



UTM
UNIVERSITI TEKNOLOGI MALAYSIA

FACULTY OF COMPUTING
UTM Johor Bahru

SESSION 2023/2024 SEMESTER 1

SECD2523
DATABASE

LAB EXERCISE : SQL DML3

NAME
NURAISSA SALSABILA BINTI MOHD FADZAL

MATRIC NO
A22EC0249

LECTURER'S NAME
DR. ROZILAWATI BINTI DOLLAH

SECTION
10

Section 6 Lesson 9 Exercise 1: Joining Tables Using JOIN

Part 1: Creating Natural Joins.

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

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

```
1 SELECT *
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	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 0.05 seconds [Download](#)

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.

The screenshot shows the Oracle APEX SQL Workshop interface. The query editor contains the following SQL code:

```
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 a table with 8 columns: ID, FIRST_NAME, LAST_NAME, ADDRESS_LINE_1, ADDRESS_LINE_2, CITY, EMAIL, and PHONE_NUMBER. The table contains 3 rows of data.

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

Part 2: Creating Joins with the USING Clause

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

The screenshot shows the Oracle APEX SQL Workshop interface. The query editor contains the following SQL code:

```
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 displays a table with 8 columns: ID, FIRST_NAME, LAST_NAME, ADDRESS_LINE_1, ADDRESS_LINE_2, CITY, EMAIL, and PHONE_NUMBER. The table contains 3 rows of data.

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.

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

```
1 SELECT *
2 FROM items JOIN price_history
3 USING (item_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 table contains 8 rows of data.

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	12/09/2023	12/09/2023	99.99	-	-
im0101044	gloves	catcher mitt	clothing	brown	m	il010230124	06/17/2017	06/17/2016	4.99	12/09/2023	12/09/2023
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	-	-

8 rows returned in 0.04 seconds

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.

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

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

The results show 3 rows returned in 0.01 seconds. The data is as follows:

Customer Number	First Name	Last Name	Phone Number	Customer Email	Sale Representative ID	Sale Representative First Name	Sale Representative Last Name	Sale Representative 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	Galand	01442736589	margal87@delphiview.com	sr03	Barry	Speed	bspeed@obl.com

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

1. 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.

The screenshot shows the Oracle APEX SQL Workshop interface with a three-way join query. The SQL command entered is:

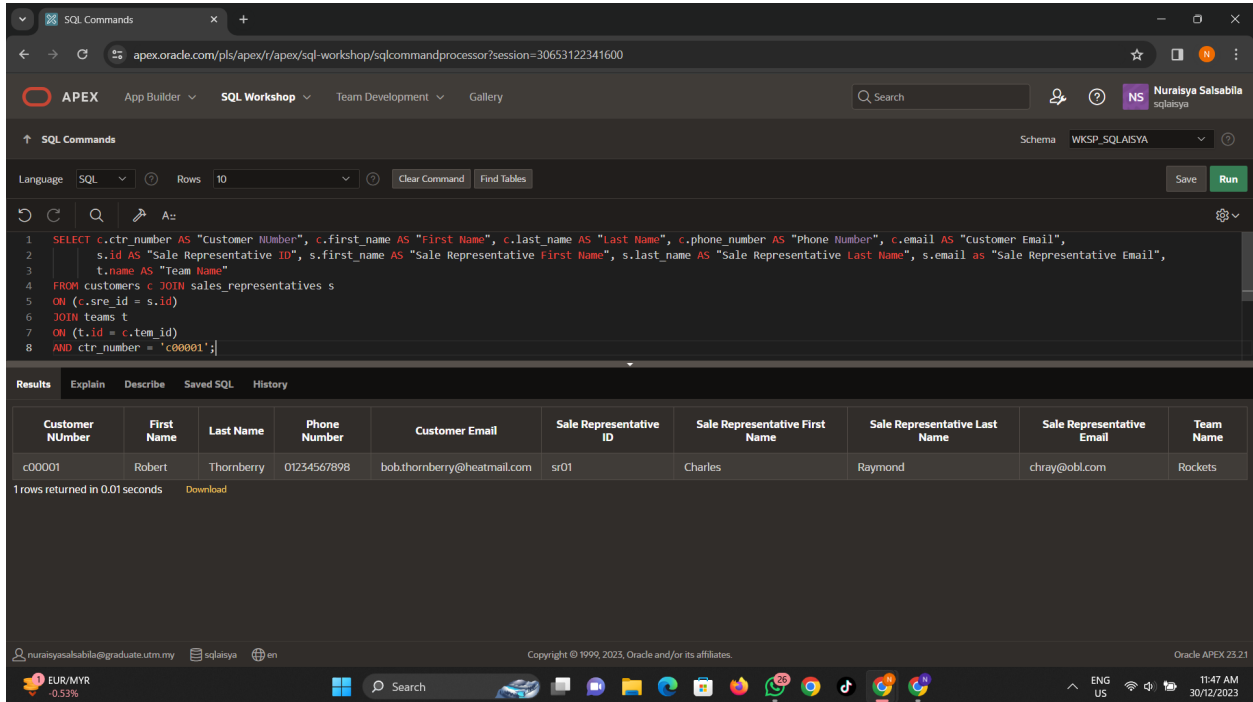
```
1 SELECT c.ctr_number AS "Customer Number", c.first_name AS "First Name", c.last_name AS "Last Name", c.phone_number AS "Phone Number", c.email AS "Customer Email",
2       s.id AS "Sale Representative ID", s.first_name AS "Sale Representative First Name", s.last_name AS "Sale Representative Last Name", s.email AS "Sale Representative Email",
3       t.name AS "Team Name"
4 FROM customers c JOIN sales_representatives s
5 ON (c.sre_id = s.id)
6 JOIN teams t
7 ON (t.id = c.tem_id);
```

