功能說明：針對文字內容依起始位置及長度並依遮罩字元進行替換

SELECT [comm].[fn\_masking] (@str,@start,@len,@mask)

參數說明

@str：要遮罩文字

@start：遮罩起始位置(<=0表總長度+@start)

@len：遮罩長度(<=0遮罩到總長度+@len)

@mask：要遮罩的字元

**範例一：遮罩前面內容**

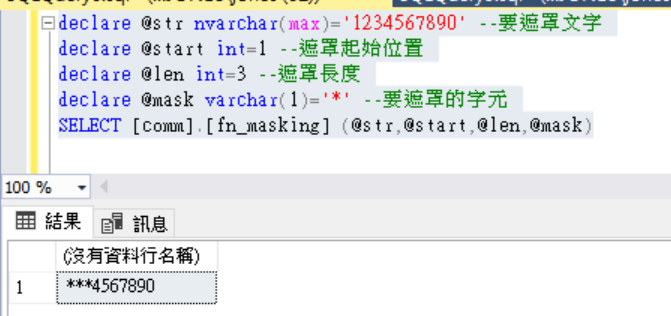
declare @str nvarchar(max)='1234567890' --要遮罩文字

declare @start int=1 --遮罩起始位置

declare @len int=3 --遮罩長度

declare @mask varchar(1)='\*' --要遮罩的字元

SELECT [comm].[fn\_masking] (@str,@start,@len,@mask)



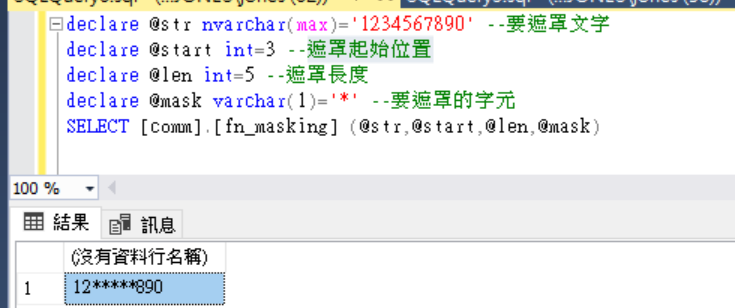
**範例二：遮罩中間內容**

declare @str nvarchar(max)='1234567890' --要遮罩文字

declare @start int=3 --遮罩起始位置

declare @len int=5 --遮罩長度

declare @mask varchar(1)='\*' --要遮罩的字元



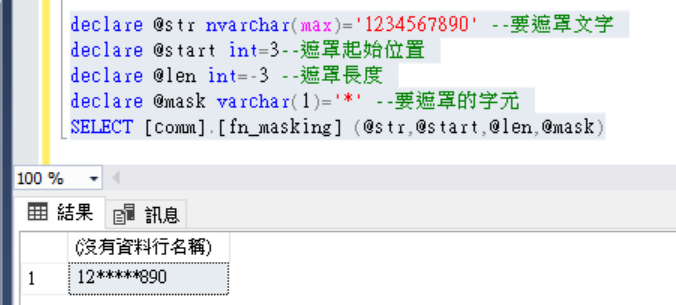
declare @str nvarchar(max)='1234567890' --要遮罩文字

declare @start int=3--遮罩起始位置

declare @len int=**-3** --往後取遮罩長度

declare @mask varchar(1)='\*' --要遮罩的字元

SELECT [comm].[fn\_masking] (@str,@start,@len,@mask)



**範例三：遮罩後面內容**

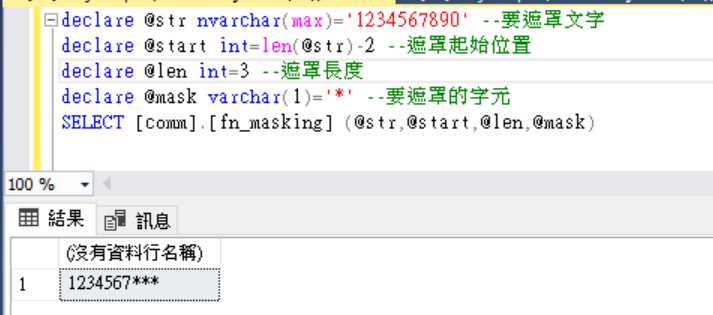
declare @str nvarchar(max)='1234567890' --要遮罩文字

declare @start int=len(@str)-2 --遮罩起始位置

declare @len int=3 --遮罩長度

declare @mask varchar(1)='\*' --要遮罩的字元

SELECT [comm].[fn\_masking] (@str,@start,@len,@mask)



