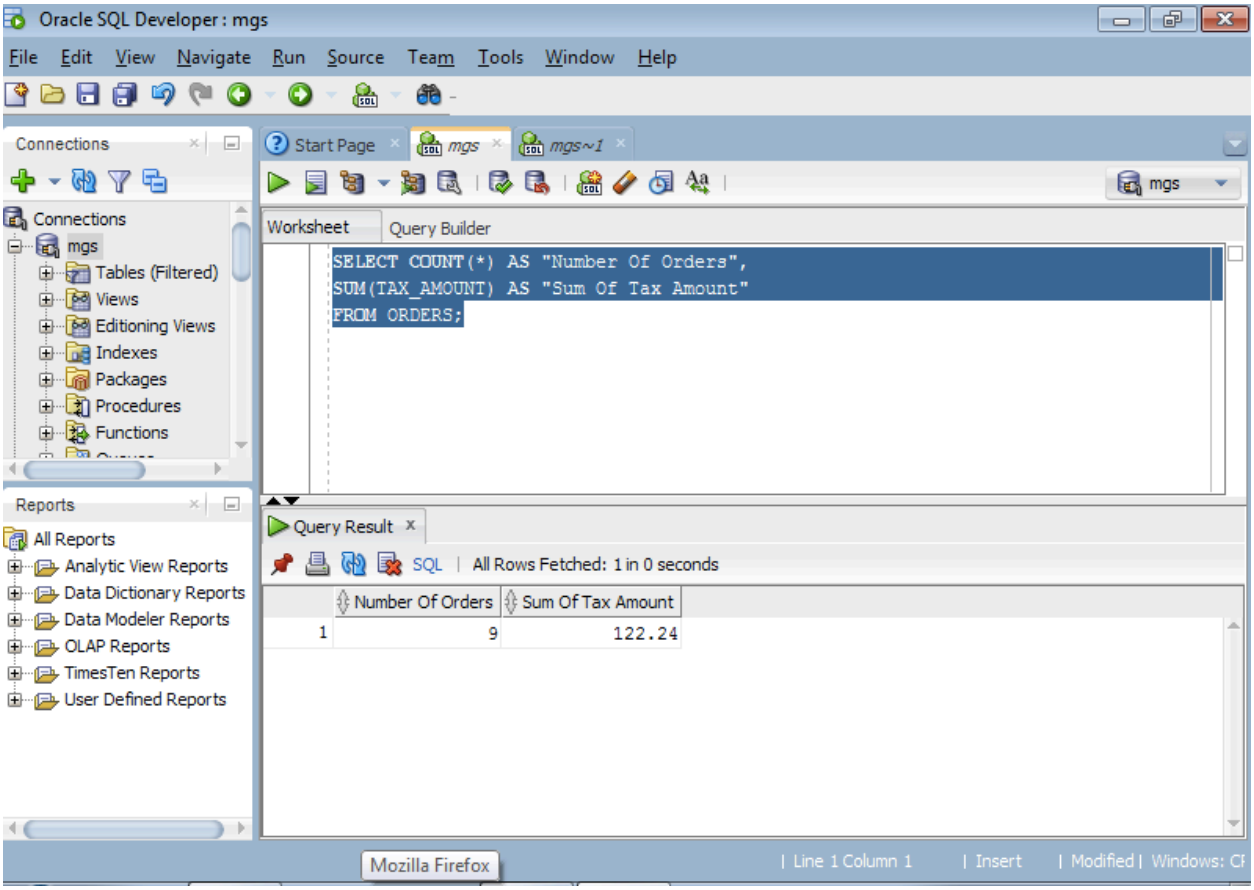
Domas Budrys Assignment 3

CHAPERT 5:

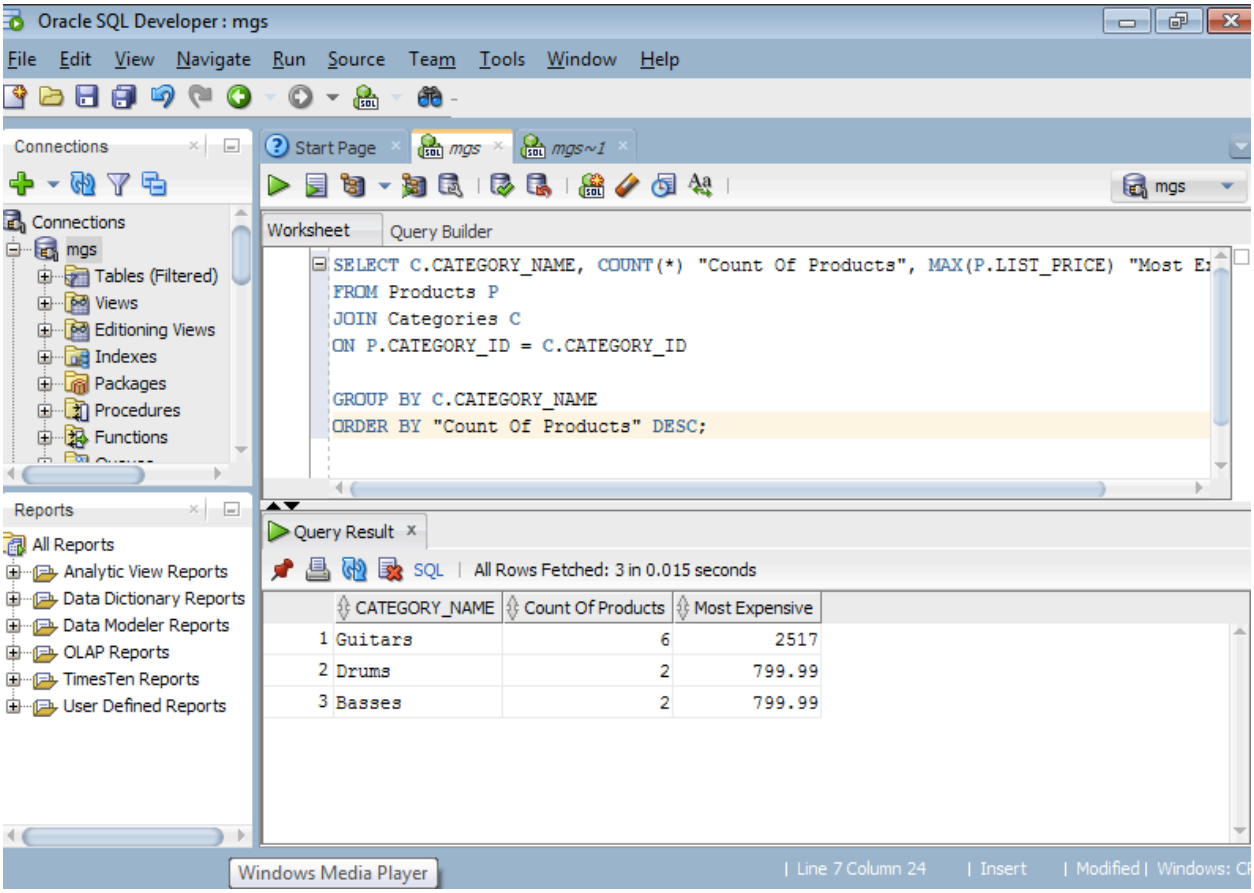
1.

```
SELECT COUNT(*) "Number Of Orders",
SUM(TAX_AMOUNT) "Sum Of Tax Amount"
FROM ORDERS;
```



