

# 画图解释 SQL join 语句

我认为 Ligaya Turmelle 的关于SQL联合（join）语句的帖子对于新手开发者来说是份很好的材料。SQL 联合语句好像是基于集合的，用韦恩图来解释咋一看是很自然而然的。不过正如在她的帖子的回复中所说的，在测试中我发现韦恩图并不是十分的匹配SQL联合语法。

不过我还是喜欢这个观点，所以我们来看看能不能用上韦恩图。假设我们有下面两张表。表A在左边，表B在右边。我们给它们各四条记录。

id	name	id	name
--	----	--	----

1	id	name	id	name
2	--	----	--	----
3	1	Pirate	1	Rutabaga
4	2	Monkey	2	Pirate
5	3	Ninja	3	Darth Vader
6	4	Spaghetti	4	Ninja

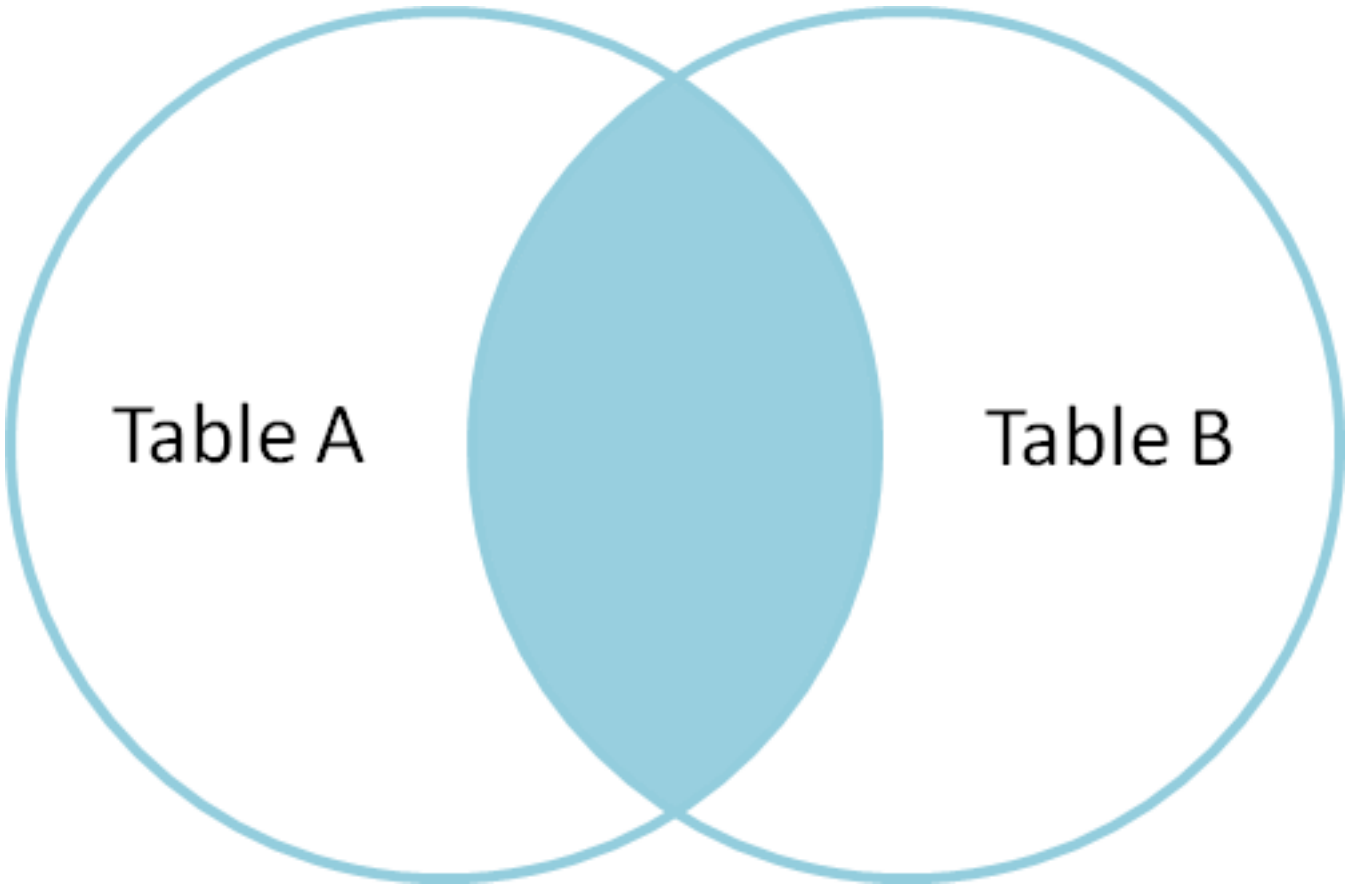
我们用过name字段用几种不同方式把这些表联合起来，看能否得到和那些漂亮的韦恩图在概念上的匹配。

