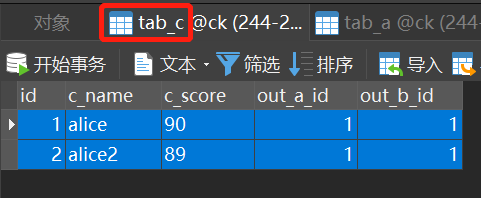
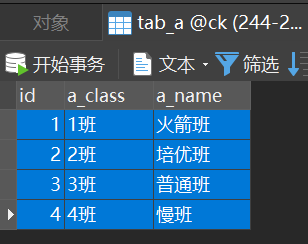
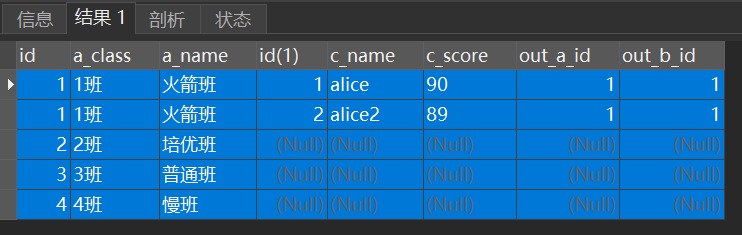
================================ LEFT JOIN =====================================

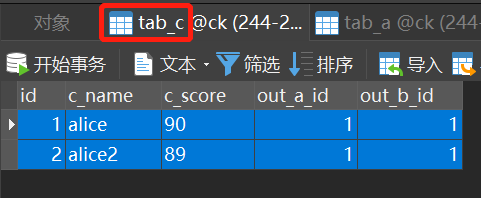
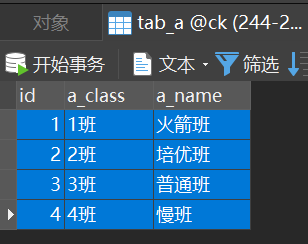
{{{注意：JOIN并不保证联合后的表的数据的顺序，和原先的任意一个表的数据的顺序一致，可参见 特别的JOIN.docx}}}



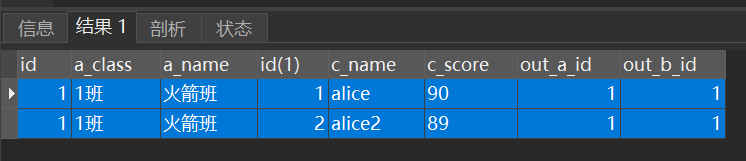
SELECT \* FROM tab\_a LEFT JOIN tab\_c ON tab\_a.id = tab\_c.out\_a\_id;



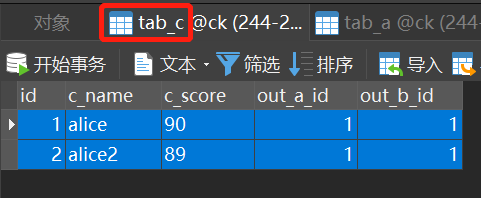
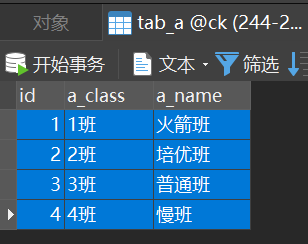
=================================INNER JOIN================================



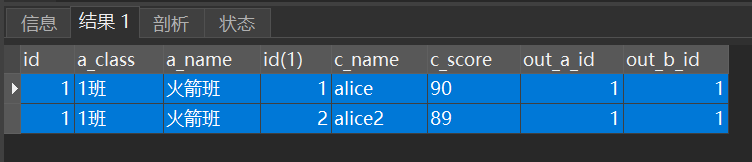
SELECT \* FROM tab\_a JOIN tab\_c ON tab\_a.id = tab\_c.out\_a\_id;



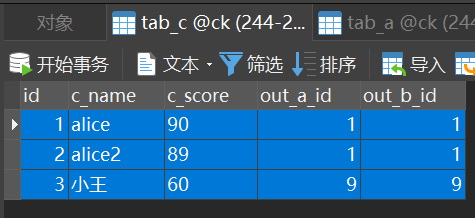
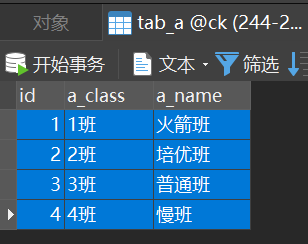
=============================== RIGHT JOIN==================================



