114年智慧行動商務系統開發班第1期

資料庫應用與開發實務學習評量

|  |  |  |  |
| --- | --- | --- | --- |
| **評量編號** | 2-3 | | |
| **評量主題** | SQL語法及T-SQL程式設計 | **實施時間** | 2025/7/18 |
| **評量形式** | 作業紀錄 | **評量日期** | 2025/7/31 |
| **評量目的** | 評定學員是否具備下列能力：   1. 撰寫各種SQL DML、SQL DDL語法之能力。 2. 使用T-SQL語言撰寫出Database、Table、Stored Procedure、Function等物件之能力。 | | |
| **評量說明** | **請先用題目給的原始資料來源檔建立指定之資料庫後，完成以下任務。**  **任務一：**依據題目要求，撰寫相對應的SQL DML Script。  **任務二：**依據題目要求，撰寫相對應的合併查詢及子查詢SQL Script。  **任務三：**依據題目要求，撰寫SQL DDL Script。  **任務四：**依據題目要求，撰寫預存程序(Stored Procedure)。  **任務五：**依據題目要求，撰寫自訂函數(Function)。 | | |
| **提交要求** | 繳交時請以FTP上傳**本作業文件檔及SQL Script原始檔**各一份，**SQL Script的副檔名請存為 .sql**，並將所有檔案放置於「完成結果檔」資料夾內，資料夾名稱請用**兩碼學號與姓名**，例如1號王小明，資料夾請命名為「01王小明」。 | | |

座號：24 姓名：鍾惟安

**請利用Northwind.sql的檔案建立本評量要使用的資料庫Northwind**

**任務一：依據題目要求，撰寫相對應的SQL DML Script。**

**利用Northwind資料庫，寫出可以查出下列資料的SQL DML語法**

1. **基本查詢**
2. 在【員工】資料表中找出所有在1993年(含)以後到職的資料。

select \* from Employees where year(HireDate) >= 1993

1. 在【訂單】資料表找出送貨郵遞區號為44087與05022及82520的資料。

select \* from Orders where ShipPostalCode in ('44087','05022','82520')

1. 在【產品】資料表中找出庫存量最多的前6名資料記錄。

select top 6 \* from Products order by UnitsInStock desc

1. 在【訂單】資料表中找出尚未有送貨日期的記錄資料。

select \* from Orders where ShippedDate is null

1. 在【訂單明細】資料表中找出訂購的產品數量介於20~40件的資料記錄。

select OrderID, SUM(Quantity) as Total from OrderDetails group by OrderID having sum(Quantity) between 20 and 40

**(二) 統計查詢**

1. 計算【產品】資料表中類別號為2的產品資料平均單價。

select avg(UnitPrice) 平均單價 from Products where CategoryID = 2

1. 在【產品】資料表中找出庫存量小於安全存量，且尚未進行採購的產品資料記錄。

select \* from Products where UnitsInStock < ReorderLevel and UnitsOnOrder = 0

1. 在【訂單明細】資料表找出訂單中包含超過5種產品的資料記錄。

select OrderID from OrderDetails group by OrderID having count(\*) > 5

1. 在【訂單明細】資料表中顯示出訂單號碼10263所有產品的價格小計。

select ProductID, UnitPrice, Quantity, Discount, UnitPrice \* Quantity \* (1 - Discount) as 小計 from OrderDetails where OrderID = 10263

1. 利用【產品】資料表資料，統計出每一個供應商各提供了幾樣產品。

select SupplierID, count(\*) 提供產品數量 from Products group by SupplierID

1. 利用【訂單明細】資料表資料，統計出各項商品的平均單價與平均銷售數量，並列出平均銷售數量大於10的資料，且將資料依產品編號由小到大排序。

select ProductID, avg(UnitPrice) 平均單價, avg(Quantity) 平均銷售數量 from OrderDetails group by ProductID having avg(Quantity) > 10 order by ProductID

