



SECD2523 - DATABASE

SEMESTER 1/20232024

SECTION 08

**LAB 4 – DML PART 3**

**Lecturer: Dr. Noor Hidayah Binti Zakaria**

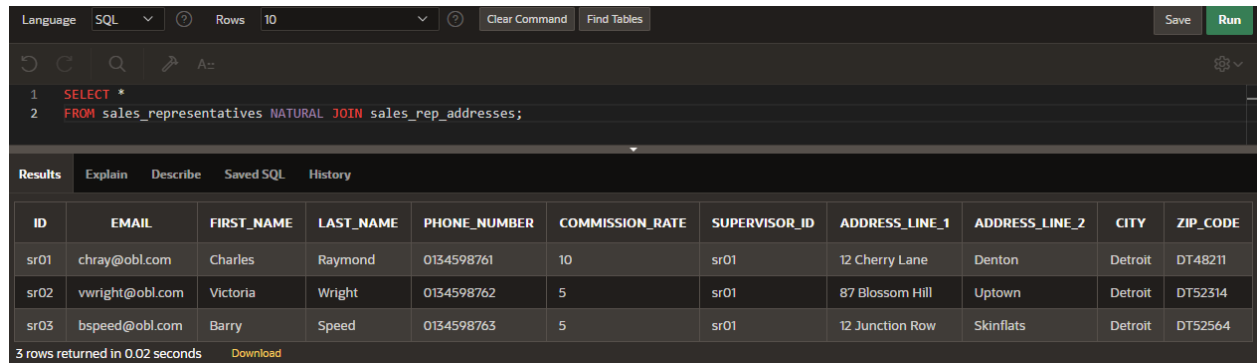
NAME	MATRIC NO.
JESSIE CHANG	A22EC0054

## 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 a SQL IDE interface. At the top, there's a toolbar with 'Language' set to 'SQL', 'Rows' set to '10', and buttons for 'Clear Command', 'Find Tables', 'Save', and 'Run'. Below the toolbar is a query editor with two lines of SQL code:

```
1 SELECT *
2 FROM sales_representatives NATURAL JOIN sales_rep_addresses;
```

Below the query editor is a 'Results' tab with sub-tabs 'Explain', 'Describe', 'Saved SQL', and 'History'. The 'Results' tab is active, showing a table with 11 columns: ID, EMAIL, FIRST\_NAME, LAST\_NAME, PHONE\_NUMBER, COMMISSION\_RATE, SUPERVISOR\_ID, ADDRESS\_LINE\_1, ADDRESS\_LINE\_2, CITY, and ZIP\_CODE. The table contains 3 rows of data:

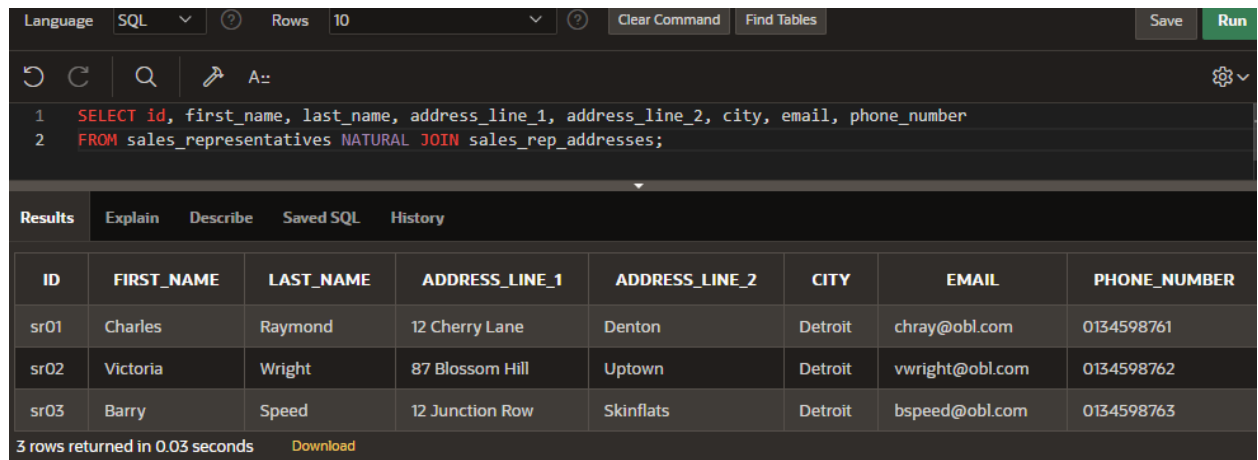
ID	EMAIL	FIRST_NAME	LAST_NAME	PHONE_NUMBER	COMMISSION_RATE	SUPERVISOR_ID	ADDRESS_LINE_1	ADDRESS_LINE_2	CITY	ZIP_CODE
sr01	chray@obl.com	Charles	Raymond	0134598761	10	sr01	12 Cherry Lane	Denton	Detroit	DT48211
sr02	vwright@obl.com	Victoria	Wright	0134598762	5	sr01	87 Blossom Hill	Uptown	Detroit	DT52314
sr03	bspeed@obl.com	Barry	Speed	0134598763	5	sr01	12 Junction Row	Skinflats	Detroit	DT52564

