

4-1 Part 2: 表單結構分析

目標

瞭解表單的結構與執行過程，以便能夠手動建立表單。

重要概念包括：

- 表單顯示過程 (Show Page process)
 - 顯示前、顯示中、顯示後的階段
 - 可以在顯示前階段執行的元件
 - 表單初始化過程
- 接受頁面過程 (Accept Page process)
- 按鈕行為設定
- Form Region 與資料表之間的關係; 表單元件(item)與資料表欄位(table column)之間的關係

1. 表單顯示過程

表單顯示過程包括三個階段：顯示前、顯示中(rendering)、顯示後。

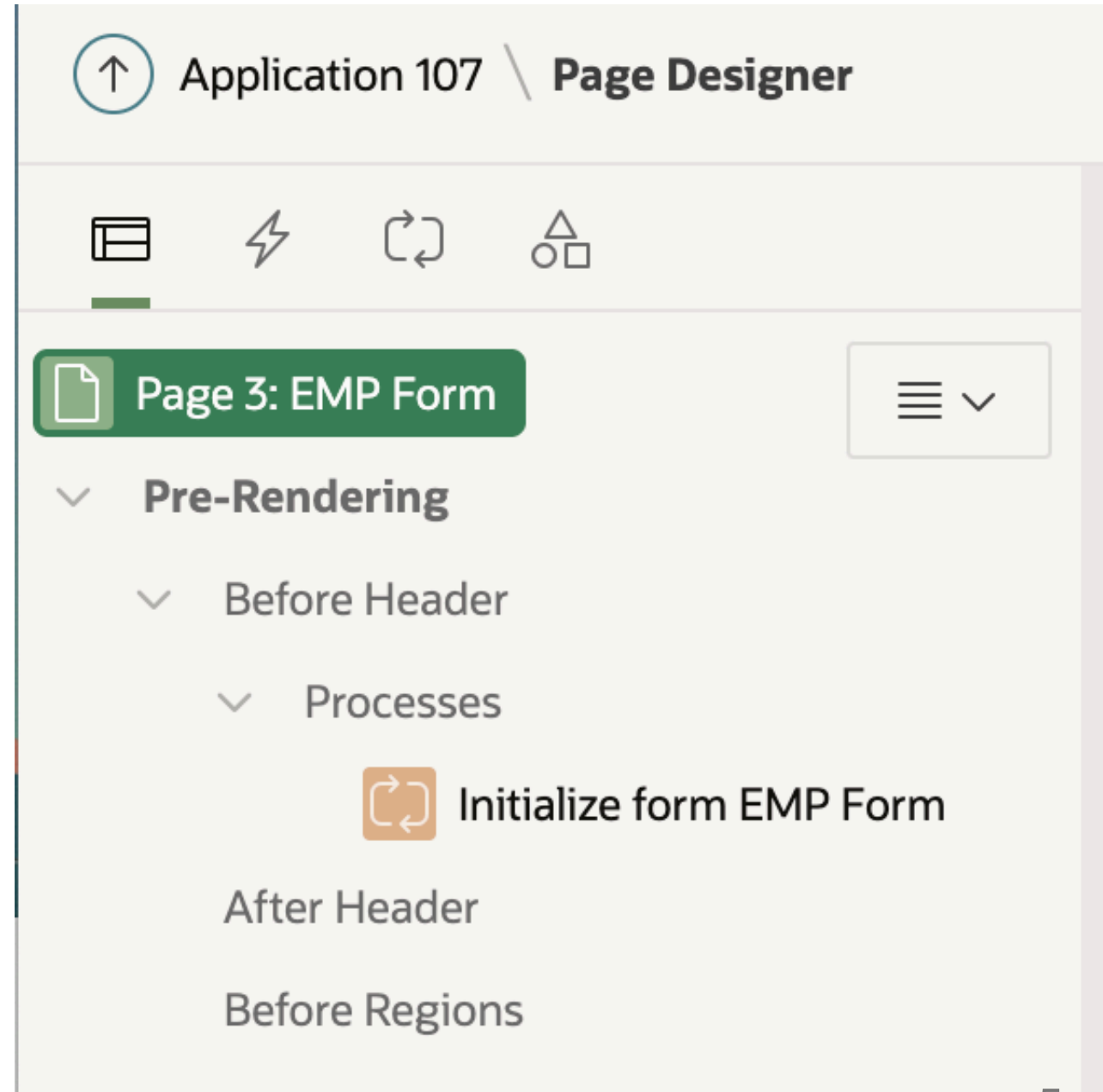
Page Designer 的左側欄位顯示了這三個階段。

顯示前(Pre-rendering)

