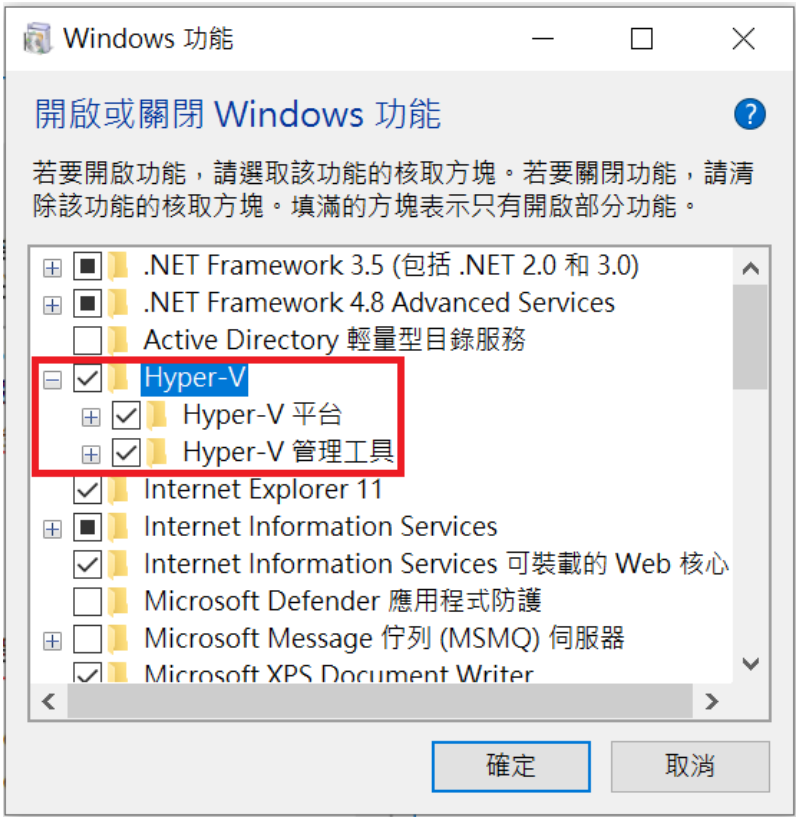
Dapper整合Oracle範例

環境設定

- Docker Desktop
  - 開啟 Hyper-V



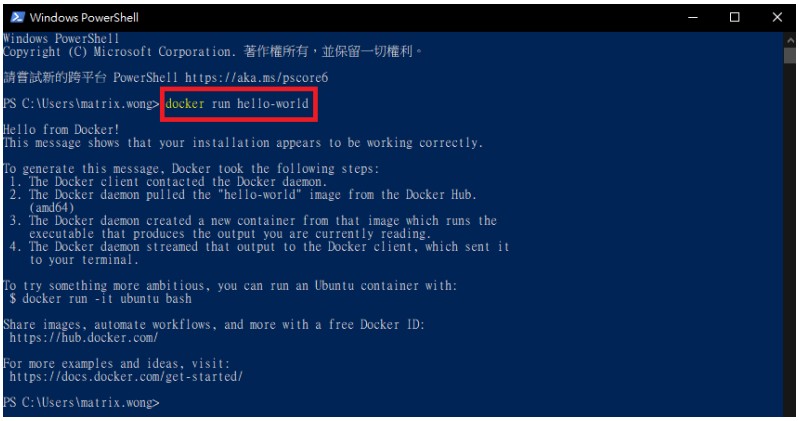
  - 安裝 WSL2

```
wsl --install
```

  - 安裝 Docker Desktop
    官方載點 https://www.docker.com/products/docker-desktop/
    說明文件 https://docs.docker.com/desktop/install/windows-install/

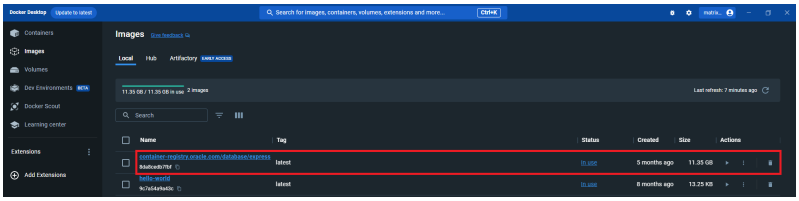
  - 測試 Docker Desktop

```
docker run hello-world
```



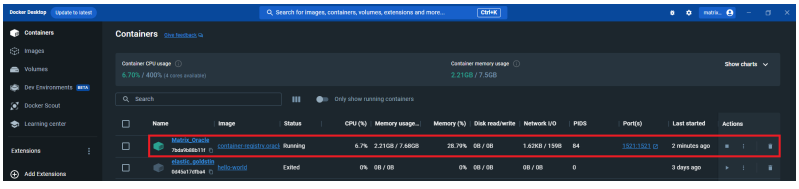
- Oracle
  - 下載 Docker Image

```
docker pull container-registry.oracle.com/database/express:latest
```

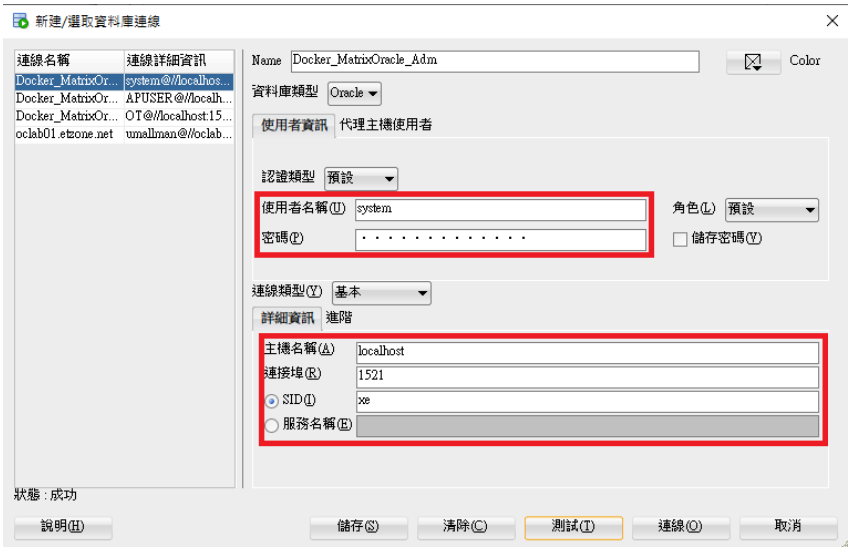


  - 建立 & 啟動 Container
    [container-name] : Container名稱
    [custom-pwd] : 預設使用者密碼

```
docker container create -it --name [container-name] -p 1521:1521 -e
ORACLE_PWD=[custom-pwd] container-
registry.oracle.com/database/express:latest
```



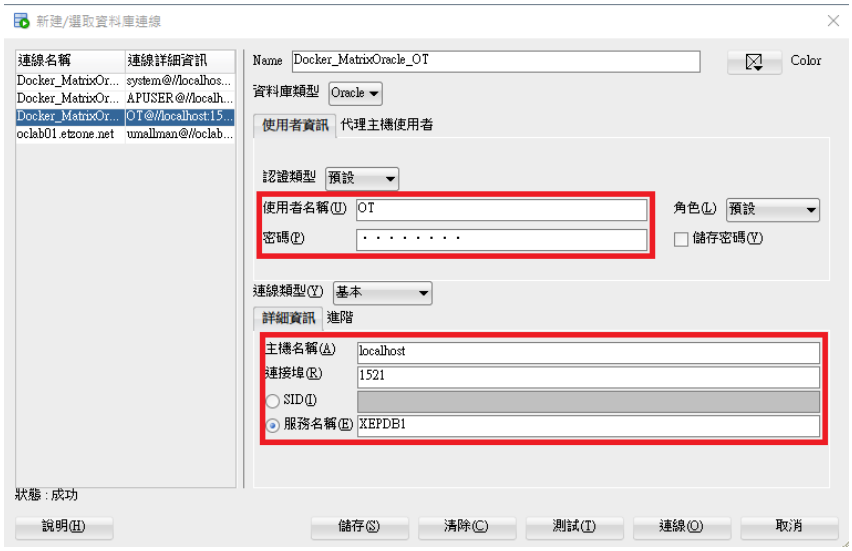
- SQL Developer 連線設定
  使用者名稱 : system
  密碼 : [custom-pwd]



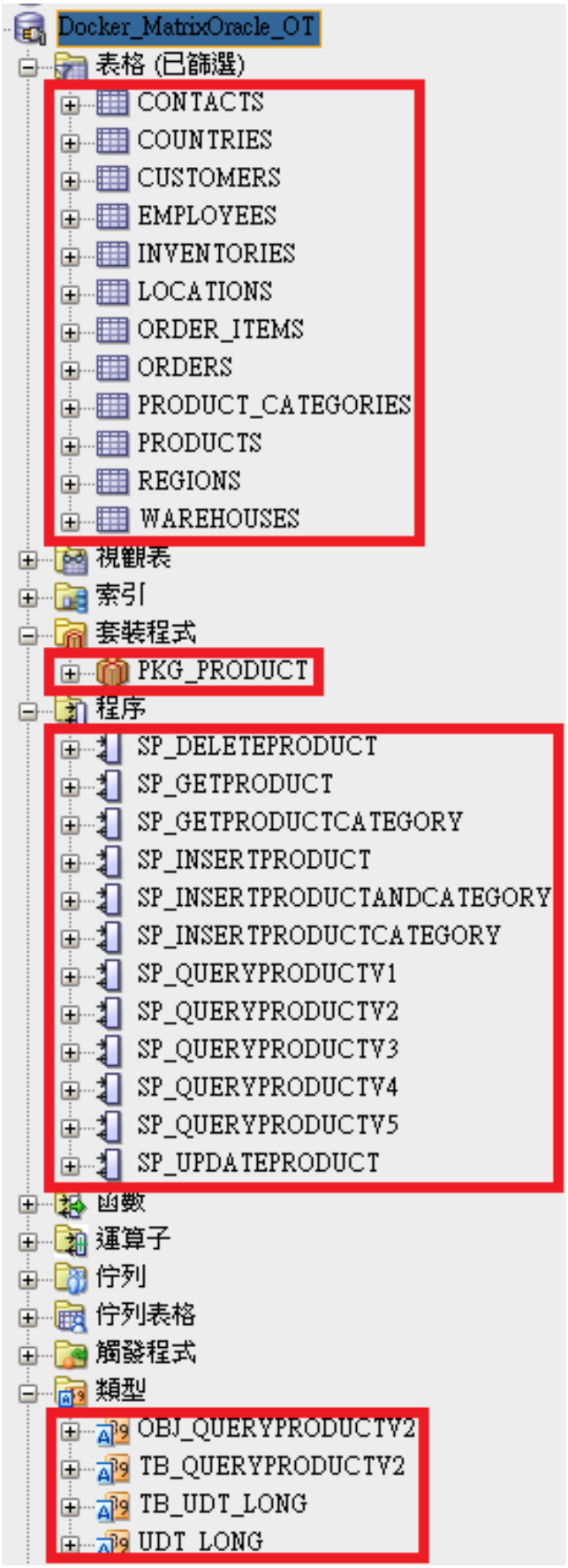
DB初始化設定

- SQL Script
  - InitialScript
    ot_create_user.pls : 建立使用者/權限
    ot_schema.pls : 建立測試資料表
    ot_data.pls : 建立測試資料
  - Type
    UDT_XXXXXX.pls : User-Defined Type
    OBJ_XXXXXX.pls : Type As Object
    TB_XXXXXX.pls : Type As Table Of Object
  - StoredProcedure
    SP_XXXXXX.pls : Procedure
  - Package
    PKG_XXXXXX.pls : Package

