**Part 3: Triggers**

**Problem Statement:**

In your two-week SQL course, you've observed an employee in your institute managing the student details in two tables: `Student\_details` and `Student\_details\_backup`. The employee routinely deletes records from the `Student\_details` table after students complete their courses, while simultaneously maintaining a backup in the `Student\_details\_backup` table by inserting the deleted records. Your goal is to assist the employee by creating a trigger that automatically inserts the deleted student details into the backup table before any deletion occurs in the primary table.

**Data Description:**

**Student\_details:**

- Attributes: Student\_id (Primary Key), Student\_name, Mail\_id, Mobile\_no.

- Purpose: This table holds the current records of all students enrolled in courses at your institute.

**Student\_details\_backup:**

- Attributes: Student\_id (Primary Key), Student\_name, Mail\_id, Mobile\_no.

- Purpose: This table serves as a backup for the `Student\_details` table, capturing all the records before any deletion occurs in the primary table.