



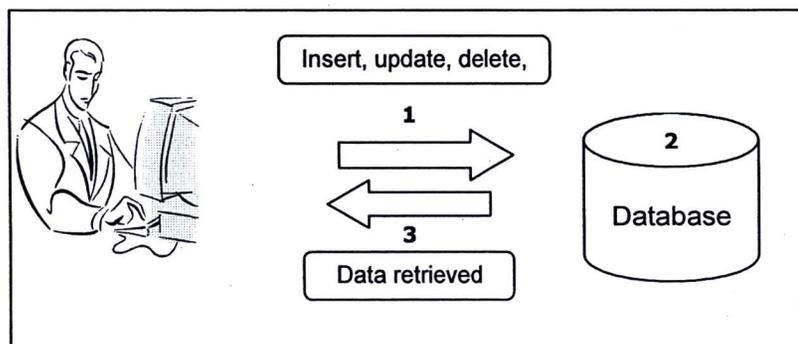
บทที่ 3 วิธีการดำเนินการวิจัย

ฐานข้อมูลแบบแอททิฟสามารถนำมาประยุกต์ใช้งานได้กับฐานข้อมูลที่สนับสนุนการทำงานของทริกเกอร์ และคำสั่งที่อยู่ในรูปแบบของแอททิฟ ซึ่งฐานข้อมูล MySQL 5.0 ก็เป็นฐานข้อมูลหนึ่งที่สนับสนุนการทำงานของทริกเกอร์ และคำสั่งที่อยู่ในรูปแบบของแอททิฟ ดังนั้นผู้พัฒนาสามารถพัฒนาเอเจนต์เพื่อให้ทำงาน เป็นส่วนติดตั้งเพิ่มเติมให้กับฐานข้อมูล แล้วสามารถทำให้ฐานข้อมูลนั้นมีคุณสมบัติเป็นฐานข้อมูลแอททิฟ มีรายละเอียดขั้นตอนและวิธีการเพื่อพัฒนาเอเจนต์ดังต่อไปนี้

1. ศึกษาโมเดลการทำงานของฐานข้อมูลรีเลชันนอล
2. ศึกษาโมเดลการทำงานของฐานข้อมูลแอททิฟ
3. การพัฒนาเอเจนต์แบบติดตั้งเพิ่มเติมให้กับฐานข้อมูลรีเลชันนอล

1. ศึกษาโมเดลการทำงานของฐานข้อมูลรีเลชันนอล

จากการศึกษาโมเดลการทำงานของฐานข้อมูลรีเลชันนอลพบว่า ฐานข้อมูลรีเลชันนอลมีการทำงานแบบแพชชีฟหรือมีการทำงานแบบตามสั่ง คือ ฐานข้อมูลจะมีการทำงานก็ต่อเมื่อผู้ใช้เขียนโปรแกรมสั่งการเช่น การเพิ่มข้อมูล การลบข้อมูลและการปรับปรุงข้อมูล ฐานข้อมูลรีเลชันนอลไม่สามารถตอบสนองการทำงานได้โดยอัตโนมัติ ซึ่งสามารถแสดงแบบจำลองการทำงานของฐานข้อมูลรีเลชันนอลในภาพที่ 15

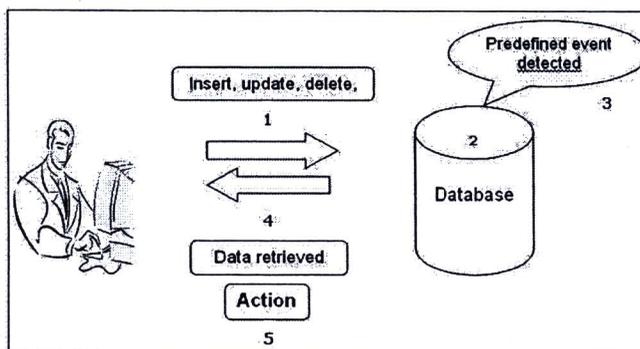


ภาพที่ 15 แบบจำลองการทำงานของฐานข้อมูลรีเลชันนอล

จากภาพที่ 15 พบว่าหากมีเหตุการณ์ใดเหตุการณ์หนึ่งเกิดขึ้นในฐานข้อมูล เช่น การเปลี่ยนแปลงของข้อมูล ปริมาณจำนวนข้อมูลลดลง หรือมีข้อมูลจำนวนหนึ่งหายไป ผู้ใช้งานไม่สามารถทราบถึงการเปลี่ยนแปลงดังกล่าวได้จนกว่าจะตรวจสอบไปยังฐานข้อมูลเพื่อดึงรายการข้อมูลต่าง ๆ ในฐานข้อมูลขึ้นมาเพื่อทำการตรวจสอบ

