

Structure Query Language (SQL)

1- និយមន័យ

SQL គឺជាភាសាពិសេសមួយ ដែលបានកើតឡើងពីការអភិវឌ្ឍនៃ Relational Model ហើយត្រូវបានក្លាយទៅជាភាសាសំរាប់ Relational Database។ ក្នុងឆ្នាំ ១៩៨៦ ខ្នាតគំរូសំរាប់ SQL ត្រូវបានកំណត់ដោយវិទ្យាស្ថានថ្នាក់ជាតិរបស់សហរដ្ឋអាមេរិច (American National Standard Institute = ANSI) ហើយ ត្រូវបានអនុម័តឲ្យក្លាយទៅជាខ្នាតអន្តរជាតិ ដោយ ISO (International Standard Organization) ក្នុងឆ្នាំ ១៩៨៧។

SQL ត្រូវបានយ៉ាងទូលំទូលាយធ្វើជា Query Language របស់ DBMS ក្នុងគោលបំណង មួយចំនួនដូចក្រោម៖

- បង្កើតនូវកិច្ចការគ្រប់គ្រងទិន្នន័យដូចជាការបន្ថែមការ ការកែប្រែ និងការលុបទិន្នន័យជាដើម។
- បង្កើតនូវការរុករកទិន្នន័យក្នុងទម្រង់ធម្មតា ឬស្មុគស្មាញ ដើម្បីបំប្លែងទិន្នន័យដើមឲ្យក្លាយទៅជាព័ត៌មានដែលគេចង់បាន។

នៅក្នុងមេរៀនយើងនិងសិក្សា SQL Statement មួយចំនួនដើម្បីរៀបចំចាត់ចែងទិន្នន័យ ដូចជា៖

- Select Statement : មាននាទីសំរាប់រុករកទិន្នន័យនិងទាញយកទិន្នន័យមកបង្ហាញចេញពី Database មួយតាមតម្រូវការរបស់យើង។
- Insert Statement មាននាទីសំរាប់បន្ថែម Record ថ្មីចូលទៅកាន់ Table មួយ។
- Update Statement មាននាទីសំរាប់កែប្រែតំលៃរបស់ Field មួយ ឬច្រើនដែលស្ថិតនៅក្នុង Table មួយ។
- Delete Statement មាននាទីសំរាប់លុប Record មួយ ឬច្រើនចេញពី Table មួយ។

2- Select Statement

គឺជា Statement ដែលមាននាទីសំរាប់រុករក និងទាញយកទិន្នន័យចេញពី Table មួយឬច្រើនដែលស្ថិតក្នុង Database មួយមកបង្ហាញតាមតម្រូវការរបស់យើង។

Select Statement មានទ្រង់ទ្រាយដូចខាងក្រោម៖

Select Predicate {

- * (សំរាប់បង្ហាញគ្រប់ Fields ទាំងអស់)
- TableName.* (សំរាប់បង្ហាញគ្រប់ Fields ទាំងអស់ ក្នុង Table មួយ)
- FieldName1, FieldName2,, FieldNameN
- TableName.FieldName(ចំពោះ Field ដែលមានឈ្មោះដូចគ្នា)
- FieldExpression as NewFieldName

From {

- TableName/QueryName
- Join Table Expression

Where General Condition

Group By FieldName1,FieldName2,.....,FieldNameN Having GroupCondition

Order By FieldName {

- ASC
- DESC

- Select : គឺជា Keyword ដែលមានលក្ខណៈដូចទៅនឹងប្រមាណវិធីចំណោល សំរាប់កំណត់យកឈ្មោះរបស់ Fields ទាំងឡាយដែលយើងចង់បង្ហាញ។
- FROM : គឺជា Keyword ដែលមាននាទីសំរាប់កំណត់ឈ្មោះរបស់ Table/Query មួយ ឬក៏កន្សោមនៃការតភ្ជាប់រវាង Table ចាប់ពី ២ ឡើងវិញទៅដែលបញ្ជាក់អំពីប្រភពរបស់ Fields ទាំងឡាយដែលយើងចង់បង្ហាញ ឬប្រើប្រាស់នៅក្នុងកន្សោមលក្ខខណ្ឌ។
- Where : គឺជា Keyword ដែលមាននាទីសំរាប់កំណត់លក្ខខណ្ឌទូទៅ (General Condition) ដើម្បីរកទិន្នន័យទៅតាមលក្ខខណ្ឌដែលយើងបានកំណត់។
- Group By : គឺជា Keyword ដែលមាននាទីសំរាប់ប្រមូលផ្តុំជាក្រុមនូវតំលៃ ឬ Record ដូចគ្នារបស់ Field ដែលយើងបានកំណត់ ដើម្បីរាប់ចំនួនសរុប គណនាផលបូកសរុប រកមធ្យមភាគ រកតំលៃតូចបំផុត ឬធំបំផុតនៃ Field ផ្សេងទៀតទៅតាមក្រុមនៃតំលៃ ឬ Record ដូចគ្នាទាំងនោះ។

