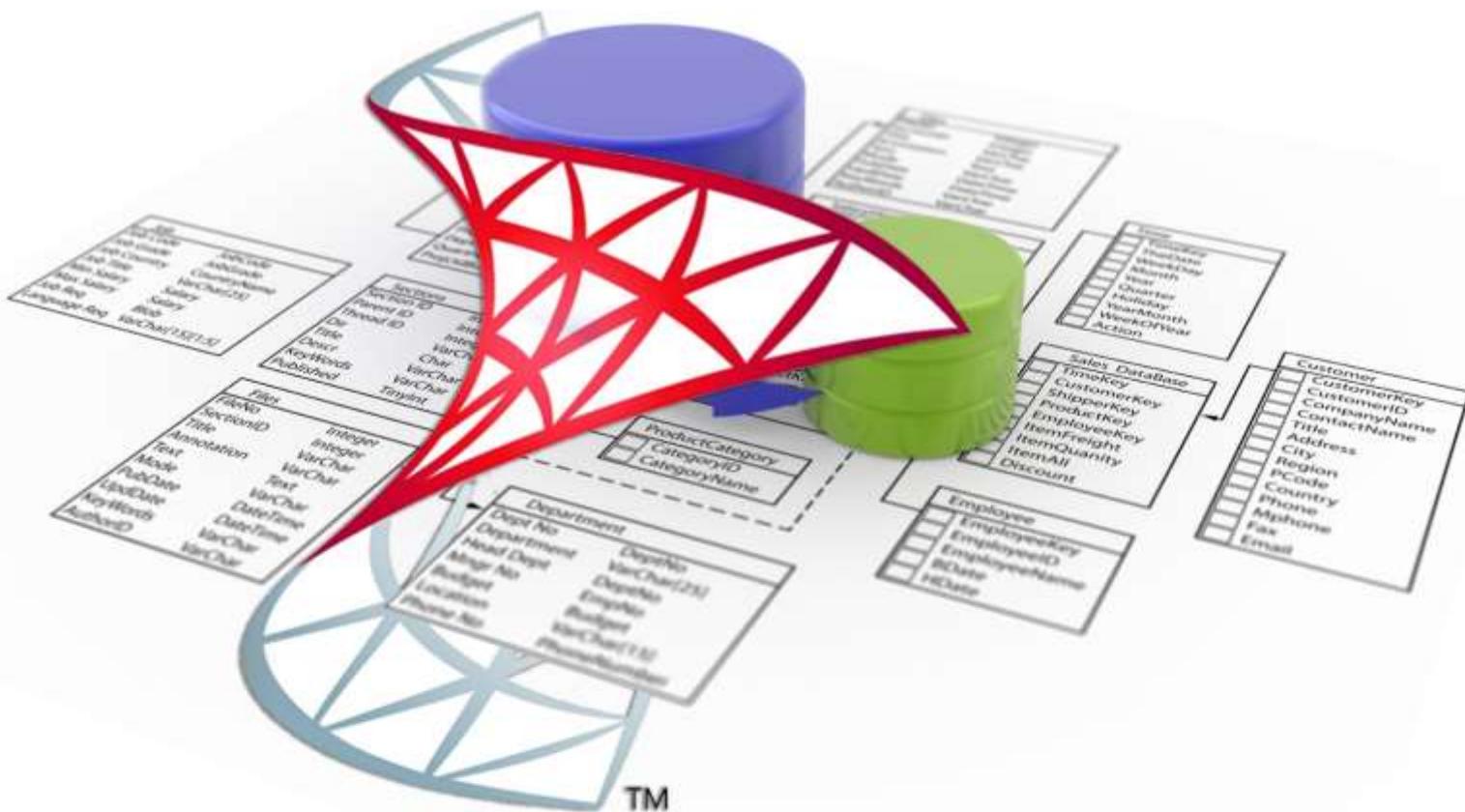


# Enter<sup>it</sup>

មន្ទីរបច្ចេកទេសព័ត៌មាននិង  
www.enteritc.com

## Microsoft® SQL Server® 2008



រៀបរៀងដោយ ហុ មុនី  
Prepared by: Ho Mony

ផ្ទះលេខ 173 ផ្លូវ 179 ភៀង 424 សង្កាត់ទំនប់ទីក ខណ្ឌចំការមន ភ្នំពេញ  
Tel: 010/012 603 314 | www.enteritc.com | facebook.com/enteritc

ଶ୍ରୀମତ୍ ପଦମାତ୍ରା

សញ្ញថ្លែងនេះ ការប្រើប្រាស់កំពុង គឺជាមធ្យាបាយយោងសំខាន់អ្នយក្តីការដោះស្រាយបញ្ហា និងដួយសំនួលដល់ការងារប្រចាំថ្ងៃបស់មនុស្សដើរសំនួលបាន ភាពធ្វើនការងារ ត្រូវតែម៉ាងនសំបុរាណតុកគឺតិចផ្សេងៗទៀត ហើយក្នុងការប្រើប្រាស់កំពុងទាំងម្រប ដែលជាបាតុមានមុខងារឡើងក្នុងការប្រើប្រាស់កំពុង ត្រូវមានសមត្ថភាព និងចំនះដឹងត្រប់ត្រាន់ក្នុងការប្រើប្រាស់ទៅបើរាយ

ដើម្បីចូលរួមបើកស្តូបវិស័យពីមានវិញ្ញា នៅក្នុងព្រះរាជាណាចក្រកម្ពុជា មជ្ឈមណ្ឌលនឹងធ្វើ ពីមានវិញ្ញា ត្រូវបានបង្កើតឡើងដើម្បីនឹងដើរ ក្នុងគោលចំនួន ដើម្បីផ្តល់បណ្តាលចំនួនដើម្បីផ្តល់កិច្ចការនិងវិញ្ញាសាល់ សិស្ស និស្សិត និងអ្នកសិក្សា ទេនានចំនួនដើម្បីព្រាកចិត្តអង់គ្លេសនៃ ជាតិសេសមានសមត្ថភាពថ្មាស់លោកស៊ីអ៊ីប្រាស់កំពុទ្ធនា គឺចនេះស្មើ សិស្ស និស្សិត និងអ្នកសិក្សាចារិក្សាចាំងអស់ ឱតខំប្រើប្រាស់ សិក្សា ត្រូវរារាំង និងចាប់យកនូវចំនួនដើម្បី ដើល មជ្ឈមណ្ឌល បានបណ្តុះបណ្តាលបន្ទុន ដើម្បីពេលរែលបានដិលអ្នកបានចំនាយ ត្រូវបានបន្ទាយដោយសារធម្មប្រយោជន៍ដែល សំរាប់ជីវិតរបស់អ្នកនាពេលនោតតា

# ក្រុមអនុករណ៍បង្កេដនៃ មន្ទីរណាបន្ទិនធីជំពូក ព័ត៌មានវិទ្យា

# ព្រះរាជាណាចក្រកម្ពុជា

សាស្ត្រព្រៃន:

ប្រធានាជន

សាស្ត្រពេលាភ័ណ៌:

ប្រធានាជន

សាស្ត្រពេលាភ័ណ៌ និងសាស្ត្រពេលាភ័ណ៌:

ប្រធានាជន

ចំណែកអាជីវកម្ម

ចំណែកអាជីវកម្មនៃការបង្កើតរបស់ខ្លួន:

ប្រធានាជន

ចំណែកអាជីវកម្ម

គោរៈ សោរ៉ាណា

ស្រុកស្រុកដែលមានភាពខ្ពស់:

រក្សាសិទ្ធិគ្រប់ប៉ាង © 2013 ដោយ មជ្ឈមណ្ឌលអនិនីស ព័ត៌មាននិងផែនការ សំរាប់ជាតិ ធនធានប្រើប្រាស់ផ្លូវការ។  
ត្រូវដោលស្នើសុំការបង្កើតរបស់ខ្លួន និងប្រើប្រាស់ផ្លូវការ ដើម្បីជួយបានការងារ និងការបង្កើតរបស់ខ្លួន។  
និងការបង្កើតរបស់ខ្លួន និងការងារ និងការបង្កើតរបស់ខ្លួន។

អាជីវកម្ម:

ផ្លូវលេខ 179 ផ្លូវ 173 សង្កាត់ ទំនប់ទីក ឧណ្ណាចំការមន រាជធានីភ្នំពេញ

Tel: [010-012] 603 314

Website: [www.enteritc.com](#) | [facebook.com/enteritc](#)

# ចំណាំ

មេរ្តៃនទិ 1: និយមន៍បយ Microsoft SQL Server 2008 .....	1
មេរ្តៃនទិ 2: រច្ឆច Connect SQL Server 2008 .....	11
មេរ្តៃនទិ 3: សិក្សាតី Normalization និង Relationship .....	19
មេរ្តៃនទិ 4: ការបង្កើត Database និង Table .....	25
មេរ្តៃនទិ 5: សិក្សាតី Constraints និង Database Diagrams .....	39
មេរ្តៃនទិ 6: Select ទិន្នន័យធាយប្រើប្រាស់ Queries .....	53
មេរ្តៃនទិ 7: សិក្សាលំនិតិ T-SQL គ្នា Queries .....	71
មេរ្តៃនទិ 8: សិក្សាតី Modifying Data .....	89
មេរ្តៃនទិ 9: សិក្សាតី T-SQL Programming .....	99
មេរ្តៃនទិ 10: សិក្សាតី T-SQL String and Date .....	107
មេរ្តៃនទិ 11: សិក្សាតី Controlling Flow .....	119
មេរ្តៃនទិ 12: សិក្សាតី Stored Procedures .....	131
មេរ្តៃនទិ 13: សិក្សាតី Triggers .....	138





# គេរើលទី 1: និយោបាយ Microsoft SQL Server 2008

## 1. ឯកចាល់

SQL Server គឺជា Relational Database Management System ដែលបង្កើតឡើងដោយ Microsoft ដែលវាគីជា Database Product មួយមានមុខងារយ៉ាងសំខាន់បំផុតក្នុងការក្រឹត្តិថ្លែងយោប័យកមកប្រើប្រាស់ ទៅតាមតម្លៃការដោយ Application ដើម្បីគ្រប់គ្រងព័ត៌មាន Connect ជាមួយវា (VB, C#, Java, Etc.) តាមរយៈ Local ឬ Network ។

SQL Server គឺជា Database Storage Technology មួយដែលបង្កើតឡើងដោយ SQL programming language ។ វាបំនើការងារផ្លូវការ Backend Database ដែលក្នុងនោះគ្រប់គ្រងព័ត៌មានយើង Frontend Application សំរាប់បង្ហាញពីមាន និងធ្វើការជាមួយគ្នា។

## 2. SQL Server History

Version	Year
SQL 4.2	ខែតុលា 1992
SQL 6.0	ខែ មិថុនា 1995
SQL 6.5	ខែមេស៊ា 1996
SQL 7.0	ខែមករា 1999 (New Architecture)
SQL 2000	ខែ សីហា 2000
SQL 2005	ខែ វិច្ឆិកា 2005
SQL 2008	ខែសីហា 2008

## 3. SQL Server Edition:

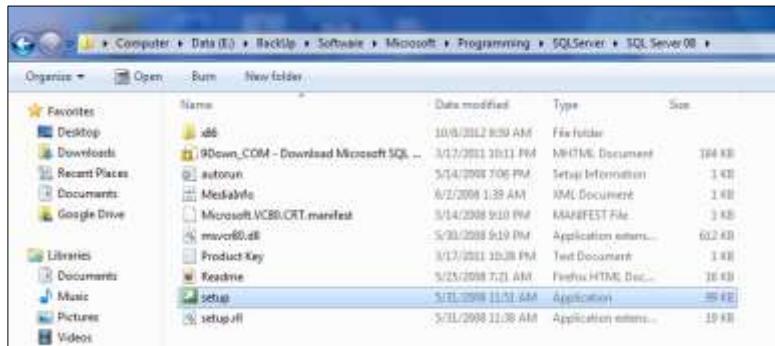
- > Datacenter : គឺជាប្រភពទៅ full-featured edition នៃ SQL Server ។
- > Enterprise : គឺជាប្រភពមួយដែលបញ្ចូលទាំង core database engine និង add-on services ដែលវានាទាំងគ្រប់គ្រងជាមួយនឹង ទិន្នន័យជំពូទ័រ ។
- > Standard : គឺជាប្រភពមួយដែលបញ្ចូលទាំងជាមួយនឹង stand-alone services ដែលរាយានលក្ខណៈខ្ពស់ពី Enterprise Edition ព្រមទាំង Support មុខងារពិច្ចុចណារៈ ។
- > Web : សំរាប់ប្រើប្រាស់ជាមួយនឹង Web hosting តែប៉ុណ្ណោះ ។
- > Express : គឺជាប្រភព Free Edition មួយសំរាប់ប្រើប្រាស់ផ្សេងៗ ឬសាកល្បងតែប៉ុណ្ណោះ ។

#### **4. Database Objects:**

- ជូចគ្គជាមួយនឹង Relational Database Management System ដើម្បីត្រួតពិនិត្យ SQL Server មាន Objects មួយចំនួនសំខាន់ៗ ដើម្បីធ្វើការជាមួយវាមាយបានល្អ និងមានប្រសិទ្ធភាព។ ក្នុងនោះ Database Object របស់ SQL Server មានជាដីជាតិ:
- > Tables : គឺជា Table សំរាប់ Store ទិន្នន័យផែលក្នុងនោះវាត្រូវបានកំណត់ និងបង្កើតជាយករាយការរូម បញ្ជាល់តាម columns, indexes, និង Constraints ។
  - > Database Diagrams : វាល្អបន្ទុវត្ថុ Graphic View នៃ សំនួរបស់ Tables និងទំនាក់ទំនងរវាងពួកវាដាមួយគ្នា ផែលផ្លូវក្រោះបើ Foreign Key Constraints ។ យើងនាថប្រើប្រាស់ Database diagram ដើម្បីធ្វើការកែត្រា Table បានដងដែរ។
  - > Views: វាត្រូវបានប្រើប្រាស់ដើម្បីរក្សាទុកនូវ Queries ដើម្បីបាន Select Data ចេញពី Table មួយ ឬប្រចើន។
  - > Stored Procedures : វាតីរក្សាទុកនូវ Queries នូវ View ដើម្បីបង្កើតកិរាណធ្វើការកែត្រា Table បានដែរ ហើយ Store procedures នាថមាន Parameters និង Procedural Code ដើមសរស់របជាមួយនឹង Transact-SQL (T-SQL) ដើម្បី .NET language ។
  - > Functions : SQL Servers គឺមាន built-in functions ជាផ្លូវការសំរាប់នោយយើង Call ហើយចេញពី T-SQL ហរកករប្រើប្រាស់ហើយនឹងកិរាណធ្វើការបង្កើត Function ដោយខ្លួនរួមបង្កើតបាន ដងដែរ ផែលមានលក្ខណៈជូចថែនឹង ការសាន់ត្រួតពិនិត្យ គឺមាន Parameters និង Return Data ។

#### **5. កែត្រា Install SQL Server 2008:**

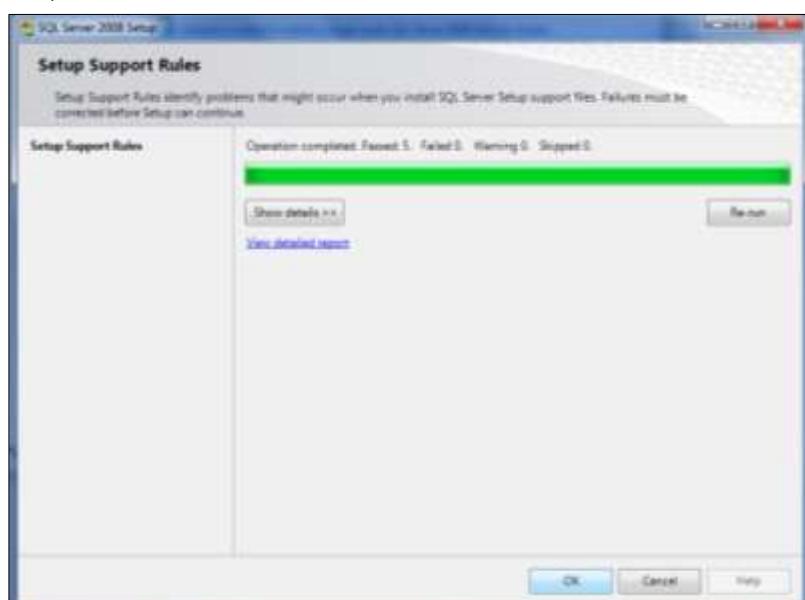
1. ស្មូមចូលឡើកនៃ Folder Setup របស់វាបើយចូច Double Click ឬ File Setup របស់វា >



2. ស្មូមចូច New SQL Server stand-alone installation or add features to an existing installation >

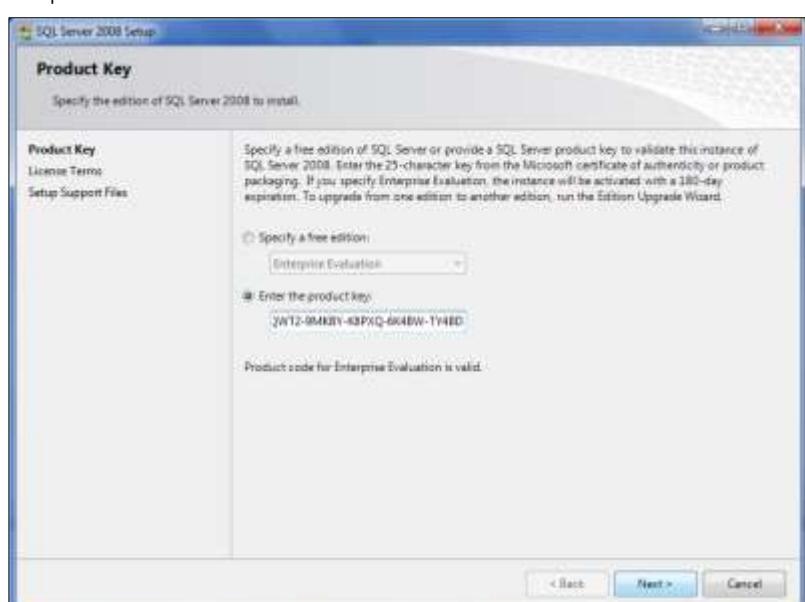


3. ចូល OK Button >



4. វិនិយោគ Enter the product Key ឬបើយសូមបញ្ជូន Product Key ផ្តល់ទៅក្នុងប្រអប់ >

5. ចូល Next Button >

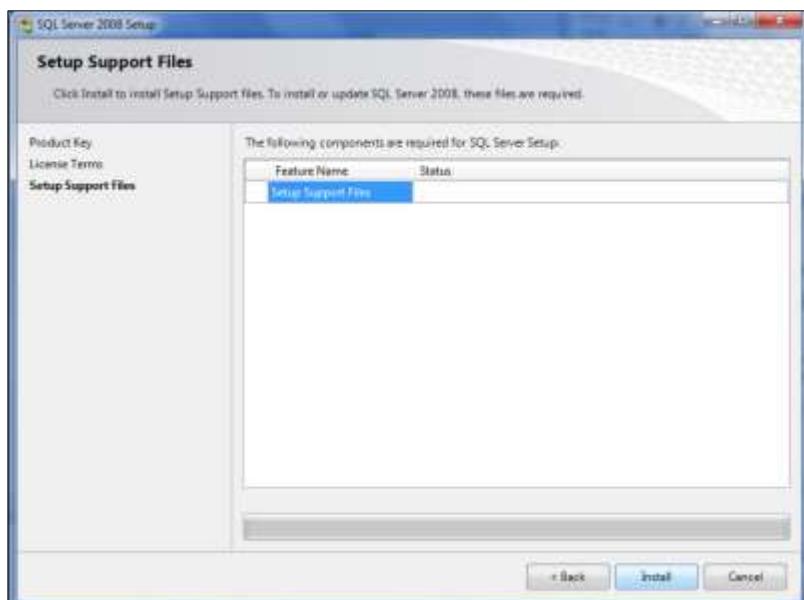


6. ស្វែង Tick យើង I accept the license term >

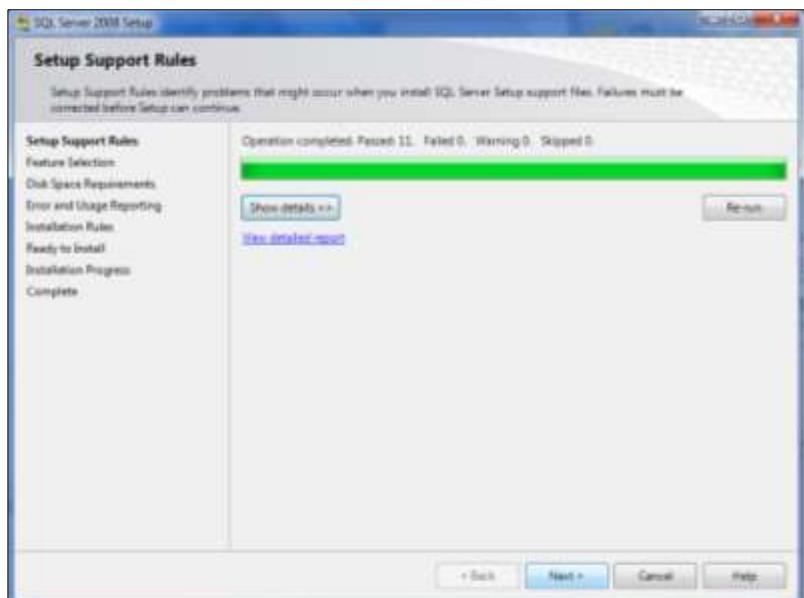
## 7. Next Button &gt;



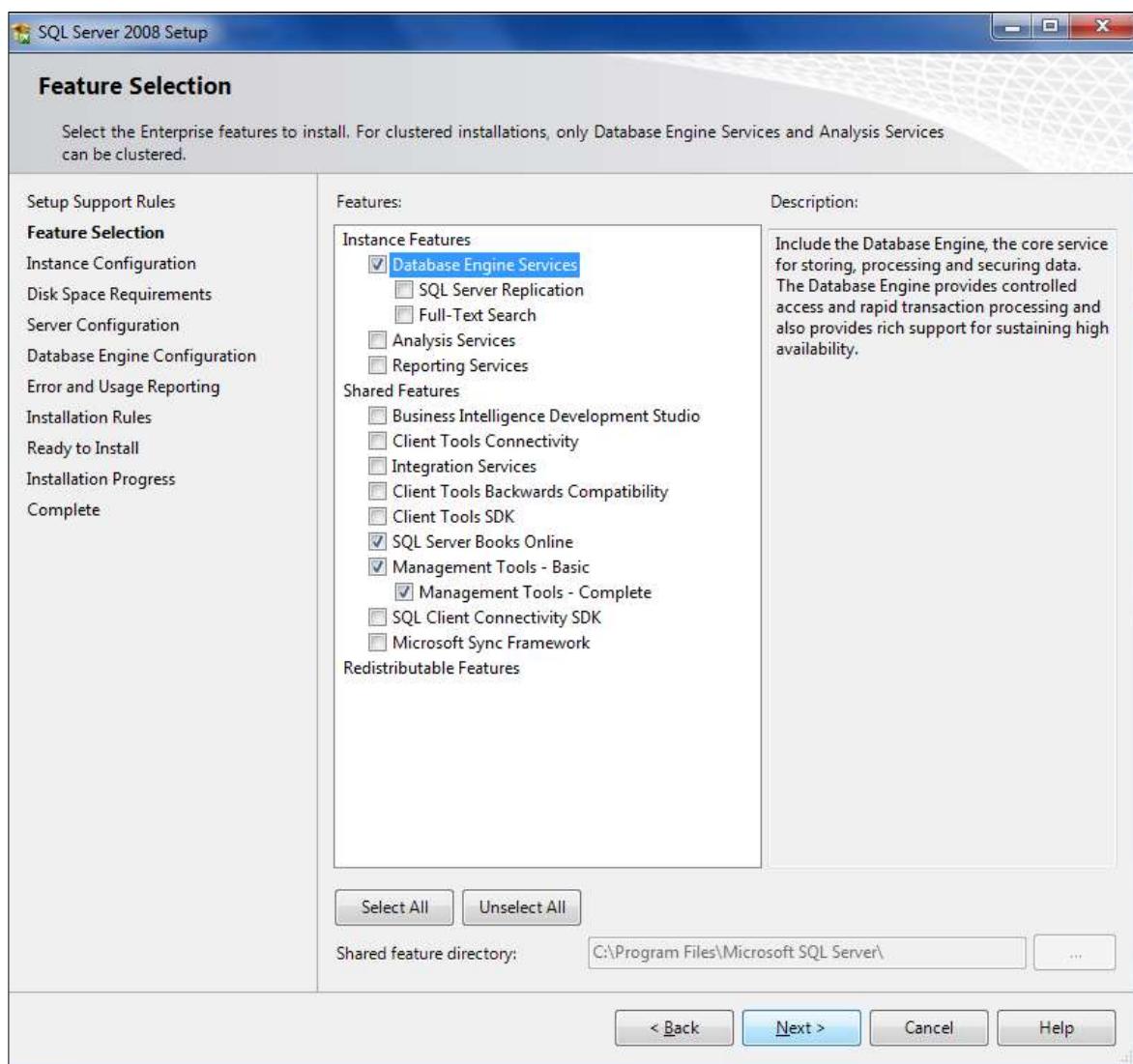
## 8. ඇඟි Install Button &gt;



## 9. Next Button &gt;

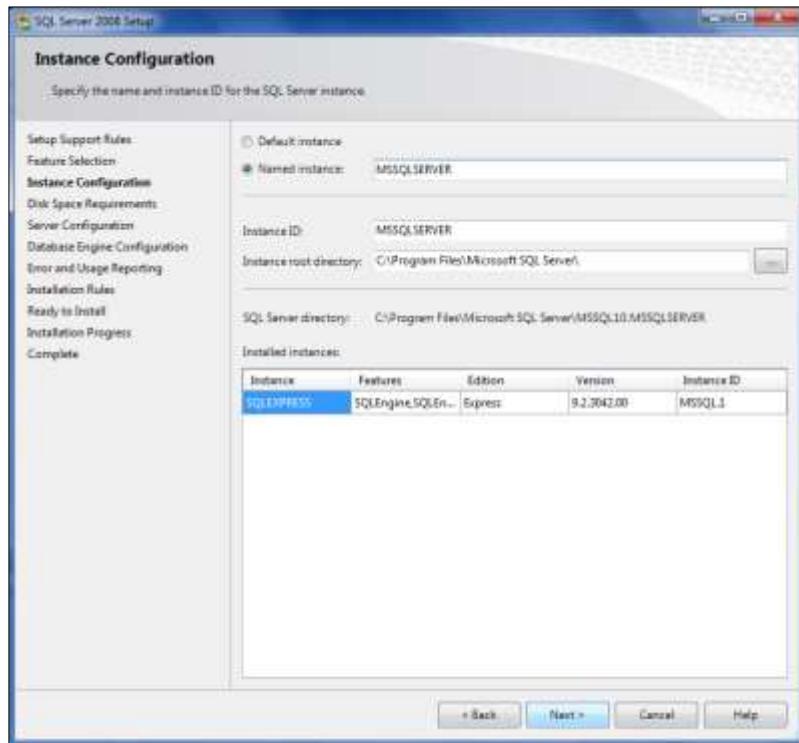


10. ស្ម័ប Tick យក Database Enging Service ដើម្បីធ្វើការ Install Database Services >
11. ស្ម័ប ធានា Tick ត្រង់ Client Tools Connectivity ក្នុងករណីផែលខ្លួន Install លើម៉ាស៊ីនធម្មតា ហើយចូរ Tick យក វាប្រសិនបើបើង Install លើម៉ាស៊ីន Server >
12. ស្ម័ប Tick យក SQL Server Books Online និង Management Tools è Basic >
13. Next Button >

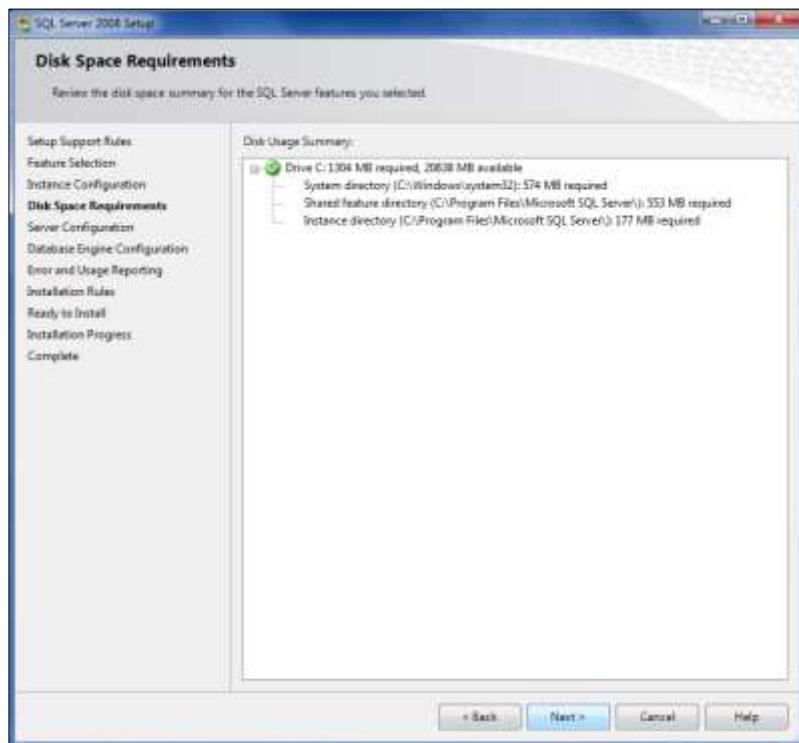


14. ស្ម័បទិន្នន័យក Named instance >

15. Next Button >



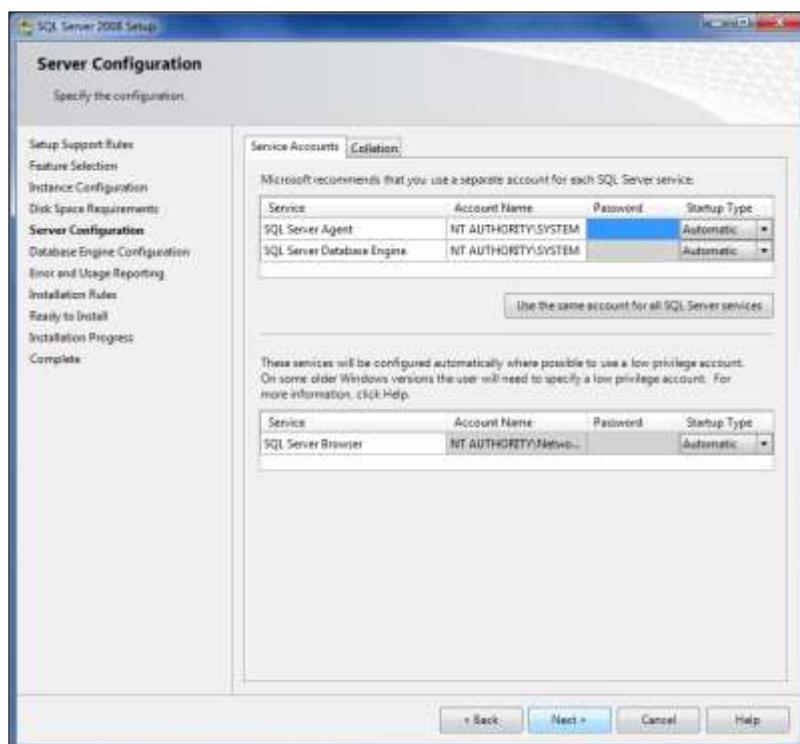
16. Next Button &gt;



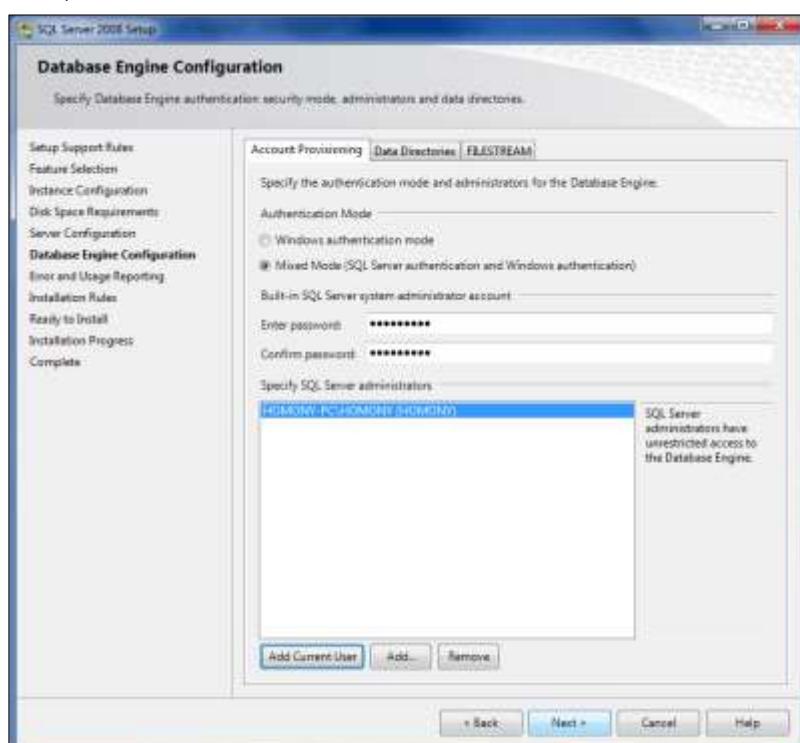
17. ត្រង់តិចបន្ថែម Microsoft recommends នឹងប្រឡងថា Account Name ស្មូមនឹសបយក SYSTEM និង Startup Type នឹងប្រឡងកម្រិត Automatic >

18. ត្រង់តិចបន្ថែម These services will be នឹងប្រឡងថា Startup Type ស្មូមនឹសបយក Automatic Automatic >

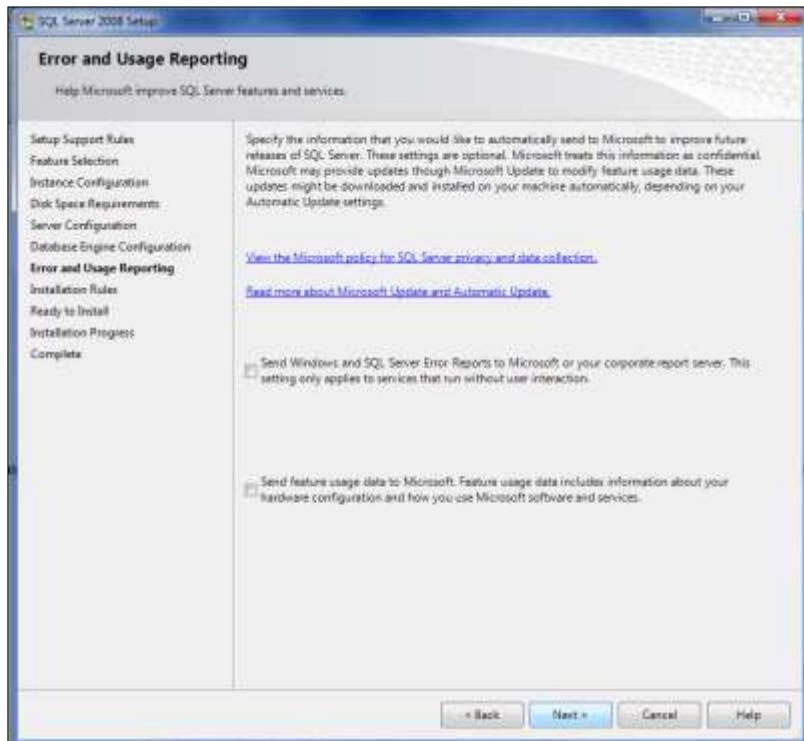
19. Next Button >



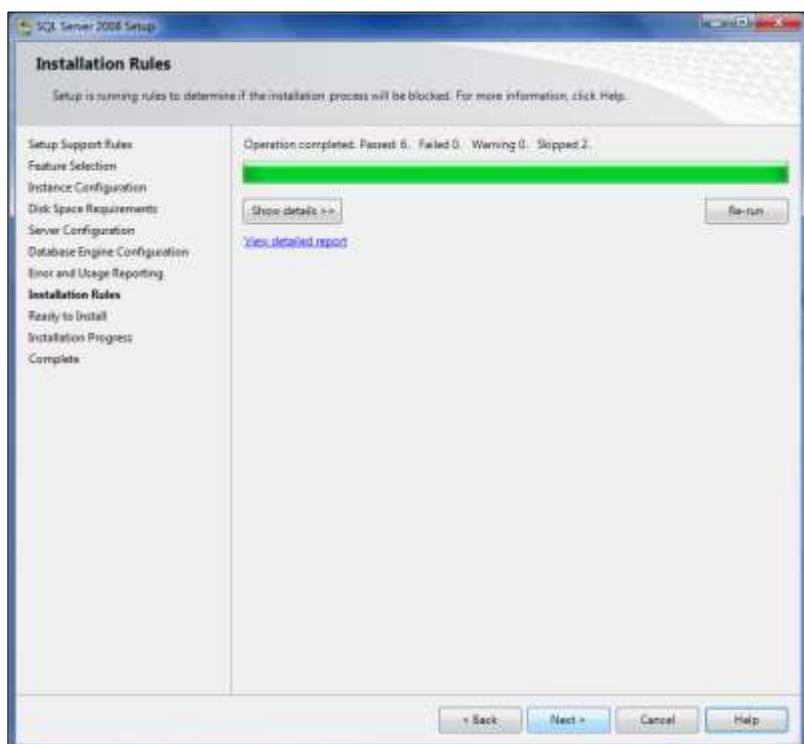
20. នឹងយក Mixed Mode >
21. គ្រប់ពាក្យ Enter password ស្មូលចំពោះ Password របស់ SQL Server Admin សំភាគ Log in >
22. គ្រប់ពាក្យ Confirm password ស្មូលចំពោះផ្តល់ព័ត៌មានអ្នកប្រើប្រាស់ >
23. ចូល Add Current User Button ដើម្បីផ្តល់ព័ត៌មាន assign current user ជាអ្នកប្រើប្រាស់ SQL Server ដើរ >
24. ចូល Next Button >



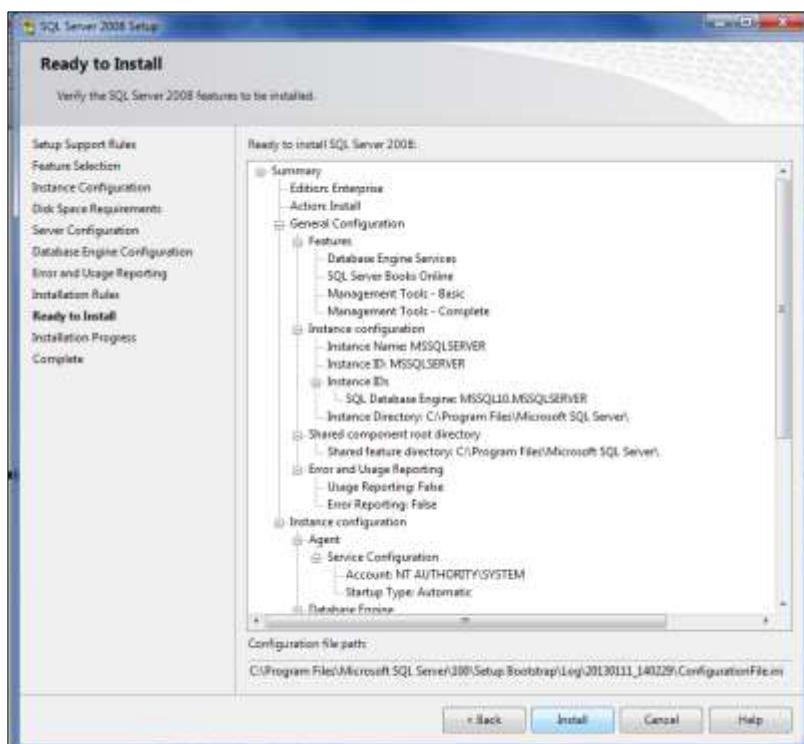
25. ស្មូលធ្វើ: Tick ទាំងពីរចំពោះ >
26. Next Button >



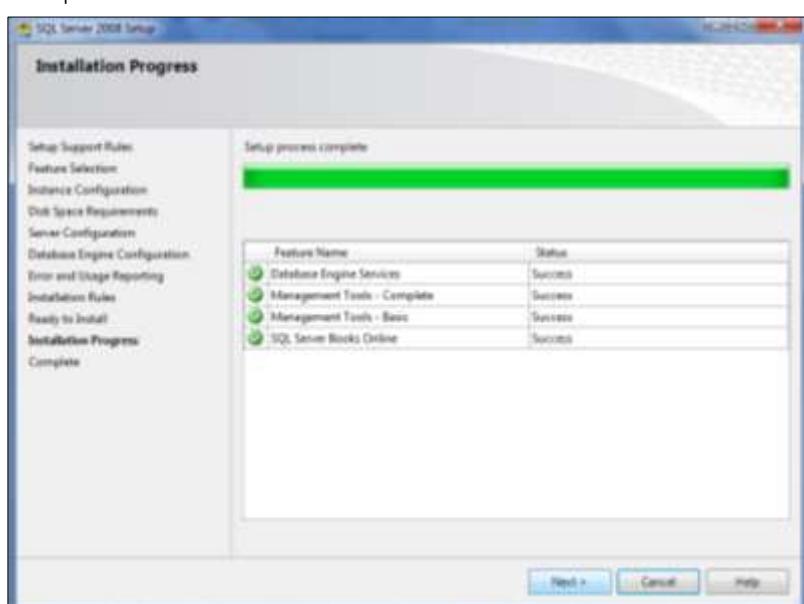
27. Next Button &gt;



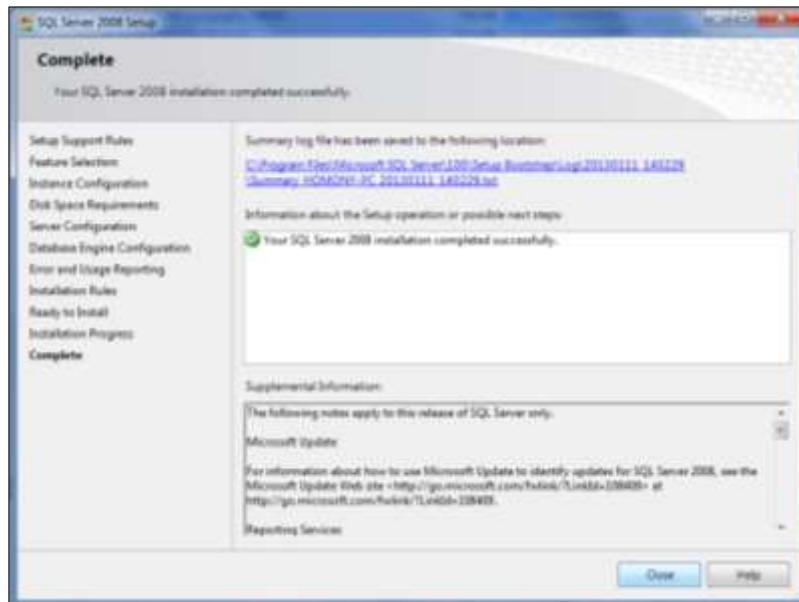
28. එමු පිටපතේ පිටපතේ පිටපතේ පිටපතේ &gt;



