# Bài tập Chương 7

## Bài tập 7.1

Các ứng dụng phù hợp cho Hệ CSDL Phân tán (DDBMS - top-down):

1. Hệ thống Ngân hàng Toàn cầu: Dữ liệu khách hàng và tài khoản được phân tán chiến lược đến các chi nhánh khu vực để tăng tốc độ truy cập cục bộ, nhưng toàn bộ hệ thống được thiết kế và quản lý như một thực thể thống nhất.
2. Mạng lưới Bán lẻ Lớn: Dữ liệu kho hàng và bán hàng được phân tán đến từng cửa hàng hoặc trung tâm phân phối, nhưng được quản lý tập trung để theo dõi tồn kho và logistics trên toàn hệ thống.
3. Hệ thống Đặt vé Hàng không: Dữ liệu về chuyến bay, chỗ ngồi và hành khách được phân tán trên các máy chủ toàn cầu để đảm bảo tính sẵn sàng cao và thời gian phản hồi nhanh, được kiểm soát bởi một lược đồ toàn cục duy nhất.

Các ứng dụng phù hợp cho Hệ Đa CSDL Phân tán (MDBS - bottom-up):

1. Cổng thông tin Y tế Tích hợp: Tích hợp dữ liệu bệnh nhân từ nhiều bệnh viện, phòng khám độc lập, mỗi nơi có hệ thống CSDL riêng, để cung cấp một cái nhìn tổng thể về lịch sử y tế.
2. Hệ thống Thông tin Sinh viên của một Đại học: Cung cấp một giao diện duy nhất để truy cập dữ liệu từ các CSDL riêng biệt và tự trị của thư viện, phòng đào tạo, và phòng công tác sinh viên.
3. Cơ quan Chính phủ Tích hợp Dữ liệu: Kết hợp dữ liệu từ các cơ quan khác nhau như thuế, dân số, và giao thông, mỗi cơ quan đã có sẵn hệ thống CSDL của riêng mình.

## Bài tập 7.2

Việc có một lược đồ khái niệm toàn cục (Global Conceptual Schema - GCS) có cả ưu và nhược điểm.

Ưu điểm (Ủng hộ việc có GCS):

* Cung cấp tính trong suốt về vị trí cho người dùng và ứng dụng.
* Đơn giản hóa việc viết các truy vấn toàn cục.
* Tạo cơ sở cho việc tối ưu hóa truy vấn toàn cục.
* Cho phép định nghĩa các ràng buộc toàn vẹn trên toàn hệ thống.

Nhược điểm (Phản đối việc có GCS):

* Khó khăn và tốn kém để thiết kế và duy trì, đặc biệt khi các CSDL địa phương thay đổi.
* Hạn chế tính tự trị (autonomy) của các hệ thống CSDL thành phần.
* Có thể không khả thi trong các môi trường quy mô rất lớn hoặc cực kỳ năng động (ví dụ: mạng P2P).
* Điểm trung tâm có thể trở thành nút thắt cổ chai về hiệu năng và quản trị.

## Bài tập 7.3

Thuật toán chuyển đổi lược đồ quan hệ sang mô hình Thực thể-Kết hợp (E-R):

1. Xác định các Tập thực thể: Mỗi quan hệ trong lược đồ quan hệ mà không phải là một quan hệ kết hợp (quan hệ chỉ chứa các khóa ngoại) được chuyển đổi thành một tập thực thể. Các thuộc tính của quan hệ trở thành thuộc tính của tập thực thể.
2. Xác định các Thuộc tính khóa: Khóa chính của mỗi quan hệ được xác định là khóa của tập thực thể tương ứng.
3. Xác định các Tập kết hợp (Relationship Sets):
   * Các quan hệ kết hợp (biểu diễn mối quan hệ nhiều-nhiều) được chuyển đổi thành các tập kết hợp.
   * Các ràng buộc khóa ngoại (foreign key) được chuyển đổi thành các tập kết hợp (thường là một-nhiều hoặc một-một).
4. Xác định Bản số (Cardinality): Dựa vào các ràng buộc khóa và khóa ngoại để xác định bản số của các tập kết hợp (1:1, 1:N, M:N).

