
Khóa đào tạo Oracle – Developer

Oracle Form Designer

Mục lục

1	Giới thiệu	4
2	Các bước để tạo một Form cơ bản	5
2.1	Khởi tạo Oracle Forms Designer	5
2.2	Tạo, xoá Form	5
2.3	Lưu trữ và thực hiện	5
3	Các thành phần cơ bản của một Form	6
3.1	Windows và Canvas-Views	7
3.1.1	Windows	7
3.1.2	Canvas-View	8
3.2	Bocks	9
3.2.1	Khái niệm	9
3.2.2	Tạo, xoá một block	10
3.2.3	Các thuộc tính cơ bản của block	11
3.3	Items	12
3.3.1	Text item	12
3.3.2	Display item	14
3.3.3	List item	15
3.3.4	Button	15
3.3.5	Check box	16
3.3.6	Radio Group	16
3.3.7	Chart item	16
3.3.8	image	17
3.3.9	OLE container	17
3.4	Relations	18
3.5	Record group, LOVs và Editors	18
3.5.1	Record group	18
3.5.2	Editors	18
3.5.3	LOVs	19
3.6	Triggers	19
3.6.1	Pre - Triggers	20
3.6.2	When - Triggers	20
3.6.3	Post - Triggers	21
3.6.4	Key - Triggers	21
3.6.5	On-Triggers	21
3.7	Messages và Alerts	21
3.7.1	Messages	21
3.7.2	Alerts	22
3.8	Một số thủ tục, hàm, biến hệ thống hay dùng	22
4	Tạo ứng dụng với Form Designer	25
4.1	Tạo Form	25
4.2	Tạo menu	28
4.3	Gọi các sản phẩm khác của Oracle từ Oracle Form	29
5	Các tính năng mới trong Oracle form 6i	30
5.1	Forms Developer	30
5.2	Các tính năng mới	30
5.2.1	Smart Server và hỗ trợ WebDB Listener	30
5.2.2	HTTP và SSL Support	30
5.2.3	OEM Integration	30
5.2.4	CGI-BIN Load Balancing	31
5.3	Xây dựng ứng dụng chạy tại nhiều nơi	32
5.3.1	Lợi thế phát triển	32
5.3.2	Viết một lần, dùng nhiều nơi	32
5.3.3	Cấu hình Form Builder theo Web-based	32
5.3.4	Cấu hình Graphic Builder theo Web-based	32
5.3.5	Middle Tier	32
5.3.6	Forms Server	32
5.4	Xây dựng Form dùng bảng đối tượng (object table)	33

1 Giới thiệu

Oracle forms là một trong những sản phẩm quan trọng trong bộ Developer/2000. Nó cung cấp các phương tiện phát triển giao diện, các xử lý, các thao tác với thông tin được lưu trữ trong cơ sở dữ liệu, tạo khả năng kết nối và trao đổi thông tin với các ứng dụng khác như Oracle reports, Oracle graphics. Ta có thể sử dụng Oracle Forms Designer để thiết kế các ứng dụng, sử dụng Oracle Forms Runform để chạy các ứng dụng đã được bằng Oracle Forms Designer.

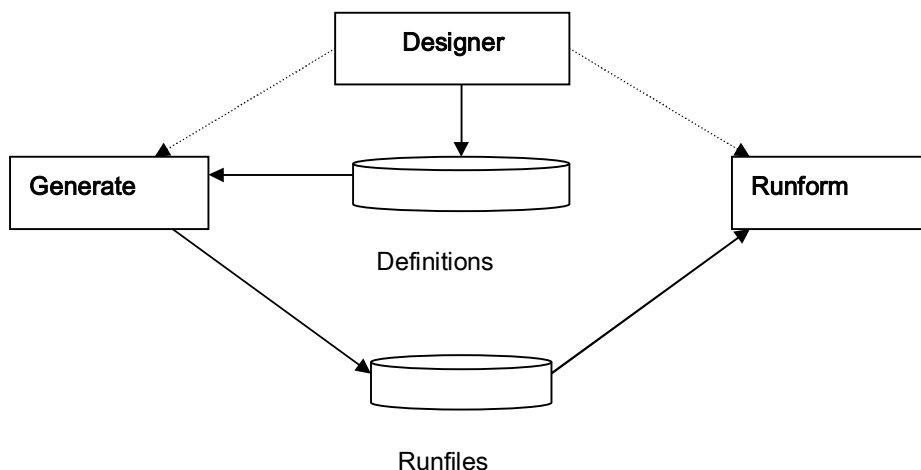
Oracle forms modules

Một ứng dụng Oracle forms gồm nhiều modules (files). Oracle forms modules gồm 3 loại:

- Form: Đại diện cho các đối tượng và dữ liệu mà người sử dụng có thể nhìn thấy hoặc thực hiện các thao tác(*.FMB, *.FMT, *.FMX)
- Menu: Là cấu trúc phân cấp các chức năng mà ta có thể chọn để thực hiện (*.MMB, *.MMT, *.MMX)
- Library: Là tập hợp các đoạn mã chương trình PL/SQL như procedures, functions, packages ...(*.PLL, *.PLD, *.PLX)

Thành phần của Oracle forms

Oracle forms gồm ba thành phần chính là:



- Oracle Forms Runform: Là chương trình thực thi ứng dụng forms đã biên dịch và các module trong nó. Nó chỉ thực thi được các file được sinh bởi Oracle Form Generate.
- Oracle Form Generate: Biên dịch các file tạo bởi Oracle Form Designer thành các file mà Oracle Forms Runform có thể thực thi được.
- Oracle Form Designer: Cho phép ta thiết kế (ví dụ như giao diện, các thư viện ...) và lưu trữ các thông tin như form, menu, library modules.

2 Các bước để tạo một Form cơ bản

2.1 Khởi tạo Oracle Forms Designer

Có 2 cách để gọi Oracle Forms Designer

- Tìm đến biểu tượng Forms Designer và nháy đúp con trỏ trên biểu tượng để vào Forms Designer; hoặc
- Gọi file F45DES32.EXE [username/pasword] trong thư mục *bin* của thư mục đã cài bộ Oracle developer/2000 (ví dụ: C:\ORAWIN95\BIN).

2.2 Tạo, xoá Form

Tạo một Form trong Oracle Forms Designer phải :

- Vào chức năng File/New/Form ;hoặc
- Đặt con trỏ vào biểu tượng forms trên Object Navigator sau đó nhấn vào biểu tượng *Create* (hình dấu [+]).

Muốn xoá form đặt hộp chọn vào tên form cần xoá sau đó nhấn phím *del* hoặc nhấn vào biểu tượng *delete* trên Object Navigator

2.3 Lưu trữ và thực hiện

Muốn lưu trữ chọn chức năng file/save hoặc File/Save As sau đó đưa đường dẫn và tên file cần lưu trữ. File ngầm định sẽ có đuôi *.FMB.

Để chạy form vào chức năng File/Run hoặc nhấn vào biểu tượng *Run* để chạy. Trong trường hợp login vào CSDL, Form Designer sẽ hỏi tên user và mật khẩu (có thể vào chức năng file/connect để login vào CSDL).

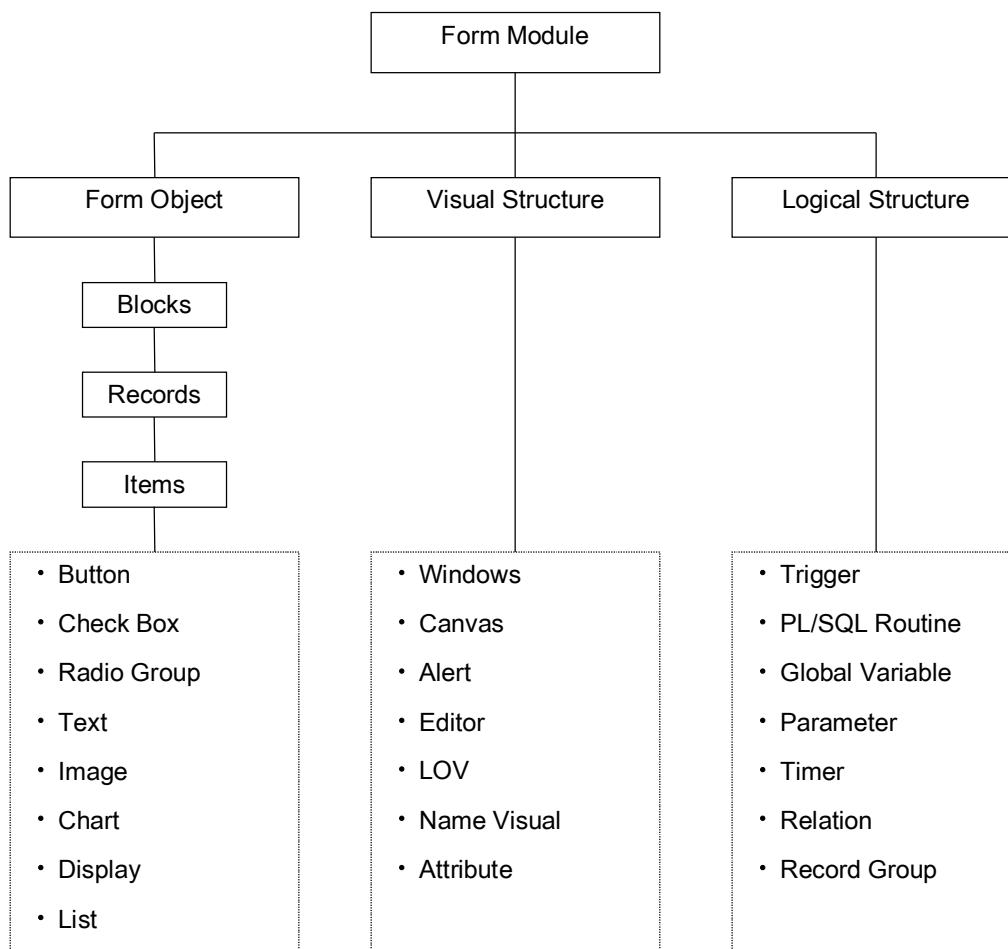
Khi chọn chức năng chạy form, form sẽ tự động được biên dịch và tự sinh ra file chạy. Nếu có lỗi sẽ có thông báo hiện lên.

Để biên dịch form chọn chức năng File/Compile hoặc File/Compile All.

Để sinh File thực hiện vào chức năng File/Administration/Generate. File ngầm định có đuôi là *.FMX và cùng tên với file được tạo bởi Oracle Form Designer.

3 Các thành phần cơ bản của một Form

Thành phần cơ bản của một Form module gồm



Các thuộc tính cơ bản của form

Name: Tên của form

Class: Lớp mà form thừa hưởng các thuộc tính

Display properties

- Coordinate information: Chỉ đơn vị dùng để đặt kích thước các object và vị trí. Có thể là character hoặc real units (centimeters, inches, pixels, or points).
- Title: Là dòng tiêu đề của form
- Current record attribute: Đặt Visual Attribute cho các record hiện thời trong form
- Horiz/vert. MDI toolbar: Chỉ định canvas-view nào sẽ hiển thị dưới dạng horizontal toolbar hoặc vertical toolbar trên window ứng dụng MDI. Canvas-view chỉ định phải có kiểu Horizontal Toolbar.

Functional properties

- Cursor mode: Xác định trạng thái của cursor trong các giao dịch. Có 2 giá trị Open tức cursor vẫn mở trong các giao dịch. Close tức cursor sẽ đóng lại sau mỗi lần commit.

- **Savepoint mode:** Chỉ ra rằng Oracle Forms sẽ đặt savepoints cho ứng dụng đang chạy tại lúc khởi động form hay trước mỗi quá trình post hay commit.
- **First navigation block:** Là tên của block mà Oracle Forms sẽ định hướng tới ngay khi khởi động và sau các thao tác CLEAR_FORM.
- **Console window:** Chỉ định tên của window mà sẽ được hiển thị bởi Oracle Forms. Console bao gồm dòng trạng thái và dòng message được hiển thị dưới đáy của window.

Menu properties

- **Menu module:** Chỉ định tên của menu được sử dụng với form. Khi thuộc tính này được đặt là ngầm định (Default) thì Oracle Forms chạy form với menu ngầm định. Nếu đặt là NULL thì form sẽ không có menu khi chạy.
- **Menu style:** Chỉ định kiểu mà Oracle Forms sẽ sử dụng để chạy các menu chỉ định bởi Menu Module (gồm pull-down, full-screen hoặc bar).
- **Use file:** Cho phép ta chỉ định vị trí của file .MMX khi gán menu tới form module.
- **Starting menu:** Chỉ định tên của một menu trong menu module mà Oracle Forms sẽ sử dụng làm menu chính hoặc top-level.
- **Menu role:** Chỉ ra role cần thiết để Oracle Forms sử dụng được menu. Với thuộc tính này thì có thể cho phép hoặc không cho phép người sử dụng thực hiện các chức năng trên ứng dụng.

3.1 Windows và Canvas-Views

3.1.1 Windows

Khái niệm

Window là một cửa sổ màn hình giống như một khung bức tranh rỗng (chưa có nội dung). Window có các chức năng cho phép phóng to, thu nhỏ, cuộn lên-xuống, di chuyển vị trí.

Một form có thể có nhiều window. Tất cả các form khi tạo mới sẽ tự động tạo một window ngầm định với tên là WINDOW0. Có thể tạo các window bằng cách chèn thêm (insert) từ Object Navigator.

Mỗi một window được tạo hầu như đồng thời với việc tạo một canvas-view. Canvas-view sẽ là nền cho giao diện để đặt các đối tượng (như item, boilerplate text và graphics). Cũng có thể đặt tương ứng canvas-view với window bằng cách đặt thuộc tính trong canvas-view.

Tại thời điểm chạy ứng dụng, window sẽ được hiển thị khi có lời gọi từ chương trình hoặc khi có sự định hướng xuất hiện (Navigation) của một item trên một canvas-view mà được gán tới window. Oracle Forms hiển thị window với nền canvas-view tương ứng

Trong cửa sổ thuộc tính của window ta có thể đặt các thuộc tính của window.

Cách tạo, xoá một window

Để tạo mới một window ta chuyển hộp chọn trên cửa sổ Object Navigator vào đối tượng windows sau đó nhấn vào biểu tượng *Create*. Ta có thể nhấp đúp chuột vào window để gọi cửa sổ thuộc tính để có thể thay đổi tên ngầm định của window hoặc các thuộc tính khác.

Muốn xóa một window đặt con chuột vào biểu tượng window cần xóa sau đó nhấn phím *del* hoặc nhấn vào biểu tượng *delete* trên Object Navigator

Các thuộc tính cơ bản của window

Name: Tên của Window

Class: Thuộc lớp nào ngầm định là <null>

Display properties

- X pos: Tọa độ ngang của window
- Y pos: Tọa độ đứng của window
- Width: Độ rộng của window
- Height: Chiều cao của window
- Bevel: Kiểu khung hiển thị của window
- Title: Tiêu đề hiển thị của window
- Font, Background, Foreground color : Xác định font, màu nền, màu chữ hiển thị

Functional properties

- View: Chỉ định canvas-view được hiển thị khi có lệnh hiển thị window từ trong chương trình.
- Vertical/Horizontal toolbar: Xác định xem toolbar nào sẽ hiển thị trong window.
- Window Style: Xác định kiểu window hiển thị là document hay dialog.
- Modal: Xác định xem window là modal hay modeless.
- Remove on exit: Xác định modeless window sẽ bị đóng khi ra khỏi window
- Biểu tượng name/ title: Biểu tượng/ tiêu đề sẽ hiển thị khi minimized window

GUI hint properties

- Closeable: Cho phép đóng window từ window manager
- Fixed Size: Window có kích thước cố định và không thể thay đổi
- Inherit Menu: Window hiển thị với form menu hiện thời
- Moveable: Cho phép di chuyển window từ window manager
- Zoomable: Cho phép thay đổi kích thước window từ window manager

3.1.2 Canvas-View

Khái niệm

Canvas-View là vùng sẽ được hiển thị lúc chạy ứng dụng. Quan hệ giữa canvas và view (khung nhìn) của nó là một khái niệm cơ bản trong Oracle Forms. View giống như một hình chữ nhật trên canvas mà những gì chứa trong view sẽ được hiển thị trên window khi chạy ứng dụng.

Có thể dùng view của canvas để thay đổi kích thước vùng hiển thị. Khi view cùng kích thước với canvas thì tất cả nội dung trên canvas sẽ được hiển thị. Khi view nhỏ hơn thì chỉ có một phần canvas trong view được hiển thị. Canvas-View luôn hiển thị trên window mà ta đã gán tới. Có 3 loại Canvas-View: là content, stacked, toolbar

Với content hoặc toolbar canvas-view: khung nhìn được xác định bởi window tại đó canvas hiển thị trong nó. Việc thay đổi kích thước window tại thời điểm chạy sẽ ảnh hưởng đến việc hiển thị nội dung trong canvas.

Với stacked canvas-view: kích thước của view có thể được chỉ định tại lúc thiết kế bằng cách giá trị cho các thuộc tính tương ứng. Có thể cho ẩn hay hiện các stacked canvas-view khi thiết kế chương trình. Có thể đặt thuộc tính để hiển thị các thanh cuộn (có chức năng hiển thị các vùng khác trên canvas chưa được hiển thị).

