Unit 3: Database Design for Educational Web Applications

Unit 3 — Moodle Introduction

Unit 3 introduces the theory and practice of relational database design for eLearning applications. You will learn to model data using UML class diagrams, design normalised database schemas, and implement them in MySQL using phpMyAdmin. Mastering database design is critical: a poorly designed database causes data inconsistency, performance problems, and security vulnerabilities that are extremely difficult to fix after deployment.

MULTIMEDIA

Pedagogically Meaningful Animation: A step-by-step animated diagram in the Unit 3 H5P video demonstrates the normalisation process—showing an unnormalised table being split into 1NF, then 2NF, then 3NF—with colour coding to highlight which column is being moved and why. The animation directly addresses the abstract nature of functional dependencies, which research shows is a common conceptual stumbling block for novice database designers (Coronel & Morris, 2019).

Unit Aim

The aim of Unit 3 is to enable students to design, normalise, and implement a relational database schema suitable for a web-based eLearning application, using UML class diagrams for modelling and MySQL/phpMyAdmin for implementation.

Intended Learning Outcomes

- Model data requirements for an eLearning system using a UML class diagram, correctly representing entities, attributes, relationships, and multiplicities (Bloom's Level 3 — Create).
- Apply the three normal forms to a given database schema, identifying and resolving functional dependencies, partial dependencies, and transitive dependencies (Bloom's Level 3 — Apply).
- Implement a relational database schema in MySQL using phpMyAdmin, defining appropriate data types, primary keys, foreign keys, and constraints (Bloom's Level 3 — Apply).
- Construct SQL SELECT, INSERT, UPDATE, and DELETE queries to retrieve and manage data in an eLearning database (Bloom's Level 3 — Apply).

3.1 Database Fundamentals

A database is an organised collection of structured data. In web-based eLearning applications, the database is the heart of the system: it stores user accounts, course content, assessment results, discussion posts, enrolment records, and every other piece of information the application needs to function.

Definition: Relational Database

A database that organises data into tables (relations). Each table has rows (tuples/records) and columns (attributes/fields). Tables are linked through relationships defined by primary keys and

foreign keys. Relational databases are managed by a Relational Database Management System (RDBMS) such as MySQL, PostgreSQL, or Oracle.

Term	Definition and Educational Example
Table (Relation)	A structured grid of rows and columns representing one entity type. Example: A users table stores all user accounts.
Row (Record / Tuple)	A single instance of the entity. Example: One row in the users table represents one student.
Column (Attribute / Field)	A single property of the entity. Example: The email column stores each student's email address.
Primary Key (PK)	A column or combination of columns whose values uniquely identify each row. Cannot be NULL. Should be an auto-incrementing integer in most cases.
Foreign Key (FK)	A column in one table that references the Primary Key of another table, establishing a link between them. Enforces referential integrity.
Index	A data structure that speeds up data retrieval. Primary keys are automatically indexed. Columns used frequently in WHERE clauses should also be indexed.
NULL	The absence of a value. NULL is not the same as zero or empty string. Be deliberate about which columns are allowed to be NULL.

3.2 Data Modelling with UML Class Diagrams

Relationship	Definition and eLearning Example
One-to-One (1:1)	Each record in Table A corresponds to exactly one record in Table B, and vice versa. Rare in practice. Example: One user account has exactly one user profile.
One-to-Many (1:N)	One record in Table A corresponds to many records in Table B. The most common relationship type. Example: One course has many enrolments.
Many-to-Many (M:N)	Many records in Table A correspond to many records in Table B. Implemented using a junction (bridge) table. Example: Many students can enrol in many courses.

3.3 Database Normalisation

Normalisation is the process of organising a database to reduce data redundancy and improve data integrity. It proceeds through a series of normal forms (NF), each eliminating a specific type of problematic dependency. For most eLearning applications, achieving Third Normal Form (3NF) is the practical target.

3.3.1 First Normal Form (1NF)

A table is in 1NF if: all column values are atomic (indivisible), there are no repeating groups or arrays in any column, and there is a primary key.

Common 1NF Violation

Storing multiple values in a single cell violates 1NF. For example, a courses table with a column enrolled_students containing '201, 204, 207' is NOT in 1NF. Each enrolment must be a separate row in a related enrolments table.

3.3.2 Second Normal Form (2NF)

A table is in 2NF if it is in 1NF AND every non-key attribute is fully functionally dependent on the entire primary key (no partial dependencies). Partial dependencies only arise in tables with composite primary keys.

3.3.3 Third Normal Form (3NF)

A table is in 3NF if it is in 2NF AND there are no transitive dependencies (no non-key attribute depends on another non-key attribute). Example: a students table with columns (student_id PK, email, department_id, department_name). Here, department_name depends on department_id, not on student_id—this is a transitive dependency. The solution is to create a separate departments table.

