

pureQuery：IBM 最新的 Java 資料庫應用程式撰寫參照範例 節省時間、減少程式碼、隨時可以執行

級別：中級

[Azadeh Ahadian \(azadeh@us.ibm.com\)](mailto:azadeh@us.ibm.com)，IBM Eclipse 應用程式開發工具

2007 年 8 月 16 日

更新日期：2007 年 12 月 14 日

pureQuery 專案為資料庫應用程式開發人員，提供以 GUI 為基礎的簡易工具，可大幅提升設計和實作階段的生產力。pureQuery 所提供的 Java™ 資料存取平台，可簡化資料存取應用程式相關的開發和管理工作。此平台是以 Java 資料存取物件為基礎，其中包括 Java 持續性，以及所有資料 CRUD 作業。您可以在 Java 的自然物件導向 (OO) 程式設計參照範例中，善加利用這些物件，以撰寫商業邏輯和基礎程式碼。

前言

關聯式資料存取的物件化

大部分業務都建立於資訊管理系統之上，資料管理系統的基礎架構，是以資料為根本，儲存於一或多個關聯式資料庫當中。這些系統由軟體程式組成，軟體程式的唯一用途為資訊處理，使用 Java 等管理語言實作。

Java 是一種物件導向式語言，可輕鬆用於封裝、繼承和多型性。您可用 Java 實作資訊處理程式的運算和邏輯觀點，也可以用 Java 存取和操作關聯式資料。Java 就物件定義資訊，而關聯式資料庫則使用資料列。物件是其所屬類別的唯一實例，因此實際上，物件彼此間相互獨立。同樣地，關聯式資料庫的資料列也是唯一的，只是資料列需要透過主索引鍵值識別。另一方面而言，只要有一或多個其他物件參照某個物件，該物件就會保持獨立，繼續存在。不過，資料列是以表格元素的形式存在，而且遭到移除後，就會消失。

因此，雖然物件和資料列在許多方面很相似，但是在透過物件化的層次使關聯資料列抽象化，設計高效率程式碼的維護，即為物件程式模組化內在價值的基礎。我們將物件化定義為一種程序，物件可將使用 SQL 執行大量非結構化行內資料查詢實務，轉化為物件導向的封裝、繼承和多型性等基本概念。簡言之，在純正的物件導向參照範例中，物件化使用能執行資料庫作業（建立、取代、更新和刪

除)的方法，以建立物件。

要將程式化的關聯式資料存取和操作物件化，最常用的方式是運用特殊的 API，以及可提供一或多個採用字串的 SQL 陳述式。整個關聯式資料處理，就是以這些封裝的 SQL 陳述式為基礎，SQL 陳述式在程式邏輯中扮演重要角色。不過對 Java 語言來說，這些 SQL 陳述式並非透通的，因此無法獲得編譯時期的最佳化以及語意與語法驗證的好處，遑論省略設計時期的特性，例如針對基礎設計和實作程序，節省大量時間的內容輔助。

因此，要提高程式設計師的效率和生產力，其中一個主要挑戰就是將關聯式資料轉換成物件，然後透過自然的物件導向程式設計，在撰寫基礎程式碼時善加利用這些物件。除了要擁有良好的編碼技能，且可將軟體拆解成模組化區段（特別是其資料存取層）之外，還必須熟知如何安排資料存取 API 呼叫的順序，並熟悉基礎資料存取程式設計模型的相關技術（例如 JDBC），以進行這項程序。

程式碼名稱爲 pureQuery 的專案，為資料庫應用程式開發人員，提供以 GUI 為基礎的簡易工具，可大幅提升設計和實作階段的生產力。為達此目標，需透過使用者起始的自動轉換，將關聯式資料轉換成物件，以供資料存取和操作使用。然後，您即可在自然物件導向程式設計參照範例中，善加利用這些物件，以撰寫商業邏輯和基礎程式碼。pureQuery 的功能整合查詢語言和 Java，有效排除了傳統的 JDBC 程式設計。

本文將概述 pureQuery 的設計時期，可顯示一些設計時期的特性，以及 pureQuery 大幅提升生產力的逐步使用實務範例。

何謂 pureQuery？

pureQuery 是 IBM 以 Java 為主的資料存取平台，可簡化資料存取應用程式的開發和管理。pureQuery 設計時期的特性是提供工具來協助開發人員指定資料庫連線（例如使用 DB2®），讓 Eclipse Java 專案察覺基礎資料庫構件（例如表格、儲存程序等等）。可自動提供設計和執行時期工具，將關聯式資料當作物件管理。所有資料庫構件的連線和自動探索作業完成後，使用者即可透過 GUI，以直觀的方式將任何基礎關聯式表格實體轉換成 Java 物件。您可以透過自動產生相關的 SQL 陳述式，以及自動建立可封裝這些陳述式的母項 Java 物件，來完成前述轉換。產生 Java 物件和內含的 SQL 陳述式之後，您可以在自然的 Eclipse IDE 程式設計環境中，使用完全整合的 SQL 和 Java 編輯功能，進一步加以自訂。

在 Java 編輯器中內建 SQL 的緊密整合，可讓所需的程式設計生產力提升，又能提供完善的設計時期體驗，同時讓使用 SQL 編碼如撰寫 Java 程式碼一般簡單。編輯器整合包括完整支援語意和語法驗證，方法是透過設計時期編譯、內容輔助、語法豐富化，以及在 Java 內部以動態且有選擇性的方式執行 SQL，錯誤會在錯誤窗格中顯示，如同 Java 錯誤。

此外，還緊密整合在設計和執行時期之間的交握協商，您得以在設計時期執行 SQL 陳述式。考量任何主變數和所使用的參數、預覽結果集，並自動回復執行 SQL 陳述式可能導致的任何副作用。

在設計時期對選定 SQL 陳述式進行額外效能分析，可啓動 Visual Explain 執行這項作業，如此可為查詢要求之最佳化工具實作，提供圖形表示法。查詢要求將拆解成個別元件，其中每個唯一元件，都會使用圖示表示。Visual Explain 還會針對查詢最佳化工具所考慮和選擇的資料庫物件，提供相關資訊。Visual Explain 可為查詢實作提供詳細資訊，協助您從整合過的 SQL-Eclipse Java 內部，在設計時期即可瞭解何處耗費最大成本。

pureQuery 物件化程序

pureQuery 系統提供三種不同方式，協助您將關聯式存取層自動轉換成基於物件的模型。

- **資料庫驅動物件化 (DDO)：**
最常見的 pureQuery 使用方式是選取表格，並為資料存取和操作（即 CRUD，create、replace、update、delete）產生物件。pureQuery 會在此選定表格中，產生基於物件導向的類別，並執行 CRUD 作業所需的介面。還會產生一個特殊類別，為所建立的物件進行單元測試。實際上，這是「表格轉換成 Java」的一種程序，其中以 Java 為主的 CRUD 作業，將根據基礎表格的性質而產生。我們將在[快速瀏覽](#)小節中進一步討論。
- **查詢驅動物件化 (QDO)：**
查詢驅動物件化通常指從一組資料庫查詢陳述式（即 SQL），開始建置應用程式的程序。需要建立物件階層時，可使用於建置基礎應用程式的現有查詢陳述式。實際上，這是「SQL 轉換成 Java」的一種程序，其中查詢陳述式將妥善轉換成以 Java 為主的物件。您可以點按以選取查詢陳述式，並經由彈出式選單的直觀 GUI 方式，輕鬆要求 pureQuery 系統產生相關的物件類別階層。基本上，此系統可使用查詢陳述式剖析結果的中繼資料（若有在基礎資料庫連線中維護的中繼資料，亦可使

用)，自動產生所需的物件類別。

- **物件關聯對映 (ORM)：**

有時雖有用來代表關聯式表格的 Bean 類別，但 Bean 類別和屬意表格之間，並未實際鏈結。pureQuery 提供直觀機制，您可透過豐富的內容輔助基礎架構，為類別名稱和位於基礎資料庫的給定表格名稱，執行一對一對映，也可以為類別的受保護（或公開）變數與該表格的直欄執行對映。實際上，這就是「Java 轉換成 SQL」的程序，其中以 Java 為主的實體，將完善轉換成基於 SQL 的 CRUD 作業。

另外有些時候，存在一個內含所有必要關聯對映中繼資料的給定 Bean 類別，但所對映的基礎實體表格卻遺失了。此時，您可以使用 pureQuery 提供的直觀式 UI 工具，從 Bean 類別衍生和產生所需的 DDL 陳述式，以便建立必要的基礎資料庫構件（即表格和直欄）。

快速瀏覽

以下小節將檢視在 [pureQuery 物件化程序](#) 小節中所述的 pureQuery 常見用法，即 DDO、QDO 和 ORM 程序。最後還會簡介 pureQuery 和 Eclipse Java Editor 之間的緊密整合，另外也會說明提供的附加價值功能，包括在設計和執行之期間，用以執行 SQL 陳述式測試的交握協商。

進行以下實務練習之前，請先確定已安裝下列項目：

- DB2 for Linux®、UNIX® 和 Windows® 8.0 或更新版本，以及安裝項目隨附的 SAMPLE 資料庫。
- Developer Workbench 9.5 版
- pureQuery 外掛程式 1.0 版

連線下載位於 [本文資源](#) 小節的 DB2 和 IBM Viper Developer 9.5 公開測試版（Developer Workbench 和 pureQuery 外掛程式的原始檔）。

資料庫驅動物件化 (DDO) 實務範例

在此章節中，我們將檢視簡單的員工資料範例情節，目標是在物件導向參照範例中，以程式化的方式存取和操作員工資料。

下列步驟中，您將建立可察覺 pureQuery 的 Java 專案，Java 專案會將 DB2 SAMPLE 資料庫中的 Employee 表格轉換成物件模型，提供任何 Java 應用程式

式中的自然物件導向方法使用。

這種使用 pureQuery 的資料庫應用程式開發，其設計模式包含下列基本步驟：

- 建立 Eclipse Java 專案
- 將 Java 專案附加至資料庫連線
- 自動產生 pureQuery 程式碼，將基礎連線中的任何給定資料庫表格構件轉換成 Java 物件
- 透過直觀的編輯、執行測試和除錯，進一步自訂

