



**UTM**  
UNIVERSITI TEKNOLOGI MALAYSIA

**FACULTY OF COMPUTING**  
UTM Johor Bahru

## LAB 4 DML 3

|             |                                  |
|-------------|----------------------------------|
| NAME        | ADAM ISMAIL HASSAN AMER ABOURAYA |
| MATRIC NO.  | A22EC0002                        |
| Course Code | SECD2523                         |
| Section     | 08                               |

# 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 an SQL Worksheet interface with a sidebar on the left containing links to Home, SQL Worksheet, My Session, Schema, Quick SQL, My Scripts, My Tutorials, and Code Library. The main area displays a SQL query:

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

Below the query, the results are displayed in 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 | DT48711  |
| 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  |

Below the table, there is a "Download CSV" button and the text "3 rows selected." At the bottom of the interface, a footer line reads: "2023 Oracle - Live SQL 23.4.2, running Oracle Database 19c EE Extreme Perf - 19.17.0.0.0 - Database Documentation - Ask Tom - Dev Gym" and "Built with ❤️ using Oracle APEX - Privacy - Terms of Use".

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 Worksheet interface with a query using a NATURAL JOIN. The query is as follows:

```
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;
```

The results table displays the following 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   |

Below the table, it indicates "3 rows selected." and provides a "Download CSV" 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.

The screenshot shows a SQL Worksheet interface with a query using the USING clause. The query is as follows:

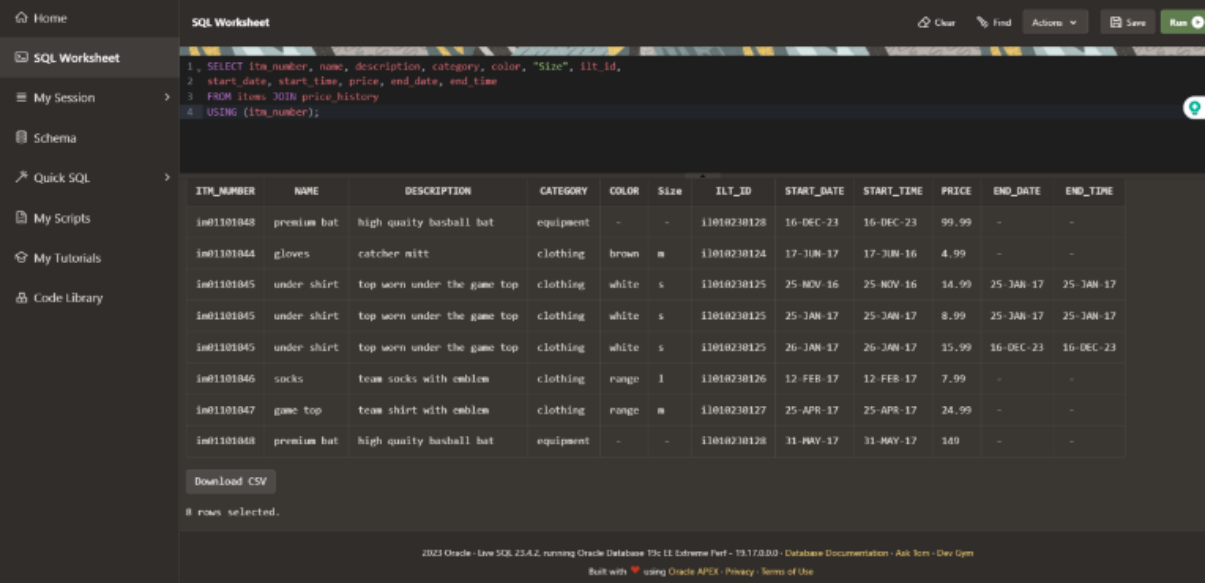
```
1. SELECT id, first_name, last_name, address_line_1, address_line_2, city, email, phone_number
2. FROM sales_representatives JOIN sales_rep_addresses
3. USING (id);
```

The results table displays the same data as the previous screenshot:

| 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   |

Below the table, it indicates "3 rows selected." and provides a "Download CSV" button.

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



The screenshot shows an SQL Worksheet interface with a sidebar on the left containing links to Home, SQL Worksheet, My Session, Schema, Quick SQL, My Scripts, My Tutorials, and Code Library. The main area displays a SQL query and its results.

