📘 Intermediate SQL – Lesson 4: Updating and Deleting Data

🎯 What You'll Learn

In this lesson, you'll learn how to:

- Update specific rows in a table using the `UPDATE` statement
- Delete records from a table using the `DELETE` statement
- Safely modify or remove data using best practices and precautions

---

🔄 What is the Difference Between INSERT, UPDATE, and DELETE?

Operation	Purpose
`INSERT`	Adds a **new row** to a table
`UPDATE`	**Modifies** values in an existing row
`DELETE`	**Removes** an entire row from a table

✏️ Using UPDATE

The `UPDATE` statement allows you to change data in a specific row or rows.
To avoid accidentally updating the whole table, you must always use a **`WHERE` clause** to target exactly the record(s) you want to modify.

You'll specify:

- The table you're updating
- The column(s) and new value(s)
- A condition to match the row(s) you want to update

💡 Example Use Case:

Updating employee Holly Flax's missing details (like ID, age, and gender) by targeting her row based on her name.

🗑️ Using DELETE

The `DELETE` statement is used to remove a row permanently from a table.
Like `UPDATE`, it must be paired with a **`WHERE` clause** to ensure you're only deleting what you intend to.

⚠️ Warning:

Deleting without a `WHERE` clause will **erase the entire table**, and **there is no undo** in SQL.

✅ Best Practice:

Before you run a `DELETE`, first test it by running a `SELECT` with the same condition.
This lets you confirm which rows would be deleted, acting as a **safety check**.

📌 Recap

✅ Use `UPDATE` to modify data within existing rows
✅ Use `DELETE` to remove entire rows from a table
✅ Always use `WHERE` to target updates or deletions
✅ Use `SELECT` as a safeguard before executing a `DELETE`
✅ Be extra cautious — deletions are **permanent**