

# Deliverable 3 - TEAM 5

Members:

Nandhu Unnikrishnan (Member 1)

Nauman Choudhary (Member 2)

Gia Bao Vo (Member 3)

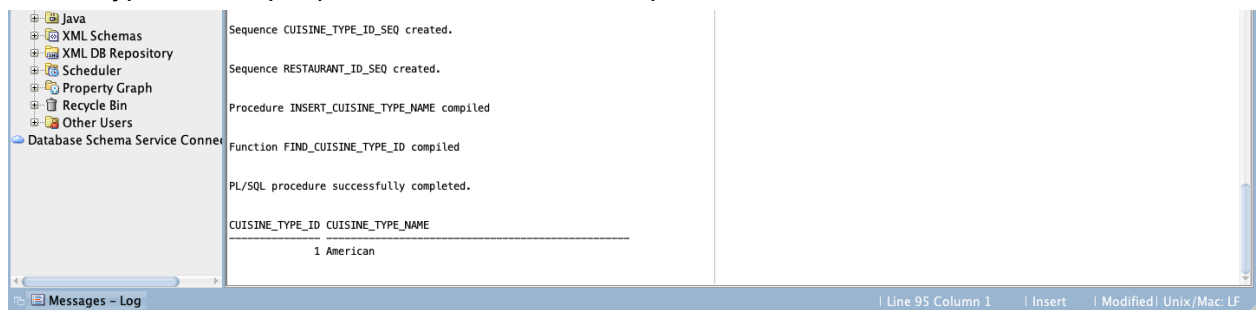
Sadia Alam (Member 4)

Shikha Gondalia (Member 5)

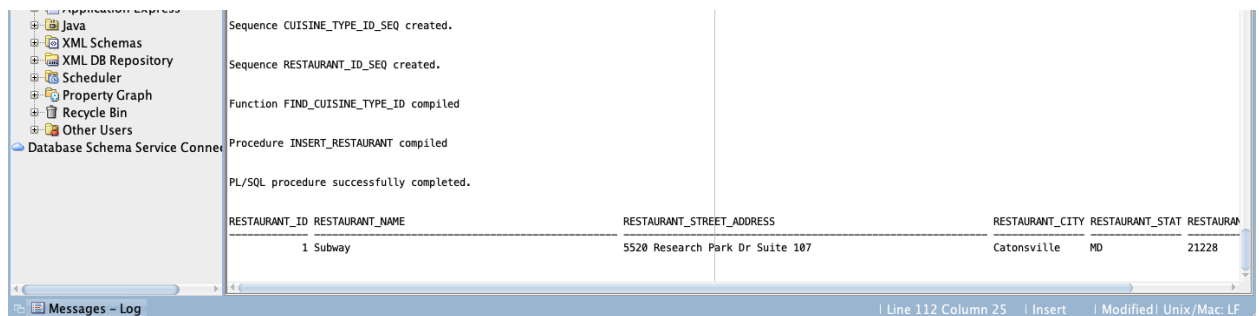
## Member 1: Nandhu Unnikrishnan

Responsible for the Cuisine Types and Restaurants tables. Must create these tables.

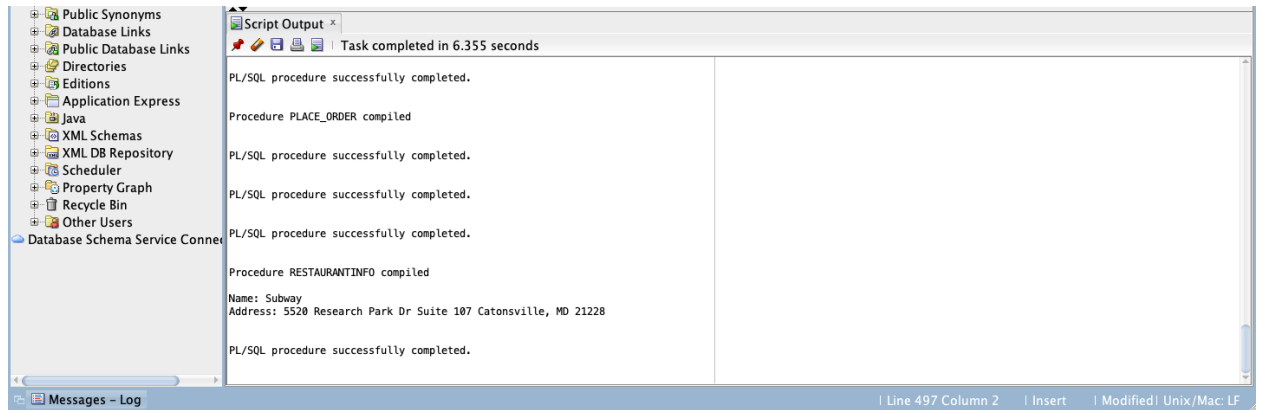
1. Add cuisine type: Given the name of a cuisine type, add it to the table. The cuisine type is the input parameter to the PL/SQL procedure.



