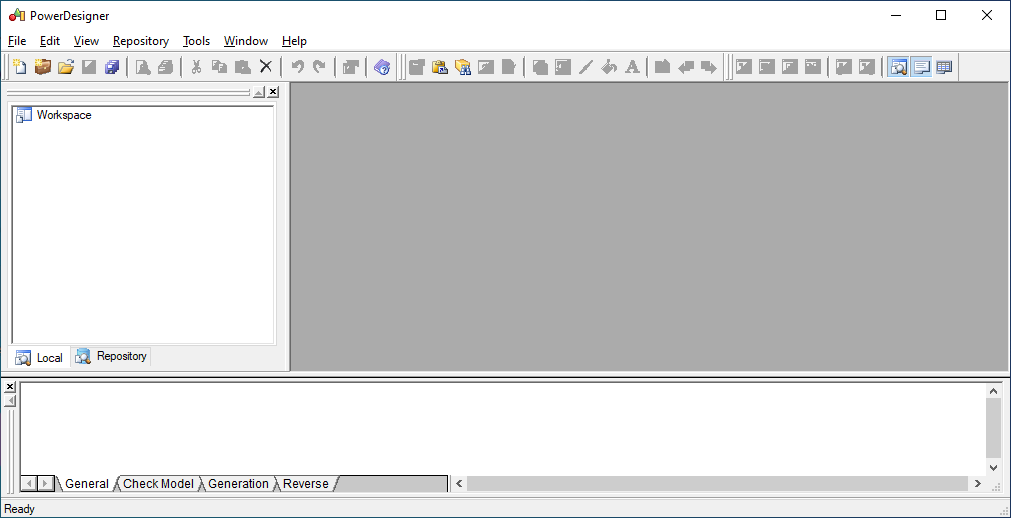
NỘI DUNG HƯỚNG DẪN THỰC HÀNH

MÔN PHÂN TÍCH THIẾT KẾ HỆ THỐNG THÔNG TIN

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

**I.1. Khởi động Power Designer:**

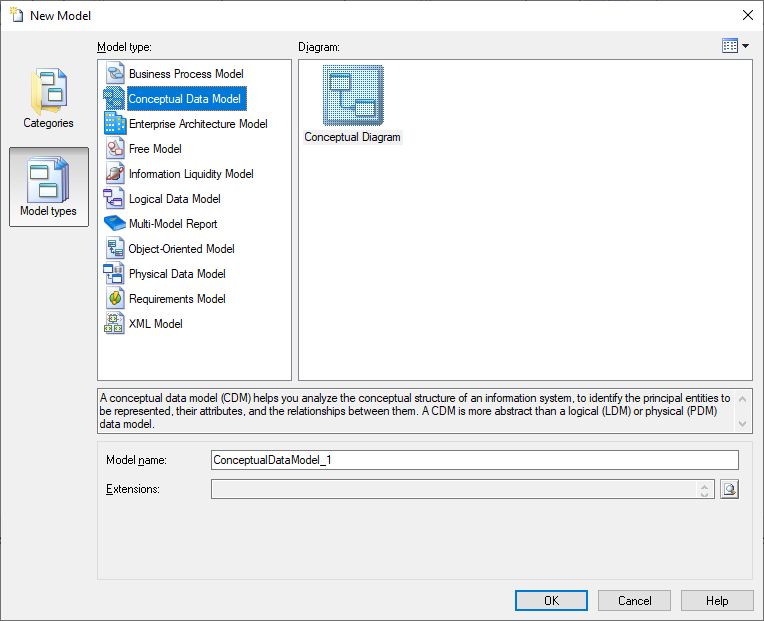
*Start\ All Programes\ SyBase\ PowerDesigner*



* Obiect Browser Window: hiển thị các mô hình và các đối tượng trên mô hình theo cấu trúc cây phân cấp. Cho phép di chuyển nhanh giữa các mô hình. Nút gốc của cây là WordSpace được lưu trên file với phần mở rộng là ( SWS), chứa thông tin môi trường thiết kế hiện hành hỗ trợ cho việc mở lại các mô hình sau này.
* Work Area: Vùng thiết kê mô hình.
* Output Window: hiển thị tiến trình thực hiện một tác vụ trên PowerDesigner, Ví dụ tiến trình tạo mô hình dữ liệu vật lý (PDM) từ mô hình quan niệm dữ liệu (CDM) sẽ được hiển thị trong window này.
* Result List : Hiển thị kết quả của việc tìm kiếm hoặc kiểm tra mô hình.

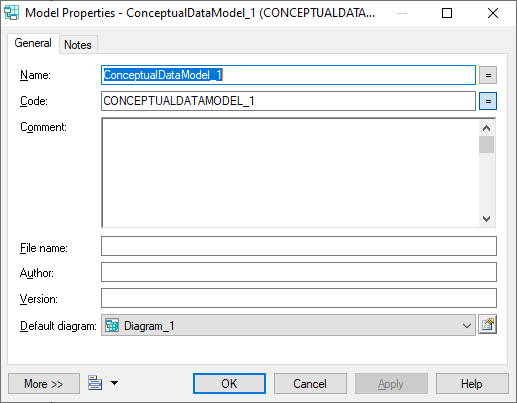
**I.2. Tạo CDM mới**

1. Chọn File 🡪 New Model…
2. Chọn Conceptual Data Model, Sửa tên mô hình và click OK.



**I.3. Khai báo thuộc tính CDM**

1. Chọn Model 🡪 Model Properties (hoặc click phải vào tên mô hình ở khung Workspace)

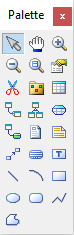


1. Nhập tên (Name), diễn giải(Comment).
2. Click Ok

**I.4. Lưu mô hình CDM**

Chọn File 🡪 Save, Nhập tên file, phần mở rộng mặc định là CDM.

