บทที่ 3

MiniRDBMS

3.1 โครงสร้างระบบ

ระบบการจัดการฐานข้อมูลเชิงสัมพันธ์ MiniRDBM เป็นระบบจัดการฐานข้อมูลเชิงสัมพันธ์ขนาดเล็ก พัฒนาบนระบบปฏิบัติการ Linux การดำเนินงานประกอบด้วยสองส่วนหลัก ส่วนแรกเป็นส่วนของการกำหนดโครงสร้างฐานข้อมูลพร้อมเงื่อนไขบังคับอย่างง่าย โดยผ่านการใช้ภาษาฐานข้อมูลที่เรียกว่า DDL ส่วนที่สองเป็นส่วนของการดำเนินงานพื้นฐานกับข้อมูลในฐานข้อมูลเพื่อให้ผู้ใช้สามารถติดต่อกับฐานข้อมูลในการนำเข้าข้อมูล ลบข้อมูล และปรับปรุงแก้ไขข้อมูลเก่าที่มีอยู่ในฐานข้อมูลได้ โดยการใช้ภาษาฐานข้อมูลที่เรียกว่า DML ข้อมูลทุกอย่างในระบบจะถูกจัดเก็บไว้ในแฟ้มข้อมูลแบบไม่มีการเรียงลำดับใด ๆ ที่เรียกว่า Heap File มีการจัดทำปทานุกรมข้อมูลของระบบในรูปแบบฐานข้อมูลเชิงสัมพันธ์ และมีการจัดเตรียมเนื้อที่ขนาดเล็กจำนวนหนึ่งในหน่วยความจำหลักที่เรียกว่าบัฟเฟอร์ (Buffer) สำหรับจัดเก็บข้อมูลชั่วคราวที่อ่านมาจากฐานข้อมูลเพื่อใช้งานและเพื่อบันทึกลงฐานข้อมูลเมื่อดำเนินงานเสร็จเรียบร้อย

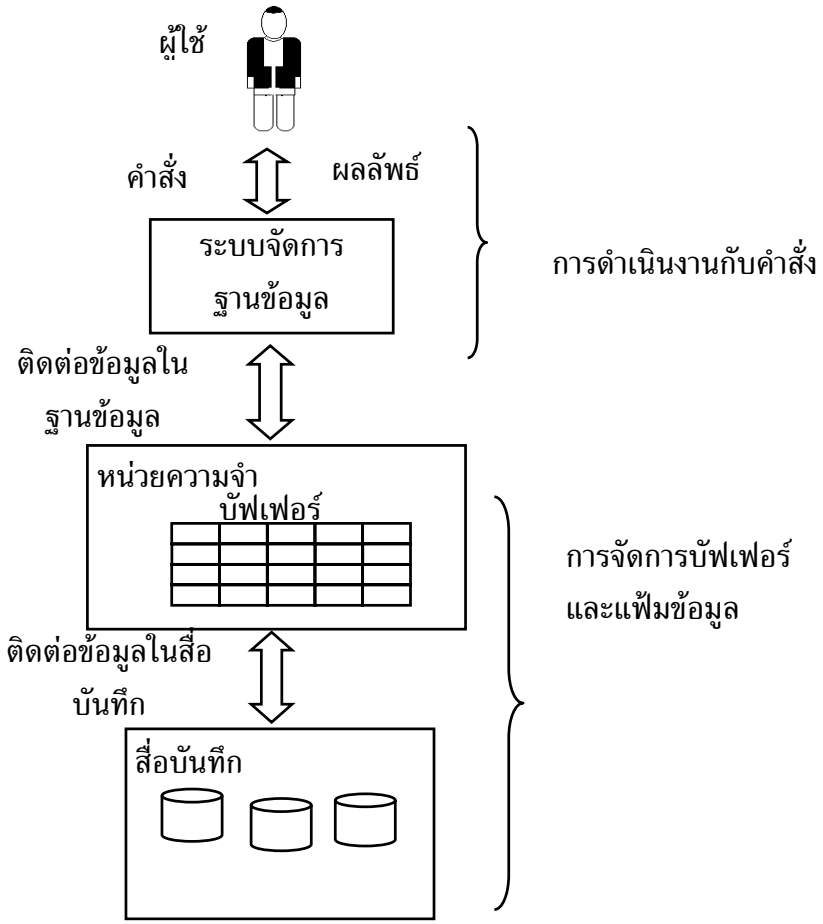
การดำเนินงานของ MiniRDBMS เมื่อมีผู้ใช้ติดต่อใช้ระบบแสดงได้ดังภาพประกอบ 3-1 โดยผู้ใช้ติดต่อใช้งาน MiniRDBMS ด้วยการป้อนแถวคำสั่งและ MiniRDBMS ดำเนินงานกับคำสั่งโดยใช้ปทานุกรมข้อมูลในการตรวจสอบความถูกต้องของข้อมูลที่ใช้ จากนั้นติดต่อกับข้อมูลในฐานข้อมูลผ่านระบบจัดการบัฟเฟอร์และระบบจัดการแฟ้มข้อมูลตามลำดับ

โครงสร้างการดำเนินงานในภาพรวมของระบบแบ่งการทำงานออกเป็น 3 ส่วนใหญ่ ๆ ดังแสดงในภาพประกอบ 3-2

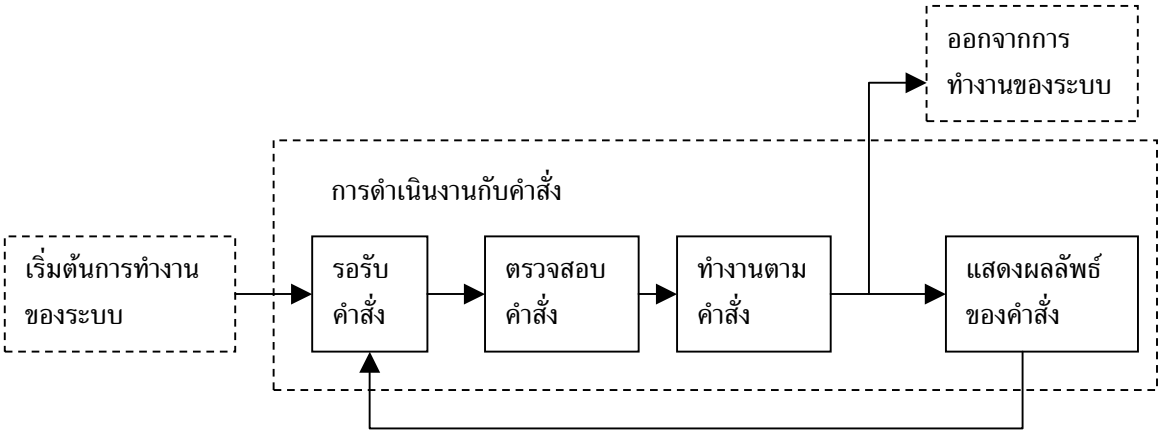
1. ส่วนเริ่มต้นการทำงานทั้งหมดของระบบ (Load System) ในส่วนนี้จะจัดเนื้อที่ในหน่วยความจำเพื่อเป็นบัฟเฟอร์พักข้อมูลระหว่างการประมวล กำหนดค่าเริ่มต้นต่าง ๆ และเปิดแฟ้มปทานุกรมข้อมูลเพื่อเตรียมพร้อมสำหรับการทำงานของระบบ

2. ส่วนการดำเนินงานกับคำสั่ง ๆทั้งหมดของระบบ (Process User Command) ในส่วนนี้จะเริ่มต้นตั้งแต่รับคำสั่งจากผู้ใช้ ตรวจสอบคำสั่ง ทำงานตามคำสั่ง และแสดงผลลัพธ์ของคำสั่ง เพื่อให้ผู้ใช้สามารถใช้ฐานข้อมูลในระบบได้