- SQL Developer 連線設定
  使用者名稱 : OT
  密碼 : Orcl1234

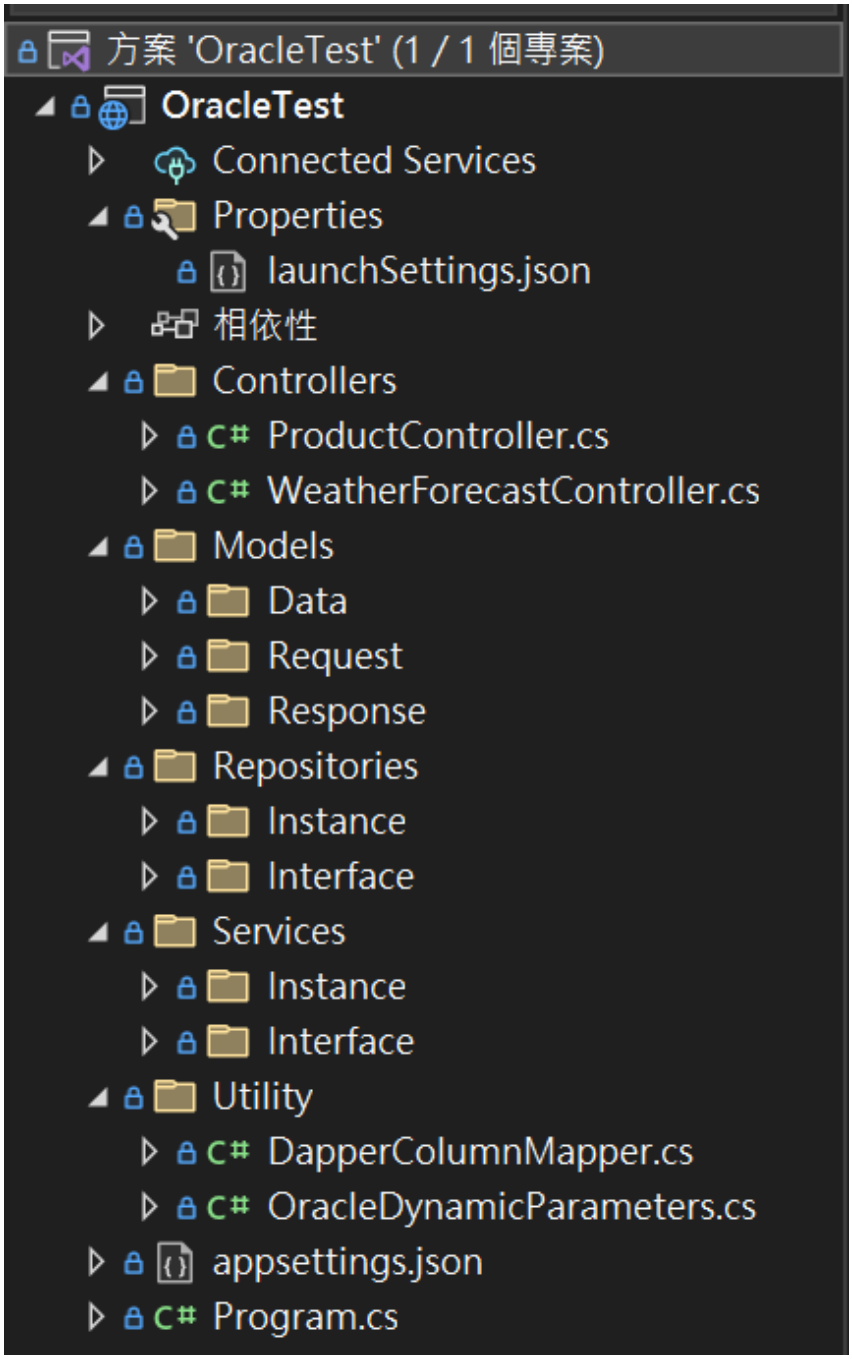


- 初始資料預覽


Docker_MatrixOracle_OT
表格 (已篩選)
CONTACTS
COUNTRIES
CUSTOMERS
EMPLOYEES
INVENTORIES
LOCATIONS
ORDER_ITEMS
ORDERS
PRODUCT_CATEGORIES
PRODUCTS
REGIONS
WAREHOUSES
視觀表
索引
套裝程式
PKG_PRODUCT
程序
SP_DELETEPRODUCT
SP_GETPRODUCT
SP_GETPRODUCTCATEGORY
SP_INSERTPRODUCT
SP_INSERTPRODUCTANDCATEGORY
SP_INSERTPRODUCTCATEGORY
SP_QUERYPRODUCTV1
SP_QUERYPRODUCTV2
SP_QUERYPRODUCTV3
SP_QUERYPRODUCTV4
SP_QUERYPRODUCTV5
SP_UPDATEPRODUCT
函數
運算子
佇列
佇列表格
觸發程式
類型
OBJ_QUERYPRODUCTV2
TB_QUERYPRODUCTV2
TB_UDT_LONG
UDT_LONG

專案架構介紹

- 概述
  Controllers : API接口
  Servies : 流程控制
  Repositories : 資料庫操作
  Models : API介接物件/資料庫物件
  Utility : 共用元件

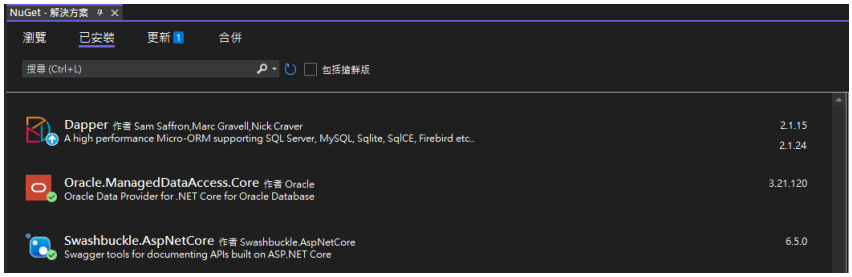


- 參考

  Dapper : Micro-ORM

  Oracle.ManagedDataAccess.Core : Oracle DB 資料驅動

  Swashbuckle.AspNetCore : Swagger



- 設定

  - appsettings.json

    DBConnectionStrings : DB 連線字串

    DbType : Repository實作類別 (RAW or SP)

```
{
  "DBConnectionStrings": {
    "Oracle": "Data Source=localhost:1521/XEPDB1; User Id=OT; Password=Orcl1234;"
  },
  "DbType": "RAW",
  "Logging": {
    "LogLevel": {
      "Default": "Information",
      "Microsoft.AspNetCore": "Warning"
    }
  },
  "AllowedHosts": "*"
}
```

  - Program.cs

    DI 註冊 : DB連線/Repository/Service

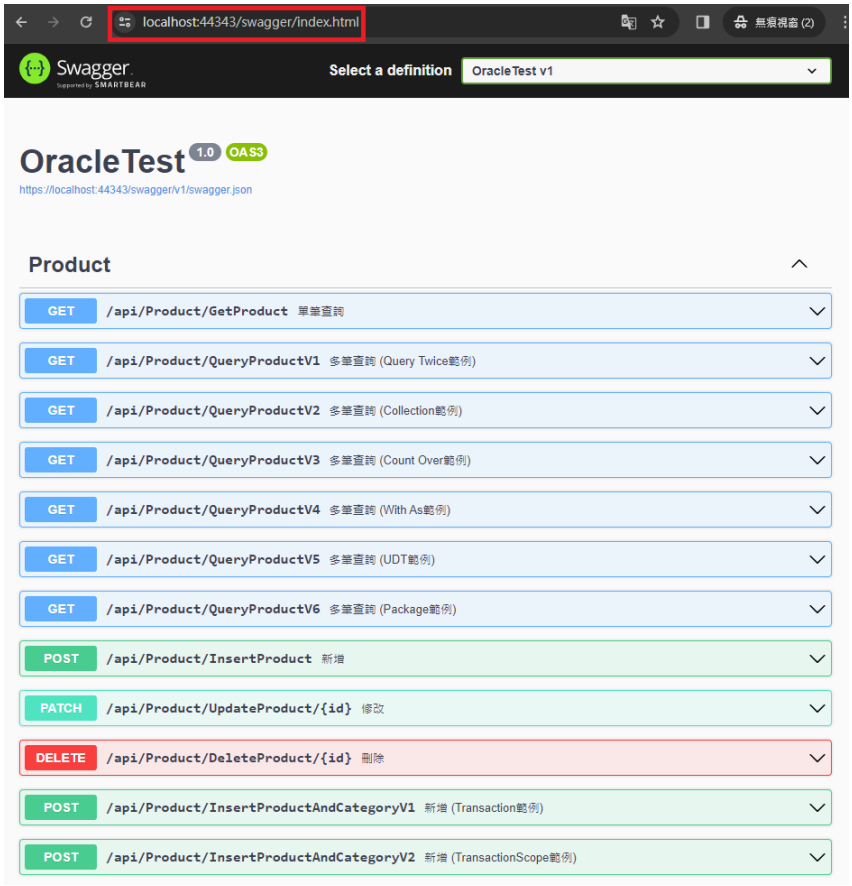
    Dapper 自定義欄位對應 : DB Table Column & Project Model

```
var builder = WebApplication.CreateBuilder(args);

// Add services to the container.

builder.Services.AddControllers();
// Learn more about configuring Swagger/OpenAPI at https://aka.ms/aspnetcore/swashbuckle
builder.Services.AddEndpointsApiExplorer();
builder.Services.AddSwaggerGen(c => {
    // XML 檔案: 文件註解標籤
    var xmlPath = Path.Combine(AppContext.BaseDirectory, "OracleTest.xml");
    c.IncludeXmlComments(xmlPath);
});

#region 註冊DB連線
var dbConnectString = (string)builder.Configuration.GetValue(typeof(string), "DBConnectionStrings:Oracle");
builder.Services.AddScoped<IDbConnection, OracleConnection>(db => new OracleConnection(dbConnectString));
#endregion

#region 註冊Repository
var dbType = builder.Configuration.GetValue(typeof(string), "DbType");
switch (dbType)
{
    case "RAW":
        builder.Services.AddScoped<IProductRepository, ProductRawRepository>();
        builder.Services.AddScoped<IProductCategoryRepository, ProductCategoryRawRepository>();
        break;

    case "SP":
        builder.Services.AddScoped<IProductRepository, ProductSPRepository>();
        builder.Services.AddScoped<IProductCategoryRepository, ProductCategorySPRepository>();
        break;
}
#endregion

#region 註冊Service
builder.Services.AddScoped<IProductService, ProductService>();
#endregion

#region Dapper自定義欄位對應
TypeMapper.Initialize("OracleTest.Models.Response");
#endregion
```

