

บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

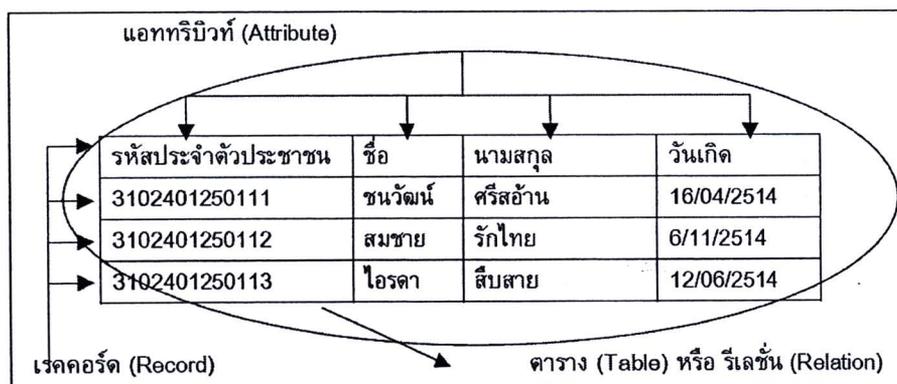
รายละเอียดในบทที่ 2 นี้จะกล่าวถึงทฤษฎีต่าง ๆ ที่เกี่ยวข้องและอ้างอิงถึงในงานวิจัย โดยแบ่งออกเป็น 8 ส่วน โดยส่วนแรกกล่าวถึงทฤษฎีฐานข้อมูลรีเลชันนอล (Relational Database System) ส่วนที่สองแนะนำทฤษฎีฐานข้อมูลแบบแอททริบิวต์ ส่วนที่สามกล่าวถึงทริกเกอร์ ส่วนที่สี่นำเสนอข้อมูลตัวเลข (Numeric Data) ส่วนที่ห้านำเสนอทฤษฎีเอเจนต์ ส่วนที่หกกล่าวถึงภาษาจาวา (Java Language) ส่วนที่เจ็ดกล่าวถึงฐานข้อมูลมายเอสคิวแอล (MySQL 5.0) ส่วนสุดท้ายคือภาษาเอสคิวแอล (SQL)

1. ทฤษฎีฐานข้อมูลรีเลชันนอล

ระบบฐานข้อมูลรีเลชันนอล เป็นฐานข้อมูลที่ใช้โมเดลเชิงสัมพันธ์ ด้วยแนวคิดแบบจำลองนี้มีการเก็บข้อมูลเป็นตารางเพื่อให้ง่ายต่อการเข้าใจและประยุกต์ใช้งาน ในแต่ละตารางแบ่งออกเป็นแถว ๆ และในแต่ละแถวจะแบ่งเป็นคอลัมน์ (Column) เนื่องจากแบบจำลองนี้เกิดจากทฤษฎีทางคณิตศาสตร์ (Kim, 1979) ทำให้มีคำศัพท์เฉพาะแตกต่างออกไปดังตารางที่ 1 และแสดงตัวอย่างรายละเอียดส่วนประกอบต่าง ๆ ของโมเดลแบบความสัมพันธ์ได้ดังภาพที่ 2

ตารางที่ 1 คำศัพท์ที่ใช้ในงานในฐานข้อมูลรีเลชันนอล

ศัพท์เฉพาะ	ศัพท์ทั่วไป
ความสัมพันธ์ (Relation)	ตาราง (Table)
ทูเปิล (Tuple)	แถว (Row) หรือ เรคคอร์ด (Record) หรือ ระเบียบ
แอททริบิวต์ (Attribute)	คอลัมน์ (Column) หรือ ฟیلด์ (Field)
คาร์ดินัลลิตี (Cardinality)	จำนวนแถว (Number of rows)
ดีกรี (Degree)	จำนวนแอททริบิวต์ (Number of attribute)
คีย์หลัก (Primary key)	ค่าเอกลักษณ์ (Unique identifier)
โดเมน (Domain)	ขอบข่ายของค่าของข้อมูล (Pool of legal values)



ภาพที่ 2 โมเดลแบบรีเลชันนอล

ด้วยเหตุที่ว่าฐานข้อมูลรีเลชันนอล สามารถอธิบายความสัมพันธ์ที่ซับซ้อนได้ดีมากกว่าโมเดลอื่น ๆ เพราะผู้ใช้สามารถกำหนดความสัมพันธ์ของตารางในฐานข้อมูลได้ ผู้ใช้สามารถค้นหา เปลี่ยนแปลงข้อมูล เพิ่มเติม หรือลบ ข้อมูลได้ โดยไม่ทำลายความสัมพันธ์ระหว่างตาราง เนื่องจากความสัมพันธ์ระหว่างตารางถูกกำหนดด้วย ทฤษฎีทางคณิตศาสตร์ เรื่องเซต (Set) (Codd, 1990) เช่น การเชื่อมตาราง (Join Table) จึงทำให้ฐานข้อมูลรีเลชันนอล เป็นที่นิยมอย่างสูงตลอดหลายสิบปีที่ผ่านมา

คุณสมบัติของรีเลชันนอล

- (1) ไม่มี ทูเปิลคู่ใด ๆ ที่ซ้ำกัน
- (2) ลำดับของ ทูเปิลไม่มีความสำคัญ
- (3) ลำดับของ แอททริบิวท์ไม่มีความสำคัญ
- (4) ค่าของ แอททริบิวท์จะเป็นค่าเดียว คือ ค่าของข้อมูลที่ปรากฏอยู่ใน ตารางจะเป็นค่าเดียว เป็นลิสต์ของหลาย ๆ ค่าไม่ได้ ซึ่งความสัมพันธ์ที่มีคุณสมบัตินี้จะถูกเรียกว่าเป็นความสัมพันธ์ที่อยู่ในรูปแบบที่เป็นบรรทัดฐาน (Normal Form)
- (5) ค่าของข้อมูลในแต่ละ แอททริบิวท์จะบรรจุค่าของข้อมูลประเภทเดียวกัน

1.1 แนวคิดของ E.F Codd

ในปี 1969, Dr. E.F.Codd เสนอแบบจำลองฐานข้อมูลที่อาศัยพื้นฐานของกลุ่มข้อมูลเชิงเลขคณิต (Mathematical Concept of relational sets) ตั้งแต่ปี 1970 และได้เสนอโมเดลแบบรีเลชันนอล (Relational Model) โดย Dr.E.F. Codd ได้เสนอกฎ 12 ข้อ ที่เป็นบรรทัดฐานว่าระบบฐานข้อมูลใดบ้างที่ถือได้ว่าเป็นฐานข้อมูลแบบรีเลชันนอล (Codd, 1970) จากกฎ 12 ข้อนี้ระบบฐานข้อมูลจึงถูกพัฒนามาขึ้นอย่างจริงจัง

กฎข้อที่ 1 กฎสารสนเทศ (The Information Rule)

กฎสารสนเทศกำหนดว่า สารสนเทศทั้งหมดในฐานข้อมูลต้องแสดงอยู่ในรูปแบบเดียวกันเท่านั้น คือ ความสัมพันธ์ ซึ่งถือได้ว่าเป็นหลักการพื้นฐานของแบบจำลองเชิงสัมพันธ์

กฎข้อที่ 2 กฎการรับประกันการเข้าถึงข้อมูล (Guaranteed Access Rule)

ค่าทุกค่าที่ถูกจัดเก็บไว้ในฐานข้อมูลต้องสามารถอ้างอิงได้ในทางตรรกะโดยการใส่ชื่อตาราง ชื่อคอลัมน์ และค่าของคีย์หลัก

กฎข้อที่ 3 การจัดการกับค่า Null อย่างเป็นระบบ (Systematic treatment of null values)

ระบบจัดการฐานข้อมูลมีการใช้ค่า Null แทนสารสนเทศที่ขาดหายไป (missing information) หรือ ไม่มีสารสนเทศที่เหมาะสม (inapplicable information) ค่า Null แตกต่างจากค่าทั่วไป เช่น แตกต่างจากค่าศูนย์ หรือ ค่าตัวเลข ดังนั้นจึงต้องมีวิธีการจัดการกับค่า Null อย่างเป็นระบบด้วย

กฎข้อที่ 4 แอ็กทีฟออนไลน์แคตตาล็อกที่อยู่บนพื้นฐานของแบบจำลองรีเลชันนอล

(Active online catalog based on the relational model)

ระบบต้องสนับสนุนรีเลชันนอลแบบออนไลน์ที่สามารถเข้าถึงได้โดยผู้ที่มีสิทธิ์ผ่านทางภาษาคิวรี

กฎข้อที่ 5 กฎของภาษา (Comprehensive data sublanguage rule)