2. ศึกษาโมเดลการทำงานของฐานข้อมูลเอกทิฟ

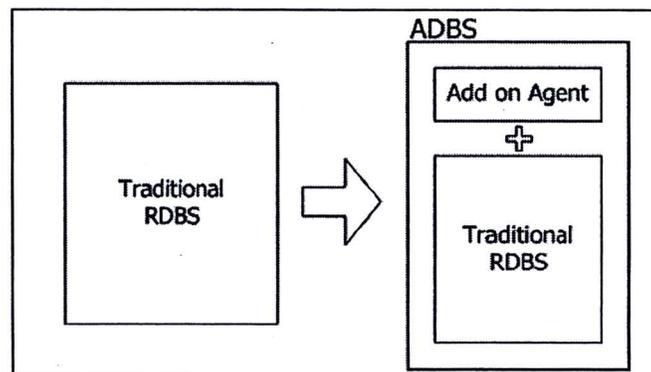
ฐานข้อมูลเอกทิฟเป็นฐานข้อมูลที่มีการตอบสนองเหตุการณ์ที่ได้กำหนดไว้ล่วงหน้าได้อย่างอัตโนมัติ การตอบสนองของฐานข้อมูลเอกทิฟนั้นจะทำตามกฎที่ได้กำหนดไว้ คุณสมบัติของฐานข้อมูลเอกทิฟนั้นมีแนวคิดมาจาก production rule ของระบบปัญญาประดิษฐ์ (Artificial Intelligent: AI) ที่แสดงในภาพของ Condition -> Action เมื่อเกิดเหตุการณ์ตรงตามเงื่อนไขที่กำหนด ระบบจะถูกกระตุ้น (triggered หรือ fired) ให้ทำงานตามกระบวนการที่กำหนดไว้ ซึ่งแนวความคิดดังกล่าวถูกนำมาประยุกต์ให้เข้ากับแนวคิด active database ในรูปของ เหตุการณ์ -> เงื่อนไข -> กระบวนการเมื่อเกิดการประมวลผลในฐานข้อมูล เป็นเหตุการณ์ที่สนใจเกิดขึ้นซึ่งทำให้สถานะในฐานข้อมูลเปลี่ยนแปลงไปตามเงื่อนไข และระบบฐานข้อมูลจะทำงานตามกระบวนการที่กำหนด กล่าวคือเมื่อผู้ใช้งานต้องการใช้งานฐานข้อมูลเอกทิฟ ผู้ใช้งานต้องทำการกำหนดเหตุการณ์ที่ต้องการตรวจจับ และเงื่อนไขในการตรวจจับ เมื่อเกิดเหตุการณ์ที่ไม่ตรงตามเงื่อนไขที่กำหนด ก็จะไม่เกิดการตรวจจับ ผู้ใช้งานก็สามารถกระทำหรือใช้งานฐานข้อมูล เช่น การเพิ่มข้อมูล การลบข้อมูล การปรับปรุงข้อมูลได้ตามปกติ แต่ถ้าหากเหตุการณ์ที่เกิดขึ้นนั้น ตรงกับเงื่อนไขในการตรวจจับ ฐานข้อมูลจะมีการตอบสนองมายังผู้ใช้งาน เช่นการแสดงความแจ้งเตือน ซึ่งการกระทำของฐานข้อมูลเอกทิฟดังกล่าวสามารถเขียนเป็นแบบจำลองการทำงานของฐานข้อมูลเอกทิฟ ดังแสดงในภาพที่16



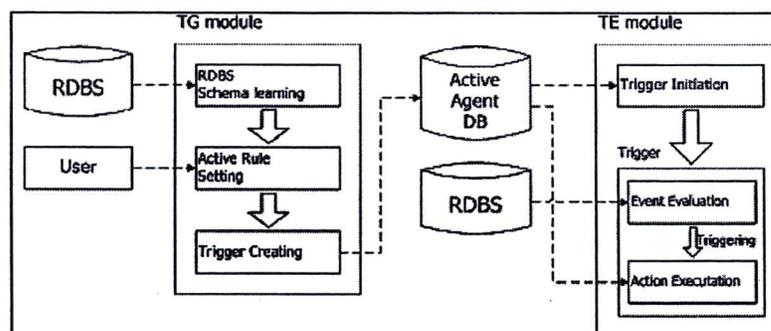
ภาพที่ 16 แบบจำลองการทำงานของฐานข้อมูลแบบเอกทิฟ

3. การพัฒนาเอเจนต์แบบติดตั้งเพิ่มเติมให้กับฐานข้อมูลรีเลชันนอล

จากการศึกษาการทำงานของฐานข้อมูลแบบรีเลชันนอลและฐานข้อมูลแบบแอททิฟ ทำให้ผู้วิจัยมีแนวคิดที่จะสร้างโปรแกรมเอเจนต์เพื่อเป็นส่วนติดตั้งเพิ่มเติมให้กับฐานข้อมูลแบบรีเลชันนอล แล้วทำให้ฐานข้อมูลแบบรีเลชันนอลมีคุณสมบัติหรือมีลักษณะเป็นฐานข้อมูลแบบแอททิฟ โดยที่ไม่มีการเปลี่ยนแปลงหรือแก้ไขโครงสร้างการทำงานของฐานข้อมูลรีเลชันนอลนั้น จากแนวคิดนี้ทำให้สามารถสร้างแบบจำลองเพื่อพัฒนา เอเจนต์เป็นส่วนติดตั้งเพิ่มเติมให้กับฐานข้อมูลรีเลชันนอล ได้ดังภาพที่ 17 และแสดงส่วนประกอบภายในของเอเจนต์ที่พัฒนาขึ้นได้ดังภาพที่ 18



ภาพที่ 17 แบบจำลองพัฒนาเอเจนต์ที่เป็นส่วนติดตั้งเพิ่มเติมให้กับฐานข้อมูลแบบรีเลชันนอล



ภาพที่ 18 ส่วนประกอบภายในเอเจนต์

จากภาพที่ 18 ภายในเอเจนต์ประกอบไปด้วย 2 โมดูล ได้แก่ ส่วนของการสร้างทริกเกอร์ (Trigger Generator (TG)) และ ส่วนของการตรวจจับ (Trigger Execution (TE))

โมดูล TG จะทำการเรียนรู้โครงสร้างของฐานข้อมูลรีเลชันนอล จากการกำหนดรายละเอียด (configuration) ของฐานข้อมูล ซึ่งผู้ดูแลระบบ (database administrator) จะเป็นผู้กำหนด ผู้ดูแลระบบจะกำหนดเงื่อนไข รวมถึง รายละเอียดต่าง ๆ ที่สำคัญ เช่น ชื่อ ทริกเกอร์

