Code sample

Volume of Fundamentals

Chapter 1 Numerical Calculation

1 Null values judgment

	A	
1	=null	
2	=0	
3	=if(A1==null,"null","not null")	null
4	=if(A2!=null,"not null","null")	not null
5	=[,1,2,3].ifn()	1

2 Random values

	A	
1	=rand()	Get a random value not less than **0** and less than **1**
2	=int(rand()*1000)	Get random integer values between **0** and **999**

3 Sign of values

	A	
1	=sign(45)	Return 1 for positive values
2	=sign(-100.34)	Return -1 for negative values
3	=sign(0)	Return 0 for zero
4	=abs(-4.6)	Return 4.6 for absolute value

4 Involution and evolution

	A	
1	=power(2,3)	Cube
2	=power(-2,3)	Cube
3	=power(4,0.5)	Square root
4	=power(27,1/3)	Cube root

5 Decimal truncation and rounding

	A	
1	=round(3451251.274,1)	Round down to 1 decimal place
2	=round(3451251.274,2)	Round down to 2 decimal places
3	=ceil(3450001.003,-2)	Complete carry and accurate to 100
4	=ceil(3450001.003,2)	Complete carry and reserve 2 decimal places
5	=floor(3451291.234,-2)	Round all and accurate to 100
6	=floor(3451281.238,2)	Round all and reserve 2 decimal places

6 Pi

	A	
1	=pi()	pi
2	=pi(4)	4*pi

7 Trigonometric function

	A	
1	=sin(pi(30/180))	Sine
2	=cos(pi()/2)	Cosine
3	=tan(pi()/4)	Tangent

	A	
4	=asin(0.5)	Arc sine
5	=acos(-0.5)	Arc cosine
6	=atan(1)	Arc tangent

8 Logarithm function

	A	
1	=lg(10000)	Logarithm with 10 as its base
2	=ln(1000)	Natural logarithm
3	=exp(A2)	Powers of e

9 Compute different expressions on conditions

	A	
1	3000	
2	=if(A1>10000, A1*0.45+450, A1>5000, A1*0.15+150, A1*0.05)	150.0
3	manager	
4	=case(A3,"president":500,"manager":300,"employee":150)	300

10 Use temp variables in the expressions

	A	
1	=(a=1,b=a*3,b+4)	7
2	=a	1
3	=b	3

Chapter 2 Strings

11 Generate strings of fixed length

	A	
1	=fill(" ",10)	" "
2	=len(A1)	10
3	=fill("ab",10)	abababababababababab

12 Search and replace substrings

	A	
1	=pos("abcdef","def")	Determine the position of **"def"** in **"abcdef"**
2	=pos("abcdefdef","def",5)	Search from the fifth character
3	=replace("abca","a","ABC")	Replace **"a"** with **"ABC"** in **"abca"**
4	=replace("abc'abc'","a","ABC")	Substrings in the quotation marks will also be replaced
5	=replace@q("abc'abc'","a","ABC")	Substrings in the quotation marks will not be replaced

13 Acquire part of a string

	A	
1	=mid("abcde",2,1)	Acquire the second character
2	=mid("abcde",3,2)	Acquire 2 characters from the third position
3	=mid("abcde",2)	Acquire characters from the second position to the end
4	=left("abcdefg",3)	The left three characters
5	=right("abcde",2)	The right two characters

14 Joining of the strings

	A	
1	="ab"+"cd"	
2	="3"+2	The result is **5** because the sting is taken as number when

		computed with numbers.
3	="ab"+1	The result is **1** because the character strings will not be converted into numbers and will be taken as 0.

15 Case identification and conversion

	A	
1	=upper("abcdef")	"ABCDEF"
2	=upper("ABCdef")	"ABCDEF"
3	=lower("abcDEF")	"abcdef"
4	=isupper("ABC")	true
5	=islower("ABC")	false
6	=islower("aBc")	false
7	=isupper("Bc")	false

16 Remove the blank spaces on both sides of a string

	A	
1	=trim(" abc ")	Remove the spaces on both sides
2	=trim@l(" abc ")	Remove the spaces on the left
3	=trim@r(" abc ")	Remove the spaces on the right

17 Match the pattern strings

	A	
1	=like("abc123","abc1?3")	**"?"** is used to match a single character
2	=like("abcefg","abc*")	**"*"** is used to match 0 or more characters

18 Match the pattern strings with '*'

	A	
1	=like("ab*123", "ab*1?3")	The escape character **"\"** can be used to match its tailing character like **"*"** in this case.
2	=like("a*bcefg","a*bc*")	true

19 Acquire character codes and return characters by encoding

	A	
1	=asc("a")	Encode the character **"a"**
2	=char(68)	Get the character of code **"68"**
3	=asc("USA")	Get the code of character **"U"**

20 Split a string into sequences

	A	
1	="12345678"	
2	=len(A1)	
3	=A2.(mid(A1,#,1))	Split it into character sequences
4	="a,[b,c],d"	
5	=A4.array ()	Return a sequence which consists of **a** ,**[b,c]**, **d** three members where the member **[b,c]** is a sequence
6	=A4.array@s()	Return a sequence which consists of **a** ,**[b,c]**, **d** three members where the member [**b,c**] is a string instead of a sequence
7	=A4.array@b()	Return a sequence which consists of **a** ,**[b,c]**, **d** four members**.** The quotation marks and brackets will not be matched during processing.

	A	
8	="a;[b;c];d".array (";")	Use ";" as the delimiter instead of the default","

21 Join the sequences into a string

	A	
1	=[1,"abc,def",[2,4],"{7,8}"]	
2	=A1.string()	Use "," as the delimiter to join and quote the strings in the quotation marks
3	=A1.string@d()	String members don't need to use quotation marks when joining a string.
4	=A1.string("&")	Use "&" as the delimiter
5	=[1,"abc,def","123"].conj@s()	Join strings directly without delimiter

22 the not necessary characters from a string

	A	
1	abcda123efag	
2	=replace(A1,"123","")	Remove "123"
3	=replace(A1,"a","")	remove "a"
4	=len(A1).(mid(A1,#,1)).select(pos("ace",~)==0).conj@s()	Remove all of the "a,c,e"

23 Take out the letter or numerical part from a string

	A	
1	2345$#dfAgsdf23*	
2	=len(A1)	16
3	=A2.(mid(A1,#,1))	Split it into character sequences
4	=A3.select(isdigit(~)).conj@s()	Numerical part
5	=A3.select(isalpha(~)).conj@s()	Letter part

24 Check if a string is consisted of letters or numbers

	A	
1	2345$#dfAgsdf23*	
2	=len(A1)	16
3	=A2.(mid(A1,#,1))	Split it into character sequences
4	=A3.count(!isdigit(~) && !isalpha(~))==0	Check if it is a alphanumeric string

25 Compute the expressions stored in a string

	A	
1	=eval("1+5")	Compute 1 + 5
2	=eval($[A1+2])	Compute A1 + 2

26 Use arguments when computing the expressions in a string

	A	
1	=eval("?+5",3)	Equivalent to "3 + 5"
2	=eval("(?1+1)/?2",3,4)	Equivalent to "(3 + 1)/4"