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

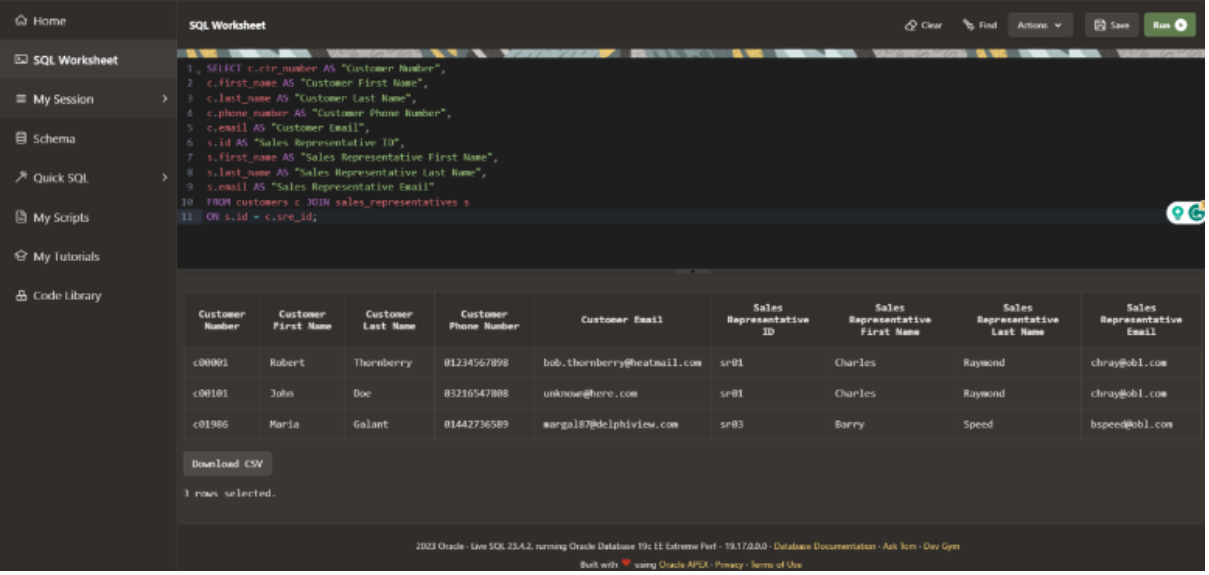
| ITM_NUMBER | NAME        | DESCRIPTION                 | CATEGORY  | COLOR  | Size | ITM_ID      | START_DATE | START_TIME | PRICE | END_DATE  | END_TIME  |
|------------|-------------|-----------------------------|-----------|--------|------|-------------|------------|------------|-------|-----------|-----------|
| im01101048 | premium bat | high quality baseball bat   | equipment | -      | -    | 11010230128 | 16-DEC-23  | 16-DEC-23  | 99.99 | -         | -         |
| im01101044 | gloves      | catcher mitt                | clothing  | brown  | m    | 11010230124 | 17-JUN-17  | 17-JUN-16  | 4.99  | -         | -         |
| im01101045 | under shirt | top worn under the game top | clothing  | white  | s    | 11010230125 | 25-NOV-16  | 25-NOV-16  | 14.99 | 25-JAN-17 | 25-JAN-17 |
| im01101045 | under shirt | top worn under the game top | clothing  | white  | s    | 11010230125 | 25-JAN-17  | 25-JAN-17  | 8.99  | 25-JAN-17 | 25-JAN-17 |
| im01101045 | under shirt | top worn under the game top | clothing  | white  | s    | 11010230125 | 26-JAN-17  | 26-JAN-17  | 15.99 | 16-DEC-23 | 16-DEC-23 |
| im01101046 | socks       | team socks with emblem      | clothing  | orange | l    | 11010230126 | 12-FEB-17  | 12-FEB-17  | 7.99  | -         | -         |
| im01101047 | game top    | team shirt with emblem      | clothing  | orange | m    | 11010230127 | 25-APR-17  | 25-APR-17  | 24.99 | -         | -         |
| im01101048 | premium bat | high quality baseball bat   | equipment | -      | -    | 11010230128 | 31-MAY-17  | 31-MAY-17  | 149   | -         | -         |

Download CSV  
8 rows selected.