3. ส่วนของการออกจากการทำงานของระบบ (Exit System) โดยจะปิดแฟ้มปทานุกรมข้อมูลและฐานข้อมูลที่กำลังใช้งานอยู่ โดยถ้าข้อมูลใดในบัฟเฟอร์มีการเปลี่ยนแปลงจะบันทึกกลับลงหน่วยความจำสำรอง และยกเลิกการจองเนื้อที่ของบัฟเฟอร์ในหน่วยความจำหลักก่อนจะหยุดการทำงานของระบบ



ภาพประกอบ 3-1 การทำงานของระบบเมื่อมีคำสั่งจากผู้ใช้ [เบญจมาศ, 2545]



ภาพประกอบ 3-2 การดำเนินงานในภาพรวมของระบบ [เบญจมาศ, 2545]

3.2 การจัดการแฟ้มข้อมูล

ข้อมูลทั้งหมดในระบบจัดการฐานข้อมูล MiniRDBMS ทั้งปทานุกรมข้อมูลและข้อมูลของฐานข้อมูลที่ผู้ใช้กำหนดและสร้างขึ้นจะถูกเก็บเป็นรีเลชั่นไว้ในแฟ้มข้อมูลที่ระบบสร้างขึ้น เรียกว่า DBFILE โดยแต่ละแฟ้มข้อมูลจะใช้เนื้อที่บนความจำสำรองหรือดิสก์ (Disk) ขนาดคงที่จำนวนหนึ่งซึ่งแบ่งออกเป็นส่วน ๆ ที่มีขนาดเท่ากัน เรียกว่า บล็อก (Block) ซึ่งเป็นหน่วยข้อมูลสำหรับการจัดสรรเนื้อที่บนดิสก์และการโอนย้ายข้อมูลระหว่างดิสก์กับหน่วยความจำหลัก

3.2.1 ประเภทของข้อมูลในแฟ้มข้อมูล

แฟ้มข้อมูลที่จัดเก็บในระบบเมื่อแบ่งตามประเภทของข้อมูลที่จัดเก็บในแฟ้มข้อมูลจะแบ่งได้สองประเภท ได้แก่ แฟ้มข้อมูลปทานุกรม และแฟ้มข้อมูลของฐานข้อมูล ภาพประกอบ 3-3 แสดงแฟ้มข้อมูลแต่ละประเภทในระบบ

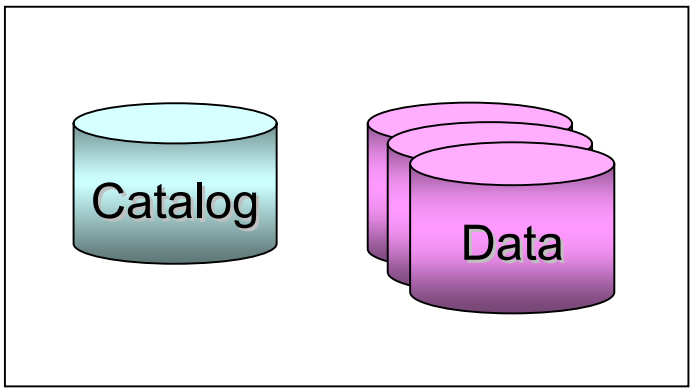
แฟ้มข้อมูลปทานุกรม

แฟ้มข้อมูลปทานุกรมเป็นแฟ้มข้อมูลสำหรับจัดเก็บรายละเอียดต่าง ๆ เกี่ยวกับปทานุกรมข้อมูลเรียกว่า Catalog File ในระบบจะมีแฟ้มข้อมูลปทานุกรมเพียงแฟ้มข้อมูลเดียวเท่านั้น ข้อมูลที่จัดเก็บในแฟ้มข้อมูลปทานุกรมประกอบด้วยข้อมูลต่อไปนี้

- นิยามฐานข้อมูลเป็นรายละเอียดของแต่ละฐานข้อมูลทั้งหมดในระบบ
- นิยามรีเลชันเป็นรายละเอียดของรีเลชันทั้งหมด
- นิยามแอตทริบิวเป็นรายละเอียดของแอตทริบิวทั้งหมดในระบบ
- เงื่อนไขบังคับ (Constraint) เป็นรายละเอียดของเงื่อนไขบังคับทั้งหมดในระบบที่กำหนดให้กับแอตทริบิวของรีเลชั่น ได้แก่ การกำหนดคีย์หลัก คีย์รอง และคีย์นอก (Foreign Key) ซึ่งการกำหนดคีย์นอกจะเป็นการกำหนดเงื่อนไขบังคับการอ้างอิงค่าแอตทริบิวของคีย์นอกจากรีเลชันที่คีย์นอกอยู่ถึงค่าแอตทริบิวเดียวกันนี้แต่เป็นคีย์หลักในอีกรีเลชั่นหนึ่งในฐานข้อมูลเดียวกัน (Referential Integrity Constraint) โดยการอ้างถึงนี้ต้องอ้างถึงค่าที่มีอยู่แล้ว
- ข้อมูลดัชนีสำหรับเก็บค่าคีย์ที่ผู้ใช้กำหนดขึ้นในแต่ละรีเลชั่น

แฟ้มข้อมูลของฐานข้อมูล

แฟ้มข้อมูลของฐานข้อมูล เป็นแฟ้มข้อมูลประเภทที่ใช้สำหรับจัดเก็บข้อมูลทั้งหมดในฐานข้อมูลหนึ่ง ๆ เรียกว่า Data File ในระบบมีแฟ้มข้อมูลของฐานข้อมูลได้มากกว่าหนึ่งแฟ้มข้อมูลขึ้นอยู่กับจำนวนฐานข้อมูลที่ผู้ใช้สร้างขึ้นในระบบ โดยทุกรีเลชันของฐานข้อมูลเดียวกันจะถูกจัดเก็บไว้ในแฟ้มข้อมูลเดียวกัน



ภาพประกอบ 3-3 ประเภทแฟ้มข้อมูลในระบบ [เบญจมาศ, 2545]

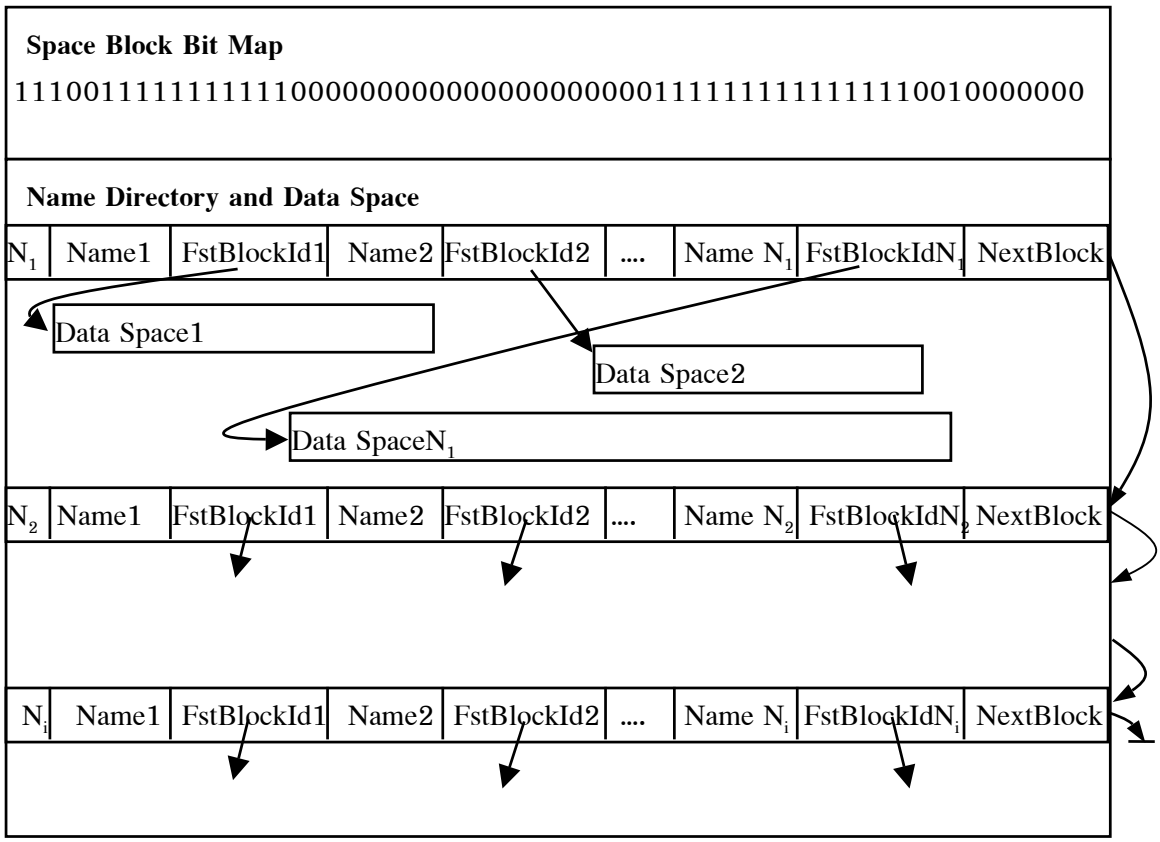
3.2.2 โครงสร้างการจัดเก็บข้อมูลของแฟ้มข้อมูลในระบบ

