AS/400 結構化查詢語言與
查詢工具實務應用設計

AS/400 Query & SQL Development Design

泛太資訊科技開發股份有限公司


單元一
QUERY

目 錄

第一章

QUERY/400特性

QUERY/400

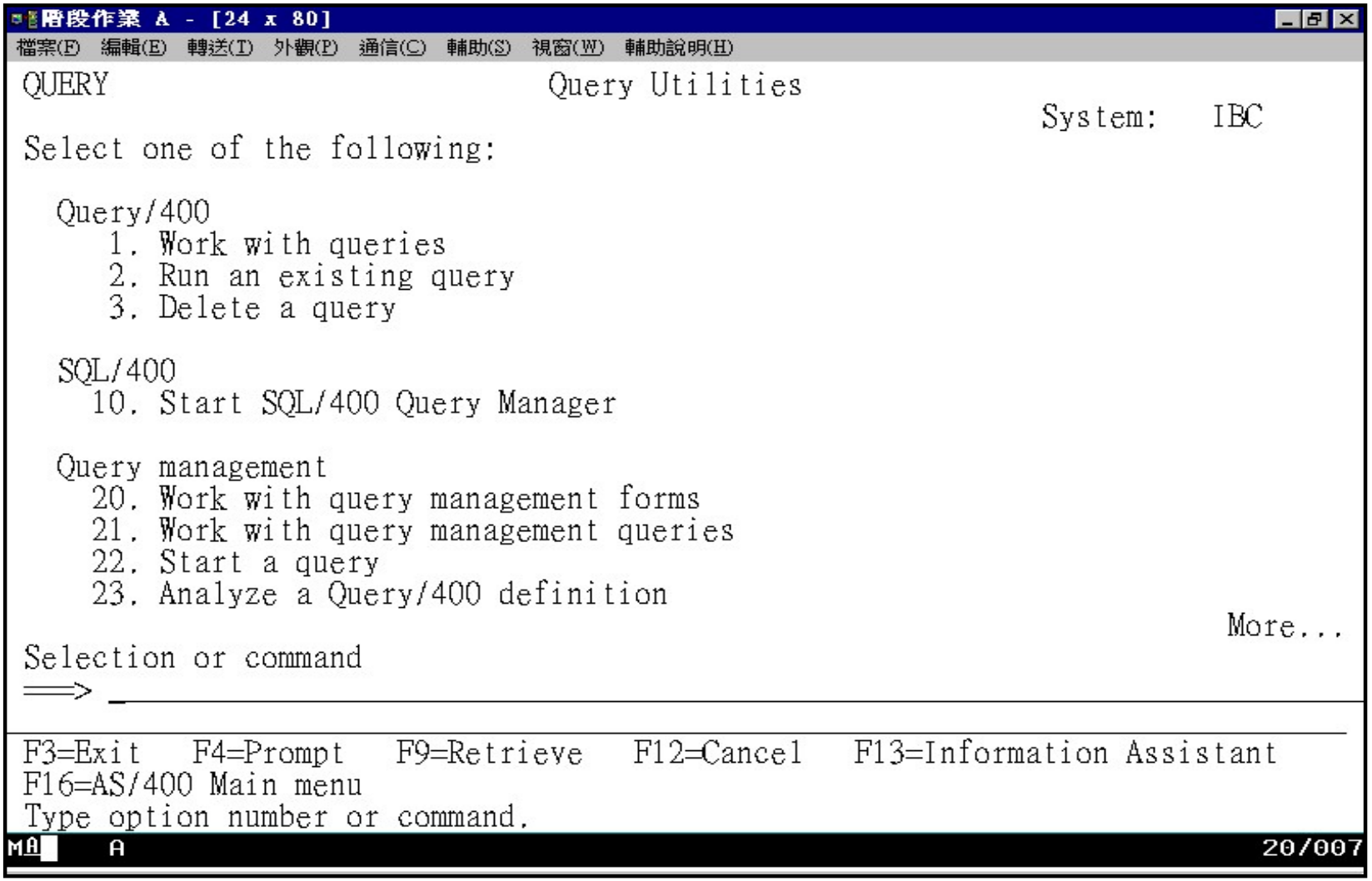
“QUERY”是一針對由DDS定義出的資料檔而產生報表的工具

- ♦ 採用畫面指引，不需撰寫程式，操作容易。
- ♦ 使用者定義輸出之格式、內容及型態。
- ♦ 輸出方式可由螢幕查詢，產生報表或輸出檔案。

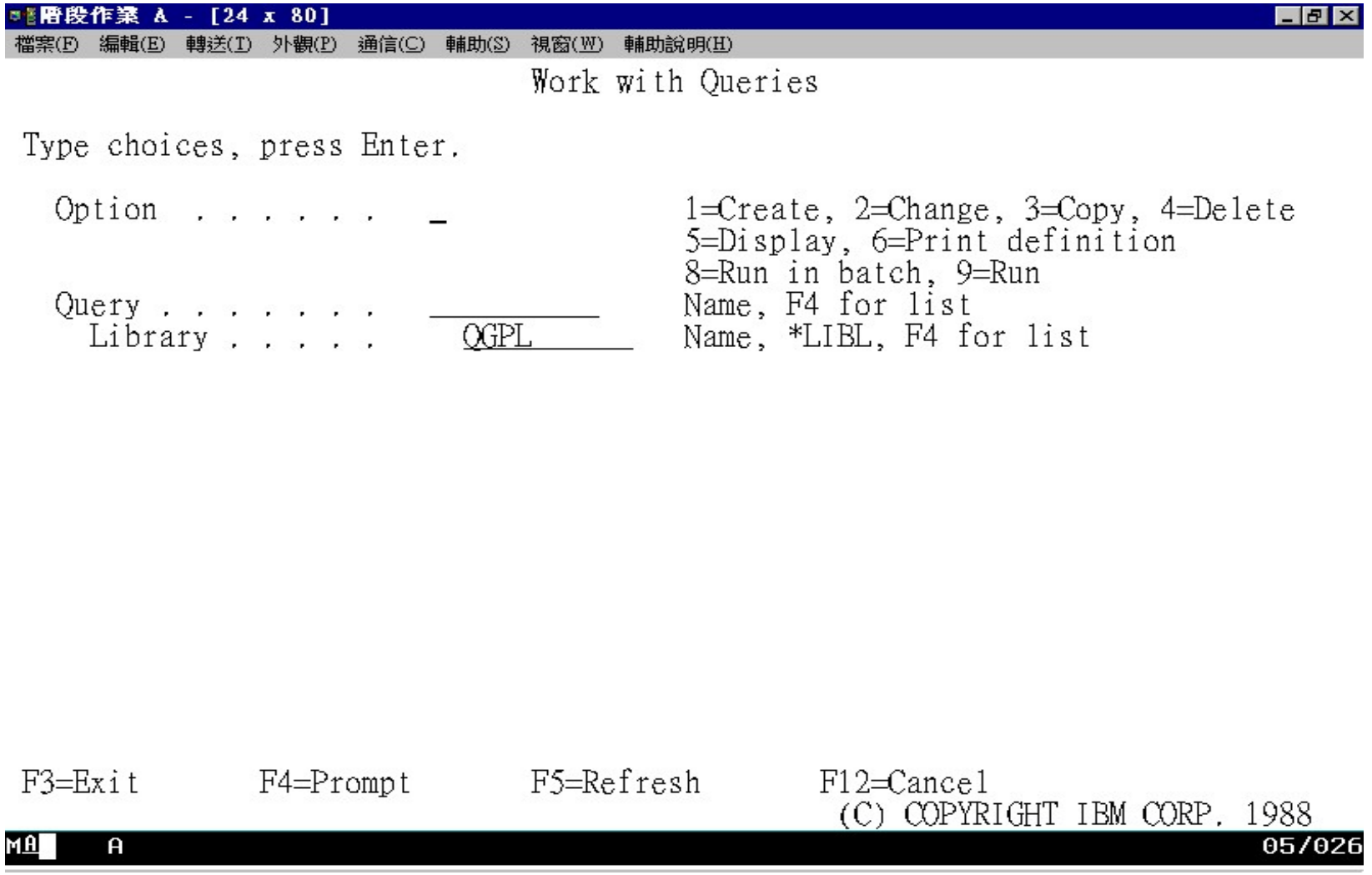
第二章

QUERY/400操作

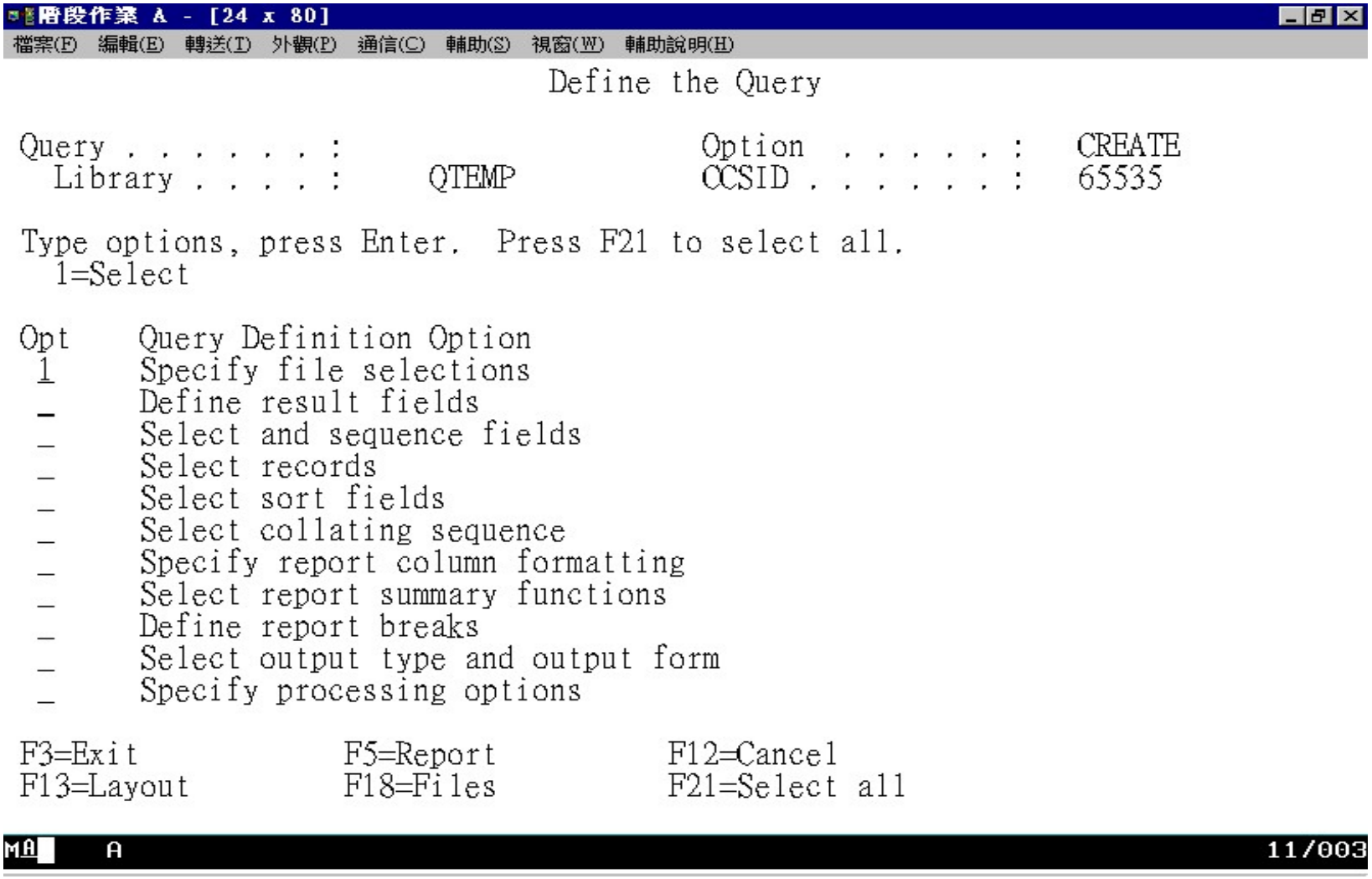
QUERY主畫面


暗段作業 A - [24 x 80]
檔案(F) 編輯(E) 轉送(T) 外觀(P) 通信(C) 輔助(S) 視窗(W) 輔助說明(H)
QUERY
Query Utilities
System: IBC
Select one of the following:
Query/400
1. Work with queries
2. Run an existing query
3. Delete a query
SQL/400
10. Start SQL/400 Query Manager
Query management
20. Work with query management forms
21. Work with query management queries
22. Start a query
23. Analyze a Query/400 definition
More...
Selection or command
==>
F3=Exit F4=Prompt F9=Retrieve F12=Cancel F13=Information Assistant
F16=AS/400 Main menu
Type option number or command.
MA A
20/007

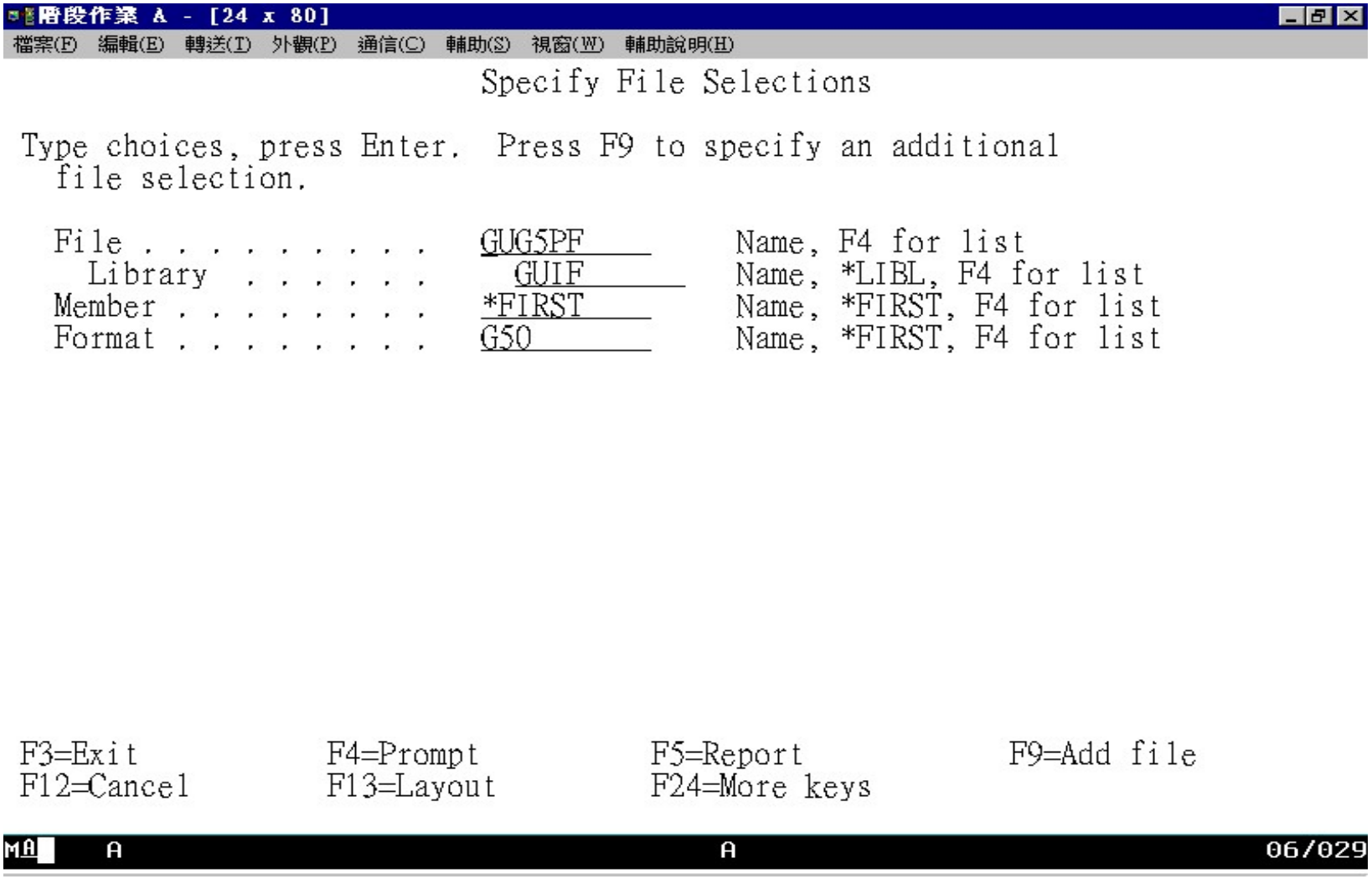
QUERY工作選項畫面



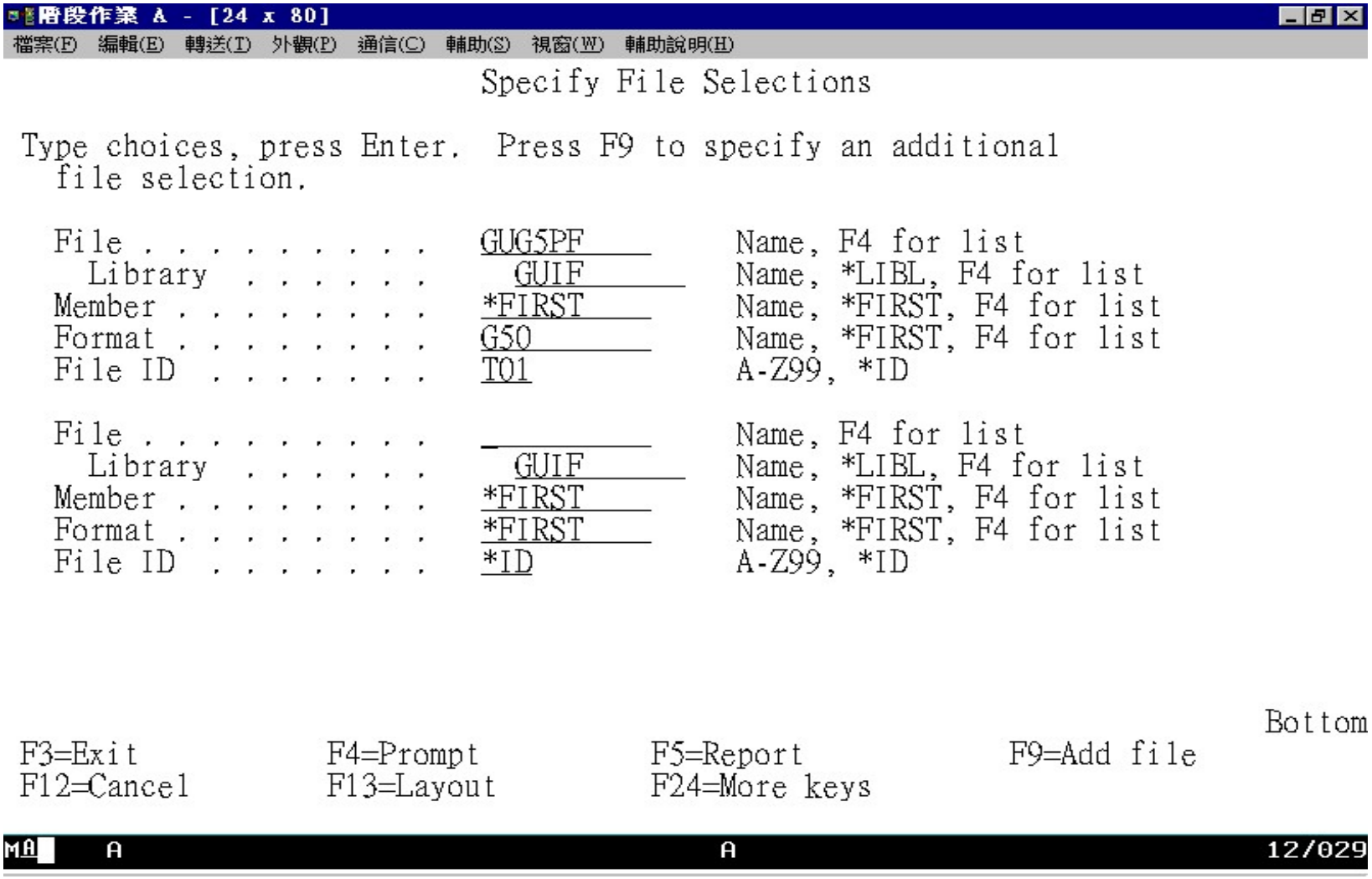
定義QUERY(定義選項)



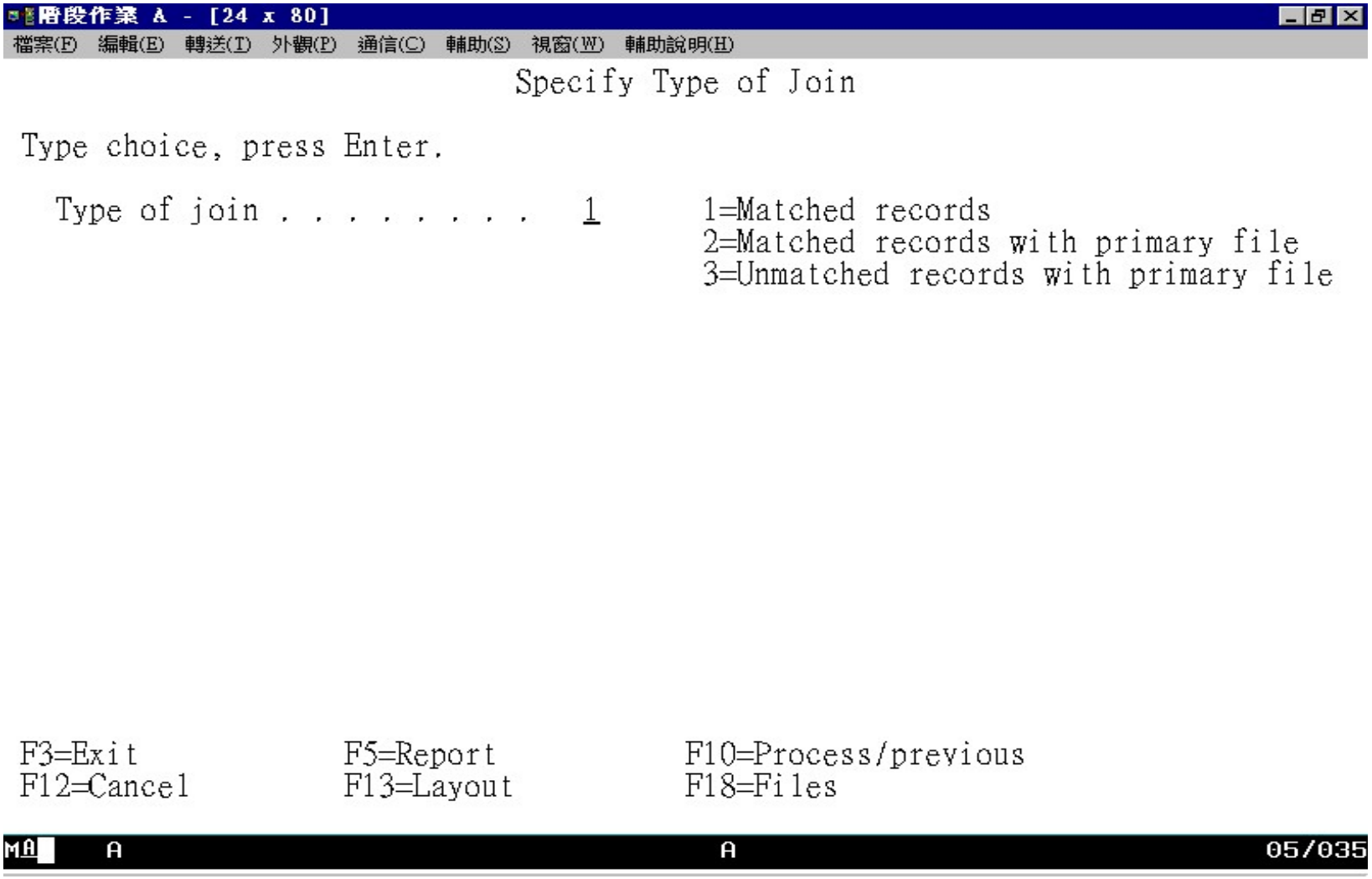
指定選取檔案(一)


階段作業 A - [24 x 80]
檔案(F) 編輯(E) 轉送(T) 外觀(P) 通信(C) 輔助(S) 視窗(W) 輔助說明(H)
Specify File Selections
Type choices, press Enter. Press F9 to specify an additional file selection.
File . . . . . GUG5PF Name, F4 for list
Library . . . . . GUIF Name, *LIBL, F4 for list
Member . . . . . *FIRST Name, *FIRST, F4 for list
Format . . . . . G50 Name, *FIRST, F4 for list
F3=Exit F4=Prompt F5=Report F9=Add file
F12=Cancel F13=Layout F24=More keys
MA A A 06/029

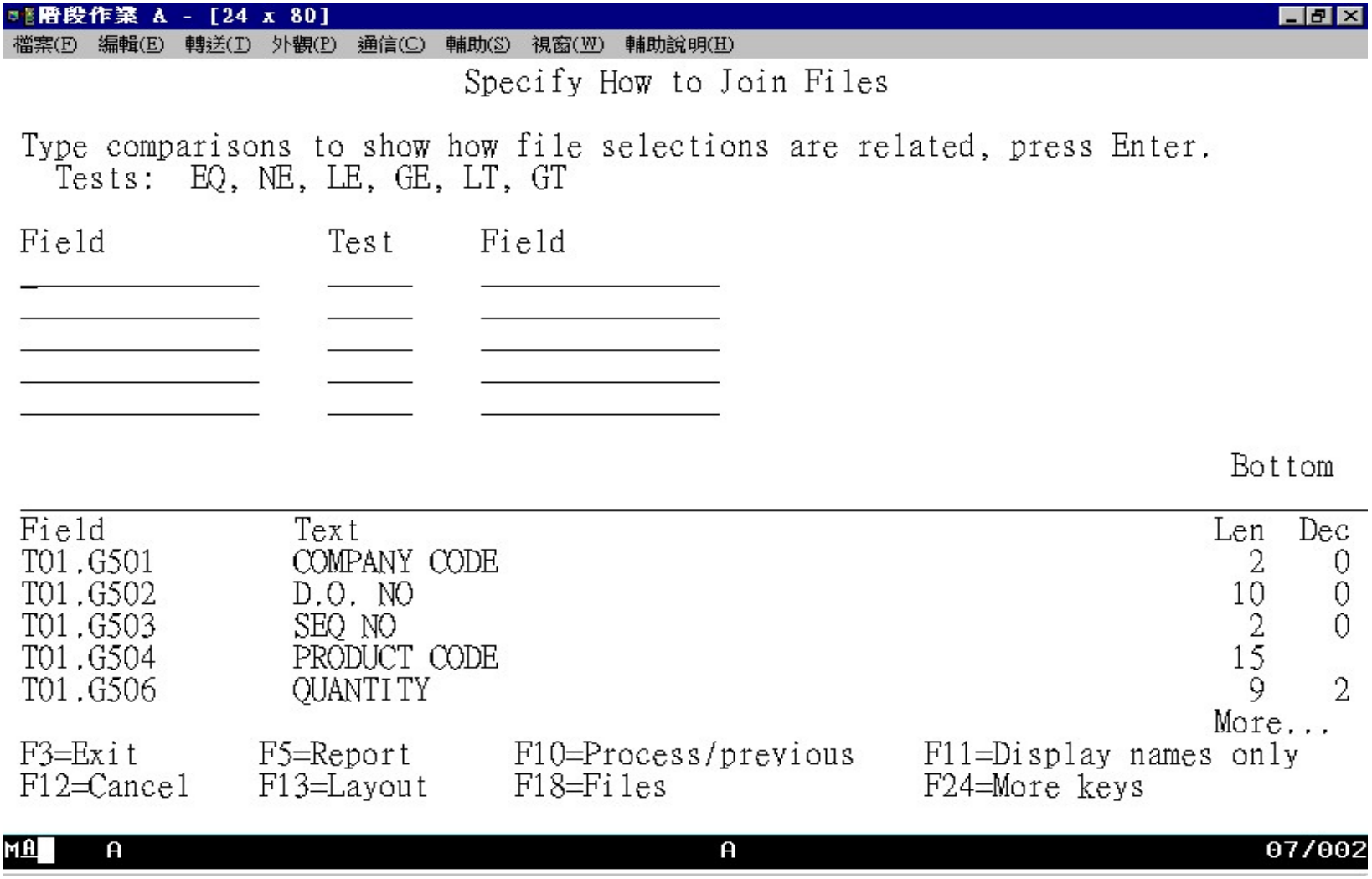
指定選取檔案(二)


