

Coursework Overview

- Module: Databases and Application Development (ACCA5026)
- Assessment: 100% of module grade
- Submission Deadline: 22nd August 2025 at 14:00
- Word Count: Equivalent to 5,000 words
- Deliverables:
 - Written report (submitted via Turnitin)
 - Zipped .sql file (via Moodle “additional files” upload)
 - Website project and report (for Part B)

Part A: Database Design & Programming (AdventureWorks Case Study)

Key Tasks:

1.Database Design:

1. Analyse requirements of AdventureWorks Cycles
2. Create **Entity Relationship Diagram (ERD)** with primary & foreign keys
3. Normalize all tables to **3rd Normal Form (3NF)**

2.Database Implementation in SQL Server:

1. Write SQL code to:
 1. Create tables with constraints (PK, FK, CHECK)
 2. Populate each table with **minimum 4 realistic records**
 3. Validate data integrity through tests

Part A: Database Design & Programming (AdventureWorks Case Study)

Key Tasks:

- Transaction Management:
 - Create stored procedures using ROLLBACK and COMMIT
 - Demonstrate transactional scenario involving multiple SQL statements
- Performance Tuning:
 - Identify important queries
 - Create secondary indexes to improve performance
 - Justify choice of indexed columns
- Deliverables for Part A:
 - .sql file including:
 - Table creation scripts
 - Constraints
 - Data population
 - Transaction tests
 - Indexing enhancements

Part A: Database Design & Programming (AdventureWorks Case Study)

Key Tasks:

Written report containing:

- Table of contents
- Introduction
- ERD diagram
- Screenshots of SQL execution
- Transaction and indexing explanation
- References and appendix

Part B: Database-Driven Website

Key Tasks:

1. Develop a PHP Website:

1. Use PHP and MySQL
2. Minimum 3 tables with 15–20 sample records
3. Back-end integration with the database

Website Features:

- Dynamic content display
- **Search function**
- **CRUD** operations (Create, Read, Update, Delete)
- Basic security:
 - User login
 - Data validation

Accompanying Short Report (~500 words):

- Overview of **SQL Injection**
- Examples and how it affects websites
- Screenshots

Deliverables for Part B:

- **Zipped folder** with:
 - Code files (HTML, CSS, PHP, etc.)
 - SQL file for database export
- **Written report** (via Turnitin)

Summary

You need to:

- 1.Design and implement a database system (Part A)
- 2.Create a transactional demo with rollback/commit
- 3.Apply indexing to improve performance
- 4.Submit SQL code as a zipped file
- 5.Write and submit a formal report
- 6.Build a PHP-based CRUD website with search (Part B)
- 7.Write a security report on SQL Injection with code and screenshots
- 8.Submit your final website and report files via Turnitin and Moodle