

## 9\_TRANTRONGCHINH

### 1 BƯỚC 1: Hãy kết nối hai bảng trên theo những cách sau:

- Sử dụng tích Decartes.
- Sử dụng JOIN: INNER JOIN, LEFT JOIN, RIGHT JOIN, FULL OUTER JOIN

```
[53]: import sqlite3
import pandas as pd

# Kết nối SQLite trong bộ nhớ
conn = sqlite3.connect("database.db")
cursor = conn.cursor()

# Tạo bảng student
cursor.execute("""
CREATE TABLE student (
    student_id INTEGER PRIMARY KEY,
    name TEXT,
    class TEXT,
    course_id INTEGER,
    score REAL
);
""")

# Thêm dữ liệu vào bảng student
students = [
    (1, 'Nguyen Minh Hoang', 'May Tinh', 12, 6.7),
    (2, 'Tran Thi Lan', 'Kinh Te', 34, 9.2),
    (3, 'Pham Van Nam', 'Toan Tin', None, 7.9),
    (4, 'Le Thanh Huyen', 'Toan Tin', 20, 7.2),
    (5, 'Vu Quoc Anh', 'May Tinh', 24, 8.0),
    (6, 'Dang Thuy Linh', 'May Tinh', 24, 5.5),
    (7, 'Bui Tien Dung', 'Kinh Te', 34, 9.2),
    (8, 'Ho Ngoc Mai', 'Toan Tin', 20, 8.8),
    (9, 'Duong Huu Phuc', 'Kinh Te', None, 7.2),
    (10, 'Cao Thi Hanh', 'May Tinh', None, 7.0)
]
cursor.executemany("INSERT INTO student VALUES (?, ?, ?, ?, ?)", students)
```

```

# Tạo bảng course
cursor.execute("""
CREATE TABLE course (
    id INTEGER PRIMARY KEY,
    course_name TEXT
);
""")

# Thêm dữ liệu vào bảng course
courses = [
    (12, 'Giai tích'),
    (34, 'Thong ke'),
    (26, 'Tin hoc')
]
cursor.executemany("INSERT INTO course VALUES (?, ?)", courses)
conn.commit()

# Kiểm tra dữ liệu đã chèn
print("Bảng student:")
print(pd.read_sql("SELECT * FROM student", conn))
print(pd.read_sql("SELECT name FROM sqlite_master WHERE type='table';", conn))

print("Bảng course:")
print(pd.read_sql("SELECT * FROM course", conn))

```

Bảng student:

	student_id	name	class	course_id	score
0	1	Nguyen Minh Hoang	May Tinh	12.0	6.7
1	2	Tran Thi Lan	Kinh Te	34.0	9.2
2	3	Pham Van Nam	Toan Tin	NaN	7.9
3	4	Le Thanh Huyen	Toan Tin	20.0	7.2
4	5	Vu Quoc Anh	May Tinh	24.0	8.0
5	6	Dang Thuy Linh	May Tinh	24.0	5.5
6	7	Bui Tien Dung	Kinh Te	34.0	9.2
7	8	Ho Ngoc Mai	Toan Tin	20.0	8.8
8	9	Duong Huu Phuc	Kinh Te	NaN	7.2
9	10	Cao Thi Hanh	May Tinh	NaN	7.0

name

0 student

1 course

Bảng course:

	id	course_name
0	12	Giai tích
1	26	Tin hoc
2	34	Thong ke

1 Bảng student chứa danh sách sinh viên với các cột:

student\_id: ID sinh viên.

name: Tên sinh viên.

class: Lớp học.

course\_id: ID khóa học mà sinh viên đang theo học.

score: Điểm số của sinh viên.

