



MS SQL SERVER TUNING PERFORMANCE

OUR TEAM



อั้งค์รา ต้วนเทศ
1163109060546



วรากรณ์ พรีเขียว
1163109061072



ณัฐวุฒิ พินสำเนียง
1163109060470



ปพล ใจกัน
1163109060348

คำจำกัดความของ MICROSOFT SQL SERVER

Microsoft SQL Server เป็นซอฟต์แวร์ซินิดของ Relational Database Management System (RDBMS) ที่ใช้บอย ซอฟต์แวร์นี้พัฒนาโดย บริษัท Microsoft ขนาดใหญ่และสามารถปรับขนาดได้ ความตั้งใจคือการใช้ Microsoft SQL Server ทุกที่ จากแล็บก็อปไป ไปยังเครือข่ายคลาวด์เซิร์ฟเวอร์และอื่น ๆ เป็นตัวเลือกของผู้ใช้ฐานข้อมูลและยังคงใช้กันอย่างแพร่หลาย การพัฒนามีแนวโน้มค่อนข้างมากตั้งแต่ SQL Server 1.0 เวอร์ชันแรกที่วางจำหน่ายจนถึงปัจจุบัน ฟังก์ชันนี้ไม่เพียง แต่เป็น RDBMS (ระบบการจัดการฐานข้อมูลเชิงสัมพันธ์) เท่านั้น แต่ยังสามารถใช้งานได้



ทำความเข้าใจกับ MICROSOFT SQL SERVER

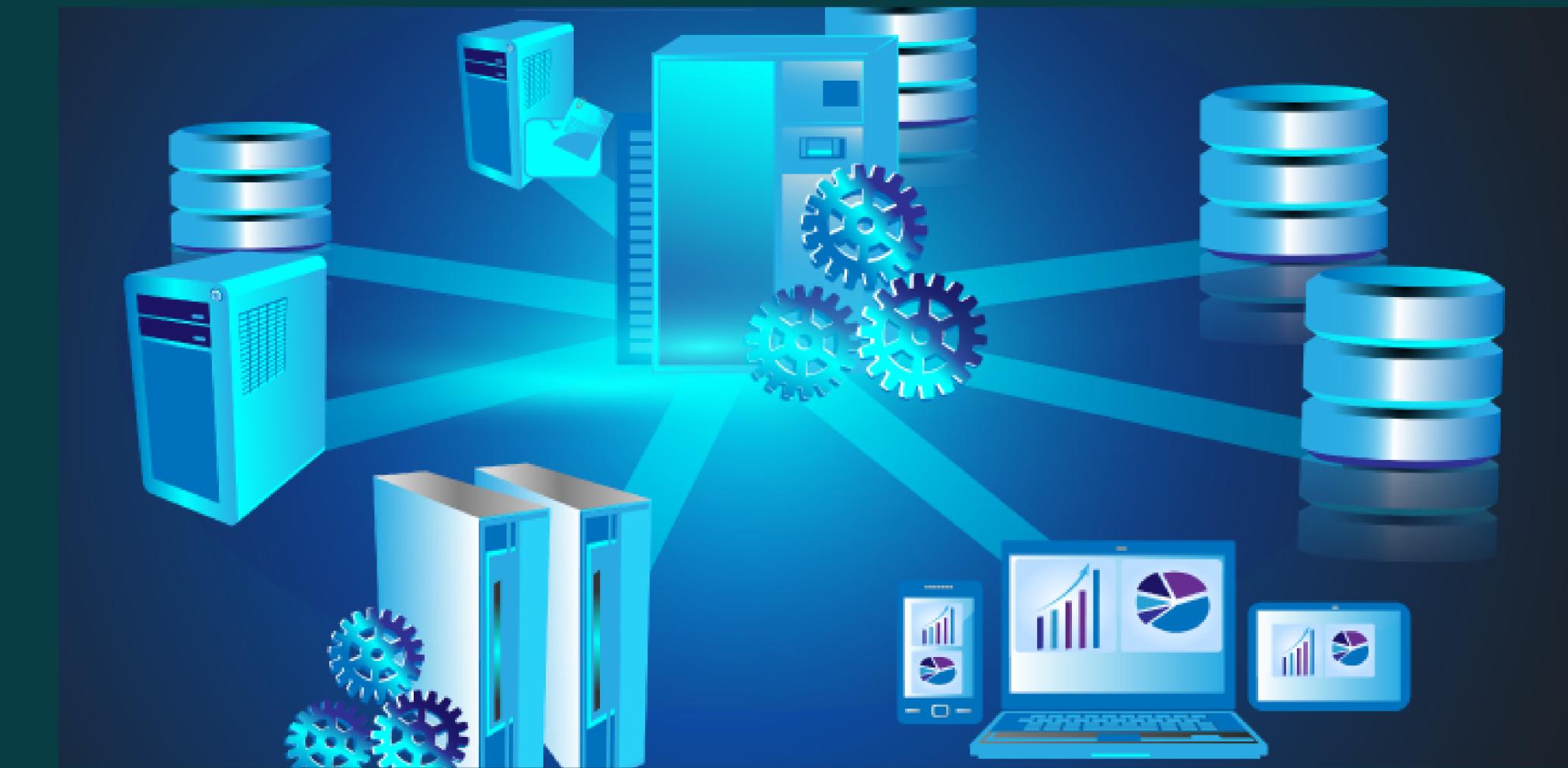
Microsoft SQL Server คืออะไร

เป็นโปรแกรมที่ใช้ในการจัดเก็บ บริหารและจัดการฐานข้อมูล จากทาง Microsoft ซึ่งมันถูกพัฒนาขึ้นมาจากSQL ซึ่งเป็นภาษาการเขียนโปรแกรมมาตรฐานสำหรับ การต่อตัวกับฐานข้อมูลเชิงสัมพันธ์ เซิร์ฟเวอร์ SQL เชื่อมโยงกับ Transact-SQL หรือ T-SQL

โดยเป็นโปรแกรมจัดการฐานข้อมูลแบบ Relational Database Management System หรือที่เรียกว่า RDBMS สามารถติดตั้งได้บน Windows, Linux, และ Docker containers ส่วนใหญ่ใช้บนระบบเครือข่าย เช่น เว็บไซต์ ร่วมกับภาษา คอมพิวเตอร์อื่น เช่น PHP, JAVA



ประเภทของ DATABASE ใน MICROSOFT SQL SERVER



MASTER DATABASE

มีความสำคัญมากสุด ใช้สำหรับเก็บข้อมูลที่สำคัญของระบบ โดยหากฐานข้อมูลนี้มีปัญหา ก็จะส่งผลกระทบต่อการทำงานของ Microsoft SQL Server ได้

MSDB DATABASE

สำคัญ รองจาก Master เก็บข้อมูลเกี่ยวกับ Alert, Job, Schedule, Backup ซึ่งถูกใช้โดยบริการของ SQL Server Agent ซึ่งเป็นบริการที่ทำงานอัตโนมัติ

DISTRIBUTION DATABASE

ใช้ในการ Replication ของฐานข้อมูล เพื่อกำให้ข้อมูลของ Microsoft SQL Server แต่ละที่มีความตรงกัน

ประเภทของ DATABASE ใน MICROSOFT SQL SERVER



MODEL DATABASE

เป็นฐานข้อมูลที่เป็นต้นแบบ (Database Template) กรณีที่เราสร้าง Database ใหม่ Microsoft SQL Server จะเอา Model Database นี้มาเป็นตัวตั้งต้น

TEMPDB DATABASE

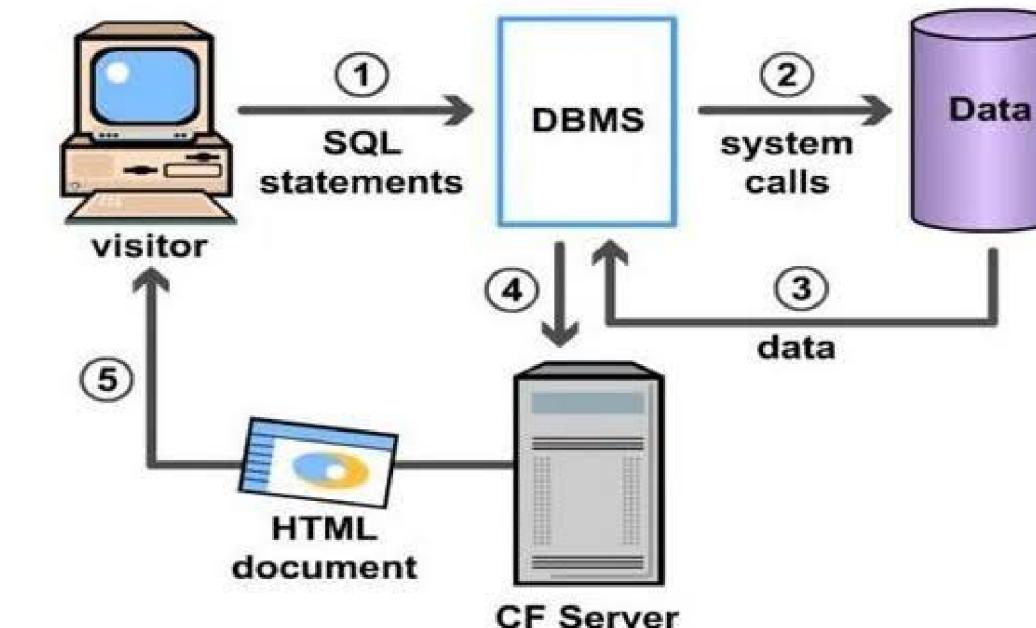
ใช้ในการเก็บข้อมูลที่เป็นชั่วคราว Temporary สำหรับ process ที่จำเป็นต้องการนำ data มาพักรอไว้ ก่อนแล้วค่อยนำไปทำงานอื่นต่อ ลบทุกครั้งที่มีการ Restart Service หรือ Shutdown ระบบ

ฟังก์ชัน *MICROSOFT SQL SERVER*

Microsoft SQL Server นั้นมีหน้าที่สำคัญในโลกของการจัดการฐานข้อมูล ซอฟต์แวร์นี้ทำการจัดการฐานข้อมูลโดยใช้คำสั่งแบบสอบถามหรือคำสั่ง SQL หรือ Structured Query Language เป็นภาษาคอมพิวเตอร์ที่เป็นที่รู้จักกันโลกและใช้ในการเข้าถึงข้อมูลที่เก็บไว้ในฐานข้อมูล

ฟังก์ชันนี้เชื่อมต่อกับกิจกรรมมากในโลกแห่งการเขียนโปรแกรม เพราะนักพัฒนาหรือโปรแกรมเมอร์ใช้ Microsoft SQL Server แม้ว่าจะไม่ได้อยู่ในภาษาการเขียนโปรแกรม ก็ยังสามารถมีอิทธิพลต่อ Microsoft SQL Server ที่มีประโยชน์อย่างมากในการพัฒนาโปรแกรมหรือแอปพลิเคชันที่ใช้เซิร์ฟเวอร์ฐานข้อมูล คุณลักษณะอื่นที่ค้นหาจาก Microsoft SQL Server คือความสามารถในการสร้างการมีเรอർและการกำคลัสเตอร์ฐานข้อมูล