- 可以在顯示前階段執行: 程序(process), 計算(computation), 分支(branch)
- 程序(Process)指在 Server 端執行的 PL/SQL 程式碼
 - e.g. 要使用表單編輯一筆資料時，我們需要於顯示前階段去資料表取得資料，並將資料放置到表單欄位(form item)
- 計算(Computation)在 Server 端，執行 SQL 或 PL/SQL 程式碼，並將計算結果放置到某個表單元件。
 - 常用於表單上的計算欄位
- 分支(Branch)是一個條件式，當滿足條件時，將導向到另一個頁面。

Page Wizard 為我們自動加入的程序:
`Initialize form FORM`，放置在顯示前階段的 Before Header 點。

- 這個程序初始化表單元件，使用表單的主鍵值去資料表取得一筆資料，並將資料放置到表單元件。



顯示中(Rendering)

- 這個階段根據頁面上的版面配置，顯示表單元件。
- 這些表單元件包括:
 - 區域元件(Region components)
 - 項目元件(Item components)
 - 按鈕(Buttons)

The image shows the APEX form rendering process and the resulting form page. On the left, a tree view under 'Pre-Rendering' shows the structure: 'Before Header' (Processes: Initialize form EMP Form), 'After Header', 'Before Regions', 'Regions' (highlighted), 'Content Body' (EMP Form), and 'Region Buttons' (CANCEL, DELETE, SAVE, CREATE). On the right, the rendered 'EMP Form' page is shown. It includes a 'PAGE HEADER' section with 'PAGE NAVIGATION', 'BREADCRUMB BAR', 'BEFORE CONTENT BODY', and 'CONTENT BODY'. The 'CONTENT BODY' contains the 'EMP Form' with fields: P3_EMPNO, P3_ENAME, P3_JOB, P3_MGR, P3_HIREDATE, P3_SAL, P3_COMM, and P3_DEPTNO. Below the fields are 'REGION CONTENT' and 'SUB REGIONS' sections, each containing buttons: CANCEL, DELETE, SAVE, and CREATE.

依條件顯示元件

我們可以設定元件的 Server-side 條件，依據條件來決定元件是否顯示。

- Server-side condition 指這些條件是在 database 上運算，而不是在 Browser 上 (client-side) 上運算

在表單中, Page Wizard 加入了四個按鈕到頁面, 這些按鈕會在不同的狀態下顯示給使用者。

三個按鈕有條件設定，分別是 DELETE, SAVE, CREATE。

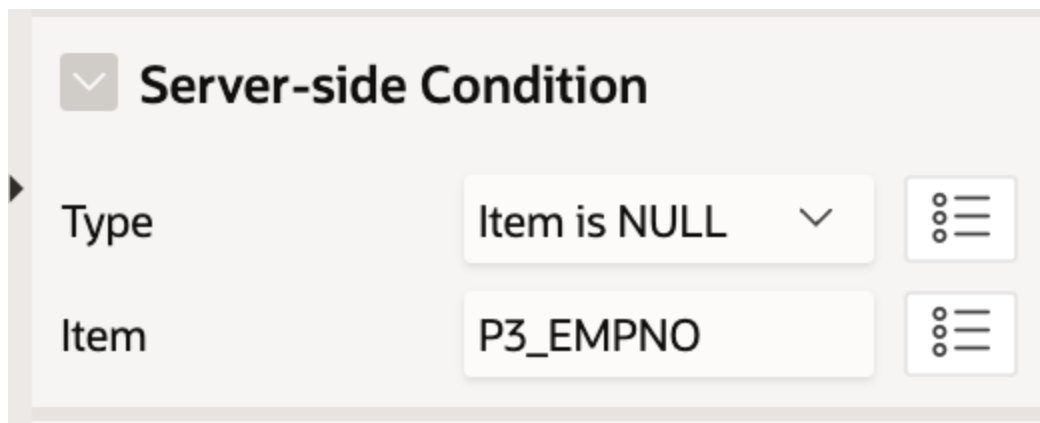
- 這三個按鈕在 Page Designer 中被標示紅點。

CREATE 按鈕的顯示條件

當我們要新增一筆資料時，我們希望顯示 CREATE 按鈕。

當新增一筆資料時，表單是空的，主鍵尚未產生。所以當主鍵值為 NULL 時，表單處於新增狀態。

依照這個邏輯，我們設定了 CREATE 按鈕的顯示條件為 `P3_EMPNO` 為 NULL。



The screenshot shows a configuration panel for a 'Server-side Condition'. It has a title 'Server-side Condition' with a dropdown arrow. Below the title, there are two rows of configuration:

Type	Condition	Icon
Type	Item is NULL	⋮
Item	P3_EMPNO	⋮

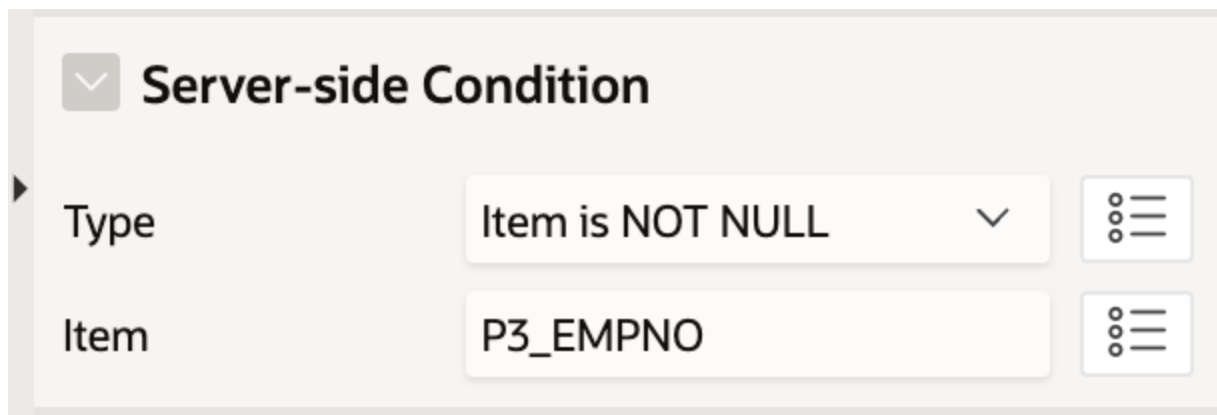
SAVE 按鈕的顯示條件

當我們要編輯一筆資料時，我們希望顯示 SAVE 按鈕。

要編輯的前題是表單已經有資料，表單的主鍵值不為 NULL。

所以當主鍵值不為 NULL 時，表單處於編輯狀態。

依照這個邏輯，我們設定了 SAVE 按鈕的顯示條件為 **P3_EMPNO** 不為 NULL。



▼ Server-side Condition

Type	Item is NOT NULL ▼	⋮
Item	P3_EMPNO	⋮

DELETE 按鈕的顯示條件

我們在編輯時允許刪除資料，此時需要顯示 DELETE 按鈕。

DELETE 按鈕的顯示條件與 SAVE 按鈕相同。

顯示後(Post-rendering)

