



UTM
UNIVERSITI TEKNOLOGI MALAYSIA

FACULTY OF COMPUTING
UTM Johor Bahru

LAB 4

DATA MANIPULATION LANGUAGE (DML 3)

SECD2523-Database

SEMESTER I, SESSION 2023/2024

Lecturer: Dr. Noor Hidayah

Name	Matric No
LEE XIN HUI	A22EC0066

Section: 08

DML 3 PART 1

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;
```

Results	Explain	Describe	Saved SQL	History							
ID	EMAIL	FIRST_NAME	LAST_NAME	PHONE_NUMBER	COMMISSION_RATE	SUPERVISOR_ID	ADDRESS_LINE_1	ADDRESS_LINE_2	CITY	ZIP_CODE	
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	
sr01	chray@obl.com	Charles	Raymond	0134598761	10	sr01	12 Cherry Lane	Denton	Detroit	DT48211	
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.

```
SELECT id, first_name, last_name, address_line_1, address_line_2, city, email, phone_number
FROM sales_representatives 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
sr02	Victoria	Wright	87 Blossom Hill	Uptown	Detroit	vwright@obl.com	0134598762
sr03	Barry	Speed	12 Junction Row	Skinflats	Detroit	bspeed@obl.com	0134598763
sr01	Charles	Raymond	12 Cherry Lane	Denton	Detroit	chray@obl.com	0134598761

3 rows returned in 0.04 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.

```
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);
```

Results	Explain	Describe	Saved SQL	History			
ID	FIRST_NAME	LAST_NAME	ADDRESS_LINE_1	ADDRESS_LINE_2	CITY	EMAIL	PHONE_NUMBER
sr02	Victoria	Wright	87 Blossom Hill	Uptown	Detroit	vwright@obl.com	0134598762
sr03	Barry	Speed	12 Junction Row	Skinflats	Detroit	bspeed@obl.com	0134598763
sr01	Charles	Raymond	12 Cherry Lane	Denton	Detroit	chray@obl.com	0134598761
3 rows returned in 0.01 seconds Download							

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);
```

Results	Explain	Describe	Saved SQL	History							
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	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	01/26/2017	01/26/2017	15.99	-	-
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
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	12/16/2023	12/16/2023
im01101044	gloves	catcher mitt	clothing	brown	m	il010230124	06/17/2017	06/17/2016	4.99	-	-
im01101048	premium bat	high quaity basball bat	equipment	-	-	il010230128	12/16/2023	12/16/2023	99.99	-	-
im01101046	socks	team socks with emblem	clothing	range	l	il010230126	02/12/2017	02/12/2017	7.99	-	-
8 rows returned in 0.02 seconds Download											

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 "Customer Number", c.first_name "Customer First Name", c.last_name
"Customer Last Name", c.phone_number "Customer Phone Number", c.email "Customer Email", s.id
"Sales Representative ID" , s.first_name "Sales Representative First Name",s.last_name "Sales
Representative Last Name", s.email "Sales Representative Email"
FROM customers c JOIN sales_representatives s
ON ( s.id = c.sre_id);
```

Results	Explain	Describe	Saved SQL	History				
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
c01986	Maria	Galant	01442736589	margal87@delphiview.com	sr03	Barry	Speed	bspeed@obl.com
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
3 rows returned in 0.02 seconds					Download			

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.

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

Results	Explain	Describe	Saved SQL	History					
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
c01986	Maria	Galant	01442736589	margal87@delphiview.com	sr03	Barry	Speed	bspeed@obl.com	Rovers
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
3 rows returned in 0.02 seconds Download									

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.

```
SELECT c.ctr_number "Customer Number", c.first_name "Customer First Name", c.last_name  
"Customer Last Name", c.phone_number "Customer Phone Number", c.email "Customer Email",  
s.id "Sales Representative ID", s.first_name "Sales Representative First Name", s.last_name "Sales  
Representative Last Name", s.email "Sales Representative Email", t.name "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';
```

Results	Explain	Describe	Saved SQL	History					
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

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');
```

Results	Explain	Describe	Saved SQL	History
Item Details				
The cost of the under shirt on this day was 14.99				
1 rows returned in 0.03 seconds Download				

DML 3 PART 2

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);
```

Results	Explain	Describe	Saved SQL	History
Rep		Supervisor		
Victoria Wright		Charles Raymond		
Barry Speed		Charles Raymond		
Charles Raymond		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);
```

Results	Explain	Describe	Saved SQL	History								
ID	NAME	NUMBER_OF_PLAYERS	DISCOUNT	CTR_NUMBER	EMAIL	FIRST_NAME	LAST_NAME	PHONE_NUMBER	CURRENT_BALANCE	SRE_ID	TEM_ID	LOYALTY_CARD_NUMBER
t003	Rovers	8	-	c01986	margal87@delphiview.com	Maria	Galant	01442736589	125.65	sr03	t003	-
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	-
t004	Jets	10	5	-	-	-	-	-	-	-	-	-
4 rows returned in 0.02 seconds Download												

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;
```

Results	Explain	Describe	Save SQL	History												
CTR_NUMBER	EMAIL	FIRST_NAME	LAST_NAME	PHONE_NUMBER	CURRENT_BALANCE	SRE_ID	TEM_ID	LOYALTY_CARD_NUMBER	ID	EMAIL	FIRST_NAME	LAST_NAME	PHONE_NUMBER	COMMISSION_RATE	SUPERVISOR_ID	
c00103	MurciaA@globaltech.com	Andrew	Murcia	0775244890	85	-	-	lc2341	sr02	vwright@pobl.com	Victoria	Wright	0134598762	5	sr01	
c01986	margal87@delphview.com	Maria	Galant	01442736589	125.65	sr03	1003	-	sr02	vwright@pobl.com	Victoria	Wright	0134598762	5	sr01	
c00012	Jjones@freemail.com	Jennifer	Jones	01505214598	0	-	-	lc1015	sr02	vwright@pobl.com	Victoria	Wright	0134598762	5	sr01	
c00001	bob.thornberry@heatmail.com	Robert	Thornberry	01234567898	150	sr01	1001	-	sr02	vwright@pobl.com	Victoria	Wright	0134598762	5	sr01	
c00101	unknown@here.com	John	Doe	03216547808	987.5	sr01	1002	-	sr02	vwright@pobl.com	Victoria	Wright	0134598762	5	sr01	
c02001	brianog@hootech.com	Brian	Rogers	01654564898	50	-	-	lc4587	sr02	vwright@pobl.com	Victoria	Wright	0134598762	5	sr01	
c00103	MurciaA@globaltech.com	Andrew	Murcia	0775244890	85	-	-	lc2341	sr03	bspeed@pobl.com	Berry	Speed	0134598763	5	sr01	
c01986	margal87@delphview.com	Maria	Galant	01442736589	125.65	sr03	1003	-	sr03	bspeed@pobl.com	Barry	Speed	0134598763	5	sr01	
c00012	Jjones@freemail.com	Jennifer	Jones	01505214598	0	-	-	lc1015	sr03	bspeed@pobl.com	Barry	Speed	0134598763	5	sr01	
c00001	bob.thornberry@heatmail.com	Robert	Thornberry	01234567898	150	sr01	1001	-	sr03	bspeed@pobl.com	Barry	Speed	0134598763	5	sr01	
More than 10 rows available. Increase rows selector to view more rows.																
10 rows returned in 0.02 seconds Download																