階段作業 A - [24 x 80]
檔案(F) 編輯(E) 轉送(T) 外觀(P) 通信(C) 輔助(S) 視窗(W) 輔助說明(H)
Specify File Selections
Type choices, press Enter. Press F9 to specify an additional file selection.
File . . . . . GUG5PF Name, F4 for list
Library . . . . . GUIF Name, *LIBL, F4 for list
Member . . . . . *FIRST Name, *FIRST, F4 for list
Format . . . . . G50 Name, *FIRST, F4 for list
File ID . . . . . T01 A-Z99, *ID
File . . . . . Name, F4 for list
Library . . . . . GUIF Name, *LIBL, F4 for list
Member . . . . . *FIRST Name, *FIRST, F4 for list
Format . . . . . *FIRST Name, *FIRST, F4 for list
File ID . . . . . *ID A-Z99, *ID
Bottom
F3=Exit F4=Prompt F5=Report F9=Add file
F12=Cancel F13=Layout F24=More keys
MA A A 12/029

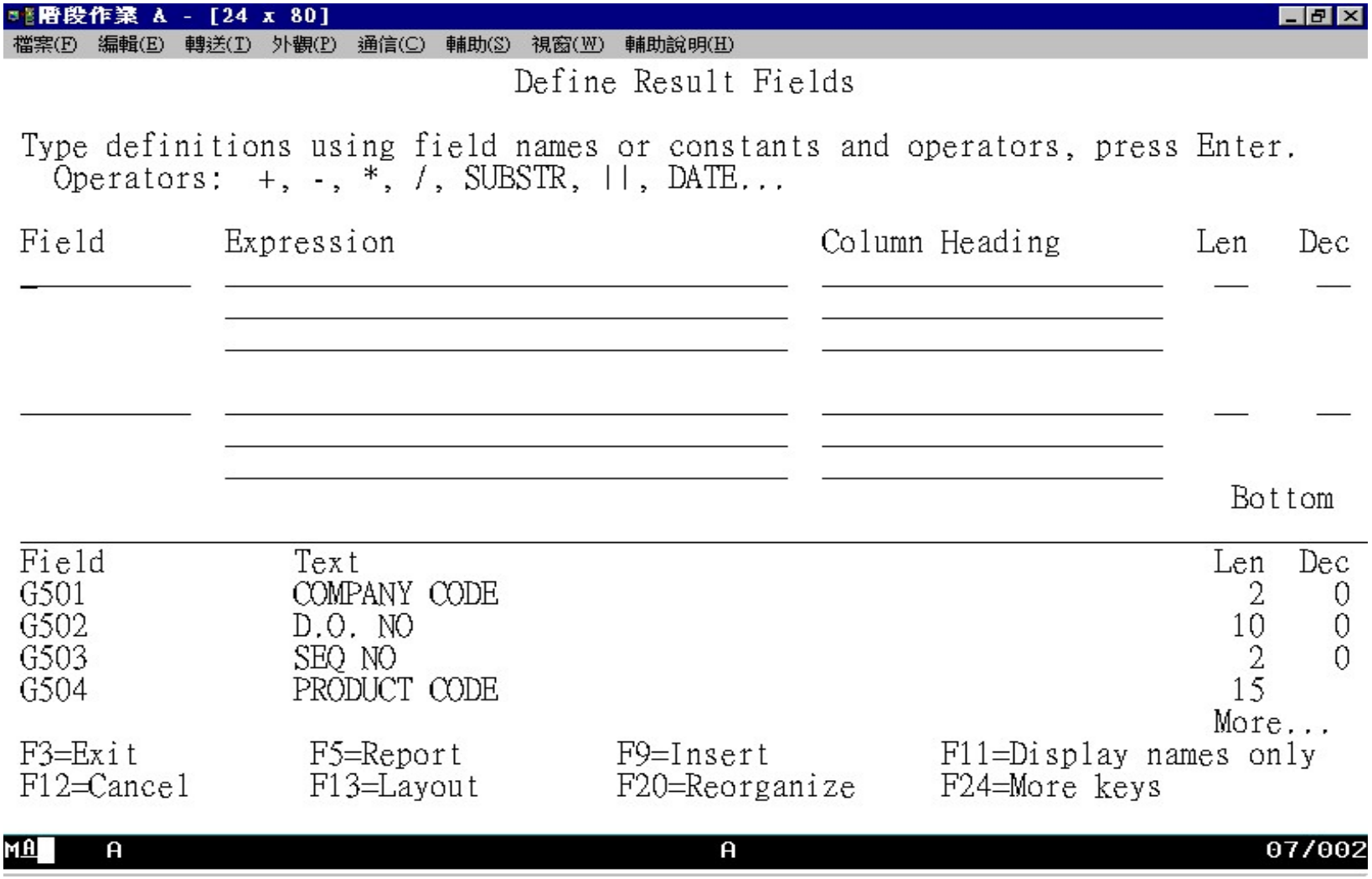
指定連結型態


階段作業 A - [24 x 80]
檔案(F) 編輯(E) 轉送(T) 外觀(P) 通信(C) 輔助(S) 視窗(W) 輔助說明(H)
Specify Type of Join
Type choice, press Enter.
Type of join 1
1=Matched records
2=Matched records with primary file
3=Unmatched records with primary file
F3=Exit
F5=Report
F10=Process/previous
F12=Cancel
F13=Layout
F18=Files
MA A A 05/035

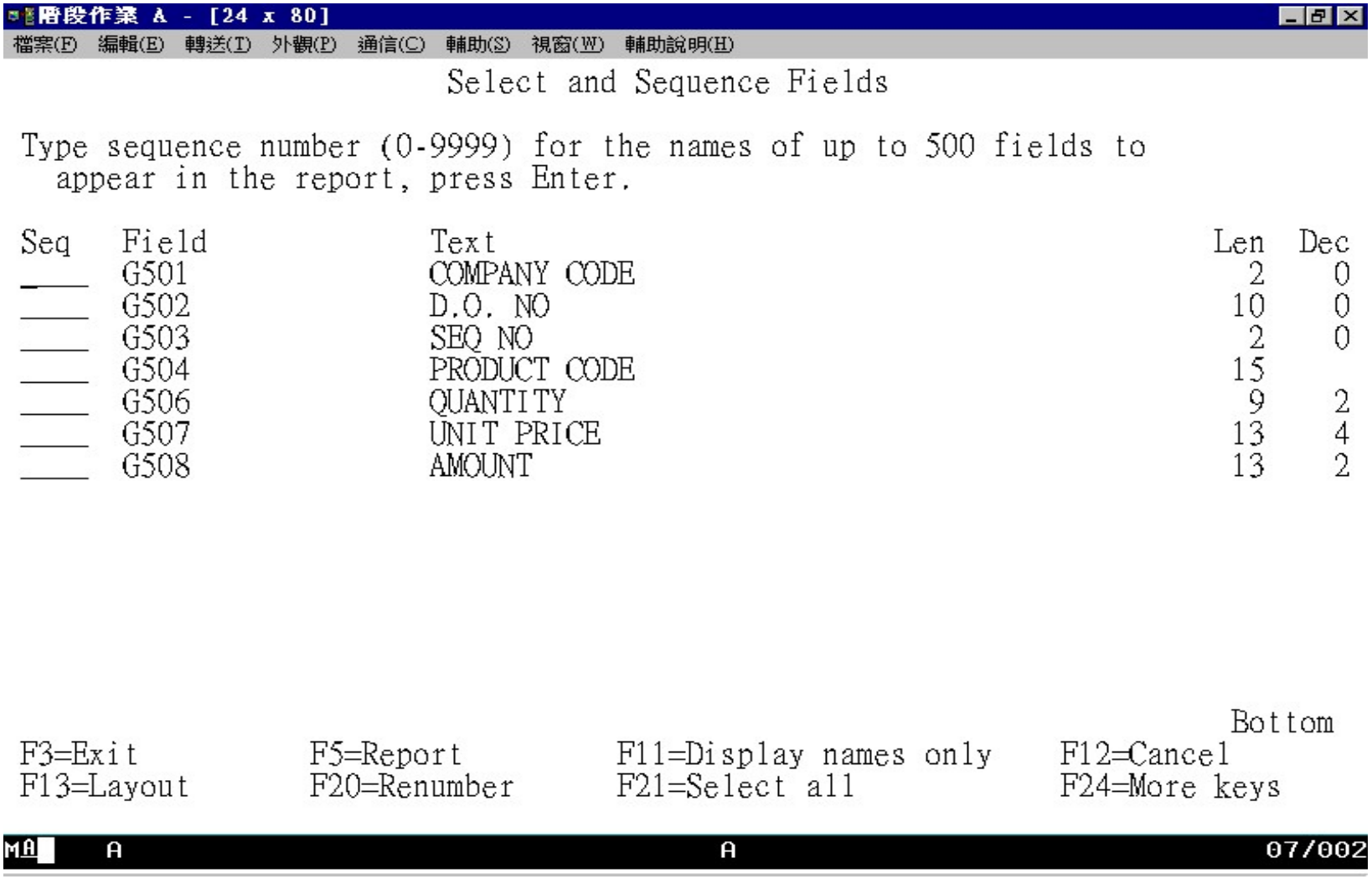
指定連結檔案


層段作業 A - [24 x 80]
檔案(F) 編輯(E) 轉送(T) 外觀(P) 通信(C) 輔助(S) 視窗(W) 輔助說明(H)
Specify How to Join Files
Type comparisons to show how file selections are related, press Enter.
Tests: EQ, NE, LE, GE, LT, GT
Field Test Field
Bottom
Field Text Len Dec
T01.G501 COMPANY CODE 2 0
T01.G502 D.O. NO 10 0
T01.G503 SEQ NO 2 0
T01.G504 PRODUCT CODE 15
T01.G506 QUANTITY 9 2
More...
F3=Exit F5=Report F10=Process/previous F11=Display names only
F12=Cancel F13=Layout F18=Files F24=More keys
MA A A 07/002

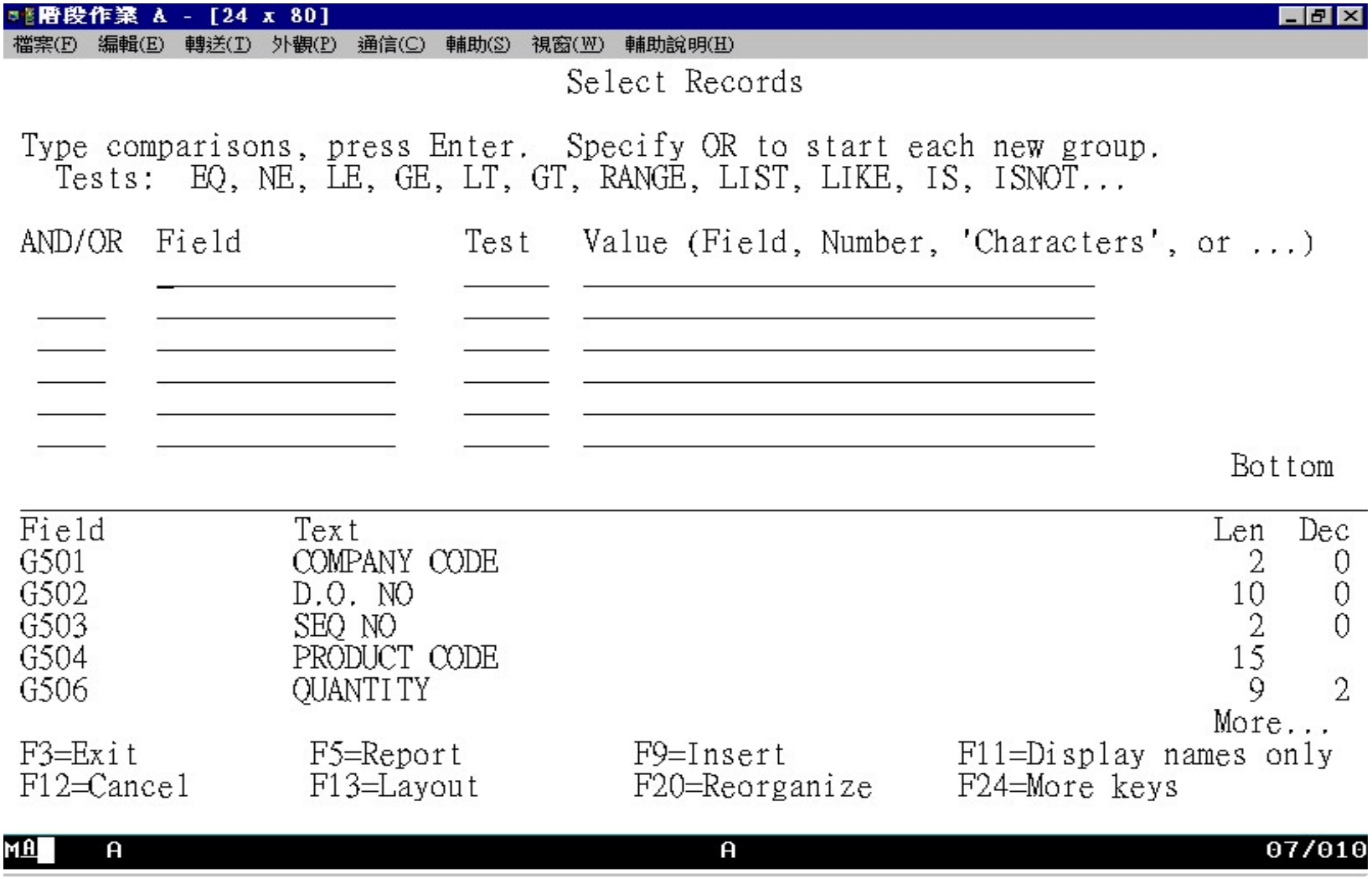
定義結果欄位


層段作業 A - [24 x 80]
檔案(F) 編輯(E) 轉送(T) 外觀(P) 通信(C) 輔助(S) 視窗(W) 輔助說明(H)
Define Result Fields
Type definitions using field names or constants and operators, press Enter.
Operators: +, -, *, /, SUBSTR, ||, DATE...
Field Expression Column Heading Len Dec
Bottom
Field Text Len Dec
G501 COMPANY CODE 2 0
G502 D.O. NO 10 0
G503 SEQ NO 2 0
G504 PRODUCT CODE 15
More...
F3=Exit F5=Report F9=Insert F11=Display names only
F12=Cancel F13=Layout F20=Reorganize F24=More keys
MA A A 07/002

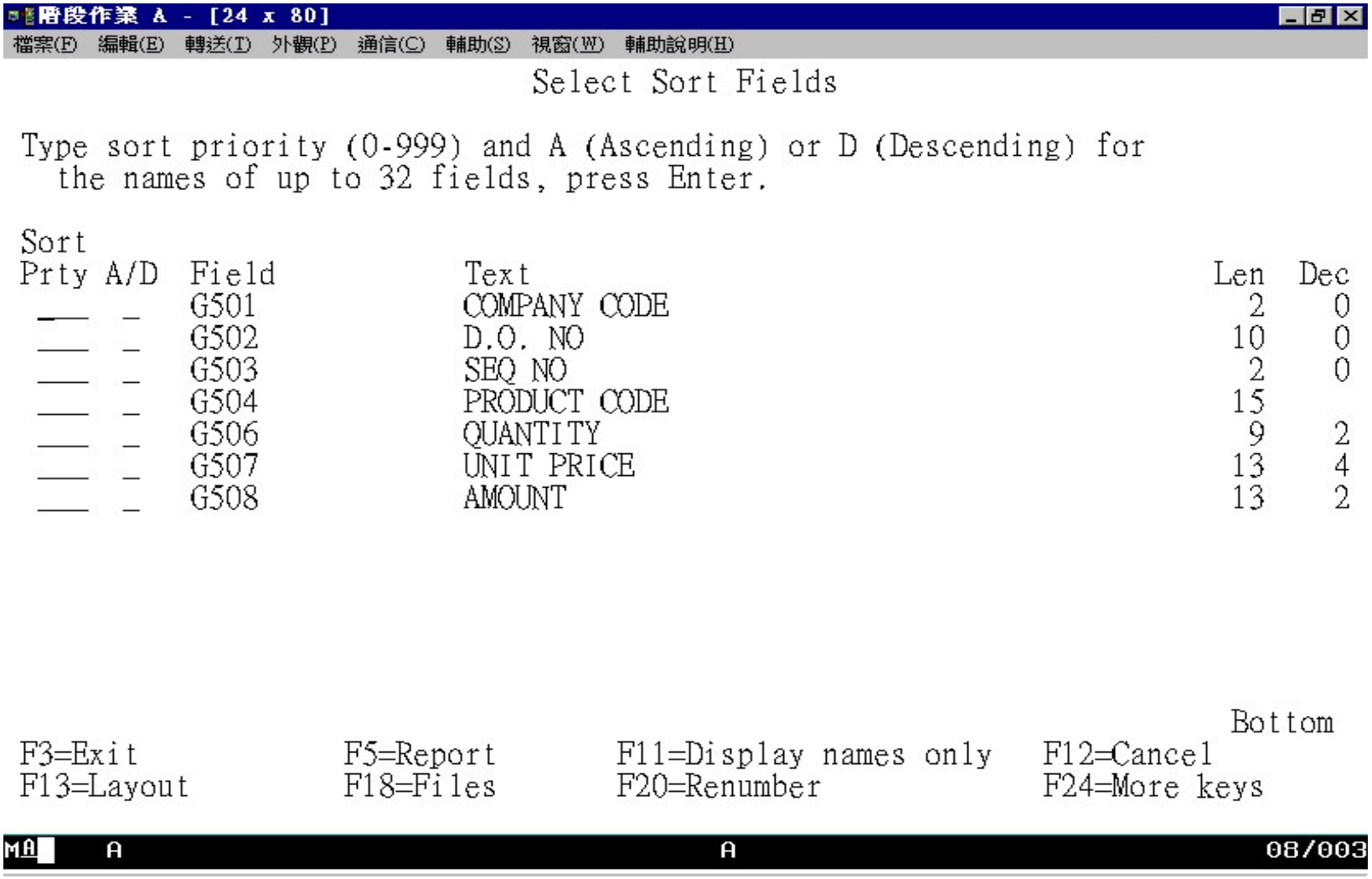
選擇欄位及排列順序


層段作業 A - [24 x 80]
檔案(F) 編輯(E) 轉送(T) 外觀(P) 通信(C) 輔助(S) 視窗(W) 輔助說明(H)
Select and Sequence Fields
Type sequence number (0-9999) for the names of up to 500 fields to
appear in the report, press Enter.
Seq Field Text Len Dec
G501 COMPANY CODE 2 0
G502 D.O. NO 10 0
G503 SEQ NO 2 0
G504 PRODUCT CODE 15
G506 QUANTITY 9 2
G507 UNIT PRICE 13 4
G508 AMOUNT 13 2
Bottom
F3=Exit F5=Report F11=Display names only F12=Cancel
F13=Layout F20=Renumber F21=Select all F24=More keys
MA A A 07/002

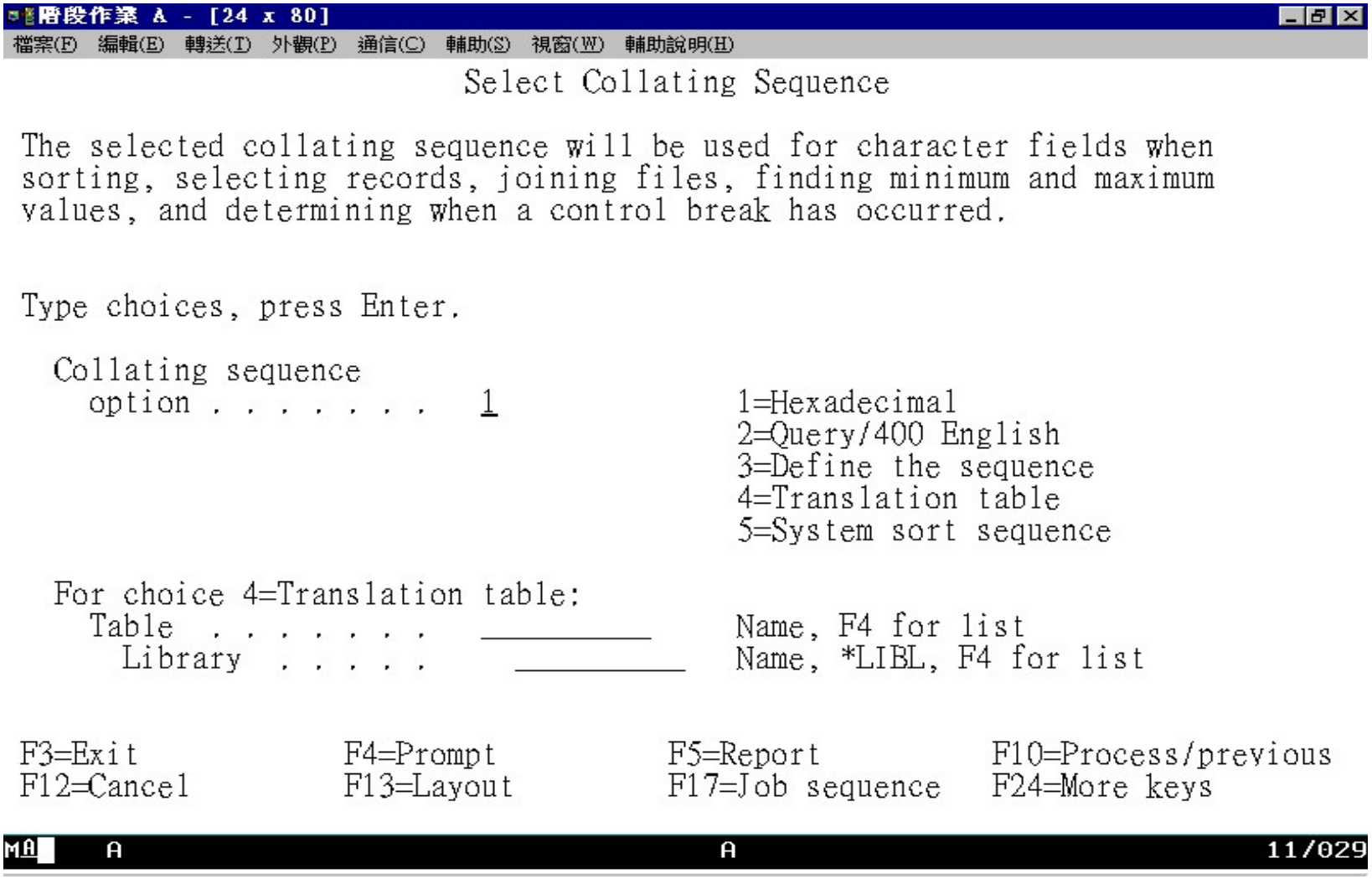
資料內容篩選


層段作業 A - [24 x 80]
檔案(F) 編輯(E) 轉送(T) 外觀(P) 通信(C) 輔助(S) 視窗(W) 輔助說明(H)
Select Records
Type comparisons, press Enter. Specify OR to start each new group.
Tests: EQ, NE, LE, GE, LT, GT, RANGE, LIST, LIKE, IS, ISNOT...
AND/OR Field Test Value (Field, Number, 'Characters', or ...)
Bottom
Field Text Len Dec
G501 COMPANY CODE 2 0
G502 D.O. NO 10 0
G503 SEQ NO 2 0
G504 PRODUCT CODE 15
G506 QUANTITY 9 2
More...
F3=Exit F5=Report F9=Insert F11=Display names only
F12=Cancel F13=Layout F20=Reorganize F24=More keys
MA A A 07/010