- 這個階段執行在頁面顯示後，通常用來執行一些後續程序。
- 我們可以在這個階段執行: 程序與計算。

Page Wizard 並沒有加入任何程序到顯示後階段。

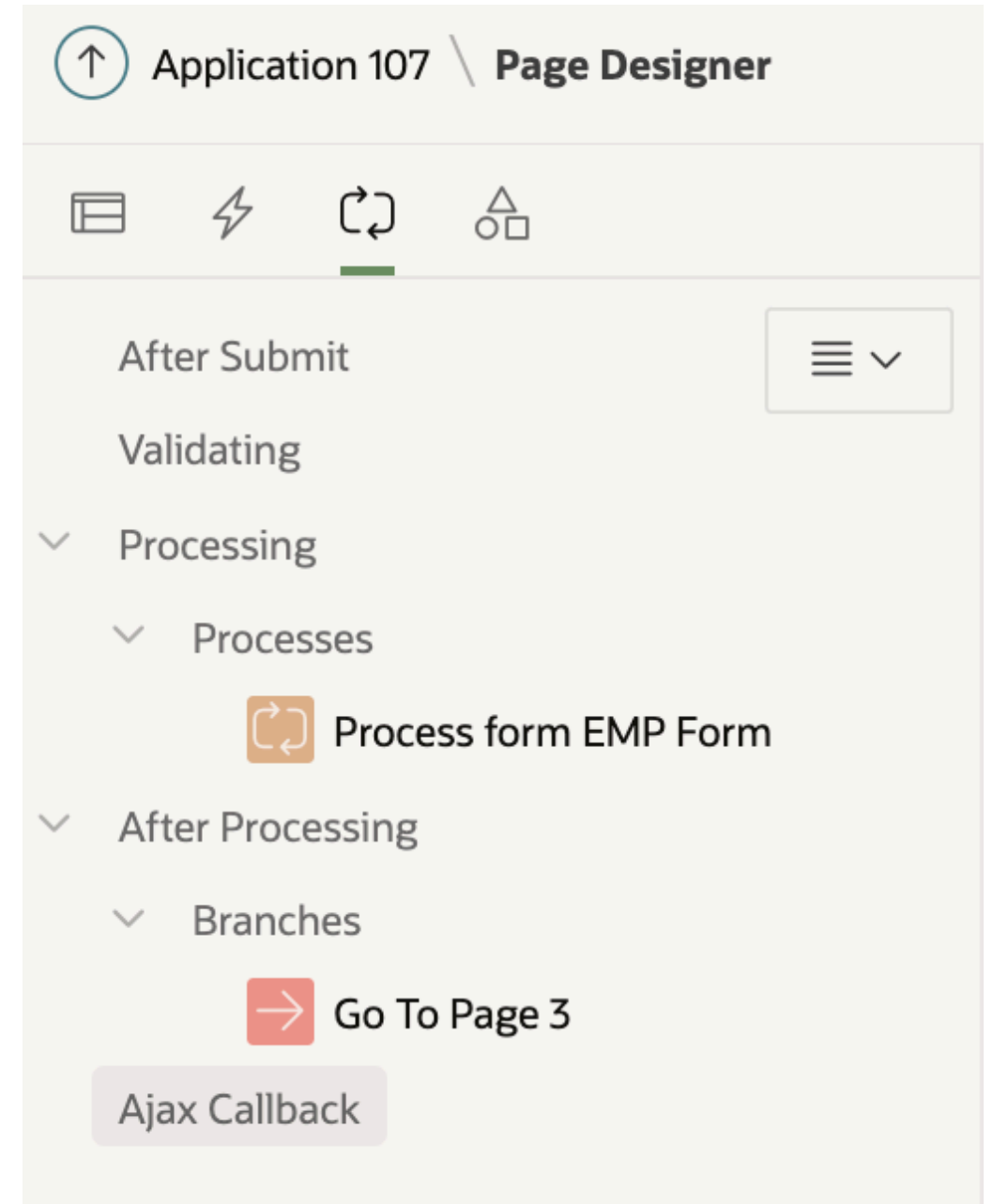
2. 接受頁面過程 (Accept Page process)

當使用者提交表單時，Apex 會執行接受頁面過程。

這個過程包括以下數個階段：

- 提交後(After Submit)
- 驗證(Validation)
- 處理(Processing)
- 處理後(After Processing)

在每個階段處理不同的工作，以下分別說明。



提交後(After Submit)

- 可以在這個階段執执行程序、計算、分支
- 例如, 在進入驗證階段前，將項目元件(Item Component)內的值格式化或轉成大寫。
 - 或者, 直接轉向到另一個頁面，而不進入驗證階段。

驗證(Validation)