27 Expression strings can change with the edition process

	A	
1	=$[B1+4]	The strings will change with the edition process in case $[] method is used in coding

Chapter 3 Datetime

28 Acquire the current date and time

	A
1	=now()

29 Split every part of the datetime

	A
1	=now()
2	=year(A1)
3	=month(A1)
4	=day(A1)
5	=time(A1)
6	=hour(A1)
7	=minute(A1)
8	=second(A1)

30 Join every part to make up the date time

	A	B	C	D	E	F
1	1989	'02	'01	'02	34	55
2	=string(A1)+"-"+string(B1)+"-"+string(C1)+" "+string(D1)+":"+string(E1)+":"+string(F1)					
3	=datetime(A2)					
4	=datetime(A2,"yyyy-MM-dd HH:mm:ss")					
5	=date(A1,int(B1),int(C1))					
6	=time(int(D1),E1,F1)					
7	=datetime(A1,int(B1),int(C1),int(D1),E1,F1)					
8	=datetime(A5,A6)					

31 Datetime away from a point

	A	
1	2006-07-05	
2	=after(A1,5)	5 days later
3	=after("1972-11-08 10:20:30",-10)	10 days before
4	=after@s(A1,5)	5 seconds later
5	=after@s("1972-11-08 10:20:30",-10)	10 seconds before
6	=after@m(A1,-1)	1 month before
7	=after@y(A1,-1)	1 year before

32 Interval between two datetime

	A	B	
1	2010-5-01 23:20:15	2010-05-03 01:01:01	
2	=interval(A1,B1)		The number of days difference between two datetimes
3	=interval@s(A1,B1)		The number of seconds difference between two datetimes
4	=interval@y(A1,"2001-01-01")		The number of years difference between two dates
5	=interval@m(A1,"2001-01-01")		The number of months difference between two dates
6	=interval@ms(A1,now())		The number of milliseconds difference between two datetimes

33 Day of a week

	A
1	2005-01-08
2	=day@w(A1)

The result is the day of the week and "**7**" stands for "**Saturday**"

34 The first and the last days of a week, a month and a quarter

	A
1	2006-03-06
2	=pdate@w(A1)
3	=pdate@we(A1)
4	=pdate@q(A1)
5	=pdate@qe(A1)
6	=pdate@m(A1)
7	=pdate@me(A1)

2: The first day of the week
3: The last day of the week
4: The first day of the quarter
5: The last day of the quarter
6: The first day of the month
7: The last day of the month

35 The number of days of a month, a quarter and a year

	A
1	2007-08-08
2	=days(A1)
3	=days@y(2006)
4	=days@y(A1)
5	=days@q(A1)

2: The days of the month in **A1**
3: The days of the year **2006**
4: The days of the current year
5: The days of the quarter in **A1**

36 Set periodic interval to generate datetime sequences

	A
1	2000-08-10 12:00:00
2	=periods@y(A1,now(),1)
3	=periods@q(A1,now(),1)
4	=periods@m(A1,now(),2)
5	=periods@d(A1,now(),7)

2: Set 1 year as the interval unit
3: Set 1 quarter as the interval unit
4: Set 2 months as the interval unit
5: Set 7 days as the interval unit

37 Get the second and last Fridays in a month/quarter/year and the total number of Fridays in this period

	A
1	=now()
2	=pdate@m(A1)
3	=pdate@me(A1)
4	=periods@d(A2, A3,1)
5	=A4.select(day@w(~)==6)
6	
7	=after(A2,6-day@w(A2))
8	=periods@dx(A7,A3,7)
9	=A8(2)
10	=A8.m(-1)
11	=A8.len()

2: The begin date of the month in **A1**
3: The end date of the month in **A1**
4: The day sequence of the month in-between
5: The Friday sequence of **A4**
6: Another alternative
7: Get the first Friday
8: Get the Friday sequences
9: Get the second Friday
10: Get the last Friday
11: Get the number of Fridays

Chapter 4 Sequence

38 Check if it is an sequence

	A	
1	=ifa([1,2,3])	true
2	=ifa(123)	false

39 Get the sequence member and sub-sequence in reversal direction

	A	
1	=[1,2,3,4,5,6].m(-3)	4
2	=[1,2,3,4,5,6].m([-2,-3])	[5,4]

40 Get the sequence member and sub-sequence in cycles

	A	
1	[1,2,3,4,5,6]	
2	=A1.m@r(10)	4
3	=A1.m@r([1,5,10])	[1,5,4]

41 Get the sub-sequence and raise no errors even if they are out of range

	A	
1	=[1,2,3,4,5,6].m@0([10,1,4,5])	The result is [1,4,5] and the members out of range do not appear

42 Get the first non-null value

	A	
1	=[null,1,5,7].ifn()	1

43 Generate a fixed length sequence consisted of the same members

	A	
1	=5*[1]	[1,1,1,1,1]
2	=3.("a")	[a,a,a]

44 Duplicate an sequence (for many times) to generate a new sequence

	A	
1	=[1,2,3]	
2	=A1.dup()	[1,2,3]
3	=3*A1	[1,2,3,1,2,3,1,2,3]

45 Generate continuous integer sequence intervals

	A	
1	=to(8)	[1,2,3,4,5,6,7,8]
2	=to(3,5)	The result is [3,4,5] for counting from 3 to 5
3	=to@s(3,5)	The result is [3,4,5,6,7] for counting 5 numbers from 3
4	=to@s(7,-3)	The result is [7,6,5]

46 Exchange member groups of an sequence

	A	
1	=[1,2,3,4,5,6,7,8].swap([2,3,4],[6,7])	[1,6,7,5,2,3,4,8]

47 Insert one or multiple members in to an sequence

	A	
1	=[1,2,3,4].insert(0,5)	[1,2,3,4,5]. Inserted into the back
2	=[1,2,3,4].insert(1,5)	[5,1,2,3,4]. Inserted into the front
3	=[1,2,3,4].insert(3,[5,6])	[1,2,5,6,3,4]. Insert multiple members

48 Delete one or multiple members from an sequence

	A	
1	=[11,12,13,14].delete(2)	The result is [11,13,14] since delete a member
2	=[11,12,13,14].delete([2,4])	The result is [11,13] since delete multiple members

49 Modify one or multiple members of an sequence

	A	
1	=[11,12,13,14]	
2	>A1(2)=6	The value in A1 is [11,6,13,14]
3	>A1([3,4])=[7,8]	The value in A1 is [11,6,7,8]

50 Modify the sequence member at the specified position and fill up the position in case out of range

	A	
1	=[11,12,13,14,15].modify(2,6)	[11,6,13,14,15]
2	=[11,12,13,14,15].modify(10,10)	[11,12,13,14,15,null,null,null,null,10]
3	=[11,12,13,14,15].modify(2,[7,8,9])	[11,7,8,9,15]

51 Insert the whole sequence into another sequence as one of its members

	A	
1	[1,2,3,4]	
2	[5,6,7,8]	
3	=A1.insert(3,[A2])	[1,2,[5,6,7,8],3,4]

