

Domas Budrys Assignment 3

CHAPERT 5:

1.

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

The screenshot displays the Oracle SQL Developer interface. The main window shows a query in the 'Query Builder' tab. The query is:

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

Below the query, the 'Query Result' tab shows the results of the query. The results are displayed in a table with two columns: 'Number Of Orders' and 'Sum Of Tax Amount'. The table contains one row of data.

	Number Of Orders	Sum Of Tax Amount
1	9	122.24

The status bar at the bottom indicates 'Line 1 Column 1 | Insert | Modified | Windows: CF'.

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;
```

The screenshot shows the Oracle SQL Developer interface. The main window displays a SQL query in the Query Builder. The query is as follows:

```
SELECT C.CATEGORY_NAME, COUNT(*) "Count Of Products", MAX(P.LIST_PRICE) "Most E  
FROM Products P  
JOIN Categories C  
ON P.CATEGORY_ID = C.CATEGORY_ID  
  
GROUP BY C.CATEGORY_NAME  
ORDER BY "Count Of Products" DESC;
```

The Query Result pane shows the results of the query, which are 3 rows fetched in 0.015 seconds. The results are displayed in a table with the following columns: CATEGORY_NAME, Count Of Products, and Most Expensive.

	CATEGORY_NAME	Count Of Products	Most Expensive
1	Guitars	6	2517
2	Drums	2	799.99
3	Basses	2	799.99

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;
```

The screenshot displays the Oracle SQL Developer interface. The 'Connections' pane on the left shows a connection to 'mgs'. The 'Query Builder' pane in the center contains the following SQL query:

```
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
```

The 'Query Result' pane at the bottom shows the results of the query, with 7 rows fetched in 0 seconds. The results are as follows:

	EMAIL_ADDRESS	Item price	Discount Amount
1	allan.sherwood@yahoo.com	4131	1830.39
2	christineb@solarone.com	2398	719.4
3	frankwilson@sbcglobal.net	2198.98	659.7
4	david.goldstein@hotmail.com	998	209.7
5	gary_hernandez@yahoo.com	799.99	120
6	barryz@gmail.com	489.99	186.2
7	erinv@gmail.com	299	0

The status bar at the bottom indicates 'Line 8 Column 27', 'Insert' mode, and 'Modified | Windows: C'.

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;
```

The screenshot shows the Oracle SQL Developer interface. The main window displays a SQL query in the Worksheet tab. The query is as follows:

```
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
```

The Query Result tab shows the results of the query, which are 3 rows. The columns are EMAIL_ADDRESS, Number Of Orders, and Total Price. The results are as follows:

	EMAIL_ADDRESS	Number Of Orders	Total Price
1	allan.sherwood@yahoo.com	3	2300.61
2	frankwilson@sbcglobal.net	3	1539.28
3	david.goldstein@hotmail.com	2	788.3

The interface also shows the Connections pane on the left with 'mgs' and 'system' connections. The Reports pane is also visible on the left. The status bar at the bottom indicates 'Line 6 Column 1' and 'Insert' mode.

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;
```

The screenshot shows the Oracle SQL Developer interface. The main window displays a SQL query in the Query Builder. The query is as follows:

```
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;
```

The Query Result pane shows the results of the query, which are 2 rows. The columns are EMAIL_ADDRESS, Number Of Orders, and Total Price.

	EMAIL_ADDRESS	Number Of Orders	Total Price
1	allan.sherwood@yahoo.com	3	2300.61
2	frankwilson@sbcglobal.net	3	1539.28

The status bar at the bottom indicates "Line 11 Column 1 | Insert | Modified | Windows: CF".