รายละเอียดของทริกเกอร์ เงื่อนไขที่ต้องการตรวจจับ ข้อความแจ้งเตือน ข้อมูลต่าง ๆ เหล่านี้ จะถูกจัดเก็บลงในฐานข้อมูลแอคทิฟเอเจนต์ ซึ่งเก็บข้อมูลแยกจากฐานข้อมูลรีเลชันนอล ทำให้การทำงานของเอเจนต์ และการทำงานของฐานข้อมูลรีเลชันนอลสามารถทำคู่ขนานกันได้ และการทำงานของเอเจนต์จะไม่ทำการเปลี่ยนแปลงข้อมูลภายในฐานข้อมูลรีเลชันนอล แต่สามารถทำให้ฐานข้อมูลรีเลชันนอล มีลักษณะเป็นฐานข้อมูลแบบแอคทิฟได้

โมดูล TE เป็น โมดูลที่กำหนดให้ทริกเกอร์เริ่มทำงาน (trigger initiation) เมื่อผู้ใช้งานเปิดการทำงานของทริกเกอร์ โปรแกรมเอเจนต์จะทำการดึงข้อมูลการตรวจจับของทริกเกอร์ที่ถูกเปิดการทำงานจากฐานข้อมูลแอคทิฟเอเจนต์ ซึ่งใน โปรแกรมเอเจนต์สามารถเปิดการทำงานทริกเกอร์ได้หลายตัวเพื่อให้ทำการตรวจจับไปพร้อมกัน แต่ทริกเกอร์ที่เปิดนั้นต้องไม่ตรวจจับตารางเดียวกัน เมื่อเกิดเหตุการณ์ขึ้นในฐานข้อมูลรีเลชันนอล และเหตุการณ์ที่เกิดขึ้นนั้นไม่ตรงกับเงื่อนไขการตรวจจับของทริกเกอร์ ทริกเกอร์ก็จะไม่ตรวจจับและไม่บันทึกผลการการตรวจจับลงในฐานข้อมูลแอคทิฟเอเจนต์ แต่หากเหตุการณ์ที่เกิดขึ้นนั้นเป็นไปตามเงื่อนไข ก็จะทำการตรวจจับและบันทึกผลการตรวจจับลงในฐานข้อมูลแอคทิฟเอเจนต์ และจะแสดงข้อความแจ้งเตือนต่อผู้ใช้งาน

3.1 การเรียนรู้โครงสร้างของฐานข้อมูลรีเลชันนอล (Relational Database Schema Learning: RDBS)

การเรียนรู้โครงสร้างของฐานข้อมูลรีเลชันนอลสามารถทำได้โดยการระบุค่าการเชื่อมต่อกับฐานข้อมูลรีเลชันนอล เช่น ชื่อฐานข้อมูล พอร์ตที่ใช้ในการเชื่อมต่อ และ รหัสผ่านเข้าฐานข้อมูล ซึ่งสามารถออกแบบหน้าจอการระบุค่าการเชื่อมต่อกับฐานข้อมูลรีเลชันนอลได้ดังภาพที่ 19



ภาพที่ 19 หน้าจอกำหนดค่าการเชื่อมต่อ

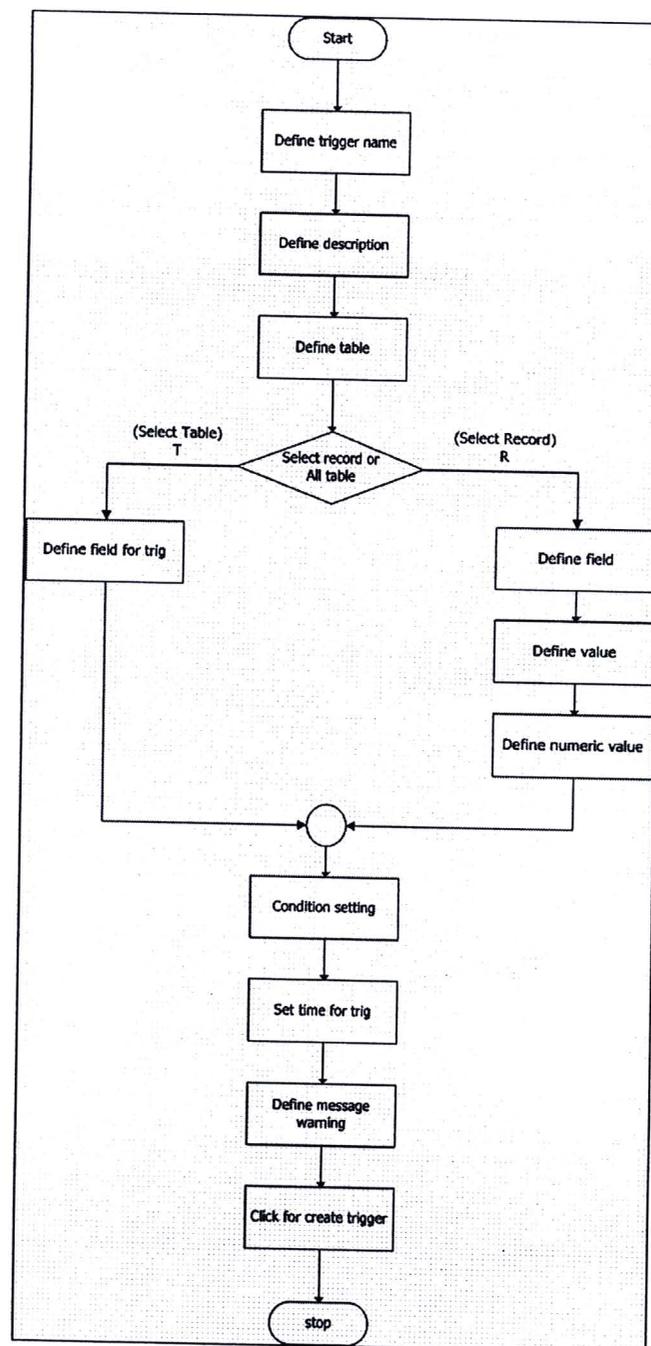
เมื่อทำการกำหนดค่าเชื่อมต่อระหว่างโปรแกรมเอเจนต์และฐานข้อมูลรีเลชันนอลแล้ว โปรแกรมเอเจนต์จะรู้จักตารางและโครงสร้างของตารางต่าง ๆ ในฐานข้อมูลนั้น ด้วยคำสั่งของ