ระบบต้องสนับสนุนอย่างน้อย 1 ภาษาที่เป็นรีเลชันนอล ที่สามารถใช้ได้ทั้งแบบโต้ตอบ และใช้ผ่านทางโปรแกรมสนับสนุนการนิยามข้อมูล รวมทั้งการวิว การจัดการข้อมูล ความถูกต้องของข้อมูล และการจัดการทรานแซกชัน

กฎข้อที่ 6 กฎการแก้ไขข้อมูลผ่านทางวิว (View updating rule)

วิวทั้งหมดตามทฤษฎีนั้นต้องสามารถปรับปรุงหรือแก้ไขได้โดยระบบ

กฎข้อที่ 7 ความสามารถในการเพิ่ม ลบ และแก้ไขข้อมูล (High-level insert, update and delete)

ระบบต้องสนับสนุนการแทรก การแก้ไข และการลบข้อมูลลักษณะของเซต

กฎข้อที่ 8 ความเป็นอิสระของข้อมูลเชิงกายภาพ (Physical data independence)

การเปลี่ยนแปลงใด ๆ ในระดับสกีมาทางกายภาพ จะไม่มีผลกระทบต่อสกีมาทางตรรกะ

กฎข้อที่ 9 ความเป็นอิสระของข้อมูลในเชิงตรรกะ (Logical data independence)

การเปลี่ยนแปลงใด ๆ ในระดับสกีมาทางตรรกะ จะไม่มีผลกระทบต่อโปรแกรมประยุกต์

กฎข้อที่ 10 ความคงสภาพที่เป็นอิสระของข้อมูล (Integrity independence)

การที่จะควบคุมข้อมูลให้มีความคงสภาพ และเป็นไปตามกฎที่ตั้งไว้ นั้น ต้องสามารถระบุได้จาก แอปพลิเคชันโปรแกรม และเก็บในรายการ ซึ่งเงื่อนไขต่าง ๆ สามารถเปลี่ยนแปลงได้ ตามความเหมาะสม โดยไม่กระทบต่อแอปพลิเคชันที่ใช้งานอยู่

กฎข้อที่ 11 ความเป็นอิสระของการกระจาย (Distribution independence)

สมบัติการกระจายข้อมูลเป็นคุณสมบัติหนึ่งของฐานข้อมูล ที่ต้องการจะเพิ่มพื้นที่ในการเก็บข้อมูล โดยที่ผู้ใช้ไม่จำเป็นต้องรู้กระบวนการทำงาน แต่ฐานข้อมูลจะทำการกระจายผ่านแอปพลิเคชันโปรแกรม

กฎข้อที่ 12 กฎการไม่มีเวอร์ชันย่อย (Non subversion rule)

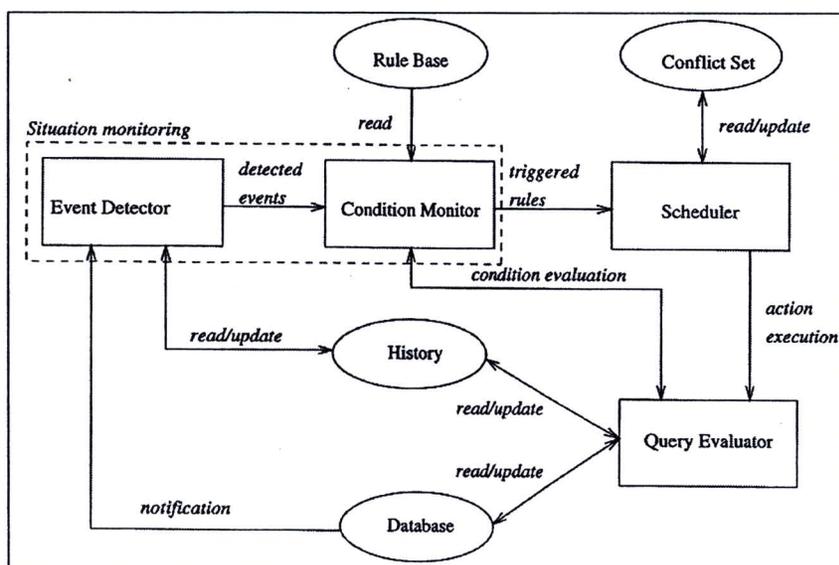
ถ้าระบบอนุญาตให้มีการติดต่อได้ในระดับต่ำสุดคือครั้งละเรคคอร์ด และการติดต่อนี้ต้องไม่ถูกใช้เส้นทางลัดโดยไม่ผ่านระบบความมั่นคงหรือเงื่อนไขความถูกต้องของข้อมูล

2. ฐานข้อมูลแบบแอคทีฟ

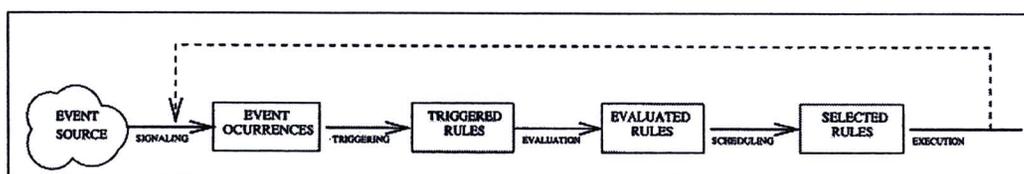
ฐานข้อมูลที่ใช้ในปัจจุบันเป็นฐานข้อมูลที่มีลักษณะเป็นแพชซีฟ กล่าวคือ เป็นฐานข้อมูลทำงานตามคำสั่ง โดยคำสั่งนั้นจะถูกส่งออกมาจากผู้ใช้งานหรือระบบ โดยมีระบบจัดการฐานข้อมูลเป็นผู้ตอบสนอง อย่างไรก็ตาม ในบางครั้งมีความจำเป็นที่ฐานข้อมูลต้องสามารถตรวจจับเหตุการณ์บางอย่างที่สำคัญ และทำการตอบสนองเหตุการณ์ที่เกิดขึ้นภายในระบบในทันที กล่าวคือ มีลักษณะที่เรียกว่า แอคทีฟ (Active) ตัวอย่างการทำงานของฐานข้อมูลแบบแอคทีฟ (Campin et al., 1997) (Chakravarthy et al., 1994) (Fratemali et al., 1995) ที่ใช้ในงานธุรกิจ เช่น ระบบการจัดสินค้าคงคลัง เมื่อมีการเบิกสินค้าบางรายการออกไปมากจนจำนวนที่คงเหลือในคลังสินค้าถึงจุดที่ต่ำกว่ากำหนด ระบบจะทำการเตือนให้ผู้รับผิดชอบทราบและเตรียมทำการจัดซื้อรายการสินค้าดังกล่าวต่อไป ในการพัฒนาระบบฐานข้อมูลทั่วไปที่มีลักษณะเป็นแบบแพชซีฟ สามารถทำได้โดยการเขียนคำสั่งโปรแกรมในการตรวจสอบสินค้าคงเหลือทุกครั้งที่มีการเบิกออกไป และทำการเตือนทุกครั้งที่จำนวนสินค้าคงคลังต่ำกว่าจุดกำหนด แต่การกระทำดังกล่าวมีปัญหาที่ ทุกโปรแกรมที่มีการเบิกสินค้านั้นมีมากกว่าหนึ่งจุด ทำให้ผู้เขียนโปรแกรมต้องฝังโปรแกรมในการตรวจสอบนี้เสมอ

สำหรับระบบจัดการฐานข้อมูล (Database Management System: DBMS) ที่สามารถตรวจจับเหตุการณ์ที่ตรงตามเงื่อนไขที่กำหนดไว้ได้ ระบบฐานข้อมูลนั้นจะถูกเรียกว่า ระบบฐานข้อมูลแบบแอคทีฟ ซึ่งความสามารถในการตรวจจับเหตุการณ์นี้ช่วยแก้ปัญหาข้างต้นด้วยการให้ระบบจัดการฐานข้อมูลตรวจสอบจำนวนสินค้าคงคลัง ทุกครั้งที่มีการเบิกจ่ายสินค้า หากสินค้า

คงคลังต่ำกว่ากำหนด ให้เรียกโปรแกรมที่ทำการเตือนผู้เกี่ยวข้องให้ทำงานทันที ความสามารถนี้ทำได้โดยการกำหนดเหตุการณ์ เงื่อนไขการตอบสนอง และวิธีการตอบสนองลงในระบบการจัดการฐานข้อมูล (Baralis et al., 1995) (Anca, 1999) ซึ่งกฎดังกล่าวสามารถใช้งานระหว่างระบบงานด้วยกันทำให้การประมวลผลเป็นไปอย่างมีประสิทธิภาพ โดยมีสถาปัตยกรรม ของ Abstract Active rule system ตามภาพที่ 3 และ ขั้นตอนการประมวลผลตาม ภาพที่ 4