แต่ละแฟ้มข้อมูลของระบบถูกออกแบบให้สามารถจัดเก็บข้อมูลของรีเลชันได้มากกว่าหนึ่งรีเลชัน โดยที่ข้อมูลของรีเลชันเดียวกันจะต้องถูกจัดเก็บในแฟ้มข้อมูลเดียวกัน ข้อมูลที่ถูกจัดเก็บในแฟ้มข้อมูล นอกจากจะเป็นข้อมูลจริงของรีเลชันต่าง ๆ แล้ว จะจัดเก็บข้อมูลส่วนอื่นด้วยเพื่อให้สามารถใช้และเข้าถึงข้อมูลในแต่ละรีเลชันได้อย่างมีประสิทธิภาพ ภาพประกอบ 3-4 แสดงโครงสร้างการจัดเก็บข้อมูลของแฟ้มข้อมูลในระบบ ซึ่งข้อมูลที่จัดเก็บในแฟ้มข้อมูลถูกแบ่งออกเป็นสามส่วน ได้แก่ สถานะของบล็อก รายชื่อรีเลชันและข้อมูลจริงของรีเลชันต่างๆ ที่ถูกจัดเก็บในแฟ้มข้อมูล

สถานะของบล็อก

ข้อมูลส่วนแรกในแฟ้มข้อมูลจะเป็นส่วนที่บอกสถานะของบล็อกทั้งหมดในแฟ้มข้อมูล เรียกว่า Space Block Bit Map จัดเก็บในรูปแบบลำดับบิตแทนสถานะของบล็อกทั้งหมดในแฟ้มข้อมูล โดยหนึ่งบิตจะระบุสถานะของบล็อกข้อมูลหนึ่งบล็อกที่สัมพันธ์กันกับตำแหน่งบิตนั้น นั่นคือบิตตำแหน่งที่ 1 จะระบุสถานะของบล็อกที่ 1 ในแฟ้มข้อมูล และบิตตำแหน่งถัดไปจะระบุสถานะของบล็อกถัดไปตามลำดับ ค่าบิตเป็น 1 แสดงสถานะไม่ว่างของบล็อกคือมีข้อมูลเก็บอยู่ แต่ถ้าค่าบิตเป็น 0 แสดงสถานะว่างของบล็อกคือไม่มีข้อมูลจัดเก็บอยู่ ดังนั้นถ้าบล็อกใด

ถูกเลือกใช้เพื่อจัดเก็บข้อมูล ค่าลำดับบิตของบล็อกนั้นก็จะมีค่าเป็น 1 และถ้ามีการคืนบล็อกให้กับระบบ ค่าลำดับบิตของบล็อกนั้นก็จะมีค่าเป็น 0 ขนาดเนื้อที่สำหรับจัดเก็บข้อมูลส่วนนี้จะขึ้นอยู่กับจำนวนบล็อกทั้งหมดในฐานข้อมูลที่สร้างถูกขึ้น



ภาพประกอบ 3-4 โครงสร้างการจัดเก็บข้อมูลของแฟ้มข้อมูลในระบบ [เบญจมาศ, 2545]

รายชื่อรีเลชั่น

ข้อมูลที่จัดเก็บในแฟ้มข้อมูลส่วนที่สองจะเป็นส่วนของการจัดเก็บรายชื่อรีเลชันทั้งหมดในหนึ่งฐานข้อมูล เรียกว่า Name Directory และหมายเลขบล็อกแรกเพื่อบอกตำแหน่งเริ่มต้นในการเข้าถึงข้อมูลของแต่ละรีเลชันในฐานข้อมูล โดยที่ชื่อรีเลชันทั้งหมดที่จัดเก็บในแฟ้มข้อมูลจะต้องไม่ซ้ำกัน ขนาดเนื้อที่และจำนวนบล็อกบนแฟ้มข้อมูลในส่วนนี้ขึ้นอยู่กับจำนวนรีเลชันที่ถูกเก็บในแฟ้มข้อมูล โครงสร้างข้อมูลที่ใช้จัดเก็บข้อมูลส่วนนี้จะเป็นโครงสร้างข้อมูลแบบลิงค์ลิสต์ดังแสดงในภาพประกอบ 3-4

ข้อมูลของรีเลชัน

ข้อมูลที่จัดเก็บในแฟ้มข้อมูลส่วนที่สามจะเป็นส่วนของการจัดเก็บข้อมูลจริงของแต่ละรีเลชัน เรียกว่า Data Space ซึ่งการเก็บข้อมูลของรีเลชันในแฟ้มข้อมูลมีข้อจำกัดดังนี้

- แต่ละแฟ้มข้อมูลสามารถจัดเก็บข้อมูลของหลายรีเลชัน แต่ข้อมูลของรีเลชันหนึ่ง ๆจะต้องจัดเก็บลงในแฟ้มข้อมูลเดียวกันเท่านั้น
- หนึ่งรีเลชันสามารถใช้เนื้อที่หรือจำนวนบล็อกสำหรับจัดเก็บข้อมูลของรีเลชันได้มากกว่าหนึ่งบล็อก แต่ข้อมูลในแต่ละบล็อกจะต้องเป็นข้อมูลจากรีเลชันเดียวกันเท่านั้น
- ขนาดเนื้อที่และจำนวนบล็อกที่ใช้ในการเก็บข้อมูลของแต่ละรีเลชันในแฟ้มข้อมูลอาจไม่เท่ากัน โดยเมื่อเพิ่มข้อมูลในรีเลชันก็สามารถใช้บล็อกใหม่และเมื่อมีการลบข้อมูลออกจากรีเลชันก็สามารถคืนบล็อกได้เช่นเดียวกัน