SELECT * FROM TableA INNER JOIN TableB
---

1	SELECT * FROM TableA			
2	INNER JOIN TableB			
3	ON TableA.name = TableB.name			
4	id	name	id	name
5	--	----	--	----
6	1	Pirate	2	Pirate

7	3	Ninja	4	Ninja
8				

内联合（inner join）只生成同时匹配表A和表B的记录集。（如下图）

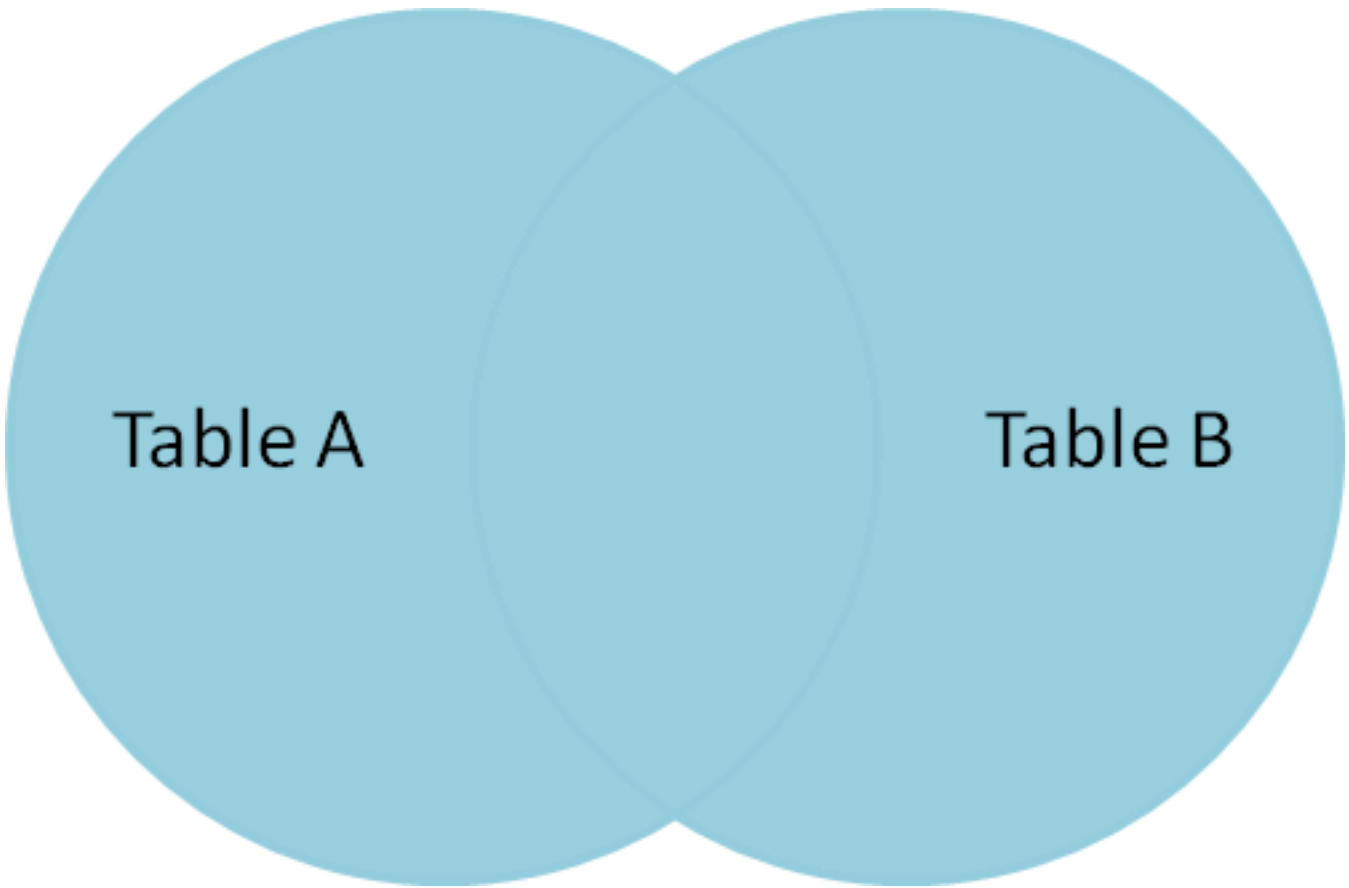


```
SELECT * FROM TableA
FULL OUTER JOIN TableB
```