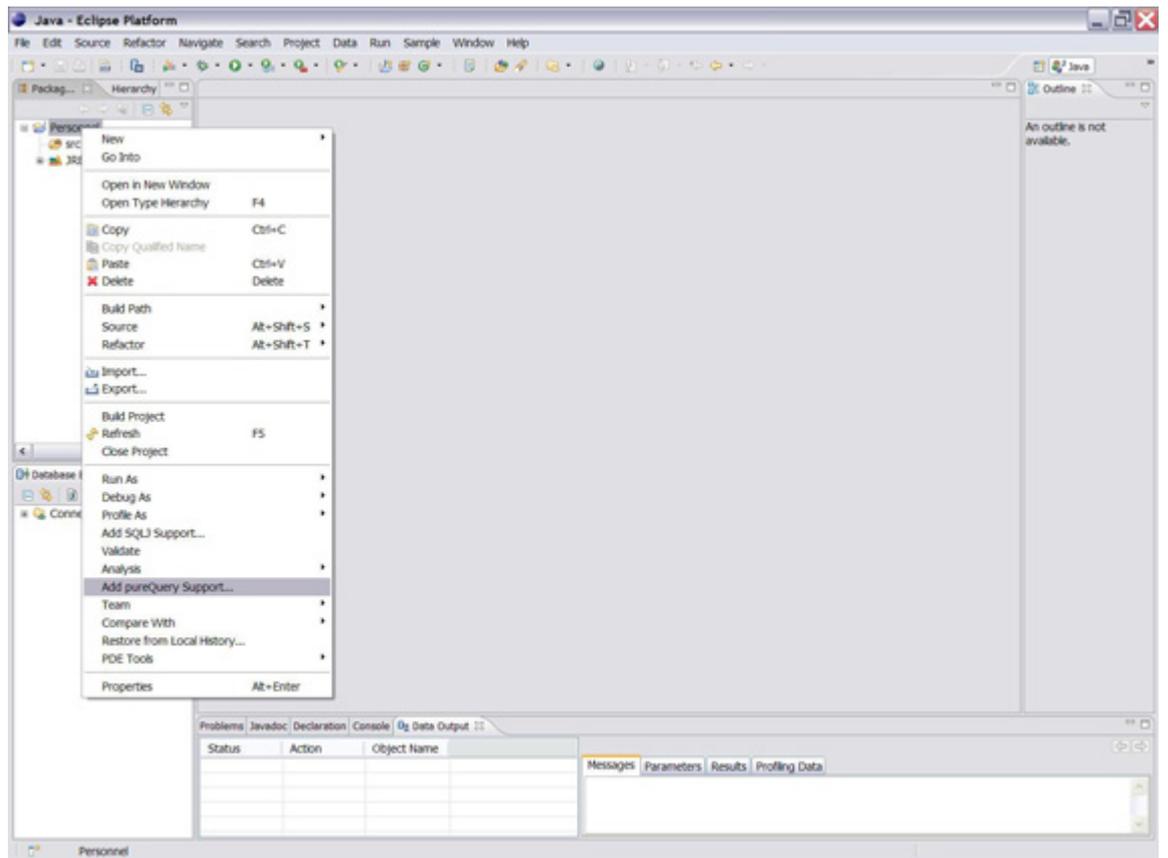
步驟 1：在 Eclipse IDE 中建立 Java 專案

建立 Java 專案時，請確定已建立來源和 Bin 資料夾。將專案命名為 "Personnel"。

步驟 2：確定您的 pureQuery 已啓用 Personnel Java 專案

1. 在 **Personnel Java** 專案上按一下滑鼠右鍵，並從功能表選取 **Add pureQuery Support**。

圖 1. 為 pureQuery 啓用 Eclipse Java 專案

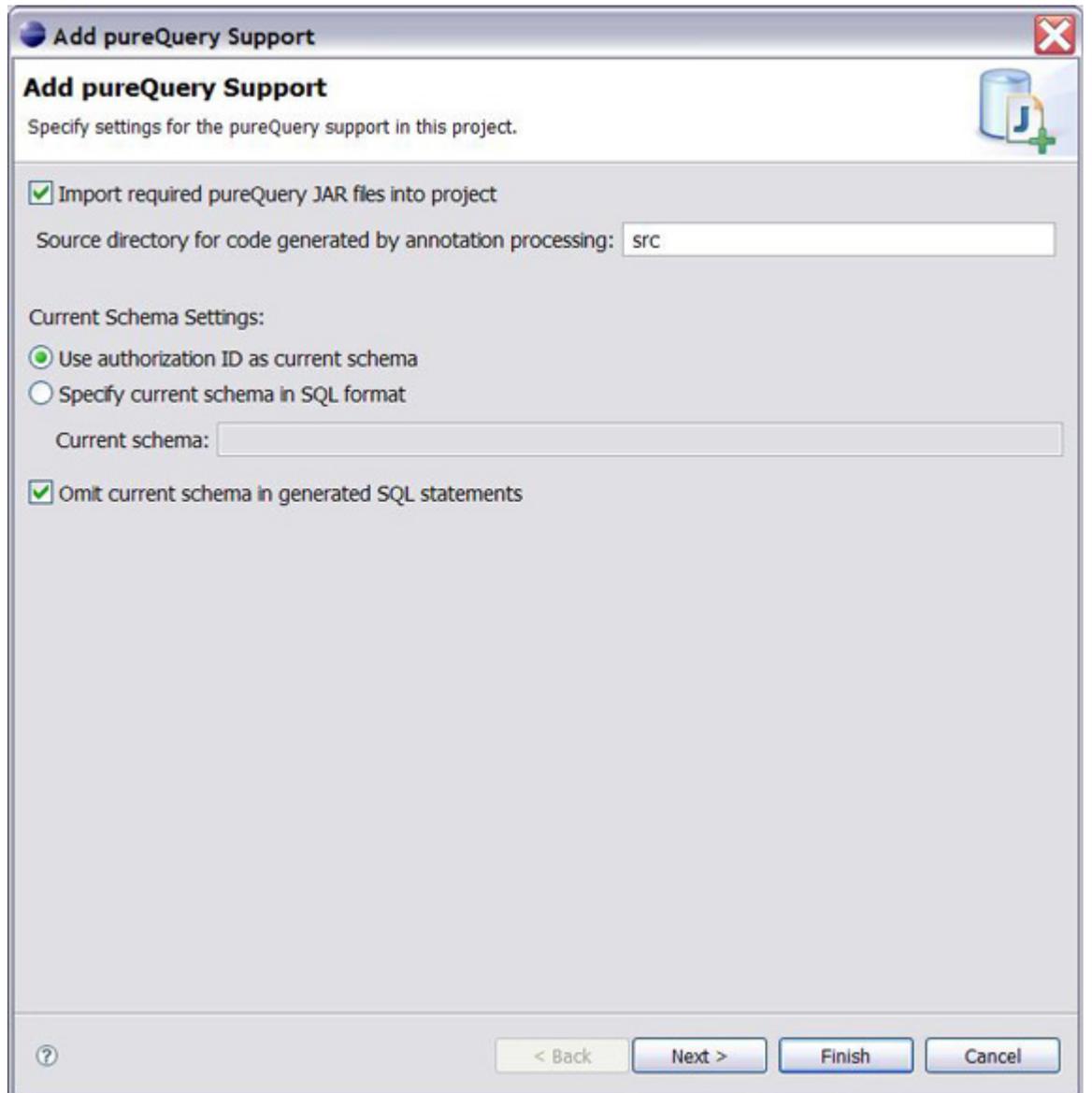


要查看較大的圖檔，請按一下[這裡](#)。

如此會啟動 **Add pureQuery Support** 精靈，可用來為 **pureQuery** 啟用 **Java** 專案。

2. 精靈的首頁提供一般選項，您可以保留預設值。最重要的是，要確定已勾選 **Import required pureQuery JAR files into project** 旁邊的勾選框。由於 **pureQuery** 有自己的特定設計和執行時期 **JAR** 檔，因此得納入 **Java** 專案。

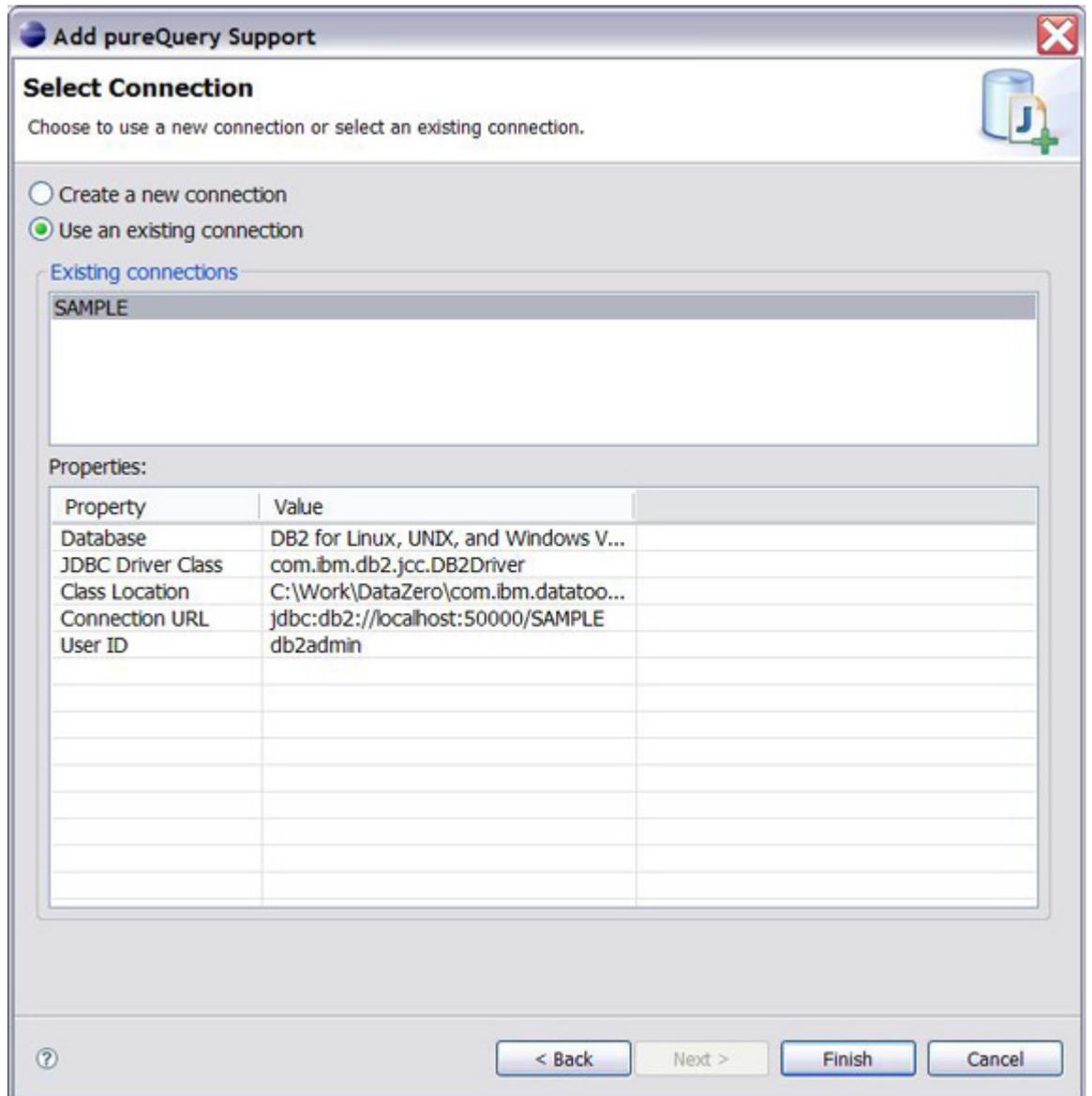
圖 2. **Add pureQuery Support** 精靈：pureQuery 一般選項頁面



按一下 **Next** 跳至 Add pureQuery Support 精靈。

3. 接下來，請建立資料庫連線。此頁面會顯示所有探索到的資料庫連線，您可以從清單中執行選取；如果不存在任何連線，精靈將導引您建立新連線。這類範例請選取 **SAMPLE** 資料庫，其中有 Employee 表格。