Cách tạo, xóa một canvas-view

Để tạo mới một canvas-view: vào cửa sổ Object Navigator, vào đối tượng canvas-views sau đó nhấn vào biểu tượng *Create*. Ta có thể gọi cửa sổ thuộc tính để thay đổi các thuộc tính của canvas-view.

Xoá một canvas-views: đặt con chuột vào tên canvas-views cần xoá sau đó nhấn phím del hoặc nhấn chuột vào biểu tượng *Delete* để xoá.

Các thuộc tính cơ bản của canvas-view

Name: Tên của canvas-view

Class: Lớp mà canvas-view kế thừa các thuộc tính

Type properties

- Canvas-view Type: Xác định kiểu của canvas-view có thể là Content, Stacked, Vertical Toolbar hoặc Horizontal Toolbar, cách canvas-view được hiển thị trong window.

Display properties

- Width: Độ rộng của Canvas-view
- Height: Chiều cao của Canvas-view
- Bevel: Kiểu khung hiển thị của Canvas-view

Functional properties

- Window: Tên window mà canvas-view sẽ hiển thị
- Raise on entry: Xác định thứ tự hiển thị của Canvas-view trong window.
- X position on canvas: Toạ độ ngang của canvas-view trong window
- Y position on canvas: Toạ độ đứng của canvas-view trong window

Stacked view properties

- View width: Xác định độ rộng của stacked canvas-view
- View height: Xác định chiều cao của stacked canvas-view
- Display X position: Xác định toạ độ ngang của stacked canvas-view
- Display Y position: Xác định toạ độ đứng của stacked canvas-view
- View horizontal/vertical scroll bar: Hiển thị các thanh cuộn ngang hoặc đứng.

3.2 Blocks

3.2.1 Khái niệm

Block là một khối có chứa các Item. Tất cả các Item dù có quan hệ tới các bảng hoặc không đều phải nằm trong các Block.

Base table blocks là khối có quan hệ tới các table (hoặc view) trên cơ sở dữ liệu. Base table blocks có thể chứa các base table item và control item. Base table item sẽ tương ứng với column.

Control block là khối mà không có quan hệ tới các bảng trên cơ sở dữ liệu và nó chỉ gồm các control item.

Tất cả Base table blocks có thể là single-record blocks hoặc multi-record blocks. Một single-record block chỉ hiển thị một bản ghi tại một thời điểm. Một multi-record block hiển thị nhiều bản ghi tại một thời điểm.

Một base table block có thể là master block hoặc detail block. Master block hiển thị các master record. Detail block hiển thị các detail record.

Ta có thể tạo thêm một block bằng cách chèn thêm đối tượng từ Object Navigator và có thể thay đổi các thuộc tính của block bằng cách thay đổi các giá trị tương ứng trong cửa sổ thuộc tính của block.

3.2.2 Tạo, xoá một block

Để tạo một Block, chọn chức năng tools/new block hoặc đặt hộp chọn vào blocks trên Object Navigator và nhấn vào biểu tượng *Create*.

Để xoá một block đặt hộp chọn vào tên block cần xoá sau đó nhấn phím *del* hoặc nhấn vào biểu tượng *delete* trên Object Navigator

General tab

- Base table: Lấy thông tin từ table nào trên cơ sở dữ liệu. Có thể tự đánh vào hoặc nhấn vào button Select ... để chọn. Nếu muốn tạo một Control block thì không cần nhập.
- Block name: Tên của block.
- Canvas: Tên canvas mà block sẽ hiển thị trong nó. Nhấn vào Select .. để chọn những canvas đã có hoặc gõ tên để tạo mới.
- Sequence ID: Vị trí của block trên Navigator.

Items tab

Items tab trong cửa sổ New Block cho phép xác định cách thức thể hiện của các item trong base table block.

- Select column: Nhấn vào đó để chọn các column cần hiển thị. Ngầm định là chọn toàn bộ. Muốn bỏ một column nào đó thì ta nháy đúp vào dấu [+], hoặc bỏ đánh dấu bằng cách nhấn vào *include*. Khi column chuyển sang dấu - tức là nó không được đưa vào block.
- Label: Là nhãn sẽ hiển thị cạnh item.
- Width: Độ rộng của Item hiển thị.
- Type: Kiểu Item sẽ hiển thị.

Layout tab

Layout tab trong cửa sổ New Block cho phép xác định cách bố trí các item trong base table block.

- Style: các item thể hiện trên canvas là Tabular hoặc form. Tabular hình thức các Item định dạng theo chiều ngang trên cùng một dòng khi Orientation được đặt là vertical, hoặc trên cùng một cột khi Orientation được đặt là Horizontal
- Form: Đặt các Item trên hai cột với label đặt bên trái mỗi Item.
- Orientation: Chỉ vị trí các record hiển thị theo hàng ngang hay hàng dọc.
- Records: Chỉ số record sẽ hiển thị trong block.
- Spacing: Chỉ khoảng cách giữa các Item.
- Integrity Constraints: Đặt các ràng buộc dữ liệu cho các item và block dựa trên các định nghĩa của các bảng trong từ điển dữ liệu (data dictionary).
- Button Palette: Oracle Forms sẽ gồm các button ngầm định.
- Scroll Bar: Oracle Forms đưa thanh scroll bar vào base table block.

Master-Detail tab

Master-Detail tab cho phép xác định các quan hệ master-detail giữa base table block và detail block.

- **Master Block:** Chỉ ra tên của Block là master trong quan hệ master-detail. Block này phải đang tồn tại trong form hiện thời.
- **Select:** Hiển thị các block trong form để có thể chọn.
- **Join Condition:** Chỉ điều kiện kết nối giữa detail record và master record.

Để xoá một block đặt hộp chọn vào tên block cần xoá sau đó nhấn phím del hoặc nhấn chuột vào biểu tượng hình dấu X để xoá.

3.2.3 Các thuộc tính cơ bản của block

Các thuộc tính của block gồm nhóm các thuộc tính sau: Display, Records, Navigation, Database, Micellaneous

Display properties

- **Scroll bar:** Có hai giá trị true hoặc false. Nếu chọn true tức là cho phép hiển thị scroll bar thì một loạt các thuộc tính sẽ xuất hiện cho phép điền các thông tin vào (như canvas-view, Scroll Bar Orientation, Scroll Bar X Position, Scroll Bar Y Position, Scroll Bar Width...
- **Current Record attribute:** Đặt Visual Attribute cho các record hiện thời trong block.

Record properties

- **Records displayed:** Số record hiển thị trong block.
- **Record orientation:** Định hướng hiển thị của các item theo hàng ngang hay dọc.

Navigation properties:

- **Navigation style:** Đặt thuộc tính xác định item tiếp theo mà con trỏ sẽ chuyển tới khi ở item đầu tiên hoặc cuối cùng (same record, change record, change block)
- **Next/Previous Navigation block:** Chỉ tên của block tiếp theo hoặc trước block hiện thời.

Database properties:

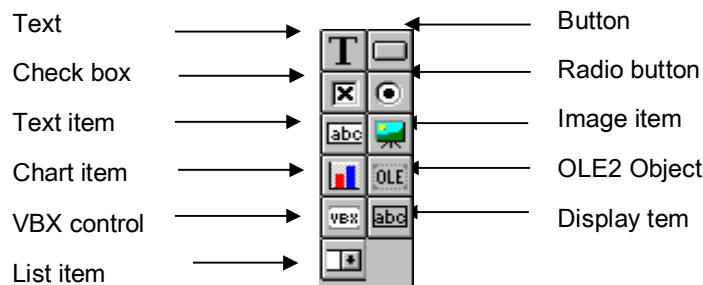
- **Base table:** Là table hoặc view mà block tham chiếu tới.
- **Query only:** Chỉ cho phép thực hiện query dữ liệu.
- **Primary key:** Có giá trị true hoặc false. Thuộc tính chỉ ra Oracle form có kiểm tra tính duy nhất của các primary key hay không trước khi thực hiện thêm số liệu vào table.
- **Delete/Insert/Update/Query allowed:** Có giá trị true hoặc false. Cho phép hay không cho phép Delete, Insert, Update, Query dữ liệu trên table
- **Update Changed Columns:** Có giá trị true hoặc false. Nếu là true chỉ các column thực sự thay đổi mới xuất hiện trong câu lệnh Update.
- **Where clause:** Xác định điều kiện lấy dữ liệu, tương đương mệnh đề where trong câu lệnh SQL
- **Order by clause:** Sắp xếp dữ liệu hiển thị theo trình tự mong muốn.
- **Records Fetched:** Xác định số lượng tối đa các bản ghi sẽ được đưa vào trên mỗi thao tác lấy dữ liệu từ bảng.

3.3 Items

Items là các đối tượng giao diện mà hiển thị thông tin để thực hiện các thao tác và các tương tác với người sử dụng.

Oracle Forms cung cấp các kiểu giao diện item mà ta có thể sử dụng để xây dựng các giao diện ứng dụng. Gồm: button, chart item, check box, display item, image item, list item, radio group, text item, OLE container, VBX control. Mỗi một item trong form đều thuộc vào một block nào đó. Các item trong một block có thể nằm trên các canvas-views khác nhau và có thể hiển thị trên các window khác nhau.

Các item trên thanh toolbar của màn hình layout (Tools/Layout Editor)



Cách tạo, xoá một item

Trong màn hình layout để tạo một item, ta phải nhấn chuột vào biểu tượng của *item* tương ứng, sau đó kéo thả chuột vào màn hình layout. Thuộc tính *canvas* chỉ ra canvas nào item đang hiển thị và *block* chỉ ra item thuộc block nào. Có thể thay đổi các giá trị này.

Cũng có thể vào Object Navigator chọn đối tượng blocks/[tên block]/items muốn đặt item, sau đó nhấn vào biểu tượng *Create*. Nháy đúp con trỏ chuột lên trên item vừa tạo sẽ hiện lên màn hình thuộc tính cho phép thay đổi các thuộc tính của item.

Muốn xoá một text item, đặt hộp chọn vào item cần xoá sau đó nhấn phím del hoặc biểu tượng *Delete* trên cửa sổ Object Navigator để xoá.

Lấy giá trị item: `<biến>:=<tên block >.<tên item>`

Gán giá trị cho item: `:<tên block >.<tên item>:=<Giá trị>`

Một số thuộc tính chung của item

3.3.1 Text item

Khái niệm

Text item Là đối tượng dạng text hiển thị các giá trị dạng string. Text item cho phép thực hiện các thao tác soạn thảo. Là kiểu ngầm định của oracle forms.

Cách tạo, xoá một text item

Giống cách tạo xoá một item ở trên, nhưng kiểu item là Text item

Các thuộc tính cơ bản

Name: Tên của item

Class: Lớp mà item kế thừa các thuộc tính

Type properties

- Item type: Mang giá trị Text item

Display properties

- Canvas: Tên canvas mà item sẽ hiển thị trong nó.
- Displayed: Có giá trị true hoặc false. Cho phép hoặc không cho phép hiển thị trên màn hình.
- X Position: Toạ độ ngang của item
- Y Position: Toạ độ đứng của Item
- Width: Độ rộng của item
- Height: Chiều cao của item
- Space between records: Khoảng cách giữa các record
- Bevel: Kiểu khung thể hiện xung quanh item
- Font name: Tên font dùng để hiển thị ngầm định là lấy font hệ thống
- Font size: Kích thước font
- Font style: Loại font
- Font width: Độ rộng của font
- Font height: Chiều cao của font
- Foreground color: Màu chữ
- Background color: Màu nền
- Fill pattern: Kiểu nền của item
- White on black: Chỉ hiển thị màu đen trắng
- Current Record attribute: Đặt Visual Attribute cho các record hiện thời trong block
- Visual attribute name: Chỉ cách đặt thuộc tính font là ngầm định hay của người sử dụng

Data properties

- Mirror item: Chỉ tên của item mà item hiện thời sẽ lấy giá trị từ nó. Đặt tính chất này cho phép việc đồng bộ giá trị dữ liệu giữa 2 item.
- Data type: Kiểu dữ liệu của item
- Maximum length: Chiều dài tối đa của giá trị mà item nhận.
- Fixed length: Chỉ ra rằng item phải có giá trị với độ dài bằng giá trị Maximum length.
- Required: Item phải được gán giá trị cho nó khi nhập số liệu.
- Format mask: Đặt khuôn dạng hiển thị cho item
- Range low value: Giá trị nhỏ nhất cho phép với dữ liệu kiểu alpha, char, date, datetime, int, number
- Range low value: Giá trị lớn nhất cho phép.
- Default value: Giá trị ngầm định
- Copy value from: Chỉ định giá trị mà Oracle Forms tham chiếu tới. Nó thường được dùng khi ta định nghĩa các quan hệ master-detail, Oracle Forms tự động đặt thuộc tính này trên các foreign key item(s) trong các detail block.

Records properties

- Items displayed: Số item sẽ được hiển thị.

Navigation properties

- Enabled: Cho phép định hướng tới item bằng chuột
- Navigable: cho phép việc định hướng tới item bằng các thao tác ngấm định. Khi đặt giá trị là false thì oracle forms trượt qua item và chuyển đến item tiếp theo.
- Next navigation item: Chỉ định item tiếp theo.
- Previous navigation item: Chỉ định item trước nó.

Database properties

- Base table item: Item có quan hệ tới column trên table
- Primary key: Có giá trị true hoặc false. Thuộc tính xác định item là primary key ứng với column trên table.
- Delete/Insert/Update/Query allowed: Có giá trị true hoặc false. Cho phép hay không cho phép Delete, Insert, Update, Query dữ liệu trên column.
- Case insensitive query: Xác định việc phân biệt các giá trị dạng chữ hoa hoặc chữ thường trong quá trình xử lý query hay không.
- Update only if null: Xác định việc chỉ thực hiện update dữ liệu trên các column mang giá trị null.

Functional Properties

- Case restriction: Xác định việc các giá trị là chữ hoa, chữ thường, hoặc cả hai loại (Upper, lower, mixed)
- Alignment: Xác định kiểu căn lề của nội dung item (ví dụ left, right, center)
- Multi-line: Cho phép text item gồm nhiều dòng
- Keep position: Giữ lại vị trí con trỏ cho lần sau.
- Auto skip: Xác định con trỏ tự nhảy đến item tiếp theo
- Vertical scroll bar: giá trị true hoặc false. Xác định hiển thị của vertical scroll bar

Miscellaneous properties

- Lov: Chỉ tên lov (list of values) gán tới the text item.
- Lov x/y position: Chỉ toạ độ ngang/đứng hiển thị lov
- Editor: Cho phép xem và soạn thảo text item bởi chương trình soạn thảo hệ thống (ví dụ: notepad)
- Editor x/y : chỉ toạ độ ngang/đứng hiển thị editor

3.3.2 Display item

Khái niệm

Display item là đối tượng dùng để hiển thị các giá trị dưới dạng string. Display item không cho phép định hướng con trỏ tới và soạn thảo nội dung trong nó.

Cách tạo, xoá Display item

Giống cách tạo xoá một item ở trên nhưng kiểu item là Display item

Các thuộc tính cơ bản

Xem phần thuộc tính của text item.

Type properties

- Item type: Mang giá trị display item

3.3.3 List item

Khái niệm

List item là một đối tượng dùng để hiển thị một danh sách các giá trị dưới dạng string cho phép chọn một giá trị.

Cách tạo, xoá List item

Giống cách tạo xoá một item ở trên nhưng kiểu item là list item

Các thuộc tính cơ bản

Xem phần thuộc tính của text item

Type properties

- Item type: Mang giá trị List item

Navigation properties

- Mouse navigation: Oracle Forms sẽ cho phép thực hiện định hướng tới item bằng chuột.

Functional Properties

- List elements: Danh sách các phần tử của nhóm các giá trị và danh sách các giá trị tương ứng với nó.
- List style: Kiểu hiển thị của list item (poplist, combo box hoặc T-list).
- Other values: Giá trị mà được gán ứng với các giá trị không có trong danh sách.

3.3.4 Button

Khái niệm

Là một cái hộp có nhãn bên trong hoặc là một biểu tượng mà có thể nhấn vào đó để thực hiện các hành động

Cách tạo, xoá một button

Giống cách tạo xoá một item ở trên nhưng kiểu item là button

Các thuộc tính cơ bản

Xem phần thuộc tính của text item

Type properties

- Item type: Mang giá trị button

Functional Properties

- Label: Là nhãn của button
- Biểu tượng: Chỉ định button dạng biểu tượng hay kiểu thông thường.
- Biểu tượng name: Tên của biểu tượng
- Default button: Xác định button là button ngầm định.

3.3.5 Check box

Khái niệm

Là một nhãn hiển thị với một hộp đánh dấu trạng thái.

Cách tạo

Giống cách tạo xoá một item ở trên nhưng kiểu item là check box

Các thuộc tính cơ bản

Xem phần thuộc tính của text item

Type properties

- Item type: Mang giá trị check box

Functional Properties

- Label: Là nhãn của check box
- Checked value: Giá trị mà hộp chọn sẽ đánh dấu
- Unchecked value: Giá trị mà hộp chọn sẽ không đánh dấu
- Check box other values: Trạng thái của hộp chọn ứng với các giá trị khác với giá trị trong checked, và unchecked.

