SECD2523
DATABASE

SECTION 10

SQL LAB 4 - DML 3 PART 1

PREPARED FOR:

MADAM ROZILAWATI BINTI DOLLAH @ MD ZAIN

PREPARED BY:

BATRIESYA IRDINA BINTI KHAIRUL HEZAL	**A22EC0141**

Section 6 Lesson 9 Exercise 1: Joining Tables Using JOIN

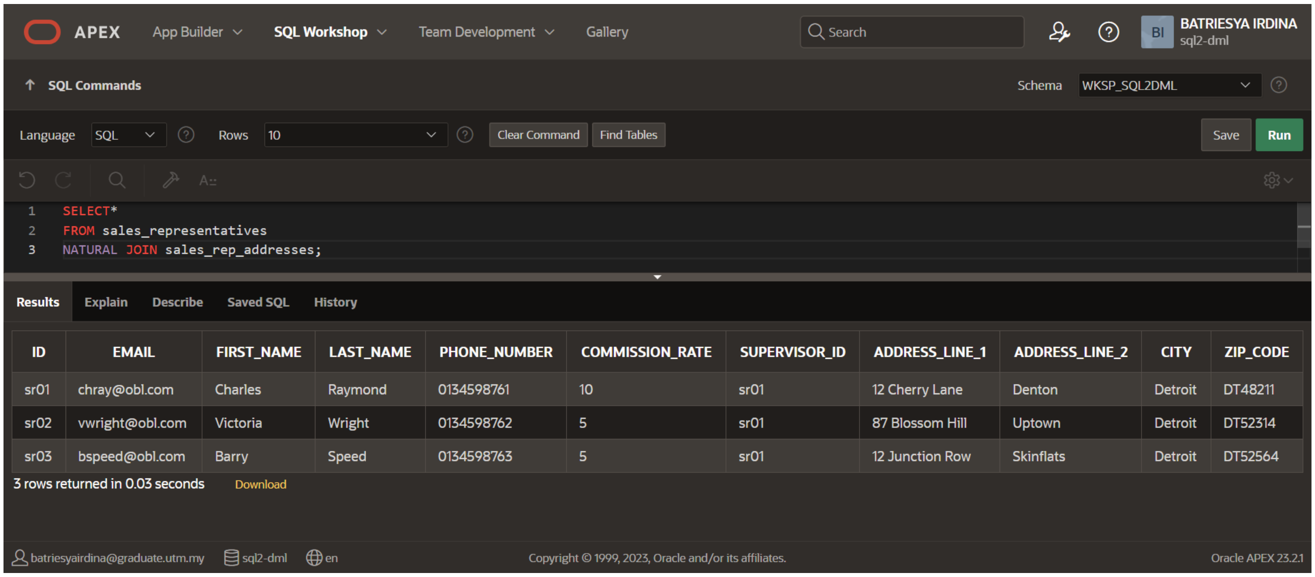
Write SELECT statements using data from multiple tables using equijoins and non-equijoins (S6L9 Objective 1)

In this exercise, you will write SELECT statements to access data from more than one table.

Part 1: Creating Natural Joins

1. Display all of the information about sales representatives and their addresses using a natural join.

   - **SELECT***
     FROM sales_representatives
     NATURAL JOIN sales_rep_addresses;



