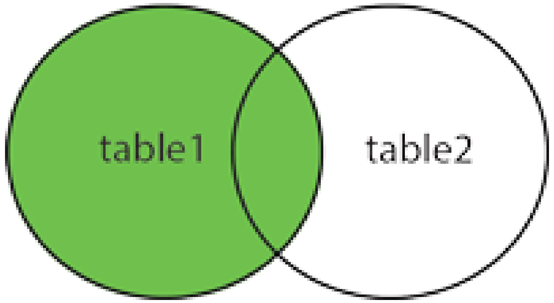
Announcement

- Term project presentation and report
  - 14th October 2021 (In class + Lab)
- Final exam
  - 17th October 2021 8am-11am

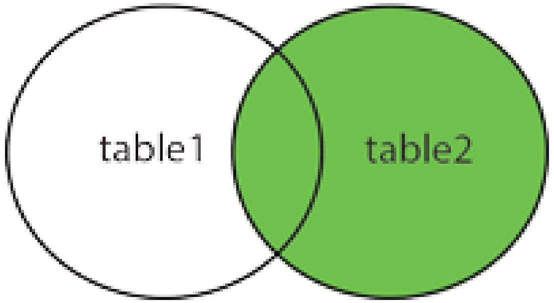
2. OUTER JOIN (3 types)

- LEFT OUTER JOIN or LEFT J
- RIGHT OUTER JOIN or RIGH
- FULL OUTER JOIN or FULL J

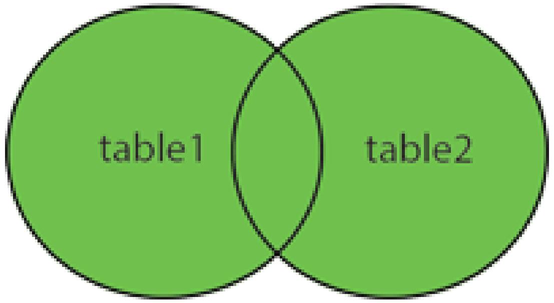
LEFT JOIN



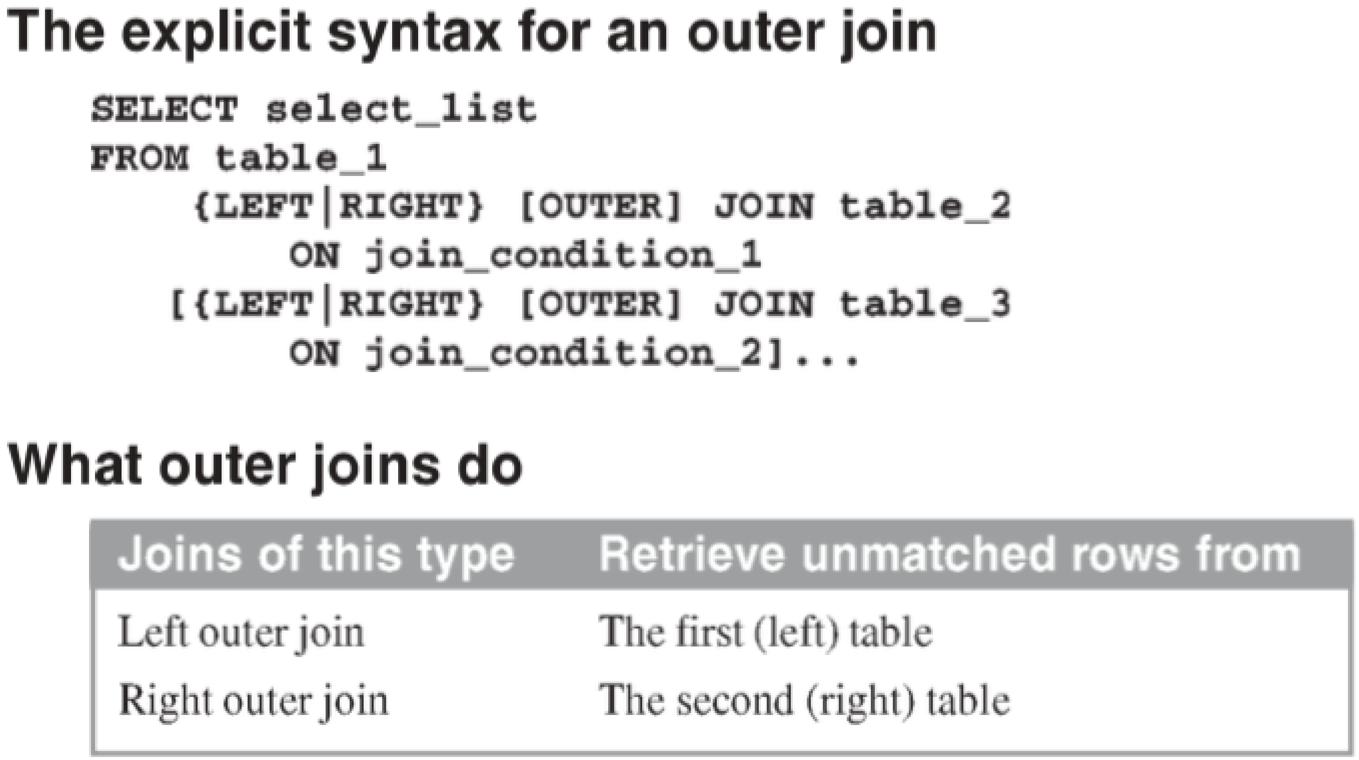
RIGHT JOIN



FULL OUTER JOIN



How to work with outer joins



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]...
```

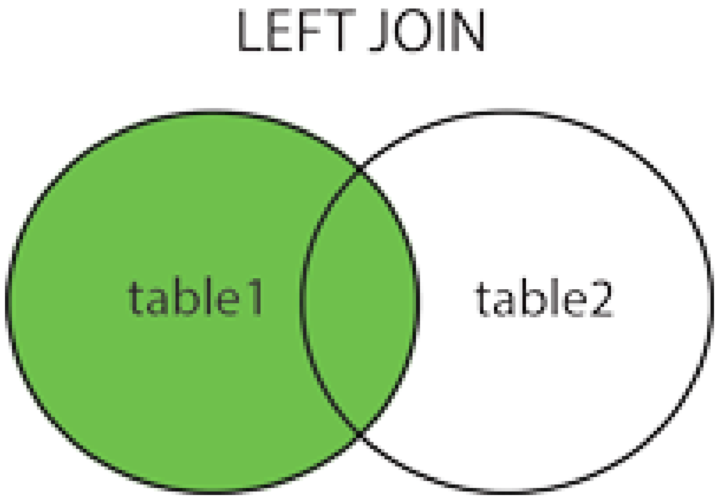
What outer joins do

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

The key OUTER is marked as optional [enclosed in square brackets]

The LEFT JOIN

- The LEFT JOIN keyword returns all rows from the left table (table1), with the matching rows in the right table (table2). The result is NULL in the right side when there is no match.



The LEFT JOIN (cont.)

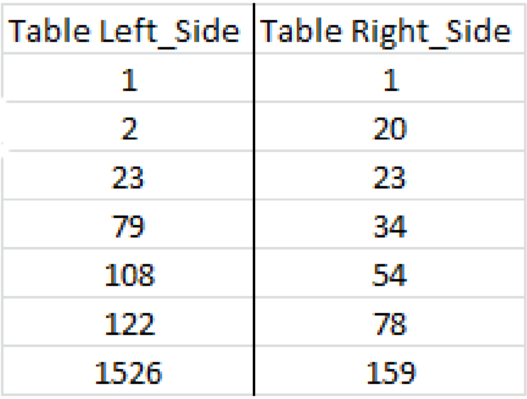
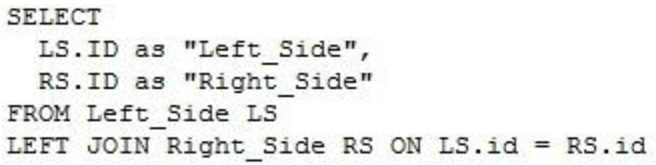
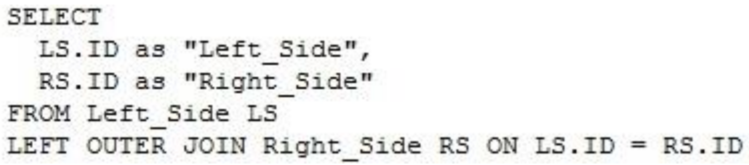


Table Left_Side	Table Right_Side
1	1
2	20
23	23
79	34
108	54
122	78
1526	159

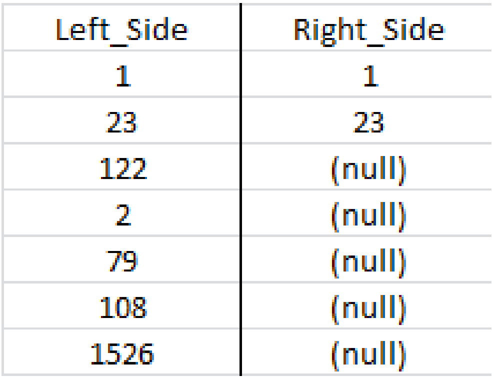