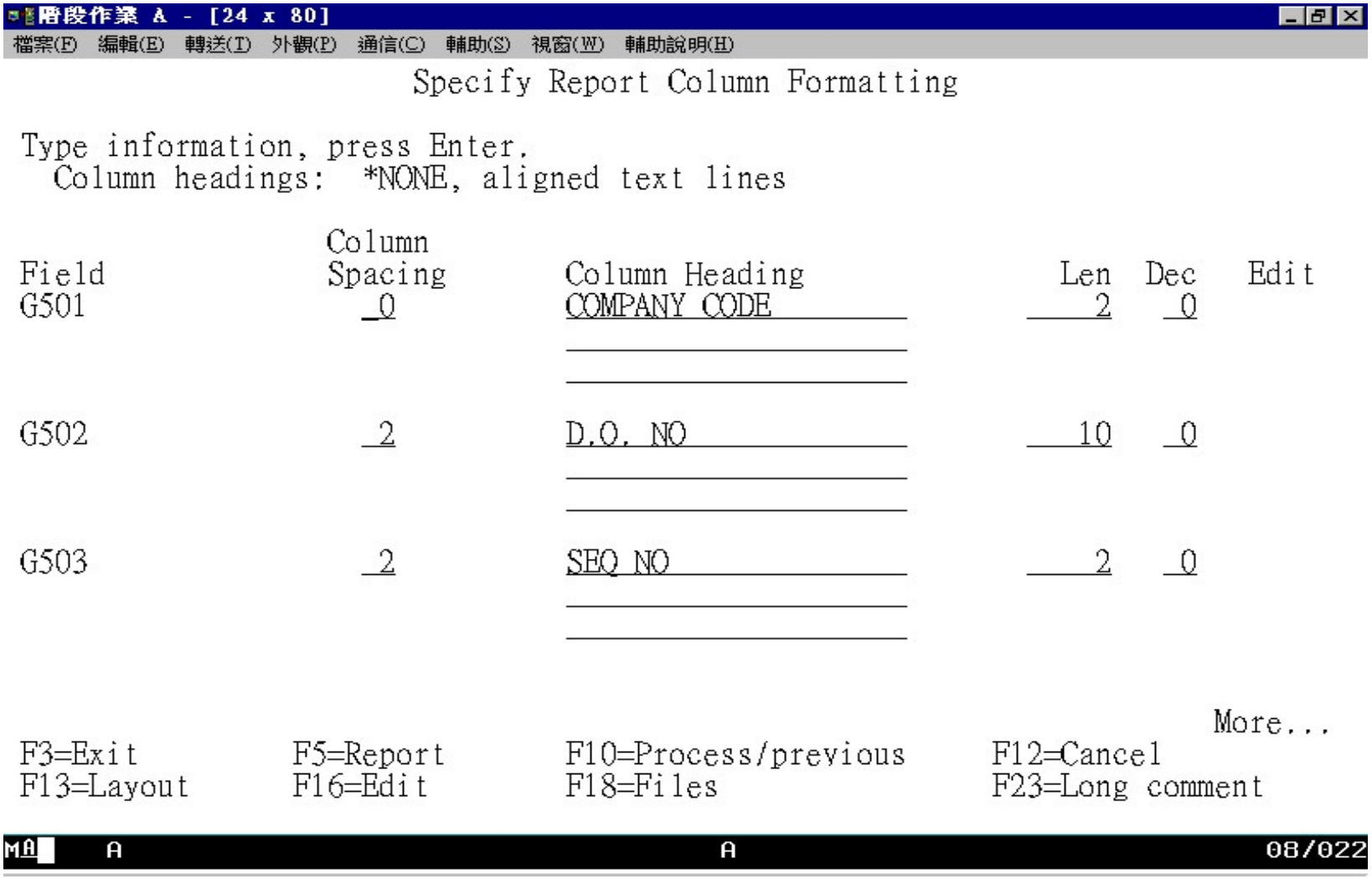
欄位排序設定

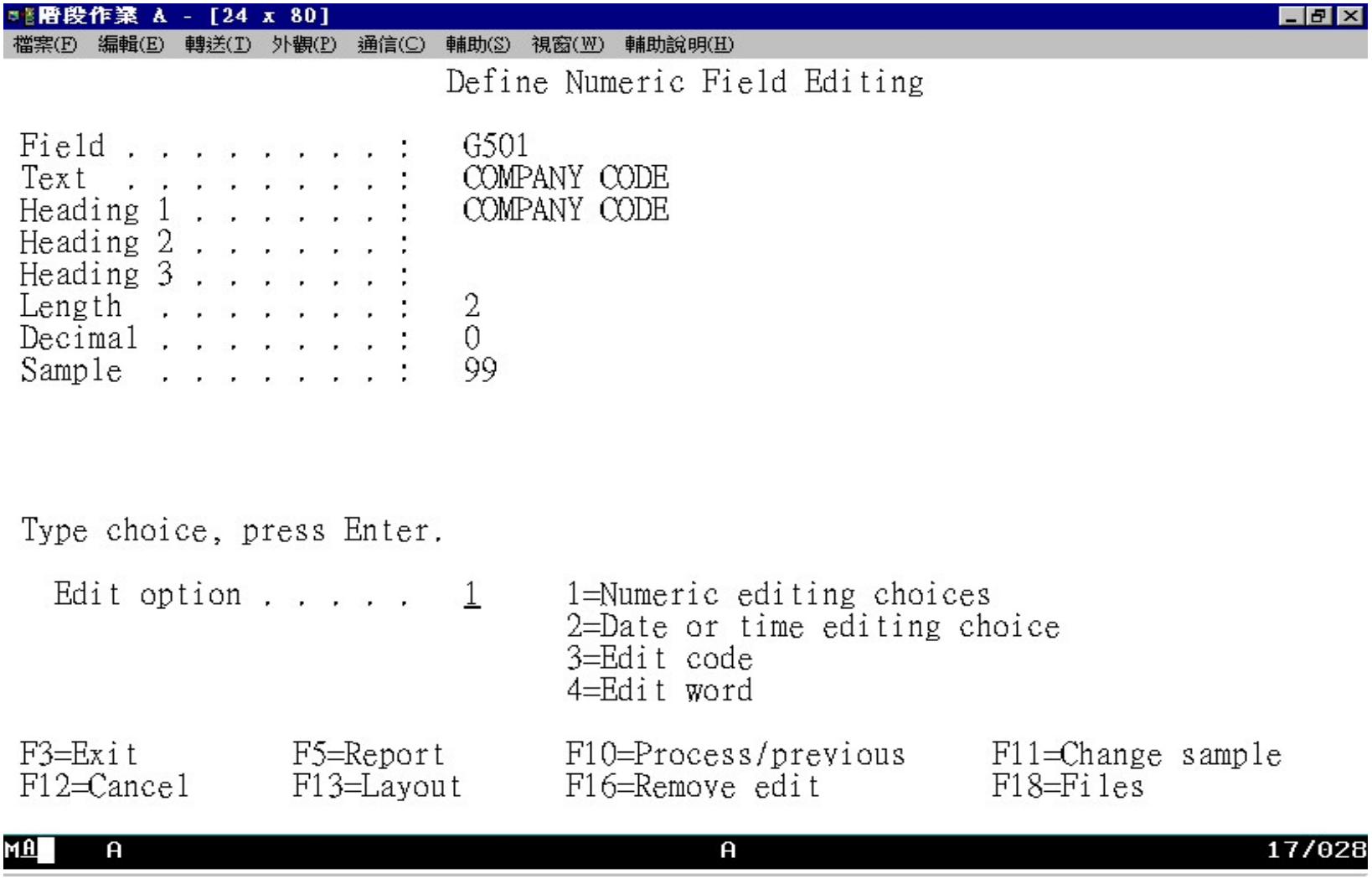

欄段作業 A - [24 x 80]
檔案(F) 編輯(E) 轉送(T) 外觀(P) 通信(C) 輔助(S) 視窗(W) 輔助說明(H)
Select Sort Fields
Type sort priority (0-999) and A (Ascending) or D (Descending) for
the names of up to 32 fields, press Enter.
Sort
Prty A/D Field Text Len Dec
___ _ G501 COMPANY CODE 2 0
___ _ G502 D.O. NO 10 0
___ _ G503 SEQ NO 2 0
___ _ G504 PRODUCT CODE 15
___ _ G506 QUANTITY 9 2
___ _ G507 UNIT PRICE 13 4
___ _ G508 AMOUNT 13 2
Bottom
F3=Exit F5=Report F11=Display names only F12=Cancel
F13=Layout F18=Files F20=Renumber F24=More keys
MA A A 08/003

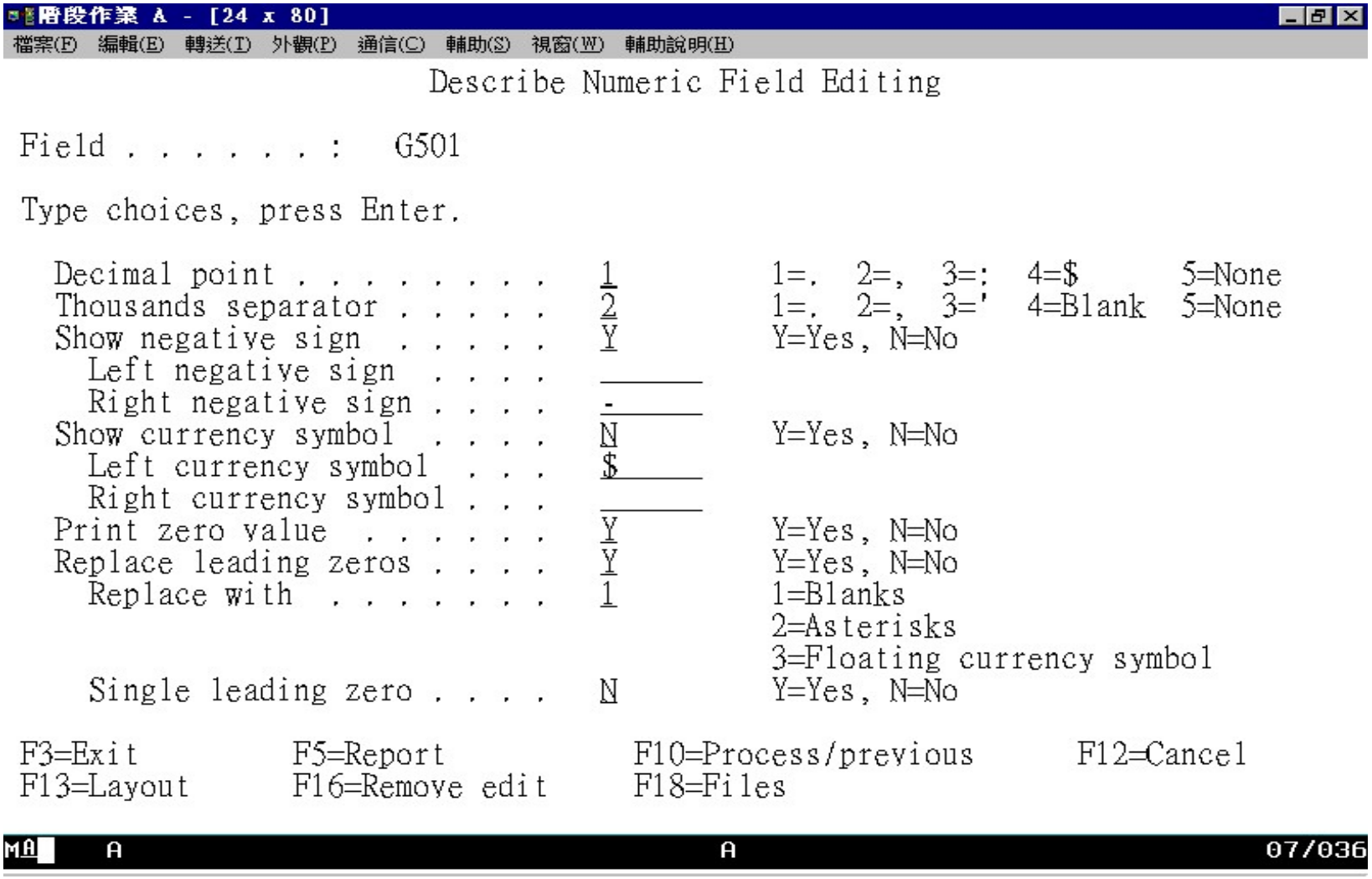
選擇排列順序標準

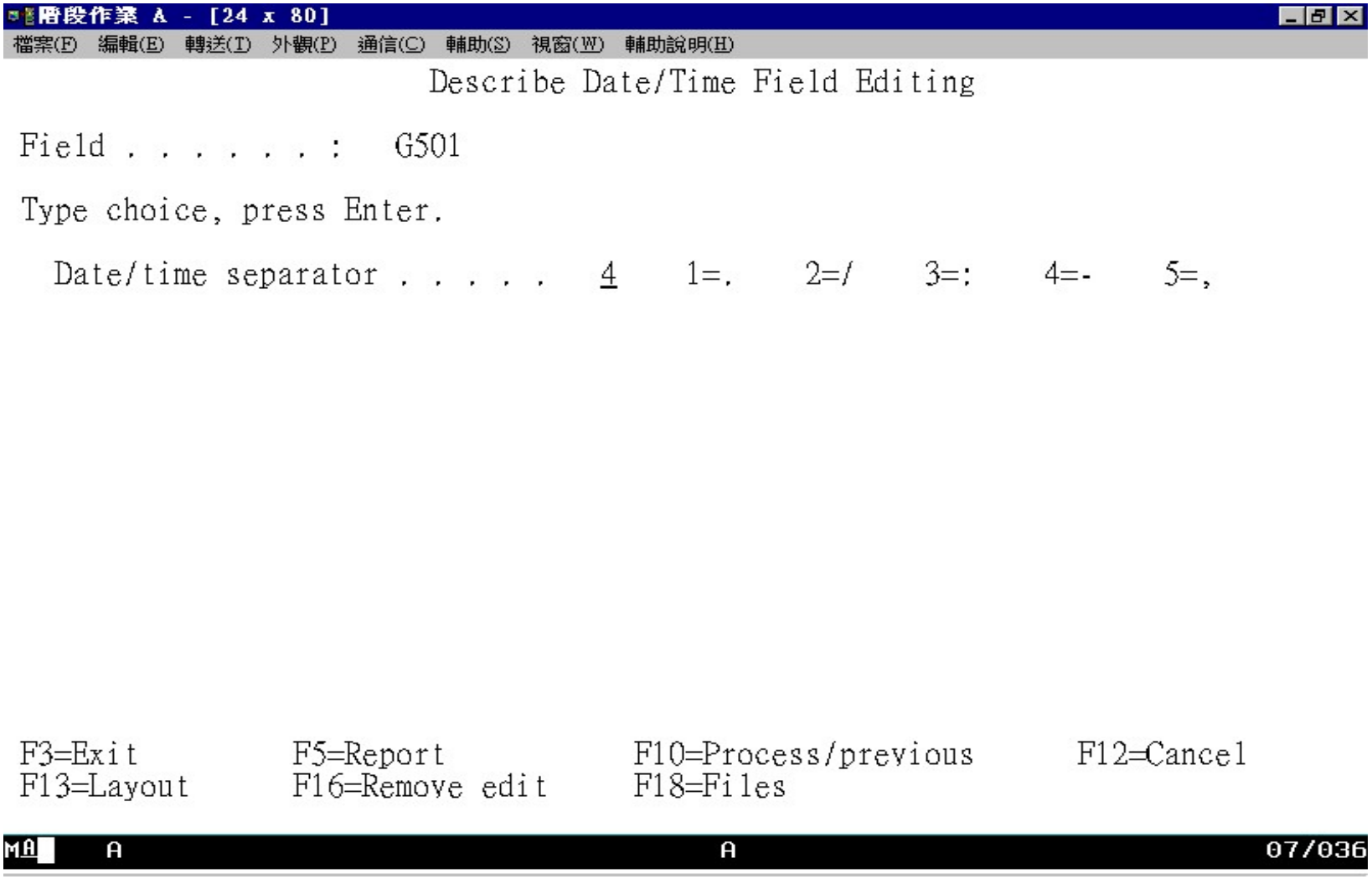

5250 階段作業 A - [24 x 80]
檔案(F) 編輯(E) 轉送(T) 外觀(P) 通信(C) 輔助(S) 視窗(W) 輔助說明(H)
Select Collating Sequence
The selected collating sequence will be used for character fields when
sorting, selecting records, joining files, finding minimum and maximum
values, and determining when a control break has occurred.
Type choices, press Enter.
Collating sequence
option 1
1=Hexadecimal
2=Query/400 English
3=Define the sequence
4=Translation table
5=System sort sequence
For choice 4=Translation table:
Table . . . . . Name, F4 for list
Library . . . . . Name, *LIBL, F4 for list
F3=Exit F4=Prompt F5=Report F10=Process/previous
F12=Cancel F13=Layout F17=Job sequence F24=More keys
MA A A 11/029

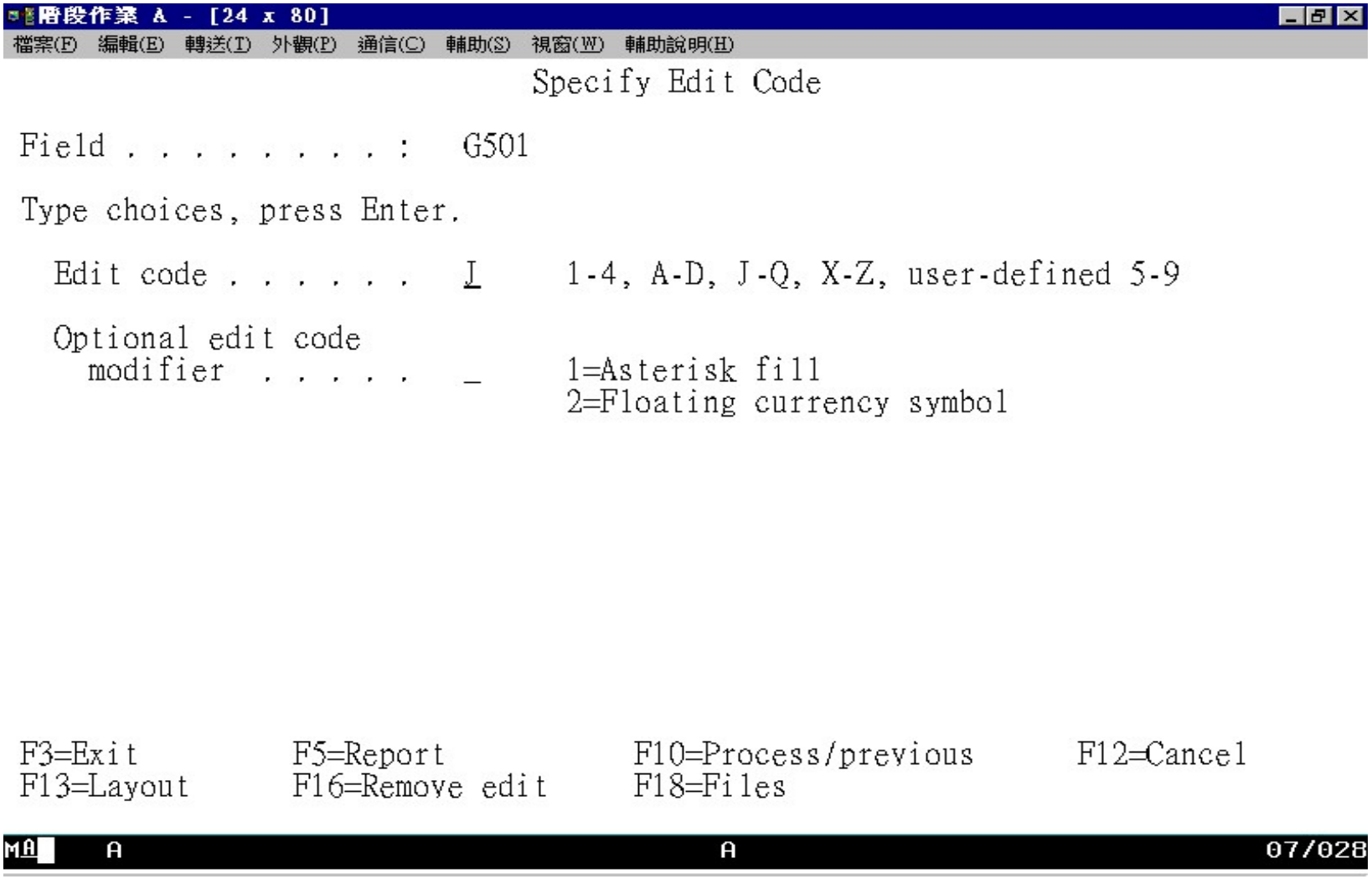
定義報表欄位格式

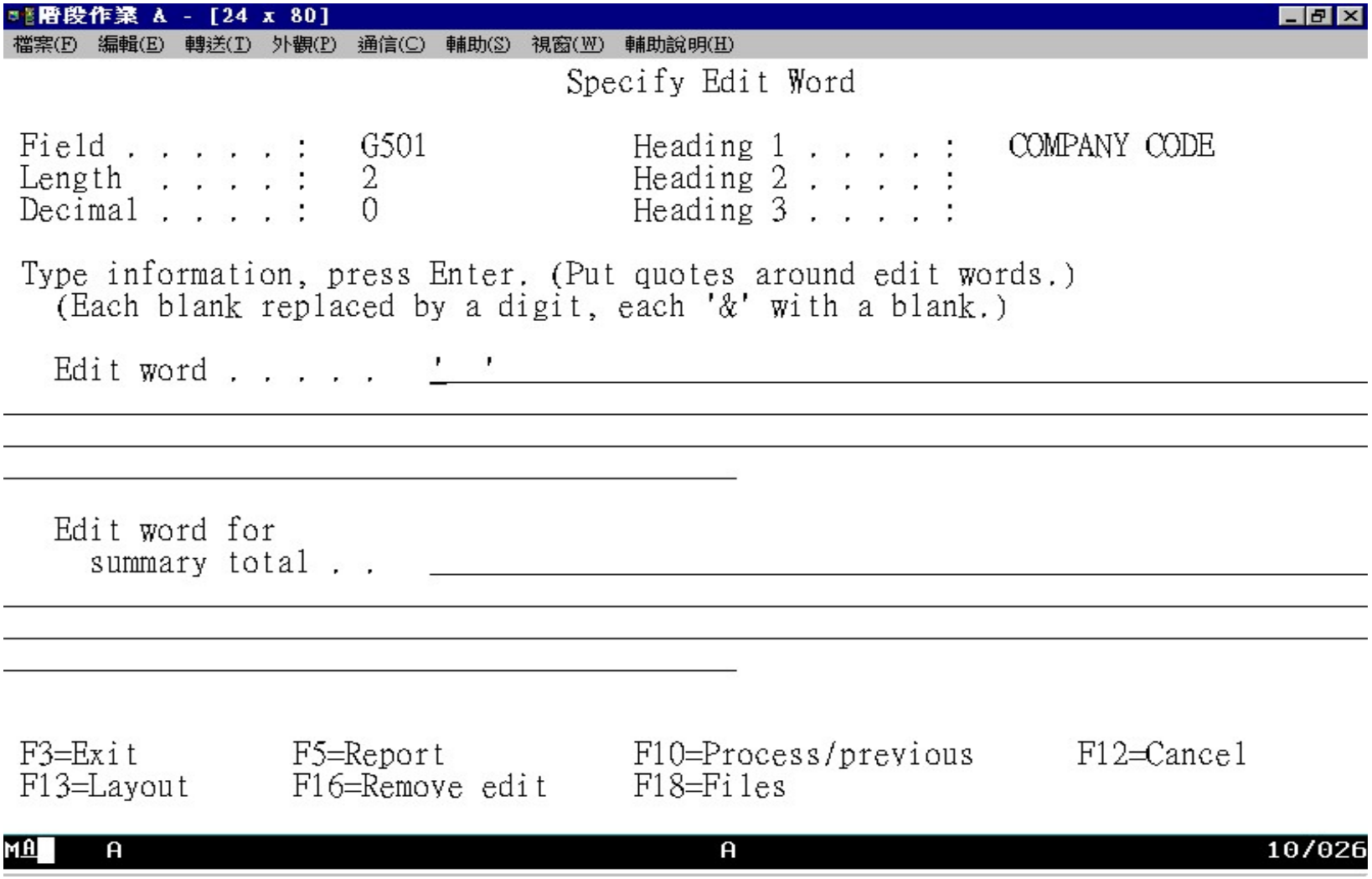

階段作業 A - [24 x 80]
檔案(F) 編輯(E) 轉送(T) 外觀(P) 通信(C) 輔助(S) 視窗(W) 輔助說明(H)
Specify Report Column Formatting
Type information, press Enter.
Column headings: *NONE, aligned text lines
Field
Column Spacing
Column Heading
Len
Dec
Edit
G501
0
COMPANY CODE
2
0
G502
2
D.O. NO
10
0
G503
2
SEQ NO
2
0
More...
F3=Exit
F5=Report
F10=Process/previous
F12=Cancel
F13=Layout
F16=Edit
F18=Files
F23=Long comment
MA
A
A
08/022


欄段作業 A - [24 x 80]
檔案(F) 編輯(E) 轉送(T) 外觀(P) 通信(C) 輔助(S) 視窗(W) 輔助說明(H)
Define Numeric Field Editing
Field . . . . . : G501
Text . . . . . : COMPANY CODE
Heading 1 . . . . . : COMPANY CODE
Heading 2 . . . . . :
Heading 3 . . . . . :
Length . . . . . : 2
Decimal . . . . . : 0
Sample . . . . . : 99
Type choice, press Enter.
Edit option . . . . . 1 1=Numeric editing choices
2=Date or time editing choice
3=Edit code
4=Edit word
F3=Exit F5=Report F10=Process/previous F11=Change sample
F12=Cancel F13=Layout F16=Remove edit F18=Files
MA A A 17/028

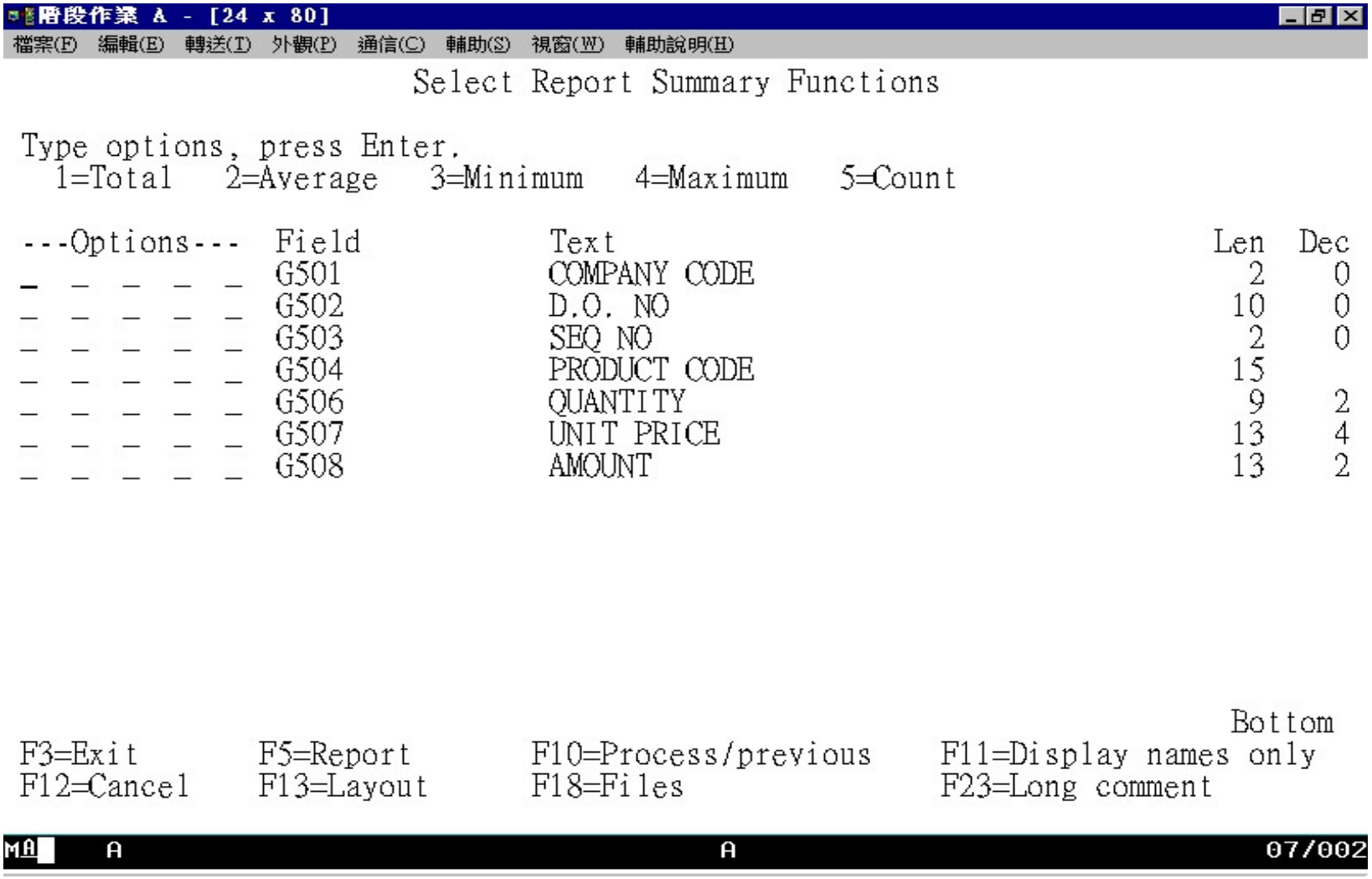

層段作業 A - [24 x 80]
檔案(F) 編輯(E) 轉送(T) 外觀(P) 通信(C) 輔助(S) 視窗(W) 輔助說明(H)
Describe Numeric Field Editing
Field . . . . . : G501
Type choices, press Enter.
Decimal point . . . . . 1 1=. 2=, 3=: 4=$ 5=None
Thousands separator . . . . . 2 1=. 2=, 3=' 4=Blank 5=None
Show negative sign . . . . . Y Y=Yes, N=No
Left negative sign . . . .
Right negative sign . . . . -
Show currency symbol . . . . N Y=Yes, N=No
Left currency symbol . . . $
Right currency symbol . . .
Print zero value . . . . . Y Y=Yes, N=No
Replace leading zeros . . . . Y Y=Yes, N=No
Replace with . . . . . 1 1=Blanks
2=Asterisks
3=Floating currency symbol
Single leading zero . . . . N Y=Yes, N=No
F3=Exit F5=Report F10=Process/previous F12=Cancel
F13=Layout F16=Remove edit F18=Files
MA A A 07/036