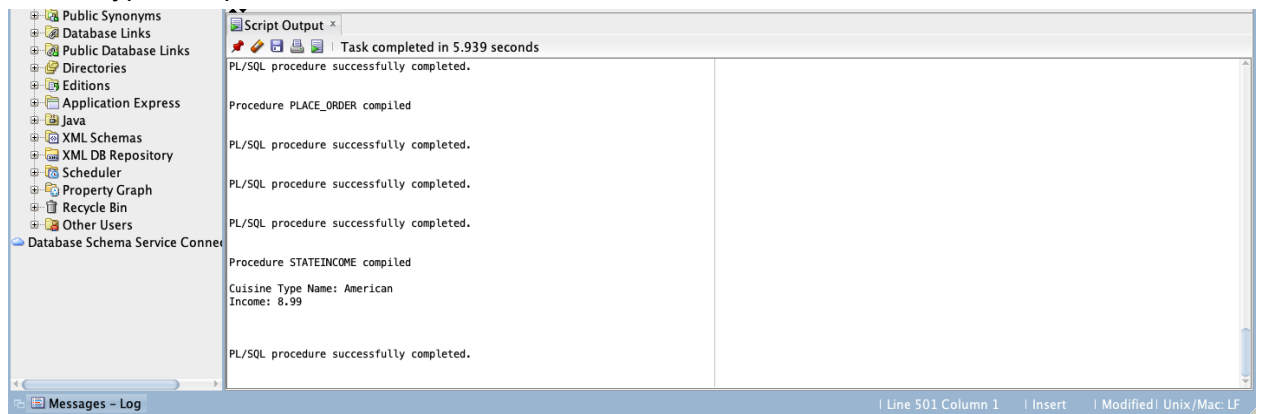
2. Add restaurant: Add a new restaurant in the table with all pertinent input information.



3. Display restaurant by cuisine: Given a cuisine type, show name and address about all restaurants that offer that cuisine.



4. Report Income by state. Generate a report that lists the income of restaurants per cuisine type and per state.



## Member 2: Nauman Choudhary

Responsible for the Waiters table. Must create the table.

5. Hire waiter: Given all pertinent information as parameters, hire a waiter at a restaurant. The waiter name and restaurant ID must be input parameters. Use the function FIND\_RESTAURANT\_ID (see helper functions below) first to get the rID.

WAITER_ID	WAITER_NAME	RESTAURANT_NAME
1	1 Nauman	HotDawg
2	2 Bao	HotDawg
3	3 Nandhu	Pizza Palace
4	4 Sadia	Windows Cookies
5	5 Shikha	Pizza Palace
6	6 John	Subway
7	7 Marry Johnson	Chick-fil-A
8	8 Tony Parker	Kabob Place

6. Show list of waiters: Given a restaurant ID (you need to call the appropriate helper function), show all info about each employee

RESTAURANT_ID	RESTAURANT_NAME	RESTAURANT_STREET_ADDRESS	RESTAURANT_CITY	RESTAURANT_STATE	RESTAURANT_ZIP	CUISINE_TYPE_ID
1	Subway	5520 Research Park Dr Suite 107	Catonsville	MD	21228	1
2	HotDawg	21 Westcott	Liverpool	CA	14156	3
3	Pizza Palace	45 Eastdoor	New York City	NY	12307	3
4	Windows Cookies	99 Southeast	New York City	NY	98425	1
5	Donald	1234	germantown	NY	12230	5
6	6rubby	23456	hewlett	MD	32124	4
7	Chick-fil-A	100 Hiltop Circle	Baltimore	MD	20904	1
8	Kabob Place	200 Melody Dr	Baltimore	MD	20904	2

7. Report tips: Show total tips by each waiter.

```

Procedure WAITER_TIPS compiled

Waiter ID          Waiter Name          Total TIP Earned
-----
1                  Marry Johnson          $2.80
2                  Tony Parker            $2.40

PL/SQL procedure successfully completed.

```

8. Report tips by state: Show total tips earned by waiters per state.

```

Procedure WAITER_ST_TIPS compiled

Waiter ID          Waiter Name          State          Total TIP Earned
-----
1                  Marry Johnson          MD              $2.80
2                  Tony Parker            MD              $2.40

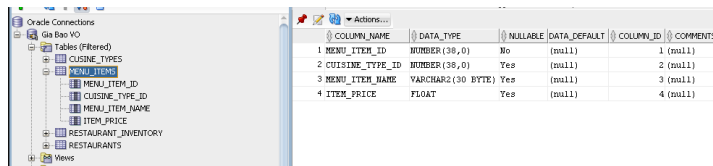
PL/SQL procedure successfully completed.

```

### Member 3: Gia Bao Vo

Responsible for the Menu Items and Restaurant Inventory tables. Must create these tables.

9. Create menu item: Given a cuisine type id, create a menu item (name and price) for that cuisine type. Use the function FIND\_CUISINE\_TYPE\_ID (see helper functions below) first to get the ID.



COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
MENU_ITEM_ID	NUMBER(38,0)	No	(null)	1	(null)
CUISINE_TYPE_ID	NUMBER(38,0)	Yes	(null)	2	(null)
MENU_ITEM_NAME	VARCHAR2(30 BYTE)	Yes	(null)	3	(null)
ITEM_PRICE	FLOAT	Yes	(null)	4	(null)