At the bottom of the results table, it says '3 rows returned in 0.02 seconds' and there is a 'Download' button.

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 a SQL IDE interface. At the top, there's a toolbar with 'Language' set to 'SQL', 'Rows' set to '10', and buttons for 'Clear Command', 'Find Tables', 'Save', and 'Run'. Below the toolbar is a query editor with two lines of 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;
```

Below the query editor is a 'Results' tab with sub-tabs 'Explain', 'Describe', 'Saved SQL', and 'History'. The 'Results' tab is active, showing 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:

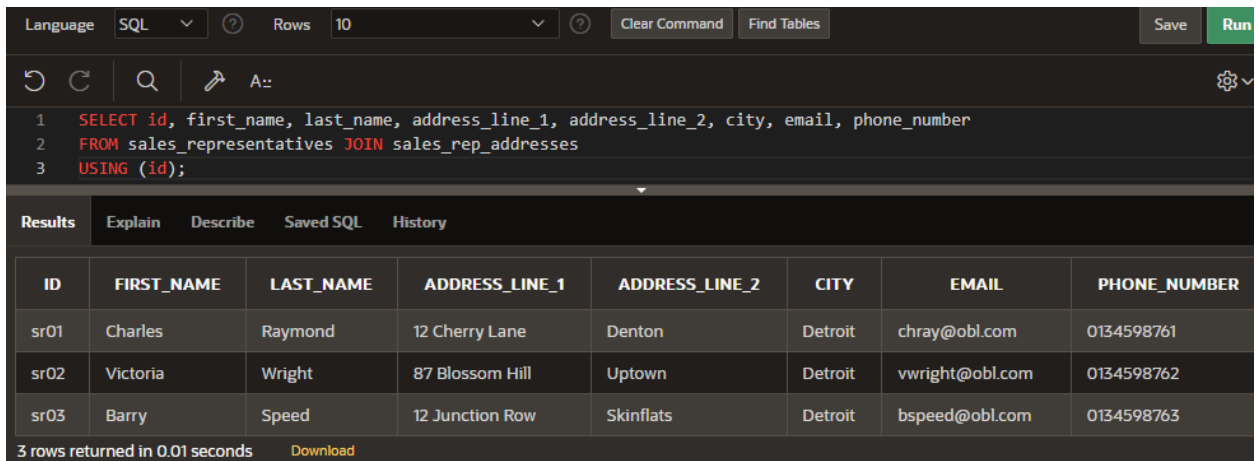
ID	FIRST_NAME	LAST_NAME	ADDRESS_LINE_1	ADDRESS_LINE_2	CITY	EMAIL	PHONE_NUMBER
sr01	Charles	Raymond	12 Cherry Lane	Denton	Detroit	chray@obl.com	0134598761
sr02	Victoria	Wright	87 Blossom Hill	Uptown	Detroit	vwright@obl.com	0134598762
sr03	Barry	Speed	12 Junction Row	Skinflats	Detroit	bspeed@obl.com	0134598763

At the bottom of the results table, it says '3 rows returned in 0.03 seconds' and there is a 'Download' button.

## 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);
```



Language: SQL Rows: 10 Clear Command Find Tables Save Run

```
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);
```