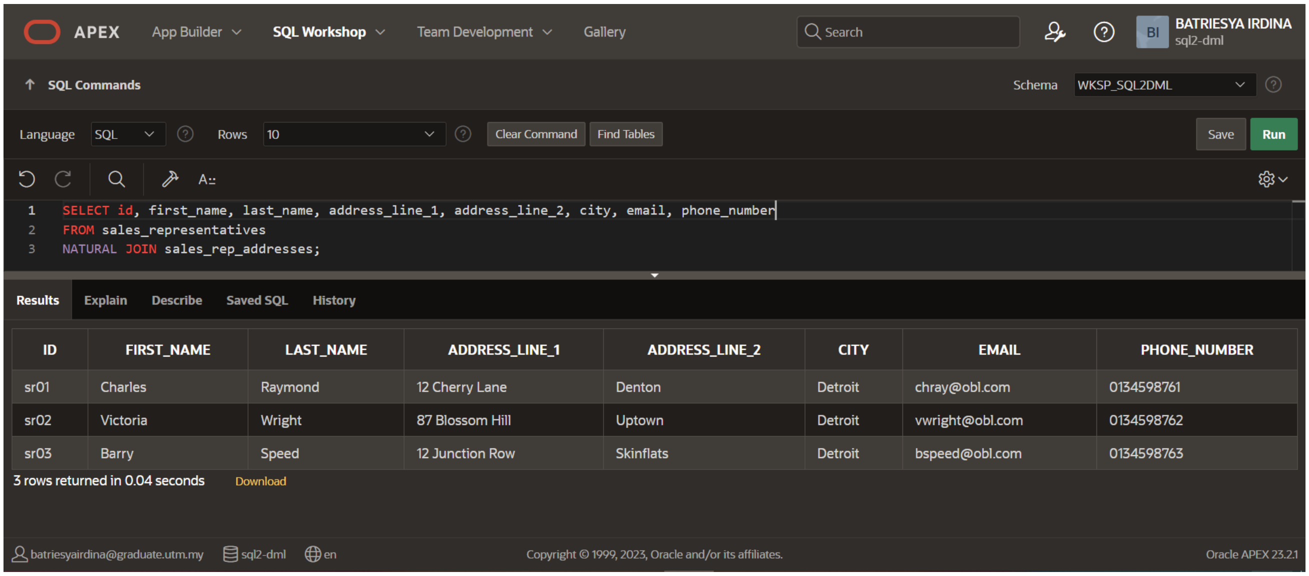
```
SELECT*
FROM sales_representatives
NATURAL JOIN sales_rep_addresses;
```

ID	EMAIL	FIRST_NAME	LAST_NAME	PHONE_NUMBER	COMMISSION_RATE	SUPERVISOR_ID	ADDRESS_LINE_1	ADDRESS_LINE_2	CITY	ZIP_CODE
sr01	chray@obl.com	Charles	Raymond	0134598761	10	sr01	12 Cherry Lane	Denton	Detroit	DT48211
sr02	vwright@obl.com	Victoria	Wright	0134598762	5	sr01	87 Blossom Hill	Uptown	Detroit	DT52314
sr03	bspeed@obl.com	Barry	Speed	0134598763	5	sr01	12 Junction Row	Skinflats	Detroit	DT52564

3 rows returned in 0.03 seconds

2. Adapt the query from the previous question to only show the id, first name, last name, address line 1, address line 2, city, email and phone_number for the sales representatives.

   - **SELECT id, first_name, last_name, address_line_1, address_line_2, city, email, phone_number**
     FROM sales_representatives
     NATURAL JOIN sales_rep_addresses;



```sql
SELECT id, first_name, last_name, address_line_1, address_line_2, city, email, phone_number
FROM sales_representatives
NATURAL JOIN sales_rep_addresses;
```

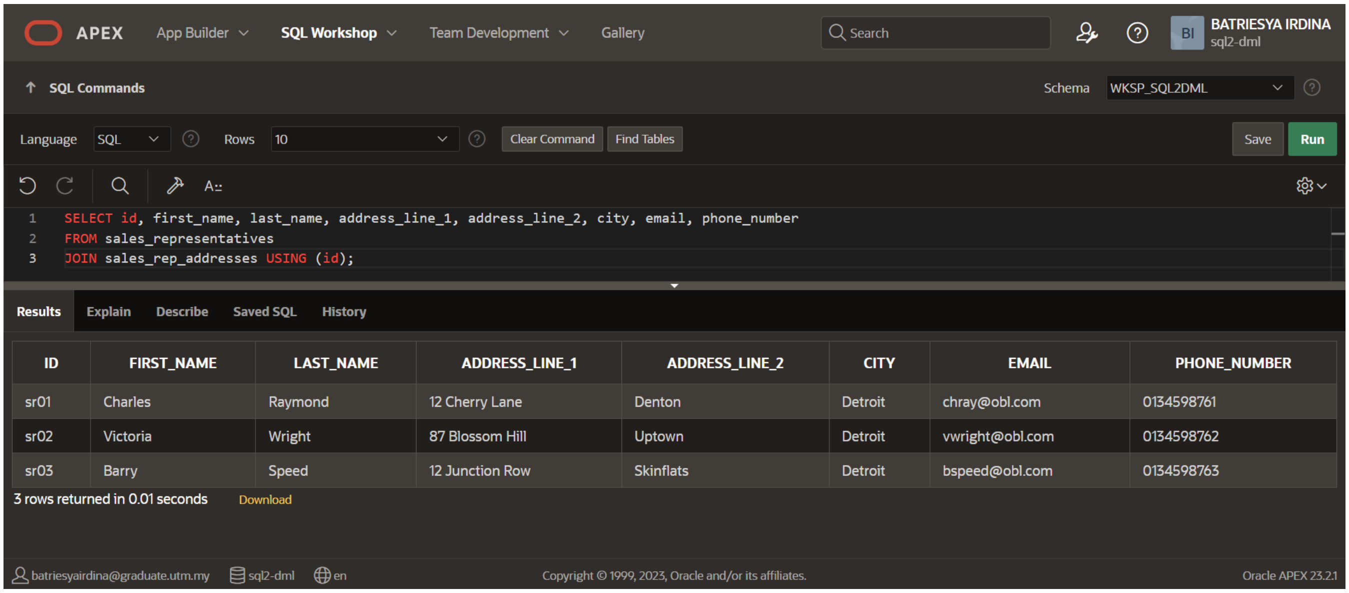
ID	FIRST_NAME	LAST_NAME	ADDRESS_LINE_1	ADDRESS_LINE_2	CITY	EMAIL	PHONE_NUMBER
sr01	Charles	Raymond	12 Cherry Lane	Denton	Detroit	chray@obl.com	0134598761
sr02	Victoria	Wright	87 Blossom Hill	Uptown	Detroit	vwright@obl.com	0134598762
sr03	Barry	Speed	12 Junction Row	Skinflats	Detroit	bspeed@obl.com	0134598763

