

บทที่ 2

หลักการและทฤษฎี

2.1 บทนำสู่การบริหารขององค์กร

การบริหารขององค์กรนับได้ว่าเป็นมีความสำคัญมากในเกือบจะทุกๆประเภทของธุรกิจ ไม่ว่าจะเป็นธุรกิจที่เกี่ยวข้องกับสินค้า หรือการให้บริการ ผู้จัดการฝ่ายผลิตหรือฝ่ายปฏิบัติการของหน่วยงานจะต้องรับผิดชอบในการควบคุมต้นทุนค่าใช้จ่ายในการดำเนินงาน ซึ่งต้นทุนชนิดหนึ่งที่มีความสำคัญอย่างมากของการปฏิบัติการ ก็คือ ค่าใช้จ่ายที่ลงทุนไปในวัตถุดิบ วัสดุสิ้นเปลือง งานระหว่างผลิต และผลิตภัณฑ์สำเร็จรูปที่ยังมิได้ทำการจัดส่ง ถ้าการลงทุนในค่าใช้จ่ายเหล่านี้มากเกินไป จะส่งผลให้ค่าใช้จ่ายของเงินทุนสูง ค่าใช้จ่ายของการดำเนินงานสูง และทำให้ประสิทธิภาพของการผลิตลดลง เมื่อมีการใช้พื้นที่มากเกินไปในการดูแลรักษาขององค์กร

การควบคุมขององค์กรเป็นสิ่งสำคัญที่ผู้บริหารควรให้ความสนใจและเอาใจใส่อย่างใกล้ชิด ทั้งนี้เพราะขององค์กรเป็นทรัพย์สินที่มีมูลค่าสูงที่สุดในกลุ่มของทรัพย์สินหมุนเวียนของการผลิต ปัญหาที่เกิดขึ้นในการควบคุมขององค์กรอาจจะเป็นสาเหตุหนึ่งที่นำมาซึ่งความล้มเหลวของกิจการได้ ในธุรกิจอุตสาหกรรมถ้าวัตถุดิบและชิ้นส่วนประกอบต่างๆ มีอยู่ไม่เพียงพอกับความต้องการของการผลิตแล้ว ก็อาจจะทำให้เกิดปัญหาถึงขั้นการผลิตหยุดชะงักได้ และอาจส่งปัญหาถึงขั้นการส่งสินค้าไม่ทันตามกำหนดเวลาของลูกค้า ซึ่งอาจจะเป็นเหตุให้ลูกค้าขาดความเชื่อถือและสูญเสียลูกค้าได้ แต่ถ้าเราพยายามมีขององค์กรไว้มากๆเพื่อป้องกันมิให้เกิดการขาดแคลนวัตถุดิบ ชิ้นส่วน หรือผลิตภัณฑ์สำเร็จรูป เราจำเป็นจะต้องใช้เงินเป็นมูลค่ามหาศาลเพื่อที่จะถือครองขององค์กรนั้นไว้ เช่น ต้นทุนราคาขององค์กร และต้นทุนในการจัดให้มีขององค์กร ในการควบคุมขององค์กรที่ดีจึงเป็นเรื่องที่เกี่ยวข้องกับความพยายามในการทำให้วัตถุประสงค์ 2 ประการในการดำเนินการให้มีขององค์กรเกิดความสมดุลในระดับที่เหมาะสมที่สุด วัตถุประสงค์ประการแรกคือ เพื่อให้การลงทุนทั้งสิ้นในขององค์กรต่ำที่สุด วัตถุประสงค์ประการที่สองคือ พยายามทำให้ระดับการให้บริการลูกค้าและการให้บริการแผนกผลิตของบริษัทเองสูงที่สุด ดังนั้นในการควบคุมขององค์กรที่ดีย่อมทำให้เกิดผลดีทั้งในแง่ของการเพิ่มประสิทธิภาพและลดค่าใช้จ่ายในการดำเนินงาน

2.1.1 ประเภทและความสำคัญของของคงคลัง

เมื่อเรามองของคงคลังในมุมของการผลิต สามารถแบ่งประเภทของคงคลังออกได้เป็น 4 ประเภท ดังนี้

2.1.1.1 วัตถุดิบและชิ้นส่วนที่สั่งซื้อ (Raw Materials and Purchased Components) ของคงคลังเหล่านี้เป็นวัสดุขั้นต้นที่ใช้ในการทำชิ้นส่วนและผลิตภัณฑ์สำเร็จรูป สำหรับชิ้นส่วนที่สั่งซื้อก็เปรียบเสมือนวัตถุดิบ แตกต่างกันไปก็เพียงแต่ว่า บริษัทภายนอกเป็นผู้ดำเนินการผลิตชิ้นส่วนนั้นทั้งหมดหรือเพียงบางส่วน

2.1.1.2 ของคงคลังระหว่างกระบวนการผลิต (In-process Inventory) หลังจากกระบวนการผลิตเริ่มต้นโดยการนำวัตถุดิบ และชิ้นส่วนประกอบที่สั่งซื้อจากภายนอกเข้าสู่กระบวนการผลิต จะมีอยู่ช่วงเวลาหนึ่ง (ช่วงเวลานำของการผลิต) ก่อนที่กระบวนการผลิตจะเสร็จสิ้น ช่วงเวลาระหว่างนั้น ของคงคลังเหล่านี้จะอยู่ในระหว่างกระบวนการผลิต เพื่อรอคอยการผลิตขั้นต่อไปให้เป็นผลิตภัณฑ์สำเร็จรูป

2.1.1.3 ผลิตภัณฑ์สำเร็จรูป (Finished Product) ผลิตภัณฑ์สำเร็จรูปอาจจะเก็บอยู่ในโรงงานหรือในคลังสินค้าก่อนที่จะส่งให้กับลูกค้า ของคงคลังประเภทนี้ประกอบด้วยชิ้นส่วนเพื่อบริการและผลิตภัณฑ์ขั้นสุดท้าย

2.1.1.4 ของคงคลังที่เป็นเครื่องมือและชิ้นส่วนเพื่อการซ่อมบำรุงและซ่อมแซม (Maintenance, Repair, and Tooling Inventories) ของคงคลังเหล่านี้ได้แก่ เครื่องมือกัด และอุปกรณ์จับยึดชิ้นงานที่ใช้กับเครื่องจักรในโรงงาน และชิ้นส่วนเพื่อการซ่อมแซมที่จำเป็นต่อการปรับเครื่องจักรเมื่อเครื่องจักรเกิดเสียหายขึ้นมา รวมทั้งชิ้นส่วนที่เป็นอะไหล่เครื่องไฟฟ้าก็รวมอยู่ในของคงคลังประเภทนี้ด้วย

ของคงคลังเหล่านี้มีส่วนสำคัญในการดำเนินงานมาก โดยเฉพาะอย่างยิ่งในโรงงานอุตสาหกรรม ซึ่งพอสรุปให้เห็นถึงความสำคัญของคงคลังแต่ละประเภทได้ดังต่อไปนี้

ของคงคลังที่เป็นผลิตภัณฑ์สำเร็จรูป

- ช่วยป้องกันความผิดพลาดอันเกิดจากความต้องการผลิตภัณฑ์ที่มีมากกว่าที่พยากรณ์ไว้ การผิดพลาดจะไม่ได้มีการตอบสนองถ้ากิจการไม่มีของคงคลังที่เป็นผลิตภัณฑ์สำเร็จรูปเก็บไว้ ทำให้ธุรกิจต้องขาดกำไรที่ควรจะได้ไป และอาจทำให้ความเชื่อถือของลูกค้าที่มาติดต่อธุรกิจลดลง และในกรณีที่รุนแรงก็อาจทำให้ลูกค้าหันไปซื้อสินค้าจากคู่แข่งกันได้ แต่ถ้าเรามีของคงคลังเก็บไว้จำนวนหนึ่ง ก็จะทำให้ความเสียหายดังกล่าวบรรเทาลงได้

ขององค์กรระหว่างกระบวนการผลิต

- ช่วยให้การผลิตสามารถดำเนินไปได้อย่างสม่ำเสมอ ถึงแม้ว่าการทำงานในแต่ละหน่วยผลิตจะมีความเร็วไม่เท่ากัน เช่น หน่วยผลิตที่มีความเร็วช้า เราอาจจะให้ผลิตเก็บไว้ล่วงหน้า

ขององค์กรที่เป็นวัตถุดิบหรือชิ้นส่วนที่สั่งซื้อ

- เพื่อป้องกันการขาดแคลนวัตถุดิบหรือชิ้นส่วนอันเนื่องจากการล่าช้าด้วยเหตุผลหลายประการ เช่น การเปลี่ยนแปลงกำหนดเวลาในการขนส่งของผู้ขาย ผู้ขายขาดแคลนวัตถุดิบไม่สามารถผลิตชิ้นส่วนที่สั่งได้ทัน หรือเกิดการนัดหยุดงานที่โรงงานผู้ขาย หรือเกิดอุทกภัย เป็นต้น ด้วยเหตุนี้จึงต้องมีวัตถุดิบคงเหลือไว้ให้เพียงพอ วัตถุดิบหรือชิ้นส่วนใดที่สำคัญต้องมีการเก็บไว้ให้มากพอ

2.1.2 เหตุผลและความจำเป็นที่ต้องมีขององค์กร

จากความสำคัญขององค์กรที่ได้กล่าวมาข้างต้น จึงพอสรุปถึงเหตุผลและความจำเป็นที่ต้องมีขององค์กรไว้ดังนี้

2.1.2.1 เพื่อลดค่าใช้จ่ายในการขนส่งและการผลิต

2.1.2.2 ปรับให้เกิดความสมดุลระหว่างความต้องการที่เกิดขึ้นและการจัดหาขององค์กรเข้ามาเก็บไว้ในคลัง การขาดสมดุลไม่ว่าจะมีความต้องการสูงกว่าปริมาณที่จัดหาเข้ามาเก็บไว้ในคลัง หรือจัดหาของเข้ามาเก็บไว้ในคลังมากกว่าความต้องการย่อมหมายถึง การมีสต็อกมากเกินไปหรือการขาดสต็อก

2.1.2.3 เพื่อให้การผลิตสามารถดำเนินได้อย่างต่อเนื่อง โดยพิจารณาขององค์กรเป็นส่วนหนึ่งของการผลิต

2.1.2.4 เพื่อตอบสนองความต้องการของตลาดที่มีความไม่แน่นอน ทำให้มีสินค้าตอบสนองลูกค้าอย่างต่อเนื่อง (รศ.พิภพ ลลิตาภรณ์, 2543. หน้า1-3)

2.2 ระบบแฟ้มข้อมูล (File System)

ในอดีต องค์กรต่าง ๆ มักจัดเก็บเอกสารไว้ในแฟ้มเอกสารต่าง ๆ ซึ่งมีความเกี่ยวข้องกันทางด้านข้อมูลน้อยหรืออาจไม่มีเลย ขึ้นอยู่กับความต้องการในการใช้ข้อมูลนั้น ๆ เช่น ประวัติการรักษาพยาบาล ที่โดยทั่วไปมักจะแยกเก็บในแฟ้มเอกสารเฉพาะคนไข้แต่ละคน หรือประวัติพนักงาน ที่อาจจัดเก็บในแฟ้มเอกสารแยกตามฝ่ายที่สังกัด ซึ่งข้อมูลของพนักงานแต่ละคนในแต่ละ

ละแฟ้มเอกสารจะมีความเกี่ยวข้องกันตามฝ่ายที่สังกัด เช่น แฟ้มเอกสารประวัติพนักงานของฝ่ายธุรการ แฟ้มเอกสารประวัติพนักงานของฝ่ายการเงิน ฯลฯ เป็นต้น แต่ต่อมาเมื่อองค์กรมีขนาดใหญ่ขึ้นจากเดิมที่สามารถค้นหาเอกสารจากแฟ้มเอกสารเพียงแฟ้มเดียว ก็เริ่มต้องค้นหาเอกสารจากแฟ้มเอกสารต่าง ๆ จำนวนมากขึ้น ส่งผลให้การค้นหาเอกสารเป็นงานที่ต้องใช้เวลา และมีความยากลำบากมากขึ้น การจัดเก็บเอกสารในคอมพิวเตอร์จึงถูกริเริ่มนำมาใช้ในองค์กรแทนการจัดเก็บในรูปแบบเดิม แต่การจัดเก็บเอกสารในคอมพิวเตอร์ในยุคแรก ๆ ยังคงไม่ค่อยมีประสิทธิภาพมากนัก เนื่องจากยังคงมีรูปแบบการจัดเก็บคล้ายกับการจัดเก็บเอกสารในลักษณะเดิมอยู่ โดยเป็นเพียงการนำเอาเอกสารต่าง ๆ ในแต่ละแฟ้มเอกสาร มาจัดเก็บในรูปของแฟ้มข้อมูลแทน และด้วยเหตุนี้ จึงจำเป็นที่จะต้องอาศัยผู้ที่มีความชำนาญ เช่น โปรแกรมเมอร์หรือนักวิเคราะห์ระบบ เข้ามาช่วยกำหนดโครงสร้างของแฟ้มข้อมูลเพื่อที่จะสามารถนำแฟ้มข้อมูลนั้นไปจัดเก็บข้อมูลและนำไปประมวลผลได้ตามความต้องการ

จากบทบาทของคอมพิวเตอร์ที่เข้ามามีอิทธิพลต่อการดำเนินงานภายในองค์กร ได้ส่งผลให้การจัดเก็บข้อมูลในแฟ้มข้อมูล มีการใช้งานแพร่หลายมากยิ่งขึ้น จากเดิมที่มีเพียง 2 หรือ 3 แฟ้มข้อมูล ได้เพิ่มจำนวนขึ้นเป็น 10 ถึง 20 แฟ้มข้อมูล ดังนั้น จึงต้องมีการเข้ามาควบคุมทางด้านโครงสร้าง และการใช้งานแฟ้มข้อมูลต่าง ๆ ให้มีความเหมาะสมต่อการใช้งานมากขึ้น และรวบรวมแฟ้มข้อมูลเหล่านี้ เข้าเป็นระบบที่เรียกว่า "ระบบแฟ้มข้อมูล (File System)"

การใช้งานระบบแฟ้มข้อมูล จะต้องอาศัยโปรแกรมเมอร์พัฒนาโปรแกรมเพื่ออ่านข้อมูลจากแฟ้มข้อมูลต่าง ๆ ขึ้นมาประมวลผล เพื่อให้ได้ผลลัพธ์ตามที่ผู้ใช้ต้องการ ภาษาคอมพิวเตอร์ที่ใช้ในการพัฒนาโปรแกรม โดยทั่วไปจะได้แก่ ภาษาคอมพิวเตอร์ในยุคที่ 3 (Third-generation Language (3GL)) เช่น ภาษา COBOL, FORTRAN, BASIC ฯลฯ เป็นต้น แต่ภาษาคอมพิวเตอร์เหล่านี้มีข้อจำกัดในการเรียกใช้ข้อมูลจากแฟ้มข้อมูล เนื่องจากภาษาคอมพิวเตอร์เหล่านี้จะอ้างถึงข้อมูลในแฟ้มข้อมูลตามโครงสร้างทางกายภาพของข้อมูลภายในแฟ้มข้อมูลนั้น เช่น เมื่อต้องการอ้างถึงข้อมูลในแฟ้มข้อมูลลูกค้า ซึ่งมีโครงสร้างดัง ตารางที่ 2.1