1	
2	SELECT * FROM TableA
3	FULL OUTER JOIN TableB
4	ON TableA.name = TableB.name
5	id   name   id   name
6	--   ----   --   ----
7	1   Pirate   2   Pirate
8	2   Monkey   null null
9	3   Ninja   4   Ninja

10	4	Spaghetti	null	null
11	null	null	1	Rutabaga
12	null	null	3	Darth Vader

全外联合（full outer join）生成表A和表B里的记录全集，包括两边都匹配的记录。如果有一边没有匹配的，缺失的这一边为null。（如下图）

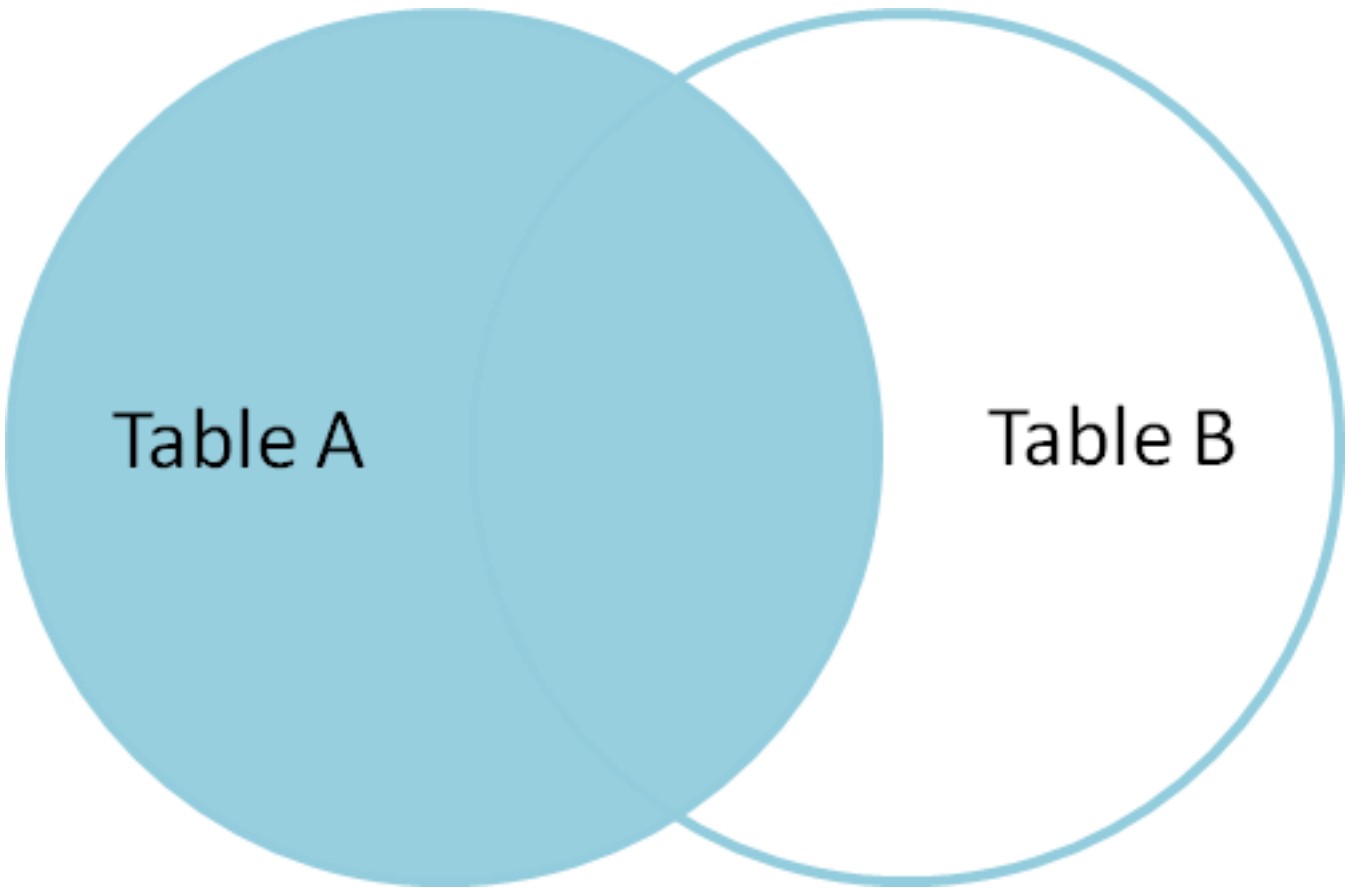


```
SELECT * FROM TableA
LEFT OUTER JOIN TableB
```