3 rows returned in 0.04 seconds

Part 2: Creating Joins with the USING Clause

1. Adapt the previous query answer to use the USING clause instead of a natural join.

   - **SELECT id, first_name, last_name, address_line_1, address_line_2, city, email, phone_number
FROM sales_representatives
JOIN sales_rep_addresses USING (id);**



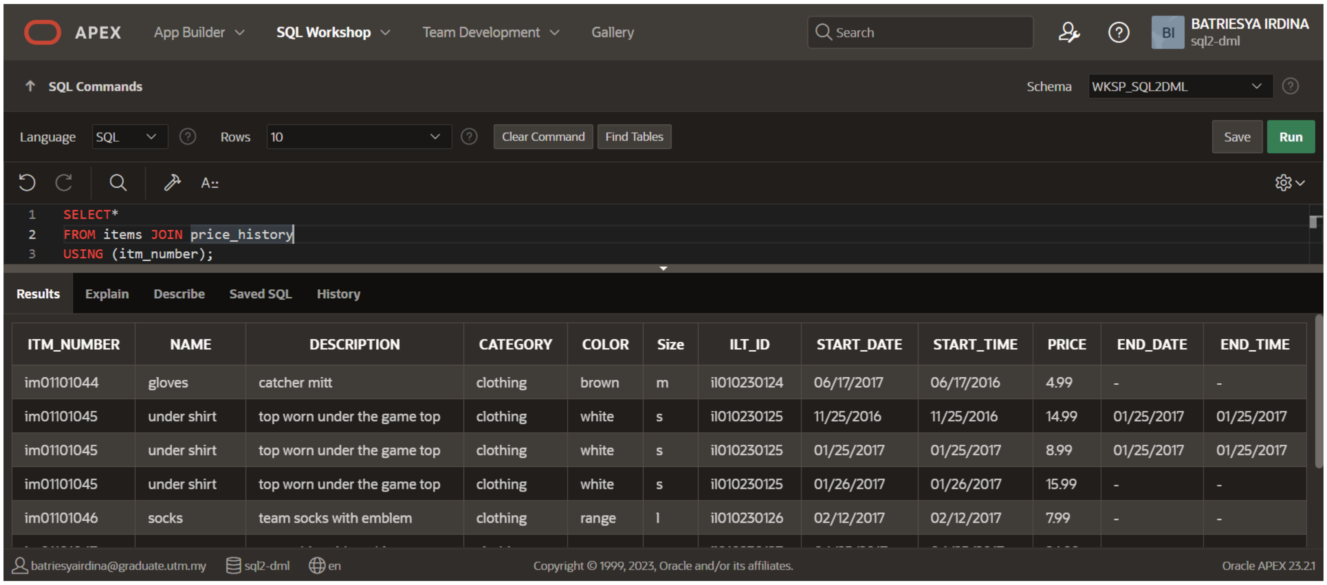
```sql
SELECT id, first_name, last_name, address_line_1, address_line_2, city, email, phone_number
FROM sales_representatives
JOIN sales_rep_addresses USING (id);
```