Một số nhận xét:

2 Phân tích bảng course

Bảng course chứa danh sách khóa học với các cột:

id: ID khóa học.

course\_name: Tên khóa học.

```
[55]: # 1 Tích Descartes (CROSS JOIN)
query = """
SELECT * FROM student
CROSS JOIN course;
"""

cross_join_df = pd.read_sql_query(query, conn)
print("CROSS JOIN Result:")
display(cross_join_df)

# INNER JOIN
query = """
SELECT * FROM student
INNER JOIN course
ON student.course_id = course.id;
"""

inner_join_df = pd.read_sql_query(query, conn)
print("INNER JOIN Result:")
display(inner_join_df)

# LEFT JOIN
query = """
SELECT * FROM student
LEFT JOIN course
ON student.course_id = course.id;
"""

left_join_df = pd.read_sql_query(query, conn)
print("LEFT JOIN Result:")
display(left_join_df)

# RIGHT JOIN
query = """
SELECT * FROM course
```

```

LEFT JOIN student
ON student.course_id = course.id;
"""
right_join_df = pd.read_sql_query(query, conn)
print("RIGHT JOIN Result:")
display(right_join_df)

# FULL OUTER JOIN
query = """
SELECT * FROM student
LEFT JOIN course ON student.course_id = course.id
UNION
SELECT * FROM student
RIGHT JOIN course ON student.course_id = course.id;
"""
full_outer_join_df = pd.read_sql_query(query, conn)
print("FULL OUTER JOIN Result:")
display(full_outer_join_df)

```

CROSS JOIN Result:

	student_id	name	class	course_id	score	id	course_name
0	1	Nguyen Minh Hoang	May Tinh	12.0	6.7	12	Giai tich
1	1	Nguyen Minh Hoang	May Tinh	12.0	6.7	26	Tin hoc
2	1	Nguyen Minh Hoang	May Tinh	12.0	6.7	34	Thong ke
3	2	Tran Thi Lan	Kinh Te	34.0	9.2	12	Giai tich
4	2	Tran Thi Lan	Kinh Te	34.0	9.2	26	Tin hoc
5	2	Tran Thi Lan	Kinh Te	34.0	9.2	34	Thong ke
6	3	Pham Van Nam	Toan Tin	NaN	7.9	12	Giai tich
7	3	Pham Van Nam	Toan Tin	NaN	7.9	26	Tin hoc
8	3	Pham Van Nam	Toan Tin	NaN	7.9	34	Thong ke
9	4	Le Thanh Huyen	Toan Tin	20.0	7.2	12	Giai tich
10	4	Le Thanh Huyen	Toan Tin	20.0	7.2	26	Tin hoc
11	4	Le Thanh Huyen	Toan Tin	20.0	7.2	34	Thong ke
12	5	Vu Quoc Anh	May Tinh	24.0	8.0	12	Giai tich
13	5	Vu Quoc Anh	May Tinh	24.0	8.0	26	Tin hoc
14	5	Vu Quoc Anh	May Tinh	24.0	8.0	34	Thong ke
15	6	Dang Thuy Linh	May Tinh	24.0	5.5	12	Giai tich
16	6	Dang Thuy Linh	May Tinh	24.0	5.5	26	Tin hoc
17	6	Dang Thuy Linh	May Tinh	24.0	5.5	34	Thong ke
18	7	Bui Tien Dung	Kinh Te	34.0	9.2	12	Giai tich
19	7	Bui Tien Dung	Kinh Te	34.0	9.2	26	Tin hoc
20	7	Bui Tien Dung	Kinh Te	34.0	9.2	34	Thong ke
21	8	Ho Ngoc Mai	Toan Tin	20.0	8.8	12	Giai tich
22	8	Ho Ngoc Mai	Toan Tin	20.0	8.8	26	Tin hoc
23	8	Ho Ngoc Mai	Toan Tin	20.0	8.8	34	Thong ke
24	9	Duong Huu Phuc	Kinh Te	NaN	7.2	12	Giai tich
25	9	Duong Huu Phuc	Kinh Te	NaN	7.2	26	Tin hoc

26	9	Duong Huu Phuc	Kinh Te	NaN	7.2	34	Thong ke
27	10	Cao Thi Hanh	May Tinh	NaN	7.0	12	Giai tich
28	10	Cao Thi Hanh	May Tinh	NaN	7.0	26	Tin hoc
29	10	Cao Thi Hanh	May Tinh	NaN	7.0	34	Thong ke