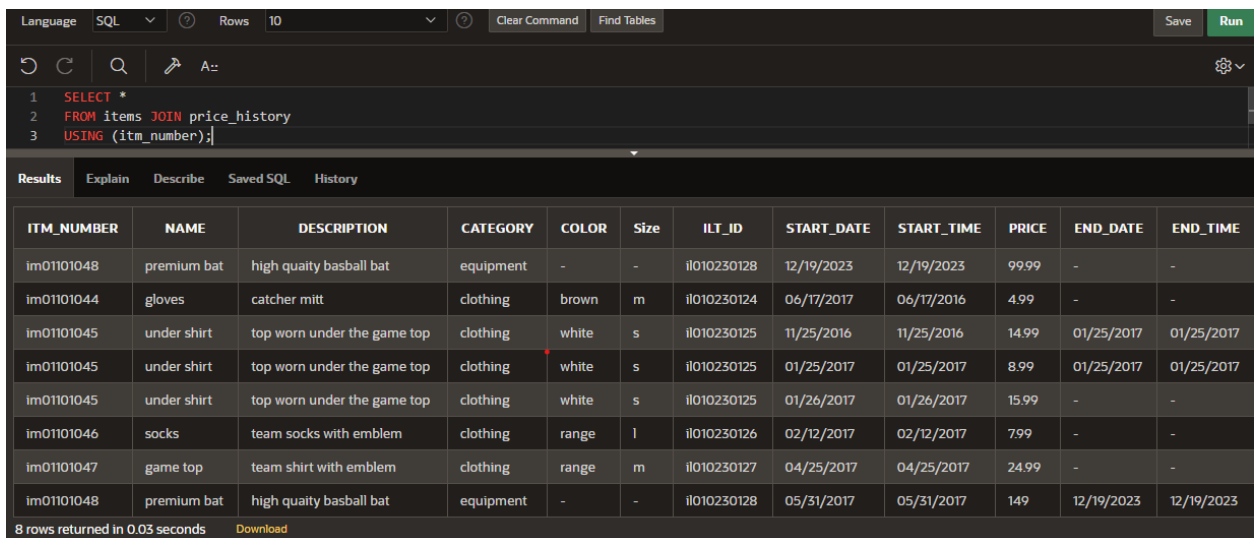
Results Explain Describe Saved SQL History

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 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);
```



Language: SQL Rows: 10 Clear Command Find Tables Save Run

```
1 SELECT *  
2 FROM items JOIN price_history  
3 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
im01i01048	premium bat	high quaity baseball bat	equipment	-	-	il010230128	12/19/2023	12/19/2023	99.99	-	-
im01i01044	gloves	catcher mitt	clothing	brown	m	il010230124	06/17/2017	06/17/2016	4.99	-	-
im01i01045	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
im01i01045	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
im01i01045	under shirt	top worn under the game top	clothing	white	s	il010230125	01/26/2017	01/26/2017	15.99	-	-
im01i01046	socks	team socks with emblem	clothing	range	l	il010230126	02/12/2017	02/12/2017	7.99	-	-
im01i01047	game top	team shirt with emblem	clothing	range	m	il010230127	04/25/2017	04/25/2017	24.99	-	-
im01i01048	premium bat	high quaity 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