3.3.6 Radio Group

Khái niệm

Là một nhóm gồm hai hay nhiều radio button trong đó có một radio button được chọn trong danh sách. Thay đổi giá trị của nó bằng cách thay đổi các lựa chọn radio button.

Cách tạo

Giống cách tạo xoá một item ở trên nhưng kiểu item là radio group

Các thuộc tính cơ bản

Xem phần thuộc tính của text item

Trong mỗi radio group sẽ có một hoặc nhiều radio button

Type properties

- Item type: Mang giá trị radio button

Functional Properties

- Label: Là nhãn của radio button
- Value: Giá trị của trả lại cho radio group của radio button.

3.3.7 Chart item

Khái niệm

Là một hình hộp hình chữ nhật có thể hiển thị các biểu đồ tạo bởi Oracle graphics. Muốn hiển thị nội dung của nó ta phải viết các đoạn mã chương trình trong các trigger.

Cách tạo, xoá một chart item

Giống cách tạo xoá một item ở trên nhưng kiểu item là Chart item

Các thuộc tính cơ bản

Xem phần thuộc tính của text item

3.3.8 image

Khái niệm

Là một hình hộp hình chữ nhật với một kích thước nào đó mà có thể hiển thị các ảnh được lưu trong cơ sở dữ liệu hoặc trên file. Các ảnh này chỉ xuất hiện trên màn hình trong lúc chạy chương trình và nhờ các lệnh.

Cách tạo, xoá một image

Giống cách tạo xoá một item ở trên nhưng item ở đây là Image

Các thuộc tính cơ bản

Xem phần thuộc tính của text item

Type properties

- Item type: Mang giá trị image

Functional Properties

- Compression: Xác định ảnh được đọc từ file có được nén khi chuyển sang khuôn dạng của Oracle hay không.
- Quantity: Xác định chất lượng hiển thị hình ảnh (high, medium, low).
- Sizing style: Xác định kiểu hiển thị bức ảnh khi kích thước của nó không phù hợp với kích thước bức ảnh thật.
- Horizontal/Vertical scroll bar: Xác định sự hiển thị của các thanh cuộn đứng hoặc ngang.

3.3.9 OLE container

Khái niệm

Là vùng có chứa và hiển thị một OLE object. OLE object được tạo từ một ứng dụng OLE server. OLE object có thể được nhúng hoặc được link trong OLE container.

Cách tạo

Giống cách tạo xoá một item ở trên nhưng kiểu item là OLE container.

Các thuộc tính cơ bản

Xem phần thuộc tính của text item

Type properties

- Item type: Mang giá trị OLE container.

Functional Properties

- OLE in-place activation: Chỉ ra rằng OLE container có được sử dụng để soạn thảo các đối tượng nhúng và liên kết nó với đối tượng bên ngoài hay không.
- OLE activation style: Chỉ sự kiện để kích hoạt OLE container item
- OLE Resize style: Cách hiển thị một OLE object trong OLE container
- OLE tenant Types: Chỉ ra OLE object nào sẽ sở hữu OLE container
- Show OLE popup menu: Xác định xem có dùng chuột phải để hiển thị popup menu trong OLE object hay không.
- OLE popup menu items: Xác định popup menu được hiển thị trong OLE object

- Show OLE tenant type: Xác định xem có hiển thị đường viền xung quanh OLE container hay không. Kiểu của đường viền phụ thuộc vào kiểu object.
- OLE class: Xác định class của OLE object
- OLE tenant aspect: Chỉ ra kiểu xuất hiện của OLE object trong OLE container

3.4 Relations

Relation là một đối tượng logic chỉ ra mối quan hệ giữa một bên là master block và một bên là detail block. Trong Object Navigator quan hệ này sẽ xuất hiện ở master block.

Khi tạo relation thì Oracle Forms sẽ tự sinh ra các trigger và các mã PL/SQL để đảm bảo quan hệ giữa master và detail blocks. Phần mã này phụ thuộc vào kiểu quan hệ mà ta chọn.

Tạo một Relation

Để tạo một Relation có thể tạo từ khi tạo mới block (Xem phần trên)

Hoặc vào cửa sổ Object Navigator chọn vào Relations của master block sau đó nhấn biểu tượng *Create* để tạo -> hiện lên màn hình New relation

Các thông tin cần điền:

- Relation name: Tên của relation
- Master block: Block cha
- Detail block: Block con
- No-isolated: Thuộc tính ngầm định. Không cho xoá master record nếu detail records vẫn còn tồn tại trong database.
- Isolated: Cho phép xoá master record không chịu ảnh hưởng của detail records
- Cascading: Cho phép xoá master record đồng thời tự động xoá hết các detail record tương ứng. Thao tác này không dùng được cho quan hệ nhiều mức (Lớn hơn 2).
- Join conditon: Điều kiện kết nối giữa master block và detail block

3.5 Record group, LOVs và Editors

3.5.1 Record group

Record group là một object trong form module mà chứa mảng các giá trị gồm các hàng và cột giống như table. Nội dung dữ liệu của nó được lấy nhờ câu lệnh SQL tương ứng.

Để tạo một record group vào cửa sổ Object Navigator chọn vào mục Record groups sau đó nhấn vào biểu tượng *Create* -> hiện lên màn hình new record group. Tại đó ta có thể chọn:

Based on query below: Sau đó gõ câu lệnh SQL tương ứng.

Static values: Sẽ hiện lên các column và danh sách các giá trị của mỗi column.

3.5.2 Editors

Cho phép xem và soạn thảo text item với nhiều chức năng khác nhau. Editors cung cấp chức năng soạn thảo ngầm định và một chức năng soạn thảo bởi chương trình soạn thảo hệ thống (bên ngoài).

Có thể tạo chức năng này bằng các thay đổi thuộc tính trong *Miscellaneous properties* của text item có thể đặt editor bằng null (ngầm định) hoặc SYSTEM_EDITOR

3.5.3 LOVs

Lov là một object trong form module cho phép tạo một danh sách các giá trị lấy từ record group, dùng để hỗ trợ cho các text item hoặc các biến trong bất kỳ một block nào của form module.

Tạo một LOV mới

Trong cửa sổ Object Navigator chọn object LOVs và nhấn biểu tượng Create -> hiện lên màn hình new LOV với các tùy chọn:

Existing record group: Cho phép chọn các record group

V2-Style Table.column values: Cho phép lấy từ các version cũ của SQL * form

New record based on query below: Cho phép đưa lệnh SQL trực tiếp.

Một số thuộc tính cơ bản

Functional properties

- LOV type: Kiểu lov tham chiếu tới các đối tượng record group
- Record group: Chỉ tên của record group mà lov tham chiếu tới
- Column mapping: Chỉ các column trên lov sẽ trả lại giá trị vào các biến nào hoặc các column thuộc block nào.
- Auto confirm: Tự động đóng và trả lại giá trị khi thao tác chọn còn lại một dòng trên lov
- Auto display: Tự động hiển thị khi con trỏ định hướng tới text item.
- Auto refresh: Tự động thực hiện lại lệnh Query
- Long list: hạn chế hiển thị của các giá trị trước khi hiển thị lov
- Auto skip: Sau khi trả lại giá trị thì tự động chuyển đến item tiếp theo

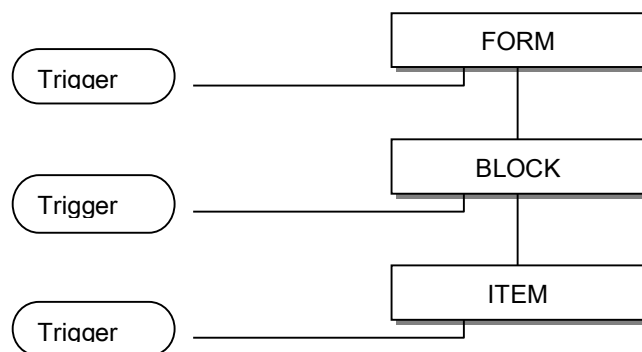
Muốn LOV có hiệu lực trên text item nào thì trong thuộc tính LOV *Miscellaneous properties* và gán thuộc tính *lov* bằng tên của LOV vừa tạo.

3.6 Triggers

Triggers là một khối chứa các mã lệnh nhằm thực hiện một chức năng nào đó trong chương trình ứng dụng. Tất cả các trigger đều có tên và chứa một hoặc nhiều dòng lệnh PL/SQL tương ứng với sự kiện mà nó phải xử lý.

Tên Trigger thường tương ứng với sự kiện (ví dụ When-Button-Pressed trigger sẽ tương ứng với sự kiện nhấn vào Button).

Một trigger phải được gán với một object xác định trên form: item, block hoặc chính trên form. Trigger gồm các mức (level) tương ứng:



Việc xác định mức đặt các trigger là rất quan trọng bởi vì nó xác định phạm vi hoạt động của trigger ứng với các sự kiện (events). Ví dụ Item-level trigger chỉ bật lên (fire) ứng với các sự kiện trên item. Nó không được bật lên ứng với sự kiện như vậy trên item khác cùng hay không cùng thuộc một block. Block-level trigger chỉ bật lên nếu có sự kiện xuất hiện trong block nhưng nó không bật lên khi xuất hiện các sự kiện giống như vậy thuộc block khác.

Mặt khác nếu định nghĩa các trigger ở các mức khác nhau cùng xử lý một sự kiện, mà sự kiện này đều nằm trong phạm vi của các trigger thì trigger ở mức thấp nhất sẽ được thực hiện. Ví dụ ta đặt trigger When-New-Item-Instance tới một block và một trigger When-New-Item-Instance khác tới text item trong block này, khi sự kiện xuất hiện trên text item thì trigger mức item sẽ được bật lên còn trigger mức block bị bỏ qua.

Ở cùng một mức các trigger cũng xuất hiện và xử lý với các cách khác nhau. Có các loại trigger sau: Pre-, Post-, When-, On-, Key-.

Tạo một trigger

Trên cửa sổ Object Navigator ứng với mỗi object (item, block, form) sẽ có mục Triggers. Ta chuyển hộp chọn vào Triggers sau đó nhấn vào biểu tượng Create để tạo mới và chọn sự kiện tương ứng (ví dụ when-button-pressed) sau đó hiện lên cửa sổ viết mã.

3.6.1 Pre - Triggers

Được bật lên trên sự kiện và xuất hiện trước khi thực hiện một hành động (ví dụ insert, update).

Một số Pre - Triggers hay dùng

- Pre-form: Được thực hiện tại thời điểm khởi động form
- Pre-insert: Được thực hiện trước khi thực hiện thao tác insert một row số liệu
- Pre-update: Được thực hiện trước khi thực hiện thao tác update một row số liệu
- Pre-delete: Thực hiện trước khi xoá một row

3.6.2 When - Triggers

Được bật khi thực hiện hành động và được thêm vào cùng với quá trình xử lý ngầm định của Oracle form

Một số When-Trigger hay dùng

- When-button-pressed: Được thực hiện khi có thao tác nhấn button bằng phím hoặc chuột
- When-clear-block: Thực hiện khi xoá dữ liệu từ block hiện thời
- When-create-record: Thực hiện khi tạo một bản ghi mới
- When-checkbox-change: Thực hiện khi check box thay đổi trạng thái
- When-radio-change: Thực hiện khi có thay đổi chọn các radio button
- When-list-change: Thực hiện khi thay đổi chọn giá trị trong danh sách.
- When-new-block-instance: Được thực hiện khi di chuyển input focus từ một block từ block khác
- When-new-item-instance: Được thực hiện khi con trỏ chuyển tới Item
- When-validate-item: Xuất hiện khi có những thay đổi giá trị của item từ người sử dụng hoặc từ các trigger và khi con trỏ di chuyển ra khỏi item.

3.6.3 Post - Triggers

Được bật trên sự kiện xuất hiện sau khi thực hiện hành động.

Một số Post-Trigger hay dùng

- Post-Change: Thực hiện khi dữ liệu của item được thay đổi với giá trị chấp nhận không phải là giá trị null.
- Post-form: Xuất hiện trong khi thoát khỏi form
- Post-query: Thực hiện sau khi dữ liệu được điền vào các record
- Post-record: Xuất hiện khi rời input focus từ bản ghi này tới bản ghi khác.

3.6.4 Key - Triggers

Được bật lên ứng với các hành động thao tác key's standard

Ví dụ Key-up: ứng với việc nhấn phím Page-up

Key-down: ứng với việc nhấn phím Page-down

Key-enter: ứng với nhấn phím enter

Key-next-item: ứng với việc nhấn phím tab

3.6.5 On-Triggers

Được bật khi thực hiện hành động và thay thế quá trình xử lý ngầm định của Oracle form

Các On-Triggers hay dùng

- On-error: Thực hiện khi có một lỗi nào đó xuất hiện
- On-message: Thực hiện khi hiện các message

3.7 Messages và Alerts

3.7.1 Messages

Là dòng thông báo xuất hiện tại dòng trạng thái của Window chỉ ra trạng thái của quá trình xử lý nào đó hoặc khi có lỗi.

Để hiển thị message ta dùng lệnh MESSAGE(message_string, user_response);

message_string: Chuỗi hiển thị cần thông báo

user_response: Chỉ hình thức hiển thị của message.

Thường hai trigger là: on-error và on-message hay liên quan đến quá trình hiển thị và xử lý các message.

Ví dụ: Thông báo lỗi on-error

```
DECLARE
err_code    CONSTANT NUMBER      := ERROR_CODE;
err_type    CONSTANT VARCHAR2(3) := ERROR_TYPE;
BEGIN
  if (err_type = 'FRM' AND err_code = 40508 ) then
    message(to_char(err_code)||': '||'Trùng mã hãy kiểm tra lại!');
  elsif (err_type = 'FRM' AND err_code = 40100 ) then
    message(to_char(err_code)||': '||'Đang ở bản ghi đầu');
  else
    message(err_type||'-'||to_char(err_code)||': '||ERROR_TEXT);
  end if;
END;
```

3.7.2 Alerts

Hiển thị dưới dạng một modal window chứa các thông tin cần thông báo và đợi trả lời từ phía người sử dụng. Tùy theo trả lời mà có thể thực hiện các xử lý tiếp theo.

Tạo một Alerts

Trong cửa sổ Object Navigator chọn mục Alerts sau đó nhấn biểu tượng *Create*, sau đó đặt thuộc tính theo yêu cầu.

Các thuộc tính cơ bản của Alert

Functional properties

- Alert style: Chỉ kiểu hiển thị của alert (stop, caution, note)
- Button 1, Button 2, Button 3: Nhãn cho các button. Phải có ít nhất một button có giá trị
- Default alert button: Button ngầm định
- Message: Nội dung thông báo cần hiển thị

Hiển thị Alert

Để hiển thị alert dùng lệnh Show_alert('<Tên alert>')

Để thay đổi nội dung thông báo của Alert dùng lệnh:

```
set_alert_properties('<tên alert>', properties, message)
```

Ví dụ: Set_Alert_Property('My_Error_Alert', alert_message_text, ' Có lỗi xuất hiện khi ghi');

```
Bt:=Show_Alert( al_id );
```

3.8 Một số thủ tục, hàm, biến hệ thống hay dùng

Một số thủ tục hay dùng

DO_KEY(built-in_subprogram_name)

Thực hiện key trigger tương ứng với các thủ tục có sẵn trong form

Built-in_subprogram_name: Chỉ định tên của built-in subprogram.

Built-in Key Trigger	Associated	Function Key
ABORT_QUERY	Key-EXIT	[Exit/Cancel]
BLOCK_MENU	Key-MENU	[Block Menu]
CLEAR_BLOCK	Key-CLRBLK	[Clear Block]
CLEAR_FORM	Key-CLRFRM	[Clear Form]
CLEAR_RECORD	Key-CLRREC	[Clear Record]
COMMIT_FORM	Key-COMMIT	[Commit]
COUNT_QUERY	Key-CQUERY	[Count Query Hits]
CREATE_RECORD	Key-CREREC	[Insert Record]
DELETE_RECORD	Key-DELREC	[Delete Record]
DOWN	Key-DOWN	[Down]
DUPLICATE_ITEM	Key-DUP-ITEM	[Duplicate Item]
DUPLICATE_RECORD	Key-DUPREC	[Duplicate Record]
EDIT_TEXTITEM	Key-EDIT	[Edit]
ENTER	Key-ENTER	[Enter]
ENTER_QUERY	Key-ENTQRY	[Enter Query]
EXECUTE_QUERY	Key-EXEQRY	[Execute Query]
EXIT_FORM	Key-EXIT	[Exit/Cancel]
HELP	Key-HELP	[Help]

GIÁO TRÌNH ĐÀO TẠO ORACLE FORM

LIST_VALUES	Key-LISTVAL	[List]
LOCK_RECORD	Key-UPDREC	[Lock Record]
NEXT_BLOCK	Key-NXTBLK	[Next Block]
NEXT_ITEM	Key-NEXT-ITEM	[Next Item]
NEXT_KEY	Key-NXTKEY	[Next Primary Key Fld]
NEXT_RECORD	Key-NXTREC	[Next Record]
NEXT_SET	Key-NXTSET	[Next Set of Records]
PREVIOUS_BLOCK	Key-PRVBLK	[Previous Block]
PREVIOUS_ITEM	Key-PREV-ITEM	[Previous Item]
PREVIOUS_RECORD	Key-PRVREC	[Previous Record]
PRINT	Key-PRINT	[Print]
SCROLL_DOWN	Key-SCRDOWN	[Scroll Down]
SCROLL_UP	Key-SCRUP	[Scroll Up]
UP	Key-UP	[Up]

Ví dụ: Muốn thực hiện thêm một bản ghi mới tại block hiện thời

```
BEGIN
  Do_Key('Create_record');
END;
```

Hoặc

```
BEGIN
  Create_record;
END;
```

Xoá dữ liệu trên block

```
BEGIN
  Do_Key('clear_block');
END;
```

EXECUTE_TRIGGER(trigger_name);

Thực hiện các trigger built-in

GO_BLOCK(block_name)

Thực hiện định hướng vào ra tới block chỉ định (navigate). Nếu không có block này thì lỗi sẽ xuất hiện.

