

## บทปฏิบัติการวิชา 674330 ระบบฐานข้อมูลขั้นสูง

### เรื่อง การสร้างอินเด็กซ์ และการสร้างวิว

วัตถุประสงค์: เมื่อจบบทปฏิบัติการนี้แล้ว นักศึกษาสามารถ

1. สร้างอินเด็กซ์แบบ Clustered index และ Nonclustered index ได้
2. สร้างและจัดการเกี่ยวกับวิว (View) ได้

#### 1. การสร้างอินเด็กซ์ด้วยโปรแกรม Microsoft SQL Server 2005

การสร้างอินเด็กซ์มีวัตถุประสงค์หลักก็เพื่อให้การค้นหาข้อมูลรวดเร็วขึ้น โดยเฉพาะอย่างยิ่งในกรณีที่ข้อมูลมีจำนวนมาก รูปแบบของอินเด็กซ์ในโปรแกรม Microsoft SQL Server 2005 มี 2 ประเภท ดังนี้

1. Clustered index
2. Nonclustered index

การสร้างอินเด็กซ์ สามารถทำได้ 2 วิธีดังนี้

1. การสร้างอินเด็กซ์พร้อมกับการสร้างตารางข้อมูล
2. การสร้างอินเด็กซ์หลังจากการสร้างตาราง

##### 1.1 การสร้างอินเด็กซ์พร้อมกับการสร้างตารางข้อมูล

คำสั่งต่อไปนี้เป็นการสร้างตาราง PERSONS พร้อมกับการสร้างอินเด็กซ์ โดยกำหนดให้ PersonID ซึ่งเป็น Primary Key เป็นอินเด็กซ์แบบ NONCLUSTERED และกำหนดให้ SocialSecurityNumber เป็นอินเด็กซ์ แบบ CLUSTERED พร้อมทั้งกำหนด constraint เป็น UNIQUE ทั้งนี้ การสร้างอินเด็กซ์พร้อมกับการสร้าง ตาราง ข้อมูลจะกระทำได้ด้วยเฉพาะฟิลด์ที่กำหนดเป็น Primary Key หรือ UNIQUE เท่านั้น

```
create table PERSONS
(
    PersonID smallint primary key nonclustered,
    FirstName varchar(30),
    LastName varchar(40),
    SocialSecurityNumber char(11) unique
    clustered
)
```

จากนั้นให้ใช้คำสั่งในการแสดงข้อมูลของอินเด็กซ์

```
sp_helpindex PERSONS
```

จากนั้นจะปรากฏหน้าจอในการแสดงข้อมูลของอินเด็กซ์ ดังนี้

	index_name	index_description	index_keys
1	PK_PERSONS_03317E3D	nonclustered, unique, primary key located on PRIMARY	PerseerID
2	UQ_PERSONS_023D5A04	clustered, unique, unique key located on PRIMARY	SocialSecurityNumber

คำสั่งต่อไปนี้เป็นตัวอย่างการสร้างตาราง PERSONS พร้อมกับการสร้างอินเด็กซ์ทั้งแบบ CLUSTERED และ NONCLUSTERED

```
create table PERSONS1
(
  PersonID smallint primary key clustered,
  FristName varchar(30),
  LastName varchar(40),
  SocialSecurityNumber char(11) unique
  nonclustered
)
```

จากนั้นใช้คำสั่งในการแสดงข้อมูลของอินเด็กซ์

**sp\_helpindex PERSONS1**

	index_name	index_description	index_keys
1	PK_PERSONS1_5F141958	clustered, unique, primary key located on PRIMARY	PersonID
2	UQ_PERSONS1_60083D91	nonclustered, unique, unique key located on PRIMA...	SocialSecurityNumber

## 1.2 การสร้างอินเด็กซ์ภายหลังการสร้างตารางข้อมูล

สามารถทำได้โดยการใช้คำสั่ง CREATE INDEX ดังตัวอย่างต่อไปนี้เป็น การสร้างอินเด็กซ์แบบ NONCLUSTERED สำหรับฟิลด์ FirstName

**CREATE INDEX INDEX\_FirstName  
ON PERSONS(FirstName)**

หลังจากการสร้างอินเด็กซ์สามารถแสดงข้อมูลของอินเด็กซ์ได้ดังนี้

**sp\_helpindex PERSONS**

	index_name	index_description	index_keys
1	index_FirstName	nonclustered located on PRIMARY	FirstName
2	PK_PERSONS__1920BF5C	clustered, unique, primary key located on PRIMARY	PersonID
3	UQ_PERSONS__1A14E395	nonclustered, unique, unique key located on PRIMA...	SocialSecurityNumber

