



**UTM**  
UNIVERSITI TEKNOLOGI MALAYSIA

**FACULTY OF COMPUTING**  
UTM Johor Bahru

## **SECD2523 – 08 Database**

Semester I , Session 2023/2024

### **Lab 4 – DML3 Part 1, 2**

Lecturer:

Dr. Noor Hidayah Zakaria

<b>Name</b>	<b>Matrics Number</b>
Teoh Zhi Yong	A22EC0125

## DML3 Part 1

### 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 Commands pane contains the following query:

```
1 SELECT * FROM
2 SALES_REPRESENTATIVES NATURAL JOIN SALES_REP_ADDRESSES;
```

The Results pane displays the output of the query, showing 3 rows returned in 0.05 seconds. The results are as follows:

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	0154598761	10	sr01	12 Cherry Lane	Denton	Detroit	DT48211
sr02	vwright@obl.com	Victoria	Wright	0154598762	5	sr01	87 Blossom Hill	Uptown	Detroit	DT52314
sr03	bspeed@obl.com	Barry	Speed	0154598763	5	sr01	12 Junction Row	Skinflats	Detroit	DT52564

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;**

The screenshot shows the APEX SQL Workshop interface. The SQL Commands pane contains the following query:

```
1 SELECT ID, FIRST_NAME, LAST_NAME, ADDRESS_LINE_1, ADDRESS_LINE_2, CITY, EMAIL, PHONE_NUMBER
2 FROM SALES_REPRESENTATIVES NATURAL JOIN SALES_REP_ADDRESSES;
```

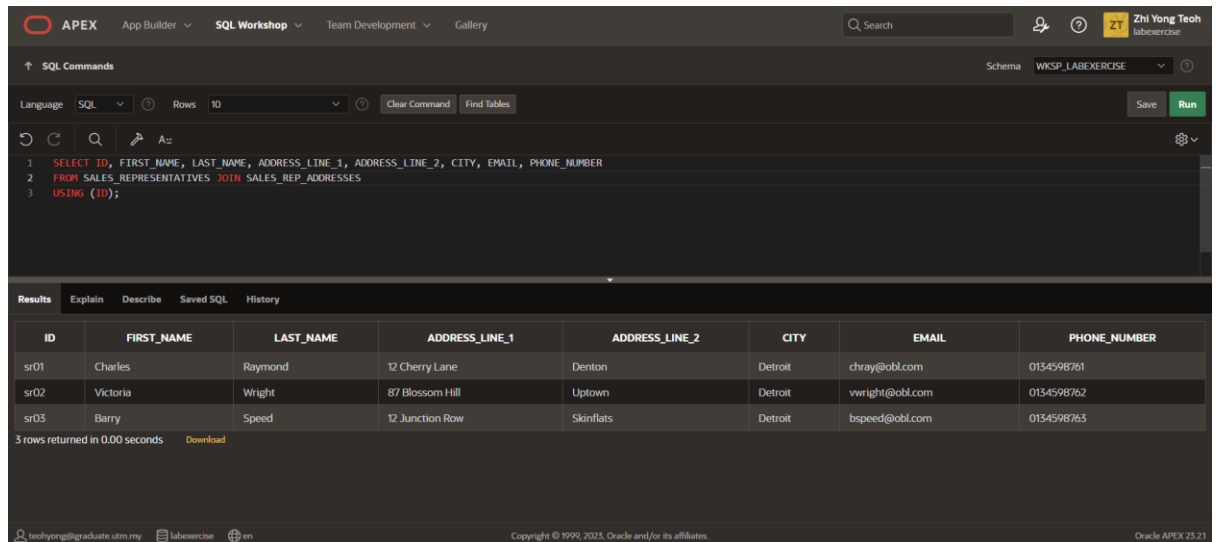
The Results pane displays the output of the query, showing 3 rows returned in 0.05 seconds. The results are as follows:

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	0154598761
sr02	Victoria	Wright	87 Blossom Hill	Uptown	Detroit	vwright@obl.com	0154598762
sr03	Barry	Speed	12 Junction Row	Skinflats	Detroit	bspeed@obl.com	0154598763

