

Lab 4

Chan Wen Kang

A22EC0146

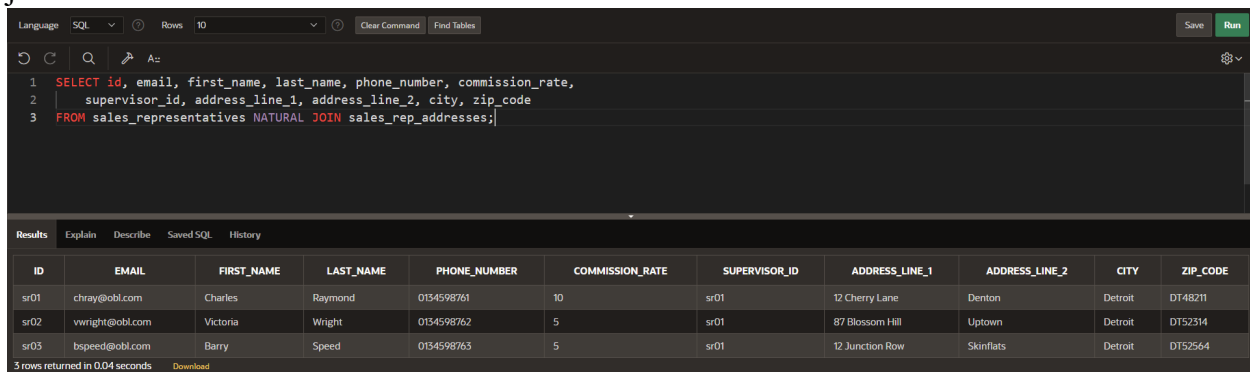
Section 6 Lesson 9 Exercise 1: Joining Tables Using JOIN

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.

Part 1: Creating Natural Joins.

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



The screenshot shows a SQL IDE interface. The top bar includes 'Language SQL', 'Rows 10', 'Clear Command', and 'Find Tables'. The main editor contains the following SQL query:

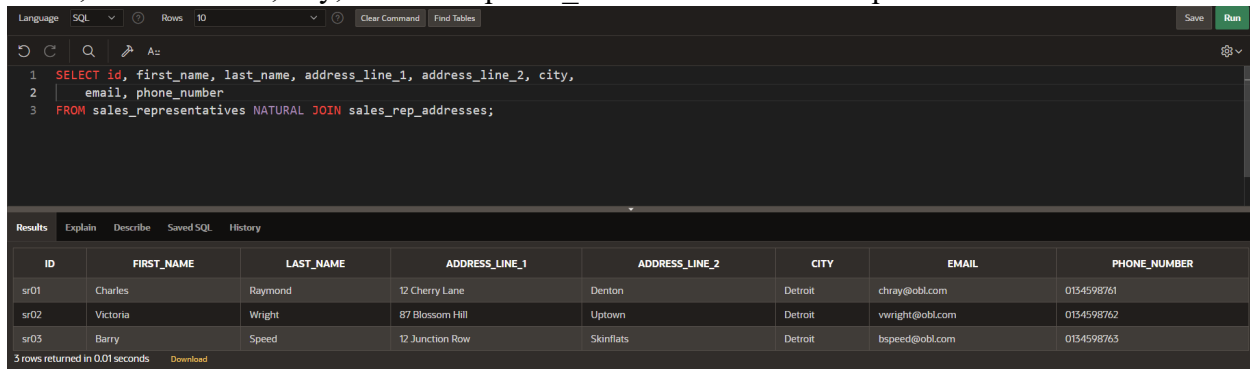
```
1 SELECT id, email, first_name, last_name, phone_number, commission_rate,  
2 supervisor_id, address_line_1, address_line_2, city, zip_code  
3 FROM sales_representatives NATURAL JOIN sales_rep_addresses;
```

The 'Results' tab is active, displaying 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

3 rows returned in 0.04 seconds

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 a SQL IDE interface. The top bar includes 'Language SQL', 'Rows 10', 'Clear Command', and 'Find Tables'. The main editor contains the following SQL query:

```
1 SELECT id, first_name, last_name, address_line_1, address_line_2, city,  
2 email, phone_number  
3 FROM sales_representatives NATURAL JOIN sales_rep_addresses;
```