DBMS (Database Management System)





ทำไมต้องมีการทำ **PERFORMANCE TUNING**

01

เมื่อต้องการให้ระบบมีความเร็วในการ **ACCESS** ข้อมูลภายใน **DATABASE** เพิ่มขึ้น

02

เมื่อต้องการให้ระบบสามารถรับ **LOAD** ของ **REQUEST** ที่เข้ามาได้มากขึ้น ตลอดจนถึง **CONNECTION** ด้วย

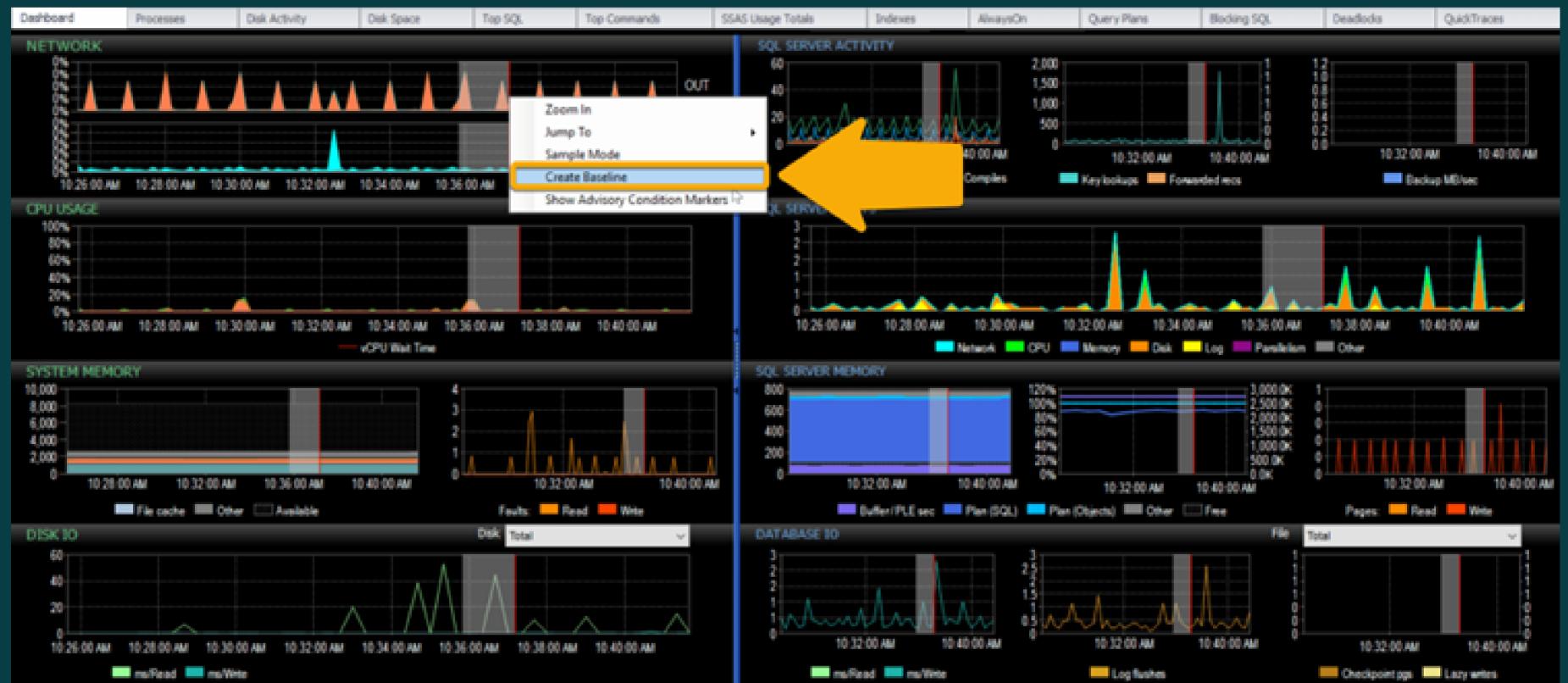
03

เมื่อต้องการให้ระบบมีความเสถียรภาพที่ดีขึ้น ไม่ **HANG** กรณีเจอ **PEAK LOAD**

สาเหตุที่เกิดปัญหา DESIGN DATABASE ไม่ดี หรือดีเกินไป

ในการออกแบบฐานข้อมูล จะมีการ Normalize Table แต่ตารางเป็นหลายตาราง ให้มีความซ้ำซ้อนน้อยที่สุด เพื่อให้สะดวกต่อการ Insert, Update, Delete แต่เมื่อเราต้องการดึงข้อมูล (Select) จากตารางต่าง ๆ ขึ้นมาใช้ ทำให้ต้อง Join, Subquery หรือแม้กระทั้ง Union จากหลายตารางมากเกินไป เพื่อให้ได้ผลลัพธ์ที่ต้องการ ซึ่งทำให้ช้าเกินกว่าที่รับได้ ถ้าข้อมูลมีจำนวน Record มากขึ้นเกินระดับหนึ่ง





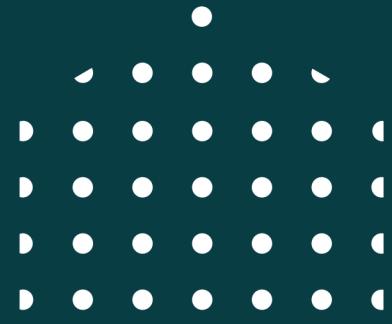
รู้ได้อย่างไรว่า MICROSOFTSQL SERVER มีประสิทธิภาพแย่ลง

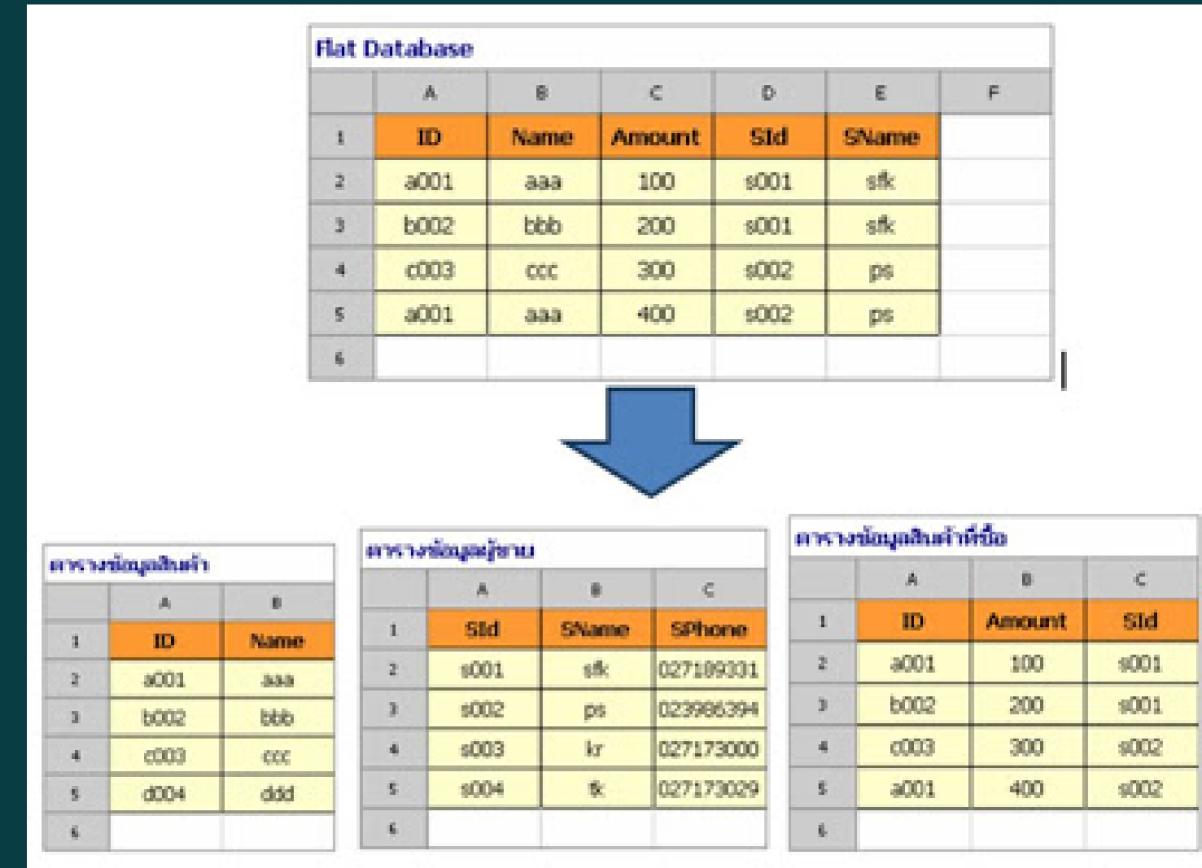
การดู Performance Baseline

Performance Baseline คือบรรดาค่าเกณฑ์วัดประสิทธิภาพพื้นฐานที่เก็บมาตอน Microsoft SQL Server ทำงานในสภาพปกติ แต่ก็ไม่ใช่สภาพที่ไม่มีการทำงานอะไรเลย แต่เป็นสภาพที่มีการทำงานปกติ และเป็นสภาพที่ใช้เป็นเกณฑ์

ประโยชน์ที่ได้

- สามารถเห็นแนวโน้มด้านประสิทธิภาพที่เปลี่ยนแปลงไปได้
- สามารถนำสภาวะปัจจุบันไปเทียบกับ **BASELINE** เพื่อค้นพบการเปลี่ยนแปลง แล้วแก้ปัญหาได้ง่ายขึ้น
- ช่วยประเมินการบริโภคทรัพยากร และวางแผนทรัพยากรล่วงหน้าได้

