**Case Study 2: Calculating Student GPA with a User-Defined Function**

**Detailed Problem Statement**:

* The university needs to calculate students’ Grade Point Averages (GPAs) to assess academic performance for transcripts, scholarship eligibility, and graduation requirements.
* Currently, GPA calculations are performed manually or through ad-hoc queries by administrative staff, which involves retrieving grades and credit hours from the StudentCourses and Courses tables, mapping letter grades to grade points (e.g., A = 4.0, B = 3.0), and computing a weighted average.
* This process is labor-intensive, prone to errors (e.g., incorrect grade mapping or division errors), and lacks scalability as the number of students and courses grows.
* Furthermore, GPA calculations are frequently required across various university processes, such as generating transcripts, academic reports, and student evaluations, necessitating a reusable and consistent method.
* The absence of a standardized, automated function leads to duplicated efforts, inconsistent results, and delays in academic reporting.

**Requirements**:

* Create a reusable database function to calculate a student’s GPA based on their completed courses, considering:
  + Letter grades (A, B, C, D, F) mapped to grade points (4.0, 3.0, 2.0, 1.0, 0.0).
  + Weighted calculations based on course credit hours.
  + Handling edge cases, such as students with no completed courses (returning a GPA of 0.00).
* The function must be easily callable within SQL queries, stored procedures, or reports to support integration with other university processes.
* The solution should operate within the SQL Server environment (UniversityDB) and ensure accuracy and performance for large datasets.
* The function should be maintainable, allowing updates to grade mappings or credit hour calculations if university policies change.

**Stakeholders**:

* Students requiring accurate GPA information for academic planning.
* Academic advisors and registrar staff needing GPA data for evaluations and reports.
* Database administrators ensuring the function’s performance and reliability.

**Current Challenges**:

* Manual GPA calculations are time-consuming and error-prone.
* Lack of a standardized method leads to inconsistent GPA results across departments.
* Ad-hoc queries are not reusable, increasing development and maintenance efforts.
* Difficulty handling edge cases, such as students with no completed courses.

**Proposed Solution**: A scalar-valued User-Defined Function (CalculateStudentGPA) that computes a student’s GPA by aggregating grade points and credit hours from completed courses, ensuring accuracy and reusability across the database system.