29. පිළි Next Button &gt;



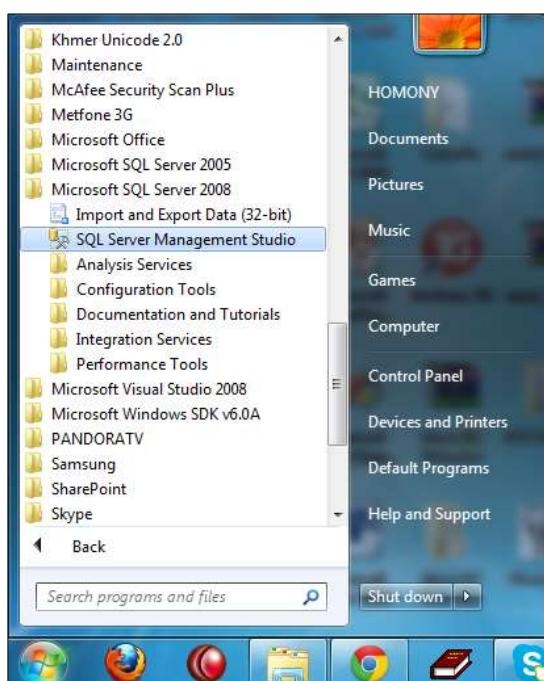
30. පිළි Close Button නේ මුළු පැවත්වන සඳහා Setup SQL Server 2008



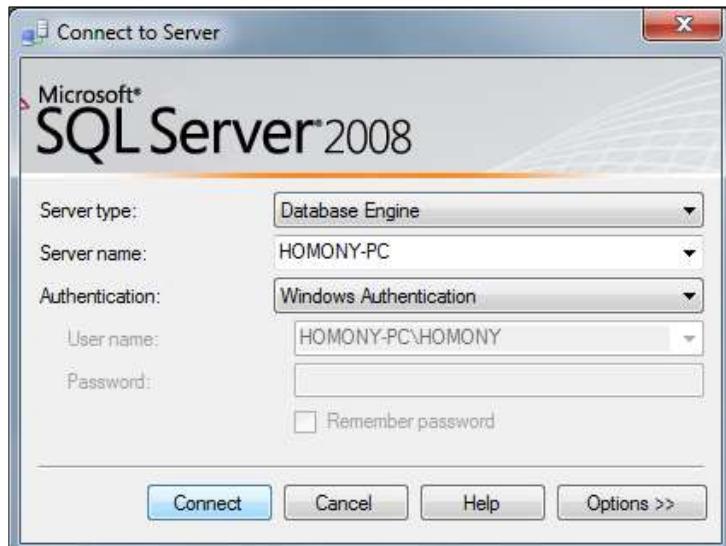
# មេរីនៃការបង្កើត 2: ការចូលទៅការណ៍ SQL Server 2008

## 1. របៀបបង្កើតការណ៍:

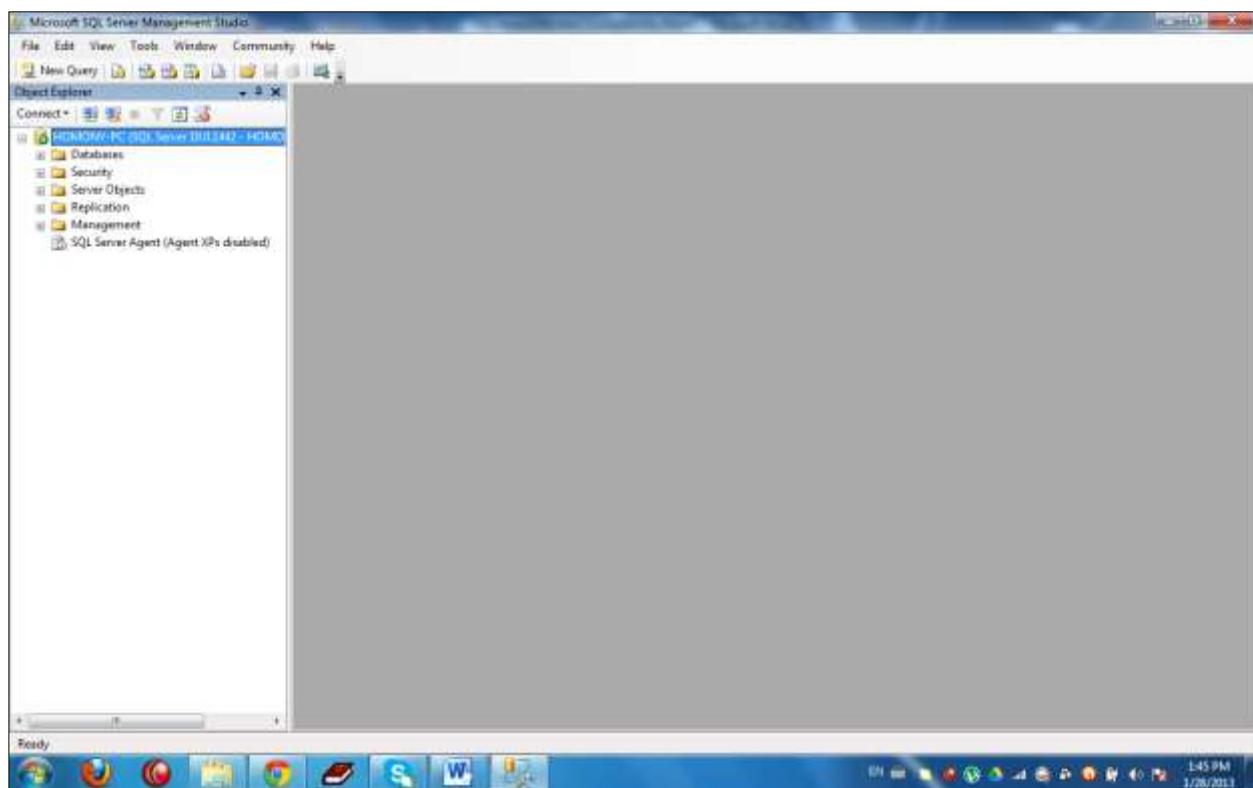
1. ចូច Start Menu >
2. All Programs >
3. Microsoft SQL Server 2008 >
4. SQL Server Management Studio >



5. ជួយប្រភេទ Server type ស្មើនឹងប្រយោជន៍ Database Engine >
6. ជួយប្រភេទ Server name ស្មើនឹងប្រយោជន៍លេខាងក្រោម Server >
7. ជួយប្រភេទ Authentication នឹងប្រយោជន៍ Windows Authentication >
8. ចូច Connect Button >

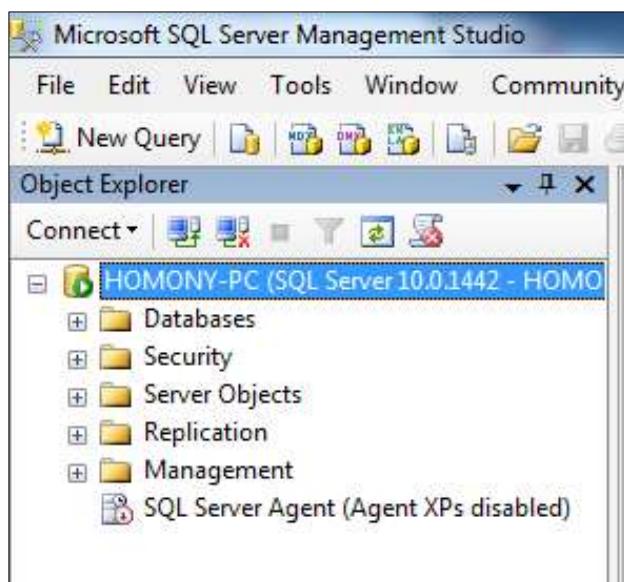


9. ចាប់មក SQL Server Management Studio នឹងបង្ហាញពីចាប់ផ្តើម



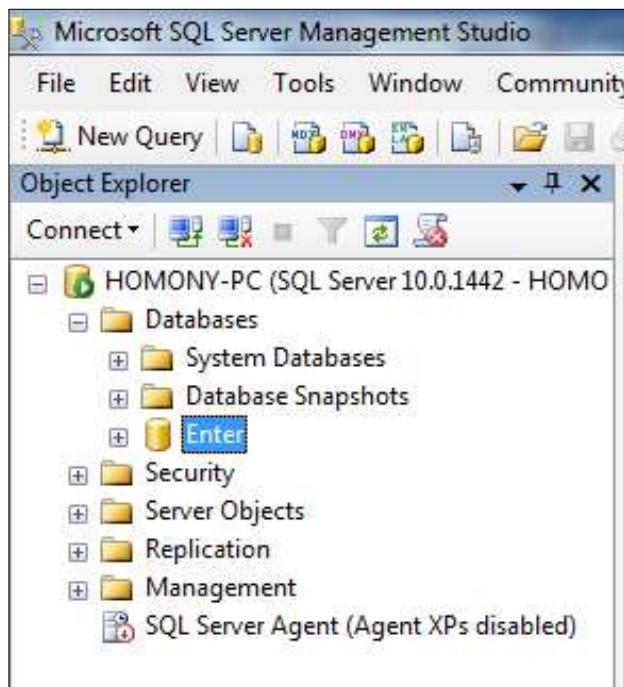
## **2. Object Explorer:**

Object Explorer គឺជាសេចក្តីបណ្តុះបណ្តាលពី ព័ត៌មានទាំងអស់របស់ Server ដែលវាត្រូវបានភាពជាបន្ទាប់មកបញ្ជាក់។ វាមែនជា tree view ទាំងមិនមែនជាបន្ទាប់មកបញ្ជាក់ទេ ប៉ុន្មោះបានការការពិចារណាដែលបានធ្វើឡើង។ Double-click លើ Folder នឹងបានបញ្ជាក់ថា វាបានបញ្ជាក់ឡើង។



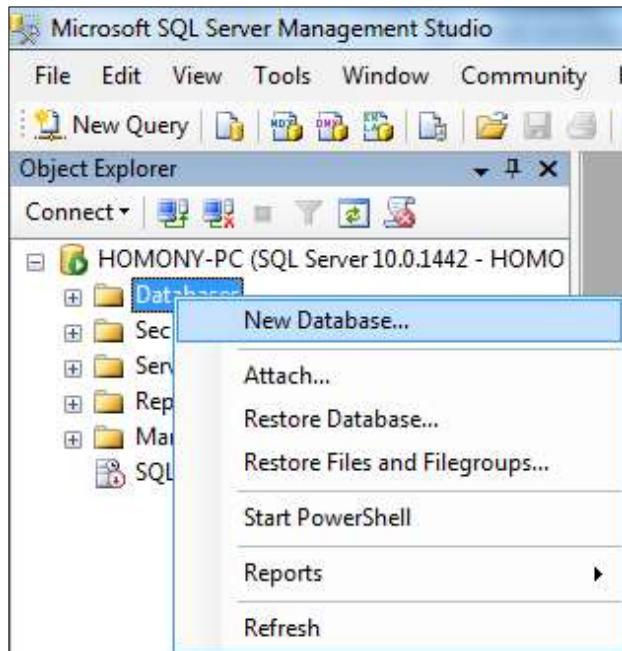
### **3. Databases:**

បន្ទាប់មកពីបាន Expand Tree View ហើយវានិងបង្ហាញពី Child Objects ដូចជានៅត្រួតដែលក្នុងនោះមាន Database ផងដែរ។ Database គឺជាយករាជ្យពីមាន ផែនក្រឹមបានរាយបច្ចេកទេរស ដើម្បីផ្តល់ភាពអាយក្រូលក្នុងការ ប្រើប្រាស់ គ្រប់គ្រង និង កែសម្រួលរវាង



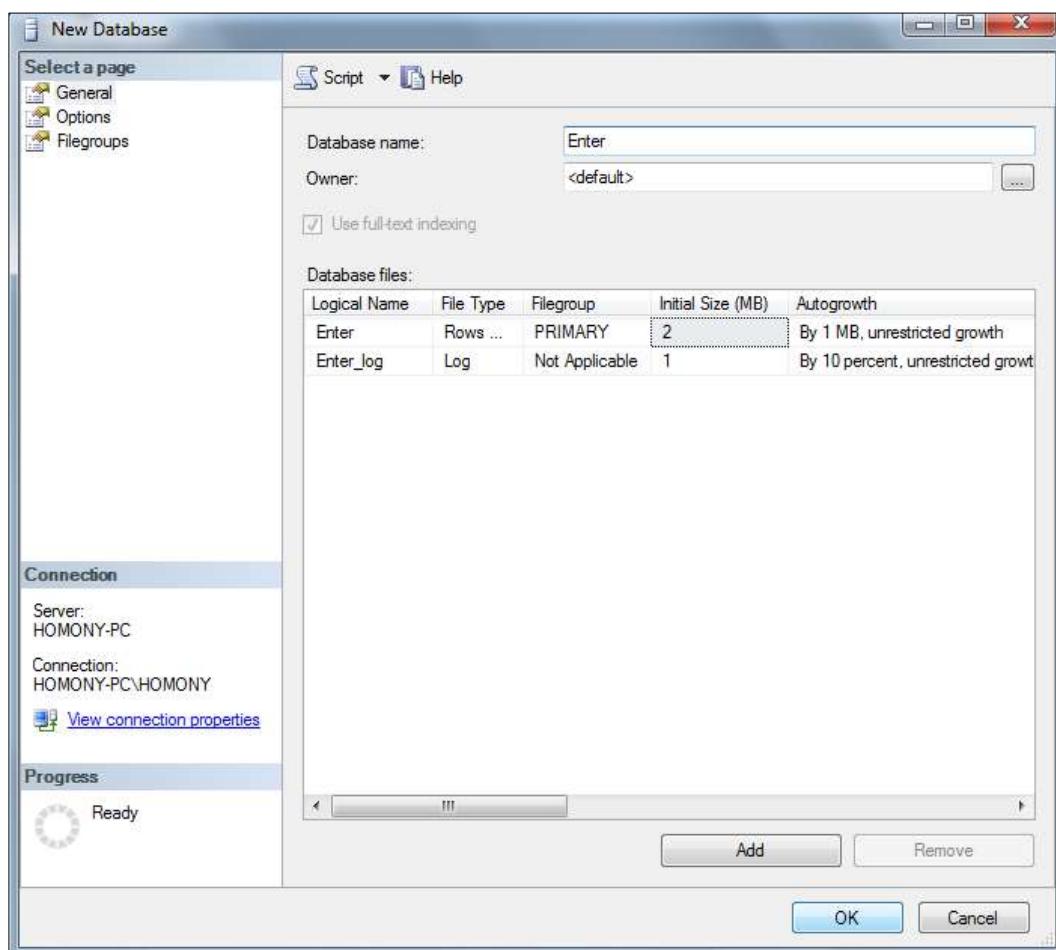
### **4. ការបង្កើត Database:**

1. ធ្វើ Mouse ស្វ័យបើ Database ហើយក ឯ New Database >



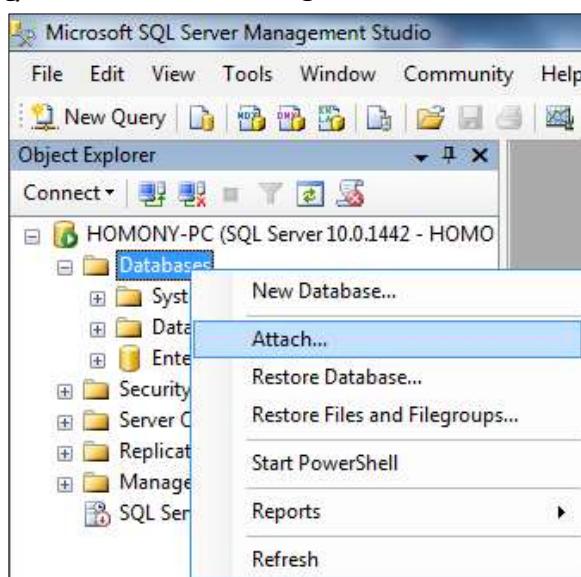
2. ជួយប្រអប់ Database name ស្មើសាក់ពេលវេលា Database (Enter) >

3. ធ្វើ OK Button

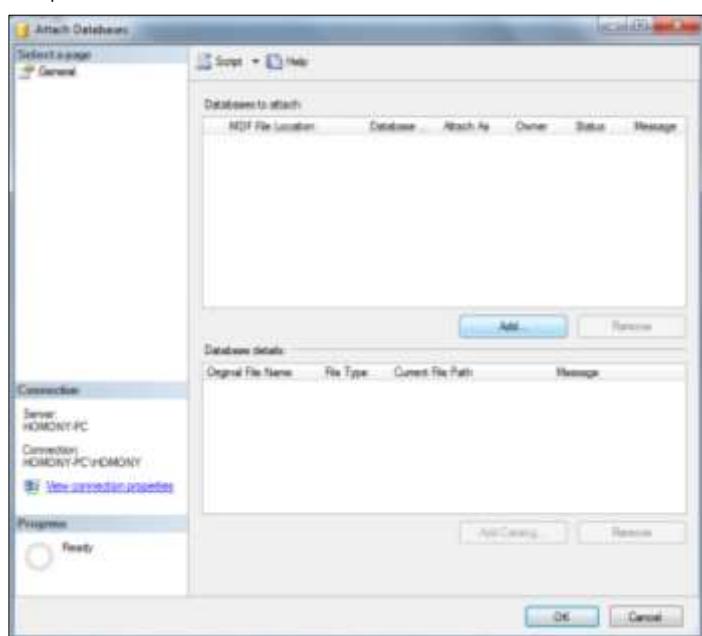


## 5. រចនា Attach Database:

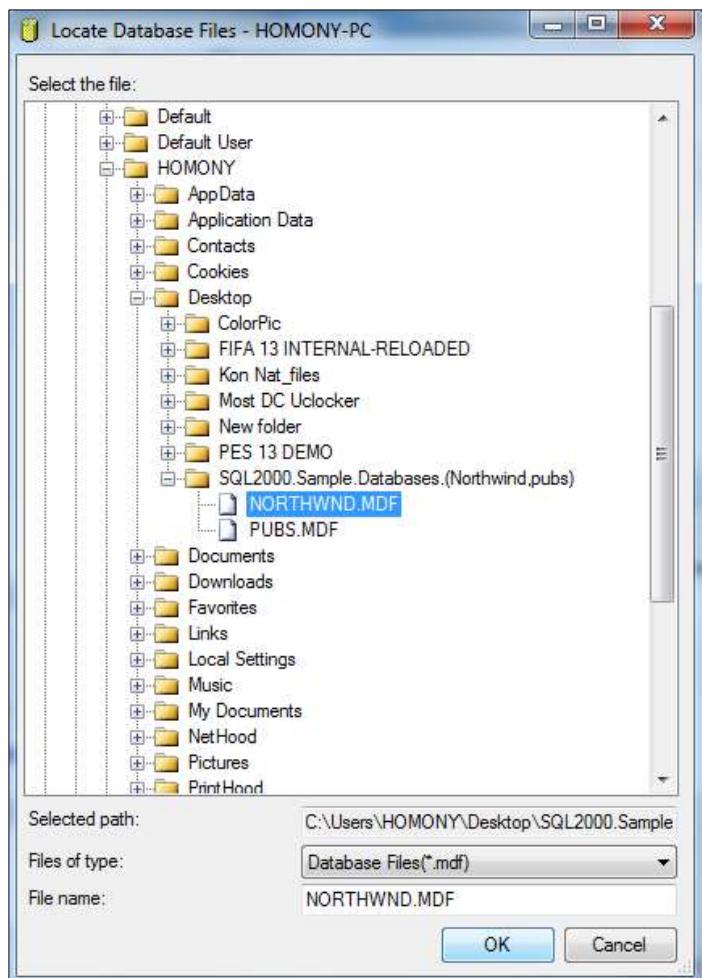
1. ធ្វើ Mouse ស្វែងប្រើ Database ហើយកិចចាAttach >



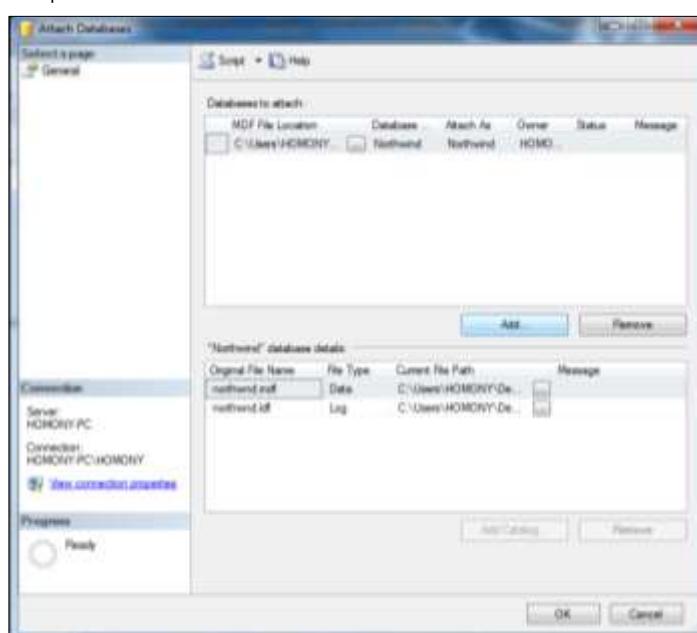
2. සුදු Add Button >



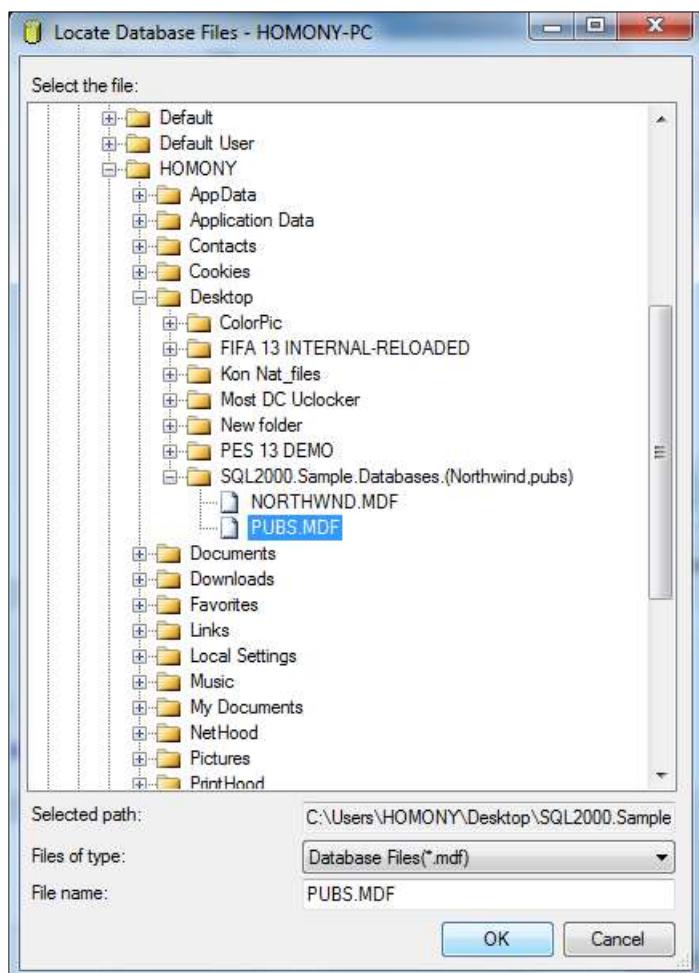
3. සුදු රීස්බයික File ඊඩඟහන Extension \*.MDF (NORTHWIND.MDF) >



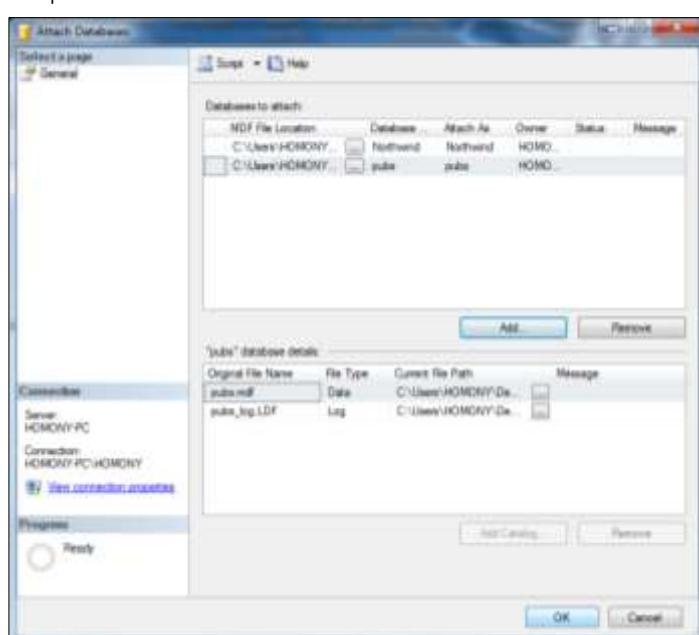
4. තුළ Add Button >



5. සුෂ්‍රේණීය උපයක් File ඊටුවනා මාන් Extention \*.MDF (PUBS.MDF) >

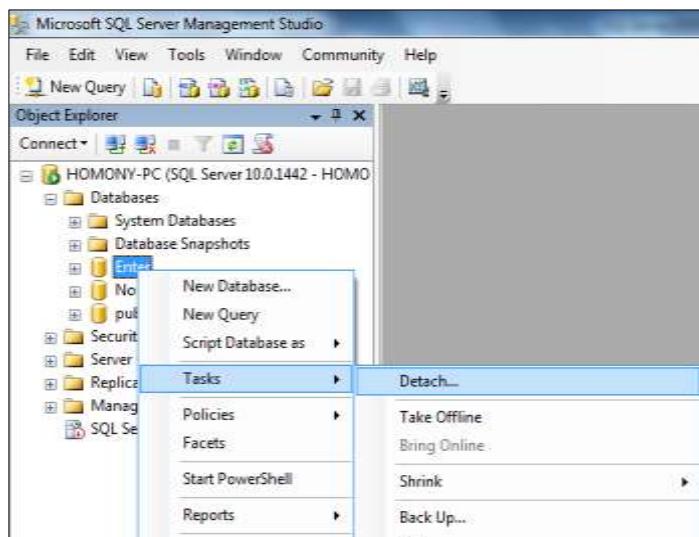


6. සුදු OK Button

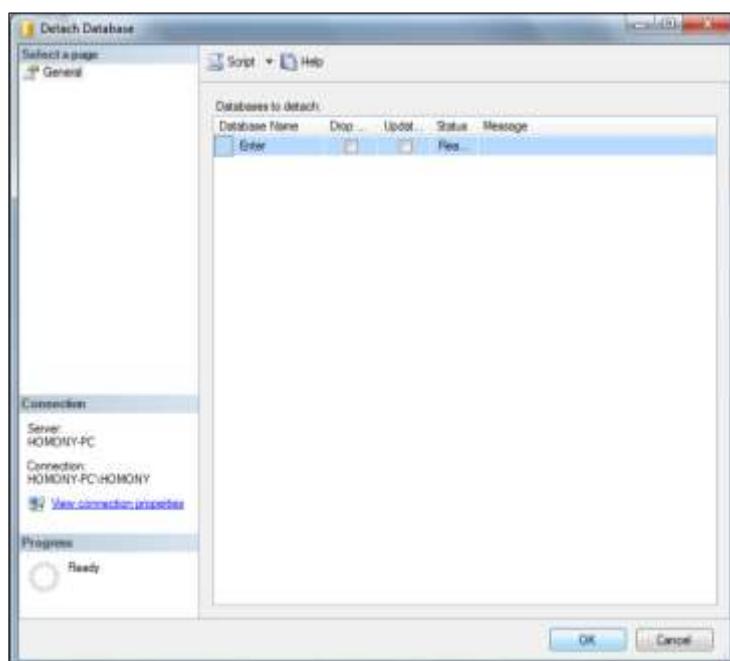


#### 6. පෙනීම Detach Database:

1. සුදු Mouse ස්ථාපී Database බාහුයෝගීතා කර >
2. යක්ක තැක්කු Tasks >
3. Detach >



4. ඇඟ OK Button



# ផែវឌីលទី 3: សិក្សាទី Normalization

## និទ្ទេ Relationship

### 1. សិក្សាទី Key Term:

- > Relation : គឺជា Tables ទាំងឡាយដែលស្ថិតនៅក្នុង Database
- > Attribute : គឺជា Columns នៃក្នុង Table ឬ នៃក្នុង Relation
- > Tuple : គឺជា Rows ដែលស្ថិតនៅក្នុង Table
- > Candidate Key: គឺជាការ រួមបញ្ចូលគ្មាន Attribute ចាប់ពី ពីរបុគ្គិនឡើងទៅដើម្បីបង្កើតបានជា Tuple  
ដើម្បីបានទិន្នន័យខ្ពស់។
- > Primary Key : គឺជា Attribute អ្នបែបដែលគ្មានទិន្នន័យស្ថិតនៅក្នុង Attribute អ្នបែប នៅទីនេះ  
ត្រូវធោនានំណែនាំខ្ពស់។ វាតើជា Key អ្នបែបដែលជាដែករបស់ Candidate Key ។
- > Foreign Key : គឺជា Attribute អ្នបែបដែល ត្រូវបានភ្លាយថា Relation Ship ជាអ្នបែបនឹង Primary និងទៀត។

### 2. សិក្សាទី Data Normalization:

Data Normalization គឺជាតីផើករក្សាការ រួចចំ Fields និង Tables នៃ Realtion Database ដោយបង់ប្រកបដោយ Table គ្នា។ ដើម្បីការបង្ហាញនិន្នន័យប្រចាំថ្ងៃ ឬច្បាស់ពី Database ។ វាគារធ្វើនៃ Normalization គឺជាការបង់ប្រកប Tables ដំឡើង ដែលមានការបង្ហាញនិន្នន័យ។ ការបង់ប្រកប Tables ដំឡើង គឺជាការបង្ហាញនិន្នន័យដោយបង់ប្រកបដោយការ add, delete ឬ edit Data នៃក្នុង Field ឬការបង្ហាញនិន្នន័យដោយការជាអ្នបែបនឹង Field នៃ Tables ដោយបង្ហាញនិន្នន័យ។

Data Normalization ត្រូវបានបង់ប្រកបជា 3 ប្រភេទដែលមានដូចជា:

- > First Normal Form
- > Second Normal Form
- > Third Normal Form ។

### 3. First Normal Form (1NF):

First Normal Form គឺមានចារចារ Attribute និងក្នុង Relation ដូចនេះ បំនែកនៃ Information ដែលត្រូវបានបង់ប្រកបប្រចាំថ្ងៃ និងក្នុងក្នុង Relation ដូចនេះ ហើយចាំបាច់ត្រូវរក្សាទុក ហើយព័ត៌មានទាំងនេះនៃក្នុង Attribute អ្នបែប ត្រូវបានបង់ប្រកបដោយការបង្ហាញនិន្នន័យ។

- > Relation មានការបង្ហាញនិន្នន័យដោយការបង់ប្រកបដោយការបង្ហាញនិន្នន័យ។

StudentID	Name
1234	Martin R. Jones
5678	Gilda Swanson

នៅត្រង់ Attribute Name គឺជូនុវត្តន៍យោះពេញរបស់ សិស្សនឹងមួយយ៉ាង ដើម្បីមានយោះត្រូវលើក ឬក្នុងករណី ដើម្បីរៀបចំការ search ឬ sort ទៅតាម last name ឬ first name របស់សិស្ស នៅត្រង់ Attribute Name គឺជាបញ្ហាមួយ ដោយទាមទារទៅយើងត្រូវធ្វើការបំបែកទៅជាទំន់ 1NF ដោយត្រូវបំបែក Attribute Name ទៅជាការនៃ First Name, Last Name ឬបន្លំមជាមួយនឹង Middle Name ទៅតាមការនៃតែប្រសើរ។

> Relation ឧអក្រមនេះគឺត្រូវបានបំបែកទៅជាទំន់ 1NF ហើយ

StudentID	LastName	FirstName	MiddleInitial
1234	Jones	Martin	R.
5678	Swanson	Gilda	

បញ្ហាមួយទៀតផើលម្អាតកើតឡើងនៅពេលប្រគល់បំបែកទៅជាទំន់ 1NF គឺករណី Repeating Groups មានន័យថា Relation បានប្រើប្រាស់ Attributes ប្រចើន ដើម្បីក្រាងកុំពេញ Data ដែលមាន Item ខ្ពស់ ។

> Relation ឧអក្រមនេះគឺមានផ្ទាល់ Repeating Groups ដែលទាមទារត្រូវបំបែកទៅជាទំន់ 1NF បន្ថែមទៀត

StudentID	Course1	Course2
1234	Elementary Chemistry	Hydroponic Gardening
5678	Zen Meditation	Elementary Chemistry

ដើម្បីធ្វើការកែត្រូវប្រវាខ័ណ៌ជាទំន់ 1NF គឺ ផ្ទាល់ Attribute Course បញ្ជូនត្រូវ ហើយបង្កើតជា Repeat Row វិញ្ញាន ក្នុងនោះចំពោះ Attribute ទាំងអស់ត្រូវតែត្រូវ Primary Key រៀបដើម្បីការពារមិនចោន Row ស្មូនកើតឡើង។

> Relation ឧអក្រមនេះគឺត្រូវបានកែត្រូវប្រវាខ័ណ៌ជាទំន់ 1NF ហើយ

StudentID	Course
1234	Elementary Chemistry
1234	Hydroponic Gardening
5678	Zen Meditation
5678	Elementary Chemistry

#### **4. Second Normal Form (2NF):**

ចំណោះ Second Normal Form ធនធានទាំងនេះ Attribute និមួយពី នៅក្នុង Relation ត្រូវធ្វើការភ្លាប់ Relation ទៅបីជាប្រើនគាមដែលអាមេរិក Relation ដែលមានទិន្នន័យរួម នៅលើលេបតែមួយ Tuple ប៉ុណ្ណោះគ្មានមួយ ព័ត៌មានទាំងអស់នៃក្នុង Tuple ត្រូវបានប្រើបានដើម្បីបង្កើតឡើង។ នៅក្នុងការបង្កើតឡើងនេះ ត្រូវបានបញ្ជាក់ថា ព័ត៌មានទាំងអស់នៃក្នុង Tuple ត្រូវបានបង្កើតឡើងជាប្រើនគាមដែលគ្មានទិន្នន័យ។ នៅក្នុងការបង្កើតឡើងនេះ ត្រូវបានបញ្ជាក់ថា ព័ត៌មានទាំងអស់នៃក្នុង Tuple ត្រូវបានបង្កើតឡើងជាប្រើនគាមដែលគ្មានទិន្នន័យ។

> Relation នាយករាជ គិមិនខាន់មានទំនើប 2NF នៅពីឱ្យទេ ល្អាច StudentID និង CourseID ត្រូវបីជាសំ Primary Key នូមដូចដើម្បីការណា record ស្ថិន បញ្ជី Field CourseName គឺពីអង្កេកទៅលើ CourseID តែម្ចាស់ល្អាច មិនមែន StudentID ទេពីឱ្យ។

<b>StudentID</b>	<b>CourseID</b>	<b>CourseName</b>
1234	2222	Penthouse Farming
5678	3333	Animal Husbandry in Manhattan

ដើម្បីបង្កើតជាចាត់វេង 2NF ចំណោះ CourseName គឺត្រូវធ្វើការកាត់ចេញទៅបង្កើតជា Relation ដូចខាងក្រោមនេះ ព័ត៌មានដែលទាក់ទងនឹង Course នេះ ១ មាននូយបង្កើតរបៀបក្នុង Relation ពីរដូចគ្នា ដែលទី ១ គឺ មាន Attribute StudentID និង CourseID ចំនួន ២ គឺ មាន Attribute CourseID និង CourseName បន្ទាប់មកធ្វើការ ត្រូវបង្កើតជាដំឡើង ឬ Relationship ទៅទាំងពីរ។

### **5. Third Normal Form (3NF):**

> ចំណោះ Relation ខាងក្រោមគឺជា ព័ត៌មានលំអិតុរបស់ CourseID នៃ Table ខាងលើដើម្បី 2NF ហើយត្រូវត្រួតពិនិត្យថា Relationship មិនគឺជាគិសន៍ល្អសេរីបង្កើតប្រាកាសជាចំណាំង 3NF

<b>CourseID</b>	<b>CourseDescription</b>	<b>Units</b>	<b>CostPerUnit</b>	<b>TotalCost</b>
2222	Penthouse Farming	4	40.00	160.00
3333	Animal Husbandry in Manhattan	3	50.00	150.00

## 6. සිංහල තීරණය Relationship:

Relationship គឺជាការត្រូវបង្កើតនៃការពិន័យ រវាង Relation (Table) ជាប្រចិនថ្មីលើក្នុងបច្ចេកទេស។ ដើម្បីទទួលឱ្យការពិន័យក្នុងបច្ចេកទេស ត្រូវបានធ្វើឡើងជាអាជីវកម្ម។ ដើម្បីធ្វើឡើង គឺជាផ្លូវការស្នើសុំការការពិន័យក្នុងបច្ចេកទេស។ ការការពិន័យក្នុងបច្ចេកទេស ត្រូវបានធ្វើឡើងជាអាជីវកម្ម។

Relationship ត្រូវបានបងចែកជា 3 ប្រធ័ន ដើម្បី ផិលមានដូចជា:

- > One-to-Many
  - > Many-to-Many
  - > One-to-One

## **7. One-to-Many Relationships:**

One-to-Many Relationship តើផ្លូវការណ៍របស់ Value និង Match ត្រាងឯកដី Primary Key នៃ Relation មួយ ជាមួយនឹង

Value នៅក្នុង Foreign Key នៃ Relation ដើម្បីពិតាពាណិជ្ជកម្មនៃ First Relation ដាក់ពាណិជ្ជកម្មនៃ Match ដាក់ពាណិជ្ជកម្ម

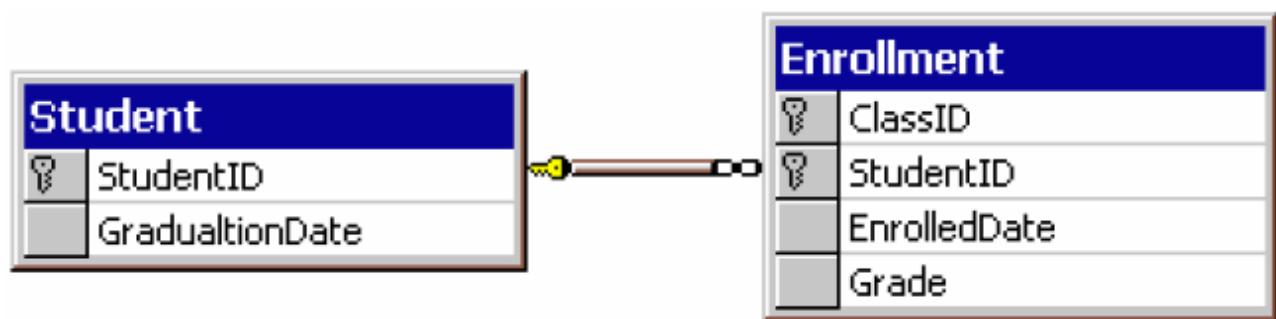
Tuples ជារូបីននៃកុំភង Second Relation ។ Ex: សិល្បៈម្នាក់នាមចចេះណែនាំរៀនបានម្រើនច្បាក់ ។

Student Enrollment Relationship គឺជាប្រភេទ master-child ប្រជាប្រភេទ "has a" និង One-to-Many Relationship ។

សិស្សរបាក់តាមមានចំនួនការចំណោមដូចជា A Student "has a" some number of class enrollments ។

ប៊ែនសិស្សម្នាក់កីវាទ តានការចេញយោង បច្ចេះល្អបោះពុម្ពយ បុកីវាទចេញបោះបានបើនផងដែរ។ ប៊ែនដូចម្ល៉ែន

ចំនួនការនេះការប្រើប្រាស់ទំនាក់ទំនងសាខាតីមានតម្លៃសម្រាប់ប្រើប្រាស់ទៅលាង។

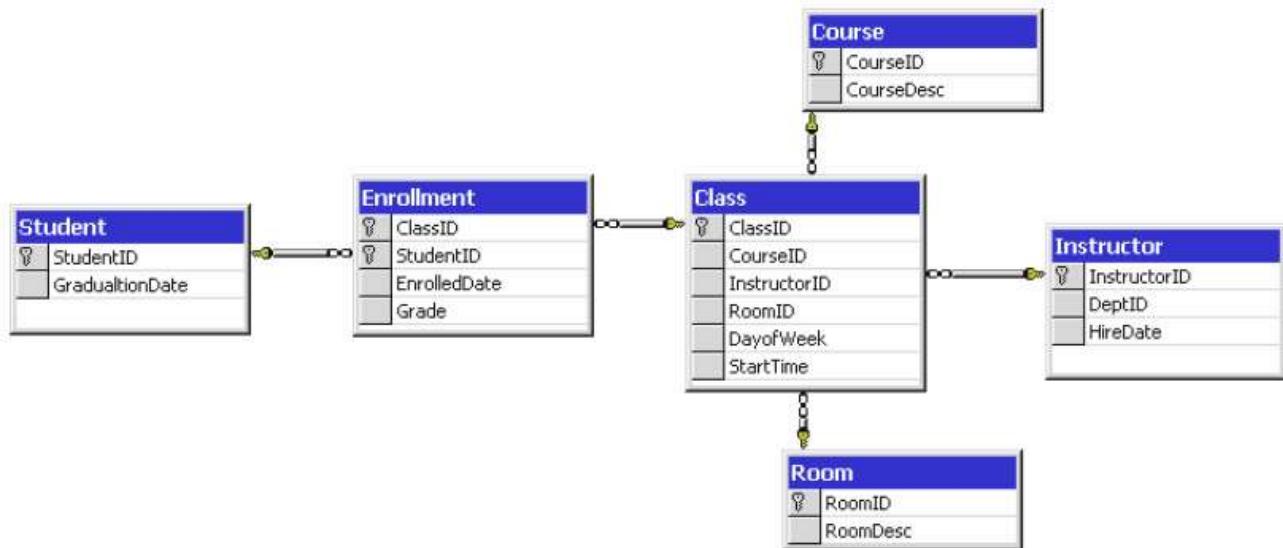


ប្រភេទមួយទៀតនេះ One-to-Many Relationship គឺ ប្រភេទ Lookup ដូច "is a" Relationship ។ Ex: Instructor និងមួយទៀតជាន់ "is a" member នៃ Department មួយនៃភ្នែក School ។



