_name_param     VARCHAR2,  
    description_param      VARCHAR2 DEFAULT (' '),  
    list_price_param       NUMBER,  
    discount_percent_param NUMBER,  
    date_added_param       DATE    DEFAULT (SYSDATE)  
)
```

AS

BEGIN

IF

list_price_param < 0 THEN

RAISE VALUE_ERROR;

END IF;

IF

discount_percent_param < 0 THEN

RAISE VALUE_ERROR;

END IF;

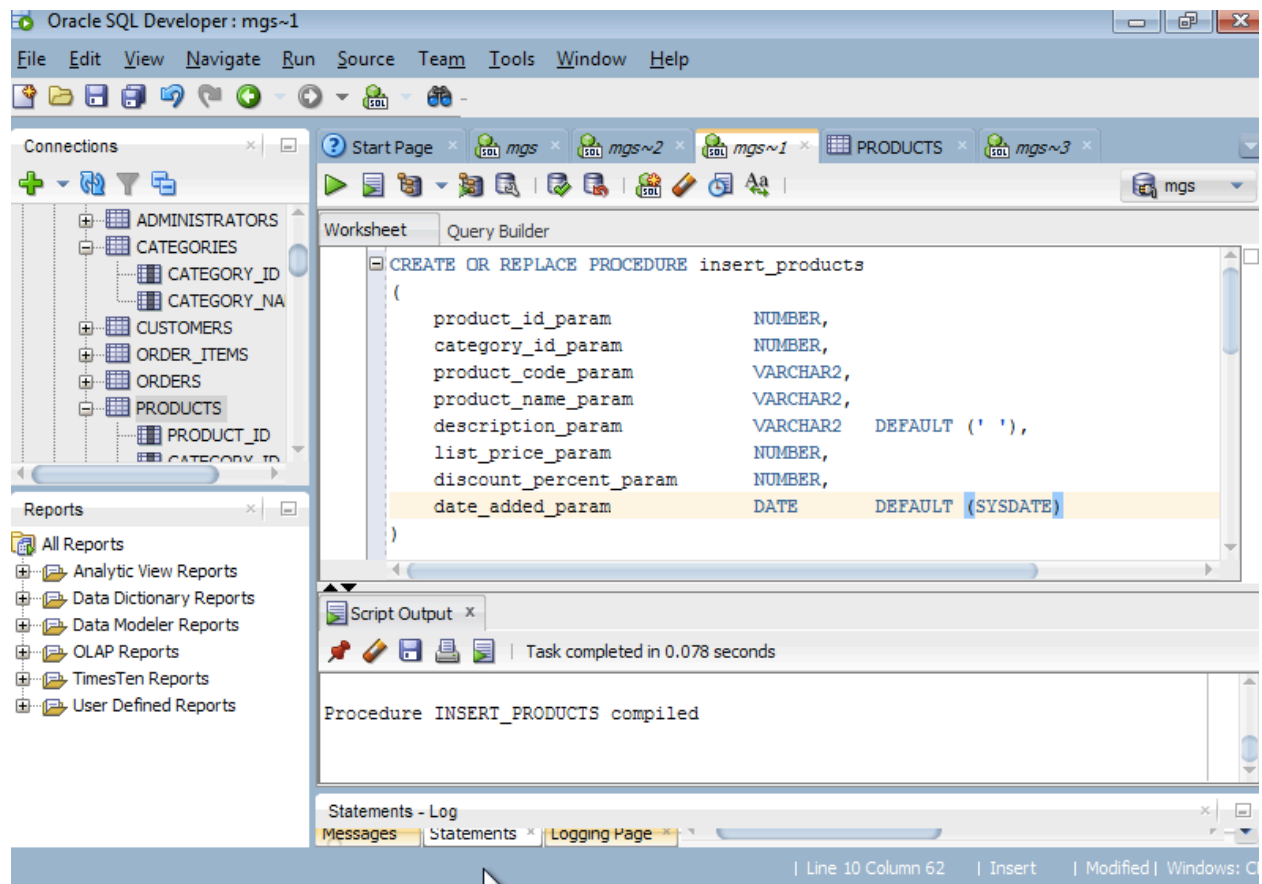
INSERT INTO PRODUCTS(PRODUCT_ID, CATEGORY_ID, PRODUCT_CODE, PRODUCT_NAME,
DESCRIPTION, LIST_PRICE, DISCOUNT_PERCENT, DATE_ADDED)

VALUES(product_id_param, category_id_param, product_code_param,
product_name_param,
description_param, list_price_param, discount_percent_param, date_added_param);

COMMIT;

END;

/