ภาพที่ 3 การทำงาน Abstract Active rule system



ภาพที่ 4 ขั้นตอนการประมวลผลฐานข้อมูลแบบเอกทิฟ

จากภาพที่ 3 และ 4 จะเห็นได้ว่า คุณสมบัติของฐานข้อมูลแบบเอกทิฟ นั้นมีแนวคิดมาจาก production rule ของระบบปัญญาประดิษฐ์ (Artificial Intelligent) ที่แสดงในรูปของ Condition -> Action เมื่อเกิดเหตุการณ์ตรงตามเงื่อนไขที่กำหนด ระบบจะถูกกระตุ้น (triggered หรือ fired) ให้ทำงานตามกระบวนการที่กำหนดไว้ ซึ่งแนวความคิดดังกล่าวถูกนำมาประยุกต์ให้เข้ากับแนวคิดฐานข้อมูลแบบเอกทิฟ ในรูปของ เหตุการณ์ -> เงื่อนไข -> กระบวนการเมื่อเกิดการประมวลผลใน

ฐานข้อมูล เป็นเหตุการณ์ที่สนใจเกิดขึ้นซึ่งทำให้สถานะในฐานข้อมูลเปลี่ยนแปลงไปตามเงื่อนไข และระบบฐานข้อมูลจะทำงานตามกระบวนการที่กำหนด

มีหลายงานวิจัยที่ศึกษาเรื่องฐานข้อมูลแบบแอคทิฟ ในด้านต่าง ๆ เช่น ความพยายามที่จะจัดการกับกฎที่มีในฐานข้อมูลแบบแอคทิฟ (Viana et al., 2006) และการวัดประสิทธิภาพของฐานข้อมูลแบบแอคทิฟ โดยใช้อัลกอริทึม (Algorithm) เพื่อวัดประสิทธิภาพการใช้งานซีพียู และหน่วยความจำ (Abbas et al., 2006) (Alesheykh et al., 2005) (Ceri et al., 2003)

2.1 Event

Event คือ เหตุการณ์ที่สนใจ เราสามารถแบ่งชนิดของเหตุการณ์ได้เป็น 2 ประเภทคือ

2.1.1 เหตุการณ์พื้นฐาน (Primitive event) เป็นเหตุการณ์ที่เกิดขึ้นเดี่ยว ๆ เช่น เหตุการณ์การเพิ่มข้อมูลใส่ในทัพเพิล (tuples) ในตารางฐานข้อมูล นอกจากนี้ primitive event เกิดเฉพาะเวลาใดเวลาหนึ่งโดยสามารถจับกลุ่มของ Primitive event ได้ดังนี้

(1) Structure operation เป็นเหตุการณ์ที่เกิดขึ้นภายในฐานข้อมูล เช่น การเพิ่มข้อมูล การลบข้อมูล และการแก้ไขข้อมูล

(2) Behavior invocation เป็นเหตุการณ์เกิดขึ้นจากการประมวลผลโดยผู้ใช้งานเป็นผู้กำหนด เช่น การใช้ before เข้าร่วมร่วมในการเขียนโปรแกรม

(3) Transaction เป็นเหตุการณ์ที่เกิดขึ้นจากการใช้คำสั่งทรานแซคชัน (transaction) เช่น คำสั่ง abort, commit, begin – transaction

(4) Abstract or user – defined เป็นเหตุการณ์ที่เกิดขึ้นโดยกลไกของการเขียนโปรแกรมของผู้ใช้งานที่เพิ่มเติมในฐานข้อมูล

(5) Exception เป็นข้อบกพร่องที่เกิดขึ้นภายในฐานข้อมูล เช่น การพยายามเข้าถึงข้อมูลโดยไม่ได้รับการอนุญาต

(6) Clock เป็นเหตุการณ์ที่เกิดขึ้น ณ เวลาใดเวลาหนึ่ง เป็นเหตุการณ์ที่มีความสัมพันธ์กัน เช่น สิบวันหลังจากการขาย และเป็นเหตุการณ์ที่เกิดขึ้นบางครั้งบางคราว เช่น ทุก ๆ วันแรกของเดือน ซึ่งเหตุการณ์เวลาจะถูกรายงาน

(7) External เป็นเหตุการณ์ที่เกิดขึ้นภายนอกฐานข้อมูล

2.1.2 เหตุการณ์ที่เกิดจากการรวม (Composite events) เป็นเหตุการณ์ที่เกิดขึ้นจากการรวมกันของเหตุการณ์พื้นฐาน หรือ เป็นเหตุการณ์ที่เกิดขึ้นในลักษณะของพีชคณิต (event algebra) ยกตัวอย่างเช่น



ตัวอย่างที่ 1

on insert to Holds or update to stored# of Holds and

on update to level of Holds and update to sales of Holds

จากตัวอย่างที่ 1 เป็นการรวมเหตุการณ์พื้นฐาน 2 เหตุการณ์ไว้ในคำสั่งเดียว คือ เหตุการณ์ที่เกิดการ insert และ เหตุการณ์ที่เกิดจากการ update

เหตุการณ์ที่เป็นลักษณะของ Composite event นั้นสามารถพบได้จากฐานข้อมูล HiPAC project (DBM88) ฐานข้อมูลประเภท OODB เช่น GJS92, GD94, CKAK94 และ CFPT 96

2.2 Condition

Condition คือ เงื่อนไขซึ่งมีลักษณะเป็นทางเลือก (optional) เมื่อเกิดเหตุการณ์ที่สนใจเกิดขึ้นจะมีการตรวจสอบต่อว่าเหตุการณ์ที่เกิดขึ้นตรงตามเงื่อนไขที่กำหนดหรือไม่ หากตรงตามเงื่อนไขที่กำหนดจะกระทำ (action) ตามที่ได้กำหนดไว้ หากเหตุการณ์ที่เกิดขึ้นไม่ตรงกับเงื่อนไขที่กำหนดก็จะไม่กระทำ

```
create trigger tgl before update on student for each row begin
if new.amount < 500 then set new.password = 'OK'; end if; end;
```

ภาพที่ 5 การสร้างเงื่อนไข

จากภาพที่ 5 ส่วนที่เป็นเงื่อนไขคือ if new.amount < 500 เมื่อเกิด event update ขึ้นที่แอตทริบิว amount ในตาราง student จากนั้นให้ทำการตรวจสอบเงื่อนไขว่า ถ้าเหตุการณ์นั้นตรงตามเงื่อนไขที่กำหนดไว้คือค่าของ amount น้อยกว่า 500 ให้กำหนดค่าใหม่ของ password มีค่าเป็น 'OK' แต่ถ้าเหตุการณ์นั้นไม่ตรงตามเงื่อนไขก็ไม่ต้องกำหนดค่าใหม่ให้กับ password

2.3 Action

Action คืองาน หรือการกระทำที่เกิดขึ้นและเป็นไปตามเงื่อนไขที่กำหนดไว้ Action นั้นสามารถปรับปรุ่ค่าในฐานข้อมูลที่เกิดเหตุการณ์ หรือฐานข้อมูลอื่นได้

จากภาพที่ 6 เมื่อเกิดเหตุการณ์ update ที่ตาราง chem ในฐานข้อมูล mylabchem แล้วทำให้ค่าของ amount น้อยกว่า 500 นั้นเป็นจริง ให้ทำการ insert ข้อมูลในฐานข้อมูล activedb ในตาราง t_result โดยกำหนดให้ tname เท่ากับค่า cnameth



```
create trigger mylabchem.Trth before update on mylabchem.chem for
each row begin
if new.amount < 500 then set new.amount = old.amount;insert into
activedb.t result set numA=0,tname =old.cnameth end if;end;"
```

ภาพที่ 6 action นอกตารางที่เกิดเหตุการณ์

3. ทริกเกอร์

ทริกเกอร์ (Potaminsto et al., 1996) ทำหน้าที่เป็น store procedure (วิสุทธิ, 2003) คือ ฟังก์ชันที่ผู้พัฒนาสามารถสร้างขึ้นและติดตั้งให้กับฐานข้อมูลเพื่อขยายความสามารถในการสืบค้น และการประมวลผล store procedure จะถูกเรียกขึ้นมาใช้งานเมื่อมีเหตุการณ์ที่น่าสนใจ วิธีการสร้างทริกเกอร์เพื่อให้ตอบสนองกับเหตุการณ์ที่เกิดขึ้น จากขอบเขตของงาน ได้พัฒนาทริกเกอร์บนพื้นฐานของภาษา SQL ซึ่งสามารถเขียนคำสั่งได้ดังนี้

3.1 การสร้างทริกเกอร์สามารถสร้างโดยใช้คำสั่ง CREATE TRIGGER ซึ่งมี Syntax คือ

```
CREATE TRIGGER name {BEFORE | AFTER}
{event | OR ...}
ON table FOR EACH {ROW | STATEMENT}
EXECUTE PROCEDURE func (arguments)
```