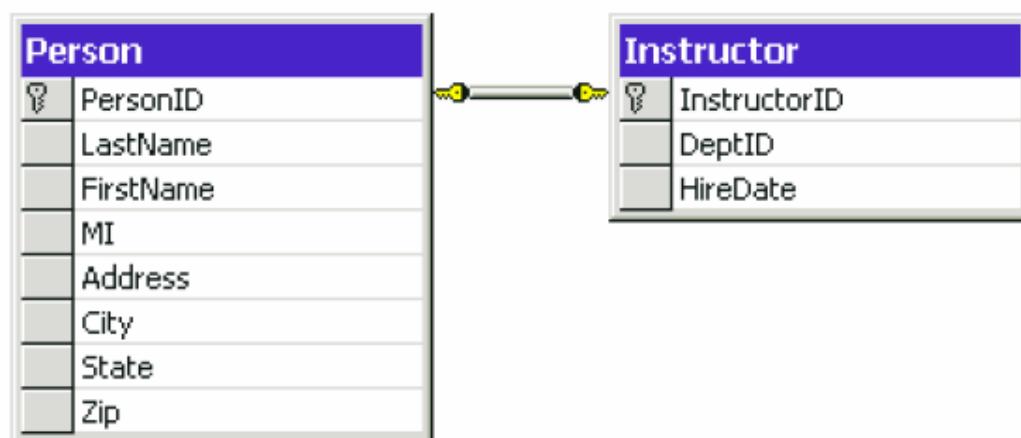
## **8. Many-to-Many Relationships:**

សិក្សាក៏ត្រូវបានបង្កើត Class ហើយ Class មួយនាមព្រមទាំងផ្លូវជាយសិក្សាបច្ចុប្បន្ន Relation ប្រធែនដីជាពីរគ្រឿងនៃគ្រឿងទី៣ Third Relation មួយទេ ។



#### **9. One-to-One Relationships:**

One-to-One Relationship គឺជា Relationship ដែល Relation មានតួរ មានតែ មួយចំណោមដែល Match ត្រាង ពេលនោះ Foreign Key កើតពីបានសិក្សាបានគឺជា Primary Key និងមិនមែនពីរគ្រឿងទៀត។ Ex: យើងមាន Table Person សំរាប់រក្សាទុក នូវពីរគ្រឿងនៃប្រភពបិកចំអនុសំគ្រួង Database ហើយក្នុងនោះ Instructor Table កើតឡើងពាណិជ្ជកម្មបច្ចេកទេសថា Instructor ត្រូវបានគិតជាប្រភពបិកចំនៃពីរគ្រឿង Person មួយចំនួនឡើងដែលមានគិតជាប្រភពបិកចំនៃពីរគ្រឿង Instructor ត្រូវបានគិតជាប្រភពបិកចំនៃពីរគ្រឿង Person Relation នាមមាន ប្រភពបិកចំនៃពីរគ្រឿង Instructor Relation ប៉ុន្ម័េរនៃពីរគ្រឿង Instructor Relation ដើម្បីនាំគិតថា Instructor Relation មានគិតជាប្រភពបិកចំនៃពីរគ្រឿង Person Relation ជានិច្ច។



ក្នុងករណីនេះ InstructorID គឺជា Primary Key ដើម្បីនឹងជា Foreign Key ដើម្បីនឹងគ្រប់ Instructor Row ទាំងអស់ ត្រូវត្រូវគិត Match ជាមួយនឹង Person Row បើនេះ PersonID គឺជា Primary Key ត្រូវបានគិតជាប្រភពបិកចំនៃពីរគ្រឿង Person Relation នៅទីព្រោះ គ្រប់ Person Row ទាំងអស់ អាចប្រើបានអាច Match ជាមួយនឹង Instructor Row នៅទីព្រោះទេ។

#### **10. សិក្សាទុក Referential Integrity:**

> Enforce Referential Integrity មាននៅលើថា Student Record មិនអាចត្រួតបាន Delete កំពុងពេលដើម្បីនា Enrollment Records ណាមួយមានបើយបានត្រូវចោរការនៃ Student នោះទេ ម្ខាក់អ្វីត Student កើមិនអាចធ្វើការ គក្សប្រ StudentID ដោយមិនបានក៏ប្រឡាតក្ខង Related Enrollment Records នោះដែរ។

> Cascading Updates and Deletes មានន័យថាបើដឹងអាចធ្វើការ កែត្រួចនឹងន័យបុណ្យបច្ចិន្ទន័យនៃកូដ Primary Key Value ដូន ហើយស្របពលជាមួយគ្នាតាមរឿងផ្សាយ កែត្រួចបុណ្យបច្ចិន្ទន័យនៃកូដ Table ដើលពាក់ព័ន្ធ ដោយស្ម័យ ប្រភពិនិត្យដីរ។ Ex: បើដឹងត្រូវការកែត្រួចន័យ StudentID នៃកូដ Student Table ដើលរាជាដា Foreign Key នៃកូដ Enrollment Table ត្រូវឡើង SQL Server នឹងធ្វើការបន្ថុ "Cascade" ដោយផ្តល់កែត្រួចន័យនៃកូដ Student Table នឹង Update ត្រូវ Records ទាំងអស់នៃកូដ Enrollment Table ។ ចំនួនក Cascading Delete នឹងត្រួតពិនិត្យបើយើង ឬបាន Data នៃកូដ Table មួយនៅរានីដឹងផ្សាយ ប្រចាំ Data នៃកូដ Table ដើលពាក់ព័ន្ធដឹងផ្សាយ។

# មេគ្រែទី 4: រាយបច្ចុប្បន្ន Database

## និង Table

### 1. សិក្សាទី Database Storage:

នៅក្នុង SQL Server គឺជានប្រើប្រាស់ Physical Storage ចំនួន 3 ប្រភពទឹកប្រើប្រាស់នៃលក្ខណនោះមានដូចជា:

- > Pages : គឺជា Storage Unit តួចជាអេកបំផុតនៃលមានចំហំត្រីមតែ 8KB ប៉ុណ្ណោះ។ វាតាមតួនាទីកត់ត្រានូវចំហំផុតនៃ Row អ្នយនៅក្នុង Table ត្រូវមកជាអ្នយនិង large text និង image columns និង Store Pages ដើម្បីត្រូវការបញ្ជីយោង។ Page អ្នយនាមជាកិច្ច Data ចេញពី Database Object តើអ្នយប៉ុណ្ណោះ។
- > Extents : គឺជា Pages ដែលមានចំនួន 8 ជាបន្ទប្បាប់ត្រាលូ ឬ 64KB ។ នៅពេលដែលបង្កើត Table ធ្វើឱ្យ Index ត្រូវ វានិងទទួលបាននូវ Single Page ដែលត្រូវក្រឡាតកនៅក្នុង Extent ។ ពេលដែលមាន Object ធ្វើត្រូវបាន បង្កើតជា បន្ទប្បាប់នោះ វានិងធ្វើការបង្កើត Pages ជាប្រើប្រាស់នៃទូទាត់ 8 បើយធ្វើការ Mix ចូលបញ្ជីនៅក្នុង Extents។ SQL Server នាគធ្វើការបង្កើត Extents បន្ថែមចៀនទៅតាមតម្លៃការ។
- > Files : Database ត្រូវបាន Store ត្រូវបានក្រឡាតកនៅក្នុង Operating System Files ដើម្បីការ Configure និងការកើនឡើងជាបន្ទប្បាប់ត្រូវបានក្រឡាតកនៅពេលដែល Data ត្រូវបាន Add បន្ថែមចៀនទៅ Database នាម ត្រូវបាន Store នៅក្នុង Files ជាប្រើប្រាស់បើនេះត្រូវគោលនយោបាយ Data ចេញពី Database តើអ្នយប៉ុណ្ណោះ។ SQL Server ប្រើប្រាស់ File Extensions ចំនួន 3 ដែលមានដូចជា:

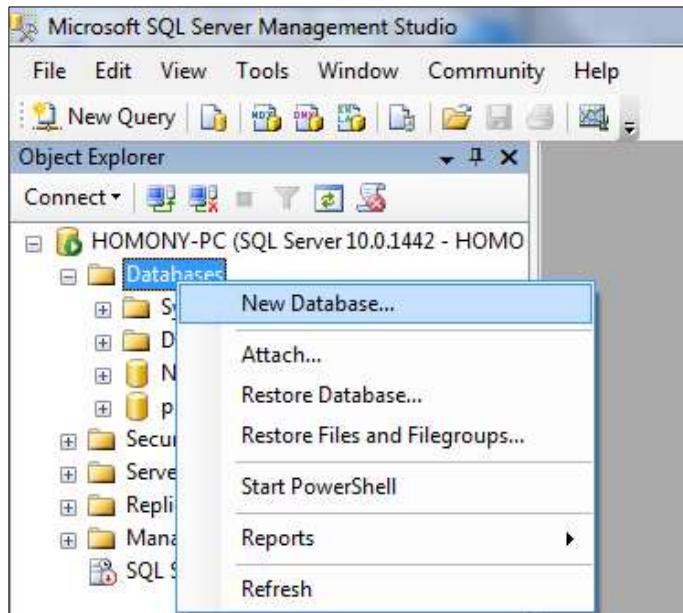
- MDF : ជាដី Primary Data File ដែលសំខាន់ជាអេកបំផុតនៃកិច្ច Database និងប្រើប្រាស់
- NDF : ជាដី Secondary Data Files ដែលជាកិច្ចិមានបន្ថែមរបស់ Database
- LDF : ជាដី Log File ដែល Store នូវដំនឹកការ ការអារម្មស់ Database

### 2. សិក្សាទី Files និង Filegroups:

Primary Data File គឺជាដីដែលត្រូវបានក្រឡាតកនៅពេលដែលបង្កើត Database ធ្វើ ហើយបង្កើតកិច្ច និងការបង្កើតបន្ថែមនូវ Files និង Filegroups បាននៅក្នុងបន្ទប្បាប់ត្រូវបានក្រឡាតកនៅក្នុង Disk Drives ឬហូត្ថាគារ នៅពេលដែល Database របស់យើងមាន Multiple filesgroups វានិងជាបង្កើតលើវិនាគារអារម្មស់ Database ត្រូវបានក្រឡាតកនៅក្នុង Smultaneous Access ទៅកាន់ Database អ្នយបាមរយៈ Hardware Devices ជាប្រើប្រាស់ និង ហើយក្នុងនោះនៅពេលដែល Data ត្រូវបាន Update នោះវានិងធ្វើការបានយ៉ាងល្អជូន ជាអ្នយនិង Filegroup ដែលបង្កើតជាប្រើប្រាស់នៅពេលដែលបង្កើតជាប្រើប្រាស់នោះតើអ្នយ។

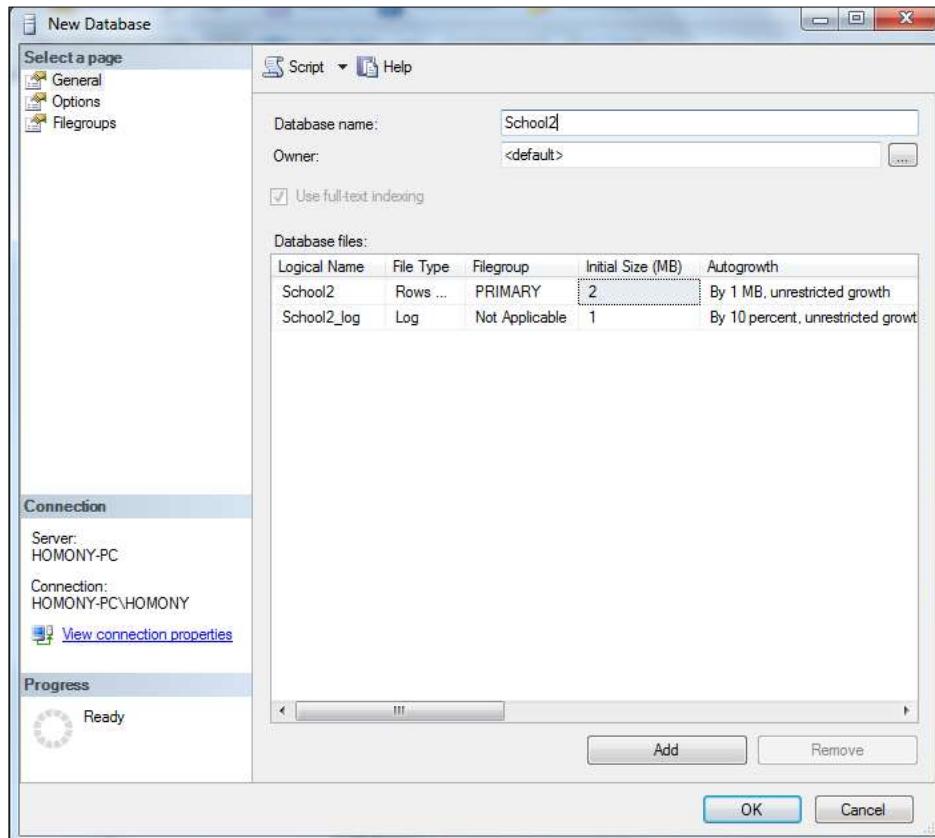
### 3. របៀបបង្កើត Database:

1. ចូច Mouse ស្វែរី Database >
2. New Database >



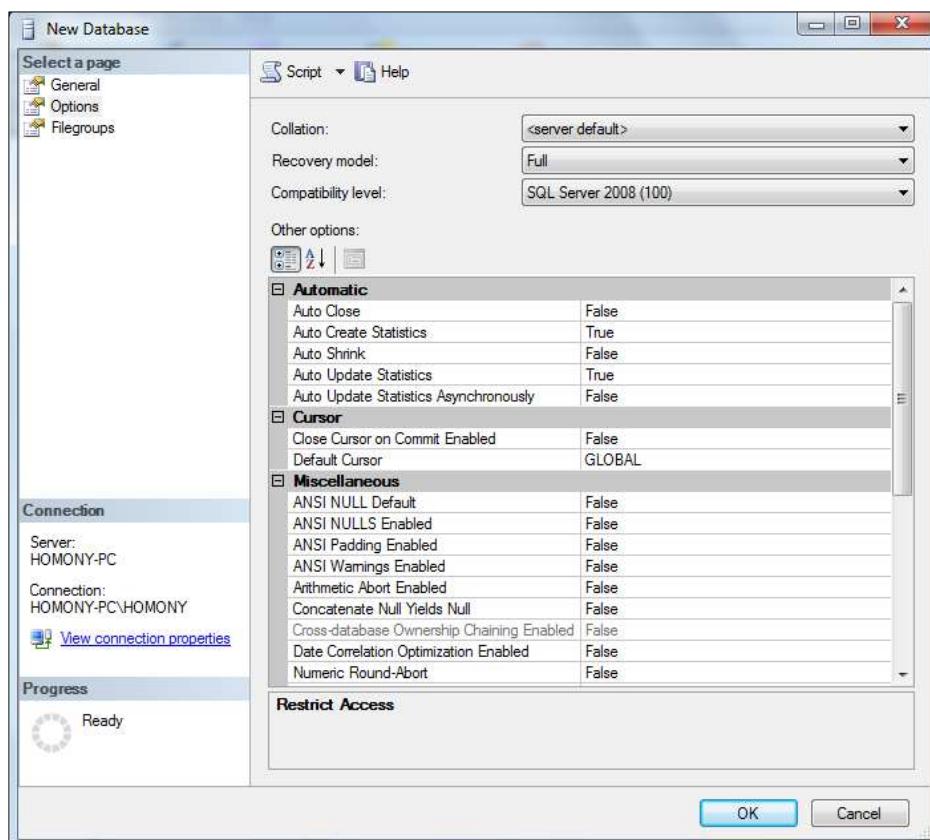
3. ផ្តល់ឈ្មោះ Database name ស្ថិតិថាក្នុងក្នុងថ្វី: Ex: School2 >

4. ចូល Options Tab >



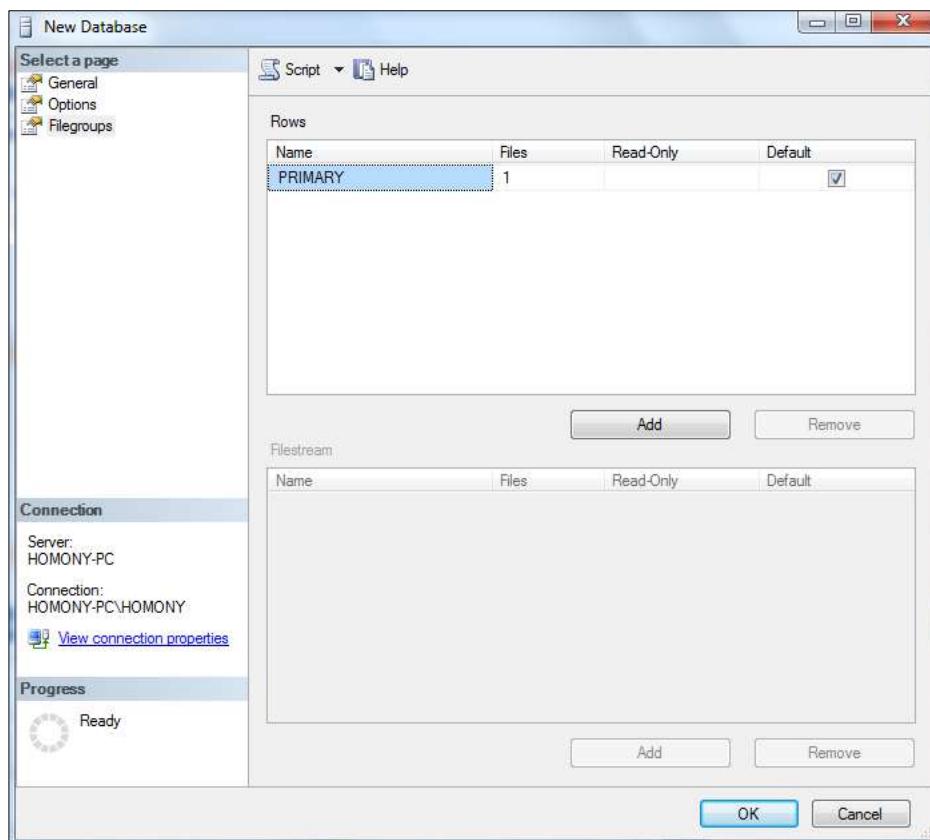
5. ស្ថិតិថាក្នុងក្នុង Option ធានាអ្នកដ្ឋានត្រូវការ >

6. ចូល Filegroups Tab >



7. ចូល Add Button ដើម្បីធ្វើការ បង្កើត File បន្ថែម >

8. ចូល OK Button



#### 4. អ្នកអនុវត្តកំណត់ឈ្មោះទៅ SQL Server Identifiers:

- > Identifiers គ្រែនិមានលេខាជាប់ពី 1 តួនាទីរហូតដល់ 128 តួនាទី

- > Identifiers គ្រឿងមានលេខាងច្បាស់ថាប់ពីជាមួយនឹង អក្សរ ឬ សញ្ញាមួយចំនួន @, \_, # ឬផ្លូវធមិនរាជជាមួយនឹង លេខ
- > Identifiers ដើម្បីបានប្រើប្រាស់ជាមួយនឹង @ គឺអាចត្រួមពេជ្តា Local Variables ឬ Parameters នៅក្នុង Transact-SQL តែប៉ុណ្ណោះ
- > Identifiers ដើម្បីបានប្រើប្រាស់ជាមួយនឹង # រាជជាតា Local Temporary Objects តែប៉ុណ្ណោះ
- > Identifiers ដើម្បីបានប្រើប្រាស់ជាមួយនឹង ## រាជជាតា Global Temporary Objects តែប៉ុណ្ណោះ
- > តួអក្សរបន្ទាប់ពីតួស៊ីមិនអាចជាតា Letters, Decimal Numbers, ឬ Symbols #,\$, \_
- > Space និង តួអក្សរពិសេសទីឡើង អាចប្រើប្រាស់បាន ឬ៖ត្រួត Identifiers គឺត្រូវបានប្រើប្រាស់ជាមួយនឹង Double Quote ឬ Square Bracket តែប៉ុណ្ណោះ។ ប៉ុន្តែភាគចំនួន Square Bracket គឺ Support បានយ៉ាងល្អប្រសិរី
- > Identifiers គឺអាចកំណត់លេខាង្លូចនិង Reserved Words នៅទេ។

## **5. សិក្សាពីការបង្កើត Table:**

បង្កើតមកពីបានធ្វើការបង្កើត Database ហើយនោះជំហានបង្កើតគឺត្រូវធ្វើការបង្កើត Table ផងដែរ។ ពេលដើម្បីបង្កើត Table គឺយើងត្រូវដាក់លេខាង្លូចរាជ្យ ហើយមែនលក្ខណៈជា Unique នៅក្នុង Database និងលេខាង្លូចគឺត្រូវមានទំនួចខ្សោចរកាមេរោគ។

Server.database.schema\_name.object\_name

ប៉ុន្តែភាគចំនួនគឺយើងតែបង្កើតប្រើប្រាស់ Syntax ត្រូវបានរក្សាទុក។

schema\_name.object\_name

Schemas គឺជាលេខាង្លូចរាជ្យ User Name ដើម្បីរាជ្យបានប្រើប្រាស់ជាយស្សែប្រភពតិត្តិសំរាប់គ្រប់គ្រង់ Objects ទាំងអស់ ដើម្បីបង្កើតឡើងជាយស្សែប្រភព Users (dbo or Database Owner) នៅក្នុង Sysadmin Server Role។ យើងអាចប្រើប្រាស់ Identifiers បានយ៉ាងត្រួមត្រូវជាមួយនឹង Schema Names ហើយយើងអាច Assign Multiple Users ជាមួយនឹង Schema បានដៃដី ១ ហើយបានបង្កើតឡើងជាយស្សែប្រភព យើងអាច Assign Permission ទៅកាន់ User ឬ Role មួយនៅលើ Schema ទាំងអស់។

ការរាយភាគចំនួនជាមួយនឹងការបង្កើត Table នៅគឺត្រូវបង្កើត Columns ដើម្បី Table មាន។ តួអក្សរនោះពេលបង្កើត Column ណាមួយនោះគឺជាកំណត់ Data Type រួមទាំង Options និងឡើងជាកំណត់ទៀតនាម Null Values បុគ្គលិក។

បង្កើតបែននេះឡើងគឺត្រូវកំណត់ Constraint នៅលើ Columns ដើម្បីមានជូនកំណត់ Foreign Key Constraints, Primary Key, ឬកំណត់ Default Values នៅក្នុង Columns ជាសិម។

## **6. SQL Server Data Type:**

តាមដឹកនាំក្រុមបង្កាញពី Data Type ទាំងអស់នៅក្នុង SQL Server 2008 ធ្វើមានការចែរចោរដាមួយនៃ Version មុនពេលដែលបានរៀបចំឡើង។

> 8 = SQL Server 2000[1]

> 9 = SQL Server 2005

> 10 = SQL Server 2008

Datatype	Min	Max	Storage	8	9	10	Type	Notes
Bigint	$-2^{63}$	$2^{63}-1$	8 bytes				Exact numeric	
Int	-2,147,483,648	2,147,483,647	4 bytes				Exact numeric	
Smallint	-32,768	32,767	2 bytes				Exact numeric	
Tinyint	0	255	1 bytes				Exact numeric	
Bit	0	1	1 to 8 bit columns in the same table requires a total of 1 byte, 9 to 16 bits = 2 bytes, etc...				Exact numeric	
Decimal	$-10^{38}+1$	$10^{38}-1$	Precision 1-9 = 5 bytes, precision 10-19 = 9 bytes, precision 20-28 = 13 bytes, precision 29-38 = 17 bytes				Exact numeric	Decimal and numeric data type is exactly the same. Precision is the total number of digits. Scale is the number of decimals. For both the minimum is 1 and the maximum is 38.
Numeric	no							
Money	$-2^{63} / 10000$	$2^{63}-1 / 10000$	8 bytes				Exact numeric	
Smallmoney	-214,748.3648	214,748.3647	4 bytes				Exact numeric	
Float	$-1.79E + 308$	$1.79E + 308$	4 bytes when precision is less than 25 and 8 bytes when precision is 25 through 53				Approximate numerics	Precision is specified from 1 to 53.
Real	$-3.40E + 38$	$3.40E + 38$	4 bytes				Approximate numerics	Precision is fixed to 7.
Datetime	1753-01-01 00:00:00.000	9999-12-31 23:59:59.997	8 bytes				Date and time	If you are running SQL Server 2008 and need milliseconds precision, use datetime2(3) instead to save

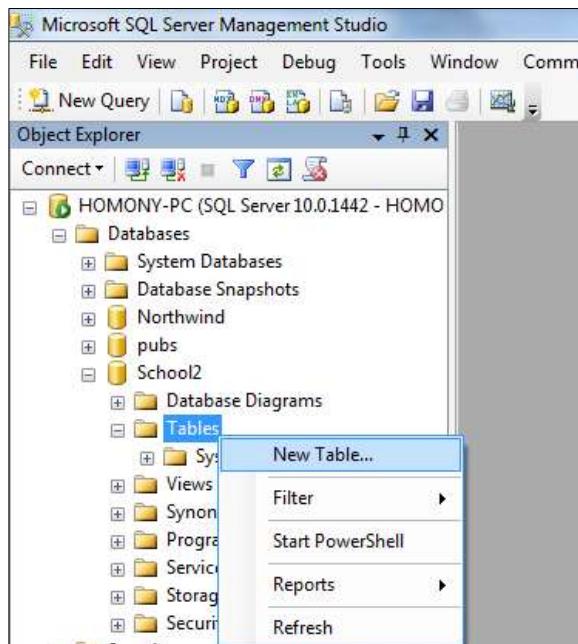
Datatype	Min	Max	Storage	8	9	10	Type	Notes
								1 byte.
Smalldatetime	1900-01-01 00:00	2079-06-06 23:59					Date and time	
Date	0001-01-01	9999-12-31		no	no		Date and time	
Time	00:00:00.0000000	23:59:59.9999999		no	no		Date and time	Specifying the precision is possible. TIME(3) will have milliseconds precision. TIME(7) is the highest and the default precision. Casting values to a lower precision will round the value.
Datetime2	0001-01-01 00:00:00.0000000	9999-12-31 23:59:59.9999999	Precision 1-2 = 6 bytes precision 3-4 = 7 bytes precision 5-7 = 8 bytes	no	no		Date and time	Combines the date datatype and the time datatype into one. The precision logic is the same as for the time datatype.
Datetimeoffset	0001-01-01 00:00:00.0000000 -14:00	9999-12-31 23:59:59.9999999 +14:00	Precision 1-2 = 8 bytes precision 3-4 = 9 bytes precision 5-7 = 10 bytes	no	no		Date and time	Is a datetime2 datatype with the UTC offset appended.
Char	0 chars	8000 chars	Defined width				Character string	Fixed width
Varchar	0 chars	8000 chars	2 bytes + number of chars				Character string	Variable width
Varchar(max)	0 chars	2^31 chars	2 bytes + number of chars	no			Character string	Variable width
Text	0 chars	2,147,483,647 chars	4 bytes + number of chars				Character string	Variable width
Nchar	0 chars	4000 chars	Defined width x 2				Unicode character string	Fixed width
Nvarchar	0 chars	4000 chars					Unicode character string	Variable width
Nvarchar(max)	0 chars	2^30 chars		no			Unicode character string	Variable width
Ntext	0 chars	1,073,741,823 chars					Unicode character string	Variable width
Binary	0 bytes	8000 bytes					Binary string	Fixed width
Varnbinary	0 bytes	8000 bytes					Binary string	Variable width
Varnbinary(max)	0 bytes	2^31 bytes		no			Binary string	Variable width
Image	0 bytes	2,147,483,647					Binary string	Variable width

Datatype	Min	Max	Storage	8	9	10	Type	Notes
		bytes						
Sql_variant							Other	Stores values of various SQL Server-supported data types, except text, ntext, and timestamp.
Timestamp							Other	Stores a database-wide unique number that gets updated every time a row gets updated.
Uniqueidentifier							Other	Stores a globally unique identifier (GUID).
Xml				no			Other	Stores XML data. You can store xml instances in a column or a variable.
Cursor							Other	A reference to a cursor.
Table							Other	Stores a result set for later processing.

<http://www.connectionstrings.com/Articles/Show/sql-server-data-type-reference>

## 7. របៀបបង្កើត Table:

1. ចូល Mouse ស្តាំបី Tables និងបង្កើត New Table >



2. គួរពរបៀប Column Name ស្មមកំណត់ថា 1 Column ទី 1 (PersonID) >

3. គួរពរបៀប Data Type ស្មមកំណត់ Data Type ទី 1 Column (int) >

4. ត្រួវបង្កើត Allow Nulls ស្មើម Clear Tick >

Column Name	Data Type	Allow Nulls
PersonID	int	<input checked="" type="checkbox"/>

5. ព្រមទាំង Column Properties នៃក្នុងប្រអប់ Size ស្មើមកំណត់ទំហំរបស់ Data Type (4) >

Property	Value
Computed Column Specification	
Condensed Data Type	int
Description	
Deterministic	Yes
DTS-published	No
Full-text Specification	
Has Non-SQL Server Subscriber	No
Identity Specification	
Indexable	Yes
Is Columnset	No
Is Sparse	No
Merge-published	No
Not For Replication	No
Replicated	No
RowGuid	No
Size	4

5. ស្មើមធ្លាប់ Expand ព្រមទាំង Identity Specification >

6. ព្រមទាំង (Is Identity) ផ្តើមបូក Yes >

Property	Value
Data Type	int
Default Value or Binding	
Table Designer	
Collation	<database default>
Computed Column Specification	
Condensed Data Type	int
Description	
Deterministic	Yes
DTS-published	No
Full-text Specification	
Has Non-SQL Server Subscriber	No
Identity Specification	
(Is Identity)	Yes
Identity Increment	1
Identity Seed	1
Indexable	Yes
Is Columnset	No
(Is Identity)	

7. ចូល Primary Key Button >

The screenshot shows the Microsoft SQL Server Management Studio interface with the 'Table Designer' open for a table named 'dbo.Table\_1'. The table structure is as follows:

Column Name	Data Type	Allow Nulls
PersonID	int	<input type="checkbox"/>
LastName	varchar(50)	<input type="checkbox"/>
FirstName	varchar(50)	<input type="checkbox"/>
MI	char(2)	<input checked="" type="checkbox"/>
Address	varchar(50)	<input checked="" type="checkbox"/>
City	varchar(50)	<input checked="" type="checkbox"/>
State	char(2)	<input checked="" type="checkbox"/>
ZipCode	varchar(15)	<input checked="" type="checkbox"/>

8. ស្វែងរក្សាទី Row ទី 2 នៅត្រង់ Column Name ហើយបង្កើត ចោនអង្គភាពការណ៍ ខាងក្រោម >

Name	Data Type	Length	Allow Nulls
LastName	varchar	50	No
FirstName	varchar	50	No
MI	char	2	Yes
Address	varchar	50	Yes
City	varchar	50	Yes
State	char	2	Yes
ZipCode	varchar	15	Yes

9. ខាងក្រោមជាលទ្ធផលដែលទទួលបាន >

Column Name	Data Type	Allow Nulls
PersonID	int	<input type="checkbox"/>
LastName	varchar(50)	<input type="checkbox"/>
FirstName	varchar(50)	<input type="checkbox"/>
MI	char(2)	<input checked="" type="checkbox"/>
Address	varchar(50)	<input checked="" type="checkbox"/>
City	varchar(50)	<input checked="" type="checkbox"/>
State	char(2)	<input checked="" type="checkbox"/>
ZipCode	varchar(15)	<input checked="" type="checkbox"/>

10. සුංචි පිහිටුව >

Column Name	Data Type	Allow Nulls
PersonID	int	<input type="checkbox"/>
LastName	varchar(50)	<input type="checkbox"/>
FirstName	varchar(50)	<input type="checkbox"/>
MI	char(2)	<input checked="" type="checkbox"/>
Address	varchar(50)	<input checked="" type="checkbox"/>
City	varchar(50)	<input checked="" type="checkbox"/>
State	char(2)	<input checked="" type="checkbox"/>
ZipCode	varchar(15)	<input checked="" type="checkbox"/>

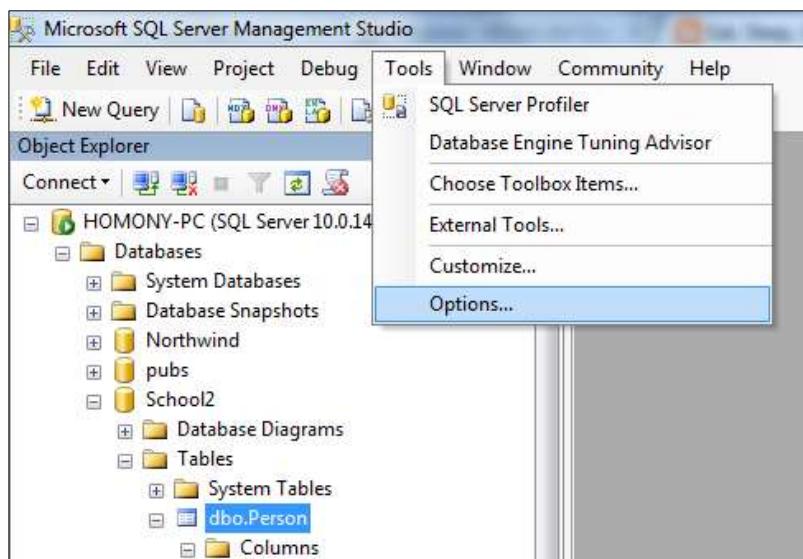
11. සුංචි පිහිටුව නිවැරදි කිරීමෙහි පිහිටුව පිහිටුව



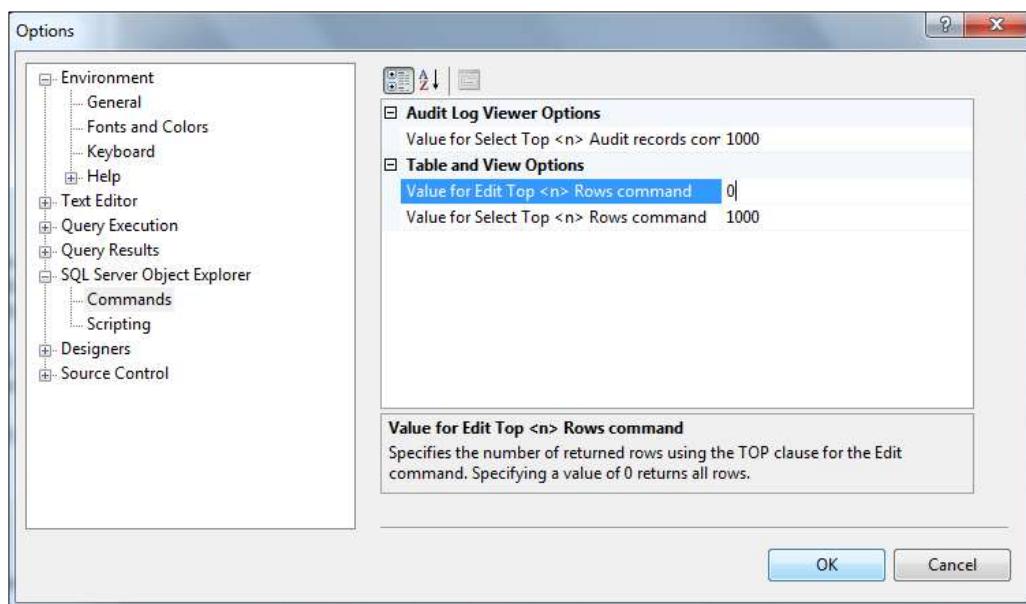
## 8. របៀប Open Table:

ជាយសារតែន្យាគ្នឹង SQL Server 2008 ត្រូវ Command Open Table នៅ គិតមាននៅលើកនោះដូចខាងក្រោម:

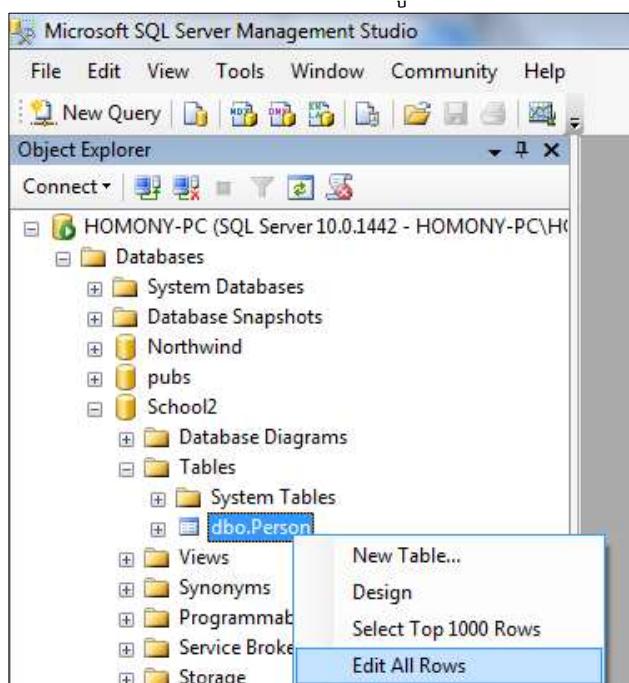
1. ចូច Tools >
2. Options >



3. ត្រួវផ្ទើកខាងក្រោម ចូច SQL Server Object Explorer >
4. ចូច Commands >
5. ត្រួវប្រអប់ Value fo Edit Top <n> Rows command ស្មមកំនត់លេខ 0 >
6. ចូច OK Button >



7. ធ្វើ Mouse ល្អាចូលឱ្យ Table វិនិយោគការ យកពាក្យ Edit All Rows >



8. ស្មបពណ៌និន្នន័យឡើតាមតម្លៃការ

HOMONY-PC.School2 - dbo.Person									
	PersonID	LastName	FirstName	MI	Address	City	State	ZipCode	
▶	6	Mony	Ho	NULL	Tuol Tompong	Phom Penh		855	
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	

## ៩. រដ្ឋភាព:

បញ្ជូនក្នុង Table មួយផែនរយៈ Course ដោយមាន Column Name និង Data Type ដូចខាងក្រោម:

HOMONY-PC.School2 - dbo.Course		
Column Name	Data Type	Allow Nulls
CourseID	int	<input type="checkbox"/>
CourseDesc	varchar(50)	<input type="checkbox"/>
PricePerUnit	smallmoney	<input checked="" type="checkbox"/>
Units	tinyint	<input checked="" type="checkbox"/>
		<input type="checkbox"/>



# គម្រោះលទ្ធផល ៥: សិក្សាតិ Constraints និង Database Diagrams

## ១. រច្ឆេត Computed Columns:

Computed Columns គឺអាចបង្កើតឡើងជាបច្ចុប្បន្នក្នុង Column ណាមួយដោយផ្តល់កម្រិតទៅជាបច្ចុប្បន្ន។ ឧបារាណក៏ថាលើក្នុង Course Table មាន Columns ចំនួនពីរ គឺ PricePerUnit និង Units ហើយបង្កើតការគណនាភាគផាយប្រកបចូលគ្នាដោយក្នុង Column តើម្មោះ Total ។ ធ្វើដោយគឺជាបច្ចុប្បន្ន Formula របស់ Column Total គឺត្រូវសរសើរការគណនា ( $[PricePerUnit] * [Units]$ ) ។

The screenshot shows the 'Course' table structure in the 'IPAKDEY7-PC.School2 - dbo.Course\*' database. The table has five columns: CourseID (int, primary key), CourseDesc (varchar(50)), PricePerUnit (smallmoney), Units (tinyint), and Total (float). The 'Total' column is highlighted with a dotted border. Below the table, the 'Column Properties' window is open for the 'Total' column. Under 'Computed Column Specification', the formula is set to  $([PricePerUnit] * [Units])$ . The 'Is Persisted' property is set to 'No'. The 'Condensed Data Type' section is also visible.