**任務二：依據題目要求，撰寫相對應的合併查詢及子查詢SQL Script。**

**利用Northwind資料庫，寫出可以查出下列資料的SQL DML語法**

1.請試寫一合併查詢，查詢出訂購日期落在1996年7月並指定送貨公司為「United Package」的有訂單之訂貨明細資料，並列出訂單號碼、產品類別名稱、產品名稱、產品訂購單價、產品訂購數量、產品價錢小計、客戶編號、客戶名稱、收貨人名字、訂購日期、處理訂單員工的姓名、貨運公司、供應商名稱等資料項目，其中產品價錢小計請以四捨五入計算至整數位。

select OD.OrderID 訂單號碼, C.CategoryName 產品類別名稱, P.ProductName 產品名稱, OD.UnitPrice 產品訂購單價, OD.Quantity 產品訂購數量, round(OD.UnitPrice \* OD.Quantity \* (1 - OD.Discount), 0) as 產品價錢小計, O.CustomerID 客戶編號, CM.CompanyName 客戶名稱, O.ShipName 收貨人名字, O.OrderDate 訂購日期, E.FirstName + ' ' + E.LastName 處理訂單員工的姓名, S.CompanyName 貨運公司, SU.CompanyName 供應商名稱

from OrderDetails OD

join Orders O on O.OrderID = OD.OrderID

join Shippers S on S.ShipperID = O.ShipVia

join Products P on P.ProductID = OD.ProductID

join Categories C on C.CategoryID = P.CategoryID

join Customers CM on CM.CustomerID = O.CustomerID

join Employees E on E.EmployeeID = O.EmployeeID

join Suppliers SU on SU.SupplierID = P.SupplierID

where

S.CompanyName = 'United Package' and

O.OrderDate between '1996-07-01' and '1996-07-31'

2. 請利用exists運算子配合子查詢式，找出哪些客戶從未下過訂單，並列出客戶的所有欄位。(不可用到in運算，亦不可用合併查詢式)

select \* from Customers C where not exists (select \* from Orders O where O.CustomerID = C.CustomerID)

3. 請利用in運算子配合子查詢式，查詢哪些員工有處理過訂單，並列出員工的員工編號、姓名、職稱、內部分機號碼、附註欄位。(不可用到exists運算，亦不可用合併查詢式)

select E.EmployeeID 員工編號, E.FirstName+' '+E.LastName 姓名, E.Title 職稱, E.Extension 內部分機號碼, E.Notes 附註欄位 from Employees E where E.EmployeeID in

(select O.EmployeeID from Orders O)

4. 請合併查詢與子查詢兩種寫法，列出1998年度所有被訂購過的產品資料，並列出產品的所有欄位，且依產品編號由小到大排序。(寫合併查詢時不得用任何子查詢式，寫子查詢時不得用任何合併查詢式)

合併查詢：

select distinct P.\* from Products P

join OrderDetails OD on OD.ProductID = P.ProductID

join Orders O on O.OrderID = OD.OrderID

where year(O.OrderDate) = 1998

order by P.ProductID

子查詢：

select \* from Products where ProductID in

(select ProductID from OrderDetails

where OrderID in

(select OrderID from Orders

where year(OrderDate) = 1998))