## 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);
```



The screenshot shows the Oracle APEX SQL Workshop interface. The SQL Commands pane contains the following query:

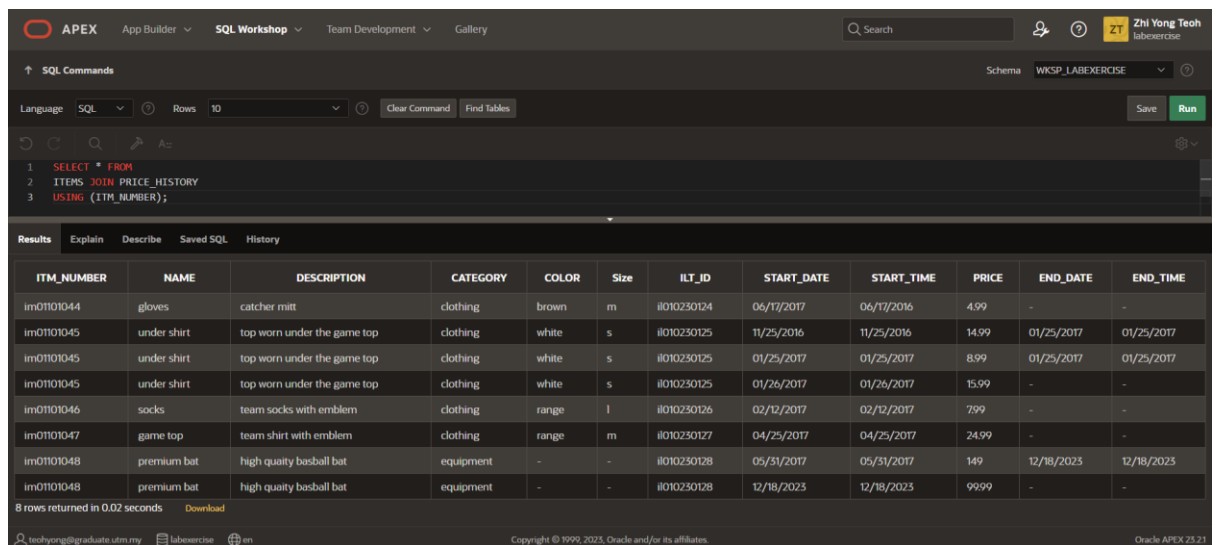
```
1 SELECT ID, FIRST_NAME, LAST_NAME, ADDRESS_LINE_1, ADDRESS_LINE_2, CITY, EMAIL, PHONE_NUMBER  
2 FROM SALES_REPRESENTATIVES JOIN SALES_REP_ADDRESSES  
3 USING (ID);
```

The Results pane shows the output of the query, which consists of 3 rows returned in 0.00 seconds. The data is as follows:

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	0154598761
sr02	Victoria	Wright	87 Blossom Hill	Uptown	Detroit	vwright@obl.com	0154598762
sr03	Barry	Speed	12 Junction Row	Skinflats	Detroit	bspeed@obl.com	0154598763

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 Commands pane contains the following query:

```
1 SELECT * FROM  
2 ITEMS JOIN PRICE_HISTORY  
3 USING (ITM_NUMBER);
```

The Results pane shows the output of the query, which consists of 8 rows returned in 0.02 seconds. The data is as follows:

ITM_NUMBER	NAME	DESCRIPTION	CATEGORY	COLOR	Size	ILT_ID	START_DATE	START_TIME	PRICE	END_DATE	END_TIME
im0101044	gloves	catcher mitt	clothing	brown	m	il010230124	06/17/2017	06/17/2016	4.99	-	-
im0101045	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
im0101045	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
im0101045	under shirt	top worn under the game top	clothing	white	s	il010230125	01/26/2017	01/26/2017	15.99	-	-
im0101046	socks	team socks with emblem	clothing	range	l	il010230126	02/12/2017	02/12/2017	7.99	-	-
im0101047	game top	team shirt with emblem	clothing	range	m	il010230127	04/25/2017	04/25/2017	24.99	-	-
im0101048	premium bat	high quality baseball bat	equipment	-	-	il010230128	05/31/2017	05/31/2017	149	12/18/2023	12/18/2023
im0101048	premium bat	high quality baseball bat	equipment	-	-	il010230128	12/18/2023	12/18/2023	9999	-	-

### Part 3: Creating Joins with the ON Clause

1. Use an ON clause to join the customer and sales representative 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.

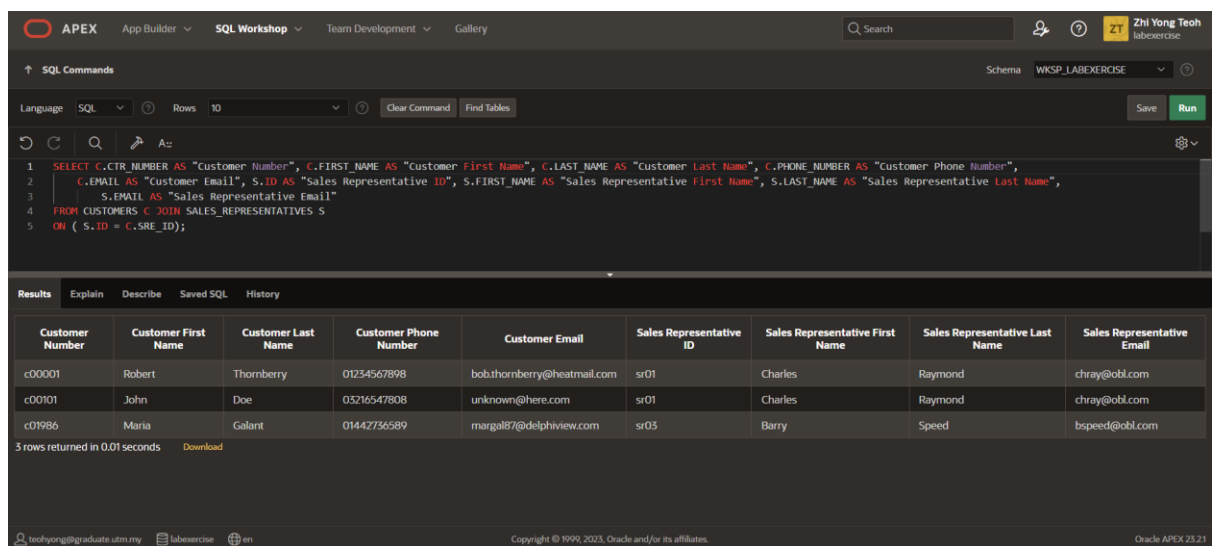
```
SELECT C.CTR_NUMBER AS "Customer Number", C.FIRST_NAME AS "Customer First Name", C.LAST_NAME AS "Customer Last Name", C.PHONE_NUMBER AS "Customer Phone Number",
```

```
C.EMAIL AS "Customer Email", S.ID AS "Sales Representative ID", S.FIRST_NAME AS "Sales Representative First Name", S.LAST_NAME AS "Sales Representative Last Name",
```

```
S.EMAIL AS "Sales Representative Email"
```

```
FROM CUSTOMERS C JOIN SALES_REPRESENTATIVES S
```

```
ON ( S.ID = C.SRE_ID);
```



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

```
1 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",
2 C.EMAIL AS "Customer Email", S.ID AS "Sales Representative ID", S.FIRST_NAME AS "Sales Representative First Name", S.LAST_NAME AS "Sales Representative Last Name",
3 S.EMAIL AS "Sales Representative Email"
4 FROM CUSTOMERS C JOIN SALES_REPRESENTATIVES S
5 ON ( S.ID = C.SRE_ID);
```

The results table shows the following data:

Customer Number	Customer First Name	Customer Last Name	Customer Phone Number	Customer Email	Sales Representative ID	Sales Representative First Name	Sales Representative Last Name	Sales Representative Email
c00001	Robert	Thornberry	01234567898	bob.thornberry@heatmail.com	sr01	Charles	Raymond	chray@obl.com
c00101	John	Doe	05216547808	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 0.01 seconds

## Part 4- Creating Three-Way Joins with the ON Clause

- Using 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 Representative ID", S.FIRST_NAME AS  
"Sales Representative First Name", S.LAST_NAME AS "Sales Representative Last Name",  
S.EMAIL AS "Sales Representative Email", T.NAME AS "Team Name"  
FROM CUSTOMERS C JOIN SALES_REPRESENTATIVES S  
ON ( S.ID = C.SRE_ID)  
JOIN TEAMS T  
ON ( C.TEM_ID = T.ID);
```