- Having : គឺជា Keyword ដែលមាននាទីសំរាប់កំណត់លក្ខខណ្ឌ ដើម្បីរុករកទិន្នន័យទៅតាមក្រុមនៃតំលៃ ឬ Record ដូចគ្នារបស់ Field ដែលយើងបានកំណត់ នៅក្នុង Group By ។
- Order By : គឺជា Keyword ដែលមាននាទីសំរាប់តំរៀប Record ទៅតាមតំលៃរបស់ Field ដែលយើងបានកំណត់លំដាប់កើន (Ascending) ប្រសិនបើយើងមិនដាក់ ឬដាក់ ASC នៅពីក្រោយឈ្មោះរបស់ Field ។ ម្យ៉ាងវិញទៀត ដើម្បីតំរៀប Record តាមលំដាប់ចុះ (Descending) ទៅតាមតំលៃរបស់ Field ដែលយើងបានកំណត់ប្រសិនបើយើងបានដាក់ DESC នៅពីក្រោមឈ្មោះរបស់ Field នោះ។

ជាទូទៅ Select Statement មាន១២ទម្រង់ដូចខាងក្រោម:

- I- Select From.....;
- II- Select From where
- III- Select From where Group By.....;
- IV- SelectFrom Where Group By.....Having.....;
- V- SelectFromWhere..... Group By Having..... Order By.....;
- VI- SelectFromWhere..... Group By Order By.....;
- VII- SelectFromWhere..... Order By.....;
- VIII- SelectFromGroup By ;
- IX- SelectFromGroup ByHaving.....;
- X- SelectFromGroup ByHaving.....Order By.....;
- XI- SelectFromGroup ByOrder By.....;
- XII- SelectFromOrder By.....

- ចំណាំ: គ្រប់ទម្រង់ទាំងអស់របស់ Select Statement ត្រូវបានដំណើរការចាប់ពី FROM ទៅផ្នែក ខាងស្តាំរហូតទាល់តែអស់ទើបដំណើរការ Select ជាចុងក្រោយ។

2.1 Predicate:

គឺជាអំនះអំនាងមួយចំនួនដែលយើងបានប្រើដើម្បីកំណត់យក Record ទៅតាម Predicate នីមួយៗដូចខាងក្រោម:

- All : មាននាទីសំរាប់បង្ហាញគ្រប់ Record ទាំងអស់។ Predicate នេះយើងពុំចាំបាច់ដាក់វាក៏បាន។
- Distinct : មាននាទីសំរាប់បង្ហាញតែម្តងគត់ចំពោះតំលៃ ឬ Record ណាដែលដូចគ្នា។
- Top n : មាននាទីសំរាប់បង្ហាញពីលើចុះក្រោមចំនួន n Records។
- Top n percent : មាននាទីសំរាប់បង្ហាញ Record ពីលើចុះក្រោមចំនួន n% នៃ Records ទាំងអស់។

Example : តើមាន Table ដូចខាងក្រោម:

Customer (<u>CustID,</u>	CustName,	Gender,	Address	Phone)
	001	Chan Dara	M	#12E0	012 333 444
	002	Keo Maly	F	#15E1	023 445 667
	003	Khim Thida	F	#128E2	011 834 567
	004	Long Darith	M	#190E0	099 458 667
	005	Yun Sopha	F	#230E1	080 334 567

ប្រសិនបើយើងសរសេរ SQL Statement ដូចខាងក្រោម នោះវានឹងបង្ហាញលទ្ធផលទៅតាម Statement នីមួយៗ៖

