

NỘI DUNG HƯỚNG DẪN THỰC HÀNH MÔN PTTK HTTT

I. MÔ HÌNH QUAN NIỆM DỮ LIỆU - CONCEPTUAL DATA MODEL(CDM)

Bước đầu tiên trong quá trình tạo mô hình quan niệm là xác định định rõ các yêu cầu của nghiệp vụ và để làm được điều này chúng ta lên mô hình mô tả các hoạt động của nghiệp vụ. Điều này giúp chúng ta quyết định:

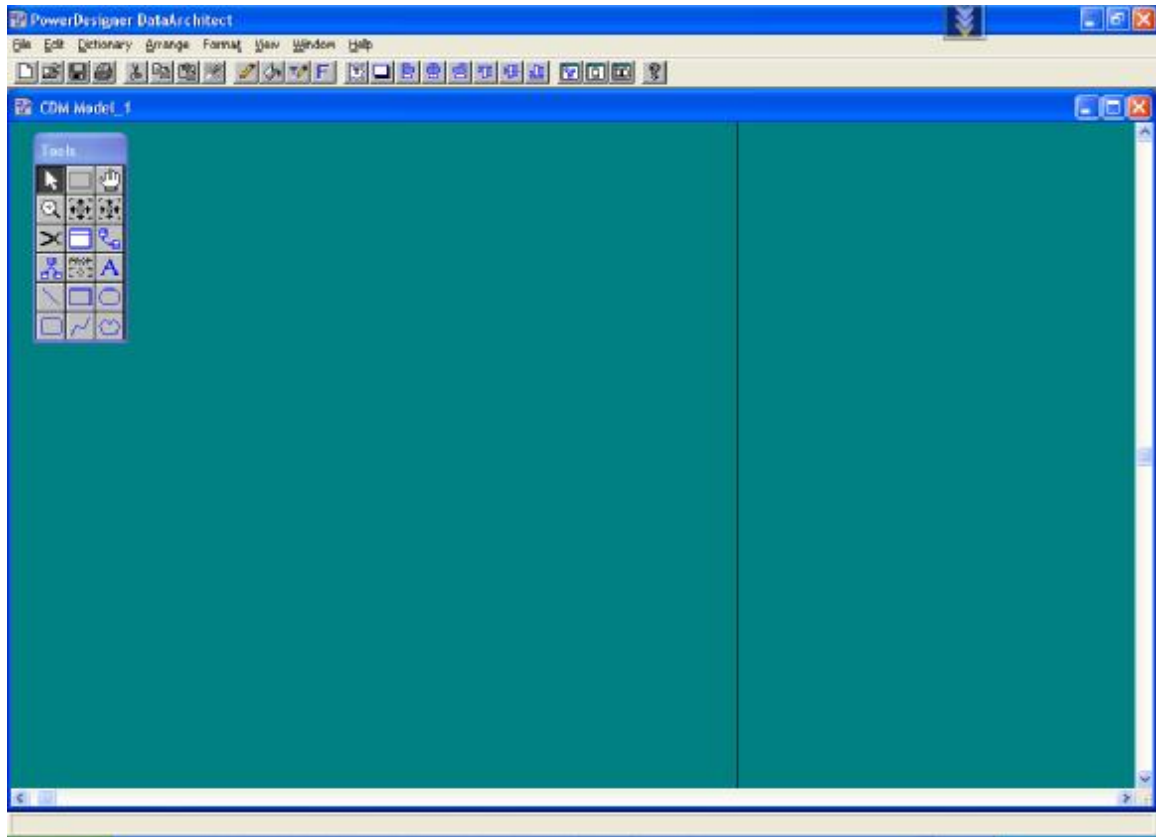
- Những thông tin gì chúng ta cần lưu trữ.
- Những thực thể nào chúng ta cần gắn vào cho từng nghiệp vụ.
- Hoạt động của từng nghiệp vụ như thế nào.

Khi ta biết rõ hoạt động của từng nghiệp vụ như thế nào thì ta dễ dàng xây dựng mô hình CDM.

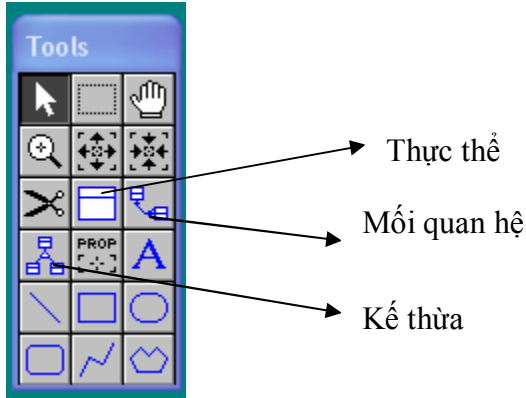
Cách sử dụng mô hình CDM trong Power Designer:

1. Khởi động Power Designer

Start/Programes/ Power Designer 6 32-bit/ DataArchitect

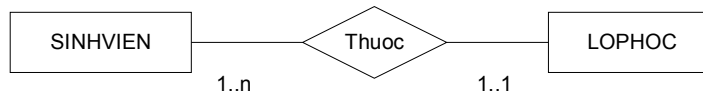


2. Hướng dẫn sử dụng thanh công cụ



3. Mối kết hợp giữa thực thể và thực thể

❖ Giả sử ta có mối kết hợp giữa hai thực thể sau:



SINHVIEN

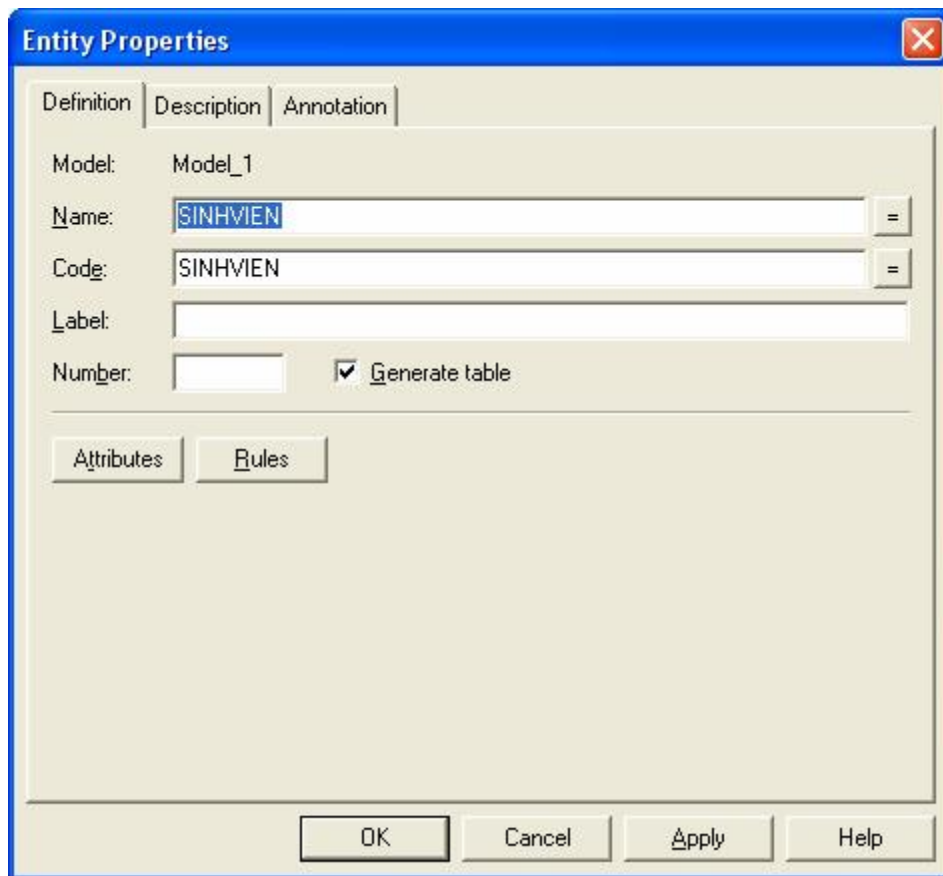
| Thuộc tính | Kiểu dữ liệu | Kích thước | Ghi chú |
|------------|--------------|------------|-----------------|
| MASV | Character | 10 | Thuộc tính khóa |
| TENSV | Character | 30 | |
| NGAYSINH | Datetime | | |
| DIACHI | Character | 30 | |

LOPHOC

| Thuộc tính | Kiểu dữ liệu | Kích thước | Ghi chú |
|------------|--------------|------------|-----------------|
| MALOP | Character | 10 | Thuộc tính khóa |
| TENLOP | Character | 30 | |
| SISO | Integer | | |

❖ *Hướng dẫn cách tạo:*

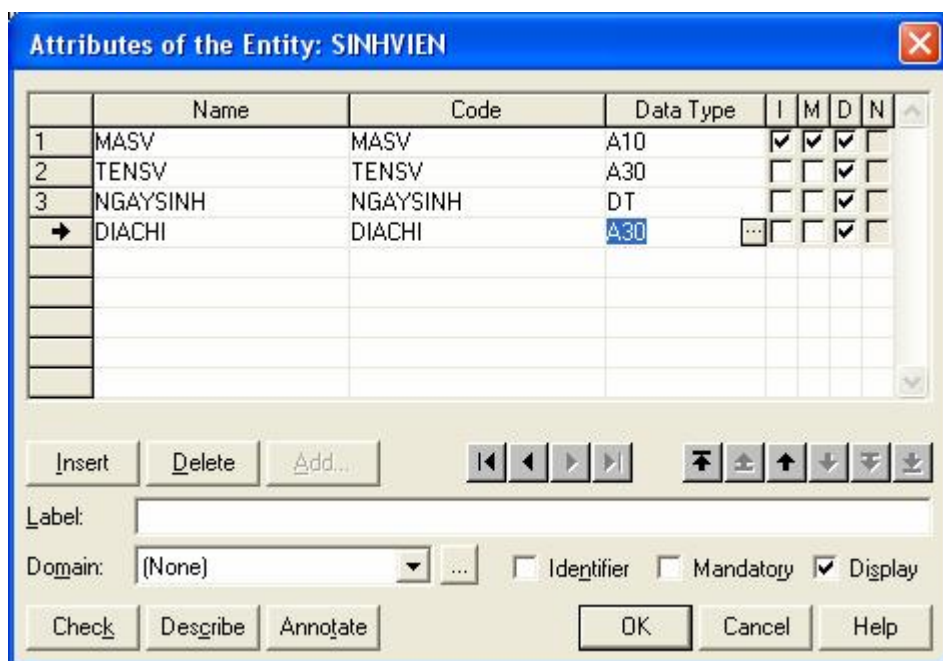
- Chọn biểu tượng thực thể, giữ chuột và kéo thả vào trong lược đồ
- Để thay đổi thông tin của thực thể ta double-click vào thực thể cần thay đổi, một cửa sổ mới mở ra cho phép chúng ta chỉnh sửa thông tin của thực thể như: tên của thực thể, thuộc tính của thực thể, các rule,....



The 'Entity Properties' dialog box is shown with the 'Definition' tab selected. It contains the following fields and options:

- Model:** Model_1
- Name:** SINHVIEN
- Code:** SINHVIEN
- Label:** (empty)
- Number:** (empty) ☒ Generate table
- Buttons:** Attributes, Rules, OK, Cancel, Apply, Help

- Để tạo thuộc tính của thực thể sinh viên, ta nhấn vào Attributes



The 'Attributes of the Entity: SINHVIEN' dialog box is shown. It contains a table with the following data:

| | Name | Code | Data Type | I | M | D | N |
|---|----------|----------|-----------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|
| 1 | MASV | MASV | A10 | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| 2 | TENSV | TENSV | A30 | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| 3 | NGAYSINH | NGAYSINH | DT | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| → | DIACHI | DIACHI | A30 | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |

Below the table are the following controls:

- Buttons:** Insert, Delete, Add...
- Navigation:** First, Previous, Next, Last, Up, Down, Add, Remove
- Label:** (empty)
- Domain:** (None)
- Options:** ☐ Identifier, ☐ Mandatory, ☒ Display
- Buttons:** Check, Describe, Annotate, OK, Cancel, Help

- Name: cho phép nhập tên của thuộc tính của thực thể
- Code: phát sinh một mã tương ứng một thuộc tính (không cần quan tâm chỉ cần click vào dấu “=” bên cạnh)
- Data Type: cho phép chọn kiểu dữ liệu cho thuộc tính, click vào dấu “...” bên cạnh, cửa sổ các kiểu dữ liệu mở ra, ta chọn kiểu dữ liệu phù hợp cho các thuộc tính, bên dưới cho có text box cho phép ta chọn kích thước chiều dài của từng kiểu dữ liệu.
- M (Mandatory): thuộc tính có bắt buộc hay không?
- P(Primary Identifier): thuộc tính có phải là khóa chính hay không?
- D(Displayed): có hiển thị thuộc tính này hay không?
- Tạo cả hai thực thể SINHVIEN và LOPHOC
- Tạo mối quan hệ (relationship) cho hai thực thể
 - Chọn biểu tượng mối quan hệ của hai thực thể từ thanh công cụ.
 - Click vào thực thể SINHVIEN và kéo qua thực thể LOP.
 - Để thay đổi kiểu của mối quan hệ: như tên của mối quan hệ, kiểu của mối quan hệ(1-1, 1-n, n-1,)

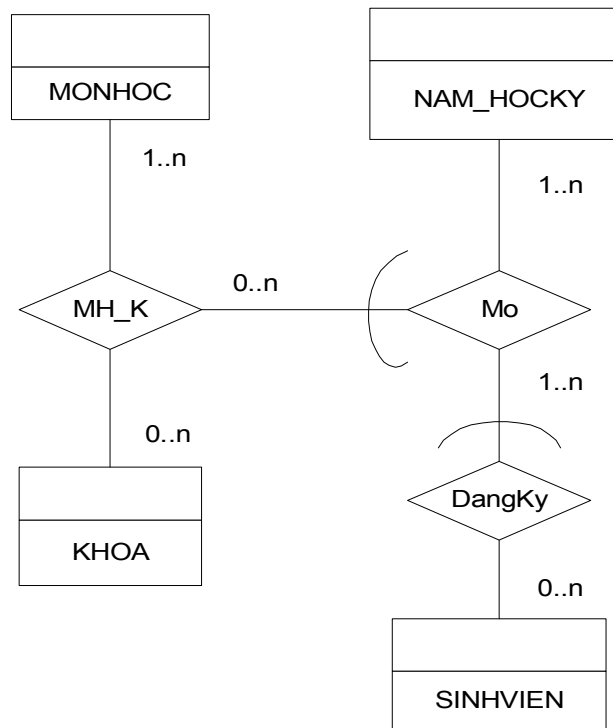
The screenshot shows the 'Relationship Properties' dialog box. At the top, there are three tabs: 'Definition', 'Description', and 'Annotation'. The 'Definition' tab is active. Below the tabs, there is a diagram showing two entities, 'SINHVIEN' and 'LOPHOC', connected by a relationship line. Below the diagram, there are two input fields for the relationship: 'Name' and 'Code'. The 'Name' field contains 'Thuoc' and the 'Code' field contains 'THUOC'. Below these fields, there is a 'Label' field. Under the 'Cardinality' section, there are four radio buttons: 'One to One', 'One to Many', 'Many to One' (which is selected), and 'Many to Many'. Below the cardinality section, there are two sections for the relationship properties. The first section is for 'SINHVIEN' to 'LOPHOC' and the second is for 'LOPHOC' to 'SINHVIEN'. Each section has a 'Mandatory' checkbox (which is checked), a 'Dependent' checkbox, a 'Dominant' checkbox, and 'Min' and 'Max' fields. The 'Min' field for both directions is '1' and the 'Max' field for 'SINHVIEN' to 'LOPHOC' is '1' and for 'LOPHOC' to 'SINHVIEN' is 'n'. There are also 'Role' fields for both directions. At the bottom, there is a 'Rules' button and a 'Generate' checkbox (which is checked). The dialog box has 'OK', 'Cancel', 'Apply', and 'Help' buttons at the bottom.

- Để thay đổi tên của mối quan hệ ta nhập tên mới vào textbox name

- Tùy theo yêu cầu của mô đề bài mà ta chọn mối quan hệ thích hợp. Ta có các mối quan hệ: 1-1, 1-n, n-1, n-n, và ta chọn trong phần Cardinality
- Dependent cho phép ta có sử dụng phụ thuộc khóa hay không?
- Mandatory: cho biết có bắt buộc ít nhất hay không?

4. Cách biểu diễn mối kết hợp mở rộng

Giả sử ta có mối quan hệ kết hợp sau:



❖ Mô tả các thuộc tính của các bảng dữ liệu:

MONHOC

| Thuộc tính | Kiểu dữ liệu | Kích thước | Ghi chú |
|------------|--------------|------------|-----------------|
| MAMH | Character | 10 | Thuộc tính khóa |
| TENMH | Character | 30 | |
| SOTINCHI | Integer | | |

KHOA

| Thuộc tính | Kiểu dữ liệu | Kích thước | Ghi chú |
|-------------|--------------|------------|-----------------|
| MAKHOA | Character | 10 | Thuộc tính khóa |
| TENKHOA | Character | 30 | |
| NAMTHANHLAP | Integer | | |

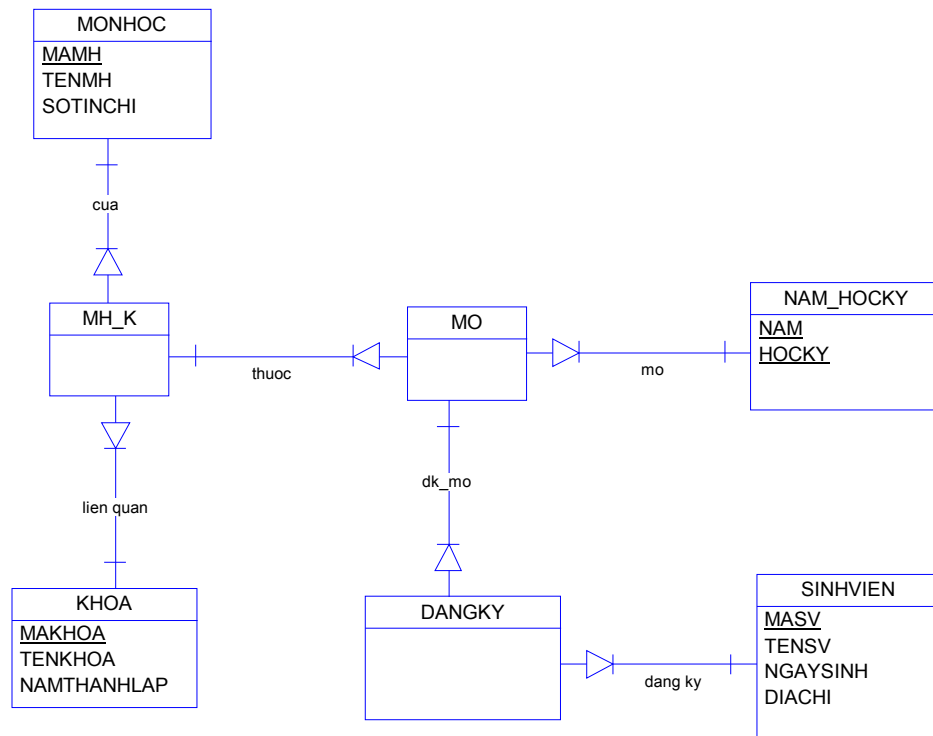
NAM_HOCKY

| Thuộc tính | Kiểu dữ liệu | Kích thước | Ghi chú |
|------------|--------------|------------|-----------------|
| NAM | Integer | | Thuộc tính khóa |
| HOCKY | Integer | | Thuộc tính khóa |

SINHVIEN

| Thuộc tính | Kiểu dữ liệu | Kích thước | Ghi chú |
|------------|--------------|------------|-----------------|
| MASV | Character | 10 | Thuộc tính khóa |
| TENSV | Character | 30 | |
| NGAYSINH | Datetime | | |
| DIACHI | Character | 30 | |

- ❖ Cách biểu diễn mô hình trong mô hình CDM của Power Designer
 - Đầu tiên các mối kết hợp và kết hợp mở rộng ta điều tạo là các thực thể nhưng không chứa thuộc tính nào cả.
 - Việc tạo các thực thể trên theo mô hình sau:



- Việc tạo các thực thể đơn giản như đã làm ở phần bài tập trên, ở đây ta quan tâm đến mối kết hợp và kết hợp mở rộng là làm sao chúng kế thừa được khóa từ những thực thể của chúng, việc này thật đơn giản trong lúc tạo mối quan hệ giữa hai thực thể ta chỉ cần check vào checkbox phụ thuộc khóa như hình bên dưới:

Relationship Properties

Definition | Description | Annotation

MONHOC MO

Name: cua = Code: RELATION_40 =

Label:

Cardinality

☐ One to One ☒ One to Many ☐ Many to One ☐ Many to Many

"MONHOC" to "MO"

☒ Mandatory ☐ Dependent ☐ Dominant Min: 1 Max: n

Role:

"MO" to "MONHOC"

☒ Mandatory ☒ Dependent ☐ Dominant Min: 1 Max: 1

Role:

Rules ☒ Generate

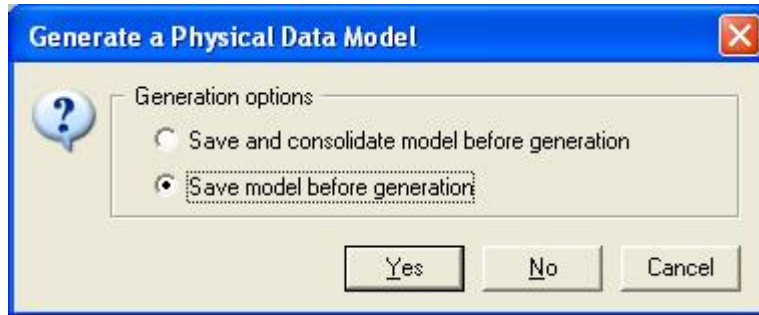
OK Cancel Apply Help

Check phụ Thuộc khóa

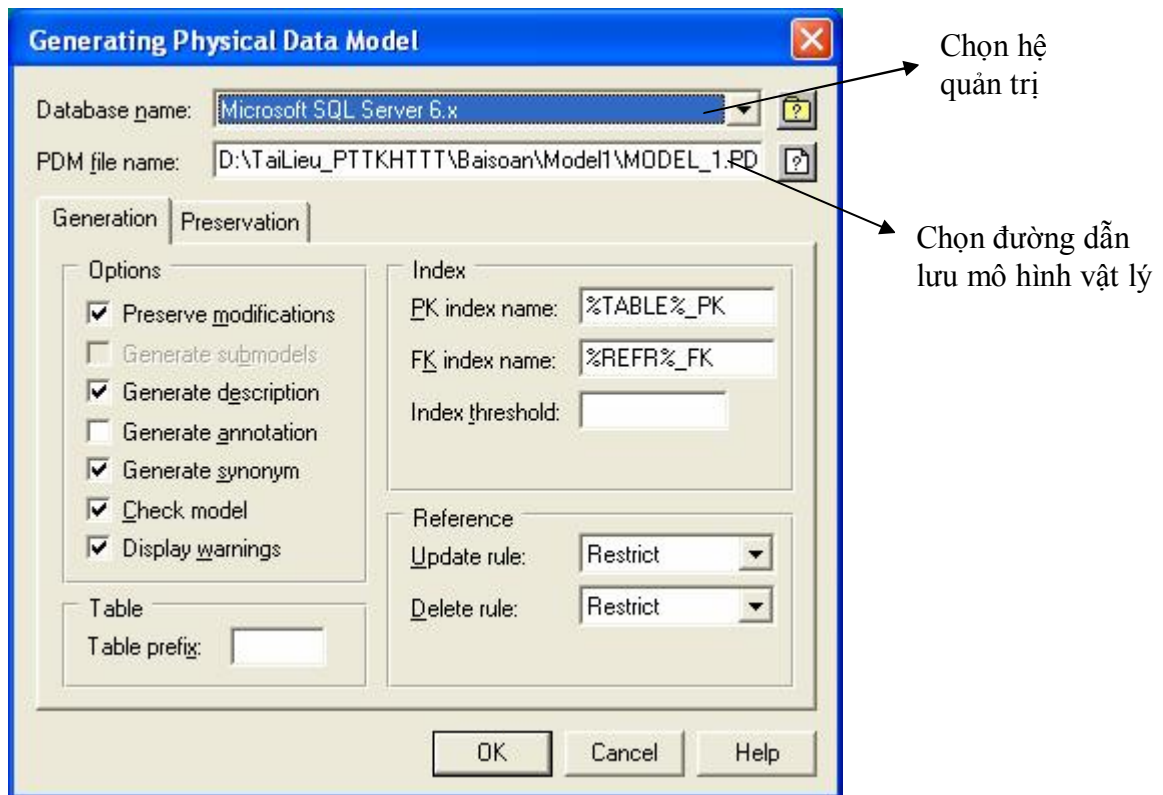
- Tương tự ta làm cho các mối kết hợp còn lại, lưu ý ở đây là các thực thể kế thừa khóa thì dấu dependent (biểu tượng hình tam giác có một gạch) sẽ nằm gần thực thể đó.

II. CÁCH CHUYỂN ĐỔI TỪ MÔ HÌNH CDM SANG MÔ HÌNH VẬT LÝ – PHYSICAL DATA MODEL(PDM)

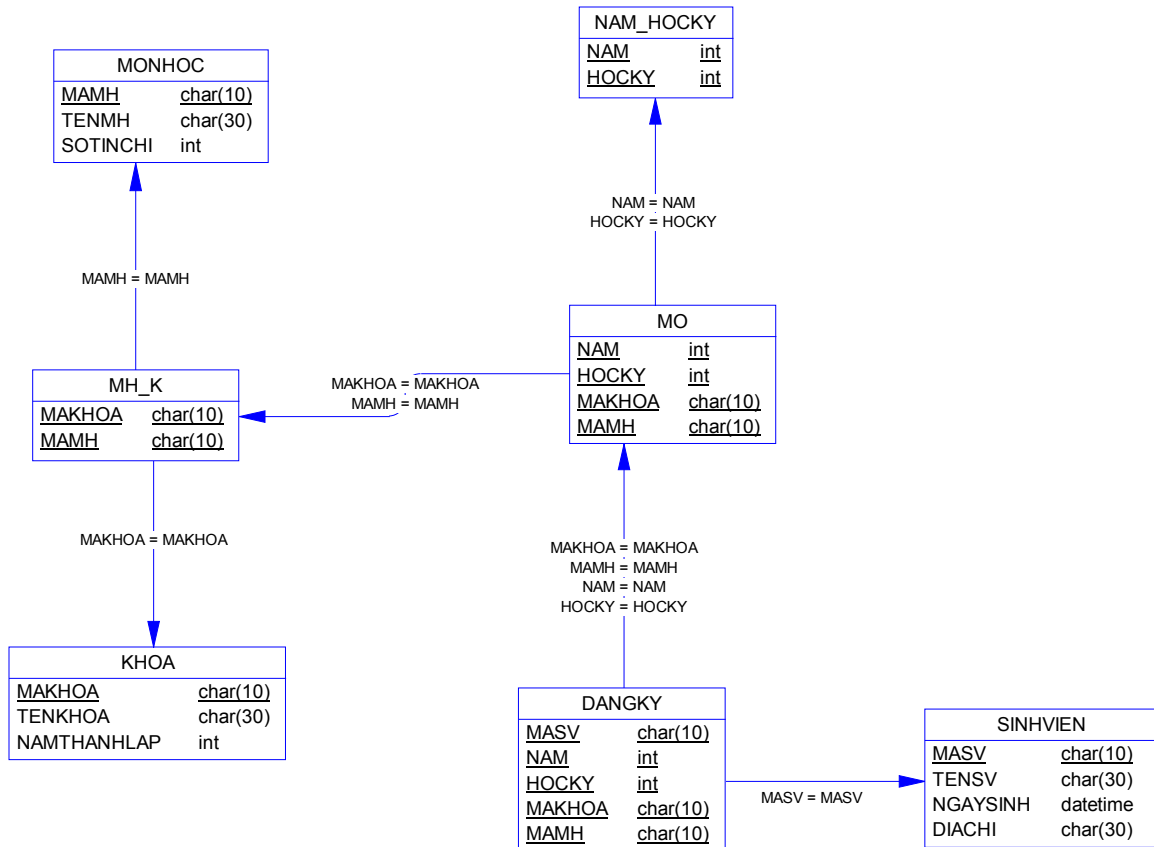
- Kiểm tra mô hình:
 - Chọn Dictionary/Check Model hoặc ta có thể nhấn phím tắt F4
- Chuyển mô hình quan niệm sang mô hình vật lý:
 - Ta kiểm tra chắc chắn mô hình không bị lỗi
 - Chọn Dictionary/Generate Physical Data Model hoặc sử dụng phím tắt Ctrl + G.
 - Một cửa sổ xuất hiện nhắc nhở chúng ta lưu mô hình trước khi chuyển nó xuống mô hình vật lý, chúng ta chọn option thứ hai và nhấn Yes để lưu mô hình như hình minh họa sau:



- Tiếp đến một cửa sổ xuất hiện cho phép ta chọn Hệ quản trị nào chúng ta dùng để lưu trữ database sau này, màn hình sau minh họa cho chọn hệ quản trị SQL Server để lưu trữ dữ liệu sau này:



- Sau khi chọn xong DBMS và đường dẫn để lưu mô hình vật lý, nhấn OK ta được mô hình vật lý như sau:



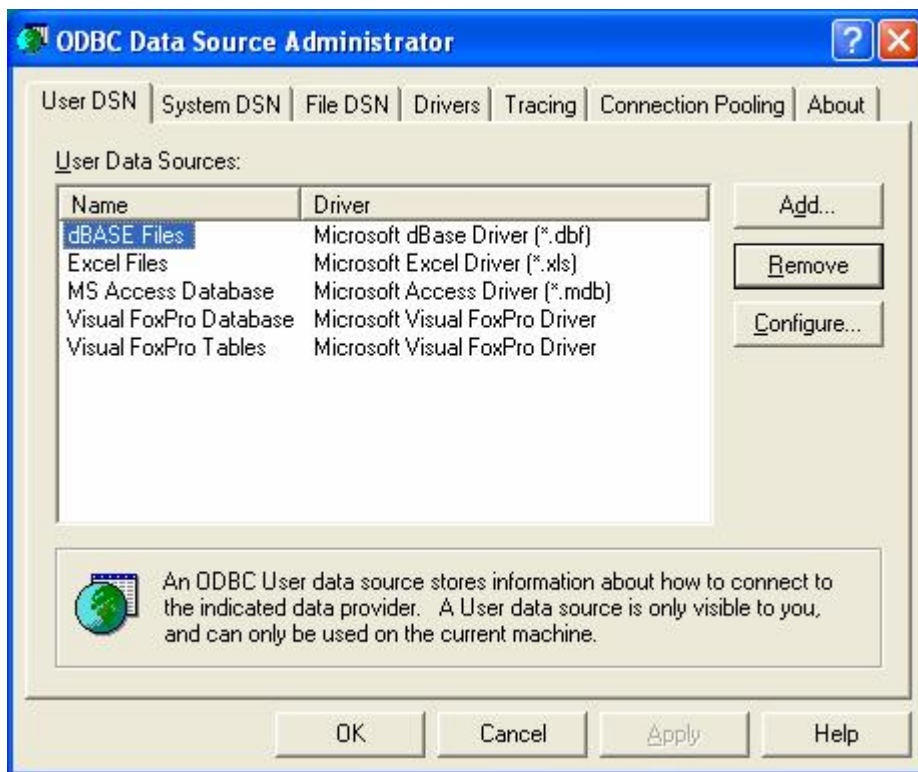
- Như vậy ở mô hình vật lý trên ta thấy rằng tất cả các mối kết hợp và kết hợp mở rộng đều sử dụng khóa của những thực thể liên quan đến chúng, chứ không phải chúng tự tạo ra.

III. CÁCH CHUYỂN ĐỔI TỪ DỮ LIỆU TỪ MÔ HÌNH PDM SANG MICROSOFT SQL SERVER

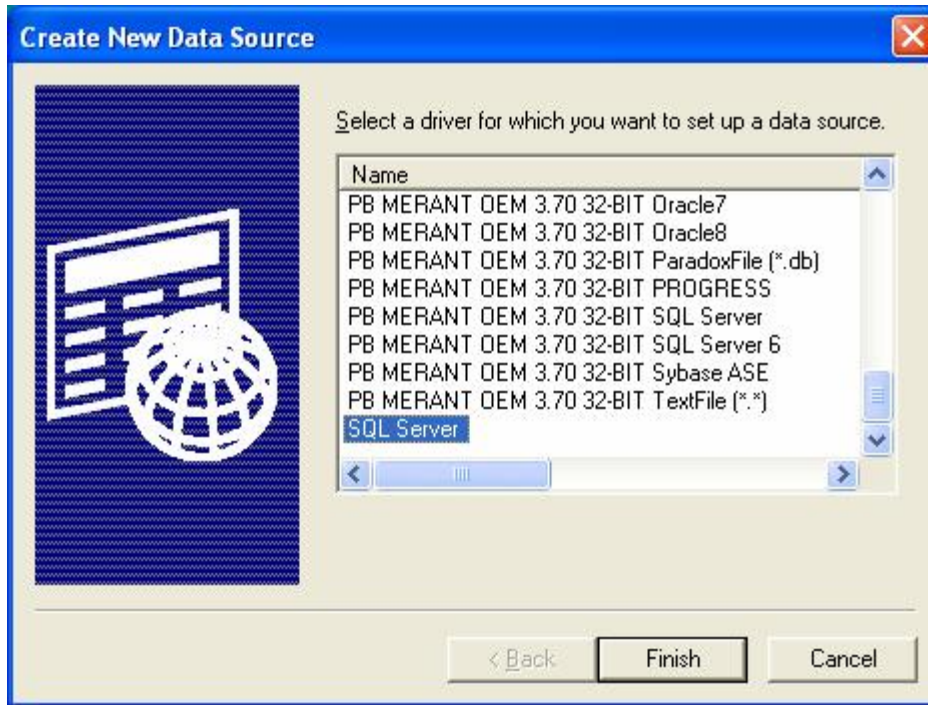
- Bước 1:
Connect và login vào SQL Server.
- Bước 2:
Chuyển từ mô hình quan niệm sang mô hình vật lý, lưu ý lúc chuyển ta nhớ chọn hệ quản trị là Microsoft SQL Server 6.x.
- Bước 2:
Từ mô hình vật lý:
 - Chọn Database/Connect. Một cửa sổ connect xuất hiện như sau:



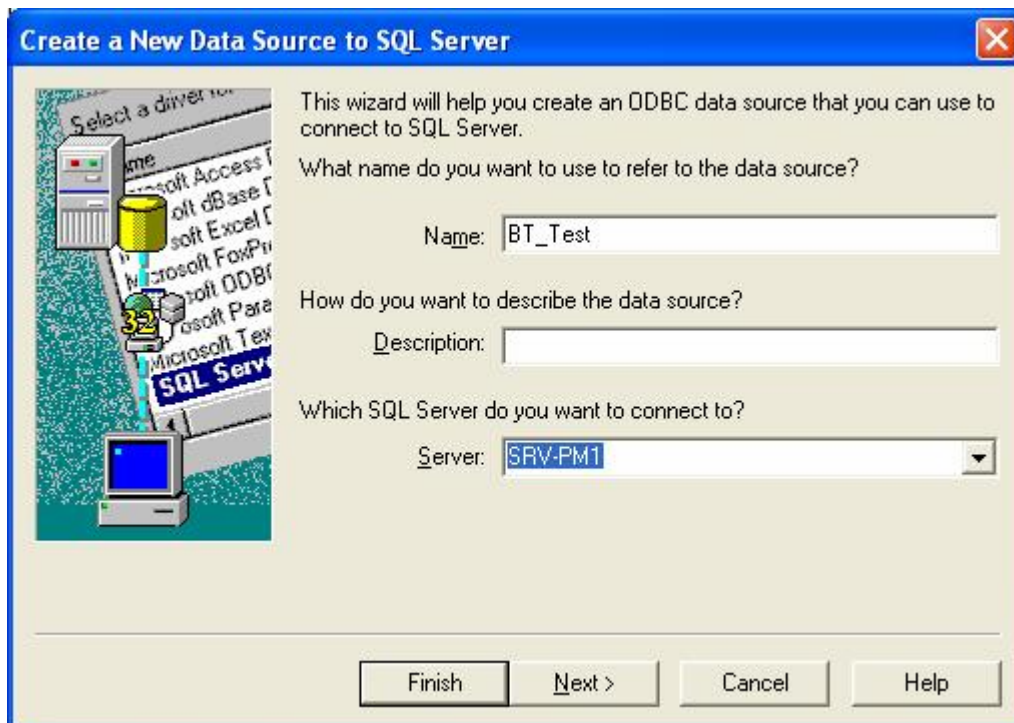
Ở đây chúng ta phải thực hiện kết nối qua một ODBC, do vậy chúng ta phải tạo một kết nối ODBC, để kết nối ta nhấn vào nút **Add**, lúc đó một màn hình ODBC Data Source Administrator xuất hiện như sau:



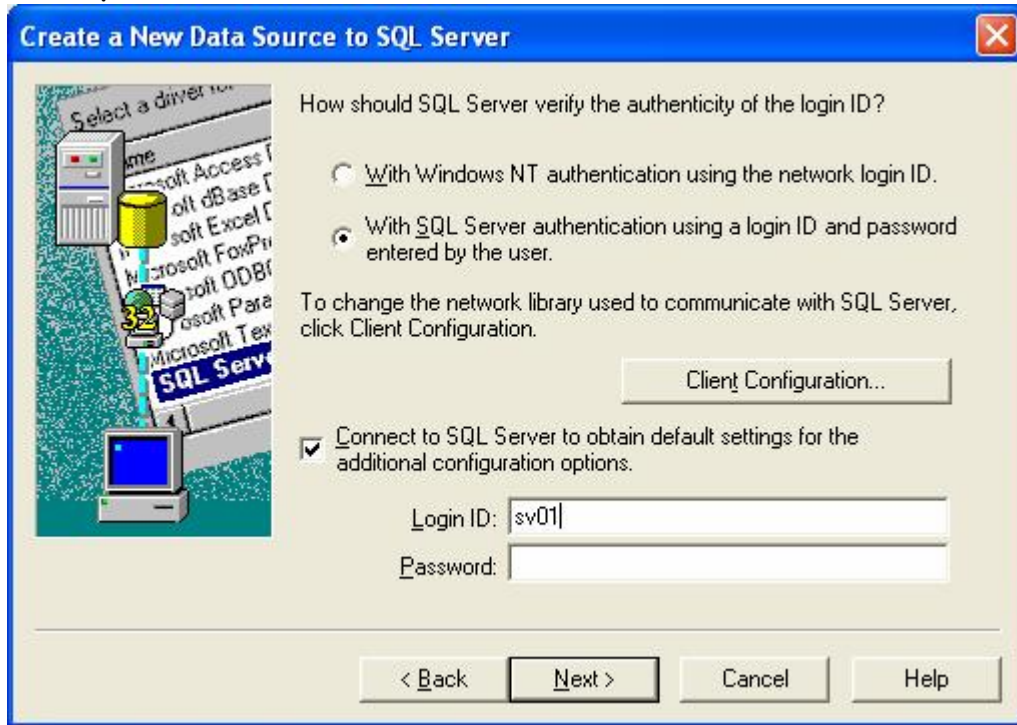
Ta tiếp tục nhấn **Add**, một cửa sổ Create Data Source xuất hiện, ta chọn trong listbox SQL Server, hình bên dưới:



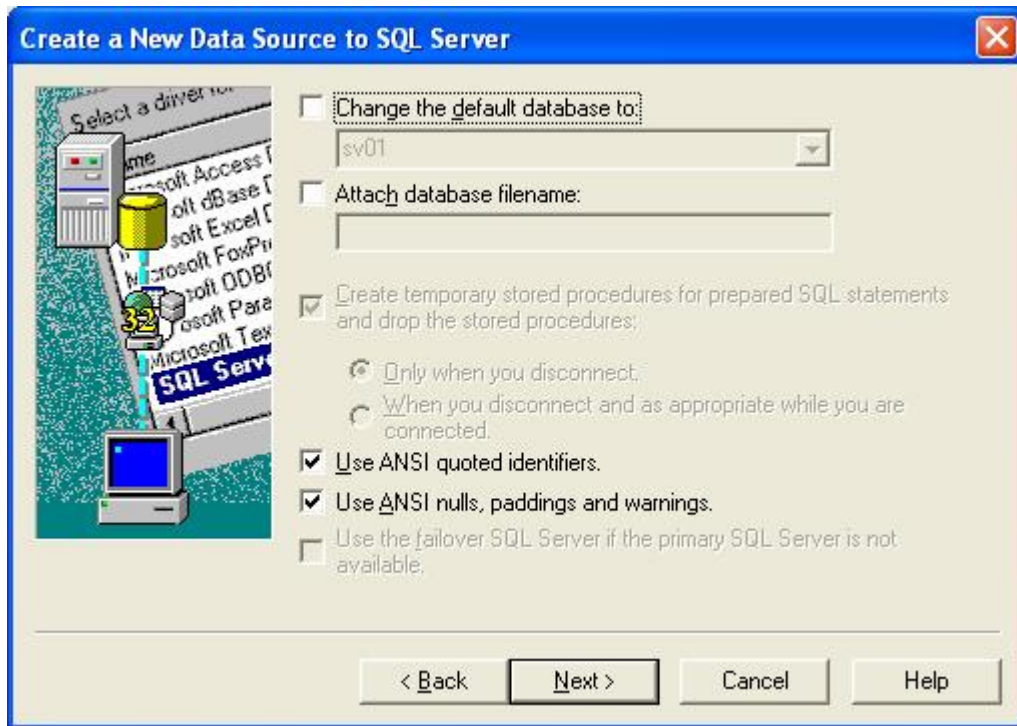
Tiếp theo ta nhấn **Finish**, một cửa sổ Create a new data source to SQL Server, trong phần Name ta gõ tên bất kỳ tùy thích (chẳng hạn BT_Test), phần Description là phần mô tả nên ta không cần phải điền vào phần này, trong phần Server bạn phải chọn server đã connect bên SQL như hình minh họa sau:



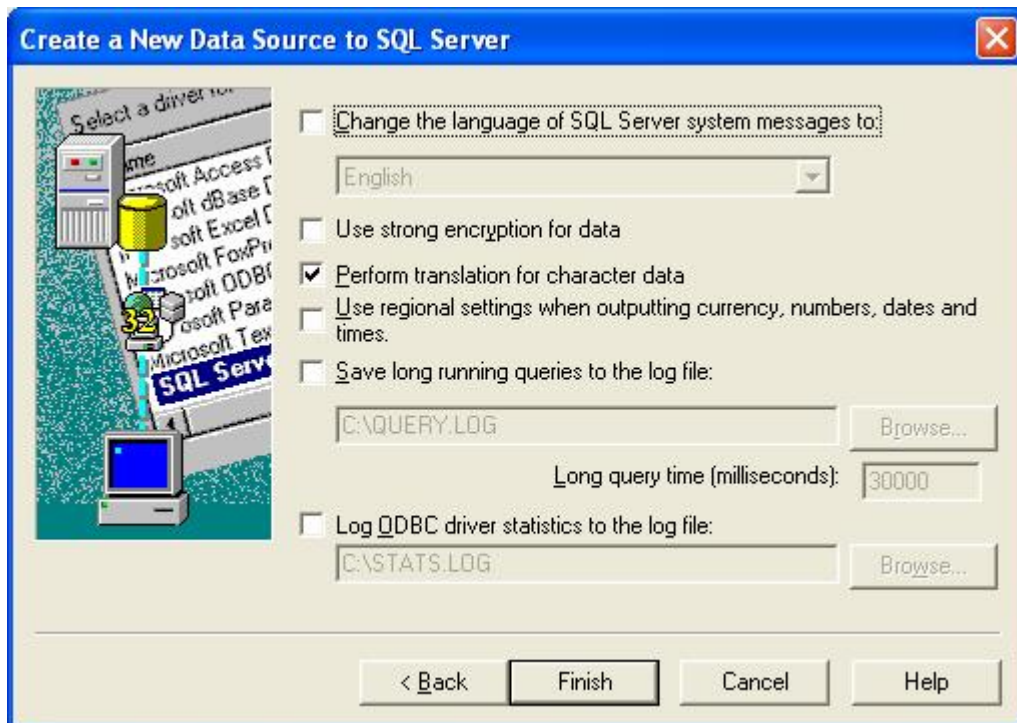
Tiếp theo ta nhấn nút **Next**, một cửa sổ khác lại xuất hiện, cho phép chúng ta chọn kiểu kết nối đến SQL Server, trong phần này ta phải chọn Option thứ hai(With SQL Server...), trong phần login ta chọn quyền login mà ta đã login như lúc login vào SQL Server(chẳng hạn như sv01), trong phần password ta bỏ trống theo hình minh họa sau:



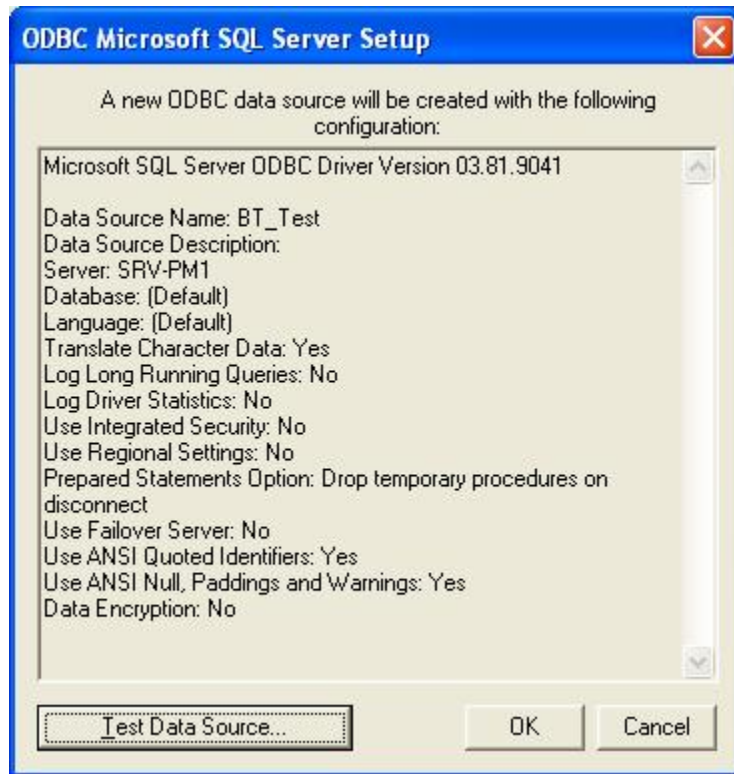
Kế tiếp là ta nhấn nút Next, ta được màn hình như sau :



Ta tiếp tục nhấn Next:



Để mặc định các giá trị ta tiếp tục nhấn Finish, thì cửa sổ ODBC Microsoft SQL Server Setup xuất hiện như sau:

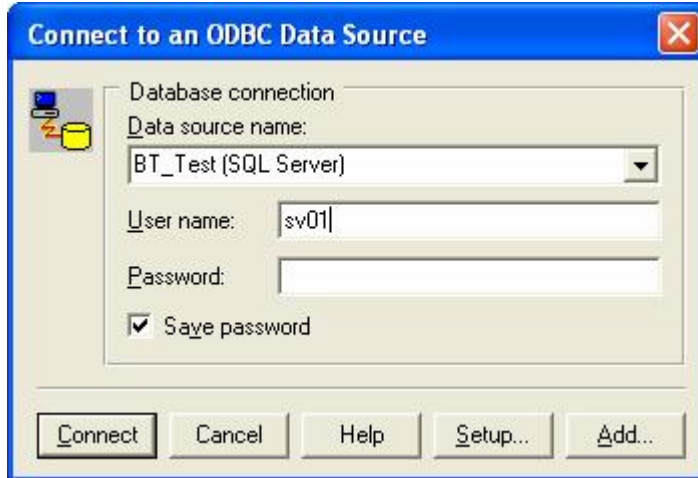


Tiếp theo ta nhấn vào Test Data Source, một màn hình thông báo kết quả connect có thành công hay không? Sau đây là màn hình thông báo connect đã thành công

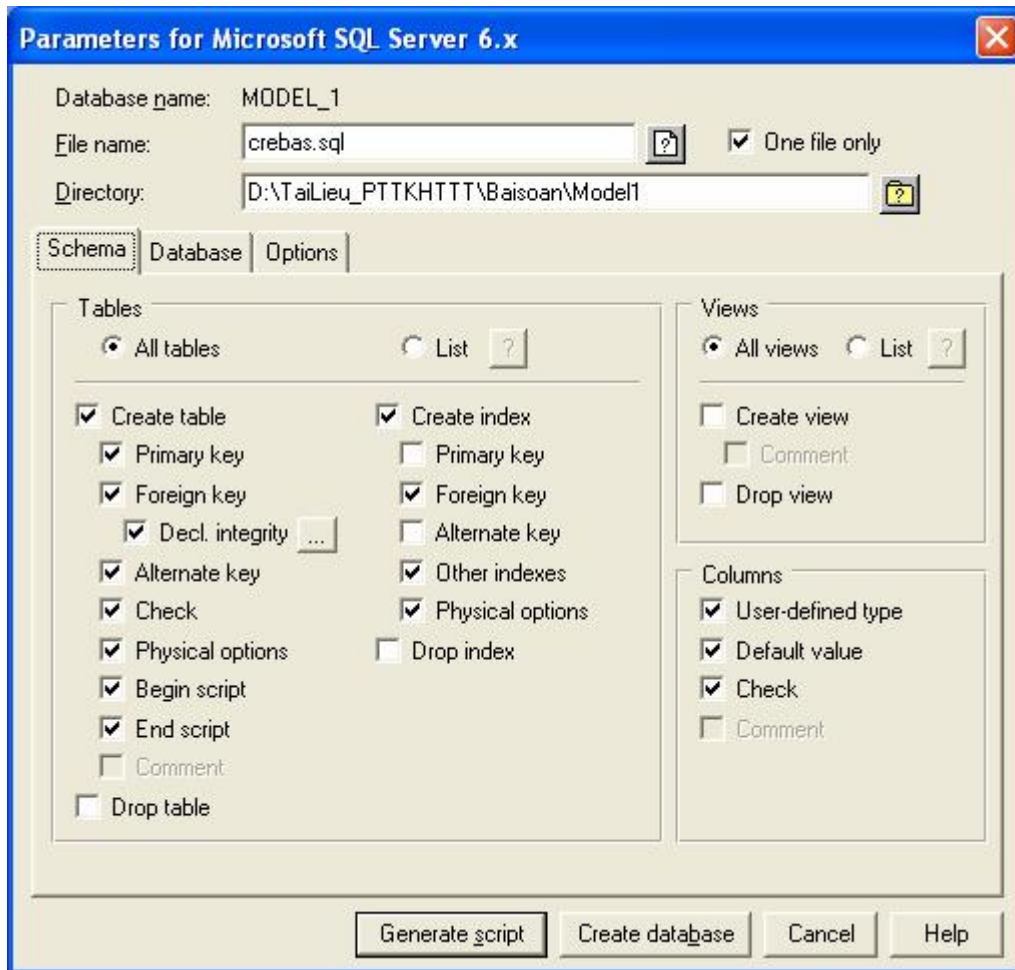


Sau đó ta nhấn nút OK liên tiếp nhau để quay về màn hình Connect ban đầu.

- Bước 3:
Ở màn hình connect ban đầu chúng ta chọn lại kết nối ODBC đã tạo là BT_Test trong phần Data Source Name, trong phần login name ta gõ vào tên mà ta dùng để login vào SQL(chẳng hạn sv01) và sau đó ta nhấn vào nút Connect. Ta có hình minh họa sau:



- Ta chọn Database/Generate Database hay nhấn phím tắt Ctrl + G, cửa sổ Parameters for Microsoft SQL 6.x xuất hiện, cửa sổ này cho phép chúng ta lựa chọn một số tham số cần thiết trước khi tạo database:



- Sau khi lựa chọn tham số xong, ta nhấn vào nút **Create database**, lúc đó toàn bộ các bảng dữ liệu sẽ được tạo ra trong database của SQL Server.

IV. CÁCH CHUYỂN ĐỔI MÔ HÌNH DỮ LIỆU TỪ MÔ HÌNH PDM SANG MICROSOFT ACCESS

- Bước 1: Vào Access tạo một database trống và đóng cửa sổ lại.
- Bước 2: Chuyển mô hình CDM sang mô hình PDM, trong lúc chuyển nhớ chọn là Microsoft Access.
- Bước 3: tạo một ODBC để kết nối với Access.
- Các bước còn lại ta làm tương tự như với SQL, nhưng lưu ý không cần connect đến server, mà ta chỉ cần chọn đường dẫn đến file Access là đủ. Sau đây là một số hình ảnh minh họa:

Generating Physical Data Model

Database name:

PDM file name:

Generation | Preservation

Options

- ☒ Preserve modifications
- ☐ Generate submodels
- ☒ Generate description
- ☐ Generate annotation
- ☒ Generate synonym
- ☒ Check model
- ☒ Display warnings

Table

Table prefix:

Index

PK index name:

FK index name:

Index threshold:

Reference

Update rule:

Delete rule:

OK Cancel Help

Connect to an ODBC Data Source

Database connection

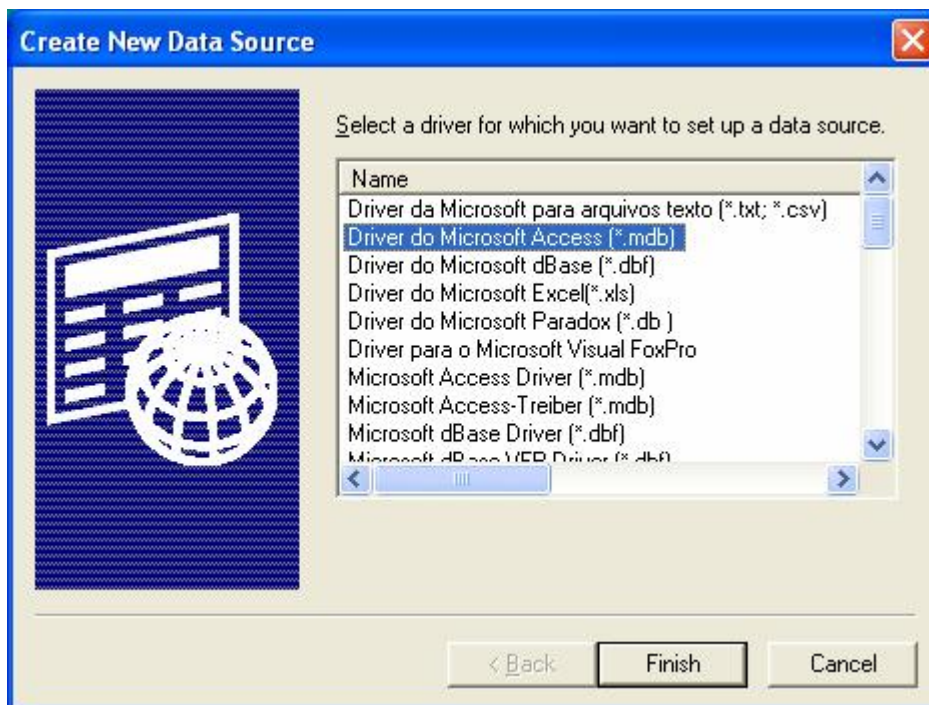
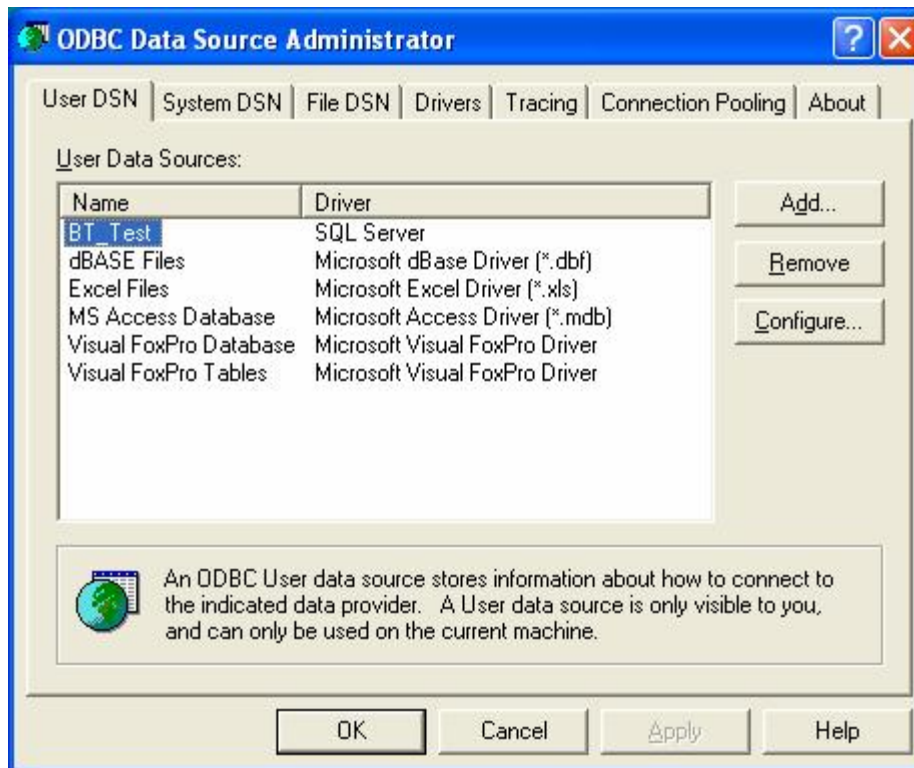
Data source name:

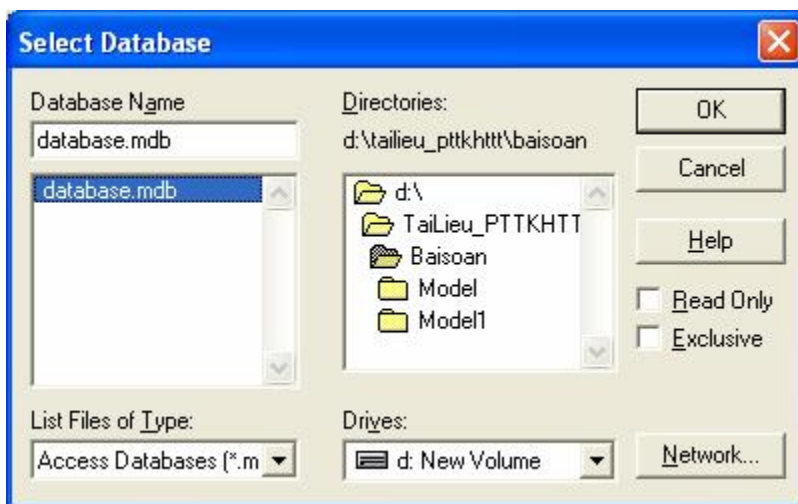
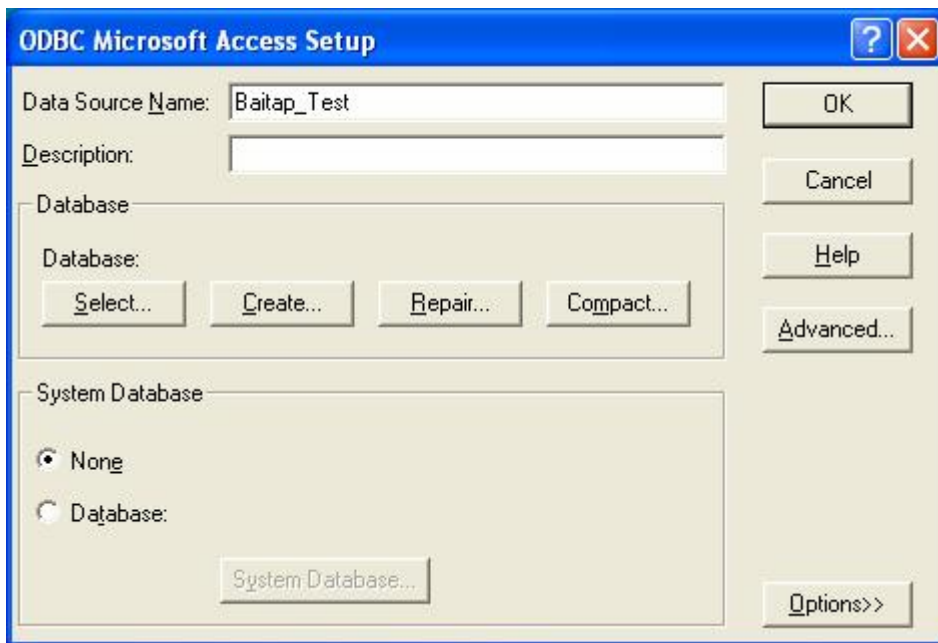
User name:

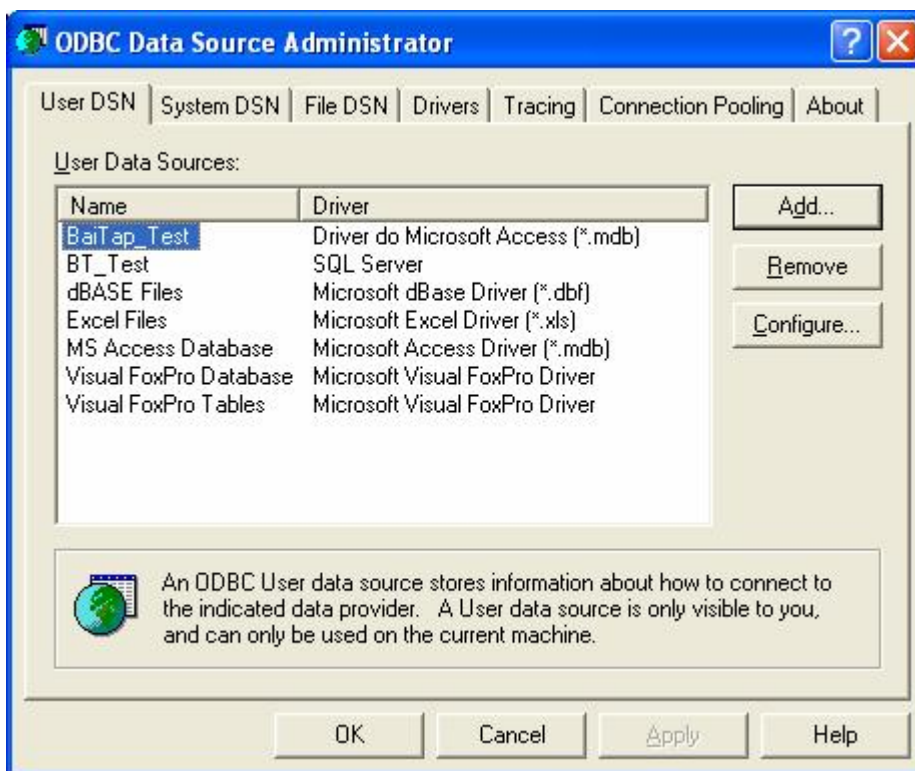
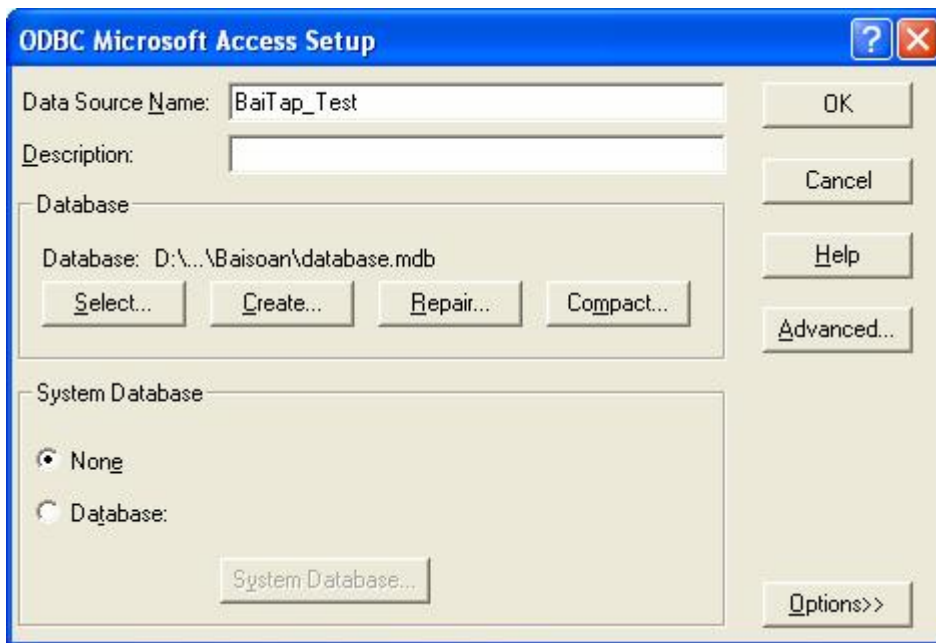
Password:

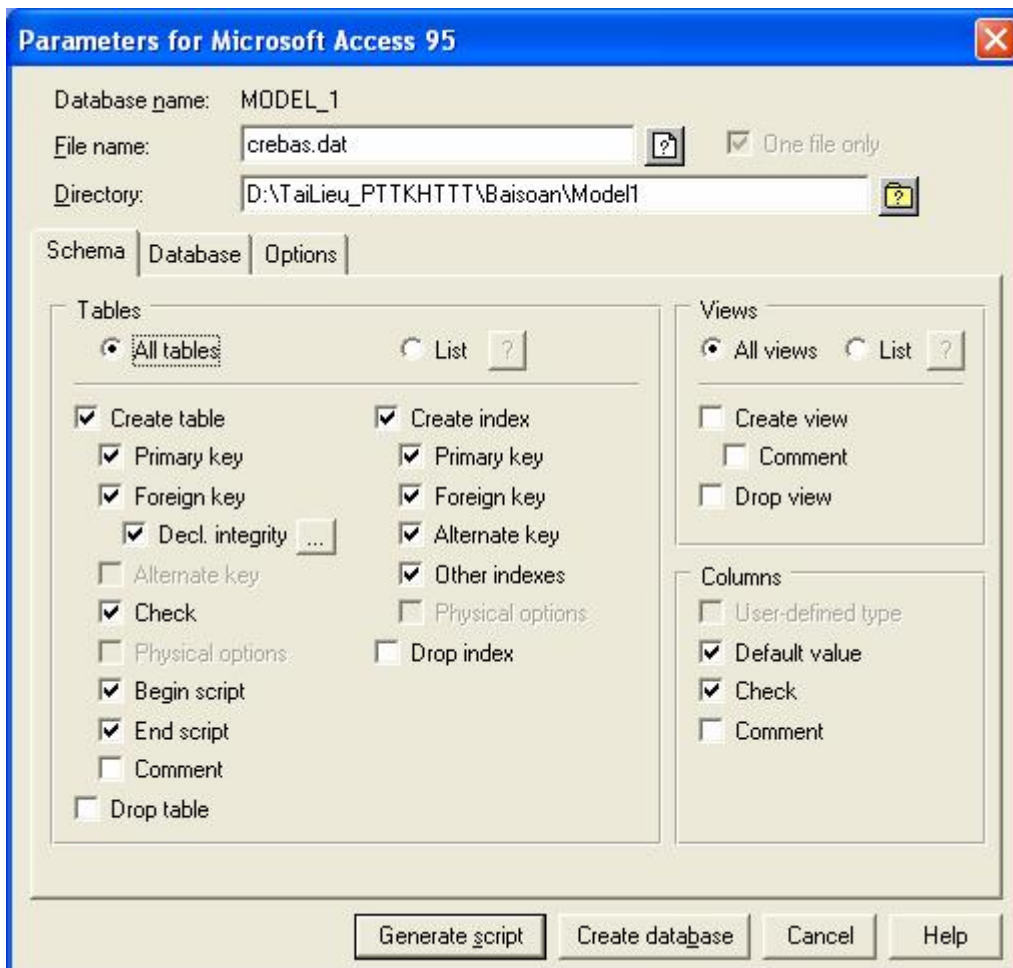
☒ Save password

Connect Cancel Help Setup... Add..









NỘI DUNG HƯỚNG DẪN THỰC HÀNH TUẦN 2

I. MÔ HÌNH HÓA XỬ LÝ

Phân tích xử lý được bao gồm trong việc mô hình hóa hệ thống thông tin. Đối tượng quan tâm của phân tích xử lý là các hoạt động hay xử lý thông tin và các dòng thông tin giữa các hoạt động xử lý này. Kết quả của giai đoạn phân tích xử lý này là lược đồ chức năng (Function schema) bao gồm các biểu diễn của hoạt động, dòng thông tin và các đặc trưng khác.

Lược đồ chức năng còn có thể được gọi là lược đồ xử lý (Process Schemal) của hệ thống thông tin ngược lại với mô hình tĩnh của hệ thống thông tin là mô hình thực thể kết hợp.

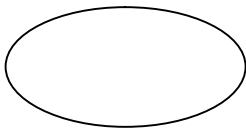

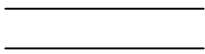
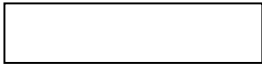
Trong thực tế tồn tại nhiều loại mô hình xử lý khác nhau:

- Một số tập trung vào dữ liệu và dòng dữ liệu giữa các hoạt động.
- Một số tập trung vào tính đồng bộ của các hoạt động bằng cách định rõ điều kiện trước và điều kiện sau của hoạt động.

Nhưng ta chỉ quan tâm đến loại mô hình đơn giản nhưng hiệu quả và khá phổ biến cho giai đoạn phân tích xử lý là mô hình dòng dữ liệu(Data Flow Diagram-DFD).

Mô hình DFD gồm các khái niệm chính : Xử lý (Process), dòng dữ liệu (Data flow), Kho dữ liệu (Data store) và Đầu cuối (Terminator hay Actor)

Sau đây là khái niệm và ký hiệu của mô hình DFD:

| Khái niệm | Ký hiệu | Ý nghĩa |
|-------------------|---|---|
| Xử lý |  | Một trong các hoạt động bên trong của HTTT, mỗi ô xử lý được đánh một số để biết được nó là ô xử lý thứ mấy và ở cấp nào. |
| Dòng dữ liệu |  | Sự thay đổi thông tin giữa các xử lý |
| Kho dữ liệu |  | Là vùng chứa thông tin |
| Tác nhân đầu cuối |  | Là một tác nhân bên ngoài hệ thống |

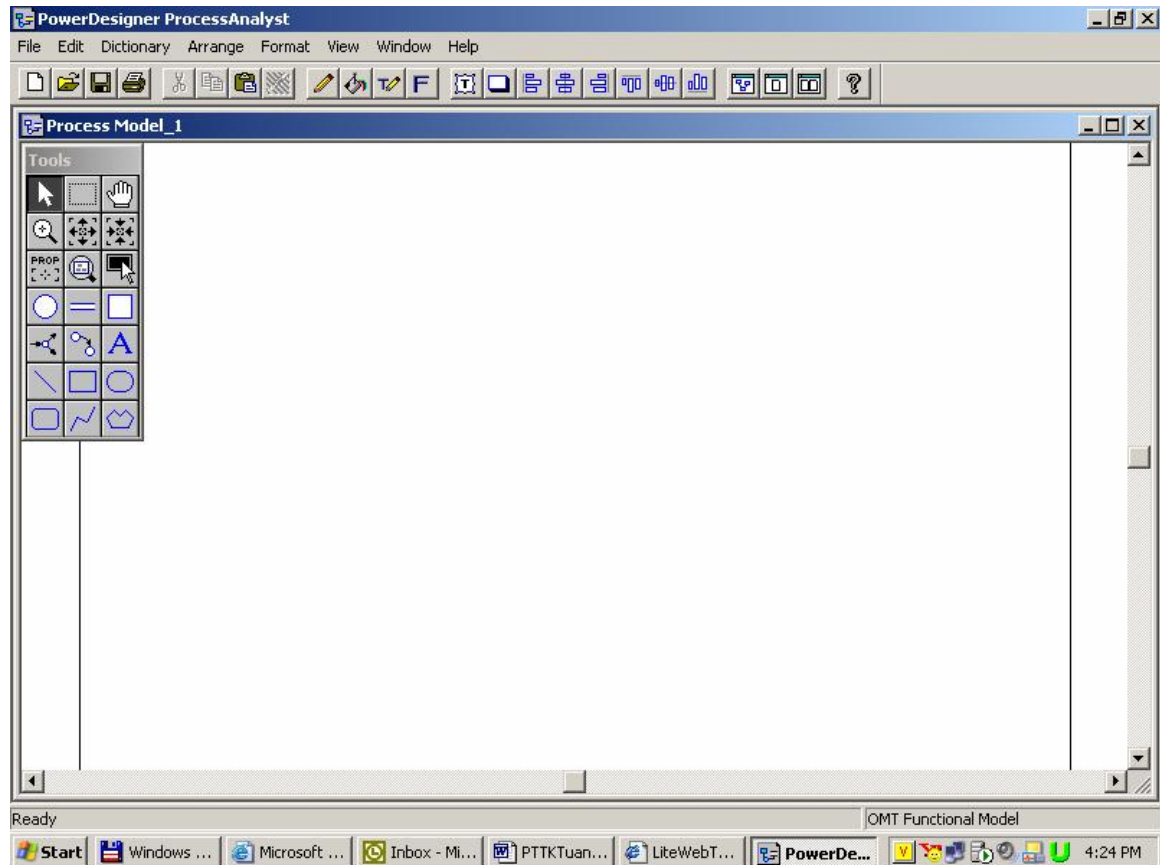
II. CÁC BIỂU DIỄN MÔ HÌNH DFD TRONG POWER DESIGNER

Trong Power Designer mô hình DFD được gọi là mô hình phân tích xử lý (*Process Analyst Model – PAM*)

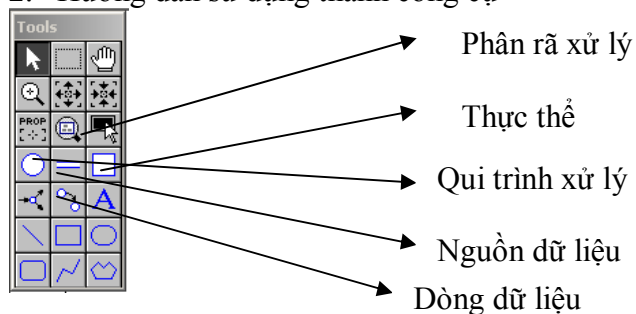
Hướng dẫn cách sử dụng mô hình PAM trong Power Designer:

1. Khởi động Power Designer

Start/Programs/ Power Designer 6 32-bit/ Process Analyst



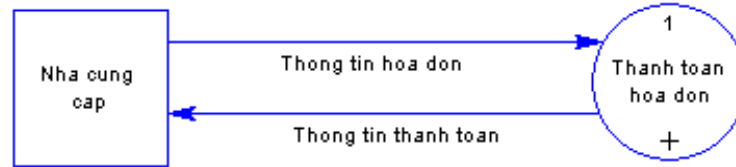
2. Hướng dẫn sử dụng thanh công cụ



III. HƯỚNG DẪN THIẾT KẾ MỘT MÔ HÌNH DFD CỤ THỂ

Xét mô hình xử lý cho quá trình thanh toán hóa đơn

1. Mô hình xử lý cấp 1:



❖ Hướng dẫn cách tạo:

- Chọn biểu tượng thực thể, giữ chuột và kéo thả vào trong lược đồ
- Để thay đổi thông tin của thực thể ta double-click vào thực thể cần thay đổi, một cửa sổ mới mở ra cho phép chúng ta chỉnh sửa thông tin của thực thể như: tên của thực thể, các rule, mô tả ...
- Chọn biểu tượng xử lý, giữ chuột và kéo thả vào trong lược đồ. Xử lý “Thanh toán hóa đơn” được đánh số 1. Để thay đổi số thứ tự, ta thay đổi giá trị trong ô Number.
- Để thay đổi thông tin của xử lý ta double-click vào xử lý cần thay đổi, một cửa sổ mới mở ra cho phép chúng ta chỉnh sửa thông tin của xử lý như: tên của xử lý, các rule, mô tả ...
- Chọn biểu tượng dòng dữ liệu, giữ chuột và kéo thả nối từ thực thể đến xử lý trong lược đồ
- Để thay đổi thông tin của dòng dữ liệu ta double-click vào dòng dữ liệu cần thay đổi, một cửa sổ mới mở ra cho phép chúng ta chỉnh sửa thông tin của dòng dữ liệu như: tên của dòng dữ liệu, dữ liệu chuyển, các rule, mô tả ...

External Entity Properties

Definition | Description | Annotation

Name:

Code:

Label:

Rules

OK Cancel Apply Help

Process Properties

Definition Specification Description Annotation

Name: Thanh toán hoa don =

Code: PRCS_4 =

Label:

Number: 1

Lowest level: ☐

Rules

OK Cancel Apply Help

Data Flow Properties


Definition Data Items Description Annotation

Nha cung cap Thanh toán hoa don

Name: Thông tin hoa don =

Code : FLOW_48 =

Label:

Type: Logical Direction: 

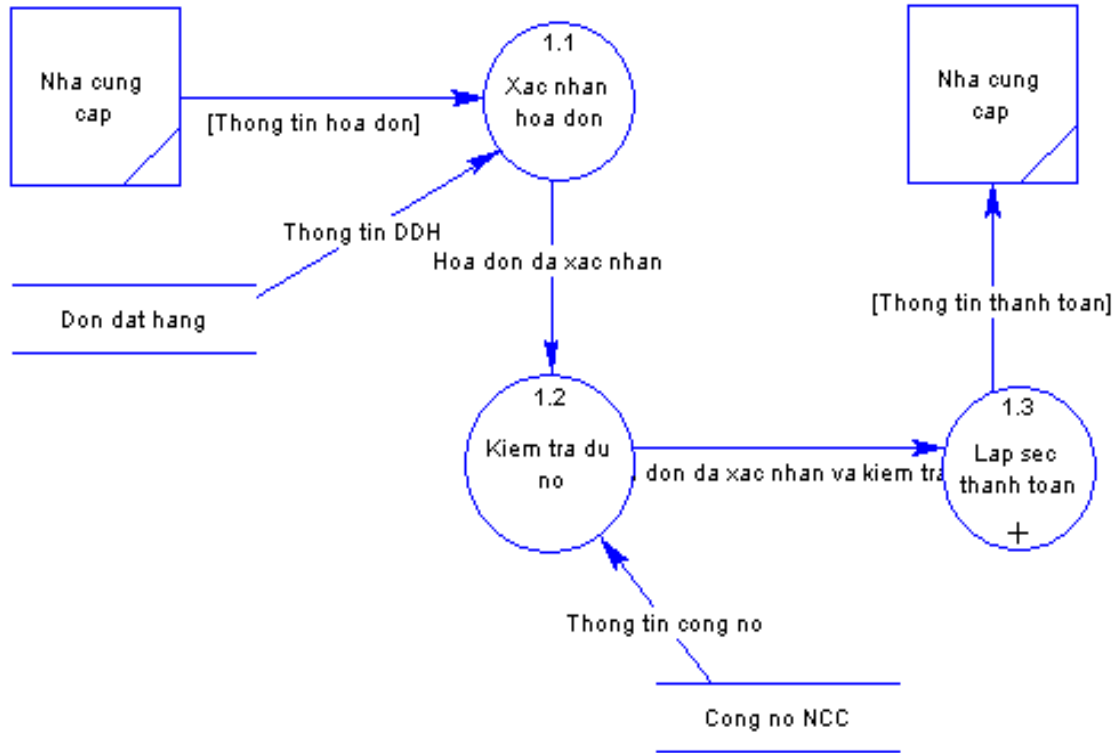
Operations: C ☐ R ☐ U ☐ D ☐

Rules

OK Cancel Apply Help

2. Mô hình xử lý cấp 2:

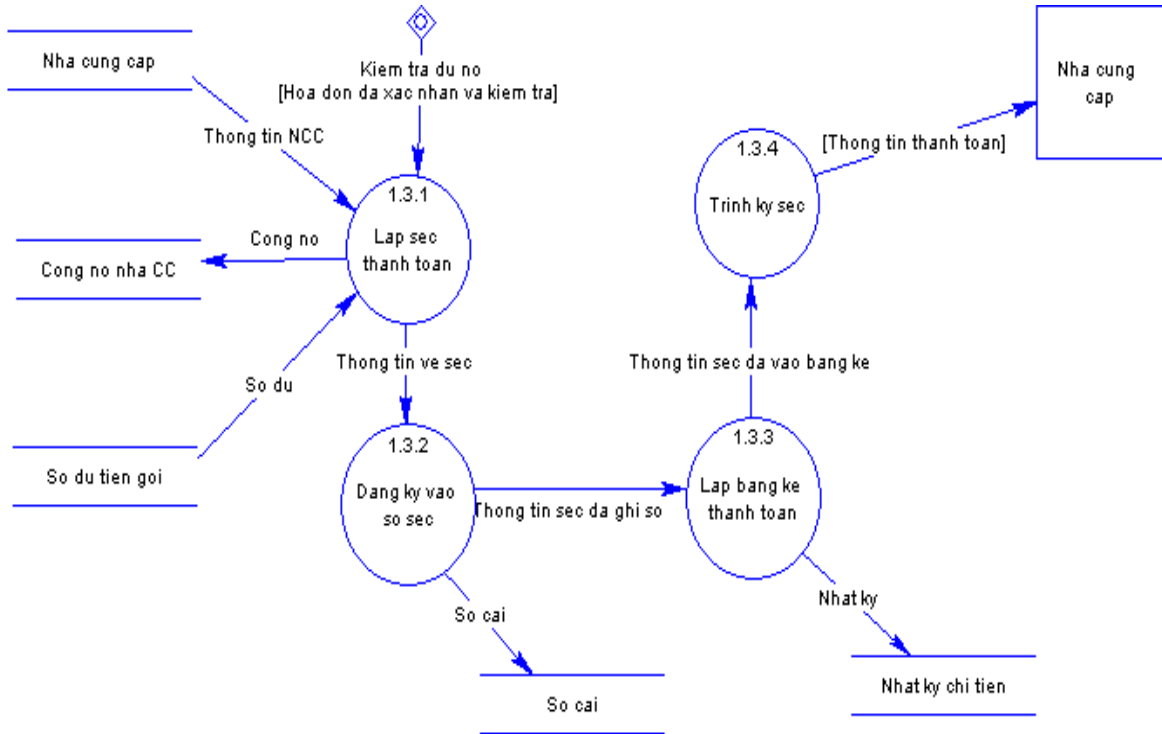
- ❖ *Phát triển mô hình xử lý Thanh toán hóa đơn: có nghĩa là tình chế ô xử lý số 1 – Thanh toán hóa đơn*



- ❖ *Hướng dẫn cách tạo:*
 - Chọn biểu tượng phân rã xử lý, giữ chuột và kéo thả đè lên xử lý “Thanh toán hóa đơn” trong lược đồ. Power Designer sẽ tự động tạo một lược đồ xử lý ở cấp chi tiết hơn cho xử lý “Thanh toán hóa đơn” (đánh số 1.1).
 - Lưu ý: Để chuyển đổi giữa các lược đồ chọn menu Window (hay phím nóng Alt-W)
 - Nếu xử lý cha bị xóa hay thay đổi số thứ tự thì xử lý con sẽ bị xóa và thay đổi số thứ tự theo.
 - Tương tự tiếp tục kéo thả nguồn dữ liệu, dòng dữ liệu để xây dựng mô hình xử lý cấp 2 cho xử lý “Đăng ký học phần”
 - Lưu ý: Khi phân rã xử lý, Power Designer sẽ tự động tạo ra hai thực thể sinh viên, do đó ta cần xóa đi một thực thể sinh viên, và điều chỉnh lại dòng dữ liệu.