- launchSettings.json
  本機IISExpress啟動網址 : https://localhost:44343

```
{
  "$schema": "https://json.schemastore.org/launchsettings.json",
  "iisSettings": {
    "windowsAuthentication": false,
    "anonymousAuthentication": true,
    "iisExpress": {
      "applicationUrl": "http://localhost:51459",
      "sslPort": 44343
    }
  },
  "profiles": {
    "OracleTest": {
      "commandName": "Project",
      "dotnetRunMessages": true,
      "launchBrowser": true,
      "launchUrl": "swagger",
      "applicationUrl": "https://localhost:7275;http://localhost:5277",
      "environmentVariables": {
        "ASPNETCORE_ENVIRONMENT": "Development"
      }
    },
    "IIS Express": {
      "commandName": "IISExpress",
      "launchBrowser": false,
      "launchUrl": "swagger",
      "environmentVariables": {
        "ASPNETCORE_ENVIRONMENT": "Development"
      }
    }
  }
}
```

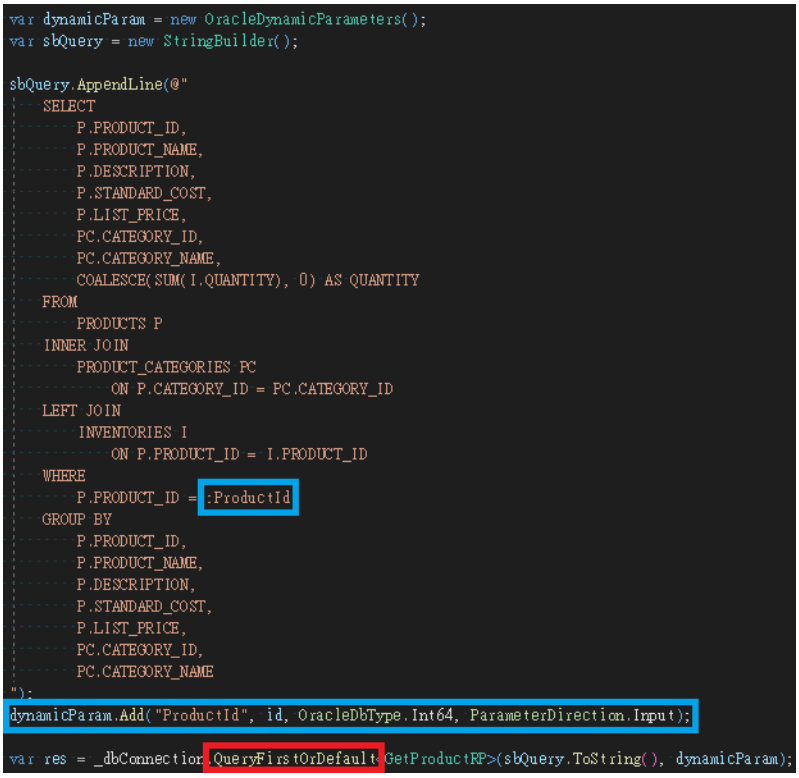
- Swagger : https://localhost:44343/swagger



CRUD範例

- 單筆查詢
  - RAW

    SQL字串中變數使用前綴：符號對應傳入參數名稱

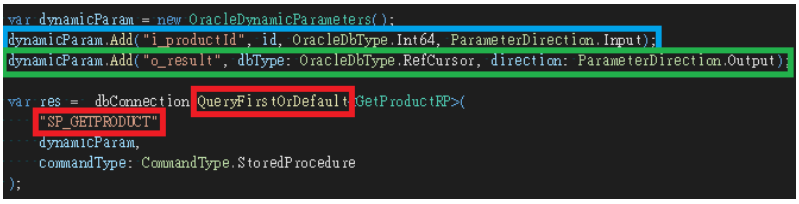


```
var dynamicParam = new OracleDynamicParameters();
var sbQuery = new StringBuilder();

sbQuery.AppendLine(@"
    SELECT
        P.PRODUCT_ID,
        P.PRODUCT_NAME,
        P.DESCRIPTION,
        P.STANDARD_COST,
        P.LIST_PRICE,
        PC.CATEGORY_ID,
        PC.CATEGORY_NAME,
        COALESCE(SUM(I.QUANTITY), 0) AS QUANTITY
    FROM
        PRODUCTS P
    INNER JOIN
        PRODUCT_CATEGORIES PC
            ON P.CATEGORY_ID = PC.CATEGORY_ID
    LEFT JOIN
        INVENTORIES I
            ON P.PRODUCT_ID = I.PRODUCT_ID
    WHERE
        P.PRODUCT_ID = :ProductId
    GROUP BY
        P.PRODUCT_ID,
        P.PRODUCT_NAME,
        P.DESCRIPTION,
        P.STANDARD_COST,
        P.LIST_PRICE,
        PC.CATEGORY_ID,
        PC.CATEGORY_NAME
");
dynamicParam.Add("ProductId", id, OracleDbType.Int64, ParameterDirection.Input);

var res = _dbConnection.QueryFirstOrDefault<GetProductRP>(sbQuery.ToString(), dynamicParam);
```

  - SP

    程式呼叫端：

    傳入 Input / Output 參數



```
var dynamicParam = new OracleDynamicParameters();
dynamicParam.Add("i_productId", id, OracleDbType.Int64, ParameterDirection.Input);
dynamicParam.Add("o_result", dbType: OracleDbType.RefCursor, direction: ParameterDirection.Output);

var res = _dbConnection.QueryFirstOrDefault<GetProductRP>(
    "SP_GETPRODUCT"
    dynamicParam,
    commandType: CommandType.StoredProcedure
);
```

    SP執行端：

使用 OPEN Cursor FOR 將查詢結果賦予 Out Cursor 回傳

```
create or replace PROCEDURE SP_GETPRODUCT(
    i_productId NUMBER,
    o_result OUT SYS_REFCURSOR
)
AS
BEGIN

    OPEN o_result FOR
    SELECT
        P.PRODUCT_ID,
        P.PRODUCT_NAME,
        P.DESCRIPTION,
        P.STANDARD_COST,
        P.LIST_PRICE,
        PC.CATEGORY_ID,
        PC.CATEGORY_NAME,
        COALESCE(SUM(I.QUANTITY), 0) AS QUANTITY
    FROM
        PRODUCTS P
    INNER JOIN
        PRODUCT_CATEGORIES PC
            ON P.CATEGORY_ID = PC.CATEGORY_ID
    LEFT JOIN
        INVENTORIES I
            ON P.PRODUCT_ID = I.PRODUCT_ID
    WHERE
        P.PRODUCT_ID = i_productId
    GROUP BY
        P.PRODUCT_ID,
        P.PRODUCT_NAME,
        P.DESCRIPTION,
        P.STANDARD_COST,
        P.LIST_PRICE,
        PC.CATEGORY_ID,
        PC.CATEGORY_NAME;

END SP_GETPRODUCT;
```

- 多筆查詢 (Query Twice)
  - RAW
    因 Oracle 無法識別多個查詢語句，故須使用 Cursor 才能接收多個回傳集合
    使用多個 OPEN Cursor FOR 將不同查詢結果分別賦予 Out Cursor 回傳

```
sbCnt.AppendLine(@"
    OPEN :o_Cnt FOR
    SELECT
        COUNT(1)
    FROM
        PRODUCTS P
    INNER JOIN
        PRODUCT_CATEGORIES PC
            ON P.CATEGORY_ID = PC.CATEGORY_ID
    WHERE 1 = 1
");
dynamicParam.Add("o_Cnt", dbType: OracleDbType.RefCursor, direction: ParameterDirection.Output);

sbResult.AppendLine(@"
    OPEN :o_Result FOR
    SELECT
        P.PRODUCT_ID,
        P.PRODUCT_NAME,
        P.DESCRIPTION,
        P.STANDARD_COST,
        P.LIST_PRICE,
        PC.CATEGORY_ID,
        PC.CATEGORY_NAME,
        COALESCE(SUM(I.QUANTITY), 0) AS QUANTITY
    FROM
        PRODUCTS P
    INNER JOIN
        PRODUCT_CATEGORIES PC
            ON P.CATEGORY_ID = PC.CATEGORY_ID
    LEFT JOIN
        INVENTORIES I
            ON P.PRODUCT_ID = I.PRODUCT_ID
    WHERE 1 = 1
");
dynamicParam.Add("o_Result", dbType: OracleDbType.RefCursor, direction: ParameterDirection.Output);

[Query Condition]

sbCnt.Append(sbQuery.ToString());
sbResult.Append(sbQuery.ToString());

[GroupBy]

[Order]

[Paging]

var res = _dbConnection.QueryMultiple($"BEGIN {sbCnt}; {sbResult}; END;", dynamicParam);

return (res.Read<int>().FirstOrDefault(), res.Read<QueryProductVlRP>());
```

  - SP
    程式呼叫端：
    傳入 Input / Output 參數

```
var dynamicParam = new OracleDynamicParameters();
dynamicParam.Add("i_categoryId", dicParams.GetValueOrDefault("CategoryId"), OracleDbType.Int64, ParameterDirection.Input);
dynamicParam.Add("i_categoryIds", dicParams.GetValueOrDefault("CategoryIds") == null ?
                                  null : string.Join(',', (IEnumerable<long>)dicParams.GetValueOrDefault("CategoryIds")),
                  OracleDbType.Varchar2, ParameterDirection.Input);
dynamicParam.Add("i_productName", dicParams.GetValueOrDefault("ProductName"), OracleDbType.Varchar2, ParameterDirection.Input);
dynamicParam.Add("i_rowStart", dicParams.GetValueOrDefault("RowStart"), OracleDbType.Int32, ParameterDirection.Input);
dynamicParam.Add("i_rowLength", dicParams.GetValueOrDefault("RowLength"), OracleDbType.Int32, ParameterDirection.Input);
dynamicParam.Add("o_cnt", dbType: OracleDbType.RefCursor, direction: ParameterDirection.Output);
dynamicParam.Add("o_result", dbType: OracleDbType.RefCursor, direction: ParameterDirection.Output);

var res = dbConnection.QueryMultiple(
    "SP_QUERYPRODUCTV1"
    dynamicParam,
    commandType: CommandType.StoredProcedure
);

return (res.Read<int>().FirstOrDefault(), res.Read<QueryProductVlRP>())
```