```
SELECT
  LS.ID as "Left_Side",
  RS.ID as "Right_Side"
FROM Left_Side LS
LEFT JOIN Right_Side RS ON LS.id = RS.id
```

=

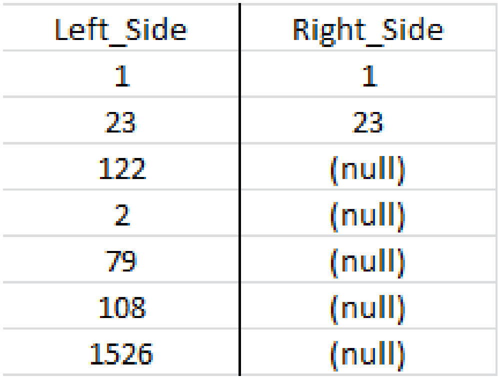


```
SELECT
  LS.ID as "Left_Side",
  RS.ID as "Right_Side"
FROM Left_Side LS
LEFT OUTER JOIN Right_Side RS ON LS.ID = RS.ID
```



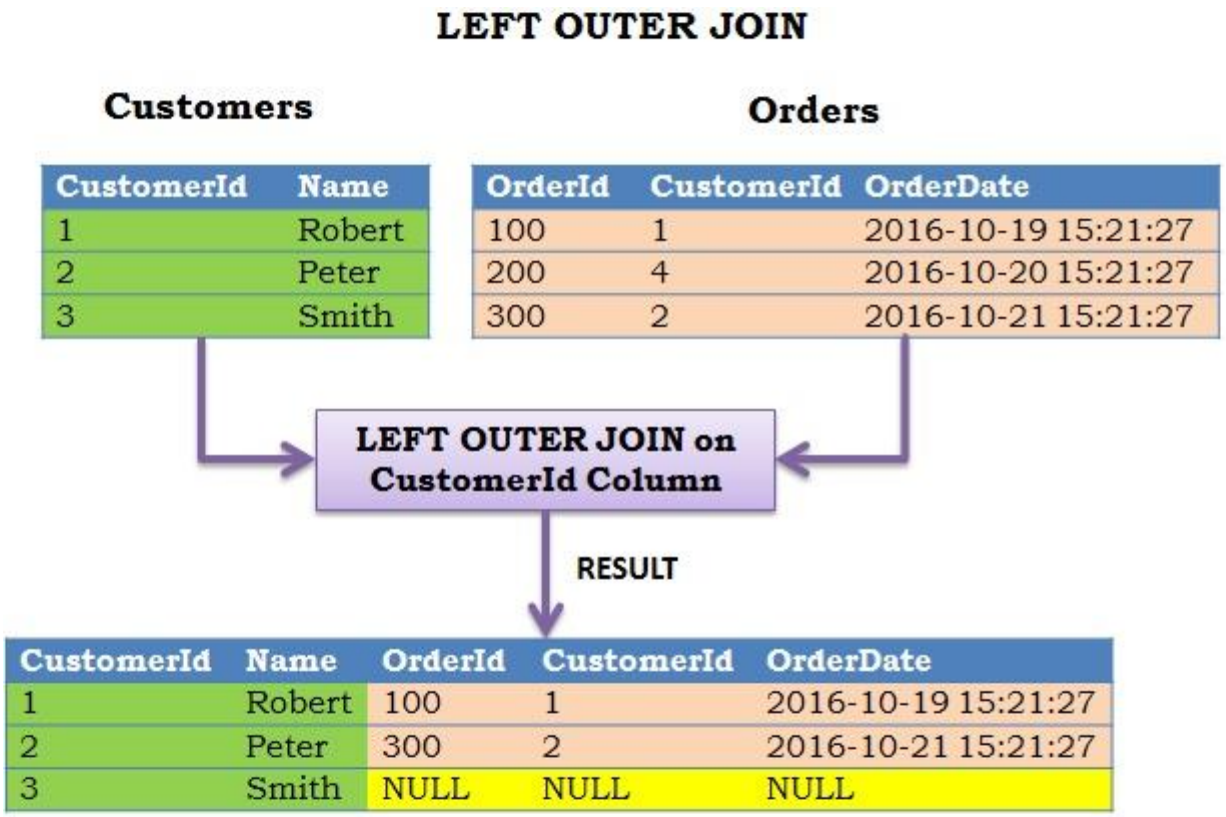
Left_Side	Right_Side
1	1
23	23
122	(null)
2	(null)
79	(null)
108	(null)
1526	(null)

=



Left_Side	Right_Side
1	1
23	23
122	(null)
2	(null)
79	(null)
108	(null)
1526	(null)

Left Join example



LEFT OUTER JOIN

Customers

CustomerId	Name
1	Robert
2	Peter
3	Smith

Orders

OrderId	CustomerId	OrderDate
100	1	2016-10-19 15:21:27
200	4	2016-10-20 15:21:27
300	2	2016-10-21 15:21:27

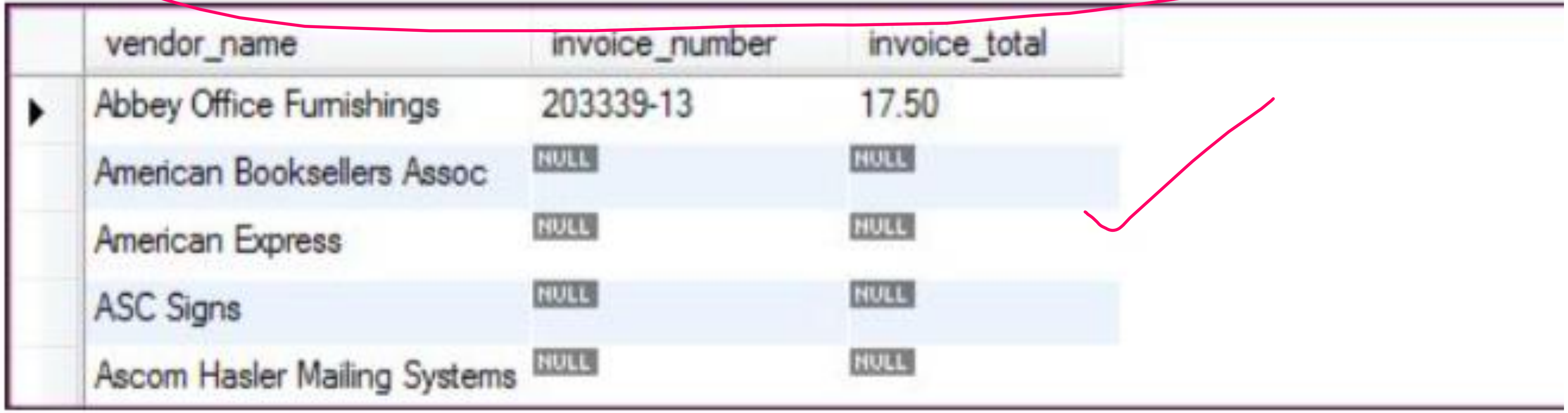
LEFT OUTER JOIN on CustomerId Column

RESULT

CustomerId	Name	OrderId	CustomerId	OrderDate
1	Robert	100	1	2016-10-19 15:21:27
2	Peter	300	2	2016-10-21 15:21:27
3	Smith	NULL	NULL	NULL

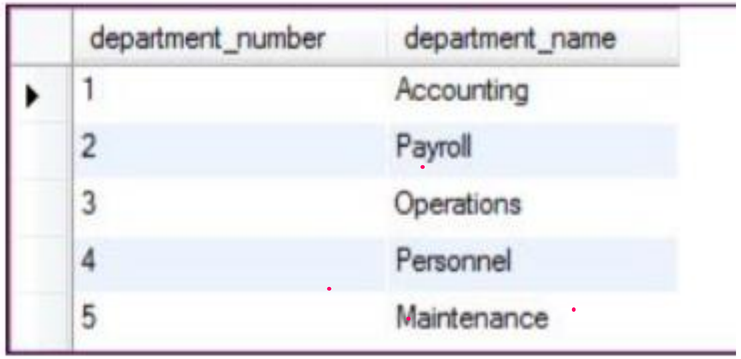
A left outer join

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



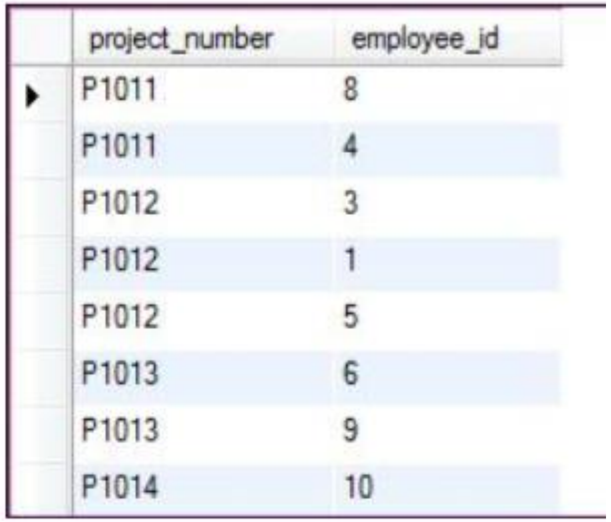
vendor_name	invoice_number	invoice_total
Abbey Office Furnishings	203339-13	17.50
American Booksellers Assoc	NULL	NULL
American Express	NULL	NULL
ASC Signs	NULL	NULL
Ascom Hasler Mailing Systems	NULL	NULL

