

SQL JOIN 類型詳解

JOIN 操作是關聯式資料庫中連接兩個或多個表格的重要操作。讓我為您詳細解釋各種 JOIN 類型及其應用。

假設資料表

以使用兩個簡單的資料表作為示例：

Employees 表

employee_id	name	email	phone	hire_date	dept_id
1	黃經理	manager.huang@3cstore.com	0912123123	2020-04-18 21:12:12.310	1
2	吳專員	wu@3cstore.com	0923234234	2022-04-18 21:12:12.310	2
3	趙銷售	chao@3cstore.com	0934345345	2025-04-18 21:12:12.310	2
4	潘大寶	troie@earliii.com	0938123456	2025-04-23 14:39:46.550	3
5	哈利波特	harry@gmail.com	0988123456	2025-04-23 15:31:02.720	NULL

(5 rows affected)

Departments 表

dept_id	dept_name
1	行銷
2	業務
3	開發

INNER JOIN (內部連接)

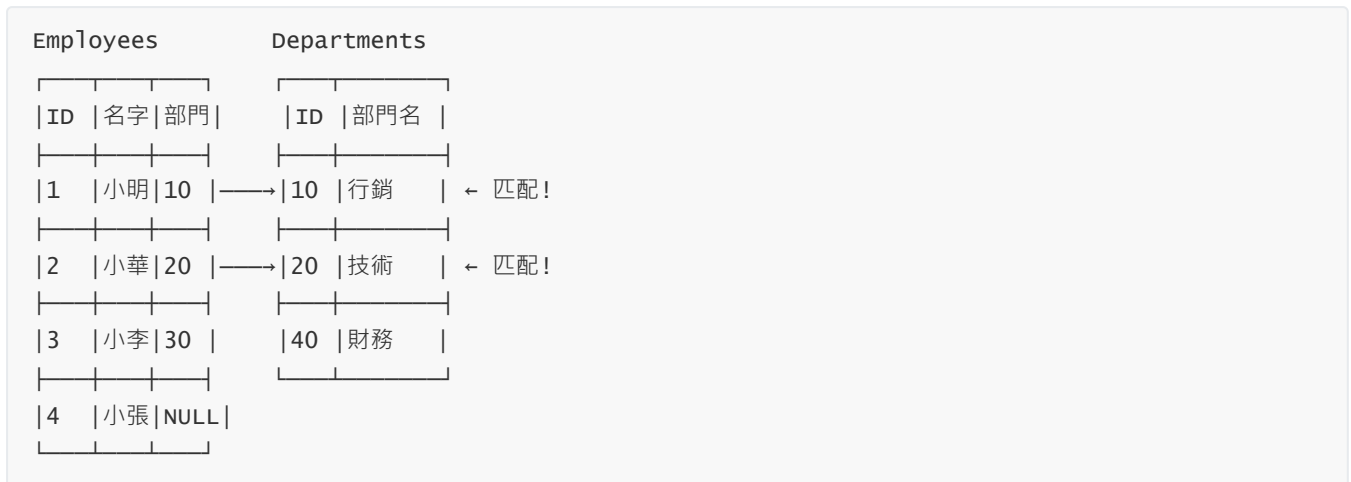
只返回兩個表中能夠匹配的資料。

```
SELECT e.name, d.dept_name
FROM Employees e
INNER JOIN Departments d ON e.dept_id = d.dept_id;
```

結果:

name	dept_name
黃經理	行銷
吳專員	業務
趙銷售	業務
潘大寶	開發

視覺化說明:



LEFT JOIN (左外部連接)

返回左表中的所有資料，以及右表中匹配的資料。若右表無匹配，則結果中對應列為 NULL。

```
SELECT e.name, d.dept_name
FROM Employees e
LEFT JOIN Departments d ON e.dept_id = d.dept_id;
```

結果:

name	dept_name
黃經理	行銷
吳專員	業務
趙銷售	業務
潘大寶	開發
哈利波特	NULL

視覺化說明:

Employees		Departments																									
<table><tr><th>ID</th><th>名字</th><th>部門</th></tr><tr><td>1</td><td>小明</td><td>10</td></tr><tr><td>2</td><td>小華</td><td>20</td></tr><tr><td>3</td><td>小李</td><td>30</td></tr><tr><td>4</td><td>小張</td><td>NULL</td></tr></table>	ID	名字	部門	1	小明	10	2	小華	20	3	小李	30	4	小張	NULL	<div>→</div>	<table><tr><th>ID</th><th>部門名</th></tr><tr><td>10</td><td>行銷</td></tr><tr><td>20</td><td>技術</td></tr><tr><td></td><td>(NULL)</td></tr><tr><td></td><td>(NULL)</td></tr></table>	ID	部門名	10	行銷	20	技術		(NULL)		(NULL)
ID	名字	部門																									
1	小明	10																									
2	小華	20																									
3	小李	30																									
4	小張	NULL																									
ID	部門名																										
10	行銷																										
20	技術																										
	(NULL)																										
	(NULL)																										
		← 匹配!																									
		← 匹配!																									
		← 無匹配																									
		← 無匹配																									

RIGHT JOIN (右外部連接)

返回右表中的所有資料，以及左表中匹配的資料。若左表無匹配，則結果中對應列為 NULL。

```
SELECT e.name, d.dept_name
FROM Employees e
RIGHT JOIN Departments d ON e.dept_id = d.dept_id;
```

結果:

name	dept_name
黃經理	行銷
吳專員	業務
趙銷售	業務
潘大寶	開發
NULL	公關

視覺化說明:

Employees		Departments																							
<table><tr><th>ID</th><th>名字</th><th>部門</th></tr><tr><td>1</td><td>小明</td><td>10</td></tr><tr><td>2</td><td>小華</td><td>20</td></tr><tr><td>3</td><td>小李</td><td>30</td></tr><tr><td>4</td><td>小張</td><td>NULL</td></tr></table>	ID	名字	部門	1	小明	10	2	小華	20	3	小李	30	4	小張	NULL	<div>→</div>	<table><tr><th>ID</th><th>部門名</th></tr><tr><td>10</td><td>行銷</td></tr><tr><td>20</td><td>技術</td></tr><tr><td>40</td><td>財務</td></tr></table>	ID	部門名	10	行銷	20	技術	40	財務
ID	名字	部門																							
1	小明	10																							
2	小華	20																							
3	小李	30																							
4	小張	NULL																							
ID	部門名																								
10	行銷																								
20	技術																								
40	財務																								
		← 匹配!																							
		← 匹配!																							
		← 無匹配																							

FULL JOIN (全外部連接)

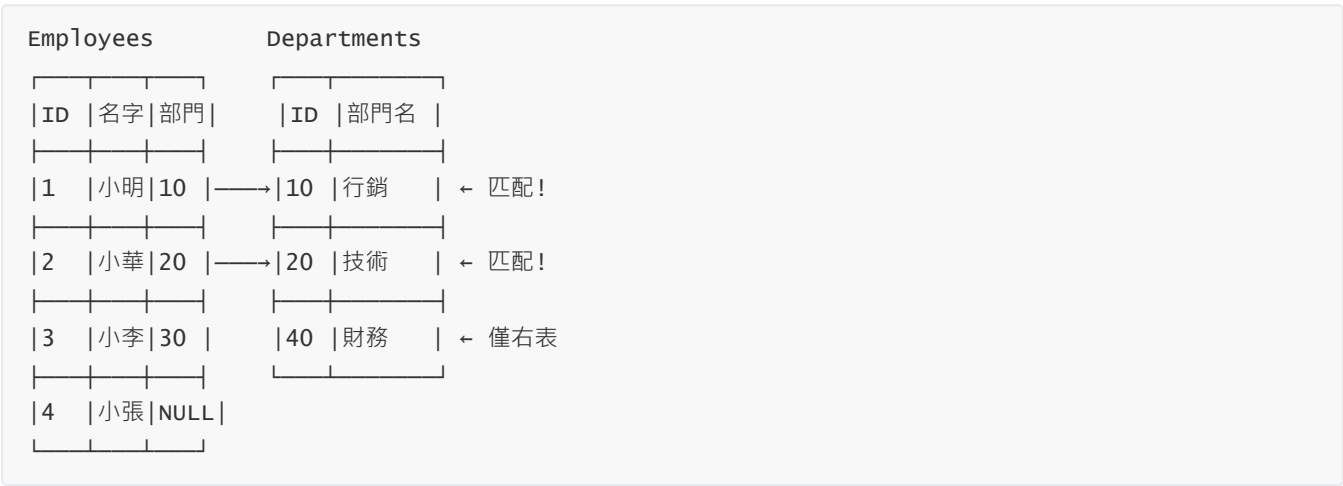
返回兩個表中所有的資料。若某一方無匹配，則結果中對應列為 NULL。

```
SELECT e.name, d.dept_name
FROM Employees e
FULL JOIN Departments d ON e.dept_id = d.dept_id;
```