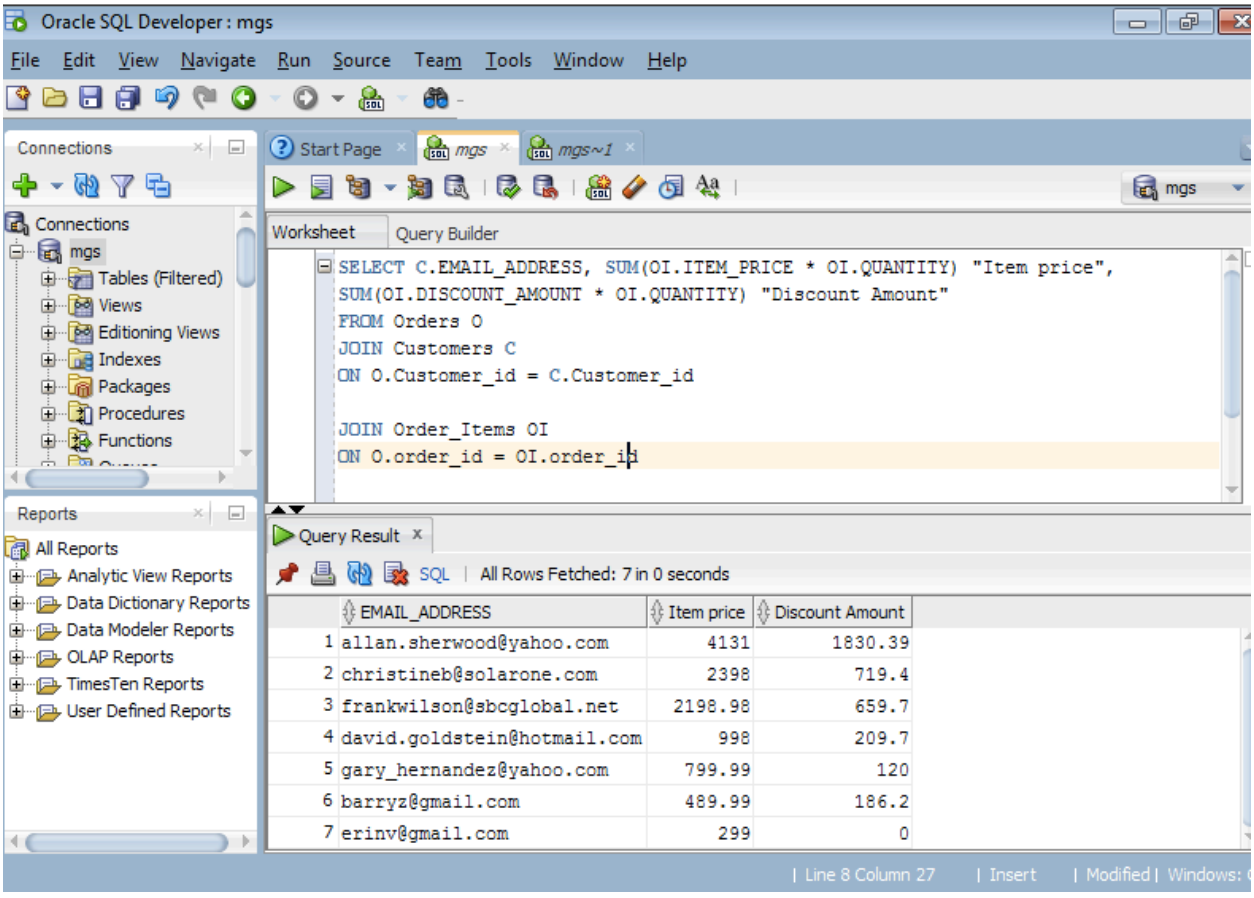
2.

```
SELECT C.CATEGORY_NAME, COUNT(*) "Count Of Products", MAX(P.LIST_PRICE) "Most
Expensive"
FROM Products P
JOIN Categories C
ON P.CATEGORY_ID = C.CATEGORY_ID

GROUP BY C.CATEGORY_NAME
ORDER BY "Count Of Products" DESC;
```



