Chapter 4

How to retrieve data from two or more tables

Objectives

Applied

- Use the explicit syntax to code an inner join that returns data from a single table or multiple tables.
- Use the explicit syntax to code an outer join.
- Code a union that combines data from a single table or multiple tables.

Objectives (cont.)

Knowledge

- Explain when column names need to be qualified.
- Describe the proper use of a table alias.
- Describe the differences between an inner join, a left outer join, a right outer join, a full outer join, and a cross join.
- Describe how to combine inner and outer joins.
- Describe the use of the implicit syntax for coding joins.
- Describe the use of the USING and NATURAL keywords for coding joins.

The explicit syntax for an inner join

```
SELECT select_list
FROM table_1
     [INNER] JOIN table_2
         ON join_condition_1
    [[INNER] JOIN table_3
         ON join_condition_2]...
```

An inner join of the Vendors and Invoices tables

```
SELECT invoice_number, vendor_name
FROM vendors INNER JOIN invoices
    ON vendors.vendor_id = invoices.vendor_id
ORDER BY invoice_number
```

invoice_number	vendor_name
0-2058	Malloy Lithographing Inc
0-2060	Malloy Lithographing Inc
0-2436	Malloy Lithographing Inc
1-200-5164	Federal Express Corporation
1-202-2978	Federal Express Corporation
10843	Yesmed, Inc

```
(114 rows)
```

The syntax for an inner join that uses table aliases

```
SELECT select_list
FROM table_1 a1
    [INNER] JOIN table_2 a2
        ON a1.column_name operator a2.column_name
   [[INNER] JOIN table_3 a3
        ON a2.column_name operator a3.column_name]...
```

An inner join with aliases for all tables

```
SELECT invoice_number, vendor_name, invoice_due_date,
    invoice_total - payment_total - credit_total
    AS balance_due
FROM vendors v JOIN invoices i
    ON v.vendor_id = i.vendor_id
WHERE invoice_total - payment_total - credit_total > 0
ORDER BY invoice_due_date DESC
```

invoice_number	vendor_name	invoice_due_date	balance_due
547480102	Blue Cross	2014-08-31	224.00
0-2436	Malloy Lithographing Inc	2014-08-30	10976.06
9982771	Ford Motor Credit Company	2014-08-23	503.20
P-0608	Malloy Lithographing Inc	2014-08-22	19351.18

An inner join with an alias for only one table

```
SELECT invoice_number, line_item_amount,
    line_item_description
FROM invoices JOIN invoice_line_items line_items
    ON invoices.invoice_id = line_items.invoice_id
WHERE account_number = 540
ORDER BY invoice_date
```

invoice_number	line_item_amount	line_item_description
I77271-O01	478.00	Publishers Marketing
972110	207.78	Prospect list
133560	175.00	Card deck advertising
97/522	765.13	Catalog design

```
(6 rows)
```

The syntax of a table name that's qualified with a database name

```
database_name.table_name
```

A join to a table in another database

```
SELECT vendor_name, customer_last_name,
    customer_first_name, vendor_state AS state,
    vendor_city AS city
FROM vendors v
    JOIN om.customers c
    ON v.vendor_zip_code = c.customer_zip
ORDER BY state, city
```

vendor_name	customer_last_name	customer_first_name	state	city
Wells Fargo Bank	Marissa	Kyle	AZ	Phoenix
Aztek Label	Irvin	Ania	CA	Anaheim
Digital Dreamworks	Neftaly	Thalia	CA	Fresno
Dataforms/West	Neftaly	Thalia	CA	Fresno
Gostanian General Building	Neftaly	Thalia	CA	Fresno
Gary McKeighan Insurance	Holbrooke	Rashad	CA	Fresno
Zylka Design	Holbrooke	Rashad	CA	Fresno
Costco	Holbrooke	Rashad	CA	Fresno
Digital Dreamworks	Holbrooke	Rashad	CA	Fresno
Dataforms/West	Holbrooke	Rashad	CA	Fresno
Lou Gentile's Flower Basket	Damien	Deborah	CA	Fresno
Wakefield Co	Neftaly	Thalia	CA	Fresno

```
(37 rows)
```