- **Block_name:** Tên block

Ví dụ:

```
BEGIN
  Go_Block('Main');
  Execute_Query;
END;
```

GO_ITEM(item_name);

Cho phép định hướng tới một item.

- **Item_name:** Tên của item

Ví dụ:

```
BEGIN
  Go_Item('emp.name');
END;
```

SET_ITEM_PROPERTY(item_name, property, value)

Đặt thuộc tính cho một item

- **item_name:** Tên của item
- **property:** Thuộc tính
- **Value:** Giá trị cần gán

GIÁO TRÌNH ĐÀO TẠO ORACLE FORM

Ví dụ: Đặt nhãn hiển thị cho cancel button là *huỷ bỏ*

Begin

Set_item_properties('emp.Cancel', lable, 'Huỷ bỏ');

End;

SET_VIEW_PROPERTY(VIEW_name, property, value)

Đặt thuộc tính cho view

SET_WINDOW_PROPERTY(window_name, property, value)

Đặt thuộc tính cho window

Hàm GET_ITEM_PROPERTY(item_name, property)

Hàm trả lại thuộc tính của item. Giá trị trả lại có dạng char

Hàm GET_VIEW_PROPERTY(View_name, property)

Hàm trả lại thuộc tính của view. Giá trị trả lại có dạng char

Hàm GET_WINDOW_PROPERTY(Window_name, property)

Hàm trả lại thuộc tính của window. Giá trị trả lại có dạng char

Một số biến hệ thống hay dùng

Oracle Forms cung cấp một số biến hệ thống để ghi lại các trạng thái trong quá trình thực thi ứng dụng. Ta có thể đọc các biến này để lấy thông tin cần thiết.

SYSTEM.BLOCK_STATUS
SYSTEM.CURRENT_BLOCK
SYSTEM.CURRENT_DATETIME
SYSTEM.CURRENT_ITEM
SYSTEM.CURRENT_FORM
SYSTEM.CURRENT_VALUE
SYSTEM.CURSOR_BLOCK
SYSTEM.CURSOR_ITEM
SYSTEM.CURSOR_RECORD
SYSTEM.CURSOR_VALUE
SYSTEM.FORM_STATUS
SYSTEM.LAST_QUERY
SYSTEM.LAST_RECORD
SYSTEM.MODE
SYSTEM.MOUSE_BUTTON_PRESSED
SYSTEM.MOUSE_CANVAS
SYSTEM.MOUSE_FORM
SYSTEM.MOUSE_ITEM
SYSTEM.MOUSE_RECORD
SYSTEM.MOUSE_WINDOW
SYSTEM.MOUSE_X_POS
SYSTEM.MOUSE_Y_POS
SYSTEM.RECORD_STATUS
\$\$DATE\$\$
\$\$DATETIME\$\$
\$\$TIME\$\$

4 Tạo ứng dụng với Form Designer

4.1 Tạo Form

Ví dụ:

Tạo một form như sau:

DEPT

Deptno Dname

Loc

<< < > >> Query Save

EMP

Empno	Ename	Job	Mgr	Hiredate	Sal	Comm
7782	CLARK	MANAGER	7839	09-JUN-81	2450	
7839	KING	PRESIDENT		17-NOV-81	5000	
7934	MILLER	CLERK	7782	23-JAN-82	1300	

1. Gọi form Designer
2. Thực hiện chức năng File/connect: Gõ tên Scott/tiger sau đó nhấn OK
3. Thực hiện chức năng File/New/Form
4. Thực hiện chức năng Tools/New block
5. Tại Base table: Nhấn select -> Nhấn OK -> Chọn bảng DEPT -> Nhấn OK
6. Đổi tên canvas thành DEPT_EMP
7. Chuyển sang Item tab -> Nhấn Select column
8. Chuyển sang Layout tab -> Tại Style chọn Form -> Đánh dấu Button Palette -> Nhấn OK
9. Thực hiện chức năng Tools/New block
10. Tại Base table: Nhấn select -> Nhấn OK -> Chọn bảng EMP
11. Nhấn OK
12. Chuyển sang Item tab -> Nhấn Select column
13. Chuyển sang Layout tab -> Tại Records gán thành 5 -> Đánh dấu Scrollbar

14. Chuyển sang Master/Detail tab -> Nhấn select -> chọn DEPT->Nhấn OK
14. Nhấn OK
15. Vào cửa sổ Object Navigator-> Chọn windows -> Chọn WINDOW0 -> Nháy đúp vào đó để gọi ủa sổ thuộc tính -> Đổi tên (Name) thành WDEPT_EMP -> Tại thuộc tính Title gõ vào "Ví dụ tạo form Master/Detail"
16. Quay sang màn hình layout (Tools/Layout editor)
17. Chuyển tools bar tại block EMP sang bên phải.
18. Thực hiện File/save ghi lại file với tên dept_emp
19. Nhấn vào Biểu tượng Run hoặc File/Run để chạy

Như vậy muốn tạo Form ta phải tạo một form mới, sau đó tạo các đối tượng khác như:

Các block, Canvas-view, window, item.

Muốn các block hiển thị thì phải đặt nó vào trong một canvas-view. Ngầm định khi tạo block, sẽ tự tạo canvas-view nếu canvas-view chưa có. Còn nếu tồn tại có thể chọn trong danh sách

Canvas-view được tạo phải được gán với một window tương ứng.

Ta có thể tạo thêm các item trong mỗi block nhằm thực hiện chức năng nào đó. (Xem phần Item)

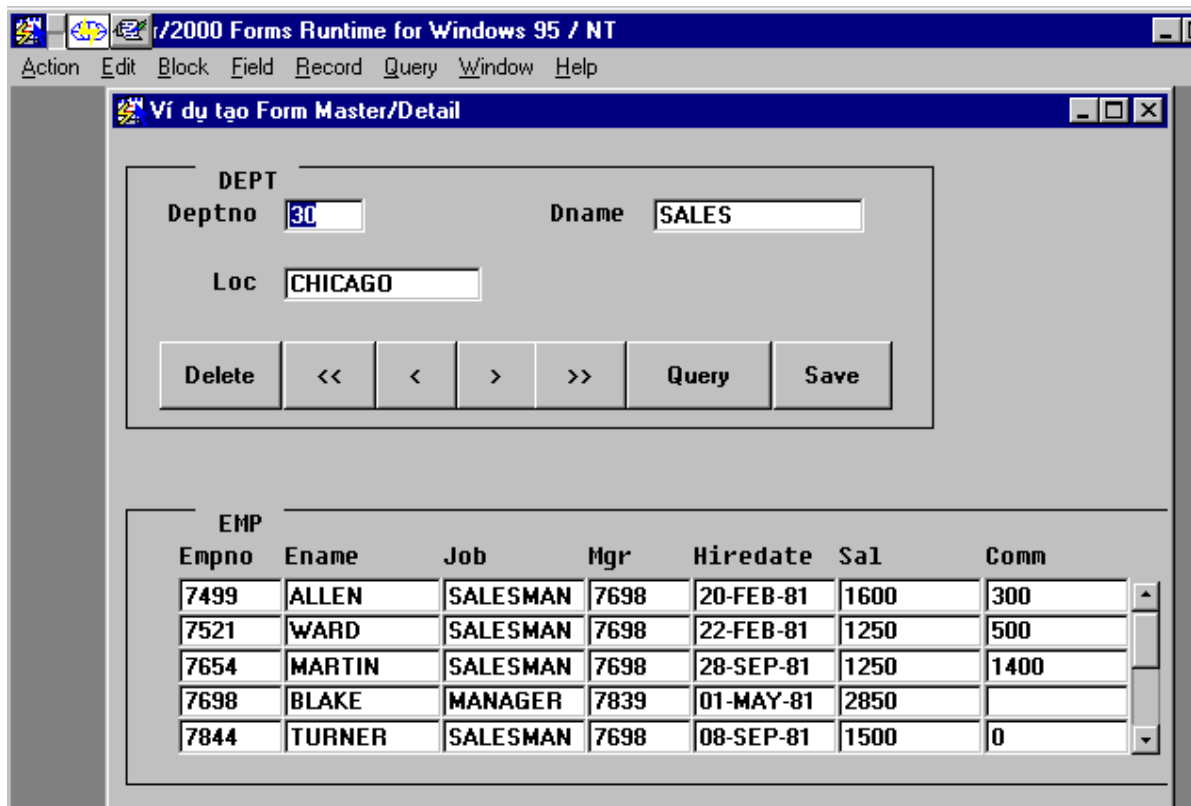
Ví dụ: Trong ví dụ trên, tạo thêm một button Delete cho phép xóa bản ghi tại DEPT (master). Phép xóa được thực hiện ngay cả khi bản ghi emp (detail) có record.

Ta phải thay đổi thuộc tính của quan hệ DEPT_EMP

1. Vào cửa sổ Object Navigator
2. Chọn Blocks/DEPT/Relations/DEPT_EMP
3. Nháy đúp gọi cửa sổ thuộc tính đổi thuộc tính Functional/Master deletes thành Cascading
4. Quay sang cửa sổ Layout editor -> Chọn push button -> đặt vào vị trí kích thước như hình dưới
5. Nháy đúp vào item vừa tạo để gọi cửa sổ thuộc tính -> Gán Name: DELETE, Gán Functional/label: Delete
6. Nhấn trở phải tại item vừa tạo (DELETE) chọn PL/SQL editor -> chọn When-button-pressed ->OK
7. Tại cửa sổ soạn thảo đánh vào dòng lệnh sau:

```
go_block('dept');  
do_key('delete_record');
```

8. Nhấn Close
9. Nhấn biểu tượng Run để chạy ứng dụng



Gọi chạy một form trong ứng dụng với nhiều form

Trong một ứng dụng không chỉ có một form mà còn có nhiều form. Để gọi một form chạy từ menu hoặc từ một trigger của form khác ta dùng các thủ tục Open_form, Call_form.

OPEN_FORM

OPEN_FORM(form_name,activate_mode,session_mode, paramlist);

Thủ tục này không thể được gọi trong Enter query mode. Nó sẽ khởi động một form trong modeless window

- Form_name: Tên form.
- Activate_mode: Có đặt form này activate hay không
- Session_mode: Mạng giá trị NO_SESSION chỉ ra form mới mở sẽ cùng một session với form hiện thời. SESSION Chỉ ra tạo một session mới.
- Paramlist: Danh sách biến được chuyển tới form gọi.

Ví dụ: Gọi form stocks

```
Open_Form('stocks',NO_ACTIVATE);
```

CALL_FORM

CALL_FORM(formmodule_name, display, switch_menu, query_mode,paramlist);

Gọi chạy một form bất kỳ lúc nào. Trong khi gọi vẫn giữ form cha active. Oracle Forms gọi form với cùng các tham chiếu như form cha (Parent form).

- Formmodule_name: Tên form được gọi

GIÁO TRÌNH ĐÀO TẠO ORACLE FORM

- Display: Kiểu hiển thị. HIDE sẽ xoá form gọi trước khi hiển thị form được gọi. NO_HIDE hiển thị form được gọi mà không xoá form gọi.
- switch_menu: Xác định menu của form được gọi. NO_REPLACE có nghĩa form vẫn giữ menu ngầm định của ứng dụng. DO_REPLACE có nghĩa Oracle Forms thay thế menu ngầm định của ứng dụng bằng menu của form được gọi.
- Query_mode: Gồm các giá trị sau. NO_QUERY_ONLY có nghĩa Oracle Forms chạy form trong normal mode cho phép thực hiện các thao tác như inserts, updates và deletes. QUERY_ONLY có nghĩa Oracle Forms chỉ chạy trong Query Only mode với form được gọi.
- Paramlist: Danh sách biến được gửi tới form được gọi.

Ví dụ:

DECLARE

pl_id ParamList;

theFormName VARCHAR2(20) := 'addcust';

BEGIN

pl_id := Get_Parameter_List('tempdata');

IF Id_Null(pl_id) THEN

Call_Form(theFormName);

ELSE

Call_Form(theFormName, HIDE, NO_REPLACE, NO_QUERY_ONLY, pl_id);

END IF;

Call_Form('lookcust',NO_HIDE,DO_REPLACE,QUERY_ONLY);

END;

4.2 Tạo menu

Muốn tạo một menu, chọn chức năng File/New/Menu hoặc vào cửa sổ Object Navigator chọn Menus sau đó nhấn biểu tượng *Create* để tạo mới. Di con trỏ vào menu vừa tạo và nhấp kép tại tên menu để vào màn hình layout editor. Tại đây ta có thể thiết kế cây menu theo yêu cầu của ứng dụng.

Nhấn vào biểu tượng *Create Down* để tạo menu bên dưới với menu hiện thời. Nhấn vào biểu tượng *Create Right* để tạo menu ngang với menu hiện thời. Nhấn vào biểu tượng *Delete* để xoá menu hiện thời.

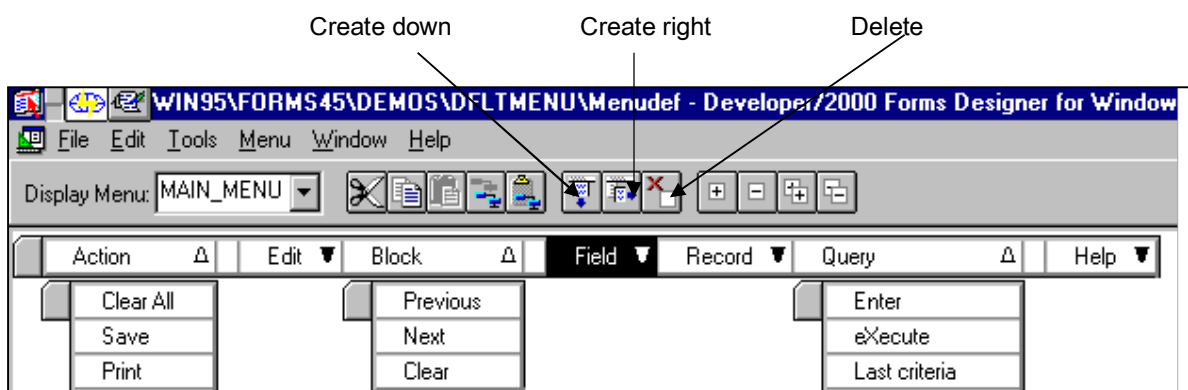
Ta có thể thay đổi *Nhãn hiển thị* theo yêu cầu. Có thể thay đổi các thuộc tính của mỗi phần tử trên menu:

Để thực hiện chức năng nào đó theo yêu cầu ta phải viết các thủ tục tương ứng với yêu cầu đó. Thực hiện bằng cách nhấn trỏ phải tại tên menu sau đó chọn PL/SQL editor để viết các lệnh xử lý.

Để ghi lại ta chọn chức năng File/Save để ghi lại menu (Ngầm định *.mmx). Để menu có thể sử dụng được trong form module ta phải Generate ra file dạng .mmx (File/Administration/Generate). Sau đó gán file này (*.mmx) vào trong phần thuộc tính form. (*xem phần thuộc tính của form*)

Để đặt quyền hạn truy nhập cho từng mục trên menu ta có thể đặt thuộc tính trên Security/Menu item roles để gán các role.

Ví dụ: ta chỉ muốn người có quyền trên role Admin thì mới thực hiện được chức năng này thì trong



4.3 Gọi các sản phẩm khác của Oracle từ Oracle Form

Để chạy một sản phẩm (products) khác từ Oracle Forms ta dùng thủ tục RUN_PRODUCT được xây dựng trong bộ Oracle Forms.

RUN_PRODUCT(product, document, commmode, execmode, location, list, display);

- Product: Chỉ sản phẩm Oracle được gọi: FORMS, REPORTS, GRAPHICS, hoặc BOOK.
- Document: Chỉ định document hoặc module được mở thực hiện.
- Commmode: Chỉ định cách thức sử dụng trao đổi thông tin với Product gồm hai giá trị là Synchronous và Asynchronous.
 - *Synchronous: Xác định điều khiển được trả lại Oracle Forms chỉ sau khi product được gọi kết thúc. Các hoạt động khác của chương trình gọi sẽ không được thực hiện.*
 - *Asynchronous: Xác định điều khiển được trả lại ngay chương trình gọi.*
- Execmode: Chỉ định cách thức sử dụng để gọi chạy product. Gồm các giá trị sau BATCH hoặc RUNTIME. khi gọi form luôn là RUNTIME.
- Location: Chỉ định vị trí của document hoặc module muốn thực thi.
- List Chỉ định tên hoặc danh sách biến được gửi tới product được gọi.
- Display: Chỉ tên Oracle Forms chart item sẽ chứa display sinh bởi Oracle Graphics.