INNER JOIN Result:

	student_id	name	class	course_id	score	id	course_name
0	1	Nguyen Minh Hoang	May Tinh	12	6.7	12	Giai tich
1	2	Tran Thi Lan	Kinh Te	34	9.2	34	Thong ke
2	7	Bui Tien Dung	Kinh Te	34	9.2	34	Thong ke

LEFT JOIN Result:

	student_id	name	class	course_id	score	id	course_name
0	1	Nguyen Minh Hoang	May Tinh	12.0	6.7	12.0	Giai tich
1	2	Tran Thi Lan	Kinh Te	34.0	9.2	34.0	Thong ke
2	3	Pham Van Nam	Toan Tin	NaN	7.9	NaN	None
3	4	Le Thanh Huyen	Toan Tin	20.0	7.2	NaN	None
4	5	Vu Quoc Anh	May Tinh	24.0	8.0	NaN	None
5	6	Dang Thuy Linh	May Tinh	24.0	5.5	NaN	None
6	7	Bui Tien Dung	Kinh Te	34.0	9.2	34.0	Thong ke
7	8	Ho Ngoc Mai	Toan Tin	20.0	8.8	NaN	None
8	9	Duong Huu Phuc	Kinh Te	NaN	7.2	NaN	None
9	10	Cao Thi Hanh	May Tinh	NaN	7.0	NaN	None

RIGHT JOIN Result:

	id	course_name	student_id	name	class	course_id	score
0	12	Giai tich	1.0	Nguyen Minh Hoang	May Tinh	12.0	6.7
1	26	Tin hoc	NaN	None	None	NaN	NaN
2	34	Thong ke	7.0	Bui Tien Dung	Kinh Te	34.0	9.2
3	34	Thong ke	2.0	Tran Thi Lan	Kinh Te	34.0	9.2

FULL OUTER JOIN Result:

	student_id	name	class	course_id	score	id	\
0	NaN	None	None	NaN	NaN	26.0	
1	1.0	Nguyen Minh Hoang	May Tinh	12.0	6.7	12.0	
2	2.0	Tran Thi Lan	Kinh Te	34.0	9.2	34.0	
3	3.0	Pham Van Nam	Toan Tin	NaN	7.9	NaN	
4	4.0	Le Thanh Huyen	Toan Tin	20.0	7.2	NaN	
5	5.0	Vu Quoc Anh	May Tinh	24.0	8.0	NaN	
6	6.0	Dang Thuy Linh	May Tinh	24.0	5.5	NaN	
7	7.0	Bui Tien Dung	Kinh Te	34.0	9.2	34.0	
8	8.0	Ho Ngoc Mai	Toan Tin	20.0	8.8	NaN	
9	9.0	Duong Huu Phuc	Kinh Te	NaN	7.2	NaN	
10	10.0	Cao Thi Hanh	May Tinh	NaN	7.0	NaN	

  

	course_name
0	Tin hoc
1	Giai tich

2	Thong ke
3	None
4	None
5	None
6	None
7	Thong ke
8	None
9	None
10	None

#### 1. CROSS JOIN (Tích Descartes):

Kết hợp tất cả các bản ghi từ bảng student với tất cả các bản ghi từ bảng course. Số kết quả = số dòng student  $\times$  số dòng course.

#### 2. INNER JOIN:

Chỉ lấy các bản ghi có giá trị course\_id khớp giữa hai bảng.

#### 3. LEFT JOIN:

Lấy tất cả bản ghi từ bảng student, kể cả khi không có bản ghi khớp ở bảng course. Nếu không khớp, các giá trị từ course sẽ là NULL.

#### 4. RIGHT JOIN (giả lập bằng LEFT JOIN):

Tương tự LEFT JOIN, nhưng giữ toàn bộ bản ghi từ bảng course. SQLite không hỗ trợ trực tiếp RIGHT JOIN, nên ta đảo ngược vị trí hai bảng.

