



**UTM**  
UNIVERSITI TEKNOLOGI MALAYSIA

**SECD2523  
DATABASE**

**SECTION 10**

**SQL LAB 4 - DML 3 PART 1**

**PREPARED FOR:**

**MADAM ROZILAWATI BINTI DOLLAH @ MD ZAIN**

**PREPARED BY:**

<b>BATRIESYA IRDINA BINTI KHAIRUL HEZAL</b>	<b>A22EC0141</b>
---	------------------

## Section 6 Lesson 9 Exercise 1: Joining Tables Using JOIN

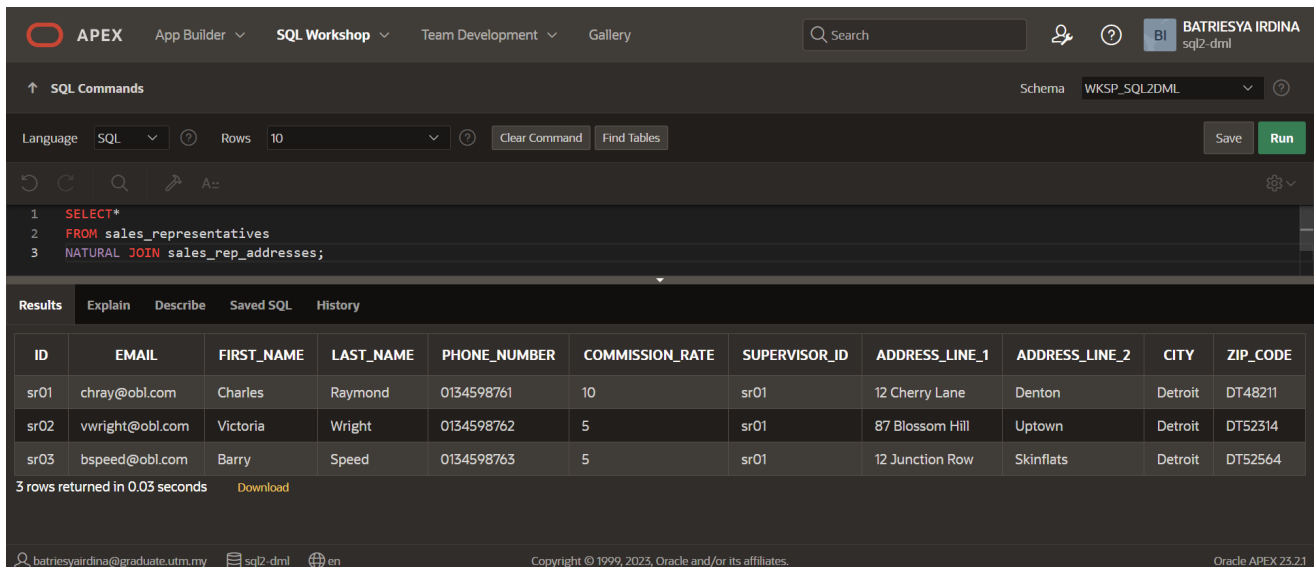
### Write SELECT statements using data from multiple tables using equijoins and non-equijoins (S6L9 Objective 1)

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;
```



The screenshot shows the APEX SQL Workshop interface. The SQL command entered is:

```
1 SELECT*  
2 FROM sales_representatives  
3 NATURAL JOIN sales_rep_addresses;
```

The results are displayed in a table with 11 columns: ID, EMAIL, FIRST\_NAME, LAST\_NAME, PHONE\_NUMBER, COMMISSION\_RATE, SUPERVISOR\_ID, ADDRESS\_LINE\_1, ADDRESS\_LINE\_2, CITY, and ZIP\_CODE. The table contains 3 rows of data.

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;
```

**APEX** App Builder SQL Workshop Team Development Gallery

Search

Schema: WKSP\_SQL2DML

Language: SQL Rows: 10 Clear Command Find Tables Save Run

```

1 SELECT id, first_name, last_name, address_line_1, address_line_2, city, email, phone_number
2 FROM sales_representatives
3 NATURAL JOIN sales_rep_addresses;

```

**Results** Explain Describe Saved SQL History

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 Download

batriesyairdina@graduate.utm.my sql2-dml en Copyright © 1999, 2023, Oracle and/or its affiliates. Oracle APEX 23.21

## 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);**

**APEX** App Builder SQL Workshop Team Development Gallery

Search

Schema: WKSP\_SQL2DML

Language: SQL Rows: 10 Clear Command Find Tables Save Run

```

1 SELECT id, first_name, last_name, address_line_1, address_line_2, city, email, phone_number
2 FROM sales_representatives
3 JOIN sales_rep_addresses USING (id);

```

**Results** Explain Describe Saved SQL History

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 Download

batriesyairdina@graduate.utm.my sql2-dml en Copyright © 1999, 2023, Oracle and/or its affiliates. Oracle APEX 23.21