SQL Worksheet History

Worksheet Query Builder

```
select*from restaurant_inventory;
select*from menu_items;

--Use the function FIND_CUISINE_TYPE_ID (see helper functions below) first to get the ID.
CREATE OR REPLACE FUNCTION find_cuisine_type_id (
  cname IN VARCHAR
) RETURN int IS
  ctype_id int;
BEGIN
  SELECT
    cuisine_type_id
  INTO ctype_id
  FROM
    cuisine_types
  WHERE
    cuisine_type_name = cname;

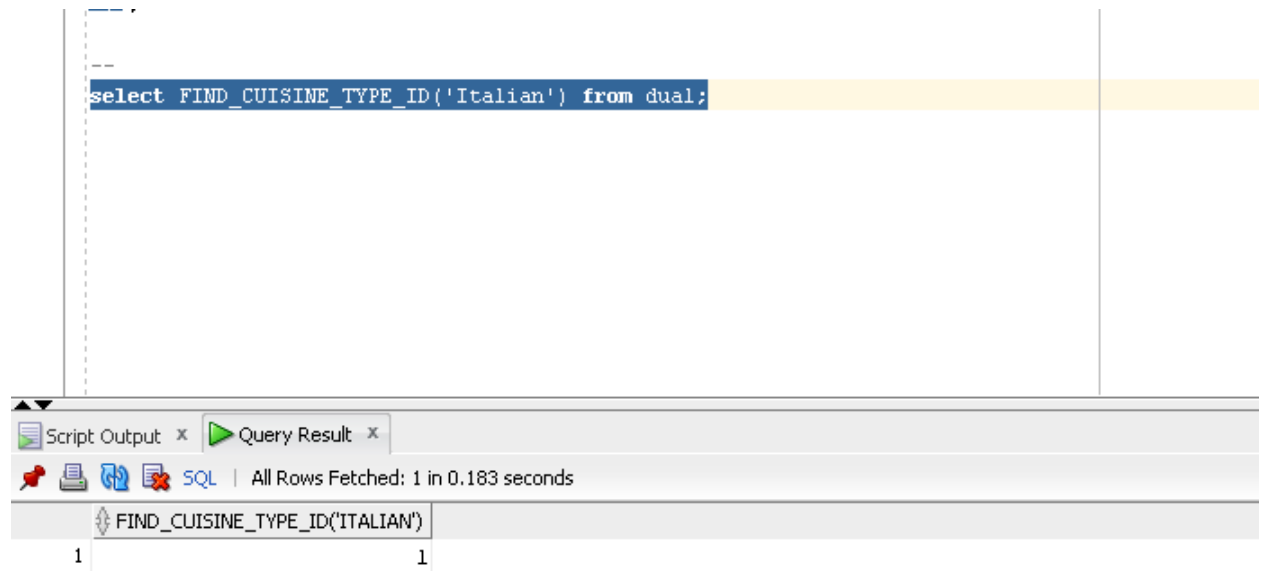
  RETURN ctype_id;
EXCEPTION
  WHEN no_data_found THEN
    dbms_output.put_line('No data Found, this Cuisine is not in database');
END;

--
select FIND_CUISINE_TYPE_ID('Italian') from dual;
```

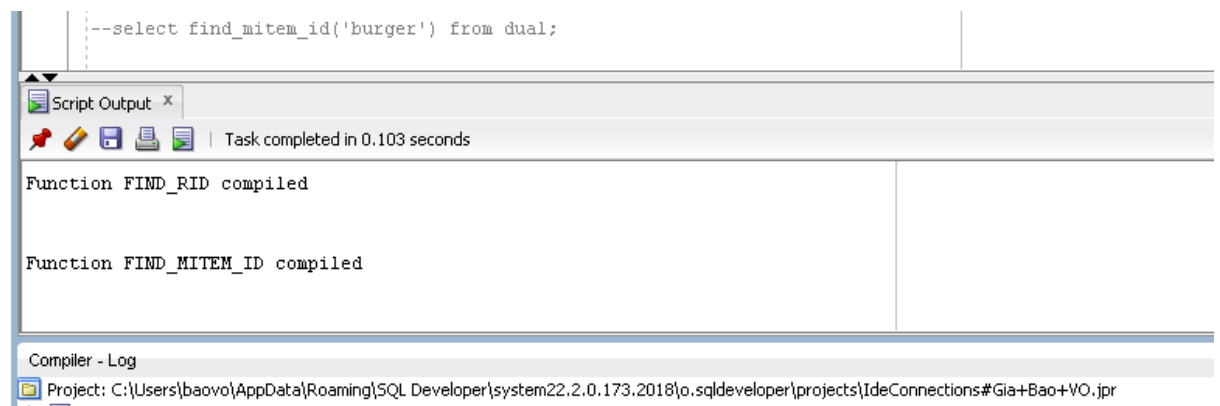
Script Output x Query Result x

Task completed in 0.105 seconds

Function FIND\_CUISINE\_TYPE\_ID compiled



10. Add menu item to Restaurant Inventory: Given all pertinent information, add a menu item with a given quantity to a given restaurant in the Restaurant Inventory table. You will need to call helper functions to find IDs (see helper functions below).



Gia Bao VO7.sql x Gia Bao VO~2.sql x Gia Bao VO10.sql x Gia Bao VO9.s

SQL Worksheet History

Worksheet Query Builder

```
--there is an order of an item.
```

```
CREATE OR REPLACE PROCEDURE add_into_restaurant_inventory (  
    menuID int,  
    restaurantID int,  
    meIT_name IN VARCHAR,  
    I_quant IN NUMBER  
) IS  
  
BEGIN  
    INSERT into restaurant_inventory  
    (menu_item_id,  
    restaurant_id ,  
    menu_item_name ,  
    quantity )  
values  
    (menuID ,  
    restaurantID,  
    meIT_name,  
    I_quant);  
  
Exception  
When others Then  
    dbms_output.put_line('Error Found While Inserting data');  
End add_into_restaurant_inventory;
```

Script Output x

Task completed in 0.148 seconds

Procedure ADD\_INT0\_RESTAURANT\_INVENTORY compiled

SQL Worksheet | History

Worksheet | Query Builder

```

restaurant_id ,
menu_item_name ,
quantity      )
values
(menuID ,
restaurantID,
menuItem_name,
I_quant);

Exception
When others Then
dbms_output.put_line('Error Found While Inserting data');
End add_into_restaurant_inventory;

-- testing to see if its added into inventory

exec add_into_restaurant_inventory(find_mitem_id('burger'),find_rid('Donald'), 'burger', 50);
exec add_into_restaurant_inventory(find_mitem_id('fries'),find_rid('Donald'), 'fries', 150);
exec add_into_restaurant_inventory(find_mitem_id('lasagna'),find_rid('rubby'), 'lasagna', 10);
exec add_into_restaurant_inventory(find_mitem_id('meatballs'),find_rid('rubby'), 'meatballs', 5);
exec add_into_restaurant_inventory(find_mitem_id('steak'),find_rid('Donald'), 'steak', 2);

select*from restaurant_inventory;

---Report Menu items: Generate a report to show totals of each menu item by type of cuisine.

