🧾 Wittig School Student Management System – Project Report

Author: Rodrigue Deguenon

Date: June 2025

📌 Project Objective

The goal of this project was to design and implement a relational database system that simulates the academic management structure of a higher education institution. The system tracks student enrollments, course offerings, instructor assignments, and department affiliations. The project also includes data analysis through SQL queries to extract meaningful operational insights. Project deliverables include an Entity Relationship Diagram (ERD), SQL schema, mock data generation, and analytical queries.

⚙️ Project Workflow

1. Database Design (ERD & Schema)

Tool Used: MySQL Workbench

- Designed the Entity Relationship Diagram (ERD) with tables: `students`, `instructors`, `courses`, `departments`, and `enrollments`.
- Mapped out relationships such as one-to-many and many-to-many using an EER Diagram.
- Forward engineered the data model to generate SQL scripts for table creation.

Entity Relationship Diagram (ERD): ERD

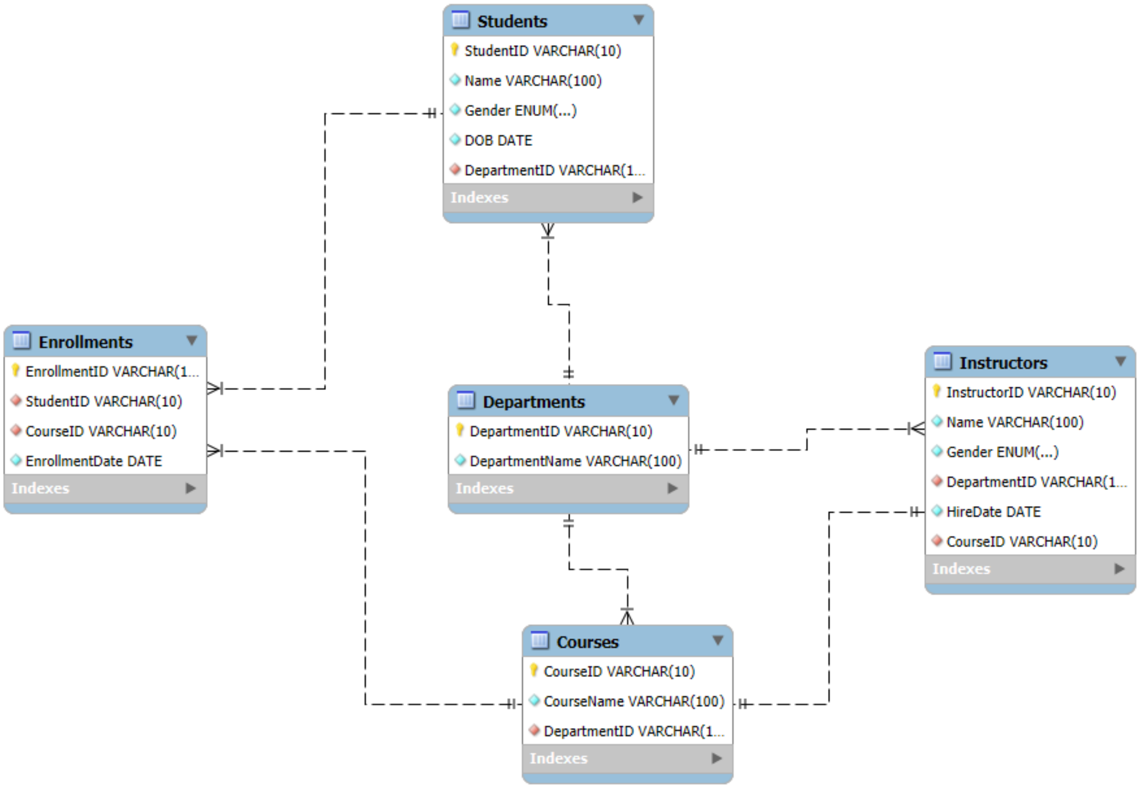

Figure: Entity Relationship Diagram (ERD) of the Wittig School Student Management System

Database creation (DDL) script Link: SQL Schema Script

2. Sample Data Generation

- **Tool Used**: AI (ChatGPT) + Excel

- Used AI to generate mock data for each table.
- Ensured valid formats, consistent foreign key references, and realistic academic data.