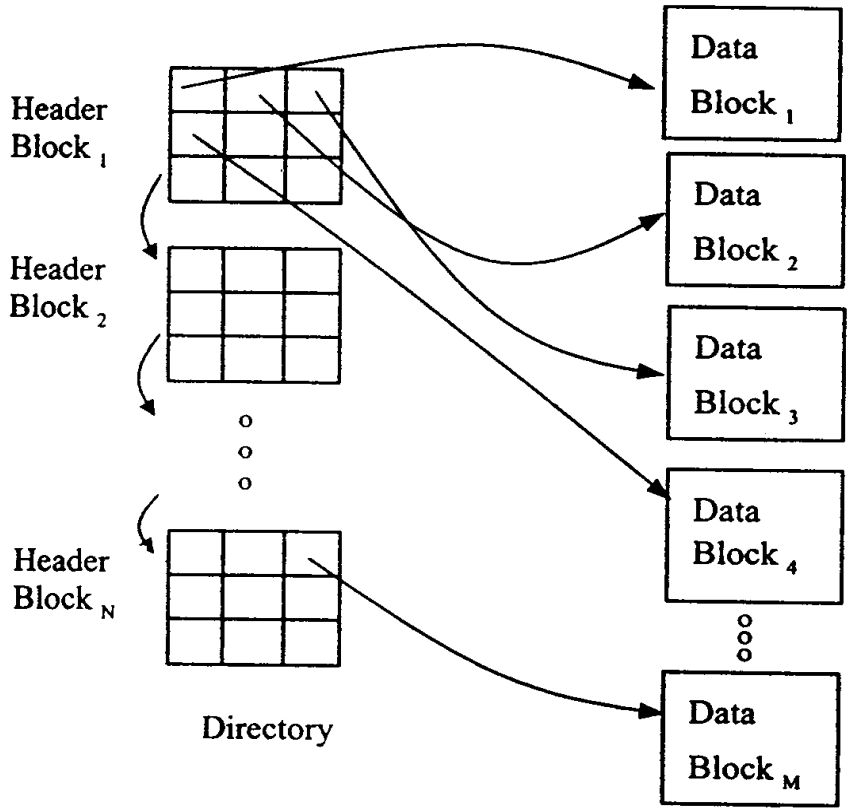
3.2.3 โครงสร้างการจัดเก็บข้อมูลของรีเลชัน

แต่ละทุเปิลของรีเลชันจะถูกจัดเก็บเป็นเรคอร์ดในบล็อกของแฟ้มข้อมูลในส่วนที่เรียกว่า Data Space โดยที่แต่ละบล็อกจะมีขนาดคงที่เท่า ๆ กัน แต่ขนาดของเรคอร์ดที่จัดเก็บในแต่ละบล็อกมีขนาดแตกต่างกันได้ ซึ่งในฐานข้อมูลเชิงสัมพันธ์ เรคอร์ดของต่างรีเลชันอาจมีขนาดแตกต่างกันได้

โครงสร้างการจัดเก็บเรคอร์ดในแฟ้มข้อมูลมีได้หลายรูปแบบ ใน MiniRDBMS ใช้โครงสร้างการจัดเก็บเรคอร์ดของรีเลชันแบบไม่มีการเรียงลำดับใด ๆ ที่เรียกว่า Heap File ดังนั้นเรคอร์ดต่าง ๆ จะถูกจัดเก็บไว้ในตำแหน่งใดก็ได้ในแฟ้มข้อมูลที่มีที่ว่างพอสำหรับเรคอร์ดนั้น ๆ โดยไม่คำนึงลำดับของการจัดเก็บเรคอร์ดในรีเลชัน ภาพประกอบ 3-5 แสดงตัวอย่างโครงสร้างการจัดเก็บเรคอร์ดในรีเลชันหนึ่ง โดยบล็อกของข้อมูลต่าง ๆ ที่จัดเก็บในรีเลชันแบ่งเป็นสองประเภทดังนี้

- Data Block เป็นบล็อกข้อมูลสำหรับจัดเก็บข้อมูลจริงของรีเลชัน
- Header Block เป็นบล็อกข้อมูลสำหรับจัดเก็บตำแหน่งของ Data Block ที่จัดเก็บข้อมูลของรีเลชัน และขนาดเนื้อที่ว่างของ Data Block นั้น

โครงสร้างการจัดเก็บเรคอร์ดแบบนี้จะมีเฉพาะส่วนของ Header Block เท่านั้นที่จะเชื่อมโยงกันเป็นลิงค์ลิสต์แบบทิศทางเดียว เพื่อให้ง่ายต่อการจัดเก็บพื้นที่ว่างบน Data Block ของรีเลชัน แต่ในกรณีที่มีการเพิ่มข้อมูลของรีเลชันและไม่มีเนื้อที่ว่างใน Data Block เก่าของรีเลชันจะต้องขอใช้บล็อกข้อมูลใหม่ และกรณีที่มีการลบข้อมูลเก่าออกจากรีเลชัน ทำให้เกิดบล็อกว่างจะต้องคืนบล็อกว่างให้กับระบบ ทั้งสองกรณีจะต้องมีการเปลี่ยนแปลงแก้ไขข้อมูลในส่วนของ Header Block ด้วย



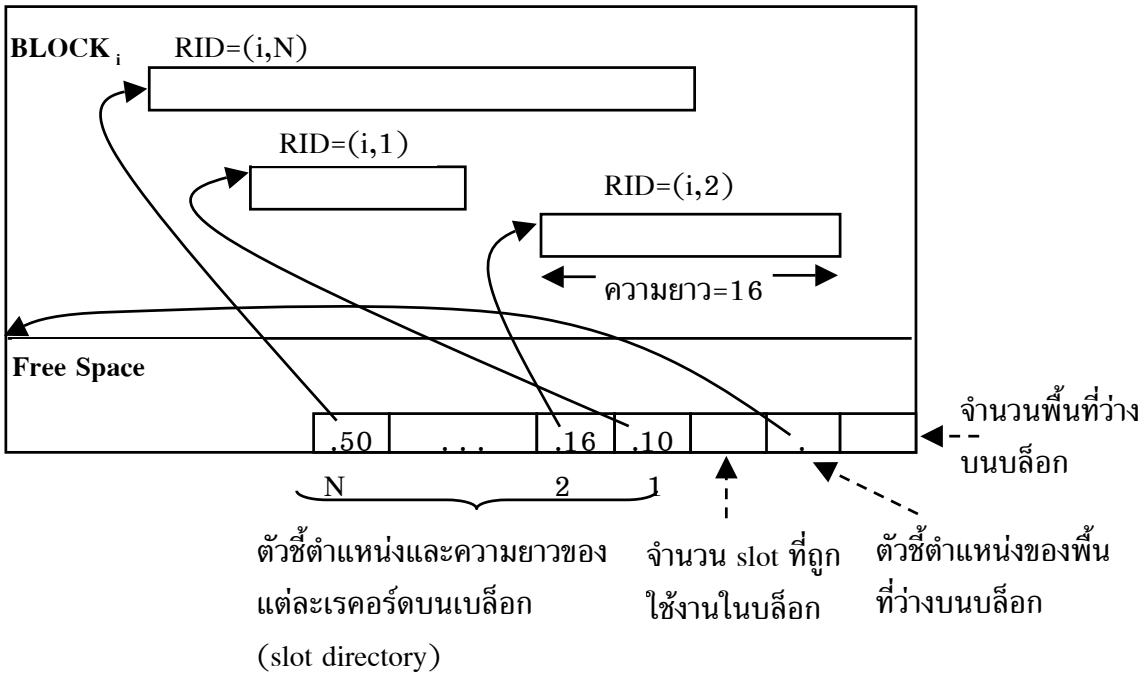
ภาพประกอบ 3-5 โครงสร้างการจัดเก็บข้อมูลของรีเลชัน [เบญจมาศ, 2545]

3.2.4 โครงสร้างของบล็อกข้อมูล

แฟ้มข้อมูลในระบบแบ่งออกเป็นบล็อกขนาดเท่า ๆ กัน แต่ละบล็อกข้อมูลสามารถจัดเก็บเรคอร์ดไว้ได้มากกว่าหนึ่งเรคอร์ด ภาพประกอบ 3-6 แสดงโครงสร้างของแต่ละบล็อกข้อมูล (Data Block Structure) ในแฟ้มข้อมูลบนดิสก์ โดยแต่ละบล็อกบนแฟ้มข้อมูลจะเก็บเรคอร์ดพร้อมทั้งรายละเอียดที่เกี่ยวข้องเพื่อความสะดวกในการจัดเก็บและการเข้าถึงเรคอร์ดในบล็อก ได้แก่ ตำแหน่งของพื้นที่ว่างบนบล็อก กลุ่มข้อมูลที่บอกตำแหน่งและความยาวของแต่ละเรคอร์ดในบล็อก เรียกว่า Slot Directory และจำนวน Slot ที่ถูกใช้งานในบล็อก โดยที่แต่ละ Slot จะบอกตำแหน่งและความยาวของเรคอร์ดหนึ่ง ๆ ในบล็อก