ภาพที่ 7 Syntax คำสั่ง CREATE TRIGGER

จากภาพที่ 7 จะเห็นได้ว่าการประกาศทริกเกอร์เป็นการสร้างกฎในฐานข้อมูล โดยกำหนดว่าเมื่อคำสั่งจะมีเหตุการณ์ (Event) เกิดขึ้นกับตาราง (table) ที่กำหนดไว้ก่อน หรือ หลัง (BEFORE|AFTER) ที่จะประมวลผลตามที่เกิดในเหตุการณ์ ให้ระบบจัดการฐานข้อมูลเรียกใช้ store procedure (Func) ที่ระบุไว้ นอกจากนี้ในตารางที่ 2 ได้แสดงความหมายตัวแปรและตัวเลือกต่าง ๆ ในคำสั่ง CREATE TRIGGER

ตารางที่ 2 ตัวแปรและตัวเลือก ในคำสั่ง CREATE TRIGGER

ตัวแปร / ตัวเลือก	คำอธิบาย
Name	ชื่อของทริกเกอร์ที่จะประกาศ
Table	ชื่อของตารางที่จะกระตุ้นให้ทริกเกอร์ทำงานเมื่อเกิดเหตุการณ์ที่กำหนดให้
Event	เหตุการณ์ที่เกิดขึ้นกับตารางประกอบด้วย INSERT, DELETE หรือ UPDATE โดยสามารถประกาศมากกว่าหนึ่งเหตุการณ์ในแต่ละทริกเกอร์ด้วยการใช้ OR
Func	ชื่อของ Store Procedure หรือฟังก์ชันที่ทริกเกอร์จะเรียกใช้เมื่อเกิดเหตุการณ์ที่กำหนด
BEFORE AFTER	เป็นการกำหนดให้ทริกเกอร์ทำการเรียกฟังก์ชันขึ้นมาทำงาน ก่อน หรือ หลัง ที่ event (INSERT, DELETE หรือ UPDATE) จะเกิดขึ้น
ROW STATEMENT	เป็นการกำหนดว่าทริกเกอร์จะทำงานสำหรับแต่ละรายการข้อมูล (row) หรือ ทำงานครั้งเดียวเมื่อคำสั่ง SQL ทั้งหมดทำงานเสร็จ (Statement)

3.2 Trigger procedure

Trigger procedure มีข้อกำหนดคือ เป็น store procedure ที่ไม่ต้องประกาศ argument แต่ตอนใช้งานจริงนั้นสามารถส่งค่า argument ได้ และต้องมีการคืนค่า argument ใช้สำหรับฟังก์ชันที่คืนค่าที่ฐานข้อมูล MySQL ไม่สามารถนำมาประมวลผลได้ โดยตรง ตัวอย่าง การสร้างและลบทริกเกอร์ การรวมค่าที่ถูก insert ลงมายัง column amount ในตาราง account มีขั้นตอนดังนี้

ขั้นตอนที่ 1 สร้างตารางชื่อ account มี 2 column คือ acct_num เลขจำนวนเต็ม และ amount เป็นทศนิยม 2 ตำแหน่ง ดังภาพที่ 8

```
mysql> CREATE TABLE account (acct_num INT, amount DECIMAL(10,2));
Query OK, 0 rows affected (0.19 sec)
```

ภาพที่ 8 source code ของคำสั่งสร้างตาราง account

ขั้นตอนที่ 2 สร้างทริกเกอร์โดยนำค่าใหม่ที่ถูกรับ insert ลง column amount มาบวกกันและเก็บผลลัพธ์ไว้ในตัวแปร @sum ดังภาพที่ 9

```
mysql> CREATE TRIGGER ins_sum BEFORE INSERT ON account
-> FOR EACH ROW SET @sum = @sum+NEW.amount;
Query OK, 0 rows affected (0.13 sec)
```

ภาพที่ 9 source code ของคำสั่งสร้างทริกเกอร์ ชื่อ ins_sum

ขั้นตอนที่ 3 ทำการกำหนดค่าเริ่มต้นของ @sum ให้เท่ากับ 0 ดังภาพที่ 10

```
mysql> SET @sum = 0;
Query OK, 0 rows affected (0.00 sec)
```

ภาพที่ 10 source code ของคำสั่งกำหนดค่าเริ่มต้นของ @sum

ขั้นตอนที่ 4 ทำการเพิ่มค่าลงในแต่ละ column ในตาราง account ดังภาพที่ 11

```
mysql> INSERT INTO account VALUES(137,14.98), (141,1937.50), (97, -100.00);
Query OK, 3 rows affected (0.00 sec)
Records: 3 Duplicates: 0 Warnings: 0
```

ภาพที่ 11 source code ของคำสั่งการเพิ่มค่าลงในแต่ละ column ในตาราง account

ขั้นตอนที่ 5 ทำการแสดงค่าของตัวแปร @sum ที่รวมเอาค่าที่ insert ลง column amount มาบวกกัน คือ $14.98+1937.50+(-100.00) = 1852.48$ ดังภาพที่ 12

```
mysql> SELECT @sum AS 'Total amount inserted';
+-----+
| Total amount inserted |
+-----+
| 1852.48                |
+-----+
1 row in set (0.00 sec)
```

ภาพที่ 12 source code ของคำสั่งแสดงค่าของตัวแปร @sum

ขั้นตอนที่ 6 การลบทริกเกอร์ มี syntax ดังนี้

```
DROP TRIGGER ชื่อschema.ชื่อtrigger
```

ภาพที่ 13 syntax ของคำสั่งลบทริกเกอร์

```
mysql> DROP TRIGGER test.ins_sum;
Query OK, 0 rows affected (0.09 sec)
```

ภาพที่ 14 source code ของคำสั่งลบทริกเกอร์ ชื่อ ins_sum

4. ข้อมูลตัวเลข

การศึกษาเรียนรู้ประเภทข้อมูล ทำให้สามารถเลือกประเภทของข้อมูลที่น่าไปใช้ในการทำวิจัยได้อย่างถูกต้อง และเหมาะสมตามจุดประสงค์ ข้อมูลตัวเลข (numerical data) เป็นข้อมูลที่ใช้ในการคำนวณ หรือการจัดเรียงข้อมูลเปรียบเทียบในฟิลด์นั้น ๆ ประกอบด้วยข้อมูลย่อย ๆ ได้แก่ จำนวนเต็ม จำนวนทศนิยม และจำนวนจริง จากการศึกษาและค้นคว้าพบว่า มีหลายงานวิจัยที่ได้นำข้อมูลตัวเลขมาประกอบในการทำวิจัย หรือนำมาเป็นข้อมูลตัวอย่างในการทำวิจัย อาทิเช่น การประมาณการสืบค้นในฐานข้อมูลเชิงตัวเลข (Approximate Query Answering In Numerical Database) (Hachem et al., 1996) และ การตรวจสอบตัวดำเนินการในฐานข้อมูลเชิงตัวเลข (Diagnostics and Engine management / numeric database) (Hidalgo, 2007) จากการศึกษางานวิจัยที่ได้กล่าวมาในเบื้องต้นนั้น สามารถสรุป คุณสมบัติของข้อมูลตัวเลขได้ดังนี้

- 4.1 ข้อมูลตัวเลขบ่งบอกถึงปริมาณในเชิงของรูปธรรมได้อย่างชัดเจน
- 4.2 ข้อมูลตัวเลขมีหลายชนิดด้วยกัน เช่น ชนิดของเลขจำนวนเต็ม ชนิดของทศนิยม และจำนวนจริง
- 4.3 ในเชิงของการจัดเก็บข้อมูล ข้อมูลตัวเลขนั้นจะใช้พื้นที่ในหน่วยความจำ และใช้เวลาในการค้นหา น้อยกว่าข้อมูลประเภทอื่น ๆ
- 4.4 ข้อมูลตัวเลขสามารถใช้สมการทางคณิตศาสตร์ช่วยในการประมวลได้
- 4.5 ข้อมูลตัวเลขสามารถกำหนดช่วงของข้อมูลและขอบเขตของข้อมูลได้ เช่น ค่ามาก (Maximum) หรือ ค่าน้อย (Minimum)

คุณสมบัติของข้อมูลตัวเลขที่กล่าวไว้ ทำให้ข้อมูลตัวเลขมีความเหมาะสมที่จะใช้เป็นข้อมูลทดสอบการทำงานของฐานข้อมูลแบบแอคทิฟ สิ่งที่ทำให้ข้อมูลตัวเลขมีความน่าสนใจอีกประการหนึ่ง คือ ข้อมูลตัวเลขมีการเปลี่ยนแปลงมากกว่าข้อมูลชนิดอื่น ๆ ซึ่งการเปลี่ยนแปลงนั้นอาจส่งผลกระทบต่อกฎหรือเงื่อนไขที่กำหนดขึ้น