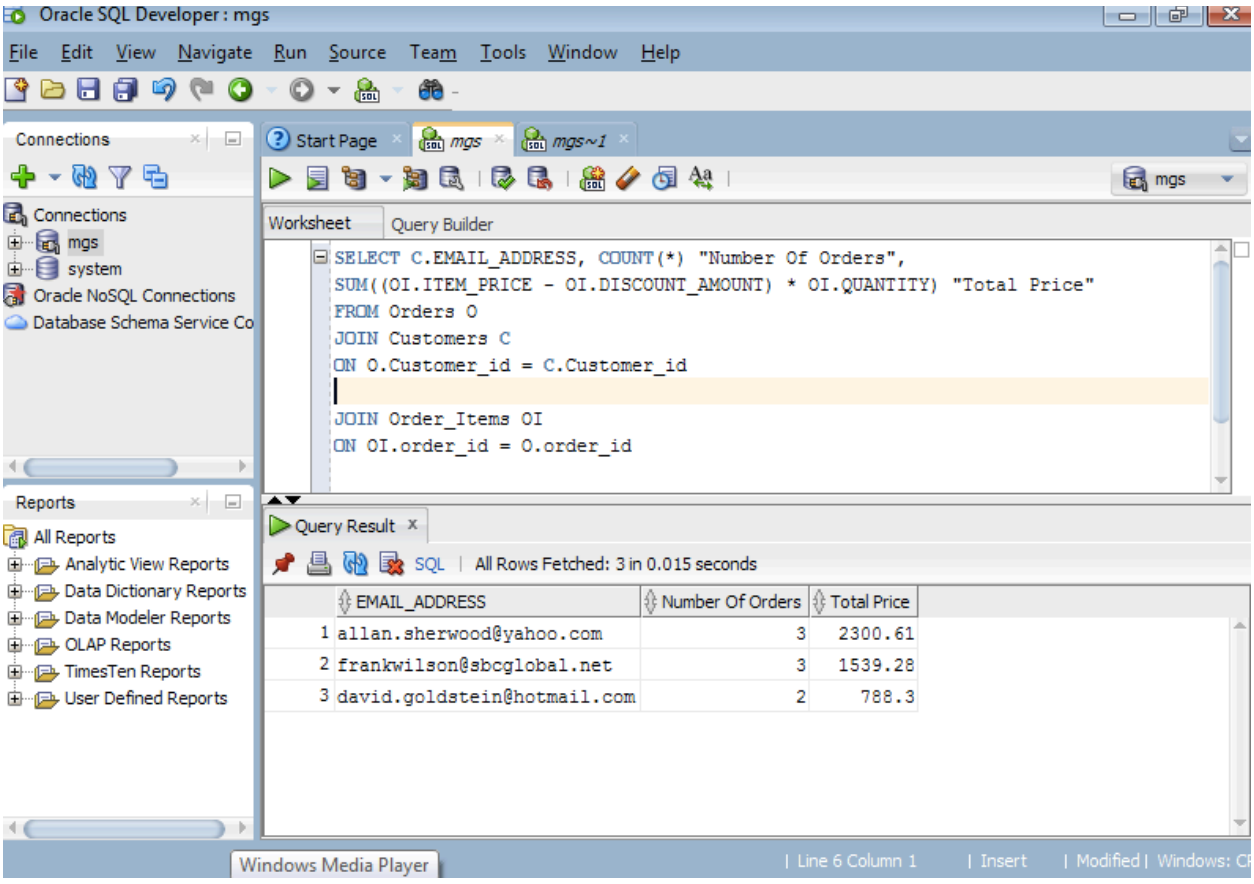
3.

```
SELECT C.EMAIL_ADDRESS, SUM(OI.ITEM_PRICE * OI.QUANTITY) "Item price",
SUM(OI.DISCOUNT_AMOUNT * OI.QUANTITY) "Discount Amount"
FROM Orders O
JOIN Customers C
ON O.Customer_id = C.Customer_id

JOIN Order_Items OI
ON O.order_id = OI.order_id

GROUP BY C.EMAIL_ADDRESS
ORDER BY "Item price" - "Discount Amount" DESC;
```



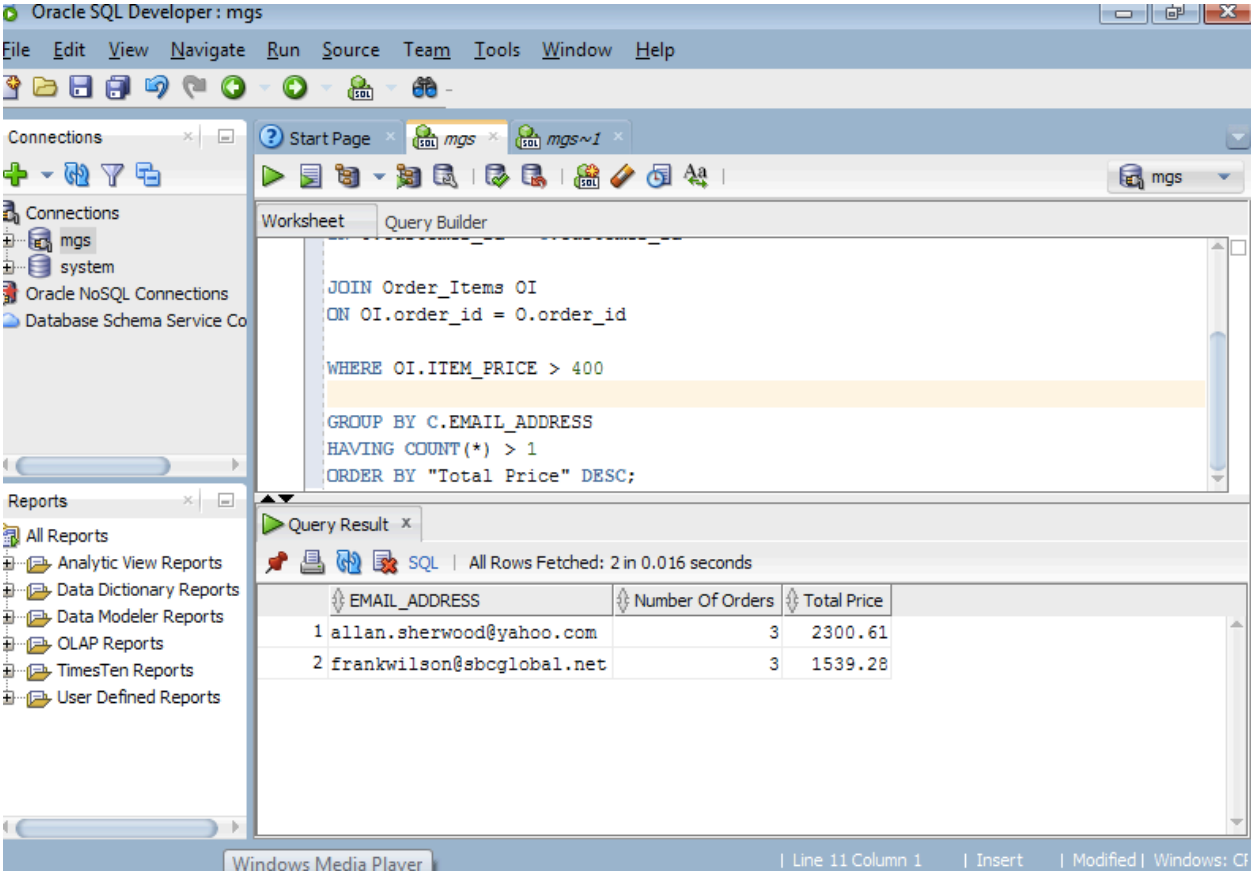
4.

```
SELECT C.EMAIL_ADDRESS, COUNT(*) "Number Of Orders",
SUM((OI.ITEM_PRICE - OI.DISCOUNT_AMOUNT) * OI.QUANTITY) "Total Price"
FROM Orders O
JOIN Customers C
ON O.Customer_id = C.Customer_id

JOIN Order_Items OI
ON OI.order_id = O.order_id

GROUP BY C.EMAIL_ADDRESS
HAVING COUNT(*) > 1
ORDER BY "Total Price" DESC;
```