ID	FIRST_NAME	LAST_NAME	ADDRESS_LINE_1	ADDRESS_LINE_2	CITY	EMAIL	PHONE_NUMBER
sr01	Charles	Raymond	12 Cherry Lane	Denton	Detroit	chray@obl.com	0134598761
sr02	Victoria	Wright	87 Blossom Hill	Uptown	Detroit	vwright@obl.com	0134598762
sr03	Barry	Speed	12 Junction Row	Skinflats	Detroit	bspeed@obl.com	0134598763

3 rows returned in 0.01 seconds

2. Display all of the information about items and their price history by joining the items and price_history tables.

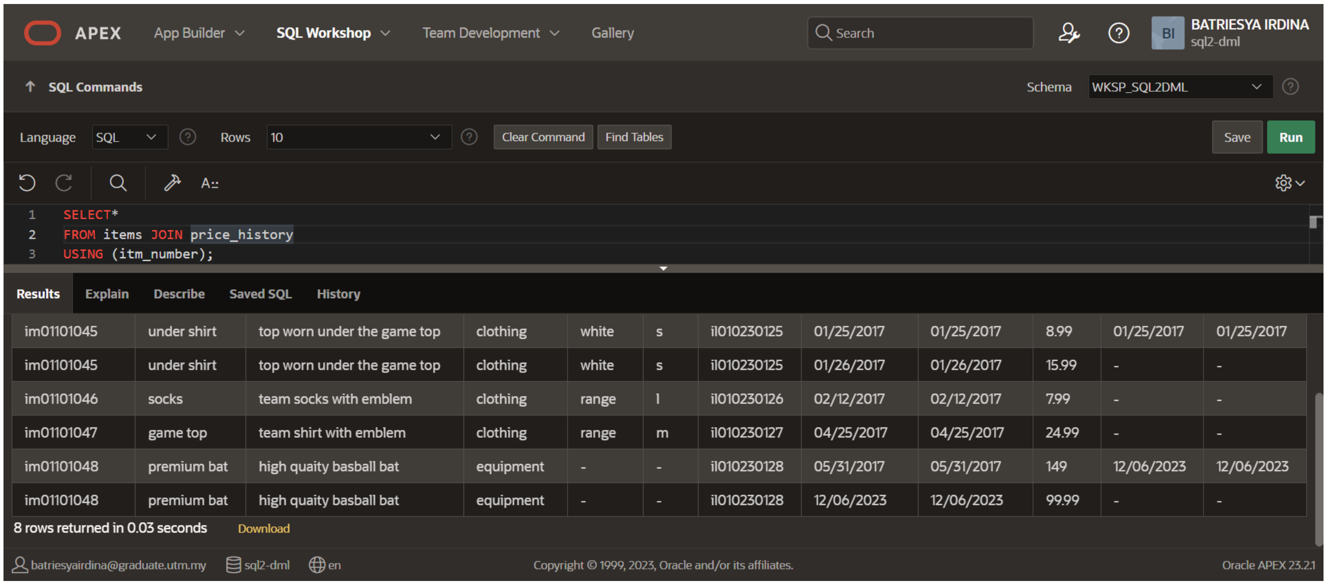
   - **SELECT***
FROM items JOIN price_history
USING (itm_number);



```
SELECT*
FROM items JOIN price_history
USING (itm_number);
```

ITM_NUMBER	NAME	DESCRIPTION	CATEGORY	COLOR	Size	ILT_ID	START_DATE	START_TIME	PRICE	END_DATE	END_TIME
im01101044	gloves	catcher mitt	clothing	brown	m	il010230124	06/17/2017	06/17/2016	4.99	-	-
im01101045	under shirt	top worn under the game top	clothing	white	s	il010230125	11/25/2016	11/25/2016	14.99	01/25/2017	01/25/2017
im01101045	under shirt	top worn under the game top	clothing	white	s	il010230125	01/25/2017	01/25/2017	8.99	01/25/2017	01/25/2017
im01101045	under shirt	top worn under the game top	clothing	white	s	il010230125	01/26/2017	01/26/2017	15.99	-	-
im01101046	socks	team socks with emblem	clothing	range	l	il010230126	02/12/2017	02/12/2017	7.99	-	-

batriesyairdina@graduate.utm.my
sql2-dml
en
Copyright © 1999, 2023, Oracle and/or its affiliates.
Oracle APEX 23.2.1



```
SELECT*
FROM items JOIN price_history
USING (itm_number);
```

im01101045	under shirt	top worn under the game top	clothing	white	s	il010230125	01/25/2017	01/25/2017	8.99	01/25/2017	01/25/2017
im01101045	under shirt	top worn under the game top	clothing	white	s	il010230125	01/26/2017	01/26/2017	15.99	-	-
im01101046	socks	team socks with emblem	clothing	range	l	il010230126	02/12/2017	02/12/2017	7.99	-	-
im01101047	game top	team shirt with emblem	clothing	range	m	il010230127	04/25/2017	04/25/2017	24.99	-	-
im01101048	premium bat	high quaity basball bat	equipment	-	-	il010230128	05/31/2017	05/31/2017	149	12/06/2023	12/06/2023
im01101048	premium bat	high quaity basball bat	equipment	-	-	il010230128	12/06/2023	12/06/2023	99.99	-	-