The screenshot shows the APEX SQL Workshop interface. The SQL Commands pane 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", C.PHONE_NUMBER AS "Customer Phone Number",  
2 C.EMAIL AS "Customer Email", S.ID AS "Sales Representative ID", S.FIRST_NAME AS "Sales Representative First Name", S.LAST_NAME AS "Sales Representative Last Name",  
3 S.EMAIL AS "Sales Representative Email", T.NAME AS "Team Name"  
4 FROM CUSTOMERS C JOIN SALES_REPRESENTATIVES S  
5 ON ( S.ID = C.SRE_ID)  
6 JOIN TEAMS T  
7 ON ( C.TEM_ID = T.ID);
```

The Results pane shows the following data:

Customer Number	Customer First Name	Customer Last Name	Customer Phone Number	Customer Email	Sales Representative ID	Sales Representative First Name	Sales Representative Last Name	Sales Representative Email	Team Name
c00001	Robert	Thornberry	01234567898	bob.thornberry@heatmail.com	sr01	Charles	Raymond	chray@obl.com	Rockets
c00101	John	Doe	05216547808	unknown@here.com	sr01	Charles	Raymond	chray@obl.com	Celtics
c01986	Maria	Galanit	01442736589	margal87@delphiview.com	sr03	Barry	Speed	bspeed@obl.com	Rovers

3 rows returned in 0.05 seconds

## Part 5: Applying Additional Conditions to a Join

- Using 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 Representative ID", S.FIRST_NAME AS "Sales Representative First Name", S.LAST_NAME AS "Sales Representative Last Name",  
       S.EMAIL AS "Sales Representative Email", T.NAME AS "Team Name"  
FROM CUSTOMERS C JOIN SALES_REPRESENTATIVES S  
ON ( S.ID = C.SRE_ID)  
JOIN TEAMS T  
ON ( C.TEM_ID = T.ID)  
WHERE C.CTR_NUMBER = 'c00001';
```

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 'Zhi Yong Teoh' are on the right. The 'SQL Commands' tab is active, displaying a SQL query. The query is as follows:

```
1 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",  
2       C.EMAIL AS "Customer Email", S.ID AS "Sales Representative ID", S.FIRST_NAME AS "Sales Representative First Name", S.LAST_NAME AS "Sales Representative Last Name",  
3       S.EMAIL AS "Sales Representative Email", T.NAME AS "Team Name"  
4 FROM CUSTOMERS C JOIN SALES_REPRESENTATIVES S  
5 ON ( S.ID = C.SRE_ID)  
6 JOIN TEAMS T  
7 ON ( C.TEM_ID = T.ID)  
8 WHERE C.CTR_NUMBER = 'c00001';
```

Below the query editor, the 'Results' tab is selected, showing a table with 10 columns: Customer Number, Customer First Name, Customer Last Name, Customer Phone Number, Customer Email, Sales Representative ID, Sales Representative First Name, Sales Representative Last Name, Sales Representative Email, and Team Name. The table contains one row of data for customer 'c00001'.

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

At the bottom of the results section, it states '1 rows returned in 0.01 seconds' with a 'Download' link.

## Part 6: Retrieving Records with Nonequijoins

1. Write a query that will display name and 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 ' || P.PRICE AS "Item Details"  
FROM ITEMS I JOIN PRICE_HISTORY P  
ON (TO_DATE('12-Dec-2016', 'DD-MM-YYYY') BETWEEN P.START_DATE AND  
P.END_DATE) AND (I.ITM_NUMBER = 'im01101045');
```

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 'Zhi Yong Teoh' are on the right. The 'SQL Commands' tab is active, showing a query with three lines. Below the query editor, the 'Results' tab is selected, displaying a table titled 'Item Details' with one row: 'The cost of the under shirt on this day was 14.99'. The status bar at the bottom indicates '1 rows returned in 0.01 seconds' and includes a 'Download' link. The footer shows the user 'teohyong@graduate.utm.my', the workspace 'labexercise', and the Oracle APEX version '23.21'.