3. Mô hình xử lý cấp 3:

- ❖ *Tiếp tục phát triển mô hình xử lý Lập séc thanh toán: nghĩa là ta phân rã ô xử lý 1.3- Lập séc thanh toán*



- ❖ *Hướng dẫn cách tạo:*
Thực hiện tương tự như các bước trên.

IV. MỘT SỐ LƯU Ý KHI THIẾT KẾ MÔ HÌNH XỬ LÝ

1. Khi xây dựng một mô hình xử lý nên tuân theo một số hướng dẫn sau đây:

- Mọi dòng dữ liệu ra của ô xử lý phải dựa trên dòng dữ liệu vào của ô xử lý đó (đây là điều kiện cần)
- Chỉ những dữ liệu nào thật cần cho ô xử lý mới đưa vào (điều kiện đủ)
- Một ô xử lý nên độc lập chức năng với ô xử lý khác
- Các ô xử lý luôn ở trạng thái sẵn sàng hoạt động

2. Quá trình phân cấp xử lý nên tuân theo các qui định sau:

- Quá trình phân cấp sẽ được phân cấp đến một chi tiết nào đó mà người thiết kế xem là đủ
- Mỗi cấp chỉ chứa từ 3 đến 7 ô xử lý, nếu nhiều hơn thì khó quan sát một cách tổng quát
- Ở cấp trên ta chỉ đặc biệt quan tâm đến nội dung của quá trình xử lý, còn các hoạt động hoặc sai cần xử lý thì chúng ta nên để từ cấp 3 trở đi
- Nếu đi vào một cấp chi tiết mà có thêm kho dữ liệu mới hay đầu cuối mới thì vẫn là hợp lý.

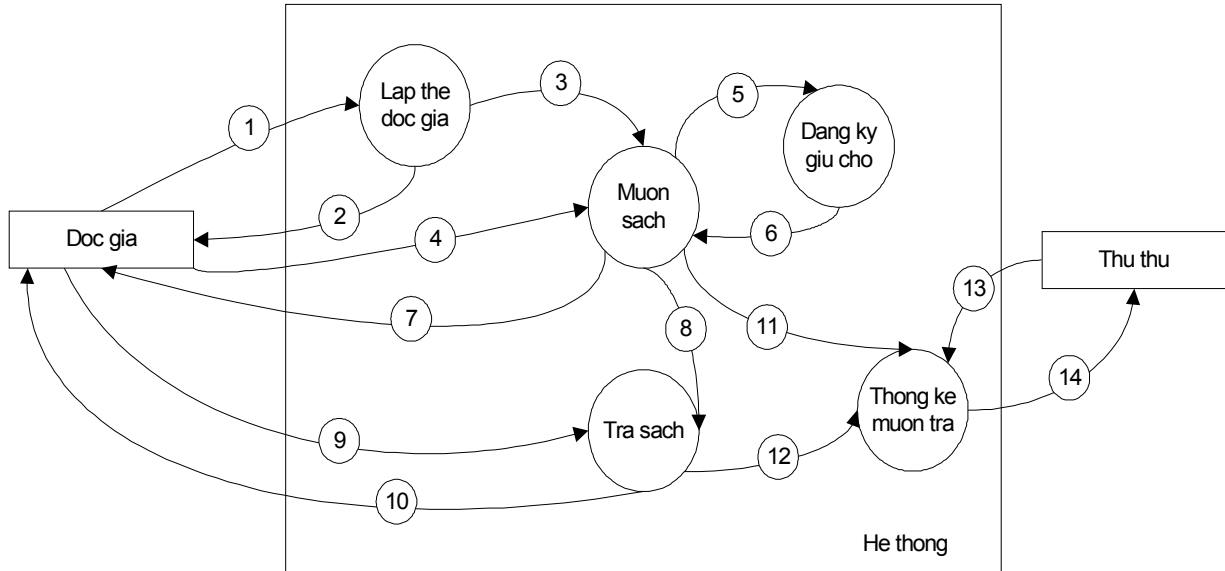
3. Chất lượng của lược đồ xử lý:

Trong phần này chúng ta sẽ xem xét các đặc trưng để đánh giá chất lượng của lược đồ xử lý là : độc lập chức năng, đầy đủ, tính đúng đắn, dễ đọc và tối thiểu:

- Tính độc lập chức năng:
Đặc trưng này được xét khi ô xử lý có tính tự trị đầy đủ(ví dụ nó có thể thực hiện các chức năng của nó một cách độc lập). Tính chất này chi tiết như sau:
 - Cách biệt: mỗi ô xử lý có thể được phân tích độc lập chi tiết
 - Dễ tích hợp: khi tinh chế một ô xử lý sẽ dễ dàng tích hợp vào phần còn lại của lược đồ xử lý.
 - Uyển chuyển: một ô xử lý dễ thay đổi mà không kéo theo sự thay đổi của các ô xử lý khác.
- Tính đầy đủ: một lược đồ xử lý đầy đủ là khi nó biểu diễn đầy đủ tất cả các đặc trưng của lĩnh vực ứng dụng ở cấp chi tiết.
- Tính đúng đắn: một lược đồ xử lý là đúng đắn khi nó chỉ sử dụng các khái niệm của mô hình xử lý để biểu diễn lĩnh vực ứng dụng
- Tính dễ đọc: một lược đồ xử lý mang tính dễ đọc khi nó biểu diễn lĩnh vực ứng dụng một cách tự nhiên và có thể dễ dàng hiểu được mà không cần giải thích gì thêm.
- Tính tối thiểu: một lược đồ xử lý là tối thiểu nếu mọi khái niệm trong lĩnh vực ứng dụng chỉ xuất hiện một lần trong lược đồ.

XÂY DỰNG MÔ HÌNH QUAN NIỆM XỬ LÝ BẰNG DFD

1. Xác định sơ đồ hệ thống con của qui trình

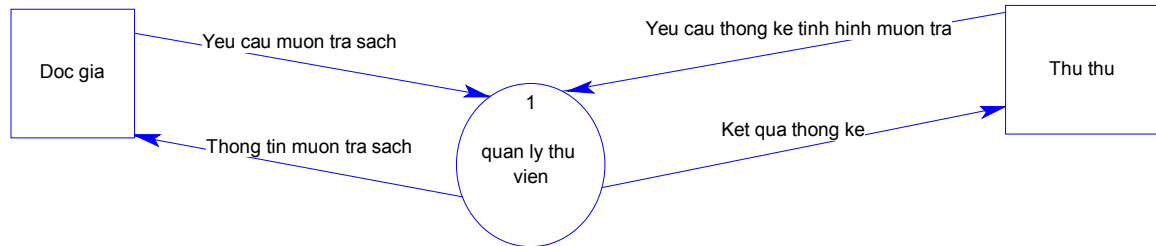


Giải thích các dòng dữ liệu lưu chuyển:

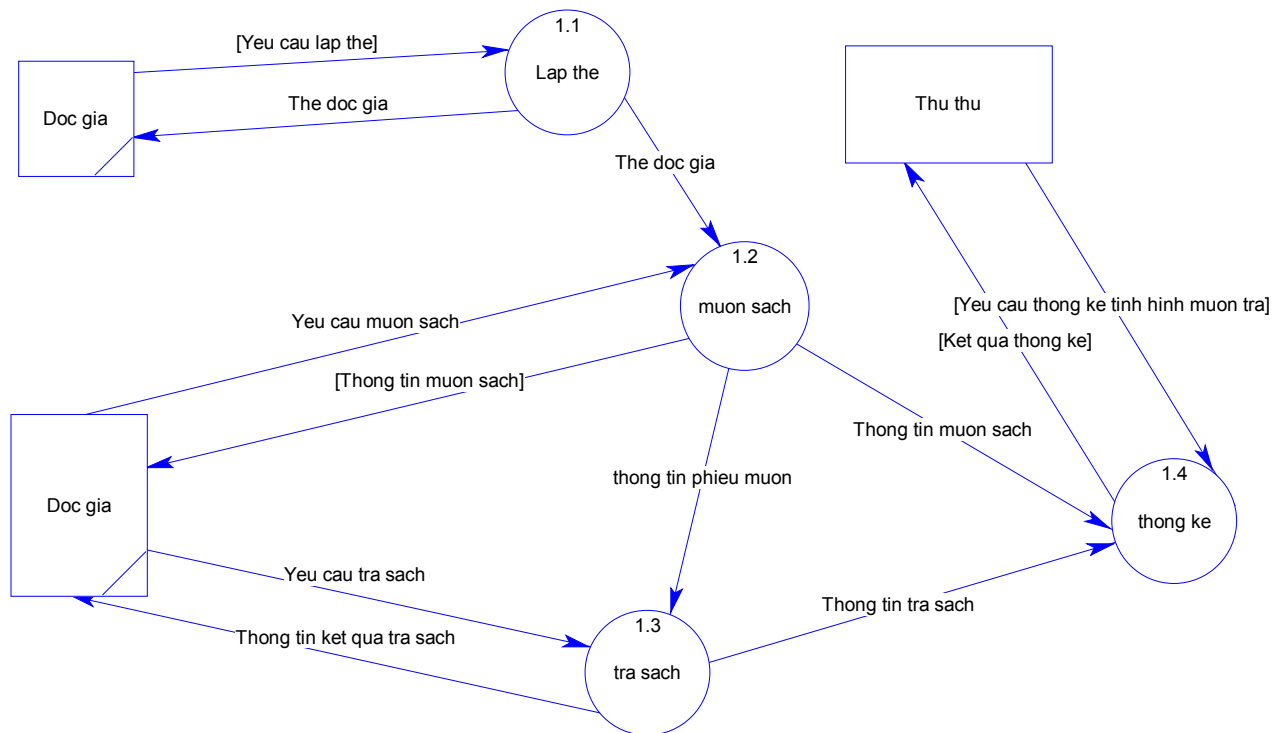
| Số | Diễn giải | Số | Diễn giải |
|----|--|----|---|
| ① | Thông tin độc giả | ⑧ | Thông tin sách mượn |
| ② | Thẻ độc giả | ⑨ | Thông tin sách trả |
| ③ | Thẻ độc giả | ⑩ | Thông báo sách được trả + phiếu thanh toán tiền phạt nếu có |
| ④ | Cuốn sách muốn mượn | ⑪ | Phiếu mượn sách |
| ⑤ | Thông tin cuốn sách giữ chỗ | ⑫ | Phiếu trả sách |
| ⑥ | Thông tin cuốn sách giữ chỗ đã giữ chỗ | ⑬ | Yêu cầu thống kê |
| ⑦ | Cuốn sách được mượn | ⑭ | Thông tin thống kê |

2. Mô hình quan niệm tổ chức theo phương pháp DFD

2.1 Mô hình cấp 1



2.3 Mô hình cấp 2



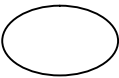
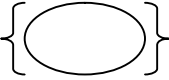


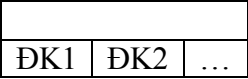
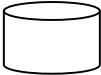

2.4 Mô hình hóa cấp 3 cho ô xử lý mượn sách



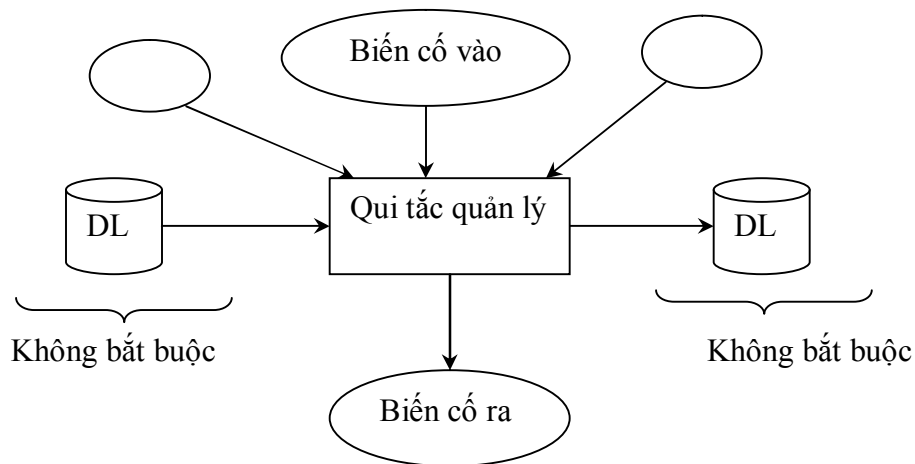
- ❖ **Bài tập cho sinh viên:** tương tự cách làm trên sinh viên tiếp tục xây dựng các cấp cho các qui trình xử lý còn lại.

MÔ HÌNH TỰA MERISE

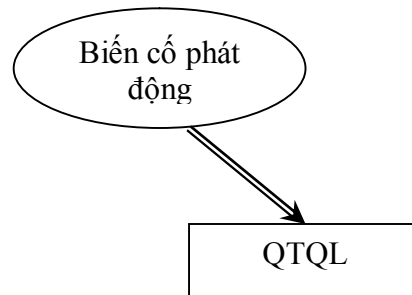
1. Hệ thống ký hiệu

| Ký hiệu | Tên gọi |
|---|---|
|  | Biến cố. Mỗi biến cố có đặc tính là nó thuộc biến cố ngoài môi trường hay biến cố trong hệ thống, có đặc tính là biến cố vào hay biến cố ra. Đối với biến cố vào được phân ra làm hai dạng: loại biến cố phát động và biến cố điều kiện |
|  | Tập các biến cố Tập biến cố gồm hai loại: tập biến cố vào và tập biến cố ra |
|  | Điều kiện phát động biến cố |
|  | Quy tắc quản lý (QTQL) |
|  | Quy tắc quản lý có điều kiện ra |
|  | Dữ liệu. Gồm có hai loại: dữ liệu vào và dữ liệu ra cho của quy tắc xử lý |
|  | Phương tiện biểu diễn dữ liệu trên giấy |

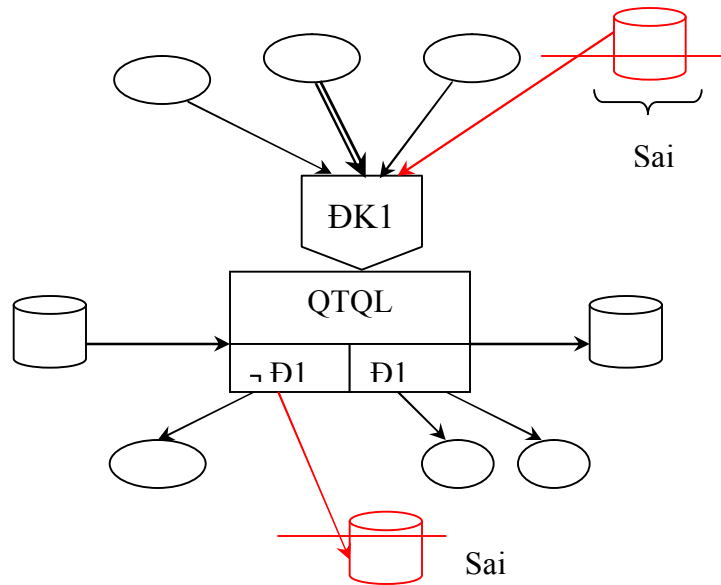
2. Một số lưu ý cho mô hình tựa Merise



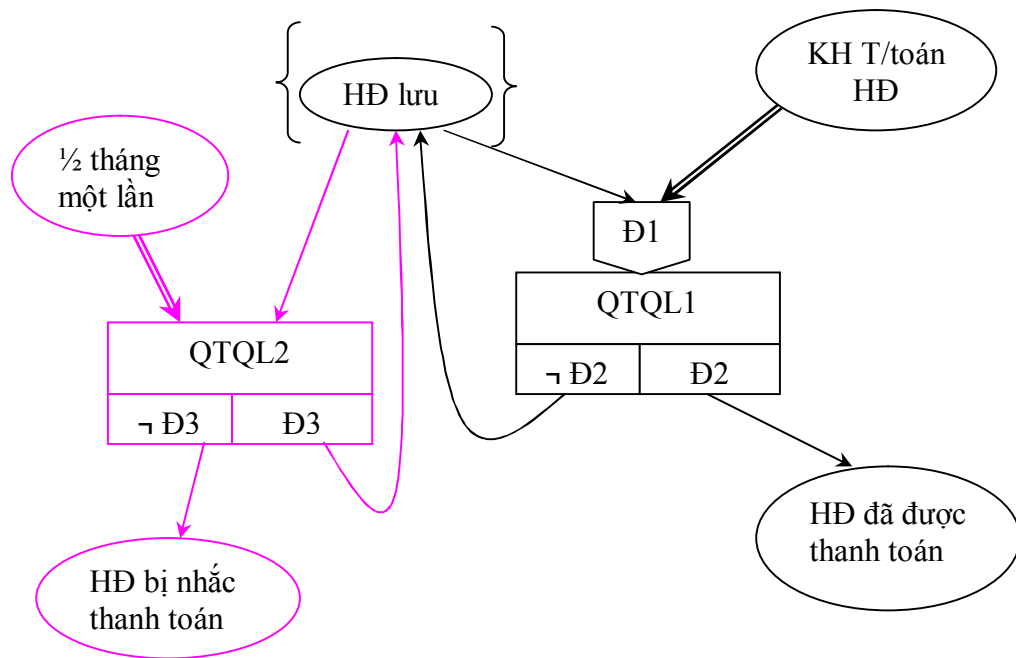
- ❖ Nếu không có gì thì dấu mũi tên đi vào có ý nghĩa “và”, nghĩa là tất cả các biến cố sẵn sàng thì mới bắt đầu thực hiện qui tắc quản lý.
- ❖ Biến cố quan trọng thường xảy ra sau cùng (còn gọi là biến cố phát động), với biến cố phát động thì mũi tên vào được biểu diễn bằng mũi tên kép



- ❖ Trước khi thực hiện qui tắc quản lý có thể có điều kiện phát động và được biểu diễn thành một biểu thức điều kiện. Biểu thức điều kiện gắn liền với nội dung của các biến cố vào.
- ❖ Kết quả ra của một qui tắc quản lý có thể có điều kiện ra và được gọi là điều kiện ra của qui tắc quản lý. Điều kiện ra của qui tắc quản lý gắn liền với nội dung của qui tắc quản lý.
- ❖ Dữ liệu không là một biến cố vào hay ra của một qui tắc xử lý.