5. เอเจนต์

นิยามของเอเจนต์มีได้หลายความหมาย เราสามารถบรรยายนิยามเอเจนต์ในรูปแบบของงาน ความอิสระและประสิทธิภาพในการสื่อสาร ตัวอย่างงานวิจัยที่นิยามความหมายของเอเจนต์ เช่น Smith Cypher และ Spohrer (Smith et al., 2007) ได้ให้คำนิยามของเอเจนต์ ว่าเป็นซอฟต์แวร์ที่ถูกพัฒนาขึ้นตามวัตถุประสงค์ โดยอาจประกอบไปด้วยหลายแอปพลิเคชันรวมกัน (Hayes et al., 1995) นิยามการทำงานของเอเจนต์ว่า เอเจนต์มีการทำงาน 3 รูปแบบ คือ หนึ่งคือการเรียนรู้เงื่อนไขที่มีการเปลี่ยนแปลงในสภาวะแวดล้อม สองคือการกระทำที่ส่งผลตามเงื่อนไข และ สามคือเหตุผลที่เกี่ยวกับ การเรียนรู้ การแก้ไขปัญหา การอนุมาน และการตัดสินใจกระทำ

ในงานวิจัยนี้ได้ใช้นิยามของเอเจนต์ของ Selker (Selker, 1994) ว่าเป็นโปรแกรมคอมพิวเตอร์ที่จำลองความสัมพันธ์ของมนุษย์ และ Wooldridge และ Jennings (Wooldridge et al., 1995) กล่าวถึง คุณสมบัติของ เอเจนต์ ดังนี้

- 5.1 ออโตโนมี (Autonomy) : เอเจนต์สามารถแก้ปัญหาหรือสามารถควบคุมการกระทำได้ด้วยตัวเอง
- 5.2 โซเชียล อาบิลิตี (Social ability) : เอเจนต์สามารถตอบสนองต่อสิ่งที่เกิดขึ้นได้อย่างเหมาะสม และช่วยแก้ปัญหาของซอฟต์แวร์และมนุษย์ได้
- 5.3 รีแอกทีวิตี (Reactivity): รับรู้สิ่งแวดล้อมที่เกิดขึ้น เช่น สิ่งแวดล้อมทางกายภาพ ส่วนต่อประสานผู้ใช้ กลุ่มของเอเจนต์อื่น ระบบอินเทอร์เน็ต หรือ เกิดจากการผสมผสานในหลาย ๆ ส่วน และ มีการตอบสนองทันทีที่ต่อเหตุการณ์นั้น ๆ
- 5.4 โปรแอคทีฟเนส (Pro-activeness): เอเจนต์ ไม่ได้มีเพียงการตอบสนองต่อสิ่งแวดล้อมเท่านั้น แต่ยังสามารถ สร้างพฤติกรรมเพื่อแก้ไขปัญหาที่ต้องการได้ โดยการเพิ่มส่วนของการทำงานแบบยืดหยุ่นตามเงื่อนไข

6. ภาษาจาวา (Java Language)

ภาษาจาวาเป็นภาษาคอมพิวเตอร์ที่พัฒนาขึ้น โดยทีมวิจัยของบริษัทซันไมโครซิสเต็มส์ ซึ่งเริ่มต้นเมื่อปี 1991 โดยทางบริษัทซันไมโครซิสเต็มส์ได้สนับสนุนให้มีการพัฒนาโครงการที่ชื่อ

Green Project ที่มีจุดมุ่งหมายที่จะออกแบบภาษาคอมพิวเตอร์ขนาดเล็กเพื่อใช้กับอุปกรณ์อิเล็กทรอนิกส์ต่างๆ เช่น สวิตช์บล็อกของเคเบิลทีวี ซึ่งอุปกรณ์เหล่านี้จะมีข้อจำกัดในด้านหน่วยความจำและหน่วยประมวลผลกลางที่จะมีความแตกต่างกัน ดังนั้น ภาษาคอมพิวเตอร์ที่พัฒนาขึ้นจะต้องไม่ขึ้นอยู่กับแพลตฟอร์ม (Platform Independent) ผลงานของทีมวิจัยนี้ทำให้ได้ภาษาคอมพิวเตอร์ใหม่ที่ชื่อว่า “Oak” ซึ่งมีโครงสร้างและคำสั่งคล้ายภาษา C++ เนื่องจากทีมวิจัยของบริษัทซันไมโครซิสเต็มส์มีความคุ้นเคยกับระบบปฏิบัติการยูนิกซ์ซึ่งมักจะใช้ภาษา C++ ในการพัฒนาโปรแกรม ภาษา “Oak” ต่อมาได้เปลี่ยนชื่อมาเป็น “Java” เนื่องจากทีมวิจัยได้ทราบภายหลังว่า “Oak” เป็นชื่อภาษาคอมพิวเตอร์ที่มีอยู่แล้ว

ภาษาจาวาเป็นภาษาคอมพิวเตอร์เชิงออบเจกต์ (Object Oriented Programming) โดยมีจุดเด่นตรงที่สามารถทำงานได้กับระบบคอมพิวเตอร์หลายแพลตฟอร์ม และผู้ที่คิดค้นภาษาจาวาจนได้รับการยกย่องว่าเป็นบิดาของภาษาจาวาก็คือ James Gosling

6.1 เทคโนโลยีจาวา

เทคโนโลยีจาวาประกอบไปด้วยองค์ประกอบหลัก ๆ 3 ส่วนคือ

- 6.1.1 Java Virtual Machine (JVM) เป็นเทคโนโลยีจาวาที่ทำหน้าที่เป็นอินเตอร์พรีเตอร์ ที่จะแปลโปรแกรมจาวาไบต์โค้ด (Java Bytecode) ให้เป็นภาษาที่เครื่องเข้าใจได้
- 6.1.2 Java Runtime Environment (JRE) เป็นเทคโนโลยีจาวาที่ใช้ในการรันโปรแกรมภาษาจาวา อาทิเช่น โปรแกรมจาวาประยุกต์ (Java Application) และโปรแกรมจาวาแอปเพล็ต (Java Applet) JRE จะประกอบด้วย JVM และ Java Application programming Interface (Java API) ที่จะรวบรวมคลาสและอินเตอร์เฟสต่างๆ ที่จำเป็นต่อการใช้งานของโปรแกรมภาษาจาวา
- 6.1.3 Java Software Developer Kit (JDK) เป็นชุดพัฒนาโปรแกรมภาษาจาวาที่จะประกอบไปด้วย JVM ตัวแปลภาษาจาวา (Java Compiler) เครื่องมือ (Tool) อื่น ๆ ในการพัฒนาโปรแกรม และ API ทั้งหมดในภาษาจาวา ซึ่ง API จะเป็นมาตรฐานคำสั่งต่าง ๆ ของภาษาจาวา

6.2 Java Software Developer Kit

JDK คือ ชุดพัฒนาโปรแกรมภาษาจาวาของบริษัทซันไมโครซิสเต็มส์ ประกอบด้วยโปรแกรมต่าง ๆ อาทิเช่น โปรแกรมคอมไพเลอร์ (Javac.exe) โปรแกรมอินเตอร์พรีเตอร์ (java.exe) และโปรแกรมดีบั๊กเกอร์ เป็นต้น แต่ชุดโปรแกรม JDK จะเป็นเพียงเซ็ทย่อยของ

โปรแกรมประเภท Software Development Kit (SDK) โดย JDK จะไม่มีโปรแกรมอิดิเตอร์ (Editor) สำหรับการเขียนซอร์สโค้ดหรือรันโปรแกรม ดังนั้น ผู้ใช้จะต้องใช้โปรแกรม SDK อื่นช่วยในการเขียนซอร์สโค้ด

นอกจากนี้ โปรแกรมที่อยู่ใน JDK จะเป็นโปรแกรมที่ต้องเรียกใช้งานผ่านทาง command line จึงไม่สะดวกในการใช้งาน แต่ก็มีข้อดีตรงที่ทำให้ผู้ที่เริ่มต้นในการพัฒนาโปรแกรมสามารถเข้าใจขั้นตอนต่าง ๆ ของการพัฒนาโปรแกรมโดยใช้ภาษาจาวาได้ดียิ่งขึ้น โดยไม่มุ่งเน้นการใช้เครื่องมือหรือคำสั่งต่าง ๆ ที่มีอยู่ในชุดพัฒนาโปรแกรมภาษาจาวาโปรแกรมอื่น ๆ