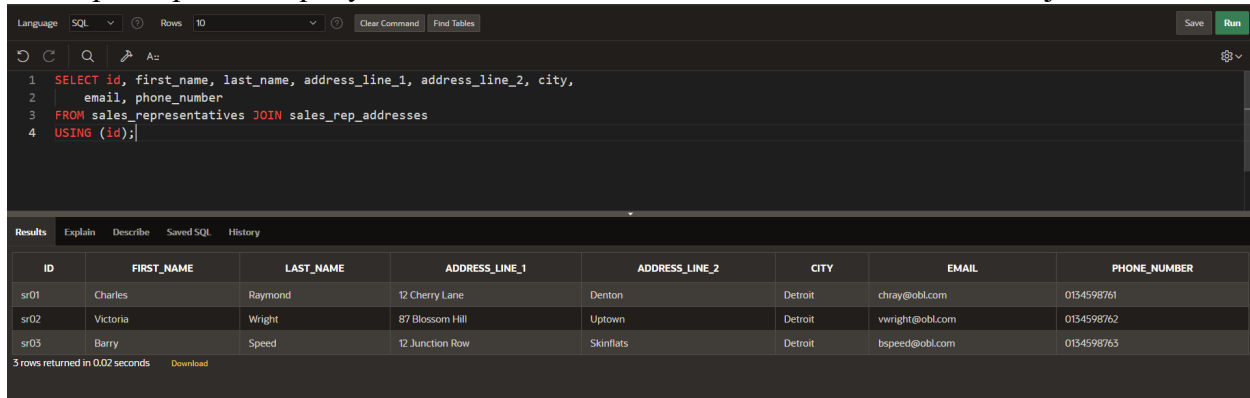
The 'Results' tab is active, displaying a table with 8 columns: ID, FIRST_NAME, LAST_NAME, ADDRESS_LINE_1, ADDRESS_LINE_2, CITY, EMAIL, and PHONE_NUMBER. The table contains 3 rows of data.

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.

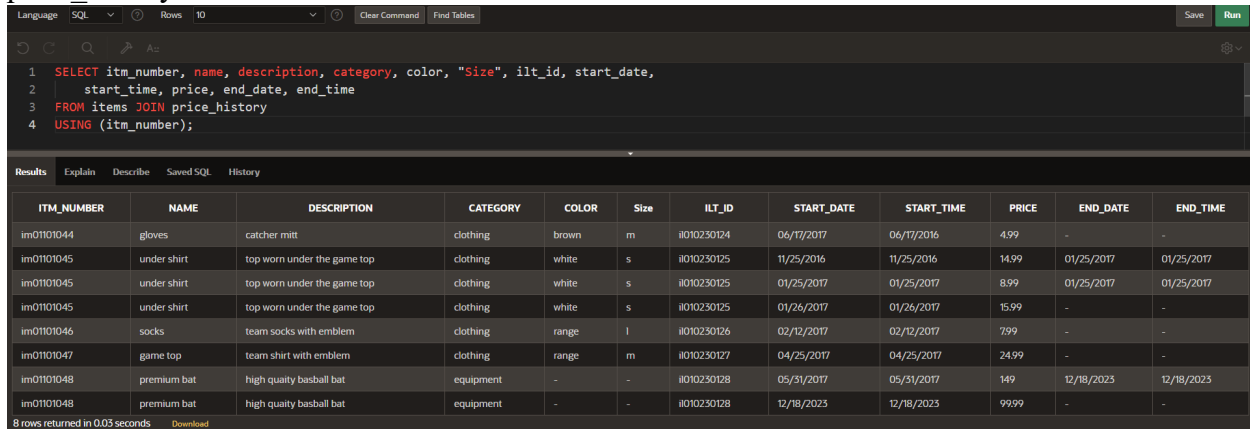


```
1 SELECT id, first_name, last_name, address_line_1, address_line_2, city,
2        email, phone_number
3 FROM sales_representatives JOIN sales_rep_addresses
4 USING (id);
```

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 0.02 seconds [Download](#)

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



```
1 SELECT itm_number, name, description, category, color, "Size", ilt_id, start_date,
2        start_time, price, end_date, end_time
3 FROM items JOIN price_history
4 USING (itm_number);
```

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	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	12/18/2023	12/18/2023
im0101048	premium bat	high quality baseball bat	equipment	-	-	il010230128	12/18/2023	12/18/2023	99.99	-	-

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.