- ❖ Bất kỳ một nghiệp vụ quản lý nào cũng phải có điểm dừng. Do vậy khi mô hình có tính chất lặp thường phải có qui tắc quản lý nhắc nhở cho hệ thống. Ví dụ:



Ghi chú:

QTQL1: Xử lý một lần thanh toán hóa đơn

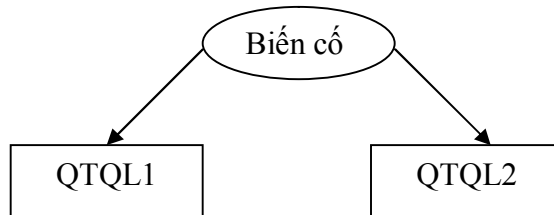
QLQL2: Nhắc thanh toán hóa đơn

Đ1: So khớp số hóa đơn mà khách hàng thanh toán với hóa đơn lưu

Đ2: Hóa đơn thanh toán xong

Đ3: Hóa đơn còn trong hạn thanh toán

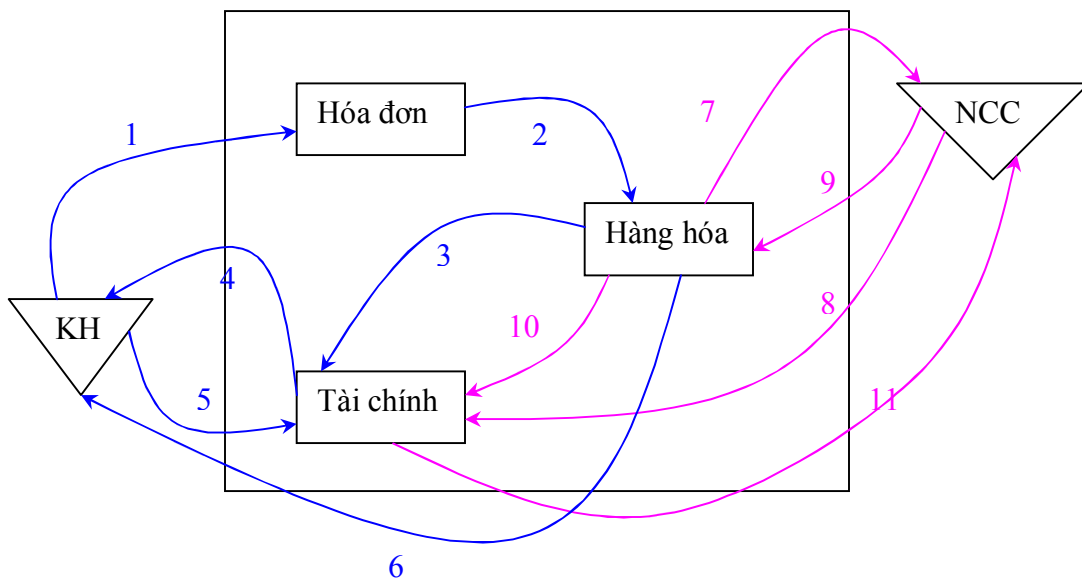
- ❖ Khi một biến cố cùng trở vào hai qui tắc xử lý thì nó mang ý nghĩa là hoặc sẽ thực hiện qui tắc xử lý này, hoặc sẽ thực hiện qui tắc xử lý kia, chứ nó không mang ý nghĩa là dùng chung cho hai qui tắc xử lý cùng xảy ra một lúc.



- ❖ Khi gặp một biến cố phức tạp ta có thể phân rã thành các biến cố ở dạng đơn giản hơn.

3. Phương pháp xây dựng mô hình quan niệm xử lý

- ❖ Bước 1: xây dựng các hệ thống con và lập sơ đồ thông lượng thông tin (giữa hệ thống con hoặc giữa hệ thống con và môi trường bên ngoài). Sau bước này phải nhận diện được tất cả các qui trình của hệ thống con. Ví dụ:

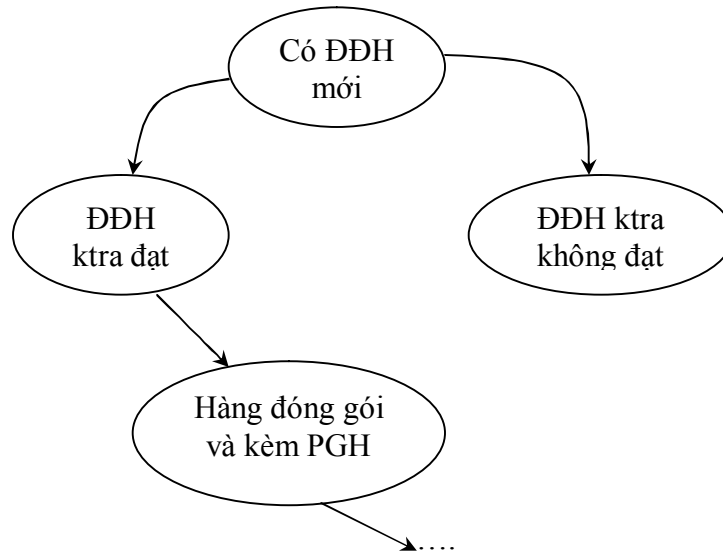


Ghi chú:

- 1: Hóa đơn mua hàng của khách hàng
- 2: Hóa đơn giải quyết
- 3: Hàng mua được giao
- 4: Hóa đơn thanh toán
- 5: Hóa đơn đã thanh toán
- 6: Hàng giao

- 7: Hóa đơn mua hàng
- 8: Hóa đơn thanh toán
- 9: Hàng giao
- 10: Hàng đã giao
- 11: Hóa đơn được thanh toán

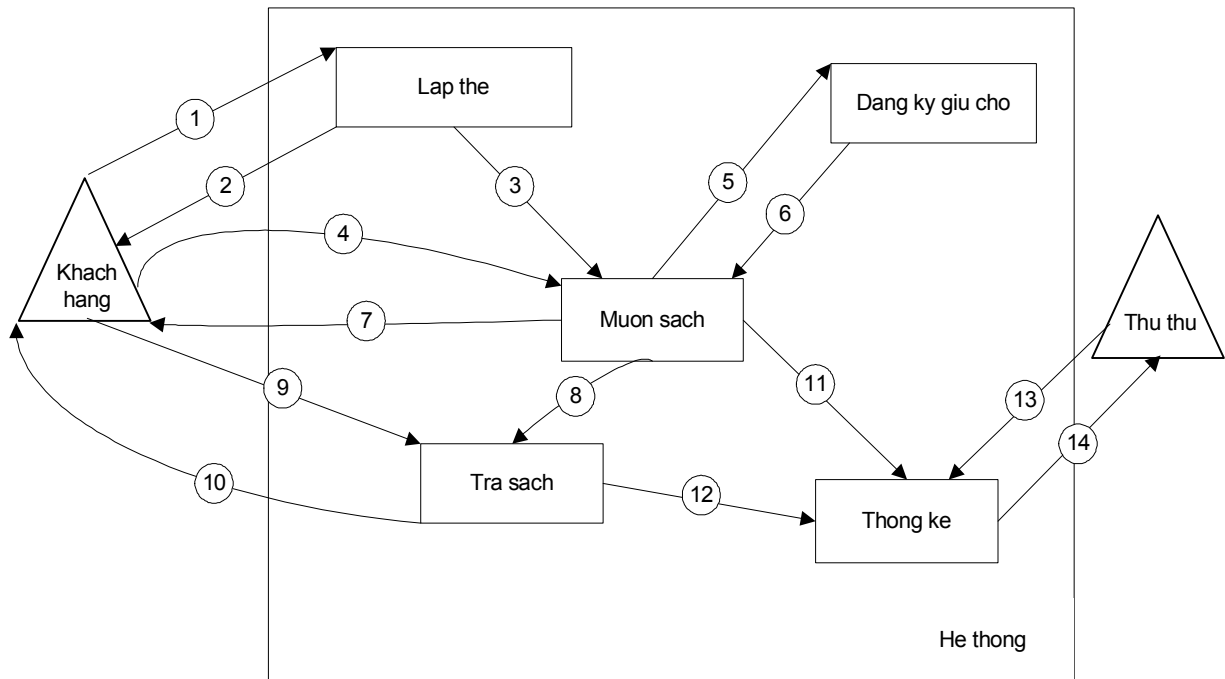
- ❖ Bước 2: Lập sơ đồ liên hoàn các biến cố cho từ qui trình xử lý. Ví dụ: trong hệ thống con Đơn đặt hàng, có qui trình giải quyết đơn đặt hàng, thì qui trình giải quyết đơn đặt hàng được biểu diễn như sau:



-
- ❖ Bước 3: Lập mô hình quan niệm xử lý cho từng qui trình

XÂY DỰNG MÔ HÌNH QUAN NIỆM XỬ LÝ THEO PHƯƠNG PHÁP TỰA MERISE

1. Xác định sơ đồ hệ thống con của qui trình

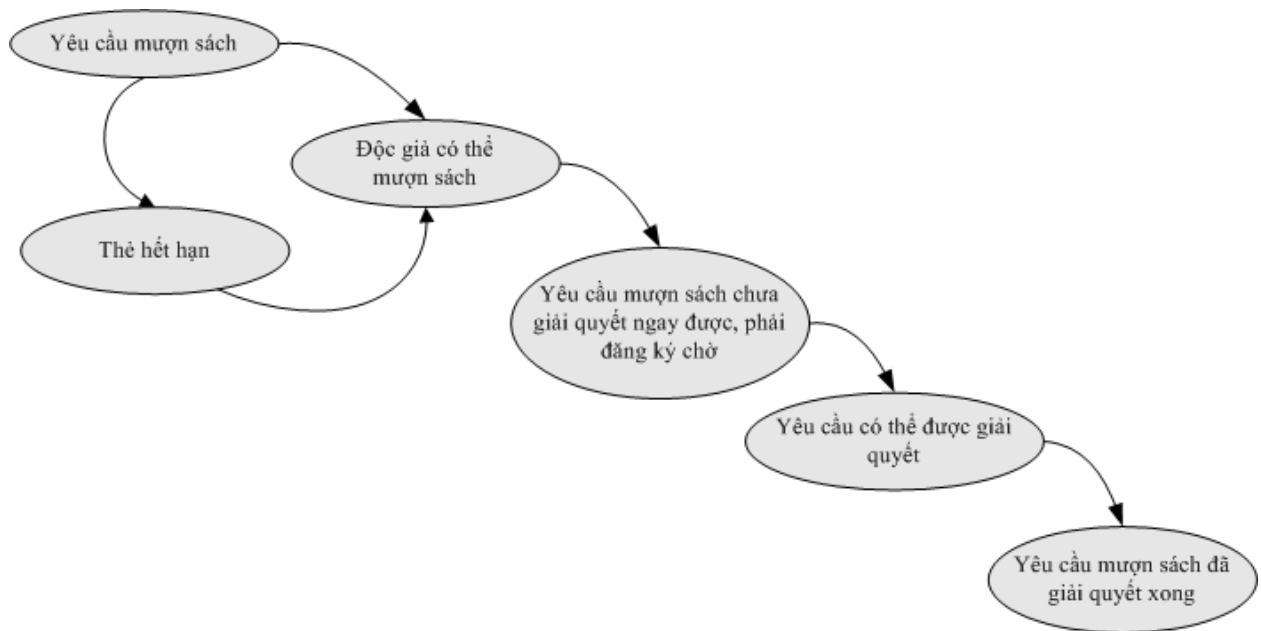


Mô tả sơ đồ dòng dữ liệu cho sơ đồ hệ thống con của qui trình

| Số | Diễn giải | Số | Diễn giải |
|----|--|----|---|
| ① | Thông tin độc giả | ⑧ | Thông tin sách mượn |
| ② | Thẻ độc giả | ⑨ | Thông tin sách trả |
| ③ | Thẻ độc giả | ⑩ | Thông báo sách được trả + phiếu thanh toán tiền phạt nếu có |
| ④ | Cuốn sách muốn mượn | ⑪ | Phiếu mượn sách |
| ⑤ | Thông tin cuốn sách giữ chỗ | ⑫ | Phiếu trả sách |
| ⑥ | Thông tin cuốn sách giữ chỗ đã giữ chỗ | ⑬ | Yêu cầu thống kê |
| ⑦ | Cuốn sách được mượn | ⑭ | Thông tin thống kê |

2. XÂY DỰNG MÔ HÌNH QUAN NIỆM XỬ LÝ CHO QUI TRÌNH MƯỢN SÁCH

2.1 Sơ đồ liên hoàn các biến cố

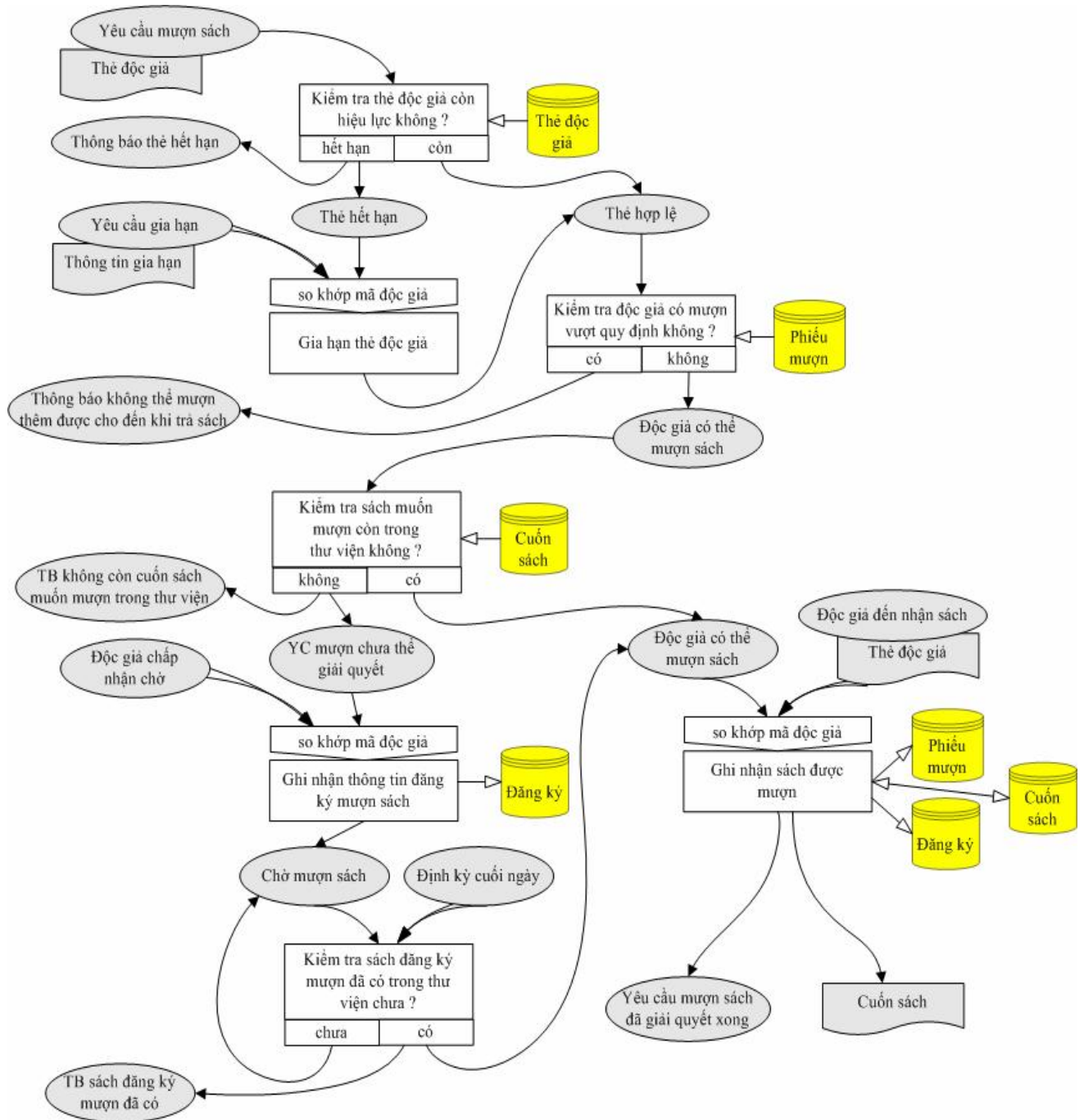


2.2. Mô tả quan niệm xử lý theo Tựa Merise

2.2.1 Mô tả các nguyên tắc biến cố vào ra

| STT | Nguyên tắc xử lý | Biến cố vào | Biến cố ra |
|-----|---|---|--|
| 1 | Kiểm tra thẻ độc giả còn hiệu lực không ? | Yêu cầu mượn sách + Thẻ độc giả | Kết thúc qui trình |
| | | | Thông tin về tồn kho các loại thuốc |
| 2 | Gia hạn thẻ độc giả | Thẻ hết hạn + Yêu cầu gia hạn + Thông tin gia hạn | Thẻ hợp lệ (có thể thực hiện việc mượn sách) |
| ... | | | ... |

2.2.2 Mô hình quan niệm xử lý quy trình mượn sách theo Tựa Merise

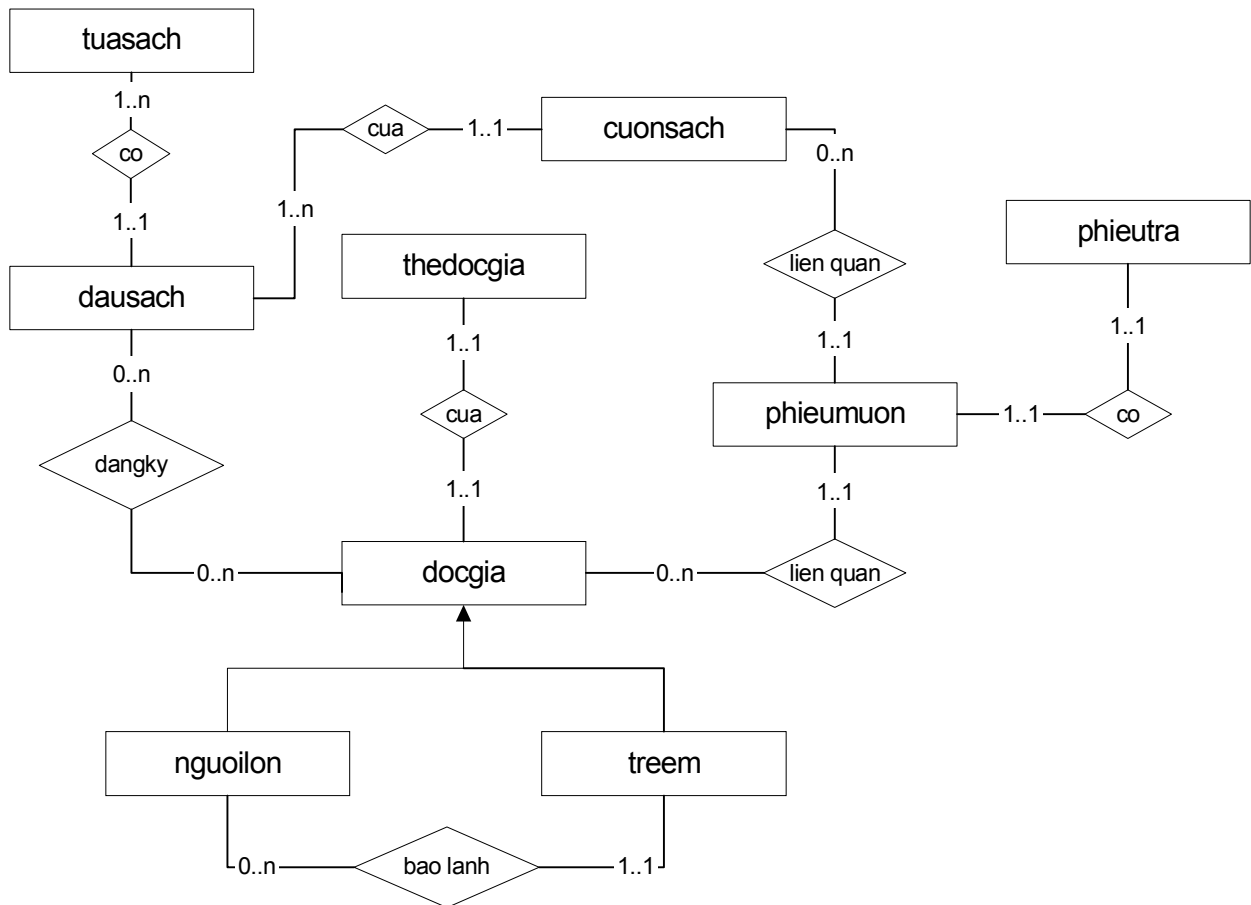


Bài tập:

1. Sinh viên bổ sung đầy đủ các thông tin cho bảng mô tả các nguyên tắc xử lý còn thiếu dựa vào mô hình xử lý tựa Merise cho quy trình mượn sách.
2. Xây dựng sơ đồ liên hoàn các biến cố và mô hình xử lý cho quy trình trả sách.

