

# SQL连接查询

通过连接运算符可以实现多个表查询。连接是关系数据库模型的主要特点，是它区别于其它类型数据库管理系统的一个标志。

连接可以在SELECT 语句的FROM子句或WHERE子句中建立，在FROM子句中指出连接时有助于将连接操作与WHERE子句中的搜索条件区分开来。

# 交叉连接（cross join）：

交叉连接返回左表中的所有行，左表中的每一行与右表中的所有行组合。交叉连接也称作笛卡尔积，因为它返回的是两表的乘积。

例如：下面的语句1和语句2的结果是相同的。

语句1：隐式的交叉连接，没有显示的声明cross join。

```
select * from table1, table2;
```

语句2：显式的交叉连接，使用cross join。

```
select * from table1 cross join table2;
```

实际应用中，Cross join一般用得比较少



举个例子：

```
mysql> select * from table1;
+-----+-----+
| sid  | bid  |
+-----+-----+
|    1 |   98 |
|    2 |   87 |
+-----+-----+
2 rows in set (0.00 sec)
```

```
mysql> select * from table2;
+-----+-----+
| sid  | sname |
+-----+-----+
|    1 | Mike  |
|    2 | James |
+-----+-----+
2 rows in set (0.00 sec)
```

```
mysql> select a.sid,a.bid,b.sid,b.sname
-> from table1 as a cross join table2 as b;
```

sid	bid	sid	sname
1	98	1	Mike
2	87	1	Mike
1	98	2	James
2	87	2	James

```
4 rows in set (0.00 sec)
```

# 内连接（inner join）：

内连接返回链接表中符合连接条件和查询条件的数据行。（所谓的链接表就是数据库在做查询形成的中间表）。INNER JOIN 产生的结果是AB的交集

现有两张表，Table A 是左边的表。Table B 是右边的表。其各有四条记录，其中有两条记录name是相同的：

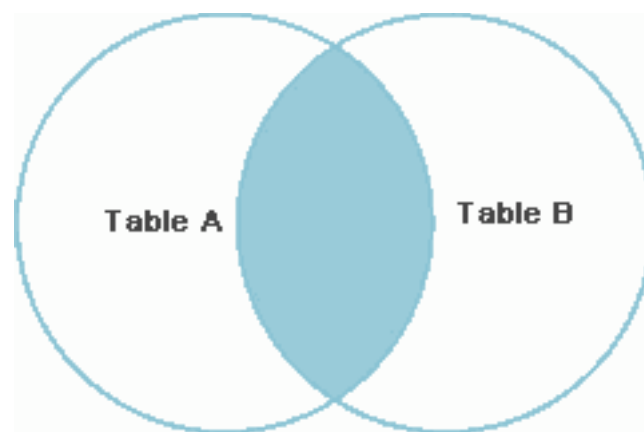
A表		B表	
id	name	id	name
1	Pirate	1	Rutabaga
2	Monkey	2	Pirate
3	Ninja	3	Darth Vade
4	Spaghetti	4	Ninja

# 内连接 (inner join) :

INNER JOIN 产生的结果是AB的交集

```
SELECT * FROM TableA INNER JOIN TableB ON  
TableA.name = TableB.name
```

结果集			
(TableA.)		(TableB.)	
id	name	id	name
1	Pirate	2	Pirate
3	Ninja	4	Ninja




# 外连接（outer join）：

外连接不但返回符合连接和查询条件的数据行，还返回不符合条件的一些行。

外连接分三类：左外连接（left outer join）、右外连接（right outer join）和全外连接（full outer join）。其中，outer关键字可省略。

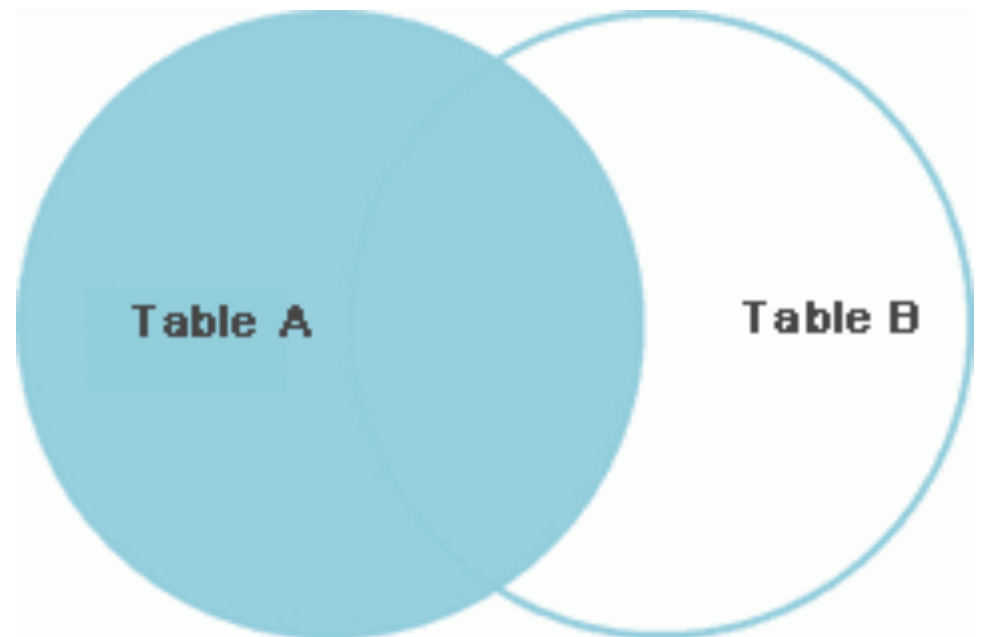
三者的共同点是都返回符合连接条件和查询条件（即：内连接）的数据行



left join 或 left outer join:

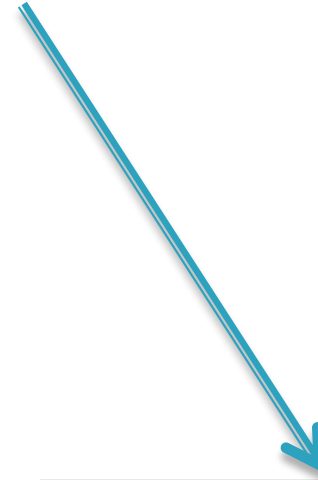
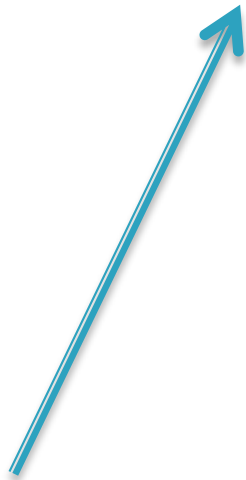
LEFT [OUTER] JOIN 产生表A的完全集，而B表中匹配的则有值，没有匹配的则以null值取代。

```
SELECT * FROM TableA LEFT OUTER JOIN TableB ON  
TableA.name = TableB.name
```



left join 或 left outer join:

SELECT \* FROM TableA LEFT OUTER JOIN TableB ON  
TableA.name = TableB.name



A表		B表	
id	name	id	name
1	Pirate	1	Rutabaga
2	Monkey	2	Pirate
3	Ninja	3	Darth Vade
4	Spaghetti	4	Ninja

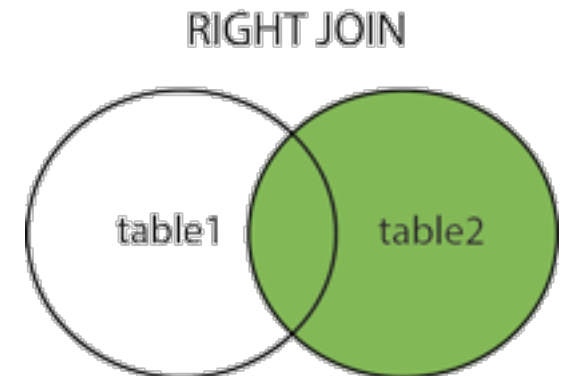
结果集			
(TableA.)		(TableB.)	
id	name	id	name
1	Pirate	2	Pirate
2	Monkey	null	null
3	Ninja	4	Ninja
4	Spaghetti	null	null



right join 或 right outer join:

RIGHT [OUTER] JOIN 产生表B的完全集，而A表中匹配的则有值，没有匹配的则以null值取代。

```
SELECT * FROM TableA RIGHT OUTER JOIN TableB ON  
TableA.name = TableB.name
```



思考：大家可以想想right join 和 left join的异同

# full join 或 full outer join:

FULL OUTER JOIN 关键字只要左表（table1）和右表（table2）其中一个表中存在匹配，则返回行。

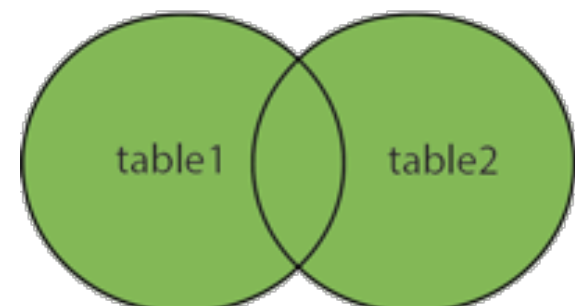
FULL OUTER JOIN 关键字结合了 LEFT JOIN 和 RIGHT JOIN 的结果。

FULL [OUTER] JOIN 产生A和B的并集。对于没有匹配的记录，则会以null做为值。

```
SELECT * FROM TableA FULL OUTER JOIN TableB ON TableA.name =  
TableB.name
```

然而mysql是不支持full join的，大家稍微了解一下就好


FULL OUTER JOIN



# union:

UNION 操作符用于合并两个或多个 SELECT 语句的结果集。  
请注意，UNION 内部的 SELECT 语句必须拥有相同数量的列。列也必须拥有相似的数据类型。同时，每条 SELECT 语句中的列的顺序必须相同。

```
select
    column_name(s) from table_name1
union
select
    column_name(s) from table_name2
```



# union:

举个例子，有如下两张表

sid	bid
1	98
2	87
3	100

sid	sname
1	Mike
2	James
4	kim

经过union操作之后得到的结果如下所示：

```
mysql> select sid from table1 union select sid from table2;
```

sid
1
2
3
4

假如需要对3个表进行拼接，则先对其中两个表先拼接，以inner join为例举个例子：

语法格式可以概括为：

```
SELECT * FROM
```

```
(表1 INNER JOIN 表2 ON 表1.字段号=表2.字段号)
```

```
INNER JOIN 表3 ON 表1.字段号=表3.字段号
```

拼接3个或3个以上的表可以以此类推，若对这部分有疑问，请百度或者谷歌自行探索一下，本次作业部分需要对3个表进行拼接操作的。

大家可以自己动手在cmd里敲一下这节课讲过的mysql命令了，自己动手敲代码会有更深刻的理解，敲代码然后自己想想总结一下这几种连接操作的异同。

然后大家把这周的实验课的作业给做了，  
详情请查看 实验五.pdf

一定要自己独立自主完成，遇到问题请自己百度谷歌进行搜索或者互相讨论来解决，作为大三的学生，应该要有自己解决问题的能力了，这点是很重要的，希望你们能够提高problem-solving和teamwork的能力。

Enjoy the lab!



谢谢