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 "Sales Representative Email"  
FROM customers c JOIN sales_representatives s  
ON (c.sre_id = s.id);
```

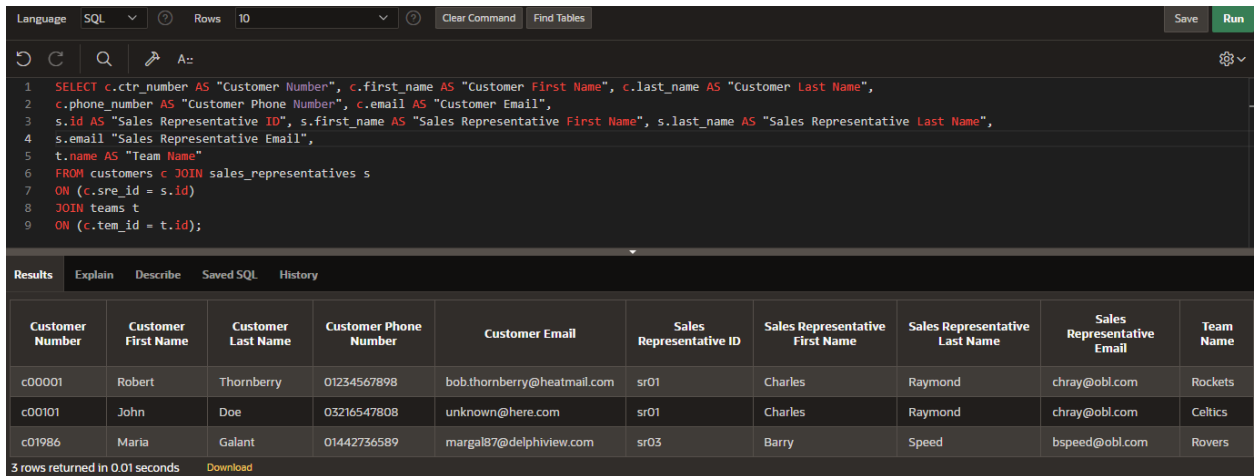
The screenshot shows a SQL IDE interface. At the top, there's a toolbar with 'Language' set to 'SQL', 'Rows' set to '10', and buttons for 'Clear Command', 'Find Tables', 'Save', and 'Run'. Below the toolbar is a search bar and a settings icon. The main area displays a SQL query with line numbers 1 through 6. The query is a SELECT statement joining 'customers' (aliased as 'c') and 'sales\_representatives' (aliased as 's') on the condition 'c.sre\_id = s.id'. The columns selected are: c.ctr\_number, c.first\_name, c.last\_name, c.phone\_number, c.email, s.id, s.first\_name, s.last\_name, and s.email. Below the query editor, there's a 'Results' tab with sub-tabs for 'Results', 'Explain', 'Describe', 'Saved SQL', and 'History'. The 'Results' tab is active, showing a table with 9 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, and Sales Representative Email. The table contains 3 rows of data. At the bottom left, it says '3 rows returned in 0.04 seconds' and there's a 'Download' button.

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

## 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 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 (c.sre_id = s.id)  
JOIN teams t  
ON (c.tem_id = t.id);
```



The screenshot shows a SQL IDE interface with the following components:

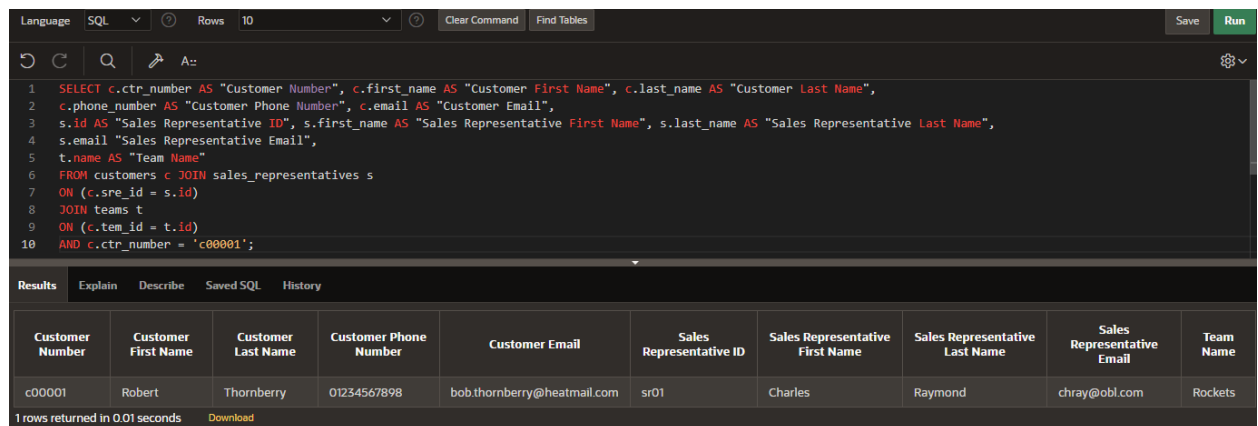
- Language:** SQL
- Rows:** 10
- Buttons:** Clear Command, Find Tables, Save, Run
- Query Editor:** Contains the SQL query for a three-way join.
- Results Tab:** Active, showing the query results in a table.
- Results Table:** Contains 10 columns and 3 rows of data.
- Status Bar:** Shows "3 rows returned in 0.01 seconds" and a "Download" button.

