

Super Basic SQL

- Super basic concepts—they are what you would imagine them to be: tables, records/rows, attributes/columns/fields, keys (later we'll see that the devil is in the details)
- Code that shows what these structures look like are comprise the data definition language (DDL)
- Code that shows how to filter or change these structures' content comprises the data manipulation language (DML)

SQL DDL

CREATE TABLE table name (column list);

column list consists of one or more column definitions separated by commas

column definition consists of a column name followed by a data type and optional modifiers—all separated by spaces

- Sample data types: int, varchar, float, date
- Sample modifiers: primary key, references table name(column name), serial

SQL DML: Create Data

```
INSERT INTO table [(columns)] VALUES (values);
```

columns and values are comma-separated lists of column names and value expressions, respectively—they're supposed to correspond

If columns is omitted, then the values are assumed to be in the order that the columns are stored in the table

Delimit strings with single quotes ('); encode a single quote with...two single quotes ('')

SQL DML: Read Data

```
SELECT [DISTINCT] (* | expression list) FROM source(s)  
WHERE condition;
```

expression list is a comma-separated list of column names or a range of other possibilities

source(s) consist of table name(s)—but can also be more

condition is a boolean expression referencing columns from the given source(s) (yes you may use AND, OR, and NOT)

DISTINCT removes duplicate results

Even Moar SELECT

- The `SELECT` statement is the workhorse of SQL—we have barely scratched its surface here
- You can use `ORDER BY` to customize how table rows are sorted (otherwise the database decides)
- You can use `GROUP BY` in order to compute aggregate values—i.e., values that compute something over multiple rows (e.g., `AVG`, `SUM`, `COUNT`)
- You can nest `SELECT` queries (subqueries), call a swath of built-in functions, and much more

SQL Miscellanea

- In case you haven't already noticed, the semicolon (`;`) serves as the SQL statement separator
- Two hyphens (`--`) start a comment from there until the end of the line (like `//` in C-like languages or `#` in Python, among others)
- SQL keywords and identifiers are generally case-insensitive, although reserved words are sometimes capitalized to make it easier to pick them out from table or column names and literal values