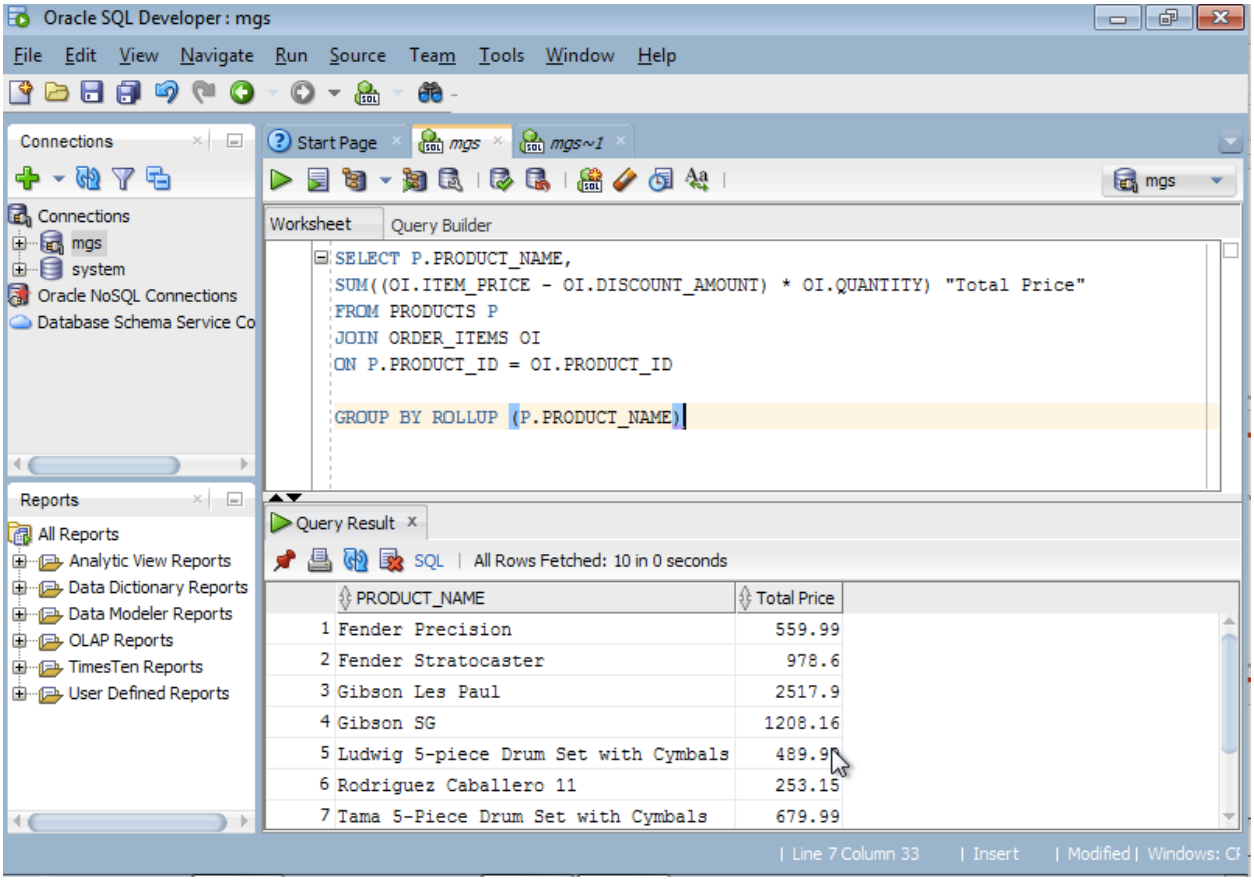
5.

```
SELECT C.EMAIL_ADDRESS, COUNT(*) "Number Of Orders",
SUM((OI.ITEM_PRICE - OI.DISCOUNT_AMOUNT) * OI.QUANTITY) "Total Price"
FROM Orders O
JOIN Customers C
ON O.Customer_id = C.Customer_id

JOIN Order_Items OI
ON OI.order_id = O.order_id

WHERE OI.ITEM_PRICE > 400

GROUP BY C.EMAIL_ADDRESS
HAVING COUNT(*) > 1
ORDER BY "Total Price" DESC;
```