ตารางที่ 2.1 โครงสร้างแฟ้มข้อมูล

C0001	ABC Industry	111 Moo 4 Samutprakarn	552-000
C0002	Thai Steel Co., Ltd.	100 Petchkasaem Ratchaburi	321-180
C0003	ITT Computer Co., Ltd.	90 III Bldg. Bangkok	433-0015
C0004	KKK (Thailand) Co., Ltd.	18 Sathorn Bldg. Bangkok	236-8897

ในภาษา COBOL จะต้องมีการนิยาม (Declare) โครงสร้างของ Record เช่น ประเภทของข้อมูล ขนาดของแต่ละ Field ไว้ในส่วน Data Division ตามโครงสร้างทางกายภาพของ Record ในแฟ้มข้อมูลลูกค้าก่อน จึงจะสามารถอ้างถึง Field ต่าง ๆ ในโปรแกรมได้ดังตัวอย่างต่อไปนี้

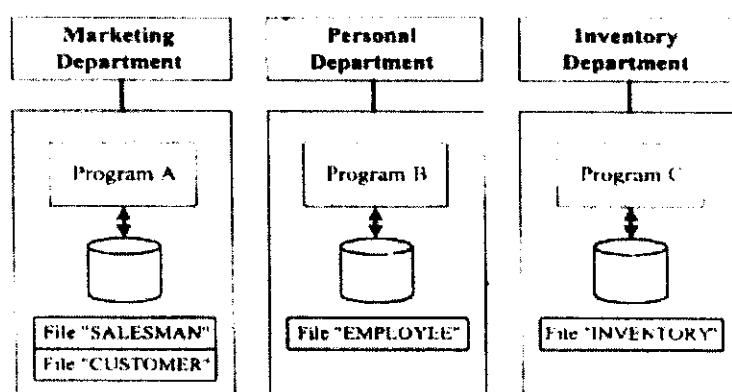
01 CUSTOMER	
02 CUST_NO PIC X(50).	
02 CUST_NAME	PIC X(50).
02 CUST_ADDRESS	PIC X(50).
02 CUST_PHONE	PIC X(8).

ด้วยเหตุนี้ จึงส่งผลให้การพัฒนาแต่ละโปรแกรมขึ้นใช้งานกับแฟ้มข้อมูลต่าง ๆ มีความซับซ้อน และต้องใช้เวลาค่อนข้างมาก รวมทั้งทำให้แต่ละโปรแกรมถูกผูกติดอยู่กับแฟ้มข้อมูลต่าง ๆ ดังนั้นเมื่อต้องมีการเปลี่ยนแปลงโครงสร้างของแฟ้มข้อมูลใดแฟ้มข้อมูลหนึ่ง จึงต้องแก้ไขโปรแกรมต่าง ๆ ที่มีการเรียกใช้ข้อมูลจากแฟ้มข้อมูลนั้นตามไปด้วย ส่งผลให้ค่าใช้จ่ายในการบำรุงรักษาค่อนข้างสูง โดยเฉพาะอย่างยิ่งในกรณีที่ต้องมีการแก้ไขโปรแกรมจำนวนมาก

ในยุคเริ่มต้นของการใช้ระบบแฟ้มข้อมูล แต่ละหน่วยงานในองค์กร จะมีการสร้างระบบแฟ้มข้อมูลขึ้นใช้งานภายในหน่วยงานของตนเอง เช่น ระบบแฟ้มข้อมูลการขายของฝ่ายการตลาด ระบบแฟ้มข้อมูลพนักงานของฝ่ายการพนักงาน ระบบแฟ้มข้อมูลสินค้าคงคลังของฝ่ายคลังสินค้า ฯลฯ เป็นต้น ซึ่งแต่ละหน่วยงานนั้น จะมีการพัฒนาโปรแกรมที่ใช้กับระบบแฟ้มข้อมูลของตนเอง

ขึ้น เพื่อใช้งานภายในหน่วยงานนั้น ๆ เช่น ฝ่ายการตลาดที่มีการพัฒนาโปรแกรมที่ใช้ข้อมูลจากระบบเพิ่มข้อมูลการขายในการออกไปสั่งซื้อ ใบกำกับสินค้า รายงานแสดงยอดการขายในแต่ละเดือน ฯลฯ หรือฝ่ายคลังสินค้าที่มีการพัฒนาโปรแกรมที่ใช้ข้อมูลจากระบบเพิ่มข้อมูลสินค้าคงคลังในการพิมพ์รายงานแสดงยอดคงคลังของสินค้าต่าง ๆ หรือฝ่ายการพนักงานที่มีการพัฒนาโปรแกรมที่ใช้ข้อมูลจากระบบเพิ่มข้อมูลพนักงานในการคำนวณเงินเดือน ค่าตอบแทนเงินได้ ฯลฯ

ดังรูปที่ 2.1 (กิตติ-จำลอง ,2544. หน้า523-525)



รูปที่ 2.1 ระบบเพิ่มข้อมูล

2.2.1 โครงสร้างเพิ่มข้อมูล

โครงสร้างข้อมูลจะประกอบด้วยโครงสร้างพื้นฐานเพิ่มข้อมูลจะเรียงลำดับจากหน่วยเล็กไปหน่วยใหญ่ดังนี้

2.2.1.1 บิต (Bit)

จะประกอบด้วยเลขฐานสอง (Binary) ที่ใช้แทนค่าหน่วยที่เล็กที่สุดของข้อมูลในระบบคอมพิวเตอร์ โดยบิตจะมีอยู่เพียงหนึ่งในสองสถานะเท่านั้นคือ 0 หรือ 1

2.2.1.2 ไบต์ (Byte)

จะประกอบด้วยการนำบิตหลายๆบิตมารวมกันเป็นไบต์ ยกตัวอย่างเช่น 1 ไบต์ มี 8 บิต ก็คือการนำเลข 0 กับ 1 มาเรียงต่อกันจนครบ 1 ไบต์ จึงสามารถสร้างรหัสแทนข้อมูลขึ้นมาเพื่อใช้สำหรับแทนตัวอักษรหรืออักขระที่แตกต่างกันได้ถึง 256 ตัวด้วยกัน

2.2.1.3 ฟิลด์ (Field)

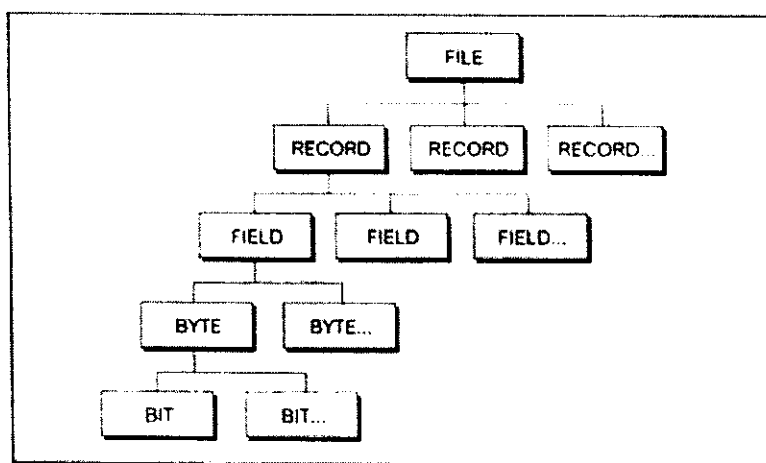
คือการนำอักขระตั้งแต่หนึ่งตัวขึ้นไปมารวมกันเพื่อให้เกิดความหมาย เช่น ฟิลด์ STD_NAME เป็นฟิลด์ที่ใช้เก็บข้อมูลชื่อนักศึกษา

2.2.1.4 เรคอร์ด (Record)

คือกลุ่มของฟิลด์ที่สัมพันธ์กัน เช่น ในหนึ่งเรคอร์ดประกอบด้วยฟิลด์ต่างๆที่ข้องเกี่ยวกับรวมกันเป็นชุด เช่น เรคอร์ดของประวัตินักศึกษา ประกอบไปด้วยฟิลด์ของรหัสนักศึกษา ชื่อ-สกุล วันเกิด ที่อยู่ จังหวัด เป็นต้น

2.2.1.5 ไฟล์ (File)

คือกลุ่มของเรคอร์ดที่สัมพันธ์กัน เช่น แฟ้มประวัติของนักศึกษาประกอบไปด้วยเรคอร์ดของนักศึกษาทั้งหมดในมหาวิทยาลัย(โอกาส เอี่ยมสิริวงศ์ , 2546.หน้า 16-18)



รูปที่ 2. 2 โครงสร้างแฟ้มข้อมูลตามลำดับความสัมพันธ์

2.2.2 ชนิดของข้อมูล

2.2.2.1 ข้อมูลชนิดข้อความ (Text)

เป็นข้อมูลที่ประกอบด้วยอักขระต่างๆที่นำมารวมกันโดยไม่มีรูปแบบที่ชัดเจน ข้อมูลเหล่านี้จะมีความหมายในตัวเองไม่จำเป็นต้องนำมาตีความหมายอีก

2.2.2.2 ข้อมูลชนิดรูปแบบ (Formatted Data)

เป็นข้อมูลที่ประกอบด้วยอักขระต่างๆ ซึ่งมีรูปแบบแน่นอน โดยอาจจัดเก็บอยู่ในรูปแบบของรหัส และจำเป็นต้องนำรหัสดังกล่าวนี้มาตีความอีกครั้งหนึ่งเมื่อใช้งาน เช่น รหัสสาขาวิชา CS คือโปรแกรมของสาขาวิทยาการจัดการคอมพิวเตอร์ เป็นต้น

2.2.2.3 ข้อมูลชนิดรูปภาพ (Image)

เป็นรูปภาพที่ใช้แทนข้อมูล ซึ่งในปัจจุบันนิยมใช้งานมากขึ้น โดยรูปภาพดังกล่าวอาจสร้างด้วยโปรแกรม หรือเป็นภาพที่ได้จากการถ่ายด้วย กล้องดิจิทัลหรือจากการสแกนภาพรวมทั้งภาพจากวิดีโอ

2.2.2.4 ข้อมูลชนิดเสียง (Audio/Sound)

เป็นข้อมูลอีกชนิดหนึ่งที่จัดเก็บเป็นลักษณะของเสียง เช่น midi,digital audio(โอบาส เอี่ยมสิริวงศ์ ,2546. หน้า 18-19)

2.2.3 ประเภทของแฟ้มข้อมูล

2.2.3.1 Master File

เป็นไฟล์จัดเก็บข้อมูลที่มีจะไม่มีการเปลี่ยนแปลงหรือมีสภาพค่อนข้างคงที่ เช่น แฟ้มข้อมูลประวัตินักศึกษา จะประกอบด้วยข้อมูลต่างๆ เช่น รหัสนักศึกษา ชื่อ-สกุล ที่อยู่ คณะ สาขา เป็นต้น

2.2.3.2 Transaction File

เป็นไฟล์ที่จัดข้อมูลการดำเนินธุรกรรมประจำวันที่มีมีความความเคลื่อนไหวอยู่เสมอ เช่น แฟ้มข้อมูลการลงทะเบียนเรียนของนักศึกษาที่จะต้องมีการลงทะเบียนเรียนในทุกๆภาคการศึกษา หรือแฟ้มข้อมูลรายการถอนเงินในบัญชีลูกค้าธนาคาร เป็นต้น

2.2.3.3 Document File

เป็นไฟล์เอกสารหรือไฟล์รายงาน (report file) ต่างๆ ที่เคยผ่านกระบวนการพิมพ์ด้วยโปรแกรมมาก่อน และทำการจัดเก็บในรูปของไฟล์เอกสารด้วยการสำเนาเก็บไว้ในคอมพิวเตอร์ เมื่อต้องการใช้งานก็สามารถเรียกขึ้นมาใช้งานได้อย่างรวดเร็ว เพราะไม่ต้องผ่านโปรแกรมเพื่อประมวลเป็นรายงานอีก

2.2.3.4 Archival File

เป็นไฟล์ข้อมูลที่บรรจุไปด้วย Master File และ Transaction File ซึ่งประกอบด้วยเรคอร์ดต่างๆที่ถูกลบหรือถูกเคลื่อนย้ายจากสื่ออุปกรณ์ออนไลน์ไปจัดเก็บไว้ในสื่อ

อุปกรณ์ที่เป็นแบบออฟไลน์ ตัวอย่างเช่น นำลงไปบันทึกลงในเทป, Archival File เป็นไฟล์ที่อาจจำเป็นต้องจัดเก็บไว้ตามระเบียบหรือกฎเกณฑ์ของรัฐบาลที่กำหนดไว้เพื่อใช้ในการตรวจสอบ และอาจเห็นไฟล์ที่จำเป็นต่อความต้องการมากในกรณีที่ต้องการข้อมูลเก่าเพื่อนำมาตรวจสอบวิเคราะห์ในภายหลัง ในบางครั้งอาจเรียก Archival File ว่า Historical File

2.2.3.5 Table Look-up File

เป็นไฟล์หรือตารางที่ใช้สำหรับการอ้างอิงเพื่อใช้งานร่วมกัน โดยข้อมูลต่างๆ ที่จัดเก็บลงในไฟล์นี้ค่อนข้างคงที่หรือมักไม่ค่อยเปลี่ยนแปลง เช่น ตารางภาษี ตารางรหัสไปรษณีย์ เป็นต้น

2.2.3.6 Audit File

เป็นไฟล์ชนิดพิเศษที่จัดเก็บเรคอร์ดที่ถูกอัปเดตลงในไฟล์ต่างๆ โดยเฉพาะอย่างยิ่ง Master File และ Transaction File ซึ่งจะใช้ร่วมกับ Archive File ในการกู้คืน (recover) ข้อมูลที่เสียหาย ไฟล์ประเภทนี้มักจะถูกกำหนดไว้ในระบบฐานข้อมูลเพื่อให้สามารถกู้คืนระบบในกรณีข้อมูลเกิดความเสียหายในระหว่างการประมวลผล (โอบาส เอี่ยมสิริวงศ์, 2546. หน้า 19-20)