**I.5. Hướng dẫn sử dụng thanh công cụ Palette**



Kế thừa

Mối quan hệ

Thực thể

**I.6. Tạo thực thể (Entity)**

* ***Giả sử ta có 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**

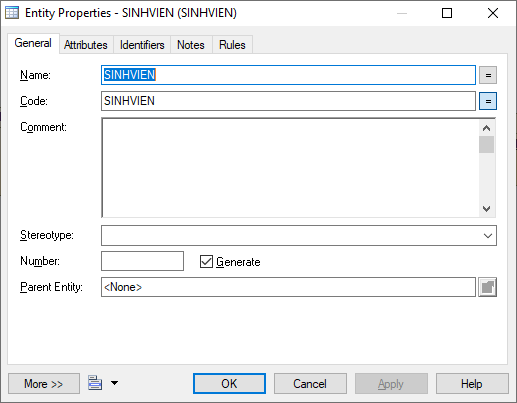
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Thuộc tính | Kiểu dữ liệu | Kích thước | Ghi chú |  |
| MALOP | Character | 10 | Thuộc tính khóa |
| TENLOP | Character | 30 |  |
| SISO | Integer |  |  |

* ***Hướng dẫn cách tạo:***

1. Click chọn biểu tượng Entity , và click vào trong lược đồ. Click phải để kết thúc.
2. Khai báo thông tin của thực thể:

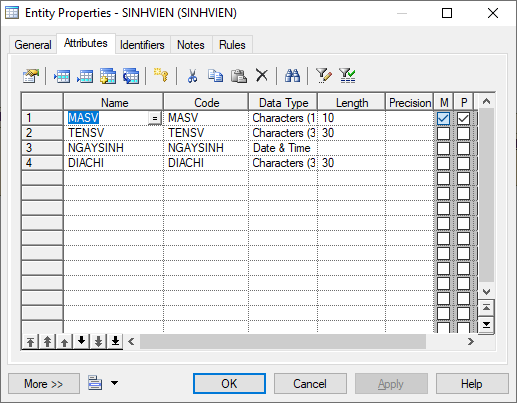
Double-click vào thực thể, một cửa sổ mới mở ra cho phép chúng ta chỉnh sửa thông tin của thực thể như: tên của thực thể, thuộc tính của thực thể, các rule,…

*+ Thẻ General:*



* + Name: Tên thực thể. Tên này hiển thị trên mô hình
  + Code: Tên tắt của thực thể. Tên này được dùng khi chuyển sang CSDL vật lý
  + Comment: Diễn giải về thực thể
  + Number: Số mẫu tin sẽ lưu trữ trong thực thể
  + Generate: Được chọn nếu entity sẽ được chuyển thành table trong PDM

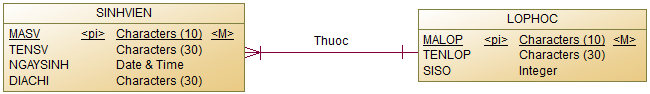
*+ Thẻ Attributes: Khai báo thuộc tính của thực thể*



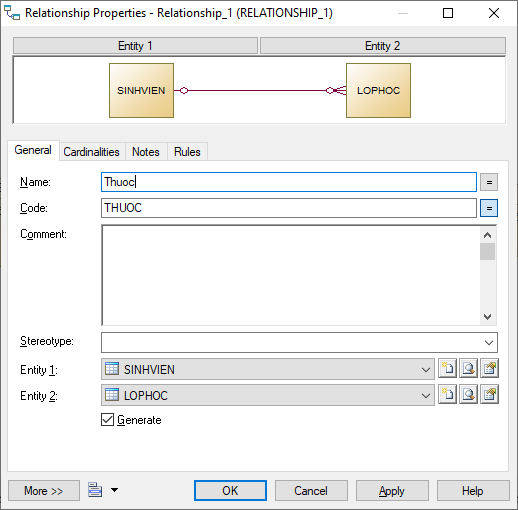
* Name: cho phép nhập tên của thuộc tính của thực thể
* Code: phát sinh một mã tương ứng một thuộc tính (không cần quan tâm chỉ cần click vào dấu “=” bên cạnh)
* Data Type: cho phép chọn kiểu dữ liệu cho thuộc tính, click vào dấu “…” bên cạnh, cửa sổ các kiểu dữ liệu mở ra, ta chọn kiểu dữ liệu phù hợp cho các thuộc tính, bên dưới cho có textbox cho phép ta chọn kích thước chiều dài của từng kiểu dữ liệu.
* M (Mandatory): Not Null hay không?
* P (Primary Indentifier): thuộc tính có phải là khóa chính hay không?
* D (Displayed): có hiển thị thuộc tính này hay không?

**I.7. Tạo mối quan hệ (relationship)**

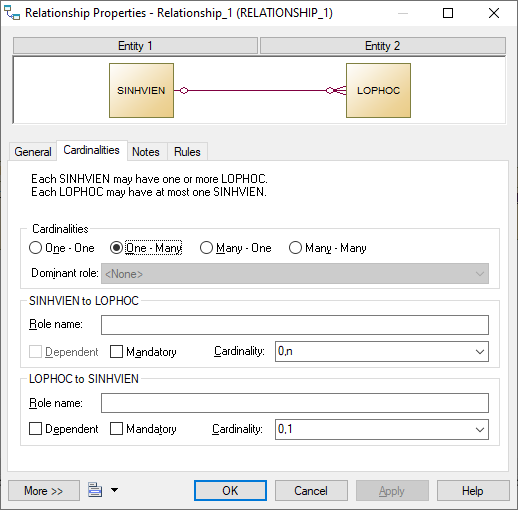
* ***Giả sử thực thể SINHVIEN có mối quan hệ với thực thể LOPHOC như sau:***



* Chọn biểu tượng mối quan hệ  từ thanh công cụ Palette.
* Click vào thực thể SINHVIEN và kéo qua thực thể LOP.
* Để thay đổi tên của mối quan hệ ta nhập tên mới vào textbox name

