**Oracle SET OPERATORS syntax**

Here you will see many Examples of using SET OPERATORS with Oracle server, and you may try these primers on your own and play with them to grasp this material.

I) Firstly, we will create two sets of rows that we will use for our SET examples.

1. We will use full SALES REP names from Employees table
2. We will use full CONTACT names for all customers with credit limit lower than $500

SELECT SUBSTR(last\_name || ', ' || first\_name,1,20) "Sales Rep Name"

FROM employees

WHERE lower(job\_title) LIKE 'sales%'

ORDER BY 1;

**Sales Rep Name**

**--------------------**

Boyd, Imogen

Cole, Isabella

Crawford, Lucy

**Cruz, Chloe**

Dixon, Elizabeth

Ellis, Grace

Fisher, Lily

Freeman, Florence

Gibson, Scarlett

Gomez, Freya

Harrison, Evie

Henry, Elsie

Hicks, Sofia

Hunter, Millie

Jordan, Poppy

Kennedy, Maya

Marshall, Isabelle

Mason, Layla

Mcdonald, Ruby

Morales, Rosie

Murray, Phoebe

Ortiz, Daisy

Owens, Sophie

Porter, Eva

Reynolds, Sophia

Simpson, Sienna

Stevens, Matilda

Sullivan, Ava

Tucker, Evelyn

Wallace, Ella

Warren, Esme

Webb, Charlotte

Wells, Alice

West, Mia

Woods, Jessica

35 rows selected.

SELECT SUBSTR(last\_name || ', ' || first\_name,1,20) "Contact Name",

SUBSTR(name,1,20) "Customer Name", credit\_limit "Limit"

FROM customers JOIN contacts USING (customer\_id)

WHERE credit\_limit < 500

ORDER BY 1 ;

**Contact Name Customer Name Limit**

**-------------------- -------------------- ----------**

Arnold, Rena Assurant 300

Atkinson, Jaclyn Global Partners 400

Barrera, Willie Starbucks 400

Brady, Amber Cameron Internationa 200

Buchanan, Brandie DaVita HealthCare Pa 300

Buckley, Trula Hartford Financial S 400

Cage, Christian Emerson Electric 100

Calhoun, Cora Kimberly-Clark 400

Callahan, Nathan Farmers Insurance Ex 100

Cochran, Mireya Paccar 400

Costner, Daniel Staples 200

Cruise, Matthias Alcoa 100

**Cruz, Napoleon Cognizant Technology 400**

Daltrey, Maurice Eli Lilly 200

Derek, Dianne Danaher 200

Edwards, Guillaume AutoNation 200

Emerson, Lavera Plains GP Holdings 100

Gamble, Peter MGM Resorts Internat 100

Glass, Daniel ManpowerGroup 300

Hannah, Matthias Community Health Sys 100

Hasan, Maurice Abbott Laboratories 200

Head, Fern US Foods Holding 100

Higgins, Diane Dollar General 200

Holden, Kandi Molina Healthcare 300

Ingram, Arlyne Kohl?s 300

Jackson, Guillaume U.S. Bancorp 200

Kaufman, Ka Southwest Airlines 200

Levy, Jeni Centene 100

Mahoney, Maurice Progressive 200

Martin, Geraldine Aflac 200

Mason, Meenakshi International Paper 100

Massey, Wendell BB&T Corp. 200

Mcdonald, Tisha Autoliv 400

Montoya, Sharyl Penske Automotive Gr 200

Odom, Marlene Cummins 400

Ortiz, Shyla AbbVie 100

Pacino, Charlie Amgen 200

Padilla, Tereasa J.C. Penney 400

Phillips, Merrilee Consolidated Edison 400

Riley, Felicitas Xerox 400

Roberson, Maryrose Xcel Energy 400

Roth, Tess DTE Energy 200

Schneider, Geraldine Whirlpool 200

Schultz, Al Altria Group 400

Sen, Dianne Tenet Healthcare 200

Stone, Flor Raytheon 100

Sutherland, Charlie Union Pacific 200

Velazquez, Misti Aramark 300

48 rows selected.

Here I will change first name of Sales Rep Cruz from Chloe to Napoleon, and that will give us situation where Full Name of one Sales Reps is the same as of one Customer Contacts.

SQL> **UPDATE employees SET first\_name = 'Napoleon'**

**WHERE last\_name = 'Cruz';**

1 row updated.