52 Compare in dictionary mode

	A	
1	=cmp(["a","b","c"],["d","e","f"])	-3
2	=cmp(["d","b","c"],["a","e","f"])	3

Chapter 5 TSeq and Records

53 Create a null table sequence

	A	
1	=create(fld1,fld2,fld3)	Create a null TSeq
2	=A1.create()	Create null TSeq of the same structure
3	=A1.dsc(fld1+fld3:c1,fld1*fld3:c2)	Add computed field
4	=A1.create(fld1, fld3; c1, c2)	Create a null TSeq and use the selected normal fields or the computed fields of A1

54 Judge if it is a record or a TSeq

	A	
1	[1,2,3]	
2	=create(fld1).record(A1)	
3	=ifr(A1(1))	false, Judge if it is a record

	A	
4	=ifr(A2(1))	**true**, Judge if it is a record
5	=ift(A1)	**false**, Judge if it is a TSeq
6	=ift(A2)	**true**, Judge if it is a TSeq

55 Find out the TSeq where a record locates

	A	
1	=*r*.home()	Get the TSeq carrying the record

56 Visit fields of the records and assign value to them

	A	
1	=*r*.Name	Visit by column name
2	=*r*.#2	Visit by column number
3	>*r*.#2="Tom"	Assignment
4	>*r*.Name="Tom"	Assignment

57 Acquire column number and the number of fields

	A	
1	=create(name, gender, job, age)	
2	=A1.fno(gender)	column number
3	=A1.fno()	the number of fields

58 Fill sequences into the records as field values

	A	
1	>*r*.paste([1,2,3])	Fill into the records in order

59 Get the field values of records and join them into a sequence

	A	
1	=*r*.field()	Get the field values of a record and join them into a sequence

60 Alter the data structure of the TSeq

	A	
1	=create(number,name,birthday)	
2	>A1.alter(number,name,birthday,position)	Add a field
3	>A1.alter(number,name,position)	Remove a field
4	>A1.alter(id:number,name,position)	Alter the name
5	>A1.alter(id,position,name)	Exchange the orders

61 Add/remove/modify the computed fields on the TSeq

	A	
1	=demo.query("select NAME,UNITPRICE,QUANTITY from RECEIPT")	
2	=A1.dsc(UNITPRICE*QUANTITY:Amount)	Add or modify a computed field
3	=A1.dsc(:Amount)	Remove a computed field

62 Change field values of a record into fields

	A	
1	=demo.query("select NAME,EVENT,SCORE from GYMSCORE")	
2	=A1.group(NAME)	
3	=A2.new(NAME,~.select@1(EVENT:"BalanceBeam").SCORE:BalanceBeam,~.select@1(EVENT:"Floor").SCORE:Floor)	Turn a record into a field

63 Append new records

	A
1	=create(Athlete,Event,Score)
2	>A1.insert(0,90:Score,"s1": Athlete,"Vault":Event)
3	>A1.insert(0:2,90+#:Score,"s2"+string(#):Athlete,"Floor":Event)

Chapter 6 Data Maintenances of TSeq (RSeq)

64 Copy a TSeq entirely

	A	
1	=demo.query("select NAME,EVENT, SCORE from GYMSCORE")	
2	=A1.dup@t()	Copy the TSeq entirely
3	=A1.dup()	Copy as a RSeq only

65 Insert one or multiple null or non-null records into a TSeq

	A	
1	=demo.query("select NAME,EVENT,SCORE from GYMSCORE")	
2	>A1.insert(2)	Insert a null record before the second record
3	>A1.insert(0:10,~:NAME)	Append 10 records at the end of the TSeq and set the **"NAME"** field value

66 Remove one or multiple records from a TSeq

	A	
1	=demo.query("select NAME, EVENT, SCORE from GYMSCORE")	
2	>A1.delete(2)	Remove the second record
3	>A1.delete([4,6,1])	Remove multiple records
4	>A1.delete(A1.select(SCORE<15))	Remove records on specified conditions

67 Modify field values of one or multiple records

	A	
1	=demo.query("select NAME,EVENT, SCORE from GYMSCORE")	
2	>A1.run(SCORE+2:SCORE)	Modify all the records
3	>A1.select(EVENT=="Vault").run(SCORE+ 2:SCORE)	Modify part of the records

68 Modify the record at a specified position and fill up the positions automatically in case record is out of range

	A	
1	=demo.query("select NAME,EVENT, SCORE from GYMSCORE")	TSeq
2	=A1.modify(5,"":EVENT, 15:SCORE)	Modify the fifth record and fill up the positions automatically in case the record is out of range

69 Add a calculation column to a TSeq

	A

1	=demo.query("select NAME,UNITPRICE, QUANTITY from RECEIPT")
2	>A1.derive(UNITPRICE*QUANTITY:Amount) Add the "**Amount**" field

70 Create a new TSeq based on the specified TSeq (RSeq)

	A
1	=demo.query("select NAME,ABBR,CAPITAL, POPULATION from STATES")
2	=A1.new(NAME, ABBR) Create a new TSeq based on an original TSeq
3	=A1.new(NAME:State,ABBR,CAPITAL:Capital) Change the field name

71 Combine or split the TSeq

	A	B
1	=demo.query("select * from STUDENTS1")	
2	=demo.query("select * from STUDENTS2 ")	
3	=A1.append(A2)	Combine these TSeqs

72 Insert sequence into the newly created TSeq to generate new records

	A	B
1	1	Tom
2	2	Jack
3	3	Andy
4	=create(id,name)	
5	>A4.record([A1:B3])	

73 Get the field values of a TSeq (RSeq) and join them into a sequence

	A	B
1	1	Tom
2	2	Jack
3	3	Andy
4	=create(id,name)	
5	>A4.record([A1:B3])	
6	=A4.field()	Return the same sequence as **[A1:B3]**

74 Add the sequence members into a TSeq (RSeq) as the field values

	A	B
1	1	Tom
2	2	Jack
3	3	Andy
4	=create(id, name).insert(1:3)	
5	>A4.paste@a([A1:B3])	

75 Add the sequence members to a TSeq (RSeq) vertically as the field values

	A	B	C
1	1	2	3
2	Tom	Jack	Andy
3	=create(id, name).insert(1:3)		
4	>A3.paste@h([A1:C1],1)		The first column
5	>A3.paste@h([A2:C2],2)		The second column

76 Copy the field values of a TSeq (RSeq) into another TSeq (RSeq)

	A
1	=create(ordernumber, ordervalue).insert(0:10)
2	=demo.query("select NAME,UNITPRICE, QUANTITY from RECEIPT")
3	>A1.paste(A2)

Chapter 7 External Files
77 Read/write a text file

	A	
1	=file("D:\\test.txt")	
2	>A1.write("USA")	Write out a string
3	=A1.read()	Read and return it as a string

78 Record log in a text file

	A	
1	=file("D:\\test.log")	
2	>A1.write@a(string(now())+": Start getting data")	"@a" indicates appending and writing out

79 Read out the TSeq from a text file

	A	
1	=file("D:\\employee.txt")	
2	=A1.import()	
3	=A1.import@t()	The first line is used for field names

