**SQL Syntax Top-Ten Basics**

Too simple to mention? Many made mistakes!

1. SELECT - FROM - WHERE - GROUP BY - HAVING - ORDER BY
	1. A condition with GROUP BY should either be “WHERE (condition) GROUP BY” or “GROUP BY (field) HAVING (condition)” – very, very, very basic, but many mistakes ☹
2. With GROUP BY, a field name cannot be in SELECT (1) w/o aggregate function, or (2) w/o in GROUP BY fields
3. Alias is only good for display, but not for math or logic comparisons
4. WHERE states conditions to filter rows, while HAVING filters GROUPs; so WHERE can’t be used with aggregate function; on the other hand, HAVING can’t be used if no GROUP
5. Multiple conditions in WHERE can be connected using AND or OR; but the fields must be restated each time:
	1. WHERE Book\_Type = ‘Fiction’ OR ‘Children’ is wrong;
	2. WHERE Book\_Type = ‘Fiction’ OR Book\_Type = ‘Children’ is right
6. GROUP BY can have any fields
	1. Advice: group by fields with grouping natures (city, major, gender, type, …), but not fields with unique values (DOB, phone number, sales amount, etc)
7. GROUP BY can have multiple fields; the more fields are in GROUP BY, the more groups (and smaller groups) there would be, since a bigger group can be broken down by the second GROUP BY field (say by City, TypeOfService, a 9-city, 3-type table of restaurants could end up 27 groups, since each of the 9 cities could have three types; or each of the 3 types could be found in nine cities)
8. ORDER BY can have multiple fields – primary sort and secondary sort (review by yourself) (🡸 🡸 🡸 And, there is no “SORT” in SQL – no such reserve word!!! )
	1. ORDER BY and GROUP BY: do not mistake one for the other (and should not) –
		1. ORDER BY sorts the individual rows according to the value of one, two, or more fields; while
		2. GROUP BY does NOT have individual rows AT ALL: a query with GROUP BY only display group statistics (such as AVG, COUNT, MAX, SUM) and no individual row values.
	2. Summary: (1) ORDER means sort! (2) GROUP BY means “collapsing individual rows into groups” – so there’s NO INDIVIDUAL ROW values in the output.
9. (Not so basic but still pretty basic) IN – when your field’s value should be ONE among several values, use IN: Cust\_ID IN (‘3’,’5’,’8’,’11’) – please do NOT use “=” here: “=” is for exactly equal to ONE value, but not “one among the following values”.
10. (None: there’re not even ten items to be put on this list.)