1	SELECT * FROM TableA			
2	LEFT OUTER JOIN TableB			
3	ON TableA.name = TableB.name			
4	id	name	id	name
5	--	----	--	----
6	1	Pirate	2	Pirate
7	2	Monkey	null	null

8	3	Ninja	4	Ninja
9	4	Spaghetti	null	null
10				

左外联合（left outer join）生成表A的所有记录，包括在表B里匹配的记录。如果没有匹配的，右边将是null。（如下图）



```
SELECT * FROM TableA
LEFT OUTER JOIN TableB
```

1	SELECT * FROM TableA
2	LEFT OUTER JOIN TableB
3	ON TableA.name = TableB.name
4	WHERE TableB.id IS null
5	id name id name
6	-- ---- -- ----
7	2 Monkey null null
8	

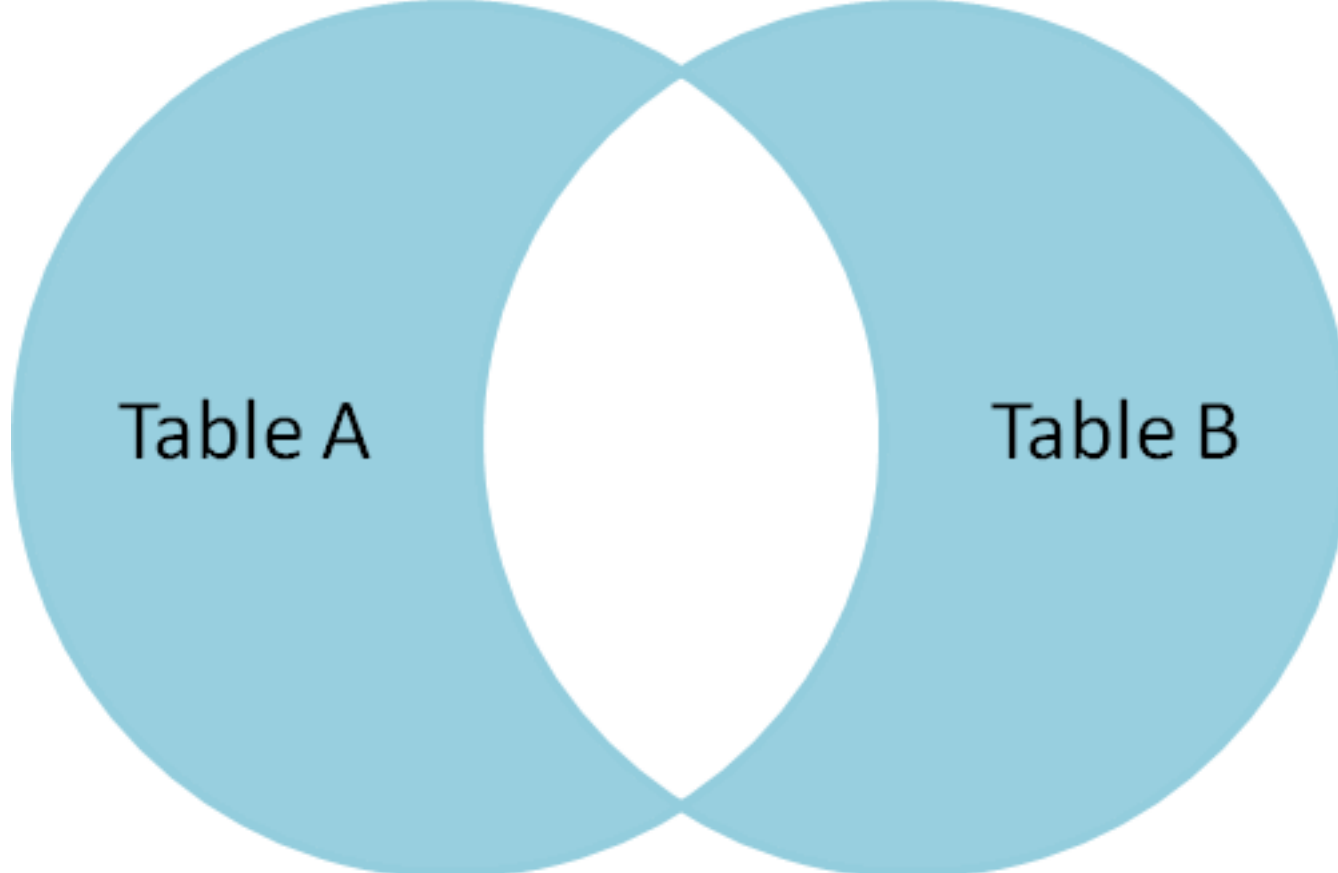
9	4	Spaghetti	null	null
---	---	-----------	------	------