```

Script Output | Query Result

SQL | All Rows Fetched: 5 in 0.029 seconds

	MENU_ITEM_ID	RESTAURANT_ID	MENU_ITEM_NAME	QUANTITY
1	12	1	steak	2
2	1	2	burger	50
3	2	6	fries	150
4	6	7	lasagna	10
5	7		meatballs	5

11. Update menu item inventory: Given a restaurant id, a menu item id, along with a given quantity, reduce the inventory of that menu item by the amount specified by the quantity. This is to keep the inventory updated every time there is an order of an item.

Gia Bao VO7.sql x Gia Bao VO~2.sql x Gia Bao VO10.sql x Gia Bao VO9.sql x Gia Bao VO~11.sql x

SQL Worksheet History

Worksheet Query Builder

```
--Add menu item to Restaurant Inventory: Given all pertinent information
--add a menu item with a given quantity to a given restaurant
--in the Restaurant Inventory table.

CREATE OR REPLACE PROCEDURE add_into_menu_items (
    c_name          VARCHAR,
    menu_item_name  IN VARCHAR,
    price           IN NUMBER
) IS
BEGIN
    Insert into menu_items
    ( menu_item_id,
      cuisine_type_id,
      menu_item_name,
      item_price)
    values
    (menu_id_seq.nextval,c_name,menu_item_name,price);

    exception
    when others then
        dbms_output.put_line('Error Found While Inserting data');
End add_into_menu_items;

--test to see if its added

exec add_into_menu_items(find_cuisine_type_id('Italian'), 'burger', '12');
exec add_into_menu_items(find_cuisine_type_id('Italian'), 'fries', '4');
exec add_into_menu_items(find_cuisine_type_id('Italian'), 'pasta', '17');
exec add_into_menu_items(find_cuisine_type_id('Italian'), 'salad', '8');
exec add_into_menu_items(find_cuisine_type_id('Italian'), 'salmon', '16');
exec add_into_menu_items(find_cuisine_type_id('Italian'), 'lasagna', '25');
```

Script Output x

Task completed in 0.127 seconds

Procedure ADD\_INT0\_MENU\_ITEMS compiled



SQL Worksheet History

Worksheet Query Builder

```

exec add_into_menu_items(find_cuisine_type_id('Italian'), 'burger', '12');
exec add_into_menu_items(find_cuisine_type_id('Italian'), 'fries', '4');
exec add_into_menu_items(find_cuisine_type_id('Italian'), 'pasta', '17');
exec add_into_menu_items(find_cuisine_type_id('Italian'), 'salad', '8');
exec add_into_menu_items(find_cuisine_type_id('Italian'), 'salmon', '16');
exec add_into_menu_items(find_cuisine_type_id('Italian'), 'lasagna', '25');
exec add_into_menu_items(find_cuisine_type_id('Italian'), 'meatballs', '11');
exec add_into_menu_items(find_cuisine_type_id('Italian'), 'spaghetti', '14');
exec add_into_menu_items(find_cuisine_type_id('Italian'), 'pizza', '35');

exec add_into_menu_items(find_cuisine_type_id('American'), 'wings', '13');
exec add_into_menu_items(find_cuisine_type_id('American'), 'icecream', '5');
exec add_into_menu_items(find_cuisine_type_id('American'), 'steak', '20');
exec add_into_menu_items(find_cuisine_type_id('American'), 'nuggets', '12');
exec add_into_menu_items(find_cuisine_type_id('American'), 'subs', '18');
exec add_into_menu_items(find_cuisine_type_id('American'), 'sushi', '22');
exec add_into_menu_items(find_cuisine_type_id('American'), 'tacos', '11');
exec add_into_menu_items(find_cuisine_type_id('American'), 'soup', '23');
exec add_into_menu_items(find_cuisine_type_id('American'), 'mcarronnie', '15');

select * from menu_items;

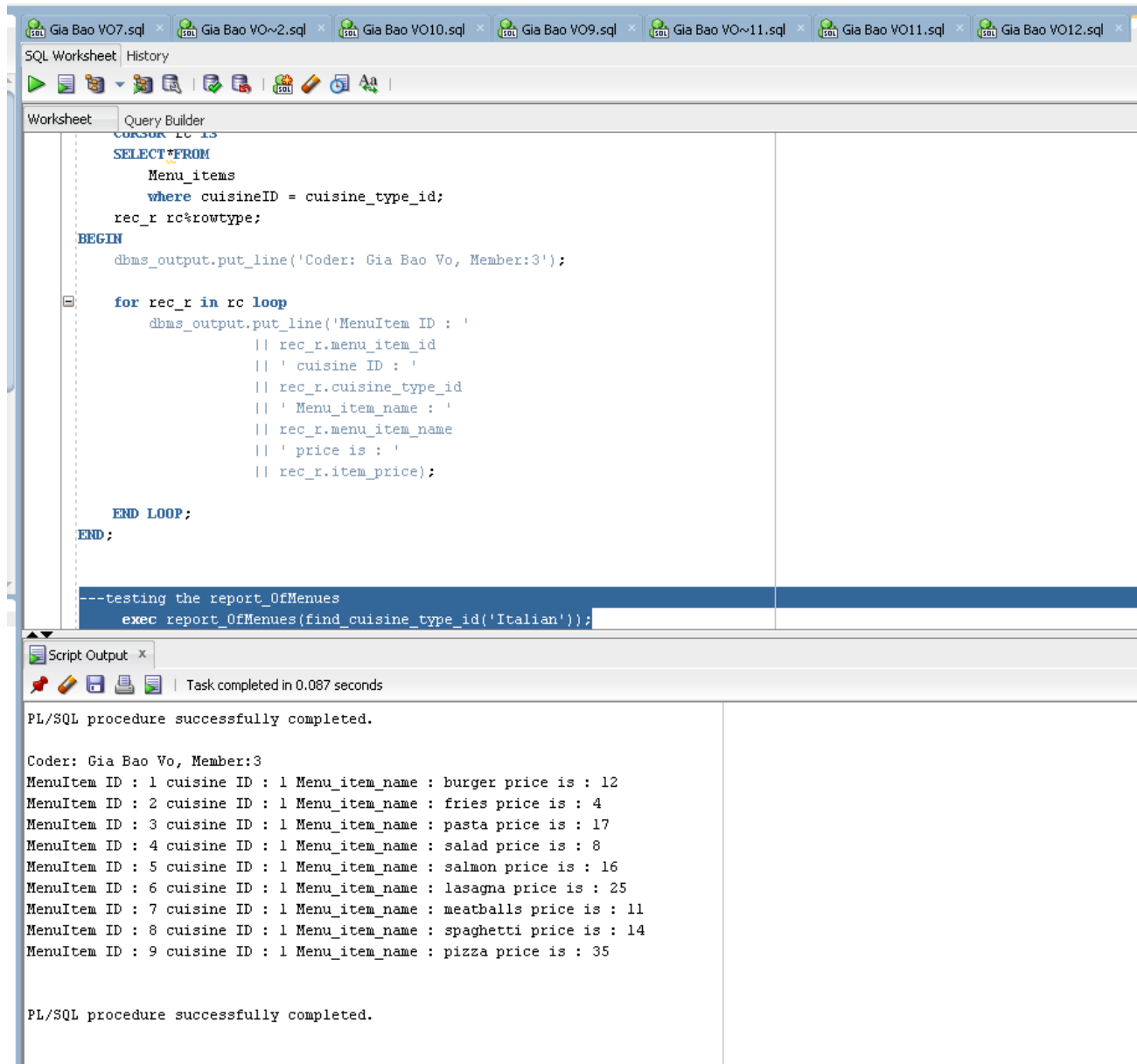
```

