



SQL and SQLite

Model-Answer Approach

[Visit our website](#)

Auto-graded task 1

The solution takes a structured approach to performing SQL database operations. It begins by creating a `Student` table using the `CREATE TABLE` command, with `STU_NUM` set as the primary key to ensure unique identification of each record. Next, the `INSERT INTO` command populates the table with student data, laying the groundwork for subsequent operations.

The solution retrieves specific records using a `SELECT` query that filters students by `COURSE_CODE`, demonstrating how to extract targeted information from the database. It then updates individual records with the `UPDATE` command, such as changing a student's course code based on their `STU_NUM`. This step ensures that the database remains accurate and up-to-date.

A `DELETE` command is used to remove a specific record, illustrating how logical operators can be employed to precisely target data for deletion. Additionally, a more complex update is performed, modifying `PROJ_NUM` for students who started before a certain date and are enrolled in specific courses.

Finally, the solution concludes by removing the `Student` table entirely with the `DROP TABLE` command.

Auto-graded task 2

The solution uses a structured approach to perform database operations with SQLite in a Python script named **database_manip.py**. It starts by connecting to an SQLite database or creating it if it doesn't exist. A `cursor` object is used to execute SQL commands. First, it creates the `python_programming` table with `CREATE TABLE IF NOT EXISTS`, avoiding errors if the table already exists.

Next, it inserts data by defining a list of tuples with student information and using `executemany` for efficiency. It retrieves records with grades between 60 and 80 using a `SELECT` command and prints them for verification. Carl Davis's grade is updated to 65 with an `UPDATE` command, and Dennis Fredrickson's row is deleted with a `DELETE` command. The script then updates the grades of all students with an `id` greater than 55 to 80.

A common pitfall is forgetting to commit changes, which would prevent data from being saved. The script includes a `commit` command to ensure changes are stored. Finally, the connection is closed to free resources. This approach is clear and logical, making it easy for beginners to understand and use basic database operations in Python.