The Departments table



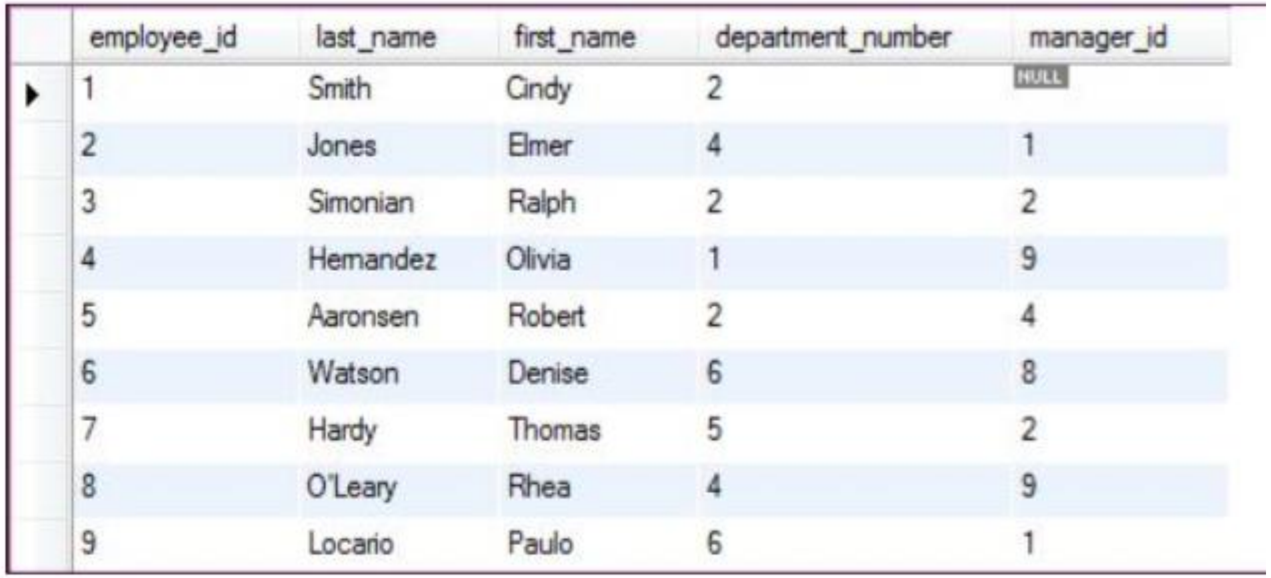
department_number	department_name
1	Accounting
2	Payroll
3	Operations
4	Personnel
5	Maintenance

The Projects table



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

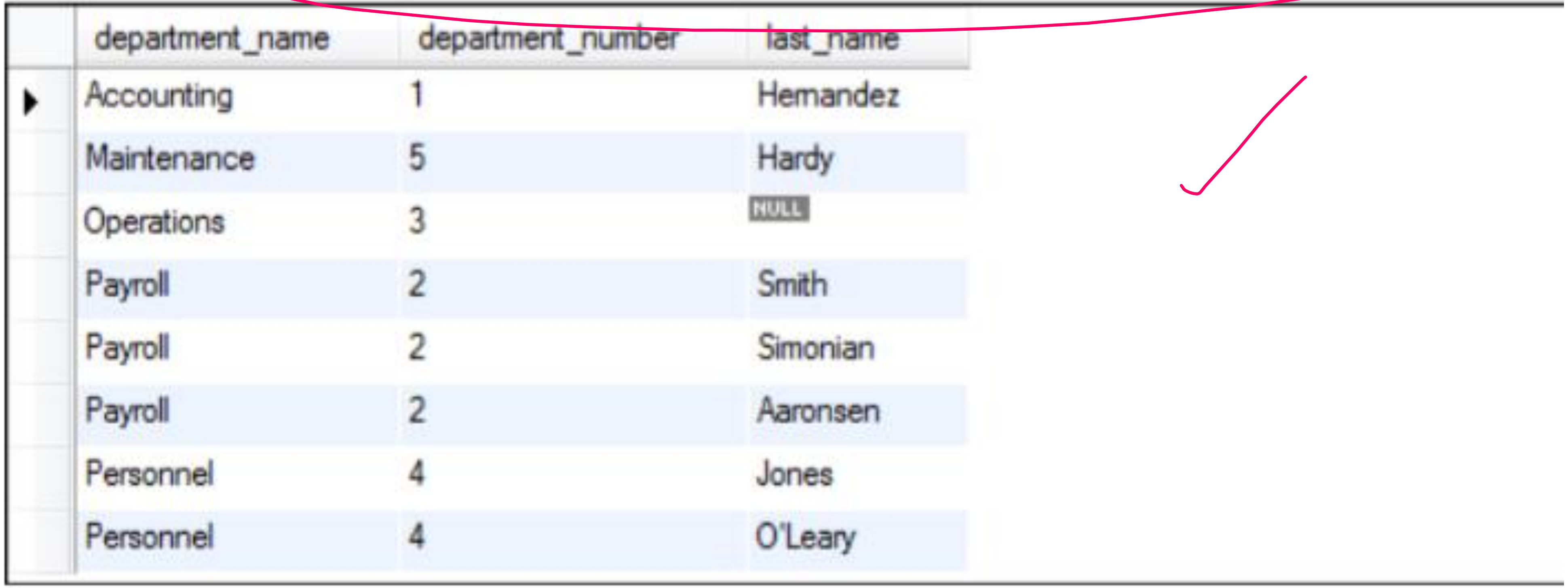
The Employees table



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

A left outer join

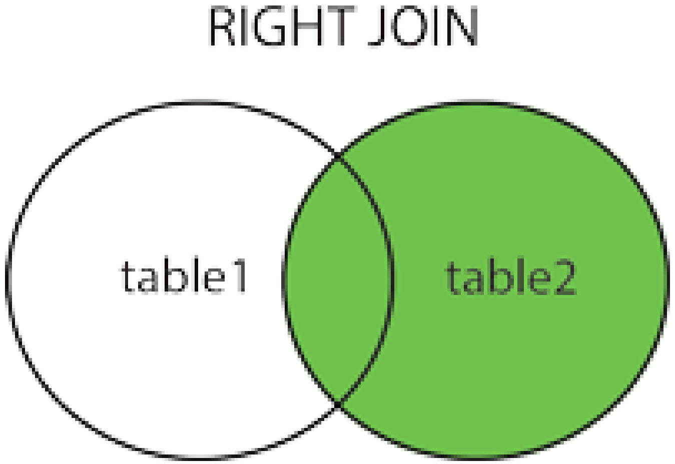
```
SELECT department_name, d.department_number, last_name
FROM departments d
    LEFT JOIN employees e
    ON d.department_number = e.department_number
ORDER BY department_name
```



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

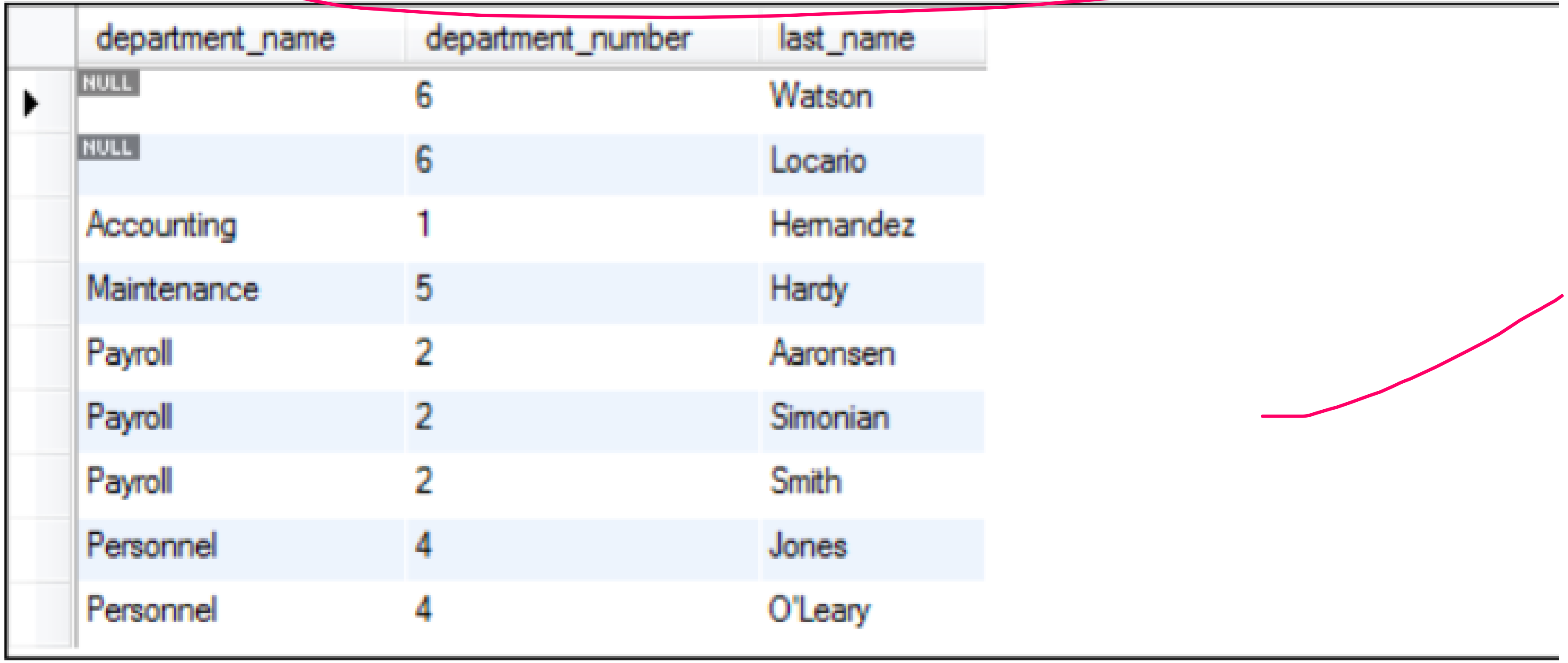
The RIGHT JOIN

- The RIGHT JOIN keyword returns all rows from the right table (table2), with the matching rows in the left table (table1). The result is NULL in the left side when there is no match.



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	Aaronsen
Payroll	2	Simonian
Payroll	2	Smith
Personnel	4	Jones
Personnel	4	O'Leary

The RIGHT JOIN

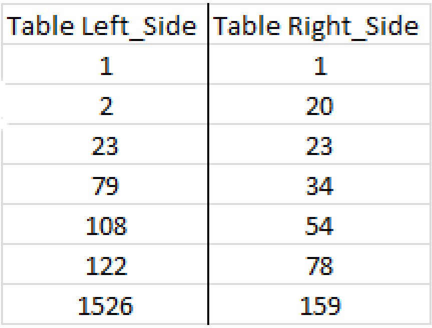
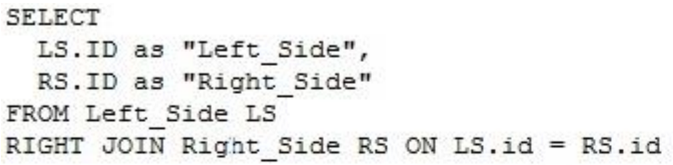
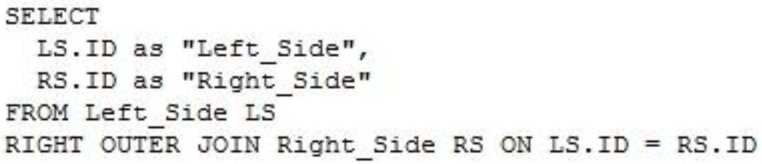


Table Left_Side	Table Right_Side
1	1
2	20
23	23
79	34
108	54
122	78
1526	159

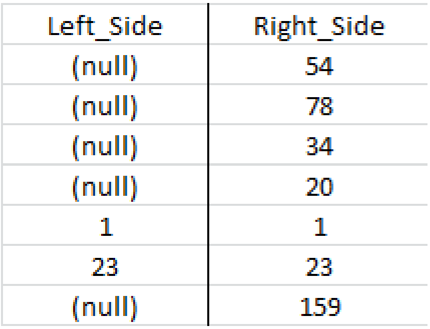


