

SELECT whatever column you want to show up **AS** alias for the column , additional column **AS** additional alias

like: `SELECT first_name AS name, occupation AS job`

each additional field in select clause should be separated by a comma, and no comma at the end

FROM whatever table you're pulling stuff from **AS** alias for the table

JOIN whatever other table you want stuff from **AS** alias for the table

ON (`firsttable/alias.field = secondtable/alias.field` **AND** any other restrictions you want for either of the tables)

like: `ON (table_a.user_id=table_b.user_id AND table_a.lc_id=3 AND table_b.active=1)`

each additional restriction defined in the "ON" clause should be separated by an "AND"

WHERE any restriction based on any field in any of the tables you've drawn from **AND** any other restriction

each additional restriction defined in the "WHERE" clause should be separated by an "AND"

GROUP BY `tablename.field`

if grouping by multiple fields, each additional field in the "GROUP BY" should be separated by a comma

like `GROUP BY table1.state, table1.city`

HAVING any other restriction you want based on any of the calculated fields. So if you only want states with over some numbers members, you can do `"HAVING count(distinct user_id) > 10,000"` (note that this works best if you only group by 1 things, if you group by both state and city, it won't know what's it counting well. Also know that you can use aliases here, so your SELECT clause has `'COUNT(DISTINCT user_id) AS 'members'` , then your having clause can read: `'HAVING 'members'>10,000'`).

each additional restriction defined in the "HAVING" clause should be separated by an "AND"

ORDER BY `tablename.field`

each additional field in the "ORDER BY" should be separated by a comma -- like `"ORDER BY table1.state, table1.city"`

LIMIT the number of rows you want returned (most queries don't limit, but it can be useful sometimes)

Tips:

1) The only two clauses required in every query are "SELECT" and "FROM"

2) If you use other types of clauses (JOIN, WHERE, GROUP BY, etc) they have to appear in the order shown above (unless you're using subqueries)

3) aliases are optional but can be useful. You don't need to put an "AS" there to alias something, but it's cleaner if you do. The only things you can alias are columns in the SELECT clause and tables in the FROM and JOINS