

- Be as understandable and “natural language” as possible when giving English descriptions of queries — try to capture what something means rather than just reading the syntax. For example, “employees without a supervisor” is more meaningful than just “employees without `superssn`” and “estimated monthly hours” is more meaningful than “`hours*4`”.
- Capture the full association. For example:

```
SELECT deptnum, projloc  
FROM PROJECT
```

isn’t just departments and project locations, it is departments with *their* project locations.

- While SQL keywords are not case sensitive, table names and column names are.
- Remember three-valued logic: expressions containing operations like `=`, `<`, `<>`, etc evaluate to `UNKNOWN` if (at least) one of the values involved is `NULL`. `IS NULL` or `NOT IS NULL` is needed to determine if a value is (or isn’t) `NULL`.
- Avoid duplicates when appropriate to do so — consider whether there could be duplicates in a result set and what is desired/expected.
- You can use a column alias defined in the `SELECT` clause in `ORDER BY` (ordering is done as the very last step, actually after the `SELECT` is applied to determine the output column list) but not in `WHERE` (`WHERE` is done before `SELECT` so the alias doesn’t exist yet).
- `ORDER BY` takes a comma-separated list of columns e.g. `ORDER BY a,b`. Writing `ORDER BY a AND b` instead is unfortunately valid syntax but it doesn’t do what you want — it instead orders by the result of the expression `a AND b`, which likely evaluates to 0 (false) for every row and thus has no effect on the sort order.
- Be careful to avoid off-by-one errors with more than (`>`) vs. at least (`>=`) and less than (`<`) vs. at most (`<=`).
- For #12, base pay is the \$20/hour applied to all of the hours worked (including any over 40). Overtime pay is the extra \$5/hour for just the hours over 40.