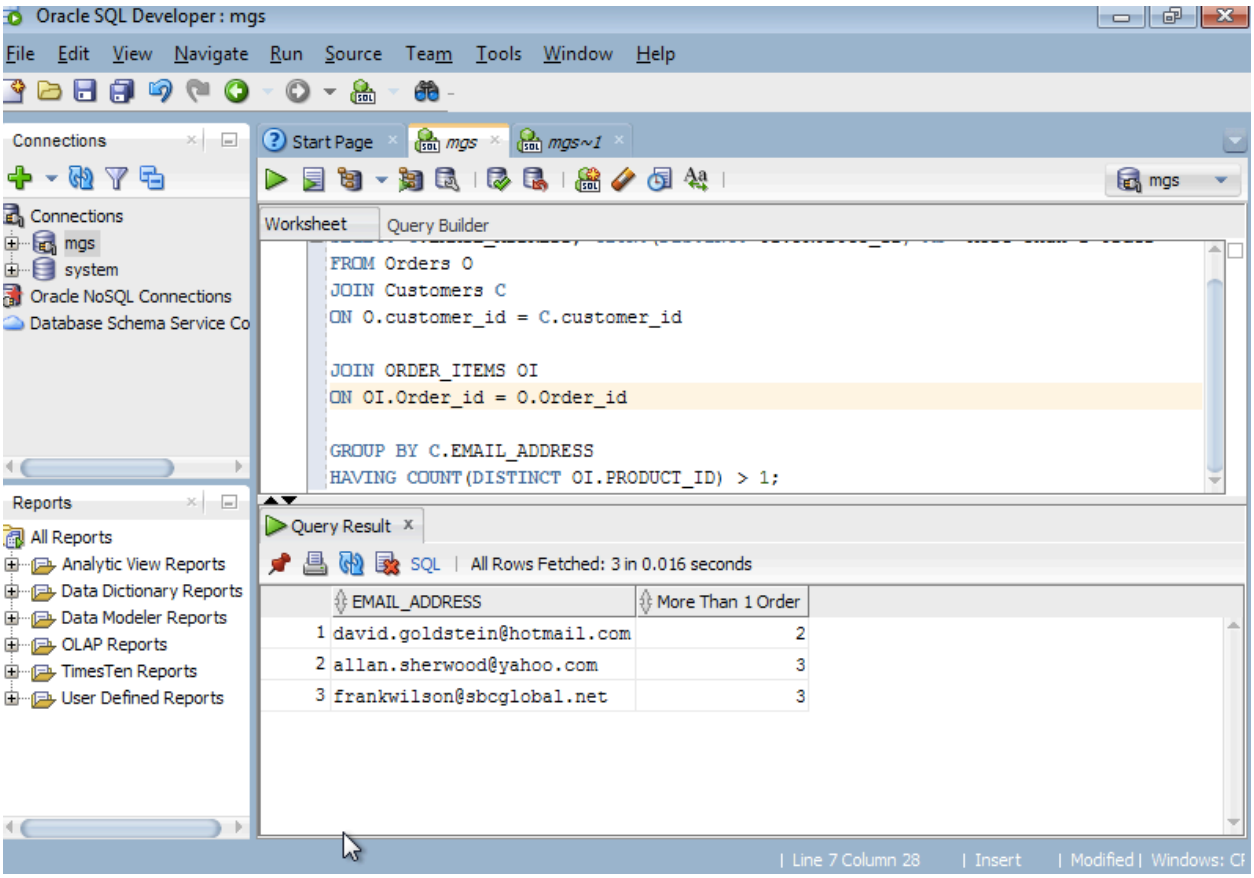
6.

```
SELECT P.PRODUCT_NAME,
SUM((OI.ITEM_PRICE - OI.DISCOUNT_AMOUNT) * OI.QUANTITY) "Total Price"
FROM PRODUCTS P
JOIN ORDER_ITEMS OI
ON P.PRODUCT_ID = OI.PRODUCT_ID

GROUP BY ROLLUP (P.PRODUCT_NAME)
```



7.

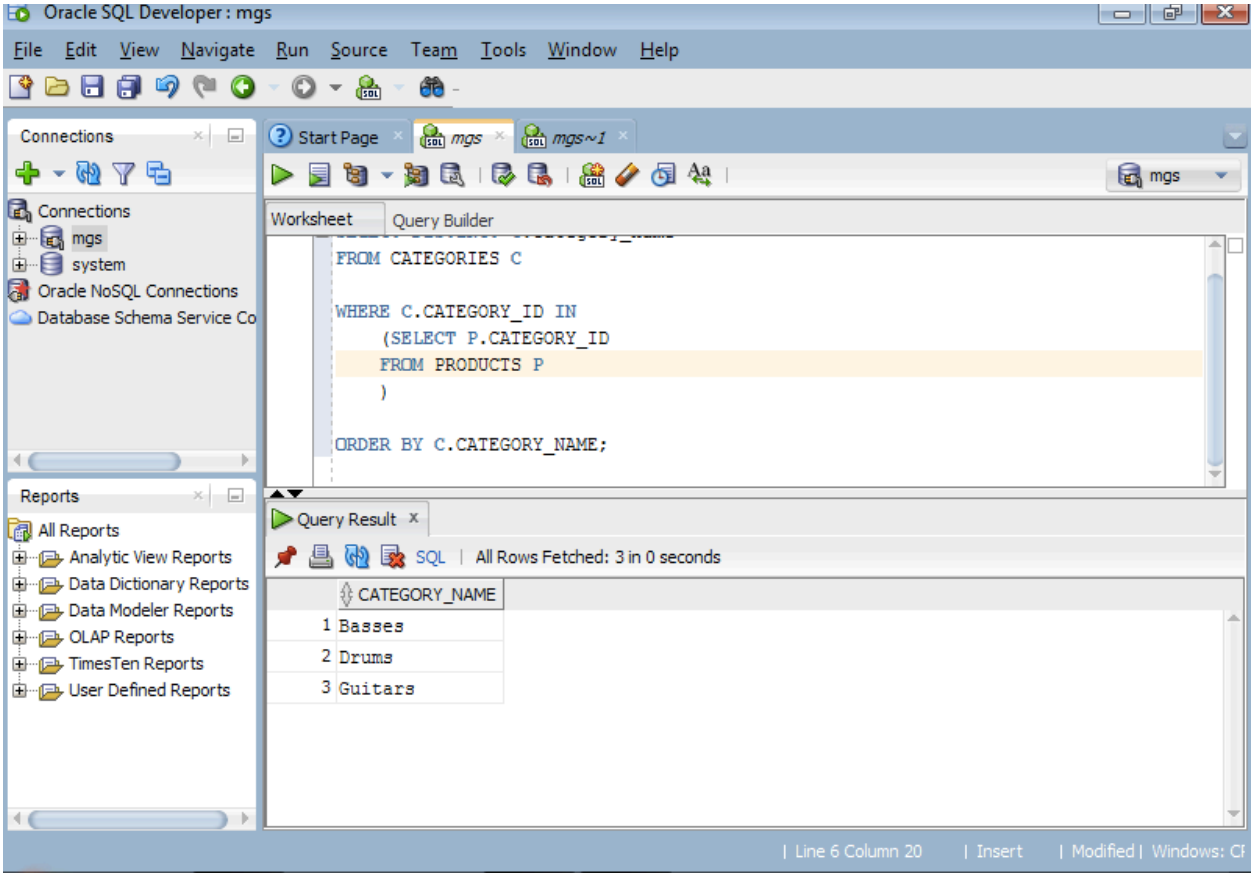
```
SELECT C.EMAIL_ADDRESS, COUNT(DISTINCT OI.PRODUCT_ID) AS "More Than 1 Order"
FROM Orders O
JOIN Customers C
ON O.customer_id = C.customer_id

JOIN ORDER_ITEMS OI
ON OI.Order_id = O.Order_id

GROUP BY C.EMAIL_ADDRESS
HAVING COUNT(DISTINCT OI.PRODUCT_ID) > 1;
```