order by ProductID

**任務三：依據題目要求，撰寫SQL DDL Script。**

1. 請寫出建立一個名為【MySchool】資料庫的SQL DDL Script。

**DDL Script：**

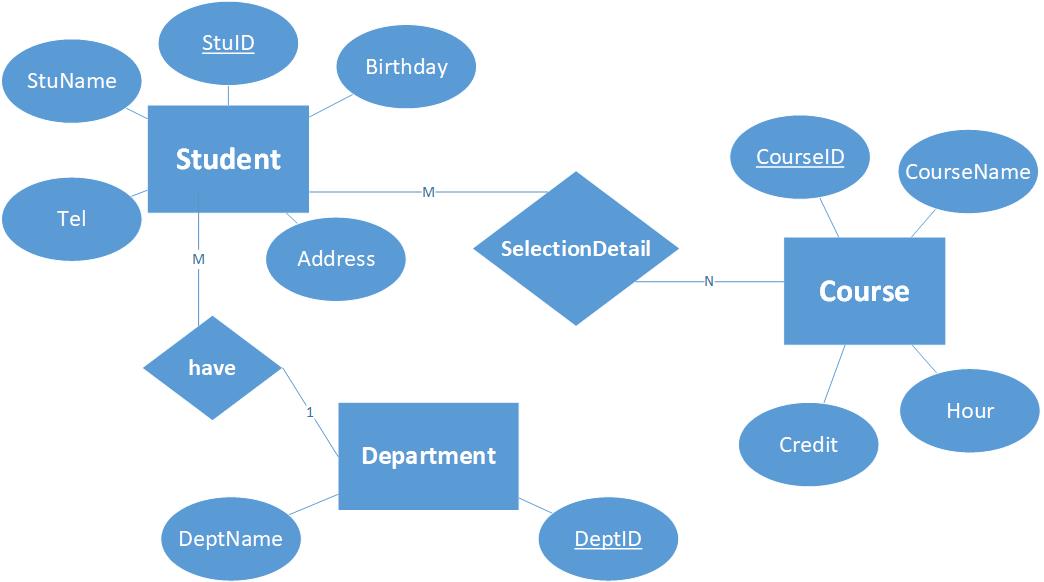
create database MySchool

go

use MySchool

go l

1. 請參考ER圖及下列資料表規格，寫出相對應之SQL DDL Script，使其可於【MySchool】資料庫中建立這些資料表。



**【Student】資料表**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **欄位** | **資料型態** | | **長度** | | **是否null** | **鍵值** | **備註** |
| StuID | nchar | | 10 | | N | P.K | S113518001 |
| StuName | nvarchar | | 20 | | N |  |  |
| Tel | nvarchar | | 20 | | N |  |  |
| Address | nvarchar | | 100 | | Y |  |  |
| Birthday | datetime | | － | | Y |  | 2000-4-8 |
| DeptID | | nchar | 1 | N | | F.K | Ref. Department.DeptID |

**DDL Script：**

create table Student(

StuID nchar(10) not null primary key,

StuName nvarchar(20) not null,

Tel nvarchar(20) not null,

[Address] nvarchar(100) null,

Birthday datetime null,

DeptID nchar(1) not null,

foreign key(DeptID) references Department(DeptID)

)

**【Course】資料表**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **欄位** | **資料型態** | **長度** | **是否null** | **鍵值** | **備註** |
| CourseID | nchar | 5 | N | P.K |  |
| CourseName | nvarchar | 30 | N |  |  |
| Credit | int | － | N |  | Default 0 |
| Hour | int | － | N |  | Default 2 |
| DeptID | nchar | 1 | N | F.K | Ref. Department.DeptID |

**DDL Script：**

create table Course(

CourseID nchar(5) not null primary key,

CourseName nvarchar(30) not null,

Credit int not null default 0,

Hour int not null default 2,

DeptID nchar(1) not null,

foreign key(DeptID) references Department(DeptID)

)

**【SelectionDetail】資料表**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **欄位** | **資料型態** | **長度** | **是否null** | **鍵值** | **備註** |
| StuID | nchar | 10 | N | P.F.K | ref. Student.StuID |
| CourseID | nchar | 5 | N | P.F.K | ref. Course.CourseID |
| Year | int | － | N |  | Default (Year(Getdate())) |
| Term | tinyint | － | N |  |  |
| Score | int | － | N |  | Default 0 |

**DDL Script：**

create table SelectionDetail(

StuID nchar(10) not null,

CourseID nchar(5) not null,

[Year] int not null default Year(Getdate()),

Term tinyint not null,

Score int not null default 0,

primary key (StuID, CourseID), --複合主鍵

foreign key (StuID) references Student(StuID),

foreign key (CourseID) references Course(CourseID)

)

**【Department】資料表**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **欄位** | **資料型態** | **長度** | **是否null** | **鍵值** | **備註** |
| DeptID | nchar | 1 | N | P.K | A-Z |
| DeptName | nvarchar | 30 | N | U.Q |  |

**DDL Script：**

create table Department(

DeptID nchar(1) not null primary key,

DeptName nvarchar(30) not null unique,

)