3.4 SQL: Implementing and Querying the Database

SQL (Structured Query Language) is the standard language for interacting with relational databases. SQL statements are divided into four categories: DDL (Data Definition Language) for creating and altering tables; DML (Data Manipulation Language) for inserting, updating, and deleting data; DQL (Data Query Language) for retrieving data; and DCL (Data Control Language) for managing permissions.

UDL NOTE

Multiple Means of Action and Expression (UDL Guideline 5): SQL practicals are available in three formats: (1) guided walkthroughs with step-by-step screenshots in the Moodle resources, (2) video demonstrations in the H5P Interactive Video, and (3) an interactive SQL sandbox embedded via an iframe pointing to sqliteonline.com, where students can type and test queries without installing any software. This removes the technological access barrier for students working from public computer labs.

3.5 Introduction to NoSQL: MongoDB

Characteristic	Relational (MySQL) vs. Document (MongoDB)
Data model	Tables with rows and columns vs. Collections of JSON documents
Schema	Fixed (defined at creation) vs. Flexible (per-document schema possible)
Relationships	Enforced via foreign keys vs. Embedded documents or references (manual)
Best use case	Structured transactional data (users, grades, enrolments) vs. Flexible content (course materials, analytics events, logs)

Moodle support	Primary (MySQL/PostgreSQL) vs. Not natively supported

Unit 3 Assessments — ABC Learning Design Framework

Activity	ABC Type	Description	Bloom's Level	Weight
UML Class Diagram — E-tivity 1	Production / Investigation	Design a complete UML class diagram for an eLearning system supporting: users, roles, courses, enrolments, assignments, submissions, and grades. Use draw.io or similar. Submit as PNG via Moodle Assignment.	Create (L6)	10% (ICA)
Normalisation Exercise — E-tivity 2	Practice / Investigation	Given a provided unnormalised eLearning data table, apply 1NF, 2NF, and 3NF in sequence, showing each step and explaining each transformation in 50–100 words per step.	Apply (L3)	5% (ICA)
phpMyAdmin Implementation — E-tivity 3	Practice / Production	Implement your UML design as a MySQL database using phpMyAdmin. Create all tables with correct data types, primary keys, foreign keys, and constraints. Insert at least five rows of realistic sample data per table. Export the SQL file and submit.	Apply (L3)	10% (ICA)
SQL Query Practical	Practice	Write and test five SQL queries: (1) SELECT with JOIN and WHERE; (2) aggregate with COUNT and GROUP BY; (3) INSERT; (4) UPDATE with a conditional; (5) DELETE with a safe WHERE clause. Submit SQL file with screenshot evidence.	Apply (L3)	5% (ICA)
Unit 3 Quiz — E-assessment 1	Practice	10-question auto-graded quiz covering normalisation, SQL syntax, data types, keys, and NoSQL concepts.	Remember / Apply	5% (ICA)

		Two attempts; higher score recorded.		

End-Unit Assessment 1: Moodle Quiz — Database Design & SQL

Ready-to-Use Moodle Quiz — Copy & Paste Questions

10 questions. Set: Attempts = 2, Grading = Highest, Shuffle = Yes, Feedback = Immediate.

Question & Answer	Notes
Q1. A table courses has columns: course_id (PK), title, lecturer_id, lecturer_name. The column lecturer_name depends on lecturer_id, not course_id. Which normal form violation does this represent? ✓ A) Third Normal Form (3NF) — transitive dependency B) First Normal Form (1NF) — non-atomic value C) Second Normal Form (2NF) — partial dependency D) No violation; this is acceptable design Feedback: A transitive dependency exists when a non-key column (lecturer_name) depends on another non-key column (lecturer_id) rather than the primary key. This violates 3NF. Resolve by moving lecturer information to a separate lecturers table.	
Q2. Which SQL clause is used to filter the results of a GROUP BY operation (not individual rows)? ✓ A) HAVING B) WHERE C) ORDER BY D) LIMIT Feedback: WHERE filters individual rows before grouping. HAVING filters groups after the GROUP BY clause has been applied. Example: SELECT course_id, COUNT(*) FROM enrolments GROUP BY course_id HAVING COUNT(*) > 30;	
Q3. In Moodle's eLearning database, the enrolments table has a foreign key student_id referencing students.id. A developer tries to INSERT an enrolment row with a student_id value that does not exist in the students table. What happens? ✓ A) The INSERT fails with a foreign key constraint violation B) The INSERT succeeds but the row is flagged as orphaned C) The INSERT succeeds and NULL is stored D) MySQL automatically creates the missing student row Feedback: InnoDB enforces referential integrity. If the referenced primary key does not exist in the parent table, the INSERT is rejected with Error 1452: Cannot add or update a child row: a foreign key constraint fails.	
Q4. A students table has columns (student_id PK, name, email). An enrolments table has columns (student_id FK, course_id FK, grade). What kind of relationship does this junction table implement? ✓ A) Many-to-Many between students and courses B) One-to-Many between students and grades C) One-to-One between students and courses D) One-to-Many between enrolments and courses Feedback: A junction (bridge) table with two foreign keys resolves a many-to-many (M:N) relationship. Many students can be enrolled in many courses; the enrolments table records each individual pairing.	
Q5. Which MySQL data type is most appropriate for storing a student's hashed password? ✓ A) VARCHAR(255) B) TEXT C) CHAR(60) D) BLOB Feedback: bcrypt password hashes are always 60 characters long. CHAR(60) is the most precise choice. VARCHAR(255) is acceptable but wastes marginal space. TEXT and BLOB are inappropriate for fixed-length string data.	
Q6. A UR lecturer wants to count the total number of submissions per assignment. Which SQL query achieves this? ✓ A) SELECT assignment_id, COUNT(*) AS	