2.2.4 ปัญหาของระบบแฟ้มข้อมูล

ด้วยการจัดเก็บแฟ้มข้อมูลอย่างเป็นเอกเทศและกระจายของระบบแฟ้มข้อมูล เมื่อพิจารณาดูแล้วจะพบว่าระบบแฟ้มข้อมูลต่าง ๆ มีความปลอดภัยและความคล่องตัวสูง เนื่องจากมีขนาดเล็กและแยกเก็บภายในแต่ละหน่วยงาน แต่ในขณะเดียวกัน ได้ก่อให้เกิดปัญหาต่าง ๆ ขึ้นเช่นเดียวกัน ดังนี้

2.2.4.1 Data Redundancy

เป็นปัญหาที่เกิดขึ้นเนื่องจากการจัดเก็บข้อมูลที่ซ้ำซ้อนกัน กล่าวคือ ข้อมูลชุดเดียวกันถูกจัดเก็บอยู่ใน 2 แฟ้มข้อมูลหรือมากกว่า เช่น ข้อมูลของพนักงานที่ชื่อ "ดวงใจ" และ "กนกวรรณ" ซึ่งถูกจัดเก็บอยู่ในทั้งแฟ้มข้อมูลพนักงาน (CUSTOMER) ของฝ่ายการพนักงาน และแฟ้มข้อมูลพนักงานขาย (SALESMAN) ของฝ่ายการตลาด ทั้ง ๆ ที่ข้อมูลทั้งสองเป็นข้อมูลเดียวกัน เนื่องจากพนักงานขายถือว่าเป็นพนักงานคนหนึ่งของบริษัท ดังตารางที่ 2.2 และ ตารางที่ 2.3

ตารางที่ 2.2 ปัญหาจากการเก็บข้อมูลซ้ำซ้อนจากแฟ้มข้อมูลพนักงาน

EMPLOYEE

001	นางแพง พลเมืองดี	14/25 บางพลัด กรุงเทพมหานคร	10,000	441-0036
002	นางสุดใจ แสนดี	108/44 ยานนาวา กรุงเทพมหานคร	15,000	236-0123
003	นส.ดวงใจ แสนประเสริฐ	99 บางซื่อ กรุงเทพมหานคร	8,000	423-5575
004	นายเกษม สำราญใจ	2/77 ลาดกระบัง กรุงเทพมหานคร	25,000	222-4000
005	นางกนกวรรณ เกิดสุข	130/88 บางแค กรุงเทพมหานคร	9,500	441-8523

ตารางที่ 2.3 ปัญหาจากการเก็บข้อมูลซ้ำซ้อนจากแฟ้มข้อมูลพนักงานขาย

SALESMAN

S001	นส.ดวงใจ แสนประเสริฐ	423-5575
S002	นางกนกวรรณ เกิดสุข	441-8523

ซึ่งการซ้ำซ้อนของข้อมูลในลักษณะนี้ จะส่งผลให้เสียพื้นที่ในการจัดเก็บไป

2.2.4.2 Data Inconsistency

เป็นปัญหาที่เป็นผลกระทบมาจากการจัดเก็บข้อมูลที่มีความซ้ำซ้อน เนื่องจากการที่มีข้อมูลชุดเดียวกันจัดเก็บอยู่ในหลายแฟ้มข้อมูล อาจทำให้เกิดข้อมูลชุดเดียวกันมีค่าที่ต่าง กันในต่างแฟ้มข้อมูลได้ ส่งผลทำให้ไม่ทราบว่าข้อมูลชุดใดคือข้อมูลชุดที่ถูกต้อง เช่น ในกรณีที่มีการเปลี่ยนหมายเลขโทรศัพท์ของพนักงานที่ชื่อ “ดวงใจ” เฉพาะในแฟ้มข้อมูลพนักงาน (EMPLOYEE) จะส่งผลให้หมายเลขโทรศัพท์ของพนักงานคนเดียวกันในแฟ้มข้อมูลพนักงานขาย (Field “SALE_PHONE”) และแฟ้มข้อมูลพนักงาน (Field “EMP_TEL”) ไม่ตรงกัน ดังนั้นเมื่อต้องการทราบหมายเลขโทรศัพท์ของพนักงานคนดังกล่าว ก็จะไม่ทราบว่าหมายเลขโทรศัพท์ใดเป็นหมายเลขที่ถูกต้องดัง ตารางที่ 2.4 และ ตารางที่ 2.5

ตารางที่ 2.4 ปัญหาการเปลี่ยนข้อมูลจากแฟ้มข้อมูลพนักงาน

EMPLOYEE

001	นางแพง พลเมืองดี	14/25 บางพลัด กรุงเทพมหานคร	10,000	441-0036
002	นางสุดใจ แสนดี	108/44 ยานนาวา กรุงเทพมหานคร	15,000	236-0123
003	นส.ดวงใจ แสน ประเสริฐ	99 บางซื่อ กรุงเทพมหานคร	8,000	424-5000
004	นายเกษม สำราญใจ	12/77 ลาดกระบัง กรุงเทพมหานคร	25,000	222-4000
005	นางกนกวรรณ เกิดสุข	130/88 บางแค กรุงเทพมหานคร	9,500	441-8523

ตารางที่ 2.5 ปัญหาการเปลี่ยนข้อมูลจากแฟ้มข้อมูลพนักงานขาย

SALESMAN

S001	นส.ดวงใจ แสน ประเสริฐ	423-5575
S002	นางกนกวรรณ เกิดสุข	441-8523

2.2.4.3 Data Anomaly

เป็นปัญหาที่เป็นผลกระทบมาจากการจัดเก็บข้อมูลที่มีความซ้ำซ้อนอีกลักษณะหนึ่ง เนื่องจากการที่มีข้อมูลชุดเดียวกัน ถูกจัดเก็บอยู่ในหลายแฟ้มข้อมูล อาจส่งผลให้ความสัมพันธ์ระหว่างข้อมูลในแฟ้มข้อมูลต่าง ๆ สูญเสียไปในกรณีที่มีการเพิ่ม ลบ หรือเปลี่ยนแปลงค่าของข้อมูลชุดเดียวกันในแฟ้มข้อมูลต่าง ๆ ที่สัมพันธ์กันไม่ครบถ้วน ซึ่งการสูญเสียความสัมพันธ์ระหว่างแฟ้มข้อมูลนี้ เกิดขึ้นได้ 3 ลักษณะ ดังนี้

1) Modification Anomaly เป็นการเปลี่ยนแปลงค่าของข้อมูลในแฟ้มข้อมูลต่าง ๆ ที่สัมพันธ์กันไม่ครบถ้วน เช่น มีการเปลี่ยนแปลงชื่อของพนักงานที่ชื่อ "ดวงใจ" ไปเป็น "ดวงพร" ในแฟ้มข้อมูลพนักงาน โดยไม่มีการเปลี่ยนชื่อของพนักงานคนเดียวกันในแฟ้มข้อมูลพนักงานขาย ดังตารางที่ 2.6 และตารางที่ 2.7

ตารางที่ 2.6 ปัญหาการจัดเก็บข้อมูลหลายแฟ้มข้อมูลจากแฟ้มข้อมูลพนักงาน

EMPLOYEE

001	นางแพง พลเมืองดี	14/25 บางพลัด กรุงเทพมหานคร	10,000	441-0036
002	นางสุดใจ แสนดี	108/44 ยานนาวา กรุงเทพมหานคร	15,000	236-0123
003	นส.ดวงพร แสน ประเสริฐ	99 บางซื่อ กรุงเทพมหานคร	8,000	424-5000
004	นายเกษม สำราญใจ	12/77 ลาดกระบัง กรุงเทพมหานคร	25,000	222-4000
005	นางกนกวรรณ เกิดสุข	130/88 บางแค กรุงเทพมหานคร	9,500	441-8523

ตารางที่ 2.7 ปัญหาการจัดเก็บข้อมูลหลายแฟ้มข้อมูลจากแฟ้มข้อมูลพนักงานขาย

SALESMAN

S001	นส.ดวงใจ แสน ประเสริฐ	424-5000
S002	นางกนกวรรณ เกิดสุข	441-8523

การเปลี่ยนแปลงในลักษณะนี้ จะส่งผลให้พนักงานที่เดิมชื่อ "ดวงใจ" ซึ่งเปลี่ยนชื่อเป็น "ดวงพร" และพนักงานขายที่ชื่อ "ดวงใจ" ไม่ใช่คนเดียวกัน

2) Insertion Anomaly เป็นการกำหนดข้อมูลเพิ่มเติมให้กับแฟ้มข้อมูลต่าง ๆ ที่สัมพันธ์กันไม่ครบถ้วน เช่น มีการเพิ่มข้อมูลพนักงานขายคนใหม่ที่ชื่อ "สมบุญ" เฉพาะแฟ้มข้อมูลพนักงานขาย โดยไม่ได้เพิ่มข้อมูลลงในแฟ้มข้อมูลพนักงานด้วยดังตารางที่ 2.8 และ ตารางที่ 2.9

ตารางที่ 2.8 ปัญหาการเพิ่มเติมข้อมูลอย่างไม่ทั่วถึงในแฟ้มข้อมูลพนักงาน
EMPLOYEE

001	นางแพง พลเมืองดี	14/25 บางพลัด กรุงเทพมหานคร	10,000	441-0036
002	นางสุดใจ แสนดี	108/44 ยานนาวา กรุงเทพมหานคร	15,000	236-0123
003	นส.ดวงพร แสน ประเสริฐ	99 บางซื่อ กรุงเทพมหานคร	8,000	424-5000
004	นายเกษม สำราญใจ	12/77 ลาดกระบัง กรุงเทพมหานคร	25,000	222-4000
005	นางกนกวรรณ เกิดสุข	130/88 บางแค กรุงเทพมหานคร	9,500	441-8523

ตารางที่ 2.9 ปัญหาการเพิ่มเติมข้อมูลอย่างไม่ทั่วถึงในแฟ้มข้อมูลพนักงานขาย

SALESMAN

รหัสพนักงาน	ชื่อพนักงาน	หมายเลข
S001	นส.ดวงพร แสนประเสริฐ	424-5000
S002	นางกนกวรรณ เกิดสุข	441-8523
S003	นางสมบุญใจดี	424-8899

การเพิ่มเติมข้อมูลในลักษณะนี้ จะส่งผลให้พนักงานขายที่ชื่อ "สมบุญใจ" ไม่ใช่พนักงานของบริษัท

3) Deletion Anomaly เป็นการลบข้อมูลจากแฟ้มข้อมูลต่าง ๆ ที่สัมพันธ์กันไม่ครบถ้วน เช่น กรณีที่พนักงานที่ชื่อ "กนกวรรณ" ลาออก แล้วมีการลบข้อมูลของพนักงานคนนั้นเฉพาะแฟ้มข้อมูลพนักงาน โดยไม่ได้ลบข้อมูลของพนักงานคนเดียวกันในแฟ้มข้อมูลพนักงานขาย ดังตารางที่ 2.10 และ ตารางที่ 2.11

ตารางที่ 2.10 การลบข้อมูลอย่างไม่ทั่วถึงในแฟ้มข้อมูลพนักงาน

EMPLOYEE

รหัสพนักงาน	ชื่อพนักงาน	ตำแหน่ง	เงินเดือน	หมายเลข
001	นางแพง พลเมืองดี	14/25 บางพลัด กรุงเทพมหานคร	10,000	441-0036
002	นางสุดใจ แสนดี	108/44 ยานนาวา กรุงเทพมหานคร	15,000	236-0123
003	นส.ดวงพร แสน ประเสริฐ	99 บางซื่อ กรุงเทพมหานคร	8,000	424-5000
004	นายเกษม สำราญใจ	12/77 ลาดกระบัง กรุงเทพมหานคร	25,000	222-4000

ตารางที่ 2.11 การลบข้อมูลอย่างไม่ทั่วถึงในแฟ้มข้อมูลพนักงานขาย

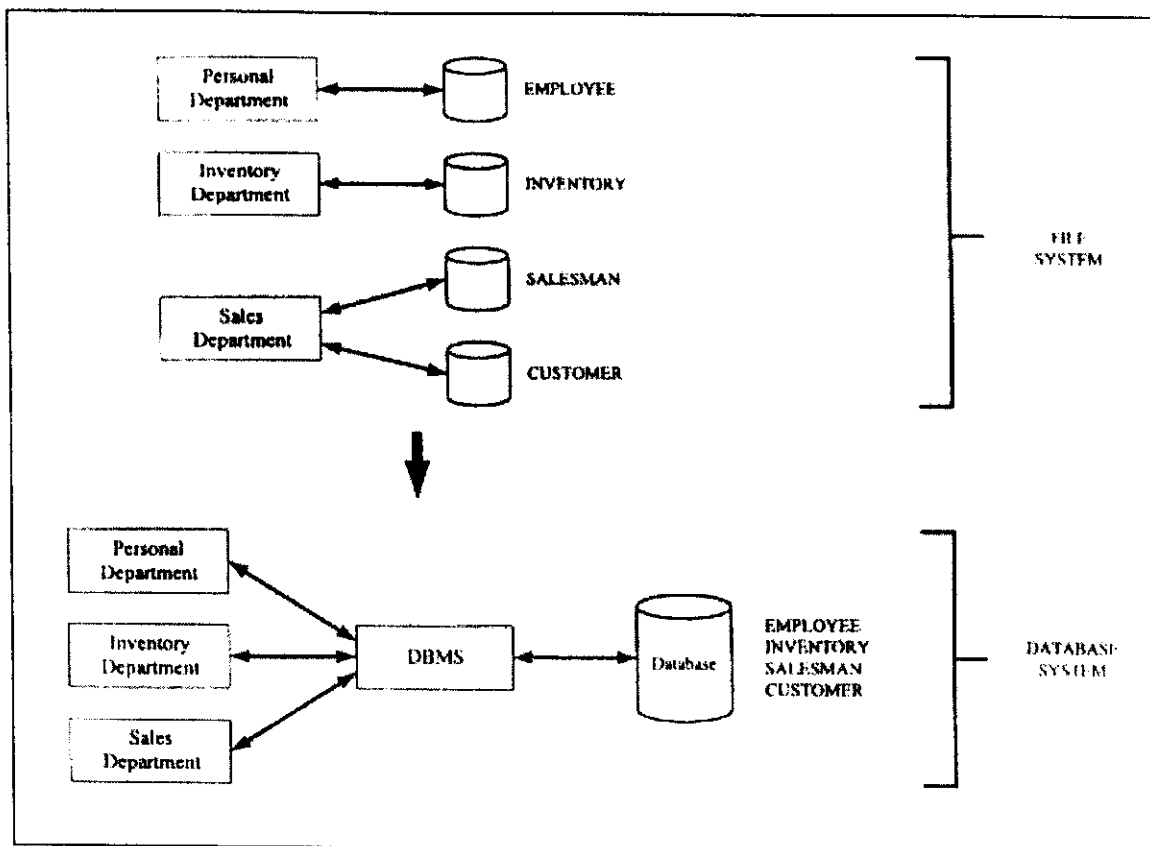
SALESMAN

รหัส	ชื่อพนักงานขาย	เบอร์โทรศัพท์
S001	นส.ดวงพร แสนประเสริฐ	424-5000
S002	นางกนกวรรณ เกิดสุข	441-8523
S003	นางสมบุญ ใจดี	424-8899