The results show 3 rows returned in 0.04 seconds. The data is as follows:

Customer Number	First Name	Last Name	Phone Number	Customer Email	Sale Representative ID	Sale Representative First Name	Sale Representative Last Name	Sale Representative Email	Team Name
c00001	Robert	Thornberry	01234567898	bob.thornberry@heatmail.com	sr01	Charles	Raymond	chray@obl.com	Rockets
c00101	John	Doe	03216547808	unknown@here.com	sr01	Charles	Raymond	chray@obl.com	Celtics
c01986	Maria	Galand	01442736589	margal87@delphiview.com	sr03	Barry	Speed	bspeed@obl.com	Rovers

Part 5: Applying Additional Conditions to a Join

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



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

```
1 SELECT c.ctr_number AS "Customer Number", c.first_name AS "First Name", c.last_name AS "Last Name", c.phone_number AS "Phone Number", c.email AS "Customer Email",  
2       s.id AS "Sale Representative ID", s.first_name AS "Sale Representative First Name", s.last_name AS "Sale Representative Last Name", s.email AS "Sale Representative Email",  
3       t.name AS "Team Name"  
4 FROM customers c JOIN sales_representatives s  
5 ON (c.sre_id = s.id)  
6 JOIN teams t  
7 ON (t.id = c.tem_id)  
8 AND ctr_number = 'c00001';
```

The Results tab shows the following data:

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

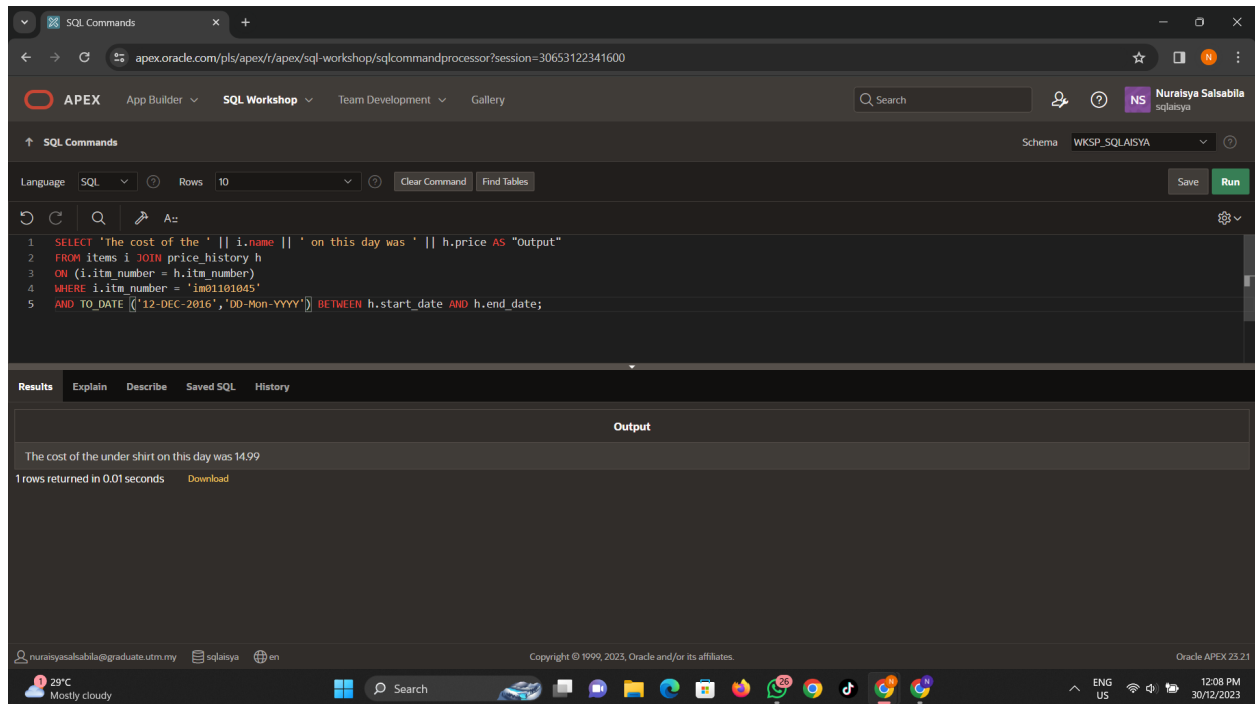
1 rows returned in 0.01 seconds

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



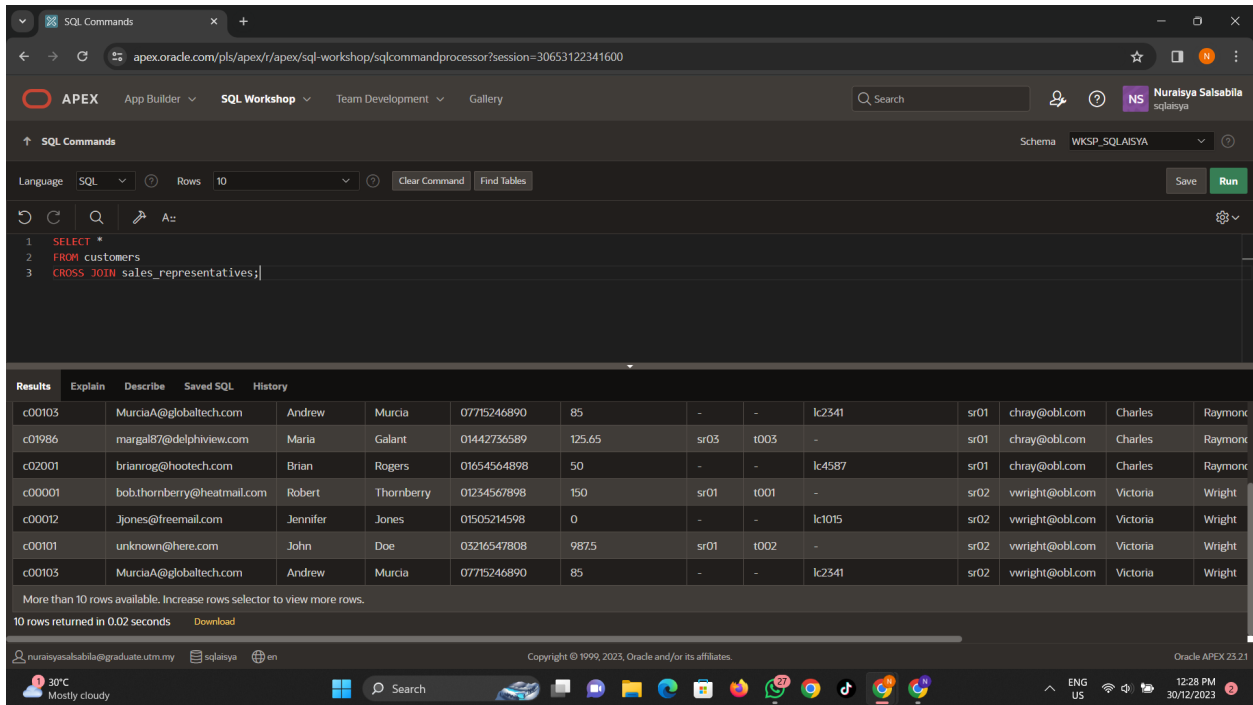
Section 6 Lesson 9 Exercise 2: Joining Tables Using JOIN

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.

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

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



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

```
1 SELECT *
2 FROM customers
3 CROSS JOIN sales_representatives;
```

The results table shows 10 rows of data, including columns for customer ID, email, first name, last name, phone number, credit limit, sales representative ID, and commission percentage.

Customer ID	Customer Email	Customer First Name	Customer Last Name	Customer Phone Number	Customer Credit Limit	Sales Representative ID	Sales Representative Commission %
c00103	MurciaA@globaltech.com	Andrew	Murcia	0775246890	85	-	-
c01986	margal87@dolphiview.com	Maria	Galant	01442736589	125.65	sr03	t003
c02001	brianrog@hooitech.com	Brian	Rogers	01654564898	50	-	-
c00001	bob.thornberry@heatmail.com	Robert	Thornberry	01234567898	150	sr01	t001
c00012	Jjones@freemail.com	Jennifer	Jones	01505214598	0	-	-
c00101	unknown@here.com	John	Doe	05216547808	987.5	sr01	t002
c00103	MurciaA@globaltech.com	Andrew	Murcia	0775246890	85	-	-

More than 10 rows available. Increase rows selector to view more rows.

10 rows returned in 0.02 seconds