階段作業 A - [24 x 80]
檔案(F) 編輯(E) 轉送(T) 外觀(P) 通信(C) 輔助(S) 視窗(W) 輔助說明(H)
Describe Date/Time Field Editing
Field . . . . . : G501
Type choice, press Enter.
Date/time separator . . . . . 4 1=. 2=/ 3=: 4=- 5=,
F3=Exit F5=Report F10=Process/previous F12=Cancel
F13=Layout F16=Remove edit F18=Files
MA A A 07/036

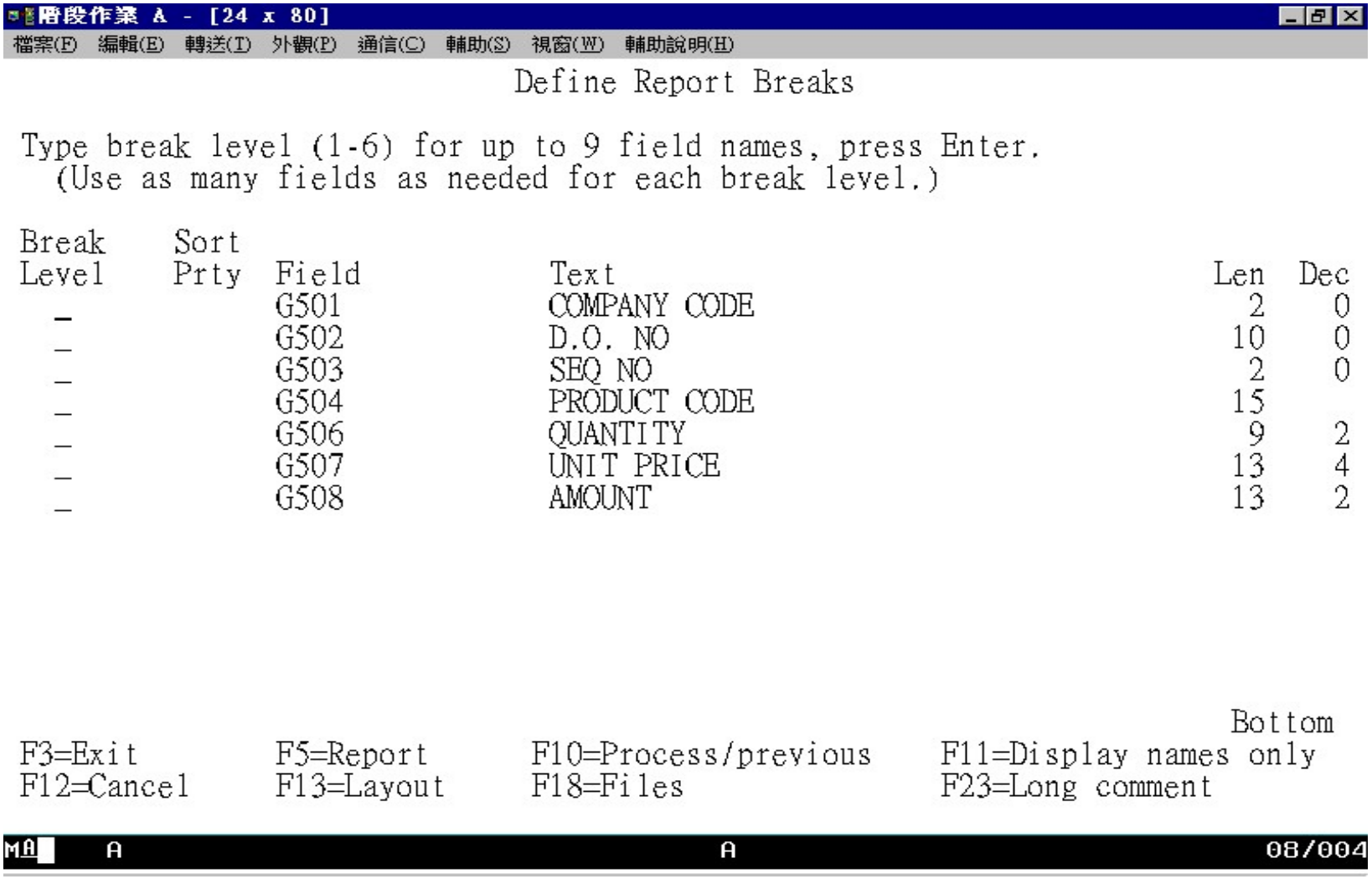

階段作業 A - [24 x 80]
檔案(F) 編輯(E) 轉送(T) 外觀(P) 通信(C) 輔助(S) 視窗(W) 輔助說明(H)
Specify Edit Code
Field . . . . . : G501
Type choices, press Enter.
Edit code . . . . . I 1-4, A-D, J-Q, X-Z, user-defined 5-9
Optional edit code
modifier . . . . . _ 1=Asterisk fill
2=Floating currency symbol
F3=Exit F5=Report F10=Process/previous F12=Cancel
F13=Layout F16=Remove edit F18=Files
MA A A 07/028

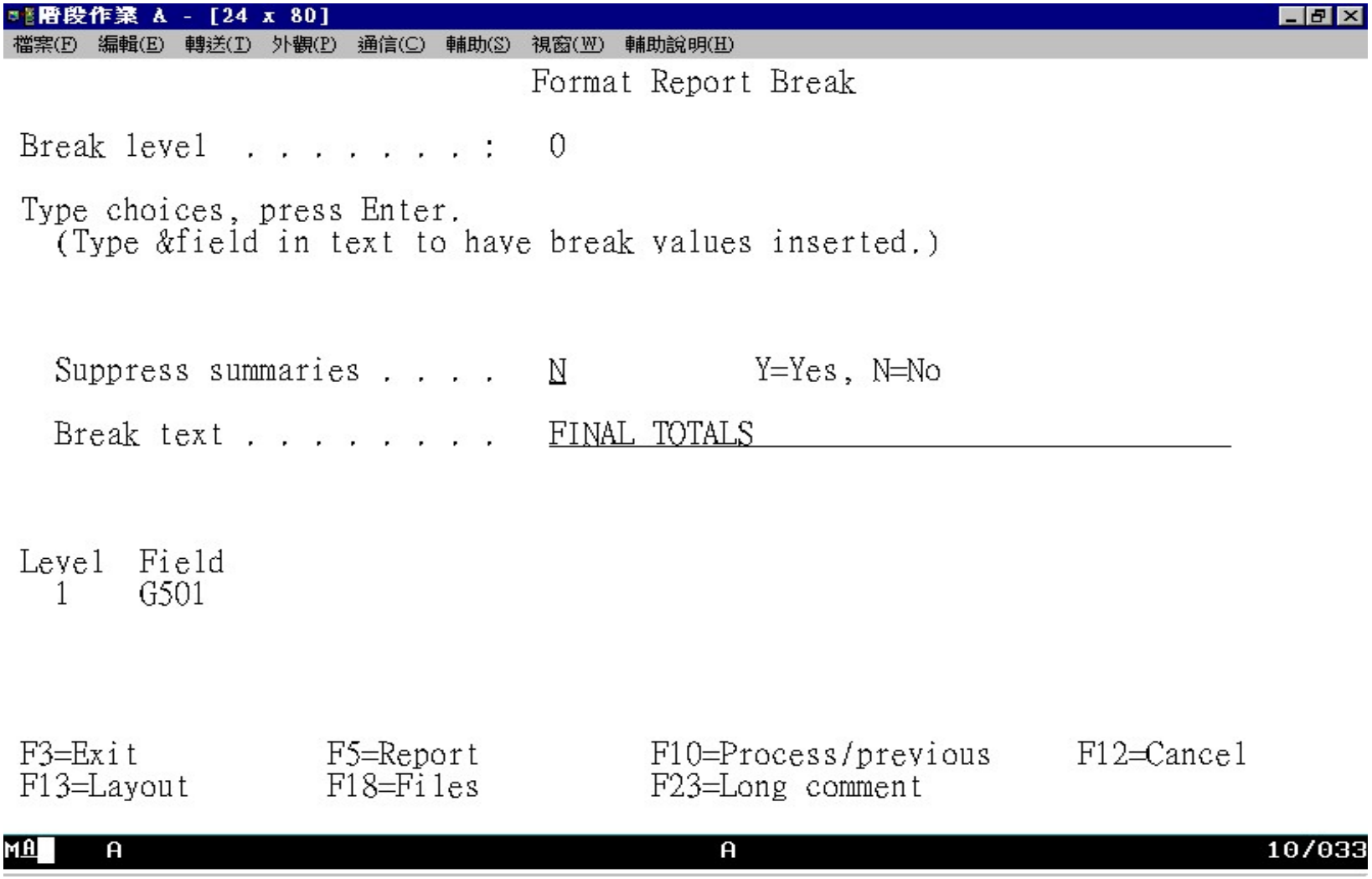

層段作業 A - [24 x 80]
檔案(F) 編輯(E) 轉送(T) 外觀(P) 通信(C) 輔助(S) 視窗(W) 輔助說明(H)
Specify Edit Word
Field . . . . . : G501
Length . . . . . : 2
Decimal . . . . . : 0
Heading 1 . . . . : COMPANY CODE
Heading 2 . . . . :
Heading 3 . . . . :
Type information, press Enter. (Put quotes around edit words.)
(Each blank replaced by a digit, each '&' with a blank.)
Edit word . . . . . ' '
Edit word for
summary total . .
F3=Exit
F5=Report
F10=Process/previous
F12=Cancel
F13=Layout
F16=Remove edit
F18=Files
MA A A 10/026

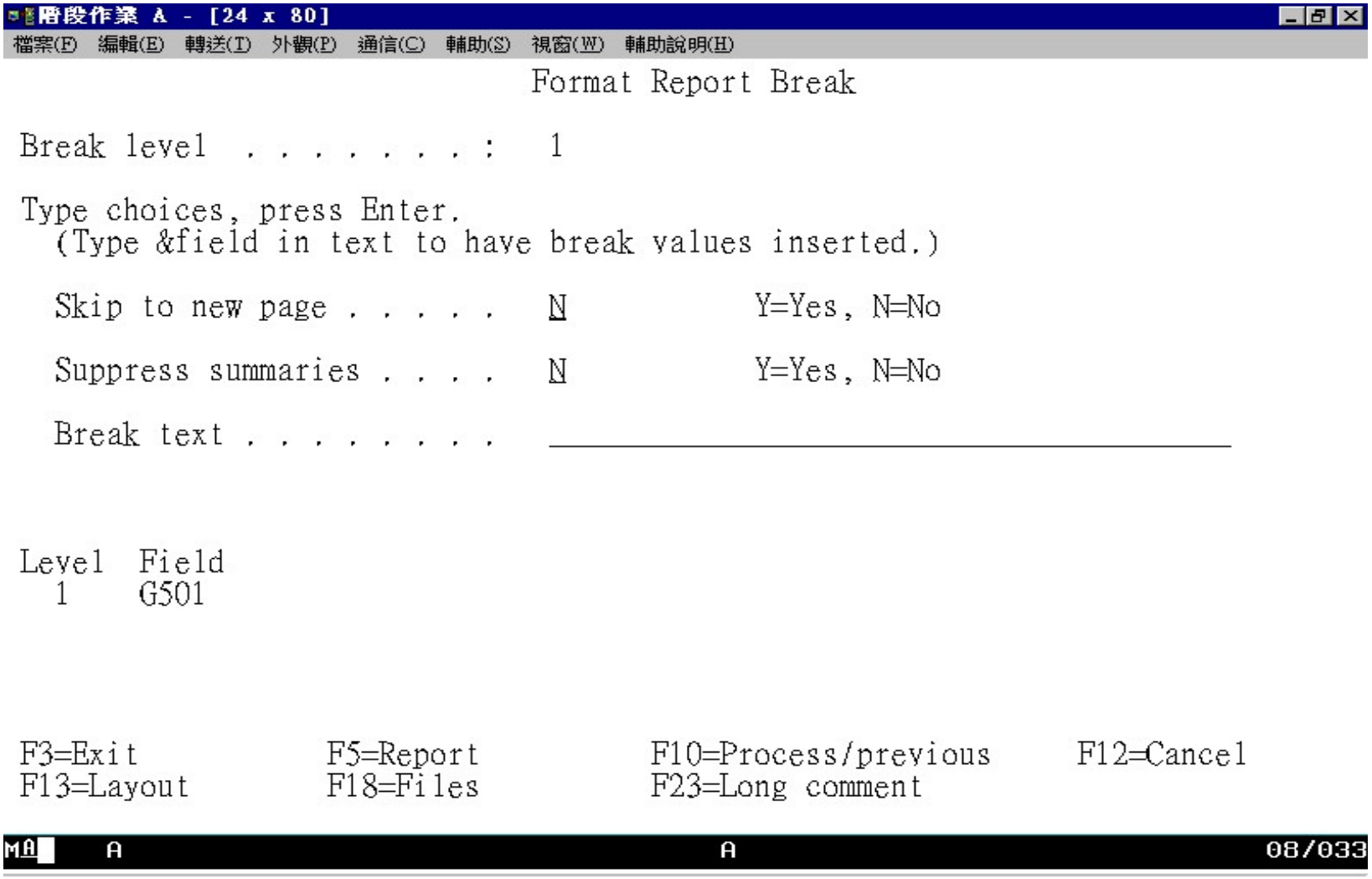
選擇報表摘要功能


層段作業 A - [24 x 80]
檔案(F) 編輯(E) 轉送(T) 外觀(P) 通信(C) 輔助(S) 視窗(W) 輔助說明(H)
Select Report Summary Functions
Type options, press Enter.
1=Total 2=Average 3=Minimum 4=Maximum 5=Count
---Options--- Field Text Len Dec
_ _ _ _ _ G501 COMPANY CODE 2 0
_ _ _ _ _ G502 D.O. NO 10 0
_ _ _ _ _ G503 SEQ NO 2 0
_ _ _ _ _ G504 PRODUCT CODE 15
_ _ _ _ _ G506 QUANTITY 9 2
_ _ _ _ _ G507 UNIT PRICE 13 4
_ _ _ _ _ G508 AMOUNT 13 2
Bottom
F3=Exit F5=Report F10=Process/previous F11=Display names only
F12=Cancel F13=Layout F18=Files F23=Long comment
MA A A 07/002

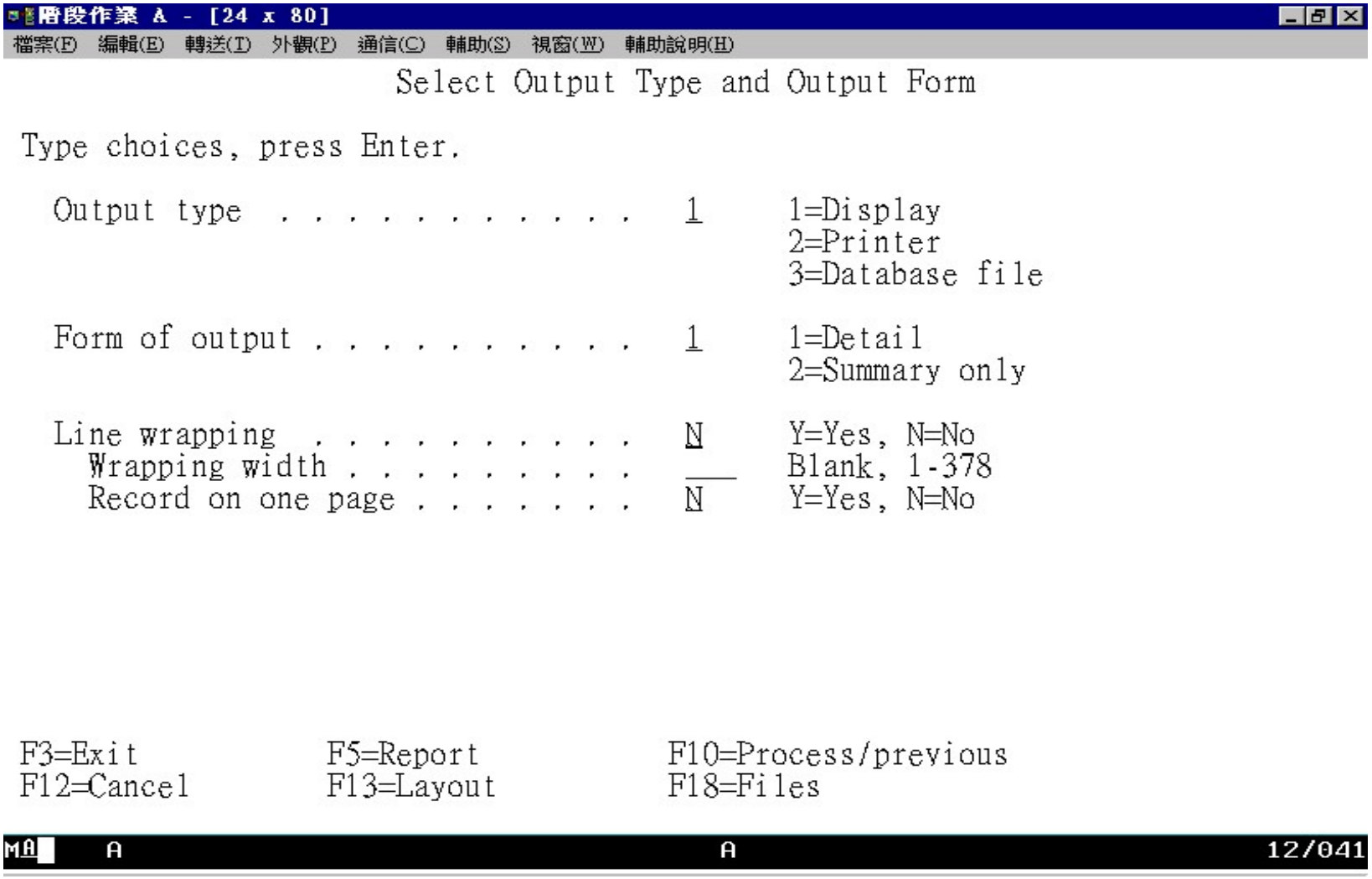
定義階層說明


層段作業 A - [24 x 80]
檔案(F) 編輯(E) 轉送(T) 外觀(P) 通信(C) 輔助(S) 視窗(W) 輔助說明(H)
Define Report Breaks
Type break level (1-6) for up to 9 field names, press Enter.
(Use as many fields as needed for each break level.)
Break Level
Sort Prty
Field
Text
Len
Dec
G501 COMPANY CODE 2 0
G502 D.O. NO 10 0
G503 SEQ NO 2 0
G504 PRODUCT CODE 15
G506 QUANTITY 9 2
G507 UNIT PRICE 13 4
G508 AMOUNT 13 2
Bottom
F3=Exit
F5=Report
F10=Process/previous
F11=Display names only
F12=Cancel
F13=Layout
F18=Files
F23=Long comment
MA A A 08/004

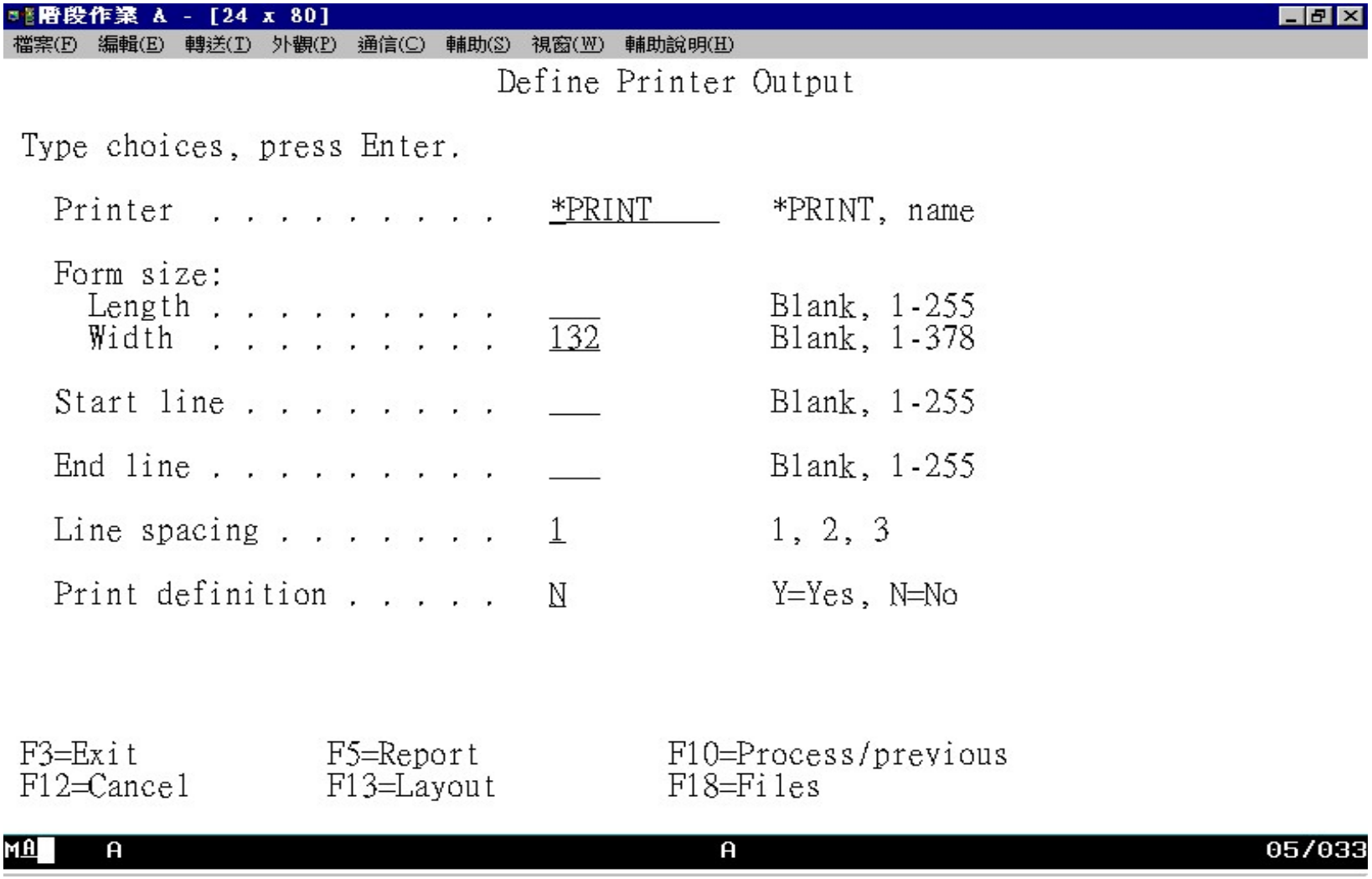

層段作業 A - [24 x 80]
檔案(F) 編輯(E) 轉送(T) 外觀(P) 通信(C) 輔助(S) 視窗(W) 輔助說明(H)
Format Report Break
Break level . . . . . : 0
Type choices, press Enter.
(Type &field in text to have break values inserted.)
Suppress summaries . . . . N Y=Yes, N=No
Break text . . . . . FINAL TOTALS
Level Field
1 G501
F3=Exit F5=Report F10=Process/previous F12=Cancel
F13=Layout F18=Files F23=Long comment
MA A A 10/033


層段作業 A - [24 x 80]
檔案(F) 編輯(E) 轉送(T) 外觀(P) 通信(C) 輔助(S) 視窗(W) 輔助說明(H)
Format Report Break
Break level . . . . . : 1
Type choices, press Enter.
(Type &field in text to have break values inserted.)
Skip to new page . . . . . N Y=Yes, N=No
Suppress summaries . . . . N Y=Yes, N=No
Break text . . . . . _____
Level Field
1 G501
F3=Exit F5=Report F10=Process/previous F12=Cancel
F13=Layout F18=Files F23=Long comment
MA A A 08/033