การลบข้อมูลในลักษณะนี้ จะส่งผลให้พนักงานที่ชื่อ "กนกวรรณ" ยังคงปรากฏเป็นพนักงานขายของบริษัทอยู่ผลกระทบทั้ง 3 ลักษณะนี้ จะก่อให้เกิด "ข้อมูลที่ไม่มีความสัมพันธ์กับแฟ้มข้อมูลใด เกิดขึ้นในแฟ้มข้อมูลใดแฟ้มข้อมูลหนึ่ง" โดยข้อมูลในลักษณะนี้ จะถือว่าเป็นข้อมูลที่ไม่มีประโยชน์ และเป็นการใช้เนื้อที่ในการจัดเก็บอย่างเปล่าประโยชน์ เช่น กรณีที่ปรากฏพนักงานที่ชื่อ "กนกวรรณ" ปรากฏเป็นพนักงานขายของบริษัทอยู่ ทั้ง ๆ ที่พนักงานคนนี้ได้ลาออกไปแล้ว (กิตติ-จำลอง , 2544.หน้า 525-528)

2.3 ระบบฐานข้อมูล (Database System)

จากปัญหาต่าง ๆ ที่เกิดขึ้นในระบบแฟ้มข้อมูล ได้ก่อให้เกิดการจัดเก็บข้อมูลในรูปแบบใหม่ขึ้นที่เรียกว่า "ฐานข้อมูล (Database)" การจัดเก็บข้อมูลในฐานข้อมูลนี้จะแตกต่างจากการจัดเก็บข้อมูลแบบแฟ้มข้อมูล เนื่องจากฐานข้อมูลเป็นการนำเอาข้อมูลต่าง ๆ ที่มีความสัมพันธ์กัน ซึ่งแต่เดิมจัดเก็บอยู่ในแต่ละแฟ้มข้อมูลมาจัดเก็บไว้ในที่เดียวกัน เช่น ข้อมูลพนักงาน สินค้าคงคลัง พนักงานขาย และลูกค้า ซึ่งแต่เดิมถูกจัดเก็บอยู่ในรูปของแฟ้มข้อมูลของฝ่ายต่าง ๆ ได้ถูกนำมาจัดเก็บรวมกันไว้ในฐานข้อมูลเดียว ซึ่งเป็นฐานข้อมูลรวมของบริษัท ส่งผลให้แต่ละฝ่ายสามารถใช้ข้อมูลร่วมกัน และสามารถแก้ไขปัญหาต่าง ๆ ที่เกิดขึ้นในระบบแฟ้มข้อมูลได้ ดังรูปที่



รูปที่ 2.3 ระบบฐานข้อมูล

ข้อมูลต่าง ๆ ที่ถูกจัดเก็บเป็นฐานข้อมูล นอกจากจะต้องเป็นข้อมูลที่มีความสัมพันธ์กันแล้ว ยังจะต้องเป็นข้อมูลที่ใช้สนับสนุนการดำเนินงานอย่างน้อยอย่างใดอย่างหนึ่งขององค์กร ดังนั้นจึงอาจกล่าวได้ว่าแต่ละฐานข้อมูลจะเทียบเท่ากับระบบแฟ้มข้อมูล 1 ระบบ และจะเรียกฐานข้อมูลที่จัดทำขึ้นเพื่อสนับสนุนการดำเนินงานอย่างใดอย่างหนึ่งนั้นว่า "ระบบฐานข้อมูล" (Database System) เช่น ระบบฐานข้อมูลเงินเดือน ซึ่งเป็นฐานข้อมูลที่จัดเก็บข้อมูลต่าง ๆ ที่สนับสนุนการคำนวณเงินเดือน หรือระบบฐานข้อมูลประชากร ซึ่งเป็นฐานข้อมูลที่จัดเก็บข้อมูลต่าง ๆ ที่สนับสนุนการจัดทำสำมะโนประชากร เป็นต้น หรือระบบฐานข้อมูล (Database System) คือ การนำเอาข้อมูลต่าง ๆ ที่มีความสัมพันธ์กันมาเก็บรวบรวมกันไว้ภายในฐานข้อมูลเดียว และสามารถให้ข้อมูลร่วมกันได้

ระบบฐานข้อมูล (Database System) คือ การจัดเก็บข้อมูลอย่างเป็นระบบซึ่งผู้ใช้สามารถเรียกใช้ข้อมูลดังกล่าวได้ในลักษณะต่างๆ เช่น การเพิ่มข้อมูล (Add Data) การเรียกใช้ข้อมูล (Retrieve Data) การแก้ไขและลบข้อมูล (Update & Delete Data) ตลอดจนการเคลื่อนย้ายข้อมูล (Move Data) ไปตามกำหนด

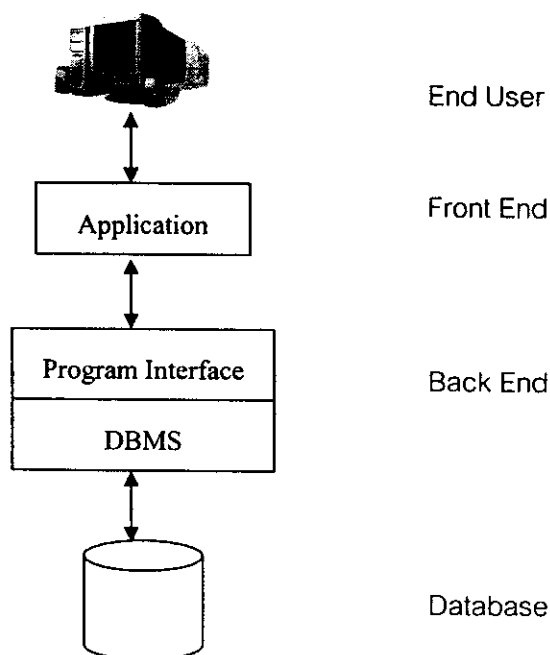
หรือระบบฐานข้อมูล อาจหมายถึง การจัดทำฐานข้อมูลขึ้นเพื่อสนับสนุนการดำเนินงานหรือกิจกรรม อย่างใดอย่างหนึ่ง

2.3.1 โครงสร้างของระบบฐานข้อมูล (Structure of database)

ระบบฐานข้อมูลสามารถแบ่งออก ตามลักษณะโครงสร้าง ซึ่งประกอบไปด้วยโครงสร้างหลัก 2 ส่วน ได้แก่ ส่วน Front End และ Back End

2.3.1.1 Front End เป็นโปรแกรมประยุกต์ (Application) ที่อาจจะสร้างจากภาษาต่างๆ ส่วนนี้โดยปกติ จะรองรับการทำงานของผู้ใช้ (End User) เพื่อทำหน้าที่ติดต่อกับระบบ

2.3.1.2 Back End เป็นส่วนที่ทำหน้าในการจัดการกับระบบฐานข้อมูลทั้งหมด ในแง่ของการจัดเก็บ และเรียกใช้ข้อมูลจากแหล่งข้อมูลจริง ได้แก่ การปฏิบัติการต่างๆกับข้อมูล, การจัดทำ Backup, การควบคุมความถูกต้องในการใช้ข้อมูลพร้อมกัน รวมไปถึงการควบคุมความปลอดภัยของระบบ เป็นต้น



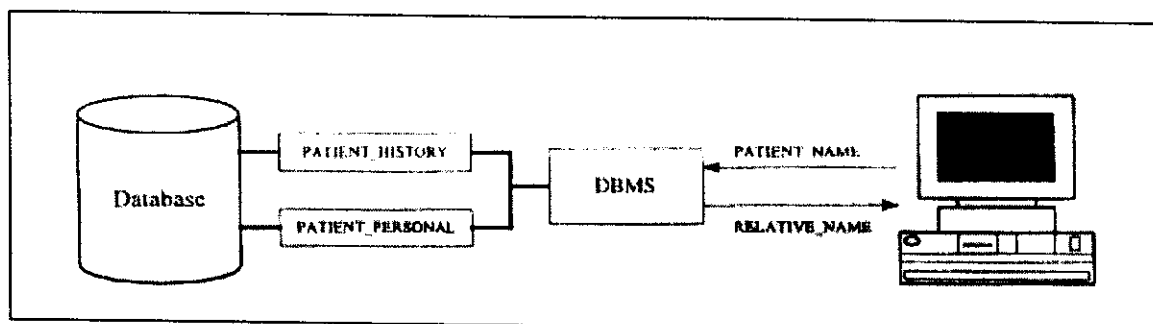
รูปที่ 2.4 โครงสร้างของระบบฐานข้อมูล

2.3.2 องค์ประกอบของระบบฐานข้อมูล

ระบบฐานข้อมูลโดยทั่วไป จะเกี่ยวข้องกับ 4 ส่วนหลัก ๆ ดังนี้

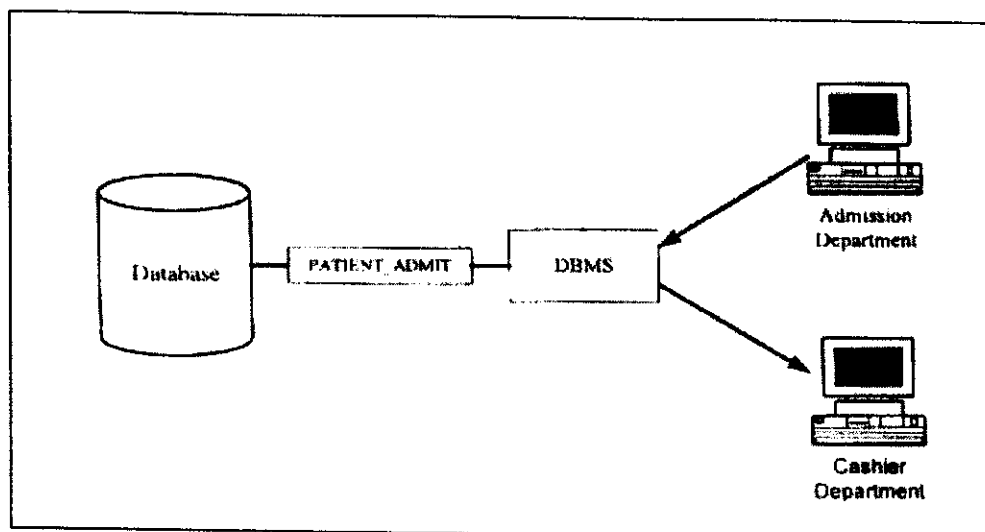
2.3.2.1 ข้อมูล (Data)

ข้อมูลที่จัดเก็บอยู่ในระบบฐานข้อมูล ไม่ว่าจะเป็นบนเครื่องคอมพิวเตอร์ส่วนบุคคล ไปจนถึงเครื่องคอมพิวเตอร์ขนาดใหญ่อย่างเช่นเครื่อง Mainframe ข้อมูลในแต่ละส่วนจะต้องสามารถนำมาใช้ประกอบกันได้ (Data Integrated) เช่น เมื่อแพทย์รักษาผู้ป่วยแพทย์จะอาศัยข้อมูลจากประวัติการรักษาพยาบาลของผู้ป่วย (PATIENT_HISTORY) มาประกอบการรักษา แต่ในกรณีฉุกเฉินที่ต้องการติดต่อญาติของผู้ป่วย ซึ่งข้อมูลในส่วนนี้ไม่ปรากฏอยู่ในประวัติการรักษาพยาบาล ทางโรงพยาบาลสามารถนำชื่อผู้ป่วย (Field "PATIENT_NAME") ไปค้นหาชื่อญาติ (Field "RELATIVE_NAME") ในทะเบียนผู้ป่วย (PATIENT_PERSONAL) ได้ โดยไม่จำเป็นต้องเก็บชื่อญาติของผู้ป่วยไว้ในประวัติการรักษาพยาบาลแต่อย่างใด รูปที่ 2.5



รูปที่ 2.5 ลำดับการเรียกดูข้อมูลบนเครื่องคอมพิวเตอร์

นอกเหนือจากคุณลักษณะนี้แล้ว ในเครื่องคอมพิวเตอร์ขนาดใหญ่ที่มีผู้ใช้จำนวนมาก ข้อมูลในฐานข้อมูลจะต้องสามารถถูกใช้ร่วมกัน (Data Sharing) จากผู้ใช้หลาย ๆ คน ได้ เช่น ข้อมูลในการจองห้องพักของผู้ป่วย (PATIENT_ADMIT) จะต้องสามารถนำไปใช้ในการออกใบเสร็จรับเงินเพื่อเก็บค่ารักษาพยาบาลโดยฝ่ายการเงินได้ในขณะเดียวกัน รูปที่ 2.6



รูปที่ 2.6_ ลำดับการเรียกดูข้อมูลบนเครื่องคอมพิวเตอร์พร้อมกันหลายเครื่อง

2.3.2.2 ฮาร์ดแวร์ (Hardware)

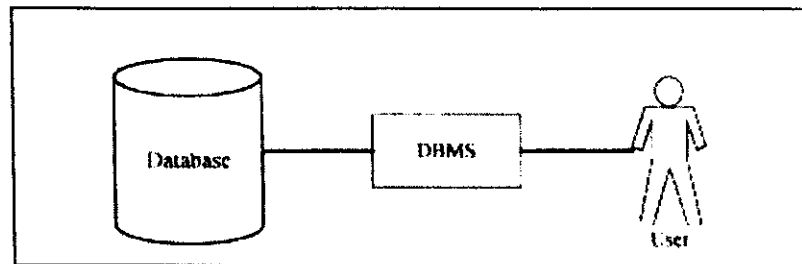
อุปกรณ์ทางคอมพิวเตอร์ที่มีส่วนเกี่ยวข้องกับระบบฐานข้อมูล จะประกอบด้วย ส่วนหลักดังนี้

1) หน่วยความจำสำรอง (Secondary Storage) เนื่องจากเป็นอุปกรณ์ทางคอมพิวเตอร์ที่ใช้จัดเก็บข้อมูลของฐานข้อมูล ดังนั้นสิ่งที่ต้องคำนึงถึงสำหรับอุปกรณ์ในส่วนนี้จึงได้แก่ ความจุของหน่วยความจำสำรองที่นำมาใช้จัดเก็บข้อมูลของฐานข้อมูลนั้น