```
SET SERVEROUTPUT ON;
```

```
BEGIN
```

```
insert_products (15, 1,'Good', 'THINGS', ' ', 111, 23, SYSDATE);
```

```
EXCEPTION
```

```
  WHEN VALUE_ERROR THEN
```

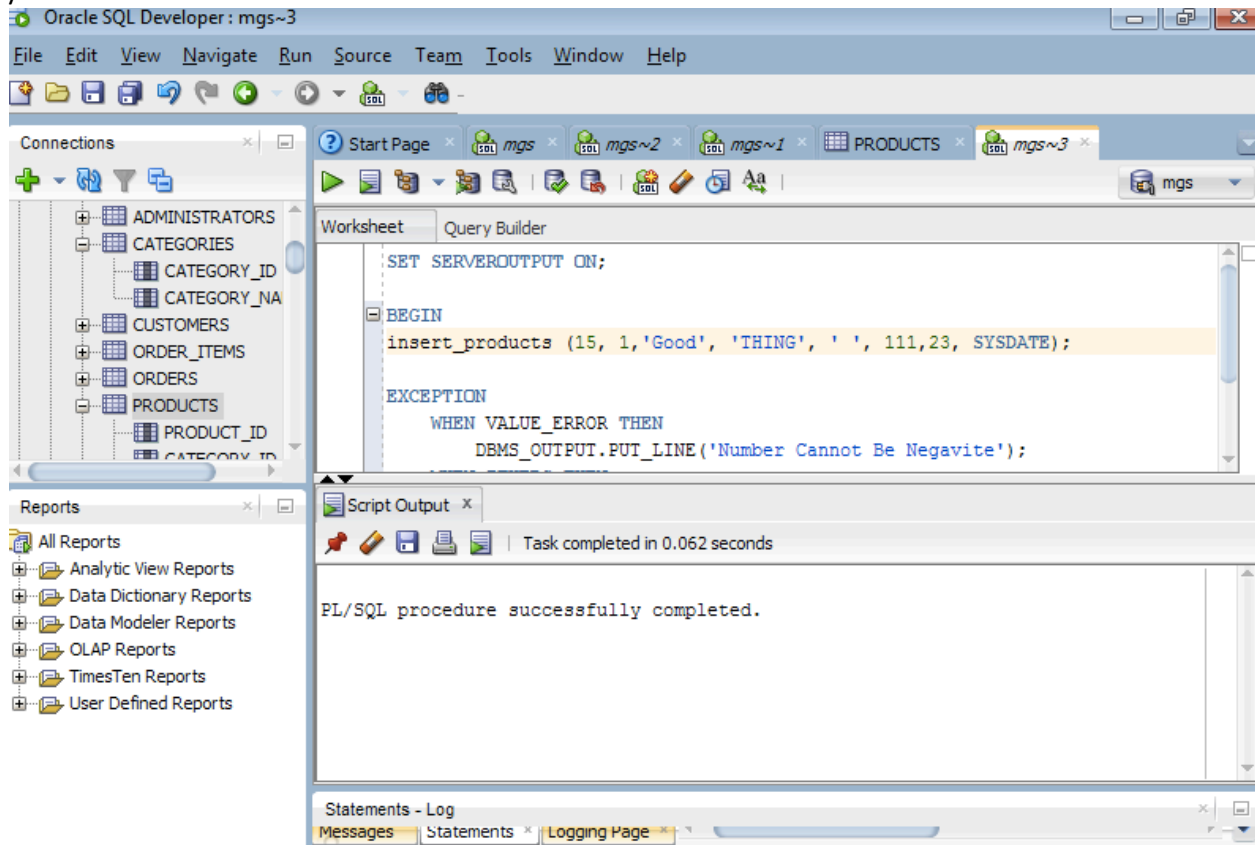
```
    DBMS_OUTPUT.PUT_LINE('Number Cannot Be Negavite');
```