    SP執行端：
    使用多個 OPEN Cursor FOR 將不同查詢結果分別賦予 Out Cursor 回傳
    INSTR函式執行 WHERE IN 效果
    OFFSET [n] ROWS FETCH NEXT [m] ROWS ONLY 執行分頁效果

```
create or replace PROCEDURE SP_QUERYPRODUCTV1(
    i_categoryId NUMBER DEFAULT NULL,
    i_categoryIds VARCHAR2 DEFAULT NULL,
    i_productName VARCHAR2 DEFAULT NULL,
```

```
    i_rowStart NUMBER DEFAULT 0,
    i_rowLength NUMBER DEFAULT 10,
    o_cnt OUT SYS_REFCURSOR,
    o_result OUT SYS_REFCURSOR
)
AS
BEGIN

    OPEN o_cnt FOR
    SELECT
        COUNT(1)
    FROM
        PRODUCTS P
    INNER JOIN
        PRODUCT_CATEGORIES PC
            ON P.CATEGORY_ID = PC.CATEGORY_ID
    WHERE 1 = 1
    AND
    (
        (i_categoryId IS NULL)
        OR
        (PC.CATEGORY_ID = i_categoryId
    )
    AND
    (
        (i_categoryIds IS NULL)
        OR
        (INSTR(','|| i_categoryIds ||',', ','||PC.CATEGORY_ID||',') > 0)
    )
    AND
    (
        (i_productName IS NULL)
        OR
        (LOWER(P.PRODUCT_NAME) LIKE i_productName
    );

    OPEN o_result FOR
    SELECT
        P.PRODUCT_ID,
        P.PRODUCT_NAME,
        P.DESCRIPTION,
        P.STANDARD_COST,
        P.LIST_PRICE,
        PC.CATEGORY_ID,
        PC.CATEGORY_NAME,
        COALESCE(SUM(I.QUANTITY), 0) AS QUANTITY
    FROM
        PRODUCTS P
    INNER JOIN
        PRODUCT_CATEGORIES PC
            ON P.CATEGORY_ID = PC.CATEGORY_ID
    LEFT JOIN
        INVENTORIES I
            ON P.PRODUCT_ID = I.PRODUCT_ID
    WHERE 1 = 1
    AND
    (
        (i_categoryId IS NULL)
        OR
        (PC.CATEGORY_ID = i_categoryId
    )
    AND
    (
        (i_categoryIds IS NULL)
        OR
        (INSTR(','|| i_categoryIds ||',', ','||PC.CATEGORY_ID||',') > 0)
    )
    AND
```

```
    (
        (i_productName IS NULL)
        OR
        (LOWER(P.PRODUCT_NAME) LIKE i_productName
    )
    GROUP BY
        P.PRODUCT_ID,
        P.PRODUCT_NAME,
        P.DESCRIPTION,
        P.STANDARD_COST,
        P.LIST_PRICE,
        PC.CATEGORY_ID,
        PC.CATEGORY_NAME
    ORDER BY
        P.PRODUCT_ID DESC
    OFFSET i_rowStart ROWS FETCH NEXT i_rowLength ROWS ONLY;

END SP_QUERYPRODUCTV1;
```

- 多筆查詢 (Collection)
  預先建立 Type 物件

```
create or replace TYPE OBJ_QUERYPRODUCTV2 AS OBJECT (
    PRODUCT_ID NUMBER,
    PRODUCT_NAME VARCHAR2(255),
    DESCRIPTION VARCHAR2(2000),
    STANDARD_COST NUMBER(9,2),
    LIST_PRICE NUMBER(9,2),
    CATEGORY_ID NUMBER,
    CATEGORY_NAME VARCHAR2(255)
);
create or replace TYPE TB_QUERYPRODUCTV2 AS TABLE OF OBJ_QUERYPRODUCTV2
```

  - RAW
    定義 Type 變數，將共用邏輯查詢結果存放在此變數中
    從 Type 變數中執行不同的查詢，再分別賦予 Out Cursor 回傳

```
sbDeclare.AppendLine(@"
    DECLARE v_TempTable TB_QUERYPRODUCTV2
");

sbTemp.AppendLine(@"
    SELECT OBJ_QUERYPRODUCTV2
    (
        P.PRODUCT_ID,
        P.PRODUCT_NAME,
        P.DESCRIPTION,
        P.STANDARD_COST,
        P.LIST_PRICE,
        PC.CATEGORY_ID,
        PC.CATEGORY_NAME
    )
    BULK COLLECT INTO
        v_TempTable
    FROM
        PRODUCTS P
    INNER JOIN
        PRODUCT_CATEGORIES PC
            ON P.CATEGORY_ID = PC.CATEGORY_ID
    WHERE 1 = 1
");

[Query Condition]

sbTemp.Append(sbQuery.ToString());

sbCnt.AppendLine(@"
    OPEN :o_Cnt FOR
    SELECT
        COUNT(1)
    FROM
        TABLE(CAST(v_TempTable AS TB_QUERYPRODUCTV2))
");
dynamicParam.Add("o_Cnt", dbType: OracleDbType.RefCursor, direction: ParameterDirection.Output);

sbResult.AppendLine(@"
    OPEN :o_Result FOR
    SELECT
        R.*,
        COALESCE(SUM(I.QUANTITY), 0) AS QUANTITY
    FROM
        TABLE(CAST(v_TempTable AS TB_QUERYPRODUCTV2)) R
    LEFT JOIN
        INVENTORIES I
            ON R.PRODUCT_ID = I.PRODUCT_ID
");
dynamicParam.Add("o_Result", dbType: OracleDbType.RefCursor, direction: ParameterDirection.Output);

[GroupBy]

[Order]

[Paging]

var res = _dbConnection.QueryMultiple($"{sbDeclare}; BEGIN {sbTemp}; {sbCnt}; {sbResult}; END;", dynamicParam);

return (res.Read<int>().FirstOrDefault(), res.Read<QueryProductV2RP>())
```

  - SP
    程式呼叫端：
    傳入 Input / Output 參數

```
var dynamicParam = new OracleDynamicParameters();
dynamicParam.Add("i_categoryId", dicParams.GetValueOrDefault("CategoryId"), OracleDbType.Int64, ParameterDirection.Input);
dynamicParam.Add("i_categoryIds", dicParams.GetValueOrDefault("CategoryIds") == null ?
                                  null : string.Join(",", (IEnumerable<long>)dicParams.GetValueOrDefault("CategoryIds")),
                 OracleDbType.Varchar2, ParameterDirection.Input);
dynamicParam.Add("i_productName", dicParams.GetValueOrDefault("ProductName"), OracleDbType.Varchar2, ParameterDirection.Input);
dynamicParam.Add("i_rowStart", dicParams.GetValueOrDefault("RowStart"), OracleDbType.Int32, ParameterDirection.Input);
dynamicParam.Add("i_rowLength", dicParams.GetValueOrDefault("RowLength"), OracleDbType.Int32, ParameterDirection.Input);
dynamicParam.Add("o_cnt", dbType: OracleDbType.RefCursor, direction: ParameterDirection.Output);
dynamicParam.Add("o_result", dbType: OracleDbType.RefCursor, direction: ParameterDirection.Output);

var res = dbConnection.QueryMultiple(
    "SP_QUERYPRODUCTV2"
    dynamicParam,
    commandType: CommandType.StoredProcedure
);

return (res.Read<int>().FirstOrDefault(), res.Read<QueryProductV2RP>())
```

SP執行端：

定義 Type 變數，將共用邏輯查詢結果存放在此變數中

從 Type 變數中執行不同的查詢，再分別賦予 Out Cursor 回傳

```
create or replace PROCEDURE SP_QUERYPRODUCTV2(
    i_categoryId NUMBER DEFAULT NULL,
    i_categoryIds VARCHAR2 DEFAULT NULL,
    i_productName VARCHAR2 DEFAULT NULL,
    i_rowStart NUMBER DEFAULT 0,
    i_rowLength NUMBER DEFAULT 10,
    o_cnt OUT SYS_REFCURSOR,
    o_result OUT SYS_REFCURSOR
)
AS
    v_TempTable TB_QUERYPRODUCTV2;
BEGIN

    SELECT OBJ_QUERYPRODUCTV2
    (
        P.PRODUCT_ID,
        P.PRODUCT_NAME,
        P.DESCRIPTION,
        P.STANDARD_COST,
        P.LIST_PRICE,
        PC.CATEGORY_ID,
        PC.CATEGORY_NAME
    )
    BULK COLLECT INTO
        v_TempTable
    FROM
        PRODUCTS P
    INNER JOIN
        PRODUCT_CATEGORIES PC
            ON P.CATEGORY_ID = PC.CATEGORY_ID
    WHERE 1 = 1
    AND
    (
        (i_categoryId IS NULL)
        OR
        (PC.CATEGORY_ID = i_categoryId
    )
    AND
    (
        (i_categoryIds IS NULL)
        OR
        (INSTR(','|| i_categoryIds ||',', ','||PC.CATEGORY_ID||',') > 0)
    )
    AND
    (
        (i_productName IS NULL)
        OR
        (LOWER(P.PRODUCT_NAME) LIKE i_productName
    );

    OPEN o_cnt FOR
    SELECT
```

```
        COUNT(1)
    FROM
        TABLE(CAST(v_TempTable AS TB_QUERYPRODUCTV2));

    OPEN o_result FOR
    SELECT
        R.*,
        COALESCE(SUM(I.QUANTITY), 0) AS QUANTITY
    FROM
        TABLE(CAST(v_TempTable AS TB_QUERYPRODUCTV2)) R
    LEFT JOIN
        INVENTORIES I
            ON R.PRODUCT_ID = I.PRODUCT_ID
    GROUP BY
        R.PRODUCT_ID,
        R.PRODUCT_NAME,
        R.DESCRIPTION,
        R.STANDARD_COST,
        R.LIST_PRICE,
        R.CATEGORY_ID,
        R.CATEGORY_NAME
    ORDER BY R.PRODUCT_ID DESC
    OFFSET i_rowStart ROWS FETCH NEXT i_rowLength ROWS ONLY;

END SP QUERYPRODUCTV2;
```