- Select All Gender from Customer; ឬ Select Gender from Customer;
វានឹងបង្ហាញលទ្ធផលដូចខាងក្រោម:

(Gender)

M
F
F
M
F

- Select Distinct Gender from Customer;

វានឹងបង្ហាញលទ្ធផលដូចខាងក្រោម:

(Gender)

M
F

- Select Top 2 * from Customer;

Customer (<u>CustID,</u>	CustName,	Gender,	Address	Phone)
	001	Chan Dara	M	#12E0	012 333 444
	002	Keo Maly	F	#15E1	023 445 667

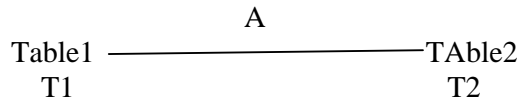
- Select Top 20 percent * from Customer;

Customer (<u>CustID,</u>	CustName,	Gender,	Address	Phone)
	001	Chan Dara	M	#12E0	012 333 444

2.2 កន្សោមនៃការភ្ជាប់រវាង Table (Join Table Expression):

កន្សោមនៃការភ្ជាប់រវាង Table មានដូចខាងក្រោម៖

A- ការភ្ជាប់រវាង 2 Tables:



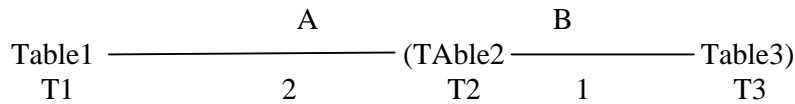
+ តាមរបៀបទី១

From Table T1 Inner Join Table2 T2 On T1.A=T2.A

+ តាមរបៀបទី២

From Table1 T1, Table2 T2 where T1.A=T2.A And Gender Condition

B- ការភ្ជាប់រវាង ៣ Tables:



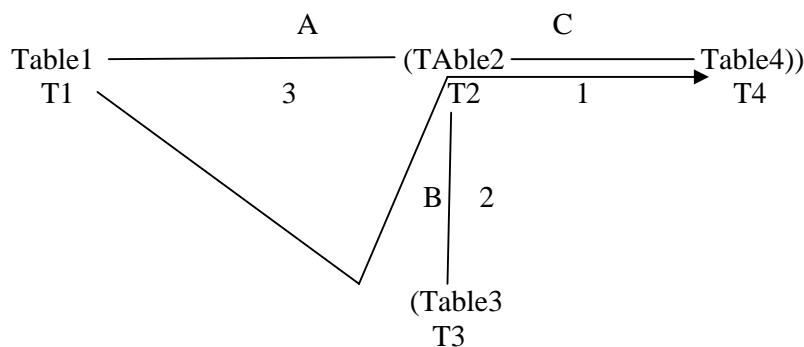
+ តាមរបៀបទី១

From Table T1 Inner Join (Table2 T2 Inner Join Table3 T3 On T2.B=T3.B) On T1.A=T2.A

+ តាមរបៀបទី២

From Table1 T1, Table2 T2, Table3 T3 where T1.A=T2.A And T2.B=T3.B And Gender Condition

C- ការភ្ជាប់រវាង 4 Tables



+ តាមរបៀបទី១

From Table T1 Inner Join (Table3 T2 Inner Join (Table2 T2 Inner Join Table4 T4 On T2.C=T4.C) On T2.B= T3.B) On T1.A=T2.A

+ តាមរបៀបទី២

From Table1 T1, Table2 T2, Table3 T3, Table4 T4 where T1.A=T2.A and T2.B=T3.B and T2.C=T4.C and Gender Condition

2.3- កន្សោមលក្ខខណ្ឌទូទៅ (General Condition):

នៅក្នុងកន្សោមលក្ខខណ្ឌ យើងអាចប្រមាណវិធីមួយចំនួនដូចជា <, <=, >, >=, =, <>, Between ...And....., Like, In, Not In, Is Null និងIs Not Null ជាដើម។

ប្រសិនបើ Field មានប្រភេទទិន្នន័យជា Text នោះយើងត្រូវដាក់ Single Quote (‘) នៅសងខាងតំលៃរបស់វា ហើយបើ Field មានប្រភេទទិន្នន័យជា Date/Time នោះយើងត្រូវដាក់ Address Sign (#) នៅសងខាងតំលៃរបស់វា។