```
  WHEN OTHERS THEN
```

```
    DBMS_OUTPUT.PUT_LINE('This is unexpected Error');
```

```
END;
```

```
/
```



```
SET SERVEROUTPUT ON;
```

```
BEGIN
```

```
insert_products (15, 1,'BAD', 'THINGS', ' ', 111, -23, SYSDATE);
```

```
EXCEPTION
```

```
  WHEN VALUE_ERROR THEN
```

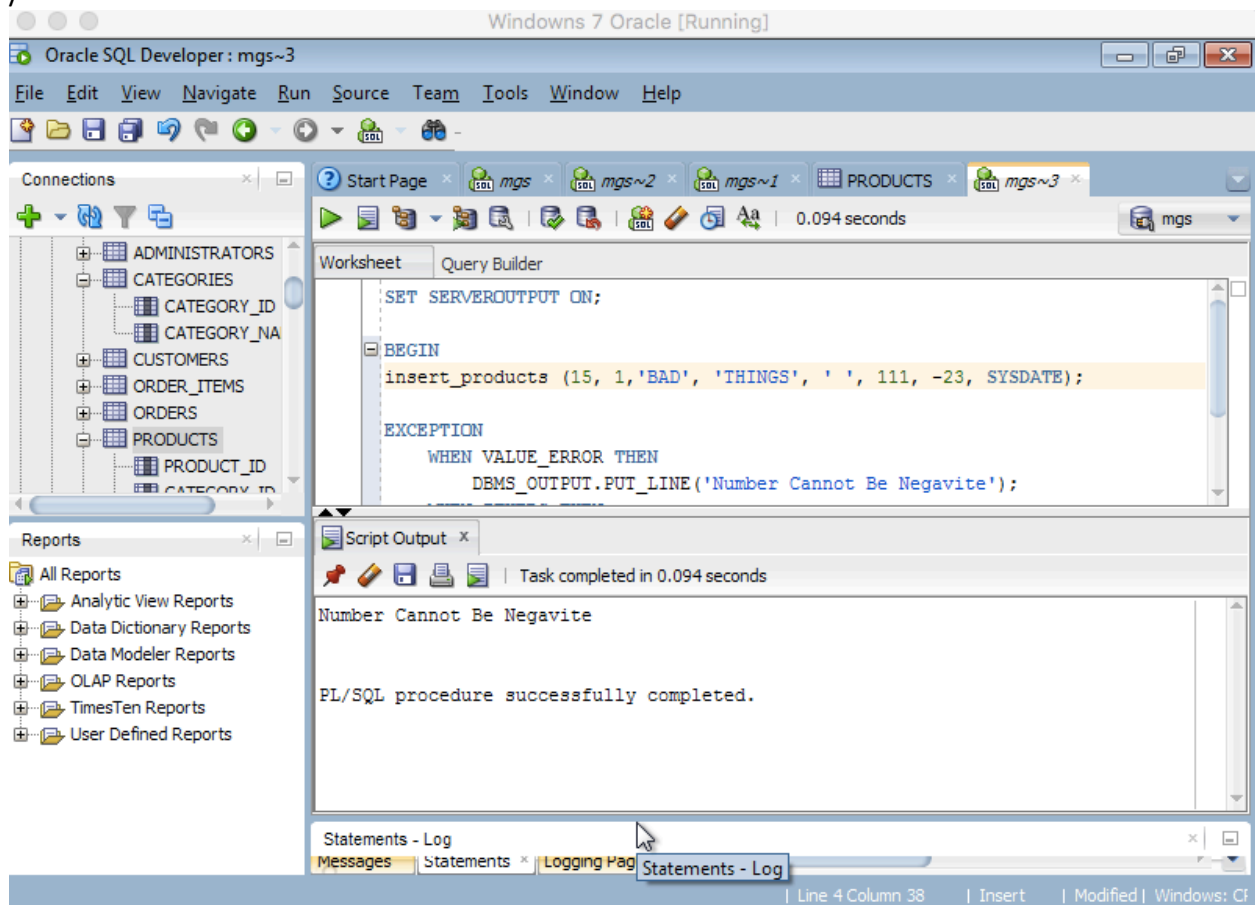
```
    DBMS_OUTPUT.PUT_LINE('Number Cannot Be Negavite');
```

