

## Updating and Deleting Records

80 minutes (2 periods)

Python Web Backend

Outcome 3.3

Outcome 3.2

Outcome 1.22

Outcome 2.20

Outcome 2.7

Outcome 1.15

Outcome 2.15

### WHAT THIS LESSON DOES

In this lesson, you'll enhance your Flask app by learning to update and delete records in a SQLite database. Follow step-by-step instructions to create edit forms, implement update and delete routes, and ensure data refreshes after changes.

### CURRICULUM OUTCOMES

| Outcome | What the specification asks for                                                                                                                                                                                  |
|---------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 3.3     | use appropriate programming languages to develop an interactive website that can display information from a database that meets a set of users' needs                                                            |
| 3.2     | create a basic relational database to store and retrieve a variety of forms of data types                                                                                                                        |
| 1.22    | read, write, test, and modify computer programs                                                                                                                                                                  |
| 2.20    | identify and fix/debug warnings and errors in computer code and modify as required                                                                                                                               |
| 2.7     | implement algorithms using a programming language to solve a range of problems                                                                                                                                   |
| 1.15    | consider the quality of the user experience when interacting with computers and list the principles of universal design, including the role of a user interface and the factors that contribute to its usability |
| 2.15    | explain what is meant by the World Wide Web (WWW) and the Internet, including the client server model, hardware components and communication protocols                                                           |

### WHAT STUDENTS SHOULD BE ABLE TO DO

- Understand how to implement update and delete operations in a Flask application with SQLite.
- Develop skills to create and manage edit forms for modifying existing database records.
- Master the use of SQL UPDATE and DELETE statements to manipulate data safely.
- Apply basic error handling and user confirmation to enhance application reliability.
- Integrate template inheritance for consistent navigation across web pages.

### BEFORE THE LESSON

- Confirm every student still has a working FlaskSQLiteIntro project with users and hobbies tables populated. Anyone who lost theirs cannot follow this lesson, so keep a known-good copy to hand out.
- Run the app yourself first and click through to /users so you know the starting state matches the code shown here.
- Check Flask is installed and the dev server starts on 127.0.0.1:5000 in your room's Python setup.

## Updating and Deleting Records

### HOW THE LESSON RUNS

Introduction

Website Navigation

Create Edit Form

SQL UPDATE Statement

Implement Update Route

SQL DELETE Statement

Implement Delete Route

Challenge

Wrap Up

### TEACHING MOVES

- 2. Website Navigation. Show why base.html matters before typing it: change the nav once, every page updates. Make sure students grasp that {% extends %} and {% block %} pair up, or their templates will render blank.
- 3. Create Edit Form. Point out the form must be pre-filled with current values so editing is not retyping from scratch. Walk one row of the table so they see where the Edit and Delete links attach per user.
- 4. SQL UPDATE Statement. Stop hard on WHERE. Demonstrate live what UPDATE with no WHERE would do to every row. This is the single most important minute of the lesson.
- 5. Implement Update Route. Highlight the ? placeholder and the tuple of values. Explain this is parameterised query, not string concatenation, and that it is what keeps user input from breaking the SQL.
- 6. SQL DELETE Statement. Reinforce the WHERE lesson again. Ask what DELETE FROM users; alone would do before you show them, then confirm it empties the whole table.
- 7. Implement Delete Route. Have them test the confirmation dialog, then delete, then check the row is gone from /users. Name that this route uses GET for simplicity and note that is not ideal practice for a destructive action.
- 8. Challenge. Steer them toward the hobbies CRUD extension, since it reuses exactly the pattern they just built. Flag that deleting a hobby linked to users needs thought about the related records.

### WHAT TO WATCH FOR

- Students omit the WHERE clause on UPDATE or DELETE and change or wipe every row instead of one. Make WHERE a rule they say aloud before running any UPDATE or DELETE. Show the damage once on a throwaway copy so it is memorable, not abstract.
- Students think a page that looks right in the browser proves the database actually changed. Have them refresh /users after every operation and confirm the change persists. Better still, reopen the record and check the stored value, not just the displayed form.
- Template inheritance renders a blank or broken page when block names do not match between base.html and the child template. Get them to check that every {% block name %} in the child matches a block of the same name in base.html, and that each block is closed.

### DIFFERENTIATION

- Emerging: Pair a student whose project is broken with one whose works, or hand them the known-good copy so they can still practise UPDATE and DELETE. Give them just the delete route to get working end to end before attempting the more complex edit form.
- Developing: Ask them to explain, before running it, exactly which rows their WHERE clause will affect. Have them trace what happens step by step from clicking Edit to the updated row appearing.
- Proficient: Set the hobbies CRUD extension, including deciding what happens to users whose hobby is deleted. Ask them to change the delete route to use POST rather than GET and explain why a destructive action should not sit on a GET request. This is strong Coursework Project evidence.