8 rows returned in 0.03 seconds

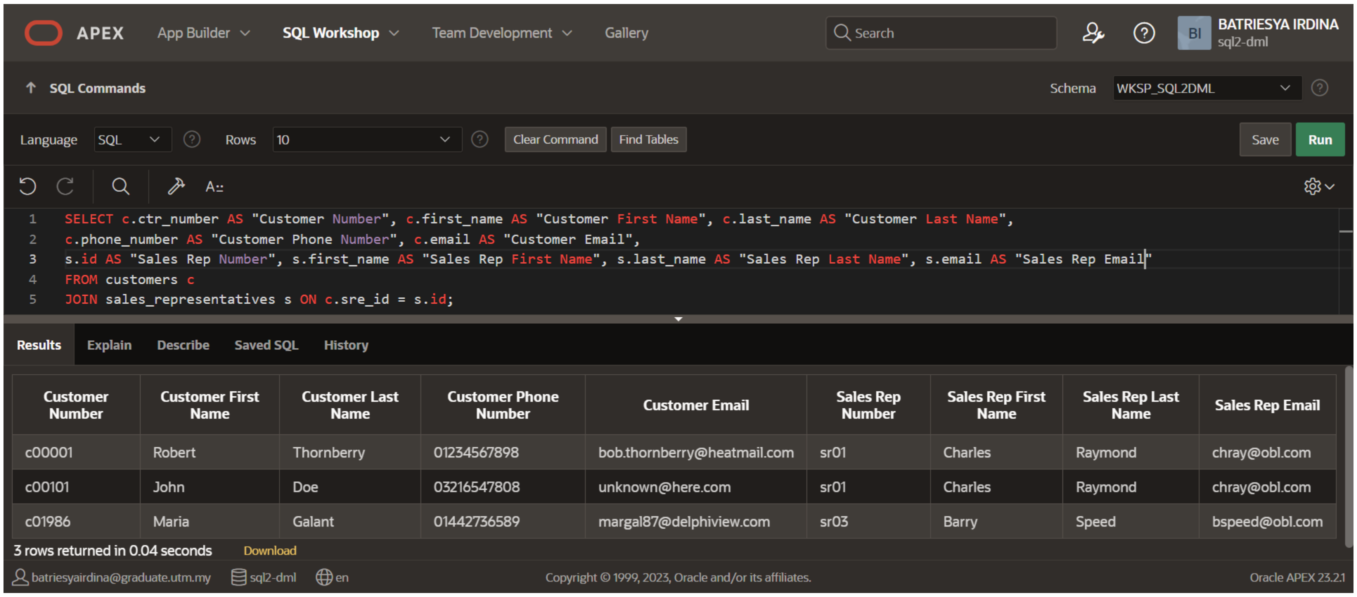
Download
batriesyairdina@graduate.utm.my
sql2-dml
en
Copyright © 1999, 2023, Oracle and/or its affiliates.
Oracle APEX 23.2.1

Part 3: Creating JOINS with the ON Clause

1. Use an ON clause to join the customer and sales representatives table so that you display the customer number, customer first name, customer last name, customer phone number,

customer email, sales representative id, sales representative first name, sales representative last name and sales representative email. You will need to use a table alias in your answer as both tables have columns with the same name.

- **SELECT c.ctr_number AS "Customer Number", c.first_name AS "Customer First Name", c.last_name AS "Customer Last Name",
c.phone_number AS "Customer Phone Number", c.email AS "Customer Email",
s.id AS "Sales Rep Number", s.first_name AS "Sales Rep First Name", s.last_name AS "Sales Rep Last Name", s.email AS "Sales Rep Email"
FROM customers c
JOIN sales_representatives s ON c.sre_id = s.id;**



```
SELECT c.ctr_number AS "Customer Number", c.first_name AS "Customer First Name", c.last_name AS "Customer Last Name",
c.phone_number AS "Customer Phone Number", c.email AS "Customer Email",
s.id AS "Sales Rep Number", s.first_name AS "Sales Rep First Name", s.last_name AS "Sales Rep Last Name", s.email AS "Sales Rep Email"
FROM customers c
JOIN sales_representatives s ON c.sre_id = s.id;
```