Example: គេមាន Tables មួយចំនួនដូចខាងក្រោម:

TbStudent(StuID, Name, Gender, BirthDate, Address, Phone, ClassID)

TbClass(ClassID, ClassName, Year, Department)

TbSubject(SubjectID, SubjectName)

TbScore(StuID, SubjectID, Semester, Score)

តាមរយៈ Tables ទាំងអស់ខាងលើ ចូរសរសេរ SQL (Select Statement) តាមគោលបំណងដូចខាងក្រោម

ក- បង្ហាញ StuID, Name, Gender នៃនិស្សិតទាំងឡាយដែលមានឈ្មោះផ្ដើមដោយអក្សរ C។

```
Select StuID, Name, Gender From TbStudent Where Name Like 'C*';
```

ខ- បង្ហាញ StuID, Name, Gender, BirthDate, នៃនិស្សិតទាំងឡាយដែលមានថ្ងៃខែឆ្នាំកំណើតនៅចន្លោះថ្ងៃទី 5/Jan/1985 ដល់ 10/Feb/1987។

```
Select StuID, Name, Gender, BirthDate from TbStudent
Where BirthDate Between #5/Jan/1985# and #10/Feb/1987#;
```

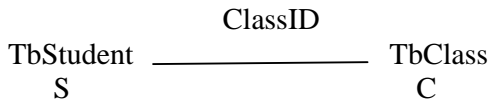
គ- បង្ហាញព័ត៌មានរបស់និស្សិតទាំងឡាយដែលគ្មានលេខទូរស័ព្ទ។

```
Select * from TbStudent Where Phone Is Null;
```

ឃ- បង្ហាញព័ត៌មានរបស់និស្សិតទាំងឡាយដែលកើតក្នុងឆ្នាំ 1985 ឬ 1987។

```
Select * from TbStudent where Year(BirthDate) In (1985, 1987);
```

ង- បង្ហាញ StuID, Name, ClassName, Year នៃនិស្សិតទាំងឡាយដែលរៀននៅក្នុង ថ្នាក់ M1 នៃឆ្នាំទី២។



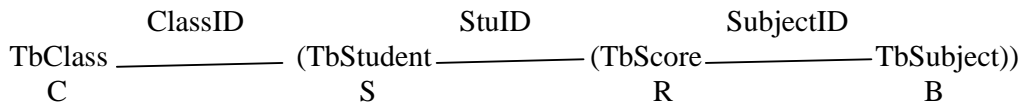
របៀបទី១៖

```
Select StuID, Name, ClassName, Year From TbStudent S Inner Join TbClass C On S.ClassID=C.ClassID Where ClassName='M1' and Year=2;
```

របៀបទី២៖

```
Select StuID, Name, ClassName, Year From TbStudent S, TbClass C Where S.ClassID=C.ClassID And ClassName='M1' and Year=2;
```

ច- ចូរបង្ហាញ StuID, Name, ClassName, Year, Semester, Subject, Score នៃនិស្សិតទាំងឡាយដែលរៀននៅក្នុងឆ្នាំទី ២ ធ្លាក់មុខវិជ្ជា DBMS ឬ MIC ក្នុងឆមាសទី១



របៀបទី១៖

```

SELECT S.StuID, Name, ClassName, Year, Semester, SubjectName, Score
FROM TbClass AS C INNER JOIN (TbStudent AS S INNER JOIN (TbScore AS R INNER
JOIN TbSubject AS B ON R.SubjectID=B.SubjectID) ON S.StuID=R.StuID) ON
C.ClassID=S.ClassID WHERE Year=2 and SubjectName In ('DBMS','MIC') And Score<50
and Semester =1;
  
```

របៀបទី២៖

```

Select S.StuID, Name, ClassName, Year, Semester, SubjectName, Score
From TbClass C, TbStudent S, TbScore R, TbSubject B Where
R.SubjectID=B.SubjectID And S.StuID=R.StuID And C.ClassID=S.ClassID
And Year=2 and SubjectName In ('DBMS','MIC') And Score<50 and Semester =1;
  