```
SELECT
  LS.ID as "Left_Side",
  RS.ID as "Right_Side"
FROM Left_Side LS
RIGHT JOIN Right_Side RS ON LS.id = RS.id
```

=

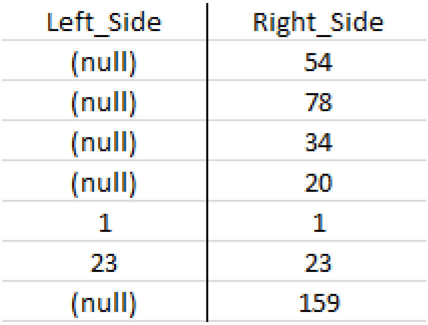


```
SELECT
  LS.ID as "Left_Side",
  RS.ID as "Right_Side"
FROM Left_Side LS
RIGHT OUTER JOIN Right_Side RS ON LS.ID = RS.ID
```



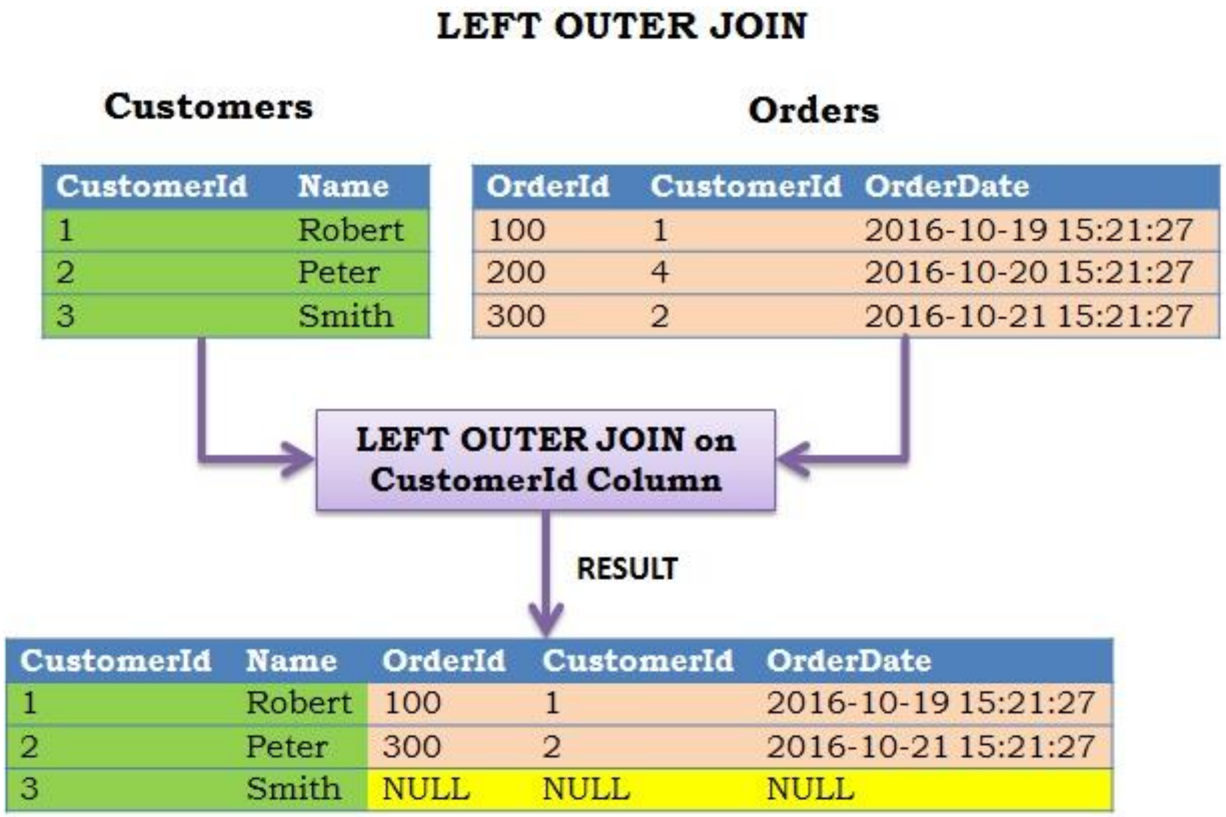
Left_Side	Right_Side
(null)	54
(null)	78
(null)	34
(null)	20
1	1
23	23
(null)	159

=



Left_Side	Right_Side
(null)	54
(null)	78
(null)	34
(null)	20
1	1
23	23
(null)	159

Right join example



LEFT OUTER JOIN

Customers

CustomerId	Name
1	Robert
2	Peter
3	Smith

Orders

OrderId	CustomerId	OrderDate
100	1	2016-10-19 15:21:27
200	4	2016-10-20 15:21:27
300	2	2016-10-21 15:21:27

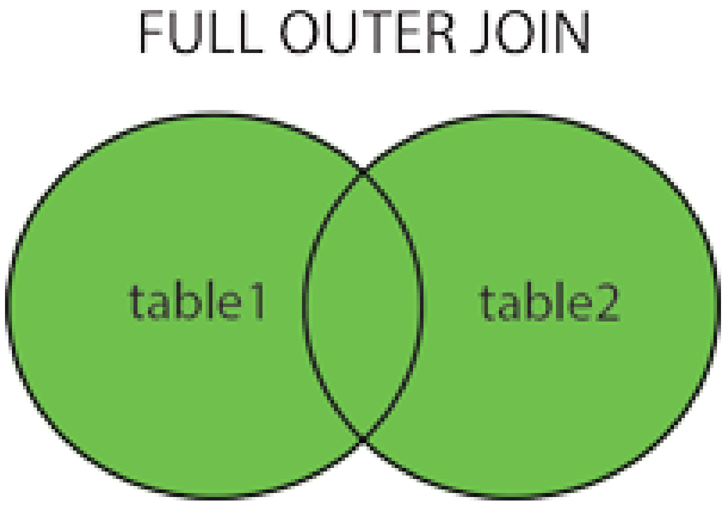
LEFT OUTER JOIN on CustomerId Column

RESULT

CustomerId	Name	OrderId	CustomerId	OrderDate
1	Robert	100	1	2016-10-19 15:21:27
2	Peter	300	2	2016-10-21 15:21:27
3	Smith	NULL	NULL	NULL

The FULL JOIN

- The FULL OUTER JOIN keyword returns all rows from the left table (table1) and from the right table (table2).The FULL OUTER JOIN keyword combines the result of both LEFT and RIGHT joins.

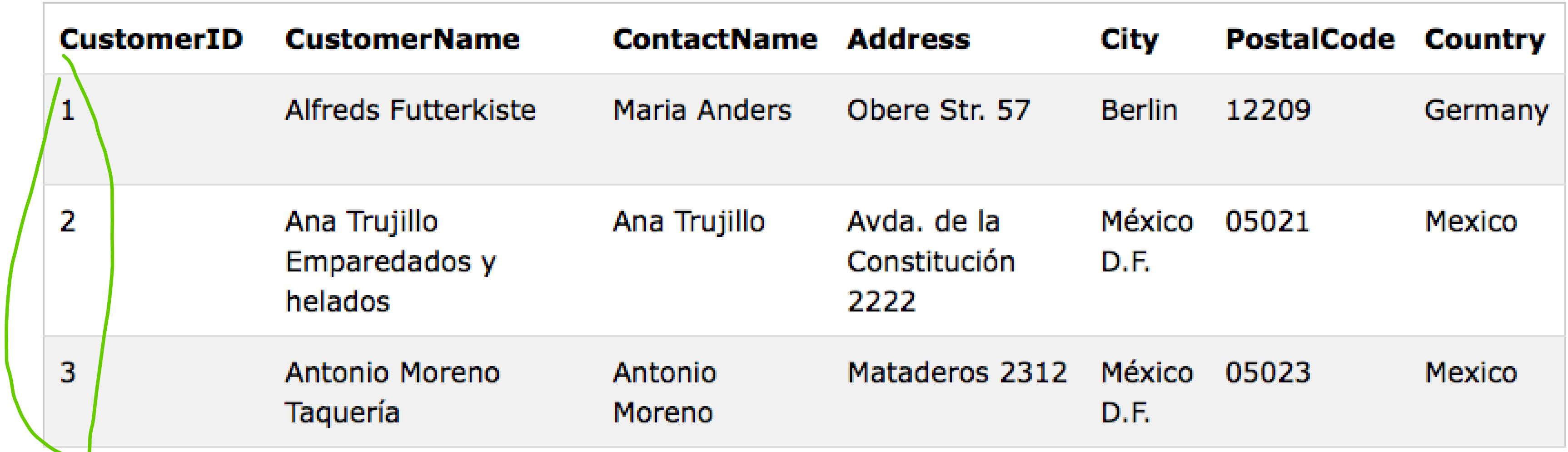


The FULL JOIN