CHAPERT 6:

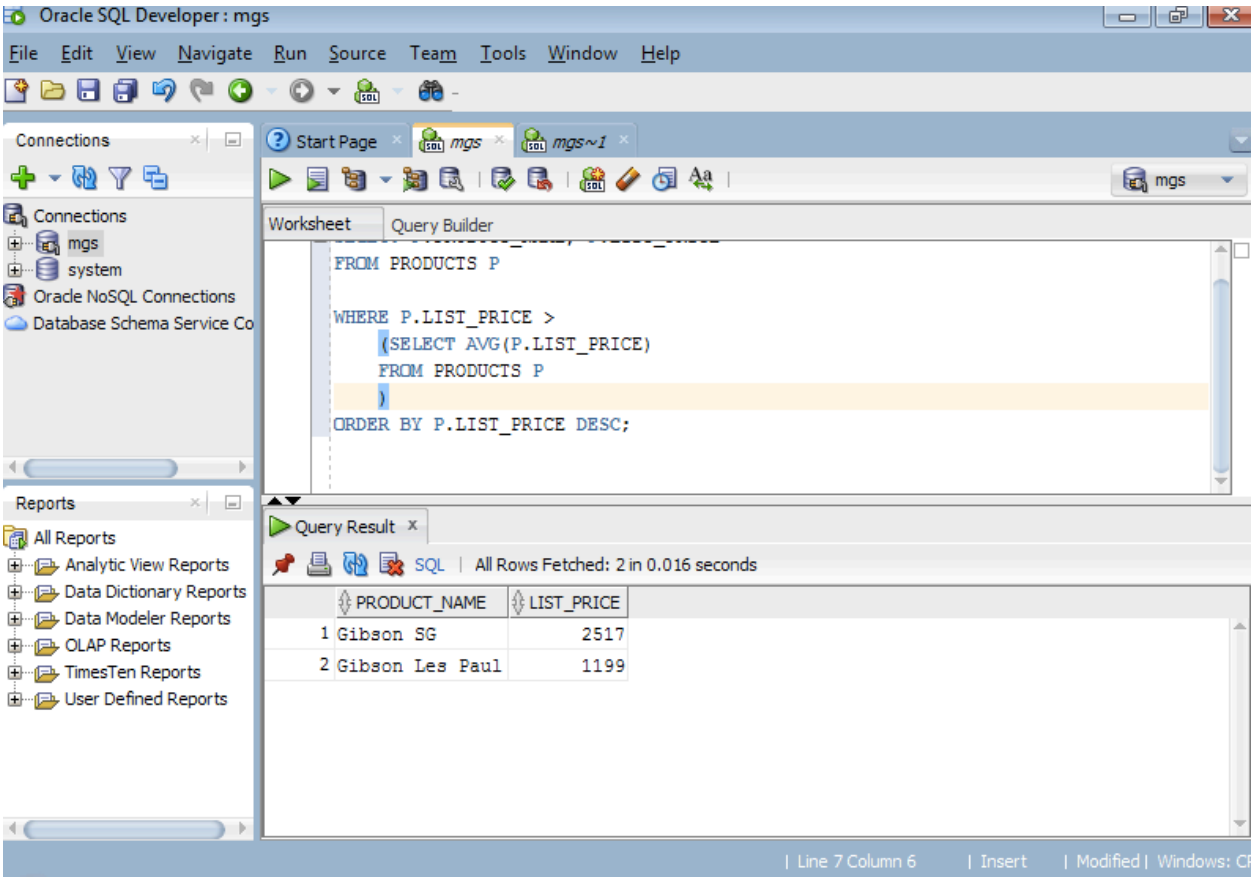
8.

```
SELECT DISTINCT C.category_name
FROM CATEGORIES C

WHERE C.CATEGORY_ID IN
   (SELECT P.CATEGORY_ID
   FROM PRODUCTS P
   )

ORDER BY C.CATEGORY_NAME;
```