圖 3. Add pureQuery Support 精靈：pureQuery 資料庫連線頁面

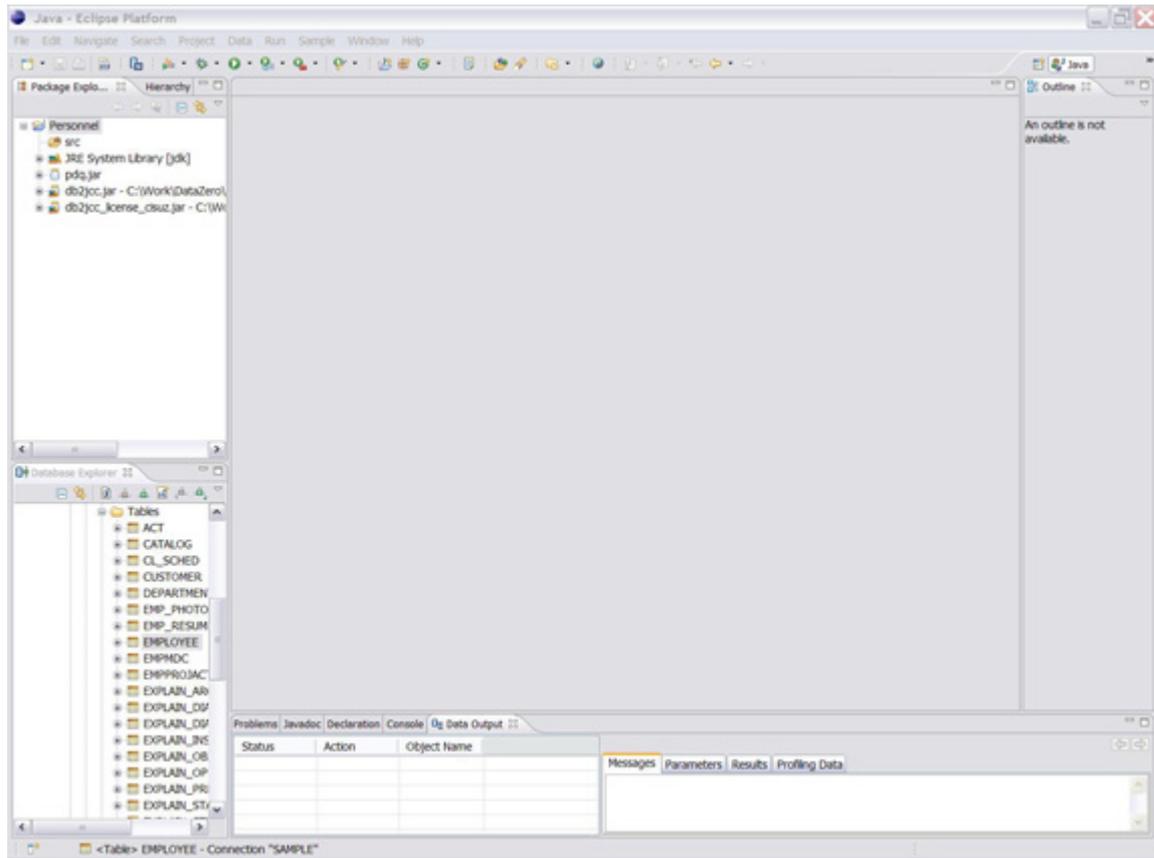


4. 按一下 **Finish** 按鈕以跳出精靈，並回到已啓用 Personnel Java 專案的 pureQuery 中。

步驟 3：將關聯式構件轉換成 Java 物件，並自動產生 pureQuery 程式碼

一旦執行資料庫連線，Java 專案會察覺資料庫，在 IDE 的「資料庫瀏覽器」樹狀結構視圖中，便會自動顯示該特定連線的所有資料庫構件，如圖 4 所示。

圖 4. 資料庫瀏覽器從 SAMPLE 資料庫填入所有構件



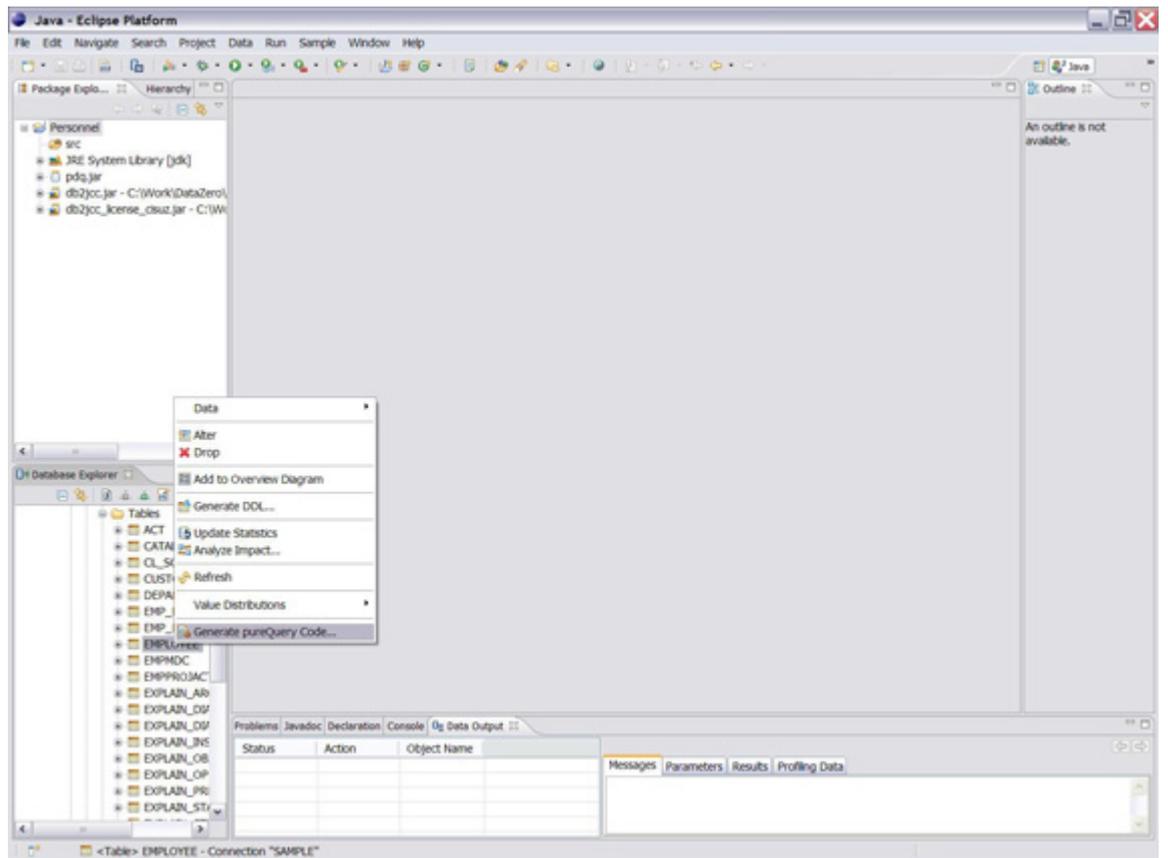
要查看較大的圖檔，請按一下[這裡](#)。

pureQuery 的強大之處，在於能自動產生所需的程式碼，而將任何基礎關聯式資料庫表格構件，轉換成 Java 物件。您只要選取屬意的資料庫構件，並啟動 pureQuery 的 Code Generation 精靈，即可輕鬆完成前述作業。

請遵循步驟，將關聯式構件轉換成 Java 物件，並自動產生 pureQuery 程式碼：

1. 這類範例請從資料庫瀏覽器的樹狀結構視圖選取 **Employee** 表格，並在該節點上按一下滑鼠右鍵，如圖 5 所示。

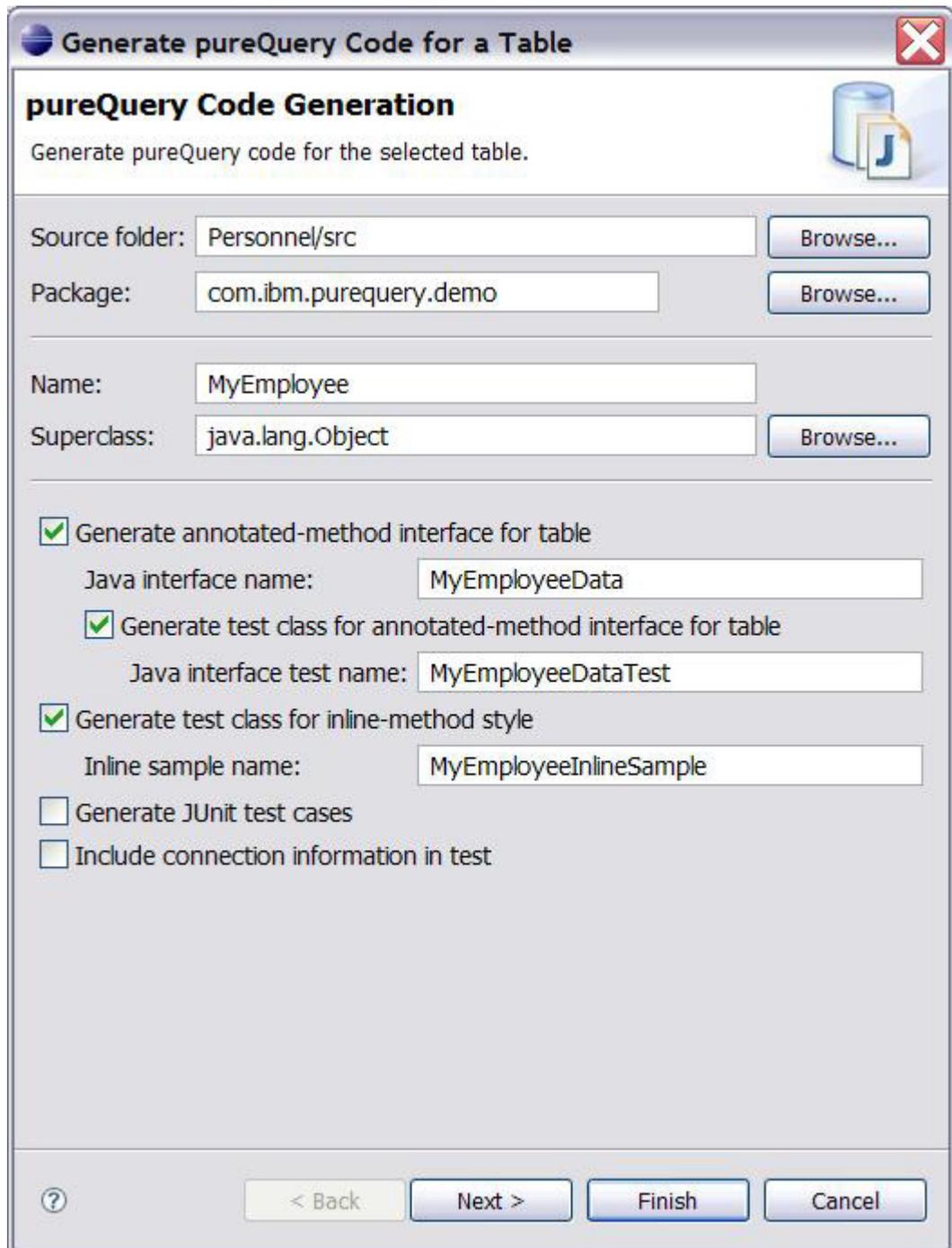
圖 5. 將 Employee 表格轉換成 Java 物件：自動產生 pureQuery 程式碼



要查看較大的圖檔，請按一下[這裡](#)。

2. 從彈出式選單選取 **Generate the pureQuery code** 選項。如此可啓動 Generate pureQuery Code 精靈，如圖 6 所示。

圖 6. pureQuery Code Generation 精靈



3. 指定 Java Bean 的一般選項。

Generate pureQuery Code for a Table 精靈（如 圖_6 所示）首頁所提供的屬性，用來說明代表 Employee 表格的 Bean 類別，您可以保留預設值。在此檢視部分屬性：

- **Source folder**：此欄位指定所有已產生 Java 檔的實際所在位

置，您可以保留預設值：**Personnel/Src**。

- **Name**：此欄位表示已產生的 Java Bean 名稱。依預設，Bean 的名稱將會和「資料庫瀏覽器」中選定資料庫表格構件的名稱相同，pureQuery 程式碼便為此產生（在此為 Employee 表格）。您可以變更名稱，結果最終產生的檔案會出現註釋，指出 Bean 和對應資料庫表格構件之間的對映。

這類範例請變更名稱為 **MyEmployee**，產生的註釋如下所示：

```
@Table(name="EMPLOYEE", schema="DB2ADMIN")
```

- **Generate annotated-method interface for table**：將產生包含資料庫建立/更新/刪除/插入方法的介面檔。基本上，這些方法將促成資料庫資料列的物件化，如先前 1.0 小節所述。建置專案時，也會為此介面產生內含所有必要 SQL 陳述式的預設實作。請注意，您可以選擇性地變更預設介面的名稱。
- **Generate test class for annotated-method interface for table**：將產生特殊類別，供 Bean 介面的單元測試使用。您可以選擇性地變更測試類別的名稱。
- **Generate test class for Inline-method style**：將產生可包含明確行內 SQL 陳述式的檔案，若依預設，將用作介面實作方法。簡化的 API 提供一致的方法，讓您以 SQL 查詢資料庫資料來源。無須變更任何程式設計或使用者體驗，就能配置應用程式靜態執行資料庫 SQL。您可以選擇性地變更測試類別的名稱。
- 請注意，在以上兩種情況中，皆可選取 **Generate JUnit test cases** 勾選框，選擇性地要求系統自動產生一組 JUnit 測試實例。
- **Include Connection information and user name**：此選項會在所有產生的測試類別中，納入連線字串和使用者名稱。

4. 指定如何定義所產生的 Bean 變數。

您可以在精靈的第二頁（如圖 7 所示），選擇性地指定是否要產生公用或受保護欄位，以及是否要包含公用 Accessor 方法。

圖 7. 產生 pureQuery 程式碼以定義 Bean 欄位

Generate pureQuery Code for a Table

Bean Fields

Specify how to define the bean fields.

Select the scope of the bean fields:

Public fields with no accessor or mutator methods

Protected fields with public accessor and mutator methods

Map the table columns to the bean fields:

| Column Name | Column Type | Field Name | Field Type |
|-------------|-------------|--------------|------------|
| EMPNO | CHAR | empno | String |
| FIRSTNME | VARCHAR | firstnme_new | String |
| MIDINIT | CHAR | midinit | String |
| LASTNAME | VARCHAR | lastname_new | String |
| WORKDEPT | CHAR | workdept_new | String |
| PHONENO | CHAR | phoneno | String |
| HIREDATE | DATE | hiredate | Date |
| JOB | CHAR | job | String |
| EDLEVEL | SMALLINT | edlevel | int |
| SEX | CHAR | sex | String |
| BIRTHDATE | DATE | birthdate | Date |
| SALARY | DECIMAL | salary | BigDecimal |
| BONUS | DECIMAL | bonus | BigDecimal |
| COMM | DECIMAL | comm | BigDecimal |
| | | | |
| | | | |
| | | | |

您也可以選擇性地變更 Bean 的變數名稱（依預設，其名稱應設為和 Employee 表格直欄名稱相同）。請注意，每個變數名稱都產生一個特殊註釋，變數名稱和個別對映直欄的名稱有所不同。註釋的形式如下：

```
@Column(name="name of the table column")
```

且出現在變數宣告之前。舉例而言，在此特定案例中，將 Bean 的欄位名稱由 **firstnme** 變更為 **firstnme_new**。在 Java 檔中產生且前置於宣告的註釋如下：

```
@Column(name="FIRSTNAME") protected String firstName_new
```

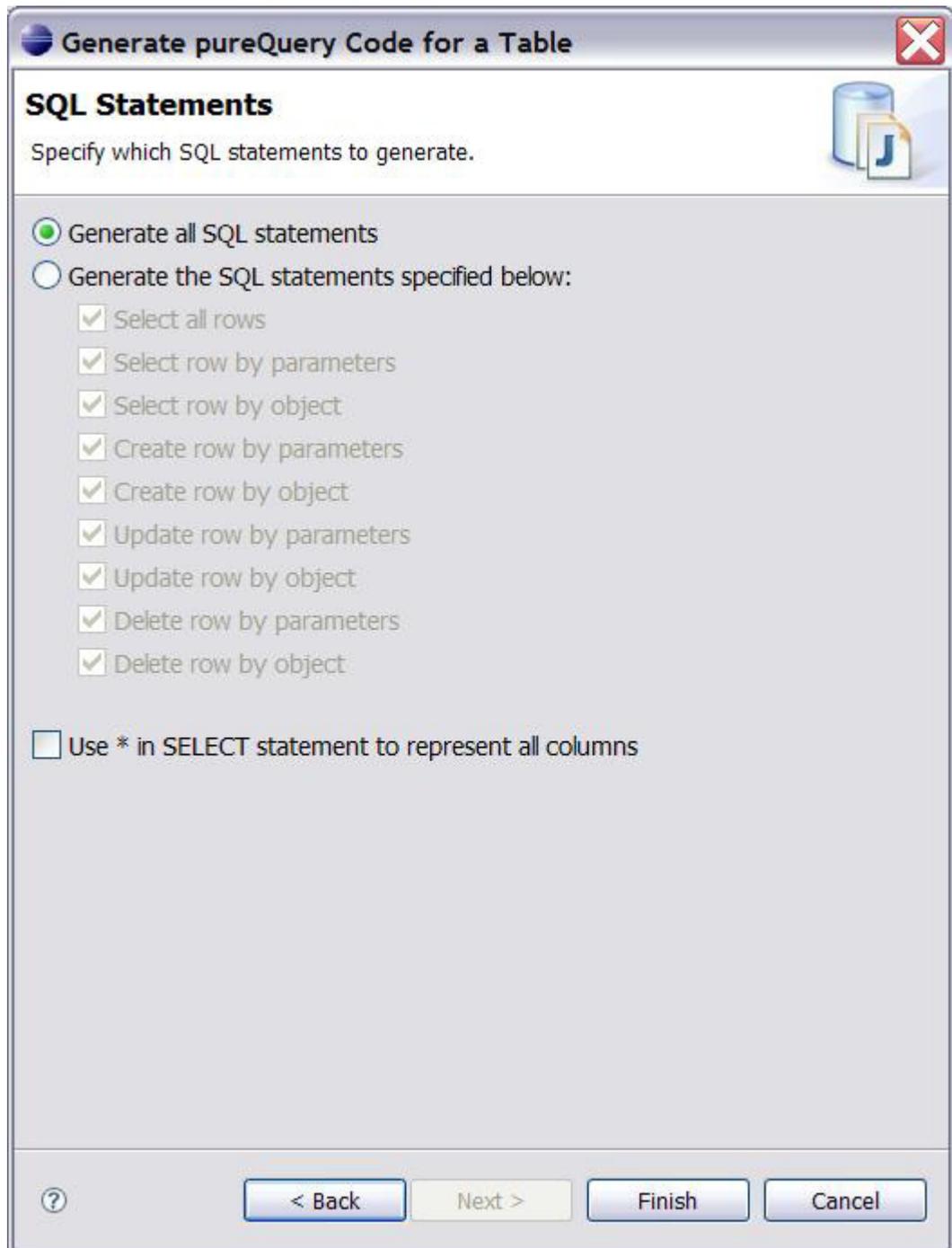
凡是和表格直欄名稱不同的變數名稱，都會出現相同狀況(參閱圖 9)。

這類範例請確定已勾選 **Protected Fields with Public accessor methods**。您也可以選擇性地變更變數類型名稱，使其不同於對映直欄資料類型名稱。