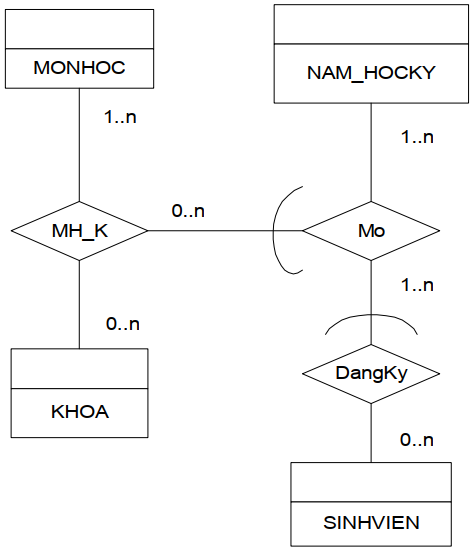


* Tùy theo yêu cầu của mô đề bài mà ta chọn mối quan hệ thích hợp. Ta có các mối quan hệ: 1-1, 1-n, n-1, n-n, và ta chọn trong phần Cardinatily
* Dependent cho phép ta có sử dụng phụ thuộc khóa hay không?
* Mandatory: cho biết có bắt buộc ít nhất hay không?



**I.8. Mối quan hệ 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 |  |  |

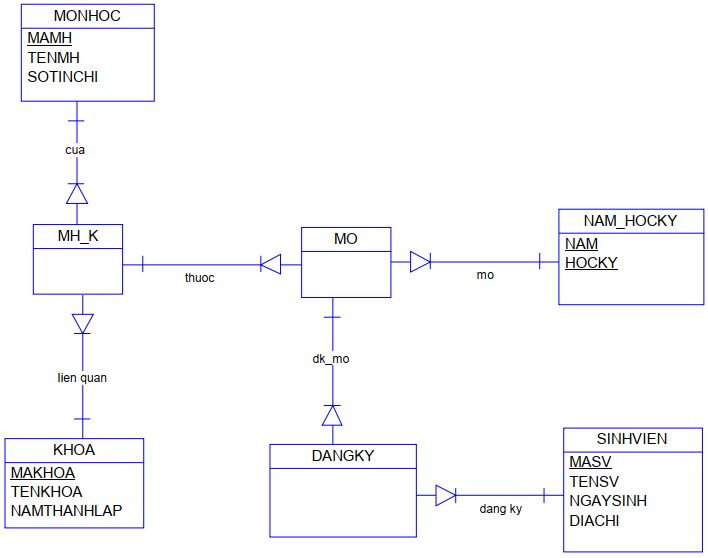
**NAM\_HOCKY**

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

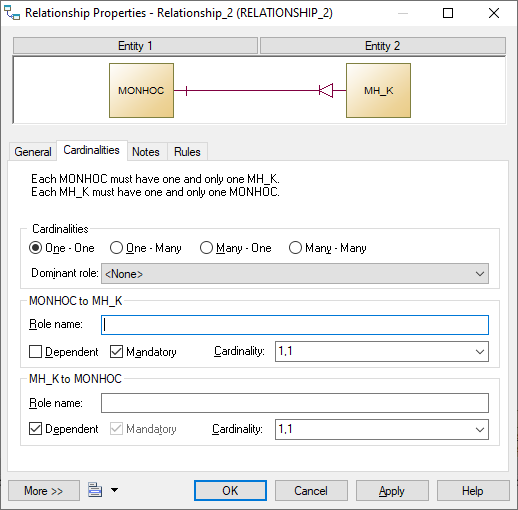
**SINHVIEN**

|  |  |  |  |
| --- | --- | --- | --- |
| 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***
* Đầu tiên các mối kết hợp và kết hợp mở rộng đều được tạo là các thực thể nhưng không chứa thuộc tính nào cả.



* 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 sau.



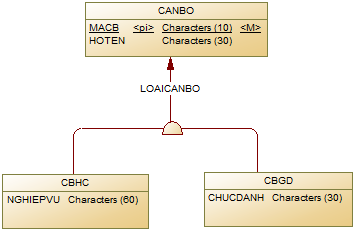
Check

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.

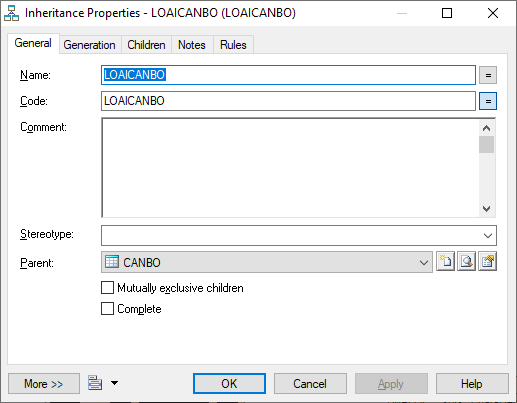
**I.9. Mối quan hệ kế thừa (inheritance)**



1. Click công cụ Inheritance link trong thanh Palette
2. Drag and drop từ thực thể con đến thực thể cha.
3. Nếu muốn khai báo thêm thực thể con thì drag and drop từ ký hiệu hình bán nguyệt tới thực thể con được thêm.

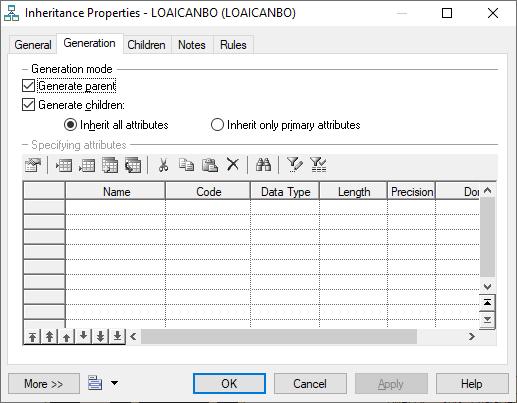
Nếu muốn thay đổi tên và khai báo các đặc tính của nó thì bấm đúp vào hình bán nguyệt, sẽ xuất hiện hộp thoại inheritance properties.