```

2.4- Aggregate Function

គឺជាអនុគមន៍ចំនួន ៥ ដែលយើងប្រើក្នុង Select Statement៖

- SUM : មាននាទីសំរាប់គណនាផលបូកសរុបនៃតំលៃ Field មួយឬក៏ស្រុះប្រមូលទៅតាមក្រុមនៃតំលៃដូចគ្នារបស់ Field ដែលយើងបានកំណត់ក្នុង Group By ។
- AVG: មានតំលៃសំរាប់រកមធ្យមនៃតំលៃរបស់ Field មួយ ឬក៏ស្រុះប្រមូលទៅតាមក្រុមនៃតំលៃដូចគ្នារបស់ Field ដែលយើងបានកំណត់ក្នុង Group By ។
- COUNT : មាននាទីសំរាប់រាប់ចំនួនសរុបនៃតំលៃរបស់ Field មួយទៅតាមក្រុមនៃតំលៃដូចគ្នារបស់ Field ដែលយើងបានកំណត់ក្នុង Group By ។
- MIN : មាននាទីសំរាប់រកចំនួនតូចបំផុតនៃតំលៃរបស់ Field មួយ ឬក៏ស្រុះប្រមូលទៅតាមក្រុមនៃតំលៃដូចគ្នារបស់ Field ដែលយើងបានកំណត់ក្នុង Group By ។
- MAX : មាននាទីសំរាប់រកចំនួនធំបំផុតនៃតំលៃរបស់ Field មួយ ឬក៏ស្រុះប្រមូលទៅតាមក្រុមនៃតំលៃដូចគ្នារបស់ Field ដែលយើងបានកំណត់ក្នុង Group By ។

- យើងត្រូវកំណត់ក្នុង Group By ចំពោះ Field ទាំងឡាយដែលយើងបាន Select បន្ថែមក្រៅពី Aggregate Function ។ ម្យ៉ាងទៀតប្រសិនបើនៅក្នុងកន្សោមលក្ខខណ្ឌបានប្រើ Aggregate Function នោះយើងត្រូវកំណត់កន្សោមលក្ខខណ្ឌនេះនៅក្នុង Having ។

Example01: គេមានតារាងដូចខាងក្រោម៖

SaleDetail (<u>SaleID,</u>	<u>PID,</u>	SaleQty,	UnitPrice,	Discount)
	001	01	20	\$5.00	0.00%
	001	02	10	\$10.00	0.00%
	002	01	10	\$5.00	0.00%
	002	02	10	\$10.00	0.00%
	002	04	20	\$5.00	0.00%
	003	01	10	\$5.00	0.00%
	003	02	10	\$10.00	0.00%

ប្រសិនបើយើងសរសេរ SQL Statement ដូចខាងក្រោម៖

```
Select PID, Sum(SaleQty) As TotalQty, Count(*) As [Count of Sales], AVG(SaleQty) As Average, Min(SaleQty) as Minimum, Max(SaleQty) as Maximum From SaleDetail Group By PID;
```

នោះលទ្ធផលនឹងបង្ហាញដូចខាងក្រោម៖

(PID,	TotalQty,	Count of Sales,	Average,	Minimum,	Maximum)
01	40	3	13.33	10	20
02	30	3	10.00	10	10
04	20	1	20.00	20	20

Example 02: គេមានតារាងដូចខាងក្រោម៖

TbStudent (StuID, Name, Gender, BirthDate, Address, Phone, ClassID)
 Number Text Text Date/Time Text Text Number

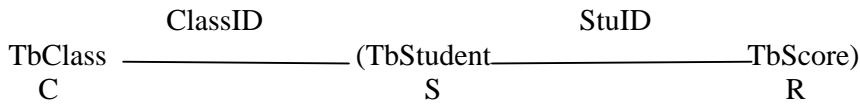
TbClass(ClassID, ClassName, Year, Department)
 Number Text Number Text

TbSubject(SubjectID, SubjectName)
 Number Text

TbScore (StuID, SubjectID, Semester, Score)
 Number Number Number Number

តាមរយៈ Tables ទាំងអស់ខាងលើ ចូរសរសេរ SQL (Select Statement) តាមគោបំណងដូចខាងក្រោម៖

ក- បង្ហាញ StuID, Name, ពិន្ទុសរុប, និងមធ្យមភាគ ទៅតាមនិស្សិតម្នាក់ៗ ដែលរៀននៅ ក្នុងឆ្នាំទី ២ មានមធ្យមភាគប្រចាំឆមាសទី១ តិចជាង៥០ ឬច្រើនជាង៧៥ ដោយតំរៀបតាម លំដាប់ចុះនៃមធ្យមភាគនេះ។