5. 請指定想要產生的不同 SQL 陳述式。

在精靈的最後一頁(參閱圖 8)，您可以選擇性地變換目標，選取想要產生的不同 SQL 陳述式。這類範例請選取 **Generate all SQL statements**。

圖 8. 產生不同的 SQL 陳述式



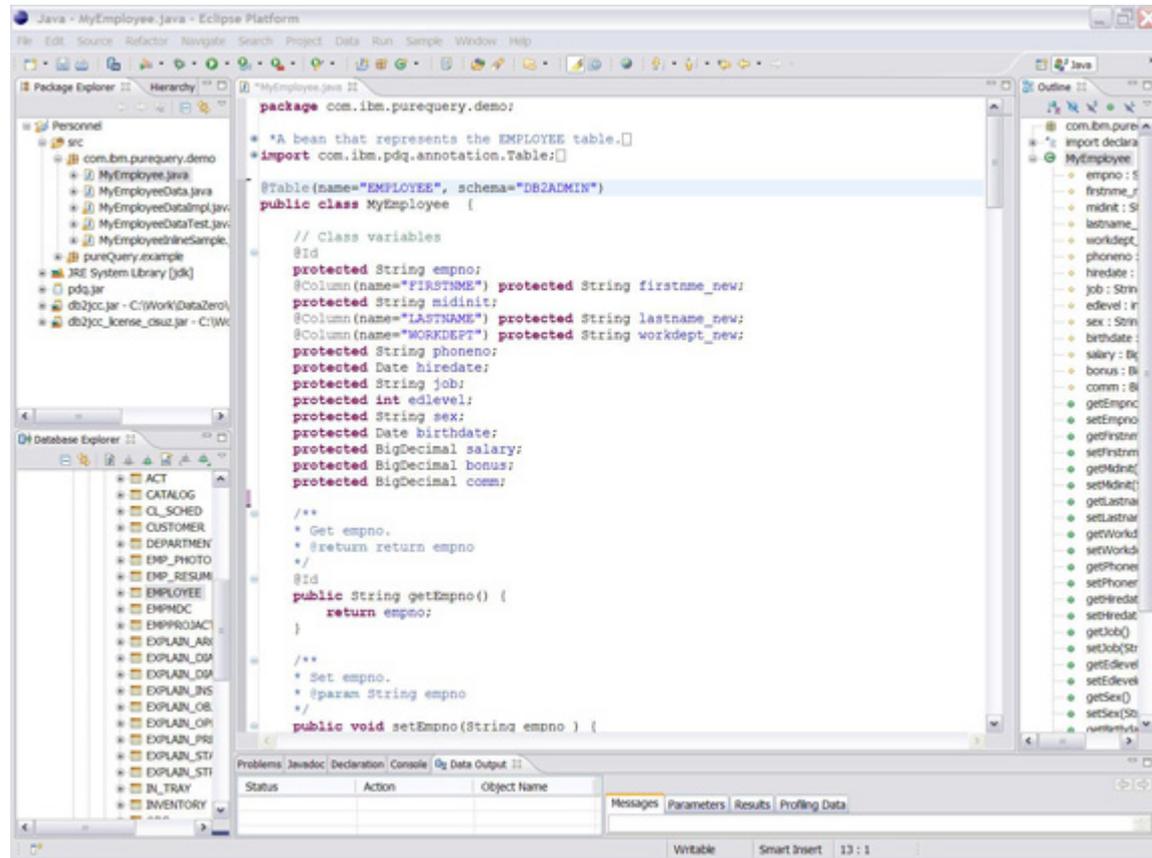
6. 按一下 **Finish**，以產生 pureQuery 程式碼。

按下 **Finish** 按鈕後，精靈除了會產生下列五個主檔，以完成先前在 pureQuery 物件化程序所述的 DDO 程序，還會產生公用程式套件。讓我們檢視產生的構件：

- 構成物件階層的檔案

1. **MyEmployee.java** (參閱圖 9)：此為主要 Bean 類別。這個類別包含變數與個別的 Accessor 方法，以及和 Employee 表格直欄的逐一對映。

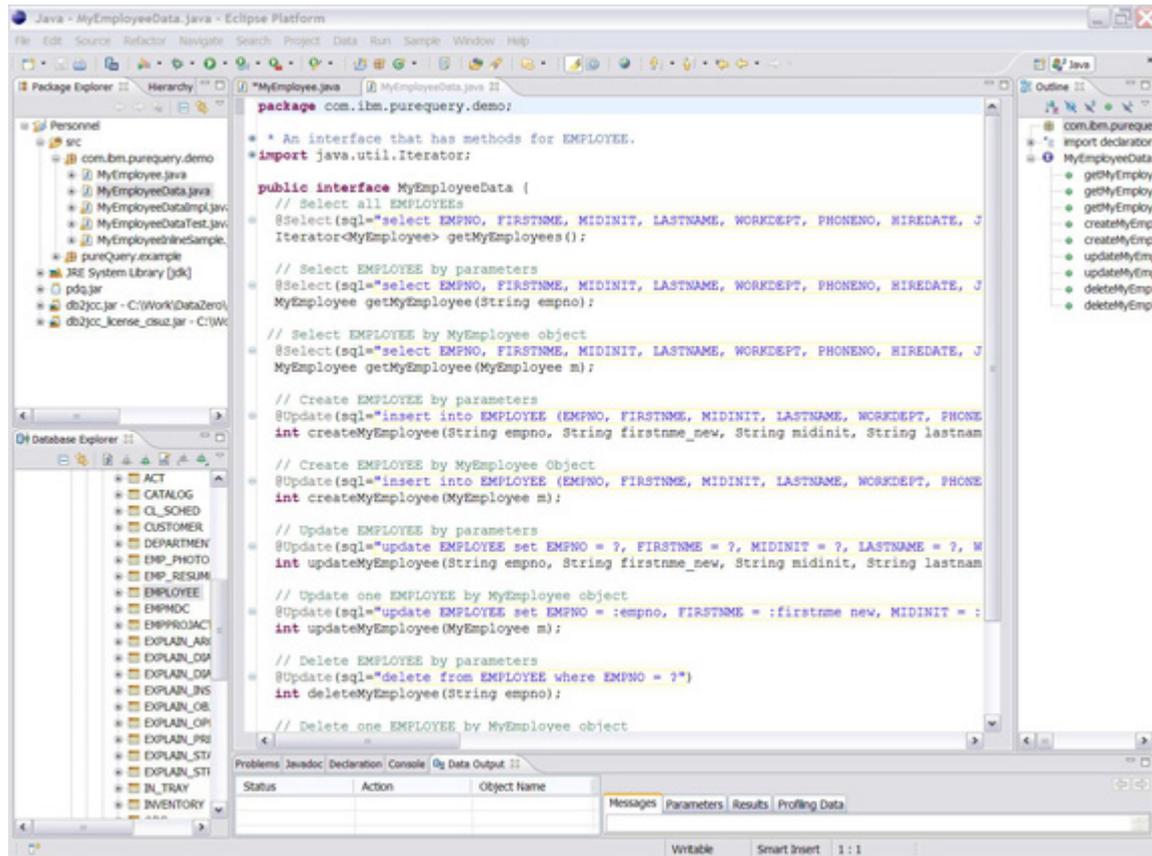
圖 9. 代表 Employee 表格的 Java Bean 類別



要查看較大的圖檔，請按一下[這裡](#)。

2. **MyEmployeeData.java** (參閱圖 10)：此介面檔抽象化資料庫存取層，以進行資料查詢和操作。包含抽象方法，唯一的實作目的，是在基礎表格中執行 CRUD 作業。

圖 10. MyEmployeeData.java 包含 CRUD 作業的介面

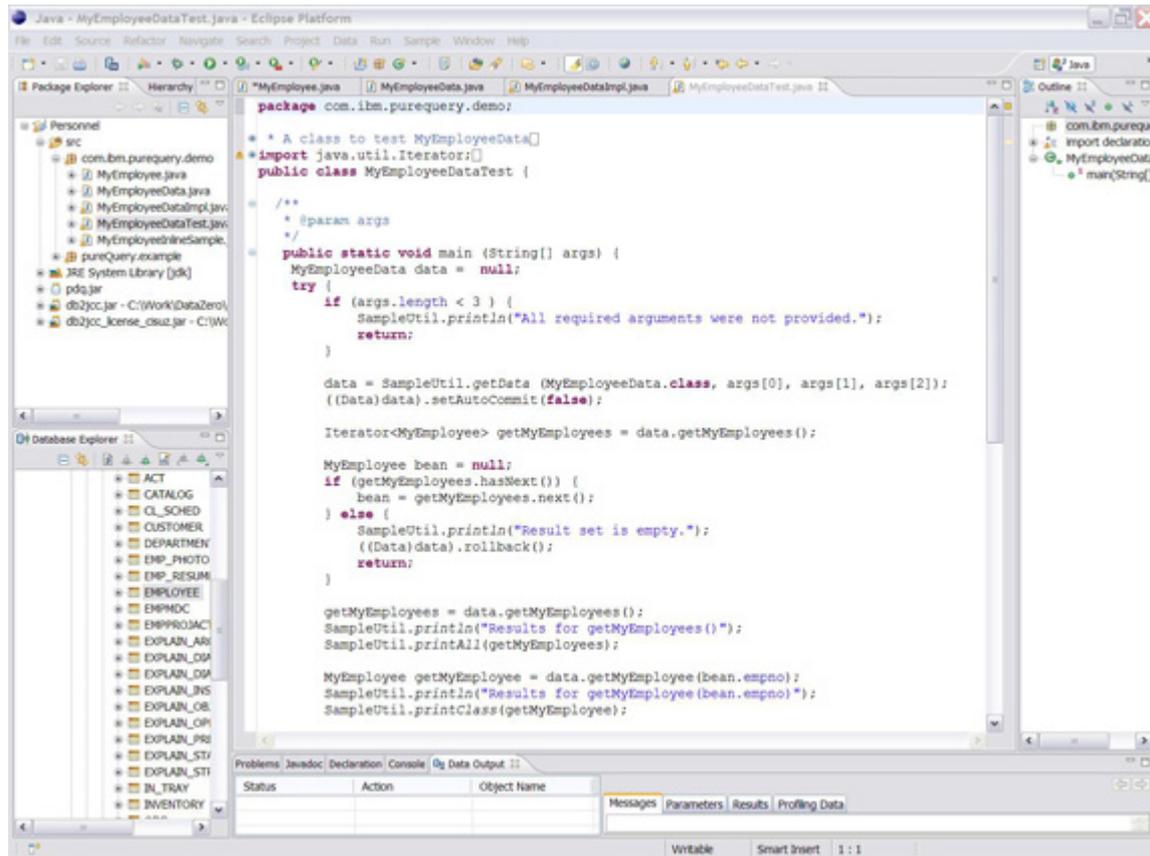


要查看較大的圖檔，請按一下[這裡](#)。

3. **MyEmployeeDataImpl.java** (參閱圖 11)：這是先前提過 MyEmployeeData 介面的預設實作，此實作包含封裝所有資料存取和操作所需之 SQL 陳述式的方法。

請注意：只有完成專案建置時，才會產生此檔案。您可以在 Eclipse IDE 主功能表中設定 Build Automatically option of the Project，或由開發人員手動建置專案，以產生檔案。