Code:

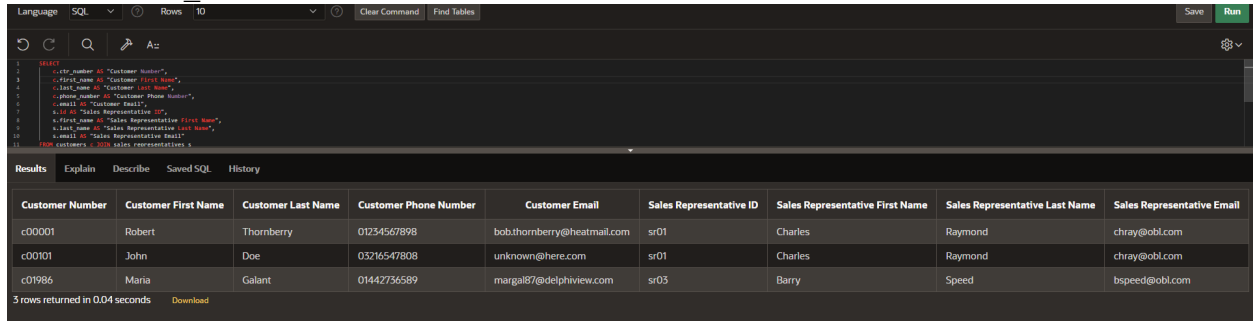
SELECT

```
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"

FROM customers c JOIN sales_representatives s

ON s.id=c.sre_id;



The screenshot shows a SQL IDE interface. The top bar includes a language dropdown set to 'SQL', a 'Rows' indicator showing '10', and buttons for 'Clear Command' and 'Find Tables'. Below the bar is a query editor with a SQL query. The 'Results' tab is active, displaying 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 of the results section, it states '3 rows returned in 0.04 seconds' and provides a 'Download' link.

```
1 SELECT
2   c.sre_id AS "Sales Representative ID",
3   c.first_name AS "Sales Representative First Name",
4   c.last_name AS "Sales Representative Last Name",
5   c.email AS "Sales Representative Email",
6   c.id AS "Customer ID",
7   c.first_name AS "Customer First Name",
8   c.last_name AS "Customer Last Name",
9   c.phone_number AS "Customer Phone Number",
10  c.email AS "Customer Email"
11 FROM customers c JOIN sales_representatives s
```

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 0.04 seconds [Download](#)

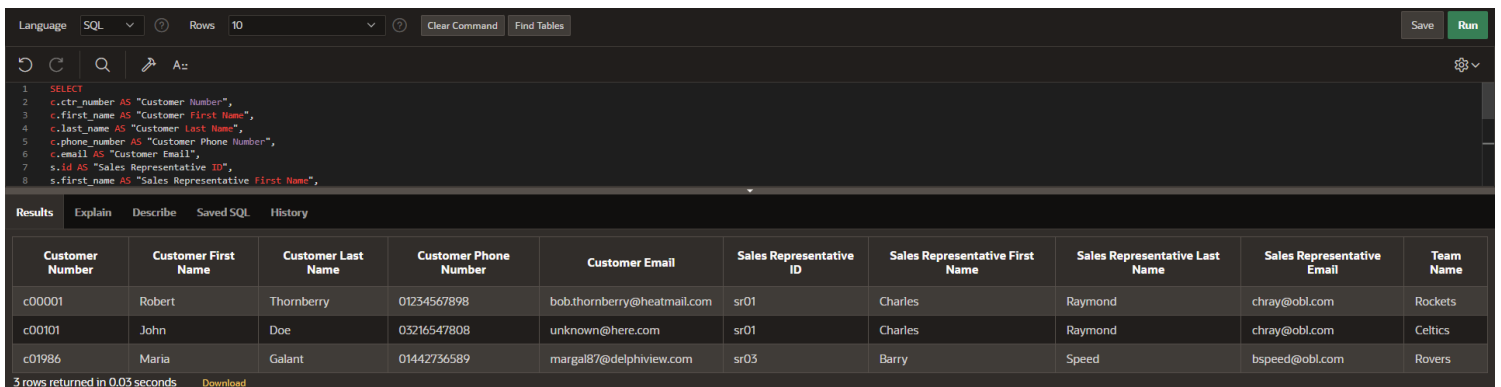
Part 4- Creating Three-Way Joins with the ON Clause

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.

Code:

```
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. The top bar includes a language dropdown set to 'SQL', a 'Rows' dropdown set to '10', and buttons for 'Clear Command' and 'Find Tables'. On the right are 'Save' and 'Run' buttons. The main editor area contains the SQL query from the previous block. Below the editor is a 'Results' tab, which is active, showing a table with 10 columns and 3 rows of data. The columns are: 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, and Team Name. The data rows are: (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), and (c01986, Maria, Galant, 01442736589, margal87@delphiview.com, sr03, Barry, Speed, bspeed@obl.com, Rovers). At the bottom left, it says '3 rows returned in 0.05 seconds' and there is 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

3 rows returned in 0.05 seconds Download

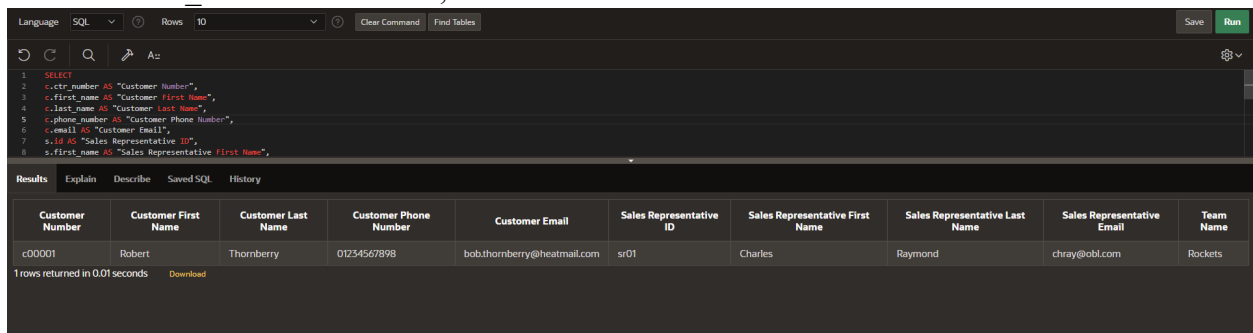
Part 5: Applying Additional Conditions to a Join

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

Code:

```
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  
WHERE c.ctr_number = 'c00001';
```



The screenshot shows a SQL IDE interface. The top bar includes a language dropdown set to 'SQL', a 'Rows' count of 10, and buttons for 'Clear Command' and 'Find Tables'. Below the bar is a command editor with the SQL query. The bottom section displays the query results in a table format. The table has 10 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, Sales Representative Email, and Team Name. A single row of data is shown, corresponding to customer c00001. Below the table, it indicates '1 rows returned in 0.01 seconds' and provides a 'Download' link.

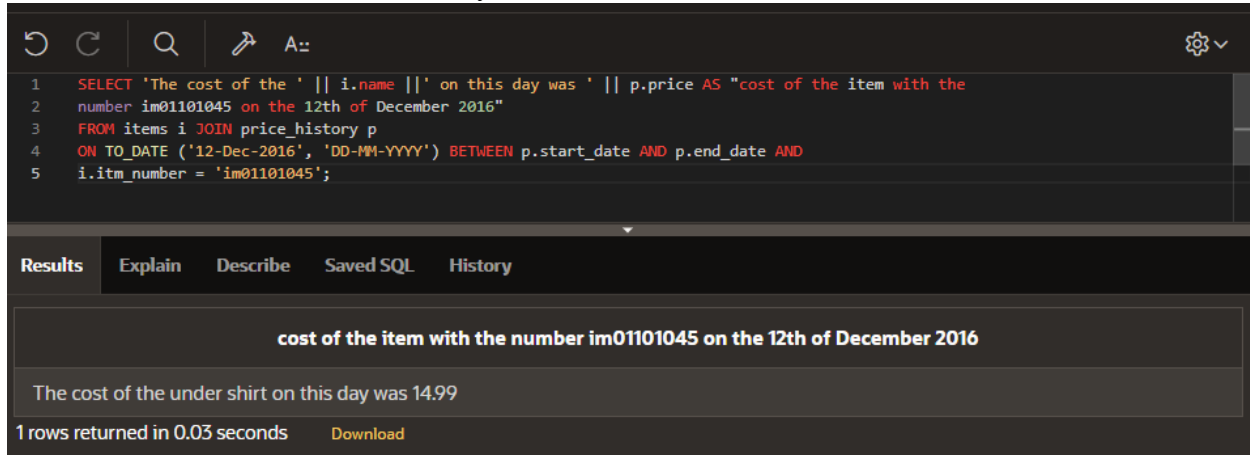
Customer Number	Customer First Name	Customer Last Name	Customer Phone Number	Customer Email	Sales Representative ID	Sales Representative First Name	Sales Representative Last Name	Sales Representative Email	Team Name
c00001	Robert	Thornberry	01234567898	bob.thornberry@heatmail.com	sr01	Charles	Raymond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