របៀបទី១៖

```

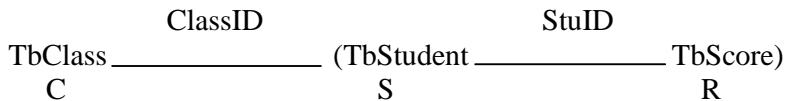
Select S.StuID, Name, Sum(Score) As [Total Score], AVG(Score) as Average From TbClass
C Inner Join (TbStudent S Inner Join TbScore R On S.StuID=R.StuID) On
S.ClassID=C.ClassID Where Year = 2 And Semester = 1 Group By S.StuID, Name, Having
Avg(Score)<50 Or Avg(Score)>75 Order By Avg(Score) Desc;
    
```

របៀបទី២៖

```

Select S.StuID, Name, Sum(Score) As [TotalScore], AVG(Score) As Average From
TbClass C, TbStudent S, TbScore R Where S.StuID=R.StuID And S.ClassID = C.ClassID
And Year=2 And Semester =1 Group By S.StuID, Name Having Avg(Score)<50 Or
Avg(score)>75 Order By Avg(Score) DESC;
    
```

ខ- បង្ហាញ StuID, Name, ពិន្ទុសរុបទៅតាមនិស្សិតម្នាក់ៗដោយកំណត់យកនិស្សិតចំនួន៥នាក់ដែលរៀន នៅក្នុងឆ្នាំទី៤ មានពិន្ទុសរុបប្រចាំឆមាសទី២ មានពិន្ទុសរុបខ្ពស់ជាងគេដោយមានមធ្យមភាគចាប់ពី ៨០ ឡើងទៅ។



របៀបទី១៖

```
Select Top 5 S.StuID, Name, Sum(Score) As [TotalScore] From TbClass C Inner Join  
(TbStudent S Inner Join TbScore R On S.StuID = R.StuID) On S.ClassID=C.ClassID  
Where Year=4 And Semester =2 Group By S.StuID, Name Having Avg(Score)>=80  
Order By Sum(Score) DESC;
```

របៀបទី២៖

```
Select Top 5 S.StuID, Name, Sum(Score) As [TotalScore] From TbClass C, TbStudent S,  
TbScore R Where S.StuID = R.StuID And S.ClassID=C.ClassID and Year=4 And Semester =2  
Group By S.StuID, Name Having Avg(Score)>=80 Order By Sum(Score) DESC;
```

2.5 Sub-Query:

គឺជា Select Statement ដែលត្រូវបានប្រើក្នុង Select Statement មួយផ្សេងទៀត ឬក៏ក្នុង Insert, Update និង Delete Statement ។ គេប្រើ Sub-Query នៅពេលដែលគេចង់ធ្វើសកម្មភាពក្នុង Table មួយដែលលក្ខខណ្ឌរបស់វាទាក់ទងនឹង Field នៃ Table មួយឬច្រើនផ្សេងទៀត។

- ចំណាំ: នៅពេលដែលយើងប្រើ Sub-Query នោះយើងត្រូវចាប់ផ្តើមចេញពី Table ណាដែលយើងចង់ធ្វើសកម្មភាព។

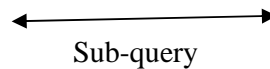
Example 01: គេមានតារាងដូចខាងក្រោម:

Product (PID,	PName,	StockQty)
	01	A	200
	02	B	100
	03	C	100
	04	D	150
	05	E	200
	06	F	100
	07	G	200

SaleDetail (SaleID,	PID,	SaleQty,	UnitPrice,	Discount)
	001	01	20	\$5.00	0.00%
	001	02	10	\$10.00	0.00%
	002	01	10	\$5.00	0.00%
	002	02	10	\$10.00	0.00%
	002	04	20	\$5.00	0.00%
	003	01	10	\$5.00	0.00%
	003	02	10	\$10.00	0.00%

ចូរសរសេរ SQL Statement ដោយប្រើ Sub-query ដើម្បីបង្ហាញនូវផលិតផលទាំងឡាយ ដែលគេ ពុំបានលក់ម្តងទាល់តែសោះ។