## ២. សិក្សាតិ Constraints:

យើងអាចបង្កើត Constraints ដើម្បីធ្វើការការពារ និងទទួលបានការពារ ដោយត្រូវកំណត់ថា data ដែលត្រូវបញ្ចូលមានភាពត្រឹមត្រូវឡើងទៀត និងស្ថានភាពជាកេស្តីដែលត្រូវការ។ យើងអាចបង្កើត Constraints ជាបច្ចុប្បន្នក្នុង Columns ឬក្នុងពារមិនត្រូវបង្កើត Table ប្រកួតាបាន។ មាន Constraints ចំនួន 6 ប្រភេទខាងក្រោម ដែលយើងអាចបង្កើត Constraints ជាបច្ចុប្បន្នក្នុងការបង្កើត Columns បានដោយខាងក្រោម:

- > Primary Key: គឺសំរាប់កំណត់ Column មួយប្រចូល រក្សាទុកទិន្នន័យខាងក្រោមជានិច្ឆ័ន់ ដើម្បីបង្កើតក្នុង Row និងក្នុង Table ។ ចំពោះ Table មួយទាំងមាន Primary Key តែមួយប៉ុណ្ណោះ។

- > Foreign Key: គឺសំរាប់កំណត់ Column មួយច្បាស់ដែលត្រូវចូលរួមធ្វើការភ្លាម Relationship រវាង Tables ជាប្រើប្រាស់  
ផោយវាត្រូវ Reference ទៅកាន់ Primary Key តែមួយចំណោះៗ Tables ប្រចាំនាទមាន Foreign Keys ប្រចាំនាទដីរ  
ដើម្បីភ្លាម Relationship បានរហូតដល់ 63 Tables ដៃនៃត្រូវនៅក្នុង Database តែមួយ។
- > Allowing Null Values: គឺសំរាប់កំណត់ពេលបញ្ចូលកើតបាន ផោយនាថ្នាមបញ្ចូលកើតបាន មួយនាថ្នាមបញ្ចូល  
ទិន្នន័យចូលកើតបាន ឬអូកវា Blank ឬ Null ចោលកើតបាន ចំពោះ NOT NULL Constraint និងត្រូវតែប្រាប់  
ទិន្នន័យចូលនៅក្នុង Column ជានិច្ឆ័ន់។
- > Default: Default Constraints គឺសំរាប់ធ្វើការកំណត់នូវ តំបន់ដែលមានស្រាប់នៅក្នុង Column មានមួយ។
- > Check: វាតិចាយកកំណត់ Limit នៃ Range of values ដែលត្រូវបានបញ្ចូលនៅក្នុង Column មួយដោយផ្តល់ករណី Boolean  
expression ។
- > Unique: វាតិចាយកកំណត់ Limit នៃ Range of values ដែលត្រូវបានបញ្ចូលនៅក្នុង Column ឱ្យបញ្ជាក់ថា នៅក្នុង Table មួយនាទមាន  
Unique Constraints រហូតដល់ 249 ។

### 3. ការរៀបចំឡាយ Default Constraints:

Default Constraint គឺរាបកំណត់ជាតិលើមានមួយដែលត្រូវបានបញ្ចូលផោយស្ម័យប្រភើនៅក្នុង Row និងនៅពេល ដែលត្រូវ  
Data ត្រូវបានបញ្ចូលនៅក្នុង Column ។ យើងនាថ្នាម Define Default Constraint បានតែមួយចំណោះៗនៅក្នុង Column មួយ។  
ឧបារណ៍ខាងក្រោមបង្ហាញពីការបង្កើត Default Constraint ផោយកំណត់មែនទិន្នន័យទេ ("") នៅក្នុង Column Grade នៃ  
Person Table ។ ភាគីនេះនៅពេលដែលកំណត់ Default Constraint ប្រើប្រាស់គីត្រកំណត់ពេលវេលាដែលមិនអាចបញ្ចូលកើតបាន។  
ស្ថិតិថ្នាមបង្កើត Table មួយទៀតឈ្មោះ Enrollment ប្រើប្រាស់ជាពេលវេលាដែលកំណត់ Default Constraints  
ដូចខាងក្រោម:

Column Name	Data Type	Allow Nulls
ClassID	int	<input checked="" type="checkbox"/>
StudentID	nchar(10)	<input checked="" type="checkbox"/>
EnrolledDate	datetime	<input checked="" type="checkbox"/>
Grade	varchar(2)	<input checked="" type="checkbox"/>

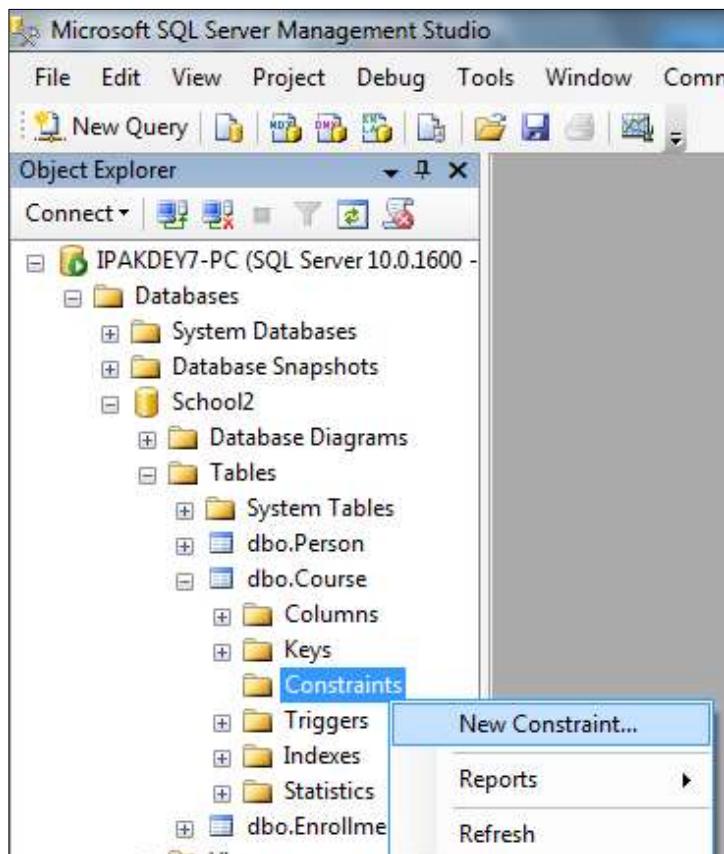
**Column Properties**

	<b>(General)</b>
(Name)	Grade
Allow Nulls	No
Data Type	varchar
Default Value or Binding	" "
Length	2

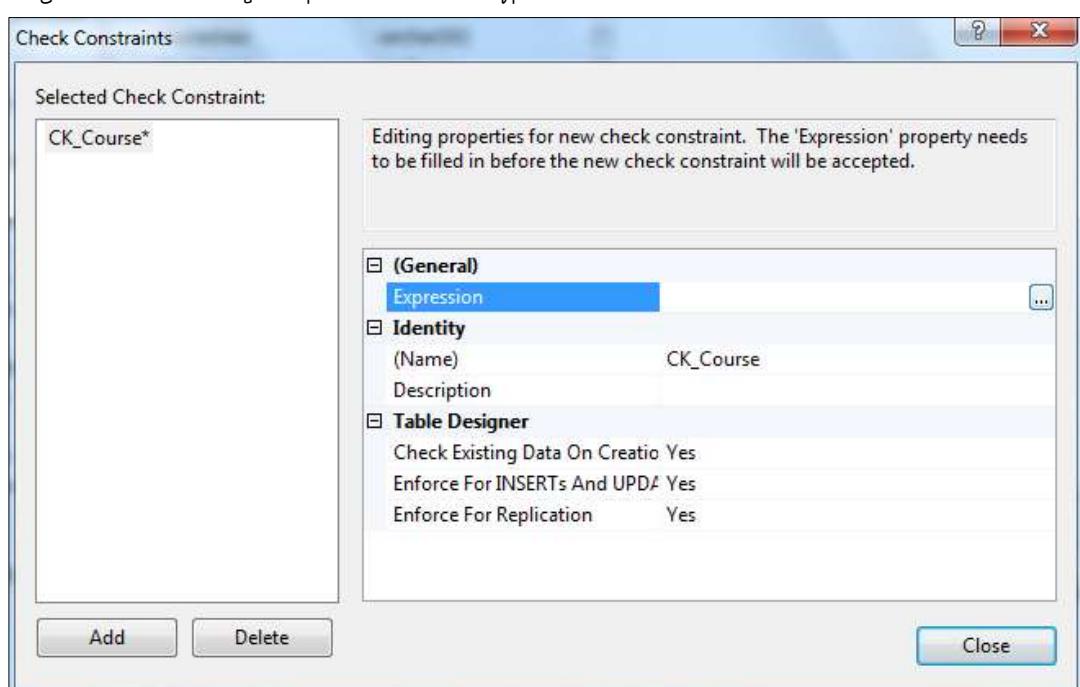
#### **4. ការគ្រប់គ្រាន់ Check Constraints:**

Check Constraint គឺជាការ Check Data មុនពេលផើល Record ត្រូវបាន Save ។ ប្រសិនបើ Data ផើលបញ្ចប់មិនត្រឹមត្រូវទេ Constraint ផើលបានកំពែនៅឡើង Record នឹងមិនត្រូវបាន Save ឡើងដូច្នេះ។

1. ចូច Expand Tables > dbo.Course > Constraints >
2. ចូច Mouse ស្វែរលើ Constraints ហើយការក្រួច New Constraint >

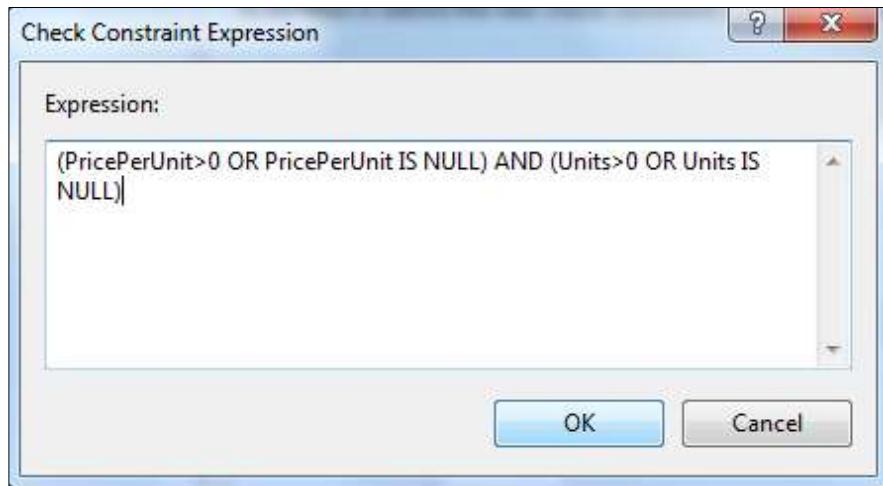


3. ត្រួច Expression ស្មូមចច្ចុ Button នៅខាងក្រោម >

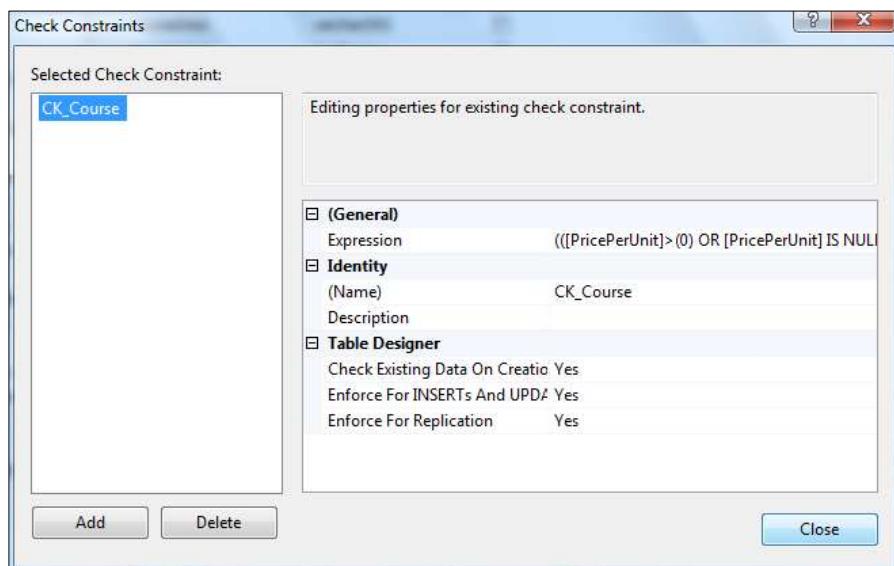


4. ស្មើសារលក្ខណៈ Check Constraint ចូលឡើងត្រួតព្រមទាំងត្រួតពីរ >  
 (PricePerUnit>0 OR PricePerUnit IS NULL) AND (Units>0 OR Units IS NULL).

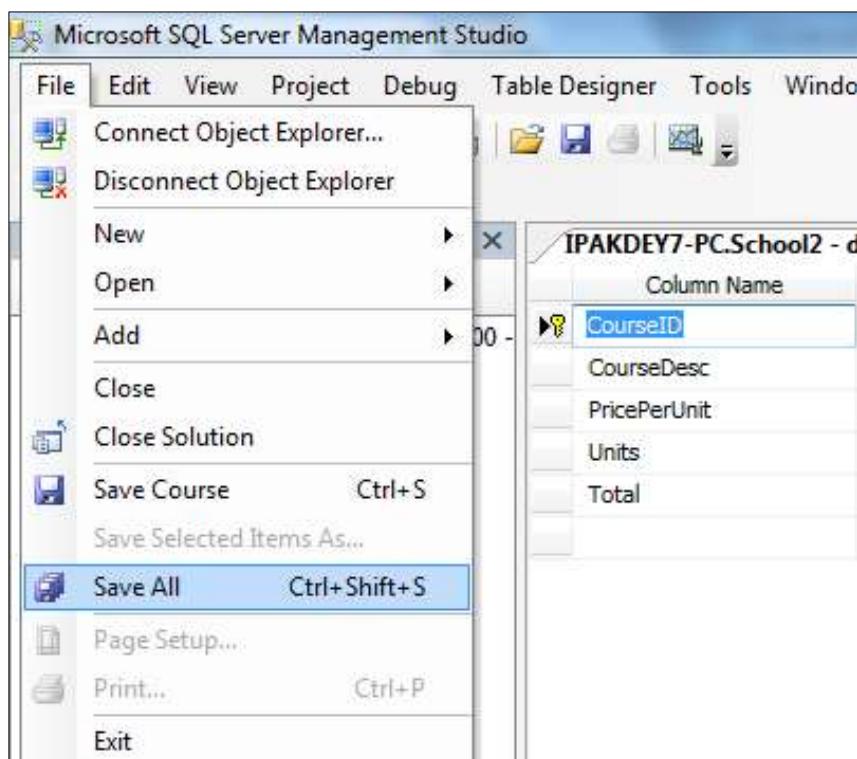
5. ធ្វើ OK Button >



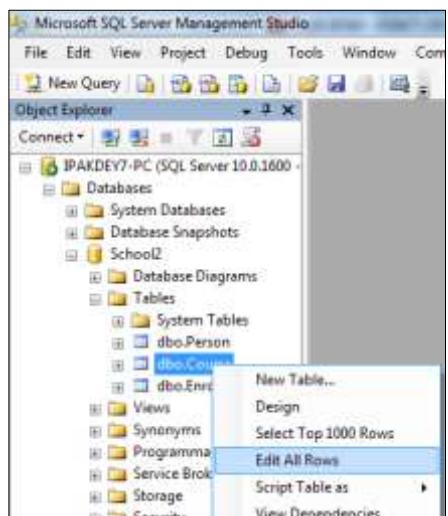
6. ធ្វើ Close Button >



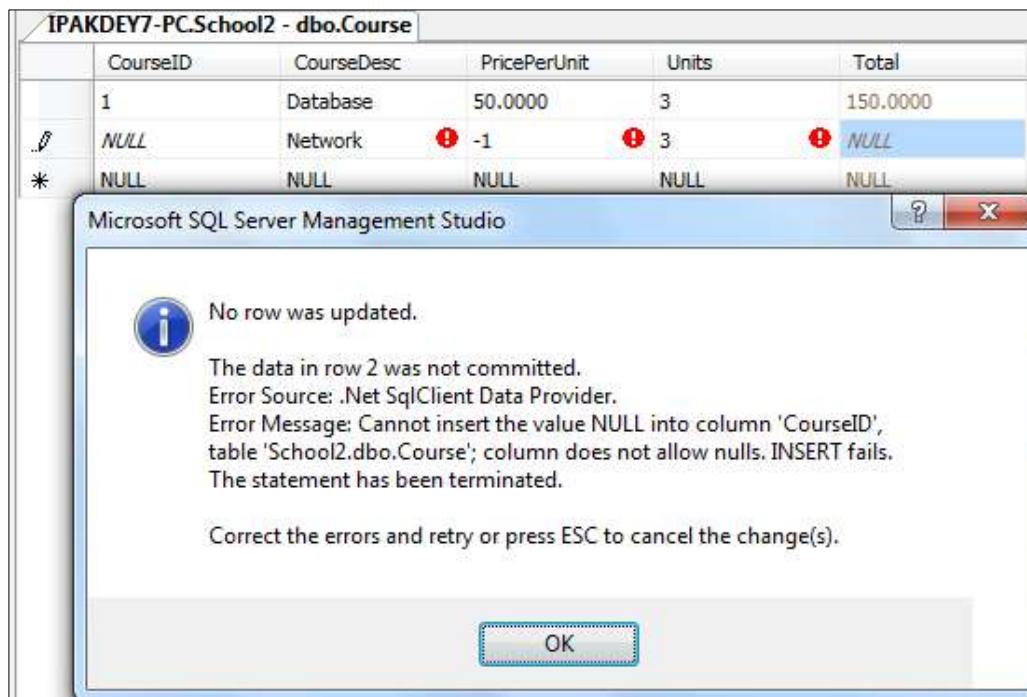
7. ធ្វើ File > Save All >



8. មុន Mouse ស្វែរឱ្យ dbo.Course បើយនិសបយក Edit All Rows >



9. បន្ទាប់មកសូមធ្វើការបញ្ចូលទិន្នន័យស្ថិតិថាមក្រាម នោះវានឹងបង្ហាញផ្លូវការ Error ឡើងស្ថិតិថាមក្រាម



## 5. គារពេញច្រោយ Triggers:

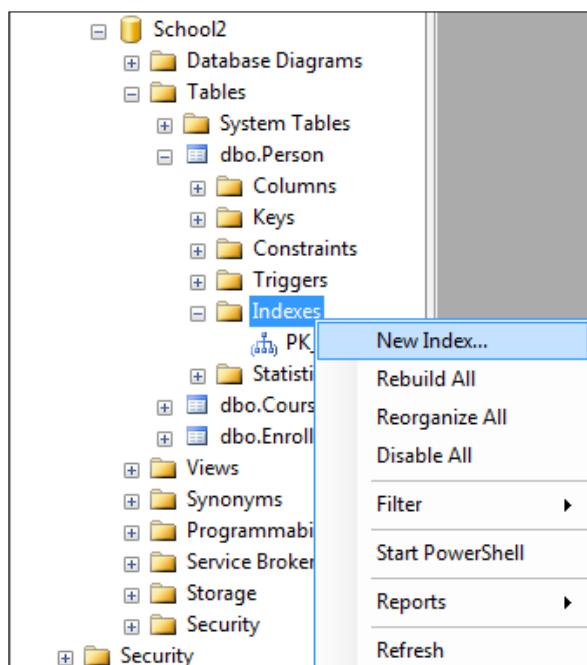
Trigger គឺជា Transact-SQL Procedure ដែលត្រូវបាន Execute នៅពេល INSERT, UPDATE, ឬ DELETE Statement ត្រួតពេញឡើង។ ឈើងមិនបានការពេញឡើងទេ នៅពេលពេញឡើង។ Trigger ដើម្បីរក្សាការកំណត់បញ្ជីលក្ខណៈពេលដំឡើង Business Rule មិនកំណត់ជាមួយនឹង Constraint បាន។ មាន Business Rules ខ្លះដើម្បីរក្សាការកំណត់ខ្លួនដោយពាណិជ្ជកម្ម rule ដែលទាក់ទងនៅនឹង Tables ដាច់ពីនា។ Trigger អាច Reference Values ចេញពី Columns, Tables ឬ Database ដែលមិនបានបង្កើត បាន។ ហើយយើងអាចធ្វើវាដាមួយនឹង Transact-SQL បានដូចខាងក្រោម។

ដើម្បីរក្សាការប្រើប្រាស់ Triggers នៅក្នុងការប្រើប្រាស់ T-SQL និង Transaction ជាមុនសិន។

## 6. គារបញ្ជី Indexes:

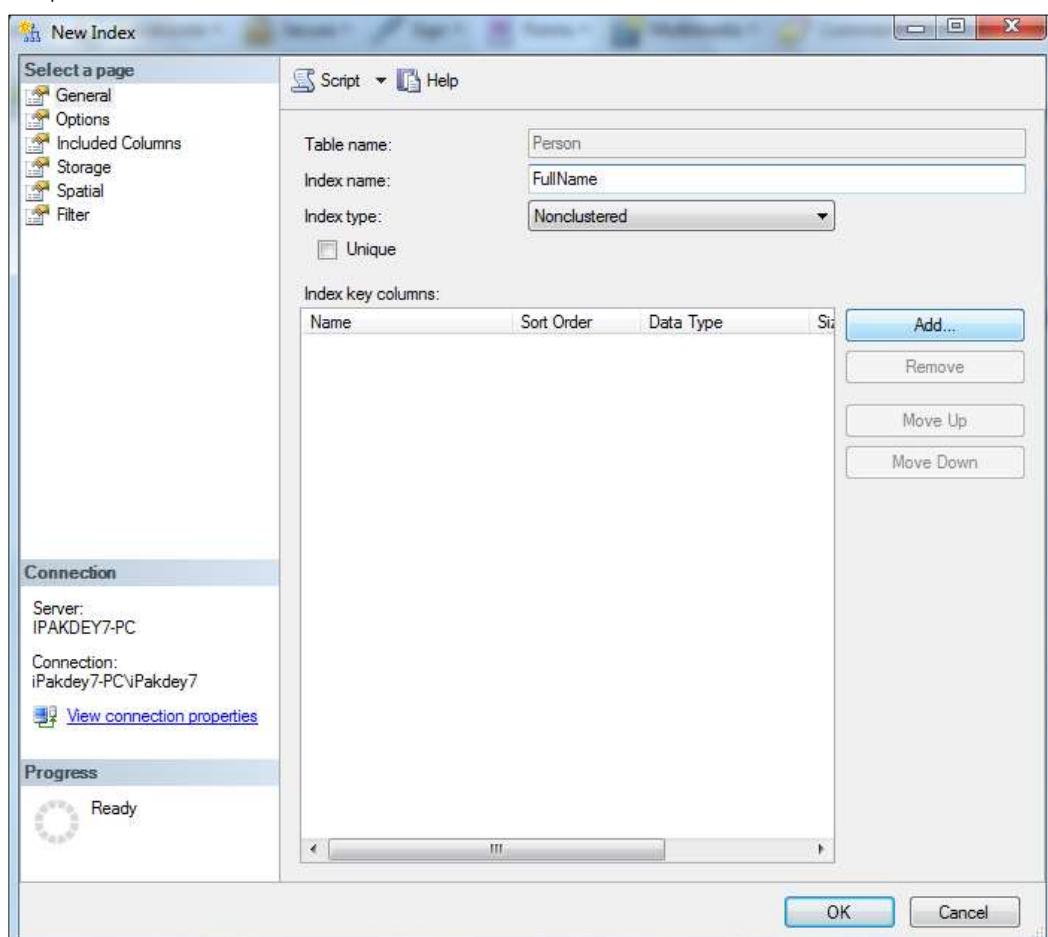
Indexes គឺជាបច្ចុប្បន្ននៃការបញ្ចីការងារដែលបង្ហាញពីការងារដែលបានបញ្ជីឡើង។ Indexes គឺជាបច្ចុប្បន្ននៃការបញ្ចីការងារដែលបង្ហាញពីការងារដែលបានបញ្ជីឡើង។

1. Expand dbo.Person >
2. មុន Mouse ស្វែរលើ Indexes ហើយចូលក្នុង New Index >



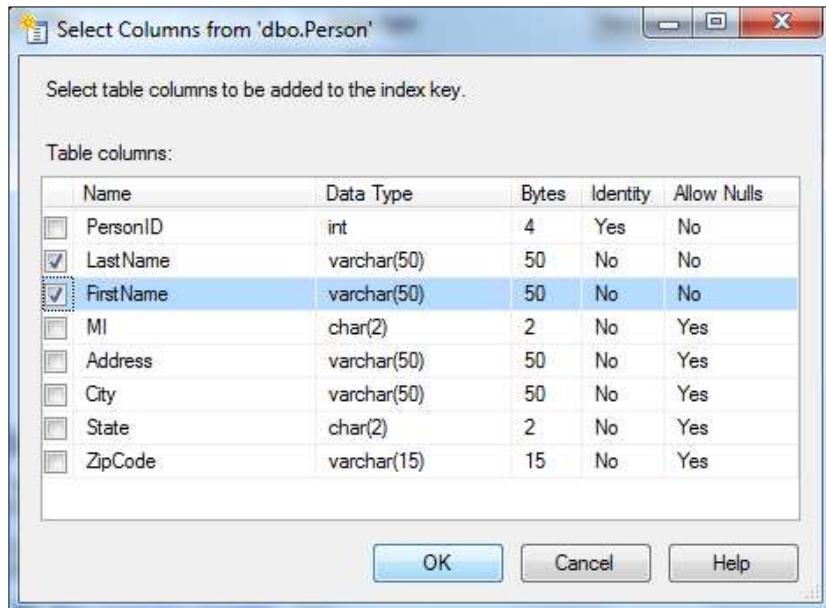
3. සිංහ Index name සුජක්ස් යොදා ( FullName ) >

4. ප්‍රාග් Add Button >

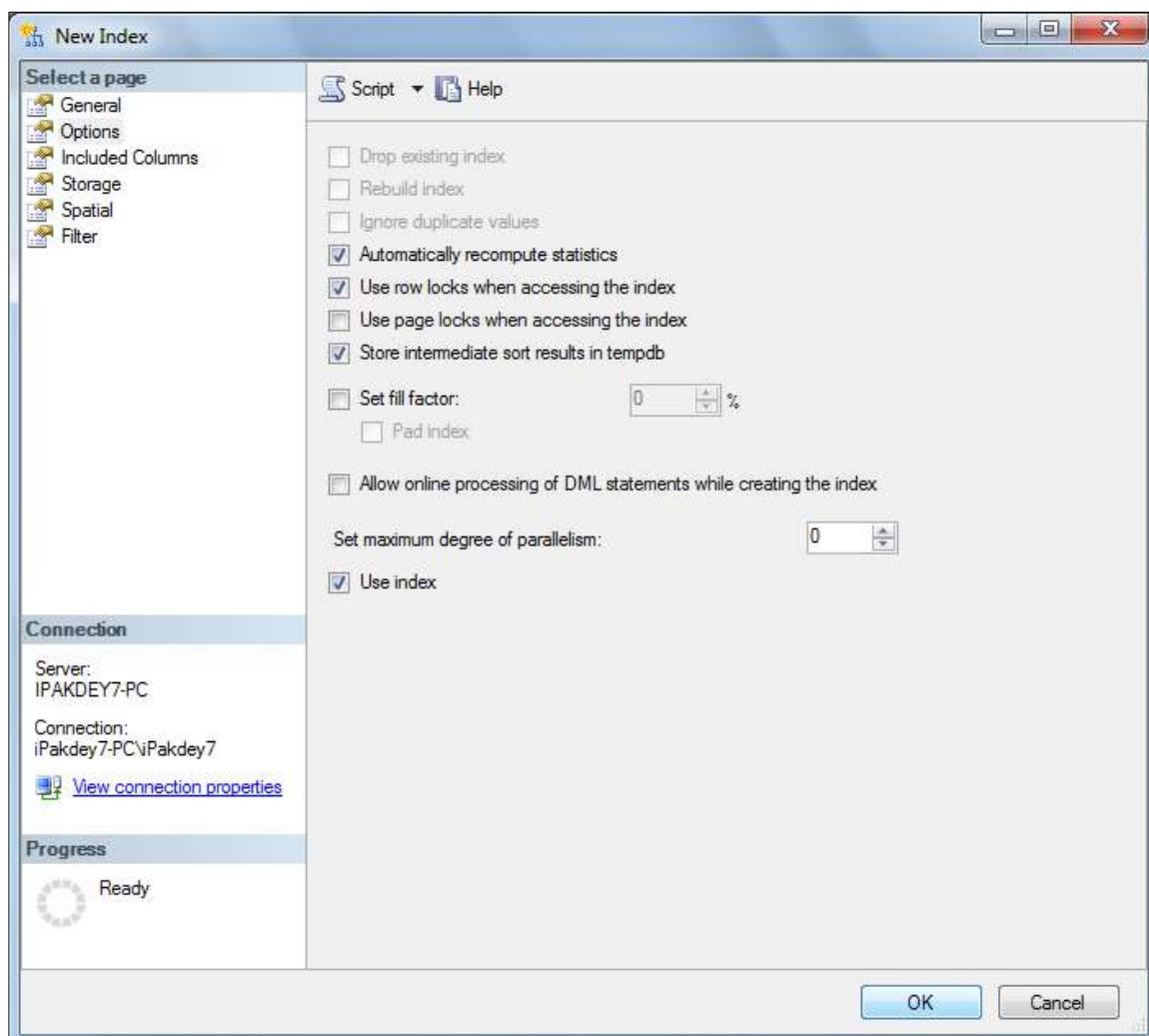


5. සුජ Tick යොදා LastName සිංහ FirstName >

6. OK Button >



7. ស្មើ Tick ឬក Option ដូចខាងក្រោម ហើយចូល OK Button



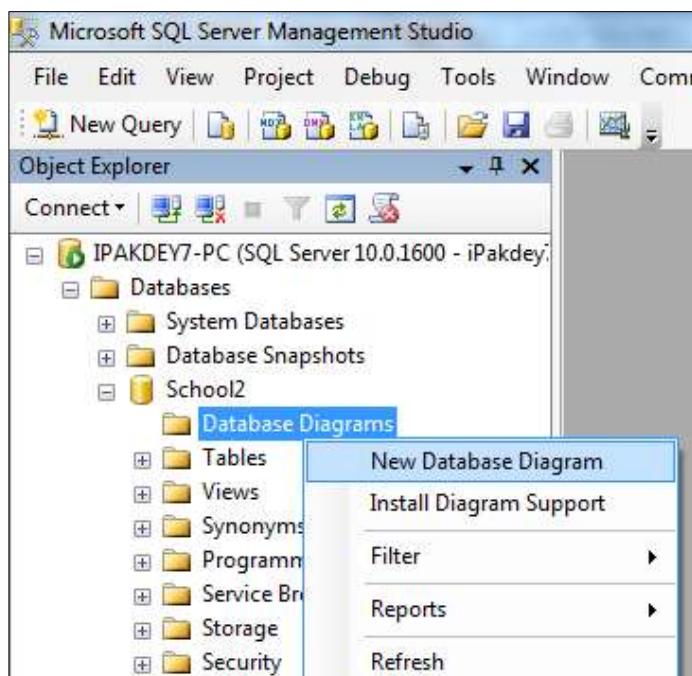
## 7. របៀបបង្កើត Database Diagrams:

1. ស្មើបញ្ជីត Table ធនធាន Class មួយចន្លែមឡើត ហើយផ្តល់នាំចូល >

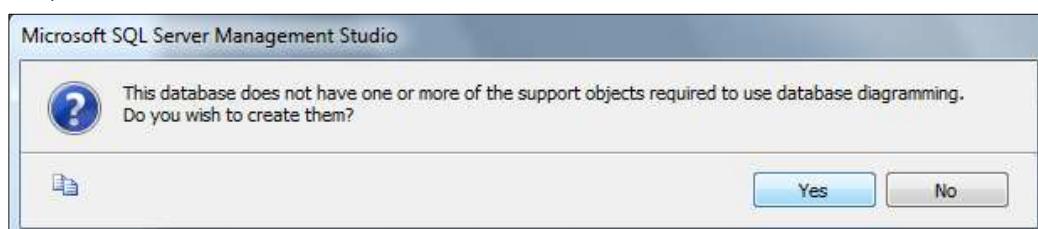
Column Name	Data Type	Allow Nulls
ClassID	int	<input type="checkbox"/>
CourseID	int	<input type="checkbox"/>
InstructorID	int	<input type="checkbox"/>
RoomID	int	<input type="checkbox"/>
DayofWeek	varchar(50)	<input type="checkbox"/>
StartTime	time(7)	<input type="checkbox"/>

2. ច్రికెట Expand School2 >

3. ច్రికెట Mouse ស్వాప్ కే Database Diagrams యక్కాగ్య New Database Diagram >

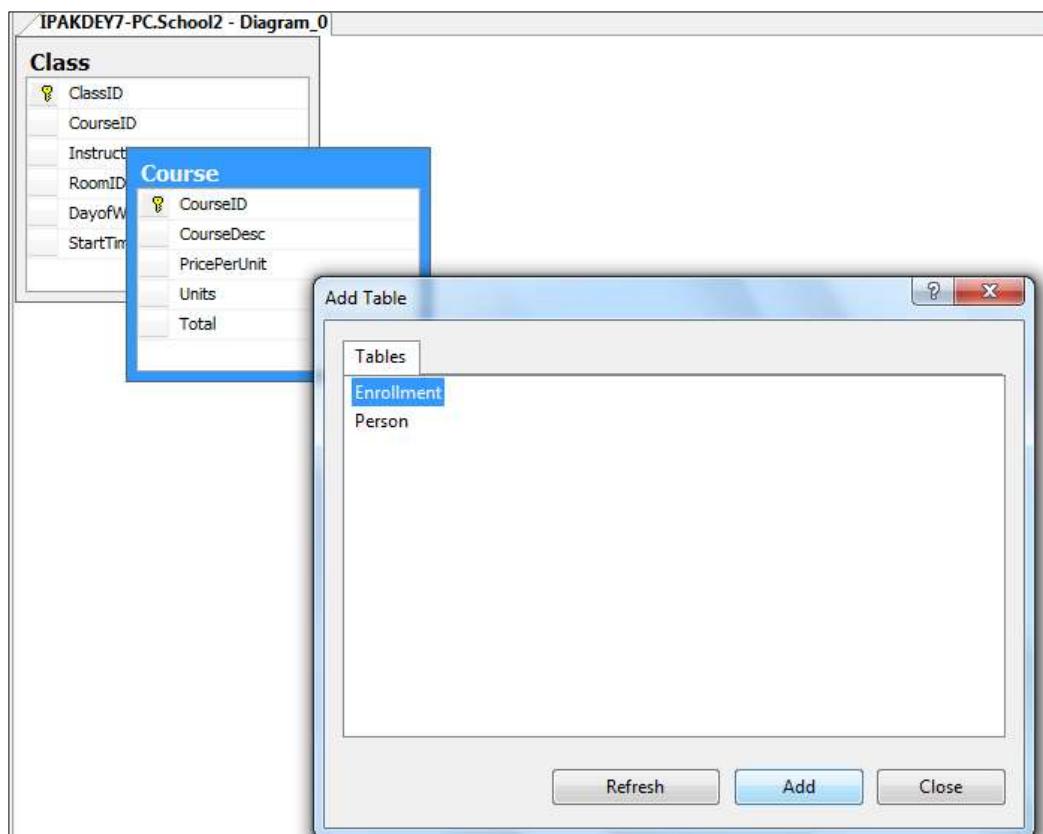


4. ច్రికెట Yes Button >

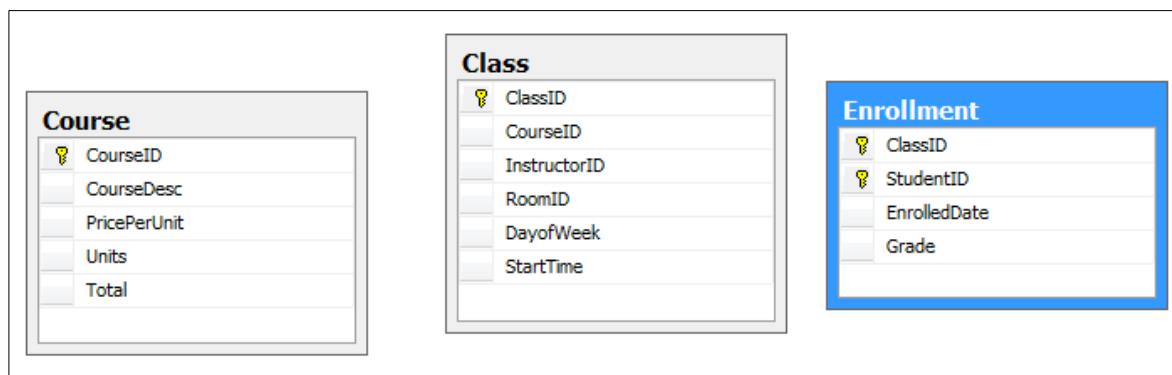


5. ស్వాప్ కే Add Table తోను 3 సీలబ్రాస్ ప్రైమర్ ను Class, Course లోనే Enrollment >

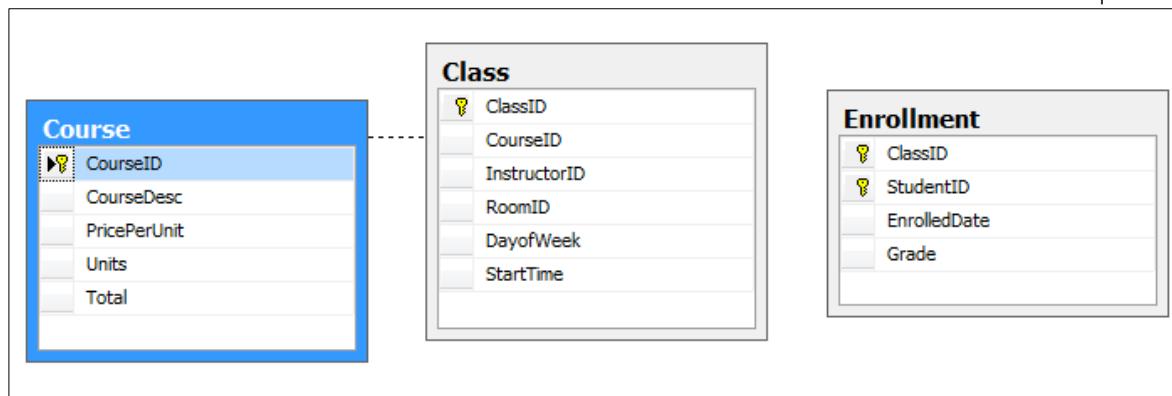
6. ច్రికెట Close Button >



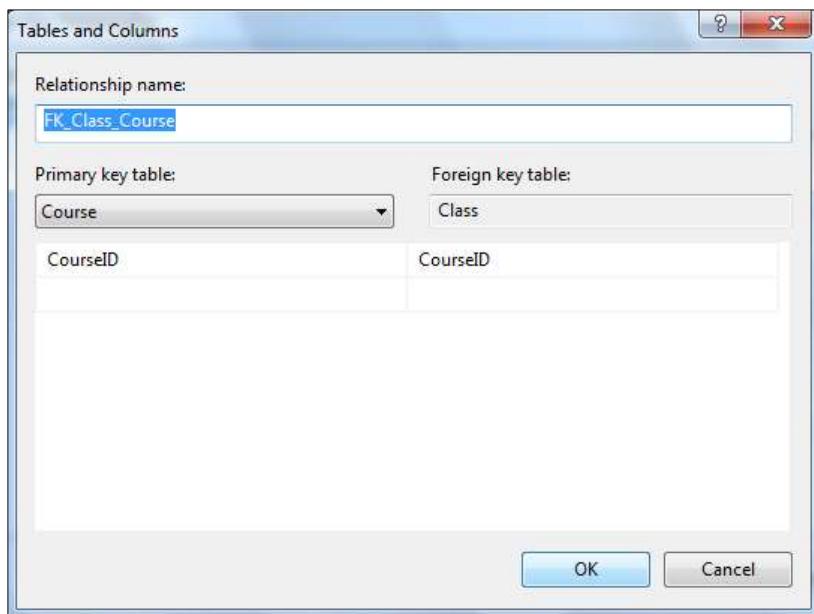
7. បញ្ជាក់ថា មិនមែនការរៀបចំបានឡើងទៅមួយច្បាស់ទៀត >



8. ធ្វើ Select លើ CourseID នហត្ថលេខាព័ត៌មាន និងពិនិត្យលាក់លើ CourseID ដែលស្នើសុំឡើងទៀត >

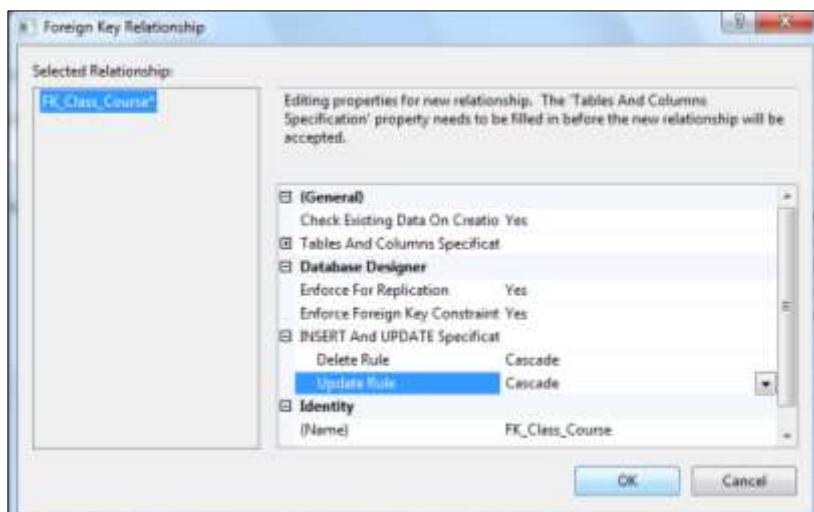


9. ធ្វើ OK Button >

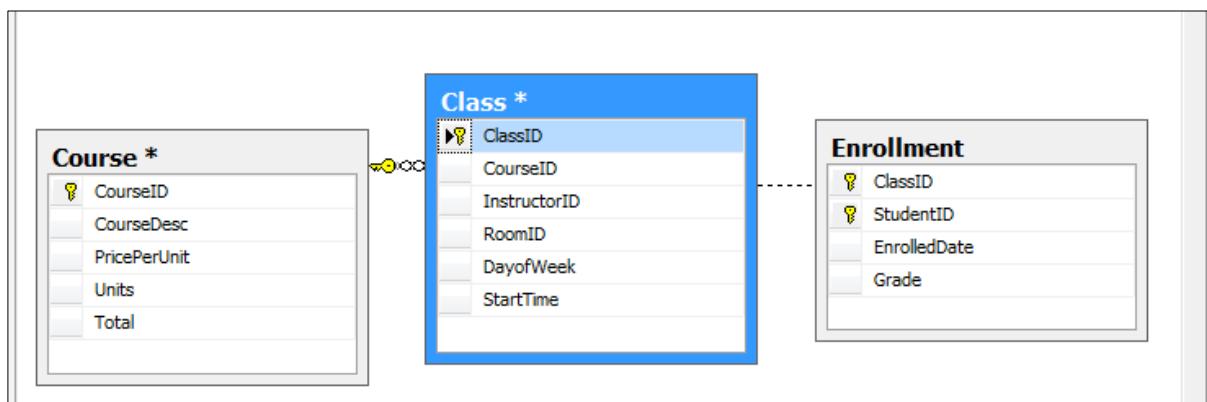


10. សូមជ្រើសរើសយក Cascade នៅត្រង់ Delete Rule និង Update Rule >

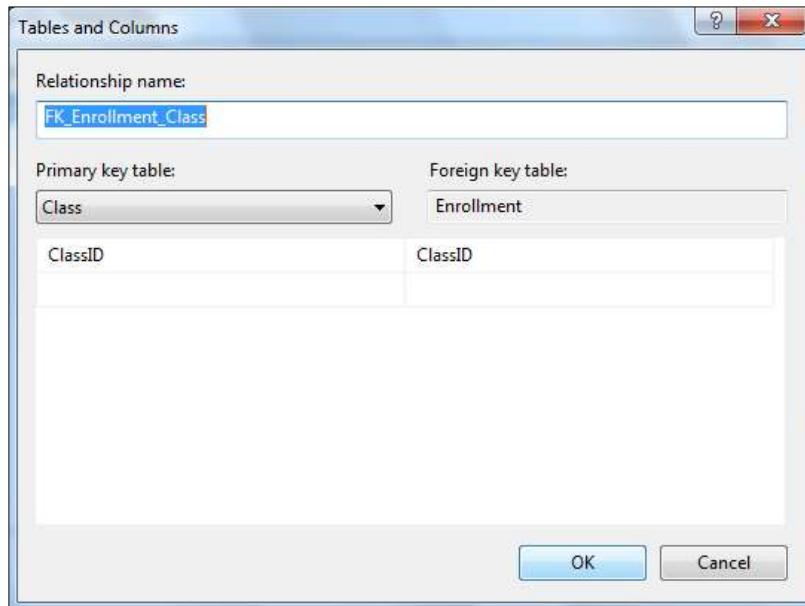
11. ចូល OK Button >



12. ចូល Select លើ ClassID របស់ Table Class ហើយចាប្រឡាតាំងហាក់លើ ClassID ដើលស្ថិតនៅត្រង់ Table Enrollment >