80 Save the TSeq in a text file

	A	
1	=demo.query("select EID,NAME,STATE, GENDER, BIRTHDAY,HIREDATE,DEPT, SALARY from EMPLOYEE")	
2	=file("D:/employee.txt")	
3	>A2.export(A1)	
4	>A2.export@t(A1)	Set field name as the title

Chapter 8 Database
81 Retrieve the data from database to TSeq via SQL

	A
1	=demo.query("select * from EMPLOYEE")
2	=demo.query("select * from EMPLOYEE where EID=?",1)

82 Return the single value result computed via SQL

	A
1	=demo.query@1("select count(*) from EMPLOYEE")
2	=demo.query@1("select count(*) from EMPLOYEE where SALARY>?",10000)

83 Use the stored procedure of database to return one or multiple TSeqs

	A	
1	=db.proc("{call proc1(?,?)}",:101:"o":a,:101:"o":b)	Execute the stored procedure and return 2 TSeqs
2	=A1(1)	The first TSeq

| 3 | =A1(2) | The second TSeq |
| 4 | =a | Use a variable name to visit the first TSeq |

84 Run SQL statement on the database to modify the data

	A	
1	>demo.execute("update SCORES set SCORE=? where STUDENTID=10", 90)	Update
2	=demo.query("selct * from LIQUORS")	
3	>demo.execute(A2,"update LIQUORS set STOCK=? where NAME=?", wineStock, wineName)	Update in batches
4	>demo.execute([1,3,5],"delete from product where productnumber=?",~)	Delete in batches
5	>demo.execute(A2,"insert into LIQUORS (LID, NAME, TYPE, PRODUCTION, STOCK) values (?,?)",wineID, wineName, wineType, wineProduction, wineStock)	Insert in batches

85 Use the program codes to connect and close a database

	A	
1	=connect("demo")	Connect a database
2	>A1.close()	Close the connection

86 Manage the transaction submission by the program code itself

	A	B	
1	=connect@e("demo")		Create a connection
2	=A1.execute@k(....)		
3	=A1.error()		Read the previous item of database execution error message
4	if A3==0	>A1.commit()	Submitted if there are no errors
5	else	>A1.rollback()	Roll back if there are errors
6	>A1.close()		Close the connection

87 Get the database error messages

	A	
1	=connect@e("demo")	
2	...	
3	=A1.error()	Error codes
4	=A1.error@m()	Error messages

88 Use a cursor to fetch more data in batches

	A	B	C	
1	=demo.cursor("select * from STOCKRECORDS")			
2	for			
3		=A1.fetch(1000)		Get **1000** records and return them as a TSeq
4		if B3==null	break	Break when the data retrieving is finished
5		...		

89 Write a TSeq (RSeq) into the database

	A
1	=demo.query("select ID, NAME,GENDER,AGE from STUDENTS")
2	=A1.primary(ID)
3	=demo.update(A1,STUDENTS1,ID, NAME)

4	=demo.update@u(A1,STUDENTS1,ID, NAME)	Generate "**update**" only
5	=demo.update@i(A1,STUDENTS1, ID, NAME)	Generate "**insert**" only
6	=demo.update@a(A1,STUDENTS1, ID,NAME)	Empty the target table before inserting data

Chapter 9 Program Logic

90 Join the data in a program cellset into a sequence

	A	B	C	D
1	1	2	3	4
2	=[A1:D1]			

91 Implement the switch/case structure

	A	B
1	=80	
2	if A1>=90	
3		> A10="excellent"
4	else if A1>=80	
5		> A10="good"
6	else if A1>=60	
7		> A10="pass"
8	else	
9		> A10="fail"
10		**good**

92 Get the occurrence of the current loop

	A	B	C
1	for	if #A1==10000	break

93 Proceed /exit the outer loop

	A	B	C	D	
1	for [3,2,1]				
2		for [5,1,3]			
3			if A1>B2	next A1	Proceed to the next loop
4			if A1== B2	break A1	Exit the outer loop

94 Clear the used variables to free memory

	A	
1	>var1=to(100)	
2	=demo.query("select * from EMPLOYEE")	
3	…	
4	> var1=null,A2=null	Clear variable **var1** and cell **A2**

95 Pass multiple arguments to the subroutine

	A	B	
1	func		Multiple arguments are arranged in order
2		=A1	
3		=B1	
4		return B2+ B3	
5	=func(A1,11,21)		

96 Return multiple values by subroutine

	A	B
1	func	
2		return [1,2,3,4]
3		
4	=func(A1)	

97 Comment multi-line codes

	A	B
1	//This is an example about...	
2		1. Remark 1...
3		2. Remark 2...
4	=1+3	

The words from line 1 to line 3 are all comments

98 Write a long statement in multiple cells

	A	B	C	D
1	68			
2	==if(A1>100:"excellent",	A1>80:"good",	A1>60:"pass",	"fail")

99 Use macro in codes

	A	B
1	[1,2,3,4]	
2	func	return A1.${A2}()
3	func	return A1.${lower(A3)}()
4	=func(A2,"sum")	
5	=func(A3,"Avg")	

Return **A1.sum()**
Return **A1.avg()**

Volume of Operation

Chapter 1 Set, Aggregation and Repetitions

100 Judge an integer sequence

	A
1	[1,2,3,4]
2	=A1.pselect(~!=int(~))==0

Check if any member is not integer

101 Judge an ascending integer sequence

	A
1	[1,2,3,4,5]
2	=A1.pselect(~!=int(~) || ~<=~[-1])==0

102 Judge if it is a member or a subset of an integer sequence

	A
1	[1,2,3,4,5,6,7]
2	=A1.pos(2)!=0
3	=A1.pos([2,3])!=0
4	=A1.pos([3,2])!=0
5	=A1.pos([2,5])!=0
6	=A1.posi([2,3])!=null
7	=A1.posi([3,2])!=null

true for member
true for continuous subset
false
false
true for subset
false

	A	
8	=A1.posi([2,5])!=null	**true**

103 Judge if members from different sequences are equal

	A	
1	[1,2,3]	
2	[3,2,1]	
3	=A1.eq(A2)	**true**

104 Perform union, intersect, and difference operations on sorted integer sequences by merging

	A	
1	=demo.query("select CLASS,STUDENTID, SUBJECT, SCORE from SCORES where CLASS=? and SUBJECT=? and STUDENTID<?","Class one", "Math",10)	
2	=demo.query("select CLASS,STUDENTID, SUBJECT,SCORE from SCORES where CLASS=? and SUBJECT=? and STUDENTID>?","Class two", "Math",5)	
3	=A1.sort(STUDENTID)	
4	=A2.sort(STUDENTID)	
5	=[A3:A4].merge(STUDENTID)	Union all
6	=[A3:A4].merge@u(STUDENTID)	Union
7	=[A3:A4].merge@i(STUDENTID)	Intersect
8	=[A3:A4].merge@d(STUDENTID)	Difference