เนื่องจากแต่ละเรคอร์ดในบล็อกข้อมูลจะมีหนึ่ง Slot ใน Slot Directory ที่เก็บตำแหน่งและความยาวของเรคอร์ด ทำให้เรคอร์ดต่าง ๆ ที่จัดเก็บในบล็อกเดียวกันมีหมายเลข Slot (Slot Number) ต่างกันด้วย ดังนั้นแต่ละเรคอร์ดในแฟ้มข้อมูลจะมีหมายเลขเรคอร์ด (Record ID : RID) ที่แตกต่างกันเพื่อใช้ในการอ้างอิงถึงเรคอร์ดที่ต้องการได้ ซึ่งหมายเลขเรคอร์ดจะประกอบด้วยหมายเลขบล็อก (Block ID) ที่เรคอร์ดถูกเก็บอยู่ และหมายเลข slot ที่จัดเก็บตำแหน่งและความยาวของเรคอร์ดนั้น (RID = <Block ID , Slot Number>) ลักษณะ

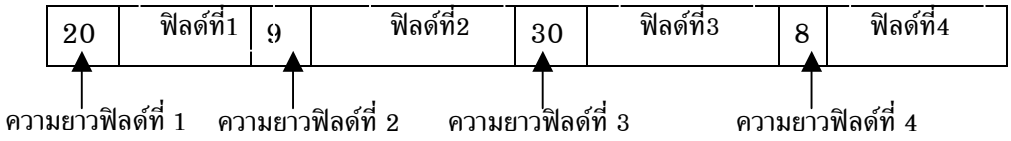
ของโครงสร้างบล็อกที่ออกแบบทำให้สามารถจัดเก็บข้อมูลได้ทั้งเรคอร์ดความยาวคงที่ และเรคอร์ดความยาวแปรได้



ภาพประกอบ 3-6 โครงสร้างของบล็อกข้อมูล [เบญจมาศ, 2545]

3.2.5 โครงสร้างของเรคอร์ด

ข้อมูลในระบบจะถูกจัดเก็บเป็นเรคอร์ดในบล็อกของแฟ้มข้อมูล โครงสร้างของการจัดเก็บเรคอร์ดมี 2 แบบ ได้แก่ เรคอร์ดความยาวคงที่ และเรคอร์ดความยาวแปรได้

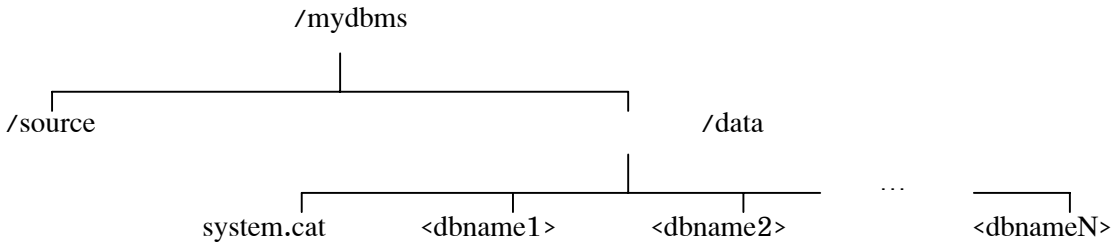


ภาพประกอบ 3-7 โครงสร้างของเรคอร์ดแบบความยาวแปรได้ [เบญจมาศ, 2545]

การสร้างและจัดการกับเรคอร์ดความยาวคงที่ทำได้ง่ายกว่าเรคอร์ดความยาวไม่คงที่ ดังนั้นการจัดเก็บข้อมูลในฐานข้อมูลจะใช้โครงสร้างเรคอร์ดขนาดความยาวคงที่ ส่วนโครงสร้างเรคอร์ดความยาวแปรได้ที่ออกแบบไว้จะถูกจัดเก็บเรคอร์ดของรีเลชันที่จะนำไปแสดงผลบนจอภาพในรูปของตารางที่จะกล่าวถึงในส่วนของการแสดงผล

3.2.6 การจัดเก็บแฟ้มข้อมูลในระบบ

แฟ้มข้อมูลทั้งหลายในระบบถูกจัดเก็บเป็นแบบโครงสร้างต้นไม้ ซึ่งเป็นลักษณะการจัดเก็บแฟ้มข้อมูลของระบบปฏิบัติการ Linux ดังแสดงในภาพประกอบ 3-8 โดยโหนดรากของต้นไม้เป็นไดเรกทอรี (Directory) ชื่อ mydbms



ภาพประกอบ 3-8 การจัดเก็บแฟ้มข้อมูลในระบบ [เบญจมาศ, 2545]

การจัดเก็บแฟ้มข้อมูลทั้งหมดในระบบจะมี mydbms เป็นไดเรกทอรีที่เข้าสู่ระบบโปรแกรมที่พัฒนา ซึ่งแบ่งเป็นไดเรคทอรีย่อย (Subdirectory) อีกสองไดเรคทอรี ดังนี้

- **source** เป็นไดเรกทอรีย่อยสำหรับจัดเก็บแฟ้มข้อมูลที่เป็นโปรแกรมต้นฉบับ (Source Program) ทั้งหมดที่พัฒนาขึ้นในการพัฒนาระบบ
- **data** เป็นไดเรกทอรีย่อยสำหรับจัดเก็บแฟ้มข้อมูลทั้งหมดของระบบ ประกอบด้วย
  - **แฟ้มข้อมูลปทานุกรม** สำหรับจัดเก็บปทานุกรมข้อมูลทั้งหมดในระบบ จะมีเพียงแฟ้มข้อมูลเดียวชื่อ system.cat
  - **แฟ้มข้อมูลของฐานข้อมูล** สำหรับจัดเก็บข้อมูลทั้งหมดในฐานข้อมูลที่ผู้ใช้สร้างขึ้น ในระบบสามารถมีแฟ้มข้อมูลประเภทนี้ได้มากกว่าหนึ่งแฟ้มข้อมูล ขึ้นอยู่กับจำนวนฐานข้อมูลทั้งหมดในระบบที่ผู้ใช้สร้างขึ้น ในแฟ้มข้อมูลจะจัดเก็บเฉพาะข้อมูลที่เป็นของฐานข้อมูลเดียวกัน และมีชื่อแฟ้มข้อมูลเหมือนกับชื่อฐานข้อมูลที่ผู้ใช้สร้างขึ้น

3.3 การจัดการบัฟเฟอร์

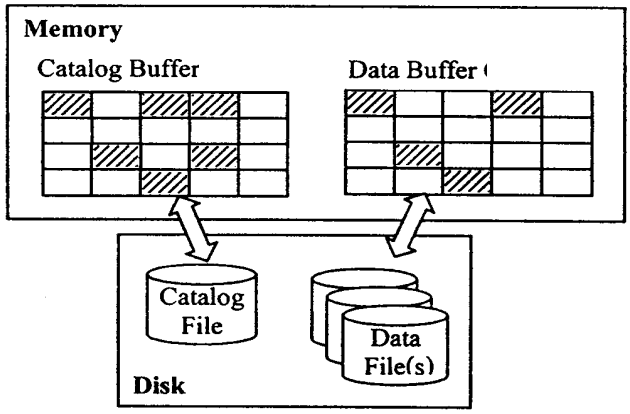
การดำเนินงานของระบบจัดการฐานข้อมูลมีเป้าหมายสำคัญอย่างหนึ่งคือ การลดจำนวนการโอนย้ายข้อมูลระหว่างดิสก์กับหน่วยความจำหลัก วิธีหนึ่งในการลดจำนวนครั้งในการเข้าถึงข้อมูลในดิสก์ คือการเก็บข้อมูลไว้ในหน่วยความจำหลักให้ได้มากที่สุดเท่าที่จะทำได้ เพื่อที่โอกาสข้อมูลที่ต้องการเข้าถึงอยู่ในหน่วยความจำหลักอยู่ก่อนแล้วมีมากขึ้น จึงไม่ต้องเข้าถึงข้อมูลในดิสก์โดยตรง อย่างไรก็ตาม ไม่มีทางเป็นไปได้ที่จะจัดเก็บข้อมูลในดิสก์ไว้ในหน่วยความจำหลัก