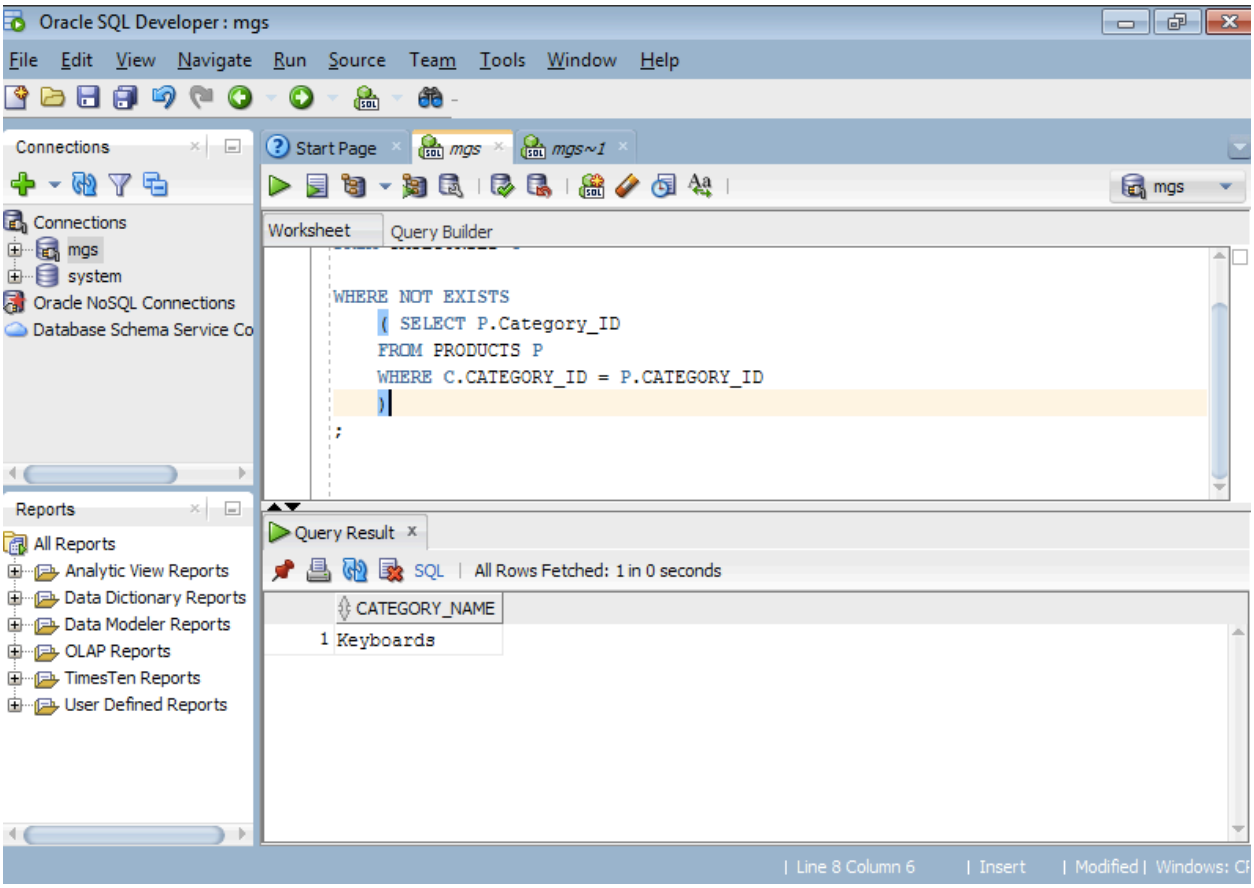
9.

```
SELECT P.PRODUCT_NAME, P.LIST_PRICE
FROM PRODUCTS P

WHERE P.LIST_PRICE >
   (SELECT AVG(P.LIST_PRICE)
   FROM PRODUCTS P
   )
ORDER BY P.LIST_PRICE DESC;
```



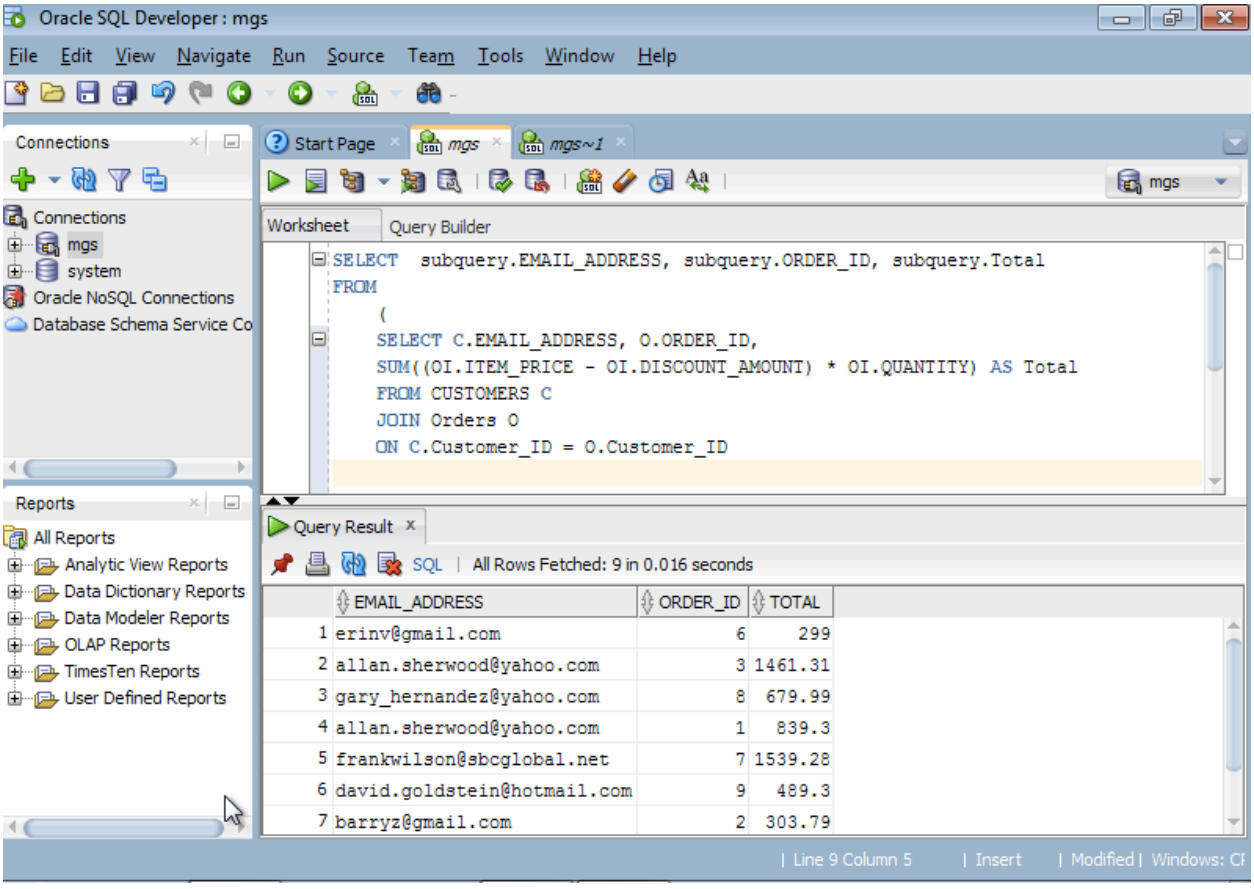
10.

```
SELECT C.Category_name
FROM CATEGORIES C

WHERE NOT EXISTS
   ( SELECT P.Category_ID
   FROM PRODUCTS P
   WHERE C.CATEGORY_ID = P.CATEGORY_ID
   )
;
```