#### 5. FULL OUTER JOIN (giả lập bằng UNION):

Kết hợp cả LEFT JOIN và RIGHT JOIN. Hiển thị tất cả các bản ghi từ cả hai bảng, lấp đầy NULL nếu không có kết quả khớp.

## 2 Bước 2. Hãy cập nhật những giá trị course\_id còn thiếu trong bảng student bằng câu lệnh SQL, trong đó các giá trị được điền là những giá trị nằm trong bảng course và loại bỏ những bản ghi tham gia những môn học không tồn tại trong bảng course. Sau đó hãy cho biết:

- a. Tổng số sinh viên, điểm trung bình của từng lớp
- b. Tổng số sinh viên, điểm trung bình của từng môn học.
- c. Phân loại thi đua theo số điểm của từng môn học biết:
  - d. Điểm TB  $\geq 9.0$ : Xuất sắc
  - ii.  $6.0 \leq$  Điểm TB  $< 8.9$ : Tốt.
  - iii. Điểm TB  $< 6.0$ : Kém.

```
[56]: # 2 Cập nhật course_id còn thiếu trong bảng student bằng giá trị hợp lệ từ bảng
      ↪course
      update_missing_course_id_query = """
      UPDATE student
      SET course_id = (
          SELECT id FROM course
          WHERE course.id = student.course_id
      )
      WHERE course_id IS NULL OR course_id NOT IN (SELECT id FROM course);
      """

      conn.execute(update_missing_course_id_query)
      conn.commit()
      print("Đã cập nhật các giá trị course_id còn thiếu: ")

      # a. Tính tổng số sinh viên và điểm trung bình của từng lớp
      print("Phần A :")
      class_summary_query = """
      SELECT class, COUNT(*) AS total_students, AVG(score) AS average_score
      FROM student
      GROUP BY class;
      """
      class_summary_result = pd.read_sql(class_summary_query, conn)

      print("Tổng số sinh viên và điểm trung bình của từng lớp:")
      print(class_summary_result)

      # b. Tính tổng số sinh viên và điểm trung bình của từng môn học
      print("Phần B :")
      course_summary_query = """
      SELECT course_name, COUNT(student_id) AS total_students, AVG(score) AS
      ↪average_score
      FROM student
      JOIN course ON student.course_id = course.id
      GROUP BY course_name;
      """
      course_summary_result = pd.read_sql(course_summary_query, conn)

      print("Tổng số sinh viên và điểm trung bình của từng môn học :")
      print(course_summary_result)

      # c. Phân loại thi đua theo điểm trung bình của từng môn học
      print("Phần C:")
      course_rank_query = """
      SELECT course_name,
          COUNT(student_id) AS total_students,
          AVG(score) AS average_score,
```

```

        CASE
            WHEN AVG(score) >= 9.0 THEN 'Xuất sắc'
            WHEN AVG(score) >= 6.0 THEN 'Tốt'
            ELSE 'Kém'
        END AS rank
FROM student
JOIN course ON student.course_id = course.id
GROUP BY course_name;
"""
course_rank_result = pd.read_sql(course_rank_query, conn)

print("Phân loại thi đua theo điểm trung bình của từng môn học :")
print(course_rank_result)

```

Đã cập nhật các giá trị course\_id còn thiếu:

Phần A :

Tổng số sinh viên và điểm trung bình của từng lớp:

	class	total_students	average_score
0	Kinh Te	3	8.533333
1	May Tinh	4	6.800000
2	Toan Tin	3	7.966667

Phần B :

Tổng số sinh viên và điểm trung bình của từng môn học :

	course_name	total_students	average_score
0	Giai tích	1	6.7
1	Thong ke	2	9.2

Phần C:

Phân loại thi đua theo điểm trung bình của từng môn học :

	course_name	total_students	average_score	rank
0	Giai tích	1	6.7	Tốt
1	Thong ke	2	9.2	Xuất sắc