## Bài tập 7.4

Lược đồ E-R toàn cục tích hợp (mô tả bằng văn bản):

* Các tập thực thể:
  + Person: Một tập thực thể tổng quát hóa với các thuộc tính chung như Name, Address. Có các tập thực thể chuyên biệt hóa là Racer, Salesperson, Director, Contact.
  + Organization: Một tập thực thể tổng quát hóa với thuộc tính Name, Address. Có các tập thực thể chuyên biệt hóa là Sponsor và Manufacturer.
  + Government\_Body: Với các thuộc tính Dept\_Name, Issuer.
  + Event: Một tập thực thể tổng quát hóa cho RACE (cuộc đua).
  + Product: Một tập thực thể cho SHOES.
* Các tập kết hợp:
  + Organizes (N:1): Giữa Director và Event(RACE).
  + Funds (M:N): Giữa Organization(Sponsor) và Event(RACE).
  + Participates\_In (M:N): Giữa Person(Racer) và Event(RACE).
  + Requires\_License (1:1): Giữa Event(RACE) và License. License được cấp bởi Government\_Body.
  + Makes (M:N): Giữa Organization(Manufacturer) và Product(SHOES).
  + Distributes (M:N): Giữa Distributor và Product(SHOES).
  + Employs (1:N): Giữa Distributor và Person(Salesperson).

## Bài tập 7.5

1. Ánh xạ LAV (Local-as-View):

Area(Id, Field) :- Works(Id, P), Area(P, Field).

1. Ánh xạ GLAV (Global-Local-as-View):

Works(Id, P), Area(P, Field) :- Teach(Id, C), In(C, Field). // Ánh xạ DB2  
Works(Id, P), Area(P, Field) :- Grant(Id, G), For(G, Field). // Ánh xạ DB3

1. Ánh xạ GAV (Global-as-View):

Works(Id, P) :- Grant(Id, G), Funds(G, P).  
Area(P, Field) :- Funds(G, P), For(G, Field).

## Bài tập 7.6

1. SoccerPlayer, Actor, Politician.
2. Actor, Politician.
3. SoccerPlayer, Politician.
4. SoccerPlayer, Actor.
5. SoccerPlayer, Politician.

## Bài tập 7.7

1. Tính đầy đủ của kết quả:
   1. Không đầy đủ: Thiếu các quốc gia không thuộc châu Âu và có dân số dưới 30 triệu.
   2. Đầy đủ: Mọi quốc gia có dân số > 40 triệu đều thuộc BigCountry.
   3. Không đầy đủ: Thiếu các quốc gia có dân số từ 20-30 triệu.
2. Các nguồn cục bộ cần thiết:
   1. EuropeanCountry, BigCountry, MidsizeOceanCountry.
   2. EuropeanCountry.
   3. EuropeanCountry, MidsizeOceanCountry.
   4. BigCountry, MidsizeOceanCountry.

## Bài tập 7.8

1. Phương pháp xác định tương ứng:
   1. Id - Key: So sánh kiểu dữ liệu (số nguyên vs. chuỗi), phân tích dữ liệu thực tế.
   2. Name - Title: So sánh tên bằng bảng từ đồng nghĩa.
   3. DeliveryPrice - Price: So sánh tên (có từ “Price” chung), so sánh kiểu dữ liệu (float, real).
   4. Description - Information: So sánh tên bằng bảng từ đồng nghĩa.
2. Tương ứng sai có thể xảy ra: Có. Ví dụ, nếu có một thuộc tính khác tên là ProductID trong ARTICLE cũng có kiểu varchar(255), phương pháp so sánh kiểu dữ liệu có thể xác định sai cặp Id - ProductID thay vì Id - Key.

## Bài tập 7.9

Kết quả so khớp lược đồ tổng thể là:

* S.a tương ứng với T.d (độ tương đồng 0.8)
* S.b tương ứng với T.f (độ tương đồng 0.9)
* S.c tương ứng với T.e (độ tương đồng 0.7)

## Bài tập 7.10

1. Các tương ứng so khớp lược đồ (ví dụ):

* (Attribute-Attribute, 1:1): MyGroup.Publication.Title - MyConference.Paper.Title
* (Attribute-Attribute, 1:1): MyGroup.Publication.Year - MyConference.ConfWorkshop.Year
* (Attribute-Relation, 1:N): MyConference.Paper.Authors - MyPublisher.Person
* (Relation-Relation, 1:1): MyGroup.GroupMember - MyPublisher.Person
* (Relation-Relation, M:N): MyGroup.Publication - MyConference.Paper