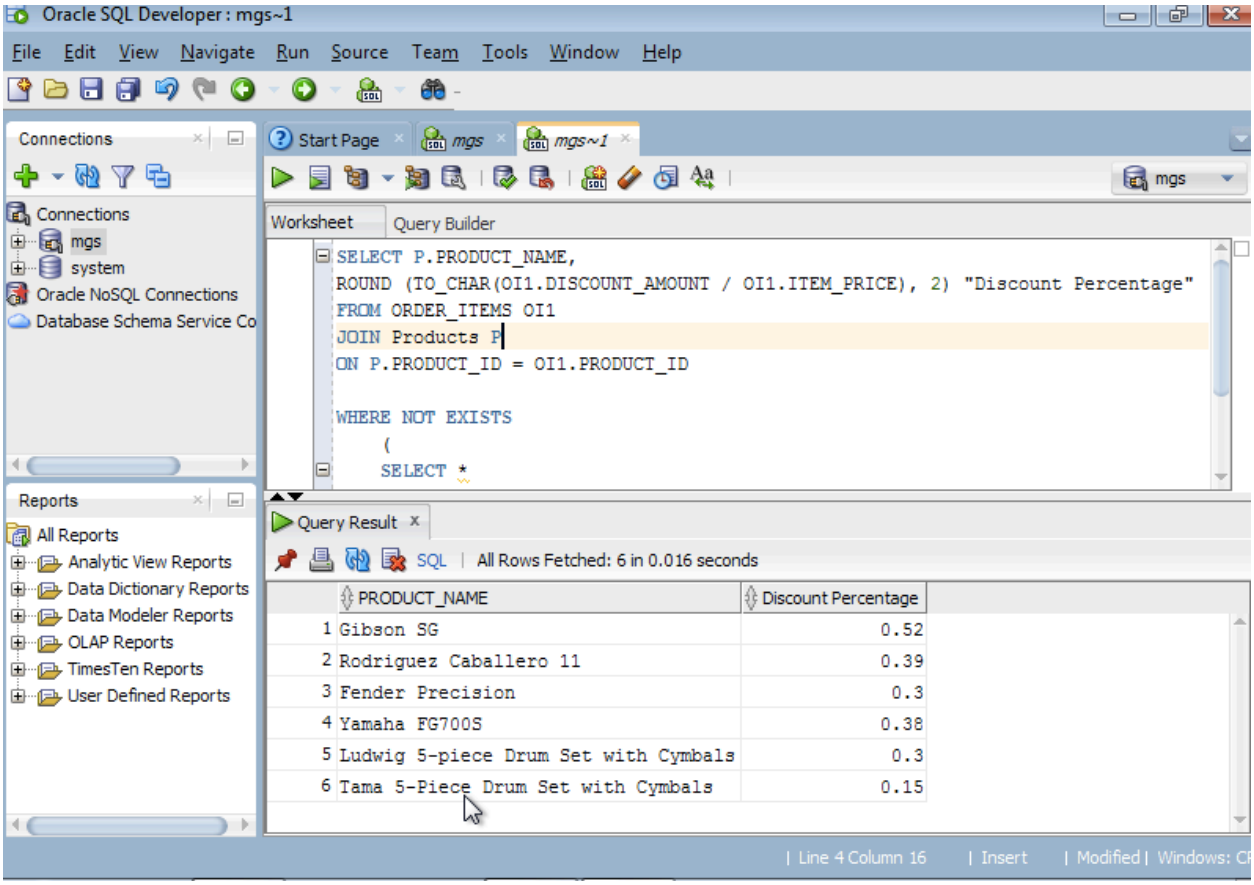
11.

```
SELECT  subquery.EMAIL_ADDRESS, subquery.ORDER_ID, subquery.Total
FROM
   (
   SELECT C.EMAIL_ADDRESS, O.ORDER_ID,
   SUM((OI.ITEM_PRICE - OI.DISCOUNT_AMOUNT) * OI.QUANTITY) AS Total
   FROM CUSTOMERS C
   JOIN Orders O
   ON C.Customer_ID = O.Customer_ID

   JOIN ORDER_ITEMS OI
   ON O.Order_ID = OI.Order_ID


   GROUP BY C.EMAIL_ADDRESS, O.ORDER_ID
   ) subquery;
```



12.

```
SELECT P.PRODUCT_NAME,
ROUND (TO_CHAR(OI1.DISCOUNT_AMOUNT / OI1.ITEM_PRICE), 2) "Discount Percentage"
FROM ORDER_ITEMS OI1
JOIN Products P
ON P.PRODUCT_ID = OI1.PRODUCT_ID

WHERE NOT EXISTS
  (
  SELECT *
  FROM ORDER_ITEMS OI2
  WHERE OI1.DISCOUNT_AMOUNT = OI2.DISCOUNT_AMOUNT
  AND
  OI1.ORDER_ID <> OI2.ORDER_ID
  );
```



13.

```
SELECT C.EMAIL_ADDRESS, O.ORDER_ID,
   (
   SELECT MIN(O.ORDER_DATE)
   FROM Orders O
   WHERE C.CUSTOMER_ID = O.CUSTOMER_ID
   ) AS "Earliest Date"

FROM CUSTOMERS C
JOIN ORDERS O
ON C.CUSTOMER_ID = O.CUSTOMER_ID;
```