2) หน่วยประมวลผล และหน่วยความจำหลัก เนื่องจากเป็นอุปกรณ์ที่จะต้องทำงานร่วมกัน เพื่อนำข้อมูลจากฐานข้อมูลขึ้นมาประมวลผลตามคำสั่งที่กำหนด ดังนั้นสิ่งที่ต้องคำนึงถึงสำหรับอุปกรณ์ในส่วนนี้ จึงได้แก่ ความเร็วของหน่วยประมวลผล และขนาดของหน่วยความจำหลักของเครื่องคอมพิวเตอร์ที่นำมาใช้ประมวลผลร่วมกับฐานข้อมูลนั้น

2.3.2.3 ซอฟต์แวร์ (Software)

ในการติดต่อกับข้อมูลภายในฐานข้อมูลของผู้ใช้ จะต้องกระทำผ่านโปรแกรมที่มีชื่อว่าโปรแกรม Database Management System (DBMS) ดังรูปที่ 2.7



รูปที่ 2.7 การติดต่อกับข้อมูลผ่าน DBMS

หน้าที่หลักของโปรแกรม DBMS ได้แก่ การทำให้การเรียกใช้ข้อมูลจากฐานข้อมูล เป็นอิสระจากส่วนของ Hardware หรือกล่าวอีกนัยหนึ่ง โปรแกรม DBMS จะมีหน้าที่ในการจัดการและควบคุมความถูกต้อง ความซ้ำซ้อนและความสัมพันธ์ระหว่างข้อมูลต่าง ๆ ภายในฐานข้อมูลแทนโปรแกรมเมอร์ ส่งผลให้ผู้ใช้สามารถที่จะเรียกใช้ข้อมูลจากฐานข้อมูลได้โดยไม่ต้องทราบถึงโครงสร้างทางกายภาพของข้อมูลในระดับที่ลึกเช่นเดียวกับโปรแกรมเมอร์ เนื่องจากโปรแกรม DBMS นี้ จะมีส่วนของ Query Language ซึ่งเป็นภาษาที่ประกอบด้วยคำสั่งต่าง ๆ ที่ใช้ในการจัดการ และเรียกใช้ข้อมูลจากฐานข้อมูล ซึ่งสามารถนำไปใช้ร่วมกับภาษาคอมพิวเตอร์อื่น ๆ เพื่อพัฒนาเป็นโปรแกรมที่ใช้สำหรับเรียกใช้ข้อมูลจากฐานข้อมูลมาประมวลผล สำหรับรายละเอียดของโปรแกรม DBMS จะกล่าวถึงในลำดับต่อไป

2.3.2.4 ผู้ใช้ระบบฐานข้อมูล (User)

ผู้ที่เรียกใช้ข้อมูลจากระบบฐานข้อมูลมาใช้งาน สามารถแบ่งออกเป็น 3 กลุ่มได้ดังนี้

1) Application Programmer ได้แก่ ผู้ที่ทำหน้าที่พัฒนาโปรแกรม (Application Program) เพื่อเรียกใช้ข้อมูลจากระบบฐานข้อมูลมาประมวลผล โดยโปรแกรมที่พัฒนาขึ้นส่วนใหญ่ มักจะใช้ร่วมกับคำสั่งในกลุ่ม Data Manipulation Language (DML) ของ Query Language เพื่อเรียกใช้ข้อมูลจากฐานข้อมูล

2) End User ได้แก่ ผู้ที่นำข้อมูลจากฐานข้อมูลไปใช้งาน ซึ่งแบ่งออกเป็น 2 กลุ่มดังนี้

- Naive Use ได้แก่ ผู้ใช้ที่เรียกใช้ข้อมูลจากฐานข้อมูลโดยอาศัยโปรแกรมที่พัฒนาขึ้น
- Sophisticated User ได้แก่ ผู้ใช้ที่เรียกใช้ข้อมูลจากฐานข้อมูลด้วยประโยคคำสั่งของ Query Language ซึ่งโดยทั่วไปผลิตภัณฑ์ทางด้านฐานข้อมูลที่จำหน่ายอยู่ใน

ห้องตลาดจะมีส่วนที่ยอมให้ผู้ใช้งาน ได้ใช้ประโยคคำสั่งของ Query Language เพื่อเรียกใช้ข้อมูลจากฐานข้อมูลได้โดยตรง สำหรับประโยคคำสั่งเหล่านี้จะถูกส่งผ่าน Query Processor ของโปรแกรม DBMS แปลงให้อยู่ในรูปของคำสั่งในกลุ่ม Data Manipulation Language

3) Database Administrator (DBA) ได้แก่ ผู้บริหารที่ทำหน้าที่ควบคุมและตัดสินใจในการกำหนดโครงสร้างของฐานข้อมูล ชนิดของข้อมูล วิธีการจัดเก็บข้อมูล รูปแบบในการเรียกใช้ข้อมูล ความปลอดภัยของข้อมูลและกฎระเบียบที่ใช้ควบคุมความถูกต้องของข้อมูลภายในฐานข้อมูล โดยอาศัยคำสั่งในกลุ่ม Data Definition Language (DDL) ซึ่งเป็นอีกส่วนหนึ่งของ Query Language เป็นตัวกำหนด (กิตติ-จำลอง ,2544. หน้า 530-532)

2.3.3 การออกแบบฐานข้อมูล

การออกแบบฐานข้อมูลที่ดีทำให้มั่นใจได้ว่าฐานข้อมูลของเราจะง่ายต่อการดูแลรักษา เราควรเก็บข้อมูลในตาราง และในตาราง ควรประกอบด้วยข้อมูลเพียงเรื่องเดียวเท่านั้น เช่น ลูกค้านั้นเมื่อปรับปรุงข้อมูลบางอย่าง เช่น ที่อยู่เพียงที่เดียว ก็จะเป็นการเปลี่ยนแปลงลักษณะที่ปรากฏโดยอัตโนมัติตลอดทั้งฐานข้อมูล

ก่อนที่จะใช้ Microsoft Access สร้างตาราง แบบสอบถาม ฟอรั่ม และวัตถุอื่นๆ ตามปกติ ควรจะร่างเค้าโครง และออกแบบงานของลงบนกระดาษก่อน และยังสามารถตรวจสอบฐานข้อมูลที่ออกแบบมาอย่างดีว่าคล้ายกับสิ่งที่ได้ออกแบบไว้หรือไม่ หรือสามารถเปิดฐานข้อมูลตัวอย่าง Northwind จากนั้นเปิดหน้าต่างความสัมพันธ์ขึ้นเพื่อตรวจสอบการออกแบบก็ได้

หลักปฏิบัติพื้นฐานสำหรับการออกแบบฐานข้อมูลที่ดี เพื่อประสิทธิภาพในการเข้าถึงข้อมูลได้อย่างถูกต้องแม่นยำ และรวดเร็ว จะประกอบไปด้วยดังนี้

- 1) กำหนดวัตถุประสงค์ของฐานข้อมูลให้ชัดเจน
- 2) กำหนดเขตข้อมูลที่ต้องการในฐานข้อมูล
- 3) กำหนดตารางที่ต้องการในฐานข้อมูล
- 4) กำหนดแต่ละเขตข้อมูลให้กับตาราง
- 5) ระบุเขตข้อมูลที่มีค่าไม่ซ้ำกันในแต่ละระเบียน
- 6) กำหนดความสัมพันธ์ระหว่างตาราง
- 7) แกะไขการออกแบบ
- 8) ป้อนข้อมูลและสร้างวัตถุฐานข้อมูลอื่นๆ
- 9) ใช้เครื่องมือวิเคราะห์ของ Microsoft Access



2.3.3.1 กำหนดวัตถุประสงค์ของฐานข้อมูลให้ชัดเจน

จะเป็นขั้นตอนแรกในการออกแบบฐานข้อมูล คือ การกำหนดวัตถุประสงค์ของฐานข้อมูล และลักษณะการใช้งานฐานข้อมูล มีขั้นตอนดังนี้

- พูดยคุยกับบุคคลที่จะใช้ฐานข้อมูลนี้ ระดมความคิดเกี่ยวกับคำถามต่างๆ ที่ต้องการและที่บุคคลเหล่านี้ต้องการให้ฐานข้อมูลเป็นคำตอบ

- ร่างรายงานที่ต้องการให้ฐานข้อมูลสร้าง
- รวบรวมรูปแบบที่ใช้บันทึกข้อมูลในปัจจุบัน

ในการที่จะกำหนดวัตถุประสงค์ของฐานข้อมูล รายการข้อมูลที่ต้องการจากฐานข้อมูลจะเริ่มปรากฏเป็นรูปร่างขึ้นมา จากนั้นสามารถกำหนดได้ว่าข้อเท็จจริงอะไรบ้างที่จำเป็นต้องเก็บไว้ในฐานข้อมูลตามวัตถุประสงค์ได้อย่างชัดเจน

2.3.3.2 กำหนดเขตข้อมูลที่ต้องการในฐานข้อมูล

ในแต่ละเขตข้อมูลจะเป็นข้อเท็จจริงเกี่ยวกับหัวข้อเฉพาะ ตัวอย่างเช่น ต้องการเก็บข้อเท็จจริงต่อไปนี้เกี่ยวกับลูกค้า ได้แก่ ชื่อของบริษัท ที่อยู่ เมือง รัฐ และหมายเลขโทรศัพท์ จึงจำเป็นต้องสร้างเขตข้อมูลแยกกันสำหรับข้อเท็จจริง ซึ่งแต่ละข้อเหล่านี้ เมื่อทำการกำหนดเขตข้อมูลที่ต้องการให้นึกถึงหลักการออกแบบ ดังต่อไปนี้

- รวบรวมข้อมูลทั้งหมดที่จำเป็นต้องใช้เท่านั้น
- เก็บข้อมูลในส่วนแบบตรรกะที่เล็กที่สุด ตัวอย่างเช่น ชื่อพนักงานมักจะถูกแยกเป็นเขตข้อมูลสองเขตข้อมูล คือ First Name และ Last Name เพื่อให้เป็นเรื่องง่ายในการเรียงลำดับข้อมูลตาม Last Name
- อย่าสร้างเขตข้อมูลสำหรับข้อมูลที่ประกอบด้วยรายการที่มีหลายหัวข้อ ตัวอย่างเช่น ในตาราง Supplies ซึ่งถ้าสร้างเขตข้อมูล Products ที่ประกอบด้วยเครื่องหมายจุลภาคคั่นแต่ละผลิตภัณฑ์ที่ได้รับ จะทำให้เป็นการยากต่อการค้นหาโดยเฉพาะผู้จัดจำหน่ายที่ขายสินค้าที่ต้องการ
- อย่ารวบรวมข้อมูลที่รับมา หรือข้อมูลที่ได้จากการคำนวณ (ข้อมูลที่เป็นผลลัพธ์ของนิพจน์) ตัวอย่างเช่น ถ้ามีเขตข้อมูล Unit Price และเขตข้อมูล Quantity อย่างสร้างเขตข้อมูลเพิ่มเติมที่เป็นผลคูณของค่าในเขตข้อมูลทั้งสองนี้
- อย่าสร้างเขตข้อมูลที่คล้ายกัน ตัวอย่างเช่น ในตาราง Supplies ถ้าสร้างเขตข้อมูล Product1, Product2 และ Product3 จะเป็นการยุ่งยากในการค้นหาผู้จัดจำหน่ายที่ขายผลิตภัณฑ์ที่เจาะจง ซึ่งจะต้องเปลี่ยนแปลงการออกแบบฐานข้อมูล และถ้าผู้จัดจำหน่ายมี

ผลิตภัณฑ์มากกว่าสามชนิด จำเป็นต้องใช้เพียงเขตข้อมูลเดียวสำหรับผลิตภัณฑ์เท่านั้น ซึ่งจะต้องใส่เขตข้อมูลนั้นลงในตาราง Products แทนการใส่ลงในตาราง Supplies

2.3.3.3 กำหนดตารางที่ต้องการในฐานข้อมูล

ควรกำหนดตารางที่ต้องการในฐานข้อมูลให้ครอบคลุมตามจุดประสงค์ และในแต่ละตารางควรประกอบด้วยข้อมูลหนึ่งหัวข้อ พร้อมด้วยรายการของเขตข้อมูล ก็ต้องใช้ข้อมูลที่เป็นสำหรับตารางด้วย ตัวอย่างเช่น ถ้ามีเขตข้อมูล Hire Date ซึ่งมีหัวข้อเป็นพนักงานแล้ว หัวข้อนี้จะอยู่ในตาราง Employees ซึ่งอาจจะมีตาราง Customers 1 ตาราง ตาราง Products 1 ตาราง และตาราง Orders 1 ตารางอีกด้วย เป็นต้น

2.3.3.4 กำหนดแต่ละเขตข้อมูลให้กับตาราง

เมื่อกำหนดแต่ละเขตข้อมูลให้กับตาราง ให้อาศัยหลักการออกแบบดังต่อไปนี้

- เพิ่มแต่ละเขตข้อมูลลงในตารางเพียงตารางเดียว
- อย่าเพิ่มเขตข้อมูลลงในตาราง หากเป็นการทำให้ข้อมูลเดียวกันปรากฏ

ในหลายระดับเขียนในตารางนั้น เพราะตารางจะประกอบด้วยข้อมูลที่มีการซ้ำซ้อนกันจำนวนมาก เขตข้อมูลนั้นอาจจะอยู่ในตารางที่ไม่ถูกต้องก็ได้ ควรเพิ่มข้อมูลผ่านฟอร์มแทน

ตัวอย่างเช่น ถ้าใส่เขตข้อมูลที่ประกอบด้วยที่อยู่ของลูกค้าลงในตาราง Orders ข้อมูลนั้นอาจมีการซ้ำกันได้มากกว่าหนึ่งระเบียบ เนื่องจากลูกค้าอาจปรากฏในใบสั่งซื้อได้มากกว่าหนึ่งใบก็เป็นได้

เมื่อข้อมูลแต่ละชิ้นมีการเก็บเพียงหนึ่งครั้ง สามารถปรับปรุงข้อมูลนั้นได้ในทีเดียว ซึ่งวิธีนี้เป็นวิธีที่มีประสิทธิภาพขึ้นและยังกำจัดความเป็นไปได้ในการซ้ำซ้อนกันของข้อมูลที่ต่างกันด้วย

2.3.3.5 ระบุเขตข้อมูลที่มีค่าไม่ซ้ำกันในแต่ละระเบียบ

เมื่อต้องการใช้ Microsoft Access เชื่อมโยงข้อมูลที่เก็บไว้ในตารางที่แยกจากกัน ตัวอย่างเช่น เมื่อต้องการเชื่อมต่อลูกค้ารายหนึ่งกับใบสั่งซื้อทั้งหมดของลูกค้ารายนั้น ในแต่ละตารางในฐานข้อมูลจะต้องมีเขตข้อมูลหรือชุดของเขตข้อมูลเช่นนี้เรียกว่าคีย์หลัก ซึ่งในแต่ละตารางจะต้องมีคีย์หลักอย่างน้อย 1 คีย์ (ต้องอาศัยความสังเกตในการเลือกคีย์หลักที่จะต้องไม่มีระเบียบไม่ซ้ำกันเท่านั้น)