- 多筆查詢 (Count Over)
  - RAW

    使用 Count Over 函式輔助，合併查詢

```
var dynamicParam = new OracleDynamicParameters();
var sbSQL = new StringBuilder();
sbSQL.AppendLine(@"
    SELECT
        P.PRODUCT_ID,
        P.PRODUCT_NAME,
        P.DESCRIPTION,
        P.STANDARD_COST,
        P.LIST_PRICE,
        PC.CATEGORY_ID,
        PC.CATEGORY_NAME,
        COALESCE(SUM(I.QUANTITY), 0) AS QUANTITY,
        COUNT(*) OVER() AS TotalCnt
    FROM
        PRODUCTS P
    INNER JOIN
        PRODUCT_CATEGORIES PC
            ON P.CATEGORY_ID = PC.CATEGORY_ID
    LEFT JOIN
        INVENTORIES I
            ON P.PRODUCT_ID = I.PRODUCT_ID
    WHERE 1 = 1
");

[Query Condition]

[GroupBy]

[Order]

[Paging]

var res = _dbConnection.Query<QueryProductV3RP>(sbSQL.ToString(), dynamicParam);

return (res.FirstOrDefault()?.TotalCnt ?? 0, res);
```

  - SP

    程式呼叫端：

    傳入 Input / Output 參數

```
var dynamicParam = new OracleDynamicParameters();
dynamicParam.Add("i_categoryId", dicParams.GetValueOrDefault("CategoryId"), OracleDbType.Int64, ParameterDirection.Input);
dynamicParam.Add("i_categoryIds", dicParams.GetValueOrDefault("CategoryIds") == null ?
                                  null : string.Join(',', (IEnumerable<long>)dicParams.GetValueOrDefault("CategoryIds")),
                 OracleDbType.Varchar2, ParameterDirection.Input);
dynamicParam.Add("i_productName", dicParams.GetValueOrDefault("ProductName"), OracleDbType.Varchar2, ParameterDirection.Input);
dynamicParam.Add("i_rowStart", dicParams.GetValueOrDefault("RowStart"), OracleDbType.Int32, ParameterDirection.Input);
dynamicParam.Add("i_rowLength", dicParams.GetValueOrDefault("RowLength"), OracleDbType.Int32, ParameterDirection.Input);
dynamicParam.Add("o_result", dbType: OracleDbType.RefCursor, direction: ParameterDirection.Output);

var res = _dbConnection.Query<QueryProductV3RP>(
    "SP_QUERYPRODUCTV3"
    dynamicParam,
    commandType: CommandType.StoredProcedure
);

return (res.FirstOrDefault()?.TotalCnt ?? 0, res)
```

    SP執行端：

使用 Count Over 函式輔助，合併查詢

```
create or replace PROCEDURE SP_QUERYPRODUCTV3
(
    i_categoryId NUMBER DEFAULT NULL,
    i_categoryIds VARCHAR2 DEFAULT NULL,
    i_productName VARCHAR2 DEFAULT NULL,
    i_rowStart NUMBER DEFAULT 0,
    i_rowLength NUMBER DEFAULT 10,
    o_result OUT SYS_REFCURSOR
)
AS
BEGIN

    OPEN o_result FOR
    SELECT
        P.PRODUCT_ID,
        P.PRODUCT_NAME,
        P.DESCRIPTION,
        P.STANDARD_COST,
        P.LIST_PRICE,
        PC.CATEGORY_ID,
        PC.CATEGORY_NAME,
        COALESCE(SUM(I.QUANTITY), 0) AS QUANTITY,
        COUNT(*) OVER() AS TotalCnt
    FROM
        PRODUCTS P
    INNER JOIN
        PRODUCT_CATEGORIES PC
            ON P.CATEGORY_ID = PC.CATEGORY_ID
    LEFT JOIN
        INVENTORIES I
            ON P.PRODUCT_ID = I.PRODUCT_ID
    WHERE 1 = 1
    AND
    (
        (i_categoryId IS NULL)
        OR
        (PC.CATEGORY_ID = i_categoryId
    )
    AND
    (
        (i_categoryIds IS NULL)
        OR
        (INSTR(','|| i_categoryIds ||',', ','||PC.CATEGORY_ID||',') > 0)
    )
    AND
    (
        (i_productName IS NULL)
        OR
        (LOWER(P.PRODUCT_NAME) LIKE i_productName
    )
    GROUP BY
        P.PRODUCT_ID,
        P.PRODUCT_NAME,
        P.DESCRIPTION,
        P.STANDARD_COST,
        P.LIST_PRICE,
        PC.CATEGORY_ID,
        PC.CATEGORY_NAME
    ORDER BY P.PRODUCT_ID DESC
    OFFSET i_rowStart ROWS FETCH NEXT i_rowLength ROWS ONLY;

END SP_QUERYPRODUCTV3;
```

- 多筆查詢 (With As)

  - RAW

    使用 With As 將共用邏輯查詢結果暫存，後續再執行實務需求

```
var dynamicParam = new OracleDynamicParameters();
var sbSQL = new StringBuilder();
sbSQL.AppendLine(@"
    WITH CTE AS
    (
        SELECT
            P.PRODUCT_ID,
            P.PRODUCT_NAME,
            P.DESCRIPTION,
            P.STANDARD_COST,
            P.LIST_PRICE,
            PC.CATEGORY_ID,
            PC.CATEGORY_NAME
        FROM
            PRODUCTS P
        INNER JOIN
            PRODUCT_CATEGORIES PC
                ON P.CATEGORY_ID = PC.CATEGORY_ID
        WHERE 1 = 1
");

[Query Condition]

sbSQL.AppendLine(@"
    )
    SELECT
        C.*,
        COALESCE(SUM(I.QUANTITY), 0) AS QUANTITY,
        COUNT(*) OVER() AS TotalCnt
    FROM
        CTE C
    LEFT JOIN
        INVENTORIES I
            ON C.PRODUCT_ID = I.PRODUCT_ID
");

[GroupBy]

[Order]

[Paging]

var res = _dbConnection.Query<QueryProductV4RP>(sbSQL.ToString(), dynamicParam);

return (res.FirstOrDefault()?.TotalCnt ?? 0, res);
```

  - SP

    程式呼叫端：

    傳入 Input / Output 參數

```
var dynamicParam = new OracleDynamicParameters();
dynamicParam.Add("i_categoryId", dicParams.GetValueOrDefault("CategoryId"), OracleDbType.Int64, ParameterDirection.Input);
dynamicParam.Add("i_categoryIds", dicParams.GetValueOrDefault("CategoryIds") == null ?
                                  null : String.Join(",", (IEnumerable<long>)dicParams.GetValueOrDefault("CategoryIds")),
                 OracleDbType.Varchar2, ParameterDirection.Input);
dynamicParam.Add("i_productName", dicParams.GetValueOrDefault("ProductName"), OracleDbType.Varchar2, ParameterDirection.Input);
dynamicParam.Add("i_rowStart", dicParams.GetValueOrDefault("RowStart"), OracleDbType.Int32, ParameterDirection.Input);
dynamicParam.Add("i_rowLength", dicParams.GetValueOrDefault("RowLength"), OracleDbType.Int32, ParameterDirection.Input);
dynamicParam.Add("o_result", dbType: OracleDbType.RefCursor, direction: ParameterDirection.Output);

var res = _dbConnection.Query<QueryProductV4RP>(
    "SP_QUERYPRODUCTV4"
    dynamicParam,
    commandType: CommandType.StoredProcedure
);

return (res.FirstOrDefault()?.TotalCnt ?? 0, res)
```

    SP執行端：

使用 With As 將共用邏輯查詢結果暫存，後續再執行實務需求

```
create or replace PROCEDURE SP_QUERYPRODUCTV4
(
    i_categoryId NUMBER DEFAULT NULL,
    i_categoryIds VARCHAR2 DEFAULT NULL,
    i_productName VARCHAR2 DEFAULT NULL,
    i_rowStart NUMBER DEFAULT 0,
    i_rowLength NUMBER DEFAULT 10,
    o_result OUT SYS_REFCURSOR
)
AS
BEGIN

    OPEN o_result FOR
    WITH CTE AS
    (
        SELECT
            P.PRODUCT_ID,
            P.PRODUCT_NAME,
            P.DESCRIPTION,
            P.STANDARD_COST,
            P.LIST_PRICE,
            PC.CATEGORY_ID,
            PC.CATEGORY_NAME
        FROM
            PRODUCTS P
        INNER JOIN
            PRODUCT_CATEGORIES PC
                ON P.CATEGORY_ID = PC.CATEGORY_ID
        WHERE 1 = 1
        AND
        (
            (i_categoryId IS NULL)
            OR
            (PC.CATEGORY_ID = i_categoryId)
        )
        AND
        (
            (i_categoryIds IS NULL)
            OR
            (INSTR(','|| i_categoryIds ||',', ','||PC.CATEGORY_ID||',') > 0)
        )
        AND
        (
            (i_productName IS NULL)
            OR
            (LOWER(P.PRODUCT_NAME) LIKE i_productName)
        )
    )
    SELECT
        C.*,
        COALESCE(SUM(I.QUANTITY), 0) AS QUANTITY,
        COUNT(*) OVER() AS TotalCnt
    FROM
        CTE C
    LEFT JOIN
        INVENTORIES I
            ON C.PRODUCT_ID = I.PRODUCT_ID
    GROUP BY
        C.PRODUCT_ID,
        C.PRODUCT_NAME,
        C.DESCRIPTION,
        C.STANDARD_COST,
        C.LIST_PRICE,
        C.CATEGORY_ID,
        C.CATEGORY_NAME
    ORDER BY C.PRODUCT_ID DESC
    OFFSET i_rowStart ROWS FETCH NEXT i_rowLength ROWS ONLY;

END SP_QUERYPRODUCTV4;
```

- 多筆查詢 (UDT)
  - RAW
    使用 UDT 傳入物件執行 WHERE IN 效果

```
#region [Query Condition]
foreach (var key in dicParams.Keys)
{
    switch (key)
    {
        case "CategoryId":
            sbSQL.AppendLine("AND PC.CATEGORY_ID = :CategoryId");
            dynamicParam.Add("CategoryId", dicParams["CategoryId"], OracleDbType.Int64, ParameterDirection.Input);
            break;

        case "CategoryIds":
            sbSQL.AppendLine("AND PC.CATEGORY_ID IN (SELECT * FROM TABLE(CAST(:CategoryIds AS TB_UDT_LONG)))");
            var aryUdtLong = ((IEnumerable<long>)dicParams["CategoryIds"])
                            .Select(e => new UdtLong() { LongData = e }).ToArray();
            dynamicParam.Add(
                "CategoryIds",
                new TbUdtLong { UdtLongData = aryUdtLong },
                OracleDbType.Object,
                ParameterDirection.Input,
                udtTypeName: "TB_UDT_LONG"
            );
            break;

        case "ProductName":
            sbSQL.AppendLine("AND LOWER(P.PRODUCT_NAME) LIKE :ProductName");
            dynamicParam.Add("ProductName", dicParams["ProductName"], OracleDbType.Varchar2, ParameterDirection.Input);
            break;
    }
}
#endregion
```