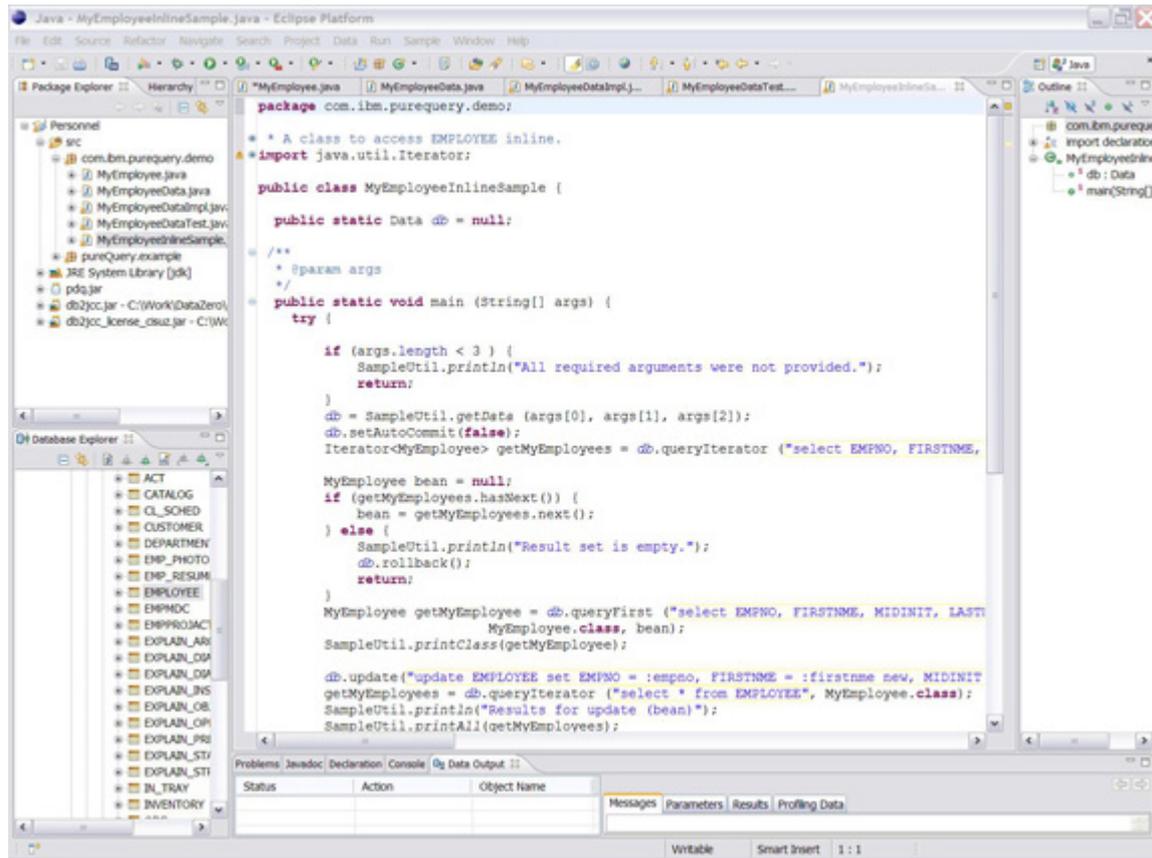
圖 11. 內含介面方法實作的類別



要查看較大的圖檔，請按一下[這裡](#)。

5. **MyEmployeeInLineSample.java** (參閱圖 13): 只有勾選 **Generate Sample Class With Inline method call syntax** 選項，才會產生此檔案。負責執行所有透過 pureQuery 執行時期 API，使用行內樣式方法呼叫所產生的 SQL 陳述式。

圖 13. 自動產生測試類別以執行行內產生的 SQL 陳述式

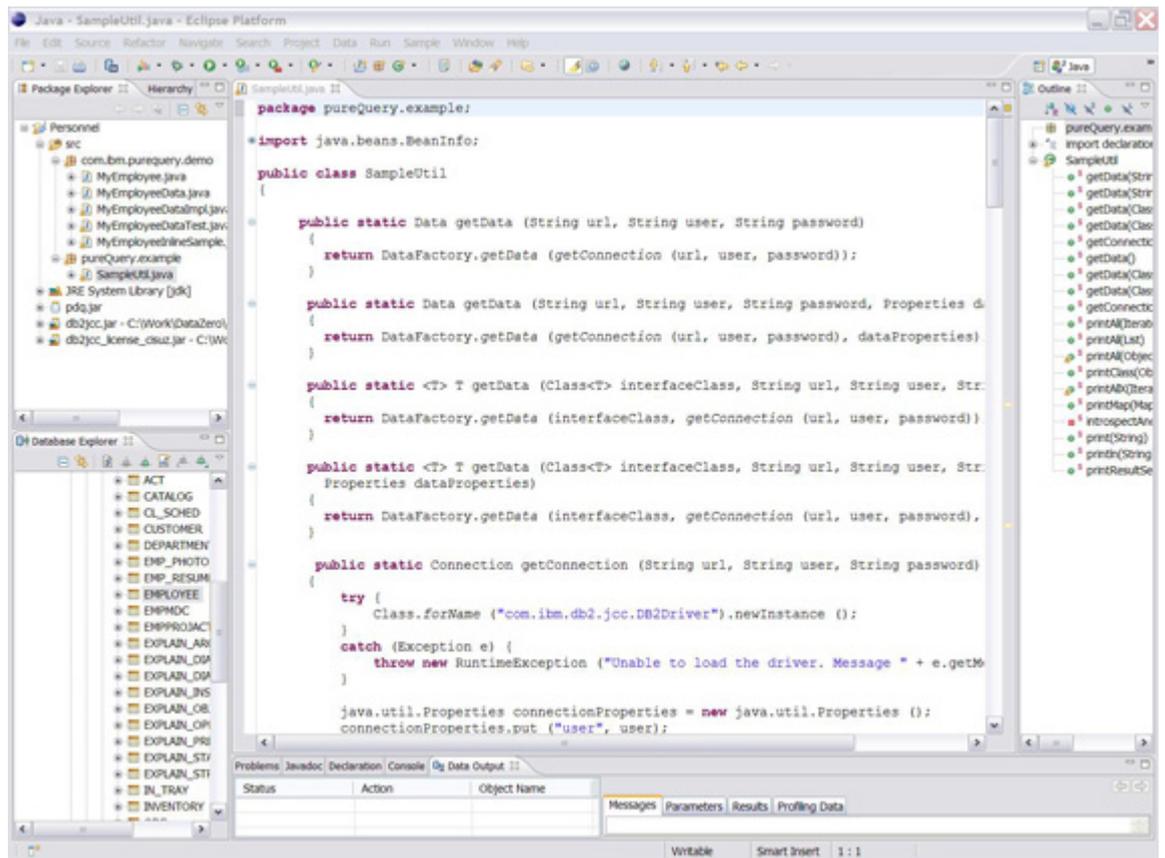


要查看較大的圖檔，請按一下這裡。

- 公用程式套件

除了前幾個小節產生的檔案外，pureQuery 系統還會產生公用程式套件，其中包括名為 **SampleUtil.java**（參閱圖 14）的檔案。這個檔案包含數種通用靜態公用程式函數（例如取得連線、列印等等），用途像是供所產生測試類別（即 **MyEmployeeDataTest.java**）列印輸出結果，以及建立基礎資料庫連線。

圖 14. 自動產生含數種通用靜態公用程式函數的範例類別



要查看較大的圖檔，請按一下[這裡](#)。

查詢驅動物件化 (QDO) 實務範例

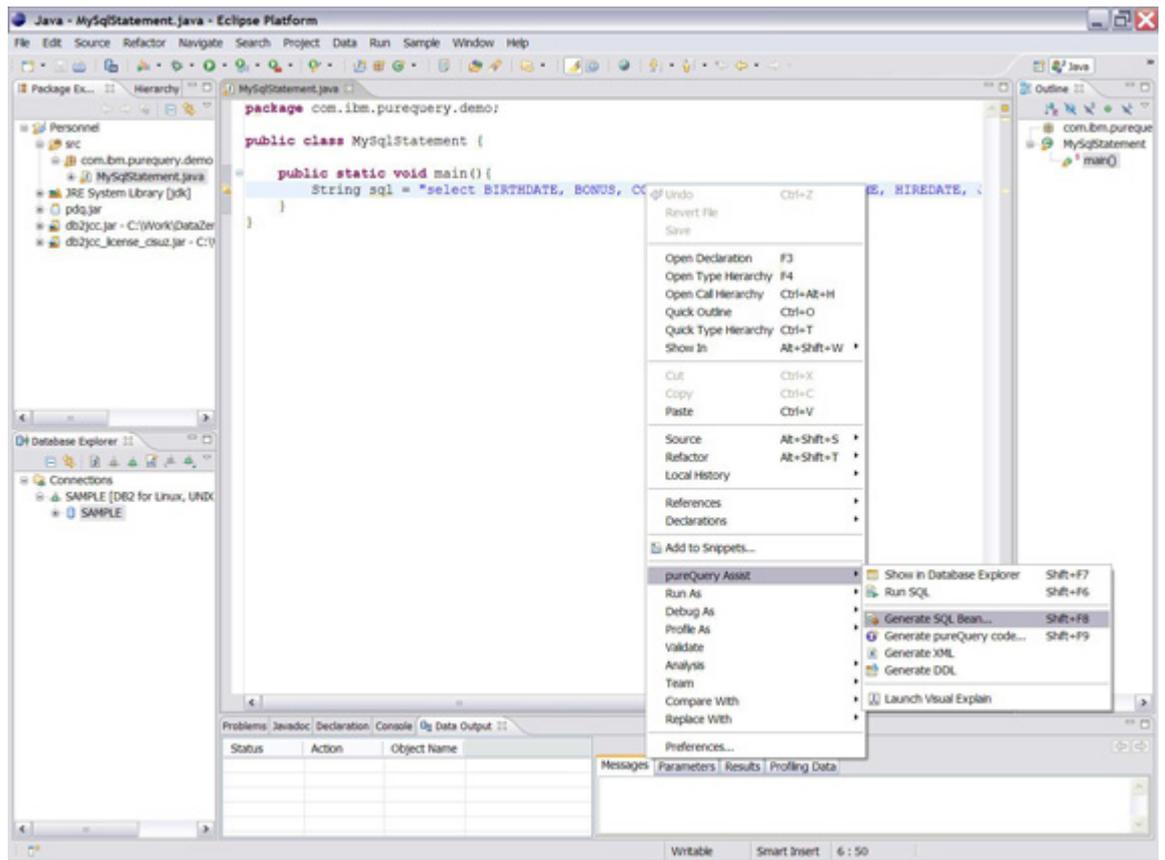
若有一或多種 SQL 陳述式，pureQuery 的直觀式工具可自動產生物件階層，供關聯式資料存取層使用。下列逐步說明展現 pureQuery 的強大之處，可使用給定的 SQL 陳述式建立物件模型。

1. 請開啓先前在 **DDO 實務範例**中建立的 Eclipse Java Personnel 專案，
2. 然後爲此專案新增 Java 檔類別，並在 Main 方法中納入以下 SQL 陳述式：

```
SELECT BIRTHDATE, BONUS, COMM, EDLEVEL, EMPNO, FIRSTNME, HIREDATE, JOB, LASTNAME,
MIDINIT, PHONENO, SALARY, SEX, WORKDEPT from EMPLOYEE
```

3. 在 SQL 陳述式的任意處點按滑鼠右鍵，然後從彈出式選單選取 **pureQuery Assist -> Generate SQL Bean** 選項，即可啓動 **Create Bean for a SQL Result Set** 精靈。