2.3.3.6 กำหนดความสัมพันธ์ระหว่างตาราง

เมื่อทำการแบ่งข้อมูลลงในตาราง และระบุเขตข้อมูลที่เป็นคีย์หลักแล้ว จำต้องมียวิธีเพื่อให้ Microsoft Access นำข้อมูลที่สัมพันธ์กันกลับมารวมกันอีกครั้ง อย่างมีความหมาย เมื่อต้องการทำเช่นนี้ ให้กำหนดความสัมพันธ์ระหว่างตารางขึ้น

การสร้างความสัมพันธ์ระหว่างตารางจะเป็นการสร้างลักษณะที่เชื่อมโยงข้อมูลแต่ละตารางเข้าหากัน เพื่อเพิ่มประสิทธิภาพการเข้าถึงข้อมูลนั่นเอง

อาจพบว่าเป็นประโยชน์อย่างมากที่จะแสดงความสัมพันธ์ในฐานข้อมูลที่ได้รับการออกแบบมาอย่างดีที่มีอยู่แล้ว เช่น ฐานข้อมูลตัวอย่าง Northwind จากตัวอย่างที่แนบมา กับโปรแกรม Microsoft Access

2.3.3.7 แกะไขการออกแบบ

หลังจากที่ออกแบบตาราง เขตข้อมูล และความสัมพันธ์ที่ต้องการแล้ว ถึงเวลาที่จะต้องศึกษาถึงการออกแบบ และการป้องกันความผิดพลาดใดๆ ที่อาจจะยังมีอยู่ ซึ่งการเปลี่ยนแปลงการออกแบบฐานข้อมูลในขณะนี้ จะง่ายกว่าการเปลี่ยนแปลงหลังจากที่ได้ใส่ข้อมูลต่างๆ ลงไปในตารางแล้ว

ใช้ Microsoft Access จากการสร้างตารางระบุความสัมพันธ์ระหว่างตาราง และพร้อมใส่ข้อมูลตัวอย่างลงไปในตารางจนเพียงพอเพื่อให้สามารถทดสอบการออกแบบได้ เมื่อต้องการทดสอบความสัมพันธ์ในฐานข้อมูล ให้ดูว่าสามารถสร้างแบบสอบถามเพื่อคำตอบที่ต้องการได้หรือไม่ โดยให้สร้างฟอร์ม และรายงานคร่าวๆ แล้วดูว่าฟอร์ม และรายงานเหล่านี้แสดงข้อมูลตามที่เราคาดไว้หรือไม่ พร้อมทั้งแก้ไขการออกแบบจนกว่าความผิดพลาดเป็นศูนย์

2.3.3.8 ป้อนข้อมูลและสร้างวัตถุฐานข้อมูลอื่นๆ

เมื่อพึงพอใจกับโครงสร้างตาราง ตามหลักการออกแบบที่ได้อธิบายไว้แล้ว ก็ถึงเวลาป้อนข้อมูลทั้งหมดที่มีอยู่ลงไปในตาราง จากนั้นสามารถสร้างวัตถุฐานข้อมูลอื่นๆ ได้ เช่น แบบสอบถาม ฟอร์ม รายงาน Data Access Page แมโครและโมดูล ในระดับหนึ่งที่ว่าวัตถุทุกชิ้นควรได้ถูกใช้งานอย่างเต็มที่

2.3.3.9 ใช้เครื่องมือวิเคราะห์ของ Microsoft Access

Microsoft Access รวบรวมเครื่องมือไว้สองชนิด ซึ่งสามารถช่วยให้ปรับรูปแบบของฐานข้อมูล Microsoft Access ได้

- ตัววิเคราะห์ตาราง สามารถวิเคราะห์การออกแบบของตารางได้ครั้งละ 1 ตาราง สามารถแนะนำโครงสร้างตารางและความสัมพันธ์ใหม่ หากเหมาะสม และยังสามารถแบ่งตารางออกเป็นตารางใหม่สองตารางที่สัมพันธ์กัน ถ้าสมเหตุสมผลกัน

- ตัววิเคราะห์ประสิทธิภาพ สามารถวิเคราะห์ฐานข้อมูลทั้งหมดและให้คำแนะนำ หรือข้อเสนอแนะสำหรับการปรับปรุงฐานข้อมูลนั้น ตัวช่วยสร้างยังสามารถนำคำแนะนำ และข้อเสนอแนะเหล่านี้ไปใช้ได้อีกด้วย (วิทยา สงวนวรรณ ,2545. หน้า 56-61)

2.4 Data Independence

ในการเรียกใช้ข้อมูลที่จัดเก็บอยู่ในระบบแฟ้มข้อมูล จะต้องอาศัยโปรแกรมที่เขียนขึ้น เพื่อเรียกใช้ข้อมูลในแฟ้มข้อมูลนั้นโดยเฉพาะ เช่น เมื่อต้องการรายชื่อของพนักงานที่มีเงินเดือนมากกว่า 5,000 บาทต่อเดือน ผู้ใช้จะต้องให้โปรแกรมเมอร์จัดทำโปรแกรม เพื่ออ่านข้อมูลจากแฟ้มข้อมูลพนักงาน และพิมพ์รายงานที่แสดงเฉพาะข้อมูลที่ตรงตามเงื่อนไขที่กำหนด ดังนั้น เมื่อเกิดมีการเปลี่ยนแปลงโครงสร้างทางกายภาพของแฟ้มข้อมูลใดแฟ้มข้อมูลหนึ่ง จึงส่งผลให้โปรแกรมต่าง ๆ ที่เรียกใช้ข้อมูลในแฟ้มข้อมูลนั้น ต้องมีการเปลี่ยนแปลงตามไปด้วย เช่น กรณีที่มีการเปลี่ยนแปลงโครงสร้างของ Index File ของแฟ้มข้อมูลพนักงาน จากเดิมซึ่งเรียงลำดับตามรหัสพนักงานมาเป็นเรียงลำดับตามชื่อแทน รายงานที่แสดงรายชื่อพนักงานที่มีเงินเดือนมากกว่า 5,000 บาทต่อเดือน ซึ่งแต่เดิมนั้นกำหนดให้เรียงลำดับตามรหัสพนักงาน จึงไม่สามารถพิมพ์ได้ ส่งผลให้ต้องมีการแก้ไขโปรแกรมตามโครงสร้าง Index File ที่เปลี่ยนแปลงไป ซึ่งการที่ข้อมูลและโปรแกรมไม่เป็นอิสระต่อกันนี้ จะเรียกว่า "Data Dependence" หรือความเป็นอิสระของข้อมูล

ภายในระบบฐานข้อมูลไม่สามารถที่จะยอมให้ความไม่เป็นอิสระระหว่างข้อมูล และโปรแกรมเกิดขึ้นได้ เนื่องจาก 2 สาเหตุหลัก ๆ ดังนี้

2.4.1 เนื่องจากในฐานข้อมูล จะต้องไม่ปรากฏข้อมูลที่ซ้ำซ้อนกันเกิดขึ้น แต่ในแง่ความเป็นจริงแล้ว ข้อมูลที่ผู้ใช้ต้องการถึงแม้จะเป็นข้อมูลเดียวกัน ก็อาจต้องการรูปแบบของข้อมูลที่ต่างกันได้ เช่น ผู้ใช้ A ต้องการข้อมูลเงินเดือนในรูปแบบของ Binary ในขณะที่ผู้ใช้ B ต้องการข้อมูลเงินเดือนในรูปแบบของ Decimal แทน ซึ่งในกรณีเช่นนี้ จึงจำเป็นที่จะต้องทำให้ข้อมูลและโปรแกรมเป็นอิสระจากกัน เพื่อให้รูปแบบของข้อมูลที่จัดเก็บอยู่ในฐานข้อมูลมีเพียงรูปแบบเดียว แล้วจึงปล่อยให้เป็นที่ของโปรแกรม DBMS ในการแปลงรูปแบบเป็นไปตามรูปแบบที่ผู้ใช้แต่ละคนต้องการแทน

2.4.2 เนื่องจาก DBA มีสิทธิ์ที่จะเปลี่ยนแปลงโครงสร้างของข้อมูลภายในฐานข้อมูลได้ ดังนั้นจึงจำเป็นที่จะต้องทำให้ข้อมูลภายในฐานข้อมูลเป็นอิสระจากโปรแกรมต่าง ๆ ที่เรียกใช้ DBA สามารถเปลี่ยนแปลงโครงสร้างของข้อมูลภายในฐานข้อมูลได้ โดยไม่ส่งผลกระทบกับโปรแกรมต่าง ๆ

ด้วยเหตุนี้จึงได้มีการกำหนดให้ข้อมูลภายในฐานข้อมูล จะต้องเป็นอิสระจากตัวโปรแกรมที่เรียกใช้ เพื่อที่จะสามารถแก้ไขโครงสร้างทางกายภาพของข้อมูลได้ โดยไม่กระทบต่อโปรแกรมที่เรียกใช้ข้อมูลจากฐานข้อมูลนั้น ซึ่งคุณลักษณะเช่นนี้ จะเรียกว่า "Data Independence"

ในการกำหนดให้ข้อมูลเป็นอิสระจากโปรแกรมที่เรียกใช้ จะแบ่งออกเป็น 2 ระดับดังนี้

- ระดับ Physical

เป็นระดับที่โครงสร้างทางกายภาพของข้อมูลเป็นอิสระจากโปรแกรมที่เรียกใช้ เช่น สามารถเปลี่ยนแปลงโครงสร้างของ Index File ออกได้โดยไม่ต้องแก้ไขโปรแกรมที่เรียกใช้ข้อมูลนั้น

- ระดับ Logical

เป็นระดับที่ความสัมพันธ์ระหว่างข้อมูลในส่วนต่าง ๆ ภายในฐานข้อมูลเป็นอิสระจากโปรแกรมที่เรียกใช้ เช่น สามารถแยกบาง Field ออกไปเป็นแฟ้มข้อมูลใหม่ ได้โดยไม่ต้องแก้ไขโปรแกรมที่เรียกใช้ข้อมูลนั้น เมื่อพิจารณาความเป็นอิสระของข้อมูลที่มีต่อโปรแกรมที่เรียกใช้ในทุก 2 ระดับ จะสังเกตเห็นว่า การกำหนดให้ข้อมูลเป็นอิสระจากโปรแกรมที่เรียกใช้ในระดับ Logical จะกระทำได้ยากกว่าในระดับ Physical เนื่องจากความสัมพันธ์ระหว่างข้อมูลในส่วนต่าง ๆ ภายในแฟ้มข้อมูลเดียวกัน จะมีความเกี่ยวข้องกับโปรแกรมมากกว่าโครงสร้างทางกายภาพของข้อมูล (กิตติ-จำลอง , 2544.หน้า532-533)

2.5 Database Management System (DBMS)

เป็นโปรแกรมที่ทำหน้าที่เป็นตัวกลางในการติดต่อระหว่างผู้ใช้กับฐานข้อมูล เพื่อจัดการและควบคุมความถูกต้อง ความซ้ำซ้อน และความสัมพันธ์ระหว่างข้อมูลต่าง ๆ ภายในฐานข้อมูล ซึ่งต่างจากระบบแฟ้มข้อมูลที่หน้าที่เหล่านี้จะเป็นหน้าที่ของโปรแกรมเมอร์ ในการติดต่อกับข้อมูลในฐานข้อมูลไม่ว่าจะด้วยการใช้คำสั่งในกลุ่มคำสั่ง DML หรือ DDL หรือจะด้วยโปรแกรมต่าง ๆ ทุกคำสั่งที่ใช้กระทำกับข้อมูลจะถูกโปรแกรม DBMS นำมาแปล (Compile) เป็นการกระทำ (Operation) ต่าง ๆ ภายใต้คำสั่งนั้น ๆ เพื่อนำไปกระทำกับตัวข้อมูลภายในฐานข้อมูลต่อไป สำหรับส่วนการทำงานต่าง ๆ ภายในโปรแกรม DBMS ที่ทำหน้าที่ในการแปลคำสั่งไปเป็นการกระทำต่าง ๆ ที่จะกระทำกับข้อมูลนั้น ประกอบด้วยส่วนการทำงานต่าง ๆ ดังนี้

2.5.1 Database Manager

เป็นส่วนที่ทำหน้าที่กำหนดการกระทำต่าง ๆ ให้กับส่วน File Manager เพื่อไปกระทำกับข้อมูลที่เกิดขึ้นในฐานข้อมูล (File Manager เป็นส่วนที่ทำหน้าที่บริหาร และจัดการกับข้อมูลที่เกิดขึ้นในฐานข้อมูลในระดับกายภาพ)

2.5.2 Query Processor

เป็นส่วนที่ทำหน้าที่แปลงประโยคคำสั่งของ Query Language ให้อยู่ในรูปแบบของคำสั่งที่ Database Manager เข้าใจ

2.5.3 Data Manipulation Language Precompiler

เป็นส่วนหนึ่งที่ทำหน้าที่แปล (Compile) ประโยคคำสั่งของกลุ่มคำสั่ง DML ให้อยู่ในรูปแบบที่ส่วน Application Programs Object Code จะนำไปเข้ารหัสเพื่อส่งต่อไปยังส่วน Database Manager ในการแปลประโยคคำสั่งของกลุ่มคำสั่ง DML ของส่วน Data Manipulation Language Precompiler นี้จะต้องทำงานร่วมกับส่วน Query Processor