The Customers table

customer_id	customer_last_name	customer_first_name	customer_address	customer_city	customer_state	custom
1	Anders	Maria	345 Winchell Pl	Anderson	IN	46014
2	Trujillo	Ana	1298 E Smathers St	Benton	AR	72018
3	Moreno	Antonio	6925 N Parkland Ave	Puyallup	WA	98373
4	Hardy	Thomas	83 d'Urberville Ln	Casterbridge	GA	31209
5	Berglund	Christina	22717 E 73rd Ave	Dubuque	IA	52004
6	Moos	Hanna	1778 N Bovine Ave	Peoria	IL	61638

(24 rows)

The Employees table

employee_id	last_name	first_name	department_number	manager_id
1	Smith	Cindy	2	NULL
2	Jones	Elmer	4	1
3	Simonian	Ralph	2	2
4	Hernandez	Olivia	1	9
5	Aaronsen	Robert	2	4
6	Watson	Denise	6	8

(9 rows)

An inner join with two conditions

```
SELECT customer_first_name, customer_last_name
FROM customers c JOIN employees e
    ON c.customer_first_name = e.first_name
   AND c.customer_last_name = e.last_name
```

customer_first_name	customer_last_name
Thomas	Hardy

```
(1 row)
```

A self-join that returns vendors from cities in common with other vendors

```
SELECT DISTINCT v1.vendor_name, v1.vendor_city,
    v1.vendor_state
FROM vendors v1 JOIN vendors v2
    ON v1.vendor_city = v2.vendor_city AND
       v1.vendor_state = v2.vendor_state AND
       v1.vendor_name <> v2.vendor_name
ORDER BY v1.vendor_state, v1.vendor_city
```

vendor_name	vendor_city	vendor_state
Wells Fargo Bank	Phoenix	AZ
Computer Library	Phoenix	AZ
AT&T	Phoenix	AZ
Aztek Label	Anaheim	CA
Blue Shield of California	Anaheim	CA
Coffee Break Service	Fresno	CA
Crown Printing	Fresno	CA
Wakefield Co	Fresno	CA

```
(84 rows)
```

A statement that joins four tables

```
SELECT vendor_name, invoice_number, invoice_date,
    line_item_amount, account_description
FROM vendors v
    JOIN invoices i
        ON v.vendor_id = i.vendor_id
    JOIN invoice_line_items li
        ON i.invoice_id = li.invoice_id
    JOIN general_ledger_accounts gl
        ON li.account_number = gl.account_number
WHERE invoice_total - payment_total - credit_total > 0
ORDER BY vendor_name, line_item_amount DESC
```

vendor_name	invoice_number	invoice_date	line_item_amount	account_description
Blue Cross	547480102	2014-08-01	224.00	Group Insurance
Cardinal Business Media, Inc.	134116	2014-07-28	90.36	Direct Mail Advertising
Data Reproductions Corp	39104	2014-07-10	85.31	Book Printing Costs
Federal Express Corporation	263253270	2014-07-22	67.92	Freight
Federal Express Corporation	263253268	2014-07-21	59.97	Freight
Federal Express Corporation	963253264	2014-07-18	52.25	Freight
Federal Express Corporation	263253273	2014-07-22	30.75	Freight
Ford Motor Credit Company	9982771	2014-07-24	503.20	Travel and Accomodations

```
(11 rows)
```

The implicit syntax for an inner join

```
SELECT select_list
FROM table_1, table_2 [, table_3]...
WHERE table_1.column_name operator table_2.column_name
  [AND table_2.column_name operator table_3.column_name]...
```

Join the Vendors and Invoices tables

```
SELECT invoice_number, vendor_name
FROM vendors v, invoices i
WHERE v.vendor_id = i.vendor_id
ORDER BY invoice_number
```

invoice_number	vendor_name
0-2058	Malloy Lithographing Inc
0-2060	Malloy Lithographing Inc
0-2436	Malloy Lithographing Inc
1-200-5164	Federal Express Corporation
1-202-2978	Federal Express Corporation

```
(114 rows)
```

Join four tables

```
SELECT vendor_name, invoice_number, invoice_date,
    line_item_amount, account_description
FROM  vendors v, invoices i, invoice_line_items li,
    general_ledger_accounts gl
WHERE v.vendor_id = i.vendor_id
  AND i.invoice_id = li.invoice_id
  AND li.account_number = gl.account_number
  AND invoice_total - payment_total - credit_total > 0
ORDER BY vendor_name, line_item_amount DESC
```

vendor_name	invoice_number	invoice_date	line_item_amount	account_description
Blue Cross	547480102	2014-08-01	224.00	Group Insurance
Cardinal Business Media, Inc.	134116	2014-07-28	90.36	Direct Mail Advertising
Data Reproductions Corp	39104	2014-07-10	85.31	Book Printing Costs
Federal Express Corporation	263253270	2014-07-22	67.92	Freight
Federal Express Corporation	263253268	2014-07-21	59.97	Freight

```
(11 rows)
```