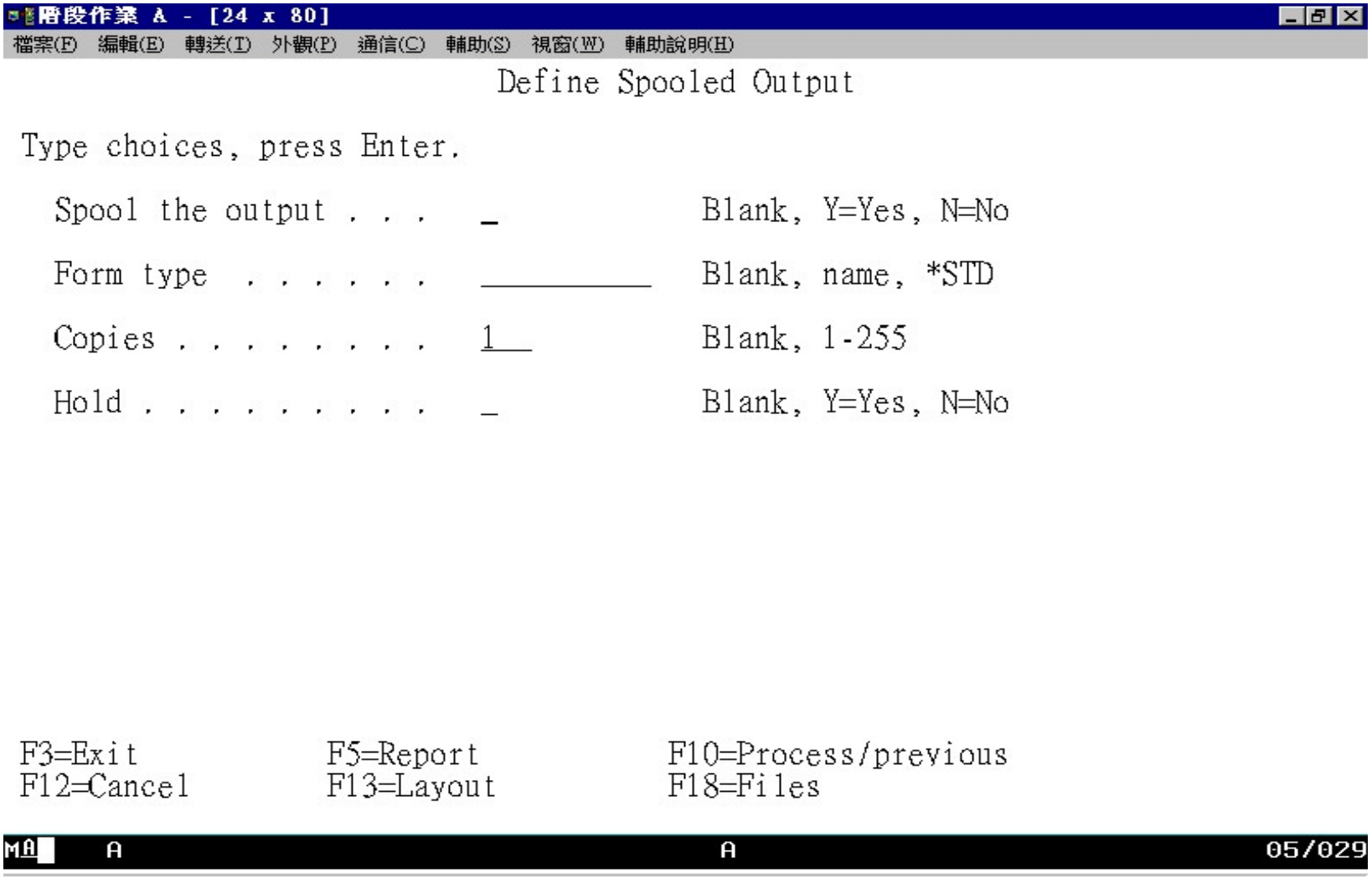
選擇輸出方式


階段作業 A - [24 x 80]
檔案(F) 編輯(E) 轉送(T) 外觀(P) 通信(C) 輔助(S) 視窗(W) 輔助說明(H)
Select Output Type and Output Form
Type choices, press Enter.
Output type . . . . . 1 1=Display
2=Printer
3=Database file
Form of output . . . . . 1 1=Detail
2=Summary only
Line wrapping . . . . . N Y=Yes, N=No
Wrapping width . . . . . Blank, 1-378
Record on one page . . . . . N Y=Yes, N=No
F3=Exit F5=Report F10=Process/previous
F12=Cancel F13=Layout F18=Files
MA A A 12/041

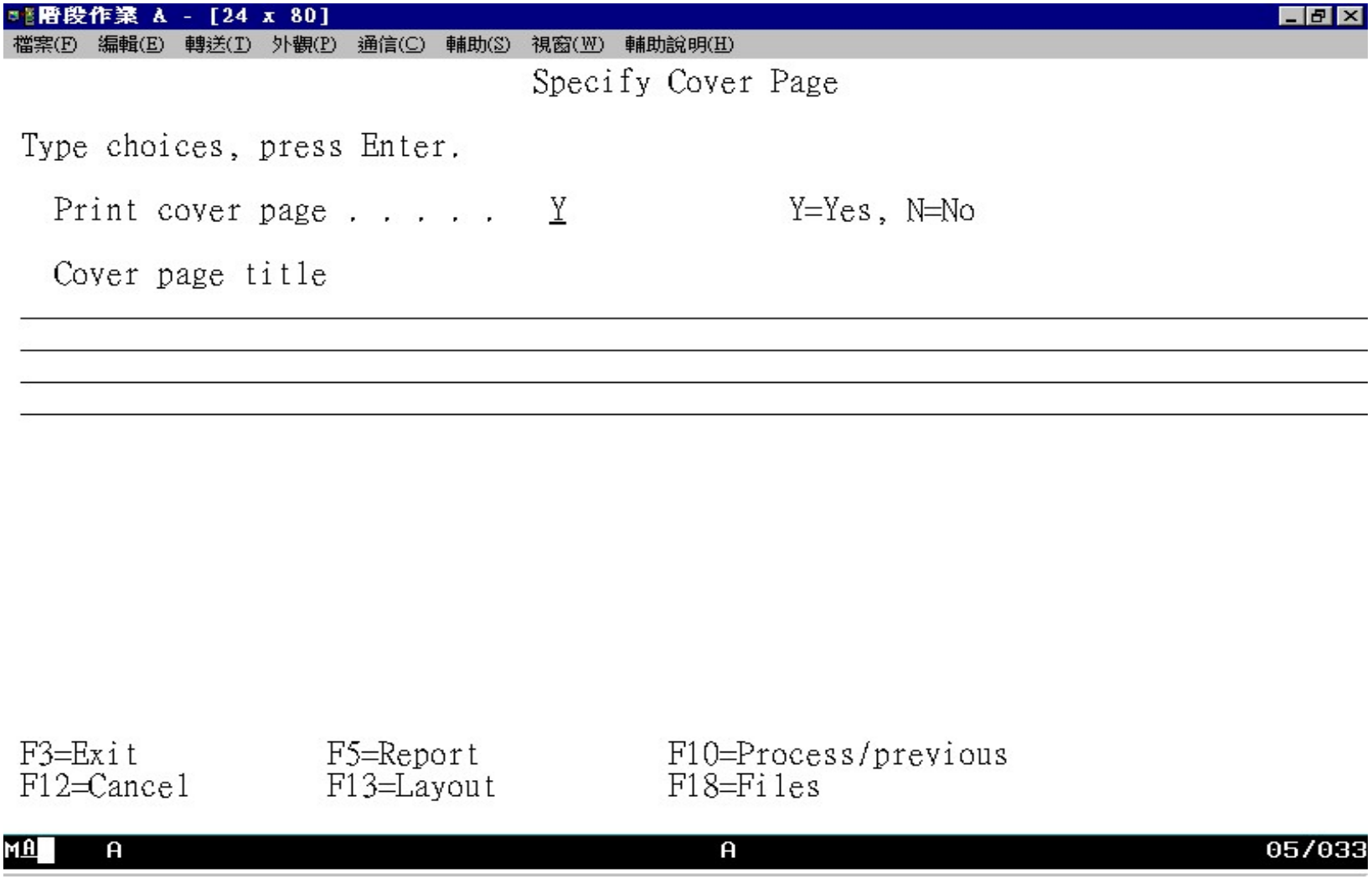
報表設定


暫段作業 A - [24 x 80]
檔案(F) 編輯(E) 轉送(T) 外觀(P) 通信(C) 輔助(S) 視窗(W) 輔助說明(H)
Define Printer Output
Type choices, press Enter.
Printer . . . . . *PRINT *PRINT, name
Form size:
Length . . . . . Blank, 1-255
Width . . . . . 132 Blank, 1-378
Start line . . . . . Blank, 1-255
End line . . . . . Blank, 1-255
Line spacing 1 1, 2, 3
Print definition . . . . . N Y=Yes, N=No
F3=Exit F5=Report F10=Process/previous
F12=Cancel F13=Layout F18=Files
MA A A 05/033

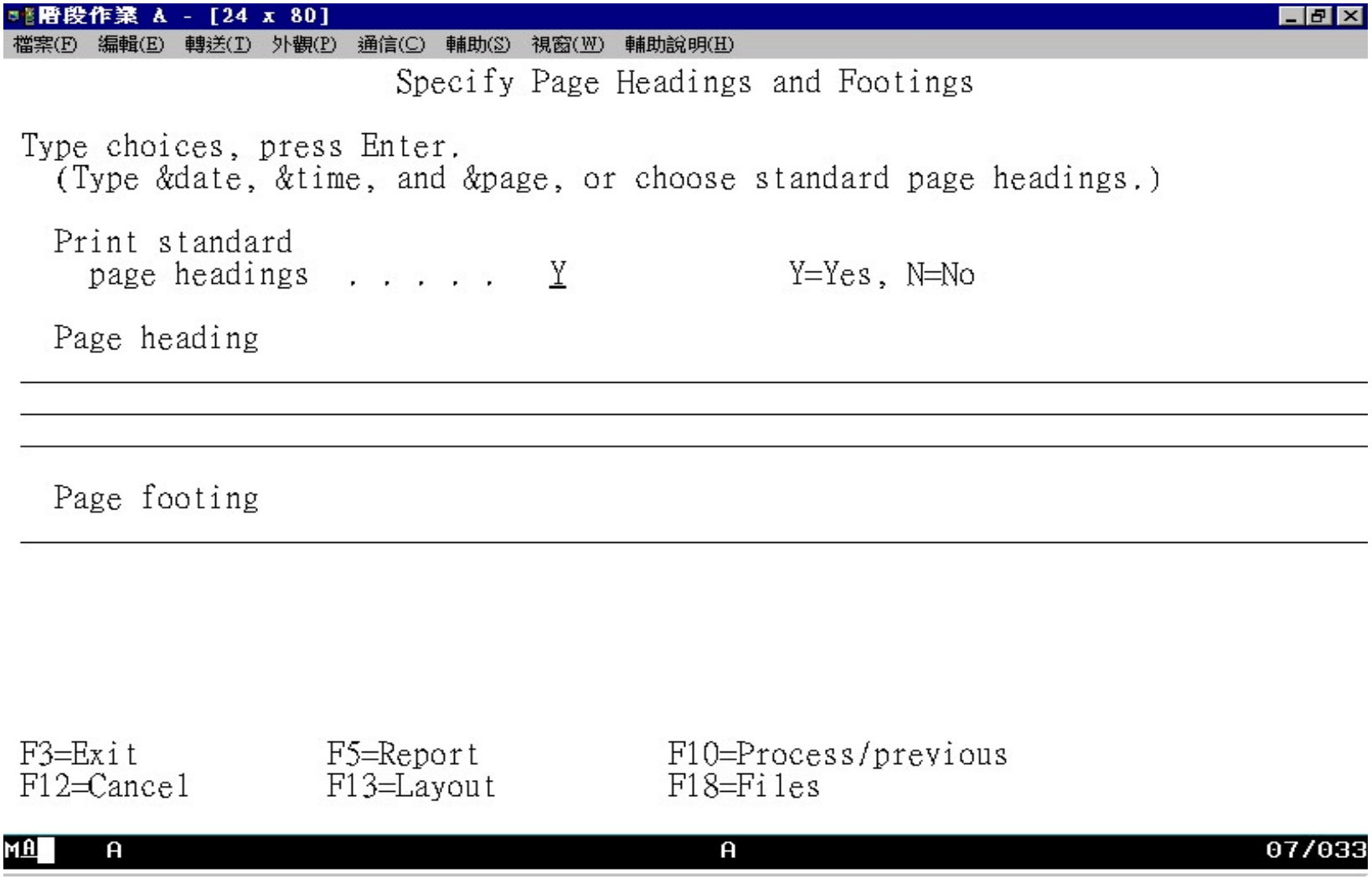
報表輸出設定


階段作業 A - [24 x 80]
檔案(F) 編輯(E) 轉送(T) 外觀(P) 通信(C) 輔助(S) 視窗(W) 輔助說明(H)
Define Spooled Output
Type choices, press Enter.
Spool the output . . . _ Blank, Y=Yes, N=No
Form type . . . . . _____ Blank, name, *STD
Copies . . . . . 1__ Blank, 1-255
Hold . . . . . _ Blank, Y=Yes, N=No
F3=Exit F5=Report F10=Process/previous
F12=Cancel F13=Layout F18=Files
MA A A 05/029

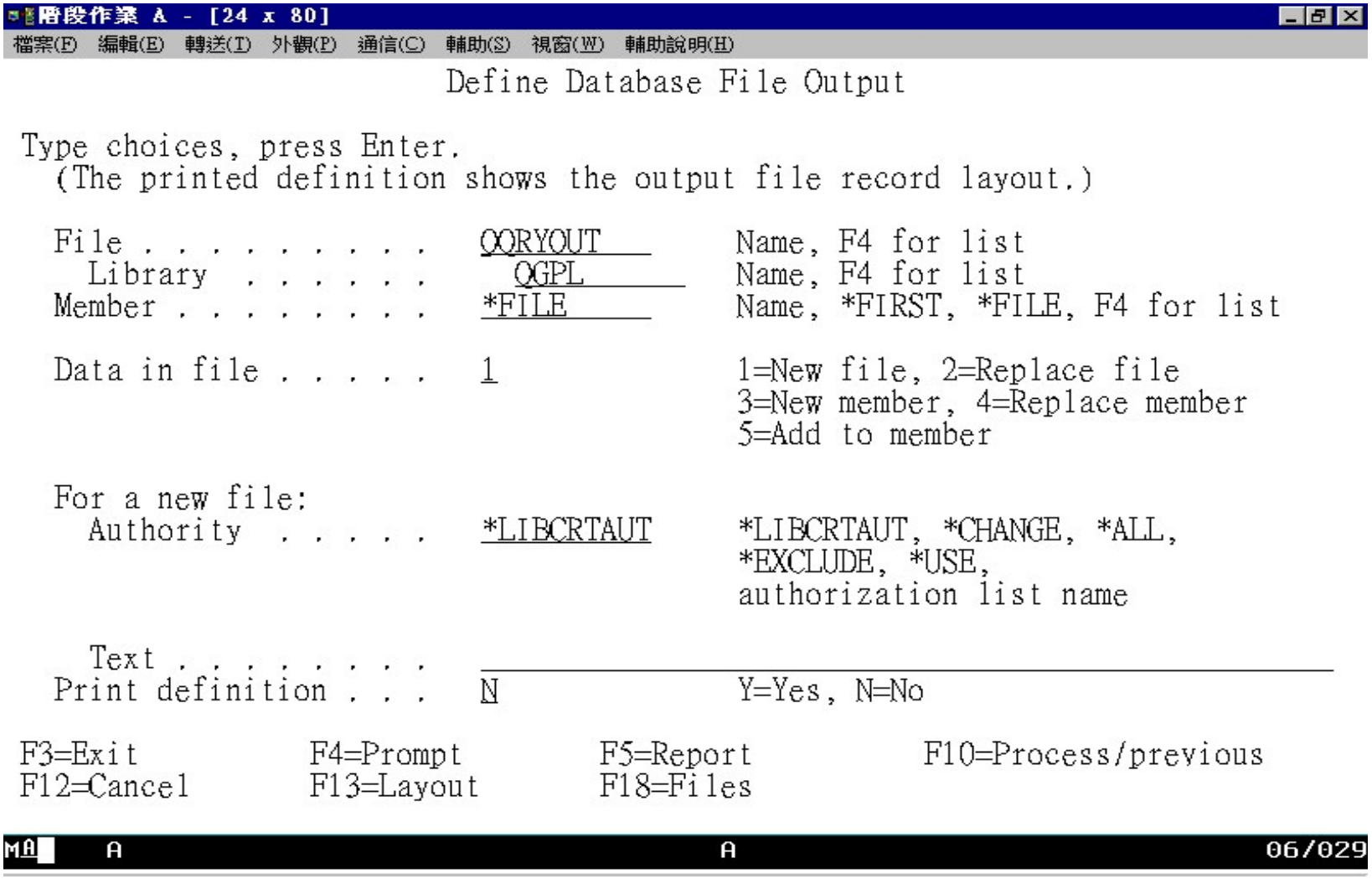
封面設定


階段作業 A - [24 x 80]
檔案(F) 編輯(E) 轉送(T) 外觀(P) 通信(C) 輔助(S) 視窗(W) 輔助說明(H)
Specify Cover Page
Type choices, press Enter.
Print cover page . . . . . Y Y=Yes, N=No
Cover page title
F3=Exit F5=Report F10=Process/previous
F12=Cancel F13=Layout F18=Files
MA A A 05/033

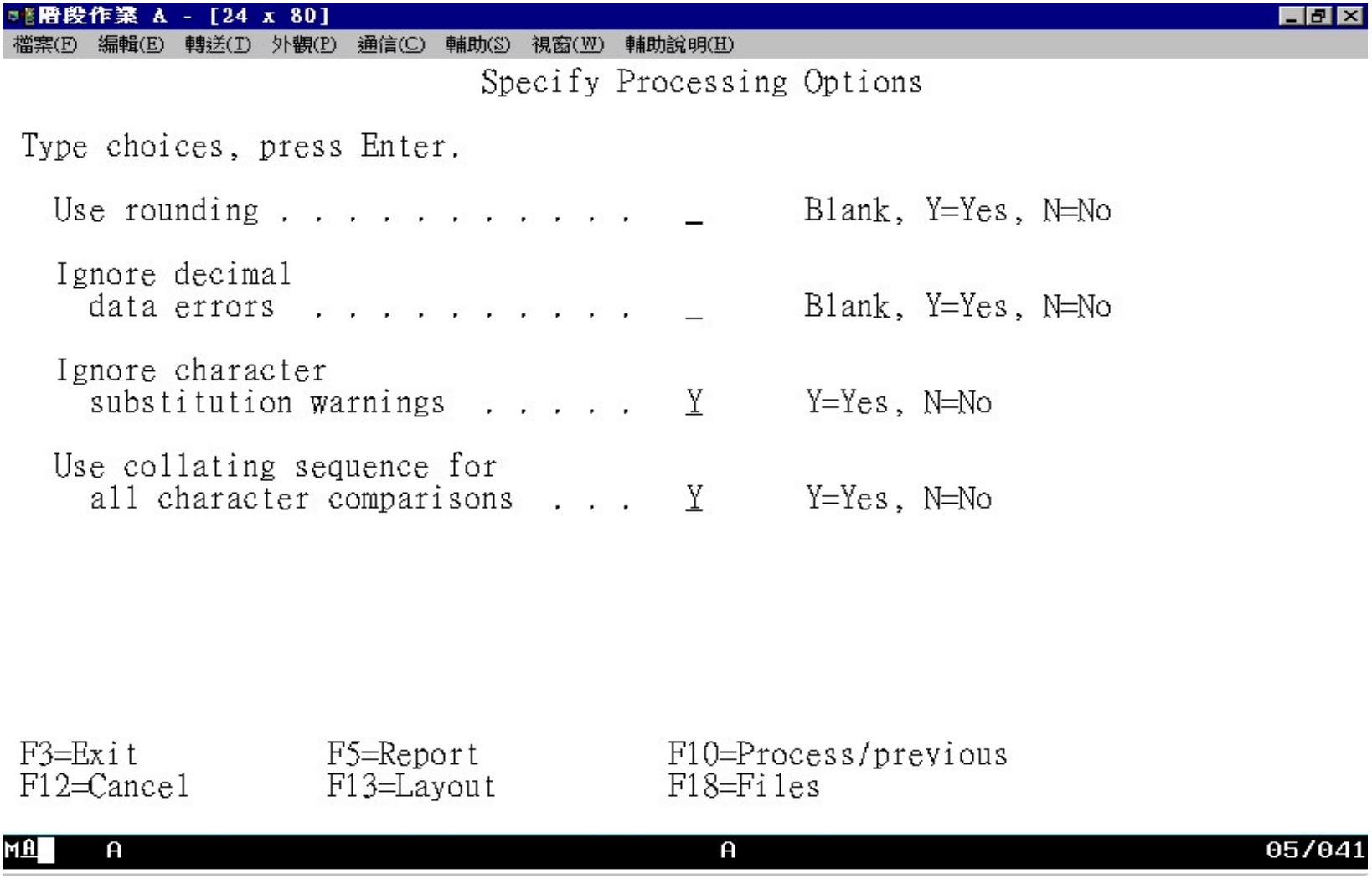
報頭表尾設定


層段作業 A - [24 x 80]
檔案(F) 編輯(E) 轉送(T) 外觀(P) 通信(C) 輔助(S) 視窗(W) 輔助說明(H)
Specify Page Headings and Footings
Type choices, press Enter.
(Type &date, &time, and &page, or choose standard page headings.)
Print standard
page headings . . . . . Y Y=Yes, N=No
Page heading
Page footing
F3=Exit F5=Report F10=Process/previous
F12=Cancel F13=Layout F18=Files
MA A A 07/033

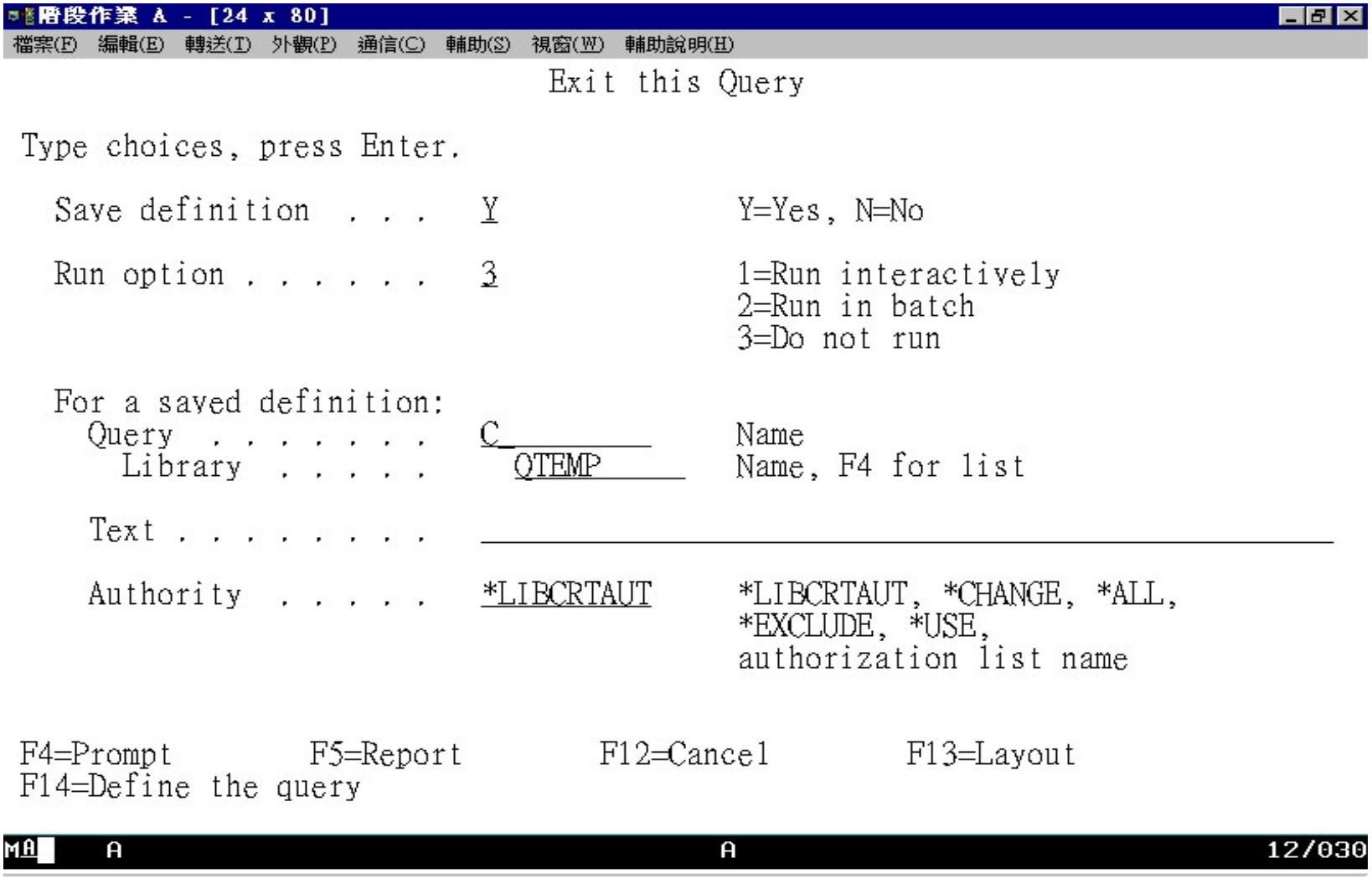
定義資料庫輸出


階段作業 A - [24 x 80]
檔案(F) 編輯(E) 轉送(T) 外觀(P) 通信(C) 輔助(S) 視窗(W) 輔助說明(H)
Define Database File Output
Type choices, press Enter.
(The printed definition shows the output file record layout.)
File . . . . . QQRYOUT Name, F4 for list
Library . . . . . QGPL Name, F4 for list
Member . . . . . *FILE Name, *FIRST, *FILE, F4 for list
Data in file . . . . . 1 1=New file, 2=Replace file
3=New member, 4=Replace member
5=Add to member
For a new file:
Authority . . . . . *LIBCRTAUT *LIBCRTAUT, *CHANGE, *ALL,
*EXCLUDE, *USE,
authorization list name
Text . . . . . .
Print definition . . . N Y=Yes, N=No
F3=Exit F4=Prompt F5=Report F10=Process/previous
F12=Cancel F13=Layout F18=Files
MA A A 06/029

設定處理選項


階段作業 A - [24 x 80]
檔案(F) 編輯(E) 轉送(T) 外觀(P) 通信(C) 輔助(S) 視窗(W) 輔助說明(H)
Specify Processing Options
Type choices, press Enter.
Use rounding . . . . . _ Blank, Y=Yes, N=No
Ignore decimal
data errors . . . . . _ Blank, Y=Yes, N=No
Ignore character
substitution warnings . . . . . Y Y=Yes, N=No
Use collating sequence for
all character comparisons . . . Y Y=Yes, N=No
F3=Exit F5=Report F10=Process/previous
F12=Cancel F13=Layout F18=Files
MA A A 05/041

QUERY程式設定(結束)



第三章

練習

練習一

隨堂練習
檔案如附件一、附件二

七月份員工獎金分發明細表

日期:10/20/94
時間:15:32:04

部門別	員工編號	員工姓名	性別	出生日期	獎金
01	A1103	陳正成	男	1960/09/22	1500
	A1101	陳中田	女	1968/05/01	2000
				01部門合計:	
				TOTAL	3500
02	A1107	李敏敏	女	1968/02/18	3000
				02部門合計:	
				TOTAL	3000
03	A1108	黃介仁	男	1962/02/22	2000
				03部門合計:	
				TOTAL	2000
				總計:	
				TOTAL	8500

* * * E N D O F R E P O R T * * *

部門員工生日清單

日期:10/20/94
時間:14:59:26

部門別	員工編號	員工姓名	性別	出生日期
01	A1103	陳正成	男	1960/09/22
	A1102	林仲如	女	1964/04/02
	A1101	陳中田	女	1968/05/01
01部門合計:				
COUNT 3				
02	A1104	黃邦明	男	1962/05/15
	A1107	李敏敏	女	1968/02/18
	A1106	周小香	女	1968/03/17
	A1105	李貴夫	男	1968/04/15
02部門合計:				
COUNT 4				
03	A1108	黃介仁	男	1962/02/22
	A1109	李國彥	男	1962/08/24
03部門合計:				
COUNT 2				
總計:				
COUNT 9				

