```
import mysql.connector
import os
import sys

conn = None
cursor = None

def initialize(dbname):
    global conn
    global cursor
    conn = mysql.connector.connect(host='localhost', user='root',
password='password', database=f'{dbname}')
    cursor = conn.cursor()
    cursor.execute('''
        CREATE TABLE IF NOT EXISTS Students (
            student_id INT AUTO_INCREMENT PRIMARY KEY,
            name VARCHAR(255) NOT NULL
        )
    ''')
    cursor.execute('''
        CREATE TABLE IF NOT EXISTS Grades (
            grade_id INT AUTO_INCREMENT PRIMARY KEY,
            student_id INT,
            subject ENUM('Physics', 'Chemistry', 'Mathematics'),
            grade FLOAT,
            max_marks FLOAT NOT NULL,
            FOREIGN KEY (student_id) REFERENCES Students(student_id)
        )
    ''')
    conn.commit()

def fetch_db():
    conn = mysql.connector.connect(host='localhost', user='root',
password='password')
    cursor = conn.cursor()
    cursor.execute("SHOW DATABASES LIKE 'Class%'")
    databases = cursor.fetchall()
    conn.close()
    return databases

def add_name():
    name = input("Enter student name: ")
    cursor.execute('INSERT INTO Students (name) VALUES (%s)', (name,))
    student_id = cursor.lastrowid
    max_marks = float(input("Enter maximum marks for all subjects: "))
    subjects = ['Physics', 'Chemistry', 'Mathematics']
    for subject in subjects:
        grade = float(input(f"Enter grade for {subject}: "))
        cursor.execute('''INSERT INTO Grades (student_id, subject, grade,
max_marks) VALUES (%s, %s, %s, %s)''', (student_id, subject, grade, max_marks))
    conn.commit()
    print(f"Student {name} and their grades have been added.")
```

```
def view():
    name = input("Enter student name: ")
    cursor.execute('''
        SELECT subject, grade, max_marks
        FROM Grades
        JOIN Students ON Grades.student_id = Students.student_id
        WHERE Students.name = %s
    ''', (name,))
    grades = cursor.fetchall()
    if grades:
        print(f"Grades for {name}:")
        for subject, grade, max_marks in grades:
            print(f"{subject}: {grade}/{max_marks}")
    else:
        print(f"No grades found for student {name}.")

def percentage():
    name = input("Enter student name: ")
    cursor.execute('SELECT student_id FROM Students WHERE name = %s', (name,))
    student = cursor.fetchone()
    if not student:
        print(f"No student found with name {name}.")
        return
    student_id = student[0]
    cursor.execute('''SELECT SUM(grade) AS total_grade, SUM(max_marks) AS
total_max_marks FROM Grades WHERE student_id = %s''', (student_id,))
    result = cursor.fetchone()
    if result and result[1] > 0:
        total_grade, total_max_marks = result
        percentage = (total_grade / total_max_marks) * 100
        print(f"Percentage for {name}: {percentage:.2f}%")
    else:
        print(f"No valid grades found for student {name}.")

def list_rank():
    cursor.execute('SELECT name, student_id FROM Students')
    students = cursor.fetchall()
    if students:
        student_percentages = []
        for name, student_id in students:
            cursor.execute('''SELECT SUM(grade), SUM(max_marks) FROM Grades WHERE
student_id = %s''', (student_id,))
            total_grade, total_max_marks = cursor.fetchone()
            if total_max_marks and total_max_marks > 0:
                percentage = (total_grade / total_max_marks) * 100
                student_percentages.append((name, percentage))
            else:
                student_percentages.append((name, 0))
        student_percentages.sort(key=lambda x: x[1], reverse=True)
        print("List of all students with their ranks:")
        rank = 1
        for name, percentage in student_percentages:
            print(f"Rank: {rank}, Name: {name}, Percentage: {percentage:.2f}%")
            rank += 1
    else:
        print("No students found.")

def pause():
    input("\nPress Enter to continue...")
    os.system('cls' if os.name == 'nt' else 'clear')

def new_db(cls):
    conn = mysql.connector.connect(host='localhost', user='root', password='password')
    cursor = conn.cursor()
    cursor.execute(f"CREATE DATABASE Class{cls}")
    conn.commit()
    cursor.close()
    conn.close()
    print('Done.')

def start():
    os.system('cls' if os.name == 'nt' else 'clear')
    print("\nStudent Grade Management System")
    print("SELECT A CLASS DATABASE")
    lst = fetch_db()
    x = len(lst)
    for i in range(x):
        print(f"{i + 1}. {lst[i][0]}")
    print(f"{x + 1}. Create new database")
    print(f"{x + 2}. Exit")
    choice = int(input(f"Enter your choice (1-{x + 2}): "))
    if choice > x + 2:
        print("Invalid input, try again.")
        os.system('cls' if os.name == 'nt' else 'clear')
        start()
    elif choice == x + 2:
        sys.exit()
    elif choice == x + 1:
        z = int(input('Enter the class for which you want to create database: '))
        new_db(z)
        pause()
        start()
    else:
        var = lst[choice - 1][0]
        initialize(var)

start()

while True:
    os.system('cls' if os.name == 'nt' else 'clear')
    print("\nStudent Grade Management System")
    print("1. Add student and grades")
    print("2. View grades by student name")
    print("3. Calculate percentage for a student")
    print("4. List all students with their ranks")
    print("5. Exit")
    choice = input("Enter your choice (1-5): ")
    if choice == '1':
        add_name()
        pause()
    elif choice == '2':
        view()
        pause()
    elif choice == '3':
        percentage()
        pause()
    elif choice == '4':
        list_rank()
        pause()
    elif choice == '5':
        print("Exiting the system.")
        break
    else:
        print("Invalid choice. Please enter a number between 1 and 5.")

conn.close()
```