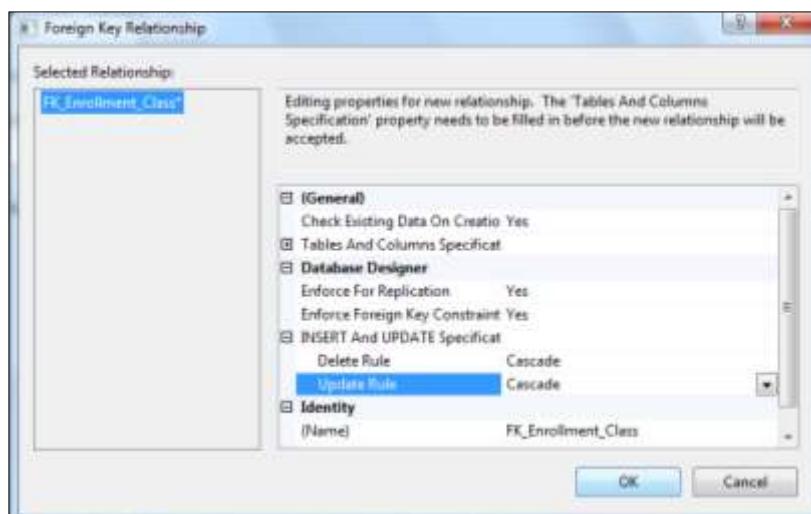


13. ចូល OK Button >

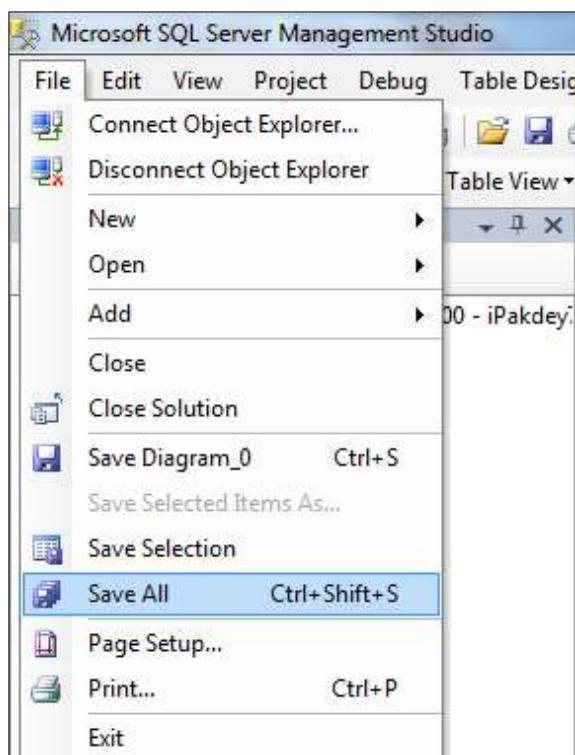


14. ស្មោគដើម្បីបន្រើបន្រាយក Cascade នៅត្រង់ Delete Rule និង Update Rule >

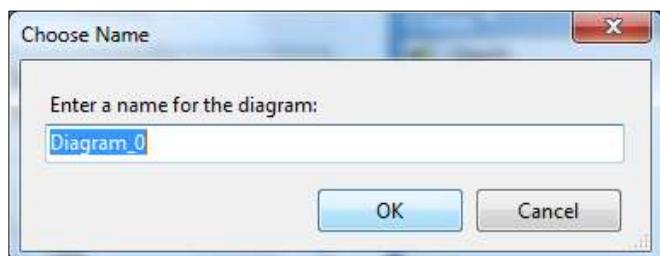
15. ធ្វើ OK Button >



16. ធ្វើ File > Save All >



17. ស្មមជាក់ល្អោះ ហើយចូច OK Button &gt;



18. ចូច Yes Button



## 8. វឌ្ឍនភាព:

- ចូរបង្កើត Database មួយល្អោះ Orders ហើយត្រួច Recovery Model របស់វាស្មមកំនត Simple ។
- ចូរបង្កើត Tables ចំនួន 2 ផើល Table ទី 1 ល្អោះ Customers និង ទី 2 គឺ CustomerOrders ។
- Table Customers តិះមាន Columns និង Data Type ដូចខាងក្រោម:

Name	Data Type	Length	Allow Nulls
CustomerID	int		No
LastName	varchar	30	No
FirstName	varchar	30	No
Address	varchar	50	Yes
City	varchar	20	Yes
State	char	2	Yes
ZipCode	varchar	11	Yes

> Columns CustomerID គឺជា Primary Key

2.2. Table CustomerOrders គឺមាន Columns និង Data Type ដូចខាងក្រោម:

Name	Data Type	Length	Allow Nulls
OrderID	int	4	No
CustomerID	int	4	No
OrderDate	smalldatetime	4	No
DeliveryDate	smalldatetime	4	Yes

> Columns OrderID គឺជា Primary Key

> ត្រូវកំណត់ Default Value នៅក្នុង Column OrderDate ដោយប្រកាសពេលបច្ចុប្បន្ន

> បង្កើត CHECK Constraint ដោយធានាថា DeliveryDate ត្រូវជាកាលបរិច្ឆេទ ក្រោម OrderDate

# លេខទូរទាត់ 6: Select និងបញ្ជី បញ្ជីបញ្ជី Queries

## 1. សិក្សាតិ Transact-SQL:

T-SQL គឺជាដឹងសារស្ថាប័នដែលផ្តល់ព័ត៌មានមួយចំនួន។ Structured Query Language (SQL) ជាមធ្យាការប្រើប្រាស់ដើម្បី Select និង Manipulate Data បានយ៉ាងមានប្រសិទ្ធភាព។ សារៈប្រយោជន៍របស់វាតិមានសមត្ថភាពក្នុងការ retrieve data ចេញពី Database ជាមធ្យាការប្រើប្រាស់ដើម្បីផ្តល់ព័ត៌មានស្ថាប័ន។

- > add, delete, និងកែសំណួលទិន្នន័យនៃក្នុង Database
- > បង្កើត Database Elements ដូចជា បង្កើត Database, Table, Indexes, និង Element ជាជំនួយទិន្នន័យ
- > គ្រប់ការ Access ទៅកាន់ Database Elements
- > Protect និង Enforce Rules នៃ ទិន្នន័យរបស់ Database ។

ក្នុងនោះវាមាន Capabilities ដំឡើងនូវប្រព័ន្ធដែលមានស្ថាប័ន។

1. Data Definition Language (DDL): អាចប្រើប្រាស់ជាបន្ទីរបង្កើត និងកែសំណួល Object នៃក្នុង Database របស់យើង។ មានសម្រាប់បង្កើតអាជីវការ CREATE, ALTER, និង DROP Object ក្នុង Database បាន។
2. Data Manipulation Language (DML): អាចប្រើប្រាស់ជាបន្ទីរបង្កើតនិងកែសំណួលទិន្នន័យស្ថាប័ន SELECT, INSERT, UPDATE, DELETE, និង TRUNCATE ទិន្នន័យជាជំនួយ។

លើសពីនេះទៅទៀត T-SQL គឺអាចប្រើប្រាស់ជាបន្ទីរបង្កើត Variables, Temporary Objects, និងការប្រើប្រាស់ Built-In System Function សំរាប់ប្រព័ន្ធប្រើប្រាស់ Aggregate Data, ជើងឯកជាបន្ទីរបង្កើតនិងកែសំណួលទិន្នន័យស្ថាប័ន។ T-SQL គឺមាន Features ស្របតាមគ្រប់បញ្ជាផ្ទៃនិង Programming Languages ដើម្បីក្នុងនោះរួមមាន Looping, Error Handling ។

## 2. សិក្សាតិ Schemas and Naming:

នៃក្នុង SQL Server 2008, Database Object ត្រូវកំណត់ឈ្មោះជាមាន 4 Parts :

**[Server\_name . [database\_name] . [schema\_name] . ]**

**database\_name . [schema\_name . ] object\_name**

> server\_name សំរាប់កំនត់ linked server name ឬ remote server name (បើកំនត់ blank មានន័យថា Current Server) ។

> database\_name សំរាប់កំនត់ database name (បើកំនត់ blank មានន័យថា Current Database) ។

> schema\_name សំរាប់កំនត់ schema name ឬ user name ដើម្បីការជាមួយនឹង Object (បើកំនត់ blank មានន័យថា Current user) ។

> Object\_name សំរាប់កំនត់ Object name ( Table, Database, E.) ។

Ex:

-- Use Northwind;

SELECT CompanyName FROM dbo.Customers;

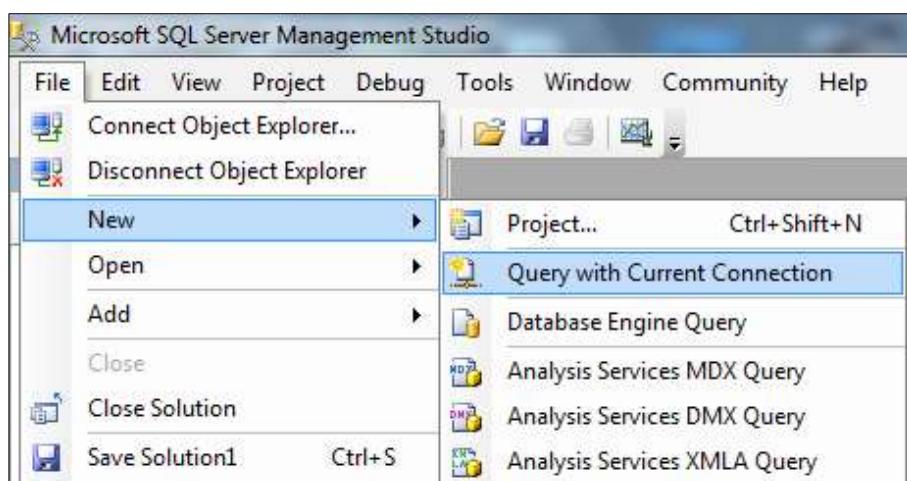
-- "Select CompanyName FROM Customers;" Will also work.

ធ្វើឱ្យធ្វើការប្រគល់នៅក្នុងមិនិច្ច:

1. File Menu >

2. New >

3. Query with Current Connection >



4. ក្នុងផ្ទាំង SQLQuery2.sql សូមសរសេរក្នុង TSQL ទៅតាមតម្លៃការ >

5. ចូច Execute Button >

6. នៅក្នុងផ្ទាំង Results នឹងបង្ហាញលទ្ធផលជូនិចខាងក្រោម

```
use Northwind;
select CustomerID from Customers;
```

CustomerID
ALFKI
ANATR
ANTON
AROUT
BERGS
BLAUS

### 3. Select Statement:

ជាមួយនឹង SELECT Statement គឺអាចប្រើប្រាស់បានយ៉ាងច្រើនរហូតដល់បញ្ជីខាងក្រោមគឺជាមូលដ្ឋានត្រូវបានប្រើប្រាស់រាជរដ្ឋាភិបាល: សាមញ្ញតម្លៃទីនៅក្នុងរបៀប

Syntax:

```
SELECT <col1, col2,...> FROM <table>
```

Example:

```
SELECT FirstName, LastName FROM dbo.Employees;
```

Output:

```
SELECT FirstName, LastName FROM dbo.Employees;
```

	FirstName	LastName
1	Nancy	Davolio
2	Andrew	Fuller
3	Janet	Leverling

Query Processor គឺ Ignores Tabs, Carriage Returns (Enter), និង Extra Spaces ក្នុងនោះ T-SQL Statement

នាមត្រូវបានសរសេរជាប្រើប្រាស់ Line មិនមែនតម្លៃយ៉ាង Single Line នោះទេ។

```

SELECT
    FirstName, LastName
FROM
    dbo.Employees;

```

ការសរស់រក្សាទុកដឹងលក្ខណៈជូនច្បាប់និងរក្សាទុកដឹងលក្ខណៈជូន

```
SELECT FirstName, LastName FROM dbo.Employees;
```

ចំពោះសញ្ញា Semicolo ; គឺអាម័បាត់នៅខាងចុងនៃ Statement បុច្ចិនជាក់កើបាន។

#### **4. Select All Columns:**

ឬឱចអាមេរិប្រាល់ សញ្ញាផ្លាយ (Asterisk) ដើម្បីធ្វើការ Select យក all Columns ចេញពី Table ។

Example:

```
SELECT * FROM dbo.Employees;
```

Output:

The screenshot shows the SQL Query window with the query: `SELECT * FROM dbo.Employees;`. Below it, the Results tab displays the data from the Employees table:

	EmployeeID	LastName	FirstName	Title	TitleOfCourtesy	BirthDate	HireDate
1	1	Davolio	Nancy	Sales Representative	Ms.	1948-12-08 00:00:00.000	1992-05-01 00:00:00.000
2	2	Fuller	Andrew	Vice President, Sales	Dr.	1952-02-19 00:00:00.000	1992-08-14 00:00:00.000
3	3	Leverling	Janet	Sales Representative	Ms.	1963-08-30 00:00:00.000	1992-04-01 00:00:00.000
4	4	Peacock	Margaret	Sales Representative	Mrs.	1937-09-19 00:00:00.000	1993-05-03 00:00:00.000

#### **5. Concatenating Columns:**

ឬឱចកើនអាមេរិការបង្កើត Columns ផ្តល់ខ្លួនជាយកទាញយកទិន្នន័យចេញពី Columns ដើម្បីត្រូវបានផ្តល់ជូន។

Example:

```
SELECT LastName + ', ' + FirstName FROM dbo.Employees;
```

Output:

```
SQLQuery2.sql - IP...PC\iPakdey7 (53)*
SELECT LastName + ', ' + FirstName FROM dbo.Employees;
```

	(No column name)
1	Davolio, Nancy
2	Fuller, Andrew
3	Leverling, Janet
4	Peacock, Margaret
5	Buchanan, Steven

#### **6. Naming Columns:**

បើដឹងកីវាទៀតការបង្កើត Columns ផ្លាស់ខ្លួនធាយកចិន្ទន៍យប់ពី Columns ដើម្បីត្រូវបានដាក់ឈ្មោះ។

Example:

```
SELECT LastName + ', ' + FirstName AS FullName
FROM dbo.Employees;
```

Output:

```
SQLQuery2.sql - IP...PC\iPakdey7 (53)*
SELECT LastName + ', ' + FirstName AS FullName
FROM dbo.Employees;
```

	FullName
1	Davolio , Nancy
2	Fuller , Andrew
3	Leverling , Janet
4	Peacock , Margaret

#### **7. Naming Columns:**

បើដឹងកីវាទៀតការបង្កើត Columns ផ្លាស់ខ្លួនបើយកតែប្រាប់រាយការណ៍ដែលបានដាក់ឈ្មោះ។

Example:

```
SELECT LastName + ', ' + FirstName AS FullName
FROM dbo.Employees;
```

## Output:

```
SQLQuery2.sql - IP...PC\iPakdey7 (53)*
SELECT LastName + ' , ' + FirstName AS FullName
FROM dbo.Employees;
```

Results Messages

	FullName
1	Davolio , Nancy
2	Fuller , Andrew
3	Leverling , Janet
4	Peacock , Margaret
5	Buchanan , Steven

ប្រសិនបើឯងមាន Table ច្បាស់គ្នា: Column ណាមួយមាន Space នៅក្នុងក្រោរប្រើប្រាស់ Square Brackets [ ] តាក់ហាំពីទៅវា ដើម្បីកំណត់មាន Error កើតឡើង។

## Example:

```
SELECT LastName + ' , ' + FirstName AS [Full Name]
FROM dbo.Employees;
```

## Output:

```
SQLQuery2.sql - IP...PC\iPakdey7 (53)*
SELECT LastName + ' , ' + FirstName AS [Full Name]
FROM dbo.Employees;
```

Results Messages

	Full Name
1	Davolio , Nancy
2	Fuller , Andrew
3	Leverling , Janet
4	Peacock , Margaret

ចំណោះ As Clause គឺមានលក្ខណៈជាឌភីអតិថិជនដែលបានបង្កើតឡើងជាប្រព័ន្ធដូចខាងក្រោម

**Ex:**

SQLQuery2.sql - IP...PC\iPakdey7 (53)\*

```

SELECT LastName + ' , ' + FirstName [Full Name]
FROM dbo.Employees;

```

	Full Name
1	Davolio , Nancy
2	Fuller , Andrew
3	Leverling , Janet
4	Peacock , Margaret

មានវិធីសារគ្រប់មុនក្នុងការរំភ័យឡើងថា Column ដែលយើងបានបង្កើតជាយូណីនឹងដឹង:

SQLQuery2.sql - IP...PC\iPakdey7 (53)\*

```

SELECT FullName = LastName + ' , ' + FirstName
From dbo.Employees;

```

	FullName
1	Davolio, Nancy
2	Fuller, Andrew
3	Leverling, Janet
4	Peacock, Margaret

#### **8. Using DISTINCT to Limit Values:**

វាទីវិធីសារគ្រប់ក្នុងការ Retrieve Data ដែលស្មូនត្រា តែមួយប៉ុណ្ណោះ។

Example:

The screenshot shows the SQL Query window with the query:

```
SQLQuery2.sql - IP...PC\iPakdey7 (53)*
SELECT DISTINCT Title FROM dbo.Employees;
```

The Results tab displays the output:

	Title
1	Inside Sales Coordinator
2	Sales Manager
3	Sales Representative
4	Vice President, Sales

#### **9. WHERE Clause:**

យើងប្រើប្រាស់ WHERE clause ដើម្បីត្រួតពិនិត្យការស្ថិតិយវត្ថុ លក្ខណៈផ្តល់ព័ត៌មានទាំងអស់

Example:

```
SELECT CompanyName, City
FROM dbo.Customers
WHERE City = 'Paris';
```

Output:

The screenshot shows the SQL Query window with the query:

```
SQLQuery2.sql - IP...PC\iPakdey7 (53)*
SELECT CompanyName, City
FROM dbo.Customers
WHERE City = 'Paris';
```

The Results tab displays the output:

	CompanyName	City
1	Paris spécialités	Paris
2	Spécialités du monde	Paris

#### **10. T-SQL Comparison Operators:**

T-SQL comparison Operator គឺនាមប្រើប្រាស់ក្នុងការ Build Queries ដោយនាមប្រើការ Search Records ទៅតាមតម្លៃការជាក់ស្អែងដែរ។

Operator	Description
=	Equal to
$\diamond$	Not equal to
>	Greater than
<	Less than
$\geq$	Greater than or equal to
$\leq$	Less than or equal to
!=	Not equal to
!<	Not less than
!>	Not greater than
[NOT] LIKE	Pattern matching a string with wildcards
BETWEEN <i>expr1</i> AND <i>expr2</i>	An inclusive range of values
IS [NOT] NULL	Check for null value
[NOT] IN ( <i>val1</i> , <i>val2</i> ...) -or- [NOT] IN ( <i>subquery</i> )	List matching
ANY (SOME)	Tests whether one or more rows in the result set of a subquery meet the specified condition
ALL	Tests whether all rows in the result set of a subquery meet the specified condition
[NOT] EXISTS	Tests whether a subquery returns any results

#### 11. LIKE Operators:

LIKE Operator គឺត្រូវប្រើប្រាស់ដើម្បីធ្វើការ Match String Values ។

Wildcard	Meaning
%	Any string of zero or more characters.
-	Any single character.
[]	Any single character within the specified range or set (LIKE [a-f] or LIKE [abcdef]).
[^]	Any single character not within the specified range or set (LIKE [^a - f] or LIKE [^abcdef]).

តាមខាងលាស់ខាងក្រោមបង្ហាញពីការ Query យក Customers ដែលមានឈ្មោះថាប់ផ្លូវជាមុននឹង "S"

Example:

```
SELECT CompanyName
FROM dbo.Customers
```

```
WHERE CompanyName LIKE 'S%';
```

Output:

```
SQLQuery2.sql - IP...PC\iPakdey7 (53)*
SELECT CompanyName
FROM dbo.Customers
WHERE CompanyName LIKE 'S%';

Results Messages
CompanyName
1 Santé Gourmet
2 Save-a-lot Markets
3 Seven Seas Imports
4 Simons bistro
```

តាមខ្សោយបាលវិធានក្រាមបង្ហាញពីការ Query យើង Customers ដែលមានឈ្មោះចុង ជាមួយនឹង "S"

Example:

```
SELECT CompanyName
FROM dbo.Customers
WHERE CompanyName LIKE '%S';
```

Output:

```
SQLQuery2.sql - IP...PC\iPakdey7 (53)*
SELECT CompanyName
FROM dbo.Customers
WHERE CompanyName LIKE '%S%';

Results Messages
CompanyName
1 Ana Trujillo Emparedados y helados
2 Blondesddsl père et fils
3 Bólido Comidas preparadas
4 Bottom-Dollar Markets
5 B's Beverages
6 Consolidated Holdings
7 Folies gourmandes
```

តាមខ្សោយបាលវិធានក្រាមបង្ហាញពីការ Query យើង Customers ដែលមានឈ្មោះត្រូវកណ្តាល ជាមួយនឹង "S"

Example:

```
SELECT CompanyName
FROM dbo.Customers
WHERE CompanyName LIKE '%S%';
```

Output:

The screenshot shows the SQL Query window with the following content:

```
SQLQuery2.sql - IP...PC\iPakdey7 (53)*
SELECT CompanyName
FROM dbo.Customers
WHERE CompanyName LIKE '%S%';
```

The Results tab displays the following data:

	CompanyName
1	Alfreds Futterkiste
2	Ana Trujillo Emparedados y helados
3	Berglunds snabbköp
4	Blauer See Delikatessen
5	Blondesddsl père et fils
6	Bólido Comidas preparadas

ចំណាំ: Underscore ( \_ ) វិញ្ញាបន្ទូរប្រើបាលដើម្បីធ្វើការកំណត់ចំណាំនូអក្សរពិតប្រាកដដែលត្រូវផ្តើកវា 'S\_' មានន័យ ថា ស្របចំយកទិន្នន័យដែលមានតូអក្សរនៅខាងមុខជាងក្រោម S និងតូអក្សទី 2 ដូរកំបាន។

តាមខាងលាប់ខាងក្រោមបង្ហាញពីការ Query យក CustomerID ដែលមានល្អ្អាបោះតូអក្សរតាំង "B" និងខាងច្ចាង "P" បុគ្គលិកដែលមានតូអក្សរចំណាំនៃ 3 តូទៀតជាដឹកកើតបាន។

Example:

```
SELECT CustomerID
FROM dbo.Customers
WHERE CustomerID LIKE 'B___P';
```

Output:

The screenshot shows the SQL Query window with the following content:

```
SQLQuery2.sql - IP...PC\iPakdey7 (53)*
SELECT CustomerID
FROM dbo.Customers
WHERE CustomerID LIKE 'B___P';
```

The Results tab displays the following data:

	CustomerID
1	BLONP
2	BONAP

លើកអាជីវក្រឹងប្រាស់ Square Brackets ដើម្បីធ្វើការកំណត់ List of Value ដើម្បីត្រូវ Match ។

តាមខាងក្រោមនេះអាសយដ្ឋានត្រួតពិនិត្យ Query យើង CustomerID ដើម្បីមានលេខាជាប់ផ្តើមជាមួយនឹង "FRAN"

និងបញ្ចប់ជាមួយនឹង "R" ឬ "K" ។

**Example:**

```
SELECT CustomerID
FROM dbo.Customers
WHERE CustomerID LIKE 'FRAN[RK]' ;
```

**Output:**

CustomerID
FRANK
FRANR

តាមខាងក្រោមនេះអាសយដ្ឋានត្រួតពិនិត្យ Query យើង CustomerID ដើម្បីមានលេខាជាប់ផ្តើមជាមួយនឹង "FRAN"

និងបញ្ចប់ជាមួយនឹងតុនក្រួចចាប់ពី "A" រហូតដល់ "S" ។

**Example:**

```
SELECT CustomerID
FROM dbo.Customers
WHERE CustomerID LIKE 'FRAN[A-S]' ;
```

**Output:**

```
SQLQuery2.sql - IP...PC\iPakdey7 (53)*
SELECT CustomerID
FROM dbo.Customers
WHERE CustomerID LIKE 'FRAN[A-S]';

Results Messages
```

	CustomerID
1	FRANK
2	FRANR
3	FRANS

តាមខាងលក្ខណៈខាងក្រោមបញ្ជាផ្ទុកការ Query យក CustomerID ដើម្បីមានឈ្មោះថាប់ផ្លូវជាមួយនឹង "FRAN"  
និងបញ្ជាប់ជាមួយនឹងតម្លៃរហូតដល់បាន លើកលេងគឺ "R"។

### **Example:**

```
SELECT CustomerID  
FROM dbo.Customers  
WHERE CustomerID LIKE 'FRAN[^R]' ;
```

## Output:

```
SQLQuery2.sql - IP...PC\iPakdey7 (53)*
SELECT CustomerID
FROM dbo.Customers
WHERE CustomerID LIKE 'FRAN[^R]';

Results Messages
CustomerID
1 FRANK
2 FRANS
```

## **12. BETWEEN Operators:**

BETWEEN Operator គឺត្រូវប្រើបាស់ដើម្បីធ្វើការ Search ទិន្នន័យតាមលក្ខខណ្ឌពីចន្ទោះអ្នប្បញ្ញចន្ទោះអ្នប្បយ នៅពេល ក្នុងនោះ AND ត្រូវបានប្រើបាស់នៅចន្ទោះ លក្ខខណ្ឌទាំងពីរដើម្បីបញ្ជាក់ថា ពីរចន្ទោះនេះ

## Example:

```
select LastName, FirstName, PostalCode
```

```
from dbo.Employees
where PostalCode between '98103' and '98999'
```

Output:

	Last Name	First Name	Postal Code
1	Davolio	Nancy	98122
2	Fuller	Andrew	98401
3	Callahan	Laura	98105

BETWEEN Operator អាចត្រូវបានប្រើប្រាស់ជាមួយនឹង Data Range ដោយជាគិត្យានៅក្នុង String ឬ Numeric ក៏បាន។

### 13. Using IS NULL:

IS NULL គឺប្រើប្រាស់ដើម្បីធ្វើការ Test វិកសារ null values នៅក្នុងទិន្នន័យ។

Example:

```
select LastName, FirstName, PostalCode
from dbo.Employees
where PostalCode between '98103' and '98999'
```

Output:

	Last Name	First Name	Region
1	Buchanan	Steven	NULL
2	Suyama	Michael	NULL
3	King	Robert	NULL
4	Dodsworth	Anne	NULL

### 14. Multiple Condition with AND, OR, and NOT:

និយោគប្រើប្រាស់ការប្រើប្រាស់ Boolean AND, OR និង NOT Operator ដើម្បីធ្វើការ Search ជាមួយនឹងលក្ខខណ្ឌ ស្ថិតិស្សាព្យាដំឡើង។

AND Operator គឺមានចាន់ Expression ទាំងអស់ដើម្បីត្រូវបានប្រើប្រាស់ជាការពិនិត្យផែនការ។

**Example:**

```
select LastName, City, PostalCode
from dbo.Employees
where City ='Seattle' and PostalCode like '9%'
```

**Output:**

SQLQuery1.sql - IP...PC\iPakdey7 (53)\*

```
select LastName, City, PostalCode
from dbo.Employees
where City ='Seattle' and PostalCode like '9%'
```

Results

	LastName	City	PostalCode
1	Davolio	Seattle	98122
2	Callahan	Seattle	98105

OR Operator គឺមានចារចំ Expression ណាមួយដើម្បីបង្ហាញក្នុងតិន្យ នៅលក្ខណៈរបស់វាពីតាម

**Example:**

```
select LastName, City, PostalCode
from dbo.Employees
where City ='Seattle' or PostalCode like '9%'
```

**Output:**

SQLQuery1.sql - IP...PC\iPakdey7 (53)\*

```
select LastName, City, PostalCode
from dbo.Employees
where City ='Seattle' or PostalCode like '9%'
```

Results

	LastName	City	PostalCode
1	Davolio	Seattle	98122
2	Fuller	Tacoma	98401
3	Leverling	Kirkland	98033
4	Peacock	Redmond	98052
5	Callahan	Seattle	98105

NOT Operator គឺត្រូវបានប្រើប្រាស់ដើម្បីធ្វើការស្របដឹងថ្មីថ្មីនៃលក្ខណៈរបស់ពីលក្ខណៈណាមួយដើម្បីបញ្ជាក់ថា

**Example:**

```
select LastName, City, PostalCode
from dbo.Employees
where City NOT LIKE 'Seattle'
```

**Output:**

The screenshot shows a SQL Server Management Studio window. The top pane displays a query: "SQLQuery1.sql - IP...PC\Pakdey7 (53)\*" containing the following SQL code:

```
from dbo.Employees
where City NOT LIKE 'Seattle'
```

The bottom pane shows the results grid with the following data:

	Last Name	City	Postal Code
1	Fuller	Tacoma	98401
2	Leverling	Kirkland	98033
3	Peacock	Redmond	98052
4	Buchanan	London	SW1 8JR
5	Suyama	London	EC2 7JR
6	King	London	RG1 9SP
7	Dodsworth	London	WG2 7LT

### **15. Operator Precedence:**

ଟଲି ପିଲି ଯେହିଁ କି କିମ୍ବା କିମ୍ବା କିମ୍ବା କିମ୍ବା କିମ୍ବା କିମ୍ବା :

- > NOT Operator ପିଲି କିମ୍ବା 1
- > AND Operator ପିଲି କିମ୍ବା 2
- > OR Operator ପିଲି କିମ୍ବା 3

#### **Example:**

```
select LastName, FirstName, City
from dbo.Employees
where LastName like '%s%'
and city not like 'Seattle'
```

#### **Output:**

SQLQuery1.sql - IP...PC\iPakdey7 (53)\*

```
use Northwind
select LastName, FirstName, City
from dbo.Employees
where LastName like '%ss%'
and city not like 'Seattle'
```

Results

	Last Name	First Name	City
1	Suyama	Michael	London
2	Dodsworth	Anne	London

#### **16. Using IN Operator:**

ឧបាទរណីខាងក្រោមនេះបង្ហាញពីការស្វែងចិត្តនៃយោងដោយប្រើប្រាស់ OR Operator ។

**Example:**

```
select CustomerID, Country
from dbo.Customers
where Country = 'France'
or Country = 'Spain'
```

**Output:**

SQLQuery1.sql - IP...PC\iPakdey7 (53)\*

```
select CustomerID, Country
from dbo.Customers
where Country = 'France'
or Country = 'Spain'
```

Results

	CustomerID	Country
1	BLONP	France
2	BOLID	Spain
3	BONAP	France
4	DUMON	France

ឧបាទរណីខាងក្រោមនេះបង្ហាញពីការស្វែងចិត្តនៃយោងដោយប្រើប្រាស់ IN Operator វិញ្ញាយដំឡើង។

**Example:**

```
select CustomerID, Country
```

```
from dbo.Customers
where Country in ('France', 'Spain')
```

Output:

```
SQLQuery1.sql - IP...PC\iPakdey7 (53)*
select CustomerID, Country
from dbo.Customers
where Country in ('France', 'Spain')

Results
CustomerID Country
1 BLONP France
2 BOLID Spain
3 BONAP France
```

ឧបារណ៍ខាងក្រោមនេះបង្ហាញពីការសម្រាប់តម្លៃនៃយោងប្រើប្រាស់ IN Operator ភ្លាមៗជាមួយនឹង SQL Statement ដើម្បីត្រូវបានបញ្ជាផ្ទាល់។

**Example:**

```
select CustomerID
from dbo.Customers
where CustomerID not IN(select CustomerID
                           from dbo.Orders);
```

```
SQLQuery1.sql - IP...PC\iPakdey7 (53)*
select CustomerID
from dbo.Customers
where CustomerID not IN(select CustomerID
                           from dbo.Orders)

Results
CustomerID
1 FISSA
2 PARIS
```

# មេរីលិ 7: សិក្សាបំនិត

## T-SQL ស្នើសុំ Queries

### 1. ការប្រព័ន្ធសំ ORDER BY:

ORDER BY clause ត្រូវបានប្រើប្រាស់ដើម្បីធ្វើការតាំងចូលក្នុង Sort Data នៅក្នុង Column ដោយតម្លៃការណ៍ ភាពប្រើនេះ ORDER BY ត្រូវជាក់នៅខាងចុងនៃ SELECT Statement ជានិច្ឆ័ន់ និងបន្ទាប់ពី WHERE clause ។ យើងអាចប្រើប្រាស់ Keyword ASC ឬ DESC តាមលំដាប់គឺ ឬ DESC ឬ DESC តាមលំដាប់មេរ។

**Example:**

```
select LastName, City
from dbo.Employees
order by City
```

**Output:**

The screenshot shows the SQL Server Management Studio interface. In the top-left pane, there is a tree view with a node labeled 'use Northwind'. Below it, under the 'use Northwind' node, is a 'select LastName, City from dbo.Employees order by City' query. In the bottom-right pane, there is a 'Results' tab showing a table with 9 rows of data. The columns are 'LastName' and 'City'. The data is as follows:

	Last Name	City
1	Leverling	Kirkland
2	Buchanan	London
3	Suyama	London
4	King	London
5	Dodsworth	London
6	Peacock	Redmond
7	Davolio	Seattle
8	Callahan	Seattle
9	Fuller	Tacoma

ឧបាណណ៍ខាងក្រោមបង្ហាញពីការ Sort តាមលំដាប់មេរ។

**Example:**

```
select LastName, City
from dbo.Employees
order by City DESC;
```

**Output:**

The screenshot shows a SQL query window titled "SQLQuery1.sql - IP...PC\iPakdey7 (53)\*". The query is:

```
use Northwind
select LastName, City
from dbo.Employees
order by City desc;
```

The results grid displays the following data:

	LastName	City
1	Fuller	Tacoma
2	Callahan	Seattle
3	Davolio	Seattle
4	Peacock	Redmond
5	Buchanan	London
6	Suyama	London
7	King	London
8	Dodsworth	London
9	Leverling	Kirkland

ដើម្បីធ្វើការ Sort ត្រូវបង្ហាញការ Sort Columns តិចិច្ចក្នុងក្រុមហ៊ុនប្រចាំប្រចាំថ្ងៃ , ដើម្បីពន្លាក់ការ Sort Columns ត្រូវបង្ហាញការ Sort ASC ឬ DESC ហើយមធ្វើការ Sort ត្រូវបង្ហាញការ Sort Columns តិចិច្ចក្នុងក្រុមហ៊ុនប្រចាំប្រចាំថ្ងៃ

#### Example:

```
select LastName, City
from dbo.Employees
order by City DESC, LastName ASC;
```

#### Output:

The screenshot shows a SQL query window titled "SQLQuery1.sql - IP...PC\iPakdey7 (53)\*". The query is:

```
select LastName, City
from dbo.Employees
order by City DESC, LastName ASC;
```

The results grid displays the same data as the previous screenshot:

	LastName	City
1	Fuller	Tacoma
2	Callahan	Seattle
3	Davolio	Seattle
4	Peacock	Redmond
5	Buchanan	London
6	Dodsworth	London
7	King	London
8	Suyama	London
9	Leverling	Kirkland

#### 2. Sorting with Expressions:

##### Example:

```
select LastName
from dbo.Employees
order by LEN(LastName);
```

Output:

The screenshot shows a SQL query window titled "SQLQuery1.sql - IP...PC\iPakdey7 (53)\*". The query is:

```
select LastName
from dbo.Employees
order by LEN(LastName);
```

The results grid shows the following data:

	LastName
1	King
2	Fuller
3	Suyama
4	Peacock

### **3. Aggregate Functions with Group By Clause:**

គោលបំនងសំខាន់ នៃ Relational Database គឺធ្វើការ Summarize ឬ Group Data ជាក្រុមពី ដើម្បីអាយុយ្មោះបញ្ជាក់ មិន និងប្រើប្រាស់ជាមួយ Aggregate Function ត្រូវតាមនាមឈ្មោះ។  
ឧបករាលនេះជា Functions សំខាន់ៗដើម្បីប្រើប្រាស់ជាតីកញ្ចប់៖

Function	Description
COUNT	Counts the number of rows in the aggregate query.
COUNT_BIG	Same as COUNT, but returns a bigint data type.
SUM	Adds up (sums) column values. Only works with numeric columns.
AVG	Averages column values. Only works with numeric columns.
MIN	Returns the lowest column value in an aggregate query.
MAX	Returns the largest column value in an aggregate query.

ឧបាទរណ៍ឧបករាលនេះជាប្រើប្រាស់ Count() ដើម្បីធ្វើការរាបចិននៅលើ

**Example:**

```
select COUNT(*)
FROM dbo.Employees
```

```
SQLQuery1.sql - IP...PC\iPakdey7 (53)*
select COUNT(*)
FROM dbo.Employees
```

(No column name)
1 9

#### **4. Counting Column:**

**Example:**

```
select COUNT(*) AS NumberEmployees,
count(Region) As NumRegion
FROM dbo.Employees
```

**Output:**

```
SQLQuery1.sql - IP...PC\iPakdey7 (53)*
use Northwind
select COUNT(*) AS NumberEmployees,
count(Region) As NumRegion
FROM dbo.Employees
```

NumberEmployees	NumRegion
1 9	5

#### **5. Counting with WHERE Clause:**

**Example:**

```
select COUNT(*) AS NumberEmployeeSeattle,
FROM dbo.Employees
Where City = 'Seattle'
```

**Output:**

```
SQLQuery1.sql - IP...PC\iPakdey7 (53)*
use Northwind
select COUNT(*) AS NumberEmployeeSeattle
FROM dbo.Employees
Where City = 'Seattle'
```

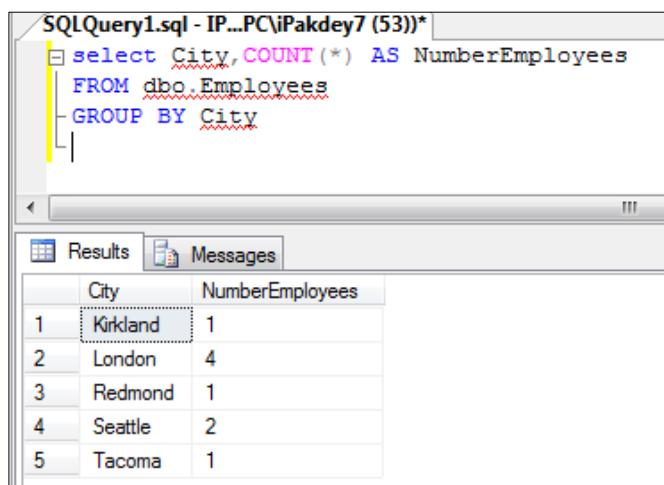
NumberEmployeeSeattle
1 2

## **6. Using GROUP BY:**

**Example:**

```
select City,COUNT(*) AS NumberEmployees
FROM dbo.Employees
GROUP BY City
```

**Output:**



The screenshot shows the SQL Query window with the following query:

```
SQLQuery1.sql - IP...PC\iPakdey7 (53)*
select City,COUNT(*) AS NumberEmployees
FROM dbo.Employees
GROUP BY City
```

The Results tab displays the output as a table:

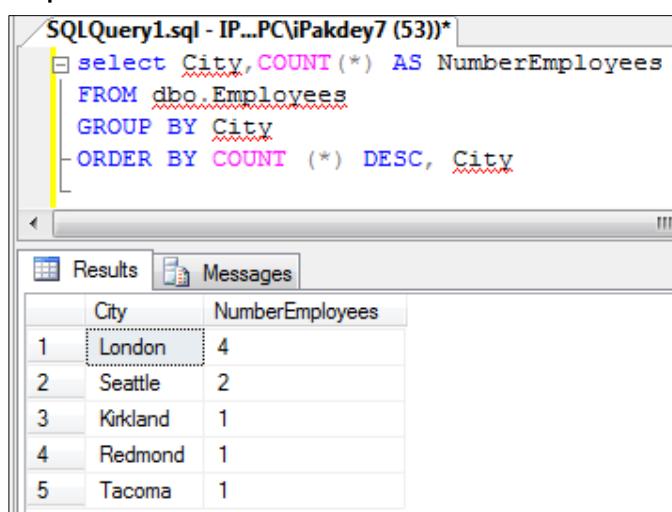
	City	NumberEmployees
1	Kirkland	1
2	London	4
3	Redmond	1
4	Seattle	2
5	Tacoma	1

## **7. Using ORDER BY with GROUP BY:**

**Example:**

```
select City,COUNT(*) AS NumberEmployees
FROM dbo.Employees
GROUP BY City
ORDER BY COUNT(*) DESC, City
```

**Output:**



The screenshot shows the SQL Query window with the following query:

```
SQLQuery1.sql - IP...PC\iPakdey7 (53)*
select City,COUNT(*) AS NumberEmployees
FROM dbo.Employees
GROUP BY City
ORDER BY COUNT(*) DESC, City
```

The Results tab displays the output as a table:

	City	NumberEmployees
1	London	4
2	Seattle	2
3	Kirkland	1
4	Redmond	1
5	Tacoma	1

## **8. Using HAVING with GROUP BY:**

ភាគធ្វើនេះ យើងតែអាចប្រើបាស់ HAVING ដើម្បីជាក់លក្ខខណ្ឌដំនឹកនៅពេលប្រើបាស់ GROUP BY ពីរព្រមទាំង WHERE Clause តើនឹងដំឡើងព័ត៌មានពេលដើរ Group ចូលគ្នា ប៉ុន្តែចំណោះ HAVING Clause វិញ ដំឡើងព័ត៌មានពេលដើរ Group ចូលគ្នា។

**Example:**

```
select City,COUNT(*) AS NumberEmployees
FROM dbo.Employees
GROUP BY City
HAVING COUNT (*) > 1
ORDER BY COUNT (*) DESC, City
```

**Output:**

	City	NumberEmployees
1	London	4
2	Seattle	2

**9. TOP Values Queries:****Example:**

```
select TOP 3
    City, count(*) as Numemployees
from dbo.Employees
group by City
order by count(*) DESC;
```

**Output:**

The screenshot shows a SQL query window titled "SQLQuery1.sql - IP...PC\iPakdey7 (53)". The query is:

```
select TOP 3
    City, count(*) as Numemployees
from dbo.Employees
group by City
order by count(*) DESC;
```

The results grid has two columns: "City" and "Numemployees". The data is:

	City	Numemployees
1	London	4
2	Seattle	2
3	Redmond	1

នឹងមានលទ្ធផលដូចខាងក្រោមដែលបញ្ជាក់ថា TOP 3 គឺជាអតិថជ្រើនប៉ុណ្ណោះទៅបានដោយប្រើប្រាស់ WITH TIES Clause.

