Software Carpentry SQL Cheat Sheet

Basic Queries

Select one or more columns of data from a table:

```
SELECT column_name_1, column_name_2 FROM table_name;
```

Select all of the columns in a table:

```
SELECT * FROM table_name;
```

Get only unique lines in a query:

```
SELECT DISTINCT column_name FROM table_name;
```

Perform calculations in a query:

```
SELECT column_name_1, ROUND(column_name_2 / 1000.0) FROM table_name;
```

Filtering

Select only the data meeting certain criteria:

```
SELECT * FROM table_name WHERE column_name = 'Hello World';
```

Combine conditions:

```
SELECT * FROM table_name WHERE (column_name_1 >= 1000) AND (column_name_2 = 'A'
OR column_name_2 = 'B');
```

Sorting

Sort results using `ASC` for ascending order or `DESC` for descending order:

```
SELECT * FROM table_name ORDER BY column_name_1 ASC, column_name_2 DESC;
```

Missing Data

Use `NULL` to represent missing data.

`NULL` is neither true nor false. Operations involving `NULL` produce `NULL`, e.g., `1+NULL`, `2>NULL`, and `3=NULL` are all `NULL`.

Test whether a value is null:

```
SELECT * FROM table_name WHERE column_name IS NULL;
```

Test whether a value is not null:

```
SELECT * FROM table_name WHERE column_name IS NOT NULL;
```

Grouping and Aggregation

Combine data into groups and calculate combined values in groups:

```
SELECT column_name_1, SUM(column_name_2), COUNT(*) FROM table_name GROUP BY
column_name_1;
```

Joins

Join data from two tables:

```
SELECT * FROM table_name_1 JOIN table_name_2 ON table_name_1.column_name =
table_name_2.column_name;
```

Combining Commands

SQL commands must be combined in the following order: `SELECT`, `FROM`, `JOIN`, `ON`, `WHERE`, `GROUP BY`, `ORDER BY`.

Creating Tables

Create tables by specifying column names and types. Include primary and foreign key relationships and other constraints.

```
CREATE TABLE survey(
        taken   INTEGER NOT NULL,
        person  TEXT,
        quant   REAL NOT NULL,
        PRIMARY KEY(taken, quant),
        FOREIGN KEY(person) REFERENCES person(ident)
);
```

Transactions

Put multiple queries in a transaction to ensure they are ACID (atomic, consistent, isolated, and durable):

```
BEGIN TRANSACTION;
  DELETE FROM table_name_1 WHERE condition;
  INSERT INTO table_name_2 values(...);
END TRANSACTION;
```

Programming

Execute queries in a general-purpose programming language by:

- loading the appropriate library
- creating a connection
- creating a cursor
- repeatedly:
    - execute a query
    - fetch some or all results
- disposing of the cursor
- closing the connection

Python example:

```
import sqlite3
connection = sqlite3.connect("database_name")
cursor = connection.cursor()
cursor.execute("...query...")
for r in cursor.fetchall():
    ...process result r...
cursor.close()
connection.close()
```