



UTM
UNIVERSITI TEKNOLOGI MALAYSIA

Database Lab4:DML3

SECD2523 – 08 Database

Semester 1 2023/2024

CHUAH KEE YONG A22EC0150

Lecturer's Name : Dr. Noor Hidayah Zakaria

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.

Copyright © 2020, Oracle and/or its affiliates. All rights reserved. Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Part 1: Creating Natural Joins.

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

```
1 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		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 0.04 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.

```
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;
3
```

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 0.00 seconds [Download](#)

Part 2: Creating Joins with the USING Clause

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

```

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);
4

```

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 0.03 seconds [Download](#)

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

```

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

```

ITM_NUMBER	NAME	DESCRIPTION	CATEGORY	COLOR	Size	ILT_ID	START_DATE	START_TIME	PRICE	END_DATE	END_TIME
im0101048	premium bat	high quality baseball bat	equipment	-	-	il010230128	12/19/2023	12/19/2023	99.99	-	-
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	12/19/2023	12/19/2023

8 rows returned in 0.03 seconds [Download](#)

Part 3: Creating Joins with the ON Clause

- 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.

```

1 SELECT c.ctr_number "Customer Number", c.first_name "Customer First Name",
2 c.last_name "Customer Last Name", c.phone_number "Customer Phone Number",
3 c.email "Customer Email", s.id "Sales Representative ID", s.first_name "Sales
4 Representative First Name", s.last_name "Sales Representative Last Name", s.email
5 "Sales Representative Email"
6 FROM customers c JOIN sales_representatives s
7 ON ( s.id = c.sre_id);
8

```

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	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 0.01 seconds [Download](#)

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.

```

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

```

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	03216547808	unknown@here.com	sr01	Charles	Raymond	chray@obl.com	Celtics
c01986	Maria	Galant	01442736589	margal87@delphiview.com	sr03	Barry	Speed	bspeed@obl.com	Rovers

3 rows returned in 0.02 seconds [Download](#)

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.

```

1 SELECT c.ctr_number "Customer Number", c.first_name "Customer First Name",
2 c.last_name "Customer Last Name", c.phone_number "Customer Phone Number",
3 c.email "Customer Email", s.id "Sales Representative ID", s.first_name "Sales
4 Representative First Name", s.last_name "Sales Representative Last Name", s.email
5 "Sales Representative Email", t.name "Team Name"
6 FROM customers c JOIN sales_representatives s
7 ON ( s.id = c.sre_id)
8 JOIN teams t
9 ON ( c.tem_id = t.id)
10 WHERE c.ctr_number='c00001';
11

```

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

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

Part 6: Retrieving Records with Nonequijoins

- 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

```

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
4 p.end_date) AND (i.item_number = 'im01101045');

```

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

1 rows returned in 0.03 seconds [Download](#)

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.

```
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);
4
```

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).

```
1 SELECT * FROM teams t LEFT OUTER JOIN customers c
2 ON (t.id = c.tem_id);
3
```

ID	NAME	NUMBER_OF_PLAYERS	DISCOUNT	CTR_NUMBER	EMAIL	FIRST_NAME	LAST_NAME	PHONE_NUMBER	CURRENT_BALANCE	SRE_ID	TEM_ID	LOYALTY_CARD_NUMBER
1001	Rockets	25	10	c00001	bob.thornberry@heatmail.com	Robert	Thornberry	01234567898	150	sr01	1001	-
1002	Celtics	42	20	c00101	unknown@here.com	John	Doe	03216547808	987.5	sr01	1002	-
1003	Rovers	8	-	c01986	margal87@delphiview.com	Maria	Galant	01442736589	125.65	sr03	1003	-
1004	Jets	10	5	-	-	-	-	-	-	-	-	-

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

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

1 SELECT * FROM customers CROSS JOIN sales_representatives;																
Results Explain Describe Saved SQL History																
CTE_NUMBER	EMAIL	FIRST_NAME	LAST_NAME	PHONE_NUMBER	CURRENT_BALANCE	SHE_ID	TEH_ID	LOYALTY_CARD_NUMBER	ID	EMAIL	FIRST_NAME	LAST_NAME	PHONE_NUMBER	COMMISSION_RATE	SUPERVISOR_ID	
000001	branning@hooitech.com	Brian	Rogers	0154644898	50	-	-	IC4687	S101	chrag@globl.com	Charles	Raymond	01546198761	10	S101	
000001	bob.thornberry@heemal.com	Robert	Thornberry	01234567898	150	S101	1001	-	S101	chrag@globl.com	Charles	Raymond	01546198761	10	S101	
000002	jones@freemal.com	Jennifer	Jones	01505214598	0	-	-	IC1015	S101	chrag@globl.com	Charles	Raymond	01546198761	10	S101	
000101	unknown@here.com	John	Doe	03216547808	9875	S101	1002	-	S101	chrag@globl.com	Charles	Raymond	01546198761	10	S101	
000103	Murcia@globtech.com	Andrew	Murcia	07715346890	85	-	-	IC3341	S101	chrag@globl.com	Charles	Raymond	01546198761	10	S101	
001989	margal8@dephview.com	Maria	Galant	01442735587	125.65	S103	1003	-	S101	chrag@globl.com	Charles	Raymond	01546198761	10	S101	
002001	branning@hooitech.com	Brian	Rogers	0154644898	50	-	-	IC4687	S102	vwright@globl.com	Victoria	Wright	01546198762	5	S101	
000001	bob.thornberry@heemal.com	Robert	Thornberry	01234567898	150	S101	1001	-	S102	vwright@globl.com	Victoria	Wright	01546198762	5	S101	
000002	jones@freemal.com	Jennifer	Jones	01505214598	0	-	-	IC1015	S102	vwright@globl.com	Victoria	Wright	01546198762	5	S101	
000101	unknown@here.com	John	Doe	03216547808	9875	S101	1002	-	S102	vwright@globl.com	Victoria	Wright	01546198762	5	S101	
More than 10 rows available. Increase rows selector to view more rows.																
87 rows returned in 0.02 seconds Download																

Copyright © 2020, Oracle and/or its affiliates. All rights reserved. Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.