

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;



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;



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);

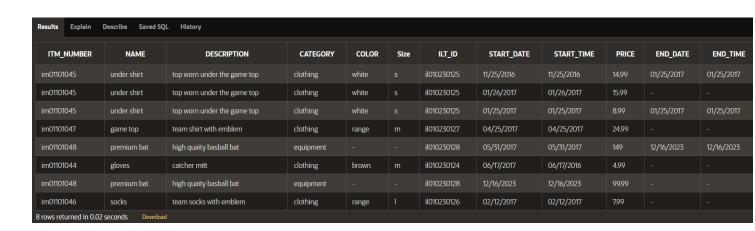


2. Display all of the information about items and their price history by joining the items and price_history tables.

SELECT *

FROM items JOIN price history

USING (itm number);



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 fist 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.	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.

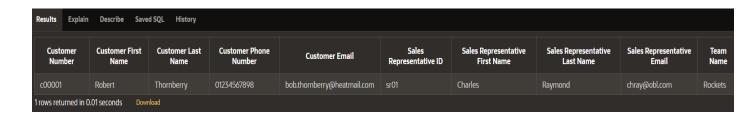
```
SELECT c.ctr_number "Customer Number", c.first_name "Customer First Name", c.last_name "Customer Last Name", c.phone_number "Customer Phone Number", c.email "Customer Email", s.id "Sales Representative ID", s.first_name "Sales Representative First Name", s.last_name "Sales Representative Last Name", 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';
```



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');



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.

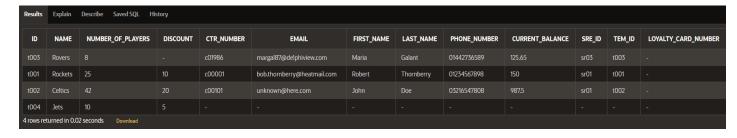
SELECT r.first_name||' '||r.last_name AS "Rep", s.first_name||' '||s.last_name AS "Supervisor" FROM sales_representatives r JOIN sales_representatives s ON (r.supervisor_id = s.id);



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);



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 Saved 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	07715246890				lc2341		vwright@obl.com	Victoria	Wright	0134598762		sr01
c01986	margal87@delphiview.com	Maria	Galant	01442736589	125.65	sr03	t003		sr02	vwright@obl.com	Victoria	Wright	0134598762		sr01
c00012	Jjones@freemail.com	Jennifer	Jones	01505214598				lc1015	sr02	vwright@obl.com	Victoria	Wright	0134598762		sr01
c00001	bob.thornberry@heatmail.com	Robert	Thornberry	01234567898	150	sr01	t001		sr02	vwright@obl.com	Victoria	Wright	0134598762		sr01
c00101	unknown@here.com	John	Doe	03216547808	987.5	sr01	t002		sr02	vwright@obl.com	Victoria	Wright	0134598762		sr01
c02001	brianrog@hootech.com	Brian	Rogers	01654564898	50			lc4587	sr02	vwright@obl.com	Victoria	Wright	0134598762		sr01
c00103	MurciaA@globaltech.com	Andrew	Murcia	07715246890				lc2341	sr03	bspeed@obl.com	Barry	Speed	0134598763		sr01
c01986	margal87@delphiview.com	Maria	Galant	01442736589	125.65	sr03	t003		sr03	bspeed@obl.com	Barry	Speed	0134598763		sr01
c00012	Jjones@freemail.com	Jennifer	Jones	01505214598						bspeed@obl.com	Barry	Speed	0134598763		sr01
c00001	bob.thornberry@heatmail.com	Robert	Thornberry	01234567898	150	sr01	t001		sr03	bspeed@obl.com	Barry	Speed	0134598763		sr01
More than 10 rows available. Increase rows selector to view more rows.															
10 rows returned in 0.02 seconds Developed															