Project: A college student courses database.

Student ID: 24083279

Student Name: Chirag Srinivas

GitHub: https://github.com/Chirag0455/Chirag_SQL_DB.git

---

1. **Introduction:**

This project involves the creation of a synthetic college enrolment database using SQLite and Python. The goal was to design a realistic dataset that represents a university environment with students, courses, and their enrolment relationships. The database was required to include at least 7 columns incorporating nominal, ordinal, interval, and ratio data types, while also demonstrating good relational database practices such as the use of foreign keys and composite keys. A secondary objective was to generate all data programmatically rather than sourcing it externally, ensuring originality and ethical handling of information. Finally, the project includes a demonstration of how the database can be queried and analysed using SQL, as well as visualized using Python

---

2. **Database Design:**

The database is organized into three interconnected tables: **students**, **courses**, and **enrolments**. The design follows a typical academic management model, where students enrol in multiple courses and each course may have many students. A composite key is used in the enrolment table to prevent duplicate enrolment entries for the same semester and year.

2.1 Tables and Keys

Students Table

- Contains unique student demographics and academic information.
- Includes all the required data types:
    - *Nominal*: major, gender
    - *Ordinal*: class_level (Freshman → Senior)
    - *Interval*: GPA
    - *Ratio*: age
- **Primary key:** student_id

Courses Table

- Represents academic courses across multiple departments.
- Stores course metadata such as course name, department, and credit value.

- **Primary key:** course_id

Enrollments Table

- Captures which students take which courses in which semester and year.
- Contains grade information tied to academic performance.
- **Foreign keys:** student_id → students, course_id → courses
- **Composite primary key:** (student_id, course_id, semester, year)
  This prevents duplicate enrollment entries and enforces referential integrity.

2.2 Ethical & Privacy Considerations

All data in this project is fully synthetic, generated using the Python *Faker* library and randomization techniques. No personal or identifiable information from real individuals is included. Since the dataset mimics student academic records, care has been taken to ensure no real-world entities or individuals can be inferred.

---

3. <u>Database Schema:</u>

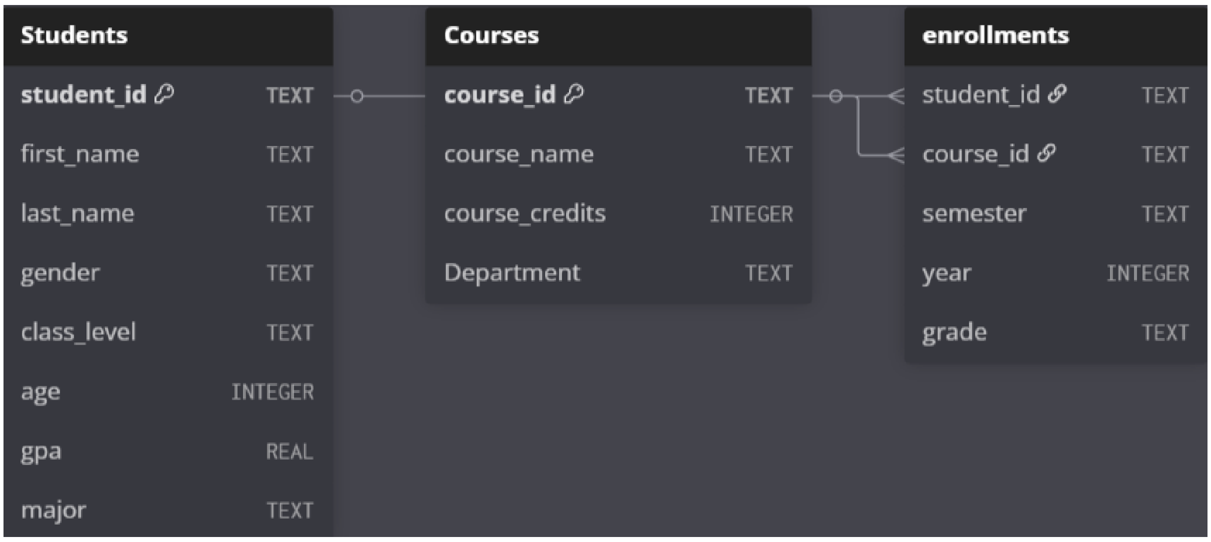

Figure-1: Schema representation of the data base

The database consists of three relational tables **Students**, **Courses**, and **Enrollments** linked through primary and foreign keys. The schema diagram included in the report visually represents these entities and their relationships.

<u>Students Table:</u>

Stores demographic and academic information about each student.
Primary Key: student_id
Key attributes: first_name, last_name, gender (nominal),
class_level (ordinal), age (ratio), gpa (interval), major (nominal).

Courses Table:

Contains details about each course offered by the institution.
Primary Key: course_id
Attributes: course_name, course_credits (ratio), department.

Represents the many-to-many relationship between students and courses.
Foreign Keys:

- student_id → Students
- course_id → Courses

Attributes: semester (nominal), year (ratio), grade.

Relationships

- One student can enroll in many courses.
- One course can have many enrolled students.
- The **Enrollments** table serves as a junction table connecting the two.