นอกจากโปรแกรม JDK ที่พัฒนาโดยบริษัทซันไมโครซิสเต็มส์แล้ว บริษัทอื่น ๆ ก็มีการพัฒนาโปรแกรม JDK ขึ้นมา โดยบางส่วนอาจใช้ซอร์สโค้ดของทางซันไมโครซิสเต็มส์ ซึ่งโปรแกรม JDK เหล่านี้จะมีมาตรฐานคำสั่งที่เหมือนกัน แต่อาจจะมีข้อแตกต่างในคุณลักษณะอื่น ๆ เช่น คอมไพเลอร์ หรือ Garbage Collection ตัวอย่างของโปรแกรม JDK อื่น ๆ มีอาทิเช่น J9 ของบริษัทไอบีเอ็ม หรือ JRockit ของบริษัท Oracle โปรแกรม SDK มีเครื่องมือเพื่อช่วยในการสร้างโปรแกรมภาษาจาวาในภาพแบบต่าง ๆ ได้ง่ายขึ้นอาทิเช่น เครื่องมือช่วยในการออกแบบโปรแกรมกราฟฟิก

สำหรับชุดพัฒนาโปรแกรม JDK นั้นจำเป็นต้องมีโปรแกรมอิดิเตอร์เพื่อใช้ในการเขียนซอร์สโค้ดของโปรแกรมภาษาจาวา ซึ่งสามารถใช้โปรแกรม TextEditor ทั่วไปได้เช่น โปรแกรม Notepad ในระบบปฏิบัติการ Windows แต่หากต้องการพัฒนาโปรแกรมให้เร็วขึ้น ผู้ใช้ควรเลือกใช้โปรแกรมอิดิเตอร์ที่ออกแบบมาเพื่อเขียนซอร์สโค้ดของโปรแกรมภาษาจาวา โดยเฉพาะ ซึ่งโปรแกรมเหล่านี้จะทำงานคล้ายกับโปรแกรมแบบ SDK โดยผู้ใช้สามารถเรียกคำสั่งในการคอมไพล์หรือการรันโปรแกรมที่พัฒนาขึ้นจากเมนูที่กำหนดไว้ได้ ตัวอย่างโปรแกรมอิดิเตอร์แบบนี้ คือ

6.3 Java Platform, Standard Edition

แพลตฟอร์ม Java SE เป็นแพลตฟอร์มจาวามาตรฐานที่ใช้ในการพัฒนาโปรแกรมภาษาจาวาเพื่อใช้งานกับเครื่องคอมพิวเตอร์ทั่วไป Virtual Machine ที่กำหนดไว้ในแพลตฟอร์มนี้คือ JVM และมี Java API ที่กำหนดไว้ต่าง ๆ อาทิเช่น java.lang, java.util, Java.io และ java.net แพลตฟอร์ม Java SE ใช้ในการพัฒนาโปรแกรมจาวาที่ใช้ในเครื่องคอมพิวเตอร์ลูกข่าย โดยแบ่งออกเป็น 2 ประเภท คือ

6.3.1 โปรแกรมจาวาประยุกต์ (Java Application) คือ โปรแกรมภาษาจาวาที่ใช้งานทั่วไป

6.3.2 โปรแกรมจาวาแอปเพล็ต (Java Applet) คือโปรแกรมภาษาจาวาที่รันบนเว็บเบราว์เซอร์

6.4 ลักษณะของภาษาจาวา

ภาษาจาวาถูกพัฒนาจากบริษัทซันไมโครซิสเต็มส์ (Sun Microsystems) ซึ่งจัดให้เป็นภาษาคอมพิวเตอร์ภาษาหนึ่งที่ใช้หลักการออกแบบตัวภาษาด้วยวิธีเชิงวัตถุ และตัวภาษาจะใช้เป็นเครื่องมือสำหรับพัฒนาโปรแกรมด้วยแนวคิดเชิงวัตถุ โดยตัวภาษามีลักษณะพิเศษดังนี้

- 6.1.1 Portability สามารถใช้งานในสภาพแวดล้อมที่แตกต่าง โดยไม่ต้องมีการปรับแต่ง
- 6.1.2 Simple ความง่ายในการเขียนโปรแกรม
- 6.1.3 Robust ความคงสภาพในการทำงาน ทำให้ไม่เกิดข้อผิดพลาดที่ไม่พึงประสงค์
- 6.1.4 Secure การรองรับมาตรฐานความปลอดภัยในการใช้งานภาพแบบต่าง ๆ
- 6.1.5 Object-Oriented มีหลักการของแนวคิดเชิงวัตถุ
- 6.1.6 Distributed มีความสามารถในการประมวลผลแบบกระจาย

เหตุผลหนึ่งที่ภาษาจาวาจัดว่าเป็นภาษาที่ได้รับความนิยมและได้รับการกล่าวถึงอย่างมากก็คือ ลักษณะคำจำกัดความที่ว่า “Write Once Run Anywhere” หรือ “เขียนครั้งเดียวใช้ได้ทุกที่” กล่าวคือ เมื่อสร้างโปรแกรมขึ้นมาจากภาษาจาวา โปรแกรมสามารถนำไปประมวลผลหรือทำงานได้ กับเครื่องคอมพิวเตอร์ทุกระบบปฏิบัติการใด เช่น วินโดว์, ลินุกซ์ หรือยูนิกซ์ โดยไม่ต้องมีการปรับปรุงตัวโปรแกรมทุกครั้งที่มีการเปลี่ยนแปลงจากระบบปฏิบัติการหนึ่งไปสู่ระบบปฏิบัติการอื่น

6.5 เครื่องมือพื้นฐานในการเขียนโปรแกรมด้วยแบบกราฟิกในภาษาจาวา

Graphic User Interface (GUI)

ปัจจุบันการพัฒนาด้าน User Interface มีมากขึ้นสามารถนำกราฟิก (Graphic) เข้ามาช่วยในการออกแบบหน้าจอหรือจอภาพให้สวยงามและยังสื่อสารกับผู้ใช้ได้อย่างมีประสิทธิภาพมากขึ้น ตัวอย่างเช่น การแสดงด้วยไอคอน ภาพภาพ ที่เหมือนจริง ลักษณะนี้เรียกว่า “Graphic User Interface” หรือ GUI

การเขียนโปรแกรมแบบ GUI (GUI Programming) จึงเป็นโปรแกรมที่ให้ผู้ใช้งานสามารถสื่อสารและทำงานกับโปรแกรมได้อย่างมีประสิทธิภาพมากขึ้น พร้อมทั้งเข้าใจ Interface หน้าทีขั้นตอนการทำงานและข้อความต่าง ๆ ของ Interface ได้ง่ายขึ้น

AWT

AWT หรือ Abstract Windowing Toolkit เป็นเครื่องมือพื้นฐานในการเขียนโปรแกรมด้วย GUI ในจาวาในการเขียนโปรแกรมด้วย GUI ต้องประกอบด้วย Component, Containers และ Layout Managers AWT Components เป็น Object ที่ Java ได้สร้างมาให้ใช้งานได้เลย ประกอบด้วย Class ดังต่อไปนี้ Component, Container, Button, Canvas, Checkbox, Choice, Dialog, Frame, Label, List, Menu, Menubar, MenuItem, Panel, Scrollbar, TextArea, TextComponent และ TextField

AWT Containers เป็น Containers อย่างหนึ่งที่สามารถนำ Component อื่น ๆ มาใส่ไว้ได้ประกอบด้วย Class ดังนี้ Applet, Container, Dialog, File Dialog, Frame, Panel, Scrollpane และ Windows

Layout Managers ใช้สำหรับแสดง Container ออกทางจอภาพ (ใช้จัดเรียง Containers) ประกอบด้วย Class ดังนี้ BorderLayout, FlowLayout, CardLayout, GridLayout และ GridBagLayout

Swing เนื่องจาก AWT เมื่อนำมาใช้ในการเขียน GUI program นั้นจะมีปัญหาบางอย่างเกิดขึ้น เช่น จะเปลืองหน่วยความจำของคอมพิวเตอร์ ทำให้มีการพัฒนา Package ชื่อว่า Swing ขึ้นมาเพื่อช่วยสนับสนุนการทำงานให้ดีขึ้น แต่ไม่ได้นำมาใช้งานแทน AWT เพราะยังใช้ AWT อยู่ Swing เป็น API (Application Programming Interface) ตัวหนึ่งใน Java Foundation Class (JFC) ซึ่งเป็นส่วนหนึ่งของ JDK โดย API ใน JFC ประกอบด้วย Accessibility, Drag and Drop, Java 2D, Swing และ Input Method Framework