  - SP
    程式呼叫端：
    傳入 UDT 物件

```
public (int, IEnumerable<QueryProductRP>) QueryProductV5(Dictionary<string, object> dicParams)
{
    try
    {
        var dynamicParam = new OracleDynamicParameters();
        dynamicParam.Add("i_categoryId", dicParams.GetValueOrDefault("CategoryId"), OracleDbType.Int64, ParameterDirection.Input);

        var aryUdtLong = ((IEnumerable<long>)dicParams.GetValueOrDefault("CategoryIds"))
                        ?.Select(e => new UdtLong() { LongData = e }).ToArray();
        dynamicParam.Add(
            "i_categoryIds",
            aryUdtLong == null ? null : new TbUdtLong { UdtLongData = aryUdtLong },
            OracleDbType.Object,
            ParameterDirection.Input,
            udtTypeName: "TB_UDT_LONG"
        );

        dynamicParam.Add("i_productName", dicParams.GetValueOrDefault("ProductName"), OracleDbType.Varchar2, ParameterDirection.Input);
        dynamicParam.Add("i_rowStart", dicParams.GetValueOrDefault("RowStart"), OracleDbType.Int32, ParameterDirection.Input);
        dynamicParam.Add("i_rowLength", dicParams.GetValueOrDefault("RowLength"), OracleDbType.Int32, ParameterDirection.Input);
        dynamicParam.Add("o_result", dbType: OracleDbType.RefCursor, direction: ParameterDirection.Output);

        var res = dbConnection.Query<QueryProductV5RP>(
            "SP_QUERYPRODUCTV5",
            dynamicParam,
            commandType: CommandType.StoredProcedure
        );

        return (res.FirstOrDefault()?.TotalCnt ?? 0, res);
    }
    catch
    {
        throw;
    }
}
```

    SP執行端：

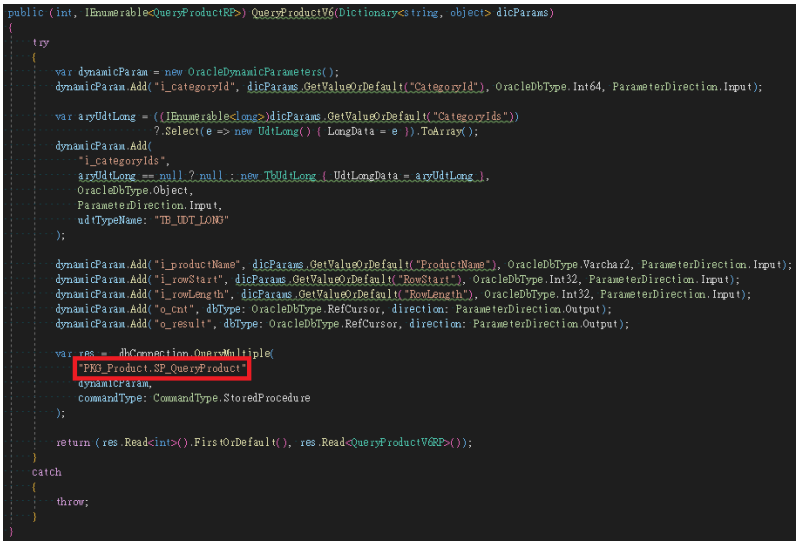
使用 UDT 物件執行 WHERE IN 效果

```
create or replace PROCEDURE SP_QUERYPRODUCTV5
    i_categoryId NUMBER DEFAULT NULL,
    i_categoryIds TB_UDT_LONG DEFAULT NULL,
    i_productName VARCHAR2 DEFAULT NULL,
    i_rowStart NUMBER DEFAULT 0,
    i_rowLength NUMBER DEFAULT 10,
    o_result OUT SYS_REFCURSOR
)
AS
BEGIN

    OPEN o_result FOR
    WITH CTE AS
    (
        SELECT
            P.PRODUCT_ID,
            P.PRODUCT_NAME,
            P.DESCRIPTION,
            P.STANDARD_COST,
            P.LIST_PRICE,
            PC.CATEGORY_ID,
            PC.CATEGORY_NAME
        FROM
            PRODUCTS P
        INNER JOIN
            PRODUCT_CATEGORIES PC
                ON P.CATEGORY_ID = PC.CATEGORY_ID
        WHERE 1 = 1
        AND
        (
            (i_categoryId IS NULL)
            OR
            (PC.CATEGORY_ID = i_categoryId)
        )
        AND
        (
            (i_categoryIds IS NULL)
            OR
            PC.CATEGORY_ID IN (SELECT * FROM TABLE(CAST(i_categoryIds AS TB_UDT_LONG)))
        )
```

- 多筆查詢 (Package)
  - SP

    程式呼叫端：

    呼叫 Package 內的 StoredProcedure



```
public (int, IEnumerable<QueryProductRP>) QueryProductV6(Dictionary<string, object> dicParams)
{
    try
    {
        var dynamicParam = new OracleDynamicParameters();
        dynamicParam.Add("i_categoryId", dicParams.GetValueOrDefault("CategoryId"), OracleDbType.Int64, ParameterDirection.Input);

        var aryUdtLong = ((IEnumerable<long>)dicParams.GetValueOrDefault("CategoryIds"))
                        ?.Select(e => new UdtLong() { LongData = e }).ToArray();
        dynamicParam.Add(
            "i_categoryIds",
            aryUdtLong == null ? null : new TbUdtLong { UdtLongData = aryUdtLong },
            OracleDbType.Object,
            ParameterDirection.Input,
            udtTypeName: "TB_UDT_LONG"
        );

        dynamicParam.Add("i_productName", dicParams.GetValueOrDefault("ProductName"), OracleDbType.Varchar2, ParameterDirection.Input);
        dynamicParam.Add("i_rowStart", dicParams.GetValueOrDefault("RowStart"), OracleDbType.Int32, ParameterDirection.Input);
        dynamicParam.Add("i_rowLength", dicParams.GetValueOrDefault("RowLength"), OracleDbType.Int32, ParameterDirection.Input);
        dynamicParam.Add("o_cnt", dbType: OracleDbType.RefCursor, direction: ParameterDirection.Output);
        dynamicParam.Add("o_result", dbType: OracleDbType.RefCursor, direction: ParameterDirection.Output);

        var res = dbConnection.QueryMultiple(
            "PKG_Product.SP_QueryProduct"
            dynamicParam,
            commandType: CommandType.StoredProcedure
        );

        return (res.Read<int>().FirstOrDefault(), res.Read<QueryProductV6RP>());
    }
    catch
    {
        throw;
    }
}
```

    SP執行端：

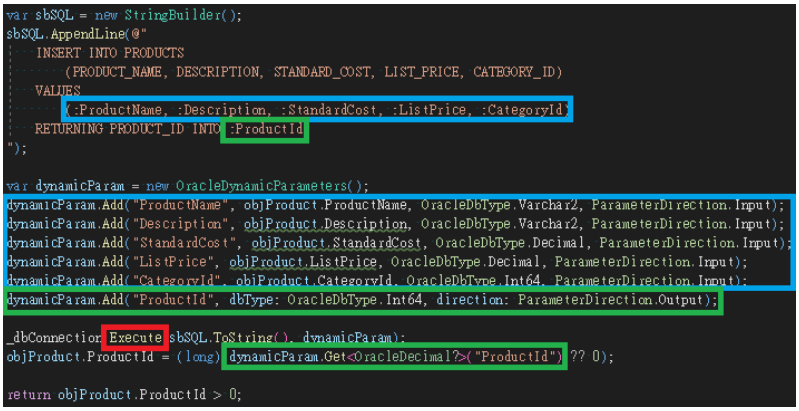
    使用 Package 內定義 Type 變數，將共用邏輯查詢結果存放在此變數中

```
create or replace PACKAGE PKG_Product AS

    TYPE OBJ_Product IS RECORD (
        PRODUCT_ID NUMBER,
        PRODUCT_NAME VARCHAR2(255),
        DESCRIPTION VARCHAR2(2000),
        STANDARD_COST NUMBER(9,2),
        LIST_PRICE NUMBER(9,2),
        CATEGORY_ID NUMBER,
        CATEGORY_NAME VARCHAR2(255)
    );
    TYPE TB_Product IS TABLE OF OBJ_Product;

    PROCEDURE SP_QueryProduct(
        i_categoryId NUMBER DEFAULT NULL,
        i_categoryIds TB_UDT_LONG DEFAULT NULL,
        i_productName VARCHAR2 DEFAULT NULL,
        i_rowStart NUMBER DEFAULT 0,
        i_rowLength NUMBER DEFAULT 10,
        o_cnt OUT SYS_REFCURSOR,
        o_result OUT SYS_REFCURSOR
    );

END PKG_Product;
```

```
create or replace PACKAGE BODY PKG_Product AS

    PROCEDURE SP_QueryProduct(
        i_categoryId NUMBER DEFAULT NULL,
        i_categoryIds TB_UDT_LONG DEFAULT NULL,
        i_productName VARCHAR2 DEFAULT NULL,
        i_rowStart NUMBER DEFAULT 0,
        i_rowLength NUMBER DEFAULT 10,
        o_cnt OUT SYS_REFCURSOR,
        o_result OUT SYS_REFCURSOR
    )
    AS
        v_TempTable TB_Product;
    BEGIN
        SELECT
            P.PRODUCT_ID,
            P.PRODUCT_NAME,
            P.DESCRIPTION,
            P.STANDARD_COST,
            P.LIST_PRICE,
            PC.CATEGORY_ID,
            PC.CATEGORY_NAME
        BULK COLLECT INTO
            v_TempTable
        FROM
            PRODUCTS P
        INNER JOIN
            PRODUCT_CATEGORIES PC
                ON P.CATEGORY_ID = PC.CATEGORY_ID
        WHERE 1 = 1
        AND
        (
            (i_categoryId IS NULL)
            OR
            (PC.CATEGORY_ID = i_categoryId)
        )
        AND
        (
            (i_categoryIds IS NULL)
            OR
            PC.CATEGORY_ID IN (SELECT * FROM TABLE(CAST(i_categoryIds AS TB_UDT_LONG)))
        )
        AND
        (
            (i_productName IS NULL)
            OR
            (LOWER(P.PRODUCT_NAME) LIKE i_productName)
        );

        OPEN o_cnt FOR
        SELECT
            COUNT(1)
        FROM
            TABLE(v_TempTable);

        OPEN o_result FOR
        SELECT
            R.*,
            COALESCE(SUM(I.QUANTITY), 0) AS QUANTITY
        FROM
            TABLE(v_TempTable) R
        LEFT JOIN
            INVENTORIES I
                ON R.PRODUCT_ID = I.PRODUCT_ID
        GROUP BY
            R.PRODUCT_ID,
            R.PRODUCT_NAME,
            R.DESCRIPTION,
            R.STANDARD_COST,
            R.LIST_PRICE,
            R.CATEGORY_ID,
            R.CATEGORY_NAME
        ORDER BY R.PRODUCT_ID DESC
        OFFSET i_rowStart ROWS FETCH NEXT i_rowLength ROWS ONLY;
    END SP_QueryProduct;

END PKG_Product;
```