为了生成只在表A里而不在表B里的记录集，我们用同样的左外联合，然后用where语句排除我们不想要的记录。（如下图）

```
SELECT * FROM TableA
FULL OUTER JOIN TableB
```

1	SELECT * FROM TableA			
2	FULL OUTER JOIN TableB			
3	ON TableA.name = TableB.name			
4	WHERE TableA.id IS null			
5	OR TableB.id IS null			
6	id	name	id	name
7	--	----	--	----
8	2	Monkey	null	null
9	4	Spaghetti	null	null
10	null	null	1	Rutabaga
11	null	null	3	Darth Vader
12				

为了生成对于表A和表B唯一的记录集，我们用同样的全外联合，然后用where语句排除两边都不想要的记录。（如下图）



---

还有一种笛卡尔积或者交叉联合（**cross join**），据我所知不能用韦恩图表示：

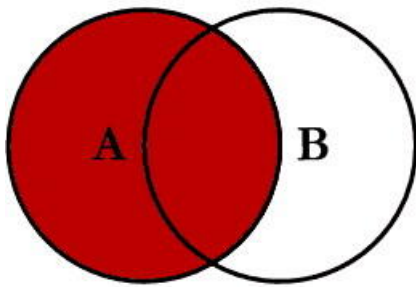
```
SELECT * FROM TableA  
CROSS JOIN TableB
```