6.6 Java Database Connectivity (JDBC)

JDBC เป็น Java API สำหรับการประมวลผลคำสั่ง SQL ซึ่งเป็น Class และ Interface ที่เป็นภาษาจาวา โดย JDBC ได้จัดเตรียม API มาตรฐานสำหรับเป็นเครื่องมือในการพัฒนา Database Application ในการเขียนโค้ดโปรแกรมภาษาจาวาซึ่ง JDBC สามารถติดต่อกับ Database ได้เกือบทุกประเภท เช่น Sysbase, Oracle, MS SQL, Informix, Access, MySQL ที่สำคัญการพัฒนา Application ด้าน Database โดยใช้จาวา สามารถนำไปรันได้หลายแพลตฟอร์ม ซึ่ง JDBC จะเข้ามาช่วยจัดการในด้านนี้ ทำให้การพัฒนา Application สามารถใช้คุณสมบัติของภาษาจาวาได้อย่างเต็มที่ ซึ่ง JDBC จะถูกประยุกต์ใช้งานในหลายด้าน เช่น ใน webpage ที่มี Applet ในการรัน Application แบบ Remote Database หรือมีการใช้ JDBC ในการติดต่อกับฐานข้อมูลซึ่ง JDBC เป็นคลาส ๆ หนึ่ง ทำให้ JDBC สามารถใช้งานได้หลาย Platform เช่น Windows, Linux, Unix, Solaris, MAC เป็นต้น

การทำงานที่สำคัญของ JDBC

6.6.1 การติดต่อกับ Data Base

6.6.2 สั่งคำสั่ง SQL ไปยัง Data Base

6.6.3 การรันผลลัพธ์จากการใช้คำสั่ง SQL แล้วทำการประมวลผล

JDBC เป็น low-level Interface ที่สามารถเรียกใช้คำสั่ง SQL ได้โดยตรง ซึ่งสามารถใช้งานได้อย่างมีประสิทธิภาพ มีขนาดเล็กและง่ายต่อการใช้งานมากกว่าตัวติดต่อ Data Base ชนิดอื่น ๆ แต่ก็ได้ออกแบบสำหรับใช้งานแบบ High-Level Interface ด้วย ซึ่ง High Level Interface คือ User Friendly ซึ่ง High-level API ของ JDBC ที่ถูกพัฒนาซึ่งมีดังนี้

JDBC ต้องการคำสั่ง SQL ที่จะทำการส่ง String ไปยัง Method ของภาษา Java ทำให้ Programmer สามารถใช้ตัวแปรในการรับส่งคำสั่ง SQL ได้

มีการแปลงตารางข้อมูลในฐานข้อมูลรีเลชันนอลไปเป็น Java Class ซึ่งเรียกว่า Object/Relational ซึ่งแต่ละแถวของตารางจะถูกแปลงไปเป็น Instant ของ Class และแต่ละ Column จะเป็นเหมือน attribute ของ Class ซึ่งนักเขียนโปรแกรมสามารถดำเนินการต่าง ๆ คล้ายว่าตารางเป็นคลาส ๆ หนึ่งในจาวา

ประเภทของ JDBC Driver

JDBC-ODBC Bridge plus ODBC Driver เป็น Driver ของ JDBC ที่ใช้ในการติดต่อการทำงานผ่าน ODBC ซึ่งจะมีการใช้ ODBC ในการติดต่อกับฐานข้อมูลก่อนไม่ว่าจะเป็นในเครื่องเดียวกันหรือติดต่อผ่าน Server แล้วจึงใช้ JDBC ไปติดต่อกับ ODBC อีกครั้งหนึ่ง

Native-API partly-Java Driver สำหรับ Driver ประเภทนี้จะเปลี่ยน JDBC calls ไปเรียก Client API ของ Oracle, Sysbase หรือ DBMS ตัวอื่น ๆ ซึ่งจะคล้ายสะพานเชื่อมต่อไปยังฐานข้อมูลอีก

JDBC-Net pure Java Driver สำหรับ Driver ประเภทนี้จะเปลี่ยน JDBC calls ไปยัง DBMS - independent net protocol ซึ่งหลังจากการเปลี่ยนไปเป็น DBMS Protocol โดย Server ซึ่ง Server ตัวนี้สามารถติดต่อเข้ากับ Java Client ไปยัง Database หลายประเภทได้ซึ่งจะสามารถใช้งานผ่านทาง Internet และ Intranet

Native - protocol pure Java Driver สำหรับ Driver ประเภทนี้จะเปลี่ยน JDBC call ไปเป็น network protocol ซึ่งจะถูกใช้โดย DBMS ซึ่งจะมีการใช้งานโดยมีการเรียกใช้งานโดยตรงจาก client ไปยัง DBMS Server

7. MySQL

MySQL (อ่านว่า “มาย-เอส-คิว-แอล”) จัดเป็นระบบจัดการฐานข้อมูลเชิงสัมพันธ์ (RDBMS: Relational Database Management System) ตัวหนึ่ง ซึ่งเป็นที่นิยมกันมากในปัจจุบัน โดยเฉพาะอย่างยิ่งในโลกของอินเทอร์เน็ต สาเหตุเพราะว่า MySQL เป็นฟรีแวร์ทางด้านฐานข้อมูลที่มีประสิทธิภาพสูง เป็นทางเลือกใหม่จากผลิตภัณฑ์ระบบจัดการฐานข้อมูลในปัจจุบัน ที่มักจะเป็นการผูกขาดของผลิตภัณฑ์เพียงไม่กี่ตัว นักพัฒนาระบบฐานข้อมูลที่เคยใช้ MySQL ต่างยอมรับในความสามารถความรวดเร็ว การรองรับจำนวนผู้ใช้ และขนาดของข้อมูลจำนวนมาก ทั้งยังสามารถใช้งานบนระบบปฏิบัติการมากมาย อาทิเช่น Unix, OS/2, Mac OS หรือ Windows ก็ตาม นอกจากนี้ MySQL ยังสามารถใช้งานร่วมกับ Web Development Platform ทั้งหลาย ไม่ว่าจะเป็น C, C++, Java, Perl, PHP, Python หรือ ASP

MySQL เป็นซอฟต์แวร์ประเภท Open Source Software สามารถดาวน์โหลด Source Code ต้นฉบับได้จากอินเทอร์เน็ต โดยไม่เสียค่าใช้จ่ายใด ๆ การแก้ไขสามารถกระทำได้ตามความต้องการ MySQL ยึดถือสิทธิบัตรตาม GPL (GNU General Public License) ซึ่งเป็นข้อกำหนดของซอฟต์แวร์ประเภทนี้ส่วนใหญ่ โดยจะเป็นการชี้แจงว่า สิ่งใดทำได้ หรือทำไม่ได้สำหรับการใช้งานในกรณีต่าง ๆ ทั้งนี้ถ้าต้องการข้อมูลเพิ่มเติม หรือรายละเอียดของ GPL สามารถหาข้อมูลได้จากเว็บไซต์

MySQL ได้รับการพัฒนาให้มีความสามารถมากยิ่งขึ้น รองรับข้อมูลจำนวนมากสามารถใช้งานหลายผู้ใช้ได้พร้อม ๆ กัน (Multi-user) มีการออกแบบให้สามารถแต่งงานออกเพื่อช่วยการทำงานให้เร็วยิ่งขึ้น รองรับข้อมูลจำนวนมาก เพื่อช่วยการทำงานเร็วยิ่งขึ้น (Multi-threaded) วิธีและการเชื่อมต่อที่ดีขึ้น การกำหนดสิทธิ์และการรักษาความปลอดภัยของข้อมูลมีความรัดกุมน่าเชื่อถือยิ่งขึ้น เครื่องมือหรือ โปรแกรมสนับสนุนทั้งของตัวเองและของผู้พัฒนาอื่นๆ มีมากยิ่งขึ้น นอกจากนี้สิ่งหนึ่งที่สำคัญคือ MySQL ได้รับการพัฒนาไปในแนวทางตามข้อกำหนดมาตรฐาน SQL ดังนั้น สามารถใช้คำสั่ง SQL ในการทำงาน MySQL ได้นักพัฒนาที่ใช้ SQL มาตรฐานอยู่แล้ว ไม่ต้องศึกษาคำสั่งเพิ่มเติม แต่อาจจะต้องเรียนรู้ถึง

ภาพแบบและข้อจำกัดบางอย่างโดยเฉพาะ ทั้งนี้ทั้งนั้น ทางทีมงานผู้พัฒนา MySQL มีเป้าหมายอย่างชัดเจนที่จะพัฒนาให้ MySQL มีความสามารถสนับสนุนตามข้อกำหนด SQL92 มากที่สุด และจะพัฒนาให้เป็นไปตามข้อกำหนด SQL99 ต่อไป

ทุกวันนี้มีการนำ MySQL ไปใช้ในระบบต่าง ๆ มากมาย ไม่ว่าจะเป็นระบบเล็ก ๆ ที่มีจำนวนตารางข้อมูลน้อย มีความสัมพันธ์ของข้อมูลในแต่ละตารางไม่ซับซ้อน เช่น ระบบฐานข้อมูลบุคคลในแผนกเล็ก ๆ ไปจนถึงระบบจัดการข้อมูลขนาดใหญ่ ที่ประกอบด้วยตารางข้อมูลมากมาย มีความสัมพันธ์ของข้อมูลในแต่ละตารางซับซ้อน เช่น ระบบสต็อกสินค้า ระบบบัญชีเงินเดือน เป็น