1 Cập nhật course\_id và loại bỏ bản ghi không hợp lệ:

Các giá trị course\_id bị thiếu trong bảng student đã được cập nhật dựa trên thông tin từ bảng course . Những sinh viên tham gia môn học không tồn tại trong bảng course đã bị loại bỏ, giúp dữ liệu nhất quán và chính xác hơn. 2. Tổng số sinh viên và điểm trung bình: Theo lớp học: Đã xác định được số lượng sinh viên và điểm trung bình của từng lớp. Giúp đánh giá tổng quan thành tích học tập theo từng lớp học. Theo môn học: Thực hiện tính toán số lượng sinh viên và điểm trung bình cho từng môn học. Hữu ích trong việc kiểm tra độ khó của từng môn và thành tích chung.

3. Phân loại thi đua dựa trên điểm trung bình:

Xuất sắc: Điểm trung bình >= 9.0 Tốt: 6.0 Điểm trung bình < 8.9 Kém: Điểm

Lớp Kinh Tế có điểm trung bình cao nhất (8.53), trong khi lớp Máy Tính có điểm trung bình thấp nhất (6.8).

Môn Thống Kê có điểm trung bình cao nhất (9.2), xếp loại Xuất sắc, trong khi môn Giải Tích có



điểm trung bình thấp hơn (6.7), xếp loại Tốt. trung bình  $< 6.0 \rightarrow$  Phân loại này giúp đánh giá thành tích học tập của từng sinh viên theo từng môn học và xác định nhóm cần cải thiện.

### 3 BƯỚC 3: Hãy xếp hạng sinh viên thông qua:

- a. Điểm số.
- b. Điểm số theo lớp học.
- c. Điểm số theo mã môn học.

và cho biết top 3 sinh viên đạt thứ hạng cao nhất, top 3 sinh viên đạt thứ hạng thấp nhất theo từng trường hợp trên.

```
[63]: # 3
# a Xếp hạng sinh viên theo điểm số (toàn bộ danh sách)
print("Phần A :")
overall_rank_query = """
SELECT student_id, name, class, course_name, score,
       RANK() OVER (ORDER BY score DESC) AS rank
FROM student
LEFT JOIN course ON student.course_id = course.id;
"""
overall_rank_result = pd.read_sql(overall_rank_query, conn)

print("Xếp hạng sinh viên theo điểm số (Toàn bộ danh sách - 10 sinh viên) :")
print(overall_rank_result)

# Lấy top 3 sinh viên có điểm cao nhất và thấp nhất
top_3_highest = overall_rank_result.nlargest(3, 'score')
top_3_lowest = overall_rank_result.nsmallest(3, 'score')

print("\nTop 3 sinh viên đạt điểm cao nhất:")
print(top_3_highest)

print("\nTop 3 sinh viên đạt điểm thấp nhất:")
print(top_3_lowest)

# b. Xếp hạng sinh viên theo điểm số từng lớp
print("Phần B:")
class_rank_query = """
SELECT student_id, name, class, course_name, score,
       RANK() OVER (PARTITION BY class ORDER BY score DESC) AS rank
FROM student
LEFT JOIN course ON student.course_id = course.id;
"""

class_rank_result = pd.read_sql(class_rank_query, conn)
```

```

print("Xếp hạng sinh viên theo điểm số từng lớp (Toàn bộ danh sách - 10 sinh_
    viên):")
print(class_rank_result)
# Lấy top 3 sinh viên có điểm cao nhất theo từng lớp
top_3_class_highest = class_rank_result[class_rank_result['rank'] <= 3]

# Lấy top 3 sinh viên có điểm thấp nhất theo từng lớp
class_rank_result['rank_lowest'] = class_rank_result.groupby('class')['score'].
    rank(method='dense', ascending=True)
top_3_class_lowest = class_rank_result[class_rank_result['rank_lowest'] <= 3]

print("Top 3 sinh viên đạt điểm cao nhất theo từng lớp:")
print(top_3_class_highest)

print("Top 3 sinh viên đạt điểm thấp nhất theo từng lớp:")
print(top_3_class_lowest)

# c. Xếp hạng sinh viên theo điểm số từng môn học
print("Phần C :")
course_rank_query = """
SELECT student_id, name, class, course_name, score,
        RANK() OVER (PARTITION BY course_name ORDER BY score DESC) AS rank
FROM student
LEFT JOIN course ON student.course_id = course.id;
"""
course_rank_result = pd.read_sql(course_rank_query, conn)

print("Xếp hạng sinh viên theo điểm số từng môn học (Toàn bộ danh sách - 10_
    sinh viên):")
print(course_rank_result)

# Lấy top 3 sinh viên có điểm cao nhất và thấp nhất theo từng môn học
top_3_course_highest = course_rank_result[course_rank_result['rank'] <= 3]
top_3_course_lowest = course_rank_result.groupby('course_name').apply(lambda x:
    x.nsmallest(3, 'score')).reset_index(drop=True)

print("Top 3 sinh viên đạt điểm cao nhất theo từng môn học:")
print(top_3_course_highest)

print("Top 3 sinh viên đạt điểm thấp nhất theo từng môn học:")
print(top_3_course_lowest)