Script Output x Query Result x

SQL | All Rows Fetched: 18 in 0.028 seconds

	MENU_IT...	CUISINE_...	MENU_IT...	ITEM_PRICE
1	1	1	burger	12
2	2	2	fries	4
3	3	3	pasta	17
4	4	4	salad	8
5	5	5	salmon	16
6	6	6	lasagna	25
7	7	7	meatballs	11
8	8	8	spaghetti	14
9	9	9	pizza	35
10	10	10	wings	13
11	11	11	icecream	5
12	12	12	steak	20
13	13	13	nuggets	12
14	14	14	subs	18
15	15	15	sushi	22
16	16	16	tacos	11
17	17	17	soup	23
18	18	18	mcarronnie	15

12. Report Menu items: Generate a report to show totals of each menu item by type of cuisine.



The screenshot shows an SQL Worksheet interface with multiple tabs at the top, including 'Gia Bao VO7.sql', 'Gia Bao VO~2.sql', 'Gia Bao VO10.sql', 'Gia Bao VO9.sql', 'Gia Bao VO~11.sql', 'Gia Bao VO11.sql', and 'Gia Bao VO12.sql'. The main window displays a PL/SQL procedure named 'report\_OfMenus' in the 'Query Builder' tab. The procedure uses a cursor to fetch menu items and a loop to print their details. Below the procedure, a test call is shown: 'exec report\_OfMenus(find\_cuisine\_type\_id('Italian'))'. The 'Script Output' tab at the bottom shows the execution results, including a success message and a list of menu items for Italian cuisine.

```
Cursor rc is
SELECT * FROM
  Menu_items
  where cuisineID = cuisine_type_id;
rec_r rc%rowtype;
BEGIN
  dbms_output.put_line('Coder: Gia Bao Vo, Member:3');

  for rec_r in rc loop
    dbms_output.put_line('MenuItem ID : '
      || rec_r.menu_item_id
      || ' cuisine ID : '
      || rec_r.cuisine_type_id
      || ' Menu_item_name : '
      || rec_r.menu_item_name
      || ' price is : '
      || rec_r.item_price);

  END LOOP;
END;
```

```
---testing the report_OfMenus
exec report_OfMenus(find_cuisine_type_id('Italian'));
```

PL/SQL procedure successfully completed.

Coder: Gia Bao Vo, Member:3

MenuItem ID : 1	cuisine ID : 1	Menu_item_name : burger	price is : 12
MenuItem ID : 2	cuisine ID : 1	Menu_item_name : fries	price is : 4
MenuItem ID : 3	cuisine ID : 1	Menu_item_name : pasta	price is : 17
MenuItem ID : 4	cuisine ID : 1	Menu_item_name : salad	price is : 8
MenuItem ID : 5	cuisine ID : 1	Menu_item_name : salmon	price is : 16
MenuItem ID : 6	cuisine ID : 1	Menu_item_name : lasagna	price is : 25
MenuItem ID : 7	cuisine ID : 1	Menu_item_name : meatballs	price is : 11
MenuItem ID : 8	cuisine ID : 1	Menu_item_name : spaghetti	price is : 14
MenuItem ID : 9	cuisine ID : 1	Menu_item_name : pizza	price is : 35

PL/SQL procedure successfully completed.