```
  WHEN OTHERS THEN
```

```
    DBMS_OUTPUT.PUT_LINE('This is unexpected Error');
```

```
END;
```

```
/
```



5.

CREATE OR REPLACE PROCEDURE update_product_discount

(

product_id_param NUMBER,

discount_percent_param NUMBER

)

AS

BEGIN

IF

discount_percent_param < 0 THEN

RAISE VALUE_ERROR;

END IF;

UPDATE PRODUCTS

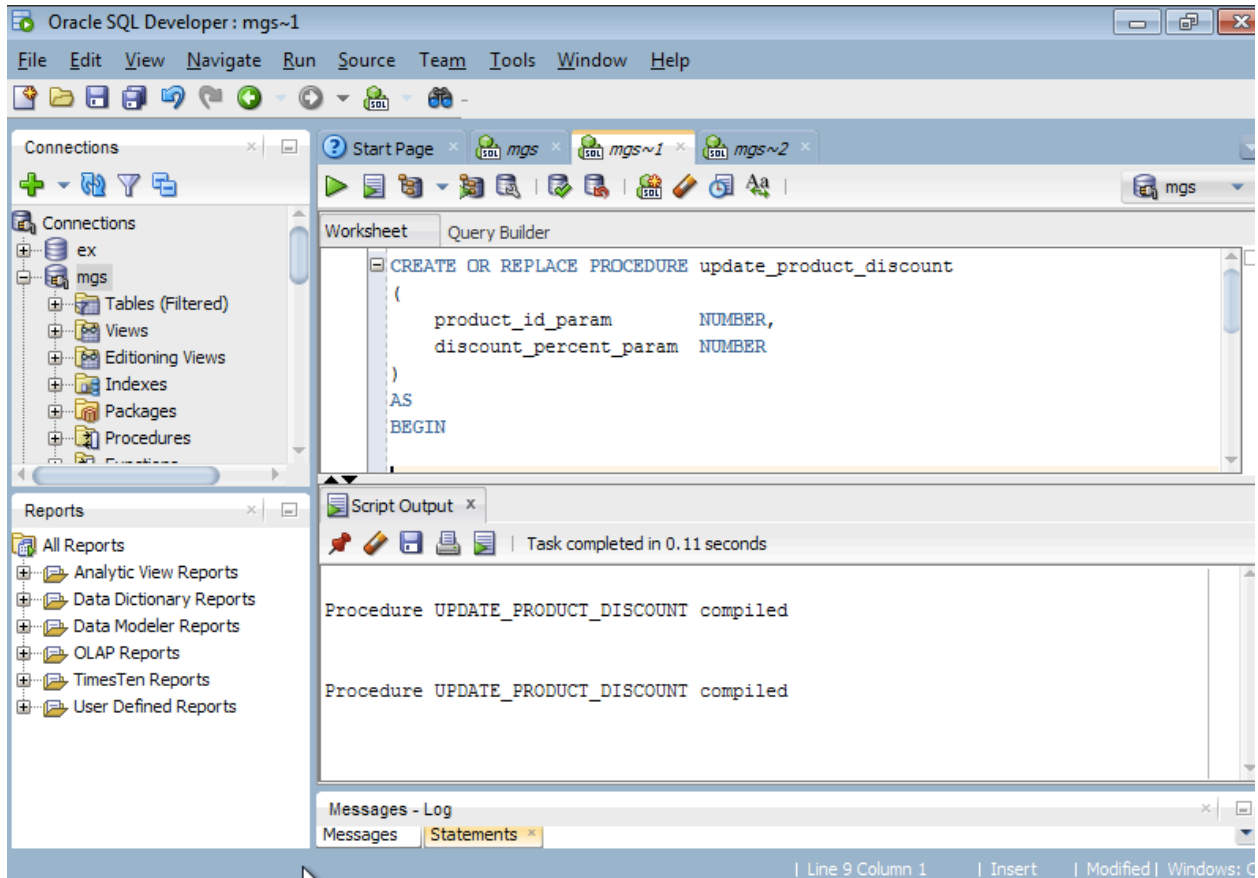
SET discount_percent = discount_percent_param

WHERE product_id = product_id_param;

COMMIT;

END;

/