105 Calculate the quadratic sum and variance

	A	
1	[1,2,3,4,5,6,7,8]	
2	=A1.sum(~*~)	Quadratic sum
3	=A1.variance()	Variance
4	=demo.query("select NAME,EVENT,SCORE from GYMSCORE")	
5	=A4.variance(SCORE)	
6	= A4.sum2(SCORE*SCORE)	

106 Calculate the weighted average

	A
1	[9,9.1,8.5,9.8,9.4]
2	[0.9,0.8,1.0,0.95,1.0]
3	=(A1**A2).sum()/A2.sum()

107 Calculate the average of an integer sequence after removing the max and the min

	A
1	[99,98,95,93,87,89,90,96,94]
2	=(A1.sum()-A1.max()-A1.min())/(A1.len()-2)
3	=(A1\A1.min()\A1.max()).avg()

108 Get the adjacent row and calculate period-over-period

	A
1	[1,2,3,4,5,6]
2	=A1.(~/~[-1]-1)

	A
3	=demo.query("select DATE,sum(CLOSING) AMOUNT from STOCKRECORDS GROUP BY DATE")
4	=A3.derive(AMOUNT/AMOUNT[-1]-1: Period-over-period)

109 Get adjacent sets and calculate a moving average

	A
1	[1,2,3,4,5,6]
2	=A1.(~{-1,1}.avg())
3	=demo.query("select STOCKID, DATE,CLOSING from STOCKRECORDS where STOCKID=?","000062")
4	=A3.(CLOSING{-3,3}.avg())

110 Calculate the proportion and cumulation

	A	
1	[1,2,3,4,5,6,7]	
2	=A1.proportion()	Proportion
3	=A1.cumulate()	Cumulate
4	=demo.query("select ABBR,POPULATION from STATES")	
5	=A4.sum(POPULATION)	
6	=A4.(POPULATION/A5)	
7	=A4.loop(~~+POPULATION)	
8	=A4.derive(POPULATION/A5:Proportion)	
9	=A4.derive(POPULATION{,0}.sum():Addup)	
10	=A4.derive(Addup[-1]+POPULATION:Addup2)	

111 Union record sequences with different data structures

	A
1	=demo.query("select * from STUDENTS ")
2	=demo.query("select * from EMPLOYEE ").derive(interval@y(BIRTHDAY,now()):AGE)
3	=A1|A2
4	=A3.select(GENDER:"F").avg(AGE)

Chapter 2 Retrieving and Locating

112 Locate a member

	A	
1	[3,2,1,5,1]	
2	=A1.pos(1)	3
3	=A1.pos@a(1)	[3,5]

113 Locate a sub sequence

	A	B	
1	[1,2,3,4,5]	[4,5]	
2	=A1.pos(B1)		4

114 Locate members matching specified conditions and return their positions

	A
1	=demo.query("select EID,NAME,STATE, GENDER,BIRTHDAY,HIREDATE,DEPT, SALARY from EMPLOYEE")

	2	=A1.select(GENDER=="M")	Locate all members matching the specified condition
	3	=A1.pselect(GENDER=="M")	Stop searching after the first matching member is found, and return the member position

115 Find members matching specified conditions from the rear to the front

	A
1	=demo.query("select EID, NAME,STATE, GENDER, BIRTHDAY,HIREDATE,DEPT,SALARY from EMPLOYEE")
2	=A1.select@z(GENDER=="M")
3	=A1.pselect@z(GENDER=="M")

116 Stop searching after the first/all members matching specified conditions are found

	A	
1	=demo.query("select NAME,EVENT, SCORE from GYMSCORE")	
2	=A1.pselect(EVENT:"UnevenBars")	Stop searching after the first matching member is found
3	=A1(A2).SCORE	The Score on a specified event
4	=demo.query("select EID,NAME,STATE, GENDER,BIRTHDAY,HIREDATE,DEPT,SALARY from EMPLOYEE")	
5	=A4.select(GENDER=="M")	Find all matching members
6	=A4.select@1(GENDER=="M")	Find the first matching member
7	=A4.pselect@a(GENDER=="M")	Locate all matching members

117 Search from the Kth member

	A
1	=demo.query("select EID,NAME,STATE,GENDER,BIRTHDAY,HIREDATE, DEPT,SALARY from EMPLOYEE")
2	=A1.pselect(GENDER=="M",8)

118 Retrieve multiple fields

	A
1	=demo.query("select EID,NAME,STATE,GENDER,BIRTHDAY,HIREDATE, DEPT, SALARY from EMPLOYEE ")
2	=A1.select(GENDER:"M",DEPT:"R&D")
3	=A1.pselect(GENDER:"M",DEPT:"R&D")
4	=A1.pselect@a(GENDER:"M",DEPT:"R&D")

119 Speed up the sorted sequence (RSeq) retrieval via binary search

	A
1	=demo.query("select * from EMPLOYEE order by GENDER,DEPT")
2	=A1.select@b(GENDER:"M",DEPT:"R&D")
3	=A1.pselect@b(GENDER:"M",DEPT:"R&D")

120 Aggregate records retrieved

	A

	A
1	=demo.query("select NAME,UNITPRICE,QUANTITY from RECEIPT")
2	=A1.sumif(UNITPRICE*QUANTITY; NAME:"Apple")
3	=A1.select(NAME:"Apple").sum(UNITPRICE*QUANTITY)

121 Filter a TSeq

	A
1	=demo.query("select EID,NAME,STATE,GENDER,BIRTHDAY,HIREDATE, DEPT, SALARY from EMPLOYEE ")
2	=A1.select@o(SALARY<6300)
3	=A1.select(SALARY>5000)
4	=A1.compose(A3)

122 Retrieve data in a RSeq (TSeq) according to the primary key value

	A
1	=demo.query("select * from SCORES")
2	=A1.primary(CLASS,STUDENTID)
3	=A1.pfind(["Class one",2])
4	=A1(A3)
5	=A1.find(["Class one",2])

123 Get the record with max/min key value and its position

	A	
1	=demo.query("select EID,NAME,STATE,GENDER, BIRTHDAY,HIREDATE,DEPT,SALARY from EMPLOYEE ")	
2	=A1.sort(HIREDATE)	Sort
3	=A2.pmax(BIRTHDAY)	The position of the youngest staff
4	=A2(to(A3-1))	The records of staff employed earlier
5	=A4.minp(BIRTHDAY).NAME	The name of the oldest staff

124 Calculate the period over period value for the selected members

	A	B	
1	=demo.query("select * from STOCKRECORDS")		
2	=A1.pselect@a(CLOSING>10)		Locate the **DATE** on which the closing prices are greater than 10
3	=A1.calc(A2, CLOSING/CLOSING[-1]-1)		Calculate the corresponding increase
4	==A2.new(	A1(~).DATE:Date,	Output the result set
5		A1(~).CLOSING:ClosingPrice,	
6		A3(#):Increase)	

Chapter 3 Sorting and Locating

125 Get members in odd positions

	A	
1	[1,2,3,4,5,6,7,8,9,10]	
2	=A1.step(2,1)	[1,3,5,7,9]

126 Calculate ranking

	A	
1	=demo.query("select NAME,EVENT,SCORE from GYMSCORE")	
2	=A1.rank(SCORE)	Calculate the ranking of all scores