```
1 SELECT 'The cost of the ' || I.NAME || ' on this day was ' || P.PRICE AS "Item Details"  
2 FROM ITEMS I JOIN PRICE_HISTORY P  
3 ON (TO_DATE('12-Dec-2016', 'DD-MM-YYYY') BETWEEN P.START_DATE AND P.END_DATE) AND (I.ITM_NUMBER = 'im01101045');
```

Item Details
The cost of the under shirt on this day was 14.99

1 rows returned in 0.01 seconds [Download](#)

## DML3 Part 2

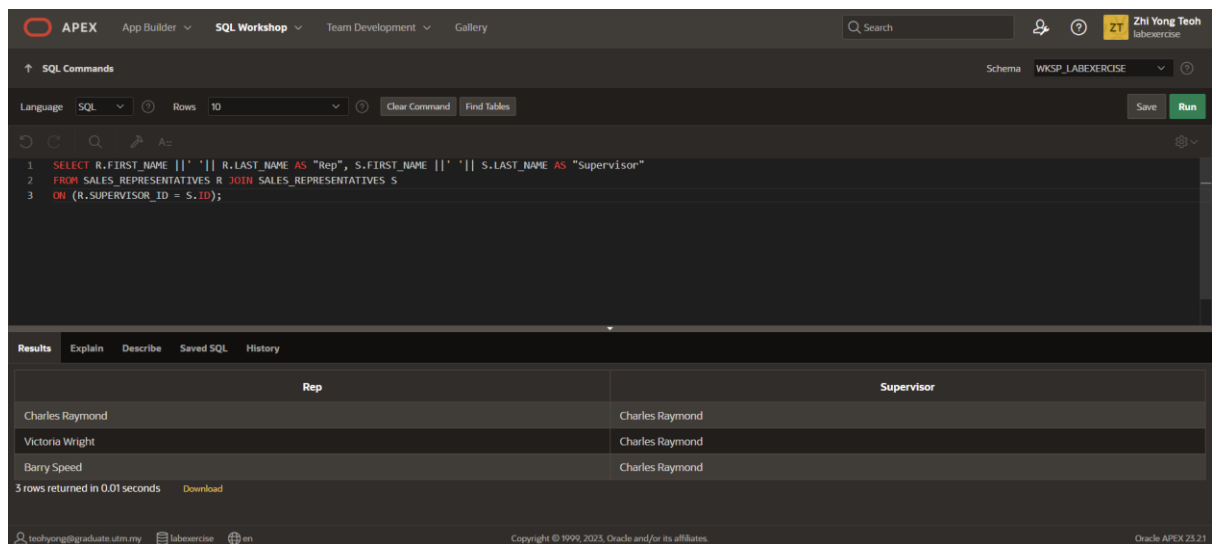
### Section 6 Lesson 9 Exercise 2: Joining Tables Using JOIN

#### Write SELECT Statements Using Data From Multiple Tables Using Equijoins and Non-Equijoins (S6L9 Objective 1)

##### Part 1 : Use a Self-Join to Join a Table to Itself (S6L9 Objective 2)

1. Write a query that will display who the supervisor is for each of the sales representatives. The information should be displayed in two columns, the first column will be the first name and last name of the sales representative and the second will be the first name and last name of the supervisor. The column aliases should be Rep and Supervisor.

```
SELECT R.FIRST_NAME || ' ' || R.LAST_NAME AS "Rep", S.FIRST_NAME || ' ' || S.LAST_NAME AS "Supervisor"
FROM SALES_REPRESENTATIVES R JOIN SALES_REPRESENTATIVES S
ON (R.SUPERVISOR_ID = S.ID);
```



The screenshot shows the APEX SQL Workshop interface. The SQL Commands pane contains the following query:

```
1 SELECT R.FIRST_NAME || ' ' || R.LAST_NAME AS "Rep", S.FIRST_NAME || ' ' || S.LAST_NAME AS "Supervisor"
2 FROM SALES_REPRESENTATIVES R JOIN SALES_REPRESENTATIVES S
3 ON (R.SUPERVISOR_ID = S.ID);
```

The Results pane shows the output of the query, which is a table with two columns: Rep and Supervisor. The table contains three rows of data:

Rep	Supervisor
Charles Raymond	Charles Raymond
Victoria Wright	Charles Raymond
Barry Speed	Charles Raymond

3 rows returned in 0.01 seconds. Download



## Part 2 : Use OUTER joins (S6L9 Objective 3)

1. Write a query that will display all of the team and customer information even if there is no match with the table on the left (team).