Ví dụ: Gọi Oracle Reports 2.5

```

DECLARED
    pl_id ParamList;
    rep_name char(100);
BEGIN
    pl_id := Get_Parameter_List('tmpdata');
    IF NOT Id_Null(pl_id) THEN
        Destroy_Parameter_List( pl_id );
    END IF;
    pl_id := Create_Parameter_List('tmpdata');
    rep_name := 'C:\report\baocao.rep';
    Add_Parameter(pl_id,'VarX',TEXT_PARAMETER,'Bao cao demo');
    Add_Parameter(pl_id,'PARAMFORM',TEXT_PARAMETER,'NO');
    run_product(REPORTS,rep_name,SYNCHRONOUS,RUNTIME,FILESYSTEM,pl_id, :baocao.daura);
END;
```

5 Các tính năng mới trong Oracle form 6i

5.1 Forms Developer

Forms Developer là một công cụ của ORACLE dùng để phát triển ứng dụng nhanh có khả năng mở rộng cao, hỗ trợ các ứng dụng internet, cho phép:

- Đưa ứng dụng lên Internet.
- Xử lý hiệu quả, mở rộng ứng dụng
- Xây dựng các ứng dụng lớn, nhất quán, có khả năng bảo trì nhanh
- Dùng lại các công cụ của third-party và database servers.

Forms Developer là một tập hợp các công cụ hỗ trợ xây dựng ứng dụng, trong đó công cụ chính là Form Builder. Trong đó còn có các công cụ phát triển khác như:

- Graphics Builder: dùng để xây dựng ứng dụng cho phép người dùng hiển thị số liệu trong CSDL dưới dạng đồ thị, đồ họa.
- Project Builder: dùng để quản lý các thành phần nhau của một ứng dụng.
- Procedure Builder: Dùng để soạn thảo, dịch, kiểm tra, tìm lỗi các đoạn lện PL/SQL
- Schema Builder: dùng để tạo, copy, sửa, xoá các đối tượng CSDL và các mối liên hệ giữa chúng
- Query Builder: dùng để lập các câu hỏi truy vấn CSDL dùng trong ứng dụng.
- Translation Builder : dùng để dịch ứng dụng sang hơn 40 ngôn ngữ khác nhau.

5.2 Các tính năng mới

5.2.1 Smart Server và hỗ trợ WebDB Listener

Cài đặt Smart Server cho phép cài đặt Forms Server với cấu hình thấp nhất so với bản trước. Khi cài đặt Smart Server được tự động cài.

Smart Server còn hỗ trợ lựa chọn cài đặt và cấu hình WebDB Listener như là Web server.

5.2.2 HTTP và SSL Support

Forms Server có thể cấu hình HTTP liên kết giữa máy trạm và máy chủ. Với phương thức này, các Forms messages có thể đi qua firewalls.

Web server cấu hình để chạy chế độ HTTP socket có thể cấu hình dùng các lớp Secure Sockets Layer (SSL) cho nội bộ, mã hoá đường truyền.

Chú ý để sử dụng SSL với Forms, Web server cần phải đặt chế độ SSL-ready, máy trạm cũng cần đặt chế độ SSL-ready.

5.2.3 OEM Integration

Oracle Enterprise Manager (OEM) là các công cụ quản lý hệ thống bằng các công cụ quản lý của Oracle. OEM Forms administrator cung cấp các chức năng cơ sở

Tự động tìm kiếm Form, dịch vụ vfa hiển thị chúng trong cây Navigator

Cho phép kiểm soát các Form, dịch vụ từ các cửa sổ đơn

Cho phép giám sát các Form, dịch vụ. Nếu có lỗi xuất hiện, OEM sẽ báo hiệu cho người quản trị hệ thống hoặc cố gắng tự động chữa lỗi đó.

5.2.4 CGI-BIN Load Balancing

Forms Server chạy chế độ load balancing chỉ khi dùng Oracle Application Server (OAS). Quá trình thực hiện đòi hỏi thêm một số cartridge.

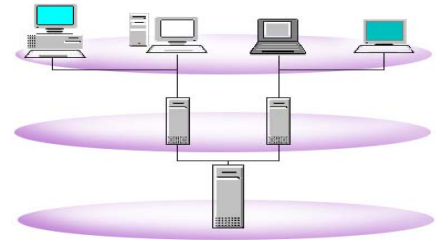
Hiện nay, dùng CGI-BIN, có thể load balancing mà không cần đến OAS và cartridge

5.3 Xây dựng ứng dụng chạy tại nhiều nơi

5.3.1 Lợi thế phát triển

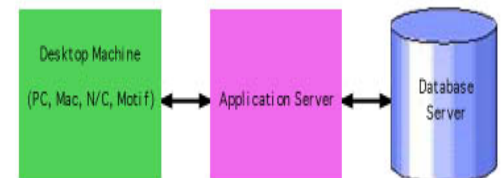
Forms Developer dùng môi trường phát triển 3 lớp three-tier, nghĩa là Web-based, hoặc two-tier (nghĩa là client/server). Điều này cho phép người phát triển lựa chọn

- Sử dụng thông tin những lúc và những nơi cần.
- Cấu hình xử lý lại trong quá trình phát triển hoặc chuyển đổi
- Quản lý thông tin với chi phí phù hợp



5.3.2 Viết một lần, dùng nhiều nơi

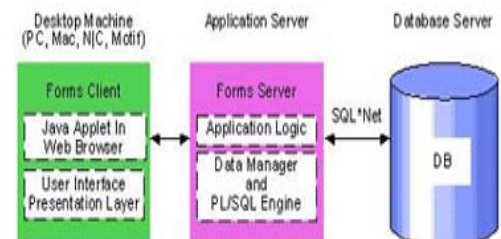
Trong môi trường three-tier, cấu hình Web-based một chương trình trong application server (middle tier) thực thi ứng dụng đã dịch và nối kết với database server. Application server nối kết với người dùng cuối hiển thị màn hình trình duyệt và chấp nhận các giá trị nhập vào của người sử dụng



5.3.3 Cấu hình Form Builder theo Web-based

Tối ưu hoá các Java applet được dùng trong form. Phương thức này là tối thiểu hoá tài nguyên máy trạm, và tối thiểu sự giao lưu trên mạng, cho phép phát triển ứng dụng với chi phí thấp trên môi trường WAN.

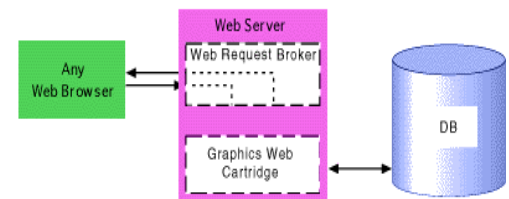
Forms Server tải các Java applet tới máy trạm thực hiện việc trình duyệt với người dùng cuối. Bạn có thể cung cấp các lớp Java để người dùng chỉnh sửa



5.3.4 Cấu hình Graphic Builder theo Web-based

Graphic Builder dùng một kết nối HTTP kết nối tới trình duyệt của người dùng cuối. Khi gọi Graphic Server đưa ra trang HTML tới trình duyệt của người dùng cuối và trang HTML này chứa các yêu cầu về hiển thị.

Graphic Server cho phép nhiều người dùng mở đồng thời.



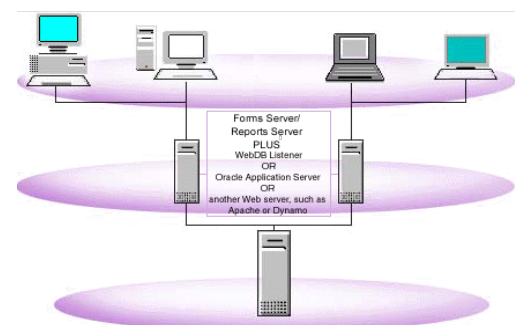
5.3.5 Middle Tier

Lớp giữa bao gồm:

- Oracle WebDB Listener
- Oracle Application Server
- Các Web server khác

5.3.6 Forms Server

Forms Server dùng để phát triển ứng dụng trên internet, cho phép chuyển đổi các ứng dụng client/server trước đây lên môi trường



không cần phải thay đổi mã nguồn. Cho phép ứng dụng chạy trên bất kỳ loại mạng nào.

Khi cập nhật Forms Server ứng dụng tự động chuyển đổi không cần viết thêm mã nguồn.

5.4 Xây dựng Form dùng bảng đối tượng (object table)

Trong Oracle 8i hỗ trợ sử dụng đối tượng tự định nghĩa - user-defined object. Các đối tượng như bảng, cột, và các cột đặc biệt khác được hiểu như REF column. Trong đó:

Object tables: là một đối tượng do người dùng định nghĩa. Trong bảng này ta định nghĩa được các thành phần, thuộc tính của các thành phần dựa trên cơ sở các kiểu dữ liệu Oracle 8 hiểu được. Một đối tượng được định nghĩa có thể được dùng lại trong việc định nghĩa một đối tượng khác, nhưng phần còn lại phải được định nghĩa bằng các kiểu dữ liệu khác của Oracle 8. Vì mỗi cột trong bảng đối tượng như là một mục dữ liệu đặc biệt, có thể coi đó là các cột trong một relational table.

Column objects trong relational tables: Lập trình viên cũng có thể đặt các cột trong bảng

6 Bài tập thực hành

6.1 Bài thực hành 1: làm quen với developer/2000

6.1.1 Nội dung

- Làm quen với Layout editor và Object Navigator trong Form Builder
- Chỉnh sửa các file truy nhập tham chiếu
- Tạo các file thực thi sẵn sàng cho bài tiếp theo dùng Project Builder

6.1.2 Các bước thực hành

1. Gọi Project Builder và chọn `Go to the Project Navigator`.
2. Gọi Form Builder, và chọn `Open an existing form` từ Welcome page.
3. Mở Orders.fmb.
4. Thay đổi preferences vì khi mở và ghi lại, Form Builder tự động đặt danh đánh file có thành chỉ có phần mở rộng là.FMB.
5. Thử mở Customers.fmb. Chú ý loại file bị đổi. Nhấn Cancel.
6. Đóng Orders form.
7. Mở Summit.fmb form.
8. Mở rộng các nhánh của Data Blocks node.
9. Mở rộng các nhánh của Database Objects node. Nếu không thể mở rộng, cần truy nhập vào CSDL trước khi mở rộng các nhánh
10. Đóng Data Blocks node.

6.2 Bài thực hành 2: chạy ứng dụng form

6.2.1 Nội dung

- Chạy ứng dụng form
- Thực hiện các truy vấn không điều kiện để truy vấn các bảng trong CSDL
- Thực hiện các truy vấn có điều kiện và các điều kiện tìm kiếm đặc biệt
- Thực hiện các thao tác insert, update, và delete trong ứng dụng form
- Ghi lại các thay đổi vào trong CSDL

6.2.2 Các bước thực hành

1. Khởi động form runtime gọi form Customers.
2. Chọn Help->Keys từ menu.
3. Thực hiện các lệnh truy vấn không điều kiện
4. Thực hiện các truy vấn có điều kiện tên khách hàng là Womansport.
5. Thực hiện truy vấn có điều kiện Sales Rep ID > 13.
6. Thực hiện các truy vấn có điều kiện sau.
 - Tìm tất cả các thành phố bắt đầu bằng từ San.
 - Tìm tất cả các khách hàng có điều kiện Credit Rating là Excellent.

GIÁO TRÌNH ĐÀO TẠO ORACLE FORM

7. Hiển thị chi tiết khách hàng của Big Johns Sports Emporium và nhấn vào Orders button để chuyển tới Orders form module.

8. Nhấn vào Image Off button để tắt hiển thị ảnh, nhấn vào Image On button để hiển thị ảnh

9. Tìm những đơn đặt hàng có Payment Type là Credit.

10. Chuyển đến bản ghi đầu tiên trong Item block và nhấn vào stock button. Inventory block hiển thị trong một cửa sổ khác. Thực hiện truy vấn để lấy thông tin về stock. Inserting, Updating, và Deleting Bản ghi

11. Nhập thêm bản ghi mới vào Order, theo các chi tiết sau.

Item	Value
Date Shipped	Ngày hiện tại (DD-MON-YYYY)
Payment Type	Cash (Radio group button)
Order Filled	No (Unchecked)

12. Nhập thêm bản ghi vào Item block theo các thông tin chi tiết sau:

Item	Value
Product ID	50530
Quantity	2

13. Ghi lại các bản ghi mới.

14. Thay đổi và cập nhật lại bản ghi order.

15. Xóa một bản ghi trong order. Điều gì sẽ xảy ra?

16. Xóa một dòng trong mục item và ghi lại.

17. Thực hiện xóa một order và ghi lại.

18. Thoát ra khỏi chế độ runtime.

6.3 Bài thực hành 3: Làm việc trong môi trường Form buider

6.3.1 Nội dung

- Không có thực hành cho phần này

6.4 Bài thực hành 4: Tạo form cơ bản

6.4.1 Nội dung

- Tạo một form module có tên CUSTOMERS. Tạo data block dùng Form Builder wizards dựa trên bảng S_CUSTOMER. Dùng Layout editor để chỉnh sửa phần hiển thị
- Tạo master-detail form module có tên ORDERS. Master block dựa trên bảng S_ORD và detail block dựa trên bảng S_ITEM. Tạo block khác dựa trên bảng S_INVENTORY .
- Gọi lại Layout wizard, và chuyển đổi hiển thị của S_ITEM và S_INVENTORY data blocks.
- Ghi và chạy form modules.

6.4.2 Các bước thực hành

1. Tạo form module. Tạo block mới dùng Data Block wizard. Dựa trên bảng S_CUSTOMER, tất cả các cột trong bảng. Dạng hiển thị là form style layout, trong content canvas mới tên là CV_CUSTOMER và hiển thị một bản ghi. Đặt tiêu đề frame là Customers.

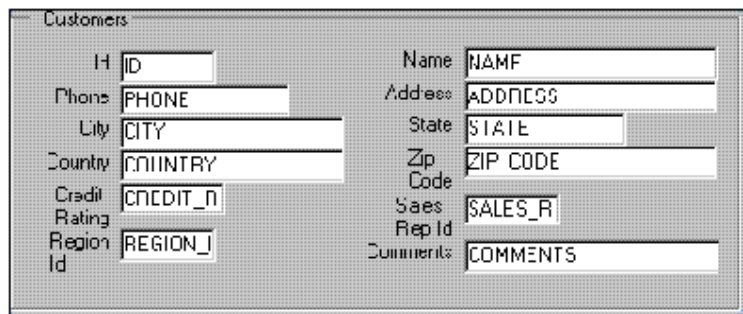
Chú ý: Đổi tên canvas bằng tay trong Object Navigator.

2. Ghi module mới thành file CUSTGxx, xx là số nhóm giáo viên cấp cho bạn. Chạy Form và thực hiện các lệnh query. Chuyển quan từng trường và trở lại Form Builder.

3. Chuyển tên form module trong Object Navigator thành CUSTOMERS.

4. Trong Layout editor, đặt lại vị trí của các items, theo hình dưới đây.

Chú ý: Cần thay đổi kích cỡ của canvas và frame.



5. Tạo form module mới. Tạo block mới dùng Data Block wizard. Dựa trên bảng S_ORD, bao gồm tất cả các cột trừ TOTAL.. Hiển thị S_ORD block trên content canvas mới tên là CV_ORDER, hiển thị 1 bản ghi. Dùng form style layout, đặt tiêu đề của frame thành Orders.

Chú ý: Đổi tên canvas bằng tay trong Object Navigator.

6. Tạo block mới dùng Data Block wizard. Dựa trên bảng S_ITEM và đưa vào tất cả các cột. Tạo mối quan hệ - relationship với master block là S_ORD. Hiển thị toàn bộ các cột ngoại trừ ORD_ID trên CV_ORDER canvas.

Hiển thị 6 bản ghi trong detail block trên cùng canvas với master block. Dùng tabular style layout, đưa vào scrollbar. Đổi vị trí của các block trong Object Navigator, chuyển S_ITEM block sau S_ORD block. Đặt tiêu đề của frame thành Items.

7. Ghi module mới có tên là ORDGxx, xx là số nhóm giáo viên cấp cho bạn.

8. Tạo block mới dựa trên S_INVENTORY, không tạo quan hệ với bất cứ block nào và hiển thị chúng lên một canvas mới. Đưa tất cả các cột trong S_INVENTORY vào trừ

OUT_OF_STOCK_EXPLANATION. Hiển thị 4 bản ghi trên content canvas có tên là CV_INVENTORY. Dùng tabular style layout, bao gồm scrollbar. Trong Object Navigator chuyển S_INVENTORY block sau S_ITEM block. Đặt tiêu đề của frame thành Stock.

9 Trong S_ITEM block chuyển hiển thị của Quantity Shipped item thành Shipped bằng cách dùng lại Layout wizard. Đầu tiên chọn frame trong Layout editor, sau đó gọi Layout wizard.