Terms to know

- Join
- Join condition
- Inner join
- Ad hoc relationship
- Qualified column name
- Table alias
- Schema
- Self-join
- Explicit syntax (SQL-92)
- Implicit syntax

The explicit syntax for an outer join

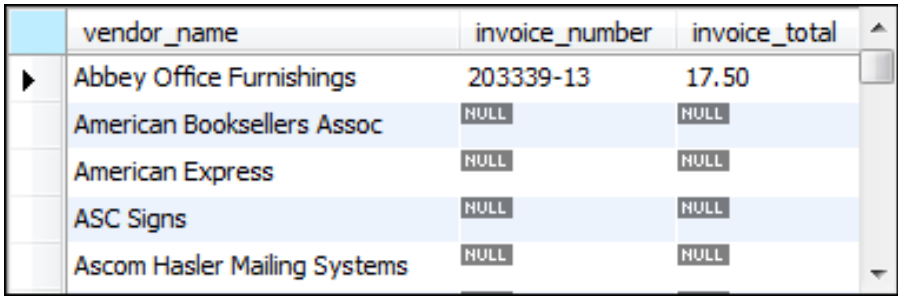
```
SELECT select_list
FROM table_1
    {LEFT|RIGHT} [OUTER] JOIN table_2
        ON join_condition_1
   [{LEFT|RIGHT} [OUTER] JOIN table_3
        ON join_condition_2]...
```

What outer joins do

Joins of this type	Retrieve unmatched rows from
Left outer join	The first (left) table
Right outer join	The second (right) table

A left outer join

```
SELECT vendor_name, invoice_number, invoice_total
FROM vendors LEFT JOIN invoices
    ON vendors.vendor_id = invoices.vendor_id
ORDER BY vendor_name
```



vendor_name	invoice_number	invoice_total
Abbey Office Furnishings	203339-13	17.50
American Booksellers Assoc	NULL	NULL
American Express	NULL	NULL
ASC Signs	NULL	NULL
Ascom Hasler Mailing Systems	NULL	NULL

```
(202 rows)
```

The Departments table

department_number	department_name
1	Accounting
2	Payroll
3	Operations
4	Personnel
5	Maintenance

The Employees table

employee_id	last_name	first_name	department_number	manager_id
1	Smith	Cindy	2	NULL
2	Jones	Elmer	4	1
3	Simonian	Ralph	2	2
4	Hernandez	Olivia	1	9
5	Aaronsen	Robert	2	4
6	Watson	Denise	6	8
7	Hardy	Thomas	5	2
8	O'Leary	Rhea	4	9
9	Locario	Paulo	6	1

The Projects table

project_number	employee_id
P1011	8
P1011	4
P1012	3
P1012	1
P1012	5
P1013	6
P1013	9
P1014	10

A left outer join

```
SELECT department_name, d.department_number, last_name
FROM departments d
    LEFT JOIN employees e
    ON d.department_number = e.department_number
ORDER BY department_name
```

department_name	department_number	last_name
Accounting	1	Hernandez
Maintenance	5	Hardy
Operations	3	NULL
Payroll	2	Smith
Payroll	2	Simonian
Payroll	2	Aaronsen
Personnel	4	Jones
Personnel	4	O'Leary

```
(8 rows)
```

A right outer join

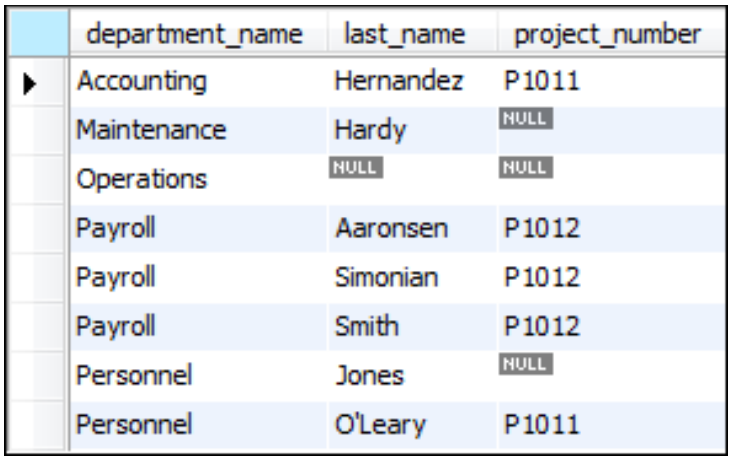
```
SELECT department_name, e.department_number, last_name
FROM departments d
    RIGHT JOIN employees e
    ON d.department_number = e.department_number
ORDER BY department_name
```

