/\*INNER JOIN\*/

/\*1\*/SELECT id\_name.id, id\_name.name FROM id\_name INNER JOIN id\_name\_null ON id\_name.id = id\_name\_null.id;

/\*2\*/SELECT id\_name.id AS a\_id, id\_name.name AS a\_name,id\_name\_null.id AS b\_id,id\_name\_null.name AS b\_name FROM id\_name INNER JOIN id\_name\_null ON id\_name.id = id\_name\_null.id;

/\*3\*/SELECT DISTINCT id\_name.id AS a\_id, id\_name.name AS a\_name,id\_name\_null.id AS b\_id,id\_name\_null.name AS b\_name FROM id\_name INNER JOIN id\_name\_null ON id\_name.id = id\_name\_null.id;

/\*\*LEFT OUTER JOIN == LEFT JOIN \*/

/\*4\*/SELECT id\_name.id, id\_name.name FROM id\_name LEFT OUTER JOIN id\_name\_null ON id\_name.id = id\_name\_null.id;

/\*\*RIGHT OUTER JOIN == RIGHT JOIN \*/

* SELECT id\_name.id, id\_name.name FROM id\_name RIGHT OUTER JOIN id\_name\_null ON id\_name.id = id\_name\_null.id;

/\*5\*/SELECT id\_name.id AS LOa\_id, id\_name.name AS LOa\_name,id\_name\_null.id AS LOb\_id,id\_name\_null.name AS LOb\_name FROM id\_name LEFT OUTER JOIN id\_name\_null ON id\_name.id = id\_name\_null.id;

/\*6\*/SELECT DISTINCT id\_name.id AS LOa\_id, id\_name.name AS LOa\_name,id\_name\_null.id AS LOb\_id,id\_name\_null.name AS LOb\_name FROM id\_name LEFT OUTER JOIN id\_name\_null ON id\_name.id = id\_name\_null.id;

* SELECT id\_name.id AS ROa\_id, id\_name.name AS ROa\_name,id\_name\_null.id AS ROb\_id,id\_name\_null.name AS ROb\_name FROM id\_name RIGHT OUTER JOIN id\_name\_null ON id\_name.id = id\_name\_null.id;

/\*7\*/SELECT id\_name\_null.id, id\_name\_null.name FROM id\_name\_null LEFT OUTER JOIN id\_name ON id\_name\_null.id = id\_name.id;

* SELECT id\_name\_null.id, id\_name\_null.name FROM id\_name\_null RIGHT OUTER JOIN id\_name ON id\_name\_null.id = id\_name.id;

/\*8\*/SELECT id\_name\_null.id AS LOb\_id, id\_name\_null.name AS LOb\_name,id\_name.id AS LOa\_id,id\_name.name AS LOa\_name FROM id\_name\_null LEFT OUTER JOIN id\_name ON id\_name.id = id\_name\_null.id;

* SELECT id\_name\_null.id AS ROb\_id, id\_name\_null.name AS ROb\_name,id\_name.id AS ROa\_id,id\_name.name AS ROa\_name FROM id\_name\_null RIGHT OUTER JOIN id\_name ON id\_name.id = id\_name\_null.id;

**/\*LEFT(RIGHT) OUTER JOIN 或者LEFT(RIGHT) JOIN ，when one table（table\_null） is null \*/**

1、SELECT id\_name.id , id\_name.name FROM id\_name LEFT OUTER JOIN table\_null ON id\_name.id = table\_null.id;

* SELECT id\_name.id , id\_name.name FROM id\_name RIGHT OUTER JOIN table\_null ON id\_name.id = table\_null.id;

2、 SELECT id\_name\_null.id , id\_name\_null.name FROM id\_name\_null LEFT OUTER JOIN table\_null ON id\_name\_null.id = table\_null.id;

* SELECT table\_null.id , table\_null.name FROM id\_name\_null RIGHT OUTER JOIN table\_null ON id\_name\_null.id = table\_null.id;

3、**/\*group by Table：** id\_name **\*/ ------>Group By本身就有Distinct的作用。**

SELECT id,name FROM id\_name GROUP BY id,name;

SELECT id,COUNT(name) FROM id\_name GROUP BY id,name;

SELECT id,COUNT(name) AS name\_cnt FROM id\_name GROUP BY id;

**/\*group by Table：** id\_name\_null **\*/**

SELECT id,name FROM id\_name\_null GROUP BY id;

**/\*group by Table：** table\_null **\*/**

SELECT id,name FROM table\_null GROUP BY id,name;

4、**/\* group by & order by ---对比如下：\*/**

SELECT id,name FROM id\_name GROUP BY id,name;

SELECT id,name FROM id\_name GROUP BY id,name ORDER BY id DESC;

5、

Having与Where的区别：

where 子句的作用是在对查询结果进行分组前，将不符合where条件的行去掉，即在分组之前过滤数据，where条件中不能包含聚组函数，使用where条件过滤出特定的行。

having 子句的作用是筛选满足条件的组，即在分组之后过滤数据，条件中经常包含聚组函数，使用having 条件过滤出特定的组，也可以使用多个分组标准进行分组。

**/\* Where、 group by、having：**id\_name **\*/**

SELECT id,name FROM id\_name WHERE id > 1;

SELECT id,name FROM id\_name WHERE id > 1 GROUP BY id,name;

SELECT id,name FROM id\_name GROUP BY id,name HAVING SUM(id) > 2;

SELECT id,name FROM id\_name WHERE id >4 GROUP BY id,name HAVING SUM(id) > 2;

1. **/\* group by、having：**id\_name **&** id\_name\_score **对比如下：\*/**

SELECT id,name FROM id\_name GROUP BY id,name HAVING SUM(id) > 2;

* SELECT id,name,score FROM id\_name\_score GROUP BY id,name HAVING SUM(id) > 2;
* SELECT id,name,score FROM id\_name\_score GROUP BY id,name HAVING SUM(score) > 15;

1. **/\* Where、 group by、having、 order by：**id\_name **&** id\_name\_score **对比如下：\*/**

SELECT id,name FROM id\_name WHERE id >3 GROUP BY id,name HAVING SUM(id) > 2 ORDER BY name;

* SELECT id,name FROM id\_name WHERE id >3 GROUP BY id,name HAVING SUM(id) > 2 ORDER BY name DESC;
* SELECT id,name,score FROM id\_name\_score WHERE id >1 GROUP BY id,name HAVING SUM(score) > 10 ORDER BY name DESC;
* SELECT id,name,score FROM id\_name\_score WHERE id >1 GROUP BY id,name,score HAVING SUM(score) > 10 ORDER BY name DESC;
* SELECT id,name,score FROM id\_name\_score WHERE id <10 GROUP BY id,name,score HAVING SUM(score) > 10 ORDER BY score;
* SELECT id,name,score FROM id\_name\_score WHERE id <10 GROUP BY id,name HAVING SUM(score) > 10 ORDER BY score;
* SELECT id,name,score FROM id\_name\_score WHERE id <10 GROUP BY id,name HAVING SUM(score) > 10 ORDER BY name;
* SELECT id,name,score FROM id\_name\_score WHERE id <10 GROUP BY id,name,score HAVING SUM(score) > 10 ORDER BY name;