บทที่ 3

แนวคิดและทฤษฎี

1. แบบจำลองข้อมูล (Data Model)

การวิเคราะห์ระบบเพื่อให้ได้แฟ้มข้อมูลต่างๆ ที่ต้องใช้ในระบบนั้น โครงสร้างของแฟ้มข้อมูลที่ได้ อาจไม่เหมาะสมกับการเปลี่ยนแปลงที่เกิดขึ้นในอนาคต เมื่อเกิดความต้องการใหม่ๆ ที่เกี่ยวข้องกับข้อมูลชุดเดียวกัน จึงจำเป็นที่จะต้องมีโครงสร้างข้อมูลที่สามารถรองรับงานที่มีอยู่เดิม รวมทั้งงานที่อาจจะเกิดขึ้นในอนาคตซึ่งมีความเกี่ยวข้องกับข้อมูลชุดนั้น โครงสร้างข้อมูลนี้ก็คือแบบจำลองข้อมูล ซึ่งต้องได้รับการออกแบบเป็นอย่างดี สามารถเก็บข้อมูลต่างๆที่มีผลกระทบต่อระบบเพื่อให้ได้ข้อมูลอ้างอิงต่างๆ ที่ใช้ร่วมกัน (Korth and Silberschatz, 1986)

แบบจำลองข้อมูลประกอบด้วยเอนทิตี (Entity) และความสัมพันธ์(Relationship) ระหว่างเอนทิตี ใช้เป็นเครื่องมือเพื่อช่วยอธิบายข้อมูล ความสัมพันธ์ระหว่างข้อมูล และกฎเกณฑ์ข้อบังคับต่างๆของข้อมูล ซึ่งจะช่วยอำนวยความสะดวกในการติดต่อระหว่างนักพัฒนาระบบ และผู้ใช้ขั้นสุดท้าย (End User) นอกจากนั้นยังสามารถใช้แก้ปัญหาที่อาจจะเกิดขึ้นในภายหลัง เช่น การมีข้อมูลที่ซ้ำซ้อน ความไม่ตรงกันของข้อมูลในแต่ละหน่วยงาน ความปลอดภัยของข้อมูล เป็นต้น (Korth and Silberschatz, 1986)

1.1 ประเภทของแบบจำลองข้อมูล

แบ่งตามแนวความคิดที่ใช้อธิบายโครงสร้างฐานข้อมูล (Database Structure) ได้เป็น 3 ประเภท (Elmasri and Navathe, 1989) คือ

1.1.1 แบบจำลองข้อมูลระดับสูง (High-Level หรือ Conceptual Data Model)

ให้แนวความคิดที่ใกล้เคียงกับวิธีที่ผู้ใช้งานมองเห็นข้อมูล โดยไม่ต้องเกี่ยวข้องกับข้อมูลที่เก็บจริง เป็นอิสระจากระบบจัดการฐานข้อมูล(DBMS) และอุปกรณ์คอมพิวเตอร์ (Elmasri and Navathe, 1989)

1.1.2 แบบจำลองข้อมูลระดับต่ำ (Low-Level หรือ Physical Data Model)

แสดงรายละเอียดว่าข้อมูลถูกเก็บในคอมพิวเตอร์อย่างไร (Elmasri and Navathe, 1989)

1.1.3 แบบจำลองข้อมูลระดับกลาง (Implementation Data Model)

อยู่ระหว่างแบบจำลองข้อมูลระดับสูง (High-Level Data Model) และแบบจำลองข้อมูลระดับต่ำ (Low-Level Data Model) ให้แนวความคิดที่เข้าใจได้โดยผู้ใช้งาน แต่ไม่รวมถึงวิธีการเก็บข้อมูลในระบบคอมพิวเตอร์ สามารถติดตั้งบนระบบคอมพิวเตอร์ได้โดยตรง ได้แก่ แบบจำลองข้อมูลเชิงสัมพันธ์ (Relational Data Model) แบบจำลองข้อมูลเครือข่าย (Network Data Model) หรือ แบบจำลองข้อมูลลำดับชั้น (Hierachical Data Model) (Elmasri and Navathe, 1989)

1.2 แบบจำลองข้อมูลความสัมพันธ์-เอนทิตี (Entity-Relationship Data Model)