```
SELECT column_name(s)
FROM table1
FULL OUTER JOIN table2
ON table1.column_name=table2.column_name;
```

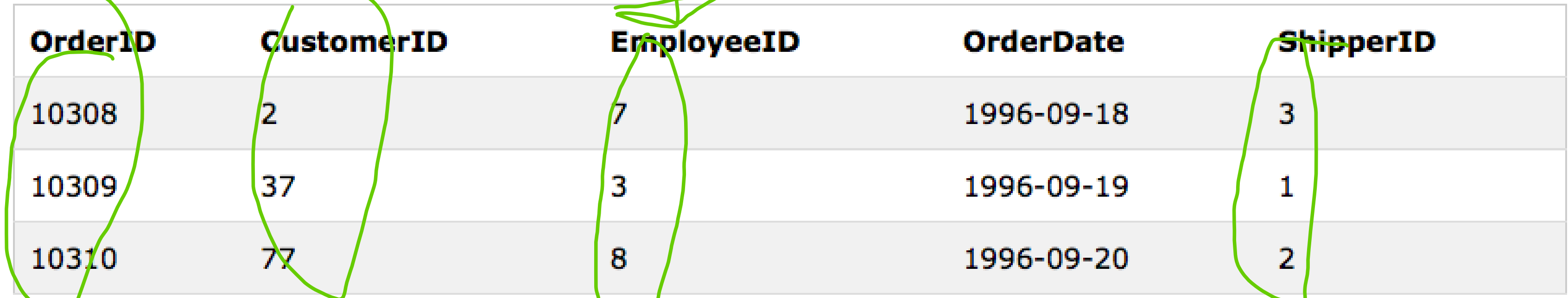
The FULL JOIN

Customers



CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
1	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209	Germany
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico
3	Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312	México D.F.	05023	Mexico

Orders



OrderID	CustomerID	EmployeeID	OrderDate	ShipperID
10308	2	7	1996-09-18	3
10309	37	3	1996-09-19	1
10310	77	8	1996-09-20	2

The FULL JOIN

SELECT Customers.CustomerName, Orders.OrderID

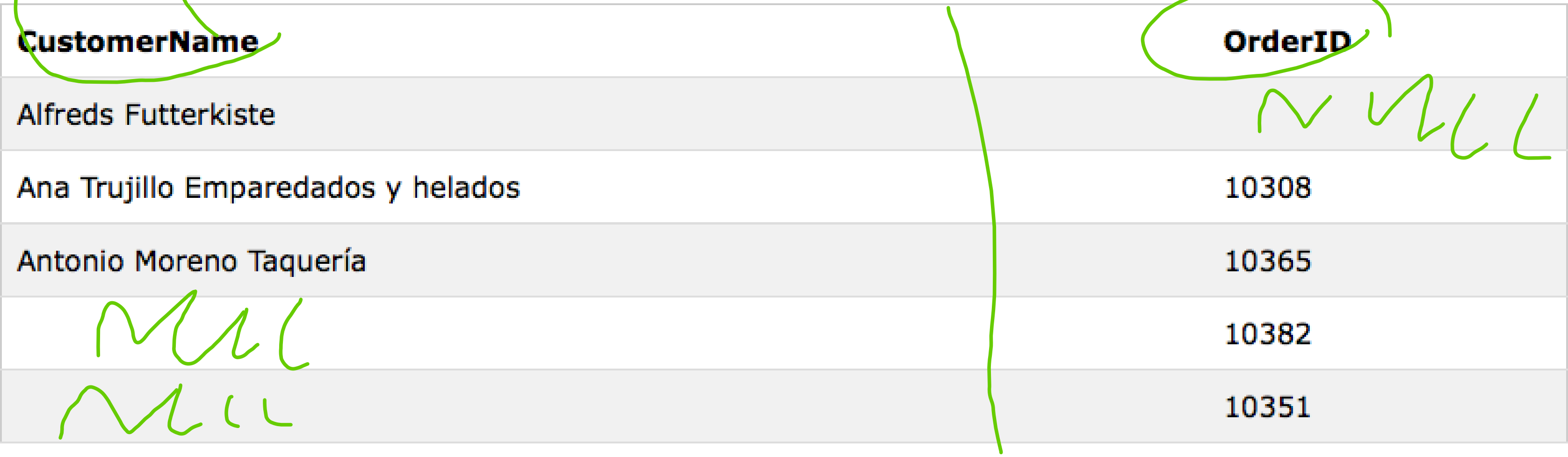
FROM Customers

FULL OUTER JOIN Orders

ON ~~Customers.CustomerID=Orders.CustomerID~~

ORDER BY Customers.Custome

The FULL OUTER JOIN keyword returns all the rows from the left table (Customers), and all the rows from the right table (Orders). If there are rows in "Customers" that do not have matches in "Orders", or if there are rows in "Orders" that do not have matches in "Customers", those rows will be listed as well.



CustomerName	OrderID
Alfreds Futterkiste	NULL
Ana Trujillo Emparedados y helados	10308
Antonio Moreno Taquería	10365
NULL	10382
NULL	10351

Workshop 3

- Use ‘ap’ database produce the same output as slide (45, 47, and 49)
- Work on term project with your teammate.

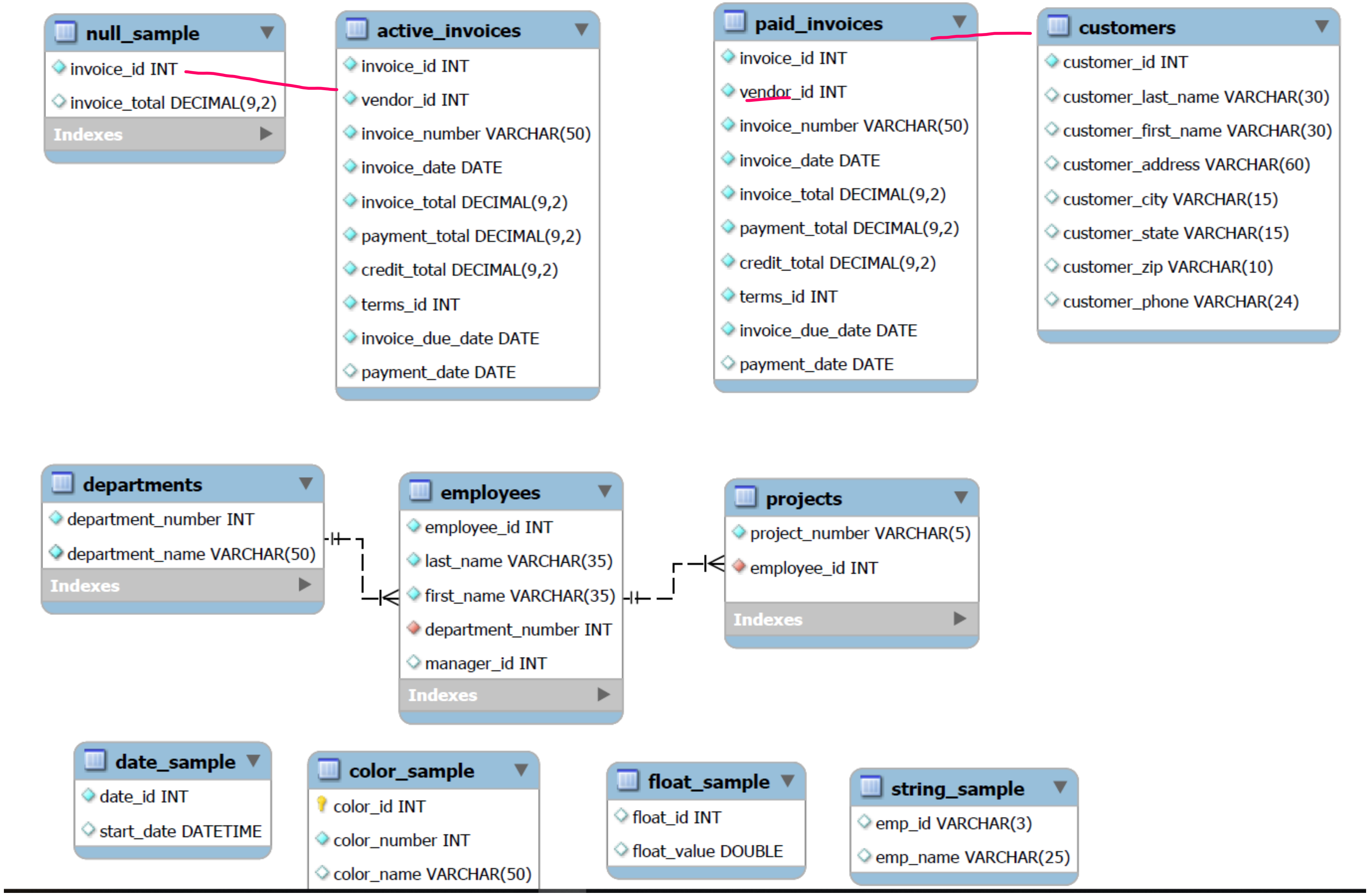
12/13/18 Agenda

- More join techniques
- Natural join
- Cross join
- Self join

Get yourself ready