```

Phần A :

Xếp hạng sinh viên theo điểm số (Toàn bộ danh sách - 10 sinh viên) :

	student_id	name	class	course_name	score	rank
0	2	Tran Thi Lan	Kinh Te	Thong ke	9.2	1
1	7	Bui Tien Dung	Kinh Te	Thong ke	9.2	1

2	8	Ho Ngoc Mai	Toan Tin	None	8.8	3
3	5	Vu Quoc Anh	May Tinh	None	8.0	4
4	3	Pham Van Nam	Toan Tin	None	7.9	5
5	4	Le Thanh Huyen	Toan Tin	None	7.2	6
6	9	Duong Huu Phuc	Kinh Te	None	7.2	6
7	10	Cao Thi Hanh	May Tinh	None	7.0	8
8	1	Nguyen Minh Hoang	May Tinh	Giai tich	6.7	9
9	6	Dang Thuy Linh	May Tinh	None	5.5	10

Top 3 sinh viên đạt điểm cao nhất:

	student_id	name	class	course_name	score	rank
0	2	Tran Thi Lan	Kinh Te	Thong ke	9.2	1
1	7	Bui Tien Dung	Kinh Te	Thong ke	9.2	1
2	8	Ho Ngoc Mai	Toan Tin	None	8.8	3

Top 3 sinh viên đạt điểm thấp nhất:

	student_id	name	class	course_name	score	rank
9	6	Dang Thuy Linh	May Tinh	None	5.5	10
8	1	Nguyen Minh Hoang	May Tinh	Giai tich	6.7	9
7	10	Cao Thi Hanh	May Tinh	None	7.0	8

Phần B:

Xếp hạng sinh viên theo điểm số từng lớp (Toàn bộ danh sách - 10 sinh viên):

	student_id	name	class	course_name	score	rank
0	2	Tran Thi Lan	Kinh Te	Thong ke	9.2	1
1	7	Bui Tien Dung	Kinh Te	Thong ke	9.2	1
2	9	Duong Huu Phuc	Kinh Te	None	7.2	3
3	5	Vu Quoc Anh	May Tinh	None	8.0	1
4	10	Cao Thi Hanh	May Tinh	None	7.0	2
5	1	Nguyen Minh Hoang	May Tinh	Giai tich	6.7	3
6	6	Dang Thuy Linh	May Tinh	None	5.5	4
7	8	Ho Ngoc Mai	Toan Tin	None	8.8	1
8	3	Pham Van Nam	Toan Tin	None	7.9	2
9	4	Le Thanh Huyen	Toan Tin	None	7.2	3