1. Phân loại đã được chỉ ra ở trên.
2. Lược đồ toàn cục hợp nhất (ví dụ):

* PUBLICATION(PubID, Title, Year, Type)
* PERSON(PersonID, FirstName, LastName, Email, Affiliation)
* VENUE(VenueID, Name, Location)
* AUTHORED\_BY(PubID, PersonID, Position)
* PUBLISHED\_IN(PubID, VenueID)

## Bài tập 7.11

1. Meaningful: Có. Nối Course và Tutor, ghép họ và tên. Complete: Không. Chỉ trả về các khóa học có gia sư. Các khóa học không có gia sư hoặc các gia sư không dạy khóa nào sẽ bị bỏ qua. Key Violations: Không.
2. Meaningful: Có, nhưng khó hiểu. Tạo ra các tuple null. Complete: Có. FULL OUTER JOIN đảm bảo tất cả các tuple từ cả hai bảng đều được bao gồm. Key Violations: Có thể. T.id có thể là NULL, vi phạm ràng buộc khóa chính của Lecture.
3. Meaningful: Có. Complete: Có. Key Violations: Có thể. Nếu C.id là NULL, nó vi phạm khóa chính của Lecture.

## Bài tập 7.12

1. LAV: AREA(ID, FIELD) :- WORKS(ID, P), AREA(P, FIELD).
2. GLAV: WORKS(ID, P), AREA(P, FIELD) :- TEACH(ID, C), IN(C, FIELD). WORKS(ID, P), AREA(P, FIELD) :- GRANT(ID, G), FOR(G, FIELD).
3. GAV: WORKS(ID, P) :- GRANT(ID, G), FUNDS(G, P). AREA(P, FIELD) :- FUNDS(G, P), FOR(G, FIELD).

## Bài tập 7.13

Logic (cụ thể là logic bậc nhất, ví dụ Datalog) hữu ích cho việc dịch và tích hợp lược đồ vì:

* Hình thức hóa Thống nhất: Cung cấp một ngôn ngữ hình thức, rõ ràng để biểu diễn cả lược đồ và các ánh xạ, loại bỏ sự mơ hồ.
* Định nghĩa View: Rất phù hợp để định nghĩa các ánh xạ GAV, LAV, GLAV dưới dạng các quy tắc logic (view).
* Suy luận (Reasoning): Cho phép hệ thống suy luận ra các ánh xạ mới, kiểm tra tính nhất quán của các ánh xạ, và phát hiện các thông tin dư thừa.
* Viết lại Truy vấn (Query Rewriting): Các thuật toán viết lại truy vấn (ví dụ, unfolding, bucket) được xây dựng một cách tự nhiên dựa trên nền tảng logic.

## Bài tập 7.14

Tối ưu hóa toàn cục trên các truy vấn toàn cục trong hệ thống đa CSDL là **có thể thực hiện được, nhưng rất hạn chế**.

Điều kiện để có thể tối ưu hóa:

* Thông tin chi phí (Cost Information): Hệ thống đa CSDL phải có khả năng ước tính chi phí thực thi các truy vấn con tại các CSDL thành phần. Điều này rất khó do tính tự trị của các CSDL thành phần (vấn đề “black-box”).
* Khả năng xử lý của CSDL thành phần: Hệ thống cần biết các toán tử (ví dụ: join, aggregate) mà mỗi CSDL thành phần hỗ trợ để quyết định nên đẩy (push down) phần xử lý nào xuống.
* Thống kê dữ liệu: Cần có các thông tin thống kê (ví dụ: số lượng tuple, phân bố giá trị) từ các CSDL thành phần.

Tối ưu hóa thường chỉ giới hạn ở việc sắp xếp thứ tự các phép nối được thực hiện tại tầng mediator và quyết định chiến lược di chuyển dữ liệu (data shipping), thay vì tối ưu hóa sâu như trong một DDBMS đồng nhất.

## Bài tập 7.15

1. GAV Rewriting: Phụ thuộc vào định nghĩa GAV. Giả sử GCS EMP(ENAME, TITLE, CITY) được định nghĩa là EMP1 ∪ EMP2, truy vấn sẽ được “unfold” thành một phép hợp của các truy vấn con trên các quan hệ cục bộ.
2. LAV Rewriting (Bucket Algorithm): Sẽ tạo các “bucket” cho mỗi subgoal trong truy vấn toàn cục và điền vào đó các view (quan hệ cục bộ) có thể trả lời subgoal đó. Sau đó, kết hợp các view từ các bucket để tạo ra các truy vấn viết lại.
3. LAV Rewriting (MinCon Algorithm): Tương tự Bucket, nhưng hiệu quả hơn vì nó xem xét các subgoal trong truy vấn một cách tổng thể, tạo ra các MinCon Description (MCD) để giảm số lượng các kết hợp cần xem xét.