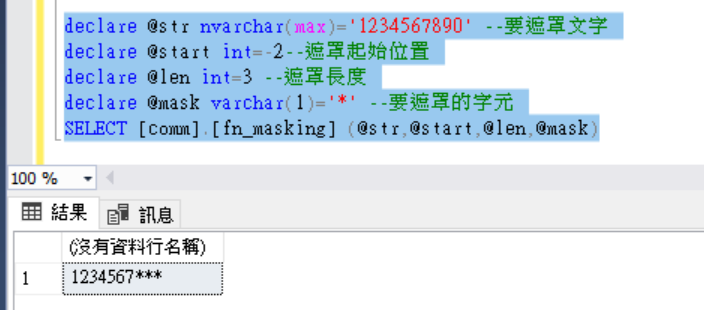
declare @str nvarchar(max)='1234567890' --要遮罩文字

declare @start int=-2 --遮罩倒數第2位置

declare @len int=3 --遮罩長度

declare @mask varchar(1)='\*' --要遮罩的字元

SELECT [comm].[fn\_masking] (@str,@start,@len,@mask)



**範例四：遮罩前後面內容**

declare @str nvarchar(max)='1234567890' --要遮罩文字

declare @start1 int=1 --遮罩起始位置

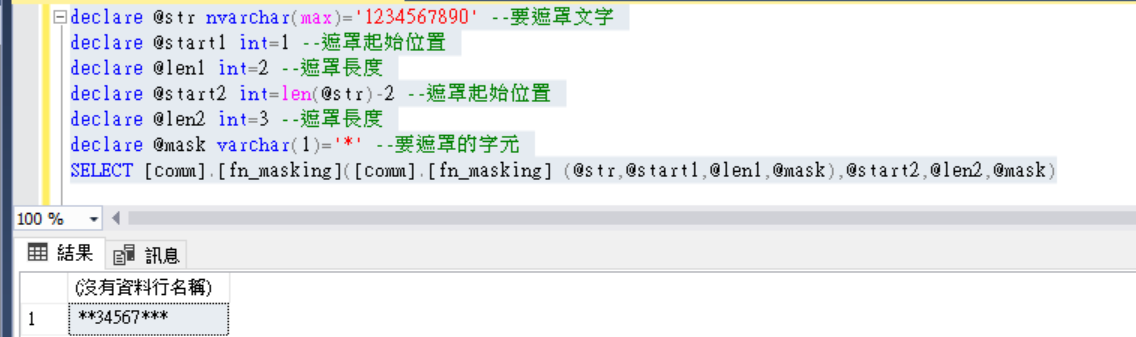
declare @len1 int=2 --遮罩長度

declare @start2 int=len(@str)-2 --遮罩起始位置

declare @len2 int=3 --遮罩長度

declare @mask varchar(1)='\*' --要遮罩的字元

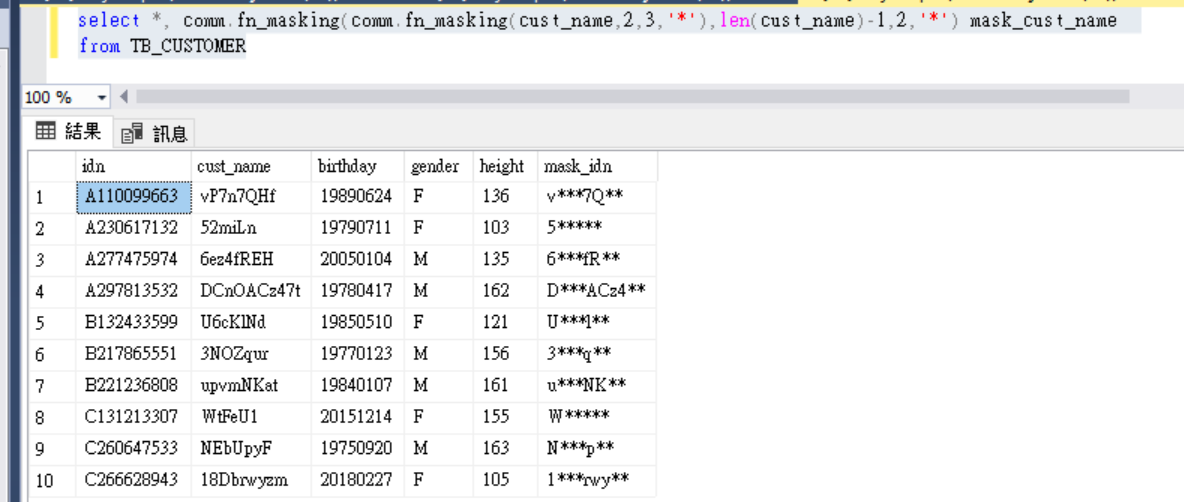
SELECT [comm].[fn\_masking]([comm].[fn\_masking] (@str,@start1,@len1,@mask),@start2,@len2,@mask)



