

ภาษาเชิงวัตถุ

1. Object-Oriented Languages

ในการแสดงให้เห็นถึงแนวทางเชิงวัตถุ หลักการต่าง ๆ จะถูกนำไปใช้ในการเขียนโปรแกรมซึ่งสามารถทำได้วิธีใดวิธีหนึ่งในนี้

นำหลักการเชิงวัตถุไปใช้เป็นเครื่องมือในการออกแบบ เพื่อดำเนินการในขั้นต่อไป ตัวอย่างเช่นการออกแบบฐานข้อมูลเชิงสัมพันธ์ เริ่มจากการออกแบบ E-R โดยแกรม จากนั้นก็แปลงไปเป็นรีเลชัน

หลักการเชิงวัตถุถูกนำไปรวมในภาษาใดภาษาหนึ่ง เพื่อใช้ในการจัดการฐานข้อมูล ด้วยวิธีการนี้ก็มีอยู่หลาย ภาษาที่สามารถรวมหลักการเชิงวัตถุได้ ดังนี้

เพิ่มความสามารถเชิงวัตถุเข้าไปใน ภาษา SQL เพื่อให้สามารถจัดการกับชนิดข้อมูลที่ซับซ้อนได้ ซึ่งจะเรียกว่า Object-relational system

นำภาษาโปรแกรมที่มีลักษณะเชิงวัตถุอยู่แล้วมาเพิ่มความสามารถทางด้านฐานข้อมูลเข้าไป ซึ่งจะถูกรเรียกว่า persistent programming languages

2. Persistent Programming Languages

Persistent data: คือข้อมูลที่ยังคงอยู่แม้ว่าโปรแกรมที่สร้างข้อมูลนั้นขึ้นมาได้ถูกปิดไปแล้วก็ตาม

Persistent programming language คือโปรแกรมภาษาที่เพิ่มส่วนสำหรับจัดการ persistent data ซึ่งแตกต่างจากภาษาที่รวม(embedded)เอา SQL เข้าไว้ในตัวภาษาน้อย 2 อย่างคือ

1. ในภาษาแบบ embedded ชนิดข้อมูลของ host language จะแตกต่างชนิดข้อมูลของภาษาจัดการข้อมูล ดังนั้นจึงเป็นหน้าที่ของนักเขียนโปรแกรมที่จะทำการแปลงชนิดข้อมูลระหว่าง host language และภาษาจัดการข้อมูล ซึ่งมีอุปสรรคหลายอย่างคือ

โปรแกรมที่ใช้ในการแปลงระหว่างอ็อบเจ็กต์และทูเปิล จะไม่ได้มีการดำเนินการในเชิงวัตถุ ซึ่งอาจจะก่อให้เกิดความผิดพลาดขึ้นได้

การแปลงข้อมูลระหว่างข้อมูลที่อยู่ในรูปแบบเชิงวัตถุกับข้อมูลที่อยู่ในรูปแบบเชิงสัมพันธ์ จะต้องเขียนโปรแกรมเป็นจำนวนมาก

ในทางกลับกัน persistent programming language และภาษาสืบค้นข้อมูลถูกรวมเข้าด้วยกัน กับ host language และทั้งสองภาษาต่างก็มีการใช้งานชนิดข้อมูลชนิดเดียวกัน ดังนั้นอ็อบเจ็กต์สามารถถูกสร้างและถูกจัดเก็บในฐานข้อมูล โดยไม่ต้องมีการเปลี่ยนแปลงรูปแบบใด ๆ ทั้งสิ้น

2. นักเขียนโปรแกรมใช้งานลักษณะ embedded SQL เพื่อไปดึงข้อมูลจากฐานข้อมูลเข้าสู่หน่วยความจำ ถ้ามีการปรับปรุงข้อมูลก็จะเขียนคำสั่งเพื่อส่งให้นำข้อมูลบันทึกกลับไปสู่ฐานข้อมูล

ส่วนใน persistent programming language นั้น นักเขียนโปรแกรมสามารถจัดการกับ persistent data ได้ โดยไม่ต้องมีการเขียนโปรแกรมเพื่อดึงข้อมูลหรือบันทึกข้อมูลกลับไปสู่ฐานข้อมูล