The screenshot shows an SQL Worksheet window with multiple tabs. The active tab is 'Worksheet', which contains a PL/SQL procedure. The procedure is designed to output menu items for a specific member. It includes a loop that iterates through menu items and a test execution for 'Italian' and 'American' cuisines. The 'Script Output' window at the bottom shows the successful completion of the procedure and the resulting output for Member 3.

```

dbms_output.put_line('Coder: Gia Bao Vo, Member:3');

for rec_r in rc loop
    dbms_output.put_line('MenuItem ID : '
        || rec_r.menu_item_id
        || ' cuisine ID : '
        || rec_r.cuisine_type_id
        || ' Menu_item_name : '
        || rec_r.menu_item_name
        || ' price is : '
        || rec_r.item_price);

END LOOP;
END;

---testing the report_OfMenues
exec report_OfMenues(find_cuisine_type_id('Italian'));

exec report_OfMenues(find_cuisine_type_id('American'));

```

Script Output x | Task completed in 0.099 seconds

PL/SQL procedure successfully completed.

```

Coder: Gia Bao Vo, Member:3
MenuItem ID : 10 cuisine ID : 2 Menu_item_name : wings price is : 13
MenuItem ID : 11 cuisine ID : 2 Menu_item_name : icecream price is : 5
MenuItem ID : 12 cuisine ID : 2 Menu_item_name : steak price is : 20
MenuItem ID : 13 cuisine ID : 2 Menu_item_name : nuggets price is : 12
MenuItem ID : 14 cuisine ID : 2 Menu_item_name : subs price is : 18
MenuItem ID : 15 cuisine ID : 2 Menu_item_name : sushi price is : 22
MenuItem ID : 16 cuisine ID : 2 Menu_item_name : tacos price is : 11
MenuItem ID : 17 cuisine ID : 2 Menu_item_name : soup price is : 23
MenuItem ID : 18 cuisine ID : 2 Menu_item_name : mcarronnie price is : 15

```

PL/SQL procedure successfully completed.






#### Member 4: Sadia Alam

Responsible for the Orders table. Must create this table.

13. Place an order: Given all required information, add an order in the Orders table. Use the FIND\_x\_ID helper functions (where x is the name of a table - see helper functions below) first to retrieve the IDs that are needed.

```
create or replace PROCEDURE PLACE_ORDER  
(
```

Script Output x

     | Task completed in 0.619 seconds

Procedure PLACE\_ORDER compiled

Coder: Sadia Alam, Member:4

PL/SQL procedure successfully completed.

Coder: Sadia Alam, Member:4

PL/SQL procedure successfully completed.

Coder: Sadia Alam, Member:4

PL/SQL procedure successfully completed.

Coder: Sadia Alam, Member:4

PL/SQL procedure successfully completed.

SELECT * FROM orders;								
Query Result x								
All Rows Fetched: 12 in 0.041 seconds								
	ORDER_ID	RESTAURANT_ID	CUSTOMER_ID	MENU_ITEM_ID	WAITER_ID	ORDER_DATE	AMOUNT_PAID	TIP
1	1	7	12	1	7	01-NOV-22	12	2.4
2	2	7	12	2	7	01-NOV-22	4	0.8
3	3	7	13	1	7	06-NOV-22	12	2.4
4	4	8	12	22	8	10-NOV-22	5.99	1.2
5	5	8	13	22	8	13-NOV-22	5.99	1.2
6	6	1	11	19	6	20-NOV-22	6.99	1.4
7	7	1	11	20	6	20-NOV-22	1	0.2
8	8	1	11	21	6	20-NOV-22	1	0.2
9	9	8	6	22	8	15-NOV-22	5.99	1.2
10	10	8	7	22	8	16-NOV-22	5.99	1.2
11	11	7	8	1	7	19-NOV-22	12	2.4
12	12	7	10	1	7	21-NOV-22	12	2.4

14. List all orders at a given restaurant on a given date.

```
-- create procedure to list orders for a given restaurant on a given date
CREATE OR REPLACE PROCEDURE LIST_ORDERS
(

```

Script Output x

Task completed in 0.205 seconds

Procedure LIST\_ORDERS compiled

Coder: Sadia Alam, Member:4

Order ID: 1  
 Restaurant ID: 7  
 Customer ID: 12  
 Menu Item ID: 1  
 Waiter ID: 7  
 Order Date: 01-NOV-22  
 Amount Paid: 12  
 Tip: 2.4

Order ID: 2  
 Restaurant ID: 7  
 Customer ID: 12  
 Menu Item ID: 2  
 Waiter ID: 7  
 Order Date: 01-NOV-22  
 Amount Paid: 4  
 Tip: .8

15. List the most popular menu item ordered for each cuisine type

<pre>-- create procedure that displays the most popular menu of a given cuisine type create or replace PROCEDURE MOST_POPULAR_MENU (     cuisine_name_param VARCHAR2 ) IS</pre>	
<div>Script Output x</div> <div>Task completed in 0.168 seconds</div>	
<pre>Procedure MOST_POPULAR_MENU compiled  Coder: Sadia Alam, Member:4 Cusine Name: Indian Most Popular Menu: Tandoori Chicken  PL/SQL procedure successfully completed.</pre>	

16. Report: Generate a report showing the top 3 restaurants of each state. The ranking is based on the total of 'amount paid' per restaurant per state.

```
-- create procedure to list the top 3 restaurants of a given state
create or replace PROCEDURE TOP_RESTAURANT_REPORT
(
    r_state_param VARCHAR2
)
IS
```

Script Output x

Task completed in 0.177 seconds

Procedure TOP\_RESTAURANT\_REPORT compiled

Coder: Sadia Alam, Member:4  
Restaurant Name: Chick-fil-A  
Restaurant Total Income: 52

Restaurant Name: Kabob Place  
Restaurant Total Income: 23.96

Restaurant Name: Subway  
Restaurant Total Income: 8.99

PL/SQL procedure successfully completed.