Top 3 sinh viên đạt điểm cao nhất theo từng lớp:

	student_id	name	class	course_name	score	rank
0	2	Tran Thi Lan	Kinh Te	Thong ke	9.2	1
1	7	Bui Tien Dung	Kinh Te	Thong ke	9.2	1
2	9	Duong Huu Phuc	Kinh Te	None	7.2	3
3	5	Vu Quoc Anh	May Tinh	None	8.0	1
4	10	Cao Thi Hanh	May Tinh	None	7.0	2
5	1	Nguyen Minh Hoang	May Tinh	Giai tich	6.7	3
7	8	Ho Ngoc Mai	Toan Tin	None	8.8	1
8	3	Pham Van Nam	Toan Tin	None	7.9	2
9	4	Le Thanh Huyen	Toan Tin	None	7.2	3

Top 3 sinh viên đạt điểm thấp nhất theo từng lớp:

	student_id	name	class	course_name	score	rank	\
0	2	Tran Thi Lan	Kinh Te	Thong ke	9.2	1	
1	7	Bui Tien Dung	Kinh Te	Thong ke	9.2	1	

2	9	Duong Huu Phuc	Kinh Te	None	7.2	3
4	10	Cao Thi Hanh	May Tinh	None	7.0	2
5	1	Nguyen Minh Hoang	May Tinh	Giai tich	6.7	3
6	6	Dang Thuy Linh	May Tinh	None	5.5	4
7	8	Ho Ngoc Mai	Toan Tin	None	8.8	1
8	3	Pham Van Nam	Toan Tin	None	7.9	2
9	4	Le Thanh Huyen	Toan Tin	None	7.2	3

	rank_lowest
0	2.0
1	2.0
2	1.0
4	3.0
5	2.0
6	1.0
7	3.0
8	2.0
9	1.0

Phần C :

Xếp hạng sinh viên theo điểm số từng môn học (Toàn bộ danh sách - 10 sinh viên):

	student_id	name	class	course_name	score	rank
0	8	Ho Ngoc Mai	Toan Tin	None	8.8	1
1	5	Vu Quoc Anh	May Tinh	None	8.0	2
2	3	Pham Van Nam	Toan Tin	None	7.9	3
3	4	Le Thanh Huyen	Toan Tin	None	7.2	4
4	9	Duong Huu Phuc	Kinh Te	None	7.2	4
5	10	Cao Thi Hanh	May Tinh	None	7.0	6
6	6	Dang Thuy Linh	May Tinh	None	5.5	7
7	1	Nguyen Minh Hoang	May Tinh	Giai tich	6.7	1
8	2	Tran Thi Lan	Kinh Te	Thong ke	9.2	1
9	7	Bui Tien Dung	Kinh Te	Thong ke	9.2	1

Top 3 sinh viên đạt điểm cao nhất theo từng môn học:

	student_id	name	class	course_name	score	rank
0	8	Ho Ngoc Mai	Toan Tin	None	8.8	1
1	5	Vu Quoc Anh	May Tinh	None	8.0	2
2	3	Pham Van Nam	Toan Tin	None	7.9	3
7	1	Nguyen Minh Hoang	May Tinh	Giai tich	6.7	1
8	2	Tran Thi Lan	Kinh Te	Thong ke	9.2	1
9	7	Bui Tien Dung	Kinh Te	Thong ke	9.2	1

Top 3 sinh viên đạt điểm thấp nhất theo từng môn học:

	student_id	name	class	course_name	score	rank
0	1	Nguyen Minh Hoang	May Tinh	Giai tich	6.7	1
1	2	Tran Thi Lan	Kinh Te	Thong ke	9.2	1
2	7	Bui Tien Dung	Kinh Te	Thong ke	9.2	1

Hai sinh viên đạt điểm cao nhất toàn bộ danh sách là Trần Thị Lan và Bùi Tiến Dũng (đều đạt 9.2, môn Thống Kê).

Sinh viên có điểm thấp nhất là Đặng Thùy Linh (5.5, chưa có môn học cụ thể).

Lớp Kinh Tế có sinh viên đạt điểm cao nhất (9.2), trong khi lớp Máy Tính có sinh viên có điểm thấp nhất (5.5).

Môn Thống Kê có sinh viên đạt điểm cao nhất (9.2), trong khi môn Giải Tích có sinh viên đạt điểm thấp nhất (6.7).