2023 Oracle - Live SQL 23.4.2, running Oracle Database 19c EE Extreme Perf - 19.17.0.0.0 - Database Documentation - Ask Tom - Dev Gps  
Built with ♥ using Oracle APEX - Privacy - Terms of Use

## 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.



The screenshot shows an SQL Worksheet interface with a sidebar on the left containing links to Home, SQL Worksheet, My Session, Schema, Quick SQL, My Scripts, My Tutorials, and Code Library. The main area displays a SQL query and its results.

```
1. SELECT c.cust_number AS "Customer Number",  
2. c.first_name AS "Customer First Name",  
3. c.last_name AS "Customer Last Name",  
4. c.phone_number AS "Customer Phone Number",  
5. c.email AS "Customer Email",  
6. s.id AS "Sales Representative ID",  
7. s.first_name AS "Sales Representative First Name",  
8. s.last_name AS "Sales Representative Last Name",  
9. s.email AS "Sales Representative Email"  
10. FROM customers c JOIN sales_representatives s  
11. ON s.id = c.sre_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 |
|-----------------|---------------------|--------------------|-----------------------|-----------------------------|-------------------------|---------------------------------|--------------------------------|----------------------------|
| c00001          | Robert              | Thornberry         | 01234567898           | bob.thornberry@heatmail.com | sr01                    | Charles                         | Raymond                        | chray@obl.com              |
| c00101          | John                | Doe                | 03216547808           | unknown@hore.com            | sr01                    | Charles                         | Raymond                        | chray@obl.com              |
| c01986          | Maria               | Galant             | 01442736589           | margal87@delphiview.com     | sr03                    | Barry                           | Speed                          | bspeed@obl.com             |

Download CSV  
3 rows selected.

2023 Oracle - Live SQL 23.4.2, running Oracle Database 19c EE Extreme Perf - 19.17.0.0.0 - Database Documentation - Ask Tom - Dev Gps  
Built with ♥ using Oracle APEX - Privacy - Terms of Use