ต้น โดยเฉพาะอย่างยิ่งในปัจจุบัน มีการใช้ MySQL เป็น Database Server เพื่อการทำงานสำหรับ Web Database Application ในโลกของอินเทอร์เน็ตมากขึ้น

7.1 สถาปัตยกรรมของ MySQL

สถาปัตยกรรม หรือ โครงสร้างภายในของ MySQL หมายถึง การออกแบบการทำงานในลักษณะของ Client/Server นั่นเอง ซึ่งประกอบด้วยส่วนหลัก ๆ 2 ส่วน คือ ส่วนของผู้ให้บริการ (Server) และ ส่วนของผู้ใช้บริการ (Client) โดยในแต่ละส่วนจะมีโปรแกรมสำหรับการทำงานตามหน้าที่ของคณส่วนของผู้ให้บริการ หรือ Server จะเป็นส่วนที่ทำหน้าที่บริหารจัดการระบบฐานข้อมูล ซึ่งหมายถึงตัว MySQL Server นั่นเอง และเป็นที่จัดเก็บข้อมูลทั้งหมด ข้อมูลที่เก็บไว้นี้มีข้อมูลที่จำเป็นสำหรับการทำงานกับระบบฐานข้อมูล และข้อมูลที่เกิดจากการที่ผู้ใช้แต่ละคนสร้างขึ้นมาจากส่วนของผู้ให้บริการ หรือ Client ก็คือผู้ใช้นั่นเอง โดยโปรแกรมสำหรับใช้งานในส่วนนี้ได้แก่ MySQL Client, Access, Web Development Platform ต่างๆ เช่น Java, Perl, PHP, ASP เป็นต้น

7.1.1 หลักการทำงานในลักษณะ Client / Server มีดังนี้

(1) ที่ฝั่งของ Server จะมีโปรแกรมหรือระบบสำหรับจัดการฐานข้อมูลทำงานรออยู่เพื่อเตรียมหรือรอคอยการร้องขอการให้บริการจาก Client

(2) เมื่อมีการร้องขอการให้บริการเข้ามา Server จะทำการตรวจสอบตามวิธีการของตน เช่น อาจจะมีการให้ผู้ให้บริการระบุชื่อและรหัสผ่าน และสำหรับ MySQL สามารถกำหนดได้ว่า จะอนุญาตหรือปฏิเสธ Client ใดๆ ในระบบที่จะเข้าใช้บริการอีกด้วย ซึ่งจะได้แสดงรายละเอียดในเรื่องต่อไป

(3) ถ้าผ่านการตรวจสอบ Server ก็จะอนุมัติการให้บริการแก่ Client ที่ร้องขอการให้บริการนั้นๆ ต่อไปและถ้าในกรณีที่มิได้รับการอนุมัติ Server ก็จะส่งข่าวสารความผิดพลาดแจ้งกลับไป Client ที่ร้องขอการให้บริการนั้น

เครื่องคอมพิวเตอร์ที่ทำหน้าที่เป็น Client หรือ Server อาจอยู่บนเครื่องเดียวกัน หรือแยกเครื่องกันก็ได้ ทั้งนี้ขึ้นอยู่กับลักษณะการทำงาน หรือการกำหนดของผู้บริหารระบบ ตามปกติถ้าเป็นการทำงานลักษณะ Web-based มีการใช้ฐานข้อมูลขนาดไม่ใหญ่นัก ตัว MySQL Server และ Client มักจะมีอยู่บนเครื่องเดียวกัน โดยเครื่องคอมพิวเตอร์ดังกล่าวจะต้องมีทรัพยากรเพื่อการทำงาน เช่น เนื้อที่ฮาร์ดดิสก์ RAM มากพอสมควร แต่สำหรับการทำงานจริง (Real-world Application) ก็มักจะแยก Client และ Server ออกเป็นคนละเครื่องกัน และสามารถรองรับงานได้ดีมากกว่า ดังนั้น ผู้บริหารระบบ หรือผู้กำหนดนโยบายสำหรับการทำงานเครือข่าย จะต้องคำนึงถึง

เรื่องที่เกี่ยวข้องเหล่านี้ให้ดี เพื่อที่จะทำให้ระบบมีการทำงานรับการให้บริการแก่ผู้ใช้者อย่างมีประสิทธิภาพและข้อมูลมีความปลอดภัยมากที่สุด

8. ภาษา SQL

ระบบฐานข้อมูลที่นิยมใช้กันทั่วโลก มีด้วยกันหลายระบบ แต่ที่ได้รับความนิยมใช้กัน ในปัจจุบันส่วนใหญ่พัฒนาขึ้นมาจากพื้นฐานของภาษา SQL (Structure Query Language) พัฒนาโดยบริษัท IBM ปี ค.ศ.1986 American National Standards Institute (ANSI) ได้กำหนดมาตรฐาน SQL ขึ้นมาเพื่อให้ผลิตภัณฑ์ทั้งหมดเป็นไปตามมาตรฐานเดียวกัน อย่างไรก็ตามการทำเช่นนี้ทำให้เกิดปัญหาบางประการขึ้น เป็นผลให้มาตรฐาน ANSI มีข้อจำกัดอยู่บ้าง เพราะ SQL มีสองชนิดคือ ชนิดได้ตอบได้กับชนิดที่ฝังอยู่ในโปรแกรม ส่วนใหญ่แล้วทั้งสองชนิดปฏิบัติงานอย่างเดียวกันแต่นำไปใช้ต่างกัน SQL ชนิดได้ตอบได้ใช้เพื่อปฏิบัติงานกับฐานข้อมูลโดยตรงเพื่อนำเอาผลลัพธ์ไปใช้งาน ส่วน SQL แบบฝัง ในโปรแกรมประกอบด้วยคำสั่งต่าง ๆ ของ SQL ที่ใส่ในโปรแกรมที่ส่วนมากแล้วเขียนด้วยภาษาอื่น เช่น COBOL, Pascal, C/C++, Visual Basic, Delphi และ Java เป็นต้น

แม้ว่าจะมีผลิตภัณฑ์ฐานข้อมูลออกมามากมายหลายยี่ห้อ แต่ด้วยมาตรฐานภาษา SQL ที่ใช้ร่วมกัน ทำให้มีความสามารถพื้นฐานเหมือนกัน อาจแตกต่างกันได้บ้างเพราะแต่ละผลิตภัณฑ์ก็ล้วนแล้วแต่พยายามสร้างจุดแข็ง ให้กับผลิตภัณฑ์ของตนเอง จึงเป็นหน้าที่ของผู้ใช้ผลิตภัณฑ์นั้น ที่ต้องศึกษาในส่วนที่แตกต่างเพื่อที่จะได้นำมาใช้งานได้เต็มที่ประสิทธิภาพ

8.1 ภาษา SQL ตามมาตรฐาน ANSI มีคุณสมบัติดังนี้

8.1.1 โครงสร้างของภาษากลับภาษาอังกฤษ สามารถเรียกดูข้อมูลที่ระบุได้ตามความต้องการ เปลี่ยนแปลง เพิ่มเติม และลบข้อมูลออกจากระบบได้

8.1.2 มีโครงสร้างไม่แน่นอน เพียงระบุความต้องการก็สามารถใช้งานได้แล้ว

8.1.3 สามารถประมวลผลข้อมูลเป็นกลุ่มได้

8.1.4 รองรับผู้ใช้งานได้ทุกประเภท เช่น ผู้ดูแลระบบฐานข้อมูลหรือโปรแกรมเมอร์

8.2 ประเภทของคำสั่งในภาษา SQL ประกอบด้วยชุดคำสั่งหลัก ๆ 3 ประการด้วยกัน ได้แก่

8.2.1 DDL (Data Definition Language) เป็นชุดคำสั่งที่ใช้ในการนิยาม กำหนด หรือ การสร้างข้อมูลบนฐานข้อมูล เช่น คำสั่งการสร้างตาราง เป็นต้น

8.2.2 DML (Data Manipulation Language) เป็นชุดคำสั่งที่ใช้ในการประมวลผลหรือจัดการกับข้อมูลในฐานข้อมูล เช่น คำสั่ง select, insert และ update

8.2.3 DCL (Data Control Language) เป็นคำสั่งที่ใช้ในการควบคุมสิทธิ์ของผู้ใช้ในการใช้ข้อมูลรวมทั้งส่วนที่ใช้ควบคุมการใช้งานฐานข้อมูลจากผู้ใช้หลายคนพร้อมกัน เช่น คำสั่ง grant, revoke