

Module Title: Database Programming	
Module Code	CMP3872
NQF Level	8
Notional Hours	98
Contact hours	4 hours/week plus 3 hours practical/week
Additional learning requirements	None
NQF Credits	16
Prerequisite	None
Compulsory/Elective	N/A
Semester Offered	Semester 2
Scheduled Review Date	TBC
Module coordinator and Contact Details	Mr. Meinolf Simbenda msimbenda@unam.na Tel: +264 81 356 6972
Module Purpose	
This intensive, project-based course provides a modern database programming using SQL-Alchemy, Python's most powerful ORM (Object-Relational Mapping) tool.	
Overarching Learning Outcome	
Students will build real-world database applications to master database design, query optimization, production-ready practices and deployment of scalable apps in Python.	
Specific Learning Outcomes	
On completing the module students should be able to: <ul style="list-style-type: none"> 1. Design and optimize relational databases using SQL-Alchemy. 2. Write advanced SQL queries, stored procedures, and triggers in open-source DBMS. 3. Integrate databases with applications using Python. 4. Improve performance via indexing, caching and query tuning. 5. Implement security, transactions, and concurrency control in open-source environments. 6. Deploy databases on open-source cloud platforms. 	

Module Content

Module 1: Relational Database Refresher & Advanced SQL
<p>Topics:</p> <ul style="list-style-type: none"> 1. Relational model 2. Normalization, and ACID properties 3. Declarative Base models 4. CRUD operations (Session.add(), commit()) 5. Complex queries (Joins vs. subqueries, recursive queries) <p>Lab 1: Advanced querying using SQL Alchemy: Build a book catalog app</p>
Module 2: Stored Procedures, Triggers & Dynamic SQL
<p>Topics:</p> <ul style="list-style-type: none"> 1. PL/SQL procedures and functions 2. Trigger use cases (audit logs, data validation) <p>Lab 2: Automating workflows with triggers</p>

Module 3: Database Connectivity & Object Relational Mapping

Topics:

1. Python Database Connection: SQLAlchemy
2. AsyncEngine/AsyncSession
3. FastAPI integration

Lab 3: Building a CRUD app with an ORM: Real-time chat app

Module 4: Performance Optimization

Topics:

1. Indexing strategies (B-trees, hash indexes)
2. Query execution plans (EXPLAIN ANALYZE)
3. Eager loading

Lab 4: Optimizing slow (E-commerce queries)

Module 5: Transactions & Concurrency Control

Topics:

1. Isolation levels (Read Committed, Serializable)
2. Deadlocks and optimistic vs. pessimistic locking
3. ACID transactions (with session.begin())
4. SQL injection prevention

Lab 5: Banking system with transfer transactions

Module 6: Schema Migrations

Topics:

1. Migration workflows (alembic revision)
2. Handling schema changes

Lab 6: Version-controlled database for a blog

Student Assessment Strategies

1. Continuous assessment tasks (Weighted to 50%)

- 2x Assignments + 2x Tests + 1 Makeup Test (Optional) – 50%
- Practical Lab Practicals - 10%
- Final Project Demo (Full-stack app with open-source DB integration) – 40%

Tools Stack

1. SQLAlchemy ORM - Database models & queries
2. PostgreSQL - Production database
3. SQLite - Development/testing
4. Alembic - Schema migrations
5. Docker - Containerized deployment

2. Written Exams (Weighted to 50%)

Prescribed Learning Resources

1. Textbooks and Digital Resources

- **Python Database Programming with SQLAlchemy** by Charles Bell (Apress)
- **Essential SQLAlchemy (2nd Ed.)** by *Jason Myers* (O'Reilly)

2. Free Online Tutorials (Best for Hands-On Practice)

- [Real Python's SQLAlchemy Guide](#)
- [SQLAlchemy ORM Tutorial](#)
- [The Architecture of Open Source Applications: SQLAlchemy](#) - Explains how SQLAlchemy works under the hood (connection pooling, query compilation).
- [SQLAlchemy 2.0 Migration Guide](#) – For maintaining legacy code (version control)

Learning and Teaching Strategies/Activities

Lectures: presentation on concepts and other theoretical foundations.

Discussion forums: reflecting on own contexts and sharing perspectives

Collaborative learning: group learning and activities carried as part of projects

Inquiry: carrying out of research to explore and understand scenarios and problems

Practicals: do practical sessions in a computer lab

Presentations and demonstrations: presentation of outcomes of projects (products, processes, impact)

Learning and Teaching Enhancement Strategies

- **Student feedback:** feedback from students using focused feedback instruments
- **Peer feedback:** student feedback on peer evaluation of each other's collaboration, participation and contribution
- **Self-evaluation:** quizzes and students' reflective journal/ portfolio on their own learning
- **Learning analytics:** use of learning management tools on student participation and online learning activities, and analyze assessment performance