## 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

The screenshot shows a SQL Worksheet interface with a query editor on the left and a results table on the right. The query is a three-way join between the customers, sales\_representatives, and teams tables. The results table displays the following 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@outmail.com | sr01                    | Charles                         | Raymond                        | chr01@bbl.com              | Rockets   |
| c00101          | John                | Doe                | 03216547888           | unknown@here.com           | sr01                    | Charles                         | Raymond                        | chr01@bbl.com              | Celtics   |
| c01986          | Maria               | Galant             | 01442736589           | margal87@delphiview.com    | sr03                    | Barry                           | Speed                          | bspeed@bbl.com             | Rovers    |

Download CSV  
3 rows selected.

2023 Oracle - Use SQL 23.4.2, running Oracle Database 19c EE Extreme Perf - 19.17.0.0.0 - Database Documentation - Ask Tom - Dev Gyn  
Built with using Oracle APEX - Privacy - Terms of Use

## 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

The screenshot shows the same SQL Worksheet interface, but the query has been updated to include a WHERE clause to filter results for the customer with number c00001. The results table now displays only one row:

| 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@outmail.com | sr01                    | Charles                         | Raymond                        | chr01@bbl.com              | Rockets   |

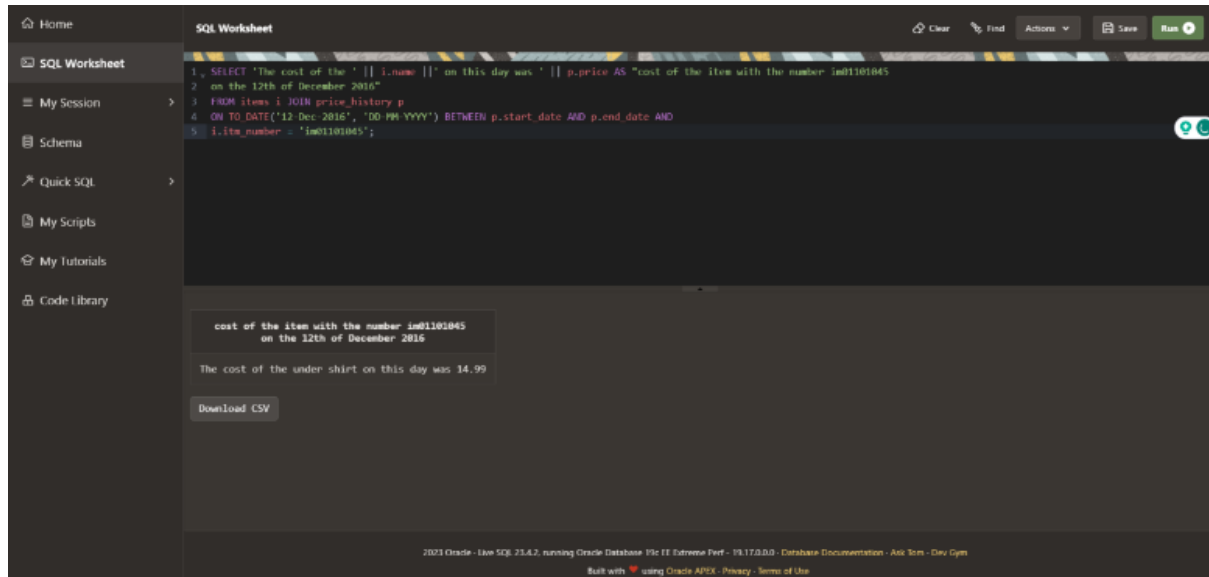
Download CSV  
3 rows selected.

2023 Oracle - Use SQL 23.4.2, running Oracle Database 19c EE Extreme Perf - 19.17.0.0.0 - Database Documentation - Ask Tom - Dev Gyn  
Built with using Oracle APEX - Privacy - Terms of Use

## 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

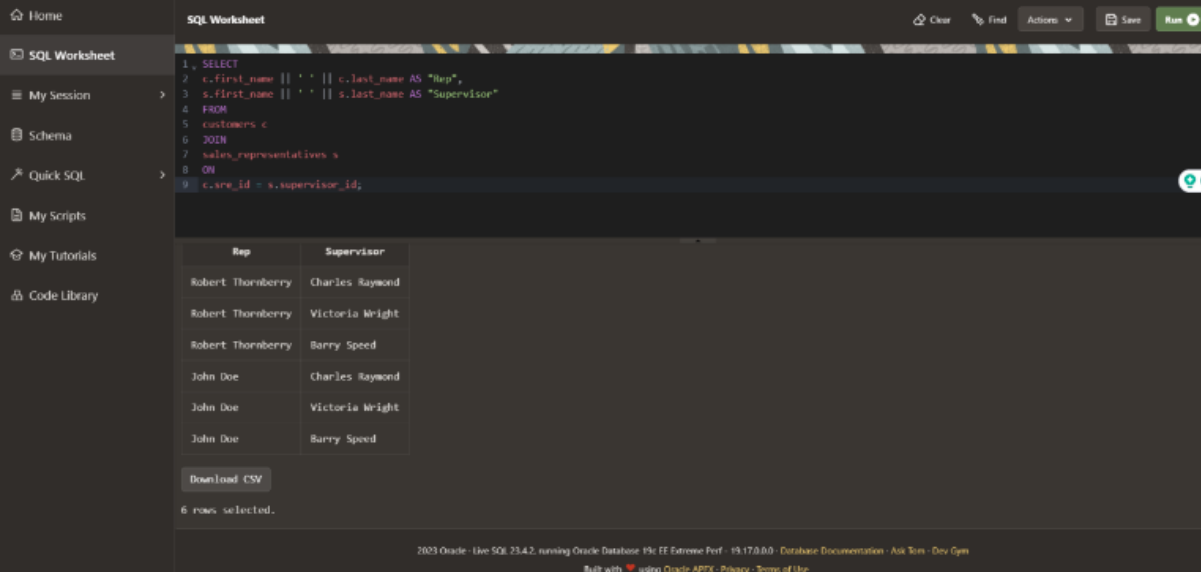


## Section 6 Lesson 9 Exercise 2: Joining Tables Using JOIN

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.



The screenshot shows an SQL Worksheet interface with a query editor on the left and a results table on the right. The query is a self-join on the EMPLOYEES table, filtering for sales representatives and their supervisors.