```
SET SERVEROUTPUT ON;
```

```
BEGIN
```

```
update_product_discount(2, -17);
```

```
EXCEPTION
```

```
    WHEN VALUE_ERROR THEN
```

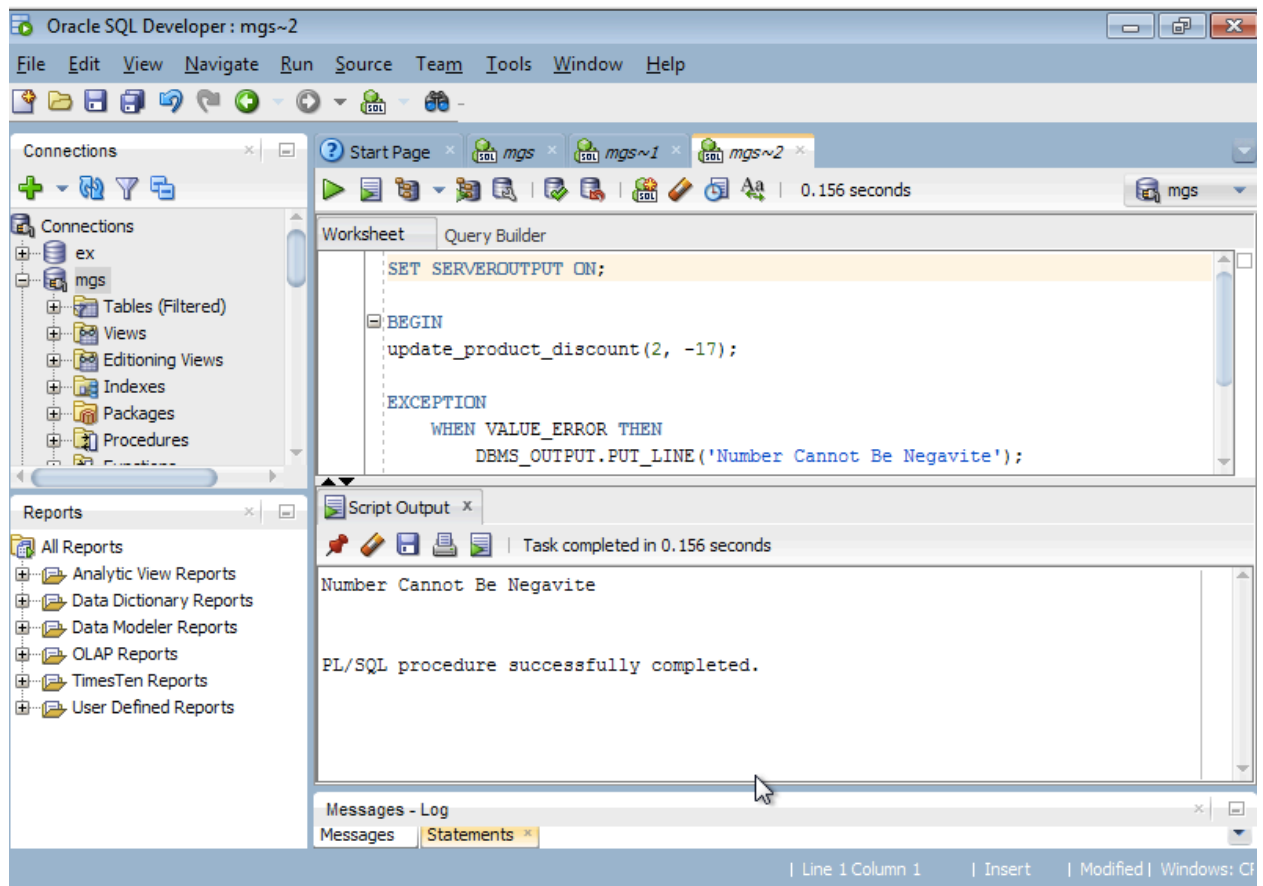
```
        DBMS_OUTPUT.PUT_LINE('Number Cannot Be Negavite');
```