*+ Thẻ General:*



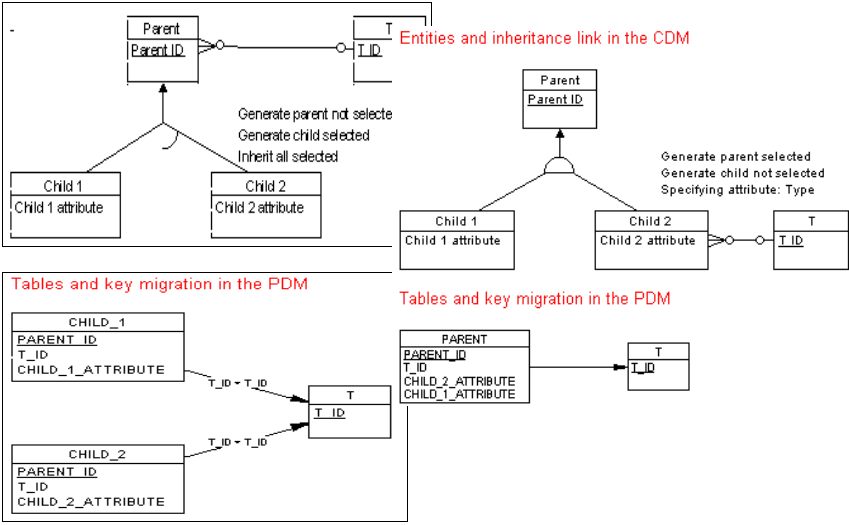
|  |  |
| --- | --- |
| Name | Tên gọi của mối quan kế thừa |
| Code | Mã của mối quan kế thừa |
| Parent | Tên của loại thực thể cha |
| Mutually exclusive children | Chỉ định một thực thể cha chỉ tương ứng với một thực thể con.  *Ví dụ: Loại thực thể Person có 2 loại thực thể con là Male và Female, mỗi thực thể Person hoặc là Male hoặc là Female.*  Chọn lựa này chỉ thể hiện trên hồ sơ thiết kế chứ không thể hiện khi chuyển sang PDM. |
| Complete | Chỉ định mỗi thực thể cha phải thuộc về một thực thể con. |

*+ Thẻ Generation:* Chỉ định cách thức chuyển đổi cấu trúc kế thừa sang PDM



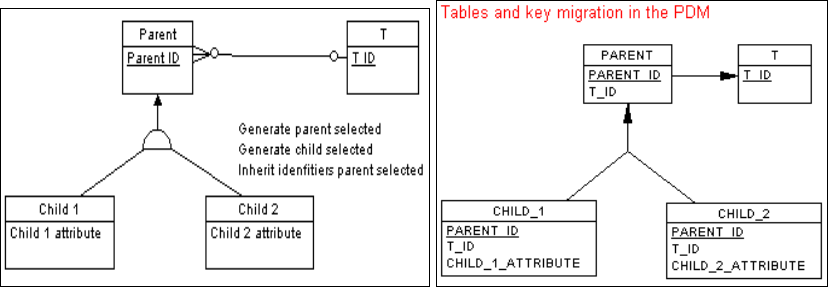
* Trường hợp chỉ chọn Generate Parent mà không chọn Generate Children thì sẽ tạo một Table tương ứng với thực thể cha và chứa thêm các thuộc tính của thực thể con. Các mối quan hệ trên các thực thể con sẽ được thể hiện trên Table đó. Trong trường hợp này bạn có thể khai báo thêm các thuộc tính đặc biệt cho table cha được tạo. Ví dụ: thuộc tính nhận dạng loại nhân viên là nhân viên hành chánh hay công nhân sản xuất.
* Trường hợp chỉ chọn Generate Children mà không chọn Generate Parent. Khi đó bạn cần chỉ định thuộc tính bảng cha ghi trên bảng con:
  + Inherit all attributes: chứa thêm các thuộc tính của thực thể cha
  + Inherit only primary attributes: chỉ chứa thêm những thuộc tính nhận dạng của thực thể cha

Khi chuyển sang PDM, Power Designer sẽ tạo các table tương ứng với các thực thể con. Các mối quan hệ với thực thể cha sẽ thể hiện trên table con.



* Trường hợp bạn chọn cả 2:

Khi chuyển sang PDM, Khóa chính của bảng con được kết hợp (concatenation) bởi thuộc tính nhận dạng của thực thể cha và của thực thể con.



**I.11. Kiểm tra mô hình**

|  |  |
| --- | --- |
| 1. Chọn Tools 🡪Check Model (F4). Xuất hiện hộp Check Conceptual Data Model 2. Chọn hay bỏ chọn những đối tượng cần kiểm tra lỗi và chú ý |  |
| 1. Click OK. Kết quả kiểm tra sẽ hiện trong khung “Output”. Đối tượng và thuộc tính bị lỗi sẽ hiện trong khung Result List. |  |

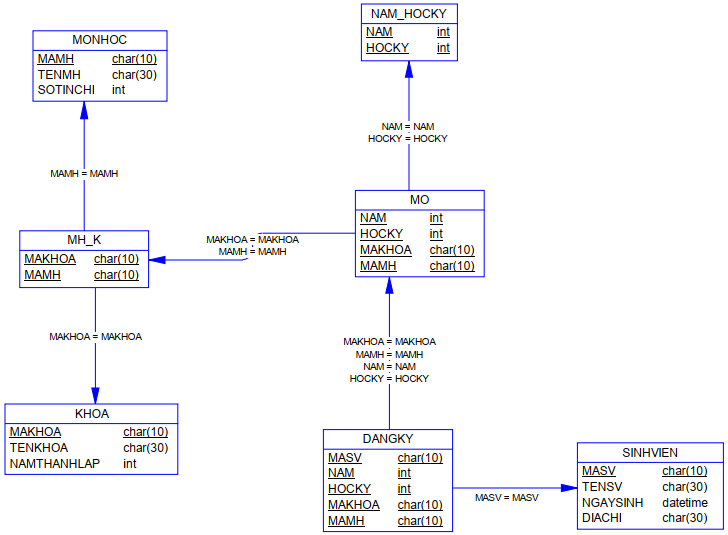
Trường hợp mô hình có lỗi, bạn có thể xem chi tiết thông báo lỗi hoặc chuyển nhanh đến đối tượng bị lỗi bằng cách: click phải vào dòng thông báo trong hộp thoại “Result List” và chọn:

1. Detail để xem chi tiết lỗi
2. Correct mở cửa sổ thuộc tính của đối tượng bị lỗi để chỉnh sửa.
3. Recheck để kiểm tra lại mô hình sau khi hiệu chỉnh

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

|  |  |
| --- | --- |
| 1. Chọn Tools 🡪 Generate Physical Data Model hoặc sử dụng phím tắt Ctrl+Shift+P. 2. Trang General, Chọn Hệ quản trị (DBMS) dùng để lưu trữ database sau này  * Mục Generate new PDM: dùng tạo mô hình PDM mới * Mục Update existing PDM: dùng cập nhật PDM đã tạo. * Nếu chọn mục Preserve modifications thì PDM mới được trộn với PDM cũ, nếu không chọn thì PDM mới sẽ thay thế PDM đang tồn tại |  |
| 1. Trang Detail, chỉ định cách đặt tên Table, các ràng buộc khóa chính, khóa phụ, khóa ngoại… |  |
| 1. Trang Selection, chọn các đối tượng cần chuyển đổi sang PDM. 2. Click OK để thực hiện |  |

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 MÔ HÌNH PDM SANG DATABASE CỦA MICROSOFT SQL SERVER**

**Bước 1:** Tạo CSDL mới trong SQL Server

1. Connect và login vào SQL Server.
2. Tạo CSDL mới chứa các đối tượng được tạo từ mô hình PDM

**Bước 2:** Tạo kết nối từ Power Designer đến CSDL trên SQL Server.

Từ mô hình vật lý:

|  |  |
| --- | --- |
| 1. Chọn Database 🡪 Connect. Một cửa sổ connect xuất hiện như hình bên. |  |
| 1. Tạo kết nối qua một ODBC: Chọn mục ODBC machine data source và click nút Configure… lúc đó một màn hình ODBC Data Source Administrator xuất hiện. |  |
| 1. Nhấn nút Add Data Source, cửa sổ Create Data Source xuất hiện. 2. Chọn kiểu data source |  |
| 1. Chọn trong listbox SQL Native Client |  |
| 1. Nhấn nút Finish, xuất hiện cửa sổ Create a new data source to SQL Server |  |
| - Name: Nhập tên Data Source  - Description không cần phải điền,  - Server: chọn tên server   1. Nhấn nút Next, sang cửa sổ chọn kiểu kết nối đến SQL Server |  |
| 1. Nhấn nút Next, sang màn hình cho phép chỉ định CSDL cần kết nối |  |
| 1. Chỉ định CSDL cần kết nối và nhấn tiếp nút Next |  |
| 1. Để mặc định các giá trị ta tiếp tục. Nhấn Finish, thì cửa sổ ODBC Microsoft SQL Server Setup xuất hiện |  |
| 1. Nhấn vào Test Data Source, một màn hình thông báo kết quả connect có thành công hay không? |  |
| 1. Sau đó ta nhấn nút OK liên tiếp nhau để quay về màn hình Connect ban đầu. |  |
| 1. Click nút Ok |  |
| 1. Ở màn hình connect ban đầu chúng ta chọn lại kết nối ODBC đã tạo là QLSV.  * Nếu kết nối với giấy phép của SQL Server thì trong phần Login ta gõ User ID và Password để login vào SQL. * Sau đó ta nhấn vào nút Connect. |  |

**Bước 3:** Sinh CSDL

|  |  |
| --- | --- |
| 1. Chọn Database 🡪 Generate Database (Ctrl + G), cửa sổ Database Generation xuất hiện, cửa sổ này cho phép chúng ta lựa chọn một số tham số cần thiết trước khi tạo database: 2. Chọn kiểu tạo : Tạo file script chứa các lệnh tạo các đối tượng trong CSDL hoặc tạo trực tiếp CSDL 3. Sau khi lựa chọn tham số xong, ta nhấn vào nút OK. |  |

**IV. MÔ HÌNH DÒNG DỮ LIỆU (DFD – Data Flow Diagram)**

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 biễu 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.

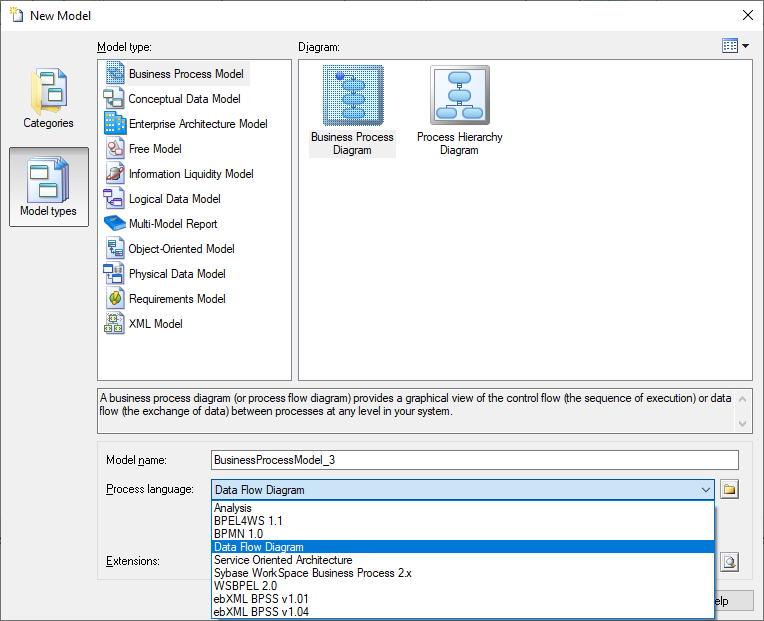
Nhưng ta chỉ quan tâm đến loại mô hình đơn giản nhưng hiệu quả và khá phổ biến cho giai đoạn phân tích xử lý là mô hình dòng dữ liệu(Data Fow Diagram-DFD). Mô hình DFD gồm các khái niệm chính : Xử lý (Process), dòng dữ liệu (Data flow), Kho dữ liệu (Data store) và Đầu cuối (Terminator hay Actor)