**範例五：資料表複合遮罩欄位遮罩**

select \*, comm.fn\_masking(comm.fn\_masking(cust\_name,2,3,'\*'),len(cust\_name)-1,2,'\*') mask\_cust\_name

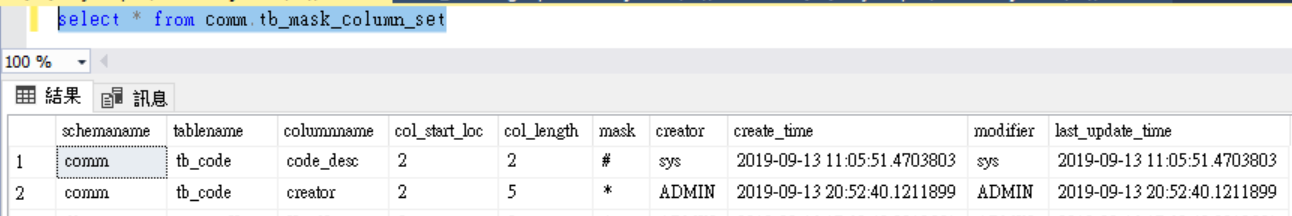
from TB\_CUSTOMER



**範例六：手動執行sql function產生具有遮罩欄位的insert sql語法**

首先先在comm.tb\_mask\_column\_set資料表建立欄位遮罩設定

select \* from comm.tb\_mask\_column\_set



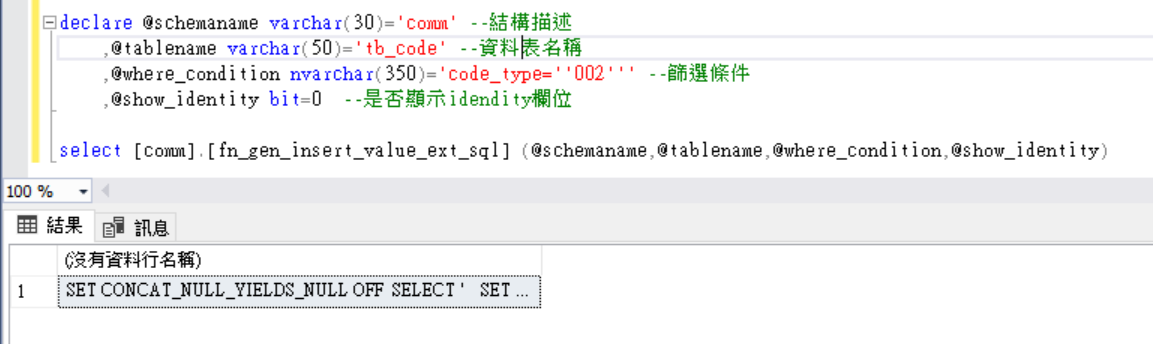
declare @schemaname varchar(30)='comm' --結構描述

