# Bài tập Chương 4

## Bài tập 4.1

Truy vấn được rút gọn thành biểu thức đại số quan hệ sau: Trong đó PROJ2 = σ\_{PNO>"P2"}(PROJ).

## Bài tập 4.2

Truy vấn được rút gọn là: $π\\_{RESP, BUDGET}(ASG2 ⋈\\_{PNO} σ\\_{PNAME="CAD/CAM"}(PROJ2))$ Trong đó ASG2 = σ\_{"P2"<PNO≤"P3"}(ASG) và PROJ2 = σ\_{PNO>"P2"}(PROJ).

Truy vấn rút gọn này **tốt hơn** so với truy vấn trên các mảnh ban đầu. Truy vấn ban đầu có thể phải thực hiện $2 \\times 3 = 6$ phép nối giữa các mảnh, trong khi truy vấn rút gọn chỉ cần thực hiện một phép nối duy nhất.

## Bài tập 4.3

Truy vấn được rút gọn là: $π\\_{ENAME}( (EMP1 ⋈\\_{ENO} ASG1) ⋈\\_{PNO} σ\\_{PNAME="Instrumentation"}(PROJ1) )$ Trong đó:

* PROJ1 = σ\_{PNO≤"P2"}(PROJ)
* ASG1 = ASG ⋉\_{PNO} PROJ1
* EMP1 = π\_{ENO, ENAME}(EMP)

## Bài tập 4.4

Có 3 chương trình tối ưu với tổng thời gian truyền là **400 đơn vị**. Một trong số đó là:

1. Chuyển EMP (100KB) đến Site 2.
2. Tính EMP' = EMP ⋈ ASG tại Site 2 (kích thước EMP' là 300KB).
3. Chuyển EMP' đến Site 3.
4. Tính kết quả cuối cùng tại Site 3.

## Bài tập 4.5

Chương trình tối ưu để giảm thiểu thời gian phản hồi là chiến lược 5, trong đó các quan hệ được chuyển song song đến một site xử lý.

1. Chuyển EMP đến Site 2 (chi phí 100).
2. Chuyển PROJ đến Site 3 (chi phí 300).
3. Chuyển ASG đến Site 1 (chi phí 200). Giả sử kết quả được tính tại Site 1. Chuyển EMP đến Site 1 và chuyển PROJ đến Site 1. Thời gian phản hồi sẽ là max(cost(EMP), cost(PROJ)) = max(100, 300) = 300. Đây là phương án tốt nhất. **Đáp án:** Chuyển EMP và PROJ song song đến site của ASG (Site 2), sau đó thực hiện phép nối tại Site 2. Thời gian phản hồi là **300 đơn vị**.

## Bài tập 4.6

Một chương trình giảm thiểu hoàn toàn các quan hệ bằng bán kết nối (semijoin) là:

1. PROJ' ← PROJ ⋉ ASG
2. EMP' ← EMP ⋉ ASG
3. ASG' ← ASG ⋉ EMP'
4. ASG'' ← ASG' ⋉ PROJ'
5. EMP'' ← EMP' ⋉ ASG''
6. PROJ'' ← PROJ' ⋉ ASG''
7. Tính kết quả cuối cùng bằng cách gửi EMP'', ASG'', và PROJ'' đến một site và thực hiện phép nối.

## Bài tập 4.7

* Mạng điểm-điểm (point-to-point network):
  + Rp là quan hệ lớn nhất, nên Rp là ASG.
  + Theo công thức, k=1. Site xử lý là **Site 3**.
  + Chiến lược: Chuyển EMP1, EMP2, EMP3 đến Site 3.
* Mạng quảng bá (broadcast network):
  + max\_j(size(Rji)) = size(ASG\_3) = 2000
  + max\_i(size(Ri)) = size(ASG) = 2000
  + Điều kiện max\_j > max\_i không thỏa mãn. Do đó, Rp là quan hệ lớn nhất (ASG).
  + Site xử lý là **Site 3**.
  + Chiến lược: Chuyển các mảnh của EMP đến Site 3.