1. Find out do we have Sales Reps with the same Full Name as Customer Contacts 🡪 use INTERSECT.

Notice that Aliases count only for the First Select statement, while ORDER BY must exist only in the Last Select statement

SELECT SUBSTR(last\_name || ', ' || first\_name,1,20) "Sales Rep Name"

FROM employees

WHERE lower(job\_title) LIKE 'sales%'

**INTERSECT**

SELECT SUBSTR(last\_name || ', ' || first\_name,1,20) "Contact Name"

FROM customers JOIN contacts USING (customer\_id)

WHERE credit\_limit < 500

ORDER BY 1;

**Sales Rep Name**

**----------------**

Cruz, Napoleon

1. Display ALL Full Names of both Sales Reps and Customer Contacts in alphabetical order (including duplicates) 🡪 use UNION ALL

SELECT SUBSTR(last\_name || ', ' || first\_name,1,20) "Sales Rep Name"

FROM employees

WHERE lower(job\_title) LIKE 'sales%'

**UNION ALL**

SELECT SUBSTR(last\_name || ', ' || first\_name,1,20) "Contact Name"

FROM customers JOIN contacts USING (customer\_id)

WHERE credit\_limit < 500

ORDER BY 1

**Sales Rep Name**

**------------------**

Arnold, Rena

Atkinson, Jaclyn

Barrera, Willie

Boyd, Imogen

Brady, Amber

Buchanan, Brandie

Buckley, Trula

Cage, Christian

Calhoun, Cora

Callahan, Nathan

Cochran, Mireya

Cole, Isabella

Costner, Daniel

Crawford, Lucy

Cruise, Matthias

**Cruz, Napoleon 🡪 same name appears twice (one duplicate)**

**Cruz, Napoleon**

Daltrey, Maurice

Derek, Dianne

Dixon, Elizabeth

Etc more rows

Webb, Charlotte

Wells, Alice

West, Mia

Woods, Jessica

1. rows selected.
2. Display ALL Full Names of both Sales Reps and Customer Contacts in alphabetical order (excluding duplicates) 🡪 use UNION

SELECT SUBSTR(last\_name || ', ' || first\_name,1,20) "Sales Rep Name"

FROM employees

WHERE lower(job\_title) LIKE 'sales%'

**UNION**

SELECT SUBSTR(last\_name || ', ' || first\_name,1,20) "Contact Name"

FROM customers JOIN contacts USING (customer\_id)

WHERE credit\_limit < 500

ORDER BY 1

**Sales Rep Name**

**------------------**

Arnold, Rena

Atkinson, Jaclyn

Barrera, Willie

Boyd, Imogen

Brady, Amber

Buchanan, Brandie

Buckley, Trula

Cage, Christian

Calhoun, Cora

Callahan, Nathan

Cochran, Mireya

Cole, Isabella

Costner, Daniel

Crawford, Lucy

Cruise, Matthias

**Cruz, Napoleon 🡪 this name shows up only once**

Daltrey, Maurice

Derek, Dianne

Dixon, Elizabeth

Etc. more rows

Wallace, Ella

Warren, Esme

Webb, Charlotte

Wells, Alice

West, Mia

Woods, Jessica

1. rows selected. **🡪 One row less, there is NO two Cruz, Napoleon names**
2. Display Full Names of only Sales Reps who do NOT have the same full name as Customer Contacts 🡪 use MINUS

SELECT SUBSTR(last\_name || ', ' || first\_name,1,20) "Sales Rep Name"

FROM employees WHERE lower(job\_title) LIKE 'sales%'

**MINUS**

SELECT SUBSTR(last\_name || ', ' || first\_name,1,20) "Contact Name"

FROM customers JOIN contacts USING (customer\_id)

WHERE credit\_limit < 500

ORDER BY 1

**Sales Rep Name**

**-----------------**

Boyd, Imogen

Cole, Isabella

Crawford, Lucy

Dixon, Elizabeth

Ellis, Grace

Fisher, Lily

Freeman, Florence

Gibson, Scarlett

Gomez, Freya

Harrison, Evie

Henry, Elsie

Hicks, Sofia

Hunter, Millie

Jordan, Poppy

Kennedy, Maya

Marshall, Isabelle

Mason, Layla

Mcdonald, Ruby

Morales, Rosie

Murray, Phoebe

Ortiz, Daisy

Owens, Sophie

Porter, Eva

Reynolds, Sophia

Simpson, Sienna

Stevens, Matilda

Sullivan, Ava

Tucker, Evelyn

Wallace, Ella

Warren, Esme

Webb, Charlotte

Wells, Alice

West, Mia

Woods, Jessica

34 rows selected. **🡪 Cruz, Napoleon is out here (it is in both sets)**

Recapitulation of SET arithmetic:

1. Figure out the INTERSECT value (**int**) for BOTH sets
2. UNION ALL always displays **a+b** rows from both sets, where **a** is number of rows in the first set and **b** in the second set
3. UNION displays **a+b-int** rows
4. MINUS displays **a-int** rows or **b-int** rows

II) Secondly, we will display only Last Name to see the difference in the SET outputs

A) Find out do we have Sales Reps with the same LAST NAME as Customer Contacts 🡪 use INTERSECT.