	A	
3	=A1.ranki(16, SCORE)	The rank of **16** points
4	=[99,98,97,96,93,87,99,95].ranki(98)	The rank of **98** in the integer sequence

127 Members with scores ranking top 10, the third, the second from the bottom of the sequence, and the median

	A	
1	=demo.query("select NAME,EVENT,SCORE from GYMSCORE")	
2	=A1.sort@o(SCORE:-1)	
3	=A1(to(10))	Members with scores ranking top **10**
4	=A1.m([3,-2])	The member with the score ranking the third, and the second from the bottom of the sequence
5	=round(A1.len()/2)	
6	=A1(A5)	median

128 Calculate for the top 3 over the previous period

	A	
1	=demo.query("select * from STOCKRECORDS where STOCKID=?","000062")	
2	=A1.sort(DATE)	Sort by **DATE**
3	=A2.psort(CLOSING:-1)	Sort by **CLOSING**
4	=A3(to(3))	The numbers of records for the three dates with highest closing prices
5	=A4.(A2.calc(A4.~, CLOSING- CLOSING[-1]))	Calculate the increases for the three days

129 Members ranking at the top 20% and the middle 50%

	A	
1	=demo.query("select NAME,EVENT,SCORE from GYMSCORE")	
2	=A1.sort(SCORE:-1)	
3	=A2.len()	Total number of people
4	=round(A3*0.2)	The number of members ranking at the top 20%
5	=A2(to(A4))	Members ranking at the top 20%
6	=round(A3*0.25)	Positions of members ranking at the first 25% of the middle
7	=round(A3*0.75)	Positions of members ranking at the last 25% of the middle
8	=A2(to(A6,A7))	Records of members ranking at the middle 50%
9	=A8(1)	The highest score of middle ranking members
10	=A8.m(-1)	The lowest score of middle ranking members

130 Select 10 members randomly

	A
1	=demo.query("select NAME,EVENT,SCORE from GYMSCORE")
2	=A1.sort(rand())(to(10))

131 Count the max continuity interval

	A
1	=demo.query("select * from STOCKRECORDS where STOCKID=?","000062")
2	=A1.sort(DATE)
3	=A2.max(a=if(CLOSING/CLOSING[-1]>=1.05,a+1,0))

The max continuity interval (day) for the increase greater than 5%

132 Sort a Tseq

	A
1	=demo.query("select * from SCORES").sort@o(SCORE:-1)
2	=demo.query("select * from SCORES").psort(SCORE:-1)
3	=demo.query("select * from SCORES").compose(A2)

133 Sort according to the specified order

	A
1	[CA,IL,KY,CO,NY]
2	=demo.query("select NAME,ABBR,CAPITAL,POPULATION from STATES")
3	=A2.align(A1,ABBR)

134 Create a binary search index for a RSeq

	A	
1	=demo.query("select NAME,EVENT,SCORE from GYMSCORE")	
2	=A1.sort(SCORE:-1)	Create a binary search index
3	=A2.select@b(SCORE:14.175)	Return matching records after the binary search is completed
4	=A1.psort(SCORE:-1)	Create an index
5	=A1(A4).pselect@b(SCORE:14.175)	
6	=A4(A5)	Return the number of the record matching the specified condition, after the binary search is completed

Chapter 4 Common Group

135 Get the distinct value of a field

	A
1	=demo.query("select NAME,EVENT,SCORE from GYMSCORE")
2	=A1.id(EVENT)

136 Delete repeated members

	A	
1	=demo.query("select NAME,TYPE,PRODUCTION from LIQUORS")	
2	=A1.id@d(TYPE)	Delete repeated members without changing the member order
3	=A1.group@1d(TYPE)	
4	=[1,2,2,3,3,4,5,6,2,3].id@d()	[1,2,3,4,5,6]

137 Delete repeated adjacent members

	A	
1	=demo.query("select NAME,TYPE, PRODUCTION from LIQUORS")	
2	=A1.id@o(TYPE)	Delete repeated adjacent members without sorting
3	=A1.group@1o(TYPE)	
4	=[1,2,2,3,3,4,5,6,2,3].id@o()	[1,2,3,4,5,6,2,3]

138 Combine the grouping results into a TSeq

	A
1	=demo.query("select NAME,EVENT,SCORE from GYMSCORE")
2	=A1.group(EVENT)
3	=A2.conj()

139 Set five members in each group

	A
1	=demo.query("select NAME,EVENT,SCORE from GYMSCORE")
2	=A1.group(int((#-1)/5))

140 Compute the aggregate after grouping

	A
1	=demo.query("select NAME,EVENT,SCORE from GYMSCORE")
2	=A1.groups(EVENT:GymEvent; sum(SCORE):TotalScore)
3	=A1.group(EVENT)
4	=A3.new(EVENT: GymEvent,~.sum(SCORE): TotalScore)

141 Get subsets of grouping results

	A	
1	=demo.query("select EID,NAME,STATE,GENDER, BIRTHDAY,HIREDATE,DEPT,SALARY from EMPLOYEE")	
2	=A1.group(DEPT)	
3	=A2.maxp(~.avg(interval@y(BIRTHDAY,now())))	The records of position players whose average age is the highest

142 Refilter or sort grouping results

	A	
1	=demo.query("select NAME,EVENT,SCORE from GYMSCORE")	
2	=A1.group(EVENT)	
3	=A2.select(~.avg(SCORE)>14.3)	
4	=A3.sort@o(~.avg(SCORE):-1)	
5	=A4(to(2))	Top **2** athletes with the max average score
6	=demo.query("select EID,NAME,STATE,GENDER, BIRTHDAY,HIREDATE,DEPT,SALARY from EMPLOYEE")	
7	=A6.groups(DEPT;count(interval@y(BIRTHDAY,now())>40):Number)	
8	=A7.select(Number>=20).(DEPT)	Positions with more than 20 employees over their 40s

143 Refilter or sort the subsets of grouping results

	A	
1	=demo.query("select NAME,EVENT,SCORE from GYMSCORE")	
2	=A1.group(EVENT)	
3	>A2.(~=~.sort(SCORE:-1))	
4	>A2.(~=~(to(2)))	
5	=A2.(~.(NAME)).isect()	Athletes with all event scores at the top **2**

144 Regroup the subsets of the grouping results

	A	
1	=demo.query("select EID,NAME,STATE,GENDER, BIRTHDAY,HIREDATE,DEPT,SALARY from EMPLOYEE ")	
2	=A1.group(DEPT)	
3	>A2.(~=~.group(month(BIRTHDAY),day(BIRTHDAY)))	Regroup the subsets
4	=A2.maxp(~.count())	
5	=A4(1). DEPT	

145 Perform the cross-row calculation for members in a group

	A	B	C	
1	=demo.query("select * from STOCKRECORDS ")			
2	=A1.group(STOCKID).(~.sort(DATE))			
3	for A2	=0		
4		if A3.pselect(B3=if(CLOSING/ CLOSING[-1] >=1.05,B3+1,0):4)>0		Harden for three days
5			=C5|A3.STOCKID	Record of the result