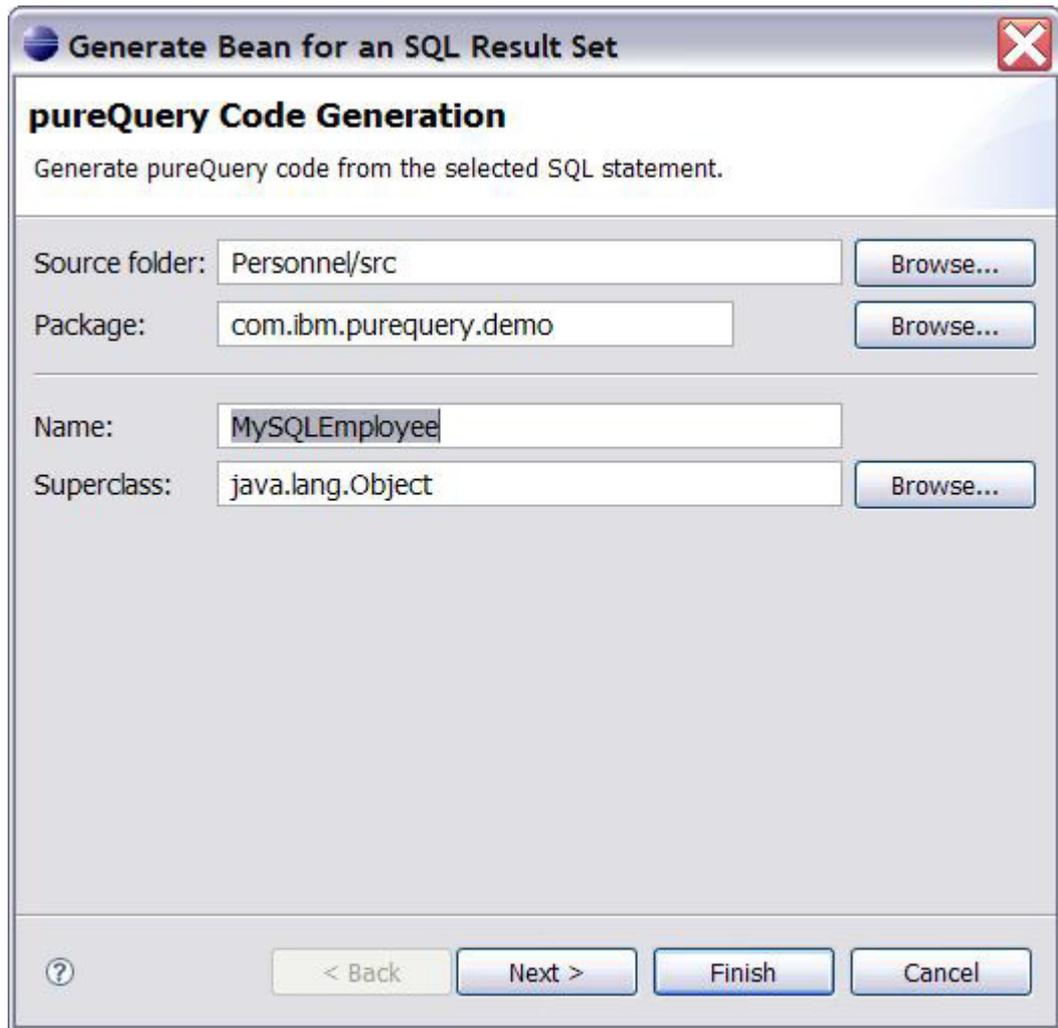
圖 15. 從 SQL 陳述式建立 Bean



要查看較大的圖檔，請按一下[這裡](#)。

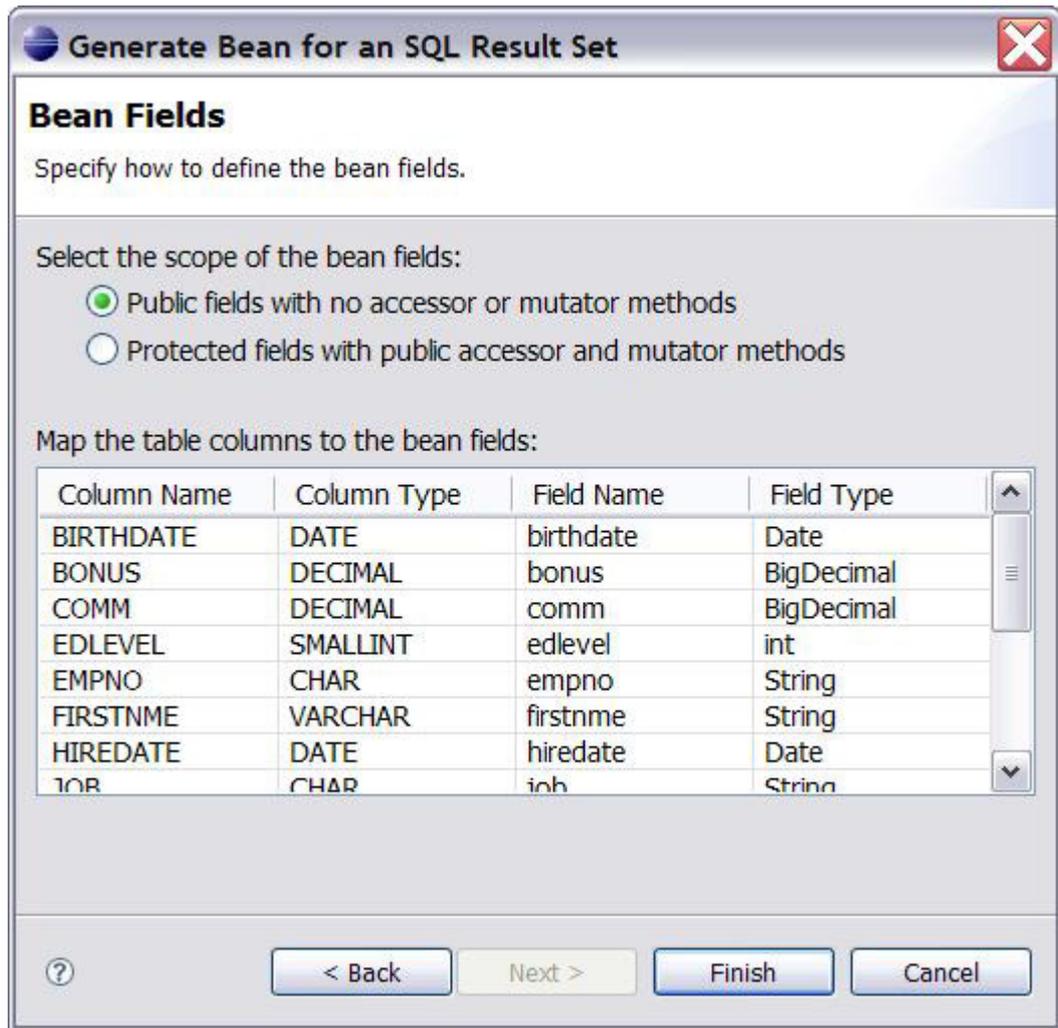
4. 在首頁為要產生的 **Bean** 類別指定名稱，在此情況下，請輸入 "MySQLEmployee"，並按一下 **Next** 按鈕。

圖 16. 指定要從 SQL 陳述式產生的 **Bean** 類別名稱



5. 您可在精靈的第二頁，指定如何定義 **Bean** 欄位，例如保留預設值。您也可以選擇性地變更變數類型名稱，使其不同於對映欄位的資料型態（參閱圖 17）。

圖 17. 指定 Bean 欄位



6. 按一下 **Finish** 按鈕，隨之產生 Bean 檔 `MySQLEmployee.java`。

要建立介面和預設實作檔，您可以使用下一個小節所介紹的 `pureQuery ORM` 程序。

物件關聯對映 (ORM) 實務範例

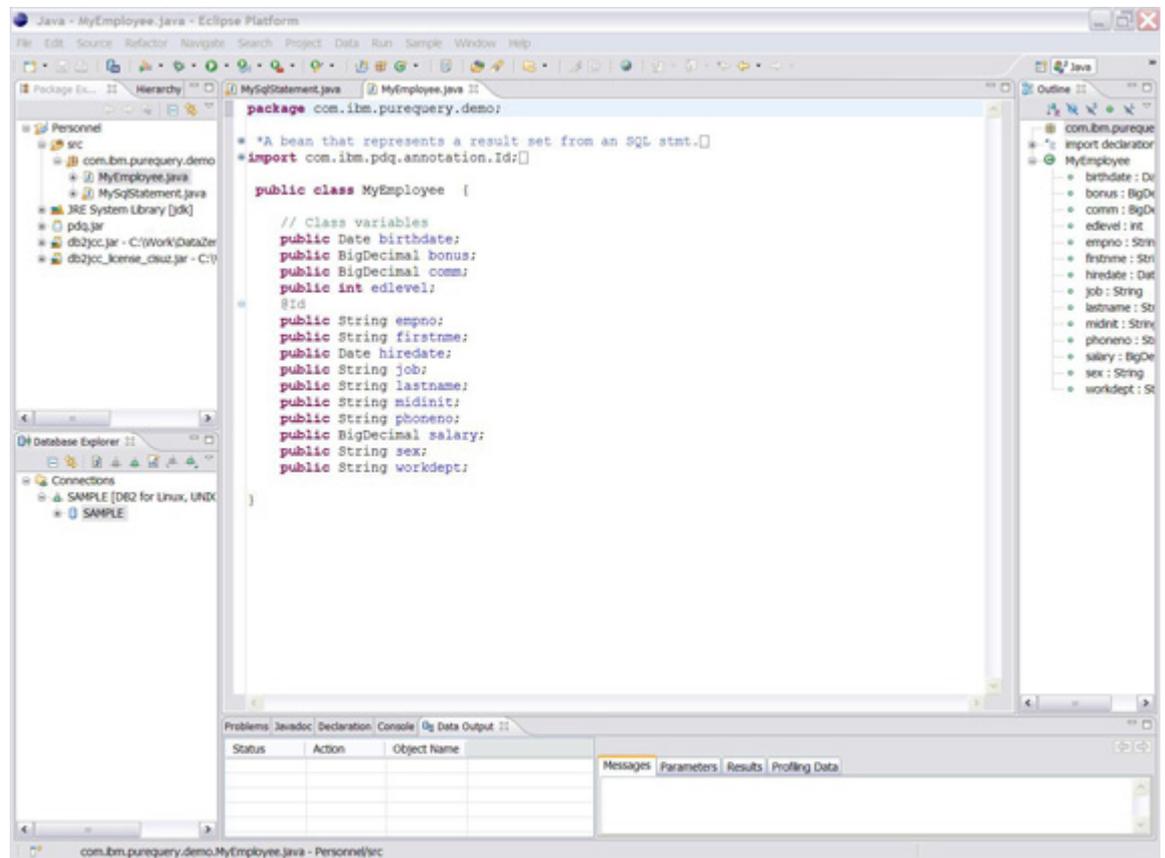
若有給定的 Java Bean 類別，`pureQuery` 所提供的直觀式機制，可在類別成員和基礎資料庫表格中任何給定關聯實體之間，進行對映。

請注意，設計時期對映必須根據表格綱目和遊戲規則進行。舉例而言，即使在設計時期，資料類型為 `Double` 的變數可以對映至類型為 `XML` 表格直欄，執行時期的結果仍不可預料，因此可能出現錯誤。下列逐步範例，說明這種對映。

1. 請開啓在 DDO 範例 中建立的相同 `Eclipse Java Personnel` 專案。

2. 要說明 ORM 特性，您可以使用現有的 Bean，或在資料庫中代表表格的 Java 使用者定義類別。此範例使用現有的 Bean，因此請刪除 MyEmployee.Java 以外的所有 Java 產生檔。

圖 18. 對映 Bean 與關聯式表格



要查看較大的圖檔，請按一下[這裡](#)。

3. 在編輯器中開啓 MyEmployee.Java 類別，類別名稱的頂端新增以下註釋：

```
@Table (name = "EMPLOYEE", Schema = "DB2ADMIN")
```

這個註釋將在類別名稱和基礎資料庫的表格之間，執行實際的對映。

4. 請在要進行對映的每一個欄位/變數的頂端，新增以下註釋。在此範例中，我們僅對映 BIRTHDAY 和 FIRSTNAME 欄位，因此請在 BIRTHDAY 變數名稱頂端新增，並

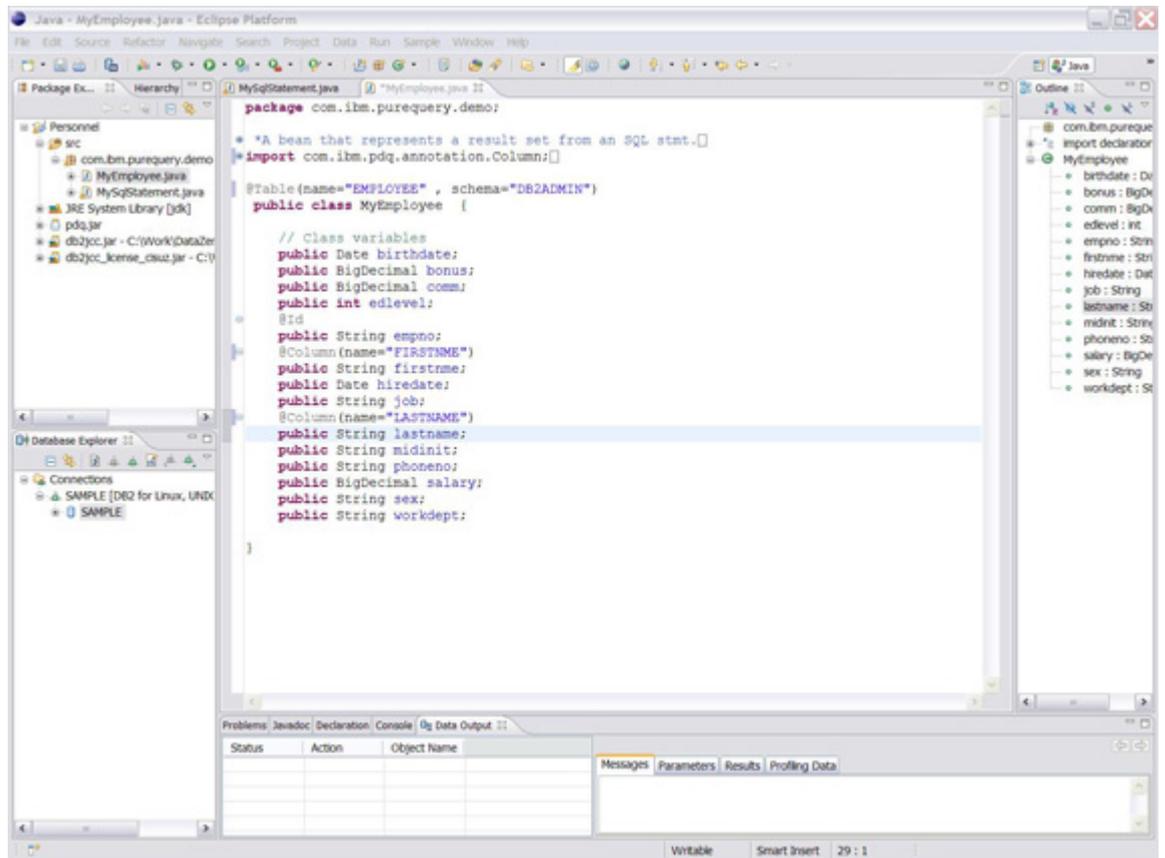
```
@Column (name = "BIRTHDAY")
```

在 FIRSTNAME 變數名稱頂端新增

```
@Column (name = "FIRSTNAME")
```

(參閱圖 19)。

圖 19. 註釋在主變數和表格欄位之間建立對映



要查看較大的圖檔，請按一下[這裡](#)。

5. 現在請進入編輯器，在 MyEmployee 類別內部任意處點按滑鼠右鍵，然後從彈出式選單選取 **pureQuery Assist -> Generate pureQuery CODE** 選項。如此將啟動先前所述的 Generate pureQuery Code for a Table 精靈，最終產生 pureQuery 相關程式碼（包括介面檔案、測試類別、實作程式碼等等）。