1	SELECT * FROM TableA
2	CROSS JOIN TableB

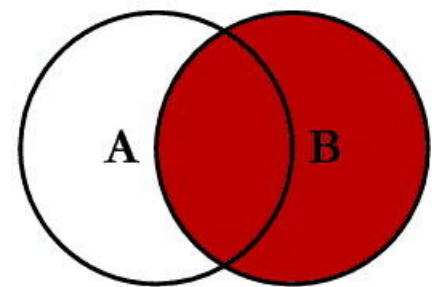
这个把“所有”联接到“所有”，产生 $4 \times 4 = 16$ 行，远多于原始的集合。如果你学过数学，你便知道为什么这个联合遇上大型的表很危险。

【2013-06-17 更新】 下图由 Moffatt 在 2008 年制作（[点击可查看大图](#)）。  
PS：Jeff Atwood 的文章写于 2007 年。

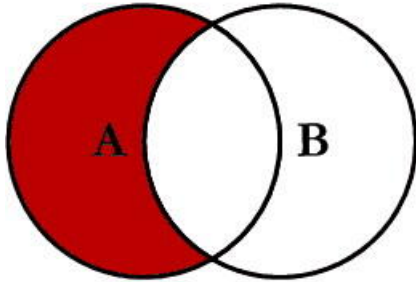
# SQL JOINS



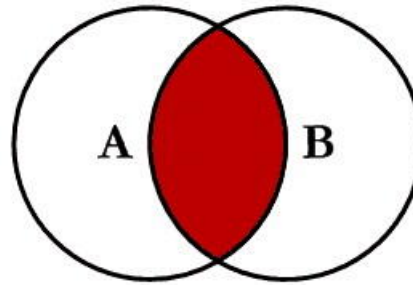
```
SELECT <select_list>  
FROM TableA A  
LEFT JOIN TableB B  
ON A.Key = B.Key
```



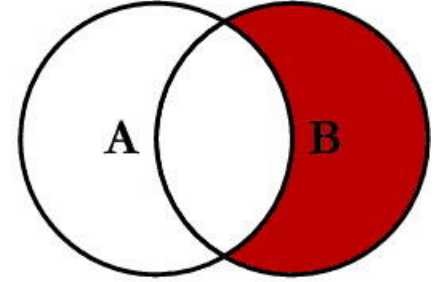
```
SELECT <select_list>  
FROM TableA A  
RIGHT JOIN TableB B  
ON A.Key = B.Key
```



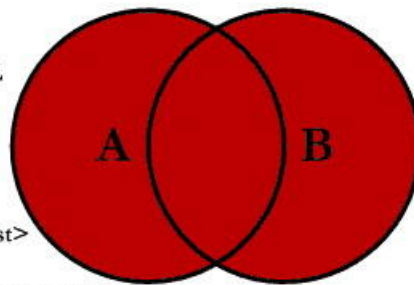
```
SELECT <select_list>  
FROM TableA A  
LEFT JOIN TableB B  
ON A.Key = B.Key  
WHERE B.Key IS NULL
```



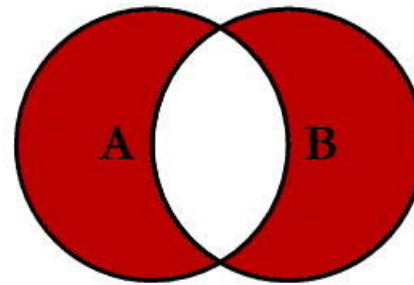
```
SELECT <select_list>  
FROM TableA A  
INNER JOIN TableB B  
ON A.Key = B.Key
```



```
SELECT <select_list>  
FROM TableA A  
RIGHT JOIN TableB B  
ON A.Key = B.Key  
WHERE A.Key IS NULL
```



```
SELECT <select_list>  
FROM TableA A  
FULL OUTER JOIN TableB B  
ON A.Key = B.Key
```



```
SELECT <select_list>  
FROM TableA A  
FULL OUTER JOIN TableB B  
ON A.Key = B.Key  
WHERE A.Key IS NULL  
OR B.Key IS NULL
```

© C.L. Moffatt, 2008

1 赞 43 收藏 10 评论