ภาษา SQL ได้แก่ คำสั่ง show tables ซึ่งเป็นคำสั่งเรียกดูตารางต่าง ๆ ภายในฐานข้อมูล และคำสั่ง describe table ซึ่งเป็นคำสั่งที่ใช้เรียกดูโครงสร้างของที่ผู้ใช้ต้องการที่จะทำการตรวจจับ

3.2 การกำหนดค่าในการตรวจจับ (Active Rule Setting) และการสร้างทริกเกอร์ (Trigger Creating)

ส่วนของการกำหนดค่าในการตรวจจับ และการสร้างทริกเกอร์ ผู้ใช้งานจะทำการกำหนดค่าต่าง ๆ ให้กับทริกเกอร์ เช่น ชื่อทริกเกอร์ รายละเอียดของทริกเกอร์เพื่อใช้อธิบายทริกเกอร์นั้น ตารางหรือเรคคอร์ดที่ต้องการตรวจจับ เวลาในการตรวจจับ เงื่อนไขที่ใช้ในการตรวจจับ รวมถึงข้อความแจ้งเตือนเมื่อทริกเกอร์สามารถตรวจจับเหตุการณ์ที่ตรงตามเงื่อนไขที่กำหนด จากนั้นจะทำการสร้างทริกเกอร์และบันทึกข้อมูลของทริกเกอร์ลงในฐานข้อมูลแอกทิฟเอเจนต์ ซึ่งสามารถออกแบบหน้าจอการกำหนดค่าของทริกเกอร์ได้ดังภาพที่ 20 และ โฟล์ชาร์ต การสร้างทริกเกอร์ได้ดังภาพที่ 21

ภาพที่ 20 หน้าจอการกำหนดค่าให้กับทริกเกอร์



ภาพที่ 21 Flow chart การสร้างทริกเกอร์

3.3 การเริ่มต้นการทำงานของทริกเกอร์ (Trigger Initiation)

เป็นขั้นตอนการเปิดการทำงานให้กับทริกเกอร์ที่สร้างขึ้น เพื่อให้ทริกเกอร์นั้นได้เริ่มการตรวจจับตารางหรือเรคคอร์ด โดยการเปิดการทำงานแต่ละครั้ง สามารถเปิดการทำงานทริกเกอร์ได้

หลายตัวเพื่อให้ทำการตรวจจับไปพร้อมกัน แต่ทริกเกอร์ที่เปิดนั้นต้องไม่ตรวจจับตารางเดียวกัน สามารถออกแบบหน้าจอการตรวจจับได้ดังภาพที่ 22

ชื่อทริกเกอร์	รายละเอียด	ตารางที่ตรวจจับ	สถานะการทำงาน
tgm1	ตรวจสอบค่าปรับน้อย...	member	กำลังทำงาน
tgm2	ตรวจจับค่าปรับมากกว่า ...	member	หยุดทำงาน
tgc1	ตรวจจับสารน้อยกว่า 300	chem	กำลังทำงาน
tgc2	ตรวจจับสารมากกว่า 1000	chem	หยุดทำงาน
tge1	ตรวจจับอุปกรณ์เหลือ...	equipment	กำลังทำงาน
tgc3	ตรวจจับสาร 300 - 500	chem	หยุดทำงาน
tge2	ตรวจจับอุปกรณ์มากกว่า ...	equipment	หยุดทำงาน
tge3	ตรวจจับอุปกรณ์อยู่ใน...	equipment	หยุดทำงาน
tgm3	ตรวจจับค่าปรับในช่วง ...	member	หยุดทำงาน

ภาพที่ 22 หน้าจอการเปิดการทำงานของทริกเกอร์

3.4 ขั้นตอนการตรวจจับ และรายงานผลการตรวจจับ

เมื่อเกิดเหตุการณ์ขึ้นในฐานข้อมูลแพสซีฟ และเหตุการณ์ที่เกิดขึ้นนั้นไม่ตรงกับเงื่อนไขการตรวจจับของทริกเกอร์ ทริกเกอร์ก็จะไม่ตรวจจับและไม่บันทึกผลการการตรวจจับลงในฐานข้อมูลเอกทิวเอเจนต์ แต่หากเหตุการณ์ที่เกิดขึ้นนั้นเป็นไปตามเงื่อนไข ก็จะทำการตรวจจับและบันทึกผลการตรวจจับลงในฐานข้อมูลเอกทิวเอเจนต์ และแสดงข้อความแจ้งเตือนต่อผู้ใช้งาน การตรวจจับของทริกเกอร์สามารถแบ่งได้ 2 การตรวจจับดังนี้ การตรวจจับที่ 1 ตรวจจับก่อนสามารถแยกได้เป็น 2 กรณีย่อย คือกรณีที่ 1 กำหนดทริกเกอร์ตรวจจับทั้งตาราง และกรณีที่ 2 คือให้ทริกเกอร์ตรวจจับเฉพาะเรคคอร์ด การตรวจจับที่ 2 ตรวจจับหลัง สามารถแยกได้เป็น 2 กรณีย่อย คือกรณีที่ 1 กำหนดทริกเกอร์ตรวจจับทั้งตาราง และกรณีที่ 2 คือให้ทริกเกอร์ตรวจจับเฉพาะเรคคอร์ด