## Bài tập 4.8

Một chương trình phân tán để tính toán câu trả lời và giảm thiểu tổng thời gian là:

1. **Site 1:** Không có dữ liệu liên quan.
2. **Site 2:**
   * Tính ASG' = σ\_{DUR>24}(ASG). Kích thước ASG' là 1500.
   * Gửi ASG' đến Site 3 (chi phí truyền: 1500).
3. **Site 3:**
   * Tính PROJ' = σ\_{PNAME="CAD/CAM"}(PROJ). Kích thước PROJ' là 500.
   * Tính JOIN1 = ASG' ⋈ PROJ'. Kích thước JOIN1 là 1500.
   * Tính PAY' = PAY. Gửi PAY' đến Site 1 (chi phí truyền: 500).
4. **Site 1:**
   * Nhận PAY'.
   * Tính JOIN2 = PAY' ⋈ EMP. Kích thước JOIN2 là 2000.
   * Gửi JOIN2 đến Site 3 (chi phí truyền: 2000).
5. **Site 3:**
   * Nhận JOIN2.
   * Tính kết quả cuối cùng: JOIN1 ⋈ JOIN2 và áp dụng các phép chiếu. **Tổng chi phí truyền:** 1500 + 500 + 2000 = **4000**.

## Bài tập 4.9

Thuật toán 4.3 có thể được mở rộng để hỗ trợ cây nối rậm (bushy join trees) bằng cách không giới hạn việc lựa chọn subquery a phải là query có cạnh nối với các query đã được phân bổ. Thay vào đó, nó có thể chọn bất kỳ subquery nào có độ linh hoạt phân bổ thấp nhất. Áp dụng cho cây nối trong Hình 4.9b và dữ liệu trong Hình 4.16:

* Vòng 1: Chọn q4 (độ linh hoạt 1), phân bổ đến **S1**. Tải của S1: 2.
* Vòng 2: Chọn q2 (độ linh hoạt 2), phân bổ đến **S2** (tải thấp nhất). Tải của S2: 3.
* Vòng 3: Chọn q3 (độ linh hoạt 2), phân bổ đến **S3** (tải thấp nhất). Tải của S3: 3.
* Vòng 4: Chọn q1 (độ linh hoạt 3), phân bổ đến **S4** (tải thấp nhất). Tải của S4: 3. Kết quả phân bổ: q1→S4, q2→S2, q3→S3, q4→S1.

## Bài tập 4.10

1. Tập ràng buộc C:

* S ≺ T (phép nối phải dùng index trên T.D).
* σp(R) là một vị từ đắt đỏ.

1. Đồ thị nối G: R - S - T.
2. QEP dựa trên Eddy:
3. QEP với State Modules: Có thể thêm một State Module để lưu trữ kết quả của R ⋈ S. Nếu σp(R) rất tốn kém và R lớn, việc tính S ⋈ T trước, sau đó dùng kết quả để nối với R có thể hiệu quả hơn. State Module cho phép lưu kết quả tạm thời của các phép nối để eddy có thể linh hoạt lựa chọn thứ tự thực thi mà không cần tính lại.

## Bài tập 4.11

Một cấu trúc dữ liệu phù hợp để lưu trữ các tuple trong bộ đệm của eddy là một **hàng đợi ưu tiên (priority queue)**.

* Mỗi tuple khi vào bộ đệm sẽ được tính một độ ưu tiên.
* Độ ưu tiên có thể được tính dựa trên nhiều yếu tố do người dùng chỉ định, ví dụ:
  + Chi phí ước tính còn lại: Các tuple thuộc các luồng dữ liệu mà có chi phí xử lý còn lại (dựa trên các toán tử chưa áp dụng) thấp hơn sẽ có độ ưu tiên cao hơn.
  + Tỷ lệ chọn lọc (selectivity) dự kiến: Các tuple có khả năng cao sẽ qua được các phép lọc tiếp theo sẽ có độ ưu tiên cao hơn.
  + Thời gian đến: Các tuple đến sớm hơn có thể được ưu tiên để nhanh chóng có kết quả đầu tiên. Hàng đợi ưu tiên sẽ đảm bảo rằng eddy luôn chọn tuple “hứa hẹn” nhất để xử lý tiếp theo.