#### Example:

```
select TOP 3 WITH TIES
    City, count(*) as Numemployees
from dbo.Employees
group by City
order by count(*) DESC;
```

#### Output:

The screenshot shows a SQL query window titled "SQLQuery1.sql - I...-PC\iPakdey7 (53)". The query is identical to the previous one:

```
select TOP 3 WITH TIES
    City, count(*) as Numemployees
from dbo.Employees
group by City
order by count(*) DESC;
```

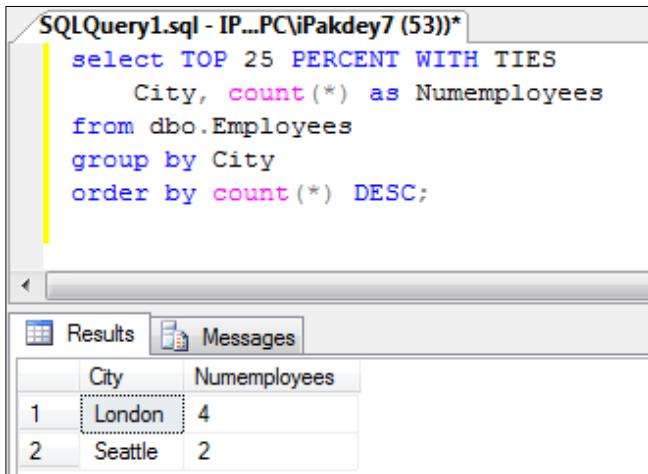
The results grid has two columns: "City" and "Numemployees". The data is:

	City	Numemployees
1	London	4
2	Seattle	2
3	Tacoma	1
4	Redmond	1
5	Kirkland	1

TOP 3 ត្រូវបានដោយប្រើប្រាស់ជាប្រភេទ Percent ដែលបានដោឡើងដោយប្រើប្រាស់ WITH TIES Clause.

#### Example:

```
select TOP 25 PERCENT WITH TIES
    City, count(*) as Numemployees
from dbo.Employees
group by City
order by count(*) DESC;
```

**Output:**


```
SQLQuery1.sql - IP...PC\iPakdey7 (53)*
select TOP 25 PERCENT WITH TIES
    City, count(*) as Numemployees
from dbo.Employees
group by City
order by count(*) DESC;
```

	City	Numemployees
1	London	4
2	Seattle	2

**10. Joining Tables:**

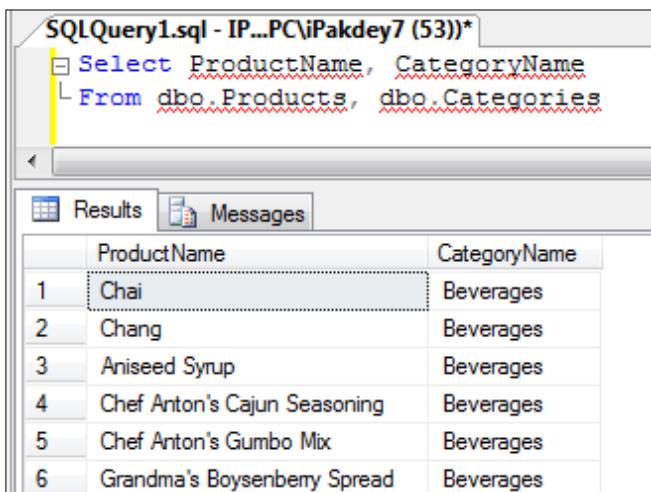
Joining Tables គឺជាការសមសរក្សាន់ T-SQL ដើម្បីធ្វើការស្របចំណួនឱ្យបញ្ជាពី Tables ជាថ្មីនៃផែលមាន Relationship

ជាមួយគ្នា។

ឧអក្សរាណនេះបង្ហាញពីការ Select Data ចេញពី Relationship Table ជាប្រភេទ Cross Join

**Example:**

```
Select ProductName, CategoryName
From dbo.Products, dbo.Categories
```

**Output:**


```
SQLQuery1.sql - IP...PC\iPakdey7 (53)*
Select ProductName, CategoryName
From dbo.Products, dbo.Categories
```

	ProductName	CategoryName
1	Chai	Beverages
2	Chang	Beverages
3	Aniseed Syrup	Beverages
4	Chef Anton's Cajun Seasoning	Beverages
5	Chef Anton's Gumbo Mix	Beverages
6	Grandma's Boysenberry Spread	Beverages

**10. Joining Tables:**

Joining Tables គឺជាការសមសរក្សាន់ T-SQL ដើម្បីធ្វើការស្របចំណួនឱ្យបញ្ជាពី Tables ជាថ្មីនៃផែលមាន Relationship

ជាមួយគ្នា។

ឧអក្សរាណនេះបង្ហាញពីការ Select Data ចេញពី Relationship Table ជាប្រភេទ Cross Join

**Example:**

```
Select ProductName, CategoryName
From dbo.Products, dbo.Categories
```

**Output:**

The screenshot shows a SQL query window titled "SQLQuery1.sql - IP...PC\iPakdey7 (53)\*". The query is:

```
SQLQuery1.sql - IP...PC\iPakdey7 (53)*
Select ProductName, CategoryName
From dbo.Products, dbo.Categories
```

The results grid has two columns: "ProductName" and "CategoryName". The data is:

	ProductName	CategoryName
1	Chai	Beverages
2	Chang	Beverages
3	Aniseed Syrup	Beverages
4	Chef Anton's Cajun Seasoning	Beverages
5	Chef Anton's Gumbo Mix	Beverages

### **11. Use of Keys in Joining:**

ជាមួយនឹងការសរស់សំរាប់ WHERE Clause បើអ្នកមានជីវិការ Select យោង Tables 2 ដើម្បាន Relationship ជាមួយគ្នា។

#### **Example:**

```
Select ProductName, CategoryName
From dbo.Products, dbo.Categories
Where Products.CategoryID = Categories.CategoryID;
```

#### **Output:**

The screenshot shows a SQL query window titled "SQLQuery1.sql - IP...PC\iPakdey7 (53)\*". The query is:

```
SQLQuery1.sql - IP...PC\iPakdey7 (53)*
Select ProductName, CategoryName
From dbo.Products, dbo.Categories
Where Products.CategoryID = Categories.CategoryID;
```

The results grid has two columns: "ProductName" and "CategoryName". The data is:

	ProductName	CategoryName
1	Chai	Beverages
2	Chang	Beverages
3	Aniseed Syrup	Condiments
4	Chef Anton's Cajun Seasoning	Condiments
5	Chef Anton's Gumbo Mix	Condiments

### **12. Inner Join:**

Inner Join គឺជា Join រវាង Relationship Tables ដោយយកទិន្នន័យណាដែល Match Criteria ត្រូវតែប៉ុណ្ណោះ។

#### **Example:**

```
Select Products.ProductName, Categories.CategoryName
From dbo.Products INNER JOIN dbo.Categories
ON Products.CategoryID = Categories.CategoryID;
```

#### **Output:**

The screenshot shows a SQL query window titled "SQLQuery1.sql - IP...PC\iPakdey7 (53)\*". The query is:

```
use Northwind
Select Products.ProductName, Categories.CategoryName
From dbo.Products INNER JOIN dbo.Categories
ON Products.CategoryID = Categories.CategoryID;
```

The results grid has columns "ProductName" and "CategoryName". The data is:

	ProductName	CategoryName
1	Chai	Beverages
2	Chang	Beverages
3	Aniseed Syrup	Condiments
4	Chef Anton's Cajun Seasoning	Condiments
5	Chef Anton's Gumbo Mix	Condiments
6	Grandma's Boysenberry Spread	Condiments

### **13. Simple Inner Join:**

**Example:**

```
Select dbo.Products.ProductName,
dbo.Categories.CategoryName,
dbo.Products.UnitPrice
from dbo.Products INNER JOIN dbo.Categories
    on dbo.Products.CategoryID = dbo.Categories.CategoryID
where UnitPrice > 50
order by ProductName;
```

**Output:**

The screenshot shows a SQL query window titled "SQLQuery1.sql - IP...PC\iPakdey7 (53)\*". The query is:

```
Select dbo.Products.ProductName,
dbo.Categories.CategoryName,
dbo.Products.UnitPrice
from dbo.Products INNER JOIN dbo.Categories
    on dbo.Products.CategoryID = dbo.Categories.CategoryID
where UnitPrice > 50
order by ProductName;
```

The results grid has columns "ProductName", "CategoryName", and "UnitPrice". The data is:

	ProductName	CategoryName	UnitPrice
1	Camarvon Tigers	Seafood	62.50
2	Côte de Blaye	Beverages	263.50
3	Manjimup Dried Apples	Produce	53.00
4	Mishi Kobe Niku	Meat/Poultry	97.00
5	Raclette Courdavault	Dairy Products	55.00
6	Sir Rodney's Marmalade	Confections	81.00
7	Thüringer Rostbratwurst	Meat/Poultry	123.79

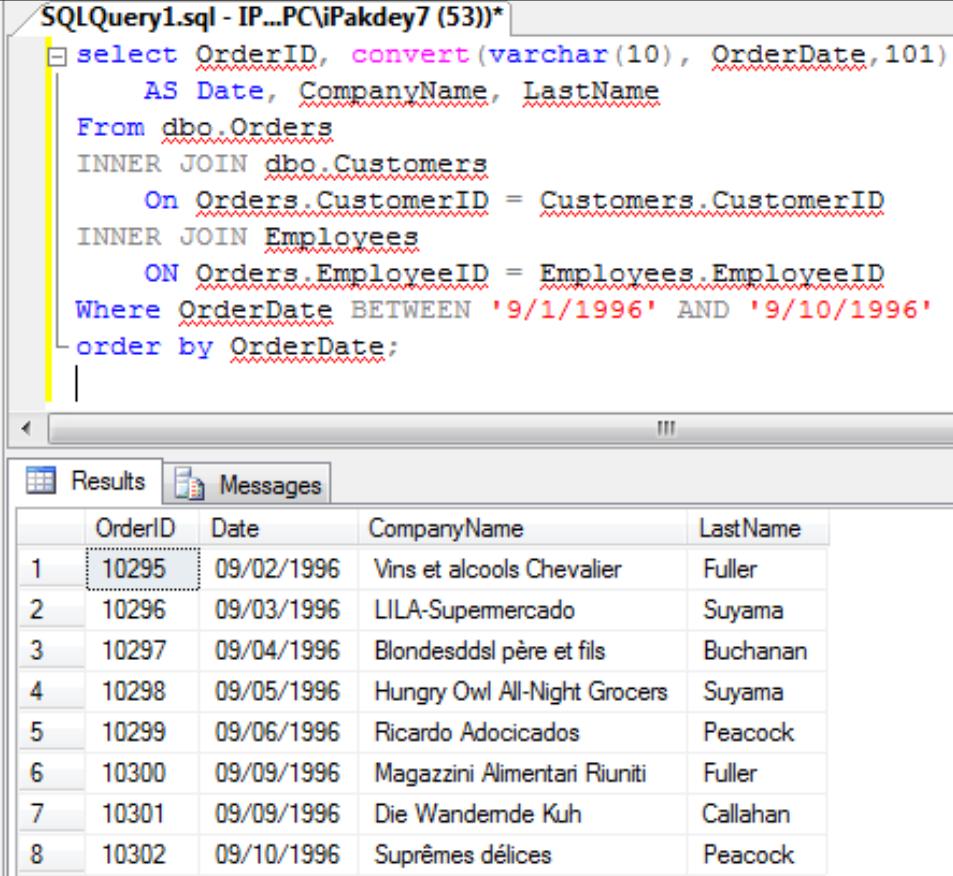
### **14. Multiple Inner Join:**

**Example:**

```

select OrderID, convert(varchar(10), OrderDate,101)
      AS Date, CompanyName, LastName
From dbo.Orders
INNER JOIN dbo.Customers
      On Orders.CustomerID = Customers.CustomerID
INNER JOIN Employees
      ON Orders.EmployeeID = Employees.EmployeeID
Where OrderDate BETWEEN '9/1/1996' AND '9/10/1996'
order by OrderDate;

```

**Output:**


The screenshot shows the SQL Server Management Studio interface. In the top pane, titled 'SQLQuery1.sql - IP...PC\Pakdey7 (53)\*', a T-SQL query is displayed. In the bottom pane, titled 'Results', the query's output is shown in a grid. The grid has four columns: OrderID, Date, CompanyName, and LastName. The data consists of 8 rows, each representing an order from September 1996.

	OrderID	Date	CompanyName	LastName
1	10295	09/02/1996	Vins et alcools Chevalier	Fuller
2	10296	09/03/1996	LILA-Supermercado	Suyama
3	10297	09/04/1996	Blondesddsl père et fils	Buchanan
4	10298	09/05/1996	Hungry Owl All-Night Grocers	Suyama
5	10299	09/06/1996	Ricardo Adocicados	Peacock
6	10300	09/09/1996	Magazzini Alimentari Riuniti	Fuller
7	10301	09/09/1996	Die Wandemde Kuh	Callahan
8	10302	09/10/1996	Suprêmes délices	Peacock

**15. Aggregates and Grouping with Joins:****Example:**

```

Select CompanyName,
       SUM([Order Details].UnitPrice * [Order Details].Quantity)
          As TotalSold
From dbo.Customers inner join dbo.Orders
      On Customers.CustomerID = Orders.CustomerID
      Inner join [Order Details]
      On Order.OrderID = [Order Details].OrderID
Where Orders.OrderDate between '9/1/1996' and '9/10/1996'
Group by CompanyName
Order by TotalSold DESC;

```

**Output:**

```

IPAKDEY7-PC.Northwind - dbo.Orders SQLQuery1.sql - IP...PC\iPakdey7 (53)*
    SUM([Order Details].UnitPrice * [Order Details].Quantity)
    As TotalSold
From dbo.Customers inner join dbo.Orders
    On Customers.CustomerID = Orders.CustomerID
    Inner join [Order Details]
        On Orders.OrderID = [Order Details].OrderID
Where Orders.OrderDate between '9/1/1996' and '9/10/1996'
Group by CompanyName
Order by TotalSold DESC;

```

	CompanyName	TotalSold
1	Hungry Owl All-Night Grocers	3127.00
2	Suprèmes délices	2708.80
3	Blondesddsl père et fils	1420.00
4	LILA-Supermercado	1050.60
5	Die Wandemde Kuh	755.00
6	Magazzini Alimentari Riuniti	608.00
7	Ricardo Adocicados	349.50
8	Vins et alcools Chevalier	121.60

**16. Outer Joins:**

ចំណោះ Outer Joins មានបីប្រភេទដូចត្រូវបញ្ជូនមកពីផលរូបមាន left, right និង full ។

- > Left Join Select ប្រកចិននៃប្រព័ន្ធគម្ពុជា Table ទី 1 ទាំងអស់
- > Right Join Select ប្រកចិននៃប្រព័ន្ធគម្ពុជា Table ទី 2 ទាំងអស់
- > Full Join Select ប្រកចិននៃប្រព័ន្ធ Table ទាំងពីរ ទាំងអស់

ឧទាហរណ៍ទី 1 នេះបង្ហាញពីការប្រើប្រាស់ inner join:

**Example:**

```

Select CompanyName, MIN(Orders.OrderDate) as FirstOrder
From dbo.Customers inner join dbo.Orders
    On Customers.CustomerID = Orders.CustomerID
Group by CompanyName
Order by CompanyName

```

**Output:**

The screenshot shows a SQL query window with the following code:

```

IPAKDEY7-PC.Northwind - dbo.Orders SQLQuery1.sql - IP...PC\iPakdey7 (53)*
Select CompanyName, MIN(Orders.OrderDate) as FirstOrder
From dbo.Customers inner join dbo.Orders
On Customers.CustomerID = Orders.CustomerID
Group by CompanyName
Order by CompanyName
  
```

The results pane displays the following data:

	CompanyName	FirstOrder
1	Alfreds Futterkiste	1997-08-25 00:00:00.000
2	Ana Trujillo Emparedados y helados	1996-09-18 00:00:00.000
3	Antonio Moreno Taquería	1996-11-27 00:00:00.000
4	Around the Horn	1996-11-15 00:00:00.000
5	Berglunds snabbköp	1996-08-12 00:00:00.000
6	Blauer See Delikatessen	1997-04-09 00:00:00.000

ឧទាហរណ៍ទี่ 2 នេះបង្ហាញពីការប្រើប្រាស់ left outer join:

**Example:**

```

Select CompanyName, MIN(Orders.OrderDate) as FirstOrder
From dbo.Customers left join dbo.Orders
On Customers.CustomerID = Orders.CustomerID
Group by CompanyName
Order by CompanyName
  
```

**Output:**

The screenshot shows a SQL query window with the same code as the previous example:

```

IPAKDEY7-PC.Northwind - dbo.Orders SQLQuery1.sql - IP...PC\iPakdey7 (53)*
Select CompanyName, MIN(Orders.OrderDate) as FirstOrder
From dbo.Customers left join dbo.Orders
On Customers.CustomerID = Orders.CustomerID
Group by CompanyName
Order by CompanyName
  
```

The results pane displays the following data, including rows for customers without orders:

	CompanyName	FirstOrder
1	Alfreds Futterkiste	1997-08-25 00:00:00.000
2	Ana Trujillo Emparedados y helados	1996-09-18 00:00:00.000
3	Antonio Moreno Taquería	1996-11-27 00:00:00.000
4	Around the Horn	1996-11-15 00:00:00.000
5	Berglunds snabbköp	1996-08-12 00:00:00.000
6	Blauer See Delikatessen	1997-04-09 00:00:00.000
7	Blondesdsl père et fils	1996-07-25 00:00:00.000
8	Bólido Comidas preparadas	1996-10-10 00:00:00.000

ឧទាហរណ៍ទี่ 3 នេះបង្ហាញពីការប្រើប្រាស់ left outer join ដើម្បីលើ:

**Example:**

```

Select CompanyName as [No Orders]
From dbo.Customers left Join dbo.Orders
On Customers.CustomerID = Orders.CustomerID
Where orders.OrderID IS NULL
  
```

Order by CompanyName;

**Output:**

The screenshot shows the SQL Server Management Studio interface with a query window titled 'IPAKDEY7-PC.Northwind - dbo.Orders / SQLQuery1.sql - IP...PC\iPakdey7 (53)\*'. The query is:

```
Select CompanyName as [No Orders]
From dbo.Customers left Join dbo.Orders
    On Customers.CustomerID = Orders.CustomerID
Where orders.OrderID IS NULL
Order by CompanyName;
```

The results grid shows two rows:

No Orders
1 FISSA Fabrica Inter. Salchichas S.A.
2 Paris spécialités

ឧទាហរណ៍ទี่ 4 នេះបង្ហាញពីការប្រើប្រាស់ right outer join:

**Example:**

```
Select CompanyName as [No Orders]
From dbo.Orders right Join dbo.Customers
    On Customers.CustomerID = Orders.CustomerID
Where orders.OrderID IS NULL
Order by CompanyName;
```

**Output:**

The screenshot shows the SQL Server Management Studio interface with a query window titled 'IPAKDEY7-PC.Northwind - dbo.Orders / SQLQuery1.sql - IP...PC\iPakdey7 (53)\*'. The query is:

```
Select CompanyName as [No Orders]
From dbo.Orders right Join dbo.Customers
    On Customers.CustomerID = Orders.CustomerID
Where orders.OrderID IS NULL
Order by CompanyName;
```

The results grid shows two rows:

No Orders
1 FISSA Fabrica Inter. Salchichas S.A.
2 Paris spécialités

ឧទាហរណ៍ទี่ 5 នេះបង្ហាញពីការប្រើប្រាស់ full outer join:

**Example:**

```
Select ProductName, CategoryName
From dbo.Products full join dbo.Categories
    On dbo.Products.CategoryID = dbo.Categories.CategoryID
Order by ProductName;
```

**Output:**

```

IPAKDEY7-PC.Northwind - dbo.Orders SQLQuery1.sql - IP...PC\iPakdey7 (53)*
Select ProductName, CategoryName
From dbo.Products full join dbo.Categories
On dbo.Products.CategoryID = dbo.Categories.CategoryID
Order by ProductName;

```

	ProductName	CategoryName
1	Alice Mutton	Meat/Poultry
2	Aniseed Syrup	Condiments
3	Boston Crab Meat	Seafood
4	Camembert Pierrot	Dairy Products
5	Camarvon Tigers	Seafood

### 17. Self Joins:

#### Example:

```

Select Employees.FirstName + ' ' + Employees.LastName
As EmployeeName, Managers.FirstName + ' ' +
Managers.LastName As ManagerName
From dbo.Employees INNER JOIN dbo.Employees As Managers
On Employees.ReportsTo = Managers.employeeID

```

#### Output:

```

SQLQuery1.sql - IP...PC\iPakdey7 (53)*
As EmployeeName, Managers.FirstName + ' ' +
Managers.LastName As ManagerName
From dbo.Employees INNER JOIN dbo.Employees As Managers
On Employees.ReportsTo = Managers.employeeID

```

	EmployeeName	ManagerName
1	Nancy Davolio	Andrew Fuller
2	Janet Leverling	Andrew Fuller
3	Margaret Peacock	Andrew Fuller
4	Steven Buchanan	Andrew Fuller
5	Michael Suyama	Steven Buchanan
6	Robert King	Steven Buchanan
7	Laura Callahan	Andrew Fuller
8	Anne Dodsworth	Steven Buchanan

යිහිප්පාගුණුවෙන් රස්ක් Employees ත්‍රීත්‍රීත් යොමු කළේ left join සිදු කිරීමෙන් පාන ඇති නොවා

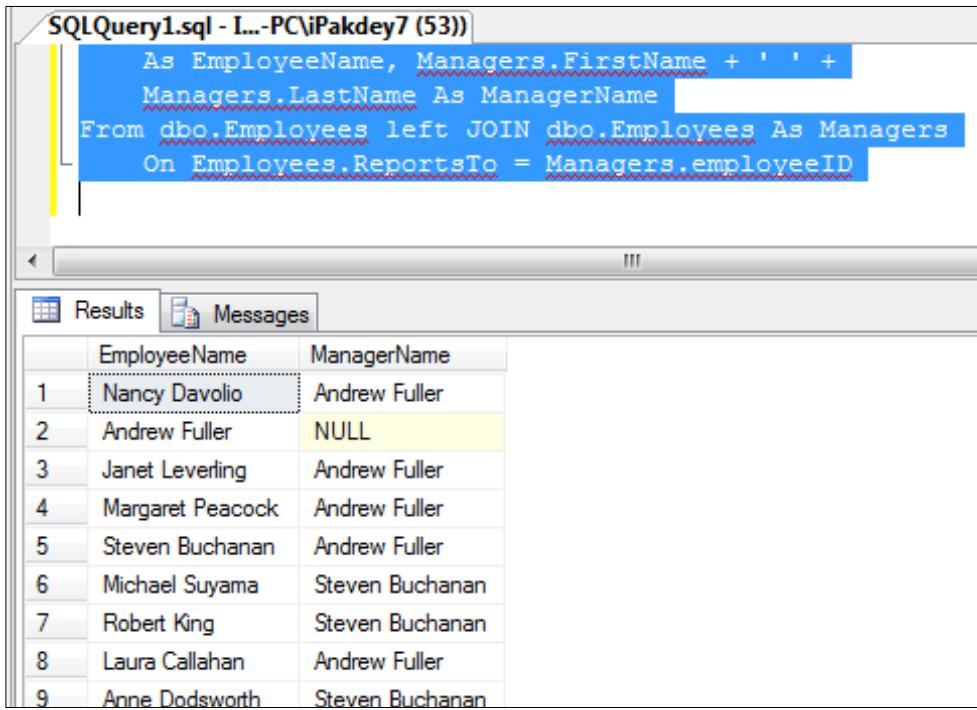
#### Example:

```

Select Employees.FirstName + ' ' + Employees.LastName
As EmployeeName, Managers.FirstName + ' ' +
Managers.LastName As ManagerName
From dbo.Employees LEFT JOIN dbo.Employees As Managers
On Employees.ReportsTo = Managers.employeeID

```

Output:



SQLQuery1.sql - IP...PC\iPakdey7 (53)\*

```

As EmployeeName, Managers.FirstName + ' ' +
Managers.LastName As ManagerName
From dbo.Employees left JOIN dbo.Employees As Managers
On Employees.ReportsTo = Managers.employeeID

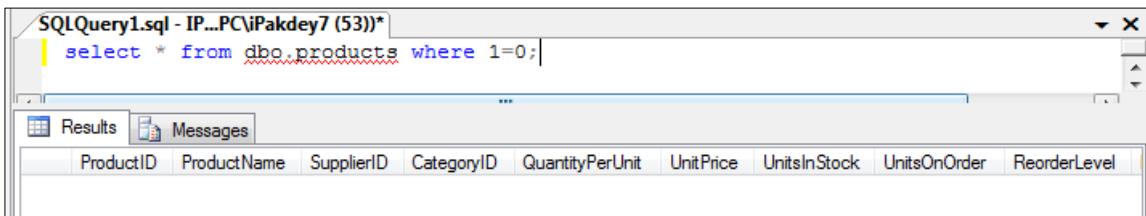
```

	EmployeeName	ManagerName
1	Nancy Davolio	Andrew Fuller
2	Andrew Fuller	NULL
3	Janet Leverling	Andrew Fuller
4	Margaret Peacock	Andrew Fuller
5	Steven Buchanan	Andrew Fuller
6	Michael Suyama	Steven Buchanan
7	Robert King	Steven Buchanan
8	Laura Callahan	Andrew Fuller
9	Anne Dodsworth	Steven Buchanan

## 18. ឧច្ចាស់:

### 1. Simple Select Queries:

- A. ច្បាសើកវា Select យក Columns ទាំងអស់ចេញពី Table Products ?

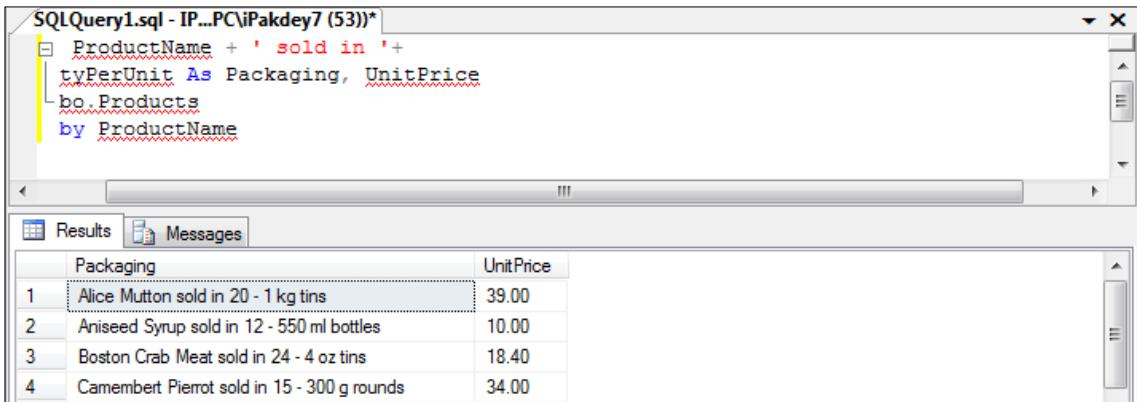


SQLQuery1.sql - IP...PC\iPakdey7 (53)\*

```
select * from dbo.products where 1=0;
```

ProductID	ProductName	SupplierID	CategoryID	QuantityPerUnit	UnitPrice	UnitsInStock	UnitsOnOrder	ReorderLevel

- B. ច្បាសើកវា Select យក ProductName, QuantityPerUnit (នៅចេញនា Columns ទាំងពីរមានពាក្យ sold in ) នៅជា Columns ធ្វើឈ្មោះ Packaging និងយក UnitPrice ចេញពី Table Products ហើយត្រូវបានតាម ProductName ?



SQLQuery1.sql - IP...PC\iPakdey7 (53)\*

```

ProductName + ' sold in ' +
tyPerUnit As Packaging, UnitPrice
- bo.Products
by ProductName

```

Packaging	UnitPrice
1 Alice Mutton sold in 20 - 1 kg tins	39.00
2 Aniseed Syrup sold in 12 - 550 ml bottles	10.00
3 Boston Crab Meat sold in 24 - 4 oz tins	18.40
4 Camembert Pierrot sold in 15 - 300 g rounds	34.00

- C. ស្មូលធ្វើការត្រូវបារក្សាទុម UnitPrice ដាច់លំនាច់បញ្ជី?

SQLQuery1.sql - IP...PC\iPakdey7 (53)\*

```

SELECT ProductName + ' sold in ' +
       tyPerUnit AS Packaging, UnitPrice
FROM Products
ORDER BY UnitPrice DESC;

```

Results

Packaging	UnitPrice
Côte de Blaye sold in 12 - 75 cl bottles	263.50
Thüringer Rostbratwurst sold in 50 bags x 30 sausgs.	123.79
Mishi Kobe Niku sold in 18 - 500 g pkgs.	97.00
Sir Rodney's Marmalade sold in 30 gift boxes	81.00
Camarvon Tigers sold in 16 kg pkg.	62.50

## 2. Aggregate Query:

- A. ស្មូលធ្វើការ Select ប្រភព ProductName, UnitsInStock, UnitsOnOrder, ហើយ ចូរកុំពួក UnitsInStock និង UnitsOnOrder ដូចត្រូវដើម្បីតាម Column ណែនាំ TotalUnits នៃពិនិត្យ Table Products ? តាក់លក្ខខណ្ឌ ត្រូវបញ្ជូនតម្លៃ UnitPrice ឬណានៅជាដំបី 30 និង UnitsOnOrder នៅជាដំបី 0 ទៅប៉ុណ្ណោះ? ច្បាស់ធ្វើការ Group តាមតាម ProductName, UnitsInStock, UnitsOnOrder ហើយតាំងរួមតាម TotalUnits ជាដំបារ៉ែម?

SQLQuery1.sql - IP...PC\iPakdey7 (53)\*

```

SELECT ProductName, UnitsInStock, UnitsOnOrder,
       SUM(UnitsInStock + UnitsOnOrder) AS TotalUnits
  FROM dbo.Products
 WHERE UnitPrice > 30 AND UnitsOnOrder > 0
 GROUP BY ProductName, UnitsInStock, UnitsOnOrder
 ORDER BY TotalUnits DESC;

```

Results

ProductName	UnitsInStock	UnitsOnOrder	TotalUnits
Wimmers gute Semmelknödel	22	80	102
Mascarpone Fabioli	9	40	49
Gnocchi di nonna Alice	21	10	31
Ipo Coffee	17	10	27

- B. ច្បាស់ធ្វើការបង្ហាញ Row ទាំងឡាយបណ្តាល់សិលមាន SUM(UnitsInStock + UnitsOnOrder) > 40 ទៅប៉ុណ្ណោះ?

SQLQuery1.sql - IP...PC\iPakdey7 (53)\*

```

SELECT ProductName, UnitsInStock, UnitsOnOrder,
       SUM(UnitsInStock + UnitsOnOrder) AS TotalUnits
  FROM dbo.Products
 WHERE UnitPrice > 30 AND UnitsOnOrder > 0
 GROUP BY ProductName, UnitsInStock, UnitsOnOrder
 HAVING SUM(UnitsInStock + UnitsOnOrder) > 40
 ORDER BY TotalUnits DESC;

```

Results

ProductName	UnitsInStock	UnitsOnOrder	TotalUnits
Wimmers gute Semmelknödel	22	80	102
Mascarpone Fabioli	9	40	49

## 3. Joining Tables with an Inner Join

- A. ច្បាស់ធ្វើការ Select ប្រភព Columns ទាំងអស់ពី Categories Table ?

SQLQuery1.sql - IP...PC\iPakdey7 (53)\*

```
select * from dbo.categories where 1=0;
```

Results Messages

CategoryID	CategoryName	Description	Picture

B. ច្បាស់ការ Insert ទិន្នន័យចូលទៅក្នុង Categories Table ?

SQLQuery1.sql - IP...PC\iPakdey7 (53)\*

```
insert into dbo.Categories(CategoryName, Description)
values ('All Diet', 'Low carb, low protein, all type of
dieting items')
```

Messages

(1 row(s) affected)

C. ច្បាស់ការ Select យក CategoryName និង ProductName មែនពីរ Relationship រវាង Categories និង Products ជាប្រធែន Inner Join បាន Order by CategoryName?

SQLQuery1.sql - IP...PC\iPakdey7 (53)\*

```
select Categories.CategoryName, Products.ProductName
from dbo.Categories inner join dbo.Products
on categories.categoryid=products.categoryid
order by categories.categoryName
```

Results Messages

CategoryName	ProductName
1 Beverages	Chai
2 Beverages	Chang
3 Beverages	Guaraná Fantástica
4 Beverages	Sasquatch Ale

D. សូមធ្វើការប្រឡងប៉ូល Left Join នឹង?

SQLQuery1.sql - IP...PC\iPakdey7 (53)\*

```
select Categories.CategoryName, Products.ProductName
from dbo.Categories left join dbo.Products
on categories.categoryid=products.categoryid
order by categories.categoryName
```

Results Messages

CategoryName	ProductName
1 All Diet	NULL
2 Beverages	Chai
3 Beverages	Chang
4 Beverages	Guaraná Fantástica
5 Beverages	Sasquatch Ale

# លេខទី 8: សិក្សាត Modifying Data

## **1. Inserting Data:**

Insert Statement គឺត្រូវបានប្រើបាស់ដើម្បីធ្វើការ Add Row ថីមួយច្បាស់ទៅក្នុង Table ។

**Syntax:**

```
Insert [into] table_or_view [(col1,col2,...)]
Values (value1,value2 ...)
```

**Example:**

```
Insert into dbo.Categories (CategoryName)
Values ('Vitamins');
```

**Output:**

```
SQLQuery1.sql - B...(BENQ\enter (52))
use Northwind;
Insert into dbo.Categories (CategoryName)
Values ('Vitamins');

(1 row(s) affected)
```

ក្នុងរាជីណិតយោងធ្វើការជាមួយនឹង Unicode Data Type ដូចជា nvarchar និង ntext គឺយើងត្រូវធ្វើការជាក់ N នៅខាងមុខទិន្នន័យណិតត្រូវ Insert ចូល ។

**Example:**

```
Insert into dbo.Categories (CategoryName)
Values ( N'Vitamins');
```

## **2. Row Affected:**

ជាមួយនឹងការ Insert Data ចូលទៅ Table វិញ នឹងត្រឡប់លទ្ធផល រាន់ខ្លួន Return ចំនួន Rows ណិតយោងនៅក្នុង Table ។ ក្នុងនោះយើងកើតមកប្រើបាស់ NOCOUNT ដើម្បីបិទកំណត់រាយអ្នកញាប់ Row Affected បានដែលជាអំពីរ។

**Example:**

```
Set nocount on;
```

```
Insert into dbo.Categories (CategoryName)
Values ('Vitamins');
Output:
```

```
SQLQuery1.sql - B...(BENQ\enter (52))*
set nocount on;
Insert into dbo.Categories (CategoryName)
Values ('Vitamins');

Messages
Command(s) completed successfully.
```

### **3. Retrieving Identity Values:**

យើងអាចប្រើប្រាស់ Syntax 3 ប្រភេទផ្តល់ចំណាំតាមការបង្កើតឡាយនូវ Identity ចុងក្រាយរបស់ Table ដែលយើងកំណុងត្រួតពារជាមួយ។

<b>@@IDENTITY</b>	<b>SCOPE_IDENTITY</b>	<b>IDENT_CURRENT</b>
Returns the last identity value generated in any table in the current session.	Returns the last identity value only within the current execution scope.	Returns the last identity value generated for a specific table in any session and any scope.

#### **Example:**

```
Select @@IDENTITY AS NewCategoryID
Select SCOPE_IDENTITY() AS NewCategoryID
Select IDENT_CURRENT('dbo.Categories') AS NewCategoryID
```

### **4. Insert Multiple Values:**

#### **Example:**

```
Insert into dbo.Products
    (ProductName, CategoryID, UnitPrice, QuantityPerUnit,
     Discontinued)
Values
    ('Pure''Sunsine''Orange Joice',1,9.99,6,0)
```

Output:

The screenshot shows the SQL Server Management Studio interface. In the top window, titled 'SQLQuery1.sql - B... (BENQ\enter (52))\*', an SQL script is run:

```
use Northwind;
Insert into dbo.Products
(ProductName, CategoryID, UnitPrice, QuantityPerUnit, Discontinued)
Values
('Pure''Sunshine''Orange Joice',1,9.99,6,0)
```

In the bottom window, titled 'Messages', the output is displayed:

```
(1 row(s) affected)
```

### **5. Insert Multiple Rows:**

Example1: ජ්‍යුත්මැන්දික Table දෙපාර්තමේන්තුවයේ Beverages

```
create table dbo.Beverages (
    ProductID int NOT NULL Primary key clustered,
    ProductName nvarchar (40) Not Null,
    CategoryID int Null,
    UnitPrice money Null,
    QuantityPerUnit nvarchar (20) Null,
    Discontinued bit Not Null Default (0))
```

Output:

The screenshot shows the SQL Server Management Studio interface. In the top window, titled 'SQLQuery1.sql - B... (BENQ\enter (52))\*', an SQL script is run:

```
create table dbo.Beverages (
    ProductID int NOT NULL Primary key clustered,
    ProductName nvarchar (40) Not Null,
    CategoryID int Null,
    UnitPrice money Null,
    QuantityPerUnit nvarchar (20) Null,
    Discontinued bit Not Null Default (0))
```

In the bottom window, titled 'Messages', the output is displayed:

```
Command(s) completed successfully.
```

Example2:

```
insert into dbo.Beverages (
    ProductID,ProductName,CategoryID,UnitPrice,QuantityPerUnit,Discontinued)
select
    ProductID,
    ProductName,CategoryID,UnitPrice,QuantityPerUnit,Discontinued
From dbo.Products
```

```
where CategoryID =1;
```

**Output:**

```
SQLQuery1.sql - B...(BENQ\enter (52))*
use Northwind;
insert into dbo.Beverages(
    ProductID,ProductName,CategoryID,UnitPrice,QuantityPerUnit,Discontinued)
select
    ProductID, ProductName,CategoryID,UnitPrice,QuantityPerUnit,Discontinued
From dbo.Products
where CategoryID =1;|
```

Messages

(15 row(s) affected)

## 6. Create New Table Using Insert Into:

**Example:**

```
select dbo.Products.* into dbo.Produce
from dbo.Products
where CategoryID =7;
```

**Output:**

```
SQLQuery1.sql - B...(BENQ\enter (52))*
use Northwind;
select dbo.Products.* into dbo.Produce
from dbo.Products
where CategoryID =7;|
```

Messages

(5 row(s) affected)

## 7. Updating Data:

**Syntax1:**

```
Update tablename
Set column1 = value1, column2=value2,...
```

**Syntax2:**

```
Update TableName
Set TableName.ColumnName = Expression
```

Where SearchCondition

**Example:**

```
select dbo.Products.* into #Produce
use Northwind;
update dbo.Categories
set
Description ='Pills that are good for you'
where CategoryName ='Vitamins';
```

**Output:**

The screenshot shows the SQL Query window titled "SQLQuery1.sql - B... (BENQ\enter (52))". The query is:

```
use Northwind;
update dbo.Categories
set
Description ='Pills that are good for you'
where CategoryName ='Vitamins';
```

The Messages pane below shows the result:

```
(2 row(s) affected)
```

**8. Updating Multiple Rows and Columns:****Example:**

```
update dbo.Products
set
    UnitPrice = UnitPrice *1.1,
    ReorderLevel = ReorderLevel + 5
where CategoryID = 2;
```

**Output:**

The screenshot shows a SQL query window titled "SQLQuery1.sql - B... (BENQ\enter (52))". The query updates the "Products" table in the "Northwind" database. It sets the "UnitPrice" to its current value multiplied by 1.1 and increases the "ReorderLevel" by 5 for products in category ID 2. The output message in the "Messages" pane indicates "(12 row(s) affected)".

```

use Northwind;
update dbo.Products
set
    UnitPrice = UnitPrice *1.1,
    ReorderLevel = ReorderLevel + 5
where CategoryID = 2;

```

(12 row(s) affected)

#### **9. Updating from Another Table:**

**Example:**

```

update dbo.Beverages
set
    dbo.Beverages.UnitPrice = dbo.Products.UnitPrice
From dbo.Beverages JOIN dbo.Products
on dbo.Beverages.ProductID = dbo.Products.ProductID;

```

**Output:**

The screenshot shows a SQL query window titled "SQLQuery1.sql - B... (BENQ\enter (52))". The query updates the "Beverages" table in the "Northwind" database. It sets the "UnitPrice" of the "Beverages" table to the "UnitPrice" of the "Products" table, joining them on the "ProductID". The output message in the "Messages" pane indicates "(15 row(s) affected)".

```

use Northwind;
update dbo.Beverages
set
    dbo.Beverages.UnitPrice = dbo.Products.UnitPrice
From dbo.Beverages JOIN dbo.Products
on dbo.Beverages.ProductID = dbo.Products.ProductID;

```

(15 row(s) affected)

#### **10. Updating with TOP:**

**Example:**

```

use Northwind;
update top (5) dbo.Beverages
set UnitPrice = UnitPrice *1.1
where Discontinued =0;

```

**Output:**

The screenshot shows a SQL Server Management Studio window. The top pane displays the following T-SQL code:

```
SQLQuery1.sql - B...(BENQ\enter (52))*
use Northwind;
update top (5) dbo.Beverages
set UnitPrice = UnitPrice *1.1
where Discontinued =0;
```

The bottom pane, titled "Messages", shows the result of the execution:

```
(5 row(s) affected)
```

### **11. Deleting Data:**

Syntax:

```
Delete [From] TableName  
[Where SearchConditions]
```

Example:

```
Delete from dbo.Beverages  
Where UnitPrice < 10;
```

Output:

The screenshot shows a SQL Server Management Studio window. The top pane displays the following T-SQL code:

```
SQLQuery1.sql - B...(BENQ\enter (52))*
Delete from dbo.Beverages
Where UnitPrice < 10;
```

The bottom pane, titled "Messages", shows the result of the execution:

```
(5 row(s) affected)
```

ដើម្បីធ្វើការ Delete Rows ទាំងអស់ចំណាំពី Table នោះយើងអាចប្រើប្រាស់ Syntax ដូចខាងក្រោម:

```
Delete from dbo.Beverages;
```