```
    WHEN OTHERS THEN
```

```
        DBMS_OUTPUT.PUT_LINE('This is unexpected Error');
```

```
END;
```

```
/
```



```
SET SERVEROUTPUT ON;
```

```
BEGIN
```

```
update_product_discount(15, 20);
```

```
EXCEPTION
```

```
    WHEN VALUE_ERROR THEN
```

```
        DBMS_OUTPUT.PUT_LINE('Number Cannot Be Negative');
```

```
    WHEN OTHERS THEN
```

```
        DBMS_OUTPUT.PUT_LINE('This is unexpected Error');
```

```
END;
```

```
/
```

The screenshot displays the Oracle SQL Developer environment. The main window shows a worksheet with the following SQL query:

```
SELECT PRODUCT_ID, DISCOUNT_PERCENT FROM Products
```

Below the query, the 'Query Result' tab is active, showing the results of the query. The results are displayed in a table with 6 rows and 2 columns: PRODUCT_ID and DISCOUNT_PERCENT. The status bar indicates 'All Rows Fetched: 11 in 0.016 seconds'.

| PRODUCT_ID | DISCOUNT_PERCENT |
|------------|------------------|
| 1 | 15 |
| 2 | 1 |
| 3 | 2 |
| 4 | 3 |
| 5 | 4 |
| 6 | 5 |

The interface also shows the 'Connections' pane on the left, listing connections 'ex' and 'mgs'. The 'Reports' pane is also visible, showing various report types. The bottom status bar indicates 'Line 1 Column 36 | Insert | Modified | Windows: CF'.

Updated Products Table:

The screenshot displays the Oracle SQL Developer interface. The main window shows a query executed in the 'Query Builder' tab. The query is: `SELECT PRODUCT_ID, DISCOUNT_PERCENT FROM Products`. The results are displayed in the 'Query Result' tab, showing 11 rows fetched in 0 seconds. The results table has two columns: `PRODUCT_ID` and `DISCOUNT_PERCENT`. The data is as follows:

| PRODUCT_ID | DISCOUNT_PERCENT |
|------------|------------------|
| 1 | 15 |
| 2 | 1 |
| 3 | 2 |
| 4 | 3 |
| 5 | 4 |
| 6 | 5 |

The interface also shows the 'Connections' pane on the left with a tree view of the database schema, including 'Tables (Filtered)', 'Views', 'Editioning Views', 'Indexes', 'Packages', and 'Procedures'. The 'Reports' pane is also visible, showing various report types like 'Analytic View Reports', 'Data Dictionary Reports', 'Data Modeler Reports', 'OLAP Reports', 'TimesTen Reports', and 'User Defined Reports'.