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

## 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 (c.sre_id = s.id)  
JOIN teams t  
ON (c.tem_id = t.id)  
AND c.ctr_number = 'c00001';
```



The screenshot shows a SQL IDE interface. At the top, there's a toolbar with 'Language' set to 'SQL', 'Rows' set to '10', and buttons for 'Clear Command', 'Find Tables', 'Save', and 'Run'. Below the toolbar is a query editor with the following SQL query:

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

Below the query editor is a 'Results' tab with sub-tabs for 'Results', 'Explain', 'Describe', 'Saved SQL', and 'History'. The 'Results' sub-tab is active, showing a table with 10 columns and 1 row of 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

At the bottom of the results table, it says '1 rows returned in 0.01 seconds' and there is a 'Download' button.

## 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 12 th of December 2016. The output of the query should look like this:

*The cost of the under shirt on this day was 14.99*

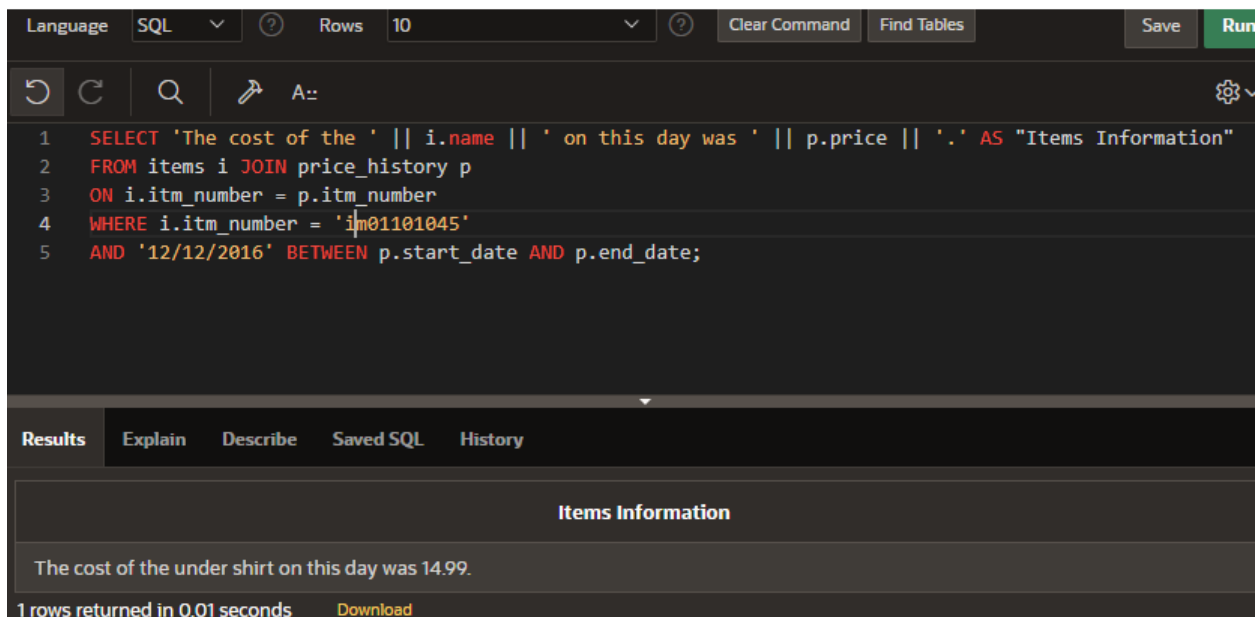
```
SELECT 'The cost of the ' || i.name || ' on this day was ' || p.price || '.' AS "Items  
Information"
```

```
FROM items i JOIN price_history p
```

```
ON i.itm_number = p.itm_number
```

```
WHERE i.itm_number = 'im01101045'
```

```
AND '12/12/2016' BETWEEN p.start_date AND p.end_date;
```



The screenshot shows a SQL IDE interface. At the top, there's a toolbar with 'Language' set to 'SQL', 'Rows' set to '10', and buttons for 'Clear Command', 'Find Tables', 'Save', and 'Run'. Below the toolbar is a command area with a SQL query. The query is as follows:

```
1 SELECT 'The cost of the ' || i.name || ' on this day was ' || p.price || '.' AS "Items Information"
2 FROM items i JOIN price_history p
3 ON i.itm_number = p.itm_number
4 WHERE i.itm_number = 'im01101045'
5 AND '12/12/2016' BETWEEN p.start_date AND p.end_date;
```

Below the command area is a 'Results' tab. The 'Results' tab is active, showing a table with one row. The table has a header 'Items Information' and a single row with the text 'The cost of the under shirt on this day was 14.99.'.

Items Information
The cost of the under shirt on this day was 14.99.

At the bottom of the results area, it says '1 rows returned in 0.01 seconds' and there is a 'Download' button.

## 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);
```

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

3 rows returned in 0.00 seconds [Download](#)



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

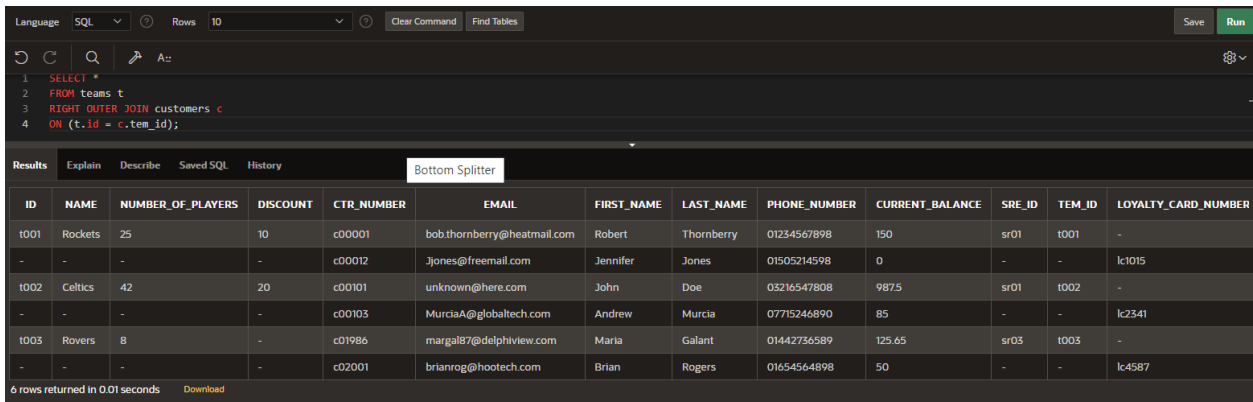
1. Write a query that will display all of the team and customer information even if there is no match with the table on the left (team).

**SELECT \***

**FROM teams t**

**RIGHT OUTER JOIN customers c**

**ON (t.id = c.tem\_id);**



The screenshot shows a SQL IDE interface with a query editor at the top and a results pane below. The query is a RIGHT OUTER JOIN between the 'teams' table (aliased as 't') and the 'customers' table (aliased as 'c') on the condition 't.id = c.tem\_id'. The results pane displays 6 rows of data, including team information (ID, NAME, NUMBER\_OF\_PLAYERS, DISCOUNT, CTR\_NUMBER, EMAIL) and customer information (FIRST\_NAME, LAST\_NAME, PHONE\_NUMBER, CURRENT\_BALANCE, SRE\_ID, TEM\_ID, LOYALTY\_CARD\_NUMBER). A 'Bottom Splitter' button is visible in the results pane header.

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	-
-	-	-	-	c00012	Jjones@freemail.com	Jennifer	Jones	01505214598	0	-	-	lc1015
t002	Celtics	42	20	c00101	unknown@here.com	John	Doe	03216547808	9875	sr01	t002	-
-	-	-	-	c00103	MurciaA@globaltech.com	Andrew	Murcia	07715246890	85	-	-	lc2341
t003	Rovers	8	-	c01986	margal87@delphview.com	Maria	Galant	01442736589	125.65	sr03	t003	-
-	-	-	-	c02001	brianrog@hootech.com	Brian	Rogers	01654564898	50	-	-	lc4587