เป็นแบบจำลองข้อมูลระดับสูง (Conceptual Data Model) โดยเป็นการรวบรวมข้อมูลต่างๆ ของระบบ แสดงในรูปของเอนทิตี (Entity) ลักษณะประจำ (Attribute) และความสัมพันธ์ระหว่างเอนทิตี โดยที่เอนทิตี (Entity) คือสิ่งต่างๆในระบบ อาจเป็นสิ่งที่มีอยู่จริงทางกายภาพ(Physical Existance) เช่น พนักงาน สินค้า หนังสือ หรือสิ่งที่มีอยู่ทางความคิด (Conceptual Existance) เช่น บริษัท รายวิชา เป็นต้น(Elmasri and Navathe,1989)

ลักษณะประจำ (Attribute) คือ คุณสมบัติของเอนทิตี

ความสัมพันธ์ (Relationship) คือ ความสัมพันธ์ระหว่างเอนทิตี

แบบจำลองข้อมูลความสัมพันธ์-เอนทิตีถูกเสนอครั้งแรกโดยเชน(Chen) เป็นเครื่องมือที่ช่วยในการติดต่อระหว่างผู้ออกแบบระบบ และผู้ใช้ขั้นสุดท้าย เนื่องจากความง่ายต่อการเข้าใจ และความสะดวกในการแทนข้อมูล แผนภาพที่ใช้การแสดงข้อมูลและความสัมพันธ์ระหว่างข้อมูล ทำให้เห็นภาพได้ชัดเจนและรัดกุมกว่าข้อมูลในรูปแบบข้อความ สามารถสื่อความหมายได้ง่าย โดยไม่จำเป็นต้องมีความรู้พื้นฐานในเรื่องแบบจำลองข้อมูลและการออกแบบฐานข้อมูล (Elmasri and Navathe, 1989)

1.3 แบบจำลองข้อมูลเชิงสัมพันธ์ (Relational Data Model)

แบบจำลองข้อมูลเชิงสัมพันธ์ เป็นแนวความคิดที่แสดงให้เห็นถึงข้อมูล ที่ถูกมองเห็นโดยผู้ใช้งาน โดยจะแสดงรายละเอียดของข้อมูลที่ปรากฏแก่ผู้ใช้ การดำเนินการกับข้อมูลและกฎเกณฑ์ต่างๆ (Fleming and Halle, 1989)

1.3.1 ส่วนประกอบของแบบจำลองข้อมูลเชิงสัมพันธ์(Fleming and Halle, 1989)

1. โครงสร้างข้อมูล (Data Structure) เป็นข้อมูลที่มองเห็นโดยผู้ใช้งานอยู่ในลักษณะของตารางความสัมพันธ์ (Relation)

2. การดำเนินการกับข้อมูลในตารางความสัมพันธ์ (Data Manipulation)

3. กฎและข้อบังคับสำหรับข้อมูลในตารางความสัมพันธ์ (Data Integrity)

1.3.2 ข้อดีของแบบจำลองข้อมูลเชิงสัมพันธ์ (Fleming and Halle, 1989)

1. แบบจำลองข้อมูล ที่เสนอต่อผู้ใช้งานอยู่ในรูปแบบที่เข้าใจง่าย การทำงานเพื่อตอบสนองต่อความต้องการของผู้ใช้จะเกี่ยวข้องเฉพาะข้อมูล ไม่ต้องคำนึงถึงความซับซ้อนของอุปกรณ์คอมพิวเตอร์

2. แบบจำลองข้อมูล แยกโครงสร้างหน่วยเก็บข้อมูล(Storage Structure) และวิธีเข้าถึงข้อมูล(Access Strategy)ออกจากตัวเชื่อมประสานกับผู้ใช้(User Interface) ซึ่งแบบจำลองข้อมูลเชิงสัมพันธ์มีข้อดีในแง่นี้เหนือกว่าแบบจำลองข้อมูลอื่นๆ

3. แบบจำลองข้อมูลเชิงสัมพันธ์ มีพื้นฐานบนทฤษฎีทางคณิตศาสตร์ที่พัฒนามาเป็นอย่างดี และวิธีการออกแบบฐานข้อมูลโดยการทำให้เป็นบรรทัดฐาน(Normalization) ทำให้ได้แบบจำลองข้อมูลที่มีพื้นฐานดี

2. ระบบฐานข้อมูล (Database System)

ฐานข้อมูล (Database) คือ กลุ่มของข้อมูลที่มีความสัมพันธ์กัน ข้อมูลในที่นี้หมายถึงข้อเท็จจริง ที่สามารถบันทึกได้และมีความหมายแน่ชัด เช่น ชื่อ หมายเลขโทรศัพท์ ที่อยู่ เป็นต้น (Elmasri and Navathe, 1989)