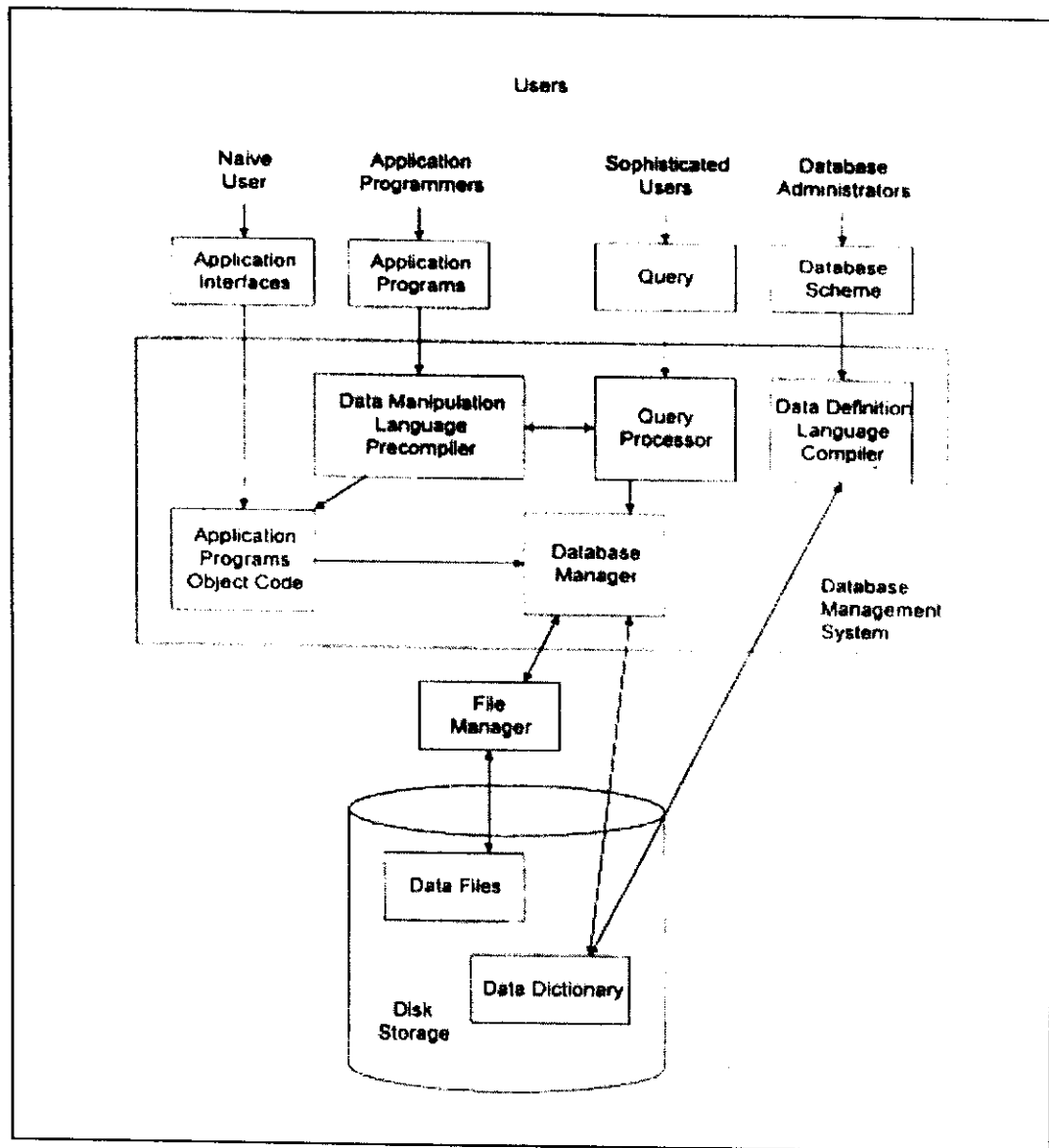
2.5.4 Data Definition Language Precompiler

เป็นส่วนที่ทำหน้าที่แปล (Compile) ประโยคคำสั่งของกลุ่มคำสั่ง DDL ให้อยู่ในรูปแบบของ MetaData ที่เก็บอยู่ในส่วน Data Dictionary ของฐานข้อมูล (MetaData ได้แก่ รายละเอียดที่บอกถึงโครงสร้างต่าง ๆ ของข้อมูล)

2.5.5 Application Programs Object Code

เป็นส่วนที่ทำหน้าที่แปลงคำสั่งต่าง ๆ ของโปรแกรม รวมทั้งคำสั่งในกลุ่มคำสั่ง DML ที่ส่งต่อมาจากส่วน Data Manipulation Language Precompiler ให้อยู่ในรูปของ Object Code ที่จะส่งต่อไปให้ Database Manager เพื่อกระทำกับข้อมูลในฐานข้อมูล

ทั้ง 5 ส่วนของโปรแกรม DBMS สามารถแสดงด้วยแผนภาพได้ดังรูปที่ 2.8



รูปที่ 2.8 ส่วนการทำงานต่างๆ ภายในโปรแกรม DBMS

โปรแกรม DBMS นี้ได้ถูกพัฒนาขึ้นมาเพื่อแก้ปัญหาทางด้าน Data Independence ที่ไม่มีในระบบแฟ้ม ดังนั้นจึงมีความเป็นอิสระจากทั้งตัว Hardware และตัวข้อมูลภายในฐานข้อมูล กล่าวคือ โปรแกรม DBMS จะมีการทำงานที่ไม่ขึ้นกับรูปแบบ (Platform) ของตัว Hardware ที่นำมาใช้กับระบบฐานข้อมูล รวมทั้งมีรูปแบบในการอ้างถึงข้อมูลที่ไม่ขึ้นอยู่กับการจัดโครงสร้างทางกายภาพของข้อมูล ด้วยการใช้ Query Language ในการติดต่อกับข้อมูลในฐานข้อมูลแทนคำสั่งของภาษาคอมพิวเตอร์ในยุคที่ 3 ส่งผลให้ผู้ใช้สามารถเรียกใช้ข้อมูลจากฐานข้อมูล โดยไม่

จำเป็นต้องทราบถึงประเภทของข้อมูลหรือขนาดของข้อมูลนั้น หรือสามารถกำหนดลำดับที่ของ Field ในการแสดงผลได้โดยไม่ต้องคำนึงถึงลำดับที่จริงของ Field นั้น

หน้าที่ของ DBMS

สำหรับหน้าที่ของโปรแกรม DBMS มีดังนี้

- 1) ทำหน้าที่แปลงคำสั่งที่ใช้จัดการกับข้อมูลภายในฐานข้อมูล ให้อยู่ในรูปแบบที่ฐานข้อมูลเข้าใจ
- 2) ทำหน้าที่ในการนำคำสั่งต่าง ๆ ซึ่งได้รับการแปลงแล้ว ไปส่งให้ฐานข้อมูลทำงาน เช่น การเรียกใช้ข้อมูล (Retrieve) การจัดเก็บข้อมูล (Update) การลบข้อมูล (Delete) การเพิ่มข้อมูล (Add) เป็นต้น
- 3) ทำหน้าที่ป้องกันความเสียหายที่จะเกิดขึ้นกับข้อมูลภายในฐานข้อมูล โดยจะคอยตรวจสอบว่าคำสั่งใดที่สามารถทำงานได้ และคำสั่งใดที่ไม่สามารถทำงานได้
- 4) ทำหน้าที่รักษาความสัมพันธ์ของข้อมูลภายในฐานข้อมูลให้มีความถูกต้องอยู่เสมอ
- 5) ทำหน้าที่เก็บรายละเอียดต่าง ๆ ที่เกี่ยวข้องกับข้อมูลภายในฐานข้อมูลไว้ใน Data Dictionary ซึ่งรายละเอียดเหล่านี้จึงมักจะถูกเรียกว่า "ข้อมูลของข้อมูล" (Metadata)
- 6) ทำหน้าที่ควบคุมให้ฐานข้อมูลทำงานได้อย่างถูกต้องและมีประสิทธิภาพ (กิตติ-จำลอง ,2544. หน้า 534-536)

2.6 Data Dictionary และ File Manager

ทุกฐานข้อมูลจะต้องมีส่วนที่ใช้เก็บข้อมูลในลักษณะ Metadata ซึ่งเป็นข้อมูลที่บอกถึงรายละเอียดของตัวข้อมูลที่เก็บอยู่ในฐานข้อมูล เช่น โครงสร้างของข้อมูล โครงสร้างของ Table โครงสร้างของ Index กฎที่ใช้ควบคุมความถูกต้องของข้อมูล (Integrity Rule) กฎที่ใช้ในการรักษาความปลอดภัยให้กับข้อมูล (Security Rule) ฯลฯ ข้อมูลเหล่านี้จัดเป็นข้อมูลที่มีความจำเป็นอย่างยิ่งต่อโปรแกรม DBMS ในการตัดสินใจที่จะดำเนินการใด ๆ กับฐานข้อมูล เช่น ข้อมูลที่เกี่ยวข้องกับกฎที่ใช้ในการรักษาความปลอดภัยให้กับข้อมูล จะถูกนำมาใช้ในการพิจารณาให้สิทธิแก่ผู้ใช้ในการใช้งานฐานข้อมูล เป็นต้น สำหรับส่วนที่ใช้จัดเก็บข้อมูลในลักษณะของ Metadata นี้ ได้แก่ Data Dictionary หรือ Catalog

สำหรับ File Manager เป็นส่วนที่ทำหน้าที่บริหารและจัดการกับข้อมูลที่เก็บอยู่ในฐานข้อมูลในระดับกายภาพ (Physical Level) (กิตติ-จำลอง ,2544.หน้า 536)

2.7 ผลกระทบของการประมวลผลด้วยระบบฐานข้อมูล

2.7.1 ข้อดีของการประมวลผลด้วยระบบฐานข้อมูล

- 1) สามารถลดความซ้ำซ้อนของข้อมูล (Data Redundancy) โดยไม่จำเป็นต้องจัดเก็บข้อมูลที่ซ้ำซ้อนกันไว้ในระบบแฟ้มข้อมูลของแต่ละหน่วยงานเหมือนเช่นเดิม แต่สามารถนำข้อมูลมาใช้ร่วมกันในคุณลักษณะ Integrated แทน
- 2) สามารถหลีกเลี่ยงความขัดแย้งของข้อมูล (Data Inconsistency) เนื่องจากไม่ต้องจัดเก็บข้อมูลที่ซ้ำซ้อนกันในหลายแฟ้มข้อมูล ดังนั้นการแก้ไขข้อมูลในแต่ละชุดจะไม่ก่อให้เกิดค่าที่แตกต่างกันได้
- 3) แต่ละหน่วยงานในองค์กร สามารถใช้ข้อมูลร่วมกันได้
- 4) สามารถกำหนดให้ข้อมูลมีรูปแบบที่เป็นมาตรฐานเดียวกันได้ เพื่อให้ผู้ใช้ข้อมูลในฐานข้อมูลชุดเดียวกัน สามารถเข้าใจและสื่อสารถึงความหมายเดียวกัน
- 5) สามารถกำหนดระบบความปลอดภัยให้กับข้อมูลได้ โดยกำหนดระดับความสามารถในการเรียกใช้ข้อมูลของผู้ใช้แต่ละคน ให้แตกต่างกันตามความรับผิดชอบ
- 6) สามารถรักษาความถูกต้องของข้อมูลได้ โดยระบุกฎเกณฑ์ในการควบคุมความผิดพลาดที่อาจเกิดขึ้นจากการป้อนข้อมูลผิด
- 7) สามารถตอบสนองต่อความต้องการใช้ข้อมูลในหลายรูปแบบ
- 8) ทำให้ข้อมูลเป็นอิสระจากโปรแกรมที่ใช้งานข้อมูลนั้น (Data Independence) ซึ่งส่งผลให้ผู้พัฒนาโปรแกรมสามารถแก้ไขโครงสร้างของข้อมูล โดยไม่กระทบต่อโปรแกรมที่เรียกใช้งานข้อมูลนั้น เช่น ในกรณีที่ต้องการเปลี่ยนขนาดของ Field สำหรับระบบแฟ้มข้อมูล จะกระทำได้ยาก เนื่องจากต้องเปลี่ยนแปลงตัวโปรแกรมที่อ้างถึง Field นั้นทั้งหมด ซึ่งต่างจากการใช้ระบบฐานข้อมูล ที่การอ้างถึงข้อมูลจะไม่ขึ้นอยู่กับการโครงสร้างทางกายภาพของข้อมูล จึงไม่ส่งผลให้ต้องแก้ไขโปรแกรมที่เรียกใช้ข้อมูลนั้นมากนัก

การจัดเก็บข้อมูลภายในฐานข้อมูลจะแตกต่างจากการจัดเก็บข้อมูลของระบบแฟ้มข้อมูล เนื่องจากในฐานข้อมูลนั้น ข้อมูลต่าง ๆ ที่สัมพันธ์กันจะถูกจัดเก็บไว้ในที่เดียวกัน ซึ่งต่างจากระบบแฟ้มข้อมูลที่ซึ่งข้อมูลต่าง ๆ จะถูกแยกจัดเก็บอยู่ในแต่ละแฟ้มข้อมูล ซึ่งด้วยวิธีนี้ส่งผลให้ข้อมูลภายในฐานข้อมูล สามารถที่จะแก้ไขปัญหาดังต่าง ๆ ไม่ว่าจะเป็นการซ้ำซ้อนของข้อมูล

ความไม่ถูกต้องของข้อมูล และการสูญเสียความสัมพันธ์ระหว่างข้อมูล ซึ่งเกิดขึ้นกับระบบเพิ่มข้อมูลได้ ในระบบฐานข้อมูลจะเกี่ยวข้องกับข้อมูล ฮาร์ดแวร์ ซอฟต์แวร์ และผู้ใช้ระบบฐานข้อมูล ข้อมูลที่จัดเก็บอยู่ในฐานข้อมูล จะมีความเป็นอิสระจากโปรแกรมที่เรียกใช้ จึงสามารถเปลี่ยนแปลงแก้ไขโครงสร้างของข้อมูลได้ง่าย สำหรับโปรแกรมที่ใช้ร่วมกับฐานข้อมูล ได้แก่ โปรแกรม DBMS ซึ่งทำหน้าที่ในการนำคำสั่งที่ใช้สำหรับเรียกใช้ข้อมูลของผู้ใช้ในระดับต่าง ๆ ไม่ว่าจะเป็น Naive Users, Application Programmers, Sophisticated Users และ Database Administrators มาแปลงเป็นการกระทำต่าง ๆ กับข้อมูลในฐานข้อมูล เพื่อให้ได้ผลลัพธ์ตามต้องการ

2.7.2 ข้อเสียของการประมวลผลด้วยระบบฐานข้อมูล

- 1) ขั้นตอนการออกแบบและการบำรุงรักษามีต้นทุนสูง เนื่องจากระบบต้องให้ผู้เชี่ยวชาญเฉพาะในการออกแบบระบบ ไม่ว่าจะเป็นทางด้าน Hardware และ Software รวมไปถึงราคาอุปกรณ์ที่ใช้มีราคาค่อนข้างสูง
- 2) ระบบมีความซับซ้อนจำเป็นต้องมีผู้ดูแลระบบที่ถูกต้องมาอย่างดี เพื่อรองรับสถานการณ์ที่ผิดพลาดอันอาจจะเกิดขึ้นได้
- 3) การเสี่ยงต่อการหยุดชะงักของระบบ เนื่องจากข้อมูลอาจถูกจัดเก็บแบบรวมศูนย์ (Centralized Database System) ความล้มเหลวของการทำงานบางส่วน อาจทำให้ระบบฐานข้อมูลโดยรวมหยุดชะงักการทำงานได้ (ณัฐพล อุ่นยัง ,2544. หน้า 22-23)