ทั้งหมด เนื่องจากขนาดของหน่วยความจำหลักมีขนาดจำกัด ดังนั้นจึงจำเป็นต้องมีการจัดสรรเนื้อที่ว่างบนหน่วยความจำหลักจำนวนหนึ่ง เรียกว่า บัฟเฟอร์ เพื่อเป็นเนื้อที่สำหรับจัดเก็บข้อมูลจากดิสก์จำนวนหนึ่งไว้ในหน่วยความจำหลักเพื่อใช้ในการประมวลผล

3.3.1 โครงสร้างบัฟเฟอร์

ในระบบจะมีการจองเนื้อที่ว่างในหน่วยความจำส่วนหนึ่งที่เรียกว่า บัฟเฟอร์ เพื่อใช้เก็บข้อมูลที่มีการเรียกใช้บ่อย ๆ เพื่อจะได้ไม่จำเป็นต้องอ่านข้อมูลขึ้นมาจากดิสก์ทุกครั้งที่ต้องการใช้ ทำให้ประหยัดเวลาในการทำงาน ในระบบมีการจองเนื้อที่ในหน่วยความจำเพื่อใช้เป็นบัฟเฟอร์สองส่วน ดังแสดงในภาพประกอบที่ 3-9 ซึ่งมีรายละเอียดดังนี้

- **Catalog Buffer** เป็นบัฟเฟอร์ส่วนที่เก็บข้อมูลของปทานุกรม
- **Data Buffer** เป็นบัฟเฟอร์สำหรับจัดเก็บข้อมูลจากฐานข้อมูลที่กำลังใช้งานอยู่ โดยข้อมูลในบัฟเฟอร์นี้จะเป็นข้อมูลจากฐานข้อมูลเดียวกันเท่านั้น

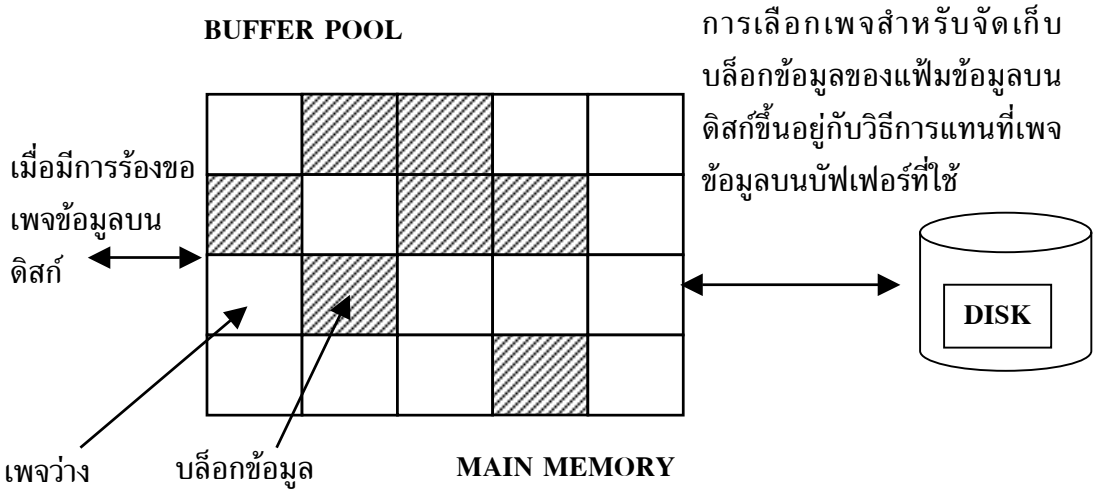


ภาพประกอบ 3-9 บัฟเฟอร์ที่ใช้ในระบบ [เบญจมาศ, 2545]

บัฟเฟอร์จะถูกแบ่งออกเป็นส่วน ๆ ขนาดเท่ากัน เรียกว่า เพจ (Page) แต่ละเพจจะมีขนาดเท่ากับขนาดข้อมูลหนึ่งบล็อกจากแฟ้มข้อมูลบนดิสก์ ในการโอนย้ายข้อมูลระหว่างดิสก์กับหน่วยความจำหลักจะเป็นการโอนย้ายข้อมูลหนึ่งบล็อกในแฟ้มข้อมูลบนดิสก์มาจัดเก็บในหนึ่งเพจของบัฟเฟอร์ หรือเป็นการนำข้อมูลหนึ่งเพจในบัฟเฟอร์บันทึกกลับลงในหนึ่งบล็อกของดิสก์

เมื่อผู้ใช้ต้องการเข้าถึงข้อมูลในแฟ้มข้อมูลบนดิสก์ ทั้งข้อมูลส่วนของปทานุกรมข้อมูลและข้อมูลจากฐานข้อมูลที่ผู้ใช้สร้างขึ้นจะเข้าถึงข้อมูลโดยการค้นหาจากบัฟเฟอร์ที่จัดเตรียมไว้ก่อน หากพบข้อมูลนั้นก็จะจัดส่งข้อมูลให้กับผู้ใช้ แต่ถ้าหากไม่พบจะอ่านข้อมูลจากดิสก์มาบันทึกไว้ในบัฟเฟอร์ แล้วจัดส่งข้อมูลให้กับผู้ใช้ ภาพประกอบ 3-10 แสดงการร้องขอข้อมูลจากเพจบนบัฟเฟอร์และการติดต่อระหว่างเพจบนบัฟเฟอร์กับบล็อกบนดิสก์ ซึ่งข้อมูลเดียวกันใน

บัฟเฟอร์และดิสก์ ข้อมูลในบัฟเฟอร์จะเป็นข้อมูลที่ทันสมัยกว่าข้อมูลในแฟ้มข้อมูลบนดิกส์ ดังนั้นจะต้องทำการบันทึกข้อมูลในบัฟเฟอร์กลับลงแฟ้มข้อมูลเพื่อให้ข้อมูลในดิสก์ถูกต้องทันสมัยเช่นกัน โดยจะมีโปรแกรมหนึ่งทำหน้าที่รับผิดชอบการจัดการเนื้อที่ในส่วนบัฟเฟอร์ เรียกว่าโปรแกรมจัดการบัฟเฟอร์ (Buffer Management)



ภาพประกอบ 3-10 โครงสร้างบัฟเฟอร์และการเข้าถึงข้อมูล [เบญจมาศ, 2545]

3.3.2 โปรแกรมจัดการบัฟเฟอร์

โปรแกรมจัดการบัฟเฟอร์ เป็นโปรแกรมที่ทำหน้าที่ในการจัดสรรเนื้อที่บัฟเฟอร์ว่างให้กับผู้ร้องขอข้อมูลต่าง ๆจากฐานข้อมูลที่จัดเก็บอยู่ในแฟ้มข้อมูลบนดิสก์ โดยจะทำหน้าที่อ่านข้อมูลหนึ่งบล็อกในแฟ้มข้อมูลบนดิสก์ที่ถูกร้องขอมาเก็บไว้ในหน่วยความจำในเพจว่างของบัฟเฟอร์ และจัดเก็บรายละเอียดในการใช้งานบัฟเฟอร์ต่าง ๆ เช่น หมายเลขบล็อกของข้อมูลในแฟ้มข้อมูลที่อ่านขึ้นมาเก็บในเพจของบัฟเฟอร์ และเมื่อมีการร้องขอบล็อกข้อมูล ในแฟ้มข้อมูลบนดิสก์ก็จะสามารถเลือกข้อมูลในบัฟเฟอร์ให้กับผู้ร้องขอได้อย่างถูกต้อง