```
SELECT * FROM
```

```
TEAMS T LEFT OUTER JOIN CUSTOMERS C
```

```
ON (T.ID = C.TEM_ID);
```

The screenshot shows the Oracle APEX SQL Workshop interface. The SQL command area contains the following query:

```
1 SELECT * FROM
2 TEAMS T LEFT OUTER JOIN CUSTOMERS C
3 ON (T.ID = C.TEM_ID);
```

The Results tab is selected, displaying a table with 13 columns: ID, NAME, NUMBER\_OF\_PLAYERS, DISCOUNT, CTR\_NUMBER, EMAIL, FIRST\_NAME, LAST\_NAME, PHONE\_NUMBER, CURRENT\_BALANCE, SRE\_ID, TEM\_ID, and LOYALTY\_CARD\_NUMBER. The table contains 4 rows of data.

ID	NAME	NUMBER_OF_PLAYERS	DISCOUNT	CTR_NUMBER	EMAIL	FIRST_NAME	LAST_NAME	PHONE_NUMBER	CURRENT_BALANCE	SRE_ID	TEM_ID	LOYALTY_CARD_NUMBER
t001	Rockets	25	10	c00001	bob.thornberry@heatmail.com	Robert	Thornberry	01234567898	150	sr01	t001	-
t002	Celtics	42	20	c00101	unknown@here.com	John	Doe	03216547808	987.5	sr01	t002	-
t003	Rovers	8	-	c01986	margal87@delphiview.com	Maria	Galant	01442736589	125.65	sr03	t003	-
t004	Jets	10	5	-	-	-	-	-	-	-	-	-

4 rows returned in 0.01 seconds [Download](#)

The footer of the interface shows the user 'Zhi Yong Teoh' with the email 'teohyong@graduate.utm.my', the schema 'WKSP\_LABEXERCISE', and the Oracle APEX version '23.21'.

## Part 3 : Generating a Cartesian Product (S6L9 Objective 4)

1. Create a Cartesian product between the customer and sales representative tables.

**SELECT \* FROM**

**CUSTOMERS CROSS JOIN SALES\_REPRESENTATIVES;**

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