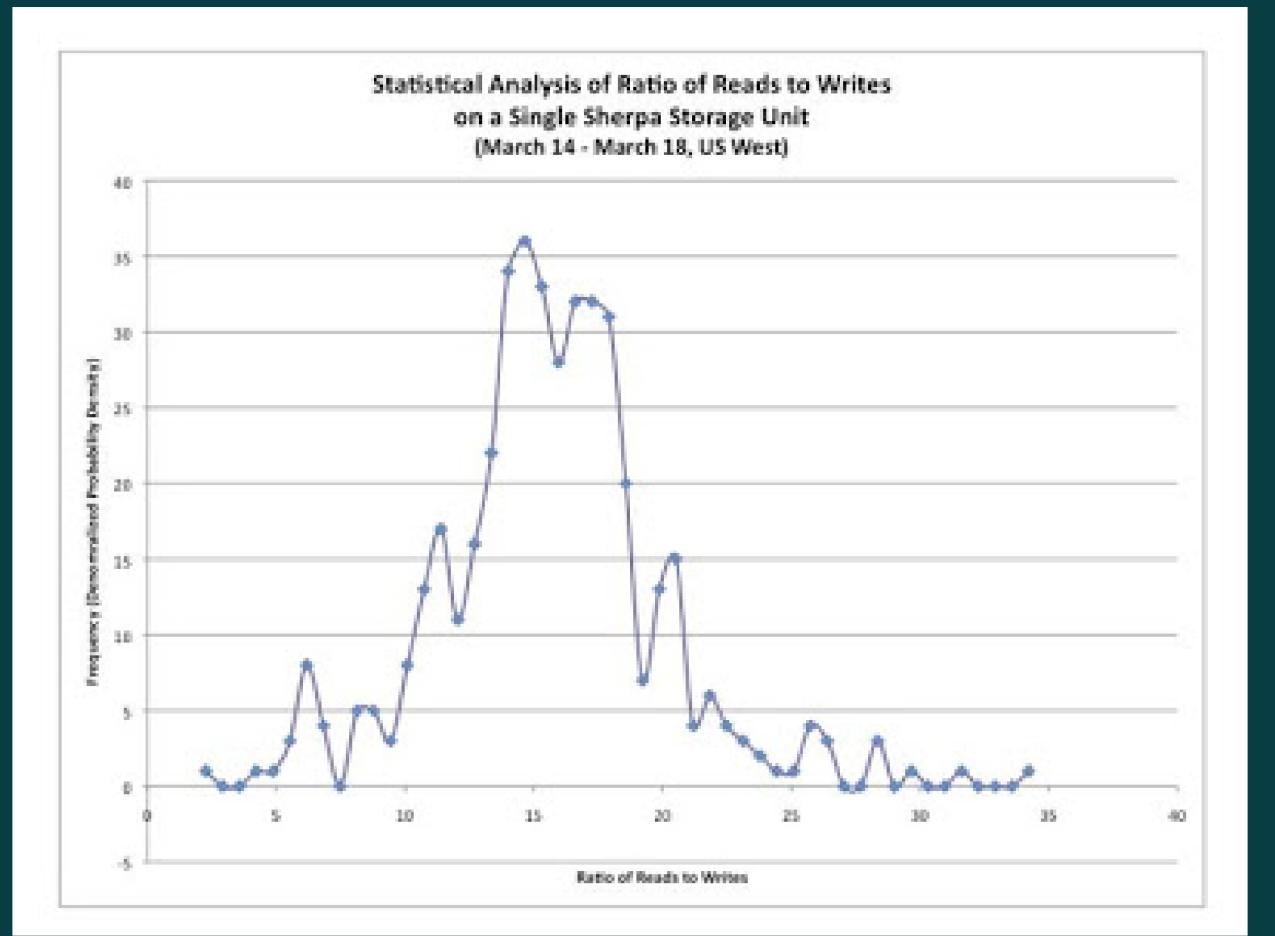




เทคนิคในการทำ PERFORMANCE TUNING

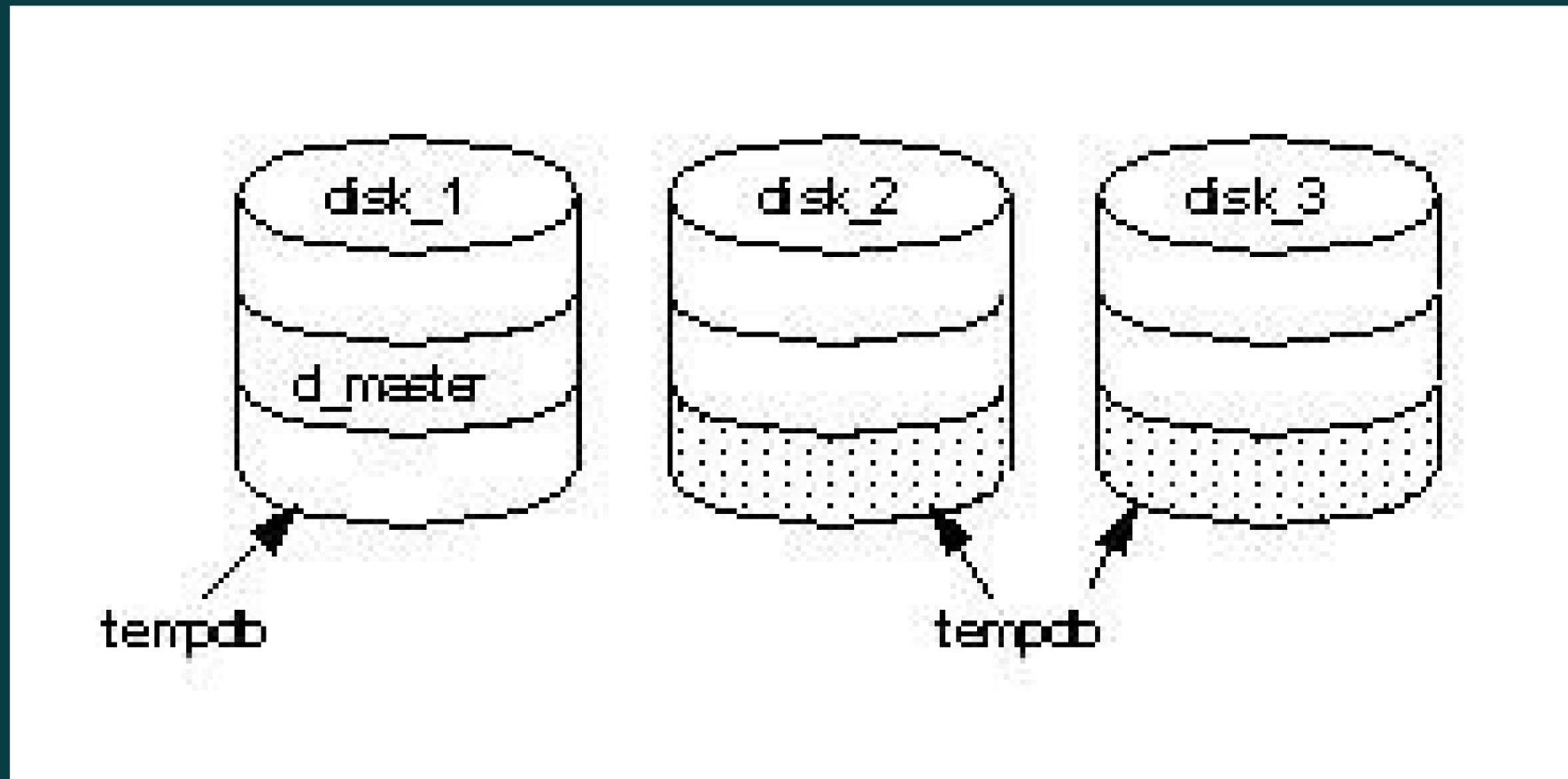
1) ใช้ Normalization เพื่อ Optimize Transactional Process

การทำ Normalization นั้น จะช่วยในมุ่งการใช้คำสั่ง Insert, Update, Delete ทำให้เกิด Performance ที่ดี เพราะจะช่วยขจัดปัญหา เรื่องการซ้ำซ้อนของข้อมูล และลดโอกาสที่จะเกิดการผิดพลาดในการประมวลผลข้อมูลในตาราง แต่จะมีผลกระทบกับคำสั่ง Select เพราะว่าใน Database หลาย ๆ ตัวที่อาจจะพบว่า Query เร็ว แต่ Insert อาจจะช้าได้



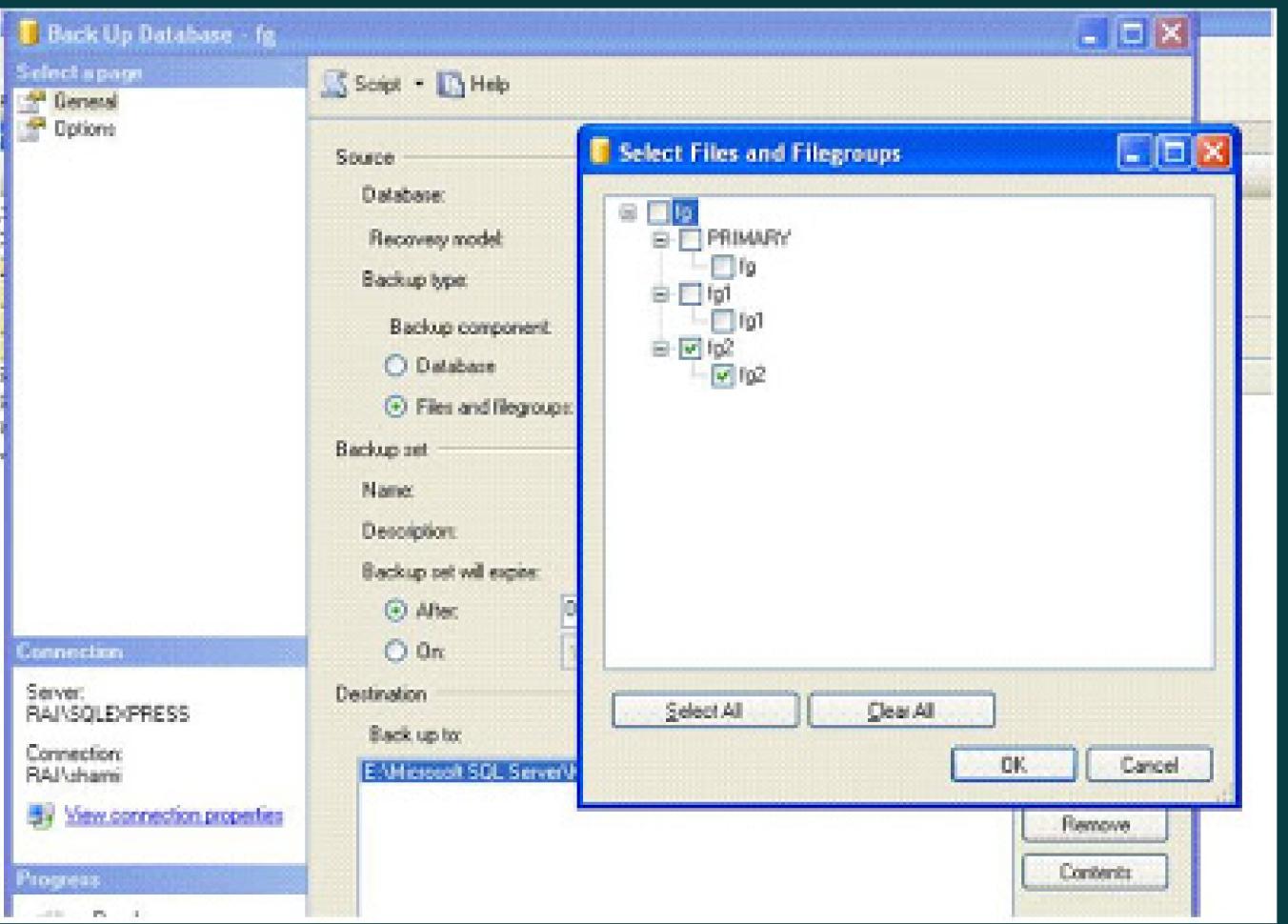
2) แยกไฟล์ MDF และ LDF ไว้กันคนละ physical disk