Google Sheets link: Sample Data

3. Data Import

- **Challenge**: Table Data Import Wizard was slow in MySQL Workbench.
- **Solution**: Used the command line and `LOAD DATA LOCAL INFILE` for faster, manual CSV imports.

```
LOAD DATA LOCAL INFILE 'students.csv'
INTO TABLE students
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\n'
IGNORE 1 ROWS;
```

4. Data Validation & Integrity Checks

- Conducted data validation checks post-import to ensure:
  - All tables were populated with the expected number of records.
  - Data types matched the schema definitions.
  - Referential integrity was maintained across foreign keys.

- Verified data integrity by checking:
  - Foreign key constraints
  - Data types and formats
  - Referential integrity across tables
- Ensured no duplicate records and all relationships were correctly established.
- Used SQL queries to validate data consistency and completeness.

```
SELECT COUNT(*) FROM students;
SELECT COUNT(*) FROM courses;
SELECT COUNT(*) FROM enrollments;
SELECT COUNT(*) FROM instructors;
SELECT COUNT(*) FROM departments;
SELECT COUNT(*) FROM enrollments WHERE StudentID IS NULL OR CourseID IS NULL;

SELECT * FROM students WHERE StudentID NOT IN (SELECT StudentID FROM enrollments);
SELECT * FROM courses WHERE CourseID NOT IN (SELECT CourseID FROM enrollments);
SELECT * FROM enrollments WHERE CourseID NOT IN (SELECT CourseID FROM courses);
SELECT * FROM enrollments WHERE StudentID NOT IN (SELECT StudentID FROM students);
SELECT * FROM departments WHERE DepartmentID NOT IN (SELECT DepartmentID FROM courses);
SELECT * FROM courses WHERE CourseID NOT IN (SELECT CourseID FROM instructors);
```

5. Data Analysis & Query Development

- Created SQL queries to uncover:

  Student & Enrollment Reports

  - How many students are currently enrolled in each course?
  - Which students are enrolled in multiple courses, and which courses are they taking?
  - What is the total number of students per department across all courses?

  Course & Instructor Analysis

  - Which courses have the highest number of enrollments?
  - Which department has the least number of students?

  Data Integrity & Operational Insights

  - Are there any students not enrolled in any courses?
  - What is the average number of courses taken by students?
  - How is the gender distribution across courses and instructors?
  - Which course has the highest male or female enrollment?

Queries and their logic are included in the Database Documentation

Analysis Script (DML) link: Insights Script

📊 Key Insights

- Top 3 courses with high enrollments are **Counseling** (205 students), **Immunology** (204 students), and **Cybersecurity** (204 students).
- **Electrical Engineering** department host the most students (210 students).
- **Biological Sciences** has the least enrollment (192 students).
- No student is left unenrolled.
- Students take an average of **9 courses**, indicating a heavy academic load compared to the typical academic load of 5-6 courses per semester in similar institutions.
- Gender balance analysis reveals equal representation across most courses, with slight variations in specific programs (e.g., more male students in Operations Systems and more female students in Evolution & Finance).
- The course with the highest male enrollment is Operations Systems, while Evolution & Finance has the highest female enrollment.

Recommendations

- **Course Load Management**: Given the average of 9 courses per student, consider implementing academic advising to help students manage their course loads effectively.
- **Resource Allocation**: Departments with high enrollments may need additional resources or faculty to maintain quality education.
- **Diversity Initiatives**: Continue promoting diversity in course offerings and faculty recruitment to better serve the student population.

✅ Conclusion

This project simulates a real-world student management system using SQL, ER modeling, and AI-powered data generation. It demonstrates full-cycle data system implementation and analytical thinking. The challenges faced during import were solved with flexibility and problem-solving, reflecting readiness for data-oriented roles. The insights derived from the data analysis provide valuable operational metrics that can inform academic planning, resource allocation, and strategic decision-making for the Wittig School.

References

- Database Documentation.
- SQL Schema Script
- Sample Data
- Insights Script

Acknowledgments

- Special thanks to Yusuf Mustapha for the project task.