ระบบจัดการฐานข้อมูล (Data Base Management System : DBMS) คือ กลุ่มของโปรแกรมที่ช่วยให้ผู้ใช้งานสามารถสร้างและบำรุงรักษาฐานข้อมูล(Elmasri and Navathe, 1989)

ระบบฐานข้อมูล(Database System) คือการประกอบกันของฐานข้อมูลและโปรแกรม (Elmasri and Navathe, 1989)

2.1 ข้อดีของฐานข้อมูล (Date, 1986)

1. ช่วยลดความซ้ำซ้อนของข้อมูล
2. หลีกเลี่ยงการเกิดความไม่ต้องกัน(Inconsistency) ของข้อมูล
3. สามารถใช้ข้อมูลร่วมกันได้
4. ทำให้เกิดมาตรฐานของข้อมูล
5. มีการกำหนดมาตรการต่างๆ เพื่อความปลอดภัยของข้อมูล

2.2 สถาปัตยกรรมของระบบจัดการฐานข้อมูล (Elmasri and Navathe, 1989)

1. ระดับภายใน (Internal Level) ใช้เค้าร่างภายใน (Internal Schema) อธิบายโครงสร้างหน่วยเก็บข้อมูล โดยใช้แบบจำลองข้อมูลกายภาพ (Physical Data Model)

2. ระดับแนวคิด (Conceptual Level) ใช้เค้าร่างเชิงแนวคิด (Conceptual Schema) อธิบายโครงสร้างของฐานข้อมูลทั้งหมดแต่จะไม่แสดงรายละเอียดของโครงสร้างหน่วยเก็บข้อมูล

3. ระดับภายนอก (External หรือ View Level) ประกอบด้วยเค้าร่างภายนอก (External Schema) หรือ ภาพระดับผู้ใช้งาน (User View) ของระบบทั้งหมด โดยที่แต่ละภาพระดับผู้ใช้งาน (User View) จะเป็นภาพที่อธิบายบางส่วนของฐานข้อมูลซึ่งผู้ใช้งานกลุ่มใดกลุ่มหนึ่งสนใจ โดยจะไม่แสดงส่วนอื่นๆ ที่ไม่เกี่ยวข้องกับผู้ใช้กลุ่มนั้น

ผู้ใช้งานจะอ้างถึงเฉพาะเค้าร่างภายนอก (External Schema) ของตนเอง ระบบจัดการฐานข้อมูลจะทำการเปลี่ยนความต้องการที่ระบุในเค้าร่างภายนอก (External Schema) ให้อยู่ในรูปของเค้าร่างเชิงแนวคิด (Conceptual Schema) จากนั้นจึงเปลี่ยนเป็นเค้าร่างภายใน (Internal Schema) เพื่อประมวลผลข้อมูลที่เก็บอยู่จริง

2.3 ความไม่พึ่งพิงของข้อมูล (Data Independence)

ความไม่พึ่งพิงของข้อมูล (Data Independence) คือ ความสามารถในการเปลี่ยนแปลงเค้าร่าง(Schema) ที่ระดับหนึ่ง โดยไม่มีผลกระทบกับเค้าร่าง(Schema) ในระดับที่สูงกว่า มี 2 ประเภท (Elmasri and Navathe, 1989) คือ

1. ความไม่พึ่งพิงในเชิงตรรก (Logical Data Independence) การเปลี่ยนเค้าร่างเชิงแนวคิด (Conceptual Schema) ทำได้โดยที่เค้าร่างภายนอก (External Schema หรือ Application Program) ไม่ต้องเปลี่ยนตาม (Elmasri and Navathe, 1989)

2. ความไม่พึ่งพิงในเชิงกายภาพ (Physical Data Independence) การเปลี่ยนแปลงเค้าร่างภายใน(Internal Schema)สามารถทำได้โดยเค้าร่างเชิงแนวคิด(Conceptual Schema) ไม่ต้องเปลี่ยนตาม (Elmasri and Navathe, 1989)

3. ฐานข้อมูลเชิงสัมพันธ์ (Relational Database)

3.1 โครงสร้างของฐานข้อมูลเชิงสัมพันธ์

ฐานข้อมูลเชิงสัมพันธ์ (Relational Database) จะแทนข้อมูลในฐานข้อมูลในลักษณะของตารางความสัมพันธ์ (Relation) โดยที่แต่ละตารางความสัมพันธ์ (Relation) จะเสมือนเป็น 1 ตาราง และแถวต่างๆ ในตารางจะแสดงค่าของข้อมูลที่มีความสัมพันธ์กัน ชื่อของตาราง (Table) และ ชื่อของสดมภ์ (Column) จะใช้แปลความหมายของค่าในแต่ละแถวของตาราง และได้กำหนดคำเพื่อใช้อธิบายตารางดังนี้ (Elmasri and Navathe, 1989)