3.4.1 ตรวจจับก่อนดำเนินการ

เมื่อทริกเกอร์เริ่มการทำงาน ทริกเกอร์จะรอดักจับเหตุการณ์ที่เกิดขึ้นกับฐานข้อมูล เมื่อเกิดเหตุการณ์ทริกเกอร์จะทำการตัดสินใจว่าเหตุการณ์ที่เกิดขึ้นส่งผลต่อข้อมูลเฉพาะเรคคอร์ดหรือข้อมูลทั้งหมดในตาราง

กรณีที่ส่งผลเฉพาะเรคคอร์ด จะทำการตรวจสอบว่าเหตุการณ์ที่เกิดขึ้นส่งผลต่อเรคคอร์ดที่กำหนดไว้หรือไม่ หากไม่ใช่จะทำการรอเหตุการณ์อื่นต่อไป แต่หากเป็นเหตุการณ์ตามที่กำหนด จะทำการตรวจสอบเงื่อนไขสำหรับเรคคอร์ด ว่าตรงตามข้อกำหนดหรือไม่ หากไม่ตรงให้ทำการรอเหตุการณ์อื่น แต่หากตรงกับที่กำหนด จะทำการคืนค่าเดิม ป้อนข้อมูลการตรวจจับนี้ไปยังฐานข้อมูลเอเจนต์ และตรวจสอบสถานะของทริกเกอร์ เพื่อให้ทำงานต่อหรือหยุดทำงาน

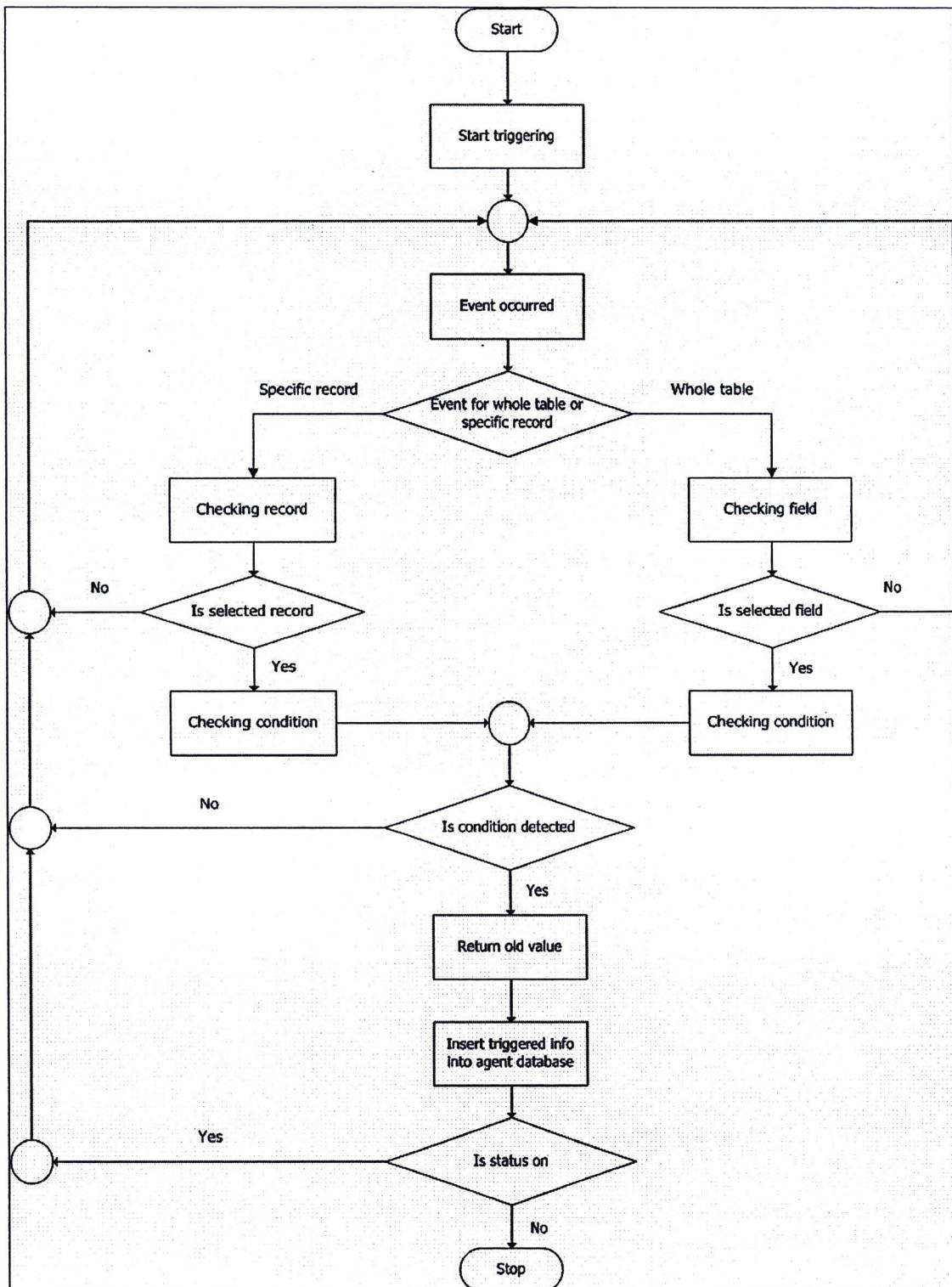
กรณีที่ส่งผลทั้งตาราง จะทำการตรวจสอบฟิลด์ที่ผู้ใช้ระบุ ว่ามีชื่อตรงตามที่กำหนดไว้หรือไม่ หากไม่ใช่ จะรอให้เกิดเหตุการณ์อื่น แต่หากมีชื่อฟิลด์ตรงตามเงื่อนไข จะทำการตรวจสอบข้อมูลทั้งหมดภายในฟิลด์ของตาราง ว่ามีข้อมูลใดบ้างที่ตรงตามเงื่อนไขตรวจสอบ หากไม่พบให้รอเหตุการณ์อื่น หากพบจะทำการคืนค่าเดิม และบันทึกข้อมูลการตรวจจับไปยังฐานข้อมูลเอเจนต์ และตรวจสอบสถานะของทริกเกอร์ เพื่อหยุด หรือทำงานต่อ ดังแสดงใน Flow chart การตรวจจับของทริกเกอร์ โดยเลือกตรวจจับก่อนในภาพที่ 23