CHUYỂN ĐỔI TỪ MÔ HÌNH ER SANG MÔ HÌNH DỮ LIỆU QUAN HỆ

I. Mô hình dữ liệu ER



II. Sự chuyển đổi từ mô hình dữ liệu ER sang mô hình dữ liệu quan hệ

Dựa theo 6 bước trong phương pháp chuyển đổi từ một mô hình dữ liệu ER sang mô hình quan hệ dữ liệu, chúng ta có mô hình quan hệ dữ liệu như sau:

docgia(ma docgia, ho, tenlot, ten, hinh).

thedocgia(ma docgia, ngaylapthe, ngayhethan)

nguoilon(ma docgia, sonha, duong, quan, dienthoai, ngaysinh)

treem(ma docgia,ma docgia nguoilon, ngaysinh)

tuasach(ma tuasach, tuasach, tacgia, tomtat)

dausach(*isbn,ma tuasach, ngonngu, bia, trangthai*)

cuonsach(ma cuonsach, isbn, tinhtrang)

dangky(isbn,ma docgia,ngay dk,ghichu)

phieumuon(isbn, ma cuonsach,ma docgia, ngaymuon, ngaytra)

phieutra(*isbn, ma cuonsach,ma docgia,ngaymuon,ngaytrathatsu, tienphat*)

III. Nhận xét và đề nghị một số mô hình quan hệ dữ liệu khác

- **Nhận xét 1 :** Vì theo yêu cầu của đề bài mã thẻ độc giả chính là mã độc giả, do vậy khóa chính của quan hệ **thedocgia** chính là **ma_docgia**. Nhưng câu hỏi đặt ra ở đây là chúng ta có nên có quan hệ **thedocgia** này hay không? Nếu có thì mục đích để làm gì? Chúng ta nhận thấy thông tin cần thiết cho một độc giả đã thể hiện đầy đủ trên quan hệ **docgia(nguoiilon,treem)**, như vậy chức năng chính của quan hệ **thedocgia** là chỉ cung cấp thông tin *ngaylaple* và *ngayhethan*, mà hai thuộc tính này được truy xuất thường xuyên để kiểm tra tình trạng thẻ độc giả có còn hợp lệ không khi độc giả mượn sách. Vậy rõ ràng khi có yêu cầu hệ thống kết xuất thông tin độc giả cùng với thông tin thẻ độc giả thì chúng ta phải thực hiện một phép kết giữa hai quan hệ **docgia** và **thedocgia** mới lấy được đầy đủ thông tin, nhưng giả sử hệ thống chúng ta rất quan tâm đến tốc độ của việc truy xuất thông tin, thì việc thực hiện phép kết này là không nên chút nào. Do vậy chúng ta cần nên gộp quan hệ **thedocgia** vào trong quan hệ **docgia**, vì khi chúng ta để lại quan hệ **thedocgia** thì cũng không có ích lợi gì hơn cho hệ thống. Do đó quan hệ **docgia** và **thedocgia** được gộp chung thành quan hệ **docgia** như sau:

docgia(*ma_docgia*, *ho*, *tenlot*, *ten*, *hinh*, *ngaylapthe*, *ngayhethan*).

- **Nhận xét 2:** Tiếp theo chúng ta thấy việc tồn tại của quan hệ **phieumuon** và **phieutra** là có nên hay không? Rõ ràng là chúng ta thấy không nên vì phải chấp nhận một sự trùng lặp dữ liệu quá lớn, muốn truy xuất thông tin thì phải kết giữa hai bảng (công việc được khuyến cáo là cần hạn chế càng tốt vì phép kết làm chậm tốc độ truy xuất thông tin), đồng thời với việc lưu trữ dư thừa như vậy sẽ làm cho kích thước lưu trữ trong hệ thống bị phình to.

Dữ liệu minh họa cho hai bảng **phieumuon** và **phieutra** (dữ liệu trùng lặp quá nhiều, lưu trữ thừa làm cho kích thước hệ thống bị phình to)

1. Phieumuon

| isbn | ma_cuonsach | ma_docgia | ngaymuon |
|------|-------------|-----------|------------|
| 1 | 1 | 1 | 01/01/2004 |
| 1 | 1 | 1 | 10/2/2004 |
| 1 | 1 | 1 | 10/6/2004 |
| 1 | 1 | 1 | 10/8/2004 |
| ... | ... | ... | ... |
| ... | ... | ... | ... |

2. Phieutra

| isbn | ma_cuonsach | ma_docgia | ngaytra | ngaytrathatsu | tienphat |
|------|-------------|-----------|------------|---------------|----------|
| 1 | 1 | 1 | 15/01/2004 | 10/01/2004 | 0 |
| 1 | 1 | 1 | 24/02/2004 | 10/03/2004 | 50.000 |
| 1 | 1 | 1 | 24/06/2004 | 20/06/2004 | 0 |
| 1 | 1 | 1 | 24/08/2004 | 20/8/2004 | 0 |
| ... | ... | ... | ... | ... | ... |
| ... | ... | ... | ... | ... | ... |

Do vậy ý tưởng đầu tiên là chúng ta nên gộp hai quan hệ **phieumuon** và **phieutra** thành quan hệ **muontra** như sau:

muontra(*isbn*, *ma_cuonsach*, *ma_docgia*, *ngaymuon*, *ngaytra*, *ngaytrathatsu*, *tienphat*)

Dữ liệu minh họa cho quan hệ **muontra**

| isbn | ma_cuonsach | ma_docgia | ngaymuon | ngaytra | Ngaytrathatsu | tienphat |
|------|-------------|-----------|------------|------------|---------------|----------|
| 1 | 1 | 1 | 01/01/2004 | 15/01/2004 | 10/01/2004 | 0 |
| 1 | 1 | 1 | 10/2/2004 | 24/02/2004 | 10/03/2004 | 50.000 |
| 1 | 1 | 1 | 10/6/2004 | 24/06/2004 | 20/06/2004 | 0 |
| 1 | 1 | 1 | 10/8/2004 | 24/08/2004 | 20/8/2004 | 0 |
| ... | ... | ... | ... | ... | ... | ... |
| ... | ... | ... | ... | ... | ... | ... |

- **Nhận xét 3:** với quan hệ **muontra** mới chúng ta nhận thấy rằng vẫn còn sự trùng lặp dữ liệu nhưng chúng ta dễ dàng truy xuất nhanh các thông tin mượn trả sách cho thư viện. Với quan hệ này rất phù hợp cho một hệ thống yêu cầu về mặt tốc độ truy xuất thông tin nhanh kể cả chấp nhận sự trùng lặp dữ liệu. Nhưng nếu hệ thống yêu cầu không được trùng lặp thông tin thì sao? Lúc này chắc chắn chúng ta không nên để nguyên quan hệ **muontra** mà phải nên phân thành hai quan hệ **muon** và **qtrinhmuon** như sau:

muon(*isbn, ma_cuonsach, ma_docgia*)

qtrinhmuon(*isbn, ma_cuonsach, ngaymuon, ma_docgia, ngay tra, ngaytrathatsu tienphat*)

Dữ liệu minh họa cho hai quan hệ **muon** và **qtrinhmuon**

1. **muon**

| isbn | ma_cuonsach | ma_docgia |
|------|-------------|-----------|
| 1 | 1 | 1 |

2. **qtrinhmuon**

| isbn | ma_cuonsach | ngaymuon | ma_docgia | ngaytra | ngaytrathatsu | tienphat |
|------|-------------|------------|-----------|------------|---------------|----------|
| 1 | 1 | 01/01/2004 | 1 | 15/01/2004 | 10/01/2004 | 0 |
| 1 | 1 | 10/2/2004 | 1 | 24/02/2004 | 10/03/2004 | 50.000 |
| 1 | 1 | 10/6/2004 | 1 | 24/06/2004 | 20/06/2004 | 0 |
| 1 | 1 | 10/8/2004 | 1 | 24/08/2004 | 20/8/2004 | 0 |
| ... | ... | ... | ... | ... | ... | ... |
| ... | ... | ... | ... | ... | ... | ... |

- **Nhận xét 4:** với dữ liệu minh họa trên ta thấy có giảm bớt sự trùng lặp thông tin và kích thước cho hệ thống. Nhưng nếu chúng ta muốn hệ thống lưu trữ tốt nhất, không bị trùng lặp thông tin, cũng như kích thước hệ thống càng nhỏ càng tốt thì sao? Lúc đó ta vẫn chọn phương án trên, nhưng để tránh việc lưu trữ nhiều và phép kết xảy ra giữa hai bảng thực hiện nhanh hơn ta chọn lại cách đặt khóa chính trên bảng **muon**, nghĩa là ta không chọn 3 thuộc tính *isbn, ma_cuonsach, ma_docgia* làm khóa chính mà chúng ta thêm vào một thuộc tính *ma_phieumuon* để làm khóa chính. Như vậy lúc đó hai quan hệ **muon**, và **qtrinhmuon** có sự thay đổi như sau:

muon(*ma_phieumuon, isbn, ma_cuonsach, ma_docgia*)

qtrinhmuon(*ma_phieumuon, ngaymuon, ngay tra, ngaytrathatsu tienphat*)

Dữ liệu minh họa cho hai bảng

1. **muon**

| ma_phieumuon | isbn | ma_cuonsach | ma_docgia |
|---------------------|-------------|--------------------|------------------|
| PM01 | 1 | 1 | 1 |

2. **qtrinhmuon**

| Ma_phieumuon | ngaymuon | ngaytra | Ngaytrathatsu | tienphat |
|---------------------|-------------------|-------------------|----------------------|-----------------|
| PM01 | 01/01/2004 | 15/01/2004 | 10/01/2004 | 0 |
| PM01 | 10/2/2004 | 24/02/2004 | 10/03/2004 | 50.000 |
| PM01 | 10/6/2004 | 24/06/2004 | 20/06/2004 | 0 |
| PM01 | 10/8/2004 | 24/08/2004 | 20/8/2004 | 0 |
| ... | ... | ... | ... | ... |
| ... | ... | ... | ... | ... |

IV. Kết luận

Như vậy dựa trên sự phân tích và nhận xét ở trên chúng ta có thể có những mô hình quan hệ có thể sử dụng:

- a. **Mô hình 1:** trường hợp xấu nhất(trùng lặp, kết xuất chậm, kích thước hệ thống lớn) ☹

docgia(*ma_docgia*, *ho*, *tenlot*, *ten*, *hinh*).
thedocgia(*ma_docgia*, *ngaylapthe*, *ngayhethan*)
nguoiilon(*ma_docgia*, *sonha*, *duong*, *quan*, *dienthoai*, *ngaysinh*)
treem(*ma_docgia*, *ma_docgia_nguoiilon*, *ngaysinh*)
tuasach(*ma_tuasach*, *tuasach*, *tacgia*, *tomtat*)
dausach(*isbn*, *ma_tuasach*, *ngonngu*, *bia*, *trangthai*)
cuonsach(*ma_cuonsach*, *isbn*, *tinhttrang*)
dangky(*isbn*, *ma_docgia*, *ngay_dk*, *ghichu*)
phieumuon(*isbn*, *ma_cuonsach*, *ma_docgia*, *ngaymuon*, *ngaytra*)
phieutra(*isbn*, *ma_cuonsach*, *ma_docgia*, *ngaymuon*, *ngaytrathatsu*, *tienphat*)

- b. **Mô hình 2:** (hệ thống truy xuất nhanh, chấp nhận sự trùng lặp thông tin)

docgia(*ma_docgia*, *ho*, *tenlot*, *ten*, *hinh*, *ngaylapthe*, *ngayhethan*)
nguoiilon(*ma_docgia*, *sonha*, *duong*, *quan*, *dienthoai*, *ngaysinh*)
treem(*ma_docgia*, *ma_docgia_nguoiilon*, *ngaysinh*)
tuasach(*ma_tuasach*, *tuasach*, *tacgia*, *tomtat*)
dausach(*isbn*, *ma_tuasach*, *ngonngu*, *bia*, *trangthai*)
cuonsach(*ma_cuonsach*, *isbn*, *tinhttrang*)
dangky(*isbn*, *ma_docgia*, *ngay_dk*, *ghichu*)
muontra(*isbn*, *ma_cuonsach*, *ma_docgia*, *ngaymuon*, *ngaytra*, *ngaytrathatsu*, *tienphat*)

- c. **Mô hình 3:** hệ thống vẫn còn sự trùng lặp thông tin, khi thực hiện phép kết vẫn còn phải kết trên cả ba thuộc tính khóa chính của quan hệ muon

```
docgia(ma_docgia, ho, tenlot, ten, hình, ngaylapthe, ngayhethan)
nguoiilon(ma_docgia, sonha, duong, quan, dienthoai, ngaysinh)
treem(ma_docgia, ma_docgia_nguoiilon, ngaysinh)
tuasach(ma_tuasach, tuasach, tacgia, tomtat)
dausach(isbn, ma_tuasach, ngonngu, bia, trangthai)
cuonsach(ma_cuonsach, isbn, tinhtrang)
dangky(isbn, ma_docgia, ngay_dk, ghichu)
muon(isbn, ma_cuonsach, ma_docgia)
qtrinhmuon(isbn, ma_cuonsach, ngaymuon, ma_docgia, ngay tra, ngaytrathatsutienphat)
```

- d. **Mô hình 4:** hệ thống không còn sự trùng lặp thông tin

```
docgia(ma_docgia, ho, tenlot, ten, hình, ngaylapthe, ngayhethan)
nguoiilon(ma_docgia, sonha, duong, quan, dienthoai, ngaysinh)
treem(ma_docgia, ma_docgia_nguoiilon, ngaysinh)
tuasach(ma_tuasach, tuasach, tacgia, tomtat)
dausach(isbn, ma_tuasach, ngonngu, bia, trangthai)
cuonsach(ma_cuonsach, isbn, tinhtrang)
dangky(isbn, ma_docgia, ngay_dk, ghichu)
muon(ma_phieumuon, isbn, ma_cuonsach, ma_docgia)
qtrinhmuon(ma_phieumuon, ngaymuon, ngay tra, ngaytrathatsutienphat)
```

- Như vậy: tùy theo yêu cầu của hệ thống mà chúng ta lựa chọn một mô hình dữ liệu quan hệ thích hợp để cài đặt.
- ❖ Bài tập sinh viên:
- ✓ Phân tích và tìm ra nhiều mô hình quan hệ khác (có đánh giá ưu và khuyết điểm của chúng)
 - ✓ Tìm và mô tả tất cả các ràng buộc toàn vẹn cho từng mô hình.

MÔ HÌNH TỔ CHỨC XỬ LÝ TỰA MERISE

1. Lập bảng thủ tục chức năng

| STT | Nguồn gốc NTQL (1) | Thủ tục chức năng(2) | Bản chất (3) | Vị trí(4) | Thời gian thực hiện (5) |
|-----|--------------------|----------------------|--------------|-----------|-------------------------|
| | | | | | |
| | | | | | |
| | | | | | |

Ghi chú: Bảng thủ tục chức năng được mô tả cho một qui trình xử lý

(1): các thủ tục chức năng được lấy từ nguồn gốc NTQL nào

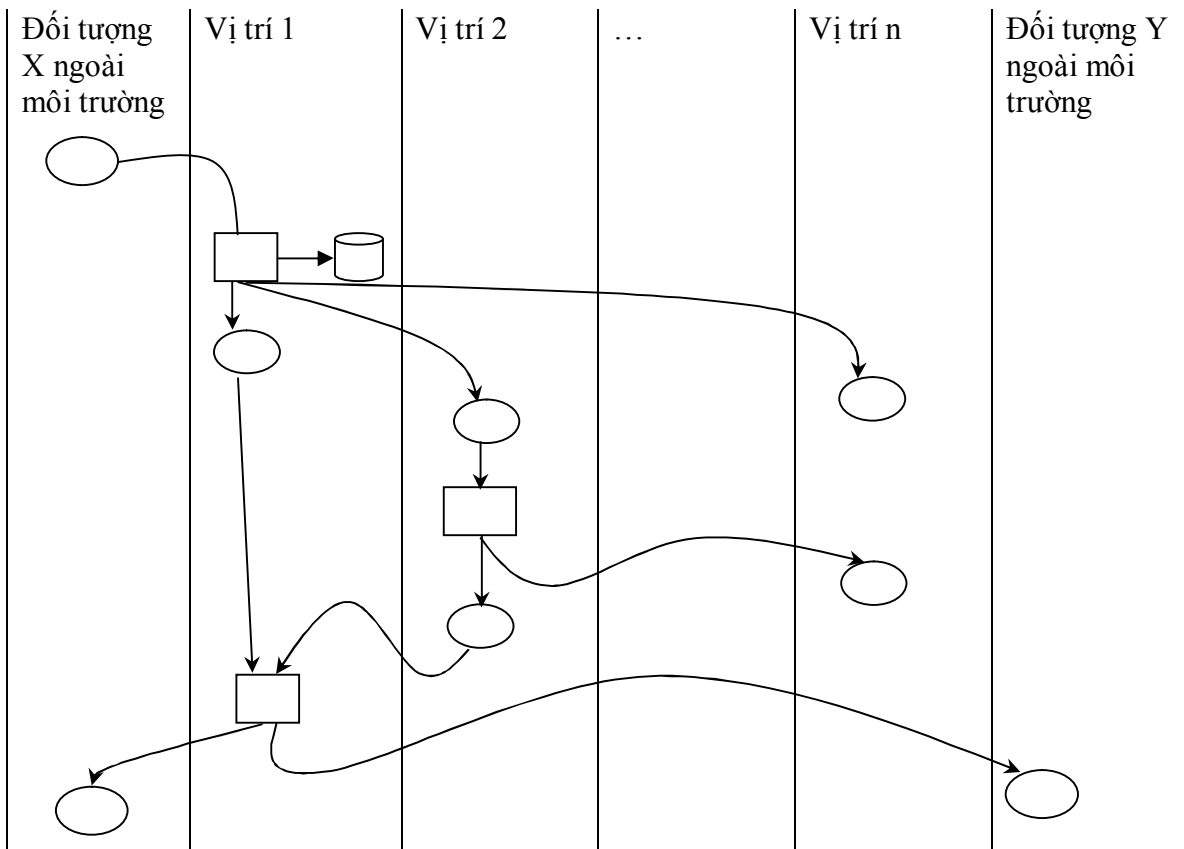
(2): Là thứ tự các công việc cho qui trình xử lý

(3): Bản chất công việc được thực hiện là thủ công hay tự động

(4): Thủ tục chức năng đó được thực hiện ở vị trí nào.

(5): Bao giờ thì thực hiện công việc đó (thực hiện ngay, cuối ngày,....)

2. Sơ đồ tổ chức xử lý theo thời gian, vị trí



XÂY DỰNG MÔ HÌNH QUAN NIỆM TỔ CHỨC XỬ LÝ THEO PHƯƠNG PHÁP TỰA MERISE

Mô hình tổ chức xử lý tựa Merise:

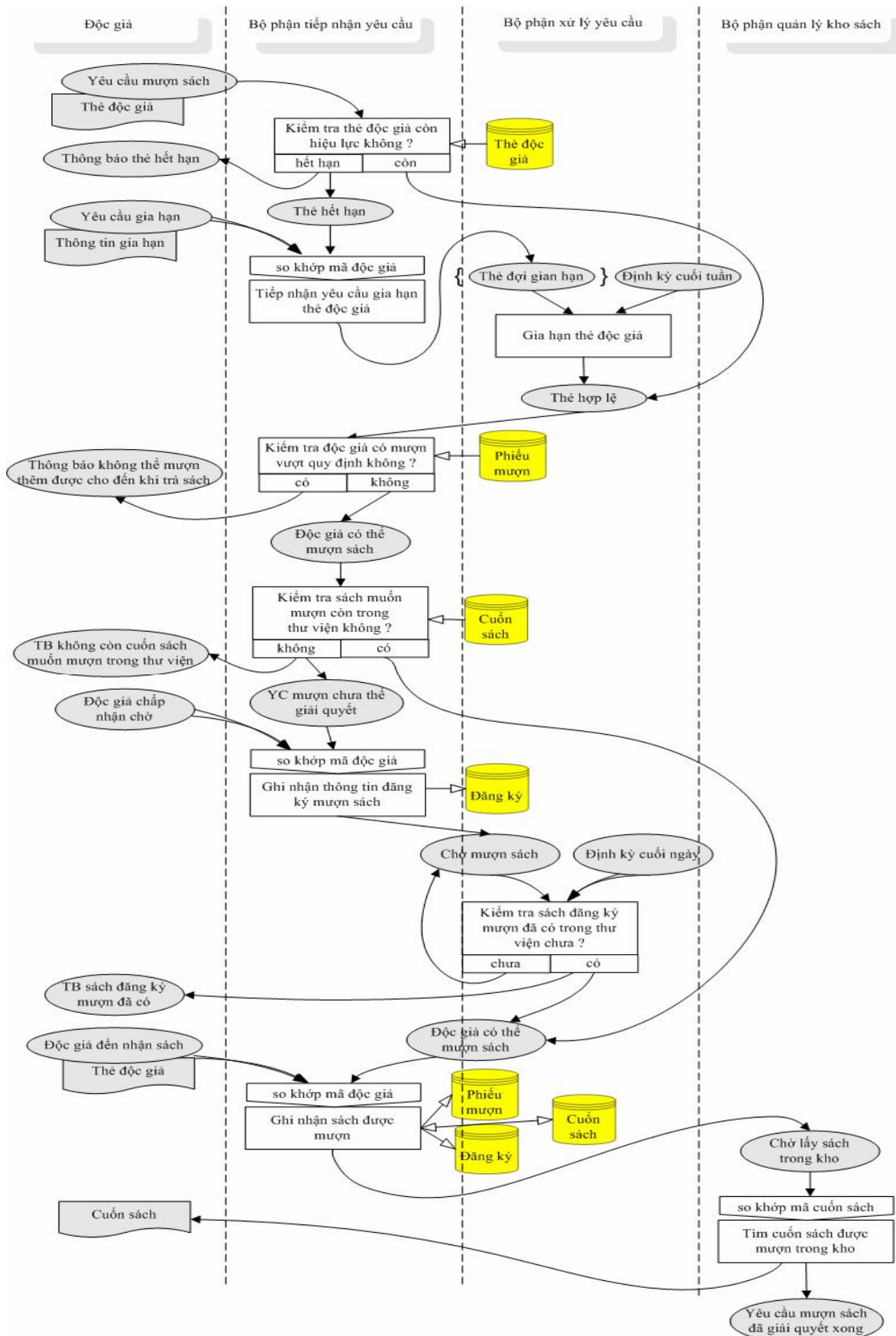
Giới thiệu:

Trong các mô hình quan niệm xử lý của quy trình mượn sách có liên quan đến các đối tượng và bộ phận sau: độc giả, bộ phận tiếp nhận yêu cầu, bộ phận xử lý yêu cầu, bộ phận quản lý kho sách.