* * * E N D O F R E P O R T * * *

附件一

08/02/94 15:28:51 PAGE 1

部門別	員工編號	員工姓名	性別	出生日期	出生地
01	A1101	陳中田	女	19,680,501	台北市
01	A1102	林仲如	女	19,640,402	台北市
01	A1103	陳正成	男	19,600,922	台南市
02	A1104	黃邦明	男	19,620,515	台中市
02	A1105	李貴夫	男	19,680,415	台中市
02	A1106	周小香	女	19,680,317	高雄市
02	A1107	李敏敏	女	19,680,218	台北市
03	A1108	黃介仁	男	19,620,222	高雄市
03	A1109	李國彥	男	19,620,824	台北市

* * * E N D O F R E P O R T * * *

(GUIF/PYZAPF-員工基本資料檔)

附件二

08/02/94 15:35:44 PAGE 1

員工編號	月份	獎金
A1101	7	2,000
A1101	8	1,800
A1103	7	1,500
A1105	8	2,500
A1107	7	3,000
A1107	8	3,000
A1108	7	2,000
A1108	8	2,500

* * * E N D O F R E P O R T * * *

(GUIF/PYZBPF-員工獎金資料檔)

練習二

列印出每張出貨單之每項產品之毛利率(格式如附表一)

條件：1、產品成本於成本檔中(GUGEPF) Price 1或 Price2。

2、無Price 1以 Price2為成本。

3、以出貨單號為Level break。

4、以80 column 報表紙列印。

5、D.O. No.為256~260。

附件一

08/02/94 09:55:57			LAB1	PAGE1	
D.O.NO	PRODUCT CODE	QUANTITY	UNIT PRICE	AMOUNT	G.P.
256	30148PKR355	62.00	226.6670	14.053.35	99.46
			Delivery Order Total:		
			TOTAL	**14,053.35**	
257	30148LZK410	2.00	1,666.6670	3,333.33	24.56-
			Delivery Order Total:		
			TOTAL	**3,333.33**	
258	30148LZK410	2.00	1,666.6670	3,333.33	24.56-
	30148PKR355	65.00	226.6670	14,733.35	99.46
			Delivery Order Total:		
			TOTAL	**18,066.68**	
259	30148PKR355	5,000.00	209.5240	1,047,620.00	
	30148PKR355	400.00	.0000	.00	++++++
			Delivery Order Total:		
			TOTAL	**1,047.620.00**	

260	30148LZK410	9.00	1,933.3330	17,399.99	7.38-
	30148LZK410	1.00	.0000	.00	++++++
	30148NJG330	7.00	.0000	.00	++++++
	30148PKQ430	24.00	428.571	10,285.70	
				++++++	
	30148PKQ430	6.00	.0000	.00	++++++
	30148PKR355	200.00	262.0000	52,400.00	99.53
	30148PKR355	60.00	.0000	.00	++++++
	30148PZN580	1.00	2,375.2380	2,375.23	81.05
	30148PZR060	3.00	1,979.0480	5,937.14	53.01
	30161GXH530	5.00	1,042.8570	5,214.28	99.07-

08/02/94	09:55:57		LAB1	PAGE 2	
D.O.NO	PRODUCT CODE	QUANTITY	UNIT PRICE	AMOUNT	G.P.
260	30161GXM530	1.00	.0000	.00	++++++

Delivery Order Total:	
TOTAL	93,612.34
FINAL TOTALS	
TOTAL	1,176,685.70

* * * E N D O F R E P O R T * * *

檔案-ID:GUGEPF

存放的LAB:GUIF

檔案名稱 FILE-NAME:產品價格資料庫檔

Internal Field Name	Field Length	Decimal Positions	Field Type	Column Heading 1	
GE01	6	0	S	CUSTOMER NO	客戶代號
GE02	1	0	A	CUSTOMER SUB-NO	分店代號
GE03	15	0	A	PRODUCT CODE	產品代號
GE04	8	0	S	EFFECTIVE DATA	產品售價有效日期
GE05	7	4	P	UNIT PRICE1	應稅價格
GE06	7	4	P	UNIT PRICE2	免稅價格
GE07	4	3	P	BONUS %	佣金
GE08	1	0	A	INVOICE TYPE	發票聯式
GE97	6	0	S	UPDATE TINE	異動時間
GE98	8	0	S	UPDATA DATE	異動日期
GE99	10	0	A	UPDATE USERID	異動者

* * * E N D O F R E P O R T * * *

劃底線者為QRY所需使用之欄位

檔案-ID:GUG3PF

存放的LIB:GUIF

檔案名稱 FILE-NAME:產品主檔

Internal Field Name	Field Length	Decimal Positions	Field Type	Column Heading 1	
G301	15	0	A	PRODUCT CODE	產品代號
G302	40	0	0	PRODUCT NAME 1	產品名稱一
G303	40	0	0	PRODUCT NAME 2	產品名稱二
G397	6	0	S	UPDATE TINE	異動時間
G398	8	0	S	UPDATA DATE	異動日期
G399	10	0	A	UPDATE USERID	異動者

＊ ＊ ＊　E N D　O F　R E P O R T　＊ ＊ ＊

劃底線者為QRY所需使用之欄位

檔案-ID：GUG4PF
存放的LIB：GUIF
檔案名稱
FILE-NAME：
出貨單主檔

Internal Field Name	Filed Length	Decimal Positions	Field Type	Column Heading 1	
G401	2	0	S	COMPANY CODE	公司別
G402	6	0	P	D.O. NO	出貨單號
G403Y	4	0	S	D.O. NO DATE-YEAR	出貨單號日期-年
G403M	2	0	S	D.O. NO DATE-MONTH	出貨單號日期-月
G403D	2	0	S	D.O. NO DATE-DAY	出貨單號日期-日
G404	6	0	S	CUSTOMER NO	客戶代號
G404S	1	0	A	CUSTOMER SUB-NO	分店代號
G405Y	4	0	S	INVOICE DATE-YEAR	發票日期-年
G405M	2	0	S	INVOICE DATE-MONTH	發票日期-月
G405D	2	0	S	INVOICE DATE-DAY	發票日期-日
G406	7	2	P	D.O. TOTAL AMOUNT	出貨總金額
G407	1	0	A	RELEADE TYPE	單據種類
G408	10	0	A	RELEASE DOC. NO	單據號碼
G409	8	0	S	RELEASE DATE	單據日期
G497	6	0	S	UPDATE TIME	異動時間
G498	8	0	S	UPDATE DATE	異動日期
G499	10	0	A	UPDATE USERID	異動者

* * * E N D O F R E P O R T * * *

劃底線者為QRY所需使用之欄位

檔案-ID：GUG5PF
存放的LIB：GUIF
檔案名稱
FILE-NAME：
出貨單明細檔

G501	2	0	S	COMPANY CODE	公司別
G502	6	0	P	D.O. NO	出貨單別
G503	2	0	S	SEQ NO	序號
G504	15	0	A	PRODUCT CODE	產品代號
G506	5	2	P	QUANTITY	數量
G507	7	4	P	UNIT PRICE	售價
G508	7	2	P	AMOUNT	金額

* * * E N D O F R E P O R T * * *

劃底線者為QRY所需使用之欄位


單元二
SQL

目 錄

第一章

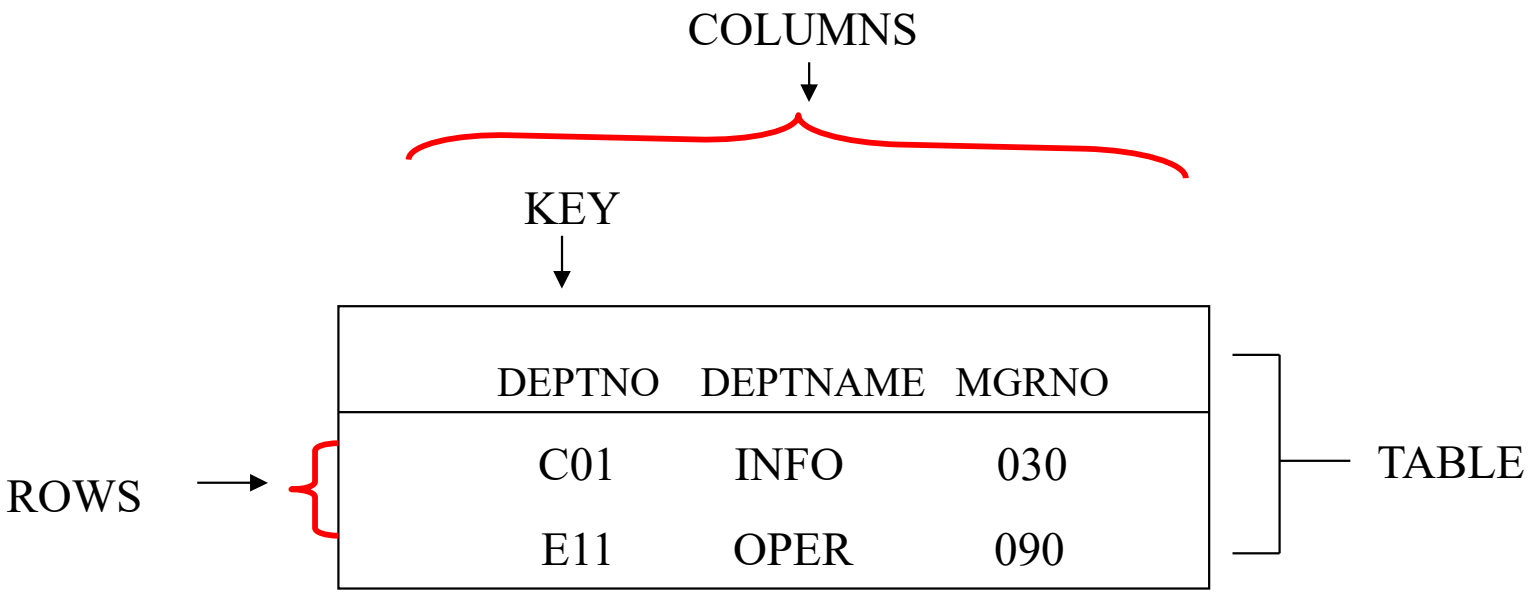
INTRODUCTION

RELATIONAL DATABASE MANAGEMENT SYSTEM

DATA PERCEIVED AS TABLES

LANGUAGE TO MANIPULATE DATA

TABLE TERMS



ENTITY： DEPARTMENT

SQL

STRUCTURED QUERY LANGUAGE

ENGLISH LIKE

EASY TO WRITE , READ , AND UNDERSTAND

SQL COMPONENTS

DML - DATA MANIPULAION LANGUAGE

- SELECT
- UPDATE
- INSERT
- DELETE

DDL - DATA DEFINITION LANGUAGE

- CREATE
- ALTER
- DROP

CONTROL

- GRANT
- REVOKE

OBJECTIVES

After completing this course, the student should be able to use SQL to

Retrieve information from a relational database management system table.

Delete, Insert, and update data in tables and views.

Create table with varying data types having varying null characteristics.

Create other relational objects.

第二章

SELECT

TABLES USED IN EXAMPLES

Q. STAFF

ID	NAME	DEPT	JOB	YEARS	SALARY	COMM
10	SANDERS	20	MGR	7	18357.50	--
20	PERNAL	20	SALES	8	18171.25	612.45
30	MARENGHI	38	MGR	5	17506.75	--
40	O'BRIEN	38	SALES	6	18006.00	846.55
50	HANES	15	MGR	10	20659.80	--
60	QUIGLEY	38	SALES	--	16808.30	850.25
70	POTHMAN	15	SALES	7	16502.83	1152.00
80	JAMES	20	CLERK	--	13504.60	128.20
90	KOONITZ	42	SALES	6	18001.75	1386.70
100	PLOTZ	42	MGR	7	18352.80	--
110	NGAN	15	CLERK	5	12508.20	206.50
120	NAUGHTON	38	CLERK	--	12954.75	180.00
130	YAMAGUCHI	42	CLERK	6	10505.90	75.60
140	FRAYE	51	MGR	6	21150.00	--
150	WILLIAMS	51	SALES	6	19456.50	637.65
160	MOLINARE	10	MGR	7	22959.20	--
170	KERMISCH	15	CLERK	4	12258.50	110.10
180	ABRAHAMS	38	CLERK	3	12009.75	236.50
190	SNEIDER	20	CLERK	8	14252.75	126.50
200	SCOUTTEN	42	CLERK	--	11508.60	84.20
210	LU	10	MGR	10	20010.00	--
220	SMITH	51	SALES	7	17654.50	992.80
230	LUNDQUIST	51	CLERK	3	13369.80	189.65
240	DANIELS	10	MGR	5	19260.25	--
250	WHEELER	51	CLERK	6	14460.00	513.30
260	JONES	10	MGR	12	21234.00	--
270	LEA	66	MGR	8	18555.50	--
280	WILSON	66	SALES	9	18674.50	811.50
290	QUILL	84	MGR	10	19818.00	--
300	DAVIS	84	SALES	5	15454.50	806.10
310	GRAHAM	66	SALES	13	21000.00	200.30
320	GONZALES	66	SALES	4	16858.20	844.00
330	BURKE	66	CLERK	1	10988.00	55.50
340	EDWARDS	84	SALES	7	17844.00	1285.00
350	GAFNEY	84	CLERK	5	13030.50	188.00

Q. ORG

DEPTNUMB	DEPTNAME	MANAGER	DIVISION	LOCATION
10	HEAD OFFICE	160	CORPORATE	NEW YORK
15	NEW ENGLAND	50	EASTERN	BOSTON
20	MID ATLANTIC	10	EASTERN	WASHINGTON
38	SOUTH ATLANTIC	30	EASTERN	ATLANTA
42	GREAT LAKES	100	MIDWEST	CHICAGO
51	PLAIN	140	MIDWEST	DALLAS
66	PACIFIC	270	WESTERN	SAN FRANCISCO
84	MOUNTAIN	290	WESTERN	DENVER

QUERY

REQUIRED SEQUENCE：

```
SELECT  NAME, YEARS,        --- Tell which column(s) to use
        SALARY
FROM    Q.STAFF             --- Tell which table(s) to use
WHERE   DEPT = 38           --- Tell which row(s) to use
ORDER BY NAME               --- Tells how to sequence
                            --- The result
```

FREE FORM QUERY

```
SELECT NAME , YEARS , SALARY FROM Q.STAFF WHERE
DEPT = 38 ORDER BY NAME
```

SELECT … FROM

SELECT some of the columns from a table

- Name the column(s) that you want in the result, in left-to-right sequence
- Commas separate multiple names

FROM names the table(s)

- Simple: TABLENAME
- Fully Qualified: AUTHID.TABLENAME

EXAMPLE

```
SELECT DEPARTNAME, DEPTNUMB
FROM Q.ORG
```

Result:

DEPTNAME	DEPTNUMB
HEAD OFFI CE	10
NEW ENGLAND	15
MID ATLANTIC	20
SOUTH ATLANTC	38
.	.
.	.

SELECT ALL OF THE COLUMNS

To select all the the columns form a table in the same left-to right sequence as originally defined:

```
SELECT    *
FROM Q.ORG
```

Result:

DEPTNUMB	DEPTNAME	MANAGER	DIVISION	LOCATION
10	HEAD OFFI CE	160	CORPORATE	NEW YORK
15	NEW ENGLAND	50	EASTERN	BOSTON
20	MID ATLANTIC	10	EASTERN	WASHINGTON
38	SOUTH ATLANTC	30	EASTERN	ALTLANTA
.	.	.	.	.
.	.	.	.	.

ROW CONTROL

WHERE

- Retrieves certain rows
- States the condition(s)

List only employees in department 20

```
SELECT DEPT, NAME,
    JOB, COMM
    FROM Q.STAFF
    WHERE DEPT = 20
```

Result:

DEPT	NMAE	JOB	COMM
20	SANDERS	MGR	-
20	PERNAL	SALES	612.45
20	KAMES	CLERK	128.20
20	SNEIFER	CLERK	126.50

WHERE EXAMPLES

Select rows by numeric data

```
SELECT DEPT, NAME,
   JOB
   FROM Q.STAFF
WHERE DEPT = 20
```

Select rows by character data

```
SELECT NAME, JOB, COMM
FROM Q.STAFF
WHERE NAME = 'SANDERS'
```

COMPARISION OPERATORS

=	Equal
<>	Not equal
>	Greater than
>=	Greater than or equal
<	Less than
<=	Less than or equal

```
SELECT ID, COMM
FROM Q.STAFF

WHERE COMM > =  1000
```

```
SELECT NAME, DEPT, YEARS
FROM Q.STAFF

WHERE JOB  <>  'MGR'
```

Notes: Not SAA, but supported by DB2 and AS/400:

¬ > Not greater than

¬ < Not less than

Not SAA, but supported by DB2, SQL/DS and AS/400:

NULL CHARACTERISTIC

A column that is --

NOT NULL must be given a value

Blanks or zeros are acceptable

Nullable does not require a value and is marked NULL

A value of NULL is :
- Not zero
- Not blank
- Unknown

NOT NULL WITH DEFAULT

A column may be described as ‘not null with default’. The system provides a default value if one is not provided.

- Zero for a numeric column
- Blank for a fixed length character column
- Zero length for a variable length character column

NULL SELECTION

To select by null value

```
SELECT NAME, JOB, COMM
FROM Q.STAFF
WHERE COMM IS NULL
```

Result:

NAME	JOB	COMM
SANDERS	MGR	-
MARENGHI	MGR	-
HANES	MGR	-
PLOTZ	MGR	-
.	.	.

To select by not null value

```
SELECT NAME, JOB, COMM
FROM Q.STAFF
WHERE COMM IS NOT NULL
```

Result:

NAME	JOB	COMM
PERNAL	SALES	612.45
O'BRIEN	SALES	846.55
QUIGLEY	SALES	650.25
ROTHMAN	SALES	1152.00
JAMES	CLERK	128.20
.	.	.

MULTIPLE CONDITIONS

Multiple conditin retrieval: AND, OR

Example - given two conditions:

JOB = 'SALES' SALARY < 17000

To have both conditions met: AND

WHERE JOB = 'SALES' AND SALARY < 17000

Result:

NAME	JOB	SALARY
QUIGLEY	SALES	16808.30
ROTHMAN	SALES	16502.83
.	.	.

To have either or both conditions met: OR

WHERE JOB = 'SALES' OR SALARY < 17000

Result:

NAME	JOB	SALARY
PERNAL	SALES	18171.25
O'BRIEN	SALES	18006.00
QUIGLEY	SALES	16808.30
ROTHMAN	SALES	16502.83
JAMES	CLERK	13504.60
.	.	.
.	.	.

MULTIPLE CONDITIONS (cont)

```
WHERE (JOB = 'SALES' AND COMM > 1200)
   OR YEARS > 10
```

Result:

NAME	JOB	YEARS	COMM
KOONITZ	SALES	6	1386.70
JONES	MGR	12	-
GRAHAM	SALES	13	200.30
EDWARDS	SALES	7	1285.00

Not the same as -

```
WHERE JOB = 'SALES'
  AND (COMM > 1200 OR YEARS > 10)
```

Result:

NAME	JOB	YEARS	COMM
KOONITZ	SALES	6	1386.70
GRAHAM	SALES	13	200.30
EDWARDS	SALES	7	1285.00

EXERCISE

Write a SELECT statement to retrieve all information about ‘SALES’ employees whose commission is either less than $250 or null.

IN

To match one of a list of values

WHERE DEPT IN (38, 20, 42)

Same as Multiple Ors on the same column:

WHERE DEPT = 38 OR DEPT = 20 OR DEPT =42

BETWEEN

To select a range of values

```
SELECT NAME, JOB, YEARS
FROM Q.STAFF
WHERE YEARS  BETWEEN 9 AND 11
```

same as

```
SELECT NAME, JOB, YEARS
FROM Q.STAFF
WHERE YEARS    >=  9
    AND YEARS  <=  11
```

Results:

NAME	JOB	YEARS
HANES	MGR	10
LU	MGR	10
LEA	MGR	9
WILSON	SALES	9
QUILL	MGR	10

PARTIAL SEARCH

To search on a subset of hcaracters: LIKE

% From zero to any number of characters lgnored

_Exactly one character ignored

Examples:

WHERE NAME LIKE 'G%'	Includes:	GRAHAM GONZALES GAFNEY
WHERE NAME LIKE '%SON'	-	WILSON
WHERE NAME LIKE '%M%N%'	-	MARENGHI ROTHMAN MOLINARE
WHERE NAME LIKE '_ _'	-	LU
WHERE NAME LIKE '_R%'	-	FRAYE GRAHAM

NEGATION

- To reverse the test

WHERE NAME NOT LIKE 'G%'

excludes: GRAHAM
GORZALES
GAFNEY

WHERE YEARS NOT BETWEEN 9 AND 11

excludes YEARS 9 through 11

RESULT TABLE ROWS

To sequence the rows

ORDER BY

To discard duplicate rows

SELECT DISTINCT

ORDER BY

To put the rows into a specified sequence:

Alphabetic list of persons in department 84

```
SELECT NAME, JOB, YEARS
FROM Q.STAFF
WHERE DEPT = 84
ORDER BY NAME
```

Result:

NAME	JOB	YEARS
DAVIS	SALES	5
EDWARDS	SALES	7
GAFNEY	CLERK	5
QUILL	MGR	10

ORDER BY (cont)

Specify column name(s) and whether

ASCending (default) or DESCending sequence

```
SELECT NAME, JOB, YEARS
FROM Q.STAFF
WHERE DEPT = 84
ORDER BY JOB, YEARS DESC
```

Result:

NAME	JOB	YEARS
GAFNEY	CLERK	5
QUILL	MGR	10
EDWARDS	SALES	7
DAVIS	SALES	5

ORDER BY YEARS DESC, JOB

Result:

NAME	JOB	YEARS
QUILL	MGR	10
EDWARDS	SALES	7
GAFNEY	CLERK	5
DAVIS	SALES	5

SELECT DISTINCT

```
SELECT DEPT
FROM Q.STAFF
```

DEPT
20
20
38
38
15
38
10
.
.

To eliminate duplicate rows

```
    SELECT DISTINCT DEPT
    FROM Q.STAFF
```

DEPT
10
15
20
38
.

Multi-columns

```
SELECT DISTINCT DEPT,JOB
    FROM Q.STAFF
```

DEPT	JOB
10	MGR
15	CLERK
15	MGR
15	SALES
20	CLERK
20	MGR
38	SALES
.	CLERK
.	.

'DISTINCT' COMMENTS

The keyword DISTINCT must immediately follow the SELECT

Result Rows are sequenced on columns from left to right

'ORDER BY' can be used to control sequencing

EXAMPLE:

```
SELECT DISTINCT DEPT, JOB
    FROM Q.STAFF
ORDER BY JOB
```

DEPT	JOB
15	CLERK
20	CLERK
38	CLERK
.	
.	
10	MGR
15	MGR
20	MGR
.	
.	
15	SALES
20	SALES

SELECT CALCULATED VALUES

+ ADD * MULTIPLY

- SUBTARACT / DIVIDE

Total earning for each person in DEPT 20

```
SELECT ID, SALARY, COMM, SALARY + COMM
 FROM Q.STAFF
 WHERE DEPT = 20
```

Result:

ID	SALARY	COMM	
10	18357.50	-	-
20	18171.25	612.45	18783.70
80	13504.60	128.20	13632.80
190	14252.75	126.50	14379.25

CONDITION ON CALCULATED VALUES

Persons whose salary plus commission exceeds $20,000

```
SELECT NAME, SALARY + COMM
FROM Q.STAFF
WHERE SALARY + COMM > 20000
```

Result:

NAME	
WILLIAMS	20094.15
GRAHAM	21200.30

'ORDER BY' CALCULATED VALUES

EXAMPLE 1:

```
SELECT NAME, SALARY + COMM
FROM Q.STAFF
WHERE SALARY + COMM > 20000
ORDER BY 2 DESC
```

Result:

NAME	
GRAHAM	21200.30
WILLIAMS	20094.15

EXAMPLE 2:

```
SELECT JOB, SALARY + COMM
FROM Q.STAFF
WHERE DEPT = 42
ORDER BY  JOB, 2
```

Result:

JOB	
CLERK	10581.50
CLERK	11592.80
MGR	--
SALES	19388.45


請做 LAB 1

第三章

ADVANCED SELECT

COLUMN FUNCTIONS

Reduce data from a column into a single value
Result contains no detail of individual underlying rows

Numeric data only

SUM - Total of the values in a column
AVG - Average of the values in a column

Any data type

MIN - Lowest value in a column
MAX - Highest value in a column
COUNT - Number of occurrences

SUM AVG MAX MIN

EXAMPLE：

```
SELECT  SUM(SALARY), AVG(SALARY),
        MIN(COMM), MAX(COMM)
FROM Q, STAFF
WHERE DEPT = 66
```

Result ：

86076.20	17215.2400000000	55.50	844.00

Result contains no detail from individual rows

Nulls are excluded

Decimal precision and scale derived from underlying column

Can use expression

AVG (SALARY + COMM)

COUNT

COUNT(*) Number of rows meeting the search condition

COUNT(DISTINCT column-name) - Number of unique values in that column. Null value not counted.

Count the departments , and persons , where the person's total salary exceeds $18,000 and list their average salary

```
SELECT COUNT(DISTINCT DEPT) , COUNT(*),
       AVG(SALARY)
FROM Q.STAFF
WHERE SALARY> 18000
```

Result :

8	16	19604.1906250000

EXERCISE

Given the following SELECT statement and example table , determine the results :

```
SELECT SUM(COLA) , AVG(COLA) ,
       MIN(COLA) , MAX(COLA)
  FORM TAB_A
  WHERE COLB = 'X'
```

TAB-A

COLA	COLB
7	X
9	X
18	O
6	X
7	X
Null	O
5	X
Null	X
2	X

SUM(COLA)	AVG(COLA)	MIN(COLA)	MAX(COLA)

LITERALS

```
SELECT  AVG(COMM) , SUM(COMM)
FROM  Q.STAFF
```

513.31	12319.45

```
SELECT 'AVG COMM :' , AVG(COMM) ,
 'SUM COMM :' , SUM(COMM)
FROM  Q.STAFF
```

AVG COMM :	513.31	SUM COMM :	12319.45

GROUP BY

With GROUP BY , a column function results in a single value for each group for each function

```
SELECT SUM(SALARY) , SUM(COMM)
    FROM  Q.STAFF
    WHERE JOB < > 'MGR'
    GROUP  BY  DEPT
```

41269.53	1468.70
45928.60	867.15
59778.80	1913.30
.	.
.	.
.	.