The screenshot shows a SQL query editor with a dark theme. The query is as follows:

```
1 SELECT 'The cost of the ' || i.name || ' on this day was ' || p.price AS "cost of the item with the
2 number im01101045 on the 12th of December 2016"
3 FROM items i JOIN price_history p
4 ON TO_DATE ('12-Dec-2016', 'DD-MM-YYYY') BETWEEN p.start_date AND p.end_date AND
5 i.item_number = 'im01101045';
```

Below the query editor, there are tabs for 'Results', 'Explain', 'Describe', 'Saved SQL', and 'History'. The 'Results' tab is selected, showing a single row of data:

cost of the item with the number im01101045 on the 12th of December 2016
The cost of the under shirt on this day was 14.99

At the bottom, it indicates '1 rows returned in 0.03 seconds' and provides a 'Download' link.

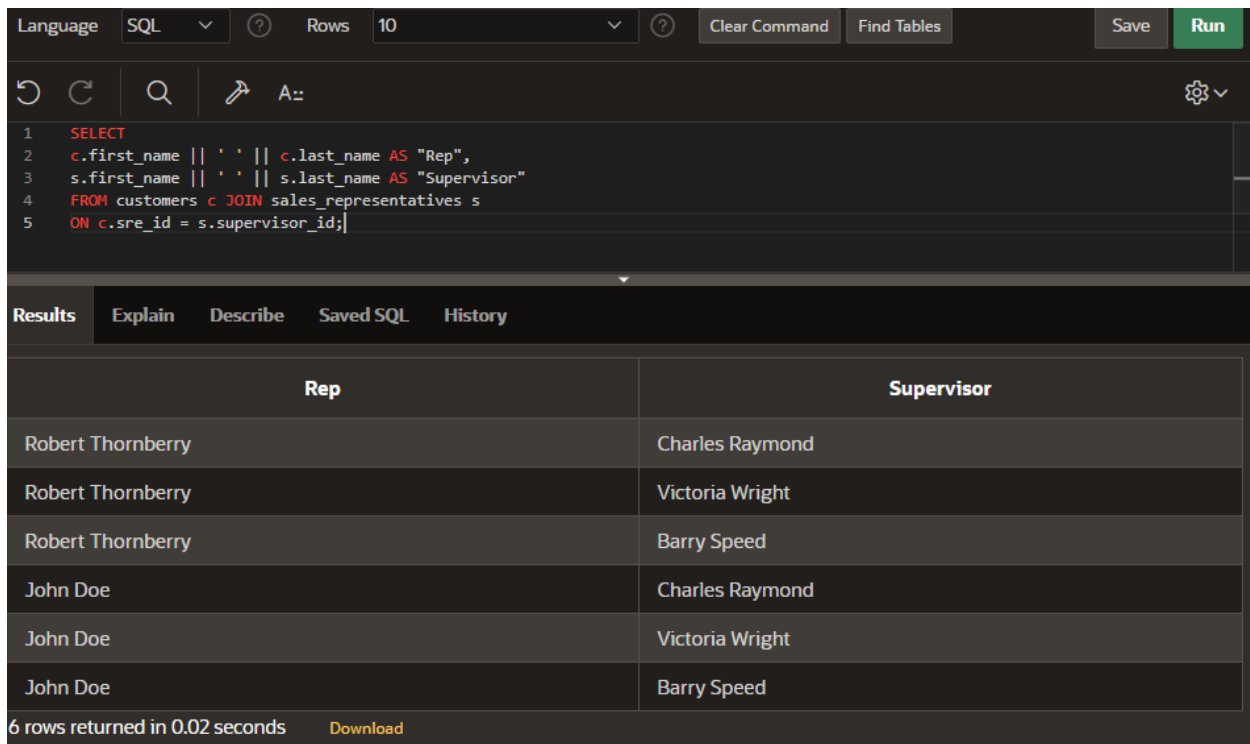
Section 6 Lesson 9 Exercise 2: Joining Tables Using JOIN

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.