ให้สร้างอินเด็กซ์ดังคำสั่งต่อไปนี้

```
CREATE UNIQUE NONCLUSTERED INDEX INDEX_FirstName  
ON PERSONS (FirstName, LastName DESC)  
WITH FILLFACTOR = 60,  
IGNORE_DUP_KEY,  
DROP_EXISTING,  
SORT_IN_TEMPDB
```

ความหมายของแต่ละคำสั่งย่อ มีดังนี้

1. เป็นการสร้างอินเด็กซ์แบบ NONCLUSTERED และแบบ UNIQUE โดยการนำเอาฟิลด์ FirstName และ LastName มารวมกันโดยเรียงลำดับจากมากไปหาน้อย (DESC)
2. WITH FILLFACTOR = 60 เป็นการให้สร้างโดยใช้พื้นที่เพียง 60% เหลืออีก 40% สำหรับเป็นพื้นที่ว่างเพื่อรองรับการเพิ่มข้อมูลในอนาคต
3. IGNORE\_DUP\_KEY เป็นการยกเลิกการเพิ่มข้อมูลถ้ามีข้อมูลซ้ำในอินเด็กซ์
4. DROP\_EXISTING เป็นการลบอินเด็กซ์ที่มีอยู่ก่อนแล้วสร้างใหม่ในชื่อเดิม
5. SORT\_IN\_TEMPDB ให้ทำการเรียงลำดับข้อมูลในฐานข้อมูล TEMPDB เพื่อให้ทำงานได้เร็วขึ้น

**sp\_helpindex PERSONS**

	index_name	index_description	index_keys
1	index_FirstName	nonclustered, ignore duplicate keys, unique located ...	FirstName, LastName(-)
2	PK_PERSONS__1920BF5C	clustered, unique, primary key located on PRIMARY	PersonID
3	UQ_PERSONS__1A14E395	nonclustered, unique, unique key located on PRIMA...	SocialSecurityNumber

## 2. การ Rebuild อินเด็กซ์

เป็นการเปลี่ยนแปลงโครงสร้างบางอย่างแล้วให้สร้างอินเด็กซ์ใหม่ โดยใช้คำสั่งดังตัวอย่างต่อไปนี้

```
DBCC DBREINDEX (PERSONS, INDEX_FirstName, 70)
```

คำสั่งนี้เป็นการ rebuild อินเด็กซ์ชื่อ INDEX\_FirstName โดยกำหนดให้ FILLFACTOR เท่ากับ 70%

### 3. การเปลี่ยนชื่ออินเด็กซ์

คำสั่งที่ใช้มีดังนี้

```
sp_rename  
@objname='PERSONS.INDEX_FriatName',  
@newname='INDEX_Frist_LastName',  
@objtype='INDEX'
```

ให้ใช้คำสั่ง sp\_helpindex PERSONS ตรวจสอบการเปลี่ยนชื่ออินเด็กซ์

### 4. การลบอินเด็กซ์

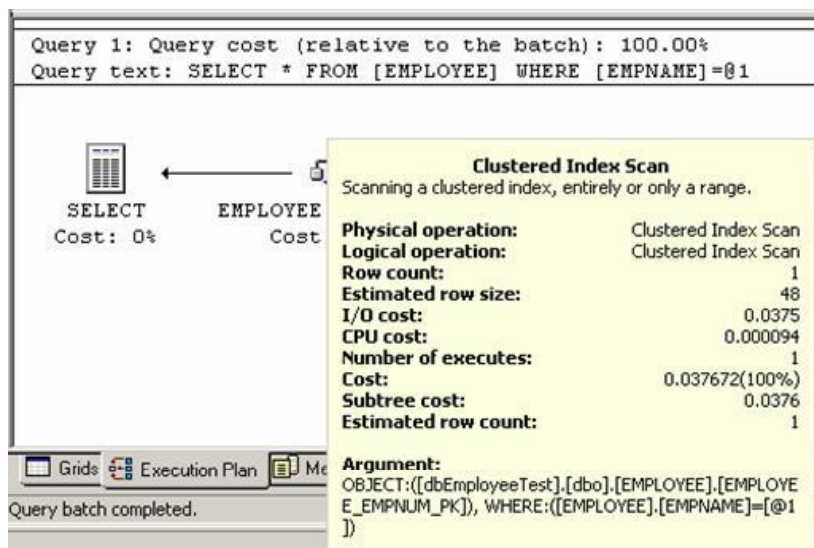
คำสั่งที่ใช้ในการลบ ได้แก่ **DROP INDEX** ชื่อตาราง.ชื่ออินเด็กซ์ ดังตัวอย่างต่อไปนี้

```
SELECT * FROM EMPLOYEE WHERE POSITION='Manager'
```