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 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 |

**IV.1. Tạo mô hình DFD**

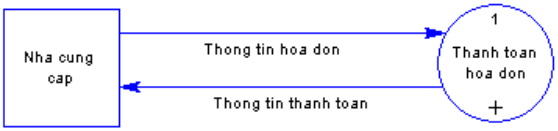
1. Chọn File / New Model…
2. Trong hộp thoại New Model, chọn Business Process Model, chọn Business Process Diagram.
3. Tại Process Language: chọn Data Flow Diagram



**IV.2. Ví dụ 1: Hướng dẫn thiết kê 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:***

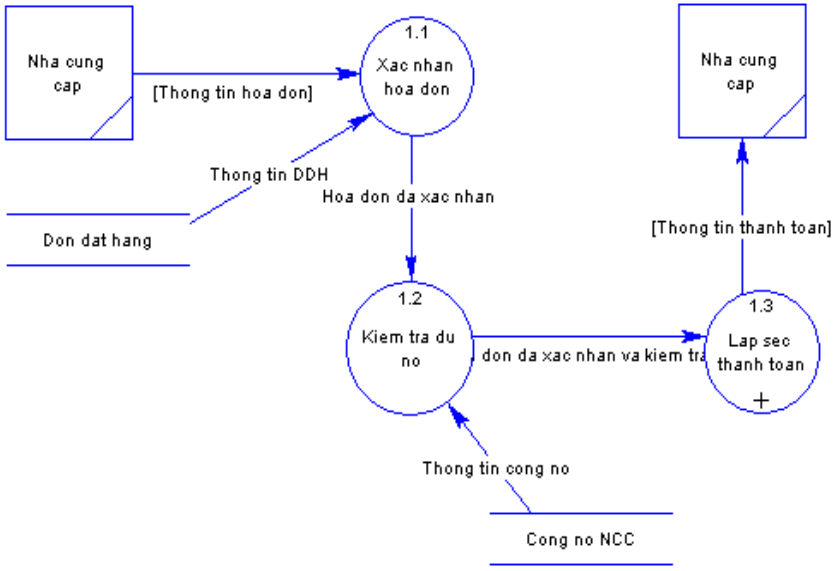


* Hướng dẫn cách tạo:

|  |  |
| --- | --- |
| * 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 ô 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 … |  |

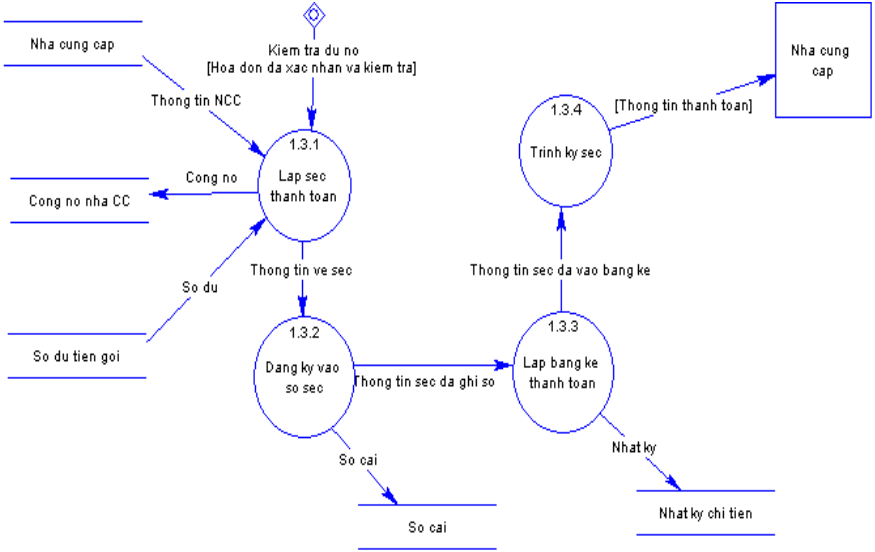
* 1. ***Mô hình xử lý cấp 2:***

Phát triển mô hình xử lý Thanh toán hóa đơn: có nghĩa là tinh 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.
  1. ***Mô hình xử lý cấp 3:***

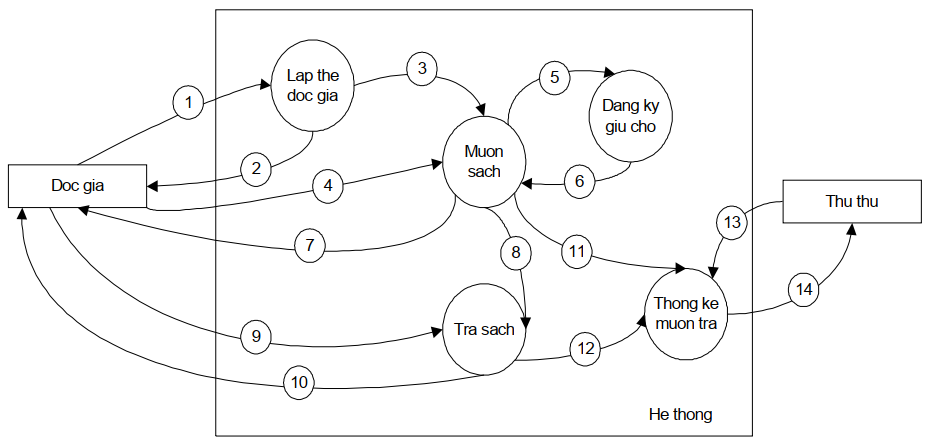
Tiếp tục phát triển mô hình xử lý Lập sec thanh toán: nghĩa là ta phân rã ô xử lý 1.3 - Lập séc thanh toán