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);
```

The screenshot shows the Oracle APEX SQL Workshop interface. The SQL command entered is:

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

The results are displayed in a table with 12 columns: ITM\_NUMBER, NAME, DESCRIPTION, CATEGORY, COLOR, Size, ILT\_ID, START\_DATE, START\_TIME, PRICE, END\_DATE, and END\_TIME. The first 10 rows of data are shown.

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	-	-

The screenshot shows the same Oracle APEX SQL Workshop interface. The SQL command is the same as in the previous screenshot. The results are displayed in a table with 12 columns. The first 8 rows of data are shown.

ITM_NUMBER	NAME	DESCRIPTION	CATEGORY	COLOR	Size	ILT_ID	START_DATE	START_TIME	PRICE	END_DATE	END_TIME
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 quality baseball bat	equipment	-	-	il010230128	05/31/2017	05/31/2017	149	12/06/2023	12/06/2023
im01101048	premium bat	high quality baseball bat	equipment	-	-	il010230128	12/06/2023	12/06/2023	99.99	-	-

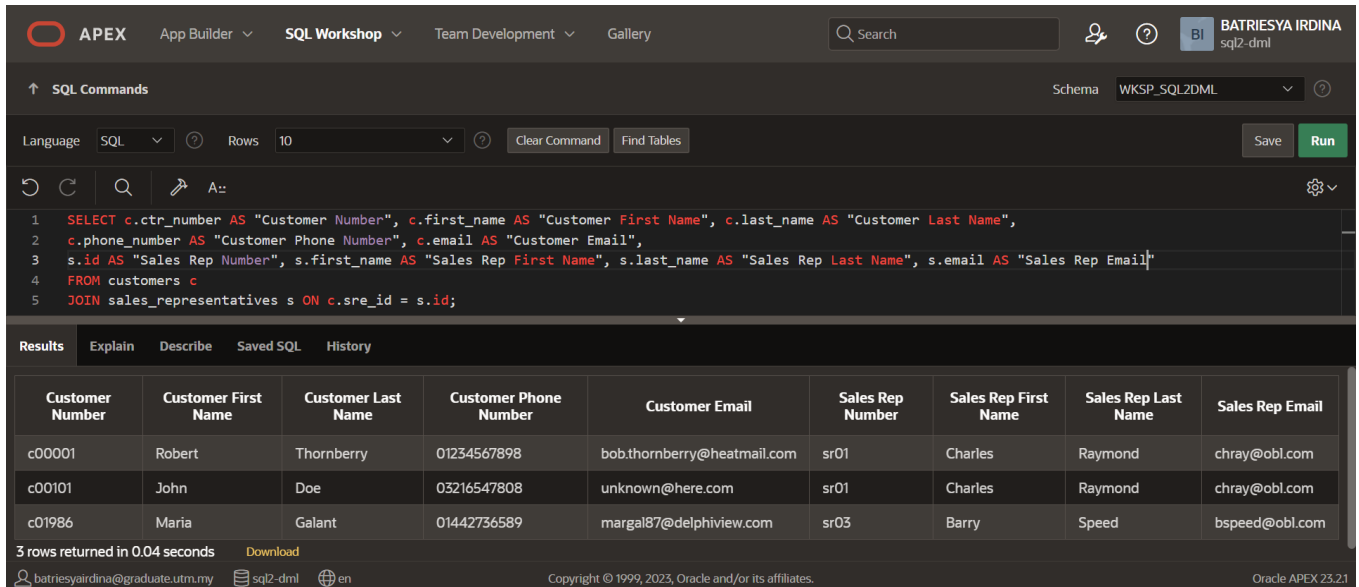
8 rows returned in 0.03 seconds

## 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;**



The screenshot displays the Oracle APEX SQL Workshop interface. The SQL command window contains the following query:

```
1 SELECT c.ctr_number AS "Customer Number", c.first_name AS "Customer First Name", c.last_name AS "Customer Last Name",  
2 c.phone_number AS "Customer Phone Number", c.email AS "Customer Email",  
3 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"  
4 FROM customers c  
5 JOIN sales_representatives s ON c.sre_id = s.id;
```

The Results tab shows the following data:

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;**

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop', 'Team Development', and 'Gallery'. A search bar and user profile 'BATRIESYA IRDINA' are also visible. The 'SQL Commands' section shows a query with 10 rows. The 'Results' tab is active, displaying a table with 9 columns: Customer Number, Customer First Name, Customer Last Name, Customer Phone Number, Customer Email, Sales Rep Number, Sales Rep First Name, Sales Rep Last Name, and Sales Rep Email. The table contains three rows of data.

```

1 SELECT c.ctr_number AS "Customer Number", c.first_name AS "Customer First Name", c.last_name AS "Customer Last Name",
2 c.phone_number AS "Customer Phone Number", c.email AS "Customer Email",
3 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"
4 FROM customers c
5 JOIN sales_representatives s ON c.sre_id = s.id
6 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

Copyright © 1999, 2023, Oracle and/or its affiliates. Oracle APEX 23.21

## Part 5: Applying additional conditions to a JOIN

- 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';**

Oracle APEX SQL Workshop interface showing a query execution result.

**SQL Commands:**

```

1 SELECT c.ctr_number AS "Customer Number", c.first_name AS "Customer First Name", c.last_name AS "Customer Last Name",
2 c.phone_number AS "Customer Phone Number", c.email AS "Customer Email",
3 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"
4 FROM customers c
5 JOIN sales_representatives s ON c.sre_id = s.id
6 JOIN teams t ON c.tem_id = t.id
7 WHERE ctr_number = 'c00001';

```

**Results:**

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

- 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;**

APEX

App Builder

SQL Workshop

Team Development

Gallery

Search

BATRIESYA IRDINA

sql2-dml

SQL Commands

Schema WKSP\_SQL2DML

Language SQL

Rows 10

Clear Command

Find Tables

Save

Run

1 SELECT 'The cost of the ' || i.name || ' on this day was ' || y.price AS "Output"

2 FROM items i

3 JOIN price\_history y ON i.itm\_number = y.itm\_number

4 WHERE i.itm\_number = 'im01101045'

5 AND TO\_DATE ('12-DEC-2016', 'DD-MM-YYYY') BETWEEN y.start\_date AND y.end\_date;

Results

Explain

Describe

Saved SQL

History

Output

The cost of the under shirt on this day was 14.99

1 rows returned in 0.01 seconds

Download

batriesyairdina@graduate.utm.my

sql2-dml

en

Copyright © 1999, 2023, Oracle and/or its affiliates.

Oracle APEX 23.2.1