โดยคำสั่ง Insert, Update, Delete จะกระทำกับไฟล์ LDF เป็นหลัก ทั้งนี้ในงาน multi-user ขนาดใหญ่แล้วจะเกิด load read-write อยู่บน physical disk ก่อนเดียวกัน ถ้าเป็นไปได้ก็ควรจะแยกไฟล์ MDF และ LDF ไว้คนละ physical disk ก็จะทำให้เกิดการแบ่ง load read-write ไปไว้คนละ physical disk เพื่อการทำงานที่ดีขึ้น



3) แยก tempdb ไว้บน harddisk ก่อนใหม่

หลักการจะเน้นกับการแยกไฟล์ MDF และ LDF ครับ คือการแยก load แต่ tempdb จะเป็น temp สำหรับ database ทุกก้อนที่อยู่ใน Server tempdb จะรับงานหนักมาก ถ้าเป็นไปได้ก็ควรแยก tempdb ไว้คนละ physical disk ได้จะดีมาก



4) สำหรับ database ขนาดใหญ่ให้ใช้วิธีการแยกไฟล์ออกเป็นหลายไฟล์

โดยใช้ File Group หลักการจะเน้นอ่อนกับการทำ partition table แต่เป็นระดับ Database แทน โดยการแยก table กระจายไปแต่ละ physical disk แต่วิธีการแยกเรา จะใช้ file group เป็นเนื้องอกการแบ่งกลุ่มให้กับข้อมูล ซึ่งสามารถทำได้ทุกเวอร์ชันของ SQL Server

การปรับแต่ง SQL ให้เร็วขึ้น

ไม่จำเป็นต้องใช้ข้อมูลทุกอย่าง

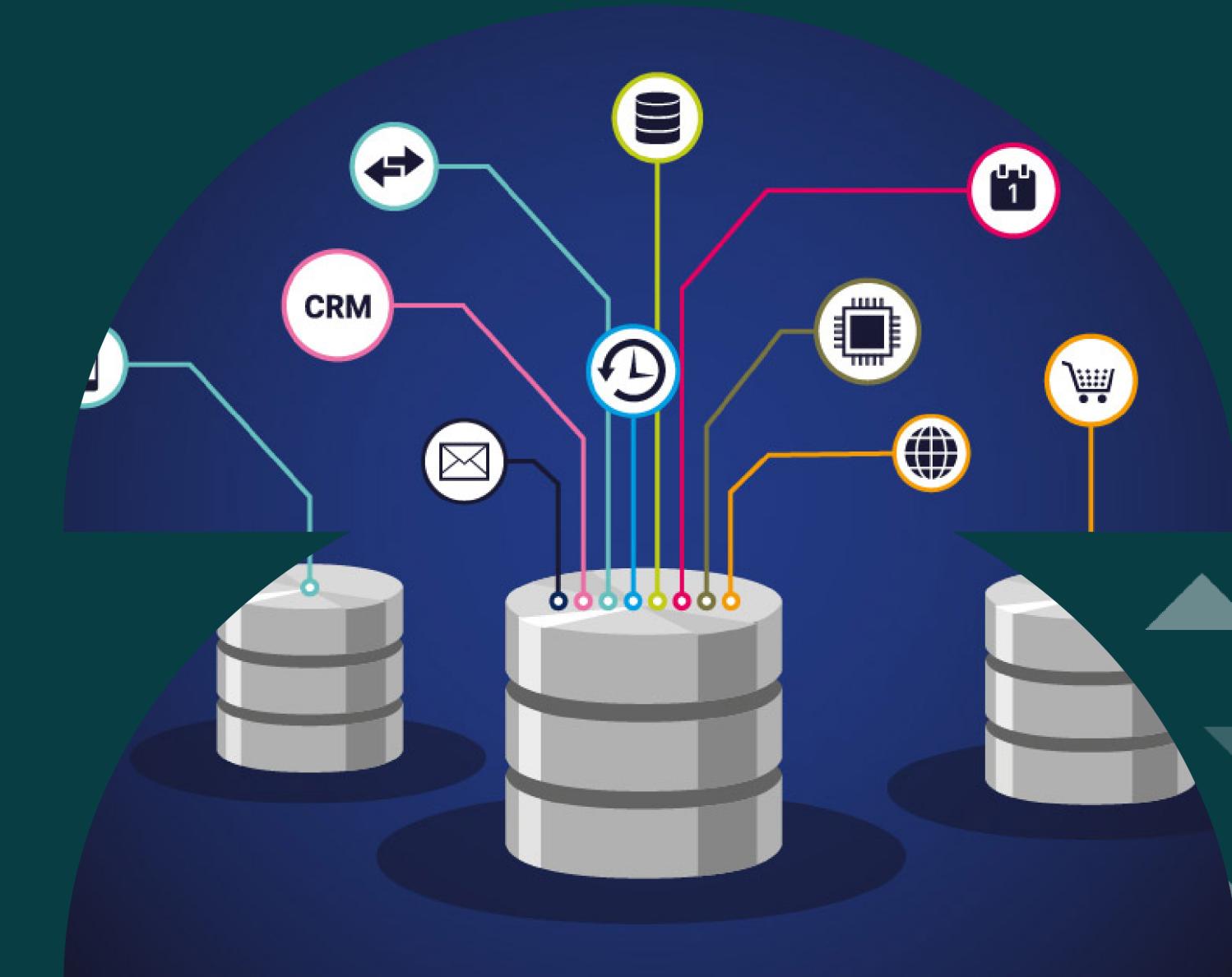
พิจารณาว่าจำเป็นต้องใช้ SELECT * หรือไม่? ควรจะดึงเฉพาะ Column ที่ต้องการเพื่อให้ได้ผลลัพธ์รวดเร็วขึ้น แทนที่จะใช้ SELECT * เพื่อดึงข้อมูลทั้งหมดอookma

ระวังการใช้ NOT IN

พยายามหลีกเลี่ยงการใช้ 'IN' หรือ 'NOT IN' เพราะการทำแบบนี้ หมายถึง กำลัง Scan กั้ง Table อยู่ เพราะ Query Engine จะໄລ่ດູໄປ ก່ຽວ Row เพื่อตรวจสอบว่า ตรงตามเงื่อนไขหรือไม่ ให้เปลี่ยนมาใช้ 'EXISTS' หรือ 'NOT EXISTS' ດູ ເພະນັນຈະສ່າງຜລກຮະບຕ່ວ່າ Query Plan ນ້ອຍກວ່າการใช้ 'NOT IN'

ใช้ WILDCARDS เก่าที่จำเป็น

การใช้ Wildcards ที่จຸດເຮີນຕົນແລະຈຸດສິ້ນສຸດຂອງ LIKE จะກຳໃຫ້ การ Query ช້າລັງ ແລະອາຈະກຳໄຟໄດ້ຜລັພຣັກທີ່ມາກເກີນກວ່າທີ່ຄຸນ ຕ້ອງການ ໄກ້ໃຊ້ Wildcards ເພາະເນື່ອຕ້ອງການໃຊ້ງານຈົງ ແລະ ເກົ່ານັ້ນ ໂດຍກ່ວ້າໄປຈະໃຊ້ກັນເພີ່ງແກ່ຈຸດເດີຍວເກົ່ານັ້ນ ດັ່ງນັ້ນ ຈຶ່ງຄວດ ຄຳນຶ່ງກົງສິ່ງທີ່ຕ້ອງການຈະໃກ້ Query Engine ກຳ





การปรับแต่ง SQL ให้เร็วขึ้น

ระวัง NESTED QUERIES

การ Run หลาย ๆ Queries ซ้อนกันไปเรื่อย ๆ เมื่อน้อยกว่า Function ก็เป็นสิ่งที่ไม่แนะนำให้ทำ และบันจะเร็วกว่าหากเขียนลงใน Table ลองพิจารณาการสร้าง Staging Tables ซึ่งนั่นหมายถึง เรากำลัง Join Subset ของข้อมูลที่เลือกลง ทำให้การ Query เร็วขึ้น

ระวังการใช้ VIEWS ซ้อน ๆ กัน

การเรียก Views ซ้อน Views ซ้อน Views ไปเรื่อย ๆ นั่นคือ เรากำลังบอกให้ Query Engine ทำการ Run หลาย ๆ Queries เพื่อ Return Row และ Column ที่ต้องการ ลองเปลี่ยนเป็นเขียนมันลงไปใน Table หากต้องการดูข้อมูลในแต่ละ วัน/สัปดาห์/เดือน แทนที่จะใช้ Views เพื่อ Filter ข้อมูล

การใช้ INDEXES

Indexes จะช่วยเพิ่มความเร็วให้ Queries ของเรา โดยการจัดลำดับข้อมูล เพื่อให้ Database Engine ทราบว่าจะหาข้อมูลที่ต้องการได้จากที่ไหน หรือมี Lookup Table เพื่อกำหนดว่าจะดำเนินการตาม哪部分 ของ Indexes ที่ใช้จะเป็นตัวกำหนดวิธีการทำงานของ Indexes และใช้ Indexes สำหรับ Column ที่จะใช้งานมันบ่อย ๆ ใน Queries



Microsoft SQL Server



ทำงานได้เป็นอย่างดีกับ Windows OS ทุกรุ่น

เมื่อพิจารณาว่ามีการจัดการและจัดการพัฒนาโดย Microsoft ผลิตภัณฑ์ทั้งหมดที่เปิดตัวควรทำงานได้ดีบน Windows OS รวมถึง Microsoft SQL Server เริ่มจาก Windows XP, Windows Vista, Windows 7, Windows 8 และแม้กระทั่ง Windows 10 สามารถติดตั้ง Microsoft SQL Server และสามารถทำงานได้ดี

สามารถทำการจัดกลุ่มข้อมูล

การจัดกลุ่มข้อมูลเป็นรูปแบบของการจัดกลุ่มข้อมูลเป็นหลายกลุ่ม ดังที่ได้กล่าวไว้ก่อนหน้านี้ Microsoft SQL Server มีข้อดีในเรื่องของการจัดกลุ่มข้อมูลเพื่อที่จะถูกใช้เป็นมูลค่าเพิ่มสำหรับโปรแกรมเมอร์หรือนักพัฒนา ด้วยการจัดกลุ่มข้อมูลนี้จะช่วยให้การจัดการฐานข้อมูลของคุณง่ายขึ้น

การควบคุมฐานข้อมูลส่วนกลาง

ใช้ Microsoft SQL Server แล้วการจัดการฐานข้อมูลจะดำเนินการจากส่วนกลาง สิ่งนี้จะช่วยลดการเกิดพลัดการสื่อสารหรือข้อผิดพลาดในการประมวลผลและการใช้ฐานข้อมูลเนื่องจากจะเกี่ยวข้องกับการใช้คอมพิวเตอร์เครื่องเดียวเป็นศูนย์ควบคุม