GROUP BY (cont)

```
SELECT DEPT, SUM(SALARY) , SUM(COMM)
FROM  Q.STAFF
WHERE JOB <> 'MGR'
GROUP BY DEPT
```

RESULT :

DEPT		
15	41269.53	1468.70
20	45928.60	867.15
38	59778.80	1913.30
:	:	:
:	:	:
:	:	:

GROUP BY … ORDER BY

To change the sort order of the results

```
SELECT  DEPT,  SUM(SALARY) , SUM(COMM)
FROM  Q.STAFF
WHERE JOB <> 'MGR'
GROUP  BY  DEPT
ORDER  BY  3
```

RESULT : DEPT

20	45928.60	867.15
:	:	:
15	41269.53	1468.70
:	:	:
38	59778.80	1913.30

‘GROUP BY’ COMMENTS

‘GROUP BY’ can be on multiple columns

GROUP BY DEPT , JOB

‘GROUP BY’ columns need not be selected

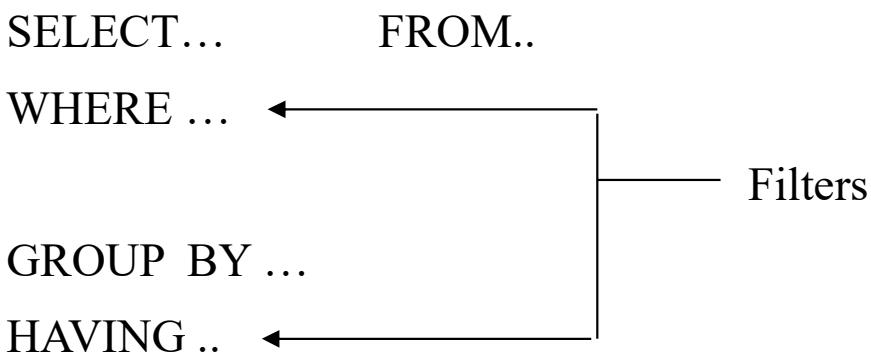
Selected columns must be a column function or in the ‘GROUP BY’

On row of summary results for each ‘GROUP BY’ value

All nulls are a single group

GROUP BY … HAVING

TO PUT CONDITIONS ON GROUPS



WHERE - chooses rows

HAVING - chooses groups

'HAVING' EXAMPLES

```
SELECT  DEPT , SUM(SALARY)
    FROM  Q.STAFF
GROUP  BY  DEPT
```

DEPT	
10	83463.45
15	61929.33
20	64286.10
38	77285.55
42	58369.05
.	.
.	.

```
SELECT  DEPT , SUM(SALARY)
    FROM  Q.STAFF
    GROUP  BY  DEPT
HAVING SUM(SALARY) > 65000
```

DEPT	
10	83463.45
38	77285.55
.	.
.	.

'HAVING' EXAMPLES (cont)

Average non-management salary by department for all departments having more than 3 non-managers

```
SELECT  DEPT , AVG(SALARY)
FROM  Q.STAFF
   WHERE  JOB <> 'MGR'
   GROUP  BY  DEPT
   HAVING  COUNT(*) > 3
```

Average non-management salary by department that has no one with less than 5 years experience . Organize the result with highest average salary first .

```
SELECT  DEPT , AVG(SALARY)
  FROM  Q.STAFF
  WHERE  JOB <> 'MGR'
  GROUP  BY  DEPT
  HAVING  MIN(YEARS) > = 5
  ORDER  BY 2  DESC
```

COLUMN FUNCTIONS SUMMARY

Column functions may be specified only in

SELECT

HAVING

SELECT may specify only

Column functions

Columns specified in ‘GROUP BY’

HAVING may specify

Any column function on any column in a table being queried . This column need not be in the SELECT .

Column FUNCTIONS may not be nested

SCALAR FUNCTIONS

Operate on individual rows

Produce a single value

Scalar functions may be nested

Column functions may be used as arguments of scalar functions

Scalar functions may be used as arguments of column functions

SUBSTR(stringname , start , length)

```
SELECT DEPTNAME , SUBSTR(DEPTNAME , 1 , 4)
FROM  Q.ORG
```

Result :

DEPTNAME
HEAD OFFICE
NEW ENGLAND
MID ATLANTIC
SOUTH ATLANTIC
GREAT
.
.

HEAD
NEW
MID
SOUT
GREA
.
.

```
SELECT DEPTNUMB , DIVISION
FROM  Q.ORG
WHERE SUBSTR(DIVISION ,3) = 'STERN'
```

Result :

DEPTNUMB	DIVISION
15	EASTERN
20	EASTERN
38	EASTERN
66	WESTERN
84	WESTERN

Notes :

- If length not specified , substring is to end of the string
- If start is beyond the end of the string-Error
- If start + length is beyond the end of the string-Error

LENGTH(argument)

```
SELECT NAME , LENGTH(NAME) ,    --Variable length
              JOB , LENGTH(JOB)     --fixed length
FROM  Q.STAFF
```

Result :

NAME		JOB	
SANDERS	7	MGR	5
PERNAL	6	SALES	5
MARENGHI	8	MGR	5
:	:	:	:

Notes :

- If argument is null , length is null.

VALUE(arg 1 , arg 2 , …, argn)

Result is first non-null value in the argument list

```
SELECT  ID , COMM , VALUE(COMM,0)
FROM  Q.STAFF
```

Result :

ID	COMM	
10	-	0.00
20	612.45	612.45
30	-	0.00
40	846.55	846.55
:	:	:

All arguments must be comparable data types (all character or all numeric)

If all arguments are null , result is null

CONVERSION FUNCTIONS

(Not SAA , but in DB2 and SQL/DS)

DECIMAL , FLOAT , INTEGER - convert number data

DIGITS - character representation of numeric value

HEX - character representation of hexadecimal digits

Example : Column SAL defined as DEC(7,2).

```
SELECT  SAL , DECIMAL (SAL,9,1), INTEGER (SAL)
FROM  EMP
WHERE  ID = 55
```

Results :

SAL		
17506.76	17506.7	17506

DATE / TIME DATA

DATE , TIME , TIMESTAMP data stored internally as packed decimal , but with no sign position

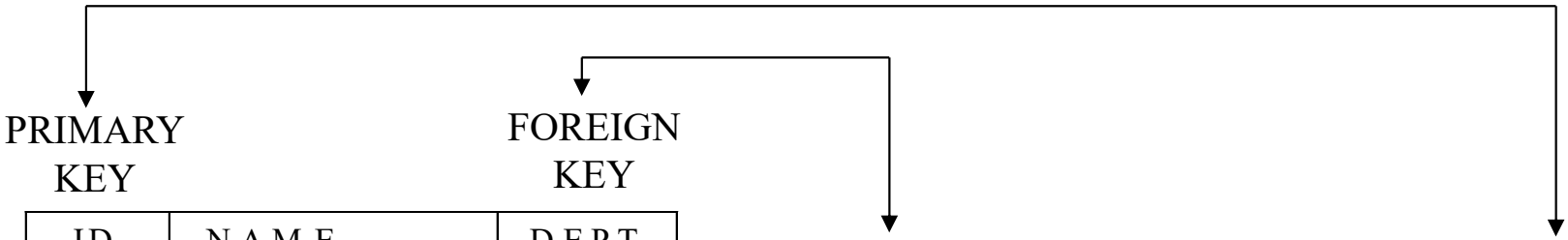
Data Type	Internal Format	Internal Length
DATE	yyyymmdd	4 bytes
TIME	hhmmss	3 bytes
TIMESTAMP	yyyymmddhhmmssnnnnnn	10 bytes

Program deals only with EXTERNAL FORM - a character string

Format	Time Format	Length	Data Format	Length
ISO	hh.mm.ss	8 bytes	yyyy-mm-dd	10 bytes
USA	hh :mm AM hh:mm PM	8 bytes	mm/dd/yyyy	10 bytes
EUR	hh.mm.ss	8 bytes	dd.mm.yyyy	10 bytes
JIS	hh:mm:ss	8 bytes	yyyy-mm-dd	10 bytes
LOCAL	? ? ?	? ? ?	? ? ?	? ? ?

Timestamp data : yyyy-mm-dd-hh.mm.ss.nnnnnn 26 bytes

JOINING TABLES



ID	NAME	DEPT
10	SANDERS	20
20	PERNAL	20
30	MARENGHI	38
40	O'BRIEN	38
50	HANES	15
60	QUIGLEY	38
70	ROTHMAN	15
80	JAMES	20
90	KOONITZ	42
100	PLOTZ	42
110	NGAN	15
⋮	⋮	⋮

DEPTNUMB	DEPTNAME	MANAGER
10	HEAD OFFICE	160
15	NEW ENGLAND	50
20	MID ATLANTIC	10
38	SOUTH ATLANTIC	30
42	GREAT LAKES	100
51	PLAINS	140
66	PACIFIC	270
84	MOUNTAIN	290

JOIN

Join implemented by FROM clause

Result can contain columns from any tables named in the FROM clause

Tables related by common data

"Join predicate" of WHERE clause

Notes :

Guideline : The join predicate should be primary key of one table equals foreign key of other table (e.g. , DEPTNUM = DEPT or MANAGER = ID) .

MULTIPLE TABLES

List the department name and manager's name for each department in the Western Division

```
SELECT  DEPTNAME , NAME
FROM  Q.STAFF , Q.ORG
WHERE  DIVISION = 'WESTERN'
    AND MANAGER = ID
```

Q.ORG

DEPTNAME	MANAGER	DIVISION
.	.	.
.	.	.
PACIFIC	270	WESTERN
PLAINS	140	MIDWEST
.	.	.
MOUNTAIN	290	WESTERN

Q.STAFF

ID	NAME
140	FRAYE
.	.
270	LEA
.	.
290	QUILL

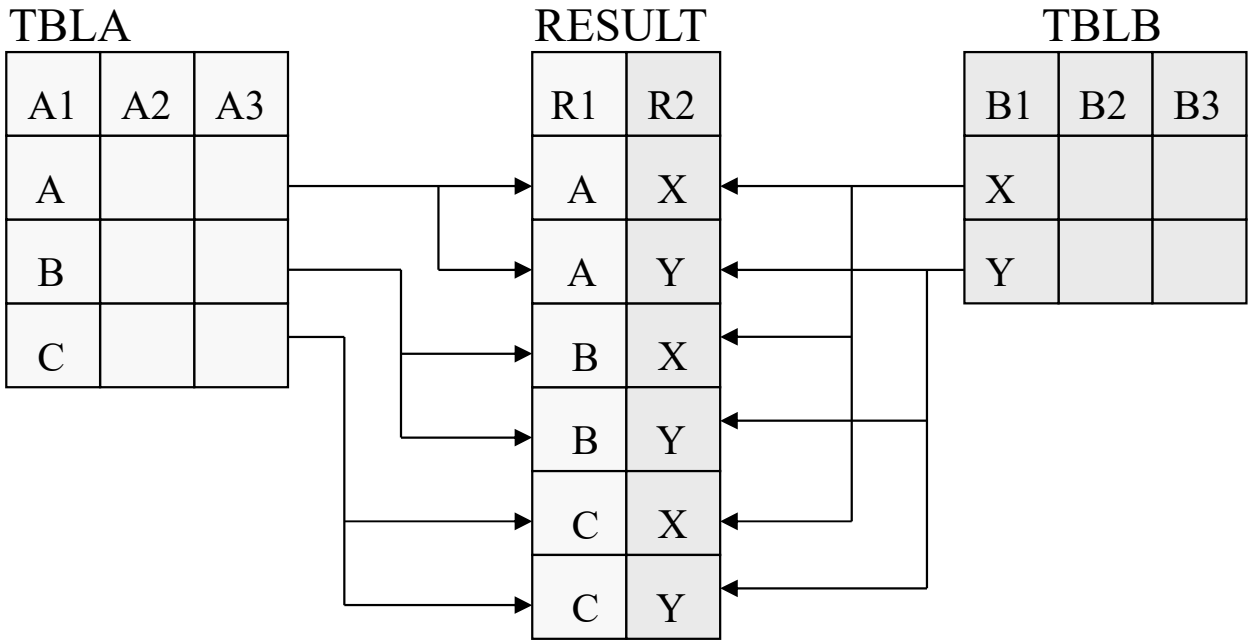
Result :

DEPTNAME	NAME
PACIFIC	LEA
MOUNTAIN	QUILL

WHERE CLAUSE

WARNING : Don’t join tables without
a join predicate .

SELECT A1 , B1 FROM TBLA , TBLB



JOINING MORE THAN TWO TABLES

```
SELECT  ID , NAME , DEPT , DEPTNAME , ORDERNO
    FROM Q,STAFF , Q.ORG , Q.SALES
    WHERE  DEPT = DEPTNUMB  AND  ID = SALESREPNO
```

Result :

ID	NAME	DEPT	DEPTNAME	ORDERNO
20	PERNAL	20	MID ATLANTIC	3456
20	PERNAL	20	MID ATLANTIC	6667
20	PERNAL	20	MID ATLANTIC	3580
40	O'BRIEN	38	SOUTH ATLANTIC	1991
.	.	.	.	.

JOINING MORE THAN TWO TABLES CONT.

Q.STAFF

ID	NAME	DEPT	JOB	YEARS	SALARY	COMM
10	SANERS	20	MGR	7	18357.50	-
20	PERNAL	20	SALES	8	18171.25	612.45
30	MARENGHI	38	MGR	5	17506.75	-
40	O'BRIEN	38	SALES	6	18005.00	846.55
50	HANES	15	MGR	10	20659.80	-
60	QUIGLEY	38	SALES	-	16808.30	650.25
70	ROTHMAN	15	SALES	7	16502.83	1152.00
80	JAMES	20	CLERK	-	13504.60	128.20
:	:	:	:	:	:	:
:	:	:	:	:	:	:

Q.ORG

DEPTNUMB	DEPTNAME	MANAGER	DIVISION	LOCATION
10	HEAD OFFICE	160	CORPORATE	NEW YORK
15	NEW ENGLAND	50	EASTERN	BOSTON
20	MID ATLANTIC	10	EASTERN	WASHINGTON
38	SOUTH ATLANTIC	30	EASTERN	ATLANTA
42	GREAT LAKES	100	MIDWEST	CHICAGO
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.

Q.SALES

ORDERNO	SALESREPNO	PRODNO	QUANTITY	CUSTNO
3456	20	10	50	1200
6667	20	160	120	4400
1991	40	150	600	4500
7777	60	30	150	8500
1020	60	30	150	8500
3333	70	50	240	9600
1115	70	101	120	8300
3580	20	190	360	4900
:	:	:	:	:

QUALIFIERS

Q.STAFF

NUMBER	NAME	DEPT	JOB	__
10	SANDERS	20	MGR	
20	PERNAL	20	SALES	
.	.	.	.	
.	.	.	.	

Q.ORG

NUMBER	DEPTNAME	MANAGER	__
10	HEAD OFFICE	160	
.	.	.	
.	.	.	
20	MID ATLANTIC	10	
.	.	.	
.	.	.	

For managers , list their name , department number and department name .

SOLUTION

Qualifying with Table Name

```
SELECT  NAME , Q.ORG.NUMBER , DEPTNAME
FROM  Q.STAFF , Q.ORG
WHERE  JOB = 'MGR'
  AND  DEPT = Q.ORG.NUMBER
```

Shorthand Qualifier : Correlation name

```
SELECT  NAME , O.NUMBER , DEPTNAME
FROM  Q.STAFF , Q.ORG O
WHERE  JOB = 'MGR'
  AND  DEPT = O.NUMBER
```

JOINING A TABLE WITH ITSELF

LIST EACH EMPLOYEE WHO EARNS MORE THAN HIS OR HER MANAGER

CONCEPTUALLY :

1. Get employee's row from VTEMPL

EMPNO	NAME	…	SALARY	-	MGRNO
40	O'BRIEN : :		18006.00		30

2. Get manager's row from VTEMPL

EMPNO	NAME	…	SALARY	-	MGRNO
30	MARENGHI : :		17506.75		30

3. Compare the two rows' values for salary

4. If query conditions are satisfied , put a row in the result table , then get another employee row and process as above , etc.

JOINING A TABLEWITH ITSELF (cont)

SOLUTION :

```
SELECT  E.NAME , E.SALARY ,
        M.NAME , M.SALARY
FROM  VTEMPL  E,VTEMPL  M
WHERE  E.MGRNO = M.EMPNO
AND  E.SALARY > M.SALARY
```

NOTE : Qualifiers are required because :
Referring to 2 rows in same table
same column names for both rows

Result :

NAME	SALARY
O'BRIEN	18006.00
.	.
.	.
.	.

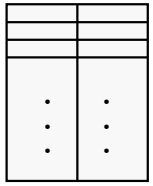
NAME	SALARY
MARENGHI	17506.75
.	.
.	.
.	.

UNION -- OVERVIEW

To merge the results of multiple queries

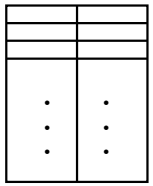
```
SELECT  A1 , A4  FROM  TBLA
  WHERE  A2 = 'Z'
```

RESULT



```
UNION
SELECT  B3 , B4 + B2  FROM  TBLB
  WHERE  B1 > = 'N'
```

RESULT



```
ORDER  BY 2
```

Notes :

- Queries execute serially
- Same number of entries in all SELECT lists
- Corresponding entries of comparable type
- UNION merges rows from resultant tables , elimination duplicate rows
- UNION ALL is like UNION , except that duplicates are NOT eliminated and no sort is performed

UNION

Would sales and clerical salary ranges overlap
If we gave clerks a 17% raise , and sales persons a 2% raise ?

```
SELECT  JOB, MIN(SALARY * 1.17),MAX(SALARY * 1.17)
      FROM  Q.STAFF
      WHERE  JOB = 'CLERK'
      GROUP  BY  JOB
```

```
UNION  ALL
SELECT  JOB, MIN(SALARY *1.02),MAX(SALARY *1.02)
      FROM  Q.STAFF
      WHERE  JOB = 'SALES'
      GROUP  BY  JOB
```

Result :

JOB		
CLERK	12291.9030	16918.2000
SALES	15763.5900	21420.0000

UNION ALL VS. UNION

```
SELECT  TEMPID , NAME
FORM  Q.APPLICANT
WHERE  ADDRESS  LIKE '% NY'
UNION  ALL
```

```
UNION
```

```
SELECT  TEMPID , LASTNAME
FORM  Q.INTERVIEW
WHERE  DISP = 'NOHIRE'
```

RESULTS :

TEMPID	NAME
410	JACOBS
440	REID
480	LEEDS
400	FROMMHERZ
430	RICHOWSKI
480	LEEDS
490	GASPARD

TEMPID	NAME
400	FROMMHERZ
410	JACOBS
430	RICHOWSKI
440	REID
480	LEEDS
490	GASPARD


請做 LAB 2

第四章

SUBQUERY

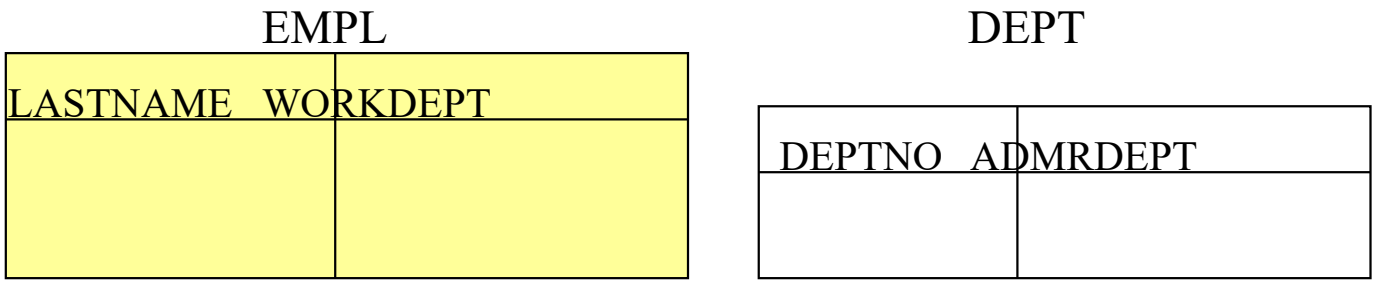
SUBQUERY

A SELECT within a search condition of another SELECT…

Results of subquery are to be used as value(s) within the “outer” SELECT

SUBQUERY EXAMPLE

List employees whose work department is under administrative department ‘A00’




```
SELECT LASTNAME , WORKDEPT
 FROM EMPL
  WHERE WORKDEPT IN (SELECT DEPTNO
                     FROM DEPT
                     WHERE
                     ADMRDEPT = ‘A00’)
```

User sees results only from highest level SELECT

Subquery values returned to higher level SELECT

EXERCISE

Display last name and first name of all employees who work in the same department as Adamson.

Write a SELECT using a subquery .

COMMENTS ON SUBQUERY

Can be in WHERE clause or in HAVING clause

Can be nested

Must be on right side of search condition

Must be enclosed in parentheses

Can return single or multiple values from only one column

Number of values subquery can return determines operator in outer SELECT

Cannot contain UNION, UNION ALL, or ORDER BY

SUBQUERY A SINGLE VALUE

List the employees whose salaries are higher than the company average

Q.STAFF

NAME	SALARY

Find average salary for the company (SUBQUERY)

```
SELECT AVG(SALARY) FROM Q.STAFF
```

16675.64

Compare each employee's salary to the result

```
SELECT NAME, SALARY FROM Q.STAFF
WHERE SALARY > (SELECT AVG(SALARY) FROM Q.STAFF)
```

Result :

NAME	SALARY
SANDERS	18357.50
PERNAL	18171.25
.	.
.	.

SUBQUERY A SINGLE VALUE

LIST THE EMPLOYEE WITH THE MAXIMUM SALARY

```
SELECT NAME , SALARY FROM Q.STAFF
 WHERE SALARY = (SELECT MAX(SALARY)
                  FROM Q.STAFF)
```

RESULT :

NAME	SALARY
MOLINARE	22959.20

SUBQUERY A LIST OF VALUES： ALL

List alphabetically employees whose salaries are higher than the average salaries for each and every department

Find average salary for each department

```
(SELECT AVG(SALARY) FROM Q.STAFF
        GROUP BY DEPT)
```

Result :

```
20865.862500000
15482.332500000
18071.525000000
.
14592.262500000
.
```

Compare each employee's salary to the list of department averages .

```
SELECT NAME, SALARY FROM Q.STAFF
WHERE SALARY > ALL (SELECT AVG(SALARY)
                    FROM Q.STAFF
                    GROUP BY DEPT)
ORDER BY NAME
```

Result :

NAME	SALARY
FRAYE	21150.00
GRAHAM	21000.00
JONES	21234.00
MOLINARE	22959.20

SUBQUERY A LIST OF VALUES: ANY or SOME

List alphabetically employees whose salaries are higher than the average salary of any department

Find average salary for each department

```
SELECT AVG(SALARY) FROM Q.STAFF
  GROUP BY DEPT
```

Result :

```
20865.862500000
15482.332500000
18071.525000000
.
14592.262500000
.
```

Compare each employee's salary to the list of department averages .

```
SELECT NAME, SALARY FROM Q.STAFF
  WHERE SALARY > ANY (SELECT AVG(SALARY)
                      FROM Q.STAFF
                      GROUP BY DEPT)
ORDER BY NAME
```

Result :

NAME	SALARY
DANIELS	19260.25
DAVIS	15454.50
EDWARDS	17844.00
FRAYE	21150.00
GONZALES	18858.20
GRAHAM	21000.00
:	:

IN

List managers in the Western Division

```
SELECT DEPT, NAME FROM Q.STAFF
   WHERE ID = ANY (SELECT MANAGER
                   FROM Q.ORG
                   WHERE DIVISION = 'WESTERN')
```

Result of the subquery is a list of MANAGERs from the Western Division departments (270 and 290)

'IN' is equivalent to a series of 'OR…=' on a single column

IN can be substituted for = ANY

```
SELECT DEPT, NAME FROM Q.STAFF
    WHERE ID IN
    (SELECT MANAGER
       FROM Q.ORG
       WHERE DIVISION = 'WESTERN')
```

SUBQUERY within HAVING CLAUSE

List those departments where average of non-managers salary is above the company average of non-managers.
Include the average salary with the highest first.

```
SELECT DEPT, AVG(SALARY) FROM Q.STAFF
    WHERE JOB < > 'MGR'
    GROUP BY DEPT
        HAVING AVG(SALARY) > (SELECT AVG(SALARY)
                              FROM Q.STAFF
                               WHERE JOB < > 'MGR')
    ORDER BY 2 DESC
```

Result :

DEPT	
66	16880.175000000
51	16235.200000000
84	15443.000000000
20	15309.533333333

CORRELATED SUBQUERY

List employees whose salaries are higher than the average salary for their department

```
SELECT NAME, SALARY
FROM Q.STAFF CN
WHERE SALARY > (SELECT AVG(SALARY)
                FROM Q.STAFF
                WHERE DEPT = CN.DEPT)
```

A correlated subquery may be executed for each row retrieved by the outer select .

SUBQUERY TO TEST TRUE/FALSE

List information about managers making less than $18,000
if there are any managers making more than $22,000

```
SELECT ID, NAME, SALARY, DEPT FROM Q.STAFF
  WHERE JOB = 'MGR' AND SALARY < 18000
  AND EXISTS (SELECT * FROM Q.STAFF
  WHERE JOB = 'MGR' AND SALARY > 22000)
```

ID	NAME	SALARY	DEPT
30	MARENGHI	17506.75	38

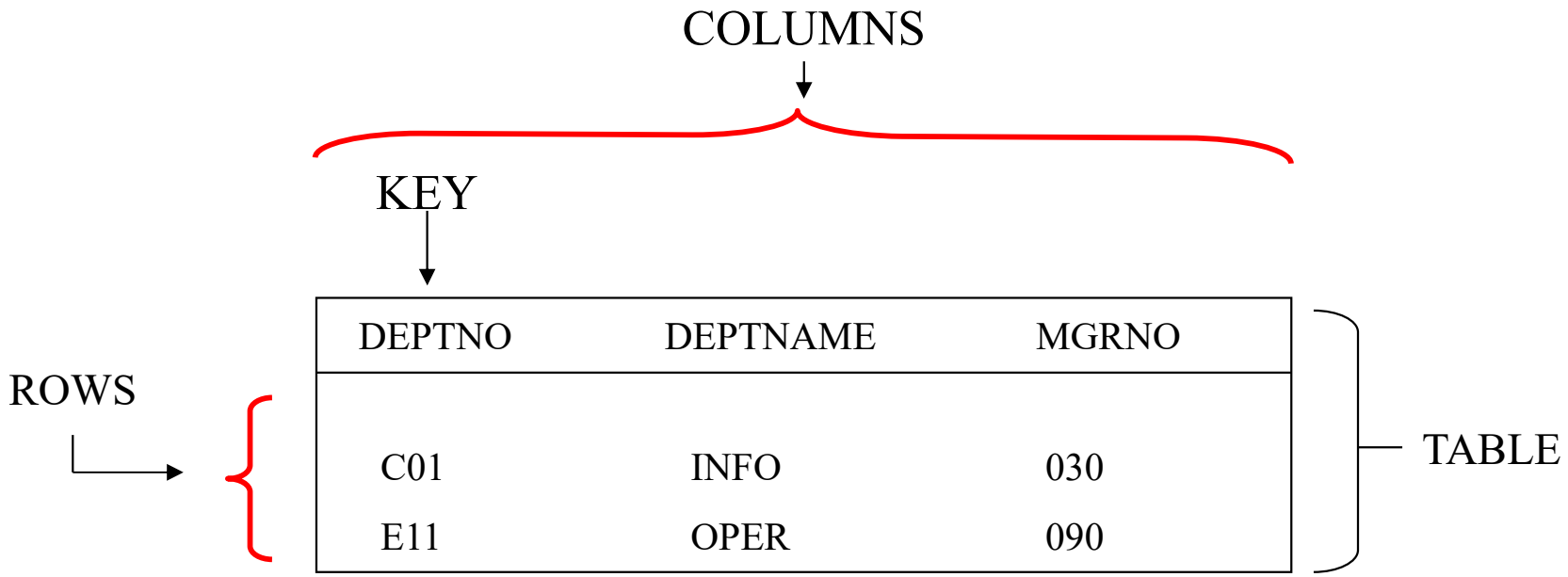
TRUE/FALSE ON INNER 'SELECT'

Decide if first 'select' takes place

第五章

TABLE

TABLE TERMS



ENTITY：DEPARTMENT

NUMERIC DATA

COLUMN DEFINITION

DATA TYPE	DESCRIPTION	RANGE
SMALLINT	Whole numbers	+or-32k*
INTEGER INT	Whole numbers	+OR-2 Billion*
DECIMAL(X ，Y) DEC(X ，Y)	X digits， of which y are to the right of the decimal point	up to 15 digits
FLOAT See Notes	floating point	See Notes

*Approximately

For DECIMAL ，define X as an odd number

NOTES: The precision and magnitude of floating point columns is dependent on the platform.Other non-SAA data type definitions may also de used for floating point columns:

FLOAT(n), REAL, DOUBLE PRECISION. See the SQL Reference MANUAL for the individual products.