2.1 Persistence of Objects

ภาษาโปรแกรมเชิงวัตถุ สนับสนุนหลักการของอ็อบเจกต์ทั้งการนิยามอ็อบเจกต์ การสร้างอ็อบเจกต์ แต่อย่างไรก็ตามอ็อบเจกต์เหล่านี้ก็เกิดขึ้นเพียงชั่วคราวเท่านั้น เมื่อปิดโปรแกรม อ็อบเจกต์เหล่านั้นก็จะหายไป ดังนั้นขั้นตอนแรกคือต้องหาวิธีทำให้อ็อบเจกต์ยังคงอยู่ ไม่หายไปเมื่อปิดโปรแกรม ซึ่งก็มีอยู่หลายวิธีคือ

Persistence by class เป็นวิธีที่ง่ายที่สุดแต่ไม่เหมาะที่จะนำมาใช้งาน วิธีการนี้เป็นการประกาศว่าคลาสสร้างสร้างขึ้นเป็นแบบ persistent นั่นคืออ็อบเจกต์ทุกอ็อบเจกต์ของคลาสจะเป็น persistent อ็อบเจกต์ไปโดยปริยาย ซึ่งวิธีการนี้ไม่มีความยืดหยุ่น เนื่องจากมันถูกใช้ทั้งแบบชั่วคราวและ persistent ในคลาสเพียงคลาสเดียว ในระบบจัดการฐานข้อมูลเชิงวัตถุ การประกาศคลาสให้เป็นแบบ persistent จะถูกมองว่าเป็นคลาสที่สามารถทำให้อ็อบเจกต์ persist ได้ แทนที่จะมองว่าเป็นคลาสที่ persist

Persistence by creation วิธีการนี้เป็นการกำหนดรูปแบบในการสร้าง persistent อ็อบเจกต์ โดยการเพิ่มรูปแบบสำหรับสร้างอ็อบเจกต์ชั่วคราว ดังนั้นอ็อบเจกต์ที่ถูกสร้างขึ้นจะเป็นแบบ persist หรือแบบชั่วคราว ก็ขึ้นอยู่กับตอนที่สร้างอ็อบเจกต์ว่าจะสร้างอย่างไร

Persistence by marking วิธีนี้จะทำการกำหนดให้อ็อบเจกต์เป็น persistent หลังจากที่ได้ถูกสร้างขึ้นมา ทุก ๆ อ็อบเจกต์ถูกสร้างขึ้นมาแบบชั่วคราว แต่ถ้าอ็อบเจกต์ไหนต้องการที่จะให้คงอยู่หลังจากจบโปรแกรม ก็จะทำทำการกำหนดก่อนที่จะจบโปรแกรม

Persistence by reference อ็อบเจกต์หนึ่งหรือหลาย ๆ อ็อบเจกต์จะถูกกำหนดให้ persist ในลักษณะเป็น root ของอ็อบเจกต์ ดังนั้นอ็อบเจกต์ทุกตัวที่ถูกอ้างถึงจาก root ก็จะเป็นแบบ persistent

2.2 Object Identity and Pointers

เมื่อ persistent object ถูกสร้าง มันจะถูกกำหนด persistent object identifier

แนวคิดของ object identifier มีลักษณะคล้าย กับ pointer แนวทางง่าย ๆ ในการสร้าง build-in identity คือใช้ pointer ชี้ไปที่ ตำแหน่งทางกายภาพของแหล่งเก็บข้อมูล โดยเฉพาะภาษาเชิงวัตถุอย่าง C++ มีการสร้างความสัมพันธ์ระหว่างอ็อบเจกต์กับตำแหน่งทางกายภาพในแหล่งเก็บข้อมูล อาจมีการเปลี่ยนแปลงได้ตลอดเวลา อย่างไรก็ตาม ได้มีการแบ่งระดับความความถาวรไว้ดังนี้

intraprocedure: Identity จะคงอยู่ในขณะที่กำลังทำคำสั่งของหนึ่งกระบวนการ(procedure)เช่น ตัวแปรชนิด local ภายในกระบวนการ