```
1. SELECT
2.   c.first_name || ' ' || c.last_name AS "Rep",
3.   s.first_name || ' ' || s.last_name AS "Supervisor"
4. FROM
5.   employees c
6 JOIN
7.   sales_representatives s
8. ON
9.   c.emp_id = s.supervisor_id;
```

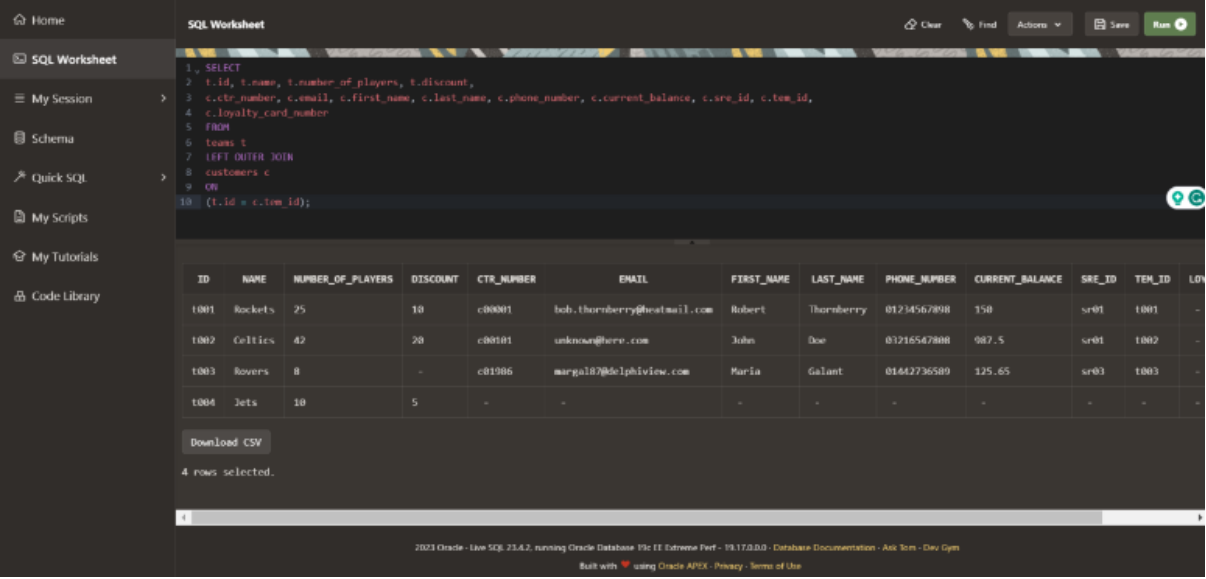
| Rep               | Supervisor      |
|-------------------|-----------------|
| Robert Thornberry | Charles Raymond |
| Robert Thornberry | Victoria Wright |
| Robert Thornberry | Barry Speed     |
| John Doe          | Charles Raymond |
| John Doe          | Victoria Wright |
| John Doe          |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

Download CSV  
6 rows selected.

2023 Oracle - Use SQL 23.4.2, running Oracle Database 19c EE Extreme Perf - 19.17.0.0.0 - Database Documentation - Ask Tom - Dev Gyn  
Built with ♥ using Oracle APEX - Privacy - Terms of Use

## 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).



The screenshot shows an SQL Worksheet interface with a query editor on the left and a results table on the right. The query is a LEFT OUTER JOIN between the TEAMS and CUSTOMERS tables, displaying all team information and matching customer information.

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

| 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 | Cellies | 42                | 20       | c00101     | unknown@here.com            | John       | Doe        | 01216567808  | 987.5           | sr01   | 1002   | -                   |
| 1003 | Rovers  | 8                 | -        | c01986     | marga187@delphiview.com     | Maria      | Galant     | 01442736589  | 125.65          | sr03   | 1003   | -                   |
| 1004 | Jets    | 10                | 5        | -          | -                           | -          | -          | -            | -               | -      | -      | -                   |

Download CSV  
4 rows selected.

2023 Oracle - Use SQL 23.4.2, running Oracle Database 19c EE Extreme Perf - 19.17.0.0.0 - Database Documentation - Ask Tom - Dev Gyn  
Built with ♥ using Oracle APEX - Privacy - Terms of Use