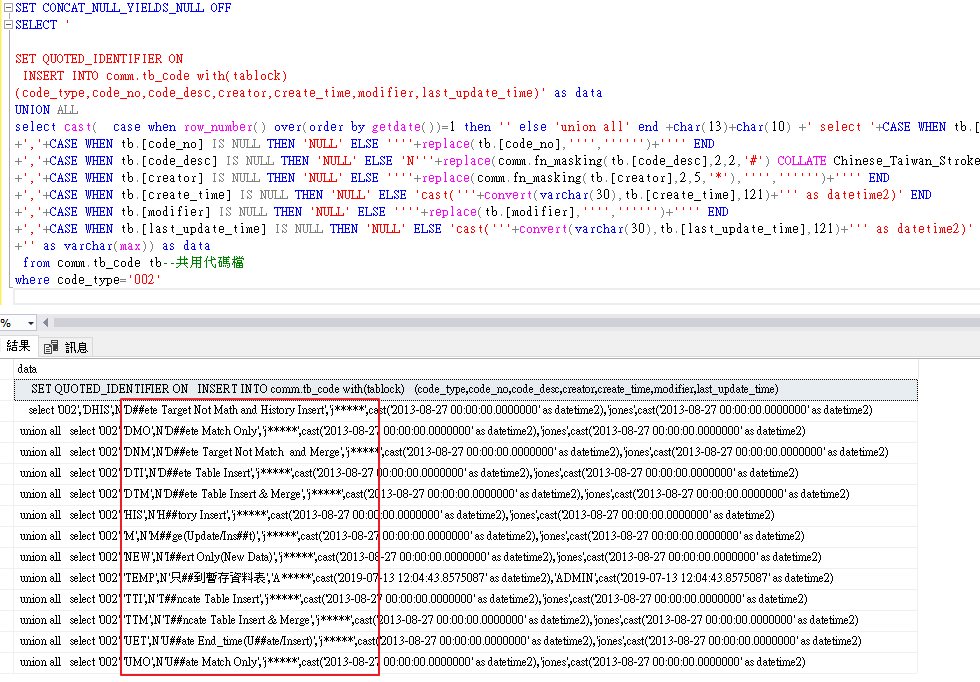
,@tablename varchar(50)='tb\_code' --資料表名稱

,@where\_condition nvarchar(350)='code\_type=''002''' --篩選條件

,@show\_identity bit=0 --是否顯示idendity欄位

select [comm].[fn\_gen\_insert\_value\_ext\_sql] (@schemaname,@tablename,@where\_condition,@show\_identity)