### **12. Using Truncate Table:**

Truncate Table កើត្រូវបានប្រើប្រាស់ដើម្បីធ្វើការលួយ Rows ចេញពី Table ដែលត្រូវបានលក្ខណៈលើវិនាទនា Delete

Syntax %

#### Example:

Trancate Table dbo.Produce

### 13. ផ្តល់ជូន:

#### 1. Adding Product:

A. ចូរធ្វើការ Add ទិន្នន័យថ្មីលទ្ធផ្លែង Table Product តាមតារាងខាងក្រោម ហើយរៀបចំឡាយ ID របស់ Identity នៃ Record ដែលបានផ្តល់ជូន។

Name	Value
ProductName	Green Tea Cola
CategoryID	1
QuantityPerUnit	12
UnitPrice	3.95
ReorderLevel	5
Discontinued	0

```
SQLQuery1.sql - B... (BENQ\enter (52))*
use Northwind;
insert into dbo.Products(
    ProductName, CategoryID, QuantityPerUnit, UnitPrice, ReorderLevel, Discontinued)
values ('Green Tea Cola', 1, 12, 3.95, 5, 0)
select SCOPE_IDENTITY() as NewProductID
```

NewProductID
81

#### 2. Editing Product:

A. ចូរធ្វើការ Update ទិន្នន័យរបស់ Table Product ដោយកំនត់ UnitPrice = 4.95 ដើម្បីផ្លូវត្រូវ ProductID = 79 ហើយបញ្ជាប់មកស្មូមធ្វើការ Select យក ProductName និង UnitPrice របស់ ProductID នេះដោយការរៀបចំ។

The screenshot shows the SQL Server Management Studio interface. In the top window, titled 'SQLQuery1.sql - B...(BENQ\enter (52))\*', the following SQL code is displayed:

```

use Northwind;
update dbo.Products
set UnitPrice = 4.95
where ProductID = 79;
select ProductName, UnitPrice
from dbo.Products
where ProductID = 79;

```

Below this, the 'Results' tab is selected, showing the output of the query:

	ProductName	UnitPrice
1	Pure'Sunsine'Orange Joice	4.95

### 3. Deleting Product:

A. ចូរធ្វើការ Delete Row ដែលមាន ProductID = 81 ចោញពី Table Products

The screenshot shows the SQL Server Management Studio interface. In the top window, titled 'SQLQuery1.sql - B...(BENQ\enter (52))\*', the following SQL code is displayed:

```

use Northwind;
delete from dbo.Products
where ProductID = 81;

```

Below this, the 'Messages' tab is selected, showing the output of the query:

(1 row(s) affected)



# ເຕັມເກຣມ 9: ໜີ້ຈຸດໃນ T-SQL Programming

## 1. Variable:

នៅក្នុង T-SQL កីឡាថម្គិនអាជីវកម្មប្រើប្រាស់ variable ដើម្បីធ្វើការ Store Data បណ្តាឃេសនូបានផែន្ទិរ។  
ដើម្បីធ្វើការប្រគល់ទិន្នន័យត្រូវប្រើប្រាស់ DECLEAR Keyword ហើយត្រូវកំណត់ Data Type របស់វានៅលម្អិតថ្មីសំរាប់  
លាកស់ផែន្ទិរ។ ត្រូវនោះប្រសិនបើវាតា Local Variable នោះត្រូវបង្ហើមសញ្ញា @ នៅខាងមុខលើក្រោមរាយផែន្ទិរ។

## Example:

```
use Northwind;
DECLARE @local char(13);
SET @local = 'Local Phone: ';
--or--
SELECT @local = 'Local Phone: ';

SELECT @local AS Value;

SELECT LastName, FirstName, @local+ HomePhone AS Phone
FROM dbo.Employees
ORDER BY LastName, FirstName;
```

## Output:

	Value		
1	Local Phone:		
	Last Name	First Name	Phone
1	Buchanan	Steven	Local Phone: (71) 555-4848
2	Callahan	Laura	Local Phone: (206) 555-1189
3	Davolio	Nancy	Local Phone: (206) 555-9857
4	Dodsworth	Anne	Local Phone: (71) 555-4444
5	Fuller	Andrew	Local Phone: (206) 555-9482
6	King	Robert	Local Phone: (71) 555-5598
7	Leverling	Janet	Local Phone: (206) 555-3412
8	Peacock	Margaret	Local Phone: (206) 555-8122
9	Suyama	Michael	Local Phone: (71) 555-7773

## 2. Data Type:

ចំណោះ Data Type របស់ Variable គឺមានប្រភេទរបស់ Column និង Table ដើម្បីជូន។

**Example:**

```
DECLARE @msg varchar(20);
SELECT @msg = 'The result is: 2+2';
PRINT @msg;
```

**Output:**

```
SQLQuery1.sql - B...BENQ\enter (52)*
DECLARE @msg varchar(20);
SELECT @msg = 'The result is: 2+2';
PRINT @msg;

Messages
The result is: 2+2
```

ចំណោះ ឧទាហរណ៍ខាងក្រោមនេះវិញ គឺរាយឱ្យបង្ហាញ Error ឬនោះ Variable ប្រភេទជា Text តើមិនអាច Store Value ជាមួយបានឡើយ

**Example:**

```
DECLARE @msg varchar(20);
SELECT @msg = 'The result is:' + (2+2);
PRINT @msg;
```

## 3. Cast and Convert:

យើងអាចប្រើប្រាស់ Function CAST និង CONVERT សំរាប់ធ្វើការ convert Data Type ទៅតាមតម្លៃការផ្តល់ជូន។

```
DECLARE @msg varchar(20);
SELECT @msg = 'The result is: ' + CAST((2+2) AS varchar);
PRINT @msg;
```

```
DECLARE @msg varchar(20);
SELECT @msg = 'The result is: ' + CONVERT(varchar, (2+2));
PRINT @msg;
```

## 4. Using Convert with Style:

យើងអាចប្រើប្រាស់ Function CAST និង CONVERT ជាមួយនឹង Style Argument សំរាប់ធ្វើការ Format ទៅតាមទំនួនដែលមានតម្លៃ។

**Example:**

```
SELECT CONVERT(varchar(12), GETDATE(),1);
```

```
SELECT CONVERT(varchar(12), GETDATE(),101);
SELECT CONVERT(varchar(12), GETDATE(),102);
SELECT CONVERT(varchar(12), GETDATE(),126);
```

**Output:**

(No column name)	
1	08/09/13
(No column name)	
1	08/09/2013
(No column name)	
1	2013.08.09
(No column name)	
1	2013-08-09T2

**5. STR:**

STR Function ដែលត្រូវបង្កើតរាយ Return String ចេញពី Numeric Expression ។

**Example:**

```
SELECT UnitPrice, STR(UnitPrice,6,1) AS Formatted
FROM dbo.Products
ORDER BY UnitPrice DESC;
```

**Output:**

SQLQuery1.sql - B...(BENQ\enter (52))\*

```
use Northwind
SELECT UnitPrice, STR(UnitPrice,6,1) AS Formatted
FROM dbo.Products
ORDER BY UnitPrice DESC;
```

		UnitPrice	Formatted
1	263.50	263.5	
2	123.79	123.8	
3	97.00	97.0	
4	81.00	81.0	

**6. Working with Nulls:****Example:**

```
SELECT CompanyName, City
```

```
FROM dbo.Suppliers
WHERE Region = NULL;
```

Output:

```
SQLQuery1.sql - B...(BENQ\enter (52))
SELECT CompanyName, City
FROM dbo.Suppliers
WHERE Region = NULL;
```

CompanyName	City

## **7. IS NULL:**

Example: IS NULL គឺត្រូវបានប្រើប្រាស់ដើម្បីធ្វើការ Test នក Null Value នៅក្នុង Column ណាមួយ ។

**Example:**

```
SELECT CompanyName, City
FROM dbo.Suppliers
WHERE Region IS NULL;
```

និងសារស្ថាដូចខាងក្រោមនេះគឺប្រើប្រាស់ ISNULL Function ដើម្បីធ្វើការ Replace តំលៃ Null ជាមួយនឹង តំលៃបានផ្តល់នូវ។

**Example:**

```
SELECT CompanyName, City
ISNULL(Region,'??') AS Regioin
FROM dbo.Suppliers
```

The screenshot shows a SQL query window titled "SQLQuery1.sql - B...(BENQ\enter (52))". The query uses the Northwind database and selects CompanyName, City, and Region from the dbo.Suppliers table. The Region column uses the ISNULL function to replace NULL values with '???'. The results grid displays 7 rows of supplier information.

	CompanyName	City	Region
1	Exotic Liquids	London	???
2	New Orleans Cajun Delights	New Orleans	LA
3	Grandma Kelly's Homestead	Ann Arbor	MI
4	Tokyo Traders	Tokyo	???
5	Cooperativa de Quesos 'Las Cabras'	Oviedo	Asturias
6	Mayumi's	Osaka	???
7	Pavlova, Ltd.	Melbourne	Victoria

## 8. NULIF:

NULIF គឺជា Function មួយប្រភេទដែលផ្តល់ឱ្យបានឈ្មោះថា ISNULL ធ្វើឡើងនៅពីការ replace តំណែលណាមួយដែលជាតំនៃ NULL វិញ។

**Example:**

```
SELECT AVG(NULIF(UnitPrice, 0)) FROM dbo.Products;
--INSTEAD OF -
SELECT AVG(UnitPrice) FROM dbo.Products;
```

## 9. COALESCE:

វាគ្រូរបានប្រើប្រាស់ដើម្បីធ្វើការ Replace តំណែលណាមួយជាមួយនឹងតម្លៃដែលផ្តល់ឱ្យបានឈ្មោះត្រូវការណីដែលរាល់ជាបាន NULL ។

**Example:**

```
SELECT CompanyName, City, ', '+COALESCE(Region, Country)
AS LOCATION
FROM dbo.Suppliers;
```

The screenshot shows a SQL query window titled "SQLQuery1.sql - B...(BENQ\enter (52))". The query selects CompanyName, City, and LOCATION from the dbo.Suppliers table. The LOCATION column is created by concatenating a comma and a space with the result of the COALESCE function, which takes Region and Country as arguments. The results grid displays 7 rows of supplier information.

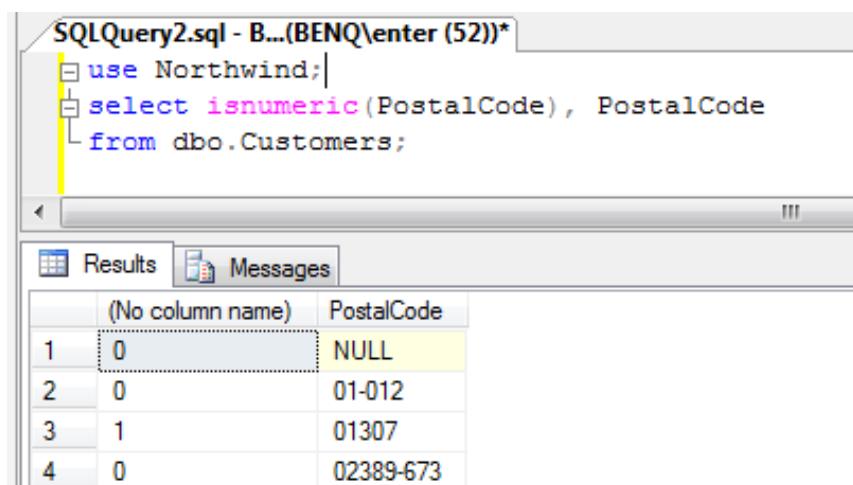
	CompanyName	LOCATION
1	Exotic Liquids	London, UK
2	New Orleans Cajun Delights	New Orleans, LA
3	Grandma Kelly's Homestead	Ann Arbor, MI
4	Tokyo Traders	Tokyo, Japan
5	Cooperativa de Quesos 'Las Cabras'	Oviedo, Asturias
6	Mayumi's	Osaka, Japan

**10. IS NUMERIC:**

វាគ្រប់បានប្រើប្រាស់ដើម្បីធ្វើត្រួតព័ត៌មាន Return តិចនៅ 1 ឬ 0 (True/False) នៅព័ត៌មានផ្តល់វាព័ត៌មាន Check មិនជាមួយនឹង Number ។

Ex:

```
select isnumeric(PostalCode), PostalCode
from dbo.Customers;
```



The screenshot shows the SQL Query window with the following code:

```
SQLQuery2.sql - B...(BENQ\enter (52))*
use Northwind;
select isnumeric(PostalCode), PostalCode
from dbo.Customers;
```

The Results tab displays the following data:

(No column name)	PostalCode
1	0
2	01-012
3	01307
4	02389-673

**11. RAND:**

វាគ្រប់បានប្រើប្រាស់ដើម្បីធ្វើត្រួតព័ត៌មាន Random ផែនមានចន្ទនោះពី 0 ដល់ 1 ។

Ex:

```
use Northwind;
--with an automatic random seed,
--different numbers every time.
select RAND(),RAND(),RAND();

--With a fixed seed,
-- the same series every time.
select RAND(456),RAND(),RAND();
```

The screenshot shows two queries in the SQL Query window:

```

SQLQuery2.sql - B...(BENQ\enter (52))*
```

```

use Northwind;
--with an automatic random seed,
--different numbers every time.
select RAND(), RAND(), RAND();

--With a fixed seed,
-- the same series every time.
select RAND(456), RAND(), RAND();
```

The Results tab displays the output of the first query:

	(No column name)	(No column name)	(No column name)
1	0.681245769315943	0.170403819984849	0.214319945101034

	(No column name)	(No column name)	(No column name)
1	0.722069995135404	0.711014519267536	0.211107652334825

## 12. RAND:

វាគ្មោះបានប្រើប្រាស់ដើម្បីធ្វើការ បង្កើតលេខក្រោមក្នុង នរាយឡើង បុច្ចេះថ្លែកាមត្រូវការ។

Ex:

```

use Northwind;
select UnitPrice,
       ROUND(UnitPrice, 0) AS RoundedDollars,
       ROUND(UnitPrice, 0,1) AS TruncatedDollars,
       ROUND(UnitPrice, 1) AS ToTensOfCents,
       ROUND(UnitPrice, -1) AS ToTensOfDollars
From dbo.Products;
```

The screenshot shows a query in the SQL Query window:

```

SQLQuery2.sql - B...(BENQ\enter (52))*
```

```

use Northwind;
select UnitPrice,
       ROUND(UnitPrice, 0) AS RoundedDollars,
       ROUND(UnitPrice, 0,1) AS TruncatedDollars,
       ROUND(UnitPrice, 1) AS ToTensOfCents,
       ROUND(UnitPrice, -1) AS ToTensOfDollars
From dbo.Products;
```

The Results tab displays the output of the query:

	UnitPrice	RoundedDollars	TruncatedDollars	ToTensOfCents	ToTensOfDollars
63	48.29	48.00	48.00	48.30	50.00
64	33.25	33.00	33.00	33.30	30.00
65	23.155	23.00	23.00	23.20	20.00
66	18.70	19.00	18.00	18.70	20.00

តាមខ្លាបរណីខាងក្រោមបង្ហាញពីការប្រើប្រាស់ ROUND ជាមួយនឹង RAND ដើម្បីបង្កើតជាលេខ Random:

```
use Northwind;
DECLARE @minID int, @maxID int
SET @minID =(SELECT MIN(EmployeeID) FROM dbo.Employees)
SET @maxID =(SELECT MAX(EmployeeID) FROM dbo.Employees)
SELECT EmployeeID AS LuckyID, LastName As LuckyName
FROM dbo.Employees
WHERE EmployeeID =
    ROUND(@minID+ (RAND () * (@maxID-@minID)) ,0);
```

SQLQuery2.sql - B...(BENQ\enter (52))\*

```
use Northwind;
DECLARE @minID int, @maxID int
SET @minID =(SELECT MIN(EmployeeID) FROM dbo.Employees)
SET @maxID =(SELECT MAX(EmployeeID) FROM dbo.Employees)
SELECT EmployeeID AS LuckyID, LastName As LuckyName
FROM dbo.Employees
WHERE EmployeeID =
    ROUND(@minID+ (RAND () * (@maxID-@minID)) ,0);
```

Results Messages

LuckyID	LuckyName
1	5 Buchanan

# លេខទូរសព្ទ 10: សិក្សាតី

## T-SQL String and Date

### **1. REPLACE and STUFF:**

REPLACE គឺត្រូវបានប្រើប្រាស់ដើម្បីធ្វើការ ដំឡើសតំលៃរបស់ String ឡើងតំលៃថ្មីដែលផ្តល់ព័ត៌មាន។

**Example:**

```
use Northwind;
SELECT QuantityPerUnit,
       REPLACE(QuantityPerUnit,'12','twelve ') AS Ten
FROM dbo.Products;
```

The screenshot shows the SQL Query window titled "SQLQuery1.sql - B... (BENQ\enter (52))". The query is as follows:

```
use Northwind;
SELECT QuantityPerUnit,
       REPLACE(QuantityPerUnit,'12','twelve ') AS Ten
FROM dbo.Products;
```

The Results tab displays the output:

	QuantityPerUnit	Ten
1	10 boxes x 20 bags	10 boxes x 20 bags
2	24 - 12 oz bottles	24 - twelve oz bottles
3	12 - 550 ml bottles	twelve - 550 ml bottles
4	48 - 6 oz jars	48 - 6 oz jars
5	36 boxes	36 boxes

ចំណោះ STUFF វិញ្ញាកីត្រូវបានប្រើប្រាស់ដើម្បីធ្វើការ Replace នូវរាយការណ៍ដែលបានដាក់ចុចដោយក្រុមហ៊ុនផ្សេងៗ។

Ex:

```
SELECT STUFF('123456', 3, 2, 'xxxx');
```

```
SQLQuery1.sql - B... (BENQ\enter (52))
SELECT STUFF ('123456', 3, 2, 'xxxx') ;
```

	(No column name)
1	12xxxx56

## 2. LEN, LEFT, and RIGHT:

- > LEN តិចជា Function សំវាប់ធ្វើការ return នូវប្រព័ន្ធសម្រាប់ String
- > LEFT សំវាប់ធ្វើការកាត់យកចំនួនតុអក្សរពីឆ្នងចែកពី String សម្រាប់
- > RIGHT សំវាប់ធ្វើការកាត់យកចំនួនតុអក្សរពីខាងស្តាំចែកពី String សម្រាប់

### Example:

```
use Northwind;
SELECT ProductName,
       LEFT(ProductName, LEN(ProductName) - 3) AS Lefty,
       RIGHT(ProductName, LEN(ProductName) - 3) AS Righty
FROM dbo.Products;
```

```
SQLQuery1.sql - B... (BENQ\enter (52))
use Northwind;
SELECT ProductName,
       LEFT(ProductName, LEN(ProductName) - 3) AS Lefty,
       RIGHT(ProductName, LEN(ProductName) - 3) AS Righty
FROM dbo.Products;
```

	ProductName	Lefty	Righty
1	Alice Mutton	Alice Mut	ce Mutton
2	Aniseed Syrup	Aniseed Sy	seed Syrup
3	Boston Crab Meat	Boston Crab M	ton Crab Meat
4	Camembert Pierrot	Camembert Pier	embert Pierrot

## 3. SUBSTRING:

SUBSTRING Function ផ្តល់ធ្វើការ return នូវ ផ្តល់កណ្តាលមួយដៃនៃ String ដោយផ្តល់ករើចំនួចចាប់ផ្តើម ផ្តល់បញ្ជីកាត់និងប្រព័ន្ធដែលបញ្ជីការកាត់យក។

**Example:**

```
use Northwind;
SELECT SUBSTRING(FirstName, 1, 1) + '.' + LastName
FROM dbo.Employees;
```

The screenshot shows the SQL Query window titled "SQLQuery1.sql - B...(BENQ\enter (52))". The query is:

```
use Northwind;
SELECT SUBSTRING(FirstName, 1, 1) + '.' + LastName
FROM dbo.Employees;
```

The results pane shows the output:

	(No column name)
1	N. Davolio
2	A. Fuller
3	J. Leverling
4	M. Peacock

**4. CHARINDEX:**

វាគ្មោះបានប្រើប្រាស់ដើម្បីធ្វើការ ស្អួលរកនឹងលេខទីកំងរបស់គ្មានក្បាលម្មួយ។

**Example:**

```
use Northwind;
SELECT HomePhone, CHARINDEX(' ', HomePhone) As StartPos
FROM dbo.Employees;
```

The screenshot shows the SQL Query window titled "SQLQuery1.sql - B...(BENQ\enter (52))". The query is:

```
SELECT HomePhone, CHARINDEX(' ', HomePhone) As StartPos
FROM dbo.Employees;
```

The results pane shows the output:

	HomePhone	StartPos
1	(206) 555-9857	5
2	(206) 555-9482	5
3	(206) 555-3412	5
4	(206) 555-8122	5

**Example:**

```
use Northwind;
SELECT HomePhone, CHARINDEX(' ', HomePhone) As StartPos
SUBSTRING(HomePhone, CHARINDEX(' ', HomePhone)+2,
LEN(HomePhone)-CHARINDEX(' ', HomePhone)+1) As Number
FROM dbo.Employees;
```

```

SQLQuery1.sql - B...(BENQ\enter (52))*
use Northwind;
select HomePhone,
       LEFT(HomePhone, CHARINDEX(')', HomePhone)) As AreaCode,
       SUBSTRING(HomePhone, CHARINDEX(')', HomePhone)+2,
       LEN(HomePhone)-CHARINDEX(')', HomePhone)+1) As Number
From dbo.Employees;
    
```

	HomePhone	AreaCode	Number
1	(206) 555-9857	(206)	555-9857
2	(206) 555-9482	(206)	555-9482
3	(206) 555-3412	(206)	555-3412
4	(206) 555-8122	(206)	555-8122

## 5. PATINDEX:

PATINDEX មានលក្ខណៈ: ប្រើប្រាស់ប្រហាក់ប្រព័លត្បាតីដែល CHARINDEX ដែលបង្កើត វាយតាម Supports Wildcard Characters បានដៃង់ជូន។

### Example:

```

use Northwind;
select ProductName, QuantityPerUnit
From dbo.Products
Where PATINDEX('24 - % g pkgs.',QuantityPerUnit) >0;

select ProductName, QuantityPerUnit
From dbo.Products
Where PATINDEX('24 - __ g pkgs.',QuantityPerUnit)>0;
    
```

```

SQLQuery1.sql - B...(BENQ\enter (52))*
use Northwind;
select ProductName,QuantityPerUnit
From dbo.Products
Where PATINDEX('24 - % g pkgs.',QuantityPerUnit) >0;

select ProductName, QuantityPerUnit
From dbo.Products
Where PATINDEX('24 - __ g pkgs.',QuantityPerUnit)>0;
    
```

	ProductName	QuantityPerUnit
1	Gustaf's Knäckebröd	24 - 500 g pkgs.
2	Mascarpone Fabioli	24 - 200 g pkgs.

	ProductName	QuantityPerUnit
1	Maxilaku	24 - 50 g pkgs.

**6. SPACE:**

SPACE Function គឺសំរាប់ធ្វើការ បង្កើតចំណាំ Space ទៅតាមតម្លៃការ ។

**Example:**

```
use Northwind;
select FirstName + SPACE(15-LEN(FirstName))
+ LastName As NameInColumns
From dbo.Employees;
```

	NameInColumns	
1	Nancy	Davolio
2	Andrew	Fuller
3	Janet	Leverling
4	Margaret	Peac...

**7. CHAR and ASCII:**

CHAR Function គឺសំរាប់ធ្វើការ Convert តំណែ ASCII Code ទៅជាការ Character ចំនួន ASCII Function គឺសំរាប់ធ្វើការ

Convert ពី Character មកជាការ ASCII Code វិញ្ញាបី

**Example:**

```
use Northwind;
select CHAR(71)+CHAR(72)
As Characters;
select ASCII('G') As ASciiColumn;
```

	Characters
1	GH

	ASciicolumn
1	71

**7. LOWER and UPPER:**

LOWER Function គឺជាវប្បធម៌ការ Convert អក្សរទៀតជា lowercase ចំនួន UPPER គឺជាវប្បធម៌ការ Convert ទៀតជា Uppercase វិញ្ញាយ

**Example:**

```
use Northwind;
select UPPER(FirstName) As FIRSTNAME,
LOWER(LastName) As lastName
From dbo.Employees;
```

The screenshot shows the SQL Server Management Studio interface. In the top-left pane, there is a code editor window titled "SQLQuery1.sql - B... (BENQ\enter (52))". It contains the following T-SQL code:

```
select UPPER(FirstName) As FIRSTNAME,
LOWER(LastName) As lastName
From dbo.Employees;
```

Below the code editor is a results grid. The grid has two columns: "FIRSTNAME" and "lastName". It displays two rows of data:

	FIRSTNAME	lastName
1	NANCY	davolio
2	ANDREW	fuller

Ex:

The screenshot shows the SQL Server Management Studio interface. In the top-left pane, there is a code editor window titled "SQLQuery1.sql - B... (BENQ\enter (52))". It contains the following T-SQL code:

```
select CompanyName
From dbo.Customers
Where CHARINDEX(Upper('the '), UPPER(CompanyName)) > 0;
```

Below the code editor is a results grid. The grid has one column: "CompanyName". It displays three rows of data:

CompanyName
Around the Horn
The Big Cheese
The Cracker Box

## 8. LTRIM and RTRIM:

LTRIM និង RTRIM គឺគ្រប់ប្រាក់ដើម្បីប្រាក់ដើម្បីការ remove leading និង trailing spaces ដែលនៅសម្រាប់អក្សរ។

**Example:**

```
use Northwind;
select LTRIM(RTRIM('      Hello      ')) + 'There' As Trimmed;
```

```
SQLQuery1.sql - B...(BENQ\enter (52))
select LTRIM(RTRIM('Hello      ')) + 'There' As Trimmed;
```

Trimmed
HelloThere

## **9. GETDATE:**

វាគ្រូរបានប្រើប្រាស់ដើម្បីធ្វើការបង្ហាញ ពេលវេលាបច្ចុប្បន្ន ។

**Example:**

```
SELECT GETDATE() AS RightNow;
```

- > MONTH(GETDATE()) គឺត្រូវបានប្រើប្រាស់ដើម្បីធ្វើការកាត់យកខែចេញពីកាលបរិច្ឆេទ
- > DAY(GETDATE()) គឺត្រូវបានប្រើប្រាស់ដើម្បីធ្វើការកាត់យកថ្ងៃចេញពីកាលបរិច្ឆេទ
- > YEAR(GETDATE()) គឺត្រូវបានប្រើប្រាស់ដើម្បីធ្វើការកាត់យកឆ្នាំចេញពីកាលបរិច្ឆេទ

**Example:**

```
SELECT MONTH(GETDATE()) As ThisMonth,
DAY(GETDATE()) As ThisDay,
YEAR(GETDATE()) As ThisYear;
```

## **10. DATEPART and DATENAME:**

DATEPART ត្រូវបានប្រើប្រាស់ដើម្បីធ្វើការ Format ឡើងឱ្យ កាលបរិច្ឆេទទៅតាមទំនំផ្សេងៗ ដើម្បីត្រូវការ។

Part of Date	Abbreviations
Year	yy, yyyy
Quarter	qq, q
Month	mm, m
Day of year	dy, y
Day	dd, d
Week	wk, ww
Weekday	dw
Hour	hh
Minute	mi, n
Second	ss, s
Millisecond	ms

### **Example:**

```
SELECT GETDATE() AS RightNow;
```

> MONTH (GETDATE ( ) )

SQLQuery1.sql - B...BENQ\enter (52)

```
select  
    DATEPART(dy, GETDATE ()) AS DayOfYear,  
    DATEPART(dd, GETDATE ()) AS DayNum,  
    DATEPART(ww, GETDATE ()) AS WeekNum,  
    DATEPART(dw, GETDATE ()) AS WeekDay,  
    DATEPART(hh, GETDATE ()) AS Hour,  
    DATEPART(mi, GETDATE ()) AS Minute,  
    DATEPART(ss, GETDATE ()) AS Seconds;
```

	DayOfYear	DayNum	WeekNum	WeekDay	Hour	Minute	Seconds
1	233	21	34	4	13	39	57

ចំណោះ DATNAME និង ត្រូវបានប្រើប្រាស់ដើម្បីធ្វើការ Format ឡើលើ កាលបរិច្ឆេទនៅតាមទំនួរដៃខែឆ្នាំ ដែលត្រូវការប្រើប្រាស់ការ return ឈ្មោះរបស់ DATE ជាមួយជាតុ String ដើម្បីរាយ។

### **Example:**

select

```
DATENAME (qq, GETDATE ()) AS Quarter,  
DATENAME (mm, GETDATE ()) AS Month,
```

```
DATENAME(dw, GETDATE()) AS Weekday,
DATENAME(hh, GETDATE()) AS Hour,
DATENAME(mi, GETDATE()) AS Minute,
DATENAME(mi, GETDATE()) AS Seconds;
```

SQLQuery1.sql - B...BENQ\enter (52)\*

```
select
    DATENAME(qq, GETDATE()) AS Quarter,
    DATENAME(mm, GETDATE()) AS Month,
    DATENAME(dw, GETDATE()) AS Weekday,
    DATENAME(hh, GETDATE()) AS Hour,
    DATENAME(mi, GETDATE()) AS Minute,
    DATENAME(mi, GETDATE()) AS Seconds;
```

	Quarter	Month	Weekday	Hour	Minute	Seconds
1	3	August	Wednesday	13	43	43

#### **11. DATEADD and DATEDIFF:**

DATEADD គឺជា Function ប្រធែន Date ដើម្បីផ្តល់ព័ត៌មានអំពីថ្ងៃខែឆ្នាំដែលបានផ្តល់នូវការបន្ថែម Date ទៅការពិនិត្យការប្រាកបដាក់។

**Example:**

```
select
    DATEADD(yy, 1, GETDATE()) AS AddYear,
    DATEADD(mm, 1, GETDATE()) AS AddMonth,
    DATEADD(dd, 1, GETDATE()) AS AddDay;
```

SQLQuery1.sql - B...BENQ\enter (52)\*

```
select
    DATEADD(yy, 1, GETDATE()) AS AddYear,
    DATEADD(mm, 1, GETDATE()) AS AddMonth,
    DATEADD(dd, 1, GETDATE()) AS AddDay;
```

	AddYear	AddMonth	AddDay
1	2014-08-21 13:48:51.497	2013-09-21 13:48:51.497	2013-08-22 13:48:51.497

ចំណោះ DATEDIFF វិញ្ញុ គឺជា Function ដើម្បីប្រើប្រាស់ តមាលាការកំណត់ថ្ងៃដែលបានផ្តល់នូវការបន្ថែមពី Start Date និង End Date ។

**Example:**

```
use Northwind;
SELECT
    OrderDate, RequiredDate, ShippedDate,
```

```
DATEDIFF(dd, OrderDate, RequiredDate) AS LeadTime,
DATEDIFF(dd, OrderDate, ShippedDate) AS DaysToShip,
DATEDIFF(dd, ShippedDate, RequiredDate) AS DayEarly
From dbo.Orders;
```

The screenshot shows the SQL Query window with the following T-SQL code:

```
use Northwind;
SELECT
    OrderDate, RequiredDate, ShippedDate,
    DATEDIFF(dd, OrderDate, RequiredDate) AS LeadTime,
    DATEDIFF(dd, OrderDate, ShippedDate) AS DaysToShip,
    DATEDIFF(dd, ShippedDate, RequiredDate) AS DayEarly
From dbo.Orders;
```

The Results window displays the following data:

	OrderDate	RequiredDate	ShippedDate	LeadTime	DaysToShip	DayEarly
1	1996-07-04 00:00:00.000	1996-08-01 00:00:00.000	1996-07-16 00:00:00.000	28	12	16
2	1996-07-05 00:00:00.000	1996-08-16 00:00:00.000	1996-07-10 00:00:00.000	42	5	37
3	1996-07-08 00:00:00.000	1996-08-05 00:00:00.000	1996-07-12 00:00:00.000	28	4	24

## 12. Using @@ROWCOUNT:

`@@ROWCOUNT` නිසා Function ස්ථාපිත කරනු ලබයි. Return කළ ප්‍රතික්‍රියා මූලික ප්‍රතික්‍රියා නිසා Rows විඳුත් සිංහල ප්‍රතික්‍රියා නිසා ප්‍රතික්‍රියා නිසා යුතු වේ.

### Example:

```
Select CompanyName
From dbo.Customers
WHERE City = 'Madrid'
Select @@ROWCOUNT AS MadridCustomerCount;
```

The screenshot shows the SQL Query window with the following T-SQL code:

```
use Northwind;
Select CompanyName
From dbo.Customers
WHERE City = 'Madrid'
Select @@ROWCOUNT AS MadridCustomerCount;
```

The Results window displays the following data:

	CompanyName
1	Bólido Comidas preparadas
2	FISSA Fabrica Inter. Salchichas S.A.
3	Romero y tomillo

	MadridCustomerCount
1	3

ចំណោះ ល្អ្ញាង Variables ជាអង្គភាពណាដើលបចាប់ផ្តើមជាមួយនឹង សញ្ញា @@ មាននំយថាតីសំដើម្បី Global Variables និង Global Variable មិនមែនជា Variables ទម្រតាស្រាវដែលគឺជាផ្លូវការនៃ Function ។



# បញ្ជីនៃគម្រោង 11: សិក្សាទី

## Controlling Flow

### **1. IF ELSE:**

If Condition តើនឹងធ្វើ execute ផ្សេងៗព្រមទាំង Statement ពេលដែល expression មានតម្លៃ Boolean = true ។ បើត្រង់ថា Condition ត្រូវបានគេធ្វើឡើង True នោះទេ តើ Else Block នឹងធ្វើការ execute វិញ្ញាបន្ទុង។

**Ex:**

```
use Northwind;
If (select COUNT(*) from dbo.Products where UnitPrice > 100) > 0
    select 'Found' As HighPrice
Else
    select 'Not Found' As HighPrice
```

**Output:**

```
SQLQuery1.sql - P... (PC2\enter (52))*
use Northwind;
If (select COUNT(*) from dbo.Products where UnitPrice > 100) > 0
    select 'Found' As HighPrice
Else
    select 'Not Found' As HighPrice
```

HighPrice
1 Found

### **2. BEGIN END:**

ភាគច្រើន យើងតែអាចប្រើប្រាស់ BEGIN END Block ត្រាប់ជាមួយនឹង IF ដើម្បីរាជ្យនៃការ multiple Statements ។

**Example:**

```
use Northwind;
If (select COUNT(*) from dbo.Products where UnitPrice > 100) > 0
    Begin
        Update dbo.Products
        Set UnitPrice = UnitPrice - .10
        Where UnitPrice > 100
        Select 'Ten cents removed from high prices' As Message
    End
Else
    Begin
        Update dbo.Products
        Set UnitPrice = UnitPrice + .10
```

```
Select 'Ten cents added to all prices' As Message
```

```
end
```

**Output:**

```
SQLQuery1.sql - P... (PC2\enter (52))*
use Northwind;
If (select COUNT(*) from dbo.Products where UnitPrice > 100) > 0
Begin
    Update dbo.Products
    Set UnitPrice = UnitPrice - .10
    Where UnitPrice > 100
    Select 'Ten cents removed from high prices' As Message
End
Else
Begin
    Update dbo.Products
    Set UnitPrice = UnitPrice + .10
    Select 'Ten cents added to all prices' As Message
End
```

Results	Messages		
	<table border="1"> <thead> <tr> <th>Message</th> </tr> </thead> <tbody> <tr> <td>1 Ten cents removed from high prices</td> </tr> </tbody> </table>	Message	1 Ten cents removed from high prices
Message			
1 Ten cents removed from high prices			

### 3. GOTO, RETURN, and Labels:

GOTO Statement គឺត្រូវបានប្រើប្រាស់ដើម្បីធ្វើការ jump ទៅកាន់ Label ណាមួយដែលបានកំណត់ឡើង ហើយនិងធ្វើការ execute ជាបន្ថែម។

**Example:**

```
If (3=3)
    GOTO JumpToHere;
Print 'you won''t see this';

JumpToHere:
Print 'you will see this';
```

**Output:**

The screenshot shows the SQL Query Editor window with the title bar "SQLQuery1.sql - P... (PC2)\enter (52)\*". Inside the editor, there is an IF block:

```

If (3=3)
    GOTO JumpToHere;
Print 'you won''t see this';

JumpToHere:
Print 'you will see this';
  
```

Below the editor is the "Messages" pane, which displays the output:

```

you will see this
  
```

RETURN Statement គឺត្រូវបានប្រើប្រាស់ដើម្បីធ្វើការ Terminates Execution ដោយគ្មានលក្ខខណ្ឌអ្នកចាំងអស់ ដើម្បីរាយការប្រើប្រាស់រាយការ return value ជាតិ៖លើនេះ សំរាប់ប្រើប្រាស់ជាមួយនឹង Stored Procedures នៅបុណ្យ។

#### Example:

```

If (select COUNT(*) from dbo.products where UnitPrice > 100) >0
Return
    Update dbo.Products
    Set UnitPrice = UnitPrice + .10
    Select 'Ten cents added to all prices' As Message
  
```

#### **4. CASE:**

Case Statement គឺមានលក្ខណៈប្រើប្រាស់ប្រហាក់ប្រចំបាបនឹង IF និង ELSE

#### Example1:

```

use Northwind;
Select CategoryName, Category =
    Case CategoryName
        When 'Meat/Poultry' Then 'Protein'
        When 'Seafood' Then 'Protein'
        When 'Grains/Cereals' Then 'Carbs'
        When 'Confections' Then 'Carbs'
        When 'Produce' Then 'Carbs'
        When 'Condiments' Then 'Seasonings'
        Else 'Other'
    End,
        Description
From dbo.Categories
Order by Category;
  
```

Output:

The screenshot shows the SQL Server Management Studio interface. In the top window, titled 'SQLQuery1.sql - P... (PC2\enter (52))\*', a T-SQL script is displayed. The script uses a CASE statement to map food categories from the 'Categories' table to a simplified category. The results are then ordered by the mapped category. In the bottom window, titled 'Results', the output is shown in a grid format with columns: CategoryName, Category, and Description.

```

use Northwind;
Select CategoryName, Category =
Case CategoryName
    When 'Meat/Poultry' Then 'Protein'
    When 'Seafood' Then 'Protein'
    When 'Grains/Cereals' Then 'Carbs'
    When 'Confections' Then 'Carbs'
    When 'Produce' Then 'Carbs'
    When 'Condiments' Then 'Seasonings'
    Else 'Other'
End,
Description
From dbo.Categories
Order by Category;

```

	CategoryName	Category	Description
1	Confections	Carbs	Desserts, candies, and sweet breads
2	Grains/Cereals	Carbs	Breads, crackers, pasta, and cereal
3	Produce	Carbs	Dried fruit and bean curd
4	Dairy Products	Other	Cheeses
5	Beverages	Other	Soft drinks, coffees, teas, beers, a...

Case Statement គឺវាជប្រើប្រាស់ជាមួយនឹង In ដើម្បីត្រូវបានក្លាយជាការប្រើប្រាស់បន្ថែមឡើត។

#### Example2:

```

use Northwind;
select CategoryName,Category =
Case
    When CategoryName In('Meat/Poultry','Seafood') Then
'Protein'
    When CategoryName In('Grains','Cereals') Then 'Carbs'
    When CategoryName ='Condiments' Then 'Seasonings'
    Else 'Other'
End,
Description
From dbo.Categories
Order By Category;

```

Output:

```
SQLQuery1.sql - P... (PC2\enter (52))*
use Northwind;
select CategoryName,Category =
    Case
        When CategoryName In('Meat/Poultry','Seafood') Then 'Protein'
        When CategoryName In('Grains','Cereals') Then 'Carbs'
        When CategoryName ='Condiments' Then 'Seasonings'
        Else 'Other'
    End,
    Description
From dbo.Categories
Order By Category;
```

	CategoryName	Category	Description
1	Confections	Other	Desserts, candies, and sweet breads
2	Dairy Products	Other	Cheeses
3	Grains/Cereals	Other	Breads, crackers, pasta, and cereal
4	Beverages	Other	Soft drinks, coffees, teas, beers, and ales

**Example3:**

```
Declare @Avg money
Select @Avg = AVG(UnitPrice) From dbo.Products
Select ProductName,UnitPrice, Ranking =
Case
    When UnitPrice > @Avg Then 'above average'
    Else 'below average'
End
From dbo.Products;
```

**Output:**

SQLQuery1.sql - P... (PC2\enter (52))\*

```


    Declare @Avg money
    Select @Avg = AVG(UnitPrice) From dbo.Products
    Select ProductName, UnitPrice, Ranking =
    Case
        When UnitPrice > @Avg Then 'above average'
        Else 'below average'
    End
    From dbo.Products;


```

	ProductName	UnitPrice	Ranking
1	Chai	18.10	below average
2	Chang	19.10	below average
3	Aniseed Syr...	11.10	below average
4	Chef Anton'...	24.30	below average
5	Chef Anton'...	23.585	below average

**Example4:**

```

From dbo.Products;
select Address = City +
    Case
        When Region Is Null Then ' ' + PostalCode
        Else ', ' + Region + ' ' + PostalCode
    end
From dbo.Customers;

```

**Output:**

```
SQLQuery1.sql - P... (PC2\enter (52))*
select Address = City +
    Case
        When Region Is Null Then ' ' + PostalCode
        Else ', ' + Region + ' ' + PostalCode
    end
From dbo.Customers;
```

The screenshot shows a SQL query in the 'Results' tab of SSMS. The query concatenates the 'City' and 'PostalCode' columns from the 'dbo.Customers' table, separated by a space if 'Region' is null, or by a comma, a space, the 'Region' name, a space, and 'PostalCode' otherwise. The results are displayed in a table with one column labeled 'Address' containing 8 rows of data.

	Address
1	Berlin 12209
2	México D.F. 05021
3	México D.F. 05023
4	London WA1 1DP
5	Luleå S-958 22
6	Mannheim 68306
7	Strasbourg 67000
8	Madrid 28023

## 5. WHILE:

While Statement គឺត្រូវបានរបៀប្រាស់ដើម្បីដំឡើងការ Loop នូវ Block Code នៅពេលផែលលក្ខខណ្ឌពិត ហើយវានឹងដោយបង្កើតឡើងការ នៅពេលផែលលក្ខខណ្ឌ មិនពិត ។ ប្រសិនបើ Code នៅក្នុង While មានច្បាស់ Lines នោះគឺត្រូវរបៀប្រាស់  
Begin End ភ្លាប់ជាមួយដើរ។  
ម្នាច់និញ្ញឡើត គឺយើងអាចរបៀប្រាស់ Condition មួយចំនួនដូចជា BREAK (ដើម្បីបញ្ចប់ Loop) និង CONTINUE (ដើម្បី skip) ជាមួយនឹង While បានដៃដើរ។

### Example:

```
use Northwind;
while (select AVG(UnitPrice) From dbo.Products) <= 12
Begin
    Update dbo.Products
        Set UnitPrice = UnitPrice * 1.01
    If (Select MAX(UnitPrice) From dbo.Products) > 150
        Break
    Else
        Continue
End;
```

### Output:

The screenshot shows a SQL query window titled "SQLQuery1.sql - P... (PC2\enter (52))". The query itself is as follows:

```

use Northwind;
while (select AVG(UnitPrice) From dbo.Products) <= 12
Begin
    Update dbo.Products
        Set UnitPrice = UnitPrice * 1.01
    If (Select MAX(UnitPrice) From dbo.Products) > 150
        Break
    Else
        Continue
End;

```

In the "Messages" tab, the output is:

```

Command(s) completed successfully.