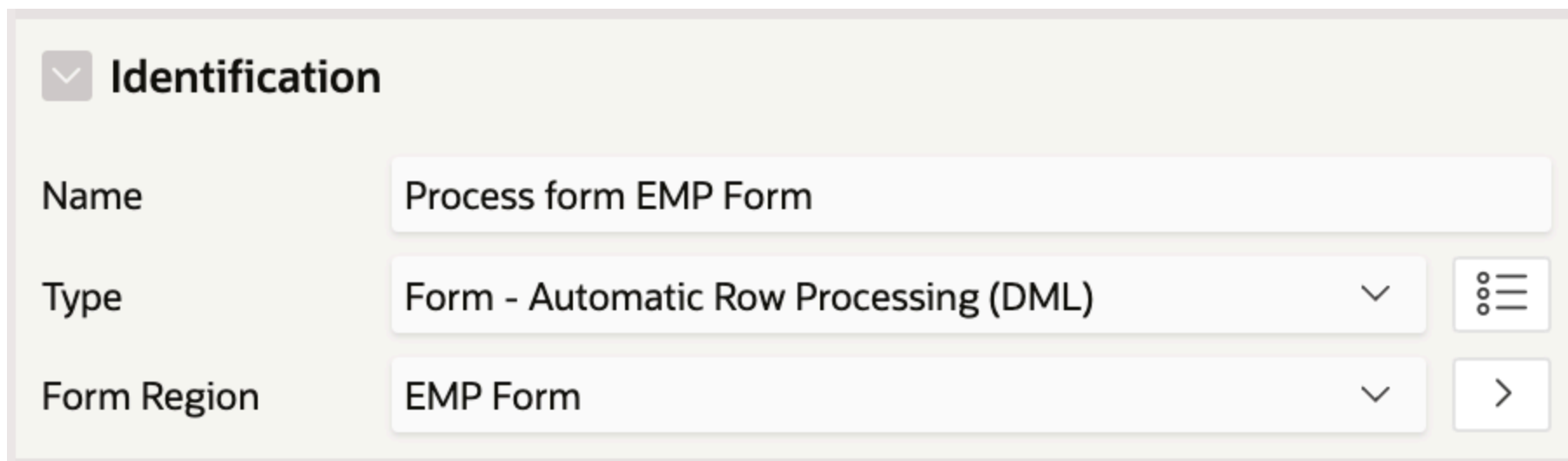
- 驗證階段用來檢查表單元件的值是否符合規則。
 - 例如, 檢查日期欄位是否為有效日期，或者檢查數字欄位是否為數字。
- 可以在這個階段執行: 驗證程序(Validation Process)、與分支(Branch)

處理(Processing)





- 在這個階段執执行程序(Process) 將表單元件上的資料寫入或更新資料表。
- 程序可以是內建程序(built-in process)或自訂的 PL/SQL 程式。

Page Wizard 在此階段加入了一個內建程序: Automatic Row Processing (DML)。

- 這個程序會根據按鈕元件行為，自動產生及執行 SQL INSERT 或 UPDATE 語句，更新資料表。
- Automatic Row Processing (DML) 需要設定要處理的 Form Region
 - 一個 Form Region 會對應到一個資料表。



The screenshot shows the 'Identification' step of the Page Wizard. It contains three fields for configuring the form:

Identification	
Name	Process form EMP Form
Type	Form - Automatic Row Processing (DML)  
Form Region	EMP Form  

系統提供的內建程序列表:

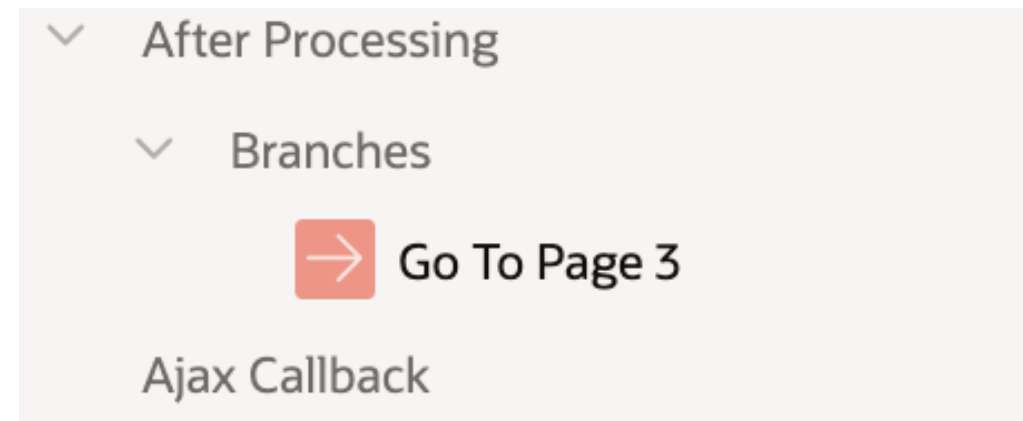
Layout	Page Search	Help
Type		
Specify which process type is executed. The following process types are supported:		
Clear Session State		
Process to clear session state stored within Application Express		
Close Dialog		
Process to close the current modal or non-modal dialog page.		
Data Loading		
Process to load data into the target table or collection based on the given data load definition.		
Execute Code		
Process to execute PL/SQL code.		
Form - Automatic Row Processing (DML)		
Process to insert, update, or delete a form region row(s).		
Form - Initialization		
Process to initialize form region items. Initialization can either be fetching data from the region source, using the primary key value(s) or simple initialization of the form region items.		
Interactive Grid - Automatic Row Processing (DML)		
Process to insert, update, or delete Interactive Grid rows.		
Reset Pagination		
Process to reset pagination of reports on the current page.		
Send E-Mail		
Process to send an email, and optionally one or more attachments, from the application.		
User Preferences		
Process to set User Preferences for the end user.		
Web Service		
Process to consume the specified Web service.		