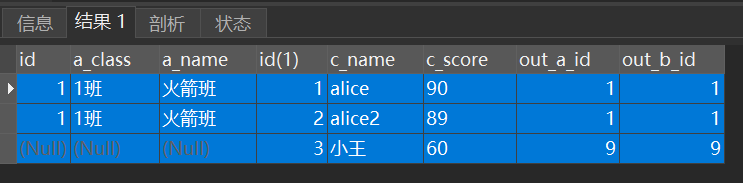
SELECT \* FROM tab\_a RIGHT JOIN tab\_c ON tab\_a.id = tab\_c.out\_a\_id;



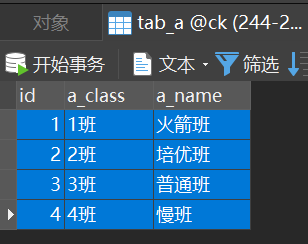
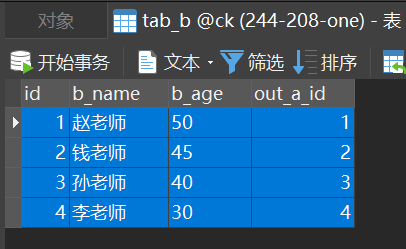
===============================RIGHT JOIN==================================

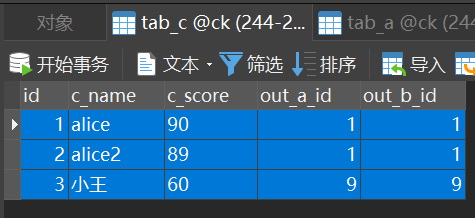


SELECT \* FROM tab\_a RIGHT JOIN tab\_c ON tab\_a.id = tab\_c.out\_a\_id;



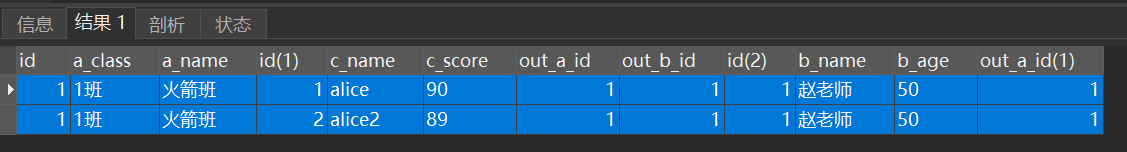
===================================多JOIN===================================



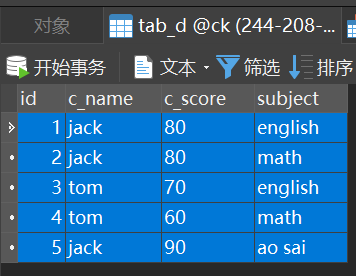
SELECT \* FROM tab\_a JOIN tab\_c ON tab\_a.id = tab\_c.out\_a\_id JOIN tab\_b ON tab\_a.id = tab\_b.out\_a\_id;

【SELECT \* FROM tab\_a JOIN tab\_b ON tab\_a.id = tab\_b.out\_a\_id JOIN tab\_c ON tab\_a.id = tab\_c.out\_a\_id; 换成这样写sql结果并无本质不同】

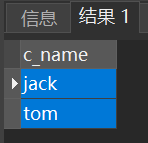


===================================JOIN后分组==================================

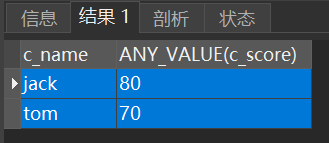
分组----------------------------------------------------------



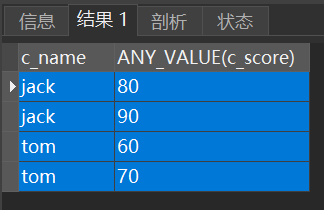
SELECT c\_name FROM tab\_d GROUP BY c\_name;