10. Trong S_INVENTORY data block, chuyển hiển thị của Amount In Stock thành In Stock bằng cách dùng Layout wizard.

11. Chạy form module. Thực hiện truy vấn, Chuyển qua Block và quan sát S_INVENTORY block. Trở lại Form Builder.

12. Chuyển tên của form module trong Object Navigator thành ORDERS và ghi lại.

6.5 Bài thực hành 5: Làm việc với Data Blocks và Frames

6.5.1 Nội dung

- Tạo một control block trong Customers form.
- Dùng Property Palette, chuyển đổi các thuộc tính trong data block S_CUSTOMER. Chuyển đổi thuộc tính của các frame liên quan để giữ bất kỳ thay đổi nào trong Layout editor sẽ được thực hiện bằng tay. Ghi và chạy form sau khi chuyển đổi.
- Tạo một control block trong Orders form.
- Tạo một visual attribute trong Orders form và sử dụng chúng để đánh dấu bản ghi hiện thời trong S_ITEM và S_INVENTORY data blocks tại thời điểm runtime. Chọn cả 2 data blocks để đặt thuộc tính tương ứng trong Property Palette.
- Chuyển đổi các thuộc tính trong S_ITEM và S_INVENTORY data blocks. Chuyển đổi thuộc tính của các frame liên quan để giữ bất kỳ thay đổi nào trong Layout editor sẽ được thực hiện bằng tay. Ghi và chạy form sau khi chuyển đổi.

6.5.2 Các bước thực hành

1. Tạo một control block trong CUSTGxx form. Tạo một new block manually, và đổi tên block này thành Control. Đặt thuộc tính Database Data Block thành, Query Allowed, Insert Allowed, Update Allowed, và Delete Allowed. Đặt thuộc tính None với Query Data Source Type property. Đặt các thuộc tính khác thành default. Chuyển dịch CONTROL block sau S_CUSTOMER block.

2. Đảm bảo rằng các bản ghi được ghi trong S_CUSTOMER block được cất trong customer ID.

3. Đặt thuộc tính của frame cho S_CUSTOMER block như sau:

Xoá tiêu đề của frame, và đặt thuộc tính Update Layout property thành Manually.

4. Ghi lại và chạy CUSTGxx form.

5. Kiểm tra hiệu ứng của các thuộc tính vừa đặt lại.

Chú ý: Lỗi hiển thị trong cửa sổ cảnh báo rằng Control block không có mục nào. Sẽ đưa thêm các mục vào Control block ở các bài sau

5. Tạo một control block trong ORDGxx form. Tạo một block mới bằng tay có tên là Control. Đặt thuộc tính Database Data Block, Query Allowed, Insert Allowed, Update Allowed, và Delete Allowed database properties thành No. Đặt thuộc tính Query Data Source Type thành None. Đặt các thuộc tính khác thành default. Đặt vị trí của CONTROL block sau S_INVENTORY block trong Object Navigator.

Chú ý: Có thể copy control block từ CUSTGxx form.

6. Tạo một visual attribute tên là Current_Record. Đặt thuộc tính Foreground Color thành White và Background Color thành DarkCyan. Chọn cả 2 block S_ITEM và S_INVENTORY đặt thuộc tính để đánh dấu hiển thị của bản ghi hiện thời.

7. Chuyển số bản ghi hiển thị trong S_ITEM block thành 4 và thay đổi kích cỡ của scrollbar cho phù hợp.

8. Đảm bảo các bản ghi được cất trong S_ITEM block theo ITEM_ID.

9. Đặt thuộc tính tự động chuyển tiếp của con trỏ thành next record, Khi người dùng nhập lệnh [Next Item] trong S_ITEM block.

10. Đặt thuộc tính của tất cả các frame thành: bỏ tiêu đề của frame, đặt thuộc tính Update Layout thành Manually.

11 Ghi lại, dịch và chạy ORDGxx form. Kiểm tra hiệu ứng của các thuộc tính vừa đặt.

6.6 Bài thực hành 6: Làm việc với text item

6.6.1 Nội dung

- Xóa mục region ID trong CUSTOMERS form.
- Sử dụng Property Palette, chuyển đổi thuộc tính của một vài text item trong S_CUSTOMER block. Ghi và chạy form sau khi chuyển đổi xong.
- Trong ORDERS form, tạo mới text items để giữ customer name và sales rep name trong S_ORD block. Đổi các thuộc tính của text item trong S_ORD, S_ITEM, và S_INVENTORY blocks. Ghi và chạy form sau khi chuyển đổi.

6.6.2 Các bước thực hành

1. Trong CUSTGxx form, bỏ Region ID item.
2. Chuyển hiển thị của Comments item thành multi-line text.
3. Tự động sinh mã số khách hàng, mã số này là duy nhất đối với bản ghi và không bị thay đổi. Dùng S_CUSTOMER_ID sequence.
4. Trong CUSTGxx form, thay đổi vị trí và độ rộng của các trong Object Navigator theo bảng sau:

Item	Độ rộng - Width
ID	60
NAME	195
ADDRESS	195
CITY	195
STATE	130
COUNTRY	195
ZIP_CODE	85
PHONE	160
CREDIT_RATING	65
SALES_REP_ID	65
COMMENTS	236

5. Ghi lại và chạy và kiểm tra sự thay đổi.

The screenshot shows an Oracle Form titled "Customer Information". It features the following fields and labels:

- ID**: A text input field.
- Name**: A text input field.
- Address**: A text input field.
- City**: A text input field.
- State**: A text input field.
- Country**: A text input field.
- Zip Code**: A text input field.
- Phone**: A text input field.
- Credit Rating**: A text input field.
- Sales Rep Id**: A text input field.
- Comments**: A large text area for notes.

6. Trong S_ORD block, tạo một text item có tên Customer_Name. Customer_Name không liên quan tới bảng S_ORD. Không insert, update, hoặc query trên item này và chỉ đi đến item bằng chuột. Đặt thuộc tính Prompt text thành Customer Name. Hiển thị item này trên CV_ORDER canvas.
7. Trong S_ORD block, tạo một text item có tên Sales_Rep_Name. Sales_Rep_Name không liên quan tới bảng S_ORD. Không insert, update, hoặc query trên item này và chỉ đi đến item bằng chuột. Đặt thuộc tính Prompt text thành Sales Rep Name. Hiển thị item này trên CV_ORDER canvas.
8. Đặt thuộc tính cho Date_Ordered hiển thị ngày hiện tại, bất cứ khi nào bản ghi mới được nhập.
9. Trong S_ITEM block, tạo text item mới có tên Item_Total. Item_Total không liên quan tới bảng S_ITEM. Không insert, update, hoặc query trên item này và chỉ đi đến item bằng chuột. Đặt thuộc tính Prompt text thành Item Total. Hiển thị item này trên CV_ORDER canvas. Khuôn hiển thị thành 999G990D99.
10. Chỉnh sửa các mục Price, Quantity và Quantity_Shipped căn bên phải.
11. Sau mục Quantity_Shipped item, các mục khác chỉ đi đến được bằng chuột và không cho phép updates.
12. Trong ORDGxx form, thay đổi kích cỡ và vị trí của các item theo bảng sau:

S_ORD Block Items	Độ rộng - Width
ID	40
DATE_ORDERED	66
CUSTOMER_ID	66
CUSTOMER_NAME	116
SALES_REP_ID	66

SALES_REP_NAME	116
DATE_SHIPPED	66
PAYMENT_TYPE	48
ORDER_FILLED	18

6.7 Bài thực hành 7: Tạo LOVs và Editors

6.7.1 Nội dung

- Tạo một LOV trong Orders form hiển thị số lượng sản phẩm và mô tả sản phẩm. Gán LOV đó vào Product_ID item trong S_ITEM data block. Ghi lại và chạy form.
- Tạo một LOV trong Customers form hiển thị số sales rep và tên của họ. Gán LOV vào Sales_Rep_ID item trong S_CUSTOMER data block. Ghi lại và chạy form.
- Tạo một editor trong Customers form, và gán nó vào Comments item. Ghi lại và chạy form.

6.7.2 Các bước thực hành

1. Trong ORDGxx form, Tạo một LOV Hiển thị số sản phẩm và mô tả sản phẩm được dùng cho trường Product_id item trong S_ITEM block. Dùng bảng S_PRODUCT, Cột Id, và Name. Đặt thuộc tính tên của LOV là Products_LOV, Title là Products, X Position là 30, Y Position là 30, Width là 200, và Height là 250. Với mỗi ID column đặt giá trị trả về thành S_ITEM.PRODUCT_ID.
2. Gán Products_LOV vào trong Product_ID item trong S_ITEM block.
3. Ghi lại và chạy để kiểm tra.
4. Trong CUSTGxx form, tạo một LOV hiển thị mã số của sales representatives và tên. Chọn first_name và last_name columns, nối chúng với nhau đặt alias thành Name. Đặt thuộc tính Name thành Sales_Rep_LOV, Title thành Sales Representatives, X Position thành 30, Y Position thành 30, Width thành 200, và Height thành 250. Đặt giá trị trả lại thành S_CUSTOMER.Sales_Rep_ID.
5. Gán Sales_Rep_Lov thành Sales_Rep_ID item trong S_CUSTOMER block.
6. Trong CUSTGx form, tạo một editor và gán chúng vào Comments item. Đặt title thành Comments, background color thành gray, và foreground color thành yellow.
7. Ghi lại và chạy để kiểm tra. Thay đổi kích cỡ của window nếu cần

6.8 Bài thực hành 8: Tạo Input Items bổ sung

6.8.1 Nội dung

- Trong Customers form, chuyển Credit_Rating item thành list item. Ghi lại và chạy form.
- Trong Orders form, chuyển Order_Filled item thành check box item.
- Trong Orders form, chuyển Payment_Type item thành radio group.
- Thêm 2 radio buttons vào radio group. Ghi lại và chạy form.

6.8.2 Các bước thực hành

1. Trong CUSTGxx form, chuyển Credit_Rating text item thành pop-up list item. Đưa thêm vào các thành phần Poor, Good, và Excellent vào dữ liệu hiển thị. Hiển thị tất cả các giá trị khác thành Poor. Các giá trị khởi tạo thành GOOD. Thay đổi kích cỡ và vị trí trong Layout Editor.
2. Ghi lại và chạy để kiểm tra sự thay đổi.

GIÁO TRÌNH ĐÀO TẠO ORACLE FORM

3. Trong ORDGxx form, chuyển đổi Order_Filled text item thành Check box. Đặt giá trị checked thành Y và giá trị unchecked state thành N. Đảm bảo các bản ghi mới chỉ nhận các giá trị Y hoặc N. Đặt nhãn thành Order Filled. Thay đổi kích cỡ của Check box, và đặt căn phải.
- 4 Chuyển đổi Payment_Type text item thành radio group. Đưa thêm các radio buttons cho Cash-tiền mặt và Credit-thẻ tín dụng để thể hiện các giá trị CASH và CREDIT.
5. Đặt ký tự S cho cash và T cho credit. Đặt nhãn cho các radio button thành Cash cho Cash radio button và Credit cho Credit radio button. Giá trị mặc định là Cash.
6. Đặt lại vị trí các items trong S_ORD block trong Object Navigator theo vị trí hiển thị trong layout.
7. Ghi lại và chạy để kiểm tra các thay đổi

6.9 Bài thực hành 9: Tạo NonInput Items

6.9.1 Nội dung

- Trong orders form, Tạo 2 display items trong S_ITEM block.
- Tạo một image item trong S_ITEM block.
- Trong orders form, tạo một iconic button trong control block.
- Trong orders form, Trên cơ sở Item_Total trong S_ITEM block tạo một control item trong chính block đó. Trên cơ sở item này tính tổng của đơn đặt hàng.
- Trong customers form, Tạo một iconic button Trong control block.
- Ghi và chạy orders và customers forms.

6.9.2 Các bước thực hành

1. Trong S_ITEM block của ORDGxx form, tạo một display item đặt tên là Description. Đặt thuộc tính Prompt là Description và hiển thị lên phía trên item.
2. Tạo một single-record image item đặt tên là Product_Image trong S_ITEM block của ORDGxx form.
3. Tạo một display item khác là Image_Description, trong S_ITEM block. Item này phải đồng bộ với Description item. Đặt thuộc tính Maximum Length cũng có giá trị tương tự như trong Description item.
- 4 Trong Control block của ORDGxx form, tạo một iconic button đặt tên là Product_LOV_Button. Sử dụng file *list.ico* (không kèm theo phần mở rộng .ico). Đặt cả hai thuộc tính Keyboard Navigable và Mouse Navigate là No.
- 5 Để hiển thị mục thông tin tổng cộng (item total information), đặt các thuộc tính sau cho Item_Total trong S_ITEM block:

format mask	là	999G990D99.
Justification	là	right.
Calculation Mode	là	Formula.
Formula	bằng	:S_ITEM.quantity_shipped * :S_ITEM.price.
Keyboard Navigable	là	No
Mouse Navigate	là	No.

- 6 Để hiển thị toàn bộ item totals tạo mới một non-database item trong S_ITEM block. Đặt các thuộc tính position, size và prompt properties theo như sau:

format mask là 9G999G990D99.

Justification property là right.

GIÁO TRÌNH ĐÀO TẠO ORACLE FORM

Number of Items Displayed bằng 1.

Tạo một summary item S_ITEM.total hiển thị tổng giá trị các item_total trong S_ITEM block. Đặt thuộc tính Query All Records của S_ITEM block là Yes. Đặt thuộc tính Keyboard Navigable và Mouse Navigate là No.

Item Id	Product Id	Price	Qty	Shipped	Item Total
ITE	PRODUC	PRICE	QTY	SHIPPED	ITEM TOTAL
ITE	PRODUC	PRICE	QTY	SHIPPED	ITEM TOTAL
ITE	PRODUC	PRICE	QTY	SHIPPED	ITEM TOTAL
ITE	PRODUC	PRICE	QTY	SHIPPED	ITEM TOTAL

Order Total: TOTAL

7. Cấu giữ, biên dịch và chạy forms để kiểm nghiệm các thay đổi. Thay đổi kích thước cửa sổ nếu cần.

Order ID: ID

Order Information

Date Ordered: DATE_ORDERED Customer ID: CUST_ID Customer Name: CUST_NAME

Sales Rep ID: SALES_REP_ID Sales Rep Name: SALES_REP_NAME

Date Shipped: DATE_SHIPPED Payment Type: ☐ Cash ☐ Credit ☒ Order Filled

IMAGE: PRODUCT IMAGE

IMAGE: DESCRIPTION

Item Id	Product Id	Description	Price	Qty	Shipped	Item Total
ITE	PRODUC	DESCRIPTION	PRICE	QTY	SHIPPED	ITEM TOTAL
ITE	PRODUC	DESCRIPTION	PRICE	QTY	SHIPPED	ITEM TOTAL
ITE	PRODUC	DESCRIPTION	PRICE	QTY	SHIPPED	ITEM TOTAL
ITE	PRODUC	DESCRIPTION	PRICE	QTY	SHIPPED	ITEM TOTAL

Order Total: TOTAL

8. Thực hiện một truy vấn trong *ORDGxx* form để đảm bảo các items mới không gây ra lỗi. Hãy tắt thuộc tính Database Item đối với các items không tương ứng với các cột trong bảng trong cơ sở dữ liệu.

9. Tạo một iconic button tương tự như trong câu 4, trong Control block của form *CUSTGxx*. Sử dụng file *list.ico* (không kèm theo phần mở rộng .ico). Đặt tên cho nút là Sales_rep_lov_button, và đặt nó cạnh Sales_Rep_ID.

10. Cấu giữ, biên dịch và chạy forms để kiểm nghiệm các thay đổi.

6.10 Bài thực hành 10: Tạo Window và Content Canvases

6.10.1 Nội dung

- Thay đổi kích cỡ và vị trí của window trong customers form. Thay đổi tên và tiêu đề của Window. Ghi lại và chạy form.
- Chỉnh sửa tên và tiêu đề của window trong orders form.
- Tạo một window mới trong orders form. Hiển thị nội dung của S_INVENTORY block trong window này. Ghi lại và chạy form.

6.10.2 Các bước thực hành

1. Sửa đổi cửa sổ trong *CUSTGxx* form. Đổi tên cửa sổ thành WIN_CUSTOMER, và đổi tên tiêu đề của nó là Customer Information. Đặt lại kích thước và vị trí cho phù hợp.
2. Cất giữ, biên dịch và chạy form để kiểm nghiệm các sự thay đổi.
3. Sửa đổi cửa sổ trong *ORDGxx* form. Gọi cửa sổ đó là WIN_ORDER. Thay đổi tiêu đề của nó thành Orders and Items.
4. Trong *ORDGxx* form, tạo một cửa sổ mới gọi là WIN_INVENTORY để phù hợp với việc hiển thị CV_INVENTORY canvas. Sử dụng các rulers trong Layout editor để đặt Height và Width cho cửa sổ. Đặt tiêu đề cho cửa sổ là Stock Levels, và thuộc tính Hide on Exit là Yes. Đặt cửa sổ mới ở vị trí cho phù hợp với WIN_ORDER.
5. Kết hợp CV_INVENTORY canvas với cửa sổ WIN_INVENTORY. Chạy form để thấy được S_INVENTORY block hiển thị trong WIN_INVENTORY khi bạn điều khiển (navigate) block này.
6. Ghi form lại.