Code:



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 with a magnifying glass icon and a dropdown menu. The main area contains a SQL query:

```
1 SELECT
2 c.first_name || ' ' || c.last_name AS "Rep",
3 s.first_name || ' ' || s.last_name AS "Supervisor"
4 FROM customers c JOIN sales_representatives s
5 ON c.sre_id = s.supervisor_id;
```

Below the query editor, there's a tabbed interface with 'Results' selected. The results are displayed in a table with two columns: 'Rep' and 'Supervisor'.

Rep	Supervisor
Robert Thornberry	Charles Raymond
Robert Thornberry	Victoria Wright
Robert Thornberry	Barry Speed
John Doe	Charles Raymond
John Doe	Victoria Wright
John Doe	Barry Speed

At the bottom, it says '6 rows returned in 0.02 seconds' and there's a 'Download' button.

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).

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

↺ ↻ 🔍 🔗 A:: ⚙️

```
1 SELECT t.id, t.name, t.number_of_players, t.discount, c.ctr_number, c.email, c.first_name,
2 c.last_name, c.phone_number, c.current_balance, c.sre_id, c.tem_id, c.loyalty_card_number
3 FROM teams t LEFT OUTER
4 JOIN customers c
5 ON (t.id = c.tem_id);
```

Results Explain Describe Saved SQL History

Rep	Supervisor
Robert Thornberry	Charles Raymond
Robert Thornberry	Victoria Wright
Robert Thornberry	Barry Speed
John Doe	Charles Raymond
John Doe	Victoria Wright
John Doe	Barry Speed

6 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.

Code:

```
SELECT c.ctr_number, c.email, c.first_name, c.last_name, c.phone_number, c.current_balance,
c.sre_id, c.tem_id, c.loyalty_card_number, s.id, s.email, s.first_name, s.last_name,
s.phone_number, s.commission_rate, s.supervisor_id
FROM customers c, sales_representatives s;
```

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
cd0001	bob.thornberry@hextel.com	Robert	Thornberry	0123456789	150	s001	1001	-	s001	chray@hdb.com	Charles	Raymond	0154567891	10	s001
cd0002	jane@hextel.com	Jennifer	Jones	01987654321	0	-	-	k1015	s001	chray@hdb.com	Charles	Raymond	0154567891	10	s001
cd0003	john@hextel.com	John	Doe	01234567890	9875	s001	1002	-	s001	chray@hdb.com	Charles	Raymond	0154567891	10	s001
cd0004	maria@hextel.com	Andrew	Marca	0776543210	85	-	-	k2341	s001	chray@hdb.com	Charles	Raymond	0154567891	10	s001
cd0005	marg@hextel.com	Maria	Galea	0143210987	32.55	s002	1003	-	s001	chray@hdb.com	Charles	Raymond	0154567891	10	s001
cd0006	bruce@hextel.com	Brian	Rogers	0154321098	50	-	-	k4567	s001	chray@hdb.com	Charles	Raymond	0154567891	10	s001
cd0007	bob.thornberry@hextel.com	Robert	Thornberry	0123456789	150	s001	1001	-	s002	vwright@hdb.com	Victoria	Wright	0154567892	5	s001
cd0002	jane@hextel.com	Jennifer	Jones	01987654321	0	-	-	k1015	s002	vwright@hdb.com	Victoria	Wright	0154567892	5	s001
cd0003	john@hextel.com	John	Doe	01234567890	9875	s001	1002	-	s002	vwright@hdb.com	Victoria	Wright	0154567892	5	s001
cd0003	maria@hextel.com	Andrew	Marca	0776543210	85	-	-	k2341	s002	vwright@hdb.com	Victoria	Wright	0154567892	5	s001

More than 10 rows available. Increase rows selector to view more rows.
10 rows returned in 0.07 seconds Download