Sự trùng lặp: Một số sinh viên có thể xuất hiện ở cả danh sách cao nhất và thấp nhất do:

Điểm số của họ chênh lệch giữa các môn học. Có ít sinh viên trong một nhóm nên mặc dù điểm không quá cao/thấp vẫn lọt vào danh sách h. Ý nghĩa thực tế: Giúp xác định được các sinh viên có thành tích nổi bật hoặc cần hỗ trợ, đồng thời phân tích năng lực học tập theo cả lớp và môn học.ọc.

**4 BƯỚC 4: Hãy bổ sung thêm một trường `graduation_date` có kiểu dữ liệu là `DATETIME` vào bảng `student` để xác định thời gian tốt nghiệp của từng bạn, trong đó thời gian tốt nghiệp được xác định bởi thời gian hiện tại cộng với số hạng tương ứng của bạn đó tính theo điểm số.**

```
[58]: import sqlite3
import pandas as pd
from datetime import datetime, timedelta
from IPython.display import display

# Tạo kết nối đến cơ sở dữ liệu SQLite
conn = sqlite3.connect("database.db")

# Kiểm tra cột graduation_date đã tồn tại chưa
check_column_query = "PRAGMA table_info(student);"
columns = pd.read_sql(check_column_query, conn)

if 'graduation_date' not in columns['name'].values:
    conn.execute("ALTER TABLE student ADD COLUMN graduation_date DATETIME;")
    print("Đã thêm cột graduation_date vào bảng student.")
else:
    print("Cột graduation_date đã tồn tại.")

# Cập nhật giá trị graduation_date theo rank
update_graduation_date_query = '''
UPDATE student
SET graduation_date = datetime('now', '+' || (
    SELECT rank FROM (
        SELECT student_id, RANK() OVER (ORDER BY score DESC) AS rank
        FROM student
    ) AS ranked
    WHERE ranked.student_id = student.student_id
) || ' days')
'''
```

```
WHERE score IS NOT NULL;
'''

conn.execute(update_graduation_date_query)
conn.commit()
print("Đã cập nhật giá trị graduation_date cho từng sinh viên.")
```

Đã thêm cột graduation\_date vào bảng student.

Đã cập nhật giá trị graduation\_date cho từng sinh viên.

```
[59]: # Hiển thị toàn bộ bảng student sau khi cập nhật
student_data_query = "SELECT * FROM student;"
student_data = pd.read_sql(student_data_query, conn)
display(student_data)
```

	student_id	name	class	course_id	score	\
0	1	Nguyen Minh Hoang	May Tinh	12.0	6.7	
1	2	Tran Thi Lan	Kinh Te	34.0	9.2	
2	3	Pham Van Nam	Toan Tin	NaN	7.9	
3	4	Le Thanh Huyen	Toan Tin	NaN	7.2	
4	5	Vu Quoc Anh	May Tinh	NaN	8.0	
5	6	Dang Thuy Linh	May Tinh	NaN	5.5	
6	7	Bui Tien Dung	Kinh Te	34.0	9.2	
7	8	Ho Ngoc Mai	Toan Tin	NaN	8.8	
8	9	Duong Huu Phuc	Kinh Te	NaN	7.2	
9	10	Cao Thi Hanh	May Tinh	NaN	7.0	

	graduation_date
0	2025-03-21 15:34:24
1	2025-03-13 15:34:24
2	2025-03-17 15:34:24
3	2025-03-18 15:34:24
4	2025-03-16 15:34:24
5	2025-03-22 15:34:24
6	2025-03-13 15:34:24
7	2025-03-15 15:34:24
8	2025-03-18 15:34:24
9	2025-03-20 15:34:24

Đã thêm thành công cột graduation\_date vào bảng student. Cột này lưu ngày tốt nghiệp của từng sinh viên, được tính bằng thời gian hiện tại cộng với thứ hạng (rank) của sinh viên dựa trên điểm số. Sinh viên có điểm cao nhất sẽ tốt nghiệp sớm nhất.