```
1 SELECT * FROM
2 CUSTOMERS CROSS JOIN SALES_REPRESENTATIVES;
```

The results are displayed in a table with the following columns: CTR\_NUMBER, EMAIL, FIRST\_NAME, LAST\_NAME, PHONE\_NUMBER, CURRENT\_BALANCE, SRE\_ID, TEM\_ID, LOYALTY\_CARD\_NUMBER, ID, EMAIL, FIRST\_NAME, LAST\_N. The table contains 18 rows, which is the Cartesian product of the 6 rows in the CUSTOMERS table and the 3 rows in the SALES\_REPRESENTATIVES table.

CTR_NUMBER	EMAIL	FIRST_NAME	LAST_NAME	PHONE_NUMBER	CURRENT_BALANCE	SRE_ID	TEM_ID	LOYALTY_CARD_NUMBER	ID	EMAIL	FIRST_NAME	LAST_N
c00001	bob.thornberry@heatmail.com	Robert	Thornberry	01234567898	150	sr01	t001	-	sr01	chray@obl.com	Charles	Raymonc
c00012	j.jones@freemail.com	Jennifer	Jones	01505214598	0	-	-	lc1015	sr01	chray@obl.com	Charles	Raymonc
c00101	unknown@here.com	John	Doe	03216547808	987.5	sr01	t002	-	sr01	chray@obl.com	Charles	Raymonc
c00103	MurciaA@globaltech.com	Andrew	Murcia	07715246890	85	-	-	lc2341	sr01	chray@obl.com	Charles	Raymonc
c01986	margal87@delphiview.com	Maria	Galant	01442736589	125.65	sr03	t003	-	sr01	chray@obl.com	Charles	Raymonc
c02001	brianrog@hootech.com	Brian	Rogers	01654564898	50	-	-	lc4587	sr01	chray@obl.com	Charles	Raymonc
c00001	bob.thornberry@heatmail.com	Robert	Thornberry	01234567898	150	sr01	t001	-	sr02	vwright@obl.com	Victoria	Wright
c00012	j.jones@freemail.com	Jennifer	Jones	01505214598	0	-	-	lc1015	sr02	vwright@obl.com	Victoria	Wright
c00101	unknown@here.com	John	Doe	03216547808	987.5	sr01	t002	-	sr02	vwright@obl.com	Victoria	Wright
c00103	MurciaA@globaltech.com	Andrew	Murcia	07715246890	85	-	-	lc2341	sr02	vwright@obl.com	Victoria	Wright

Row 1-10

This table shows the first 10 rows of the Cartesian product result set, which are identical to the first 10 rows shown in the previous table.

CTR_NUMBER	EMAIL	FIRST_NAME	LAST_NAME	PHONE_NUMBER	CURRENT_BALANCE	SRE_ID	TEM_ID	LOYALTY_CARD_NUMBER	ID	EMAIL	FIRST_NAME	LAST_N
c00001	bob.thornberry@heatmail.com	Robert	Thornberry	01234567898	150	sr01	t001	-	sr01	chray@obl.com	Charles	Raymonc
c00012	j.jones@freemail.com	Jennifer	Jones	01505214598	0	-	-	lc1015	sr01	chray@obl.com	Charles	Raymonc
c00101	unknown@here.com	John	Doe	03216547808	987.5	sr01	t002	-	sr01	chray@obl.com	Charles	Raymonc
c00103	MurciaA@globaltech.com	Andrew	Murcia	07715246890	85	-	-	lc2341	sr01	chray@obl.com	Charles	Raymonc
c01986	margal87@delphiview.com	Maria	Galant	01442736589	125.65	sr03	t003	-	sr01	chray@obl.com	Charles	Raymonc
c02001	brianrog@hootech.com	Brian	Rogers	01654564898	50	-	-	lc4587	sr01	chray@obl.com	Charles	Raymonc
c00001	bob.thornberry@heatmail.com	Robert	Thornberry	01234567898	150	sr01	t001	-	sr02	vwright@obl.com	Victoria	Wright
c00012	j.jones@freemail.com	Jennifer	Jones	01505214598	0	-	-	lc1015	sr02	vwright@obl.com	Victoria	Wright
c00101	unknown@here.com	John	Doe	03216547808	987.5	sr01	t002	-	sr02	vwright@obl.com	Victoria	Wright
c00103	MurciaA@globaltech.com	Andrew	Murcia	07715246890	85	-	-	lc2341	sr02	vwright@obl.com	Victoria	Wright

Row 11-18

This table shows the last 8 rows of the Cartesian product result set, which are identical to the last 8 rows shown in the previous table.

c01986	margal87@delphiview.com	Maria	Galant	01442736589	125.65	sr03	t003	-	sr02	vwright@obl.com	Victoria	Wright
c02001	brianrog@hootech.com	Brian	Rogers	01654564898	50	-	-	lc4587	sr02	vwright@obl.com	Victoria	Wright
c00001	bob.thornberry@heatmail.com	Robert	Thornberry	01234567898	150	sr01	t001	-	sr03	bspeed@obl.com	Barry	Speed
c00012	j.jones@freemail.com	Jennifer	Jones	01505214598	0	-	-	lc1015	sr03	bspeed@obl.com	Barry	Speed
c00101	unknown@here.com	John	Doe	03216547808	987.5	sr01	t002	-	sr03	bspeed@obl.com	Barry	Speed
c00103	MurciaA@globaltech.com	Andrew	Murcia	07715246890	85	-	-	lc2341	sr03	bspeed@obl.com	Barry	Speed
c01986	margal87@delphiview.com	Maria	Galant	01442736589	125.65	sr03	t003	-	sr03	bspeed@obl.com	Barry	Speed
c02001	brianrog@hootech.com	Brian	Rogers	01654564898	50	-	-	lc4587	sr03	bspeed@obl.com	Barry	Speed

18 rows returned in 0.01 seconds [Download](#)