請注意，由於 Java Bean 類別已存在，精靈不會再顯示 Bean 欄位給使用者。

pureQuery 的內容輔助並與 Java 編輯器緊密整合

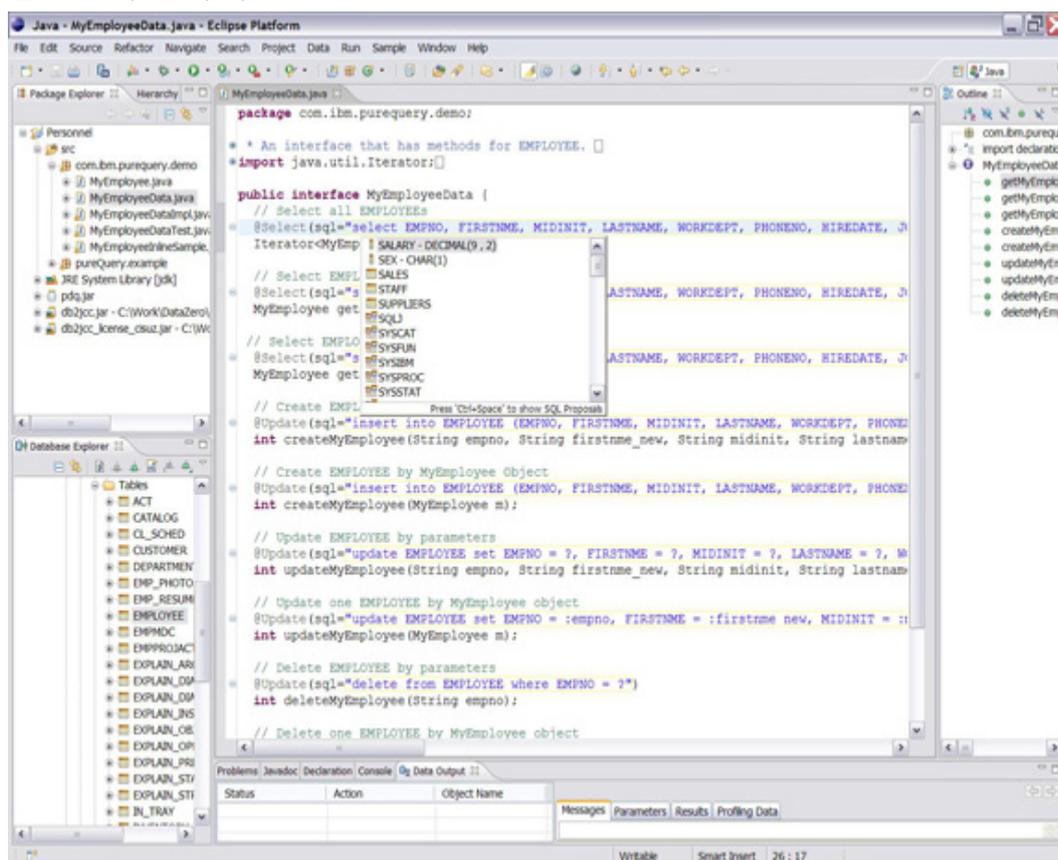
pureQuery 最強大的功能之一，在於可透過單一編輯器，完美整合 SQL 和 Java。如此緊密的整合為 Java 和 SQL 提供一致的設計時期功能集，其中除了

SQL 陳述式的語意驗證外，還包括程式碼輔助、語法驗證和豐富化功能。

程式碼輔助可提供：

- **SQL 陳述式完成** 每當輸入 SQL 關鍵字如 SELECT，或者想要使用運算子時，環境定義視窗就會出現，並顯示完成陳述式幾種可能的方式。

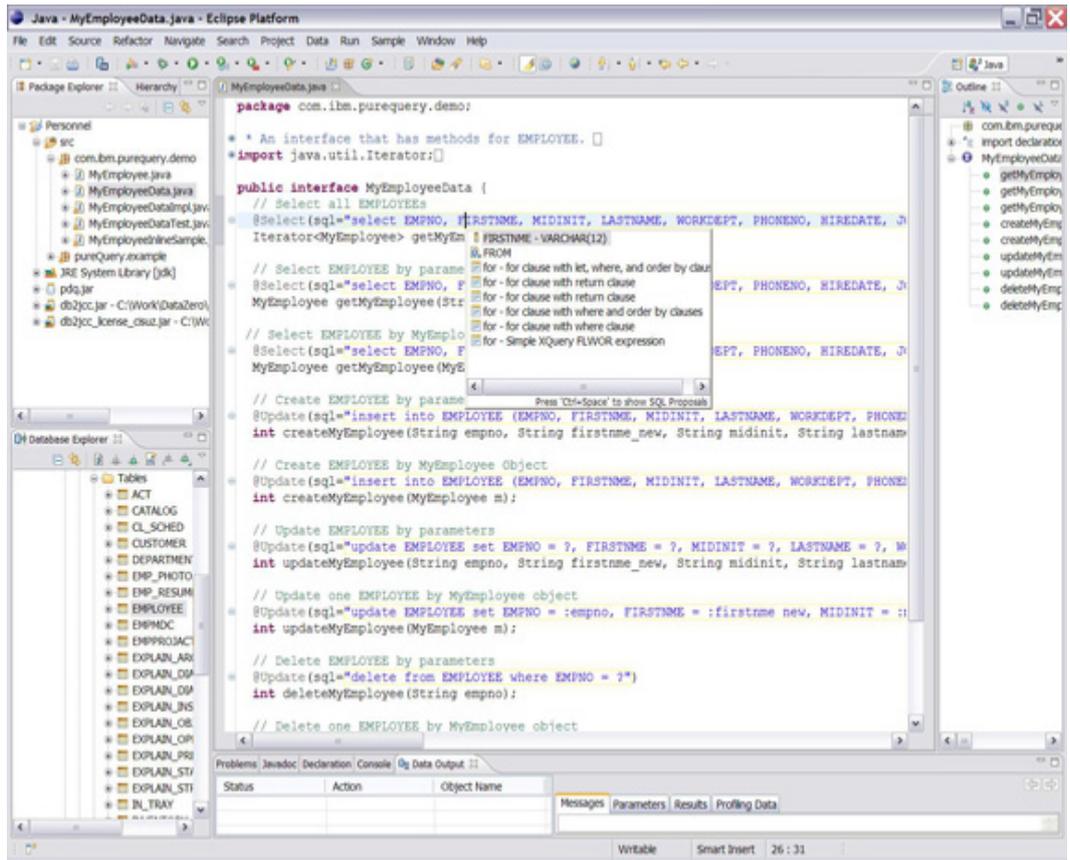
圖 20. 陳述式完成



要查看較大的圖檔，請按一下[這裡](#)。

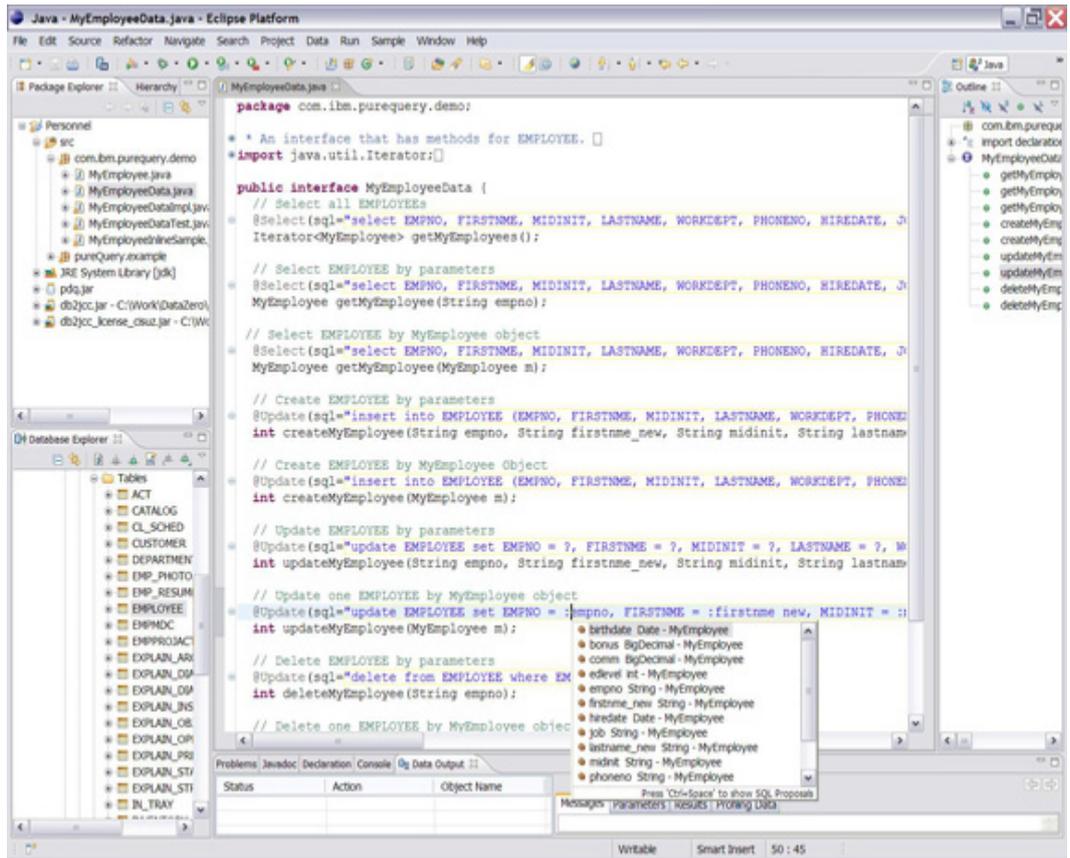
- 內容輔助顯示和給定關聯式構件（如表格、綱目等等）有關的可能實體名稱清單，或者需要在陳述式中指定參數的主變數清單（參閱圖 21 和 22）。

圖 21. 內容輔助



要查看較大的圖檔，請按一下這裡。

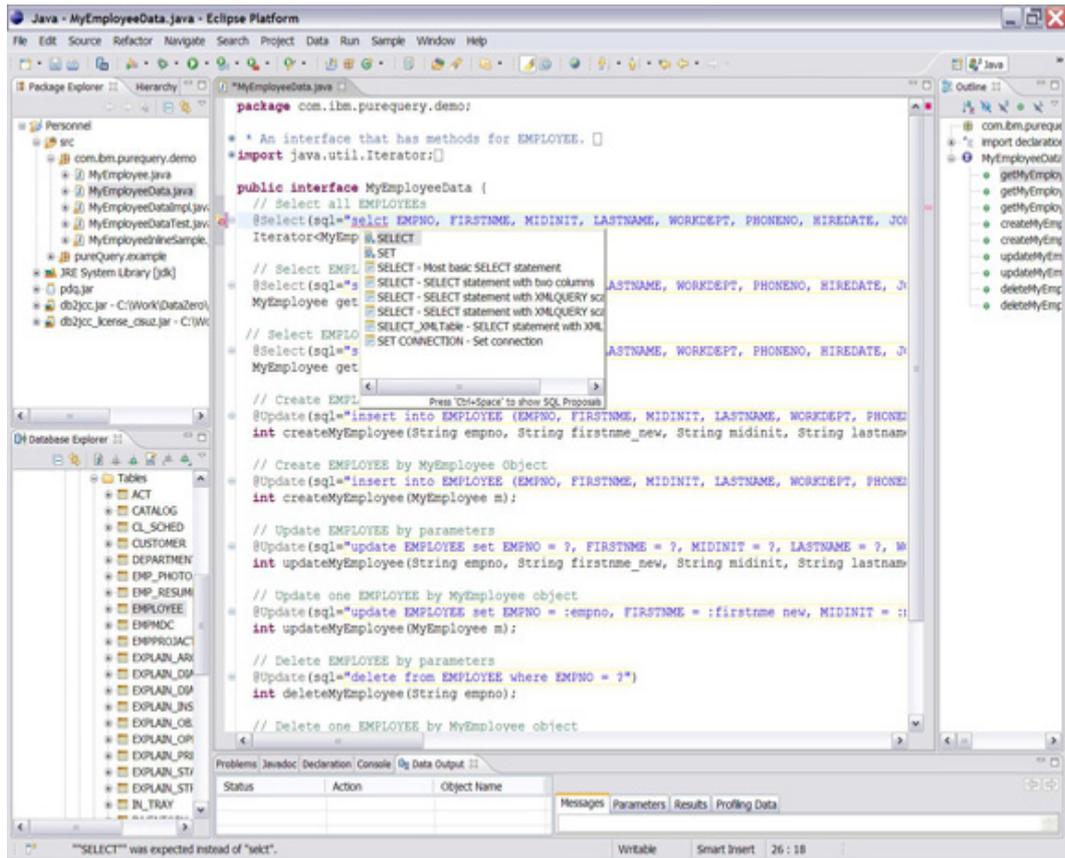
圖 22. 主變數內容輔助



要查看較大的圖檔，請按一下[這裡](#)。

- **Quick Fix** 提供線上嵌入式動態語法驗證機制，可偵測拼字等錯誤，並迅速修復。

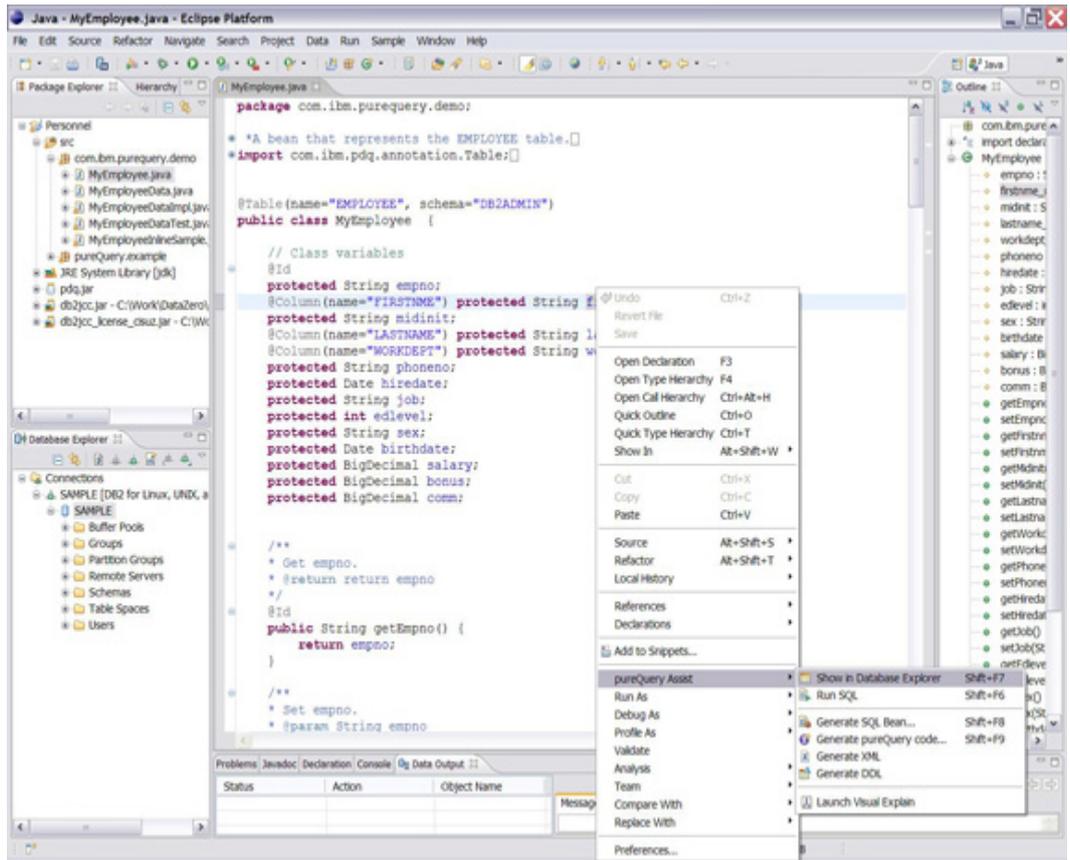
圖 23. Quick Fix：關鍵字選取 (Select) 誤拼成 Selct