- Start the XAMPP and start mysql server
- Open MySQL workbench.

EX database



null_sample
- invoice_id INT
- invoice_total DECIMAL(9,2)

Indexes

active_invoices
- invoice_id INT
- vendor_id INT
- invoice_number VARCHAR(50)
- invoice_date DATE
- invoice_total DECIMAL(9,2)
- payment_total DECIMAL(9,2)
- credit_total DECIMAL(9,2)
- terms_id INT
- invoice_due_date DATE
- payment_date DATE

paid_invoices
- invoice_id INT
- vendor_id INT
- invoice_number VARCHAR(50)
- invoice_date DATE
- invoice_total DECIMAL(9,2)
- payment_total DECIMAL(9,2)
- credit_total DECIMAL(9,2)
- terms_id INT
- invoice_due_date DATE
- payment_date DATE

customers
- customer_id INT
- customer_last_name VARCHAR(30)
- customer_first_name VARCHAR(30)
- customer_address VARCHAR(60)
- customer_city VARCHAR(15)
- customer_state VARCHAR(15)
- customer_zip VARCHAR(10)
- customer_phone VARCHAR(24)

departments
- department_number INT
- department_name VARCHAR(50)

Indexes

employees
- employee_id INT
- last_name VARCHAR(35)
- first_name VARCHAR(35)
- department_number INT
- manager_id INT

Indexes

projects
- project_number VARCHAR(5)
- employee_id INT

Indexes

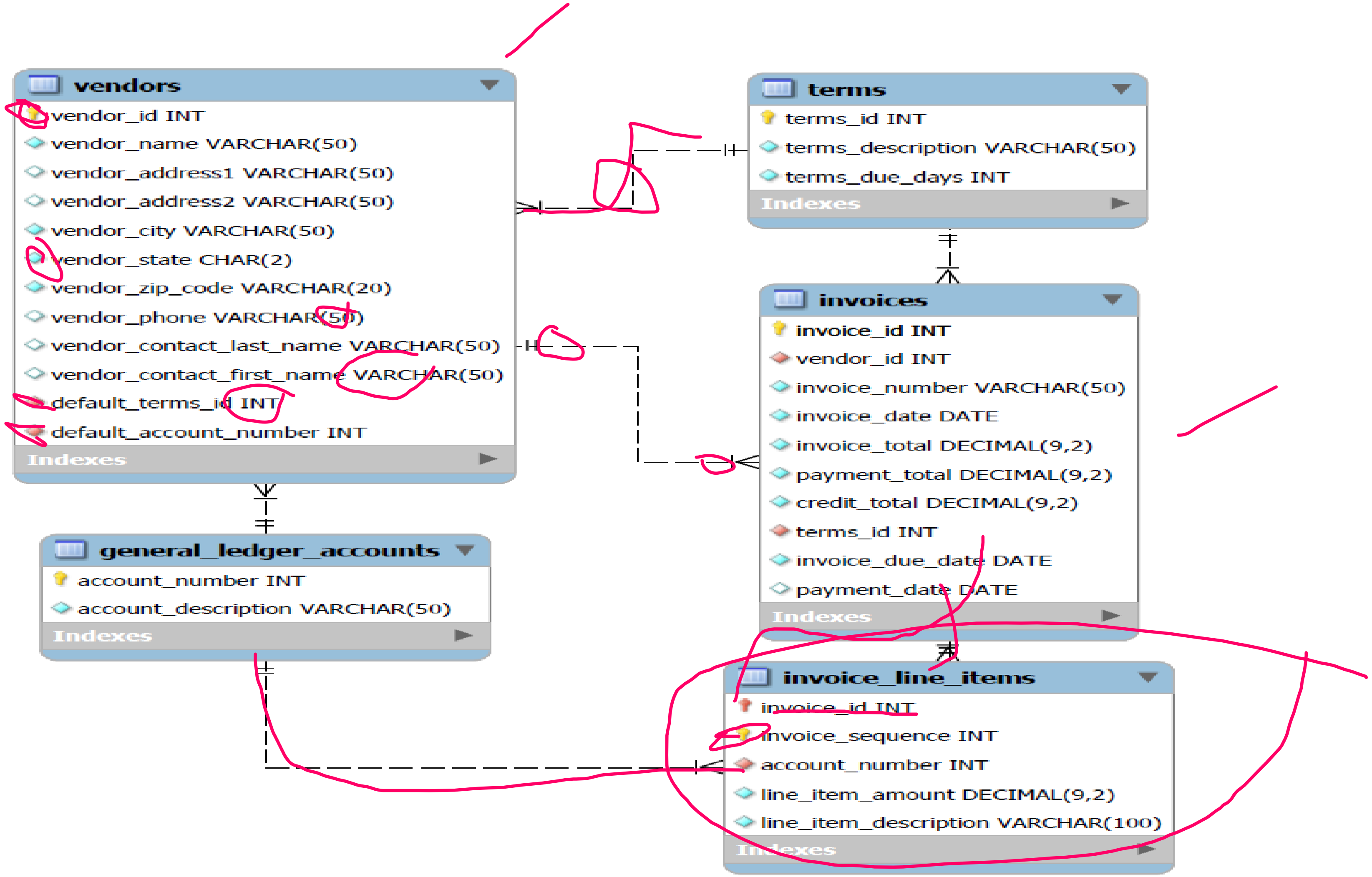
date_sample
- date_id INT
- start_date DATETIME

color_sample
- color_id INT
- color_number INT
- color_name VARCHAR(50)

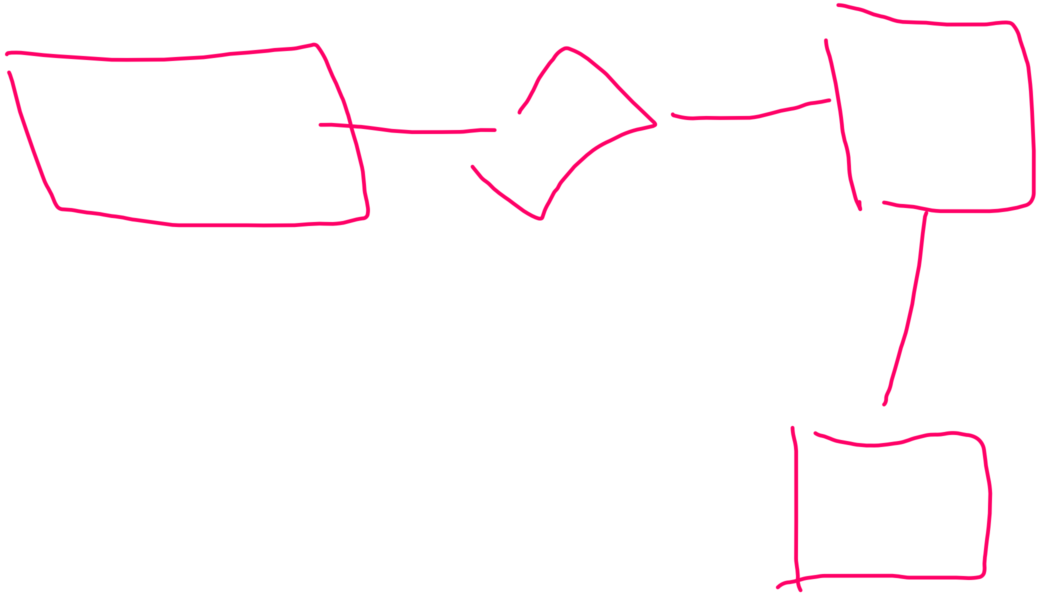
float_sample
- float_id INT
- float_value DOUBLE

string_sample
- emp_id VARCHAR(3)
- emp_name VARCHAR(25)

AP database



vendors
vendor_id INT
vendor_name VARCHAR(50)