### Member 5: Shikha Gondalia

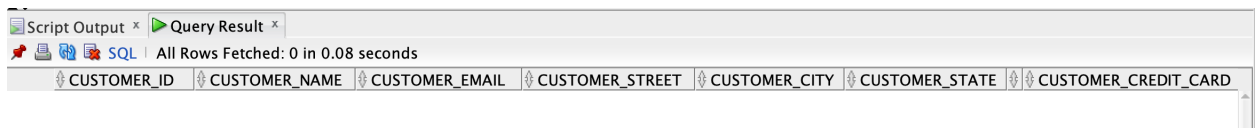
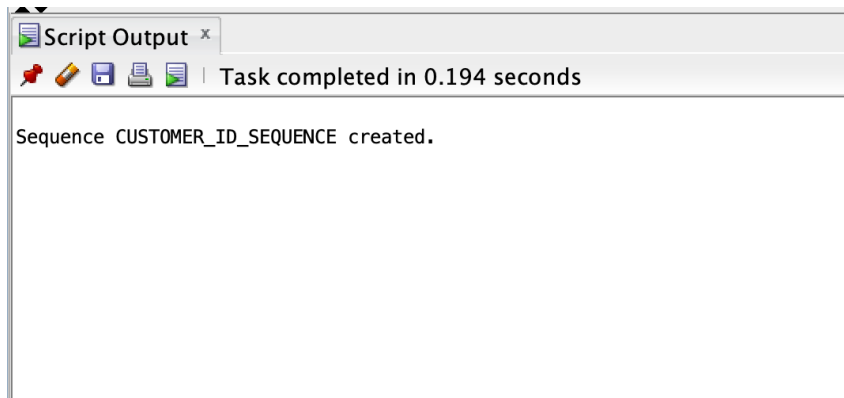
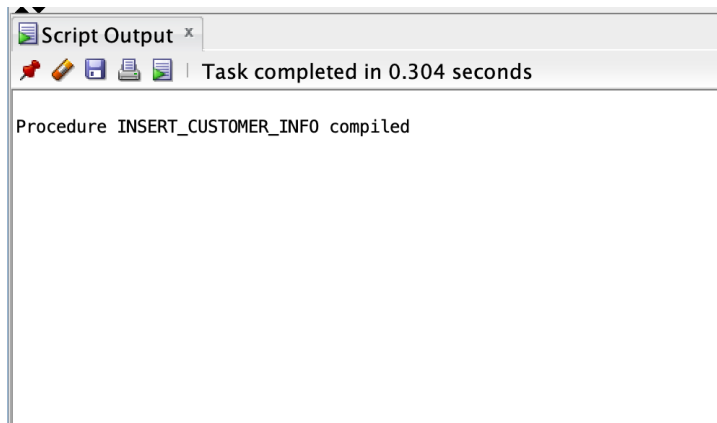
Responsible for the Customers table. Must create this table.

17. Add a customer: Given all necessary information add a customer to the DB

Script Output x

Task completed in 0.378 seconds

Table CUSTOMER created.



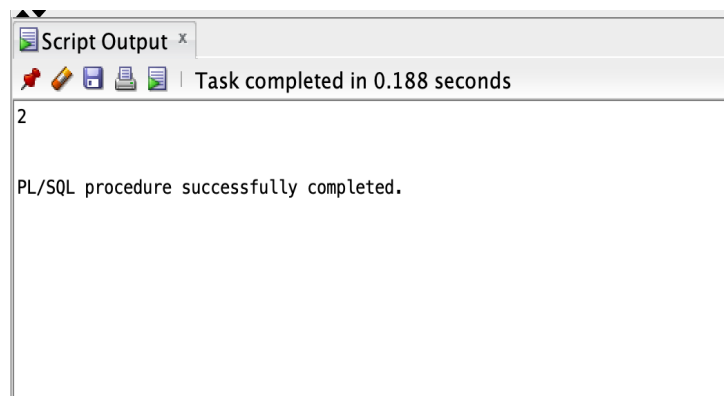
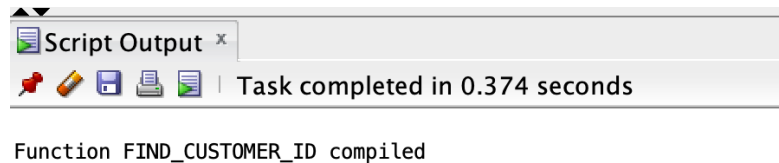
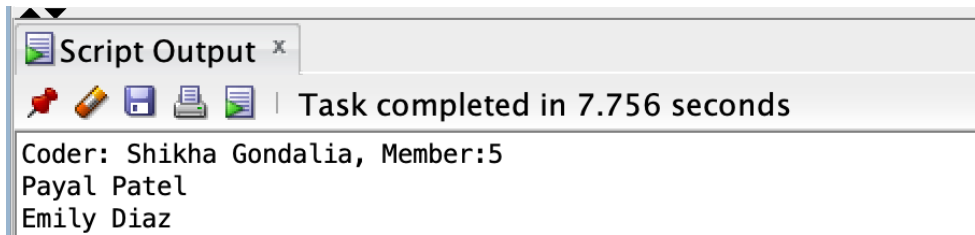
Script Output x Query Result x