จุดเด่น MICROSOFT SQL SERVER



สามารถสำรองฐานข้อมูลได้

นอกจากนี้ใน Microsoft SQL Server สิ่งอำนวยความสะดวกในการสำรองฐานข้อมูลเป็นมาตรการป้องกันในกรณีที่มีการหยุดชะงักในระหว่างกระบวนการดำเนินการ นอกจากนี้ฟังก์ชันการสำรองฐานข้อมูลนี้ยังสามารถใช้งานได้มากกว่าหนึ่งครั้งหากในอนาคตสร้างระบบที่คล้ายกัน กระบวนการของการย้ายฐานข้อมูลไปยังอุปกรณ์อื่น ๆ ก็จะอำนวยความสะดวกอย่างมาก

มีคุณสมบัติการกู้คืนและกู้คืนข้อมูล

นอกจากนี้ยังมีคุณสมบัติการสำรองข้อมูลที่เปิดออก Microsoft SQL Server ยังมีการกู้คืนข้อมูลและเรียกคืนคุณสมบัติ สิ่งนี้มีประโยชน์เมื่อมีข้อมูลบางส่วนจากฐานข้อมูลของคุณที่เสียหายดังนั้นคุณสามารถกู้คืนและกู้คืนข้อมูลได้



MICROSOFT SQL SERVER



สามารถใช้ได้กับ Windows OS เท่านั้น

สำหรับผู้ที่คุณใช้งานของ LINUX หรือ MAC OS และต้องลงทะเบียนจาก การใช้ Microsoft SQL Server เพราะจนถึงตอนนี้สามารถใช้งานได้กับ ระบบปฏิบัติการ Windows เท่านั้นเท่านั้นถึงแม้ว่าในทุกรุ่น สำหรับข้อ บกพร่องนี้อาจสันนิษฐานได้ว่า Microsoft อนุญาตให้ใช้ Microsoft SQL Server เพื่อ จำกัด เวลา Windows เท่านั้นเนื่องจากยังอยู่ในขอบเขต ของ บริษัท

ภาษาโปรแกรม จำกัด

Microsoft SQL Server เป็นที่รู้จักกันดีอยู่แล้ว เพราะ มันเข้ากันได้กับภาษาการเขียนโปรแกรมสุดยอดที่คุณเคยกับการใช้ ภาษาอื่นนอกเหนือจาก Net คุณจะต้องเตรียมพร้อม ก่อนที่จะเสียงต่อความยุ่งยากในการใช้ เพราะถูกกำหนดว่า Microsoft ได้ออกแบบซอฟต์แวร์นี้เพื่อปุ่มเน้นการใช้ ภาษา Net เท่านั้น

ราคาซื้อแพง

ซอฟต์แวร์เกือบทั้งหมดจาก Windows แบ่งเป็นราค าในราคาก่อต่องานสูงรวมถึง Microsoft SQL Server นี้ แต่แน่นอนว่ายังรองรับความสามารถที่สำคัญและ คุณสมบัติที่บลส์บนใจซึ่งมีราคามหาศาล

ไม่เหมาะสมสำหรับฐานข้อมูล ขนาดใหญ่

แม้ว่าดำเนินการโดยใช้ชื่อของ บริษัท ขนาดใหญ่ เพื่อร่วมชั้นของ Microsoft แต่ก็จริงแล้ว Microsoft SQL Server ไม่สามารถรองรับการใช้ฐานข้อมูลขนาด ใหญ่ได้ แท้จริงแล้วจนถึงขณะนี้มีเพียง Oracle Database เท่านั้นที่สามารถสนับสนุนการใช้งานขนาด ใหญ่และไม่มีใครสามารถแข่งขันกับมันได้

QUERY PLAN

ประเภทการแสดงผล QUERY EXECUTION PLAN

การแสดงผล QUERY EXECUTION PLAN นั้นเป็นการแสดงผล PLAN ที่ถูกเลือกจากขั้นตอน QUERY OPTIMIZATION เสร็จไปแล้ว ขอเรียกว่า COMPILED QUERY EXECUTION PLAN หรือ COMPILED PLAN (รายละเอียดอยู่บញความ “เตรียมพร้อมก่อนอ่าน EXECUTION PLAN” สนุกมาก) ซึ่งการแสดงผลแบ่งออกได้ 2 ประเภทดังนี้

1. ESTIMATED EXECUTION PLAN คือการแสดงผล QUERY EXECUTION PLAN ที่ถูก COMPILED แล้วแต่ยังไม่ถูก EXECUTE
2. ACTUAL EXECUTION PLAN คือการแสดงผลทั้งข้อมูลที่เป็นค่าประเมิน (ESTIMATED) และข้อมูลที่ได้จากการ EXECUTE จริง (ACTUAL)
ผู้อ่านที่ยังไม่ได้อ่านบทความ “เตรียมพร้อมก่อนอ่าน EXECUTION PLAN” ควรกับไปอ่านบทความก่อนเพื่อทำความเข้าใจกลไก QUERY OPTIMIZATION และ CARDINALITY ESTIMATION

การแสดงผล ESTIMATED EXECUTION PLAN

สามารถแสดงผล ESTIMATED EXECUTION PLAN ได้ 2 แบบคือ

1. แสดงผลเป็นเอกสาร XML ผ่านคำสั่ง SET SHOWPLAN_XML โดยมี SYNTAX ดังนี้

```
SET SHOWPLAN_XML { ON | OFF }
```

เมื่อเราป้อนคำสั่ง T-SQL ต่อจาก การกำหนด SET SHOWPLAN_XML ให้มีค่าเป็น

ON และทำการ Execute

จะพบว่า Microsoft SQL Server จะแสดง Compiled Plan ออกมาในรูปแบบเอกสาร XML โดยไม่รวมผล Compiled Plan ดังกล่าว

```

SET SHOWPLAN_XML ON
GO

SELECT
    E.BusinessEntityID
    , P.Title
    , P.FirstName
    , P.MiddleName
    , P.LastName
    , P.Suffix
    , E.JobTitle
    , D.Name AS Department
    , D.GroupName
    , EH.StartDate
FROM HumanResources.Employee as E
INNER JOIN Person.Person as P
ON P.BusinessEntityID = E.BusinessEntityID
INNER JOIN HumanResources.EmployeeDepartmentHistory as EH
ON E.BusinessEntityID = EH.BusinessEntityID
INNER JOIN HumanResources.Department as D
ON EH.DepartmentID = D.DepartmentID
WHERE EH.EndDate IS NULL;

GO

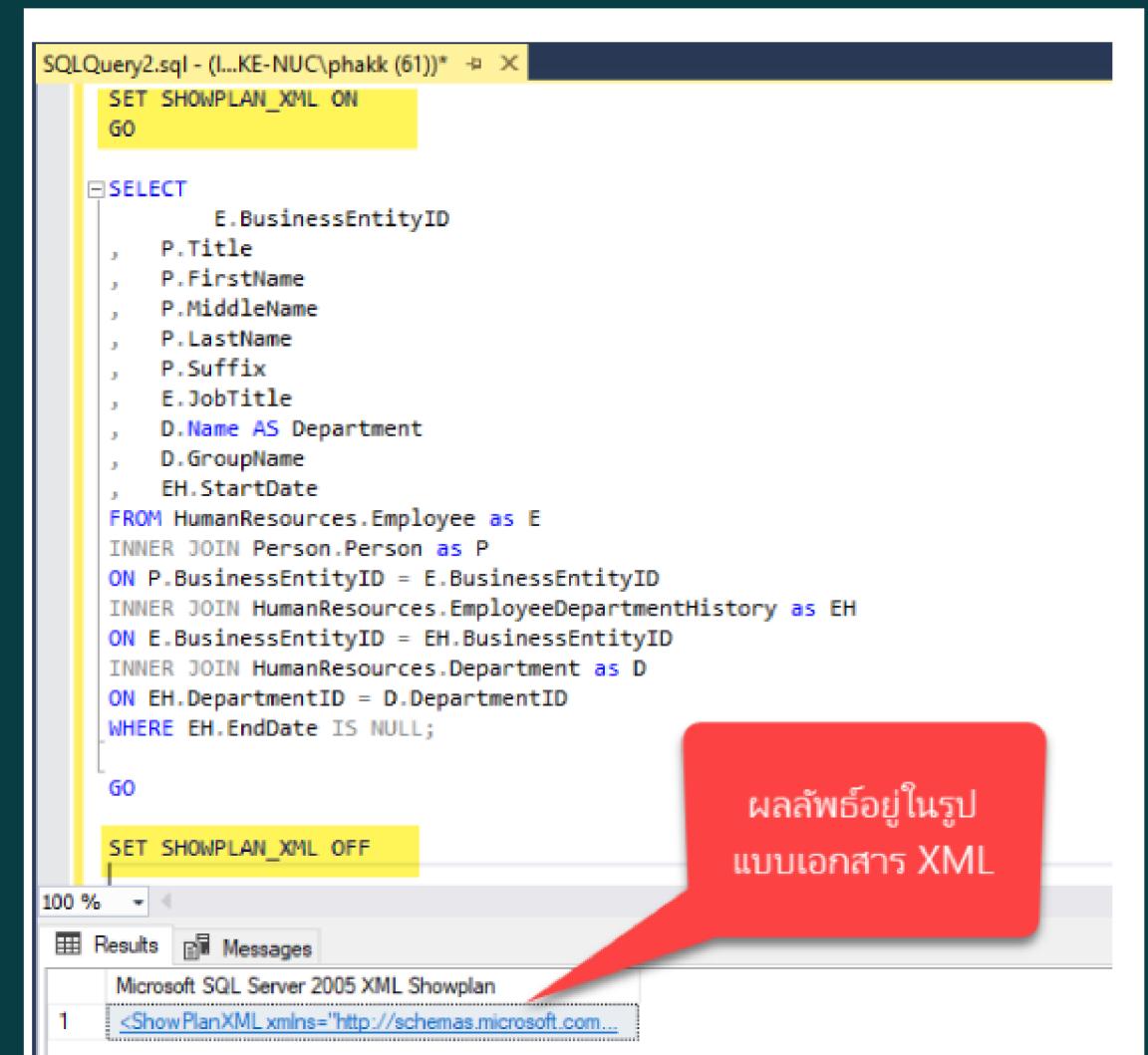
SET SHOWPLAN_XML OFF

```

แนะนำว่าเมื่อ SET SHOWPLAN_XML ให้มีค่าเป็น ON แล้ว ควร SET SHOWPLAN_XML ให้มีค่าเป็น OFF ทันทีหลังใช้งานเสร็จ

ทดสอบด้วยสคริปต์ต่อไปนี้กับฐานข้อมูล ADVENTUREWORKS ผ่าน SQL SERVER MANAGEMENT STUDIO (SSMS)

จะสังเกตเห็นว่าผู้เขียนต้องใช้ประโยค GO เพื่อให้คำสั่ง SET SHOWPLAN_XML อยู่คบลະ SCOPE (BATCH) กับคำสั่งอื่น ๆ เพราะเป็นข้อจำกัดของคำสั่งนี้