vendor_address1 VARCHAR(50)
vendor_address2 VARCHAR(50)
vendor_city VARCHAR(50)
vendor_state CHAR(2)
vendor_zip_code VARCHAR(20)
vendor_phone VARCHAR(50)
vendor_contact_last_name VARCHAR(50)
vendor_contact_first_name VARCHAR(50)
default_terms_id INT
default_account_number INT
Indexes
terms
terms_id INT
terms_description VARCHAR(50)
terms_due_days INT
Indexes
invoices
invoice_id INT
vendor_id INT
invoice_number VARCHAR(50)
invoice_date DATE
invoice_total DECIMAL(9,2)
payment_total DECIMAL(9,2)
credit_total DECIMAL(9,2)
terms_id INT
invoice_due_date DATE
payment_date DATE
Indexes
general_ledger_accounts
account_number INT
account_description VARCHAR(50)
Indexes
invoice_line_items
invoice_id INT
invoice_sequence INT
account_number INT
line_item_amount DECIMAL(9,2)
line_item_description VARCHAR(100)
Indexes

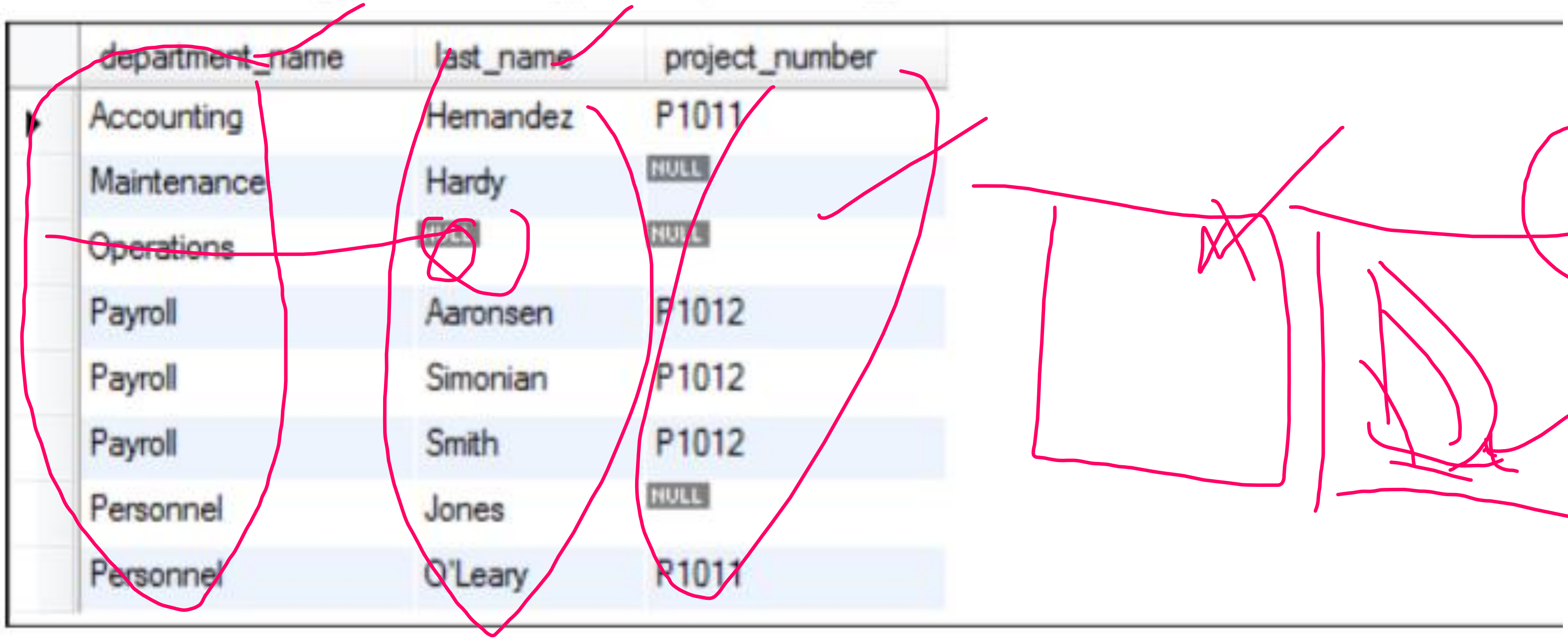


Join Technique

- Join three tables using left outer joins
- Combine an outer and an inner join
- Use the USING Keyword to join two tables
- Use the USING Keyword to join three tables

Join three tables using left outer joins

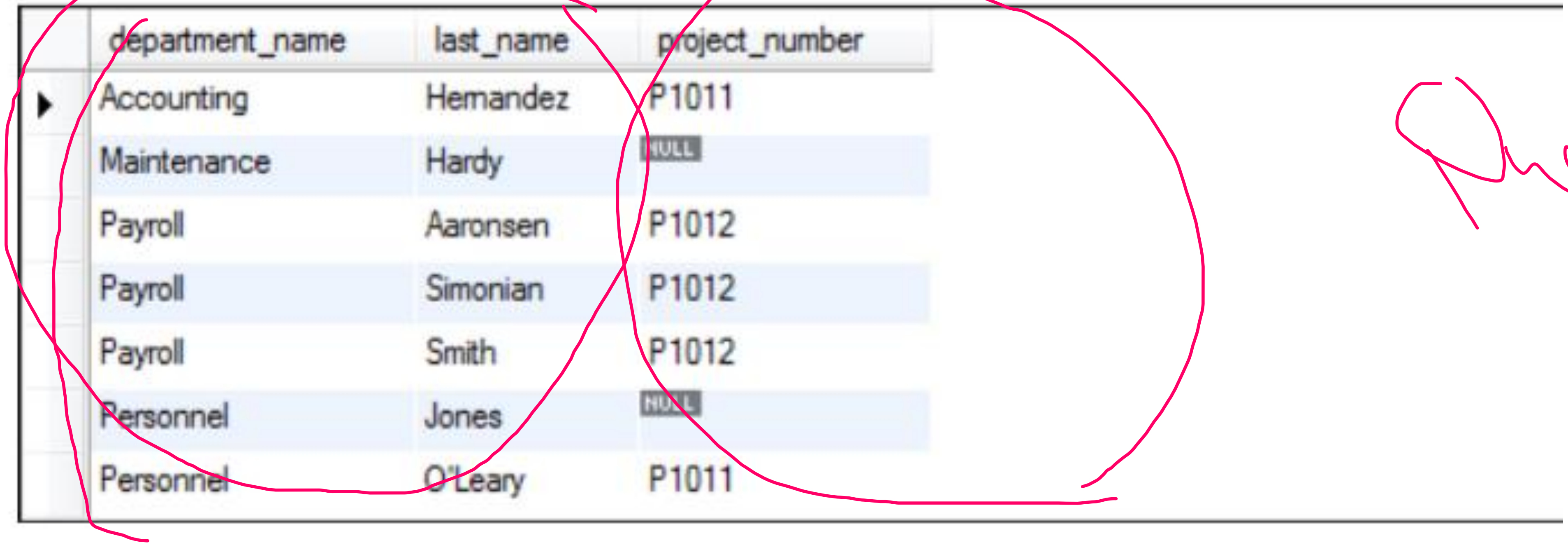
```
SELECT department_name, last_name, project_number
FROM departments d
    LEFT JOIN employees e
        ON d.department_number = e.department_number
    LEFT JOIN projects p
        ON e.employee_id = p.employee_id
ORDER BY department_name, last_name
```



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

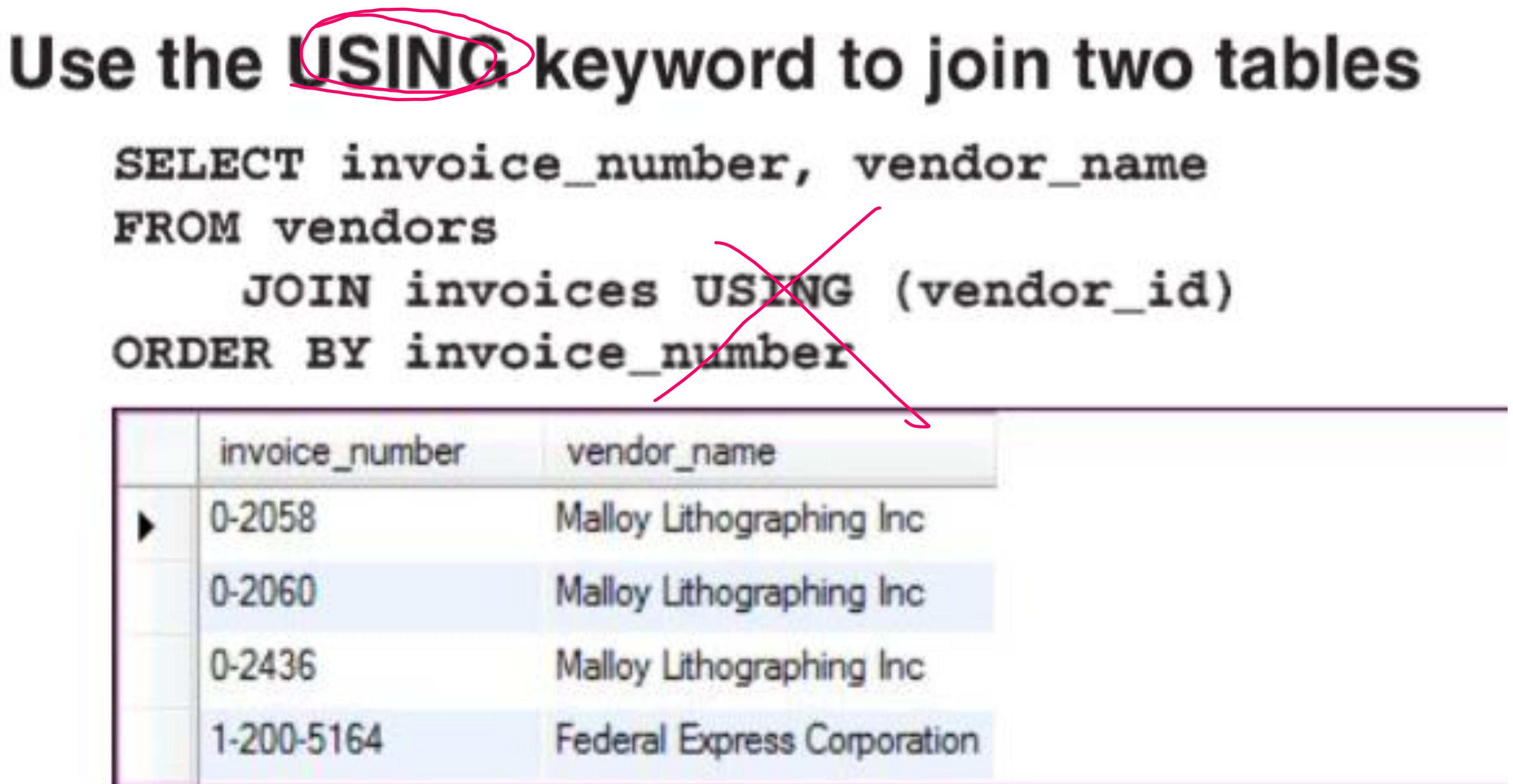
Combine an outer and an inner join