146 Get a specified member from each grouped subset

	A	
1	=demo.query("select NAME,TYPE,PRODUCTION from LIQUORS")	
2	=A1.group(TYPE).new(TYPE,~.m(-1):Last)	Group and aggregate directly
3	=demo.query("select EID,NAME,STATE,GENDER, BIRTHDAY,HIREDATE,DEPT,SALARY from EMPLOYEE ")	
4	=A3.group(DEPT)	Group first
5	=A4.(~.minp(SALARY))	Then aggregate

147 Find the member which appears most frequently

	A	
1	=demo.query("select EID,NAME,STATE,GENDER, BIRTHDAY,HIREDATE,DEPT,SALARY from EMPLOYEE ")	
2	=A1.group(DEPT)	Group
3	=A2.maxp(~.count())	Find the group with most employees
4	=A3(1). DEPT	The **DEPT** with most employees

Chapter 5 Affinity Grouping & Join

148 Group according to the specified classification

	A
1	[America,Jamaica,France,Scotland,England]
2	=demo.query("select NAME,TYPE,PRODUCTION from LIQUORS")
3	=A2.align@a(A1,PRODUCTION)

Group by **PRODUCTION**

149 Group according to the specified condition ranges

	A	B
1	?<25	Below 25
2	?>=25 && ?<=30	25 to 30
3	?>=30 && ?<=40	30 to 40
4	?>=40 && ?<=50	40 to 50
5	?>50	over 50
6	=create(Section,AgeGroup).record([A1:B5])	
7	=demo.query("select EID,NAME,STATE,GENDER, BIRTHDAY,HIREDATE,DEPT, SALARY from EMPLOYEE").derive(interval@y(BIRTHDAY,now()):AGE)	
8	=A7.enum@r(A6:Section,AGE)	
9	=A8.new(A6(#).AgeGroup:AgeGroup, ~.count():Number, ~.avg(AGE):AverageAge)	

Group by **AGE** (aligned with row 8)

150 Conditionally group with possible range overlap

	A
1	[?<5000,?>=7000,?>10000]
2	=demo.query("select EID,NAME,STATE,GENDER, BIRTHDAY,HIREDATE,DEPT, SALARY from EMPLOYEE")
3	=A2.enum@r(A1,SALARY)

151 Join tables on equivalence conditions

	A
1	=demo.query("select * from STATES")
2	=demo.query("select * from EMPLOYEE")
3	=join(A1:State.NAME;A2:Employee,STATE)

152 Join tables based on the first one (left join)

	A
1	=demo.query("select * from STATES")
2	=demo.query("select * from EMPLOYEE")
3	=join@1(A1:State,NAME;A2:Employee,STATE)

153 Join records even if specified conditions are not matched (full join)

	A
1	=demo.query("select * from STATES")
2	=demo.query("select * from EMPLOYEE")
3	=join@f(A1:State,NAME;A2:Employee,STATE)

154 Align tables if the specified fields in the respective tables are equal

	A
1	=demo.query("select * from EMPLOYEE")

	A
2	=demo.query("select * from ATTENDANCE")
3	=demo.query("select * from PERFORMANCE")
4	=join@1(A1:Employee,EID;A2:Attencance,EMPLOYEEID;A3:Performance, EMPLOYEEID)

155 Perform a common join under non-equal conditions

	A
1	=demo.query("select * from STATES")
2	=demo.query("select * from CITIES")
3	=demo.query("select * from GYMSCORE")
4	=xjoin(A1:State,left(NAME,1)=="A";A2:City,POPULATION>1000000; A3:Score,EVENT=="Floor")

156 Perform unconditional join (full interleaving) operations

	A
1	=demo.query("select * from STATES")
2	=demo.query("select * from STUDENTS")
3	=xjoin(A1:State;A2:Student)

157 Convert foreign key references into record fields

	A	
1	=demo.query("select * from CITIES")	
2	=demo.query("select * from STATES where STATEID<?",51)	
3	=A1.switch(STATEID,A2)	Create a reference between the primary table and a subtable
4	=A1.group(STATEID.REGIONID)	Directly access the primary table via reference fields
5	=A2.run(CAPITAL=A1.select@1(NAME==CAPITAL))	
6	=A1.new(NAME,STATEID.CAPITAL.NAME:StateCapital)	
7	=A1.select(STATEID.CAPITAL.POPULATION>1000000)	

158 Convert members of a subtable into record sequence fields

	A	
1	=demo.query("select EID,NAME,STATE,GENDER,BIRTHDAY, HIREDATE,DEPT,SALARY from EMPLOYEE ")	
2	=demo.query("select * from FAMILY where RELATION=?","Child")	
3	=A1.select(GENDER=="F" && A2.id(EID).pos(EID)>0)	
4	=A3.run(EID=A2.select(EID==A3.EID))	Create a reference between the primary table and a subtable
5	>A3.(EID=EID.sort(AGE:-1))	Sort a record sequence field again
6	=A3.new(NAME,EID(1).GENDER:GenderOfFirstChild,interval@y(BIRTHD AY,now())-EID(1).AGE:ReproductiveAge)	Directly count a record sequence field

Volume of SQL

Chapter 1 General Computation

159 IS NULL/NVL/COALESCE

	A
1	=demo.query("select EID,NAME,STATE,GENDER,BIRTHDAY,

	A	
	HIREDATE,DEPT, SALARY from EMPLOYEE ")	
2	=A1.select(DEPT!=null)	Not null
3	=A1.select(DEPT ==null)	Null
4	=demo.query("select NAME,UNITPRICE,QUANTITY from RECEIPT ")	
5	=A4.(NAME).ifn()	The first non-null member

160 CAST/CONVERT

	A	
1	=date("1983-09-12")	Convert character string to date
2	=string(A1,"yyyymmdd")	Convert date to character string
3	=int("5")	Convert character to integer
4	=string(5)	Convert integer to character
5	=decimal(A3)	Convert integer to decimal
6	=ifnumber(A5)	Judge if **A5** is a number
7	=double("234")	Convert character to double

161 CASE/DECODE

	A
1	1
2	=case(A1,1,"ClassOne",2,"ClassTwo","ClassThree")
3	=if(A1==1, "ClassOne", A1==2,"ClassTwo","ClassThree")

162 AND/OR/NOT, <>

	A	
1	=demo.query("select * from SALES")	
2	=A1.select(CLIENT=="PWQ" || CLIENT=="QUICK")	OR
3	=A1.select(AMOUNT>5000 && AMOUNT<10000)	AND
4	=A1.select(!(CLIENT=="TRADH"))	NOT
5	=A1.select(CLIENT!="TRADH")	<>

163 LIKE

	A
1	=demo.query("select * from SALES")
2	=A1.select(like(CLIENT,"*AY*"))

164 COUNT/SUM/AVG/MAX/MIN

	A
1	=demo.query("select * from SALES")
2	=A1.sum(AMOUNT)
3	=A1.count()
4	=A1.avg(AMOUNT)
5	=A1.max(AMOUNT)
6	=A1.min(AMOUNT)

