Unit 9: SQL Documentation

SQL development

If you're new to SQL, you can learn more from this course: SQL.

Creating tables

```
CREATE TABLE customers (id INTEGER PRIMARY KEY, name TEXT, age INTEGER, weight REAL);
```

```
CREATE TABLE customers (id INTEGER PRIMARY KEY, age INTEGER);
```

See also: specifying defaults, using foreign keys. For more details, see the following: SQLite reference for CREATE .

Inserting data

```
INSERT INTO customers VALUES (73, "Brian", 33);
```

```
INSERT INTO customers (name, age) VALUES ("Brian", 33);
```

See also: The SQLite reference for INSERT .

Querying data

```
SELECT * FROM customers;
```

```
SELECT * FROM customers WHERE age > 21;
```

```
SELECT * FROM customers WHERE age < 21 AND state = "NY";
```

```
SELECT * FROM customers WHERE plan IN ("free", "basic");
```

```
SELECT name, age FROM customers;
```

```
SELECT * FROM customers WHERE age > 21 ORDER BY age DESC;
```

```
SELECT name, CASE WHEN age > 18 THEN "adult" ELSE "minor" END "type" FROM customers;
```

Transform with CASE

See also: filtering with LIKE, restricting with LIMIT, using ROUND and other core functions. For more details, see: the SQLite reference for SELECT .

Aggregating data

```
SELECT MAX(age) FROM customers;
```

```
SELECT gender, COUNT(*) FROM
students GROUP BY gender;
```

See also: restricting results with HAVING.

Joining related tables

```
SELECT customers.name,
orders.item FROM customers JOIN
orders ON customers.id =
orders.customer_id;
```

```
SELECT customers.name,
orders.item FROM customers LEFT
OUTER JOIN orders ON
customers.id =
orders.customer_id;
```

Updating and deleting data

```
UPDATE customers SET age = 33
WHERE id = 73;
```

```
DELETE FROM customers WHERE id =
73;
```

Also see: ALTER TABLE.

Our implementation of SQL is based on a popular dialect of SQL known as SQLite .
To run SQL in the browser, we make use of these technologies:

asm.js
Emscripten
sql.js