Customer Number	Customer First Name	Customer Last Name	Customer Phone Number	Customer Email	Sales Rep Number	Sales Rep First Name	Sales Rep Last Name	Sales Rep Email
c00001	Robert	Thornberry	01234567898	bob.thornberry@heatmail.com	sr01	Charles	Raymond	chray@obl.com
c00101	John	Doe	03216547808	unknown@here.com	sr01	Charles	Raymond	chray@obl.com
c01986	Maria	Galant	01442736589	margal87@delphiview.com	sr03	Barry	Speed	bspeed@obl.com

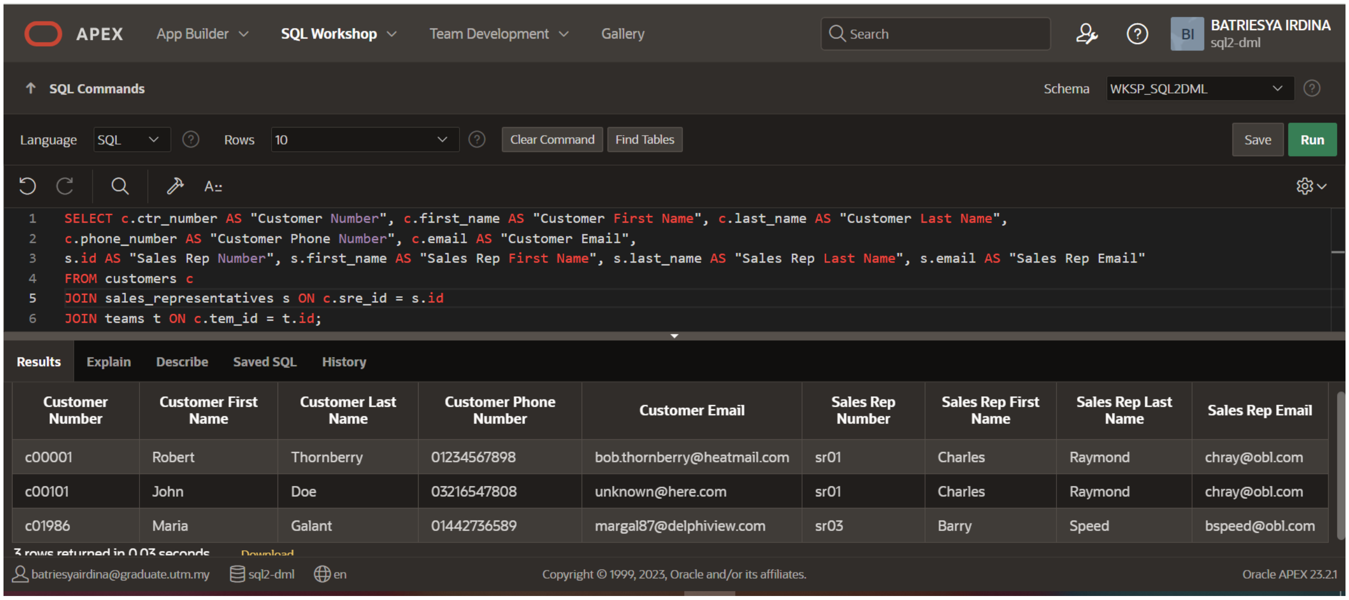
3 rows returned in 0.04 seconds

Part 4: Creating Three-Way JOINS with the ON Clause

1. Use the answer to Task 3 add a join that will allow the team name that the customer represents to be included in the results.

    - **SELECT c.ctr_number AS "Customer Number", c.first_name AS "Customer First Name", c.last_name AS "Customer Last Name",
c.phone_number AS "Customer Phone Number", c.email AS "Customer Email",
s.id AS "Sales Rep Number", s.first_name AS "Sales Rep First Name", s.last_name AS "Sales Rep Last Name", s.email AS "Sales Rep Email"
FROM customers c**

```
JOIN sales_representatives s ON c.sre_id = s.id;
JOIN teams t ON c.tem_id = t.id;
```



```
SELECT c.ctr_number AS "Customer Number", c.first_name AS "Customer First Name", c.last_name AS "Customer Last Name",
c.phone_number AS "Customer Phone Number", c.email AS "Customer Email",
s.id AS "Sales Rep Number", s.first_name AS "Sales Rep First Name", s.last_name AS "Sales Rep Last Name", s.email AS "Sales Rep Email"
FROM customers c
JOIN sales_representatives s ON c.sre_id = s.id
JOIN teams t ON c.tem_id = t.id;
```