要查看較大的圖檔，請按一下[這裡](#)。

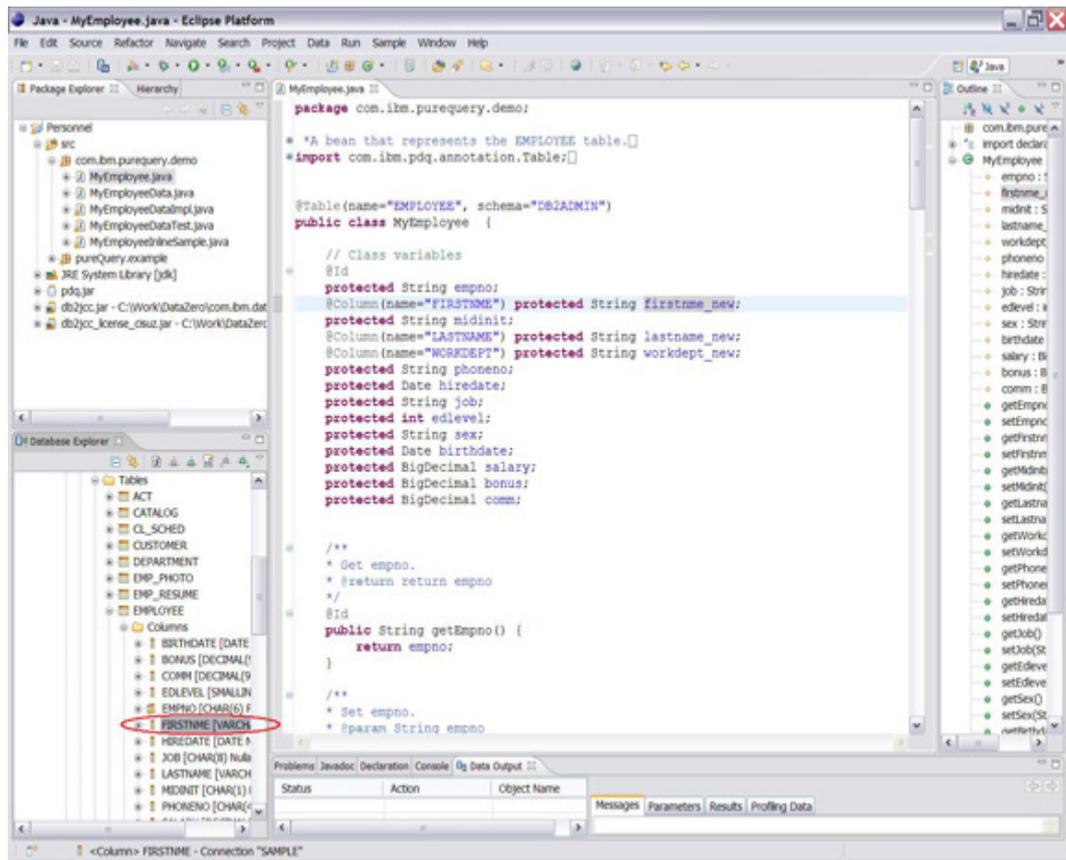
- **自動探索資料庫定義**（參閱圖 24 和 25）可點按選取任何 Bean 成員名稱，即可迅速跳至資料庫瀏覽器中的資料庫定義（例如，點按對映至表格名稱的類別名稱，將跳至資料庫瀏覽器中的表格節點，點按對映至直欄名稱的變數名稱，將跳至資料庫瀏覽器中的直欄節點等等）。

圖 24. 點按選取 Bean 類別中的成員名稱



要查看較大的圖檔，請按一下這裡。

圖 25. 點按選取跳至資料庫瀏覽器



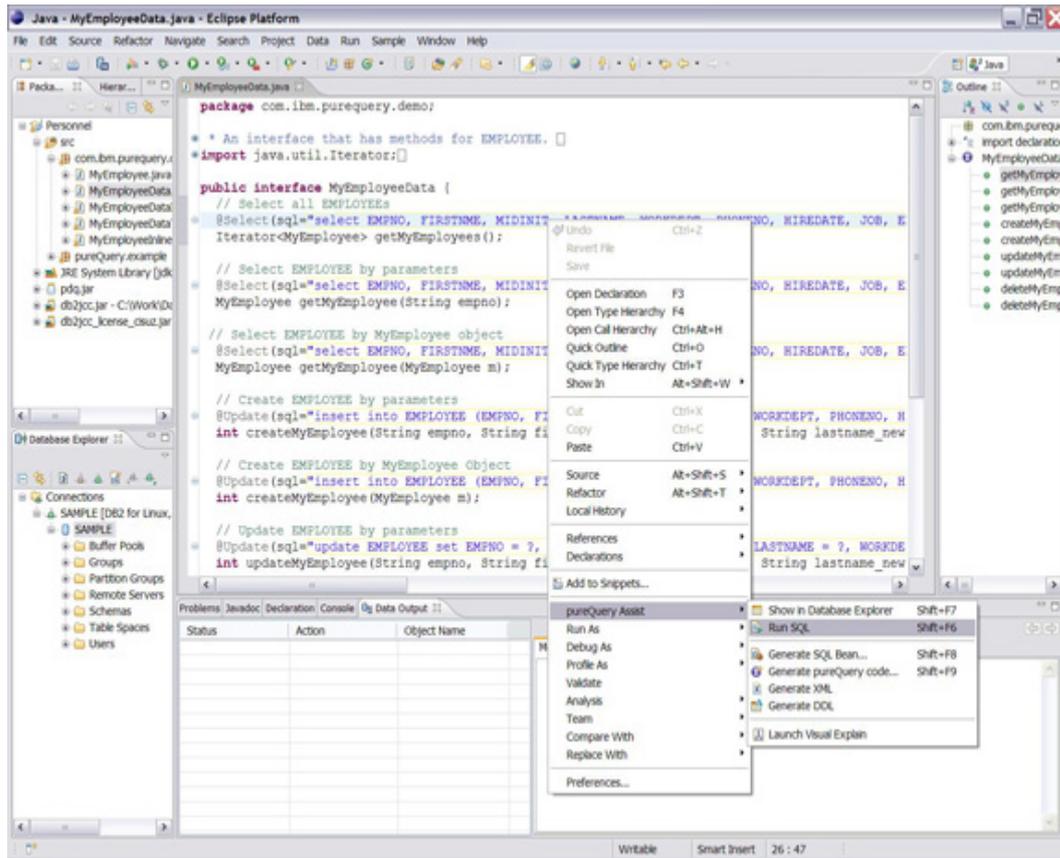
要查看較大的圖檔，請按一下[這裡](#)。

設計和執行時期的交握協商

pureQuery 設計時期功能的最大優勢之一，是可在編輯器中動態執行任何給定的 SQL 陳述式。pureQuery 藉此獨家特色，可運用回復功能，在設計時期動態執行 SQL 陳述式。

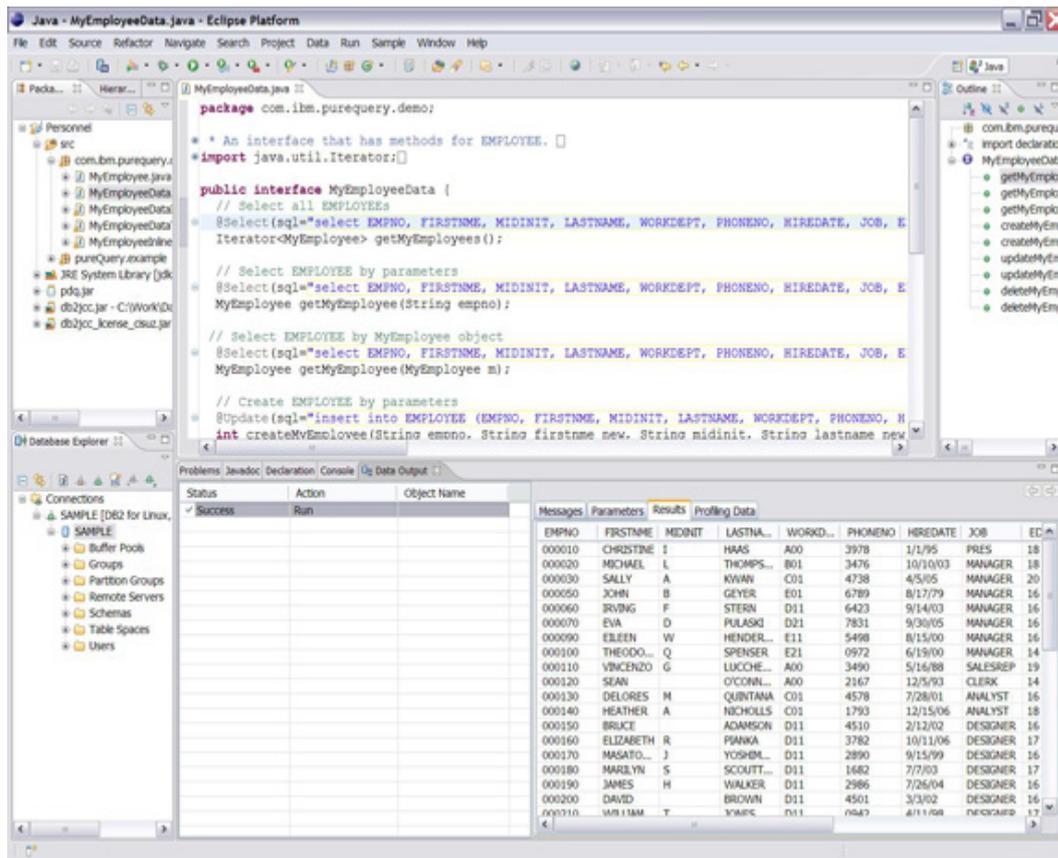
要查看這項特色，只要在 Java 編輯器中，以滑鼠右鍵點按任何所產生 SQL 陳述式下方的按鈕即可。從快速功能表選取 **pureQuery Assist** -> **Run SQL**（參閱圖 26），將執行 SQL 陳述式，並在 Eclipse IDE 的 Data View 中顯示結果集（參閱圖 27）。

圖 26. 點按選取以便在設計時期執行 SQL



要查看較大的圖檔，請按一下這裡。

圖 27. 資料視圖中顯示的設計時期 SQL 執行結果



要查看較大的圖檔，請按一下這裡。

總結

以資料庫為主的商業應用程式，目前正逐漸發展，改由 Web（或其他方式）提供服務；而資料和存取效率，正是推動此趨勢的力量。為提升生產力，以加速開發與維護應用程式，對於物件導向基礎架構的需求急速擴大，前所未見。IBM 的 pureQuery 透過完善、直觀的關聯存取物件化，提供最先進且可插入 Eclipse IDE 環境的解決方案，為開發人員提升生產力，將績效最佳化。

資源

Learn

- Learn more about the [IBM Data Studio pureQuery Runtime](#).
- Visit the [developerWorks resource page for IBM Data Studio](#) to read articles and tutorials and connect to other resources to expand your Data Studio skills.
- Learn about [DB2 Express-C](#), the no-charge version of DB2 Express Edition for the community.

Get products and technologies

- Download a free trial version of [DB2 Enterprise 9](#).
- Now you can use DB2 for free. Download [DB2 Express-C](#), a no-charge version of DB2 Express Edition for the community that offers the same core data features as DB2 Express Edition and provides a solid base to build and deploy applications.
- Download [IBM Data Studio](#).

關於作者

Azadeh Ahadian 是 IBM 在美國加州聖荷西矽谷實驗室資訊管理工具單位的軟體開發人員，擁有碩士學位，在軟體應用程式和工具開發領域的業界經驗超過八年。Azadeh 是 pureQuery 設計時期團隊的成員，研發貢獻良多。