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

**IV.3. Ví dụ 2**

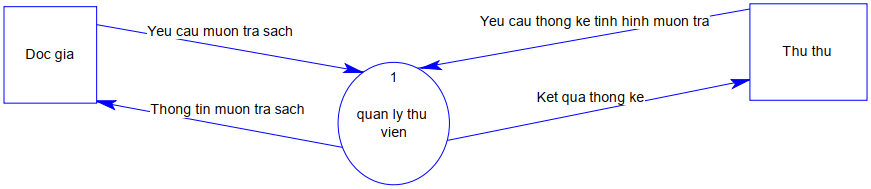
Xác định sơ đồ hệ thống con của qui trình



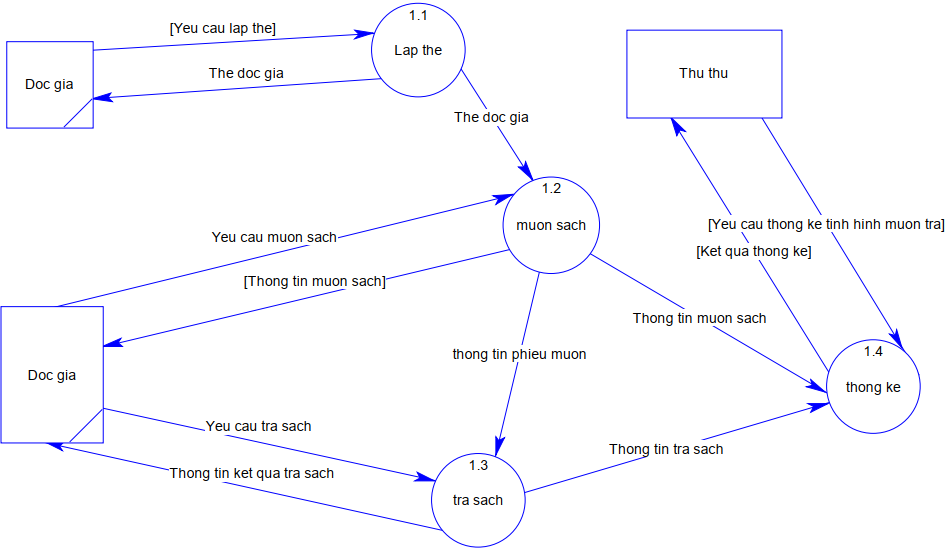
Giải thích các dòng dữ liệu lưu chuyển:



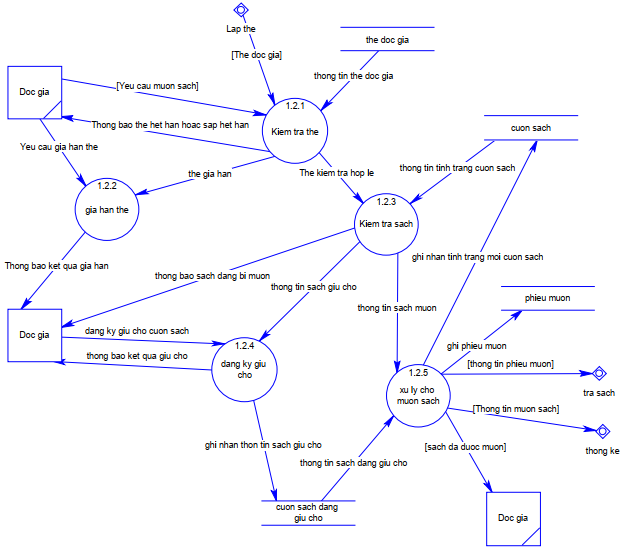
1. ***Mô hình cấp 1***



1. ***Mô hình cấp 2***



1. ***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.

**IV.4. 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
  1. ***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.
  1. ***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 lãnh 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.

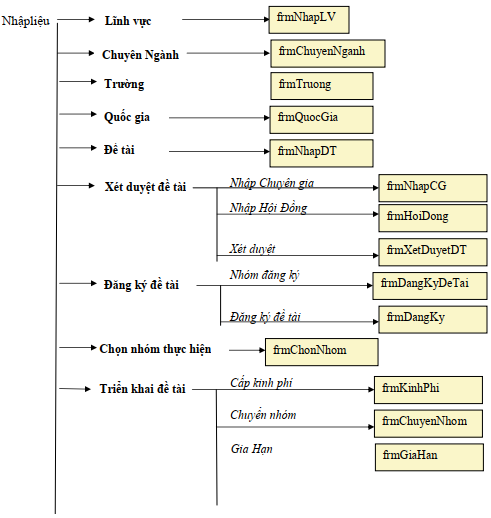
**V. THIẾT KẾ GIAO DIỆN**

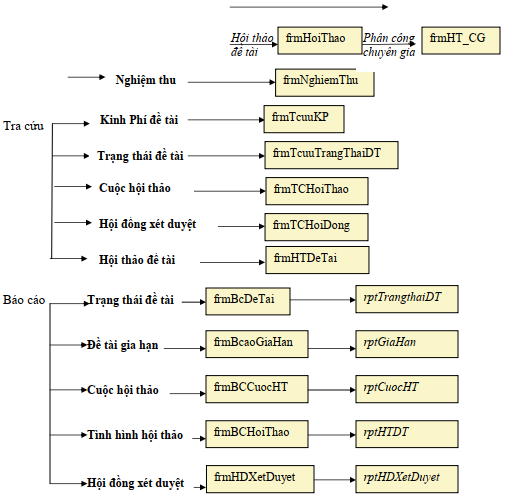
Để thiết kế giao diện, đầu tiên dựa vào các chức năng và dữ liệu mà ta đã phân tích, ta có thể xác định được bao nhiêu giao diện cần phải sử dụng, mục đích của từng giao diện và quan hệ giữa chúng như thế nào.