Bảng các thủ tục chức năng:

| STT | Nguồn gốc NTQL | Thủ tục chức năng | Bản chất | Vị trí | Thời gian |
|-----|--|--|----------|----------------------|----------------|
| 1. | Kiểm tra thẻ độc giả còn hiệu lực không ? | Tiếp nhận yêu cầu mượn sách + thẻ độc giả | TC | BP tiếp nhận yêu cầu | thực hiện ngay |
| 2. | | Kiểm tra thẻ độc giả còn hiệu lực không? | TĐ | nt | nt |
| 3. | | Thông báo nếu thẻ hết hạn | Bán TĐ | nt | nt |
| 4. | Gia hạn thẻ độc giả | Tiếp nhận yêu cầu gia hạn | TC | nt | nt |
| 5. | | Chuyển yêu cầu gia hạn cho bộ phận xử lý yêu cầu | TC | nt | cuối ngày |
| 6. | | Gia hạn thẻ độc giả | Bán TĐ | BP xử lý yêu cầu | cuối tuần |
| 7. | Kiểm tra độc giả có mượn vượt quy định không ? | Kiểm tra số sách đang mượn | TĐ | BP tiếp nhận yêu cầu | thực hiện ngay |
| 8. | | Thông báo nếu số sách đang mượn đã đủ quy định | Bán TĐ | nt | nt |
| 9. | Kiểm tra sách muốn mượn còn quyền nào trong thư viện không ? | Kiểm tra sách muốn mượn còn quyền nào trong thư viện không? | TĐ | nt | nt |
| 10. | | Thông báo đăng ký chờ nếu không còn quyền nào trong thư viện | Bán TĐ | nt | nt |
| 11. | ... | ... | ... | ... | ... |

Mô hình tổ chức xử lý:



Bài tập:

1. Sinh viên bổ sung đầy đủ các thủ tục vào bảng thủ tục chức năng ở trên của quy trình mượn sách dựa vào mô hình tổ chức xử lý tựa Merise.
2. Xây dựng mô hình tổ chức xử lý cho quy trình trả sách.

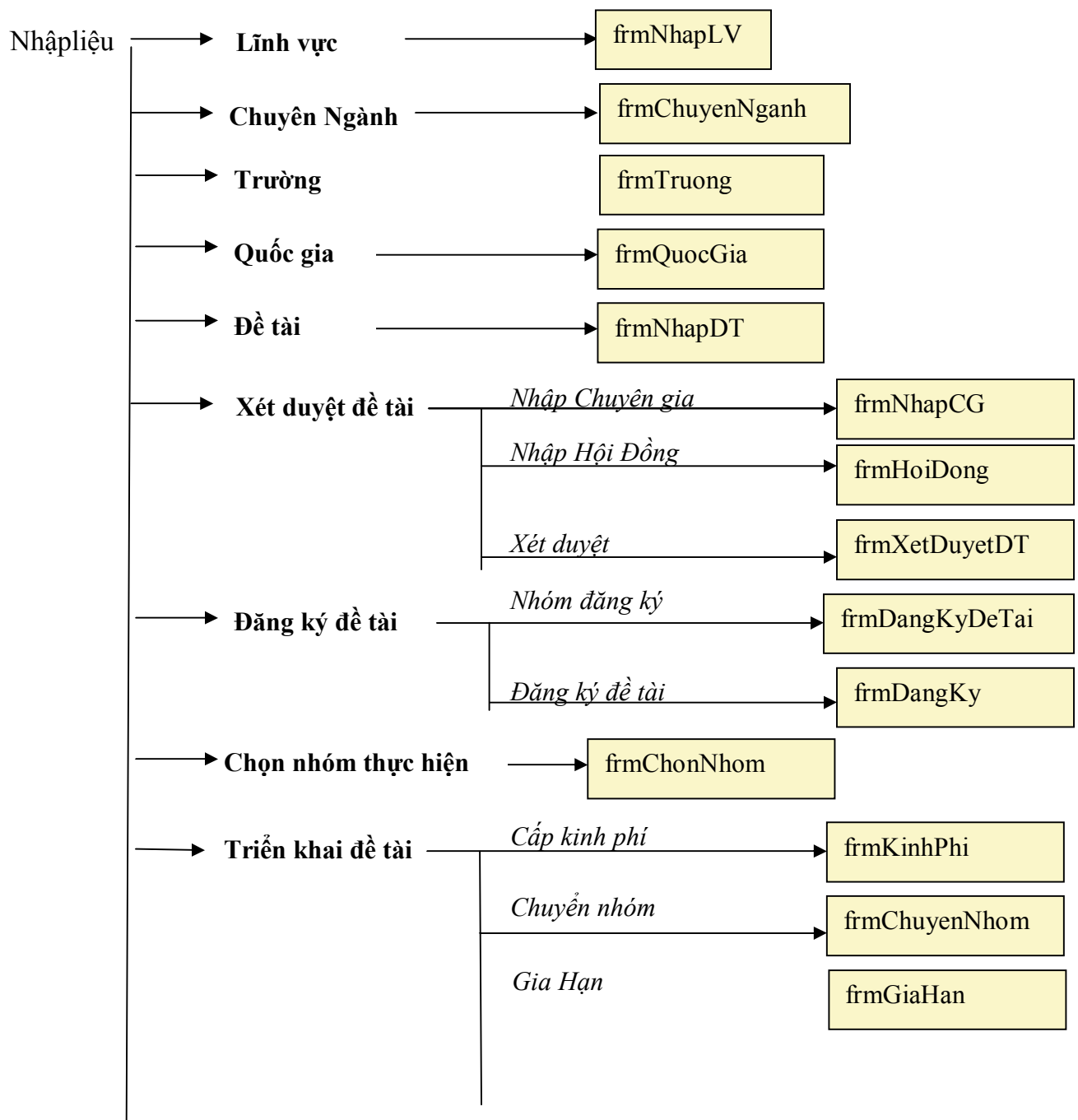
Thiết kế giao diện

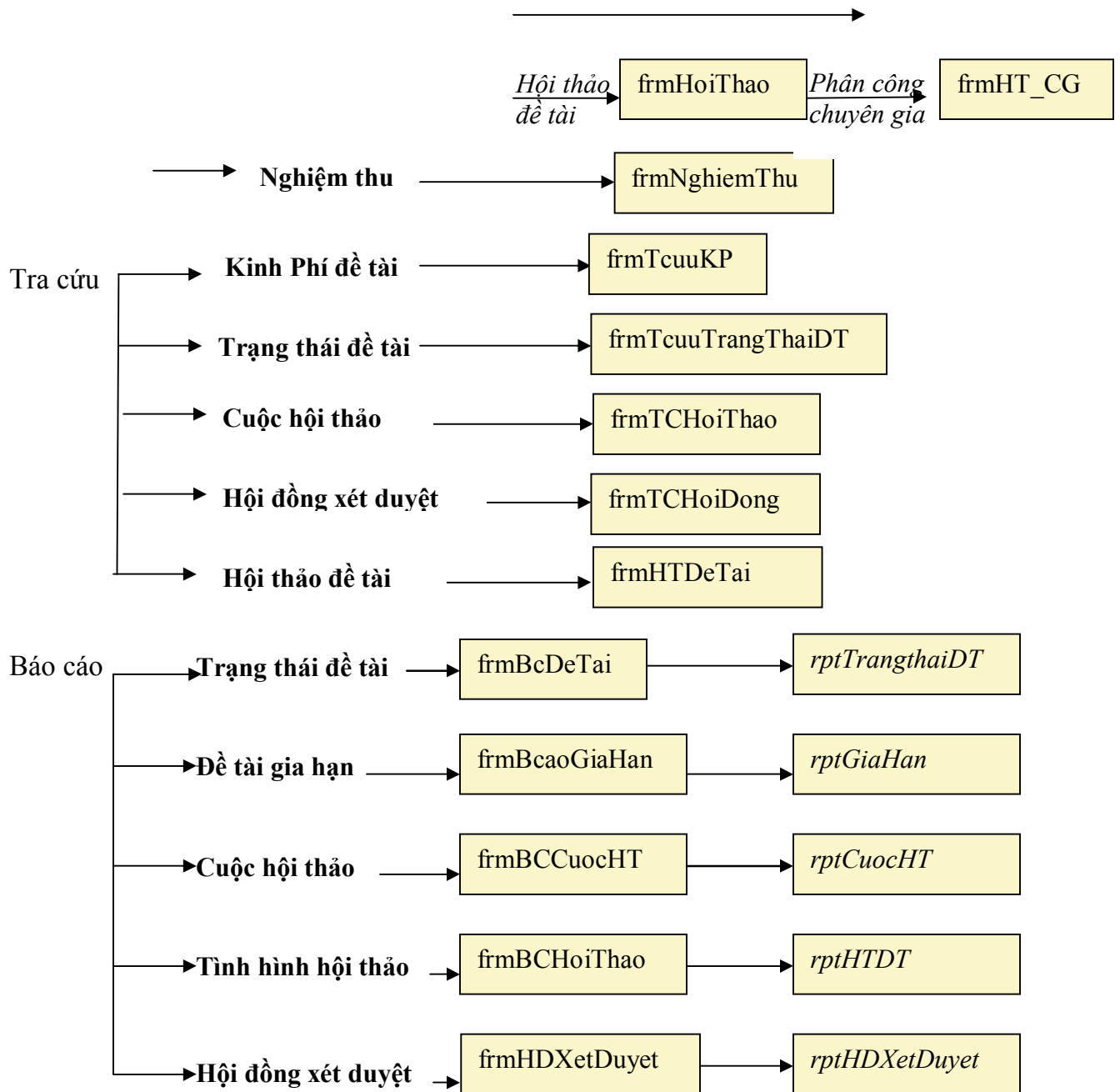
Để thiết kế giao diện, đầu tiên dựa vào các chức năng và dữ liệu mà ta đã phân tích, ta có thể xác định được bao nhiêu giao diện cần phải sử dụng, mục đích của từng giao diện và quan hệ giữa chúng như thế nào.

I Thiết kế giao diện

Chúng ta phải xác định được chương trình sẽ gồm có tất cả bao nhiêu giao diện, bao nhiêu loại và giữa các giao diện đó tương tác với nhau như thế nào.

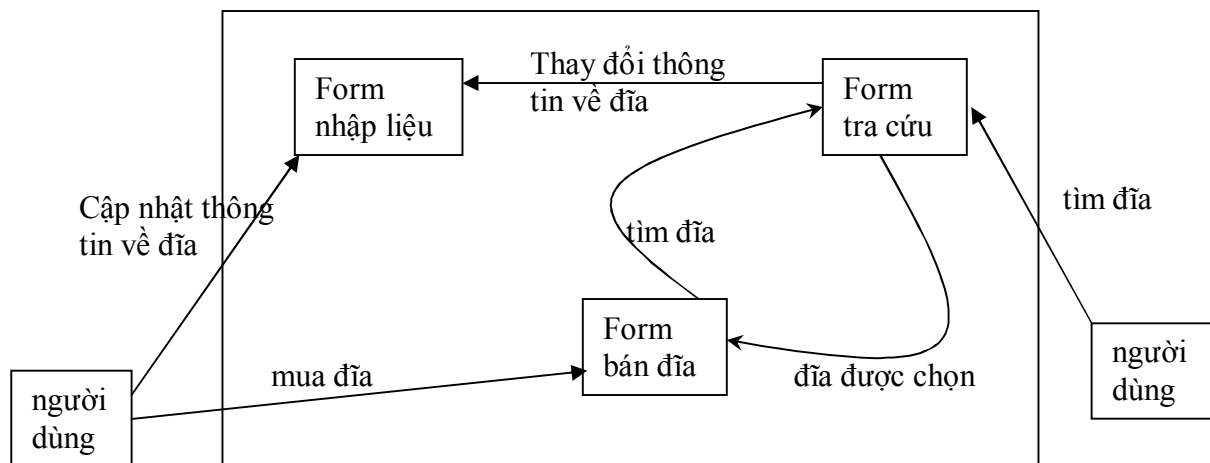
Đối với chương trình có các chức năng riêng biệt nhau, ta thể hiện như sau:





Đối với các chương trình có các giao diện tương tác với nhau, ta sử dụng mô hình DFD để diễn tả mối quan hệ giữa các giao diện.

Ví dụ: Đối với chương trình quản lý đĩa CD gồm có 3 form cơ bản sau: form nhập liệu về đĩa CD, form tìm kiếm đĩa và form bán đĩa. Ta có mô hình sau:



II Trình bày giao diện:

Chúng ta có thể sử dụng tất cả các công cụ từ đơn giản như Winword, Paint... đến phức tạp như Photoshop, Corel... để hỗ trợ việc trình bày từng giao diện. Kết hợp với bản mô tả (sưu liệu) về giao diện đó giúp cho người lập trình dễ dàng thực hiện.

VD:

Hội Đồng

Các Hội Đồng hiện có:

Mã: HD01
Tên: HD1

Thông tin Hội Đồng:

Mã Hội Đồng: HD01
Tên HD: HD1
Số Chuyên Gia: 3

Thêm Cập nhật Xóa

Danh sách đề tài được Hội đồng xét duyệt:

| Mã đề tài | Tên Đề tài | Ngày Bắt Đầu... | Ngày kết thúc... | Kết quả |
|-----------|--------------------|-----------------|------------------|---------|
| DT001 | Nghiên cứu... | 7/22/2004... | 7/22/2004... | 1 |
| DT003 | Phần mềm | 7/22/2004... | 7/22/2004... | 1 |
| DT004 | Phần cứng | 7/15/2004... | 7/16/2004... | 1 |
| DT005 | Ấm thực | 7/14/2004... | 7/15/2004... | 1 |
| DT006 | Lịch sử thời cổ... | 7/15/2004... | 7/16/2004... | 1 |
| DT007 | Đặc nhân tâm | 7/14/2004... | 7/15/2004... | 1 |
| DT008 | Quản lý thư... | 7/22/2000... | 7/22/2000... | 1 |
| DT009 | a | 7/22/2004... | 7/22/2004... | 1 |

Phân công xét duyệt đề tài Thoát

Phân Công chuyên gia vào Hội Đồng

DS Chuyên Gia không thuộc Hội Đồng

| Mã | Họ Tên | Lĩnh Vực |
|-------|-------------|------------|
| CG004 | Yến Phương | Văn học |
| CG005 | Vinh Phúc | Nghệ thuật |
| CG006 | Tiến Thành | Sử học |
| CG007 | Linh Phương | Sinh học |
| CG008 | An Nghi | Công nghệ |
| CG009 | Huyền Nhân | Nghệ thuật |
| CG010 | Sinh | Nghệ thuật |
| CG011 | ... | ... |

DS Chuyên Gia thuộc Hội Đồng:

| Mã | Họ Tên | Lĩnh Vực |
|-------|-----------|-----------|
| CG001 | Mai Ngọc | Toán học |
| CG002 | Phú Thành | Công nghệ |
| CG003 | Nhật Hoa | Công nghệ |

Tham gia 0 hội đồng

Sưu liệu

| STT | Tên Control | Loại Control | Chức năng |
|-----|-------------|------------------------|---|
| 1. | Cbomahd | ComboBox | Chứa danh sách mã hội đồng |
| 2. | Cbotenhd | Combobox | Chứa danh sách tên hội đồng |
| 3. | Grddetai | Xceed.Grid.GridControl | Hiển thị danh sách các đề tài được hội đồng xét duyệt |
| 4. | grdCGia | Xceed.Grid.GridControl | Hiển thị danh sách các chuyên gia không thuộc hội đồng |
| 5. | grdCG | Xceed.Grid.GridControl | Hiển thị danh sách các chuyên gia thuộc hội đồng |
| 6. | cmdPhai | Button | Thêm chuyên gia vào hội đồng |
| 7. | cmdTrai | Button | Xóa chuyên gia khỏi hội đồng |
| 8. | cmdXetDuyet | Button | Gọi đến frmHD_DT để phân công hội đồng xét duyệt đề tài |
| 9. | cmdThem | Button | Khởi tạo các textbox để nhập mới dữ liệu hội đồng |
| 10. | cmdCN | Button | Lưu thông tin vừa nhập hay vừa sửa chữa |
| 11. | cmdXoa | Button | Xoá dòng dữ liệu được chọn trên Combobox ô |
| 12. | cmdThoat | Button | Đóng form |

III Một số kinh nghiệm về thiết kế giao diện:

- Khi thiết kế giao diện, ta thiết kế các control như label, button, textbox,... sao cho khoảng trống trong form được giảm tối đa; các control phải cân đối về độ rộng, hẹp, font chữ...
- Thường thì trong một chương trình các form sẽ sử dụng chung một pattern về màu sắc, font chữ... (như ví dụ dưới). Để nhấn mạnh ý nghĩa của một số form nào đó, ta có thể sử dụng một kiểu riêng cho nó (thay đổi về màu sắc, cách trang trí...).

VD: form sau đây có cùng kiểu với form dùng trong ví dụ trên.

frmNhapCG

Chuyên Gia

Mã Chuyên Gia: CG009

Họ Tên: Huỳnh Nhân

Phái: ☐ Nam ☒ Nữ

Quốc tịch: VIỆT NAM

Lĩnh Vực trực thuộc: Nghệ thuật

Địa Chỉ: Nguyễn Văn Cừ

Danh sách các chuyên gia:

| Mã | Họ Tên |
|-------|-------------|
| CG001 | Mạn Ngọc |
| CG002 | Phứ Thành |
| CG003 | Nhật Hoa |
| CG004 | Yến Phương |
| CG005 | Vĩnh Phúc |
| CG006 | Tuấn Thành |
| CG007 | Linh Phương |
| CG008 | An Nghi |
| CG009 | Huỳnh Nhân |
| CG010 | ... |

Thêm Cập nhật Xóa Thoát

- Có thể sử dụng màu sắc thay đổi để làm nổi bật nội dung. Như ví dụ sau, màu sắc thay đổi trong từng dòng dữ liệu cho biết đề tài đó đã được xét duyệt hay chưa hay đã hoàn tất.

Phân công Hội đồng xét duyệt Đề tài

Mã Hội đồng: Ngày bắt đầu xét duyệt: Tên Hội Đồng: Số chuyên gia:

| Mã Đề Tài | Tên Đề Tài | Lĩnh Vực |
|-----------|-------------------|-----------------|
| DT007 | Đột biến Tiến... | Công nghệ... |
| DT008 | Cơ sở thực... | Công nghệ... |
| DT010 | Quá trình tiến... | Khoa học vũ trụ |
| DT011 | Tính bất toàn... | Khoa học vũ trụ |
| DT013 | Tính liên tục... | Khoa học vũ trụ |
| DT015 | Du lịch và di... | Văn Hóa |

| Mã đề tài | Tên Đề tài | Ngày Bắt Đầu... | Ngày kết thúc... |
|-----------|------------------|-----------------|------------------|
| DT006 | Khai thác... | 8/1/2004... | |
| DT014 | Kỹ thuật... | 7/1/2001... | 7/1/2001... |
| DT017 | Thừa kế và di... | 7/1/2001... | 7/31/2004... |

>>
<<

Nhập kết quả xét duyệt

[Thoát](#)

IV Thiết kế report:

Tùy theo yêu cầu của từng report mà ta chọn mẫu report nào cho thích hợp. Các mẫu report các bạn có thể tham khảo trong phần sample khi cài chương trình Crystal Report. Crystal Report cung cấp rất nhiều mẫu khác nhau.