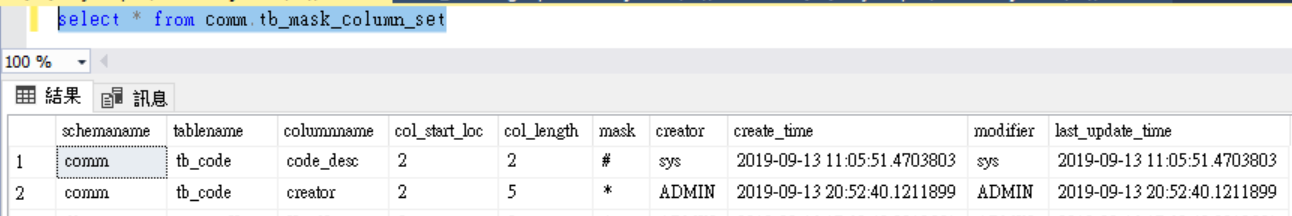




**範例七：手動執行sql function產生匯出具有遮罩欄位的資料表內容**

首先先在comm.tb\_mask\_column\_set資料表建立欄位遮罩設定

select \* from comm.tb\_mask\_column\_set



declare @schemaname varchar(30)='comm' --結構描述

,@tablename varchar(50)='tb\_code' --資料表名稱

,@where\_condition nvarchar(350)='code\_type=''002''' --篩選條件

,@field\_terminator varchar(2)=',' --欄位分隔符號

,@file\_format\_key varchar(10)='011|C' --BIG5

,@table\_alias varchar(10)='tb' --資料表別名

,@src\_export\_type\_key varchar(10)='008|T' --固定值

,@last\_data\_type\_key varchar(10)='009|ALL' --固定值

,@last\_data\_column\_name nvarchar(255)='' --固定值

,@program\_no varchar(10)='' --固定值

,@exec\_group varchar(30)='' --固定值

,@lang\_flag char(1)='' --固定值

SELECT [comm].[fn\_gen\_column\_concate\_ext\_sql] (@schemaname

,@tablename

,@where\_condition

,@field\_terminator

,@file\_format\_key

,@table\_alias

,@src\_export\_type\_key

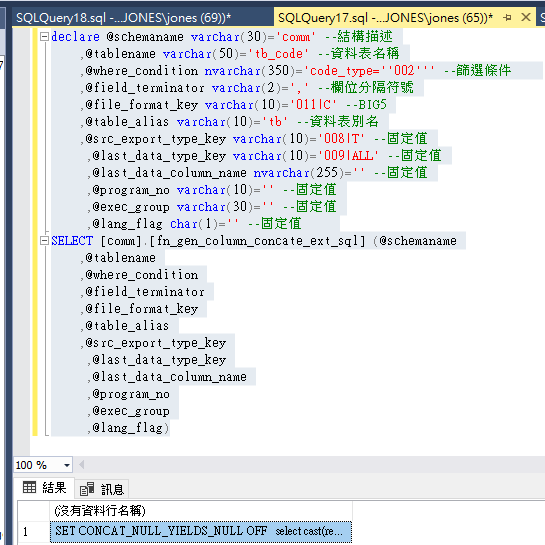
,@last\_data\_type\_key

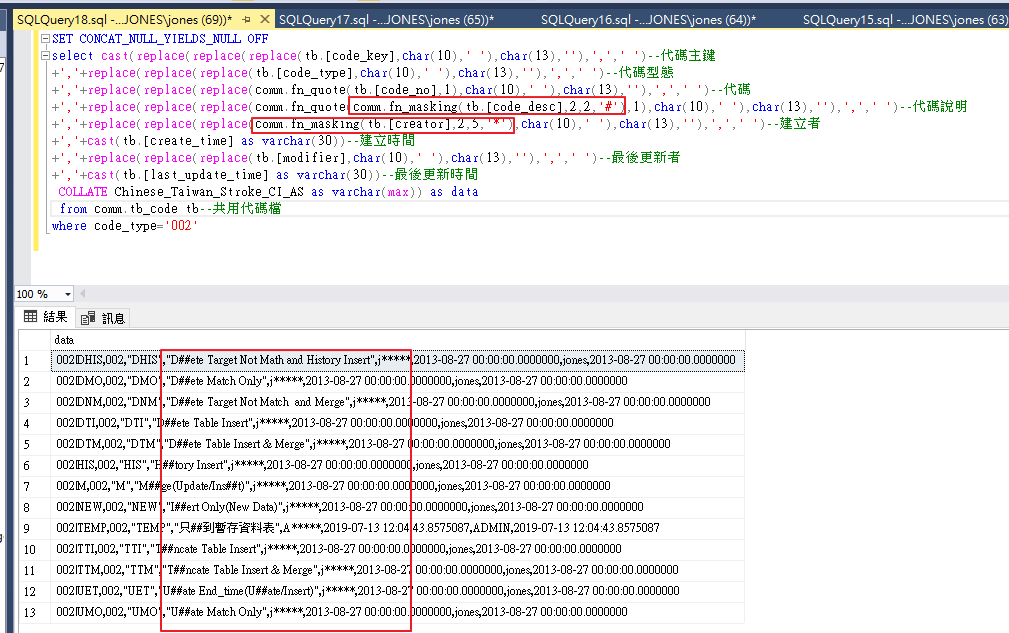
,@last\_data\_column\_name

,@program\_no

,@exec\_group

,@lang\_flag)





**範例八：透過網頁設定，手動產生匯出具有遮罩欄位的資料表內容SQL語法**

1. 首先先到資料表檢視中，選取要產生的資料表(comm.tb\_code)