department_name	department_number	last_name
NULL	6	Watson
NULL	6	Locario
Accounting	1	Hernandez
Maintenance	5	Hardy
Payroll	2	Smith
Payroll	2	Simonian
Payroll	2	Aaronsen
Personnel	4	Jones
Personnel	4	O'Leary

```
(9 rows)
```

Join three tables using left outer joins

```
SELECT department_name, last_name, project_number
FROM departments d
    LEFT JOIN employees e
        ON d.department_number = e.department_number
    LEFT JOIN projects p
        ON e.employee_id = p.employee_id
ORDER BY department_name, last_name
```



department_name	last_name	project_number
Accounting	Hernandez	P1011
Maintenance	Hardy	NULL
Operations	NULL	NULL
Payroll	Aaronsen	P1012
Payroll	Simonian	P1012
Payroll	Smith	P1012
Personnel	Jones	NULL
Personnel	O'Leary	P1011

```
(8 rows)
```

Combine an outer and an inner join

```
SELECT department_name, last_name, project_number
FROM departments d
    JOIN employees e
        ON d.department_number = e.department_number
    LEFT JOIN projects p
        ON e.employee_id = p.employee_id
ORDER BY department_name, last_name
```

department_name	last_name	project_number
Accounting	Hernandez	P1011
Maintenance	Hardy	NULL
Payroll	Aaronsen	P1012
Payroll	Simonian	P1012
Payroll	Smith	P1012
Personnel	Jones	NULL
Personnel	O'Leary	P1011

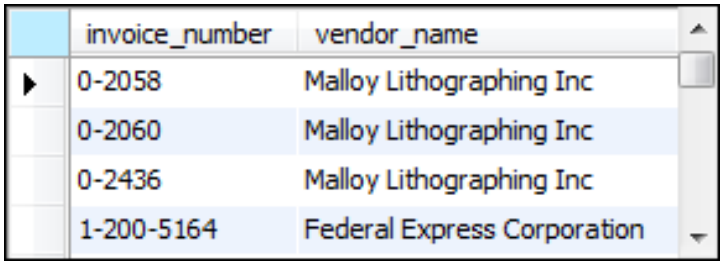
```
(7 rows)
```

The syntax for a join that uses the USING keyword

```
SELECT select_list
FROM table_1
    [{LEFT|RIGHT} [OUTER]] JOIN table_2
        USING (join_column_1[, join_column_2]...)
   [[{LEFT|RIGHT} [OUTER]] JOIN table_3
        USING (join_column_1[, join_column_2]...)]...
```

Use the USING keyword to join two tables

```
SELECT invoice_number, vendor_name
FROM vendors
    JOIN invoices USING (vendor_id)
ORDER BY invoice_number
```



invoice_number	vendor_name
0-2058	Malloy Lithographing Inc
0-2060	Malloy Lithographing Inc
0-2436	Malloy Lithographing Inc
1-200-5164	Federal Express Corporation

```
(114 rows)
```

Use the USING keyword to join three tables

```
SELECT department_name, last_name, project_number
FROM departments
    JOIN employees USING (department_number)
    LEFT JOIN projects USING (employee_id)
ORDER BY department_name
```

department_name	last_name	project_number
Accounting	Hernandez	P1011
Maintenance	Hardy	NULL
Payroll	Simonian	P1012
Payroll	Smith	P1012
Payroll	Aaronsen	P1012
Personnel	Jones	NULL
Personnel	O'Leary	P1011

```
(7 rows)
```

The syntax for a join that uses the NATURAL keyword

```
SELECT select_list
FROM table_1
     NATURAL JOIN table_2
    [NATURAL JOIN table_3]...
```

Use the NATURAL keyword to join tables

```
SELECT invoice_number, vendor_name
FROM vendors
    NATURAL JOIN invoices
ORDER BY invoice_number
```

invoice_number	vendor_name
0-2058	Malloy Lithographing Inc
0-2060	Malloy Lithographing Inc
0-2436	Malloy Lithographing Inc
1-200-5164	Federal Express Corporation

```
(114 rows)
```