```
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

on ใช้กับชื่อต่างกันได้ PK&FK แต่ using ต้องเหมือนกัน



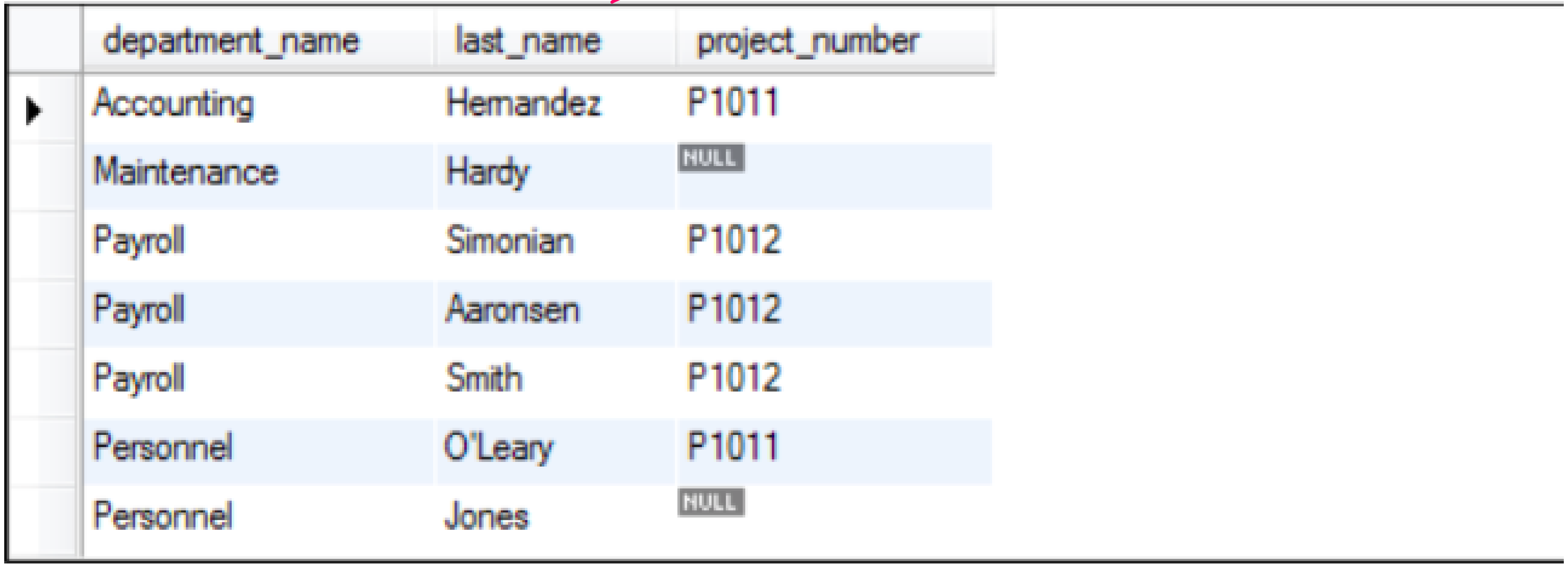
Use the USING keyword to join two tables

```
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

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	Aaronsen	P1012
Payroll	Smith	P1012
Personnel	O'Leary	P1011
Personnel	Jones	NULL