3.3.3 ขั้นตอนวิธีการแทนที่เพจข้อมูลบนบัฟเฟอร์

ขั้นตอนวิธีการแทนที่เพจข้อมูลบนบัฟเฟอร์เป็นขั้นตอนวิธีสำหรับการเลือกเพจเดิมที่อยู่ในบัฟเฟอร์ออกเพื่อแทนที่ด้วยข้อมูลเพจใหม่ ซึ่งสามารถทำได้หลายขั้นตอนวิธี ยกตัวอย่างเช่น First-In-First-Out (FIFO) algorithm เป็นขั้นตอนวิธีการเลือกเพจเดิมที่อยู่ในบัฟเฟอร์เป็นเวลานานที่สุดออกก่อน Least-Recently-Used (LRU) algorithm เป็นขั้นตอนวิธีการเลือกข้อมูลเพจเดิมที่เคยถูกอ้างถึงเมื่อเร็ว ๆ นี้น้อยที่สุดออกก่อน Most-Recently-Used (MRU) algorithm เป็นขั้นตอนวิธีการเลือกข้อมูลเพจเดิมที่เคยถูกอ้างถึงเมื่อเร็ว ๆ นี้มากที่สุดออกก่อน เป็นต้น

ในระบบเลือกใช้ขั้นตอนวิธีการแทนที่เพจข้อมูลบนบัฟเฟอร์แบบ LRU ซึ่งเป็นขั้นตอนวิธีการที่ระบบปฏิบัติการเลือกใช้อยู่บ่อย ๆ โดยมีสมมติฐานที่ว่าข้อมูลเพจใดที่เคยถูกอ้างถึงเมื่อเร็ว ๆ นี้มากที่สุดจะถูกอ้างถึงอีกครั้งหนึ่ง ดังนั้นจะเลือกข้อมูลเพจเดิมที่เคยอ้างถึงเมื่อเร็ว ๆ นี้น้อยที่สุดออกก่อนเพื่อแทนที่ด้วยข้อมูลเพจใหม่ โดยจะมีการเก็บสถานะและลำดับการใช้งานเพจบนบัฟเฟอร์ เพื่อเป็นข้อมูลในการแทนที่เพจข้อมูล และข้อมูลเหล่านี้จะถูกปรับปรุงเปลี่ยนแปลงทุกครั้งที่มีการร้องขอเพื่อเข้าถึงบล็อกข้อมูลจากดิสก์

3.4 การจัดการปทานุกรมข้อมูล

ระบบจัดการฐานข้อมูลเชิงสัมพันธ์ดำเนินงานกับข้อมูลในฐานข้อมูลเชิงสัมพันธ์ที่ประกอบด้วยตารางข้อมูลสองมิติหรือรีเลชัน ข้อมูลในปทานุกรมข้อมูลจึงถูกจัดเก็บในรูปฐานข้อมูลเชิงสัมพันธ์ด้วยเพื่อที่ระบบจัดการฐานข้อมูลจะได้สามารถดำเนินงานกับข้อมูลในปทานุกรมข้อมูลได้ด้วยวิธีการเดียวกัน โครงสร้างปทานุกรมข้อมูลของ MiniRDBMS ในรูป OR-Diagram แสดงในภาพประกอบ 3-11

3.4.1 โครงสร้างของปทานุกรม

โครงสร้างปทานุกรมข้อมูลของ MiniRDBMS มีรายละเอียดและตารางข้อมูลที่สมนัยกันดังนี้

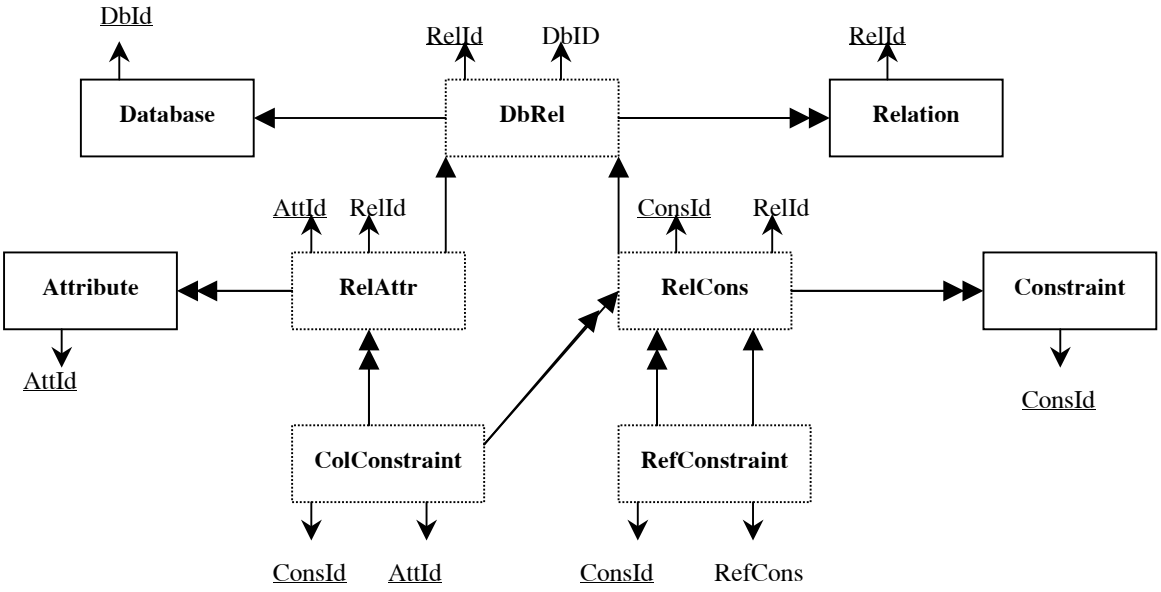
ความสัมพันธ์ระหว่าง Database กับ Relation โดย

- ฐานข้อมูลหนึ่ง ๆ จะมีจำนวนรีเลชันได้หลายรีเลชัน
- แต่ละรีเลชันจะมีฐานข้อมูลใดฐานข้อมูลหนึ่งเป็นเจ้าของได้หนึ่งฐานข้อมูลเท่านั้น

เมื่อแปลงเป็นโครงสร้างตารางข้อมูลที่สมนัยกันจะได้

Database(<u>DbId</u>, <u>Dbname</u>, DbCreate) โดยแต่ละแถวแนวนอนของตารางข้อมูล Database แทนรายละเอียดข้อมูลของฐานข้อมูลหนึ่งฐานข้อมูล

Relation(<u>RelId</u>, DbId, RelName, NoAtt, TupSize, RelCreate, RelLstUpdate) โดยแต่ละแถวแนวนอนของตารางข้อมูล Relation แทนรายละเอียดข้อมูลของรีเลชันหนึ่งรีเลชัน โดยมีข้อมูลของ DbRel จัดเก็บรวมกับข้อมูลของ Relation ด้วย



ภาพประกอบ 3-11 ปทานุกรมข้อมูลระบบ [เบญจมาศ, 2545]

ความสัมพันธ์ระหว่าง DbRel กับ Attribute โดย

- รีเลชันหนึ่ง ๆ จะมีจำนวนแอตทริบิวได้หลายแอตทริบิว
- แต่ละแอตทริบิวจะมีรีเลชันใดรีเลชันหนึ่งเป็นเจ้าของได้หนึ่งรีเลชันเท่านั้น

เมื่อแปลงเป็นโครงสร้างตารางข้อมูลที่สมนัยกันจะได้

Attribute (AttId, RelId, AttName, AttDomain, AttPos, AttLen, AttPrecision, AttScale, AttNullable, AttDefaVal) แต่ละแถวแนวนอนของตารางข้อมูล Attribute แทนรายละเอียดข้อมูลของแอตทริบิวหนึ่งแอตทริบิว โดยมีข้อมูลของ RelAttr จัดเก็บรวมกับข้อมูลของ Attribute ด้วย

ความสัมพันธ์ระหว่าง DbRel กับ Constraint โดย

- รีเลชันหนึ่ง ๆ จะมีการกำหนดเงื่อนไขบังคับได้หลายเงื่อนไขบังคับ
- แต่ละเงื่อนไขบังคับจะมีรีเลชันใดรีเลชันหนึ่งเป็นเจ้าของได้หนึ่งรีเลชันเท่านั้น