Use the NATURAL keyword in a statement that joins three tables

```
SELECT department_name AS dept_name, last_name,
    project_number
FROM departments
    NATURAL JOIN employees
    LEFT JOIN projects USING (employee_id)
ORDER BY department_name
```

dept_name	last_name	project_number
Accounting	Hernandez	P1011
Maintenance	Hardy	NULL
Payroll	Simonian	P1012
Payroll	Smith	P1012
Payroll	Aaronsen	P1012
Personnel	Jones	NULL
Personnel	O'Leary	P1011

```
(7 rows)
```

The explicit syntax for a cross join

```
SELECT select_list
FROM table_1 CROSS JOIN table_2
```

A cross join that uses the explicit syntax

```
SELECT departments.department_number, department_name,
    employee_id, last_name
FROM departments CROSS JOIN employees
ORDER BY departments.department_number
```

department_number	department_name	employee_id	last_name
1	Accounting	2	Jones
1	Accounting	5	Aaronsen
1	Accounting	8	O'Leary
1	Accounting	3	Simonian
1	Accounting	6	Watson

```
(45 rows)
```

The implicit syntax for a cross join

```
SELECT select_list
FROM table_1, table_2
```

A cross join that uses the implicit syntax

```
SELECT departments.department_number, department_name,
    employee_id, last_name
FROM departments, employees
ORDER BY departments.department_number
```

department_number	department_name	employee_id	last_name
1	Accounting	2	Jones
1	Accounting	9	Locario
1	Accounting	5	Aaronsen
1	Accounting	3	Simonian
1	Accounting	6	Watson

```
(45 rows)
```

Terms to know

- Outer join
- Left outer join
- Right outer join
- Equijoin
- Natural join
- Cross join
- Cartesian product

The syntax for a union operation

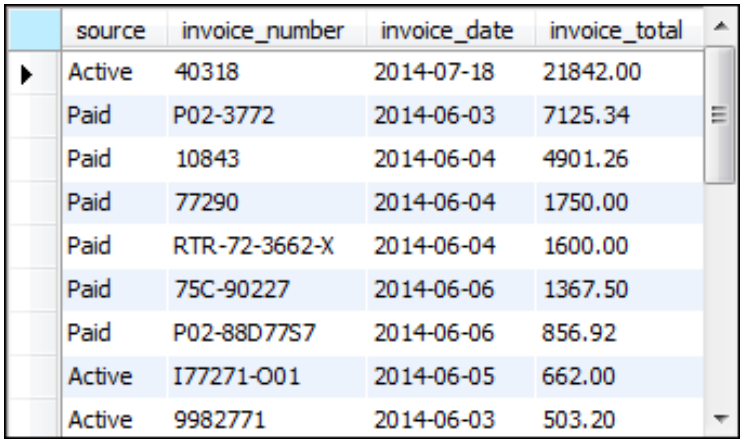
```
    SELECT_statement_1
UNION [ALL]
    SELECT_statement_2
[UNION [ALL]
    SELECT_statement_3]...
[ORDER BY order_by_list]
```

Rules for a union

- Each result set must return the same number of columns.
- The corresponding columns in each result set must have compatible data types.
- The column names in the final result set are taken from the first SELECT clause.

A union that combines result sets from two different tables

```
    SELECT 'Active' AS source, invoice_number,
        invoice_date, invoice_total
    FROM active_invoices
    WHERE invoice_date >= '2014-06-01'
UNION
    SELECT 'Paid' AS source, invoice_number,
        invoice_date, invoice_total
    FROM paid_invoices
    WHERE invoice_date >= '2014-06-01'
ORDER BY invoice_total DESC
```