ตารางความสัมพันธ์	(Relation)	หมายถึง ตาราง (Table)
ทูเพิล	(Tuple)	หมายถึง แถว (Row หรือ Record)
ลักษณะประจำ	(Attribute)	หมายถึง สดมภ์ (Column หรือ Field)
โดเมน	(Domain)	หมายถึง ค่าที่เป็นไปได้ของข้อมูลในแต่ละสดมภ์

3.2 ทฤษฎีการทำให้เป็นบรรทัดฐาน (Normalization)

การทำให้เป็นบรรทัดฐาน (Normalization) เป็นทฤษฎีที่ใช้ทำการวิเคราะห์และแยกส่วนโครงสร้างของข้อมูลออกไปเป็นชุดๆของความสัมพันธ์ ในทางปฏิบัติเป็นเทคนิคซึ่งประกอบด้วยหลายขั้นตอนซึ่งช่วยให้เกิดประโยชน์ต่างๆ เช่น ช่วยลดขนาดที่ว่างที่ใช้เก็บข้อมูล ช่วยลดความขัดแย้งกันภายในฐานข้อมูล ช่วยลดปัญหาที่เกิดขึ้นเมื่อทำการลบหรือปรับปรุงข้อมูล ทำให้โครงสร้างข้อมูลมีเสถียรภาพสูงสุด (Date, 1986)

3.2.1 รูปแบบบรรทัดฐานระดับที่ 1 (First Normal Form, 1NF)

ทุกสมาชิกในเอนทิตี (Entity) จะมีค่าลักษณะประจำ (Attribute) หนึ่งๆ ได้เพียงค่าเดียว หรือกล่าวคือจะมีกลุ่มของค่าลักษณะประจำ (Attribute) ในสมาชิกหนึ่งไม่ได้ (Date, 1986)

3.2.2 รูปแบบบรรทัดฐานระดับที่ 2 (Second Normal Form, 2NF)

ความสัมพันธ์จะมีคุณสมบัติเป็นรูปแบบบรรทัดฐานระดับที่ 2 เมื่อมีคุณสมบัติของรูปแบบบรรทัดฐานระดับที่ 1 และลักษณะประจำที่ไม่ใช่กุญแจ (Non-Key Attribute) ทุกๆลักษณะประจำ (Attribute) ต้องขึ้นตรงกับกุญแจหลัก (Primary Key) ทั้งชุด (Date, 1986)

3.2.3 รูปแบบบรรทัดฐานระดับที่ 3 (Third Normal Form, 3NF)

ความสัมพันธ์จะมีคุณสมบัติเป็นรูปแบบบรรทัดฐานระดับที่ 3 เมื่อมีคุณสมบัติของรูปแบบบรรทัดฐานระดับที่ 2 และลักษณะประจำที่ไม่ใช่กุญแจ (Non-Key Attribute) ใดๆ จะขึ้นกับลักษณะประจำ(Attribute)อื่นๆ ที่ไม่ใช่กุญแจหลักหรือกลุ่มกุญแจหลัก(Primary Key) ไม่ได้ (Date, 1986)

4. การอัดขนาดข้อมูล (Data Compression)

เทคนิคการอัดขนาดข้อมูลมีการใช้กันอยู่หลายแบบ แบ่งออกได้เป็น 2 ประเภทใหญ่ๆ คือการอัดขนาดข้อมูลที่ไม่มีการสูญเสียข้อมูลเลย คือ ข้อมูลก่อนทำการอัดขนาดข้อมูลและข้อมูลหลังทำการขยายขนาดข้อมูลกลับคืนจะเหมือนกันทุกประการ และ การอัดขนาดข้อมูลที่จะมีการสูญเสียข้อมูลบางส่วน ข้อมูลก่อนการอัดขนาดข้อมูลและหลังจากการขยายขนาดข้อมูลกลับคืนจะไม่เหมือนหรือตรงกัน การอัดขนาดข้อมูลแบบที่มีการสูญเสียข้อมูลไปบางส่วนนี้นิยมใช้ในการอัดขนาดข้อมูลประเภทรูปภาพ หรือเสียง ซึ่งการสูญหายของข้อมูลไปเพียงบางส่วนที่ไม่มีความสำคัญต่อการแปลความหมายจะไม่ทำให้ความหมายของภาพหรือเสียงนั้นผิดหรือเปลี่ยนแปลงไปจากเดิม สำหรับการอัดขนาดขนาดข้อมูลที่จะกล่าวถึงในการวิจัยนี้จะกล่าวถึงเฉพาะการอัดขนาดข้อมูลที่ใช้กับข้อมูลที่เป็นตัวอักษรเท่านั้น ซึ่งจำเป็นต้องใช้เทคนิคหรือวิธีการอัดขนาดข้อมูลที่ไม่มีการสูญเสียข้อมูลเลย ข้อความก่อนการอัดขนาดข้อความ และ หลังการขยายข้อความกลับคืนต้องเหมือนกันทุกประการ (Salton, 1989)