intraprogram: Identity จะคงอยู่ในขณะที่กำลังทำคำสั่งของหนึ่งโปรแกรมหรือคิวรี เช่นตัวแปรชนิด global

interprogram: Identity จะคงอยู่จากโปรแกรมหนึ่งไปสู่อีกโปรแกรมหนึ่ง เช่น pointers ไปยังข้อมูลในแฟ้มบนดิสก์ แต่มันสามารถเปลี่ยนแปลงได้ขึ้นอยู่กับวิธีการจัดเก็บข้อมูลในแฟ้มถ้ามีการเปลี่ยนแปลงระบบแฟ้ม

persistent: Identity จะคงอยู่ไม่เฉพาะตอนที่โปรแกรมกำลังทำงานเท่านั้น แต่จะเกี่ยวข้องกับการปรับเปลี่ยนโครงสร้างของข้อมูลด้วย ซึ่งการคงอยู่ในลักษณะนี้จะถูกใช้ในระบบเชิงวัตถุ

2.3 Storage and Access of Persistent Objects

อ็อบเจ็กต์ถูกจัดเก็บในฐานข้อมูลอย่างไร ข้อมูลของแต่ละอ็อบเจ็กต์ก็จะถูกจัดเก็บแยกเฉพาะไปแต่ละอ็อบเจ็กต์ และ ส่วนของโปรแกรมที่ใช้เขียนเมธอดก็ควรถูกจัดเก็บไว้ในส่วนของ database schema รวมไปถึงการกำหนดชนิด ข้อมูลต่าง ๆ ของคลาส อย่างไรก็ตามส่วนมากจะมีการจัดเก็บไว้ในไฟล์แยกออกมาจากฐานข้อมูล

การค้นหาอ็อบเจ็กต์ในฐานข้อมูลทำได้หลายวิธีคือ

บอกชื่อของอ็อบเจ็กต์ เหมาะกับจำนวนอ็อบเจ็กต์น้อย ๆ

ใช้ object identifiers หรือ persistent pointers ไปที่อ็อบเจ็กต์ ซึ่งสามารถถูกจัดเก็บไว้ภายนอกได้ ไม่เหมือนกับ การบอกชื่อ ตัวชี้เหล่านี้ไม่ต้องมีการทำอะไรอีก มันสามารถที่จะชี้ไปยังตำแหน่งที่เก็บข้อมูลในฐานข้อมูลได้ทันที เก็บคอลเลกชันของอ็อบเจ็กต์ และยอมให้โปรแกรมทำการค้นหาอ็อบเจ็กต์ที่ต้องการในคอลเลกชันได้ คอลเลกชัน ของอ็อบเจ็กต์สามารถมองเป็น collection type ได้ ประกอบไปด้วย sets multiset list และอื่น ๆ และมีคลาส พิเศษที่เรียกว่า class extent ซึ่งเป็นคอลเลกชันของอ็อบเจ็กต์ทุกตัวของคลาส ซึ่งเมื่อไรก็ตามที่อ็อบเจ็กต์ถูกสร้าง อ็อบเจ็กต์นั้นก็จะถูกเพิ่มเข้าไปใน class extent ทันที และ เมื่อไรที่อ็อบเจ็กต์ถูกลบ อ็อบเจ็กต์นั้นก็จะถูกลบออกจาก class extent เช่นกัน

ระบบฐานข้อมูลเชิงวัตถุแทบทุกระบบสนับสนุนการเข้าถึง persistent objects ทั้งสามวิธี โดยที่อ็อบเจ็กต์ทุกตัว ต้องมี object identifiers การกำหนดชื่อให้กับ class และ คอลเลกชันของอ็อบเจ็กต์อื่น ๆ แต่อ็อบเจ็กต์เกือบ ทั้งหมดไม่ได้ถูกกำหนดชื่อ ดังนั้นเราจะใช้ class extents เพื่อจัดการกับคลาสทุกคลาสที่มี persistent objects

ที่มา <http://www.srisangworn.go.th/home/databaselearnx/ms1t2-13.htm>