

SQL

Database Schema: Tables

Kaggle: <https://www.kaggle.com/datasets/nadintamer/top-spotify-tracks-of-2018>

```
sqlite> .schema
CREATE TABLE songs (
  id INTEGER,
  name TEXT,
  artist_id INTEGER,
  danceability REAL,
  energy REAL,
  key INTEGER,
  loudness REAL,
  speechiness REAL,
  valence REAL,
  tempo REAL,
  duration_ms INTEGER
);
CREATE TABLE artists (
  id INTEGER,
  name TEXT
);
```

Table: songs

| id | name | artist_id | danceability | energy | key | loudness | speechiness | valence | tempo | duration_ms |
|----|-----------------------|-----------|--------------|--------|-----|----------|-------------|---------|---------|-------------|
| 22 | Te Bote - Remix | 49 | 0.903 | 0.675 | 11 | -3.445 | 0.214 | 0.442 | 96.507 | 417920 |
| 43 | SICKO MODE | 64 | 0.834 | 0.73 | 8 | -3.714 | 0.222 | 0.446 | 155.008 | 312820 |
| 74 | Walk It Talk It | 45 | 0.909 | 0.628 | 2 | -5.456 | 0.201 | 0.406 | 145.905 | 276147 |
| 75 | Him & I (with Halsey) | 28 | 0.589 | 0.731 | 2 | -6.343 | 0.0868 | 0.191 | 87.908 | 268867 |
| 30 | Perfect | 26 | 0.599 | 0.448 | 8 | -6.312 | 0.0232 | 0.168 | 95.05 | 263400 |

Table: artists

| id | name |
|----|---------------------|
| 1 | 5 Seconds of Summer |
| 2 | 6ix9ine |
| 3 | Anitta |
| 4 | Anne-Marie |
| 5 | Ariana Grande |
| 6 | Bazzi |
| 7 | Bebe Rexha |
| 8 | Becky G |
| 9 | benny blanco |
| 10 | Billie Eilish |
| 11 | BlocBoy JB |
| 12 | Bruno Mars |
| 13 | Calvin Harris |
| 14 | Camila Cabello |
| 15 | Cardi B |

1. **SQL query to list the names of all songs in the database.**
 - Query should output a table with a single column for the name of each song.
 1. `SELECT name FROM songs;`
2. **SQL query to list the names of all songs in increasing order of tempo.**
 - Query should output a table with a single column for the name of each song.
 1. `SELECT name FROM songs ORDER BY tempo;`
3. **SQL query to list the names of the top 5 longest songs, in descending order of length.**
 - Query should output a table with a single column for the name of each song.
 1. `SELECT name FROM songs ORDER BY duration_ms DESC LIMIT 5;`
4. **SQL query that lists names of any songs that have danceability, energy, and valence greater than 0.75.**
 - Query should output a table with a single column for the name of each song.
 1. `SELECT name FROM songs WHERE danceability > 0.75 AND energy > 0.75 AND valence > 0.75;`
5. **SQL query that returns the average energy of all the songs.**
 - Query should output a table with a single column and a single row containing the avg energy.
 1. `SELECT AVG(energy) FROM songs;`
6. **SQL query that lists the names of songs that are by Post Malone.**
 - Query should output a table with a single column for the name of each song.
 - You should not make any assumptions about what Post Malone's artist_id is.
 1. `SELECT name FROM songs WHERE artist_id = (SELECT id FROM artists WHERE name = 'Post Malone');`
7. **SQL query that returns the average energy of songs that are by Drake.**
 - Query should output a table with a single column and single row containing the average energy.
 - You should not make any assumptions about what Drake's artist_id is.
 1. `SELECT AVG(energy) FROM songs WHERE artist_id = (SELECT id FROM artists WHERE name = 'Drake');`
8. **SQL query that lists the names of the songs that feature other artists.**
 - Songs that feature other artists will include "feat." in the name of the song.
 - Your query should output a table with a single column for the name of each song.
 1. `SELECT name FROM songs WHERE name LIKE '%feat.%';`