處理後(After Processing)

處理後階段只能加入分支(Branch)。

應用的情境: 表單處理完畢後，導向資料列表頁面或者首頁。

Page Wizard 在此階段加入了一個分支(Branch)。



Identification	
Name	Go To Page 3
Execution Options	
Sequence	1
Point	After Processing
Behavior	
Type	Page or URL (Redirect)
Target	Page 3

3. Form Region 與資料表之間的關係

- Form Region 是一個區域元件，用來顯示表單元件。

Q: Form - Automatic Row Processing (DML) 如何產生對應的 DML 語句？如何知道資料表及欄位資訊？

A: Form Region 上會有資料表的資訊。

- Form Region 的 Data Source 屬性指定了資料表

The screenshot shows the configuration for a Form Region in Oracle APEX. It is divided into two main sections: Identification and Source.

- Identification:**
 - Title: EMP Form
 - Type: Form
- Source:**
 - Location: Local Database
 - Type: Table / View
 - Table Owner: Parsing Schema
 - Table Name: EMP (This field is highlighted with a yellow border)
 - Include ROWID Column: ☐
 - Where Clause: (empty)

Q: 那如何知道表單元件(item)與資料表欄位(table column)之間的對應關係?

Form Region 上的項目元件(Item)會與資料表的欄位(table column)對應.

在 Item 的 Source 屬性，我們必須指定:

- Form Region：該表單元件屬於哪個 Form Region
- Column：該表單元件對應到資料表的哪個欄位
- Data type：該欄位的資料型態
- Primary Key：該欄位是否為主鍵

The screenshot displays the 'Item Properties' dialog box for an item named 'P3_EMPNO'. The 'Source' tab is selected and highlighted with a yellow border. The dialog is organized into several sections: Identification, Settings, Layout, Appearance, Advanced, and Source. In the 'Source' section, the 'Form Region' is set to 'EMP Form', the 'Column' is 'EMPNO', and the 'Data Type' is 'NUMBER'. Other settings include 'Value Protected' (checked), 'Sequence' (10), 'Region' (EMP Form), 'Format Mask' (empty), 'Warn on Unsaved Changes' (Page Default), 'Query Only' (unchecked), 'Primary Key' (checked), and 'Maintain Session State' (Per Request (Memory Only)).

Section	Property	Value
Identification	Name	P3_EMPNO
	Type	Hidden
Settings	Value Protected	Checked
	Sequence	10
Layout	Region	EMP Form
	Format Mask	
Advanced	Warn on Unsaved Changes	Page Default
	Query Only	Unchecked
Source	Form Region	EMP Form
	Column	EMPNO
	Data Type	NUMBER
	Primary Key	Checked
Source	Maintain Session State	Per Request (Memory Only)

4. 按鈕行為設定(Button Behavior)

在表單中，我們可以設定按鈕的行為，以執行頁面提交與頁面轉向。

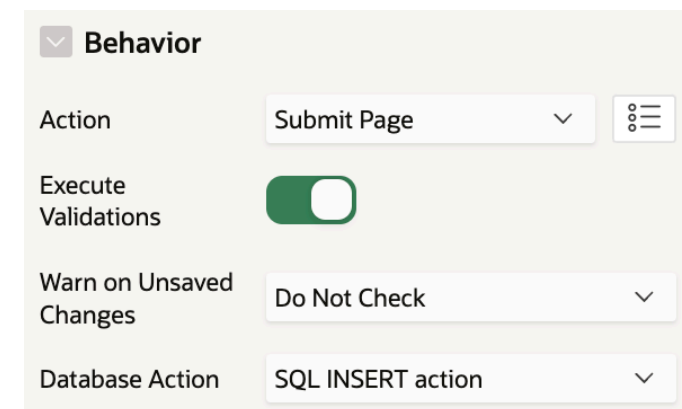
我們來看看 Page Wizard 加入的按鈕的行為設定。

CREATE 按鈕的行為

按下後，應提交表單，執行 INSERT 語句，新增一筆資料。

他的行為相關屬性設定如下：

- Action: Submit Page
- Execute Validations(執行欄位驗證): On/Off
- Database Action: SQL INSERT (產生 INSERT 語句)
 - 指示 Form - Automatic Row Processing (DML)
產生 INSERT 語句