- 新增
  - RAW

    使用 Out 變數接收 Identity 自動產生的 PKey

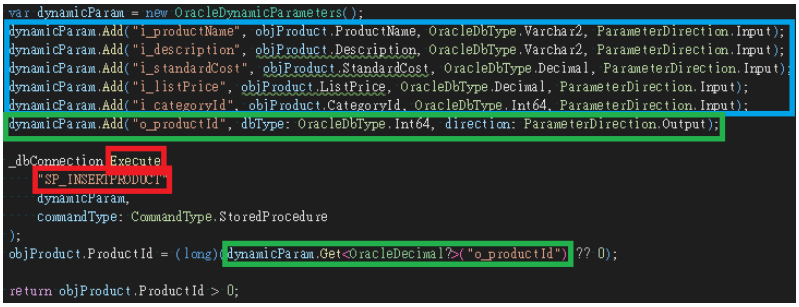


```
var sbSQL = new StringBuilder();
sbSQL.AppendLine(@"
    INSERT INTO PRODUCTS
        (PRODUCT_NAME, DESCRIPTION, STANDARD_COST, LIST_PRICE, CATEGORY_ID)
    VALUES
        (:ProductName, :Description, :StandardCost, :ListPrice, :CategoryId)
    RETURNING PRODUCT_ID INTO :ProductId
");

var dynamicParam = new OracleDynamicParameters();
dynamicParam.Add("ProductName", objProduct.ProductName, OracleDbType.Varchar2, ParameterDirection.Input);
dynamicParam.Add("Description", objProduct.Description, OracleDbType.Varchar2, ParameterDirection.Input);
dynamicParam.Add("StandardCost", objProduct.StandardCost, OracleDbType.Decimal, ParameterDirection.Input);
dynamicParam.Add("ListPrice", objProduct.ListPrice, OracleDbType.Decimal, ParameterDirection.Input);
dynamicParam.Add("CategoryId", objProduct.CategoryId, OracleDbType.Int64, ParameterDirection.Input);
dynamicParam.Add("ProductId", dbType: OracleDbType.Int64, direction: ParameterDirection.Output);

_dbConnection.Execute(sbSQL.ToString(), dynamicParam);
objProduct.ProductId = (long)(dynamicParam.Get<OracleDecimal?>("ProductId") ?? 0);

return objProduct.ProductId > 0;
```

  - SP

    程式呼叫端：

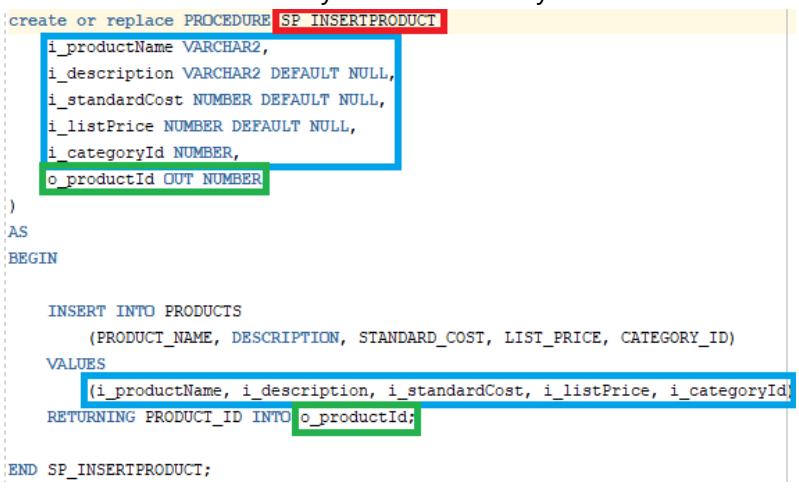
    傳入 Input / Output 參數



```
var dynamicParam = new OracleDynamicParameters();
dynamicParam.Add("i_productName", objProduct.ProductName, OracleDbType.Varchar2, ParameterDirection.Input);
dynamicParam.Add("i_description", objProduct.Description, OracleDbType.Varchar2, ParameterDirection.Input);
dynamicParam.Add("i_standardCost", objProduct.StandardCost, OracleDbType.Decimal, ParameterDirection.Input);
dynamicParam.Add("i_listPrice", objProduct.ListPrice, OracleDbType.Decimal, ParameterDirection.Input);
dynamicParam.Add("i_categoryId", objProduct.CategoryId, OracleDbType.Int64, ParameterDirection.Input);
dynamicParam.Add("o_productId", dbType: OracleDbType.Int64, direction: ParameterDirection.Output);

_dbConnection.Execute(
    "SP_INSERTPRODUCT",
    dynamicParam,
    commandType: CommandType.StoredProcedure
);
objProduct.ProductId = (long)(dynamicParam.Get<OracleDecimal?>("o_productId") ?? 0);

return objProduct.ProductId > 0;
```

    SP執行端：

    使用 Out 變數接收 Identity 自動產生的 PKey



```
create or replace PROCEDURE SP_INSERTPRODUCT
(
    i_productName VARCHAR2,
    i_description VARCHAR2 DEFAULT NULL,
    i_standardCost NUMBER DEFAULT NULL,
    i_listPrice NUMBER DEFAULT NULL,
    i_categoryId NUMBER,
    o_productId OUT NUMBER
)
AS
BEGIN

    INSERT INTO PRODUCTS
        (PRODUCT_NAME, DESCRIPTION, STANDARD_COST, LIST_PRICE, CATEGORY_ID)
    VALUES
        (i_productName, i_description, i_standardCost, i_listPrice, i_categoryId)
    RETURNING PRODUCT_ID INTO o_productId;

END SP_INSERTPRODUCT;
```

- 修改
  - RAW

```
var sbSQL = new StringBuilder();
sbSQL.AppendLine(@"
    UPDATE PRODUCTS SET
        PRODUCT_NAME = :ProductName,
        DESCRIPTION = :Description,
        STANDARD_COST = :StandardCost,
        LIST_PRICE = :ListPrice,
        CATEGORY_ID = :CategoryId
    WHERE
        PRODUCT_ID = :ProductId
");

var dynamicParam = new OracleDynamicParameters();
dynamicParam.Add("ProductId", objProduct.ProductId, OracleDbType.Int64, ParameterDirection.Input);
dynamicParam.Add("ProductName", objProduct.ProductName, OracleDbType.Varchar2, ParameterDirection.Input);
dynamicParam.Add("Description", objProduct.Description, OracleDbType.Varchar2, ParameterDirection.Input);
dynamicParam.Add("StandardCost", objProduct.StandardCost, OracleDbType.Decimal, ParameterDirection.Input);
dynamicParam.Add("ListPrice", objProduct.ListPrice, OracleDbType.Decimal, ParameterDirection.Input);
dynamicParam.Add("CategoryId", objProduct.CategoryId, OracleDbType.Int64, ParameterDirection.Input);

return _dbConnection.Execute(sbSQL.ToString(), dynamicParam) > 0;
```

  - SP

    程式呼叫端：

    傳入 Input 參數

```
var dynamicParam = new OracleDynamicParameters();
dynamicParam.Add("i_productId", objProduct.ProductId, OracleDbType.Int64, ParameterDirection.Input);
dynamicParam.Add("i_productName", objProduct.ProductName, OracleDbType.Varchar2, ParameterDirection.Input);
dynamicParam.Add("i_description", objProduct.Description, OracleDbType.Varchar2, ParameterDirection.Input);
dynamicParam.Add("i_standardCost", objProduct.StandardCost, OracleDbType.Decimal, ParameterDirection.Input);
dynamicParam.Add("i_listPrice", objProduct.ListPrice, OracleDbType.Decimal, ParameterDirection.Input);
dynamicParam.Add("i_categoryId", objProduct.CategoryId, OracleDbType.Int64, ParameterDirection.Input);

_dbConnection.Execute
    "SP_UPDATEPRODUCT"
    dynamicParam,
    commandType: CommandType.StoredProcedure
);
```

    SP執行端：

```
create or replace PROCEDURE SP_UPDATEPRODUCT
(
    i_productId NUMBER,
    i_productName VARCHAR2,
    i_description VARCHAR2 DEFAULT NULL,
    i_standardCost NUMBER DEFAULT NULL,
    i_listPrice NUMBER DEFAULT NULL,
    i_categoryId NUMBER
)
AS
BEGIN

    UPDATE PRODUCTS SET
        PRODUCT_NAME = i_productName,
        DESCRIPTION = i_description,
        STANDARD_COST = i_standardCost,
        LIST_PRICE = i_listPrice,
        CATEGORY_ID = i_categoryId
    WHERE
        PRODUCT_ID = i_productId;

END SP_UPDATEPRODUCT;
```

- 刪除
  - RAW

```
var sbSQL = new StringBuilder();
sbSQL.AppendLine(@"
    DELETE FROM
        PRODUCTS
    WHERE
        PRODUCT_ID IN :ProductIds
");

var dynamicParam = new OracleDynamicParameters();
dynamicParam.AddDynamicParams(new { ProductIds = ids });

return _dbConnection.Execute(sbSQL.ToString(), dynamicParam) > 0;
```

  - SP

    程式呼叫端：

    傳入 Input 參數

```
var dynamicParam = new OracleDynamicParameters();
dynamicParam.Add("i_productIds", string.Join(",", ids), OracleDbType.Varchar2, ParameterDirection.Input);

_dbConnection.Execute(
    "SP_DELETEPRODUCT",
    dynamicParam,
    commandType: CommandType.StoredProcedure
);
```

    SP執行端：

    INSTR函式執行 WHERE IN 效果

```
create or replace PROCEDURE SP_DELETEPRODUCT
    i_productIds VARCHAR2
)
AS
BEGIN

    DELETE FROM
        PRODUCTS
    WHERE
        INSTR(','|| i_productIds ||',', ','||PRODUCT_ID||',') > 0;

END SP_DELETEPRODUCT;
```