Select * From Product Where PID Not In (Select PID From Sale);



នោះលទ្ធផលនឹងបង្ហាញដូចខាងក្រោម:

Product (PID,	PName,	StockQty)
	03	C	100
	05	E	200
	06	F	100
	07	G	200

Example 02: គេមានតារាងដូចខាងក្រោម:

TbStudent (StuID,	Name,	Gender,	BirthDate,	Phone,	ClassID)
	Number	Text	Text	Date/Time	Text	Number

TbClass (ClassID,	ClassName,	Year,	Department)
	Number	Text	Number	Text

TbSubject (SubjectID,	SubjectName)
	Number	Text

TbScore (StuID, SubjectID, Semester , Score)
Number Number Number Number

តាមរយៈ Tables ទាំងអស់ខាងលើ ចូរសរសេរ SQL (Select statement)

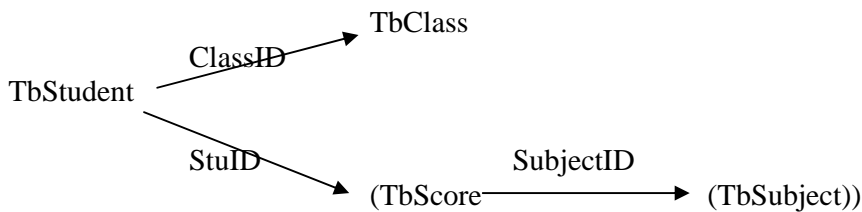
តាមគោលបំណងខាងក្រោម:

ក- បង្ហាញព័ត៌មានរបស់និស្សិតទាំងឡាយដែលរៀននៅឆ្នាំទី២។

TbStudent _____(TbClass)

```
Select * From TbStudent Where ClassID In (Select ClassID From TbClass Where Year=2)
```

ខ- បង្ហាញព័ត៌មានរបស់និស្សិតទាំងឡាយ ដែលរៀនទី៤ ដែលប្រឡងធ្លាក់មុខវិជ្ជា Linux ។



```
Select * From TbStudent Where ClassID In ( Select ClassID From TbClass Where Year=4)
And StuID In (Select Distinct StuID From TbScore Where Score<50 and SubjectID in (select SubjectId From TbSubject where Subject = 'linux'));
```

3- Insert Statement:

គឺជា Statement ដែលមាននាទីសំរាប់ចំលងទៅផ្ទុកក្នុង Field តាមលំដាប់នៃ Table មួយឬក៏ ចំលង Record ពី Record មួយទៅផ្ទុកក្នុង Table មួយផ្សេងទៀត។

Syntax1: ចំលងតំលៃទៅផ្ទុកក្នុង Field ចំនួន n តាមលំដាប់នៃ Table មួយ។

```
Insert Into TableName (FieldName1, FieldName2, ..., FieldNameN)
Values (Value1, Value2, ....., ValueN);
```

ឬម្យ៉ាងទៀតគឺ:

```
Insert Into TableName Values( value1, value2, ....., valueN);
```

ប្រសិនបើ Field មានប្រភេទទិន្នន័យជា Text នោះយើងត្រូវដាក់ Single quote (‘) នៅសងខាង Value ដែលត្រូវផ្ទុកក្នុង Field នោះ។ ម្យ៉ាងវិញទៀត ប្រសិនបើ Field មានប្រភេទទិន្នន័យ ជា Date/Time នោះយើងត្រូវដាក់ Address (#) នៅសងខាង Value ដែលត្រូវផ្ទុកក្នុង Field។

Example: គេមានតារាងដូចខាងក្រោម:

Employee	(EmpID,	Name,	Gender,	BirthDate,	Address,	Phone,	Position,	Salary)
	Number	Text	Text	Date/Time	Text	Text	Text	Currency

ដើម្បីចំលងតំលៃដូចជា 1 , Chan Dara, M, 12/Jan/84, #102E0, (012) 344-315, Manager, \$1200 ទៅផ្ទុកក្នុង Field តាមលំដាប់នៃ Table ខាងលើ នោះយើងត្រូវសរសេរ SQL Statement ដូចខាងក្រោម:

```
Insert Into Employee Values (1,'Chan Dara','M', #12/Jan/84#, '120E0','(012) 344-315','Manager', 1200);
```