## Bài tập 7.16

1. Chi phí các kế hoạch (sử dụng mô hình chi phí chung):

* cost(P1) = cost(EMP) + |EMP| \* cost(σ\_{ENO=v}(ASG)) = 100 + 100 \* 2000 = 200,100
* cost(P2) = cost(EMP) + |EMP| \* cost(σ\_{DUR>36}(ASG)) = 100 + 100 \* (0.01 \* 2000) = 100 + 2000 = 2,100
* cost(P3) = cost(σ\_{DUR>36}(ASG)) + |σ\_{DUR>36}(ASG)| \* cost(σ\_{ENO=v}(EMP)) = (0.01 \* 2000) + 20 \* 100 = 20 + 2000 = 2,020
* cost(P4) = cost(σ\_{DUR>36}(ASG)) + |σ\_{DUR>36}(ASG)| \* cost(EMP) = 20 + 20 \* 100 = 2,020

1. Kế hoạch có chi phí thấp nhất là **P3** hoặc **P4** với chi phí **2,020**.

## Bài tập 7.17

1. Chi phí các kế hoạch (sử dụng mô hình chi phí tùy chỉnh):

* cost(P1) = |EMP| + |EMP| \* |σ\_{ENO=v}(ASG)| = 100 + 100 \* 1 = 200 (giả sử 1 tuple ASG cho mỗi EMP)
* cost(P2) = 2,100 (như trên)
* cost(P3) = |σ\_{DUR>36}(ASG)| + |σ\_{DUR>36}(ASG)| \* |σ\_{ENO=v}(EMP)| = 20 + 20 \* 1 = 40
* cost(P4) = 2,020 (như trên)

1. Kế hoạch có chi phí thấp nhất là **P3** với chi phí **40**.

## Bài tập 7.18

| Tiêu chí | Query-based | Operator-based |
| --- | --- | --- |
| **Tính biểu cảm của truy vấn** | Hạn chế bởi ngôn ngữ chung (ví dụ: SQL subset). | Linh hoạt hơn, có thể khai thác các toán tử độc đáo của nguồn. |
| **Hiệu năng truy vấn** | Có thể không tối ưu nếu wrapper phải giả lập các toán tử phức tạp. | Tốt hơn, vì mediator có thể tối ưu hóa dựa trên các toán tử gốc. |
| **Chi phí phát triển Wrapper** | Cao hơn, vì wrapper phải hỗ trợ toàn bộ giao diện truy vấn chung. | Thấp hơn, wrapper chỉ cần export các toán tử mà nguồn hỗ trợ. |
| **Bảo trì hệ thống** | Dễ hơn, vì giao diện chung ổn định. | Khó hơn, mediator phải thay đổi nếu toán tử của nguồn thay đổi. |
| **Sự phát triển (Evolution)** | Khó tích hợp các tính năng mới của nguồn. | Dễ hơn, chỉ cần export toán tử mới. |

## Bài tập 7.19

1. Hàm lập kế hoạch (Planning functions) của wrapper w3:

* accessPlan(R: relation, ...): Tạo toán tử scan cho EMP hoặc ASG tại db4.
* joinPlan(R1: EMP, R2: ASG, ...): Tạo toán tử join được thực hiện cục bộ tại db4.

1. Định nghĩa mới của view toàn cục EMPASG:

EMPASG = (db1.EMP ⋈ db2.ASG) ∪ db3.EMPASG ∪ (db1.EMP ⋈ db4.ASG) ∪ (db4.EMP ⋈ db2.ASG) ∪ (db4.EMP ⋈ db4.ASG)

1. QEP cho truy vấn: QEP sẽ là một phép hợp (Union) của các kế hoạch con. Một kế hoạch con sẽ là Fetch(CITY="Paris") trên db3.EMPASG. Một kế hoạch con khác sẽ là phép nối giữa Scan(CITY="Paris") trên db1.EMP và Scan trên db2.ASG, được thực hiện tại mediator hoặc một wrapper. Các kế hoạch con khác sẽ liên quan đến db4. Tất cả sau đó được lọc theo DUR > 24.