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

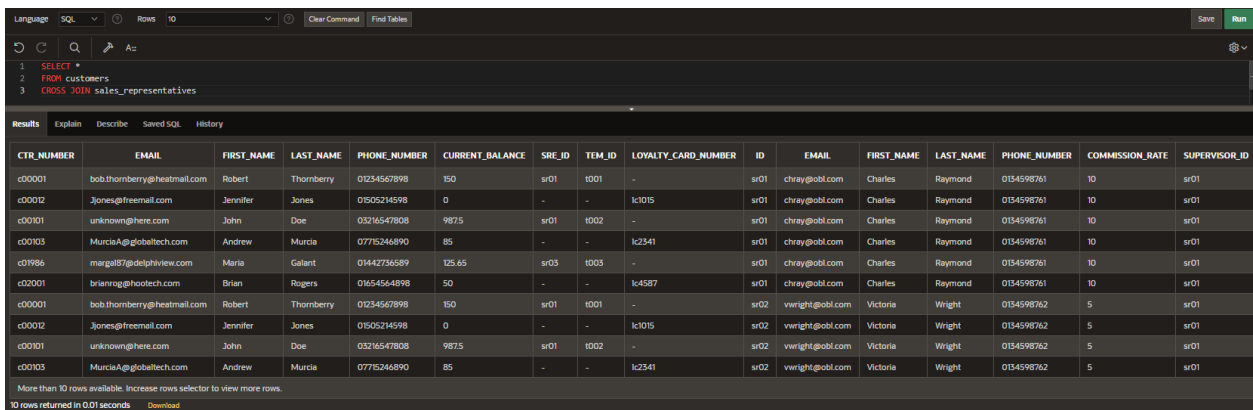
## Part 3 : Generating a Cartesian Product (S6L9 Objective 4)

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

**SELECT \***

**FROM customers**

**CROSS JOIN sales\_representatives**



The screenshot shows a SQL IDE interface with a query editor at the top and a results pane below. The query is a CROSS JOIN between the 'customers' table and the 'sales\_representatives' table. The results pane displays a Cartesian product of the two tables, showing all possible combinations of customer and sales representative records. The results pane header includes a 'Bottom Splitter' button. A message at the bottom indicates that more than 10 rows are available and that the rows selector can be increased to view more rows.

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
c00001	bob.thornberry@heatmail.com	Robert	Thornberry	01234567898	150	sr01	t001	-	sr01	chray@obl.com	Charles	Raymond	0154598761	10	sr01
c00012	Jjones@freemail.com	Jennifer	Jones	01505214598	0	-	-	lc1015	sr01	chray@obl.com	Charles	Raymond	0154598761	10	sr01
c00101	unknown@here.com	John	Doe	03216547808	9875	sr01	t002	-	sr01	chray@obl.com	Charles	Raymond	0154598761	10	sr01
c00103	MurciaA@globaltech.com	Andrew	Murcia	07715246890	85	-	-	lc2341	sr01	chray@obl.com	Charles	Raymond	0154598761	10	sr01
c01986	margal87@delphview.com	Maria	Galant	01442736589	125.65	sr03	t003	-	sr01	chray@obl.com	Charles	Raymond	0154598761	10	sr01
c02001	brianrog@hootech.com	Brian	Rogers	01654564898	50	-	-	lc4587	sr01	chray@obl.com	Charles	Raymond	0154598761	10	sr01
c00001	bob.thornberry@heatmail.com	Robert	Thornberry	01234567898	150	sr01	t001	-	sr02	vwright@obl.com	Victoria	Wright	0154598762	5	sr01
c00012	Jjones@freemail.com	Jennifer	Jones	01505214598	0	-	-	lc1015	sr02	vwright@obl.com	Victoria	Wright	0154598762	5	sr01
c00101	unknown@here.com	John	Doe	03216547808	9875	sr01	t002	-	sr02	vwright@obl.com	Victoria	Wright	0154598762	5	sr01
c00103	MurciaA@globaltech.com	Andrew	Murcia	07715246890	85	-	-	lc2341	sr02	vwright@obl.com	Victoria	Wright	0154598762	5	sr01

More than 10 rows available. Increase rows selector to view more rows.  
10 rows returned in 0.01 seconds [Download](#)