結果:

name	dept_name
黃經理	行銷
吳專員	業務
趙銷售	業務
潘大寶	開發
哈利波特	NULL
NULL	公關

視覺化說明:



CROSS JOIN (交叉連接)

將左表的每一行與右表的每一行進行組合，形成笛卡爾積。

```
SELECT e.name, d.dept_name
FROM Employees e
CROSS JOIN Departments d;
```

結果:

name	dept_name
潘大寶	業務
哈利波特	業務
黃經理	開發
吳專員	開發
趙銷售	開發
潘大寶	開發
哈利波特	開發
黃經理	公關
吳專員	公關
趙銷售	公關
潘大寶	公關
哈利波特	公關

視覺化說明:

Employees	x	Departments	=	結果 (4x3=12 筆資料)																																																	
<table><tr><td>ID</td><td>名字</td><td>部門</td></tr><tr><td>1</td><td>小明</td><td>10</td></tr><tr><td>2</td><td>小華</td><td>20</td></tr><tr><td>3</td><td>小李</td><td>30</td></tr><tr><td>4</td><td>小張</td><td>NULL</td></tr></table>	ID	名字	部門	1	小明	10	2	小華	20	3	小李	30	4	小張	NULL		<table><tr><td>ID</td><td>部門名</td></tr><tr><td>10</td><td>行銷</td></tr><tr><td>20</td><td>技術</td></tr><tr><td>40</td><td>財務</td></tr></table>	ID	部門名	10	行銷	20	技術	40	財務		<table><tr><td>名字</td><td>部門名</td></tr><tr><td>小明</td><td>行銷</td></tr><tr><td>小明</td><td>技術</td></tr><tr><td>小明</td><td>財務</td></tr><tr><td>小華</td><td>行銷</td></tr><tr><td>小華</td><td>技術</td></tr><tr><td>小華</td><td>財務</td></tr><tr><td>小李</td><td>行銷</td></tr><tr><td>小李</td><td>技術</td></tr><tr><td>小李</td><td>財務</td></tr><tr><td>小張</td><td>行銷</td></tr><tr><td>小張</td><td>技術</td></tr><tr><td>小張</td><td>財務</td></tr></table>	名字	部門名	小明	行銷	小明	技術	小明	財務	小華	行銷	小華	技術	小華	財務	小李	行銷	小李	技術	小李	財務	小張	行銷	小張	技術	小張	財務
ID	名字	部門																																																			
1	小明	10																																																			
2	小華	20																																																			
3	小李	30																																																			
4	小張	NULL																																																			
ID	部門名																																																				
10	行銷																																																				
20	技術																																																				
40	財務																																																				
名字	部門名																																																				
小明	行銷																																																				
小明	技術																																																				
小明	財務																																																				
小華	行銷																																																				
小華	技術																																																				
小華	財務																																																				
小李	行銷																																																				
小李	技術																																																				
小李	財務																																																				
小張	行銷																																																				
小張	技術																																																				
小張	財務																																																				

每種 JOIN 類型都有其特定的使用場景，選用合適的 JOIN 類型可以讓您獲取需要的資料，同時保持查詢效率。