**任務四：依據題目要求，撰寫預存程序(Stored Procedure)**

請利用**任務三**所建立的【MySchool】資料庫，建立一個名為「InsertDeptmentData」的預存程序，新增科系資料時必須呼叫執行此預存程序進行資料新增。

1. 預存程序內需檢查科系代碼(DeptID)及科系名稱(DeptName)是否已在資料庫中。
2. 若欲新增的資料值檢查到已被使用，則輸出對應的錯誤訊息且不進行Insert動作。
3. 若欲新增的資料值皆未被使用，則進行Insert動作將資料寫入資料庫。

**程式碼：**

create or alter procedure InsertDepartmentData

@DeptID nchar(1), @DeptName nvarchar(30)

as

begin

print 'Check DeptID : '+@DeptID+', DeptName : '+@DeptName

--檢查科系代碼(DeptID)是否已在資料庫中

declare @DeptIDExists nchar(1)

select @DeptIDExists = count(DeptID) from Department where DeptID = @DeptID

if @DeptIDExists > 0

begin

print '科系代碼已存在，請使用其他代碼。'

return

end

print 'DeptID OK'

--檢查科系名稱(DeptName)是否已在資料庫中

declare @DeptNameExists nvarchar(30)

select @DeptNameExists = DeptName from Department where DeptName = @DeptName

if @DeptNameExists is not null

begin

print '科系名稱已存在，請使用其他名稱。'

return

end

print 'DeptName OK'

insert into Department values (@DeptID, @DeptName)

print '科系新增成功。'

end

exec InsertDepartmentData 'A', '資訊工程系'

**任務五：依據題目要求，撰寫自訂函數(Function)**

請利用**任務三**所建立的【MySchool】資料庫，建立一個名為「getCourseID」的自訂函數，其功能為新增課程資料時可呼叫此函數自動取得一個新的課程編號。CourseID的編碼規則為英文字母C加上開課科系代碼再加上3碼流水號。(例如科系代碼為G開的第123門課程，CourseID為CG123，科系代碼為B開的第1門課程，CourseID為CB001)。

**程式碼：**

--主Function

create or alter function getCourseID(@DeptID nchar(1))

returns nchar(5)

as

begin

--檢查有無該科系

declare @Dept int;

select top 1 @Dept = count(DeptID) from Department where DeptID = @DeptID;

--print (@Dept)

if @Dept = 0

--print ('無該科系');

return '00000';

--declare @DeptID nchar(5)

--set @DeptID = 'A'

--紀錄表內該科系的課程數

declare @CountDept int;

select @CountDept = count(\*) from Course where DeptID = @DeptID;

--print @CountDept

--print (CAST(@CountDept + 1 AS nchar(3)))

declare @NewCourseID nvarchar(5);--先宣告nvarchar，不然@DeptID因為nchar的固定長度會把後面全部都補上空白

set @NewCourseID = 'C' + rtrim(@DeptID) + RIGHT('000' + CAST(@CountDept + 1 AS nvarchar), 3);--使用 rtrim() 去除 DeptID 後面的空白

--print @NewCourseID

return @NewCourseID

end

--主執行的SP，負責插入資料，因為Function內不能執行Insert也不能呼叫SP

create or alter proc InsertNewCourse

@CourseName nvarchar(30), @Credit int, @Hour int, @DeptID nchar(1)

as

begin

--取課程ID

declare @NewCourseID nchar(5);

select @NewCourseID = (select dbo.getCourseID(@DeptID));

--print @NewCourseID;

if @NewCourseID = '00000'

begin

print '無該科系'

return

end

--檢查是否已有該課程

declare @ExistsCourse nvarchar(30) = null

select top 1@ExistsCourse = CourseName from Course where @CourseName like '%' + CourseName + '%';

if @ExistsCourse is not null

begin

print '已有該課程'

return

end

insert into Course values (@NewCourseID, @CourseName, @Credit, @Hour, @DeptID);

print '加入課程成功'

end

exec InsertNewCourse '大氣科學系', 0, 0, 'D';