เมื่อแปลงเป็นโครงสร้างข้อมูลจะได้ตารางข้อมูลเชิงสัมพันธ์ที่สมนัยกันคือ

Constraint (ConsId, RelId, ConsType) แต่ละแถวแนวนอนของตารางข้อมูล Constraint แทนรายละเอียดข้อมูลของการกำหนดเงื่อนไขบังคับหนึ่งเงื่อนไขบังคับ โดยมีข้อมูลของ RelCons จัดเก็บรวมกับข้อมูลของ Constraint ด้วย

ความสัมพันธ์ระหว่าง RelCons กับ RelAttr โดย

- เงื่อนไขบังคับหนึ่ง ๆ สามารถมีแอตทริบิวที่เกี่ยวข้องกับเงื่อนไขบังคับนั้นได้มากกว่าหนึ่งแอตทริบิว
- แต่ละแอตทริบิวสามารถถูกกำหนดเงื่อนไขบังคับได้มากกว่าหนึ่งเงื่อนไขบังคับ เมื่อแปลงเป็นโครงสร้างข้อมูลจะได้ตารางข้อมูลเชิงสัมพันธ์ที่สมนัยกันคือ

ColConstraint (ConsId, AttId, AttPos) แต่ละแถวแนวนอนของตารางข้อมูล ColConstraint แทนรายละเอียดข้อมูลของการกำหนดเงื่อนไขบังคับของหนึ่งแอตทริบิว

ความสัมพันธ์ระหว่าง RelCons กับ RelCons โดย

- เงื่อนไขบังคับหนึ่ง ๆ (กรณีเงื่อนไขบังคับการกำหนดคีย์นอก) จะอ้างอิงเงื่อนไขบังคับการกำหนดคีย์หลักได้เพียงหนึ่งเงื่อนไขบังคับเท่านั้น
- แต่ละเงื่อนไขบังคับ (กรณีเงื่อนไขบังคับการกำหนดคีย์หลัก) จะสามารถถูกอ้างอิงได้หลายครั้ง

เมื่อแปลงเป็นโครงสร้างข้อมูลจะได้ตารางข้อมูลเชิงสัมพันธ์ที่สมนัยคือ

RefConstraint (ConsId, RefCons, OnDelete) แต่ละแถวแนวนอนของตารางข้อมูล RefConstraint แทนรายละเอียดข้อมูลของเงื่อนไขบังคับการกำหนดคีย์นอกของหนึ่งเงื่อนไขบังคับ

3.4.2 รายละเอียดข้อมูลในปทานุกรมข้อมูล

- ชื่อแอตทริบิวในรีเลชันหนึ่ง ๆ จะมีชื่อไม่ซ้ำกัน มีความยาวไม่เกิน 15 ตัวอักขระ จะเป็นตัวอักษรภาษาอังกฤษ ตัวเลข หรือขีดเส้นใต้ แต่ต้องไม่มีช่องว่างและต้องขึ้นต้นด้วยตัวอักษร
- รูปแบบชนิดของข้อมูล การกำหนดชนิดข้อมูลจะกำหนดค่าเป็นตัวอักษรประกอบด้วย
  - C หมายถึง ข้อมูลอักขระ
  - I หมายถึง ข้อมูลเลขจำนวนเต็ม
  - R หมายถึง ข้อมูลเลขทศนิยม
  - D หมายถึง ข้อมูลวันที่ (Date) มีรูปแบบ วัน/เดือน/ปี โดยปีต้องเป็นตัวเลข 4 ตัว
  - 0 หมายถึง ข้อมูลหมายเลขเรคอร์ดในแฟ้มข้อมูล
- ขนาดของฟิลด์ การกำหนดขนาดความยาวของข้อมูลที่จัดเก็บจริงในแต่ละฟิลด์ของรีเลชัน จะมีหน่วยความยาวเป็นไบต์ โดยจะยาวได้ไม่เกิน 255 ไบต์

รายละเอียดแต่ละรีเลชันของปทานุกรม มีดังนี้

ตาราง 3-1 โครงสร้างปทานุกรมข้อมูล Database

ชื่อแอตทริบิว	ชนิด	รูปแบบ	ขนาด	คำอธิบาย
DbId	P	0	8	รหัสฐานข้อมูล
DbName	S	C	15	ชื่อฐานข้อมูล
NoRel	N	I	2	จำนวนรีเลชันในฐานข้อมูล
DbCreate	N	D	4	วันที่สร้างฐานข้อมูล

ตาราง 3-2 โครงสร้างปทานุกรมข้อมูล Relation

ชื่อแอตทริบิว	ชนิด	รูปแบบ	ขนาด	คำอธิบาย
RelId	P	0	8	รหัสรีเลชัน
DbId	N	0	8	รหัสฐานข้อมูล
RelName	N	C	15	ชื่อรีเลชัน
NoAtt	N	I	2	จำนวนแอตทริบิวที่มีในรีเลชัน
TupLen	N	I	2	ขนาดทุเปิล
RelCreate	N	D	4	วันที่สร้างรีเลชัน
RelLstUpdate	N	D	4	วันที่ปรับปรุงรีเลชันครั้งหลังสุด

ตาราง 3-3 โครงสร้างปทานุกรมข้อมูล Attribute

ชื่อแอตทริบิว	ชนิด	รูปแบบ	ขนาด	คำอธิบาย
AttId	P	0	8	รหัสฐานข้อมูล
RelId	N	0	8	รหัสรีเลชัน
AttName	N	C	15	ชื่อแอตทริบิว
AttDomain	N	C	1	ชนิดข้อมูลของค่าแอตทริบิว
AttPos	N	I	4	ตำแหน่งเริ่มต้นของข้อมูลในระเบียน
AttLen	N	I	4	ขนาดของข้อมูลในระเบียนซึ่งมีหน่วยเป็นไบต์
AttPrecison	N	I	1	จำนวนตัวเลขหน้าตำแหน่งทศนิยม
AttScale	N	I	1	จำนวนตัวเลขหลังตำแหน่งทศนิยม
AttNullable	N	C	1	ค่าแฟลกการใส่ค่าว่าง
AttDefault	N	C	256	ค่าเริ่มต้นของข้อมูลในทุเปิล

ตาราง 3-4 โครงสร้างปทานุกรมข้อมูล Constraint

- P หมายถึง การกำหนดคีย์หลัก
- S หมายถึง การกำหนดคีร์รอง
- F หมายถึง การกำหนดคีย์นอกหรือเงื่อนไขบังคับการอ้างอิงค่าแอตทริบิว

ชื่อแอตทริบิว	ชนิด	รูปแบบ	ขนาด	คำอธิบาย
ConsId	P	0	8	รหัสเงื่อนไขบังคับ
RelId	N	0	8	รหัสรีเลชัน
ConsType	N	C	1	ประเภทของเงื่อนไขบังคับ

ตาราง 3-5 โครงสร้างปทานุกรมข้อมูล ColConstraint

ชื่อแอตทริบิว	ชนิด	รูปแบบ	ขนาด	คำอธิบาย
ConsId	P	0	8	รหัสเงื่อนไขบังคับ
AttId	P	0	8	รหัสแอตทริบิวที่มีการกำหนดเงื่อนไขบังคับ
AttPos	N	I	1	ตำแหน่งลำดับของแอตทริบิว

ตาราง 3-6 โครงสร้างปทานุกรมข้อมูล RefConstraint

ชื่อแอตทริบิว	ชนิด	รูปแบบ	ขนาด	คำอธิบาย
ConsId	P	0	8	รหัสเงื่อนไขบังคับ
ConsRef	N	0	8	รหัสเงื่อนไขบังคับของการกำหนดคีย์หลักของรีเลชันที่ถูกอ้างถึง
OnDelete	N	C	1	ค่าแฟลกการลบทุเปิลในรีเลชันแบบต่อเนื่อง