- 新增 (Transaction)
  - RAW

    使用 BeginTransaction 將兩段執行語句包在一起，確保交易一致性

```
try
{
    var sbCategory = new StringBuilder();
    sbCategory.AppendLine(@"
        INSERT INTO PRODUCT_CATEGORIES
            (CATEGORY_NAME)
        VALUES
            (:CategoryName)
        RETURNING CATEGORY_ID INTO :CategoryId
    ");

    var dpCategory = new OracleDynamicParameters();
    dpCategory.Add("CategoryName", objProductCategory.CategoryName, OracleDbType.Varchar2, ParameterDirection.Input);
    dpCategory.Add("CategoryId", dbType: OracleDbType.Int64, direction: ParameterDirection.Output);

    var sbProduct = new StringBuilder();
    sbProduct.AppendLine(@"
        INSERT INTO PRODUCTS
            (PRODUCT_NAME, DESCRIPTION, STANDARD_COST, LIST_PRICE, CATEGORY_ID)
        VALUES
            (:ProductName, :Description, :StandardCost, :ListPrice, :CategoryId)
        RETURNING PRODUCT_ID INTO :ProductId
    ");

    var dpProduct = new OracleDynamicParameters();
    dpProduct.Add("ProductName", objProduct.ProductName, OracleDbType.Varchar2, ParameterDirection.Input);
    dpProduct.Add("Description", objProduct.Description, OracleDbType.Varchar2, ParameterDirection.Input);
    dpProduct.Add("StandardCost", objProduct.StandardCost, OracleDbType.Decimal, ParameterDirection.Input);
    dpProduct.Add("ListPrice", objProduct.ListPrice, OracleDbType.Decimal, ParameterDirection.Input);
    dpProduct.Add("ProductId", dbType: OracleDbType.Int64, direction: ParameterDirection.Output);

    _dbConnection.Open();
    using (var tran = _dbConnection.BeginTransaction())
    {
        try
        {
            _dbConnection.Execute(sbCategory.ToString(), dpCategory, tran);
            objProductCategory.CategoryId = (long)dpCategory.Get<OracleDecimal>("CategoryId");

            objProduct.CategoryId = objProductCategory.CategoryId;
            dpProduct.Add("CategoryId", objProduct.CategoryId, OracleDbType.Int64, ParameterDirection.Input);

            _dbConnection.Execute(sbProduct.ToString(), dpProduct, tran);
            objProduct.ProductId = (long)dpProduct.Get<OracleDecimal>("ProductId");

            tran.Commit();
        }
        catch
        {
            tran.Rollback();
            throw;
        }
    }

    return objProduct.ProductId > 0;
}
catch
{
    throw;
}
finally
{
    _dbConnection.Close();
}
```

  - SP

    程式呼叫端：

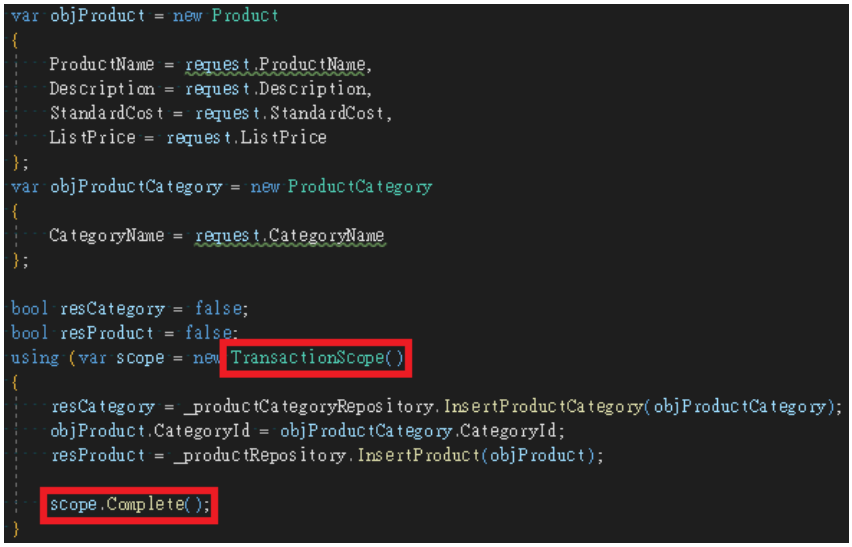
```
var dynamicParam = new OracleDynamicParameters();
dynamicParam.Add("i_categoryName", objProductCategory.CategoryName, OracleDbType.Varchar2, ParameterDirection.Input);
dynamicParam.Add("i_productName", objProduct.ProductName, OracleDbType.Varchar2, ParameterDirection.Input);
dynamicParam.Add("i_description", objProduct.Description, OracleDbType.Varchar2, ParameterDirection.Input);
dynamicParam.Add("i_standardCost", objProduct.StandardCost, OracleDbType.Decimal, ParameterDirection.Input);
dynamicParam.Add("i_listPrice", objProduct.ListPrice, OracleDbType.Decimal, ParameterDirection.Input);
dynamicParam.Add("o_productId", dbType: OracleDbType.Int64, direction: ParameterDirection.Output);

_dbConnection.Execute(
    "SP_INSERTPRODUCTANDCATEGORY"
    dynamicParam,
    commandType: CommandType.StoredProcedure
);
objProduct.ProductId = (long)(dynamicParam.Get<OracleDecimal?>("o_productId") ?? 0);

return objProduct.ProductId > 0;
```

    SP執行端：

使用 Begin Exception 區塊執行，確保交易一致性

```
create or replace PROCEDURE SP_INSERTPRODUCTANDCATEGORY
    i_categoryName Varchar2,
    i_productName VARCHAR2,
    i_description VARCHAR2 DEFAULT NULL,
    i_standardCost NUMBER DEFAULT NULL,
    i_listPrice NUMBER DEFAULT NULL,
    o_productId OUT NUMBER
)
AS
    v_categoryId NUMBER;
BEGIN

    BEGIN
        INSERT INTO PRODUCT_CATEGORIES
            (CATEGORY_NAME)
        VALUES
            (i_categoryName)
        RETURNING CATEGORY_ID INTO v_categoryId;

        INSERT INTO PRODUCTS
            (PRODUCT_NAME, DESCRIPTION, STANDARD_COST, LIST_PRICE, CATEGORY_ID)
        VALUES
            (i_productName, i_description, i_standardCost, i_listPrice, v_categoryId)
        RETURNING PRODUCT_ID INTO o_productId;

        COMMIT;
    EXCEPTION
        WHEN OTHERS THEN
            ROLLBACK;
            RAISE;
    END;

END SP_INSERTPRODUCTANDCATEGORY;
```

- 新增 (TransactionScope)
  使用 TransactionScope 將兩段執行語句包在一起，確保交易一致性



```
var objProduct = new Product
{
    ProductName = request.ProductName,
    Description = request.Description,
    StandardCost = request.StandardCost,
    ListPrice = request.ListPrice
};
var objProductCategory = new ProductCategory
{
    CategoryName = request.CategoryName
};

bool resCategory = false;
bool resProduct = false;
using (var scope = new TransactionScope())
{
    resCategory = _productCategoryRepository.InsertProductCategory(objProductCategory);
    objProduct.CategoryId = objProductCategory.CategoryId;
    resProduct = _productRepository.InsertProduct(objProduct);

    scope.Complete();
}
```

參考資料

- Docker Desktop 安裝
  https://www.runoob.com/docker/windows-docker-install.html
- Oracle 安裝/建立使用者
  https://medium.com/geekculture/run-oracle-database-21c-in-docker-351049344d0c
  https://hackmd.io/@CloudyWing/SJsMfcGMo
- Oracle 測試資料庫
  https://www.w3cschool.cn/oraclejc/oraclejc-gpa52qqy.html
- Oracle.ManagedDataAccess
  https://blog.yowko.com/dapper-oracle-manageddataaccess/
  https://www.c-sharpcorner.com/article/net-core-with-oracle-database-using-dapper/
- OracleDynamicParameter
  https://blog.darkthread.net/blog/dapper-ref-cursor/
  https://gist.github.com/vijayganeshpk/3096151
- Dapper Column Mapping
  https://dotblogs.com.tw/supershowwei/2016/08/16/175753
- Dapper ColumnAttributeTypeMapper
  https://stackoverflow.com/questions/20951531/dapper-with-attributes-mapping/20969521#20969521
  https://gist.github.com/kalebpederson/5460509
- Dapper QueryMultiple
  https://www.cnblogs.com/51net/p/16069796.html
  https://www.cnblogs.com/51net/p/16069684.html
- Dapper Transaction / TransactionScope
  https://stackoverflow.com/questions/44118876/which-transaction-is-better-with-dapper-begin-tran-or-transactionscope
  https://stackoverflow.com/questions/6860979/dapper-transactionscope
  https://kevintsengtw.blogspot.com/2015/09/dapper_28.html
- Oracle Cursor
  https://godleon.blogspot.com/2008/12/oracle-cursor-cursor-variable.html
- Oracle Stored Procedure
  https://matthung0807.blogspot.com/2017/08/oracle-stored-procedures.html
  https://medium.com/jimmy-wang/oracle%E5%9F%BA%E6%9C%AC%E4%BF%AE%E7%B7%B4-pl-sql-stored-procedures-and-functions-73b235631880
- Oracle Temporary Table / Collection
  https://stackoverflow.com/questions/41689135/how-to-create-a-temporary-table-inside-stored-procedure-before-opening-a-cursor

- Oracle Temporary Table
  https://blog.darkthread.net/blog/odpnet-temp-table/
  https://www.cnblogs.com/kerrycode/p/3285936.html

- Oracle Collection
  https://godleon.blogspot.com/2008/12/oracle-collection.html
  https://godleon.blogspot.com/2008/12/oracle-record.html
  https://www.cnblogs.com/liuning8023/archive/2012/05/10/2495019.html
  https://blog.csdn.net/liangweiwei130/article/details/38223319

- Oracle Bulk Collect / ForAll
  https://blog.51cto.com/u_15127540/3343414
  https://www.itread01.com/articles/1478240721.html

- Oracle INSTR()
  https://www.sjkjc.com/oracle-ref/instr/
  https://tomkuo139.blogspot.com/2009/01/plsql-substring-string.html
  https://www.cnblogs.com/qinyios/p/11207981.html

- Oracle Count Over
  https://juejin.cn/s/oracle%20count%20over%20partition
  https://blog.csdn.net/sinat_19671783/article/details/73925833

- Oracle With As
  https://www.cnblogs.com/mingforyou/p/8295239.html
  https://dljeng.blogspot.com/2018/01/oracle-with.html

- Oracle UDT
  https://stackoverflow.com/questions/50694795/oracle-managed-driver-user-defined-types
  https://medium.com/oracledevs/announcing-odp-net-21-3-user-defined-types-for-managed-odp-net-and-odp-net-core-836fa81da3ba
  https://github.com/oracle/dotnet-db-samples/tree/master/samples/udt
  https://www.codeproject.com/Articles/1261430/Calling-Oracles-Stored-Procedures-and-Mapping-User

- Oracle Package
  https://matthung0807.blogspot.com/2017/09/oracle-package.html
  https://www.cnblogs.com/xqzt/p/4875891.html
  https://stackoverflow.com/questions/67622904/define-custom-type-in-package
  https://stackoverflow.com/questions/15420642/pass-udt-defined-in-a-package-as-a-parameter-to-stored-proc-in-oracle

- Oracle Transaction
  https://dba.stackexchange.com/questions/84769/commit-and-rollback-in-oracle-stored-procedure
  https://blog.csdn.net/chenglaozong/article/details/85265337