Assignment: Python + MySQL Connector

Setup

Install MySQL Connector:

```
pip install mysql-connector-python
```

- Create a database called `school_db` in MySQL.
- Create two tables:
    - `students(student_id, name, age, grade)`
    - `courses(course_id, course_name, instructor)`
    - `enrollments(enroll_id, student_id, course_id)`

Questions

1. Write a Python program to **connect to MySQL** and print the list of all available databases.
2. Create a function in Python to **create the tables** (`students`, `courses`, `enrollments`) in `school_db`.
3. Write a Python program to **insert multiple students and courses** into the respective tables using `executemany()`.
4. Fetch and print all rows from the `students` table in a **tabular format**.

5. Write a program to **update the grade** of a student (given `student_id`) via a Python input prompt.

6. Delete a student from the `students` table using Python input for `student_id`.

7. Write a Python script to insert enrollment records linking students with their courses, ensuring **foreign key integrity**.

8. Fetch all students **enrolled in a particular course** (given `course_id` as input).

9. Write a program to **display students who are not enrolled** in any course.

10. Perform an **INNER JOIN** between `students` and `courses` using Python to show student names along with the courses they are enrolled in.

11. Write a Python function to **search for a student by name** (partial matches allowed using `LIKE`).

12. Add a new column `email` to the `students` table using Python and update it with random values.

13. Create a program to **count how many students are enrolled per course** and display results in descending order.

14. Write a Python script that accepts a course name as input and returns all enrolled students with their grades.

15. Export the contents of the `students` table into a **CSV file** using Python.

16. Write a Python program to **import student data from a CSV file** into the `students` table.

17. Create a stored procedure in MySQL (e.g., `GetStudentCourses(student_id)`) and call it from Python.

18. Handle exceptions in Python to gracefully catch **duplicate entries** (e.g., inserting a student with the same ID).

19. Write a program to implement **transactions** in Python:

    - Insert a student
    - Enroll them in a course
    - Roll back if the second query fails

20. Close the database connection safely in Python, ensuring both cursor and connection are properly closed.