**V.1. 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.

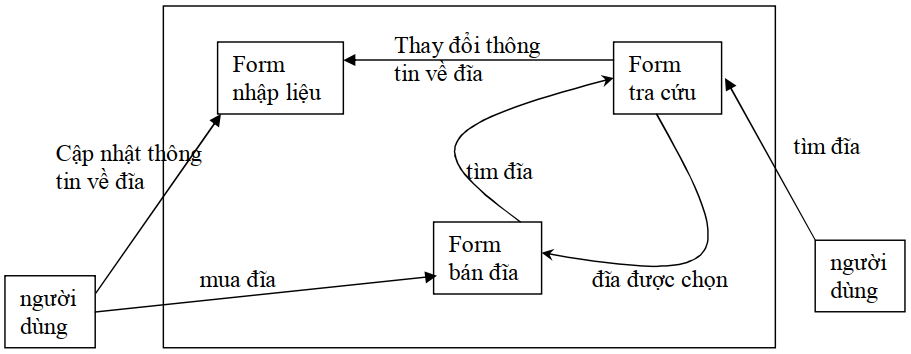
Đối với chương trình có các chức năng riêng biệt nhau, ta thể hiện như sau:





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.

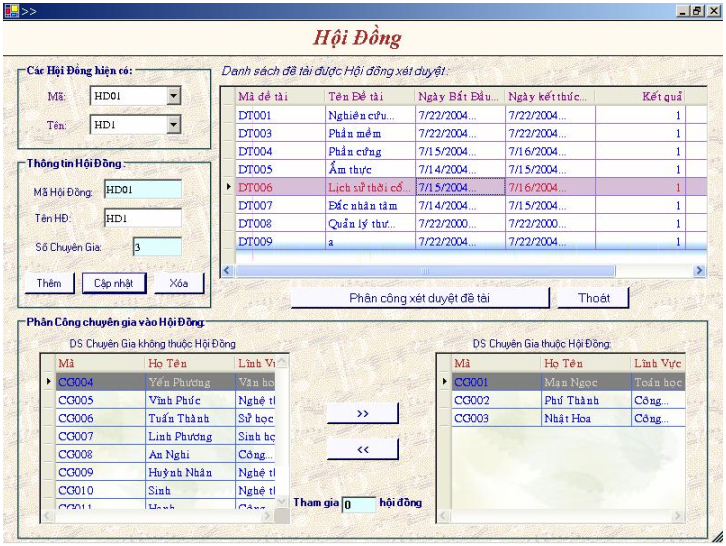
Ví dụ: Đối với chương trình quản lý đĩa CD gồm có 3 form cơ bản sau: form nhập liệu về đĩa CD, form tìm kiếm đĩa và form bán đĩa. Ta có mô hình sau:



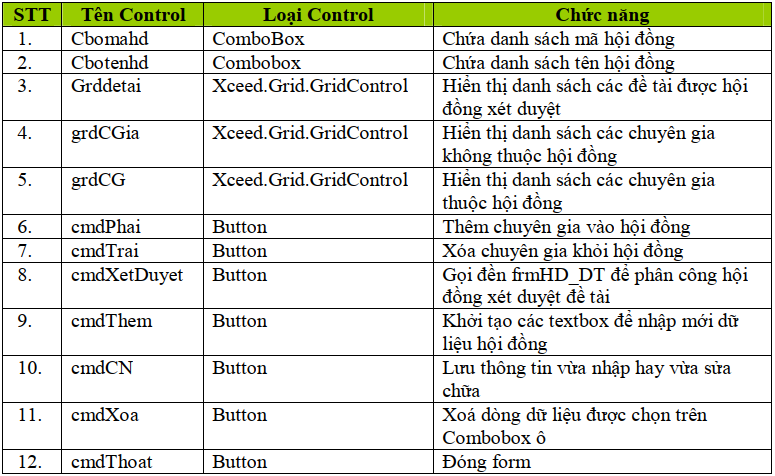
**V.2. Trình bày giao diện**

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.

Ví dụ:



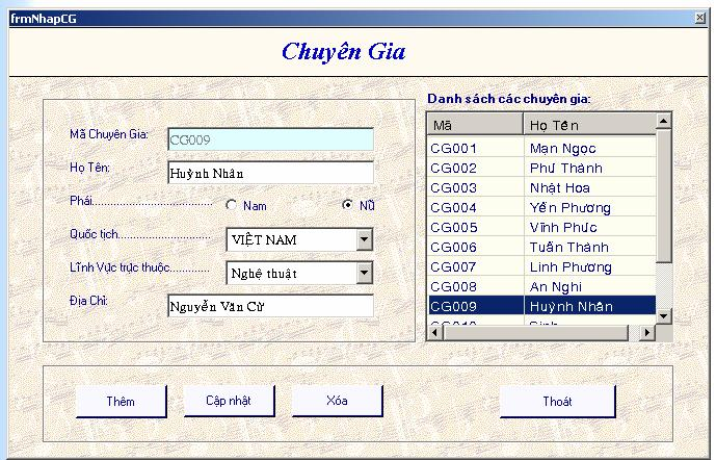
**Sưu liệu**



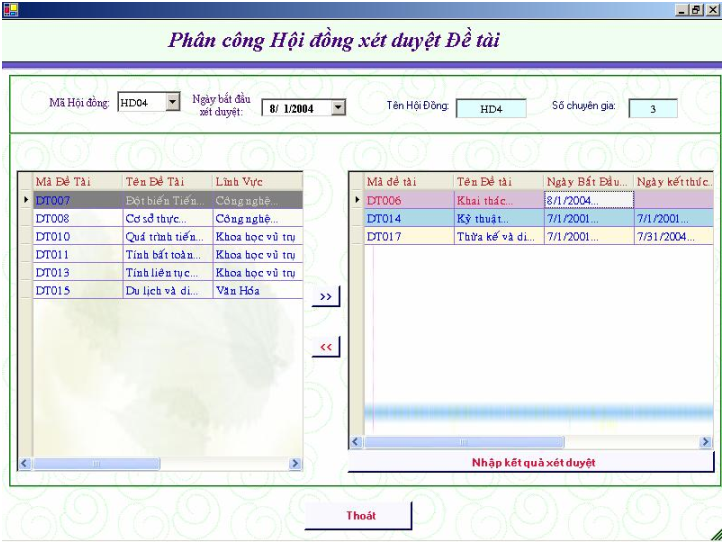
**V.3. Một số kinh nghiệm về thiết kế giao diện**

* + 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.



* + 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.



**V.4. 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.