The image shows a configuration panel titled 'Behavior' with a dropdown arrow on the left. It contains four settings:

- Action:** A dropdown menu showing 'Submit Page' with a downward arrow and a menu icon to its right.
- Execute Validations:** A toggle switch that is currently turned on (green).
- Warn on Unsaved Changes:** A dropdown menu showing 'Do Not Check' with a downward arrow.
- Database Action:** A dropdown menu showing 'SQL INSERT action' with a downward arrow.

SAVE 按鈕的行為

按下後，應提交表單，執行 UPDATE 語句，更新一筆資料。

他的行為相關屬性設定大部分與 CREATE 按鈕相同，只有 Database Action 不同。

- Database Action: SQL UPDATE (產生 UPDATE 語句)

▼ Behavior

Action

Submit Page

▼

☰

Execute
Validations

Warn on Unsaved
Changes

Do Not Check

▼

Database Action

SQL UPDATE action

▼

DELETE 按鈕的行為

按下後，應提交表單，執行 DELETE 語句，刪除一筆資料。

刪除前, 會先執行確認對話框，詢問是否確定刪除。對話框是透過 JavaScript 實現的。

所以，在按下 DELETE 按鈕後的執行程序:

1. 轉向頁面去執行 JavaScript 程式碼，顯示確認對話框。
2. 如果使用者確認刪除，則提交表單，指示 Form - Automatic Row Processing (DML) 產生 DELETE 語句。

DELETE 按鈕的行為相關屬性設定如下:4-1 Part 2: 表單結構分析

- Action: Redirect to URL
- Target: JavaScript function
- 執行的 JS code :

```
javascript:apex.confirm(htmldb_delete_message, 'DELETE');
```

上述中

- `apex.confirm()` 是 Oracle APEX JavaScript API 提供的函數，用來顯示確認對話框。
- `htmldb_delete_message` 是一個 JS 變數，用來定義對話框的訊息. 也可以直接寫入訊息字串。
- `DELETE` 是設定給 REQUEST 參數的值，用來指示 Form - Automatic Row Processing (DML) 產生 DELETE 語句。

The screenshot shows the 'Behavior' configuration panel for a button in Oracle APEX. It includes the following settings:

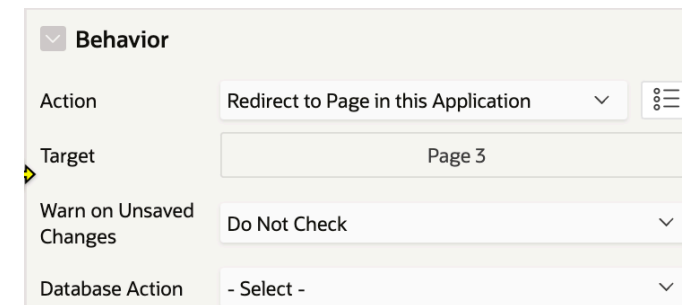
- Action:** Redirect to URL (selected from a dropdown menu).
- Target:** javascript:apex.confirm(htmldb_delete_message,'DELETE');
- Execute Validations:** Disabled (toggle switch).
- Warn on Unsaved Changes:** Do Not Check (selected from a dropdown menu).
- Database Action:** SQL DELETE action (selected from a dropdown menu).

CANCEL 按鈕的行為

按下後，應轉向到另一個頁面，並沒有提交表單到 Server。

他的行為相關屬性設定如下：

- Action: Redirect to URL
- Target: Page number



Behavior	
Action	Redirect to Page in this Application
Target	Page 3
Warn on Unsaved Changes	Do Not Check
Database Action	- Select -

5. 結語

- 瞭解表單的結構與執行過程，以便能夠手動建立表單。
- 表單顯示過程包括三個階段：顯示前、顯示中(rendering)、顯示後。
- 可設定元件的 Server-side 條件，依據條件來決定元件是否顯示。
- 表單接受頁面過程包括四個階段：提交後、驗證、處理、處理後。
- Form Region 與資料表之間的關係; 表單元件(item)與資料表欄位(table column)之間的關係
- 按鈕行為設定(Button Behavior)