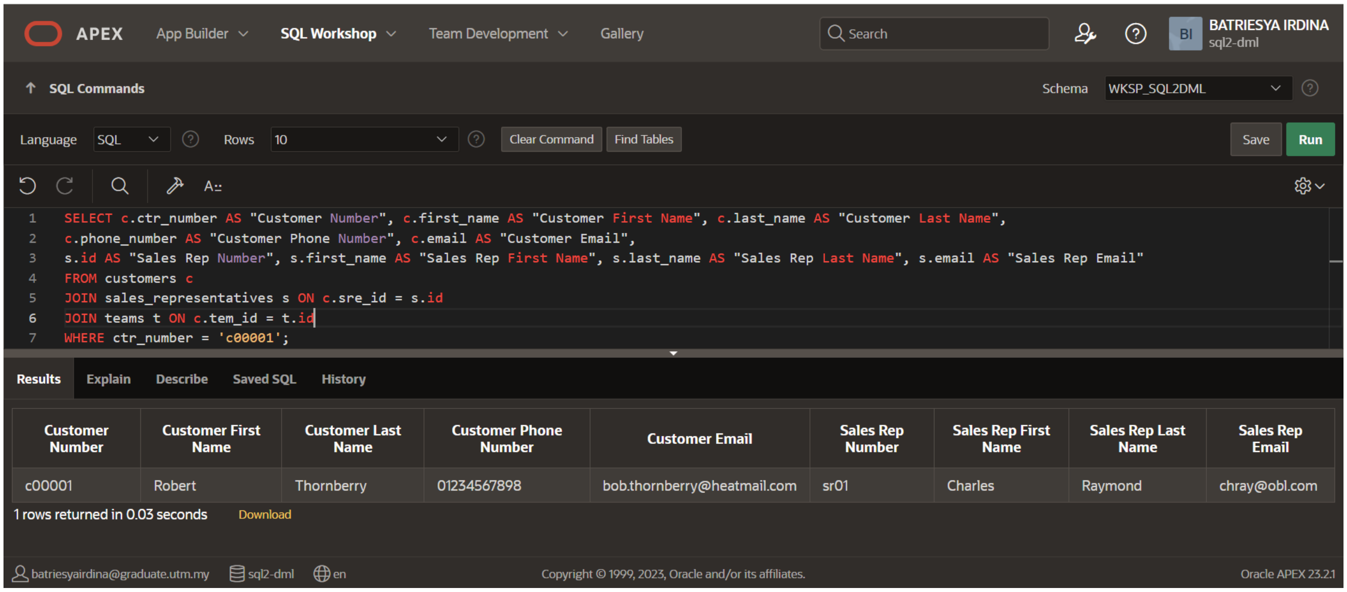
Customer Number	Customer First Name	Customer Last Name	Customer Phone Number	Customer Email	Sales Rep Number	Sales Rep First Name	Sales Rep Last Name	Sales Rep Email
c00001	Robert	Thornberry	01234567898	bob.thornberry@heatmail.com	sr01	Charles	Raymond	chray@obl.com
c00101	John	Doe	03216547808	unknown@here.com	sr01	Charles	Raymond	chray@obl.com
c01986	Maria	Galant	01442736589	margal87@delphiview.com	sr03	Barry	Speed	bspeed@obl.com

3 rows returned in 0.03 seconds

Part 5: Applying additional conditions to a JOIN

1. Use the answer to Task 4 add an additional condition to only show the results for the customer that has the number -c00001.

    -
      ```
      SELECT c.ctr_number AS "Customer Number", c.first_name AS "Customer First Name", c.last_name AS "Customer Last Name",
      c.phone_number AS "Customer Phone Number", c.email AS "Customer Email",
      s.id AS "Sales Rep Number", s.first_name AS "Sales Rep First Name", s.last_name AS "Sales Rep Last Name", s.email AS "Sales Rep Email"
      FROM customers c
      JOIN sales_representatives s ON c.sre_id = s.id;
      JOIN teams t ON c.tem_id = t.id;
      WHERE ctr_number = 'c00001';
      ```



```sql
SELECT c.ctr_number AS "Customer Number", c.first_name AS "Customer First Name", c.last_name AS "Customer Last Name",
c.phone_number AS "Customer Phone Number", c.email AS "Customer Email",
s.id AS "Sales Rep Number", s.first_name AS "Sales Rep First Name", s.last_name AS "Sales Rep Last Name", s.email AS "Sales Rep Email"
FROM customers c
JOIN sales_representatives s ON c.sre_id = s.id
JOIN teams t ON c.tem_id = t.id
WHERE ctr_number = 'c00001';
```