The screenshot shows a SQL Server Management Studio window titled "SQLQuery2.sql - (...KE-NUC\phakk (61))". The query is:

```

SET SHOWPLAN_XML ON
GO

SELECT
    E.BusinessEntityID
    , P.Title
    , P.FirstName
    , P.MiddleName
    , P.LastName
    , P.Suffix
    , E.JobTitle
    , D.Name AS Department
    , D.GroupName
    , EH.StartDate
FROM HumanResources.Employee as E
INNER JOIN Person.Person as P
ON P.BusinessEntityID = E.BusinessEntityID
INNER JOIN HumanResources.EmployeeDepartmentHistory as EH
ON E.BusinessEntityID = EH.BusinessEntityID
INNER JOIN HumanResources.Department as D
ON EH.DepartmentID = D.DepartmentID
WHERE EH.EndDate IS NULL;

GO

SET SHOWPLAN_XML OFF

```

The results pane at the bottom shows the output of the XML plan:

```

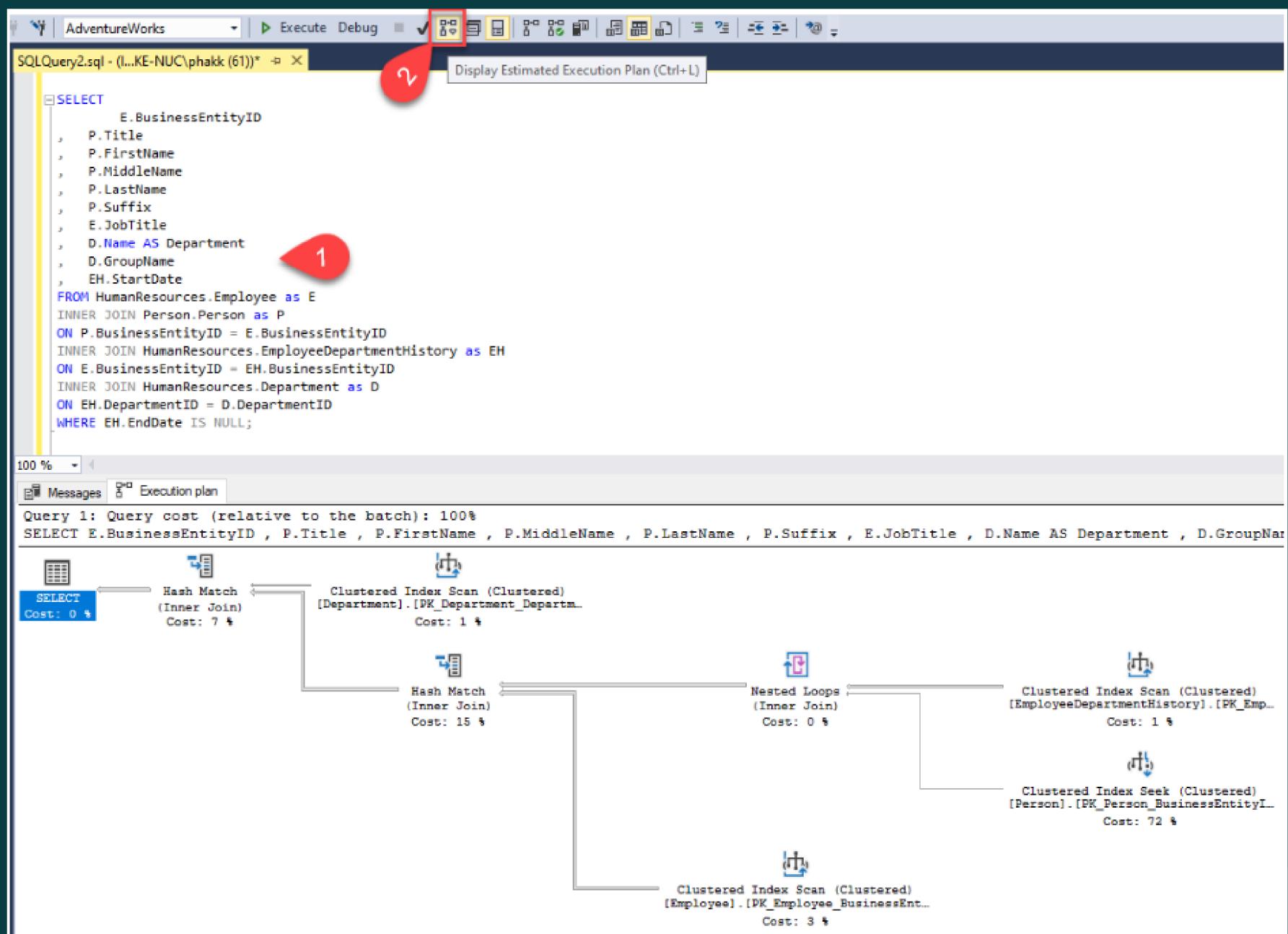
<ShowPlanXML xmlns='http://schemas.microsoft.com...>

```

A red callout bubble points to the XML output with the text: "ผลลัพธ์อยู่ในรูปแบบเอกสาร XML".

ข้อดีของการแสดงผลเป็นเอกสาร XML คือ เมื่อเราใช้ CLIENT TOOL อื่นๆ ที่ไม่ใช่ SSMS ก็สามารถ เก็บ PLAN ไปเปิดด้วย SSMS ในภายหลังได้

2. แสดงผลเป็น GRAPHIC ทันที ผ่าน SQL SERVER MANAGEMENT STUDIO (SSMS)
โดยพิมพ์ QUERY ก็ต้องการแสดง COMPILED PLAN เข้าไปจากนั้นกด CTRL-L หรือ
กดไปที่ปุ่ม DISPLAY ESTIMATED EXECUTION PLAN จะได้ผลลัพธ์เป็น COMPILED PLAN ออกมากในแท็บ
EXECUTION PLAN ดังแสดง



การแสดงผล ACTUAL EXECUTION PLAN

สามารถแสดงผล ACTUAL EXECUTION PLAN ได้ 2 แบบคือ
1. แสดงผลเป็นเอกสาร XML ผ่านคำสั่ง SET STATISTICS XML โดยมี SYNTAX ดังนี้

SET STATISTICS XML { ON | OFF }

เมื่อเราป้อนคำสั่ง T-SQL ต่อจาก การกำหนด SET STATISTICS XML ให้มีค่าเป็น ON และทำการ EXECUTE จะพบว่า MICROSOFT SQL SERVER ทำการประมวล COMPILED PLAN จนได้ RESULTSET ออกมา หลังจากนั้นจึงแสดงผล COMPLIED PLAN พร้อมข้อมูลที่เกิด ตอนประมวลผลจริงในรูปแบบเอกสาร XML

ทดสอบด้วยสคริปต์ต่อไปนี้กับฐานข้อมูล ADVENTUREWORKS ผ่าน SQL SERVER MANAGEMENT STUDIO (SSMS)

```
SET STATISTICS XML ON
GO

SELECT
    E.BusinessEntityID
    , P.Title
    , P.FirstName
    , P.MiddleName
    , P.LastName
    , P.Suffix
    , E.JobTitle
    , D.Name AS Department
    , D.GroupName
    , EH.StartDate
FROM HumanResources.Employee AS E
INNER JOIN Person.Person AS P
ON P.BusinessEntityID = E.BusinessEntityID
INNER JOIN HumanResources.EmployeeDepartmentHistory
AS EH
ON E.BusinessEntityID = EH.BusinessEntityID
INNER JOIN HumanResources.Department AS D
ON EH.DepartmentID = D.DepartmentID
WHERE EH.EndDate IS NULL;

GO

SET STATISTICS XML OFF
```

จะสังเกตเห็นว่าผู้เขียนต้องใช้ประโยค GO เพื่อให้คำสั่ง SET STATISTICS XML อยู่คุณละ SCOPE (BATCH)
กับคำสั่งอื่น ๆ เพราะเป็นข้อจำกัดของคำสั่งนี้
ผลลัพธ์ที่ได้พบว่าคิวรี่จะถูก EXECUTE และประมวลผลเอกสาร XML ในแท็บ RESULTS มาด้วย หลังจาก
EXECUTE จนได้ RESULTSET ดังแสดง

The screenshot shows a SQL query window titled "SQLQuery2.sql - (...KE-NUC\phakk (61))". The query uses the HumanResources.Employee table and joins it with Person.Person, EmployeeDepartmentHistory, and Department tables. It filters for employees whose department history ended (END_DATE IS NULL). The query is wrapped in a batch with "GO" statements and includes "SET STATISTICS XML ON" and "SET STATISTICS XML OFF" commands. The results tab displays a grid of employee data, and the messages tab shows the XML execution plan. A red callout box points to the XML plan area with the text "ผลลัพธ์อยู่ในรูปแบบเอกสาร XML".

BusinessEntityID	Title	FirstName	MiddleName	LastName	Suffix	JobTitle	Department	GroupName	StartDate	
1	NULL	Ken	J	Sánchez	NULL	Chief Executive Officer	Executive	Executive General and Administration	2009-01-14	
2	2	NULL	Temi	Duffy	NULL	Vice President of Engineering	Engineering	Research and Development	2008-01-31	
3	3	NULL	Roberto	NULL	Tamburello	NULL	Engineering Manager	Engineering	Research and Development	2007-11-11
4	4	NULL	Rob	NULL	Walters	NULL	Senior Tool Designer	Tool Design	Research and Development	2010-05-31
5	5	Ms.	Gail	A	Erickson	NULL	Design Engineer	Engineering	Research and Development	2008-01-06
6	6	Mr.	Jossef	H	Goldberg	NULL	Design Engineer	Engineering	Research and Development	2008-01-24
7	7	NULL	Dylan	A	Miller	NULL	Research and Development Manager	Research and Development	Research and Development	2009-02-08
8	8	NULL	Diane	L	Margheim	NULL	Research and Development Engineer	Research and Development	Research and Development	2008-12-29
9	9	NULL	Gigi	N	Matthew	NULL	Research and Development Engineer	Research and Development	Research and Development	2009-01-16
10	10	NULL	Michael	NULL	Raheem	NULL	Research and Development Manager	Research and Development	Research and Development	2009-05-03
11	11	NULL	Ovidiu	V	Craciun	NULL	Senior Tool Designer	Tool Design	Research and Development	2010-12-05
12	12	NULL	Thierry	B	D'Hers	NULL	Tool Designer	Tool Design	Research and Development	2007-12-11
13	13	Ms.	Janice	M	Galvin	NULL	Tool Designer	Tool Design	Research and Development	2010-12-23
14	14	NULL	Michael	I	Sullivan	NULL	Senior Design Engineer	Engineering	Research and Development	2010-12-30
15	15	NULL	Sharon	B	Salavarria	NULL	Design Engineer	Engineering	Research and Development	2011-01-18
16	16	NULL	David	M	Bradley	NULL	Marketing Manager	Marketing	Sales and Marketing	2009-07-15
17	17	NULL	Kevin	F	Brown	NULL	Marketing Assistant	Marketing	Sales and Marketing	2007-01-26
18	18	NULL	John	L	Wood	NULL	Marketing Specialist	Marketing	Sales and Marketing	2011-02-07
19	19	NULL	Mary	A	Dempsey			Marketing	Sales and Marketing	2011-02-14
20	20	NULL	Wanda	M	Benshoof			Marketing	Sales and Marketing	2011-01-07
21	21	NULL	Temy	J	Eminizer			Marketing	Sales and Marketing	2009-03-02
22	22	NULL	Sanya	E	Hampadoungae			Marketing	Sales and Marketing	2008-12-12
23	23	NULL	May	E	Gibson			Marketing	Sales and Marketing	2009-01-12
24	24	Ms.	Jill	A	Williams			Marketing	Sales and Marketing	2009-01-18

Microsoft SQL Server 2005 XML Showplan
1 cShowPlanXML xmlns="http://schemas.microsoft.com..."