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)
```

The screenshot shows the Oracle SQL Developer interface. The main window displays a SQL query in the Worksheet tab. The query is as follows:

```
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)
```

The Query Result tab shows the results of the query. The results are displayed in a table with two columns: PRODUCT_NAME and Total Price. The table contains 7 rows of data.

PRODUCT_NAME	Total Price
1 Fender Precision	559.99
2 Fender Stratocaster	978.6
3 Gibson Les Paul	2517.9
4 Gibson SG	1208.16
5 Ludwig 5-piece Drum Set with Cymbals	489.9
6 Rodriguez Caballero 11	253.15
7 Tama 5-Piece Drum Set with Cymbals	679.99

The status bar at the bottom indicates the cursor is at Line 7 Column 33, and the window is titled "mgs".

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;
```

The screenshot shows the Oracle SQL Developer interface. The main window displays a SQL query in the Worksheet tab. The query is as follows:

```
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;
```

The Query Result tab shows the results of the query. The results are displayed in a table with two columns: EMAIL_ADDRESS and More Than 1 Order. The table contains three rows of data:

EMAIL_ADDRESS	More Than 1 Order
david.goldstein@hotmail.com	2
allan.sherwood@yahoo.com	3
frankwilson@sbcglobal.net	3

The status bar at the bottom indicates the cursor is at Line 7 Column 28. The bottom right corner shows the status: Insert | Modified | Windows: CF.

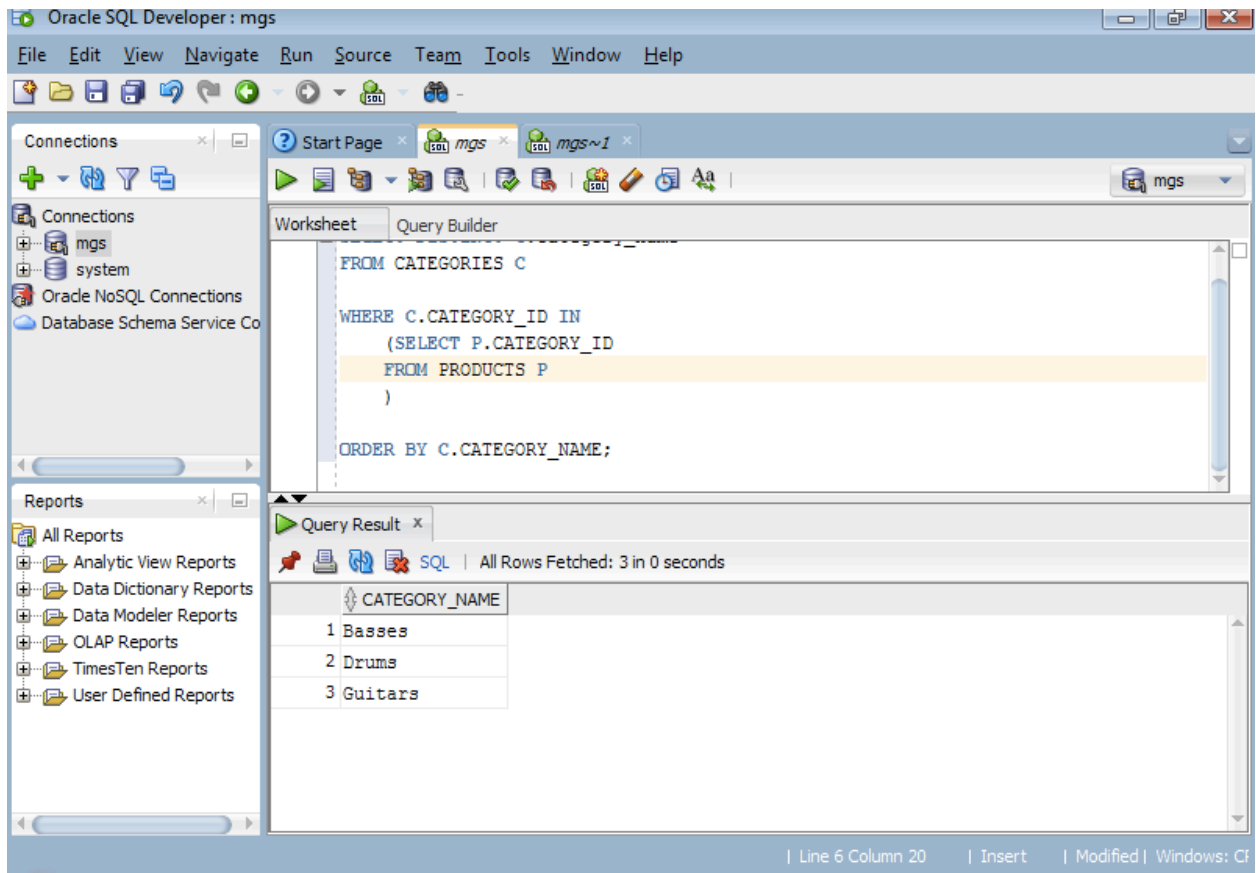
CHAPERT 6:

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;
```



The screenshot displays the Oracle SQL Developer interface. The main window shows a query in the Worksheet tab:

```
FROM CATEGORIES C  
  
WHERE C.CATEGORY_ID IN  
      (SELECT P.CATEGORY_ID  
       FROM PRODUCTS P  
       )  
  
ORDER BY C.CATEGORY_NAME;
```

The Query Result tab shows the results of the query, displaying a table with the following data:

	CATEGORY_NAME
1	Basses
2	Drums
3	Guitars

The status bar at the bottom indicates "Line 6 Column 20 | Insert | Modified | Windows: CF".

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;
```

The screenshot shows the Oracle SQL Developer interface. The main window displays a SQL query in the Worksheet tab. The query is as follows:

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

The Query Result tab shows the results of the query. The results are displayed in a table with two columns: PRODUCT_NAME and LIST_PRICE. The table contains two rows of data:

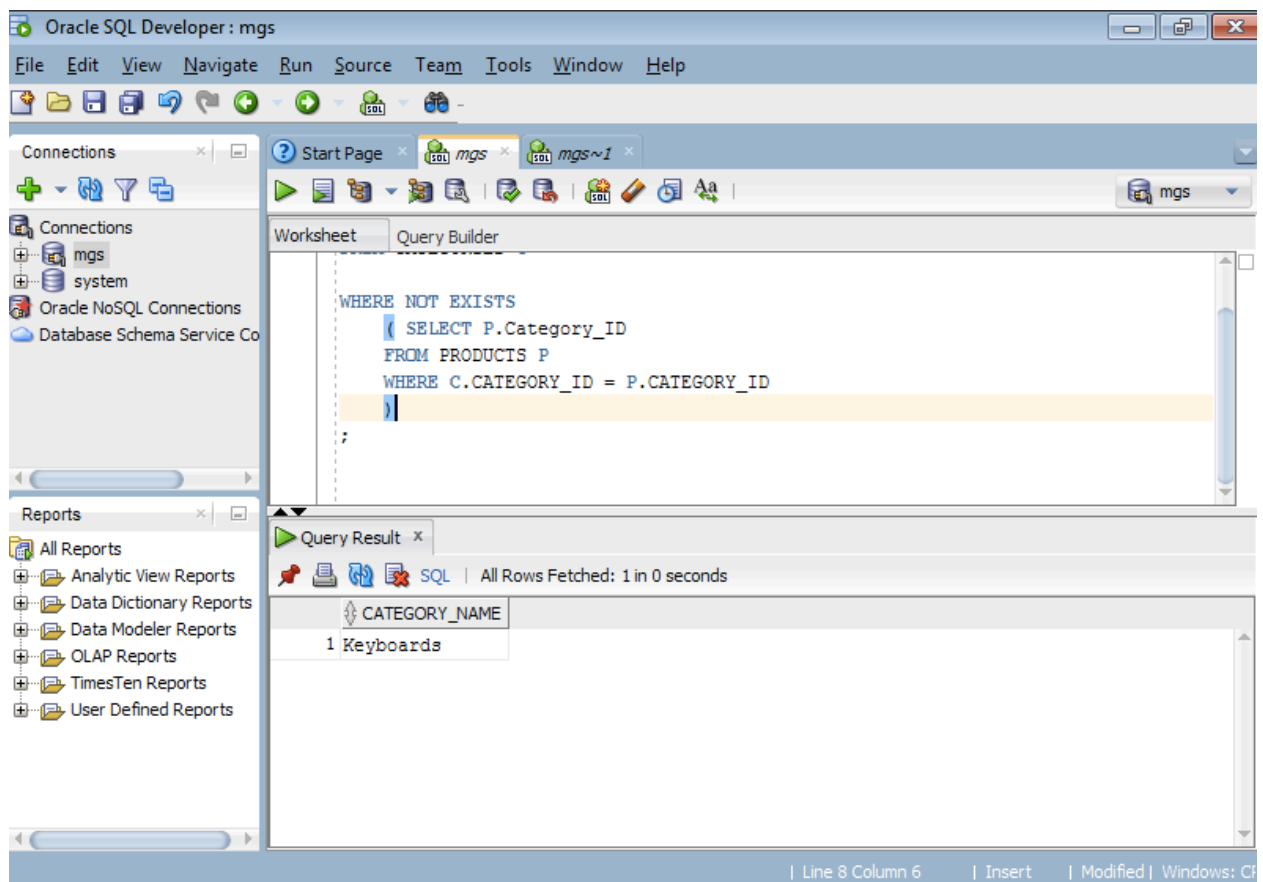
PRODUCT_NAME	LIST_PRICE
1 Gibson SG	2517
2 Gibson Les Paul	1199

