

Introduction to SQL — Guided Practice (SQLite)

This worksheet gives you hands-on experience with SQL. You'll type every command yourself and watch what happens. Run all commands inside sqlite3 or a SQLite browser such as DB Browser for SQLite.

1. Create the database

```
CREATE TABLE interns (  
  id INTEGER PRIMARY KEY,  
  first_name TEXT,  
  last_name TEXT,  
  city TEXT,  
  cohort TEXT  
);  
  
CREATE TABLE projects (  
  id INTEGER PRIMARY KEY,  
  name TEXT,  
  client TEXT,  
  start_date TEXT  
);  
  
INSERT INTO interns VALUES  
(1, 'Alex', 'Nguyen', 'Eugene', '2025A'),  
(2, 'Bianca', 'Lopez', 'Springfield', '2025A'),  
(3, 'Carlos', 'Kim', 'Eugene', '2025B');  
  
INSERT INTO projects VALUES  
(101, 'Helpdesk Revamp', 'Lane Library', '2025-06-01'),  
(102, 'WiFi Survey', 'City Parks', '2025-06-10');
```

2. View table contents

```
SELECT * FROM interns;
```

3. Selecting specific columns

```
SELECT first_name, last_name FROM interns;
```

4. Filtering rows with WHERE

```
SELECT * FROM interns WHERE city = 'Eugene';
```

5. Sorting results with ORDER BY

```
SELECT first_name, last_name, city FROM interns ORDER BY last_name;
```

6. Joining tables

```
CREATE TABLE assignments (  
  intern_id INTEGER,  
  project_id INTEGER,
```

```

    role TEXT
);

INSERT INTO assignments VALUES
(1,101,'Support'),
(2,101,'Support'),
(3,102,'Field');

SELECT interns.first_name, interns.last_name, projects.name
FROM assignments
JOIN interns ON interns.id = assignments.intern_id
JOIN projects ON projects.id = assignments.project_id;

```

7. Counting results with GROUP BY

```
SELECT city, COUNT(*) AS num_interns FROM interns GROUP BY city;
```

8. Summing numeric data

```

CREATE TABLE timelogs (
    id INTEGER PRIMARY KEY,
    intern_id INTEGER,
    project_id INTEGER,
    hours REAL
);

INSERT INTO timelogs VALUES
(1,1,101,4),
(2,2,101,3),
(3,3,102,6),
(4,1,101,5);

SELECT intern_id, SUM(hours) AS total_hours
FROM timelogs
GROUP BY intern_id;

```

9. Filtering groups with HAVING

```
SELECT intern_id, SUM(hours) AS total_hours FROM timelogs GROUP BY intern_id HAVING SUM(hours) > 5;
```

10. Updating and deleting data

```

INSERT INTO timelogs VALUES (5,3,102,2);
DELETE FROM timelogs WHERE id = 5;
SELECT * FROM timelogs;

```