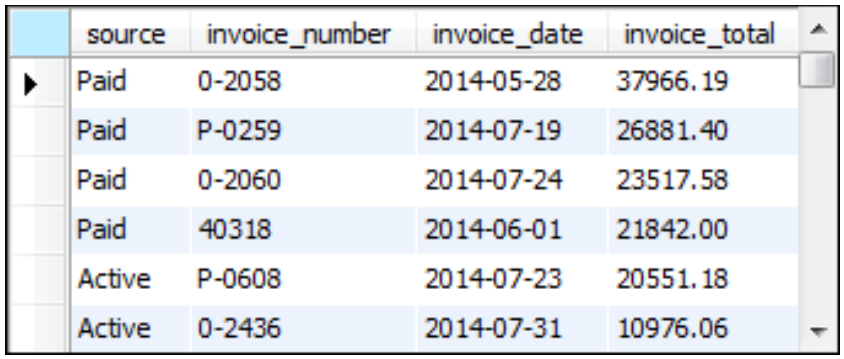


source	invoice_number	invoice_date	invoice_total
Active	40318	2014-07-18	21842.00
Paid	P02-3772	2014-06-03	7125.34
Paid	10843	2014-06-04	4901.26
Paid	77290	2014-06-04	1750.00
Paid	RTR-72-3662-X	2014-06-04	1600.00
Paid	75C-90227	2014-06-06	1367.50
Paid	P02-88D77S7	2014-06-06	856.92
Active	I77271-O01	2014-06-05	662.00
Active	9982771	2014-06-03	503.20

```
(22 rows)
```

A union that combines result sets from a single table

```
    SELECT 'Active' AS source, invoice_number,
        invoice_date, invoice_total
    FROM invoices
    WHERE invoice_total - payment_total - credit_total > 0
UNION
    SELECT 'Paid' AS source, invoice_number,
        invoice_date, invoice_total
    FROM invoices
    WHERE invoice_total - payment_total - credit_total <= 0
ORDER BY invoice_total DESC
```



source	invoice_number	invoice_date	invoice_total
Paid	0-2058	2014-05-28	37966.19
Paid	P-0259	2014-07-19	26881.40
Paid	0-2060	2014-07-24	23517.58
Paid	40318	2014-06-01	21842.00
Active	P-0608	2014-07-23	20551.18
Active	0-2436	2014-07-31	10976.06

```
(114 rows)
```

A union that combines result sets from the same two tables

```
    SELECT invoice_number, vendor_name,
        '33% Payment' AS payment_type,
        invoice_total AS total,
        invoice_total * 0.333 AS payment
    FROM invoices JOIN vendors
        ON invoices.vendor_id = vendors.vendor_id
    WHERE invoice_total > 10000
UNION
    SELECT invoice_number, vendor_name,
        '50% Payment' AS payment_type,
        invoice_total AS total,
        invoice_total * 0.5 AS payment
    FROM invoices JOIN vendors
        ON invoices.vendor_id = vendors.vendor_id
    WHERE invoice_total BETWEEN 500 AND 10000
```

A union that combines result sets from the same two tables (continued)

```
UNION
    SELECT invoice_number, vendor_name,
        'Full amount' AS payment_type,
        invoice_total AS total,
        invoice_total AS payment
    FROM invoices JOIN vendors
        ON invoices.vendor_id = vendors.vendor_id
    WHERE invoice_total < 500
ORDER BY payment_type, vendor_name, invoice_number
```

invoice_number	vendor_name	payment_type	total	payment
40318	Data Reproductions Corp	33% Payment	21842.00	7273.38600
0-2058	Malloy Lithographing Inc	33% Payment	37966.19	12642.74127
0-2060	Malloy Lithographing Inc	33% Payment	23517.58	7831.35414
0-2436	Malloy Lithographing Inc	33% Payment	10976.06	3655.02798
P-0259	Malloy Lithographing Inc	33% Payment	26881.40	8951.50620
P-0608	Malloy Lithographing Inc	33% Payment	20551.18	6843.54294
509786	Bertelsmann Industry Svcs. Inc	50% Payment	6940.25	3470.12500

```
(114 rows)
```

A union that simulates a full outer join

```
    SELECT department_name AS dept_name,
        d.department_number AS d_dept_no,
        e.department_number AS e_dept_no, last_name
    FROM departments d
        LEFT JOIN employees e
        ON d.department_number = e.department_number
UNION
    SELECT department_name AS dept_name,
        d.department_number AS d_dept_no,
        e.department_number AS e_dept_no, last_name
    FROM departments d
        RIGHT JOIN employees e
        ON d.department_number = e.department_number
ORDER BY dept_name
```

A union that simulates a full outer join (result set)

dept_name	d_dept_no	e_dept_no	last_name
NULL	NULL	6	Watson
NULL	NULL	6	Locario
Accounting	1	1	Hernandez
Maintenance	5	5	Hardy
Operations	3	NULL	NULL
Payroll	2	2	Smith
Payroll	2	2	Simonian
Payroll	2	2	Aaronsen
Personnel	4	4	O'Leary
Personnel	4	4	Jones

(10 rows)

Terms to know

- Union
- Full outer join