Customer Number	Customer First Name	Customer Last Name	Customer Phone Number	Customer Email	Sales Rep Number	Sales Rep First Name	Sales Rep Last Name	Sales Rep Email
c00001	Robert	Thornberry	01234567898	bob.thornberry@heatmail.com	sr01	Charles	Raymond	chray@obl.com

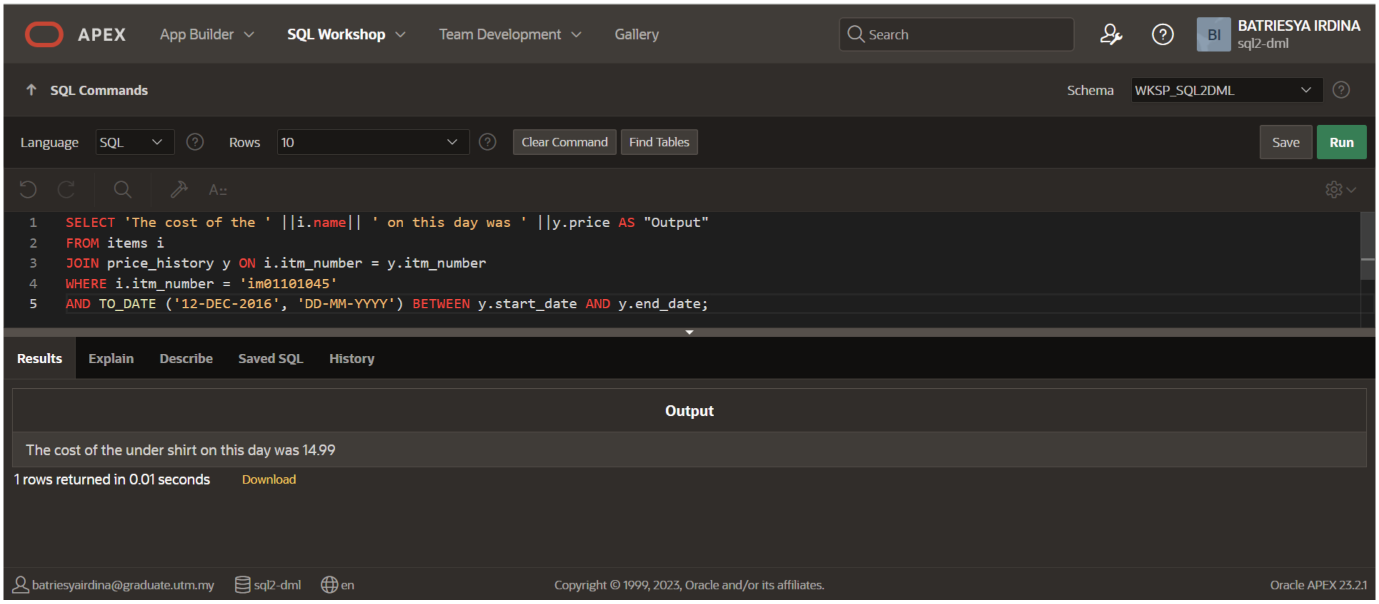
1 rows returned in 0.03 seconds

Part 6: Retrieving records with non equijoins

1. Write a query that will display the name and the cost of the item with the number im01101045 on the 12th of December 2016. The output of the query should look like this:

   The cost of the under shirt on this day was 14.99

   - **SELECT 'The cost of the ' ||i.name|| ' on this day was ' ||y.price AS "Output"
FROM items i
JOIN price_history y ON i.itm_number = y.itm_number
WHERE i.itm_number = 'im01101045'
AND TO_DATE ('12-DEC-2016', 'DD-MM-YYYY') BETWEEN y.start_date AND y.end_date;**



```
SELECT 'The cost of the ' ||i.name|| ' on this day was ' ||y.price AS "Output"
FROM items i
JOIN price_history y ON i.itm_number = y.itm_number
WHERE i.itm_number = 'im01101045'
AND TO_DATE ('12-DEC-2016', 'DD-MM-YYYY') BETWEEN y.start_date AND y.end_date;
```

Output
The cost of the under shirt on this day was 14.99

1 rows returned in 0.01 seconds