total_submissions FROM submissions GROUP BY assignment_id; B) SELECT COUNT(*) FROM submissions; C) SELECT assignment_id FROM submissions WHERE COUNT(*) > 0; D) SELECT assignment_id, SUM(submissions) FROM assignments; Feedback: COUNT(*) counts all rows. GROUP BY assignment_id produces a separate count for each assignment. WHERE cannot be used with aggregate functions—use HAVING instead.	
Q7. Why should the Engine=InnoDB be specified when creating MySQL tables in an eLearning system? ✓ A) InnoDB supports foreign key constraints and ACID-compliant transactions B) InnoDB is faster than MyISAM for all read operations C) InnoDB is the only engine that supports auto-increment primary keys D) InnoDB is required by SCORM standards Feedback: InnoDB is the only MySQL engine that enforces foreign key constraints and supports transactions. MyISAM does not enforce referential integrity, making it unsuitable for relational eLearning databases.	
Q8. Which SQL statement removes a student row with student_id = 42 from the students table while preventing accidental deletion of all rows? ✓ A) DELETE FROM students WHERE student_id = 42; B) DELETE FROM students; C) DROP TABLE students; D) TRUNCATE students; Feedback: The WHERE clause is essential to target a specific row. DELETE FROM students; removes all rows. DROP TABLE removes the entire table structure. TRUNCATE removes all rows without a WHERE option.	
Q9. In which situation is a NoSQL document database such as MongoDB more appropriate than MySQL for an eLearning application? ✓ A) Storing learner activity logs where each event may have different fields B) Storing student grade records that require ACID-compliant transactions C) Managing user enrolments that have strict referential integrity requirements D) Storing fixed-schema course metadata Feedback: Activity logs often have variable structures (different events have different properties)—a flexible document schema handles this better than a rigid relational table. All the other options benefit from relational integrity and fixed schemas.	
Q10. [Short Answer] Explain the purpose of the DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci clause in a CREATE TABLE statement and why it matters for a multilingual eLearning application at the University of Rwanda. Model Answer: utf8mb4 is the full Unicode character set that supports all Unicode code points, including characters outside the Basic Multilingual Plane (such as emoji and certain African language characters). The legacy utf8 MySQL character set only supports three-byte sequences and cannot store four-byte Unicode characters, causing data truncation. utf8mb4_unicode_ci provides linguistically accurate case-insensitive collation, essential for a multilingual eLearning system where content may be in Kinyarwanda, French, and English.	

End-Unit Assessment 2: UML Class Diagram Rubric

Criterion	Distinction (75–100%)	Merit (60–74%)	Pass (50–59%)	Fail (<50%)
Entity & Attribute Identification	All required entities present with correct, well-named attributes and	All entities present; minor attribute naming or data type issues.	Most entities present; some attributes missing or incorrect.	Significant entities or attributes absent.

	appropriate data types annotated.			
Relationship Modelling	All relationships correctly modelled with accurate multiplicities (1:1, 1:N, M:N) and junction tables used where required.	Most relationships correct; one or two multiplicity errors.	Relationships partially modelled; significant multiplicity or junction table errors.	Few or no relationships modelled.
Normalisation Evidence	Diagram demonstrates 3NF: no transitive or partial dependencies visible in entity design.	Design is largely normalised; one 2NF or 3NF violation identifiable.	Some normalisation applied; multiple violations present.	No evidence of normalisation applied.
Diagrammatic Conventions	Standard UML notation used throughout; diagram is legible, professionally presented, and labelled.	Mostly standard notation; minor convention deviations.	Notation inconsistent; diagram partially legible.	Non-standard or illegible diagram.