3.4.2 ตรวจจับหลังดำเนินการ

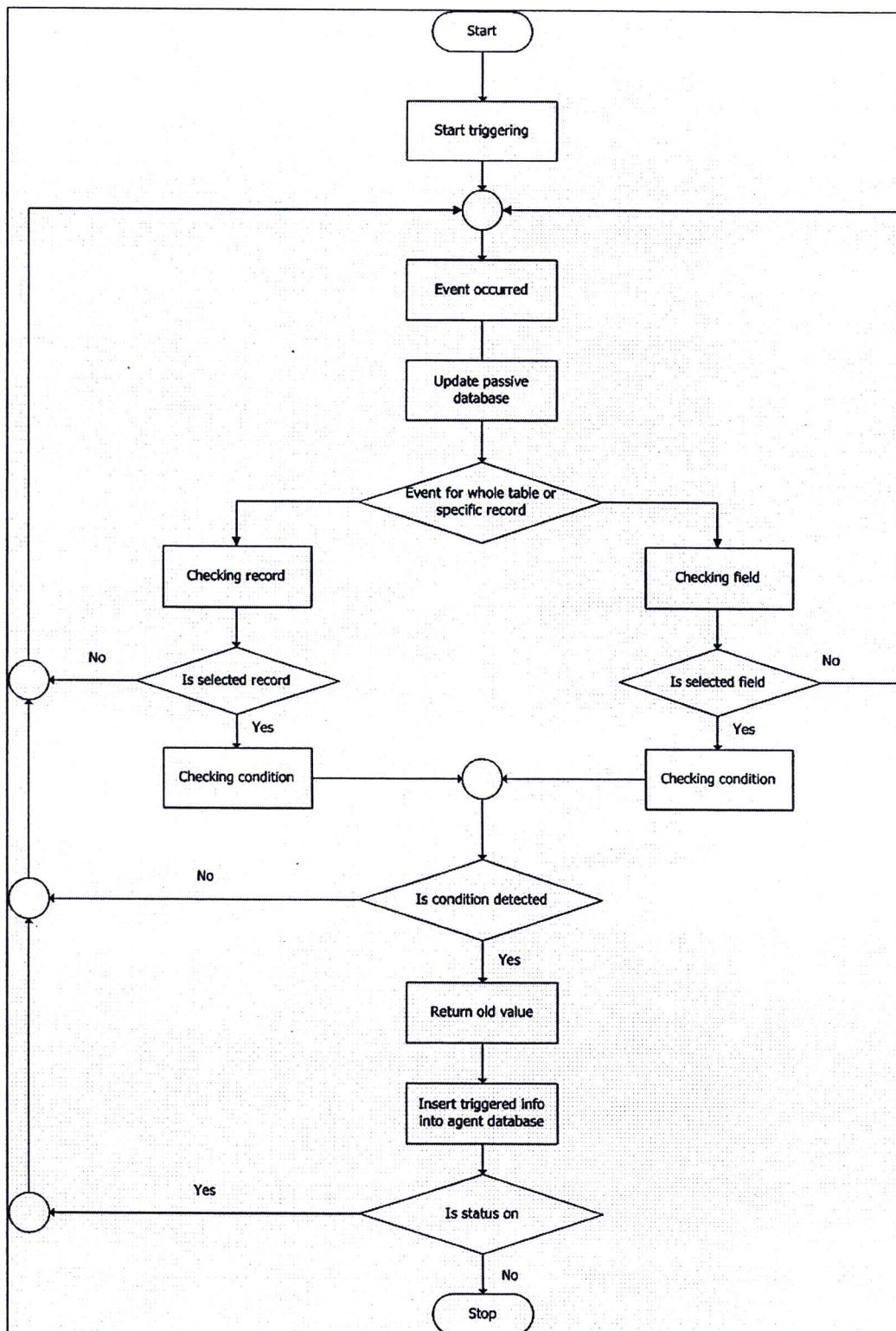
เมื่อทริกเกอร์เริ่มทำงาน ทริกเกอร์จะดักจับเหตุการณ์ที่จะเกิดขึ้นกับฐานข้อมูล เมื่อเกิดเหตุการณ์ปรับปรุงข้อมูลจะเกิดการตัดสินใจว่า การปรับปรุงนี้ มีผลเฉพาะเรคคอร์ด หรือมีผลต่อข้อมูลในตาราง

กรณีที่มีการปรับปรุงมีผลเฉพาะเรคคอร์ด ทริกเกอร์จะทำการตรวจสอบว่า เรคคอร์ดที่มีการปรับปรุงเป็นเรคคอร์ดที่ตรวจจับหรือไม่ หากไม่ใช่ให้รอตรวจสอบในการปรับปรุงข้อมูลครั้งต่อไป หากเป็นเรคคอร์ดที่ต้องการตรวจจับ จะทำการตรวจสอบเงื่อนไข ว่าข้อมูลมีการเปลี่ยนแปลงตามที่ระบุหรือไม่ หากข้อมูลไม่ได้เปลี่ยนแปลงตามที่ระบุ ให้รอตรวจสอบการปรับปรุงข้อมูลครั้งต่อไป หากข้อมูลถูกเปลี่ยนแปลงตามที่ระบุ บันทึกการตรวจจับในฐานข้อมูลเอเจนต์ และตรวจสอบสถานะของ ทริกเกอร์ เพื่อหยุด หรือทำงานต่อไป