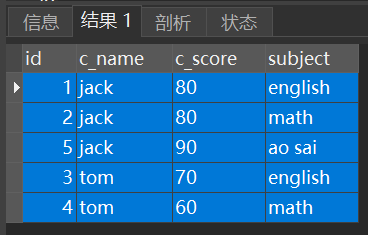
SELECT c\_name, ANY\_VALUE(c\_score) FROM tab\_d GROUP BY c\_name;



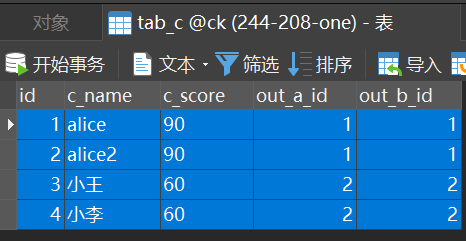
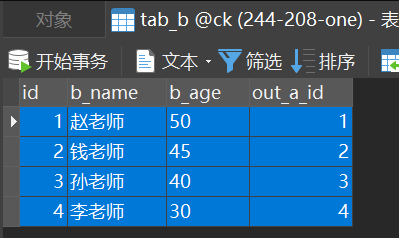
SELECT c\_name, ANY\_VALUE(c\_score) FROM tab\_d GROUP BY c\_name, c\_score;



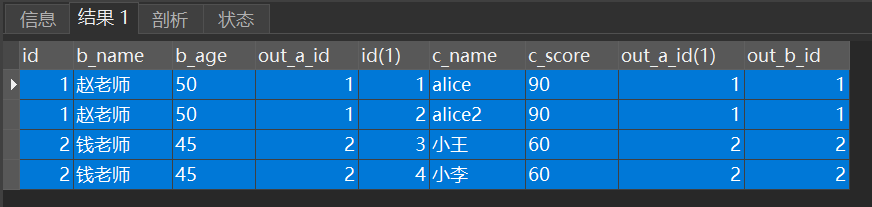
SELECT \* FROM tab\_d GROUP BY c\_name, id; （由于c\_name分组后组内又按id这个不重复列分组，故组内成员都会显示出来，也就无需在每个字段前加any\_value了，故可以直接用 select \*）【GROUP BY 后的c\_name和id顺序不同分组的含义是不同的，后一个是前一个分组后的组内分组依据，SELECT \* FROM tab\_d GROUP BY id, c\_name；的结果是不一样的】



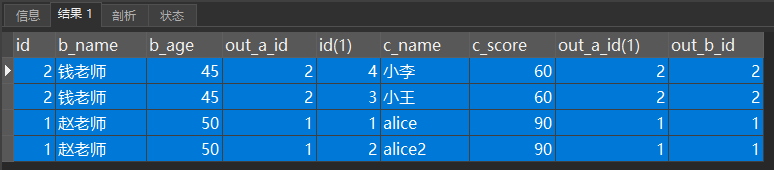
JOIN后的分组--------------------------------------------------------------------



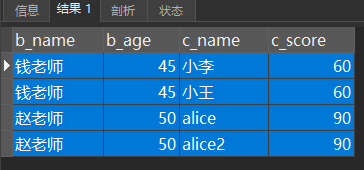
SELECT \* FROM tab\_b JOIN tab\_c ON tab\_b.id = tab\_c.out\_b\_id;



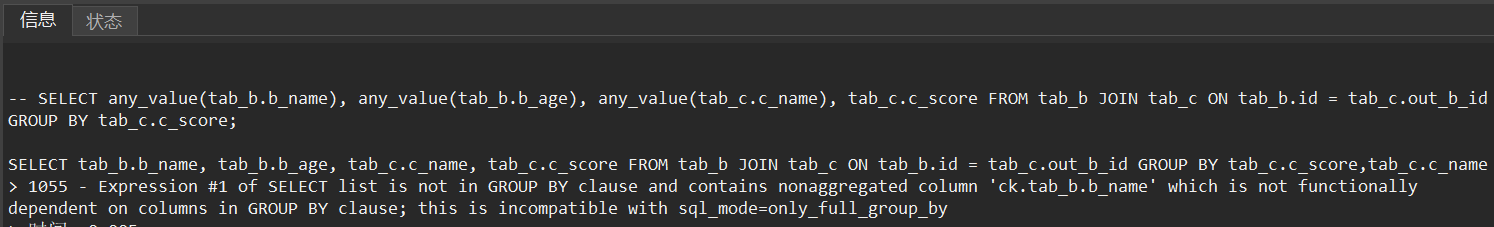
SELECT \* FROM tab\_b JOIN tab\_c ON tab\_b.id = tab\_c.out\_b\_id GROUP BY tab\_c.c\_score,tab\_c.c\_name; 【分组的同时显示组内数据。】