6.11 Bài thực hành 11: Làm việc với các Canvase khác

6.11.1 Nội dung

- Tạo một horizontal toolbar canvas trong orders form. Tạo một buttons mới trong control block, và đặt chúng vào horizontal toolbar. Ghi lại và chạy form.
- Tạo một stacked canvas trong orders form để đưa thêm dòng trợ giúp. Đặt vị trí của canvas ở chính giữa cửa sổ. Tạo một button trong control block. Button này sẽ được dùng và hiển thị trong stacked canvas. Đưa dòng trợ giúp vào stacked canvas. Ghi và chạy form.
- Tạo một tab canvas trong customers form. Tạo 3 trang tab trong canvas này, và đảm bảo rằng mỗi tab hiển thị các thông tin tương ứng. Ghi lại và chạy form.

6.11.2 Các bước thực hành

Toolbar Canvases

1. Trong *ORDGxx* form, tạo một horizontal toolbar canvas gọi là Toolbar. Trong cửa sổ WIN_ORDER, biến nó thành toolbar chuẩn cho cửa sổ. (Gợi ý đặt Height là 30).
2. Cất giữ, biên dịch và chạy form để kiểm nghiệm. Lưu ý, bây giờ toolbar sẽ chiếm một phần vị trí trên window. Thay đổi lại kích thước window cho phù hợp.
3. Tạo ba nút bấm trong Control block, theo chi tiết dưới đây, và đặt chúng vào Toolbar canvas.

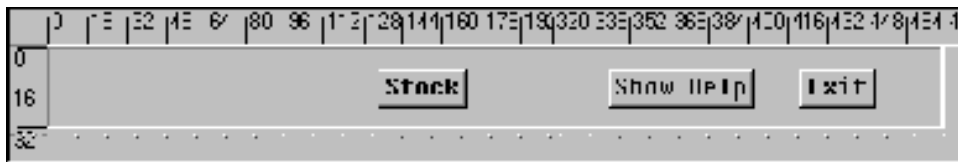
Tên nút (Button Name)Chi tiết (Details)

Stock_button	Label: Stock
--------------	--------------

GIÁO TRÌNH ĐÀO TẠO ORACLE FORM

	Mouse Navigate: No
	Keyboard Navigable: No
	Canvas: Toolbar
Show_Help_button	Label: Show Help
	Mouse Navigate: No
	Keyboard Navigable: No
	Canvas: Toolbar
Exit_button	Label: Exit
	Mouse Navigate: No
	Keyboard Navigable: No
	Canvas: Toolbar

Gợi ý các vị trí cho các nút được chỉ ra như dưới đây:



Stacked Canvases

1. Tạo một stacked canvas có tên là CV_HELP để hiển thị các trợ giúp trong cửa sổ WIN_ORDER của *ORDGxx* form. Gợi ý các thuộc tính:

<i>visible size</i>	là	Viewport
Width	là	270,
Viewport Height	là	215 (points).

Đặt các application help text vào canvas này.

2. Định vị trí khung nhìn của stacked canvas sao cho nó xuất hiện ở giữa của WIN_ORDER. Đảm bảo sẽ nó không che khuất enterable item đầu tiên.

Thực hiện điều này bằng cách định các vị trí top-left của khung nhìn trong Layout editor, trong khi hiển thị CV_ORDER. Định nghĩa Viewport X và Viewport Y, định vị trí trong Property Palette. Không di chuyển khung nhìn trong Layout Editor.

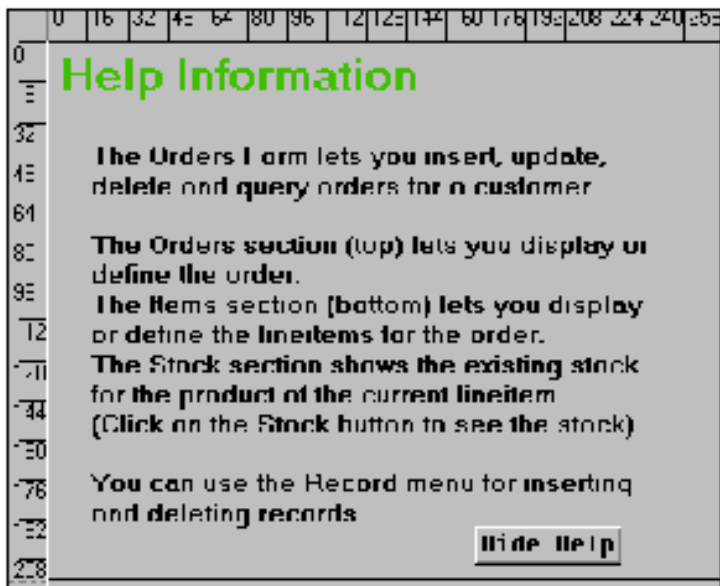
3. Thông qua Object Navigator , tổ chức CV_HELP sao cho nó là canvas nó là cuối cùng trong hàng đợi. (Việc này đảm bảo thứ tự stack đúng khi chạy).

4. Cài giữ, biên dịch và chạy form để kiểm nghiệm. Lưu ý: stacked canvas hiển thị ở mọi thời điểm, nên cần tính toán sao cho nó không che khuất item hiện thời trong form.

5. Tắt thuộc tính Visible của CV_HELP, sau đó tạo một nút bấm trong control block để làm ẩn đi các thông tin trợ giúp Help khi nó không còn cần thiết nữa.

Sau này, sẽ thêm vào các mã lệnh. Hiển thị nút bấm này trong CV_HELP canvas.

Tên nút	Chi tiết
Hide_Help_button	Label: Hide Help, Canvas: CV_HELP



Tab Canvases

Sửa đổi *CUSTGxx* form để sử dụng Tab canvas:

- 1 Trong Layout editor, xoá bỏ frame object bao trùm S_CUSTOMER block. Tạo một Tab canvas. Trong Layout editor đặt thuộc tính Background Color là gray, Tab style là Square, và Bevel là None.
- 2 Đổi tên Tab canvas này thành TAB_CUSTOMER. Tạo ba Tab pages và đặt tên chúng là Address, Billing, và Comments.
- 3 Thiết kế các Tab pages dựa trên hình minh hoạ dưới đây. Đặt các thuộc tính item sao cho chúng có thể xuất hiện (visible) trên các Tab pages tương ứng.

GIÁO TRÌNH ĐÀO TẠO ORACLE FORM

0 16 32 48 64 80 96 112 128 144 160 176 192 208 224 240 256 272 288 304 320 336 352 368 384

0 16 32 48 64 80 96 112 128 144 160 176 192 208 224 240 256 272 288 304 320

Id
ID

Customer Information

Address Billing Comments

COMMENTS

0 16 32 48 64 80 96 112 128 144 160 176 192 208 224 240 256 272 288 304 320 336 352 368 384

0 16 32 48 64 80 96 112 128 144 160 176 192 208 224 240 256 272 288 304 320

Id
ID

Customer Information

Address Billing Comments

Name NAME

Address ADDRESS

City CITY

State STATE

Country COUNTRY

Zip Code ZIP CODE

Phone PHONE

0 16 32 48 64 80 96 112 128 144 160 176 192 208 224 240 256 272 288 304 320 336 352 368 384

0 16 32 48 64 80 96 112 128 144 160 176 192 208 224 240 256 272 288 304 320

Id
ID

Customer Information

Address Billing Comments

Credit Rating
CREDIT_RATING

Sales Rep Id
SALES_REP_ID

4. Đặt lại thứ tự các items dựa theo thứ tự trang tab, để đảm bảo người sử dụng không dịch chuyển từ 1 tab này sang tab khác, mỗi khi chuyển đổi giữa các items. Đặt thuộc tính Next Navigation Item và Previous Navigation Item tùy theo thứ tự của các items trong các tab pages.

5. Cất giữ, biên dịch và chạy form.

6.12 [Bài thực hành 12: Giới thiệu về Triggers](#)

6.13 [Bài thực hành 13: Xây dựng Triggers](#)

6.13.1 Nội dung

- Dùng các built-ins để hiển thị LOVs
- Dùng When-Button-Pressed và When-Window-Closed triggers để đưa thêm tính năng cho items.
- Dùng built-ins để hiển thị và ẩn Help stacked canvas

6.13.2 Các bước thực hành

1. Trong *CUSTGxx* form, viết một trigger để hiển thị Sales_Rep_Lov, mỗi khi Sales_Rep_Lov_Button được bấm. Sử dụng Smart Triggers để tạo trigger mỗi khi bấm nút (When-Button-Pressed trigger). Tìm những built-in thích hợp trong các built-in packages, và sử dụng chức năng Paste Name and Arguments.

2. Tạo một When-Window-Closed trigger cho mỗi form level để thoát khỏi form.

3. Cất giữ, biên dịch và chạy form.

4. Trong *ORDGxx* form, viết một trigger để hiển thị Products_LOV mỗi khi Products_LOV_Button được chọn.

5. Viết một trigger để thoát khỏi form mỗi khi Exit_Button được chọn.

6. Cất giữ, biên dịch và chạy form.

7. Tạo một When-Button-Pressed trigger cho CONTROL.Show_Help_Button, sử dụng SHOW_VIEW built-in để hiển thị CV_HELP.

```
SHOW_VIEW('CV_HELP');
```

```
GO_ITEM('CONTROL.HIDE_HELP_BUTTON');
```

8. Tạo một When-Button-Pressed trigger cho CONTROL.Hide_Help_button để làm ẩn CV_HELP. Sử dụng HIDE_VIEW built-in để có được điều này.

```
HIDE_VIEW('CV_HELP');
```

```
GO_BLOCK('S_ORD');
```

9. Cất giữ, biên dịch và chạy *ORDGxx* form để kiểm nghiệm.

10. Tạo một When-Button-Pressed trigger cho CONTROL.Stock_Button, sử dụng GO_BLOCK built-in để hiển thị S_INVENTORY block.

6.14 [Bài thực hành 14: Tạo NonInput Items](#)

6.14.1 Nội dung

- Chạy mô đun này ở chế độ gỡ lỗi từng bước để theo dõi hoạt động của nó

6.14.2 Các bước thực hành

1. Mở *CUSTGxx.FMB*. Tạo một thủ tục procedure đặt tên là *List_Of_Values*. Nhập mã từ tệp *pr14_1.txt*.

```
PROCEDURE list_of_values(p_lov in VARCHAR2,p_text in VARCHAR2)
IS
v_lov BOOLEAN;
BEGIN
v_lov:= SHOW_LOV(p_lov);
IF v_lov THEN
MESSAGE('You have just selected a '||p_text);
ELSE
MESSAGE('You have just cancelled the List of Values');
END IF;
END;
```

Chỉnh sửa trigger When-Button-Pressed của *CONTROL.Sales_LOV_Button* như sau.

```
LIST_OF_VALUES('SALES_REP_LOV', 'Sales Representative');
```

Dịch và chạy form trong chế độ gỡ lỗi. Thiết lập một điểm dừng ở một trong các trigger, và giám sát ngăn xếp Call. Thử chạy từng bước qua mã để theo dõi quá trình xử lý.

6.15 Bài thực hành 15: Tạo NonInput Items

6.15.1 Nội dung

- Viết một trigger đánh giá mức độ tin cậy của khách hàng xem có phải bắt anh ta trả bằng tiền mặt.
- Tạo một toolbar button để hiển thị và che dấu các hình ảnh sản phẩm.

6.15.2 Các bước thực hành

1. Trong *ORDGxx* form viết một trigger, trigger này chạy khi có sự thay đổi *Payment Type*, chỉ cho phép các khách hàng có mức độ tin cậy tốt được trả theo thủ tục nợ. Có thể nhập tệp *pr15_1.txt*.

2. Trong *CONTROL* block, tạo một nút mới là *Image_Button* và đặt nó trên *Toolbar*. Thiết lập thuộc tính *Label* property là *Image Off*.

3. Nhập tệp *pr15_3.txt* vào một trigger để chạy khi *Image_Button* được nhấn. Tệp chứa mã xác định giá trị hiện thời của thuộc tính *visible* của mục *Product Image*. Nếu giá trị hiện thời là *True*, thuộc tính *visible* sẽ đổi thành *False* cho cả mục *Product Image* và mục *Image Description*. Cuối cùng sự thay đổi nhấn trên *Image_Button* được phản ánh qua trạng thái thay đổi kế tiếp của nó. Tuy nhiên, nếu thuộc tính *visible* hiện thời là *False* thì thuộc tính *visible* sẽ đổi thành *True* cho cả mục *Product Image* và mục *Image Description*.

Đặt Triger When-Button-Pressed trên *CONTROL.Image_Button* thành :

```
IF GET_ITEM_PROPERTY('S_ITEM.product_image',VISIBLE)='TRUE' THEN
SET_ITEM_PROPERTY('S_ITEM.product_image', VISIBLE, PROPERTY_FALSE);
SET_ITEM_PROPERTY('S_ITEM.image_description', VISIBLE, PROPERTY_FALSE);
SET_ITEM_PROPERTY('CONTROL.image_button',LABEL,'Image On');
ELSE
SET_ITEM_PROPERTY('S_ITEM.product_image', VISIBLE, PROPERTY_TRUE);
SET_ITEM_PROPERTY('S_ITEM.image_description', VISIBLE, PROPERTY_TRUE);
SET_ITEM_PROPERTY('CONTROL.image_button',LABEL, 'Image Off');
END IF;
```

4. Ghi, biên dịch và chạy form.

6.16 Bài thực hành 16: Tạo NonInput Items

6.16.1 Nội dung

- Sử dụng một alert để báo cho người sử dụng rằng khách hàng phải trả bằng tiền mặt.
- Sử dụng một alert chung hỏi người sử dụng xác nhận việc đóng form.

6.16.2 Các bước thực hành

1. Tạo một alert trong *ORDGxx* gọi là *Payment_Type_Alert* với một nút bấm OK button. Thông báo 'Khách hàng này phải trả bằng tiền mặt!'. Đặt thuộc tính Title thành Payment Type, Style thành Caution.
2. Sửa đổi When-Radio-Changed trigger trên *Payment_Type* để hiện ra *Payment_Type_Alert* thay cho thông báo khi một khách hàng phải trả bằng tiền mặt.
3. Tạo một cảnh báo - alert chung *Question_Alert* cho phép trả lời Yes hoặc No. Message property để trống cho cảnh báo này. Chọn thuộc tính Style là Stop và định nghĩa hai nút bấm trong cảnh báo: Yes và No.
4. Sửa đổi When-Button-Pressed trigger trên *CONTROL.Exit_Button* để sử dụng *Question_Alert* để hỏi người sử dụng việc đóng form. Gọi *SET_ALERT_PROPERTY* built-in để định nghĩa thông báo: 'Bạn có muốn đóng form không?'. Kiểm tra giá trị trả về *SHOW_ALERT*, và gọi *EXIT_FORM* built-in nếu người sử dụng trả lời là Yes.
- 5 Ghi, biên dịch và chạy form để kiểm tra.

6.17 Bài thực hành 17: Truy vấn Triggers

6.17.1 Nội dung

- Bố trí customer names và sales representative names vào mỗi dòng của khối *S_ORD*
- Bố trí các mô tả vào mỗi hàng của khối *S_ITEM*
- Vô hiệu hoá nút bấm Exit trong chế độ Enter Query
- Thêm 2 hộp kiểm tra cho phép lựa chọn chế độ truy vấn.

6.17.2 Các bước thực hành

1. Trong *ORDGxx* form, viết một trigger để đặt *Customer_Name* và *Sales_Rep_Name* vào mỗi dòng được trả về bởi một câu hỏi trên khối *S_ORD*.
2. Viết một trigger đặt *Description* cho mỗi dòng được trả về bởi một câu hỏi trên khối *S_ITEM*.
3. Hãy đảm bảo rằng *Exit_Button* không hiệu quả trong chế độ Enter Query. Chỉnh sửa giao diện câu hỏi mặc định. Mở mô đun form Customers. Thêm một hộp kiểm gọi là *Control.Case_Sensitive* vào form để người sử dụng có thể chỉ rõ có hay không một câu hỏi cho tên một khách hàng sẽ là case sensitive. Bạn có thể nhập tệp *pr17_4.txt* vào When-Checkbox-Changed trigger. Thiết lập giá trị thuộc tính khởi tạo là 'Y.' Trong khối *Control*, thêm một control box (gọi là *Control.Case_Sensitive* như chỉ ra dưới đây)và tạo trigger có tên là **When-Checkbox-Changed trigger trên mục CONTROL.Case_Sensitive**

```
IF NVL(:CONTROL.case_sensitive, 'Y') = 'Y' THEN
    SET_ITEM_PROPERTY('S_CUSTOMER.name', CASE_INSENSITIVE_QUERY,
                                PROPERTY_FALSE);
ELSE
    SET_ITEM_PROPERTY('S_CUSTOMER.name', CASE_INSENSITIVE_QUERY,
                                PROPERTY_TRUE);
END IF;
```

5. Thêm control box Control.Exact_Match vào form để người sử dụng có thể chỉ rõ có hay không có một điều kiện hỏi cho một tên khách hàng phải phù hợp với giá trị bảng (Nếu không có sự phù hợp được thừa nhận, giá trị tìm kiếm có thể là một phần của giá trị bảng). Bạn có thể nhập tệp *pr17_5.txt* vào Pre-Query Trigger. Thiết lập giá trị thuộc tính khởi tạo là 'Y'. Thêm một control box (gọi là Control.Exact_Match như chỉ ra dưới đây) để điều khiển khối và tạo trigger **Pre-Query trigger trên S_CUSTOMER block**

```
IF NVL( :CONTROL.exact_match, 'Y' ) = 'N' THEN
  :S_CUSTOMER.name := '%' || :S_CUSTOMER.name || '%';
END IF;
```

6.18 Bài thực hành 18: Kiểm tra tính hợp lệ

6.18.1 Nội dung

- Kiểm tra giá trị của sales representative item dùng LOV
- Viết trigger kiểm tra ngày shipped date không trước ngày ordered date
- Chỉnh vị trí của customer names, sales representative names và ID khi customer ID bị thay đổi
- Viết trigger kiểm tra tên và giá của product khi product ID bị thay đổi.