2.8 นิยามพื้นฐานในระบบฐานข้อมูล

Database มีคุณลักษณะที่สำคัญ 2 ประการ คือ ประการแรกจะต้องเป็นกลุ่มข้อมูลที่รวมเป็นหนึ่งเดียว (Integrated) และข้อมูลนี้จะถูกผู้ใช้สามารถเรียกใช้ร่วมกันได้

2.8.1 Data คือ ความเป็นจริง (Fact) ที่เกี่ยวกับบุคคล สถานที่ เหตุการณ์ หรือสิ่งของต่างๆ ซึ่งสามารถนับจำนวนได้

2.8.2 Information คือ ข้อมูลที่ถูกจัดรวบรวมให้อยู่ในรูปที่สามารถจะนำไปประกอบ การตัดสินใจ อย่างใดอย่างหนึ่งได้

2.8.3 Entity คือ สิ่งใดสิ่งหนึ่ง ได้แก่ ชื่อของบุคคล สถานที่ สิ่งของหรือการกระทำ ที่ต้องการจัดเก็บข้อมูลนั้นไว้ Entity ที่ใช้สำหรับแสดงความสัมพันธ์กัน ระหว่างข้อมูลในระบบ เช่น ลูกค้า ผู้ว่าจ้าง เป็นต้น สัญลักษณ์ที่ใช้แทน Entity คือ รูปสี่เหลี่ยมผืนผ้า

2.8.4 Attribute คือ รายละเอียดของข้อมูลใน Entity หนึ่งๆ ที่ใช้แสดงลักษณะและคุณสมบัติของ Entity ที่ถูกอ้างถึง เช่น Attribute ของ นิสิต ได้แก่ เลขประจำตัวนิสิต ชั้นปี คณะ ภาควิชา เป็นต้น ค่า Attribute คือค่าที่เก็บอยู่ใน Entity นั่นเองสัญลักษณ์ที่ใช้เขียนแทน คือ รูปวงรี

2.8.5 Entity Set คือ Entity หลายๆตัวที่มีค่าAttributeเหมือนกันและสามารถนำมารวมกันในรูปของTableเพื่อสะดวกในการเข้าถึงข้อมูลกลุ่มดังกล่าว เช่น Entity Set ของนิสิต เป็นต้น

2.8.6 Field เกิดจากการรวมตัวของข้อมูลที่เล็กที่สุดภายในคอมพิวเตอร์ เรียกว่า บิต (Bit) นำมาประกอบกันจะได้ข้อมูล เรียกว่า ไบต์ (Byte) หรือตัวอักษร (Character) หากนำอักขระมาประกอบกันเป็นกลุ่ม ก็จะได้ข้อมูลที่ขยายตัวออกเป็นรูปแบบใหม่ที่เรียกว่า Field และอาจกล่าวได้ว่า ส่วนของ Field ก็จะได้แก่ Attribute นั่นเอง

2.8.7 Record เกิดจากการรวมตัวของ Field หรือ Attribute ที่แสดงคุณสมบัติของ Entity ตัวใดตัวหนึ่ง

2.8.8 File คือ กลุ่มของ Record ชนิดเดียวกัน ที่ถูกนำมารวมกันเป็นหมวดหมู่ ข้อมูลที่อยู่ในไฟล์จะสามารถมองได้เป็น อาร์เรย์ 2 มิติ นั่นคือ ในรูปของแถวซึ่งแสดงถึงจำนวน Record และคอลัมน์ ซึ่งแทนค่าของ Attribute แต่ละตัวนั่นเอง

2.8.9 Association คือ สัญลักษณ์แสดงความสัมพันธ์กัน ระหว่าง Entity ซึ่งจะเกิดขึ้นได้กับ Entity ตั้งแต่ 2 ตัวขึ้นไป (ญัฐพล อุ่นยัง ,2544. หน้า 34-36)

2.9 Type of Entity Association

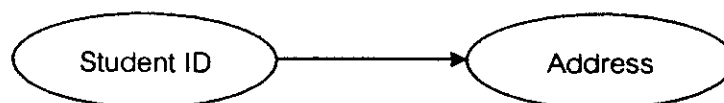
ค่าของ Entity แต่ละตัวจะถูกเก็บอยู่ในรูปของไฟล์ ในขณะเดียวกัน ค่าของ Attribute ก็จะได้แก่ ค่า Field นั่นเอง ส่วนความสัมพันธ์ ระหว่าง Entity ได้แก่ ความสัมพันธ์ระหว่างไฟล์ที่ถูกนำเสนอในรูปของการกำหนดค่าของ Field ในไฟล์หนึ่ง เพื่อแสดงความสัมพันธ์ไปยังอีกไฟล์หนึ่งนั่นเอง

ความสัมพันธ์ระหว่าง Entity สามารถเขียนแทนได้ด้วย สัญลักษณ์ หัวลูกศร แบ่งชนิดของความสัมพันธ์ออกได้เป็น 3 ลักษณะ ดังต่อไปนี้

2.9.1 ความสัมพันธ์แบบหนึ่งต่อหนึ่ง (One to One Relationships)

หมายถึง ในช่วงระยะเวลาที่กำหนด ค่าของ Entity A มีความสัมพันธ์กับค่าของ Entity B เพียงค่าเดียวเท่านั้น นั่นคือ หากทราบว่าค่าของ Entity A ก็สามารถหาค่าของ Entity B ได้ด้วย

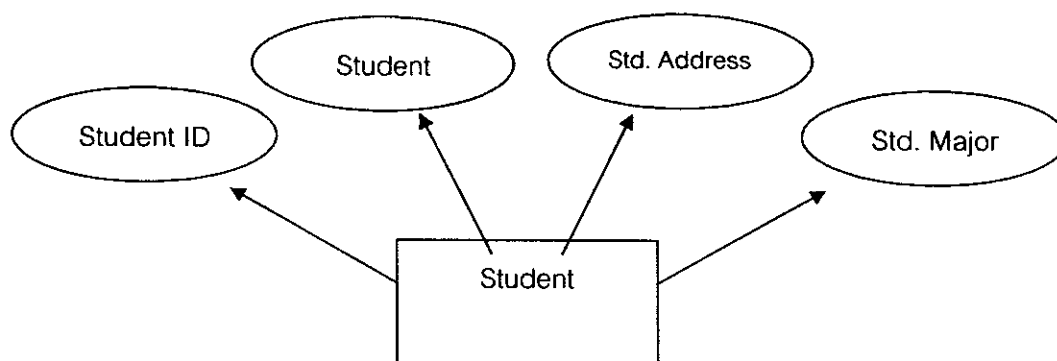
เช่น ในกรณีของเลขประจำตัวนักศึกษาจาก Entity A จะอ้างถึงค่าของ Entity B ได้เพียงค่าเดียวเท่านั้น คือที่อยู่ของนักศึกษาคนๆนั้น



รูปที่ 2.9 แสดงตัวอย่าง Entity ความสัมพันธ์แบบหนึ่งต่อหนึ่ง

2.9.2 ความสัมพันธ์แบบหนึ่งต่อกลุ่ม (One to Many Relationships)

หมายถึง ในช่วงระยะเวลาที่กำหนด ค่าของ Entity A จะมีความสัมพันธ์กับค่าของ Entity B ได้มากกว่า 1 เท่า เช่นนักศึกษา 1 คน สามารถเลือกเรียนวิชาได้หลายวิชาได้ เป็นต้น ความสัมพันธ์ในลักษณะนี้จะเกิดขึ้นเป็นส่วนใหญ่ในระบบ



รูปที่ 2.10 แสดงตัวอย่าง Entity ความสัมพันธ์แบบหนึ่งต่อกลุ่ม

เช่น ในกรณีของนักศึกษาเรียนวิชาเลือกเสรี ที่มีนักศึกษาจากหลายแผนก ลงทะเบียนเรียนรวมกันได้ เป็นต้น

2.9.3 ความสัมพันธ์แบบกลุ่มต่อกลุ่ม (Many to Many Relationships)

หมายถึง ในช่วงระยะเวลาที่กำหนด ทั้งค่าของ Entity A มีความสัมพันธ์กับค่าของ Entity B ได้มากกว่า 1 ค่า (ณัฐพล อุ่นยัง , 2544.หน้า 36-37)

2.10 การแนะนำโปรแกรม Microsoft Access 2000

Microsoft Access 2000 เป็นโปรแกรมฐานข้อมูลที่นิยมใช้กันอย่างแพร่หลาย เนื่องจาก Access 2000 เป็นโปรแกรมฐานข้อมูลที่มีความสามารถในหลายๆด้าน ใช้งานง่าย ซึ่งผู้ใช้งานสามารถ

เริ่มทำได้ตั้งแต่การออกแบบฐานข้อมูล จัดเก็บข้อมูล เขียนโปรแกรมควบคุม ตลอดจนการทำ รายงานแสดงผลข้อมูล

Access 2000 เป็นโปรแกรมฐานข้อมูลที่ใช้ง่าย โดยไม่จำเป็นต้องมีความเข้าใจในการเขียนโปรแกรม ก็สามารถใช้งานได้โดยไม่ต้องศึกษารายละเอียดในการเขียนโปรแกรมให้ยุ่งยาก และสำหรับนักพัฒนาโปรแกรมมืออาชีพนั้น Access 2000 ยังตอบสนองความต้องการในระดับสูงขึ้นไปอีก เช่น การเชื่อมต่อระบบฐานข้อมูลอื่นๆ เช่น SQL SERVER, ORACLE หรือแม้แต่ การนำข้อมูลออกสู่ระบบเครือข่ายอินเทอร์เน็ตก็สามารถทำได้โดยง่าย

2.10.1 ความสามารถของโปรแกรม Access 2000

2.10.1.1 สามารถสร้างระบบฐานข้อมูลใช้งานต่างๆ ได้โดยง่าย เพราะ Access 2000 มีเครื่องมือต่างๆ ให้ใช้ในการสร้างโปรแกรมได้โดยง่ายและรวดเร็ว

2.10.1.2 โปรแกรมที่สร้างขึ้นสามารถตอบสนองผู้ใช้งานได้ตามต้องการ เช่น การสอบถามยอดสินค้า การเพิ่มสินค้า การลบสินค้า การแก้ไขข้อมูลสินค้า เป็นต้น

2.10.1.3 สามารถสร้างรายงานเพื่อแสดงข้อมูลที่ต้องการ ตามที่ผู้ใช้งานต้องการ

2.10.1.4 สามารถสร้างระบบฐานข้อมูล เพื่อนำไปใช้ร่วมกับฐานข้อมูลอื่น ได้โดยง่าย เช่น SQL SERVER, ORACLE ได้

2.10.1.5 สามารถนำเสนอข้อมูลออกสู่ ระบบเครือข่ายอินเทอร์เน็ต ก็สามารถทำได้โดยง่าย(ณัฐพล อุ่นยัง, 2544.หน้า 91-92)

2.10.2 ส่วนประกอบต่างๆ ของข้อมูลใน Access 2000

เมื่อสร้างแฟ้มฐานข้อมูลใหม่ขึ้นมาแล้ว ส่วนประกอบต่างๆ ของแฟ้มฐานข้อมูลใน Access 2000 มีดังนี้

2.10.2.1 ตาราง (Table)

ตารางจะเป็นส่วนที่เก็บข้อมูลทั้งหมดที่มีอยู่ โดยตารางข้อมูลนี้ จะเก็บข้อมูลในรูปแบบแถวและคอลัมน์ โดยข้อมูลในแต่ละแถว เรียกว่า เรคคอร์ด (Record) และข้อมูลในแต่ละคอลัมน์ เรียกว่า ฟیلด์ (Fields)

2.10.2.2 แบบสอบถาม (Query)

แบบสอบถามเป็นเครื่องมือ ที่ใช้ในการสอบถามค้นหาข้อมูลที่ต้องการจากตารางได้อย่างอัตโนมัติ

2.10.2.3 ฟอรม (Form)

ฟอรมเป็นเครื่องมือที่ใช้สร้างส่วนติดต่อกับฐานข้อมูลให้ผู้ใช้งานได้ง่ายขึ้น ซึ่งการสร้างฟอรมนั้น ผู้สร้างสามารถสร้างให้สามารถทำงานได้หลายๆ อย่างเช่น การค้นหาข้อมูล การเพิ่มข้อมูล การลบข้อมูล และแก้ไขข้อมูล

2.10.2.4 รายงาน (Report)

รายงานเป็นเครื่องมือที่ใช้แสดงผลของข้อมูลออกมา และยังสามารถทำกราฟ และรูปภาพได้อีกด้วย ซึ่งจะทำให้รายงานดูน่าสนใจยิ่งขึ้น

2.10.2.5 เพจ (Data Access Page)

เพจเป็นเครื่องมือที่ช่วยให้เราสามารถแสดงผลของข้อมูลขึ้น WEB SITE ได้โดยง่าย

2.10.2.6 แมโคร (Macro)

แมโครเป็นคำสั่งต่างๆ ที่ช่วย Microsoft Access 2000 ทำงานได้อย่างอัตโนมัติ เช่น ต้องการเปิดแฟ้มฐานข้อมูลออกมาแล้ว ให้ทำการเปิดฟอรมอัตโนมัติ เป็นต้น

2.10.2.7 โมดูล (Module)

โมดูลมีหน้าที่คล้ายกับแมโคร แต่สามารถเขียนโปรแกรม เพื่อควบคุมการทำงานได้มากกว่า จะเป็นลักษณะการเขียนโปรแกรม ที่เรียกว่า Visual Basic for Application ซึ่ง จะใช้งานในโปรแกรมที่มีความซับซ้อนมากยิ่งขึ้น (ณัฐพล อุ่นยัง, 2544. หน้า 95-98)

2.10.3 ส่วนประกอบของหน้าจอ Microsoft Access 2000

2.10.3.1 เมนู

เมนูเป็นคำสั่งต่างๆ ที่โปรแกรมได้ทำการเตรียมไว้ให้เรา ซึ่งจะมีเตรียมไว้ในด้านบนของโปรแกรม ดังนี้ แฟ้ม, แก้ไข, มุมมอง, เทรก, เครื่องมือ, หน้าต่าง, วิธีใช้, ซึ่งแต่ละเมนู ก็จะมีคำสั่งย่อยต่างๆ กันออกไป

2.10.3.2 ทูลบาร์

เป็นคำสั่งต่างๆ ที่ โปรแกรมได้ทำการเตรียมไว้ให้เราเหมือนกับเมนู แต่ละคำสั่งที่อยู่ในทูลบาร์นั้น อยู่ในลักษณะที่เป็นสัญลักษณ์รูปภาพ ซึ่งนำมาเรียงกันหลายๆ คำสั่ง จนเป็นทูลบาร์ เพื่อที่จะใช้งานได้ง่ายและรวดเร็วขึ้น ซึ่งจะเห็นว่าบางคำสั่งก็มีอยู่ในเมนูเหมือนกัน (ณัฐพล อุ่นยัง, 2544. หน้า 99)