SQL All Rows Fetched: 13 in 0.048 seconds

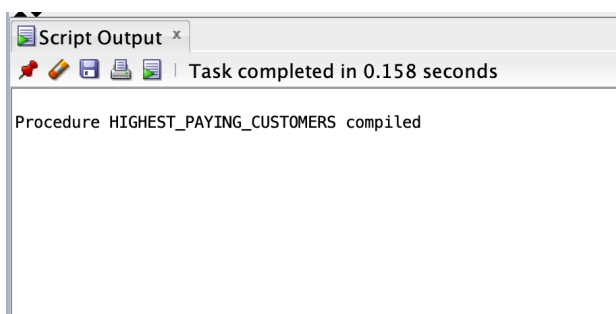
CUSTOMER_ID	CUSTOMER_NAME	CUSTOMER_EMAIL	CUSTOMER_STREET	CUSTOMER_CITY	CUSTOMER_STATE	CUSTOMER_ZIP	CUSTOMER_CREDIT_CARD
1	1 Joe Thomas	joethomas1@gmail.com	Singers Way	Elkridge	MD	21075	7834-3921-0987-4532
2	2 Payal Patel	payalpatel1876@gmail...	Abbey Road	Ellicott City	MD	21043	0987-3431-9343-9481
3	3 Megan Reed	meganreed043@gmail...	Dobbin Road	San Diego	CA	92014	0912-9321-9310-8764
4	4 Abby Miller	abbeymiller021@gma...	Lark Lane	Arlington	VA	20350	4312-8723-3210-0961
5	5 Emily Diaz	emilydiaz01@gmail.com	Woodstock Way	Ellicott City	MD	21043	2313-9834-3491-0431
6	6 Leo Smith	leosmith892@gmail.com	Blackwell Lane	Cleveland	OH	44101	8734-4091-3231-3203
7	7 Anjali Shah	anjalishah2@gmail.com	Blueberry Way	McLean	VA	22106	9736-0123-9873-2134
8	8 Rick Novak	ricknovak632@gmail...	Buttercup Way	Los Angeles	CA	90005	7643-8903-2189-0765
9	9 Emma Smith	emmasmith77@gmail.com	Burns Drive	Atlanta	GA	30305	0723-3213-1232-3241
10	10 Noah Barnett	noahbarnett70@gmai...	Clark Court	Los Angeles	CA	90005	8976-3432-0981-3432
11	11 Nandhu	n72@umbc.edu	13121 Serpentine Way	Silver Spring	MD	20904	7122-2905-1900-1257
12	12 John Carter	jcarter@gmail.com	101 River Park DR	Baltimore	MD	20821	1000-2000-3000-4000
13	13 Maya Erickson	merickson@gmail.com	102 River Park DR	Baltimore	MD	20821	1001-2001-3001-4001

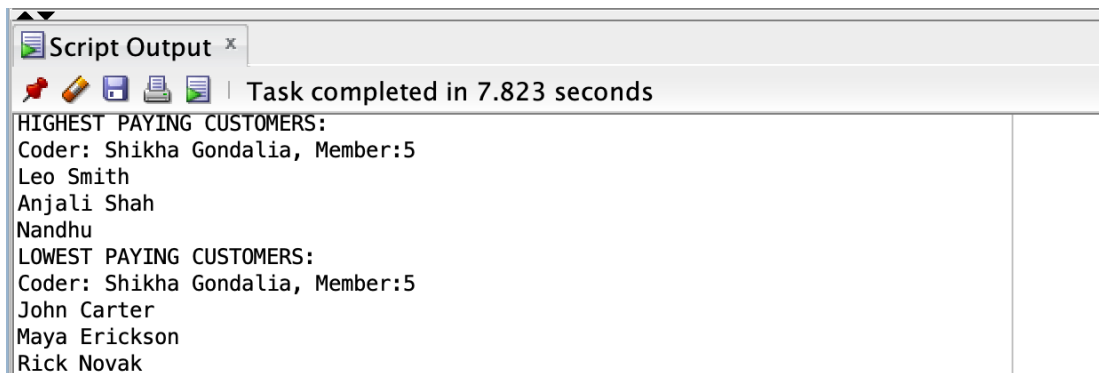
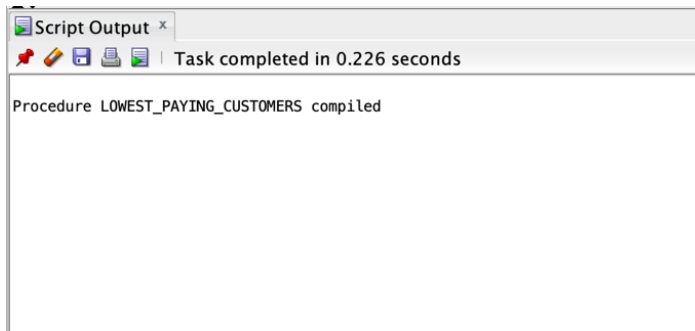




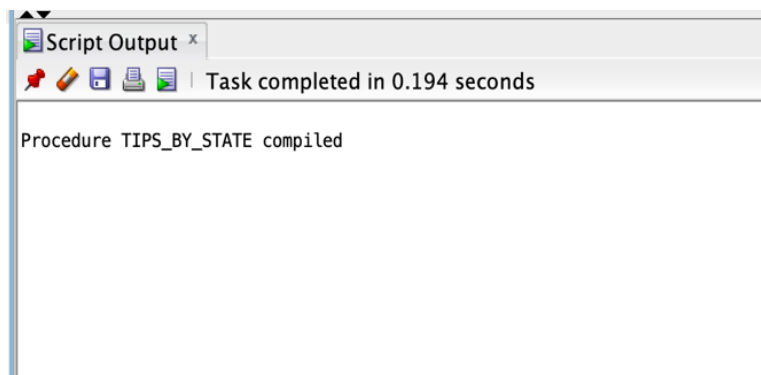


19. Report: Generate a report with the names of customers who spent the most money (top 3) so we can send them discount coupons, and also the names of the most frugal customers (bottom 3).










20. Report: States of generous customers. Generate a report that lists the states based on customers who tip generously. Show the total amount of tips by state in descending order of tip amount.



Script Output



Task completed in 7.756 seconds

Rick Novak

Coder: Shikha Gondalia, Member:5

STATES TIP AMOUNT:

MD: \$9.8

CA: \$4.8

OH: \$1.2

VA: \$1.2