กรณีที่มีการปรับปรุงมีผลกับข้อมูลหลายเรคคอร์ดในตาราง ทริกเกอร์จะทำการตรวจสอบว่าต้องการตรวจจับฟิลด์ใดในตาราง เมื่อตรวจพบ ทริกเกอร์จะทำการค้นหาเรคคอร์ดที่มีเงื่อนไขตรงตามที่กำหนดภายในตาราง หากไม่พบจะรอตรวจจับในการปรับปรุงข้อมูลครั้งต่อไป หากข้อมูลถูกเปลี่ยนแปลงตามที่ระบุ บันทึกการตรวจจับในฐานข้อมูลเอเจนต์ และตรวจสอบสถานะของทริกเกอร์ เพื่อหยุด หรือทำงานต่อไป ดังแสดงใน Flow chart การตรวจจับของทริกเกอร์ โดยเลือกตรวจจับหลังในภาพที่ 24

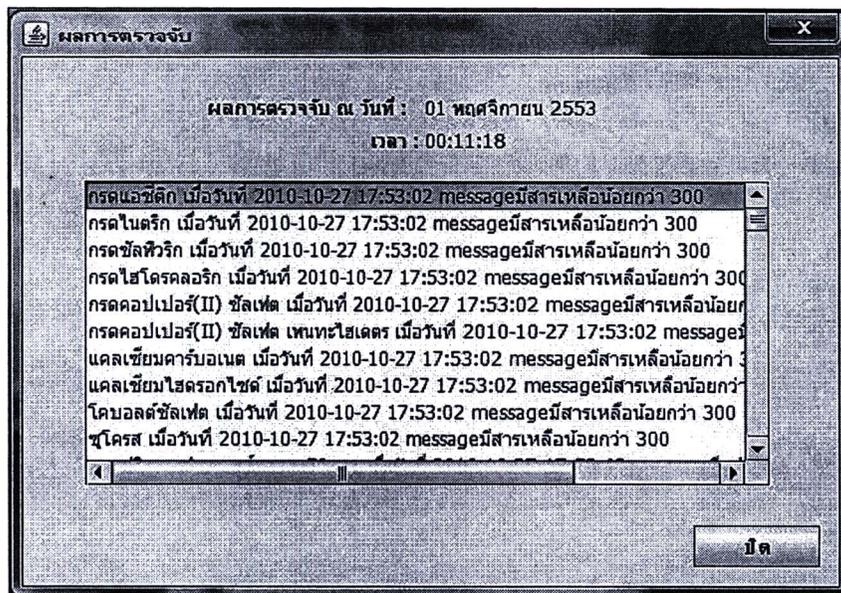


ภาพที่ 23 Flow chart การตรวจจับของทริกเกอร์ โดยเลือกตรวจจับก่อนการดำเนินการ



ภาพที่ 24 Flow chart การตรวจจับของทริกเกอร์ โดยเลือกตรวจจับหลัง

เมื่อทริกเกอร์ตรวจจับเหตุการณ์ที่เป็นไปตามเงื่อนไขแล้วจะแสดงข้อความแจ้งเตือนต่อผู้ใช้งานซึ่งข้อความแจ้งเตือน เป็นข้อความที่กำหนดคขึ้นในขั้นตอน Active rule setting หน้าจอข้อความแจ้งเตือนแสดงในภาพที่ 25



ภาพที่ 25 ข้อความแจ้งเตือนการตรวจจับ

การออกแบบรายงานผลการตรวจจับทำได้โดยการดึงข้อมูลจากฐานข้อมูลเอกทิวเอเจนต์ จากตารางที่บันทึกผลการตรวจจับของทริกเกอร์แต่ละทริกเกอร์ดังแสดงในภาพที่ 26 และแสดงรายละเอียดการตรวจจับได้ในภาพที่ 27

รายงานผลการตรวจฉบับของทริกเกอร์

รายงานการทำงานของทริกเกอร์

ข้อมูลทริกเกอร์ที่กำลังทำงาน

ชื่อทริกเกอร์	เวลาเริ่มต้นการทำงาน	ผลการตรวจฉบับ
tgcl	27/ตุลาคม/2553 17:57:00	10 รายการ
tge1	27/ตุลาคม/2553 17:56:49	5 รายการ
tgm1	27/ตุลาคม/2553 17:56:38	5 รายการ

ข้อมูลทริกเกอร์ที่รอเปิดการทำงาน

ชื่อทริกเกอร์	หยุดทำงานเมื่อเวลา	ผลการตรวจฉบับ

รีเฟรช เปิดการทำงาน ปิดการทำงาน ดูผลการตรวจฉบับ ดูข้อมูลทริกเกอร์

ภาพที่ 26 รายงานผลการตรวจฉบับของทริกเกอร์

แสดงรายละเอียดการตรวจฉบับ

แสดงรายละเอียดการตรวจฉบับ

ชื่อทริกเกอร์ : tgcl

รายละเอียดการตรวจฉบับ	เวลาที่ตรวจฉบับได้
กรดแอสซิดิก	2010-10-27 16:46:35
กรดไนตริก	2010-10-27 16:46:35
กรดซัลฟิวริก	2010-10-27 16:46:35
กรดไฮโดรคลอริก	2010-10-27 16:46:35
กรดคอปเปอร์(II) ซัลเฟต	2010-10-27 16:46:35
กรดคอปเปอร์(II) ซัลเฟต เพนตะไฮเดรต	2010-10-27 16:46:35
แคลเซียมคาร์บอเนต	2010-10-27 16:46:35
แคลเซียมไฮดรอกไซด์	2010-10-27 16:46:35
โคบอลต์ซัลเฟต	2010-10-27 16:46:35
ซูโครส	2010-10-27 16:46:35
โซเดียมคลอไรด์	2010-10-27 16:46:35
โซเดียมคาร์บอเนต	2010-10-27 16:46:35
โซเดียมคาร์บอเนตเดคะไฮเดรต	2010-10-27 16:46:35
โซเดียมไฮดรอกไซด์	2010-10-27 16:46:35