จากนั้นให้ทำการคลิกที่แท็บ Execution Plan



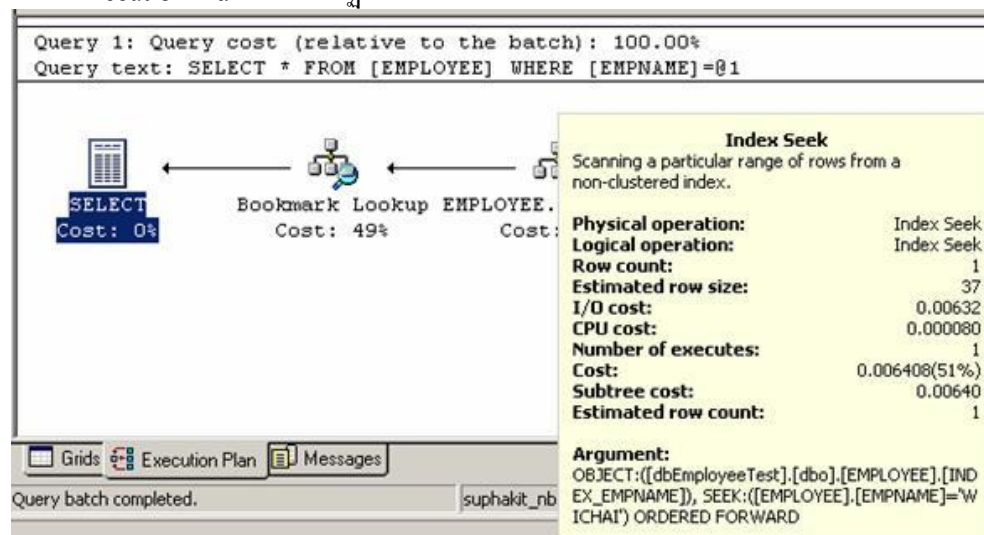
หลังจากนั้นจะปรากฏหน้าจอที่แสดงถึง Cost แล้วให้นำเมาส์ไปชี้ที่รูปจะปรากฏข้อความดังนี้



ให้ทำการสร้างอินเด็กซ์ชื่อ index01 ฟิวด์ POSITION ที่ตาราง Employee ดังนี้

```
CREATE INDEX index01 on EMPLOYEE (POSITION)
```

เมื่อสร้างเสร็จแล้วให้ลองทำการรันคำสั่ง SQL ในการเรียกข้อมูลเดิมอีกครั้ง แล้วทำการคลิกที่แท็บ Execution Plan จะปรากฏหน้าจอดังนี้



จากการเปรียบเทียบการเรียกค้นข้อมูลจากก่อนการสร้างอินเด็กซ์และหลังจากการสร้างอินเด็กซ์ปรากฏข้อมูลเปรียบเทียบที่สำคัญ ดังนี้

รายการ	ก่อนการสร้างอินเด็กซ์	หลังการสร้างอินเด็กซ์
Estimated row size	0.0375	0.00632
I/O cost	48	37
CPU cost	0.000094	0.000080
CPU cost	0.037672(100%)	0.006408(51%)
Cost	0.0376	0.00640
Subtree cost		

## 5. การสร้างวิว

วิว (View) เปรียบเสมือนหน้าต่างของข้อมูลจากตารางหนึ่ง ๆ ที่ถูกเรียกเพื่อทำการเปลี่ยนแปลงข้อมูล วิวเป็นตารางเสมือนจริง (Virtual Table) ที่มีข้อมูลมาจากตารางหลัก (Base Table) โดยไม่มีข้อมูลเก็บอยู่จริงประโยชน์ของวิว คือ ผู้ใช้ในระบบสามารถเรียกดูข้อมูลจากฐานข้อมูลในลักษณะที่ผู้ใช้นั้น ๆ ต้องการ โดยใช้ คำสั่ง ง่าย ๆ ซึ่งอาจจะเป็นการเรียกดูข้อมูลจากตารางเดียวหรือหลายตารางก็ได้ นอกจากนี้วิวยังทำให้ผู้ใช้ข้อมูล มีความเป็นอิสระในการเรียกใช้ข้อมูลหรือแก้ไขข้อมูลจากวิวที่ผู้ใช้นั้น ๆ สร้างขึ้น

การสร้างวิวทำได้โดยใช้คำสั่ง CREATE VIEW ดังตัวอย่างต่อไปนี้

การสร้างวิวชื่อ EMPVIEW ซึ่งมีข้อมูลรหัสพนักงาน ชื่อ วันที่เริ่มทำงาน และเงินเดือน

```
CREATE VIEW EMPVIEW  
AS  
SELECT EMPNUM, EMPNAME, HIREDATE, SALARY  
FROM EMPLOYEE
```

จากนั้นให้ทำการเรียกดูข้อมูลจาก EMPVIEW

```
SELECT * FROM EMPVIEW
```

### ลองทำดู 1

เงื่อนไข

ให้สร้างวิวชื่อ EMPDEP ซึ่งมีข้อมูลของชื่อพนักงาน ชื่อแผนก เงินเดือน โดยให้แสดงเฉพาะพนักงานที่มีเงินเดือนมากกว่า 20000

คำสั่งที่ใช้

.....  
.....  
.....