ผลลัพธ์อยู่ในรูป
แบบเอกสาร XML

2.แสดงผลเป็น GRAPHIC หลังจาก EXECUTE คำสั่งผ่าน SQL SERVER MANAGEMENT STUDIO (SSMS)
โดยจะต้องทำการ ENABLE โดยกด CTRL-M หรือกดปุ่ม INCLUDE ACTUAL EXECUTION PLAN ก่อนจะ EXECUTE ดังแสดง

The screenshot shows the SQL Server Management Studio interface. In the top bar, there is a red circle with the number '2' over the 'Include Actual Execution Plan (Ctrl+M)' button. In the main window, there is a red circle with the number '1' over the 'Execution plan' tab in the results pane. A green callout bubble points from the text below to this tab. The results pane displays a table of employee data from the AdventureWorks database.

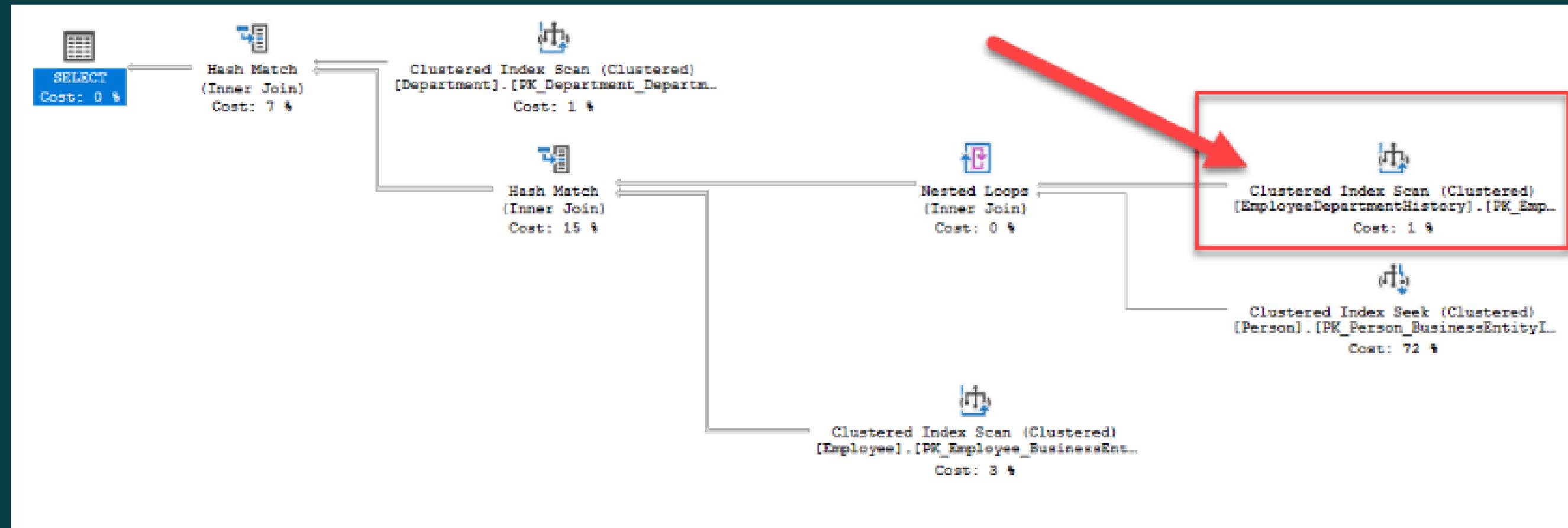
BusinessEntityID	Title	FirstName	MiddleName	LastName	Suffix	JobTitle	Department	GroupName	StartDate
1	NU...	Ken	J	Abrahaz	NU...	Chief Executive Officer	Executive	Executive General and Administration	2009-01-...
2	NU...	Teri	Lee	Engineering	Research and Development	2008-01-...
3	NU...	Roberto	NULL	Tamburini		...	Engineering	Research and Development	2007-11-...
4	NU...	Rob	NULL	Walter		...	Tool Design	Research and Development	2010-05-...
5	Ms.	Gail	A	Erickson		...	Engineering	Research and Development	2008-01-...
6	Mr.	Jossef	H	Goldberg		...	Engineering	Research and Development	2008-01-...
7	NU...	Dylan	A	Miller		...	Research and Developm...	Research and Development	2009-02-...
8	NU...	Diane	L	Margheim		...	Research and Developm...	Research and Development	2008-12-...
9	NU...	Gigi	N	Matthew	NU...	Research and Development Engin...	Research and Developm...	Research and Development	2009-01-...
10	NU...	Michael	NULL	Raheem	NU...	Research and Development Mana...	Research and Developm...	Research and Development	2009-05-...
11	NU...	Ovidiu	V	Cracum	NU...	Senior Tool Designer	Tool Design	Research and Development	2010-12-...
12	NU...	Thierry	B	D'Hers	NU...	Tool Designer	Tool Design	Research and Development	2007-12-...
13	Ms.	Janice	M	Galvin	NU...	Tool Designer	Tool Design	Research and Development	2010-12-...
14	NU...	Michael	I	Sullivan	NU...	Senior Design Engineer	Engineering	Research and Development	2010-12-...

จะปรากฏ Tab นี้
หลังปรามผลเสร็จ

พบว่าหลังจาก EXECUTE จะได้ RESULTSET เรียบร้อยแล้วจะปรากฏแท็บ EXECUTION PLAN ตามมา

เปรียบเทียบข้อมูลของการแสดงผล QUERY EXECUTION PLAN แต่ละแบบ

เมื่อนำมาสู่การประยุกต์ใช้ เนื้อหาจะได้รับความเข้าใจมากขึ้น



จะมีความแตกต่างกันของข้อมูลอยู่บ้าง

ESTIMATED EXECUTION PLAN และ ACTUAL EXECUTION PLAN

ในส่วนของ ESTIMATED EXECUTION PLAN นั้นแสดง COMPILED PLAN ที่ยังไม่ได้ EXECUTE ค่าต่างๆ ที่ได้มาเกิดจากการคำนวนของคอมโพเนนต์ในกลุ่ม QUERY OPTIMIZATION ชื่อ CARDINALITY ESTIMATOR โดยใช้ข้อมูลหลักจาก STATISTICS (ในรูปแบบของ HISTOGRAM) ที่สัมพันธ์กับแต่ละ PREDICATE (ในที่นี้คือเงื่อนไขหลังประโยค WHERE) เพื่อประเมินจำนวนแถวข้อมูล (ESTIMATED NUMBER OF ROWS) จากนั้นจึงนำไปประเมิน COST ต่างๆ ต่อไป (COMPILED PLAN ที่แสดงใน ESTIMATED EXECUTION PLAN ถูกเลือกมาแล้ว จาก หลายๆ PLAN แบ่งขั้นกันด้วยค่า COST นี้เอง)

แต่ในส่วนของ ACTUAL EXECUTION PLAN นั้นเป็นการนำ COMPILED PLAN ไปประมวลผล ผู้อ่านจะพบจำนวนแถวข้อมูลที่ได้จากการประมวลผลจริง (ACTUAL NUMBER OF ROWS) เพิ่มจากค่าประเมิน หากจำนวนแถวข้อมูลจากการประมวลผล ต่างจากจำนวนแถวข้อมูลที่ได้จากการประเมินมาก เป็นเพราะ STATISTICS (HISTOGRAM) เริ่มไม่สะท้อนค่าปัจจุบัน อาจจำเป็นต้องปรับปรุง STATISTICS แบบ MANUAL (ปกติแล้ว STATISTICS ถูกตั้งค่าให้ปรับปรุงอัตโนมัติ)

Estimated Execution Plan

Clustered Index Scan (Clustered)	
Scanning a clustered index, entirely or only a range.	
Physical Operation	Clustered Index Scan
Logical Operation	Clustered Index Scan
Estimated Execution Mode	Row
Storage	RowStore
Estimated I/O Cost	0.0038657
Estimated Operator Cost	0.0043483 (1%)
Estimated CPU Cost	0.0004826
Estimated Subtree Cost	0.0043483
Estimated Number of Executions	1
Estimated Number of Rows	290
Estimated Number of Rows to be Read	296
Estimated Row Size	19 B
Ordered	False
Node ID	5
Predicate	
[AdventureWorks].[HumanResources]. [EmployeeDepartmentHistory].[EndDate] as [EH].[EndDate] IS NULL	
Object	
[AdventureWorks].[HumanResources]. [EmployeeDepartmentHistory]. [PK_EmployeeDepartmentHistory_BusinessEntityID_StartDate_Dep artmentID] [EH]	
Output List	
[AdventureWorks].[HumanResources]. [EmployeeDepartmentHistory].BusinessEntityID, [AdventureWorks].[HumanResources]. [EmployeeDepartmentHistory].DepartmentID, [AdventureWorks]. [HumanResources].[EmployeeDepartmentHistory].StartDate	

Actual Execution Plan

Clustered Index Scan (Clustered)	
Scanning a clustered index, entirely or only a range.	
Physical Operation	Clustered Index Scan
Logical Operation	Clustered Index Scan
Actual Execution Mode	Row
Estimated Execution Mode	Row
Storage	RowStore
Number of Rows Read	296
Actual Number of Rows	290
Actual Number of Batches	0
Estimated I/O Cost	0.0038657
Estimated Operator Cost	0.0043483 (1%)
Estimated CPU Cost	0.0004826
Estimated Subtree Cost	0.0043483
Number of Executions	1
Estimated Number of Executions	1
Estimated Number of Rows	290
Estimated Number of Rows to be Read	296
Estimated Row Size	19 B
Actual Rebinds	0
Actual Rewinds	0
Ordered	False
Node ID	5
Predicate	
[AdventureWorks].[HumanResources]. [EmployeeDepartmentHistory].[EndDate] as [EH].[EndDate] IS NULL	
Object	
[AdventureWorks].[HumanResources]. [EmployeeDepartmentHistory]. [PK_EmployeeDepartmentHistory_BusinessEntityID_StartDate_Dep artmentID] [EH]	
Output List	
[AdventureWorks].[HumanResources]. [EmployeeDepartmentHistory].BusinessEntityID, [AdventureWorks].[HumanResources]. [EmployeeDepartmentHistory].DepartmentID, [AdventureWorks]. [HumanResources].[EmployeeDepartmentHistory].StartDate	