SELECT SUBSTR(last\_name,1,20) "Rep Last Name"

FROM employees

WHERE lower(job\_title) LIKE 'sales%'

INTERSECT

SELECT SUBSTR(last\_name,1,20)

FROM customers JOIN contacts USING (customer\_id)

WHERE credit\_limit < 500

ORDER BY 1;

**Rep Last Name**

**--------------**

Cruz

Mason

Mcdonald **🡪 Now we have 4 last names the same in both sets**

Ortiz

1. Display ALL Last Names of both Sales Reps and Customer Contacts in alphabetical order (including duplicates) 🡪 use UNION ALL

SELECT SUBSTR(last\_name,1,20) "Rep Last Name"

FROM employees

WHERE lower(job\_title) LIKE 'sales%'

UNION ALL

SELECT SUBSTR(last\_name,1,20)

FROM customers JOIN contacts USING (customer\_id)

WHERE credit\_limit < 500

ORDER BY 1;

**Rep Last Name**

**--------------------**

Arnold

Atkinson

Barrera

Boyd

Brady

Buchanan

Buckley

Cage

Calhoun

Callahan

Cochran

Cole

Costner

Crawford

Cruise

**Cruz**

**Cruz**

Daltrey

Derek

Dixon

Etc. More rows

Marshall

Martin

**Mason**

**Mason**

Massey

**Mcdonald**

**Mcdonald**

Montoya

Morales

Murray

Odom

**Ortiz**

**Ortiz**

Owens

Pacino

Etc. More rows

Webb

Wells

West

Woods

1. rows selected.
2. Display ALL Last Names of both Sales Reps and Customer Contacts in alphabetical order (excluding duplicates) 🡪 use UNION

SELECT SUBSTR(last\_name,1,20) "Rep Last Name"

FROM employees

WHERE lower(job\_title) LIKE 'sales%'

UNION

SELECT SUBSTR(last\_name,1,20)

FROM customers JOIN contacts USING (customer\_id)

WHERE credit\_limit < 500

ORDER BY 1;

**Rep Last Name**

**--------------------**

Arnold

Atkinson

Barrera

Boyd

Brady

Buchanan

Buckley

Cage

Calhoun

Callahan

Cochran

Cole

Costner

Crawford

Cruise

Cruz

Daltrey

Derek

Dixon

Etc. More rows

Marshall

Martin

Mason

Massey

Mcdonald

Montoya

Morales

Murray

Odom

Ortiz

Owens

Etc. More rows

Webb

Wells

West

Woods

1. rows selected.
2. Display Last Names of only Sales Reps who do NOT have the same last name as Customer Contacts 🡪 use MINUS

SELECT SUBSTR(last\_name,1,20) "Rep Last Name"

FROM employees

WHERE lower(job\_title) LIKE 'sales%'

MINUS

SELECT SUBSTR(last\_name,1,20)

FROM customers JOIN contacts USING (customer\_id)

WHERE credit\_limit < 500

ORDER BY 1;

**Rep Last Name**

**--------------**

Boyd

Cole

Crawford

Dixon

Ellis

Fisher

Freeman

Gibson

Gomez

Harrison

Henry

Hicks

Hunter

Jordan

Kennedy

Marshall

Morales

Murray

Owens

Porter

Reynolds

Simpson

Stevens

Sullivan

Tucker

Wallace

Warren

Webb

Wells

West

Woods

31 rows selected.

III) Thirdly, lets avoid JOIN between tables Locations and Warehouses to display Location Id (common column for both tables), City (only with Locations) and Warehouse Name (only with Warehouses).

Notice, how we need to use Dummy columns and TO\_CHAR function to ensure the same number of columns and the same datatypes for both SELECT statements.

SELECT location\_id "Loc#", SUBSTR(City,1,20) "City",

TO\_CHAR(null) "Warehouse Name"

FROM locations

UNION

SELECT location\_id , to\_char(null), SUBSTR(warehouse\_name,1,20)

FROM warehouses

ORDER by 1 ;

**Loc# City Warehouse Name**

**---------- -------------------- --------------------**

1 Roma

2 Venice

3 Tokyo

4 Hiroshima

5 Southlake

5 Southlake, Texas

6 South San Francisco

6 San Francisco

7 South Brunswick

7 New Jersey

8 Seattle

8 Seattle, Washington

9 Toronto

9 Toronto

10 Whitehorse

11 Beijing

11 Beijing

12 Bombay

12 Bombay

13 Sydney

13 Sydney

14 Singapore

15 London

16 Oxford

17 Stretford

18 Munich

19 Sao Paulo

20 Geneva

21 Bern

22 Utrecht

23 Mexico City

23 Mexico City

32 rows selected.

You can see that this looks like LEFT OUTER JOIN, because it shows also Locations (Cities) without Warehouses. We can improve this one by showing only Cities that host Warehouses.

SELECT location\_id "Loc#", SUBSTR(City,1,20) "City",

TO\_CHAR(null) "Warehouse Name" FROM locations

WHERE location\_id IN (SELECT DISTINCT location\_id FROM Warehouses)

UNION

SELECT location\_id , to\_char(null), SUBSTR(warehouse\_name,1,20)

FROM warehouses

ORDER by 1 ;

**Loc# City Warehouse Name**

**---------- -------------------- --------------------**

5 Southlake

5 Southlake, Texas

6 South San Francisco

6 San Francisco

7 South Brunswick

7 New Jersey

8 Seattle

8 Seattle, Washington

9 Toronto

9 Toronto

11 Beijing

11 Beijing

12 Bombay

12 Bombay

13 Sydney

13 Sydney

23 Mexico City

23 Mexico City

18 rows selected.