รีเฟรช ปิด

ภาพที่ 27 รายละเอียดการตรวจฉบับ



3.5 การออกแบบตารางฐานข้อมูลของเอเจนต์

การจัดเก็บข้อมูลของเอเจนต์บนฐานข้อมูล MySQL 5.0 ประกอบด้วยตารางเก็บข้อมูลจำนวน 3 ตาราง ได้แก่ตาราง tg_setting, t_result และ t_trigger

3.5.1 ตาราง tg_setting เป็นตารางที่เก็บข้อมูลต่าง ๆ ในการสร้างทริกเกอร์ เช่น ชื่อทริกเกอร์ รายละเอียดของทริกเกอร์ เงื่อนไขในการตรวจจับ ข้อความแสดงเมื่อตรวจพบเป็นต้น โดยตารางที่ 3 แสดงข้อมูลต่างรวมทั้งชื่อฟิลด์ที่ใช้เก็บคุณสมบัติของทริกเกอร์

ตารางที่ 3 tg_setting

ฟิลด์	ชนิด	คีย์	ว่างเปล่า	คำอธิบายข้อมูลที่เก็บ
tg_name	varchar(100)	คีย์หลัก	ไม่	ชื่อทริกเกอร์
tg_detail	varchar(200)		ไม่	รายละเอียดทริกเกอร์
tg_tbname	varchar(50)		ไม่	ชื่อตารางที่ตรวจจับ
tg_affect	varchar(50)		ไม่	ค่าใดค่าหนึ่งระหว่าง มีผลทั้งตาราง หรือ มีผลเฉพาะเรคคอร์ด
tg_fvalue	varchar(100)		ไม่	ข้อมูลฟิลด์ (หากเลือกมีผลเฉพาะเรคคอร์ด)
tg_value	varchar(100)		ไม่	ข้อมูลค่าในฟิลด์ (หากเลือกมีผลเฉพาะเรคคอร์ด)
tg_integer	varchar(100)		ไม่	ข้อมูลค่าในฟิลด์ (หากเลือกมีผลเฉพาะเรคคอร์ด) ที่เป็นตัวเลข
tg_field	varchar(30)		ไม่	ข้อมูลฟิลด์ (หากเลือกมีผลทั้งตาราง)
tg_condition	varchar(100)		ไม่	ข้อมูลเงื่อนไข
tg_at	varchar(50)		ไม่	ค่าใดค่าหนึ่งระหว่าง ตรวจสอบก่อน หรือ ตรวจสอบหลัง
tg_msg	varchar(100)		ไม่	ข้อความแจ้งเตือน
dbname	varchar(50)		ไม่	ชื่อฐานข้อมูล

3.5.2 ตาราง t_result เป็นตารางที่เก็บข้อมูลที่ทริกเกอร์สามารถทริกได้ เช่น ชื่อทริกเกอร์ที่ตรวจจับ ผลการตรวจจับ และเวลาในการตรวจจับ เป็นต้น ตารางที่ 4 แสดงข้อมูลต่าง ๆ รวมทั้งชื่อฟิลด์ที่ใช้เก็บผลของการตรวจจับ

3.5.3 ตาราง t_trigger เป็นตารางที่เก็บข้อมูลการเปิดการทำงานของทริกเกอร์ และการหยุดการทำงานของทริกเกอร์ ตารางที่ 5 แสดงข้อมูลต่าง ๆ รวมทั้งชื่อฟิลด์ที่ใช้เก็บข้อมูลการเปิดปิด

ตารางที่ 4 t_result

ฟิลด์	ชนิด	คีย์	ว่างเปล่า	คำอธิบายข้อมูลที่เก็บ
numA	int(5)	คีย์หลัก	ไม่	ลำดับ
tname	varchar(100)	คีย์นอก	ไม่	ชื่อทริกเกอร์
result	varchar(200)		ไม่	ผลการตรวจจับ
timetrig	varchar(50)		ไม่	เวลาในการตรวจจับ
refcondition	varchar(20)		ไม่	เงื่อนไขที่ใช้ในการตรวจจับ

ตารางที่ 5 t_trigger

ฟิลด์	ชนิด	คีย์	ว่างเปล่า	คำอธิบายข้อมูลที่เก็บ
numT	int(5)	คีย์หลัก	ไม่	ลำดับ
tname	varchar(50)	คีย์นอก	ไม่	ชื่อทริกเกอร์
detail	varchar(100)		ไม่	รายละเอียดของทริกเกอร์
affect	varchar(20)		ไม่	ชื่อตารางที่ทำการตรวจจับ
event	varchar(20)		ไม่	ตรวจจับก่อนหรือหลัง
condition	varchar(20)		ไม่	ข้อมูลเงื่อนไขในการตรวจจับ
trigg	varchar(100)		ไม่	ตรวจจับทั้งตารางหรือตรวจจับเฉพาะเรคคอร์ด
action	varchar(50)		ไม่	ข้อความแจ้งเตือน
status	varchar(20)		ไม่	สถานะ (เปิดการทำงาน หรือ หยุดการทำงาน)
timeStr	varchar(50)		ไม่	เวลาเปิดการทำงาน
timeStp	varchar(50)		ไม่	เวลาปิดการทำงาน