This schema ensures clean normalization, prevents duplication, and maintains referential integrity across all tables.

---

4. Data Visualization:

To further explore patterns and trends in the generated dataset, three visualizations were produced using Python's matplotlib library. These charts provide insight into students' academic performance and demographic distribution across majors and class levels.

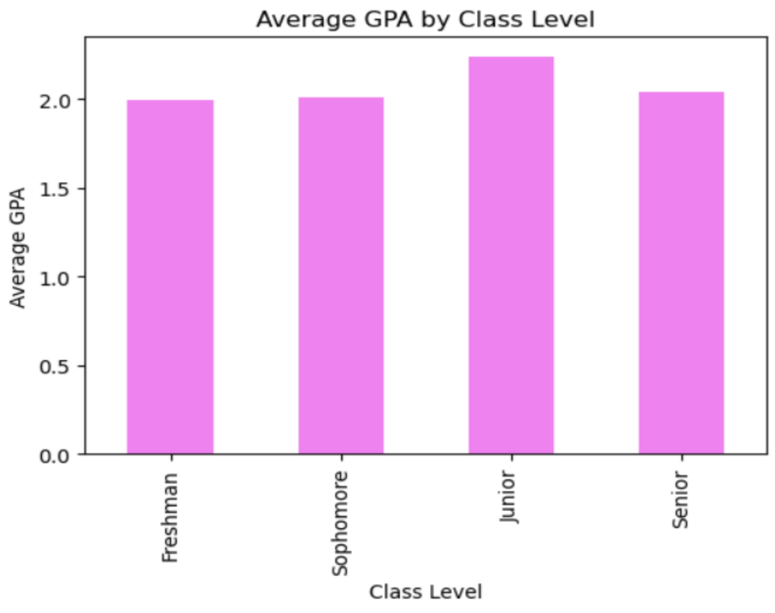

figure-2: A bar chart comparing the GPA by class level

Average GPA by Class Level

The first visualization compares the average GPA across the four class levels: Freshman, Sophomore, Junior, and Senior. The results show relatively consistent GPA averages across categories, with Juniors displaying the highest overall GPA. This suggests that academic performance tends to improve slightly as students progress through their degree before stabilizing in the final year. Because the dataset is synthetic and randomly generated, the differences are moderate, yet the visualization effectively demonstrates how ordinal categories can be analyzed.

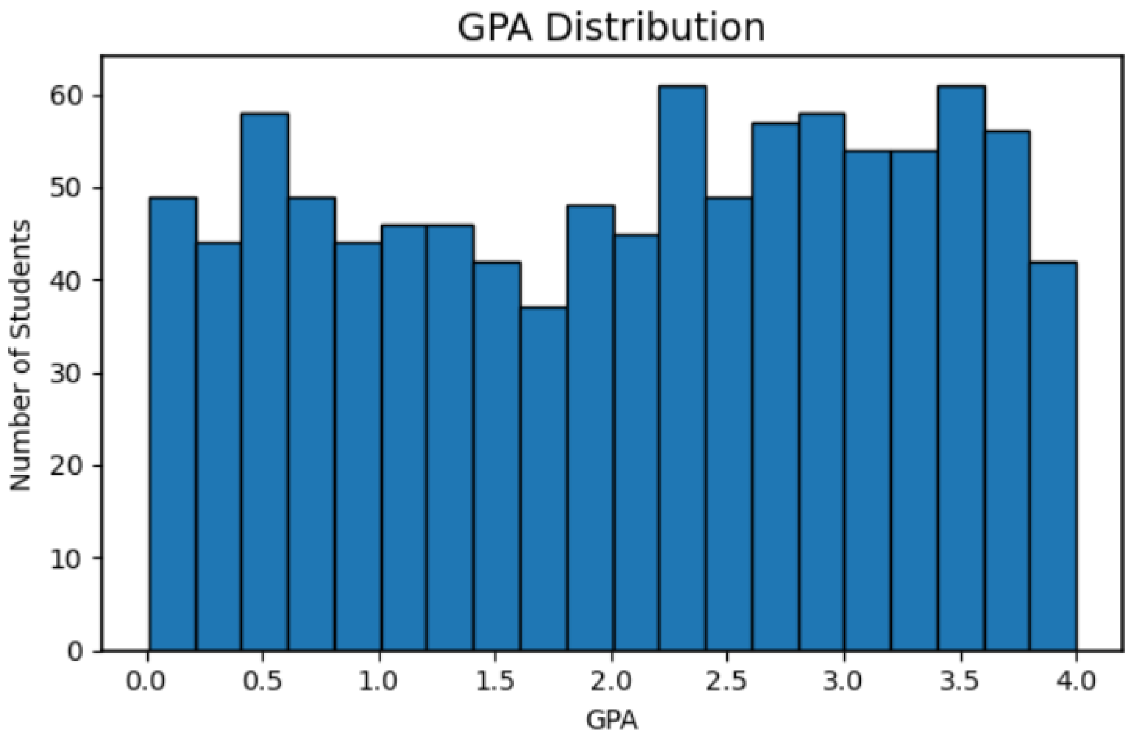

figure-3: GPA distribution of students.