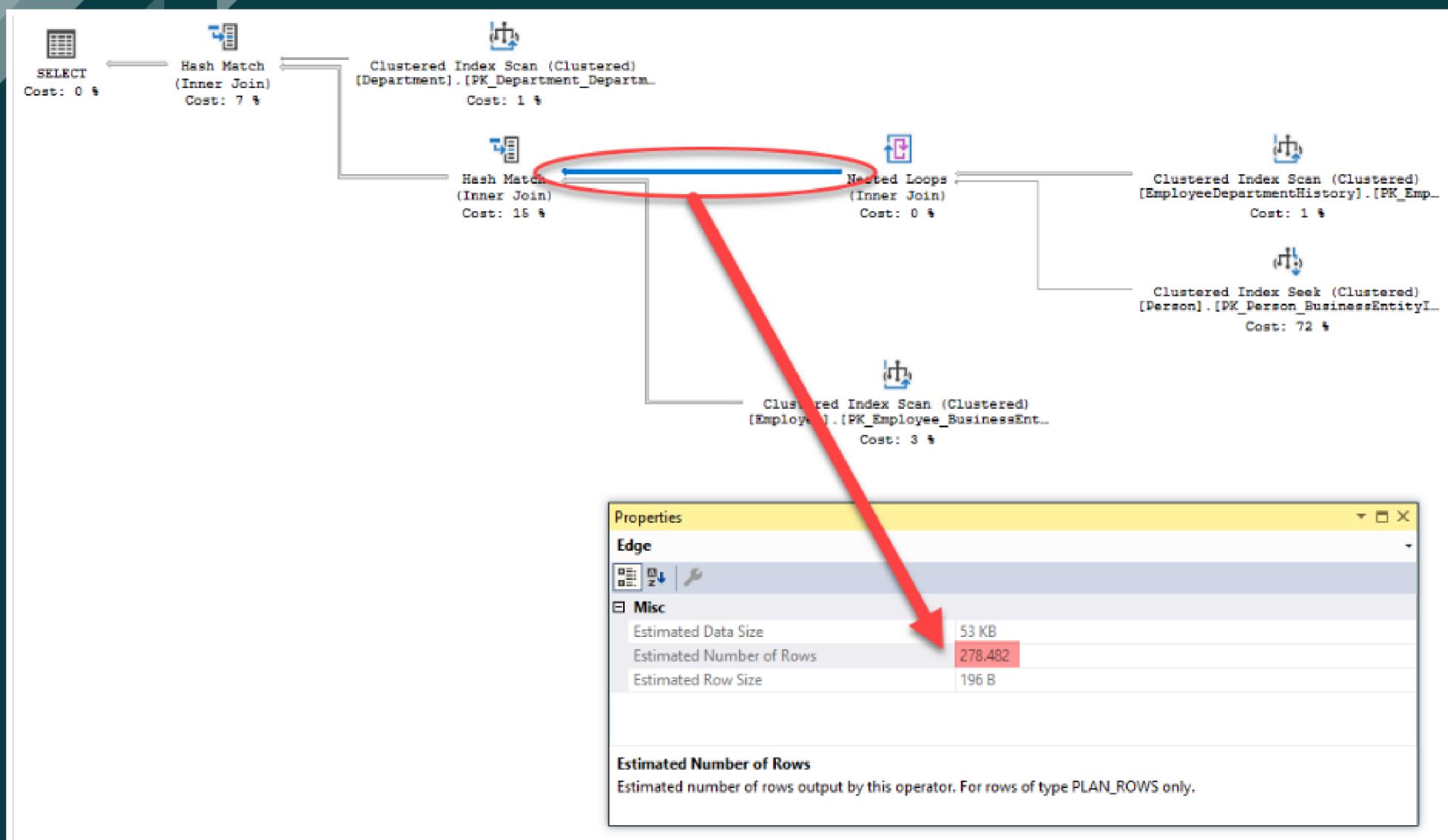
นอกเหนือจากการเอาเม้าส์ไปลอยอยู่เหนือตัวคำเบินการเพื่อแสดงข้อมูลแล้ว ยังสามารถดูรายละเอียดที่มากขึ้น

โดยการคลิกไปที่ตัวดำเนินการใดๆ และกด F4 หรือคลิกขวาเลือก PROPERTIES เพื่อแสดง PROPERTIES ของตัวดำเนินการนั้นดังแสดง

The image shows two side-by-side windows from SQL Server Management Studio. Both windows are titled 'Properties' and have a tab bar at the top with 'Estimated Execution Plan' and 'Actual Execution Plan'. The left window is labeled 'Estimated Execution Plan' and the right window is labeled 'Actual Execution Plan'. Both windows show a tree view of execution plan components. In the 'Actual Execution Plan' window, several properties are highlighted with red or green backgrounds. These include 'Defined Values' (highlighted in red), 'Object' (highlighted in red), 'Output List' (highlighted in red), 'TableCardinality' (highlighted in red), and 'Number of Rows Read' (highlighted in green). Other properties like 'Actual I/O Statistics' and 'Actual Time Statistics' are also visible.

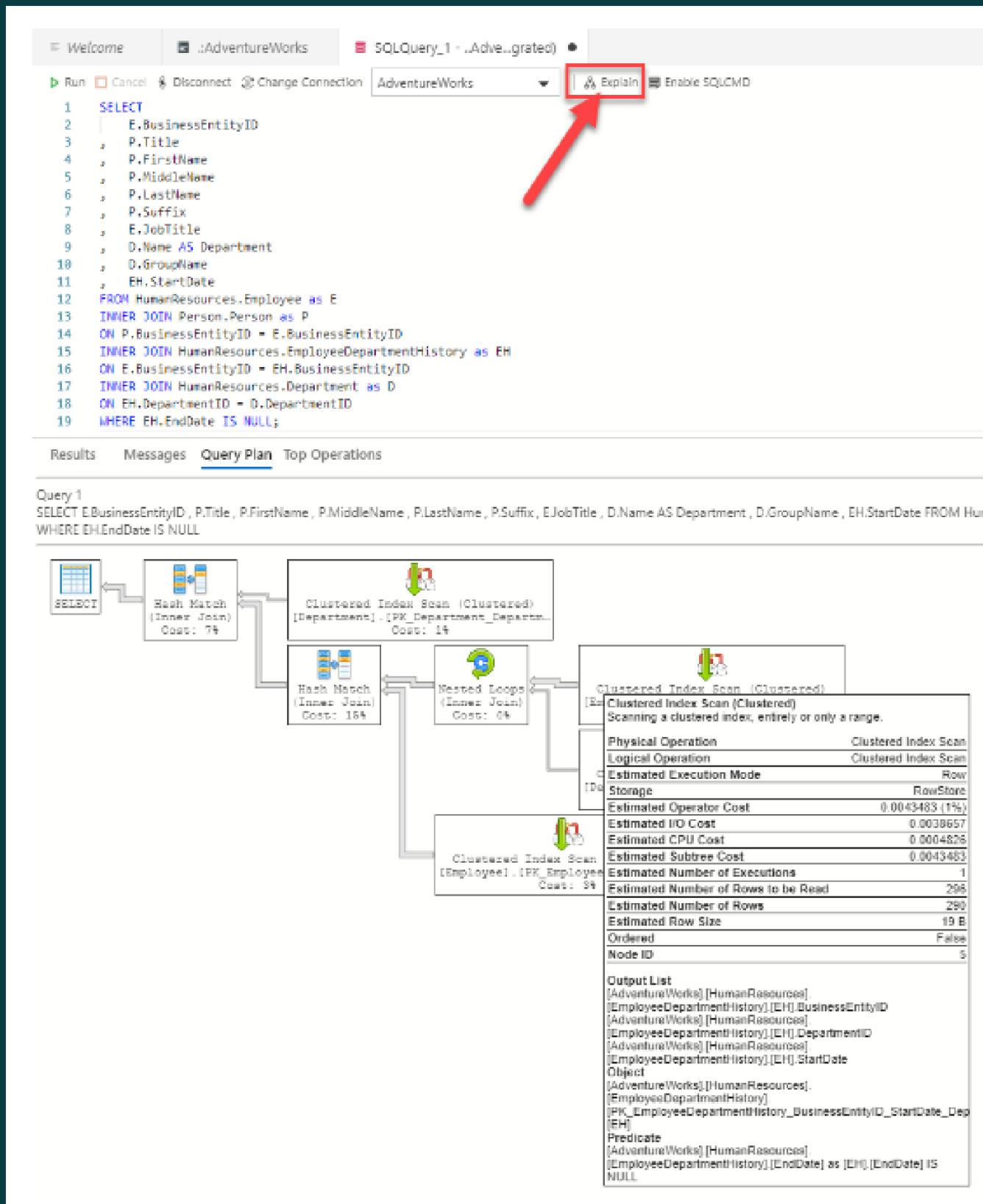
พบว่าการเรียก PROPERTIES ขึ้นมาแสดงสามารถให้รายละเอียดที่เพิ่มขึ้น โดยเฉพาะในส่วนของ ACTUAL EXECUTION PLAN จะแสดง I/O STATISTICS และ TIME STATISTICS ที่ใช้ในการประเมินผล วัดกึ่งเห็นการแตกเป็น THREADS ย่อยในการใช้ I/O หรือ CPU อีกด้วย

นอกเหนือจะสามารถแสดงค่าข้อมูลของแต่ละตัวดำเนินการภายใน COMPILED PLAN แล้ว เส้นเชื่อม (EDGE) ระหว่างตัวดำเนินการยังแสดงถึงข้อมูลที่ให้หลังจากตัวดำเนินการหนึ่งไปยังอีกตัวดำเนินการหนึ่ง ดังแสดง



นี่เป็นการแสดง ESTIMATED EXECUTION PLAN หากนำเม้าส์ไปลอยอยู่เหนือเส้นเชื่อม หรือเรียก PROPERTIES ของเส้นเชื่อมอ กมา จะพบกับค่าจำนวนแควข้อมูลที่ได้จากการประเมินมีค่าเป็น 278.482 เป็นเครื่องยืนยันว่าเป็นการคำนวนเชิงสถิติมา เพราะจำนวนแควข้อมูลจริงๆ แล้วต้องเป็นจำนวนเต็ม

นอกจาก MICROSOFT SQL SERVER MANAGEMENT STUDIO และ
อีกหนึ่งเครื่องมือจาก MICROSOFT ชื่อ AZURE DATA STUDIO ที่สามารถแสดง COMPILED PLAN
ออกมากได้เช่นกัน
โดยกดไปที่ปุ่ม EXPLAIN ข้อมูลของ ESTIMATED EXECUTION PLAN จะปรากฏออกมาดังแสดง



แต่เครื่องมือที่ผู้เขียนชอบมากเป็นพิเศษคือ PLAN EXPLORER จาก SENTRYONE



**THANK
YOU**