```

## **6. WAITFOR:**

វាតិត្របានប្រើប្រាស់សំរាប់កំនត់ Query Processor ធ្វើការ execute នៅក្នុងកំឡូងពេលដាក់លាក់ណាមួយ។

### **Example:**

```

--Pause for ten seconds
Waitfor Delay '000:00:10';
Print 'Done';

--Pause until a certain time
Waitfor Time'23:23:00';
Print 'It is night';

```

### **Output:**

The screenshot shows a SQL query window titled "SQLQuery1.sql - P... (PC2\enter (52))". The query contains two sections of code:

```

--Pause for ten seconds
Waitfor Delay '000:00:10';
Print 'Done';

--Pause until a certain time
Waitfor Time'23:23:00';
Print 'It is night';

```

## **7. Ranking Results:**

Rank Functions គឺត្រូវបានប្រើប្រាស់សំរាប់ធ្វើការ Rank ដើម្បីនឹងយកចោរមានលំដាប់ ផែនវាមាន 4 ប្រភេទសំខាន់ៗ ដូចជា:

- > Rank : Return នូវលេខលំដាប់ Rank នៃ Row និមួយៗ ហើយនឹងរំលែងលេខ ប្រសិនបើមានលេខ ស្ថិន។
- > Dense\_Rank : Return នូវលេខលំដាប់ Rank នៃ Row និមួយៗ ហើយ ប៉ុណ្ណោះរំលែងលេខនៅទេដូចជាដាមានលេខស្មូនត្តិក់ជាយ៉ាង។
- > NTile : សំរាប់ Distributes នូវ Rows ទៅជាក្រុមដែលជាក់លាក់ណាមួយ ។
- > Row\_Number : Return នូវលេខរឿងនៃ Row ។

**Ex1:**

```
use Northwind;
Select ProductName, UnitPrice,
       ROW_NUMBER() over (order by UnitPrice Desc) As RowNum,
       RANK() Over (Order by UnitPrice Desc) As Rank,
       DENSE_RANK() Over (order by UnitPrice Desc) As Rank,
       NTILE(50) Over (order by UnitPrice Desc) As NTile50
From dbo.Products
Order By UnitPrice Desc;
```

**Output:**

The screenshot shows the SQL Query Editor window with the following details:

- Title Bar:** SQLQuery1.sql - P... (PC2\enter (52))\*
- Query Text:**

```
use Northwind;
Select ProductName, UnitPrice,
       ROW_NUMBER() over (order by UnitPrice Desc) As RowNum,
       RANK() Over (Order by UnitPrice Desc) As Rank,
       DENSE_RANK() Over (order by UnitPrice Desc) As Rank,
       NTILE(50) Over (order by UnitPrice Desc) As NTile50
From dbo.Products
Order By UnitPrice Desc;
```
- Results Tab:** Shows the output of the query as a table.
- Table Data:**

	ProductName	UnitPrice	RowNum	Rank	Rank	NTile50
1	Côte de Blaye	263.40	1	1	1	1
2	Thüringer Rostbratwurst	123.69	2	2	2	1
3	Mishi Kobe Niku	97.10	3	3	3	2
4	Sir Rodney's Marmalade	81.10	4	4	4	2
5	Camarvon Tigers	62.60	5	5	5	3
6	Raclette Courdavault	55.10	6	6	6	3
7	Manjimup Dried Apples	53.10	7	7	7	4
8	Tarte au sucre	49.40	8	8	8	4

**Ex2:**

```
use Northwind;
Select ProductName, UnitPrice,
       ROW_NUMBER() over (order by UnitPrice Desc) As RowNum,
```

```
RANK() Over (Order by UnitPrice Desc) As Rank,
DENSE_RANK() Over (order by UnitPrice Desc) As Rank,
NTILE(50) Over (order by UnitPrice Desc) As NTile50
From dbo.Products
Order By ProductName;
```

**Output:**

SQLQuery1.sql - P... (PC2\enter (52))\*

```
use Northwind;
Select ProductName, UnitPrice,
ROW_NUMBER() over (order by UnitPrice Desc) As RowNum,
RANK() Over (Order by UnitPrice Desc) As Rank,
DENSE_RANK() Over (order by UnitPrice Desc) As Rank,
NTILE(50) Over (order by UnitPrice Desc) As Ntile50
From dbo.Products
Order By ProductName;
```

	ProductName	UnitPrice	RowNum	Rank	Rank	NTile50
1	Alice Mutton	39.10	14	14	14	7
2	Aniseed Syr...	11.10	64	64	53	33
3	Boston Cra...	18.50	44	44	41	22
4	Camembert ...	34.10	19	19	18	10
5	Camarvon ...	62.60	5	5	5	3
6	Chai	18.10	46	45	42	23
7	Chang	19.10	42	41	39	21

**Ex3:**

```
Select ProductName, CategoryID, UnitPrice,
DENSE_RANK() Over (Partition By CategoryID
Order By UnitPrice Desc) As DenseRank
From dbo.Products
Order By CategoryID, UnitPrice Desc;
```

**Output:**

SQLQuery1.sql - P... (PC2\enter (52))\*

```

Select ProductName, CategoryID, UnitPrice,
       DENSE_RANK() Over (Partition By CategoryID
                           Order By UnitPrice Desc) As DenseRank
From dbo.Products
Order By CategoryID, UnitPrice Desc;

```

	ProductName	CategoryID	UnitPrice	DenseRank
1	Côte de Blaye	1	263.40	1
2	Ipoh Coffee	1	46.10	2
3	Chang	1	19.10	3
4	Chai	1	18.10	4
5	Chartreuse ...	1	18.10	4
6	Steeleye St...	1	18.10	4
7	Lakkalikööri	1	18.10	4
8	Outback La...	1	15.10	5
9	Laughing Lu...	1	14.10	6
10	Sasquatch ...	1	14.10	6
11	Pure'Sunsin...	1	10.09	7
12	Pure'Sunsin...	1	10.09	7
13	Rhönbräu Kl...	1	7.85	8
14	Pure'Sunsin...	1	5.05	9
15	Guaraná Fa...	1	4.60	10
16	Green Tea ...	1	4.05	11
17	Vegie-spread	2	48.39	1
18	Northwoods...	2	44.10	2
19	Sirop d'éable	2	31.45	3

Ex4:

```

Select ProductName, CategoryID, UnitPrice,
       DENSE_RANK() Over (Partition by CategoryID
                           Order By UnitPrice Desc) As DenseRankByCategory,
       DENSE_RANK() Over (Order By UnitPrice Desc) As
OverallDenseRank,
       DENSE_RANK() Over (Order By UnitsOnOrder Desc) As
OverallByUnitsOnOrder
From dbo.Products
Order By ProductName;

```

Output:

SQLQuery1.sql - P... (PC2\enter (52))\*

```

Select ProductName, CategoryID, UnitPrice,
       DENSE_RANK() Over (Partition by CategoryID
                           Order By UnitPrice Desc) As DenseRankByCategory,
       DENSE_RANK() Over (Order By UnitPrice Desc) As
                           OverallDenseRank,
       DENSE_RANK() Over (Order By UnitsOnOrder Desc) As
                           OverallByUnitsOnOrder
  From dbo.Products
 Order By ProductName;

```

Results

	ProductName	CategoryID	UnitPrice	DenseRankByCategory	OverallDenseRank	OverallByUnitsOnOrder
1	Alice Mutton	6	39.10	3	14	10
2	Aniseed Syr...	2	11.10	12	53	3
3	Boston Cra...	8	18.50	6	41	10
4	Camembert ...	4	34.10	5	18	10
5	Camarvon ...	8	62.60	1	5	10
6	Chai	1	18.10	4	42	10
7	Chang	1	19.10	3	39	6
8	Chartreuse ...	1	18.10	4	42	10
9	Chef Anton'...	2	24.30	5	29	10
10	Chef Anton'...	2	23.585	6	31	10
11	Chocolade	3	12.85	9	50	3
12	Côte de Blaye	1	263.40	1	1	10
13	Escargots d...	8	13.35	8	49	10
14	Filo Mix	5	7.10	7	62	10
15	Flotemysost	4	21.60	7	34	10
16	Geitost	4	2.60	10	67	10
17	Genen Sho...	2	17.15	10	44	10

# មេរីលទ្ធផល 12: សិក្សាតិ

## Stored Procedures

### **1. Definition:**

Stored Procedures គឺជា Batch of T-SQL Code ដែលត្រូវបានរក្សាទុកនៅក្នុង SQL Server ដោយផ្តល់ ដែលយើង ធ្វើប្រាក់រាជីម្លើធ្វើការ បញ្ជីនល្អីននៃ Application របស់យើង, ជាព្រឹត្តការងារស្ថុគ្មោះមួយចំនួនដែលនឹងការការពារក្នុង Application នៅក្នុង Application ។

Stored Procedures គឺអាចធ្វើការងារបានល្អីតែទាំងអស់ជូនដែល Select Statement ឬ Enforce Complex Business Logic ។

### **2. Create Stored Procedures:**

Syntax:

```

CREATE PROCEDURE [dbo].[spSyntaxExample]
    Input[@parameter1 datatype [length]]
    Output[@parameter2 datatype [length]] OUTPUT
    AS
        Int, Char
    BEGIN
        --Variable declaration
        [@parameter1 datatype [length]]

        --SQL statements
        .....
    END

    GO

```

Example:

```

Create Procedure dbo.ListEmployees
As
Set Nocount On

```

```
Select EmployeeID, LastName, FirstName
From dbo.Employees
Go
```

**Output:**

The screenshot shows the SQL Server Management Studio interface. On the left, the Object Explorer displays the database structure of 'Northwind'. In the center, the 'SQLQuery6.sql' query window contains the following T-SQL code:

```
Create Procedure dbo.ListEmployees
As
Set Nocount On
Select EmployeeID, LastName, FirstName
From dbo.Employees
Go
```

On the right, the 'Messages' pane shows the message: "Command(s) completed successfully."

**3. Executer Stored Procedures:**

ធ្វើមឺនុយ execute Stored Procedures នោះតើអាមេរិកប្រើប្រាស់ជាមួយនឹង exe ឬ execute Keyword ។

The screenshot shows the SQL Server Management Studio interface. The 'Results' tab of the query window displays the output of the executed stored procedure:

```
--exec dbo.ListEmployees;
execute dbo.listEmployees;
```

The results table shows the following data:

	EmployeeID	Last Name	First Name
1	1	Davolio	Nancy
2	2	Fuller	Andrew
3	3	Leverling	Janet
4	4	Peacock	Margaret
5	5	Buchanan	Steven

**4. Working with Parameters:**

Parameters គឺជា Variable ដើម្បី Store តម្លៃ Input ដើម្បីអាមេរិក Pass ទៅក្នុង Stored Procedure នៅពេល Execute ។  
ដើម្បី declare វានោះគឺត្រូវធ្វើការ set Data Type ទៅរាយដែលស្រួលបានច្រើននៅ គឺត្រូវបង្ហាញ  
ចំណាំរបស់រាជ្យបញ្ជី សញ្ញា Commas ។

**Ex:**

```
create proc dbo.ListEmployeesByCity
@City varchar(25)
As
Set nocount on
    Select LastName,FirstName,City
    From dbo.Employees
    Where City = @City;
```

**Output:**

```
SQLQuery1.sql - P... (PC2\enter (52))*
create proc dbo.ListEmployeesByCity
@City varchar(25)
As
Set nocount on
    Select LastName,FirstName,City
    From dbo.Employees
    Where City = @City;|
```

Messages

Command(s) completed successfully.

ដើម្បីធ្វើការ execute វានោះគឺយើងអាមេរិកប្រើប្រាស់តាមទំនួន 2 ប្រភេទជាមុនការ:

**Ex:**

```
--exec dbo.ListEmployeesByCity @City ='London';
exec dbo.ListEmployeesByCity 'London';
```

**Output:**

The screenshot shows the SQL Server Management Studio interface. In the top tab bar, the left tab is titled 'SQLQuery2.sql - P... (PC2\enter (53))\*' and the right tab is 'SQLQuery1.sql - P... (PC2\enter (52))\*'. The left tab contains the following T-SQL code:

```
--exec dbo.ListEmployeesByCity @City ='London';
exec dbo.ListEmployeesByCity 'London';
```

Below the tabs, there are two buttons: 'Results' and 'Messages'. The 'Results' button is selected. A table is displayed with the following data:

	Last Name	First Name	City
1	Buchanan	Steven	London
2	Suyama	Michael	London
3	King	Robert	London
4	Dodsworth	Anne	London

##### **5. Defining Optional Parameters and Default Values:**

គ្រឿង Parameters របស់ Stored Procedures តើអាចធ្វើការកំណត់ថា Default Value មួយដែលដឹងមានត្រូវបែងចែក ឬវាបានផ្តល់ជូន។ Default Value តិចជា រាយកទណ្ឌនៃ Parameters ប្រើប្រាស់នៅពេលដឹងបញ្ជីលក្ខណៈពេលដែលបង្កើតឡើង។

**Ex:**

```
create proc dbo.ListEmployeesOptionalCity
@City varchar(25)=NULL
As
Set Nocount on
    If @City Is Null
        Select LastName, FirstName, City
        From dbo.Employees;
    Else
        Select LastName, FirstName, City
        From dbo.Employees
        Where City =@City;
```

Output:

```
SQLQuery1.sql - PC... (PC2\enter (52))*
create proc dbo.ListEmployeesOptionalCity
@City varchar(25)=NULL
As
Set Nocount on
If @City Is Null
    Select LastName, FirstName, City
    From dbo.Employees;
Else
    Select LastName, FirstName, City
    From dbo.Employees
    Where City =@City;

Messages
Command(s) completed successfully.
```

នៅពេលដឹងឱ្យឯការ Execute វានោះ គឺជាមធារិយីករណ៍កំនត់តំបន់ទៅមិនមែន Parameters ដឹងត្រូវការនៅក្បាត់ ឬ Stored Procedures ដួងបាន។

	Last Name	First Name	City
1	Buchanan	Steven	London
2	Suyama	Michael	London
3	King	Robert	London
4	Dodsworth	Anne	London

## **6. Handling Output Parameters:**

Output Parameters តើធ្វើការ Return Single Value ដើម្បីត្រួតពិនិត្យ Declare ជាមួយនឹង OUTPUT Keyword នៅខាង ខាង statement ។

**Ex:**

```
create proc dbo.InsertShipper
    @CompanyName nvarchar(40) = NULL,
    @Phone nvarchar(24) = NULL,
    @ShipperID int = NULL OUTPUT
AS
Set Nocount on
If @CompanyName Is Null
    Set @ShipperID =0;
Else
    Begin
        Insert into dbo.Shippers(CompanyName,Phone)
        values (@CompanyName,@Phone);
        Set @ShipperID = SCOPE_IDENTITY();
    End
Go
```

```
ដើម្បីធ្វើការទាញ output parameter value នៅលើត្រូវធ្វើការ declare  
  
declare @NewShipperID int  
exec dbo.InsertShipper  
    @CompanyName = 'Express2You',  
    @Phone =Null,  
    @ShipperID = @NewShipperID OUTPUT;  
Select @NewShipperID As ShipperID;  
Go
```

```
SQLQuery3.sql - P... (PC2\enter (53))* SQLQuery1.sql - P... (P

declare @NewShipperID int
exec dbo.InsertShipper
    @CompanyName = 'Express2You',
    @Phone =Null,
    @ShipperID = @NewShipperID OUTPUT;
Select @NewShipperID As ShipperID;
Go
```

## **7. Using the RETURN Statement**

យើងអាចប្រើបាស់ Return Statement ដើម្បីធ្វើការ return តើលើ Scalar ត្រូវនោះអាចមានស្ថិតជា return 0 ដើម្បរា  
ត្រូវបានប្រើបាស់សំរាប់ធ្វើការ exit ។

**Ex:**

```
Create proc dbo.InsertShipperReturn
```

```

        @CompanyName nvarchar(40) = Null,
        @Phone nvarchar(24) = Null
    As
    Set nocount on
    if @CompanyName Is Null
        return 0;
    Else
        Begin
            Declare @NewID int;
            insert into dbo.Shippers(CompanyName, Phone)
            values(@CompanyName, @Phone);
            Set @NewID = SCOPE_IDENTITY();
            return @NewID
        End
    Go

```

ហើយយើងត្រូវប្រើប្រាស់ Syntax ដូចខាងក្រោមដើម្បីធ្វើការ execute:

```

Declare @NewShipperID int;
exec @NewShipperID = dbo.insertShipperReturn
    @CompanyName ='Express2You2',
    @Phone =Null;
Select @NewShipperID As ShipperID;

```

Output:

ShipperID	
1	5

#### **8. Working with Variables:**

ហើយអាមេរិក្សប្រាស់ Variables នៅក្នុង Stored Procedures ដើម្បីធ្វើការក្រោតចែងការណ៍នៃភាពពិភាក្សាឌីជីថល់ Programming Language ដើម្បីក្រោតចែងការណ៍នៃភាពពិភាក្សាឌីជីថល់

Ex:

```

declare @Num int, @String varchar(50);

Set @Num=1;
Set @String='Hello';
Print Cast(@Num As varchar(10))+ ' ' +@String;
Select @Num =2, @String='Hello World';
Print Cast(@Num As varchar(10)) + ' ' + @String;

```

**Output:**

The screenshot shows the SQL Server Management Studio interface. In the top tab bar, there are two tabs: "SQLQuery3.sql - P... (PC2\enter (53))\*" and "SQLQuery1.sql - P... (PC2\enter (52))\*". The "SQLQuery1.sql" tab is active, displaying the provided SQL code. Below the tabs, the main pane shows the code again. At the bottom, the "Messages" pane is open, showing the results of the execution: "1 Hello" and "2 Hello World".

```

declare @Num int, @String varchar(50);
Set @Num=1;
Set @String='Hello';
Print Cast(@Num As varchar(10))+ ' ' +@String;
Select @Num =2, @String='Hello World';
Print Cast(@Num As varchar(10)) + ' ' + @String;

1 Hello
2 Hello World

```

# លេខទៅលទ្ធផល 13: សិក្សាតី

## Triggers

### **1. Create Trigger:**

Triggers គឺជាប្រធែណ៍នៃ Procedures ដូចមានការផែនលក្ខណៈ Run Automatic នៅពេលដែល Predefined Event

ត្រូវបានកែតម្រូវដើម្បីធ្វើដំឡើង រាយការព្រមទាំង Defined ដើម្បី execute ដូចម្នប់ការប្រើប្រាស់ Insert, Update, Delete Operation ។

**Example:**

1. ស្មូលបង្កើត Table dbo.TestTrigger ដូចខាងក្រោម:

```
create Table dbo.TestTrigger
(Signal varchar(50));
Go

Insert into dbo.TestTrigger(Signal)
Values('Stop');

Insert into dbo.TestTrigger(Signal)
Values ('Go');
Go
```

The screenshot shows the SQL Query window with the following code:

```
SQLQuery1.sql - PC... (PC2\enter (51))*
create Table dbo.TestTrigger
(Signal varchar(50));
Go

Insert into dbo.TestTrigger(Signal)
Values('Stop');

Insert into dbo.TestTrigger(Signal)
Values ('Go');
Go
```

Below the query window, the Messages pane displays the results of the execution:

```
(1 row(s) affected)

(1 row(s) affected)
```

2. ស្មូលបង្កើត Trigger ណែនាំ dbo.NoDeleteStop ដូចខាងក្រោម:

```

Create Trigger dbo.NoDeleteStop
    ON dbo.TestTrigger
    For Delete
As
    Declare @TestValue varchar(50);

    Select @TestValue = Signal From deleted;
    If @TestValue = 'Stop'
        Begin
            Print 'Deleting Stop signals is not allowed.';
            Rollback Tran;
        End

```

The screenshot shows the SQL Query window with the following T-SQL code:

```

Create Trigger dbo.NoDeleteStop
    ON dbo.TestTrigger
    For Delete
As
    Declare @TestValue varchar(50);

    Select @TestValue = Signal From deleted;
    If @TestValue = 'Stop'
        Begin
            Print 'Deleting Stop signals is not allowed.';
            Rollback Tran;
        End

```

A tooltip in the bottom right corner of the code editor says: "CREATE TRIGGER is not currently supported in Transact-SQL IntelliSense."

Below the query window, the Messages pane shows the command completed successfully.

### 3. සුජකරුවා

```
Delete From dbo.TestTrigger Where Signal = 'Stop';
```

The screenshot shows the SQL Query window with the following T-SQL code:

```

Delete From dbo.TestTrigger Where Signal = 'Stop';

```

The execution results in an error message in the Messages pane:

Deleting Stop signals is not allowed.  
Msg 3609, Level 16, State 1, Line 1  
The transaction ended in the trigger. The batch has been aborted.

## 2. Create INSTEAD OF Trigger:

INSTEAD OF Triggers គឺនាច Support ជាមួយនឹង Table ដើម្បីផ្តល់ព័ត៌មានប្រចាំថ្ងៃ ហើយរក្សាទុកការណ៍ទូទៅ និងការកែតម្រូវការ។

តាមខាងក្រោមនេះ នឹងបង្ហាញពីការបង្កើត View មួយដែល Join រវាង Table Products និង Categories។ ជាដូចខាងក្រោមនេះ នឹងបង្ហាញពីការកែតម្រូវការ។

នេះយើងនឹងបង្កើត Trigger ដើម្បីការកែតម្រូវការ។

Ex:

1. បង្កើត View ដូចខាងក្រោម:

```
Create View dbo.CategoriesAndProducts
As
```

```
    Select C.CategoryName,
           P.ProductID, P.ProductName
      From dbo.Products As P Inner Join
           dbo.Categories As C ON
           P.CategoryID = C.CategoryID;
```

Go

```
SQLQuery1.sql - P... (PC2\enter (51))*
Create View dbo.CategoriesAndProducts
As
    Select C.CategoryName,
           P.ProductID, P.ProductName
      From dbo.Products As P Inner Join
           dbo.Categories As C ON
           P.CategoryID = C.CategoryID;
Go
```

Messages

Command(s) completed successfully.

2. សូមបើកការ SQL Statement ដើម្បីធ្វើការ Update View បើកតម្រូវការ។

```
Update dbo.CategoriesAndProducts
Set ProductName ='Chai Tea', CategoryName ='Drinks'
Where ProductID=1
```

The screenshot shows a SQL query window titled 'SQLQuery1.sql - P... (PC2\enter (51))\*' containing the following code:

```

Update dbo.CategoriesAndProducts
Set ProductName = 'Chai Tea', CategoryName = 'Drinks'
Where ProductID=1
  
```

Below the query window is a 'Messages' pane. It displays an error message in red text:

```

Msg 4405, Level 16, State 1, Line 1
View or function 'dbo.CategoriesAndProducts' is not updatable because it contains OUTPUT parameters or triggers.
  
```

### 3. ସ୍ଥାପନା କରିବାରେ ଏହାକିମ୍ବାନ୍ତିକ କାମକାଳୀମାତ୍ରରେ କରାଯାଇଲାଗଲା:

```

Create Trigger dbo.CategoriesAndProductsUpdate
ON dbo.CategoriesAndProducts
INSTEAD OF UPDATE
AS
SET NOCOUNT ON
    Update dbo.Products
    Set Products.ProductName =
        (Select inserted.ProductName
        From inserted)
    Where Products.ProductID=
        (Select inserted.ProductID From inserted);

    Update dbo.Categories
    Set Categories.CategoryName=
        (Select inserted.CategoryName
        From inserted)
    Where Categories.CategoryName=
        (Select deleted.CategoryName From deleted);
  
```

The screenshot shows the SQL Query Editor window with the title bar "SQLQuery1.sql - P... (PC2\enter (52))\*". The main pane displays the T-SQL code for creating a trigger:

```

Create Trigger dbo.CategoriesAndProductsUpdate
ON dbo.CategoriesAndProducts
INSTEAD OF UPDATE
AS
SET NOCOUNT ON
    Update dbo.Products
        Set Products.ProductName =
            (Select inserted.ProductName
            From inserted)
        Where Products.ProductID=
            (Select inserted.ProductID From inserted);

    Update dbo.Categories
        Set Categories.CategoryName=
            (Select inserted.CategoryName
            From inserted)
        Where Categories.CategoryName=
            (Select deleted.CategoryName From deleted);

```

Below the code, the status bar shows "Command(s) completed successfully." The bottom-left corner of the window has a "Messages" tab.

4. សរួលរក្សាន់ SQL Statement ដើម្បីធ្វើការ Update:

```

Update dbo.CategoriesAndProducts
Set ProductName ='Chai Tea', CategoryName='Drinks'
    Where ProductID = 1
Go

Select * From dbo.Products Where ProductID = 1;
Select * From dbo.Categories Where CategoryName ='Drinks';
Go

```

```

SQLQuery1.sql - P... (PC2\enter (52))*
Update dbo.CategoriesAndProducts
Set ProductName ='Chai Tea', CategoryName='Drinks'
Where ProductID = 1
Go

Select * From dbo.Products Where ProductID = 1;
Select * From dbo.Categories Where CategoryName = 'Drinks';
Go

```

	ProductID	ProductName	SupplierID	CategoryID	QuantityPerUnit	UnitPrice	Units
1	1	Chai Tea	1	1	10 boxes x 20 bags	18.10	39

	CategoryID	CategoryName	Description	Picture
1	1	Drinks	Soft drinks, coffees, teas, beers, and ales	0x151C2F00020000

### **3. Create a DDL Trigger:**

DDL (Data Definition Language) Triggers គឺជាប្រព័ន្ធដែលបានរក្សាទុកដោយប្រើការជាមួយនឹង Create, Alter និង Drop Statements ដើម្បីរក្សាទុកការប្រើប្រាស់ផ្ទាល់ខ្លួនដោយប្រើប្រាស់ Database Operations ។

តាមខ្លាប់រណ៍ខាងក្រោម បង្ហាញពីការបង្កើត Database Trigger ដើម្បីរក្សាទុកការបង្កើត Add Table ។

Ex:

1. បង្កើត Database Trigger ដូចខាងក្រោម:

```

Create Trigger NoNewTables
ON DATABASE
For Create_Table
As
    Print 'No new tables allowed in this database.';
    Rollback;
Go

```

```
SQLQuery1.sql - P... (PC2\enter (52))*
Create Trigger NoNewTables
ON DATABASE
For Create_Table
As
    Print 'No new tables allowed in this database.';
    Rollback;
Go
```

Messages

Command(s) completed successfully.

2. បន្ទាប់មកសូមធ្វើការបញ្ជីត Table ដើម្បីបន្ថែមនឹងទូលបាន Error:

```
Create Table dbo.TestDDLTrigger
(Single varchar(50));
Go
```

```
SQLQuery1.sql - P... (PC2\enter (52))*
Create Table dbo.TestDDLTrigger
(Single varchar(50));
Go
```

Messages

No new tables allowed in this database.  
Msg 3609, Level 16, State 2, Line 1  
The transaction ended in the trigger. The batch has been aborted.

យើងអាចប្រើប្រាស់ Disable ឬ Enable Keyword ដើម្បីធ្វើការបិទប៊ូកវាបណ្តាល់នាមពេលនេះ ជាដាក់លូបរាងចោរជាបន្ថែម។

```
Disable Trigger NoNewTables On Database;
Go
```

នូវករណីដើមប្រើបង្កើតការប្រើប្រាស់ Trigger ថាបន្ទាន់គឺអាចប្រើប្រាស់តាម Syntax ដូចខាងក្រោម៖

```
If Exists (Select * From sys.triggers
           Where parent_class=0 and name = 'NoNewTables')
Drop Trigger NoNewTable ON Database
```



# សិក្សាតិវេស៊ីន

Special Abilities:

- ត្រួចតាមដែនដែលបានរៀបចំឡើង និង បរិញ្ញាប្រព័ន្ធទាំងអស់
- សិក្សាតិវេស៊ីននឹងបានរៀបចំឡើងការងារដោយការបង្កើតរូបរាង
- បានរៀបចំឡើង ពេលវេលាដោយការងារដោយការបង្កើតរូបរាង
- វិញ្ញាបនប្រព័ន្ធដែលបានរៀបចំឡើងដោយការងារ
- សិក្សាតិវេស៊ីននឹងបានរៀបចំឡើង ពេលវេលាដោយការងារ
- សិក្សាតិវេស៊ីននឹងបានរៀបចំឡើង ពេលវេលាដោយការងារ

## ◀) Short Course 3 Months

Group Study

### គម្រោងនៃវឌ្ឍន៍ សំរួលទៅ ហិរញ្ញវត្ថុ និង ទេសចរណ៍ ការ និង

\$ 5 only	\$ 5 only	\$ 5 only
Joomla	Sony Vegas	SQL Server

ផែនការ

Every Saturday  
7 :00 am - 8 :30 am  
12:00 pm - 1 :30 pm  
5 :00 am - 6 :30 pm

ការបង្កើតរូបរាង

លើកទី 1: 16 - November - 2013  
លើកទី 2: 23 - November - 2013  
លើកទី 3: 30 - November - 2013

## Web Development

Short course:

\$120 / 6 Months

\$150 / 6 Months

1. HTML 5  
2. CSS 3  
3. Web Hosting

4. PHP  
5. PHP&MySQL  
6. PHP Projects

Internet និង WiFi តាមរយៈគឺតិចជាអ្នកបាន  
សិក្សាតិវេស៊ីន

## ◀) Short Course 3 Months

Group Study

### គម្រោងនៃវឌ្ឍន៍ សំរួលទៅ ហិរញ្ញវត្ថុ និង ទេសចរណ៍ ការ និង

\$120 / 6 Months	\$80 / 3 Months
1. ASP.Net 2. SQL Server	1. WordPress 2. Joomla

### Programming + Database

Short course:

\$120 / 6 Months

1. Access -I  
2. Access -II  
3. Access Projects

\$80 / 6 Months

4. VB.Net -I  
5. VB.Net -II  
6. SQL Server

## Video Editing

Short course:

\$150 / 6 Months

\$180 / 6 Months

1. Sony Vegas  
2. After Effects  
3. Adobe Premiere

4. Adobe Flash  
5. Avid Liquid  
6. Particle Illusion

Internet និង WiFi តាមរយៈគឺតិចជាអ្នកបាន  
សិក្សាតិវេស៊ីន

## ◀) Short Course 3 Months

Group Study

### គម្រោងនៃវឌ្ឍន៍ សំរួលទៅ ហិរញ្ញវត្ថុ និង ទេសចរណ៍ ការ និង

\$150 / 8 Months	\$30 / 3 Months
1. Typing 2. Word 2013 3. Excel 2013	4. PowerPoint 2013 5. Internet 6. Outlook 2013

### Graphic Design

Short course:

\$150 / 8 Months

1. Photoshop -I  
2. Illustrator -I  
3. InDesign -I

\$100 / 8 Months

4. CorelDraw -I  
5. Photoshop -II  
6. Illustrator -II  
7. InDesign -II

\$70 / 8 Months

4. CorelDraw -II  
5. Photoshop -III  
6. Illustrator -III  
7. InDesign -III

## Graphic Design

Short course:

\$150 / 8 Months

\$180 / 6 Months

1. Photoshop -I  
2. Illustrator -I  
3. InDesign -I

4. CorelDraw -I  
5. Photoshop -II  
6. Illustrator -II  
7. InDesign -II

Internet និង WiFi តាមរយៈគឺតិចជាអ្នកបាន  
សិក្សាតិវេស៊ីន

## ◀) Short Course 3 Months

Group Study

### គម្រោងនៃវឌ្ឍន៍ សំរួលទៅ ហិរញ្ញវត្ថុ និង ទេសចរណ៍ ការ និង

Study Time:

Monday - Saturday

Sunday

7:00 am - 8:00 pm  
8:00 am - 5:00 pm

Monday - Saturday

Sunday

7:00 am - 8:00 pm  
8:00 am - 5:00 pm

Monday - Saturday

Sunday

7:00 am - 8:00 pm  
8:00 am - 5:00 pm

Monday - Saturday

Sunday

7:00 am - 8:00 pm  
8:00 am - 5:00 pm

Monday - Saturday

Sunday

7:00 am - 8:00 pm  
8:00 am - 5:00 pm

Monday - Saturday

Sunday

7:00 am - 8:00 pm  
8:00 am - 5:00 pm

Monday - Saturday

Sunday

7:00 am - 8:00 pm  
8:00 am - 5:00 pm

Monday - Saturday

Sunday

7:00 am - 8:00 pm  
8:00 am - 5:00 pm

Monday - Saturday

Sunday

7:00 am - 8:00 pm  
8:00 am - 5:00 pm

Monday - Saturday

Sunday

7:00 am - 8:00 pm  
8:00 am - 5:00 pm

Monday - Saturday

Sunday

7:00 am - 8:00 pm  
8:00 am - 5:00 pm

Monday - Saturday

Sunday

7:00 am - 8:00 pm  
8:00 am - 5:00 pm

Monday - Saturday

Sunday

7:00 am - 8:00 pm  
8:00 am - 5:00 pm

Monday - Saturday

Sunday

7:00 am - 8:00 pm  
8:00 am - 5:00 pm

Monday - Saturday

Sunday

7:00 am - 8:00 pm  
8:00 am - 5:00 pm

Monday - Saturday

Sunday

7:00 am - 8:00 pm  
8:00 am - 5:00 pm

Monday - Saturday

Sunday

7:00 am - 8:00 pm  
8:00 am - 5:00 pm

Monday - Saturday

Sunday

7:00 am - 8:00 pm  
8:00 am - 5:00 pm

Monday - Saturday

Sunday

7:00 am - 8:00 pm  
8:00 am - 5:00 pm

Monday - Saturday

Sunday

7:00 am - 8:00 pm  
8:00 am - 5:00 pm

Monday - Saturday

Sunday

7:00 am - 8:00 pm  
8:00 am - 5:00 pm

Monday - Saturday

Sunday

7:00 am - 8:00 pm  
8:00 am - 5:00 pm

Monday - Saturday

Sunday

7:00 am - 8:00 pm  
8:00 am - 5:00 pm

Monday - Saturday

Sunday

7:00 am - 8:00 pm  
8:00 am - 5:00 pm

Monday - Saturday

Sunday

7:00 am - 8:00 pm  
8:00 am - 5:00 pm

Monday - Saturday

Sunday

7:00 am - 8:00 pm  
8:00 am - 5:00 pm

Monday - Saturday

Sunday

7:00 am - 8:00 pm  
8:00 am - 5:00 pm

Monday - Saturday

Sunday

7:00 am - 8:00 pm  
8:00 am - 5:00 pm

Monday - Saturday

Sunday

7:00 am - 8:00 pm  
8:00 am - 5:00 pm

Monday - Saturday

Sunday

7:00 am - 8:00 pm  
8:00 am - 5:00 pm

Monday - Saturday

Sunday

7:00 am - 8:00 pm  
8:00 am - 5:00 pm

Monday - Saturday

Sunday

7:00 am - 8:00 pm  
8:00 am - 5:00 pm

Monday - Saturday

Sunday

7:00 am - 8:00 pm  
8:00 am - 5:00 pm

Monday - Saturday

Sunday

7:00 am - 8:00 pm  
8:00 am - 5:00 pm

Monday - Saturday

Sunday

7:00 am - 8:00 pm  
8:00 am - 5:00 pm

Monday - Saturday

Sunday

7:00 am - 8:00 pm  
8:00 am - 5:00 pm

Monday - Saturday

Sunday

7:00 am - 8:00 pm  
8:00 am - 5:00 pm

Monday - Saturday

Sunday

7:00 am - 8:00 pm  
8:00 am - 5:00 pm

Monday - Saturday

Sunday

7:00 am - 8:00 pm  
8:00 am - 5:00 pm

Monday - Saturday

Sunday

7:00 am - 8:00 pm  
8:00 am - 5:00 pm

Monday - Saturday

Sunday

7:00 am - 8:00 pm  
8:00 am - 5:00 pm

Monday - Saturday

Sunday

7:00 am - 8:00 pm  
8:00 am - 5:00 pm

Monday - Saturday

Sunday

7:00 am - 8:00 pm  
8:00 am - 5:00 pm

Monday - Saturday

Sunday

7:00 am - 8:00 pm  
8:00 am - 5:00 pm

Monday - Saturday

Sunday

7:00 am - 8:00 pm  
8:00 am - 5:00 pm

Monday - Saturday

Sunday

7:00 am - 8:00 pm  
8:00 am - 5:00 pm

Monday - Saturday

Sunday

7:00 am - 8:00 pm  
8:00 am - 5:00 pm

Monday - Saturday

Sunday

7:00 am - 8:00 pm  
8:00 am - 5:00 pm

Monday - Saturday

Sunday

7:00 am - 8:00 pm  
8:00 am - 5:00 pm

Monday - Saturday

Sunday

7:00 am - 8:00 pm  
8:00 am - 5:00 pm

Monday - Saturday

Sunday

7:00 am - 8:00 pm  
8:00 am - 5:00 pm

Monday - Saturday

Sunday

7:00 am - 8:00 pm  
8:00 am - 5:00 pm

Monday - Saturday

Sunday

7:00 am - 8:00 pm  
8:00 am - 5:00 pm

Monday - Saturday

Sunday

7:00 am - 8:00 pm  
8:00 am - 5:00 pm

Monday - Saturday

Sunday

7:00 am - 8:00 pm  
8:00 am - 5:00 pm

Monday - Saturday

Sunday

7:00 am - 8:00 pm  
8:00 am - 5:00 pm

Monday - Saturday

Sunday

7:00 am - 8:00 pm  
8:00 am - 5:00 pm

# ក្រសួងពេទ្យ

Computer Course:

I. Microsoft Office			
1. Word 2013	\$10	5. VB Net Advance	\$40
2. Excel 2013	\$10	6. C# Programming	\$40
3. Internet	\$10	7. C# Advance	\$40
4. PowerPoint 2013	\$10	8. Java Program	\$40
5. Word Advance	\$20	9. Java Advance	\$40
6. Excel Advance	\$20	VII. Database	
II. Graphic Design			
1. Photoshop CS6-II	\$20	1. Access 2013	\$20
2. Illustrator CS6-II	\$20	2. Access Advance	\$30
3. CorelDraw X6-II	\$20	3. SQL Server 2012	\$40
4. InDesign CS6-II	\$25	4. Oracle 11g	\$40
5. Photoshop -II	\$30	VIII. Web Development	
6. Illustrator -II	\$30	1. HTML 5	\$25
7. CorelDraw -II	\$30	2. CSS 3	\$30
8. InDesign -II	\$30	3. JavaScript	\$30
III. Account/Statistics		4. JQuery	\$30
1. QuickBooks 2010	\$30	5. PHP	\$40
2. Peachtree 2010	\$30	6. PHP & MySQL	\$40
3. SPSS 16	\$40	7. ASP.Net -II	\$40
IV. Video Editing		8. ASP.Net -II	\$40
1. Sony Vegas 12	\$30	9. Joomla	\$50
2. Adobe Flash CS6	\$30	10. Wordpress	\$50
3. After Effects CS6	\$40	VIII. Repair/ Network	
4. Premiere CS6	\$40	1. Computer Basic	\$20
5. Particle Illusion	\$40	2. Computer Repair	\$30
6. Avid Liquid	\$40	3. Repair Advance	\$30
V. Programming		4. Peer To Peer	\$30
1. C Programming	\$25	5. Server 2008-II	\$40
2. C++	\$30	6. Server 2008-II	\$40
3. C++ OOP	\$30	7. Linux Fedora-II	\$50
4. VB.Net 2012	\$40	8. Linux Fedora-II	\$50

# គម្រោងខ្មែរ

Contact Us:

○ Tel : 010 603 314

: 012 603 314

○ Website: www.enteritc.com

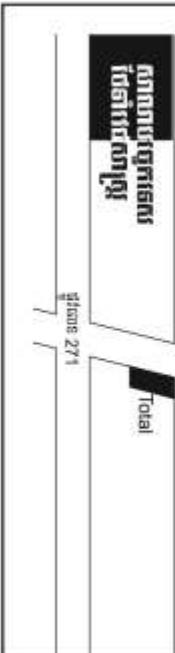
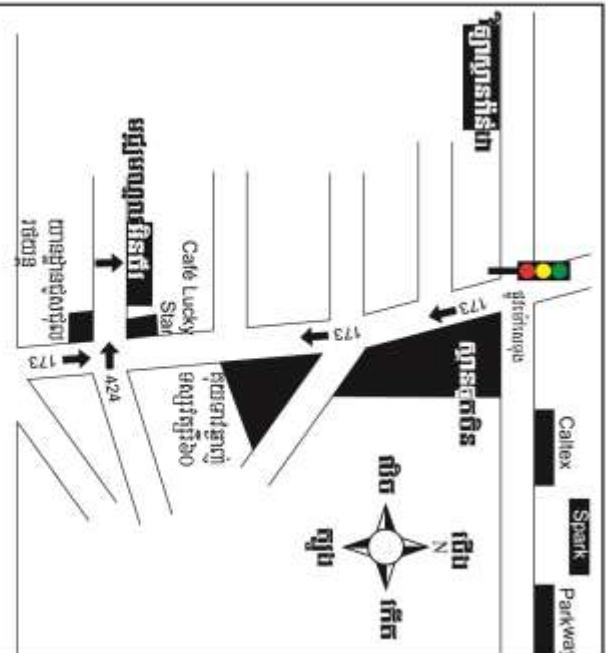
○ Email : info@enteritc.com

○ Page : facebook.com/enteritc

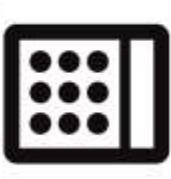
○ ផ្លូវលេខ 179 ផ្លូវ 424 តីកដែល 173 សង្កាត់ទុនប៊ែក  
ឧណ្ណ ចំការមេដ្ឋាន ហ្មិរ ភូមិពេជ្យ (ក្រុងរាជធានីភ្នំពេញ)

## គម្រោងខ្មែរ: បច្ចេកទេស

Enter Center of Information Technology



Web Development



Programming



Network