GPA Distribution

The second figure presents a histogram of the GPA distribution for all students. The GPAs span the full range from 0.0 to 4.0, as expected from the uniform random generation. The distribution appears relatively even, with no strong skew or clustering. This indicates that the randomization method produced a wide but balanced spread of academic performance, making the dataset suitable for statistical experiments or machine-learning demonstrations.

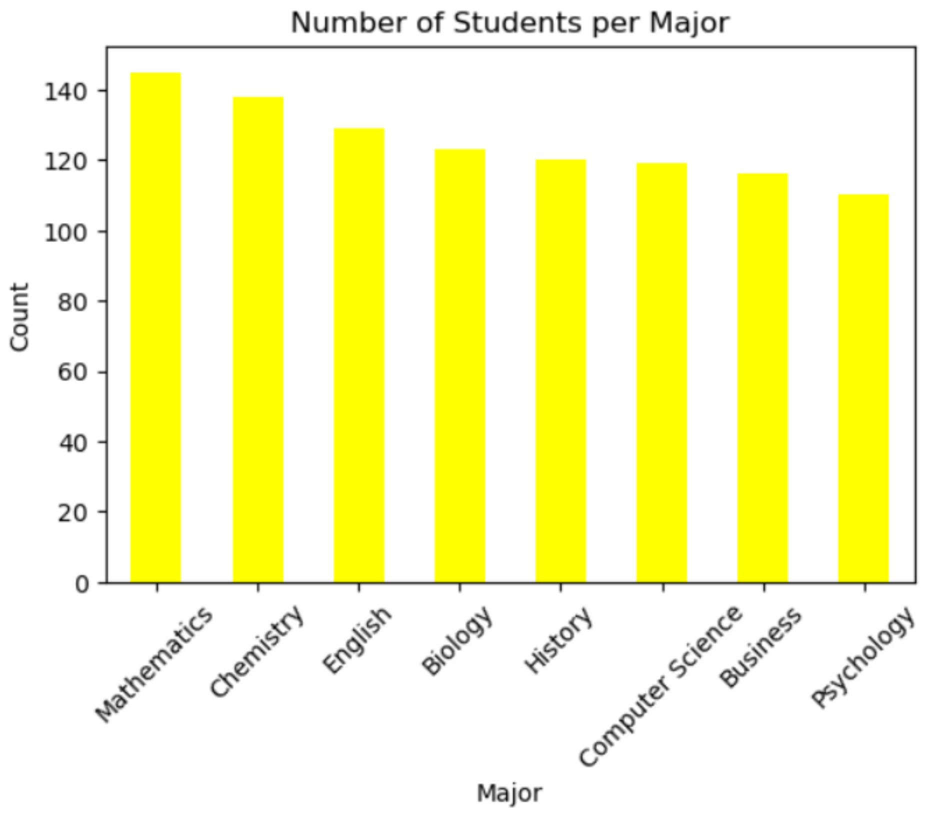

figure-4: Bar chart showing number of students per major.

Number of Students per Major

The final visualization shows the count of students within each academic major. While the student assignments to majors were randomized, the distribution remains balanced across the eight departments included in the dataset. Mathematics, Chemistry, and English appear slightly more represented, while Psychology shows the lowest count. This bar chart confirms that the dataset maintains realistic variability while avoiding extreme imbalances that could distort analysis.

5. Data Generation Methodology

The dataset was created using a Python script that populated the SQLite database with realistic and randomized values.

5.1 Tools Used

- **SQLite** for database creation.
- **Python** for data generation and insertion.
- **Faker library** to produce realistic names and demographic attributes.

5.2 Data Generation Approach

1. **Course Table**
   A fixed list of 20 representative university courses was manually defined for realism. Each course includes a department and credit count (ratio data).
2. **Students Table**
   1,000 students were generated with:

   - Names produced using Faker
   - Gender chosen from fixed options
   - Class level randomly assigned from a defined ordinal set
   - GPA drawn from a uniform distribution between 0.0 and 4.0
   - Age randomized between 17 and 30
   - Major selected from a list of realistic academic fields

3. **Enrollments Table**
   Each student was enrolled in 2-4 courses, with:

   - Semester chosen from a set of three values
   - Year between 2018-2025
   - Grade assigned from typical university grade letters

The combination of Faker-generated demographics and manually curated domain-specific values results in a dataset that is both realistic and fully synthetic.

---

6. Conclusion

This project successfully demonstrates the construction of a relational database from scratch using randomized but realistic academic data. The use of foreign keys and composite keys reinforces relational integrity, while the variety of data types satisfies all requirements for nominal, ordinal, interval, and ratio variables.

The database is large enough (over 1000 records in the primary table) to perform meaningful analysis and visualization. By exporting tables to CSV and using Python for further exploration, the project highlights the practical integration of SQL and data science workflows.

Overall, the database provides a robust foundation for educational, analytical, and experimentation purposes.