Syntax2:

```
Insert Into TableDestination (FieldName1, FieldName2, .....,FieldNameN) Select FieldName1, FieldName2,.....,FieldNameN From TableSource Where Condition;
```

ឬម្យ៉ាងទៀតគឺ:

```
Insert Into TableDestination Select * from TableSource Where Condition;
```

Example: គេមានតារាងដូចខាងក្រោម:

Sale (<u>SaleID</u> ,	SaleDate,	CusID,	EmpID)
Number	Date/Time	Number	Number
BackUpSale (<u>SaleID</u> ,	SaleDate,	CusID,	EmpID)
Number	Date/Time	Number	Number
SaleDetail(<u>SaleID</u> , <u>PID</u> ,	SaleQty,	UnitPrice,	Discount)
Number	Number	Currency	Number
BackUpSaleDetail(<u>SaleID</u> , <u>PID</u> ,	SaleQty,	UnitPrice,	Discount)
Number	Number	Currency	Number

ដើម្បីចំលង Record ពី Table ឈ្មោះ SaleDetail និង Sale ចំពោះការលក់ផលិតផលទាំងឡាយមុនឆ្នាំ ២០០៦ ទៅផ្ទុករៀងគ្នាក្នុង Table ឈ្មោះ BackUp-SaleDetail និង BackupSale នោះយើងត្រូវសរសេរ SQL Statement ដូចខាងក្រោម។

- ចំលង Records ពី Sale ទៅផ្ទុកក្នុង BackUpSale :

```
Insert Into BackUpSale Select * From Sale Where Year(SaleDate)<2006;
```

- ចំលង Records ពី SaleDetail ទៅផ្ទុកក្នុង BackUpSaleDetail:

```
Insert Into BackUpSaleDetail Select * from SaleDetail where SaleID In ( Select SaleID From Sale Where Year(SaleDate)<2006)
```

db.Execute “Insert Into TableDestination Select * from TableSource Where Condition”

4- Delete Statement:

គឺជា Statement ដែលត្រូវមាននាទីសំរាប់លុប Records ចេញពី Table ណាមួយ។

Syntax:

```
Delete From TableName Where Condition;
```

Example: ដើម្បីលុប Records ចេញពី SaleDetail និង Sale ចំពោះការលក់ផលិតផលទាំងឡាយមុនថ្ងៃទី 1/Feb/2006 នោះយើងត្រូវសរសេរ SQL Statement ដូចខាងក្រោម:

- លុប Records ចេញពី SaleDetail

```
Delete From SaleDetail Where SaleID In (select SaleID from Sale  
Where SaleDate<#1/Feb/2006#);
```

- លុប Records ចេញពី Sale:

```
Delete From Sale Where SaleDate<#1/Feb/2005#
```

5- Update Statement

គឺជា Statement ដែលមាននាទីសំរាប់កែតំលៃរបស់ Field មួយឬច្រើនដែលស្ថិតនៅក្នុង Table មួយ។

Syntax:

```
Update TableName Set FieldName1=NewValue1,....., FieldNameN=NewValueN
Where Condition;
```

Example: គេមានតារាងដូចខាងក្រោម:

Employee	(EmpID,	Name,	Gender,	BirthDate,	Address,	Phone,	Position,	Salary)
	Number	Text	Text	Date/Time	Text	Text	Text	Currency

ចូរសរសេរ SQL Statement ដើម្បីកែតួនាទី (Position) និងប្រាក់ខែរបស់ Employee ឈ្មោះ Keo Thida ឲ្យមានតួនាទីជា Supervisor និង បន្ថែម \$100 ទៅលើប្រាក់ខែចាស់:

```
Update Employee Set Position='Supervisor', Salary=Salary+100
Where Name = 'Keo Thida'
```

- ចំណាំ:

ប្រសិនបើយើងសរសេរនៅក្នុង Program ត្រង់ព្រឹត្តិការណ៍មួយ នោះយើងត្រូវប្រើតាម Syntax ដូចខាងក្រោម:

Syntax:

```
Dim db as DAO.Database
set db=CurrentDB() ឬ set db=OpenDatabase("Path)
db.Execute "Update TableName set FieldName1=NewValue1,....., FieldNameN=NewValueN
Where Condition";
```