### ลองทำดู 2

เงื่อนไข

ให้สร้างวิวชื่อ DEPSAL ประกอบด้วยข้อมูลรหัสแผนก เงินเดือนสูงสุด และเงินเดือนต่ำสุดของแต่ละแผนก โดยให้ตั้งชื่อ คอลัมน์ใหม่ในวิวเป็น DEPNO MAXSALARY และ MINSALARY ตามลำดับ

คำสั่งที่ใช้

```
CREATE VIEW DEPSAL (DEPNO, MAXSALARY, MINSALARY)
```

```
AS
```

.....  
.....  
.....

## 6. การลบวิว

ผู้ที่สร้างวิวขึ้นมาจะทำการลบวิวที่สร้างขึ้นได้ตามความต้องการ โดยการลบวิวจะไม่กระทบถึงตารางหลักที่วิวนั้นใช้ในการสร้างวิวขึ้นมา

ตัวอย่างคำสั่งในการลบวิวที่ชื่อ EMPVIEW

```
DROP VIEW EMPVIEW
```

## 7. การใช้ WITH CHECK OPTION

คำสั่งต่อไปนี้เป็นการสร้างวิวชื่อ EMPDEP2 ประกอบด้วย ข้อมูลรหัสพนักงาน ชื่อพนักงานของ พนักงานที่มีรหัสแผนก 20 โดยกำหนดให้ มี WITH CHECK OPTION ซึ่งผลของคำสั่งนี้ หากมีการเพิ่มเติม แก้ไข หรือลบข้อมูลของรหัสแผนกที่ไม่ใช่ 20 ตามที่ระบุไว้ที่อนุประโยค WHERE เครื่องจะแสดงข้อความว่าไม่ถูกต้องทันที

```
CREATE VIEW EMPDEP2
AS
SELECT EMPNUM, EMPNAME
FROM EMPLOYEE
WHERE DEPNO = '20'
WITH CHECK OPTION
```

## ลองทำดู 3

### เงื่อนไข

ให้สร้างวิวชื่อ EMPTEMP ซึ่งประกอบด้วยรหัส ชื่อ ตำแหน่งงาน และรหัสแผนกของพนักงาน และมีอนุประโยค WITH CHECK OPTION เพื่อตรวจสอบว่าการใส่รหัสแผนกในตาราง EMPLOYEE เป็นรหัสที่มีอยู่ในตาราง DEP เท่านั้น

### คำสั่งที่ใช้

.....

.....

.....

.....

## 8. การปรับปรุงแก้ไขข้อมูลในวิว

การปรับปรุงแก้ไขข้อมูลในวิวสามารถทำได้โดยใช้คำสั่ง INSERT UPDATE หรือ DELETE ทำนองเดียวกับการใช้ในการปรับปรุงแก้ไขข้อมูลในตารางหลัก แต่การปรับปรุงแก้ไขข้อมูลในวิวนั้นจะไม่สามารถทำได้ถ้าวิวนั้นๆ มีข้อมูลที่เกิดจากการสร้างวิวที่มีคุณสมบัติในข้อใดข้อหนึ่งต่อไปนี้

คุณสมบัติของวิวที่ไม่สามารถปรับปรุงแก้ไขได้	DELETE	UPDATE	INSERT
1. ข้อมูลในวิวนั้นเกิดจากการเชื่อมโยงข้อมูล (JOIN)	✓	✓	✓
2. มีการใช้ฟังก์ชันเกี่ยวข้องกับการรวม (Group Functions)	✓	✓	✓
3. มีการใช้อนุประโยค GROUP BY	✓	✓	✓
4. มีการใช้ DISTINCT	✓	✓	✓
5. มีการใช้ข้อความย่อย (Subquery)	✓	✓	✓
6. คอลัมน์บางคอลัมน์ในวิวถูกกำหนดขึ้นในลักษณะของนิพจน์ที่เป็นการคำนวณทางคณิตศาสตร์		✓	
7. คอลัมน์ในตารางหลักที่จะต้องไม่เป็นค่าว่าง (NOT NULL) ไม่ได้ถูกรวมอยู่ในวิว			✓