The status bar at the bottom indicates the current position is Line 7 Column 6, and the status is Insert. The bottom right corner shows the status: Modified | Windows: C

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;
```

The screenshot displays the Oracle SQL Developer interface. The main window shows a query in the 'Query Builder' tab. The query is a subquery that selects email addresses, order IDs, and total amounts from the CUSTOMERS, Orders, and ORDER_ITEMS tables. The 'Query Result' tab shows the results of the query, which are 9 rows of data. The results are displayed in a table with columns: EMAIL_ADDRESS, ORDER_ID, and TOTAL. The status bar at the bottom indicates 'Line 9 Column 5 | Insert | Modified | Windows: CF'.

	EMAIL_ADDRESS	ORDER_ID	TOTAL
1	erinv@gmail.com	6	299
2	allan.sherwood@yahoo.com	3	1461.31
3	gary_hernandez@yahoo.com	8	679.99
4	allan.sherwood@yahoo.com	1	839.3
5	frankwilson@sbcglobal.net	7	1539.28
6	david.goldstein@hotmail.com	9	489.3
7	barryz@gmail.com	2	303.79

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  
);
```

The screenshot displays the Oracle SQL Developer interface. The main window shows a query in the 'Query Builder' tab. The query is as follows:

```
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  
);
```

The 'Query Result' tab shows the results of the query, which are 6 rows. The columns are 'PRODUCT_NAME' and 'Discount Percentage'.

PRODUCT_NAME	Discount Percentage
1 Gibson SG	0.52
2 Rodriguez Caballero 11	0.39
3 Fender Precision	0.3
4 Yamaha FG700S	0.38
5 Ludwig 5-piece Drum Set with Cymbals	0.3
6 Tama 5-Piece Drum Set with Cymbals	0.15

The status bar at the bottom indicates 'Line 4 Column 16 | Insert | Modified | Windows: C'.

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;
```

The screenshot shows the Oracle SQL Developer interface. The main window displays a query in the Worksheet tab. The query is as follows:

```
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;
```

The Query Result tab shows the results of the query. The results are displayed in a table with the following columns: EMAIL_ADDRESS, ORDER_ID, and Earliest Date. The results are as follows:

	EMAIL_ADDRESS	ORDER_ID	Earliest Date
1	allan.sherwood@yahoo.com	1	28-MAR-12
2	barryz@gmail.com	2	28-MAR-12
3	allan.sherwood@yahoo.com	3	28-MAR-12
4	christineb@solarone.com	4	30-MAR-12
5	david.goldstein@hotmail.com	5	31-MAR-12
6	erinv@gmail.com	6	31-MAR-12
7	frankwilson@sbcglobal.net	7	01-APR-12