6.18.2 Các bước thực hành

1. Trong CUSTGxx form, hiển thị Sales_Rep_Lov khi người dùng nhập mã Sales_Rep_Id không có trong CSDL.
2. Ghi, biên dịch và chạy form để kiểm tra.
3. Trong ORDGxx form, viết một validation trigger để kiểm soát Date_Shipped là sau Date_Ordered. Viết một When-Validate-record trigger để so sánh giá trị của Date_Shipped và Date_Ordered. Nếu Date_Shipped là trước Date_Ordered ngừng trigger với một thông báo thích hợp.
4. Trong ORDGxx form, tạo một trigger để viết giá trị đúng cho các mục Customer_Name, Sales_Rep_Name, và Sales_Rep_Id khi một kiểm tra hợp lệ xuất hiện trên Customer_Id. Không chạy trigger nếu không tìm thấy khách hàng.
5. Tạo một validation trigger khác trên S_ITEM.Product_Id để lấy tên sản phẩm và viết nó vào mục Description. Không chạy trigger và hiển thị một thông báo nếu không tìm thấy sản phẩm.

6.19 Bài thực hành 19: Tạo NonInput Items

6.19.1 Nội dung

- Thực hiện một truy vấn khi form khởi động
- Thay đổi hình ảnh sản phẩm khi con trỏ di chuyển trên mỗi bản ghi của S_ITEM

6.19.2 Các bước thực hành

1. Viết một When-New-Form-Instance trigger trên ORDGxx form để thực hiện truy vấn khi khởi tạo form. Sử dụng EXECUTE_QUERY built-in.
 2. Viết một trigger chạy khi con trỏ di chuyển vào mỗi record của S_ITEM block, và đặt vào Product_Image hình ảnh sản phẩm tương ứng nếu có. Sử dụng hàm Get_Product_Image.
- Hàm Get_Product_Image function đã có sẵn. Hàm này trả về tên tệp hình ảnh với đầu vào là số hiệu sản phẩm đã cho. Nếu không tìm thấy tệp, hàm trả về giá trị No file.

```
FUNCTION get_product_image (product_number IN NUMBER) RETURN VARCHAR2 IS
v_filename VARCHAR2(20);
BEGIN
    SELECT s_image.filename INTO v_filename
    FROM s_image, s_product
    WHERE s_image.id = s_product.image_id AND s_product.id = product_number;
    IF v_filename is null THEN
        v_filename := 'No file';
    END IF;
    RETURN v_filename;
EXCEPTION
    WHEN no_data_found THEN return('No file');
END;
```

Nếu hàm trả về một tên tệp thích hợp, trigger phải chuyển tên này tới READ_IMAGE_FILE built-in.

4. Định nghĩa cùng kiểu trigger và viết mã cho S_ORD block. Nó sẽ hiển thị hình ảnh cho các sản phẩm của dòng mục đầu tiên của S_ITEM.

5. Tìm các vị trí có thể đặt đoạn mã này?

6. Ghi, biên dịch và chạy form để kiểm tra.

6.20 Bài thực hành 20: Xử lý các giao dịch (Transaction Processing)

6.20.1 Nội dung

- Tự động đặt thứ tự giá trị order ID dùng sequence
- Tự động đặt giá trị cho mục ID khi thêm bản ghi bằng giá trị cao nhất của order ID +1
- Chèn, cập nhật, xoá các row trong view
- Chỉnh sửa các commit messages trong customers form
- Chỉnh sửa màn hình login trong customers form.

6.20.2 Các bước thực hành

1. Trong *ORDGxx* form viết một transactional trigger trong S_ORD block trên S_ORD.Id với giá trị kế tiếp trong S_ORD_ID sequence.

Tạo một Pre-Insert trigger gán giá trị theo trình tự này. Nếu có một ngoại lệ (exception) xảy ra trong trigger, khi đó huỷ bỏ trigger với một thông điệp báo lỗi.

2 Trong S_ORD block đặt thuộc tính Enabled cho ID item là No.

3 Cất giữ, biên dịch và chạy form để kiểm nghiệm.

Chèn thêm một đơn đặt hàng mới (new order). Giá trị Id duy nhất cho đơn đặt hàng phải xuất hiện mỗi khi bạn cất giữ nó.

4 Tạo một trigger tương tự trên S_ITEM block để gán giá trị cho Item_Id khi có một bản ghi mới được lưu giữ. Số này lấy bằng cách thêm một giá trị cho số Item_id cao nhất, thực hiện hành động trong Pre-Insert Trigger. Đặt thuộc tính Required và Enabled là No cho Item_id.

5 Cất giữ, biên dịch và chạy form để kiểm nghiệm.

Chèn thêm một line-item record mới trong S_ITEM block, rồi cất giữ.

Ghi chú

Giải pháp 20.4 không phải là cách an toàn nhất. Giải pháp tốt hơn là giữ toàn bộ các hàng vào một bảng khác có thể khoá, nhưng giải pháp này không phù hợp trong giai đoạn này.

6. Employees form dựa trên view có các cột ở các bảng S_EMP và S_DEPT. Mở Employees form. Kiểm tra xem nó có cho phép người sử dụng thêm các nhân công mới hay sửa chữa các nhân công đã có bằng form này.

GIÁO TRÌNH ĐÀO TẠO ORACLE FORM

Nội dung các Trigger như sau:

On-Insert trigger trên EMP block (trên view employee)

```
INSERT INTO S_EMP VALUES
:EMP.ID,
:EMP.LAST_NAME,
:EMP.FIRST_NAME,
:EMP.USERID,
:EMP.START_DATE,
:EMP.COMMENTS,
:EMP.MANAGER_ID,
:EMP.TITLE,
:EMP.DEPT_ID,
:EMP.SALARY,
:EMP.COMMISSION_PCT);
```

On-Update trigger trên EMP block (trên view employee)

```
UPDATE S_EMP
SET
ID =:EMP.ID,
LAST_NAME =:EMP.LAST_NAME,
FIRST_NAME =:EMP.FIRST_NAME,
USERID =:EMP.USERID,
START_DATE =:EMP.START_DATE,
COMMENTS =:EMP.COMMENTS,
MANAGER_ID =:EMP.MANAGER_ID,
TITLE =:EMP.TITLE,
DEPT_ID =:EMP.DEPT_ID,
SALARY =:EMP.SALARY,
COMMISSION_PCT =:EMP.COMMISSION_PCT
WHERE ID = :EMP.NUM;
```

7. Có thể xóa một nhân công không?

8. Với cương vị người thiết kế, thẩm tra các trigger thực hiện đang tồn tại. Tạo một trigger thực hiện mới cho phép xóa một nhân công.

9. Mở *CUSTGxx* form. Tạo ba biến toàn cục GLOBAL.INSERT, GLOBAL.UPDATE, và GLOBAL.DELETE. Các biến này cho biết số lần chèn, cập nhật, xóa. Nên viết các trigger Post-Insert, Post-Update, và Post-Delete để khởi tạo và tăng giá trị các biến toàn cục trên.

10. Tạo thủ tục HANDLE_MESSAGE. Nhập tệp *pr20_10.txt*. Thủ tục này nhận 2 đối số: đối số thứ nhất là số hiệu thông báo, và đối số thứ hai là bộ chỉ lỗi logic. Thủ tục này sử dụng ba biến toàn cục để hiển thị một thông báo và sau đó xóa bỏ các biến toàn cục này. Nội dung thủ tục như sau

```
PROCEDURE handle_message( message_number IN NUMBER, IS_ERROR IN
BOOLEAN ) IS
BEGIN
IF message_number IN ( 40400, 40406, 40407 ) THEN
DEFAULT_VALUE( '0', 'GLOBAL.insert' );
DEFAULT_VALUE( '0', 'GLOBAL.update' );
DEFAULT_VALUE( '0', 'GLOBAL.delete' );
MESSAGE('Save Ok: ' ||
:GLOBAL.insert || 'records inserted, ' ||
:GLOBAL.update || 'records updated, ' ||
:GLOBAL.delete || 'records deleted !!!' );
ELSIF is_error = TRUE THEN
MESSAGE('ERROR: ' || ERROR_TEXT );
ELSE
MESSAGE( MESSAGE_TEXT );
END IF;
END ;
```

Gọi thủ tục khi xuất hiện lỗi với hai tham số là mã lỗi và TRUE.

Gọi thủ tục khi xuất hiện một thông báo với 2 tham số là mã thông báo và FALSE.

11. Mở mô đun form *CUSTGxx*. Viết một On-Logon trigger để kiểm soát số lượng các điểm kết nối. Sử dụng LOGON_SCREEN built-in dựa theo màn hình mặc định và LOGON để kết nối tới CSDL.. Có thể nhập tệp *pr20_11.txt*. Nội dung Trigger On-Logon tại mức Form level là:

```
DECLARE
connected BOOLEAN := FALSE;
tries NUMBER := 3;
un VARCHAR2(30);
pw VARCHAR2(30);
cs VARCHAR2(30);
BEGIN
SET_APPLICATION_PROPERTY(CURSOR_STYLE, 'DEFAULT');
WHILE connected = FALSE and tries > 0 LOOP
LOGON_SCREEN;
un := GET_APPLICATION_PROPERTY( USERNAME );
pw := GET_APPLICATION_PROPERTY( PASSWORD );
cs := GET_APPLICATION_PROPERTY( CONNECT_STRING );
LOGON( un, pw || '@' || cs, FALSE );
IF FORM_SUCCESS THEN
connected := TRUE ;
END IF;
tries := tries - 1;
END LOOP;
IF NOT CONNECTED THEN
MESSAGE('Too many tries!');
RAISE FORM_TRIGGER_FAILURE;
END IF;
END;
```

6.21 Bài thực hành 21: Viết một đoạn mã động (Flexible Code)

6.21.1 Nội dung

- Chỉ đặt các hình ảnh sản phẩm khi các mục hình ảnh được hiển thị
- Sửa chữa When-Button-Pressed trigger của Image_Button để sử dụng đối tượng ID thay vì tên đối tượng

6.21.2 Các bước thực hành

1. Trong *ORDGxx* form sửa đổi các trigger để đặt các mục Product_Image khi mục hình ảnh được hiển thị. Thêm một đoạn mã để kiểm tra Product_Image. Chỉ thực hiện các hành động của trigger nếu hình ảnh được hiển thị vào thời điểm này. Sử dụng hàm built-in GET_ITEM_PROPERTY.
2. Sửa chữa trigger When-Button-Pressed trên Image_Button để sử dụng các ID đối tượng. Sử dụng một hàm FIND_object để lấy ID cho mỗi mục được tham chiếu bởi trigger. Khai báo các biến cho các ID này và sử dụng chúng trong mỗi mục tham chiếu trong trigger.
3. Ghi nhận, biên dịch và thực hiện form để kiểm tra các điểm này.

6.22 Bài thực hành 22: Chia sẻ các Objects và Code

6.22.1 Nội dung

- Tạo một Object Group và sử dụng Object Group này trong một mô đun form mới
- Sử dụng các lớp Property
- Tạo một Object Library và sử dụng Object Library này trong một mô đun form mới
- Thiết lập và sử dụng SmartClasses

6.22.2 Các bước thực hành

1. Trong *ORDGxx* form tạo một nhóm đối tượng gọi là Stock_Objects, chứa khối S_INVENTORY, CV_INVENTORY và WIN_INVENTORY.

GIÁO TRÌNH ĐÀO TẠO ORACLE FORM

2. Ghiform.

3. Tạo một môđun form mới và sao chép vào nó nhóm đối tượng Stock_Objects.

4. Trong môđun form mới, Tạo một Property Class gọi là ClassA. Bao gồm các thuộc tính sau được thiết lập:

Property	Setting
Font Name	Arial
Format Mask	99,999
Font Size	8
Justification	Right
Delete Allowed	No
Background Color	DarkRed

5. áp dụng ClassA cho CV_INVENTORY, mục Restock_Date và mục Max_In_Stock.

6. Ghi form module với tên *STOCKxx.fmb*, biên dịch và chạy form này và lưu ý các lỗi.

7. Tạo cho Restock_Date Format Mask một thuộc tính khác. Thay đổi Format Mask cho S_INVENTORY.restock_date theo MM/DD/YYYY.

8. Ghi, biên dịch và chạy form.

9. Tạo một thư viện đối tượng với tên SUMMIT. Tạo 2 tab trong thư viện đối tượng gọi là personal và corporate. Thêm khối Control, TOOLBAR, và Question_Alert vào tab personal của thư viện đối tượng. Ghi nhận thư viện đối tượng này thành SUMMIT.OLB.

10. Tạo một form mới và tạo một khối dữ liệu trên cơ sở bảng S_DEPT. Lôi TOOLBAR canvas, khối CONTROL và Question_Alert từ thư viện đối tượng vào new form. Phân lớp các đối tượng. Một số mục không được áp dụng cho form này. Thiết lập các mục dưới đây thuộc tính Canvas là NULL: Image_button, Stock_button, Show_help_button, Product_Lov_button, Hide_Help_button. Sử dụng TOOLBAR theo khuôn dạng ngang cho form này. Thiết lập thuộc tính Window là WINDOW1 cho TOOLBAR canvas. Ghi form này thành *DEPTGxx*, biên dịch và chạy form này để kiểm tra.

11. Thử xóa một mục trên NULL canvas. Điều gì sẽ xảy ra, giải thích tại sao?

12. Tạo hai nút bấm ví dụ, một nút bấm rộng và một nút bấm vừa. Tạo một ví dụ trường ngày tháng. Thiết lập độ rộng và format mask phù hợp với tiêu chuẩn bạn đưa ra. Kéo các mục này vào thư viện đối tượng của bạn. Đánh dấu các mục này như là SmartClasses. Tạo một new form và một khối dữ liệu mới trong nó. Cung cấp những SmartClasses này vào form của bạn.

6.23 Bài thực hành 23: Tạo Multiple Form Applications

6.23.1 Nội dung

- Liên kết các form Orders và Customers bằng cách sử dụng một biến toàn cục
- Sử dụng built-ins để kiểm tra xem có phải form Orders đang thực hiện
- Sử dụng các biến toàn cục để hạn chế câu hỏi trong Orders form

6.23.2 Các bước thực hành

1. Trong *ORDGxx* form, tạo một trigger Pre-Form để chắc chắn rằng biến toàn cục Customer_Id tồn tại. Sử dụng DEFAULT_VALUE built-in, và thiết lập biến thành Null nếu nó chưa tồn tại.

GIÁO TRÌNH ĐÀO TẠO ORACLE FORM

2. Thêm một trigger để đảm bảo câu hỏi trong khối S_ORD được giới hạn giá trị bởi GLOBAL.Customer_Id. Viết một Pre-Query trigger trên khối S_ORD để gán giá trị biến toàn cục chu mục Customer_Id.
3. Ghi, biên dịch và chạy form để kiểm tra nó làm việc khi đứng một mình.
4. Trong CUSTGxx form, Tạo một nút bấm khối Control gọi là Orders_Button.
5. Định nghĩa một trigger cho Control.Orders_Button để khởi tạo GLOBAL.Customer_Id với ID của khách hàng hiện thời, sau đó mở ORDGxx form, chuyển điều khiển tới nó, sử dụng OPEN_FORM. Chắc chắn rằng tên trong Object Navigator của ORDGxx form giống với tên vật lý.
6. Ghi và biên dịch mỗi form, sau đó chạy ứng dụng để kiểm tra. Chạy CUSTGxx form, sau đó mở ORDGxx form từ nút tạo mới. So sánh hai form?
7. Thay đổi vị trí cửa sổ của ORDGxx form nếu cần.
8. Sửa đổi Orders_Button trigger trong CUSTGxx để nó sử dụng GO_FORM để chuyển điều khiển đến ORDGxx nếu form này đã đang chạy. Sử dụng FIND_FORM built-in cho mục đích này. Nên nhớ rằng bạn cần sử dụng tên môđun trong GO_FORM built-in và tên tệp trong OPEN_FORM built-in.
9. Viết một When-Create-Record trigger trên khối S_ORD để sử dụng giá trị của GLOBAL.Customer_Id như giá trị mặc định cho S_ORD.Customer_Id.
10. Thêm mã vào CUSTGxx form để GLOBAL.Customer_Id được cập nhật khi Customer_Id hiện thời có thay đổi.



This document was created with the Win2PDF "print to PDF" printer available at
<http://www.win2pdf.com>

This version of Win2PDF 10 is for evaluation and non-commercial use only.

This page will not be added after purchasing Win2PDF.

<http://www.win2pdf.com/purchase/>