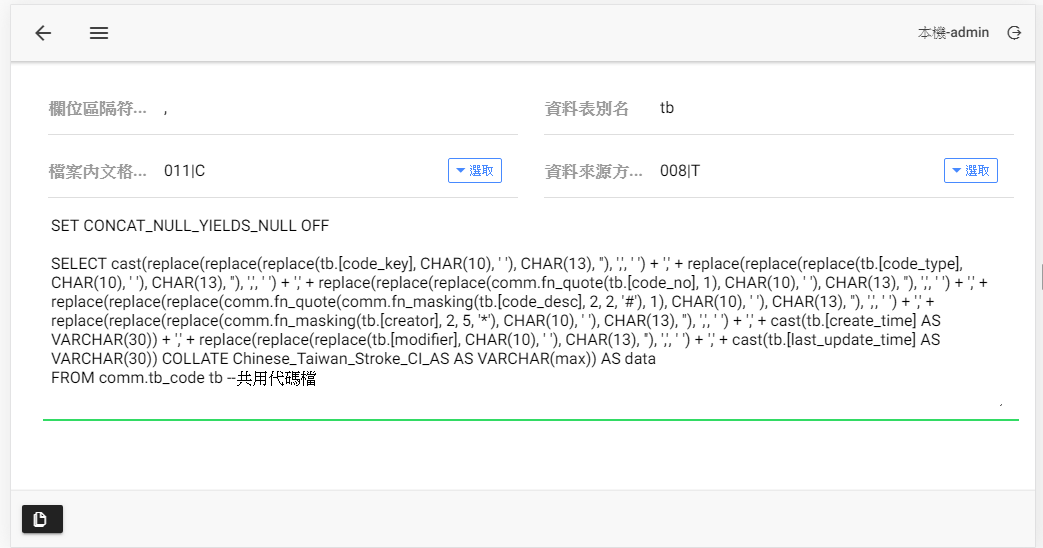


1. 點選或設定遮罩欄位，點選可以新增欄位遮罩

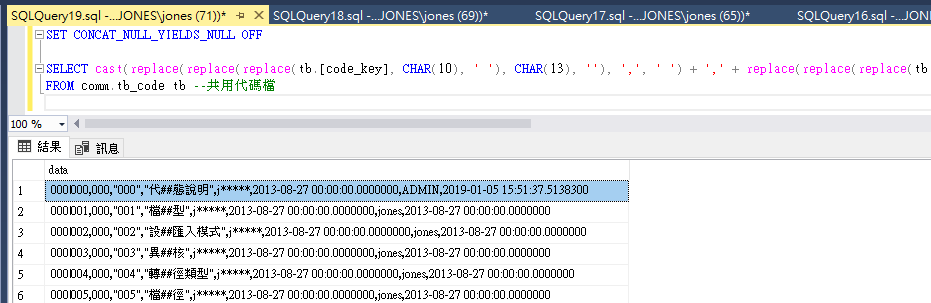


1. 回實體資料表請單點選檢視產生匯出sql語法





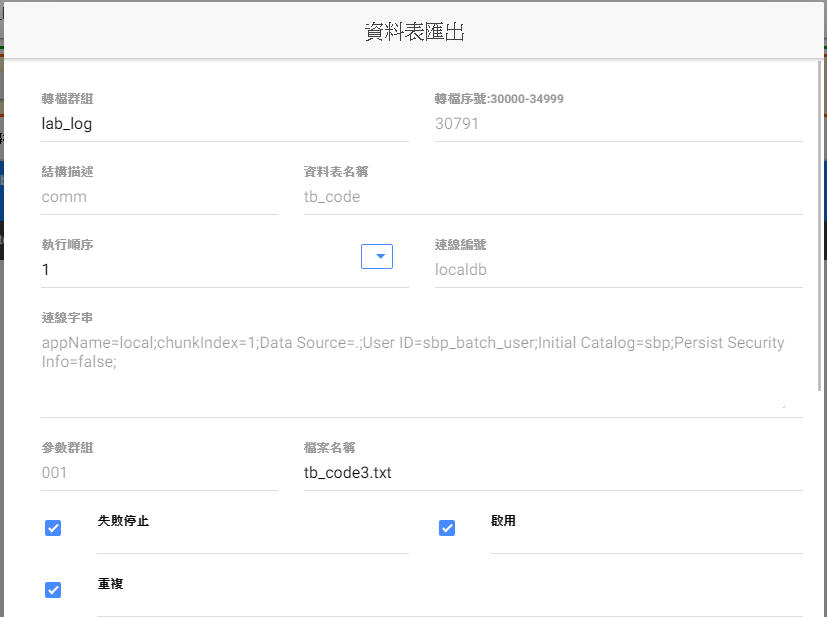
1. 點選複製sql語法，貼到ssms執行

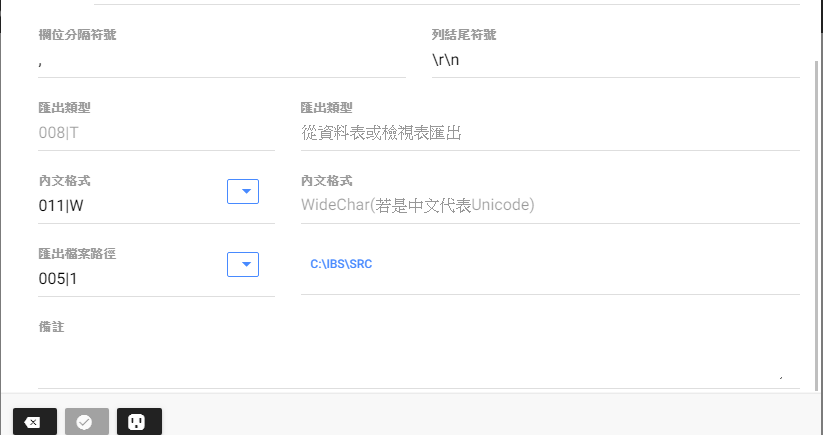


**範例九：透過網頁設定，手動執行批次產生匯出具有遮罩欄位的檔案**

1. 首先先到資料表匯出設定建立相關匯出資訊如下圖







1. 儲存後回到清單，點選手動執行



1. 系統會依照設定內容產生檔案放在指定路徑