SELECT tab\_b.b\_name, tab\_b.b\_age, tab\_c.c\_name, tab\_c.c\_score FROM tab\_b JOIN tab\_c ON tab\_b.id = tab\_c.out\_b\_id GROUP BY tab\_c.c\_score,tab\_c.c\_name;



如果某些版本的MySQL报下面这样的错

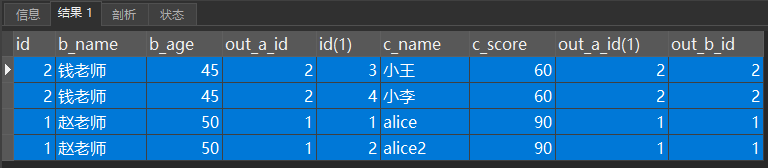


【报的错和SELECT \* 是一样的---------1055 - Expression #1 of SELECT list is not in GROUP BY clause and contains nonaggregated column 'ck.tab\_b.b\_name' which is not functionally dependent on columns in GROUP BY clause; this is incompatible with sql\_mode=only\_full\_group\_by】

那就可以用这样的办法：--------------------------------------------

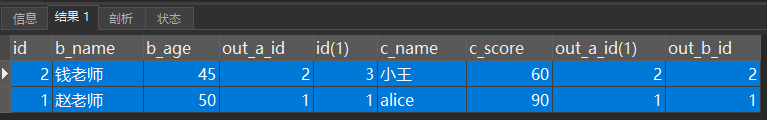
将GROUP BY中两个分组依据的第二个分组依据，即撑开组员而进行的组内分组的依据，改为id这种具备unique key、primary key属性的字段：

SELECT \* FROM tab\_b JOIN tab\_c ON tab\_b.id = tab\_c.out\_b\_id GROUP BY tab\_c.c\_score,tab\_c.id;



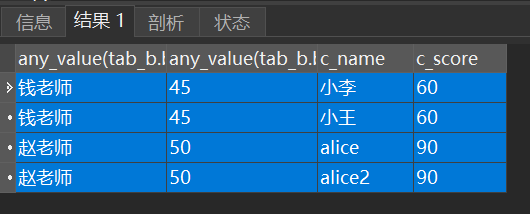
不过如果写成这样，结果有所不同，因为tab\_b表的id在本次被JOIN后的是有重复的。

SELECT \* FROM tab\_b JOIN tab\_c ON tab\_b.id = tab\_c.out\_b\_id GROUP BY tab\_c.c\_score,tab\_b.id;

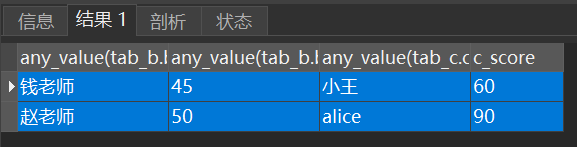


或者---------------（不推荐）---------------

SELECT any\_value(tab\_b.b\_name), any\_value(tab\_b.b\_age), tab\_c.c\_name, tab\_c.c\_score FROM tab\_b JOIN tab\_c ON tab\_b.id = tab\_c.out\_b\_id GROUP BY tab\_c.c\_score,tab\_c.c\_name;



SELECT any\_value(tab\_b.b\_name), any\_value(tab\_b.b\_age), any\_value(tab\_c.c\_name), tab\_c.c\_score FROM tab\_b JOIN tab\_c ON tab\_b.id = tab\_c.out\_b\_id GROUP BY tab\_c.c\_score;



（完）