165 IN/EXISTS

	A	
1	[1,3,5,7,9]	
2	=demo.query("select * from EMPLOYEE ")	
3	=A2.select(A1.pos(EID)>0)	IN
4	[English,Maths]	
5	=demo.query("select * from SCORES")	

	6	=A5.select(SCORE>75).group(STUDENTID)	
	7	=A6.select(~.(SUBJECT).posi(A4)!=null)	EXISTS
	8	=A7.(STUDENTID)	

Chapter 2 Data Retrieval

166 SELECT * FROM …

	A
1	=demo.query("select * from EMPLOYEE")

167 WHERE …

	A
1	=demo.query("select * from EMPLOYEE")
2	=A1.select(SALARY>5000)

168 SELECT … FROM

	A
1	=demo.query("select * from EMPLOYEE")
2	=A1.new(EID,NAME)

169 AS

	A
1	=demo.query("select * from EMPLOYEE")
2	=A1.new(EID:EmployeeNo,NAME+" "+SURNAME: EmployeeName)

170 SELECT …

	A	
1	=new(1:TypeNumber,"Beverage":Name)	Create a TSeq of only one record
2	[1,2,3,4,5]	
3	=create(TypeNumber,TypeName).record([1,"Beverage",2,"Grain"])	Create a null TSeq, and fill in data

171 ORDER BY/ASC/DESC

	A
1	=demo.query("select * from EMPLOYEE")
2	=A1.sort(BIRTHDAY:1,SALARY:-1)

172 DISTINCT

	A	
1	=demo.query("select * from SALES")	
2	=A1.id(CLIENT)	Get the distinct value
3	=A1.([CLIENT,SELLERID])	All values available
4	=demo.query("select * from SALES")	All available values in the group

173   FISRT/LAST/TOP/BOTTOM

	A	
1	=demo.query("select * from EMPLOYEE")	
2	=A1.m(1).NAME	FIRST
3	=A1.m(-1).NAME	LAST
4	=A1.m(to(3))	TOP 3

	A	
5	=A1.m(to(-1,-3))	BOTTOM 3

174 UNION/UNION ALL/INTERSECT/MINUS

	A	
1	=demo.query("select * from EMPLOYEE")	
2	=A1.select(DEPT=="Sales" || DEPT=="R&D")	
3	=A1.select(SALARY>5000)	
4	=A2|A3	UNION ALL
5	=A2&A3	UNION
6	=A2^A3	INTERSECT
7	=A2\A3	MINUS

175 SELECT … FROM (SELECT …)

	A	
1	=demo.query("select * from EMPLOYEE")	
2	=A1.select(DEPT=="Sales")	Query
2	=A2.count()	Recount the result set

176 SELECT (SELECT … FROM) FROM

	A	
1	=demo.query("select * from EMPLOYEE")	
2	=demo.query("select * from FAMILY")	
3	=A1.dup@t()	
4	=A1.run(EID=A2.select(EID:A1.EID))	Calculate the subtable reference first
5	=A1.new(NAME,EID.count():NumberOfMembers)	
6	=A3.new(NAME,A2.select(EID:A3.EID).count():NumberOfMembers)	Use the direct-write method

177 CURSOR/FETCH

	A	B	C	
1	=demo.cursor("select * from SALES")			
2	for			
3		=A1.fetch(100)		Fetch **100** records each time
4		if B3==null	break	
5		...		

Chapter 3 Group and Association

178 GROUP BY

	A	
1	=demo.query("select * from EMPLOYEE")	
2	=A1.groups(DEPT;sum(SALARY): SalarySum)	Group and aggregate
3	=A1.group(DEPT)	First, divide into groups
4	=A3.new(DEPT,~.count():EmployeeNumber)	Then, aggregate

179 HAVING

	A	
1	=demo.query("select * from EMPLOYEE")	
2	=A1.groups(DEPT;sum(SALARY): SalarySum)	Aggregate by group
3	=A2.select(SalarySum>200000)	Filter the results aggregated

	A	
4	=A1.group(DEPT)	Group
5	=A4.select(~.count()>30)	Filter grouped subsets

180 Perform the equi-join on two tables with the same level

	A
1	=demo.query("select * from STATENAME")
2	=demo.query("select * from STATEINFO")
3	=join(A1:StateName,STATEID;A2:StateInfo,STATEID)

181 Perform the equi-join on the primary table and the subtable

	A
1	=demo.query("select * from STATES")
2	=demo.query("select * from EMPLOYEE")
3	=join(A1:State,NAME;A2:Employee,STATE)

182 Non-Equijoin

	A
1	=demo.query("select * from EMPLOYEE")
2	=demo.query("select * from LIQUORS")
3	=demo.query("select * from RECEIPT")
4	=xjoin(A1:Employee,STATE=="New York";A2:Liquor,STOCK>500; A3:Food,QUANTITY>2)

183 LEFT JOIN/FULL JOIN

	A	
1	=demo.query("select * from EMPLOYEE")	
2	=demo.query("select * from STATES")	
3	=demo.query("select * from ATTENDANCE")	
4	=demo.query("select * from PERFORMANCE")	
5	=join@1(A2:State,NAME;A1:Employee,STATE)	Left join
6	=join@f(A2:State,NAME;A1:Employee,STATE)	Full join
7	=join@1(A1:Employee,EID;A3:Attendance, EMPLOYEEID;A4:Performance, EMPLOYEEID)	Align and join with the first table

Chapter 4 Data Maintenance and Structure

184 INSERT

	A	
1	=demo.query("select * from SCORES ")	
2	=A1.insert(0,"Class one",20,"PE",100)	add a new record
3	=A1.insert(5,"Class one",21,"PE",100)	Insert a new record

185 INSERT FROM SELECT …

	A	
1	=demo.query("select * from SALES")	
2	=A1.dup@t()	Duplicate the TSeq

186 DELETE … WHERE …

	A
1	=demo.query("select ORDERID,CLIENT,SELLERID,

	A	
	ORDERDATE,AMOUNT from SALES")	
2	=A1.delete(A1.select(AMOUNT<10000))	Delete the matching records

187 UPDATE … WHERE …

	A	
1	=demo.query("select ORDERID,CLIENT,SELLERID, ORDERDATE, AMOUNT from SALES")	
2	=A1.select(CLIENT:"HL").run(AMOUNT= int(AMOUNT*1.1))	Update data on conditions

188 CREATE/DROP TABLE

	A	
1	=create(ProductNo,ProductName,UnitPrice,Quantity)	Create a TSeq
2	>A1=null	Clear the TSeq

189 ALTER TABLE

	A	
1	=demo.query("select * from EMPLOYEE")	
2	=A1.alter(ID:EID,NAME,DEPT)	Alter the table structure

190 PRIMARY

	A	
1	=demo.query("select * from EMPLOYEE")	
2	=A1.primary(EID)	Set a primary key

191 CONNECT/DISCONNECT/COMMIT/ROLLBACK

	A		
1	=connect@e("demo")		Create a connection
2	>A1.execute@k(...)		
3	=A1.error()		Error message arising from the previous database operation
4	if A3==0	>A1.commit()	Commit if no errors
5	else	>A1.rollback()	Rollback if any errors
6	>A1.close()		Close the connection