CHARACTER STRING DATA

COLUMN DEFINITION

DATA TYPE	DESCRIPTION	LENGTH
CHAR(X) CHARACTER(X)	fixed length string	x characters (maximum 254)
VARCHAR(X)	variable length string (or longer than 254 char)	O to x characters (maximum 4K*)

*Approximately , assuming 4k page.True Maximum length depends on lengths of other columns, as well as “page” size.

Note:

Not SAA,but in DB2 , SQL/DS and OS/2 EE:

LONG VARCHAR - variable length string - 0 to n characters

(maximum 4K* - calculated)

DOUBLE BYTE CHARACTER SETS

COLUMN DEFINITION

DATA TYPE	DESCRIPTION	LENGTH
GRAPHIC(X)	fixed length string	x characters (maximum 127)
VARGRAPHIC(X)	variable length string (or longer than 127 char)	O to x characters (maximum 2K*)

*Approximately , assuming 4k page.True Maximum length depends on lengths of other columns, as well as “page” size.

Note:

Not SAA,but in DB2, SQL/DS:

LONG VARGRAPHIC - variable length string - 0 to n characters
(maximum 2K* - calculated)

DATE/TIME DATA

COLUMN DEFINITION

DATA TYPE	INTERNAL FORMAT	INTERNAL LANGTH
DATE	yyymmdd	4 bytes
TIME	hhmmss	3 bytes
TIMESTAMP	yyyymmddhhmmssnnnnnn(nnnnnn =microseconds)	10 bytes

CREATE TABLE

CREATE TABLE table_name (col_name…)

* 1 to 10 characters:
  A-Z $ # @ 0-9 _

(DB2,SQL/DS and OS/2 EE support up to 18 characters)

* First character must be alphabetic

Note:

Not SAA,but in DB2 and SQL/DS:

CREATE TABLE tablename
(column definitions)
IN……

For DB2: IN database. tablespace

For SQL/DS: IN dbspace Specifies where the table is to be stored.

CREATE TABLE EXAMPLE

```
CREATE TABLE CITIES

(CITY_ID          CHAR(10)          NOT NULL,
CITY _NAME        CHAR(20)           NOT NULL,
MILES_2-LA        SMALLINT           NOT NULL,
POOULATION        INT                ,
AVG_INCOME        DEC(9,2)           ,
LAST_CENS         DATE               ,
LAST_UPDAT        TIMESTAMP          NOT NULL WITH DEFAULT,
DESCRIPT          VARCHAR(300),
PRIMARY KEY       (CITY_ID)          )
```

Note:

Since a primary key has been identified,a unique index must be created on that column.

REFERENTIAL INTEGRITY

Primary key-foreign key

- Data types and lengths must be identical

Primary key

- Values must be unigue
- Unique index required
- Nulls are not allowed

Foreign key

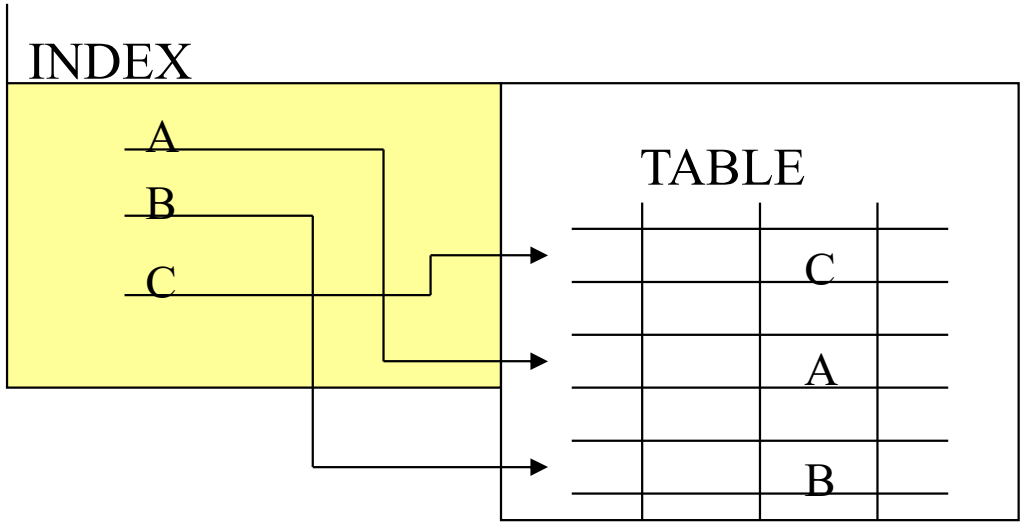
- Value must match primary key value
- Nulls are allowed

第六章

OBJECTS

INDEX

An ordered list used by the database manager to locate row(s) of a table containing a given value.



CREATE INDEX

```
CREATE INDEX index_name
     ON table_name
      (col_name ASC  |  DESC,
          col_name ASC | DESC,
            …                    )
```

Optional Parameter:

UNIOUE- prevents the index from containing two or more rows with the same key

CREATE INDEX EXAMPLES

```
CREATE  INDEX  IXEMPD
    ON  EMP(WORKDEPT)

CREATE  UNIQUE  INDEX  IXEMPN
    ON  EMP(EMPNO)

CREATE  INDEX  IXPROJ
    ON  PROJ(PROJNO DESC)

CREATE  UNIQUE  INDEX  IXDADM
    ON  DEPT(DEPTNO,ADMRDEPT)
```

VIEW

A logical table with rows & columns from one or more tables

defined through SELECT

view data not separately stored

simplifies SQL

provides security

‘CREATE VIEW’FORMAT

CREATE VIEW view-name

(col-name,col-name…)

AS SELECT select-stmt

Note:

Not SAA,but in DB2 and OS/2 EE,optional parameter:

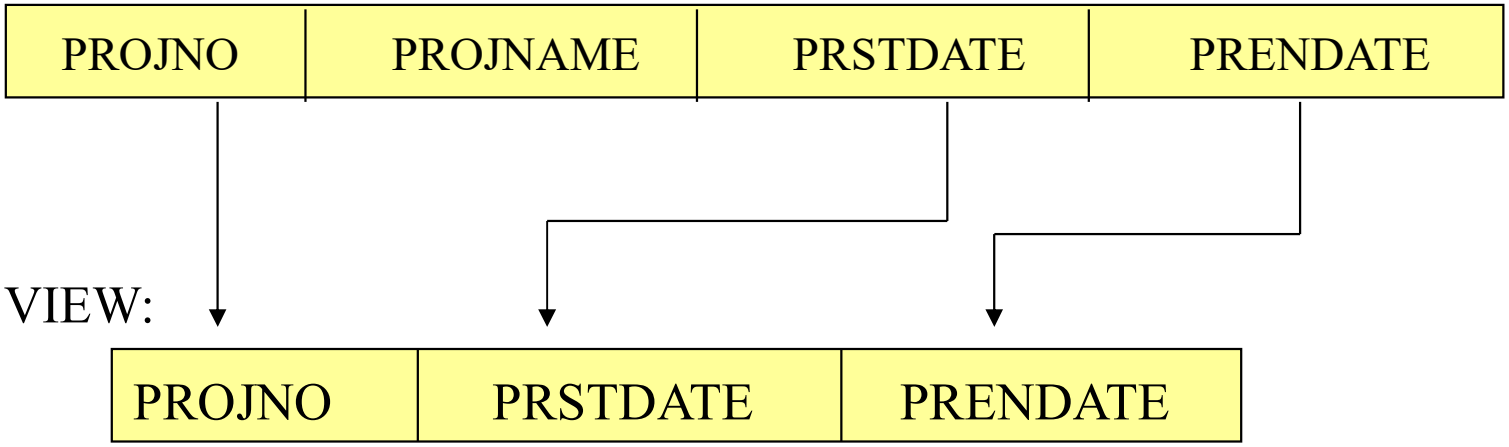
WITH CHECK OPTION

Data must stay within the VIEW definition when the view is used for UPDATEs.For Example,assume a view defined by;

ASSUMED COLUMN NAMES

```
CREATE VIEW  MYPROJ AS
   SELECT PROJNO ,PRSTDATE,PRENDATE
   FROM PROJ
```

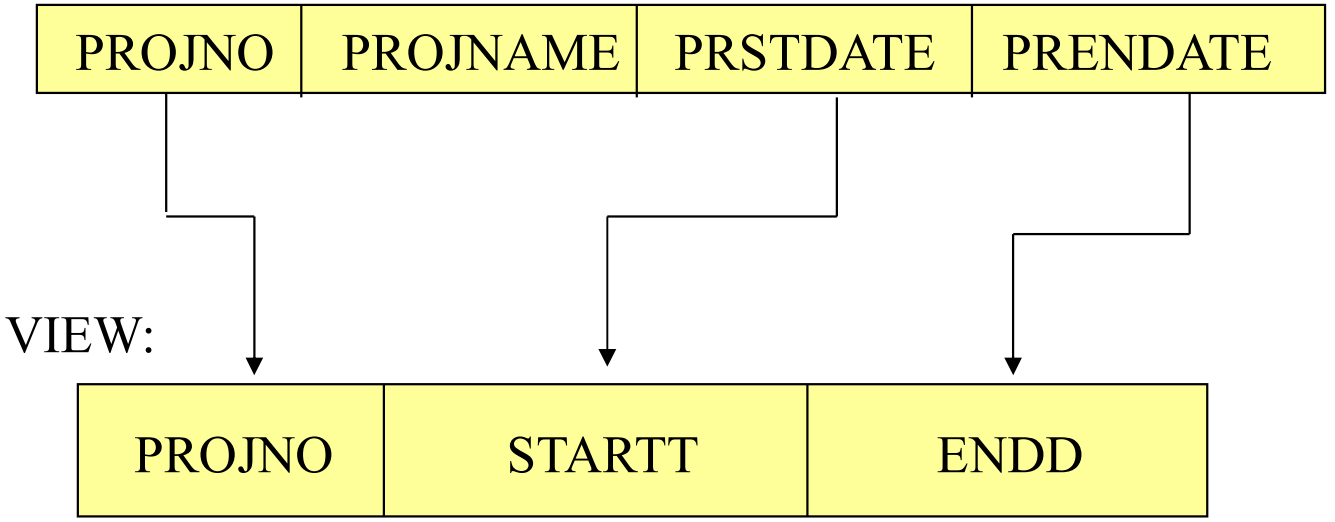
BASE TABLE:



ASSIGNING COLUMN NAMES

```
CREATE VIEW  MYPROJ
 (PROJNO,STATT,ENDD)
 AS SELECT  PROJNO ,PRSTDATE,PRENDATE
   FROM PROJ
```

BASE TABLE:



VIEW AS CONVENIENCE

Using Base Table

```
SELECT PROJNO,PRSTDATE,PRENDATE FROM PROJ
 WHERE (DEPTNO='A11' OR DEPTNO='A17')
    AND PRENDATE < '01/01/1992'
```

Creating a VIEW

```
CREATE VIEW MYPROJ AS
SELECT PROJNO,PRSTDATE,PRENDATE FROM PROJ
 WHERE(DEPTION='A11'OR DEPTNO = 'A17')
    AND PRENDATE < '01/01/1992'
```

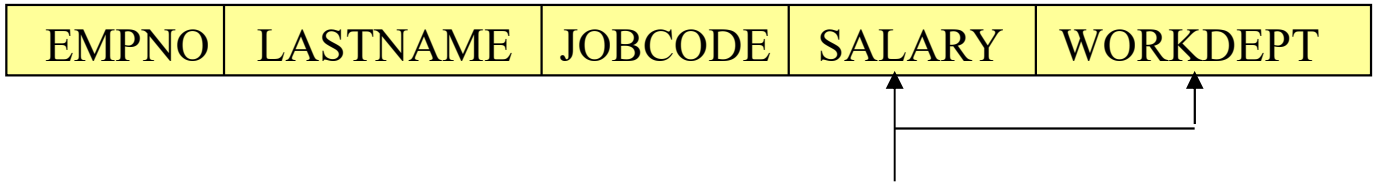
Using a VIEW

```
SELECT * FROM  MYPROJ
```

VIEW AS SECURITY

```
CREATE VIEW VEMPA17 AS
  SELECT EMPNO,JOBCODE,LASTNAME
   FROM EMPL WHERE WORKDEPT ='A17'
```

Base Table EMPL

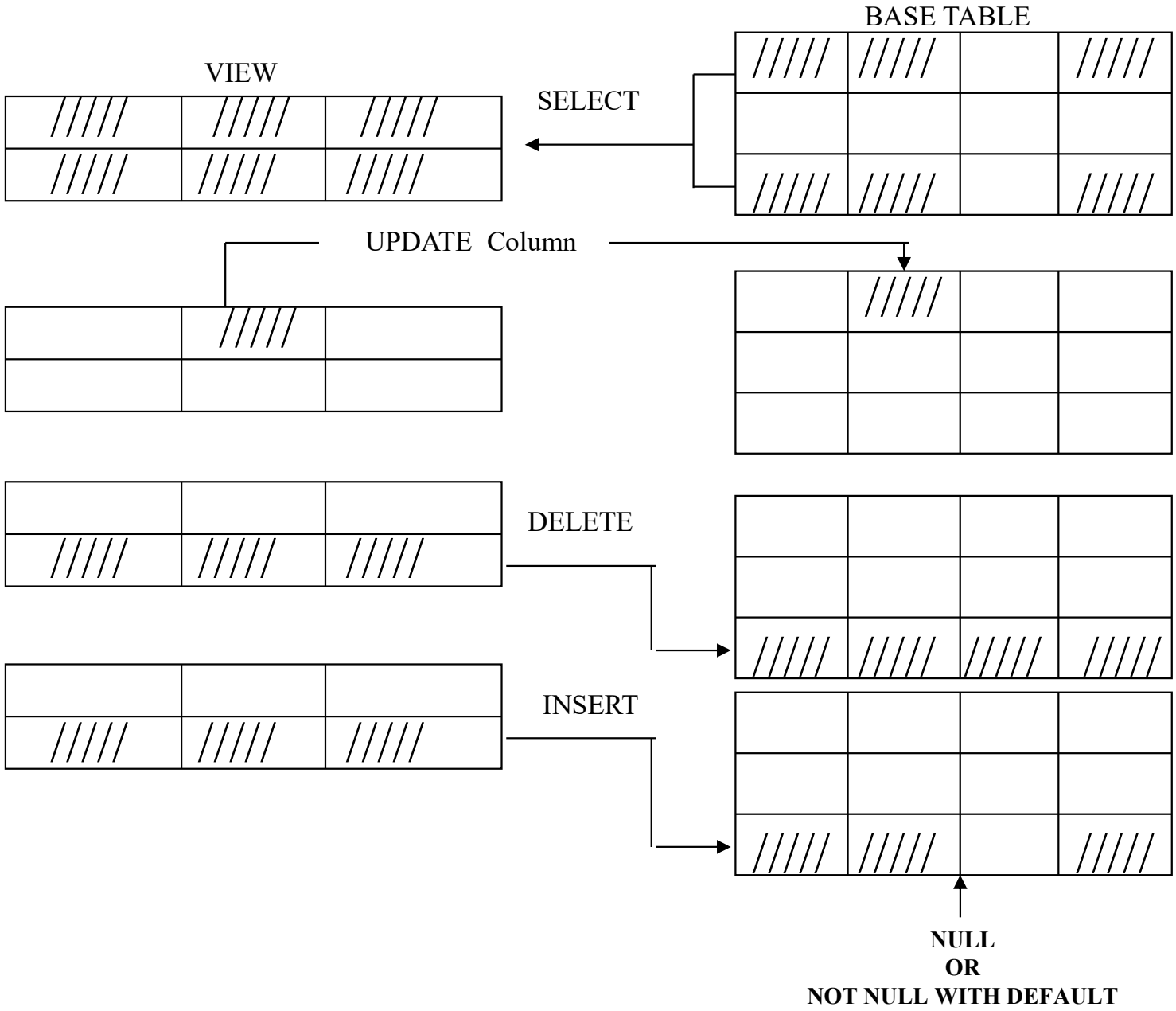


VIEW VEMPA17

EMPNO	JOBCODE	LASTNAME

Rows are for dept A17 only

PROCESSING A VIEW



PROCESSING A VIEW

When a row is inserted into a VIEW, the base TABLE columns which are not in the view must allow NULLs or be defined as NOT NULL WITH DEFAULT

If VIEW column is derived from an expression or scalar function

No UPDATE on derived column

No INSERT through the VIEW

If VIEW includes a join,GROUP BY clause,the DISTINCT keyword,or any column function

The VIEW is SELECT only

A VIEW cannot contain UNION,UNION ALL,or ORDER BY

ALTER TABLE

ALTER TABLE table_name

ADD column_name data_type

To ADD multiple columns,code multiple ALTER statements

ADDed columns must accept NULL or NOT NULL WITH DEFAULT values

ALTER TABLE EXAMPLE

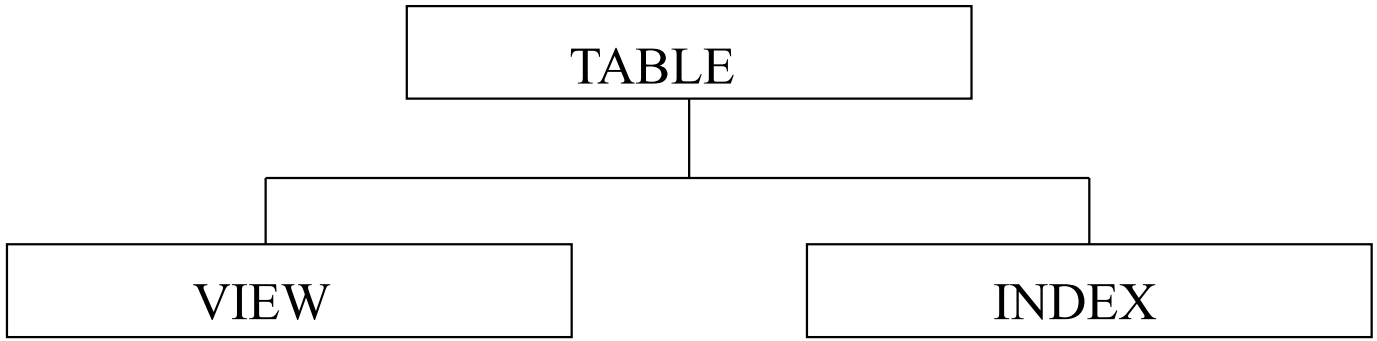
A	B	C
01	Aaa	8
02	SSS	7
03	Mnn	12

```
ALTER TABLE TABX
ADD D CHAR(3)
```

A	B	C	D
01	Aaa	8	
02	SSS	7	
03	Mnn	12	

DROP

Dropping an object causes its dependents to be dropped



第七章

CONTROL

GRANT

Used to grant privileges on an object to users,

Table or view privileges:

ALTER	(Table only)
INDEX	(Table only)
DELETE	
INSERT	
SELECT	
UPDATE	

Example:

```
GRANT SELECT ON EMPL TO PUBLIC
```

REVOKE

Used to revoke previously granted privileges on an object from users.

Example:

```
REVOKE SELECT ON EMPL FROM PUBLIC
```

第八章

MAINTAINING DATA

INSERT A Single Row

```
INSERT INTO PROJ
  VALUES('MA2114','','B01','',NULL, CURRENT DATE, NULL)
 -----or-----
INSERT INTO PROJ
  (DEPTNO, PROJNO, PROJNAME, RESPEMP, PRST DATE)
  VALUES('B01','MA2114','','',CURRENT DATE)
```

RESULT:

PROJNO	PROJNAME	DEPTNO	RESPEMP	PRSTAFF	PRSTDATE	PRENDATE
……	……...	…	……	……	……	……
……	……...	…	……	……	……	……
OP2011	SCP SYSTEM SUPPORT	E21	000320	001.00	19860101	19920201
OP2012	APPLICAT'N SUPPORT	E21	000330	001.00	19860101	19920201
OP2013	DB/DC SUPPORT	E21	000340	001.00	19860101	19920201
……	……	…	……	……	……	……
MA2114		B01		?	19920320	?

INSERT Multiple Rows

TESTPROJ

PROJNO	PROJNAME	DEPTNO	RESPEMP	PRSTAFF	PRSTDATE	PRENDATE

```
INSERT INTO TESTPROJ
  SELECT * FROM PROJ
    WHERE DEPTNO LIKE 'B%'
      OR DEPTNO IN ('E11','E21','E31')
```

RESULT:

PROJNO	PROJNAME	DEPTNO	RESPEMP	PRSTAFF	PRSTDATE	PRENDATE
PL2100	WELD LINE PLANNING	B01	000020	001.00	19910101	19910915
OP1010	OPERATION	E11	111190	115.00	19900101	19920501
OP2010	SYSTEM SUPPORT	E21	000100	004.00	19860101	19920201
OP2011	SCP SYSTEM SUPPORT	E21	000320	001.00	19860101	19920201
OP2012	APPLICAT'N SUPPORT	E21	000330	001.00	19860101	19920201
OP2013	DB/DC SUPPORT	E21	000340	001.00	19860101	19920201
MA2114		B01		?	19920320	?

UPDATE Columns

BEFORE:

PROJNO	PROJNAME	DEPTNO	RESPEMP	PRSTAFF	PRSTDATE	PRENDATE
PL2100	WELD LINE PLANNING	B01	000020	001.00	19910101	19910915
……	……..	…	……	……	……..	……..
OP2012	APPLICAT'N SUPPORT	E21	000330	001.00	19860101	19920201
OP2013	DB/DC SUPPORT	E21	000340	001.00	19860101	19920201
MA2114		B01		?	19920320	?

```
UPDATE TESTPROJ
  SET PRSTAFF = PRSTAFF *2,PRENDATE = NULL,
   PRSTDATE ='1991-06-01'
  WHERE DEPTNO ='B01'
```

AFTER：

PROJNO	PROJNAME	DEPTNO	RESPEMP	PRSTAFF	PRSTDATE	PRENDATE
PL2100	WELD LINE PLANNING	B01	000020	002.00	19910601	?
……	……..	…	……	……	……..	……..
OP2012	APPLICAT'N SUPPORT	E21	000330	001.00	19860101	19920201
OP2013	DB/DC SUPPORT	E21	000340	001.00	19860101	19920201
MA2114		B01		?	19910601	?

UPDATE Columns(cont)

A subset of rows

```
UPDATE PERS
  SET SALARY = SALARY + 300
WHERE JOB = 'CLERK'
```

All rows:

```
UPDATE PERS
  SET YEARS = YEARS + 1
```

EXERCISE

Write an update statement to give all clerks in the Q.STAFF table a 5% increase if YEARS is 10 or more.

DELETE ALL Rows

BEFORE：

PROJNO	PROJNAME	DEPTNO	RESPEMP	PRSTAFF	PRSTDATE	PRENDATE
PL2100	WELD LINE PLANNING	B01	000020	002.00	19910601	?
……	……..	…	……	……	……..	……..
OP2012	APPLICAT'N SUPPORT	E21	000330	001.00	19860101	19920201
OP2013	DB/DC SUPPORT	E21	000340	001.00	19860101	19920201
MA2114		B01		?	19910601	?

```
DELETE FROM TESTPROJ
  WHERE DEPTNO ='B01'
```

AFTER:

PROJNO	PROJNAME	DEPTNO	RESPEMP	PRSTAFF	PRSTDATE	PRENDATE
……	……..	…	……	……	……..	……..
……	……..	…	……	……	……..	……..
……	……..	…	……	……	……..	……..
OP2012	APPLICAT'N SUPPORT	E21	000330	001.00	19860101	19920201
OP2013	DB/DC SUPPORT	E21	000340	001.00	19860101	19920201

DELETE ALL Rows

BEFORE：

PROJNO	PROJNAME	DEPTNO	RESPEMP	PRSTAFF	PRSTDATE	PRENDATE
……	……..	…	……	……	……..	……..
……	……..	…	……	……	……..	……..
OP2012	APPLICAT'N SUPPORT	E21	000330	001.00	19860101	19920201
OP2013	DB/DC SUPPORT	E21	000340	001.00	19860101	19920201

```
DELETE FROM TESTPROJ
```

AFTER：

PROJNO	PROJNAME	DEPTNO	RESPEMP	PRSTAFF	PRSTDATE	PRENDATE

第九章

SQL IN PROGRAMS

EMBEDDED SQL - EXAMPLE 1

Non-Cursor Processing

```
EXEC SQL UPDATE Q.STAFF
        SET SALARY= :NEWSAL
        WHERE ID = :EMPNO
END-EXEC.

EXEC SQL SELECT AVG(SALARY)
        INTO  :AVESAL
        FROM Q.STAFF
        WHERE DEPT =:DPTNO
END-EXEC.
```

SELECT is valid only for a single result row

Places result(s) in “Host Variable(s)”

```
DECLARE   <cursor_name>   FOR
SELECT   <column_name>
FROM     <table_name>
WHERE   <condition>
ORDER BY   <column_name>
FOR UPDATE OF <column_name>

OPEN <cursor_name>

CLOSE <cursor_name>


FETCH     <NEXT , PRIOR , FIRST , LAST>
FROM     <cursor_name>
FOR   <pgm_variable> ROWS
INTO   <pgm_variable>


UPDATE <table_name>
SET   <coulmn_name>   =   <pgm_variable>
WHERE   CURRENT   OF   <cursor_name>


DELETE   FROM   <table_name>
WHERE CURRENT OF <cursor_name>
```