เทคนิคการอัดขนาดข้อความส่วนใหญ่ จะอาศัยหลักการที่ว่า ตัวอักษรแต่ละตัวอักษรหรือข้อความแต่ละข้อความมีความถี่ในการนำไปใช้ไม่เท่ากัน ตัวอักษรบางตัวอักษรหรือคำบางคำมีการนำไปใช้มากในขณะที่ตัวอักษรหรือคำบางคำแทบจะไม่มีการนำไปใช้เลย หลักการพื้นฐานในการอัดขนาดข้อความ จึงใช้วิธีการแทนตัวอักษรหรือคำที่มีความถี่ในการใช้ในเอกสารมากด้วยจำนวนบิตที่น้อยที่สุดและแทนที่ตัวอักษรหรือคำที่มีความถี่การใช้ที่น้อยกว่าด้วยจำนวนบิตที่เพิ่มขึ้น เทคนิคการอัดขนาดข้อความโดยทั่วไป จึงประกอบด้วยเทคนิค 2 ส่วนด้วยกัน คือส่วนแรกทำหน้าที่ในการหาแบบการอัดขนาดข้อความให้ได้ประสิทธิภาพสูงสุด (Modelling) ได้แก่การตรวจสอบทำสถิติข้อความว่าตัวอักษรหรือคำใดในเอกสารที่ต้องการอัดขนาดข้อความนั้นมีการใช้มากน้อยต่างกันอย่างไรเพื่อที่จะเลือกได้ว่าควรที่จะใช้รหัสขนาดเท่าใดแทนตัวอักษรหรือข้อความนั้น ส่วนที่ 2 เป็นเทคนิคหรือวิธีที่ใช้เข้ารหัสข้อความ(Code) คือการแทนที่ตัวอักษรหรือคำในเอกสารต้นฉบับด้วยรหัสที่ต้องการในเอกสารที่ทำการอัดข้อความแล้ว (Salton, 1989)

4.1 เทคนิคการอัดขนาดข้อมูลเฉพาะงาน

เทคนิคการอัดขนาดข้อมูลเฉพาะงานนี้ วิธีการที่ใช้จะขึ้นกับลักษณะข้อมูลที่จะมาทำการอัดขนาดข้อความมาก เช่น ถ้าข้อความที่จะนำมาทำการอัดขนาดข้อมูลมีตัวเลขเป็นจำนวนมาก อาจจะใช้วิธีการอัดขนาดข้อมูลโดยทำการเก็บข้อมูลตัวเลข 1 ตัวด้วยรหัส 4 บิต หรือในกรณีที่มีการซ้ำของตัวอักษรหรือข้อความเป็นจำนวนมาก เช่น ตัวอักษรที่เป็นช่องว่าง หรือเลข 0 อาจใช้วิธีการเก็บข้อมูลตัวอักษรนั้นแล้วตามด้วยจำนวนตัวอักษรที่มีการซ้ำนั้นแทนเป็นต้น (Salton, 1989)

4.2 เทคนิคการใช้รหัสที่มีขนาดคงที่

เป็นการใช้รหัสที่มีขนาดคงที่แน่นอนในการแทนข้อความ ซึ่งโดยปกติแล้วตัวอักษรในภาษาอังกฤษจะสามารถแทนได้ด้วยรหัสขนาด 5 - 6 บิต แต่ในทางปฏิบัติจะทำการแบ่งรหัสออกเป็น 2 ชุดหรือมากกว่า คือ ชุดของตัวอักษรที่มีการใช้งานมาก ชุดตัวอักษรที่มีการใช้งานน้อย ทั้งสองชุดของตัวอักษรจะแทนด้วยรหัสที่มีขนาดความกว้างเท่ากัน โดยมากอาจแทนด้วยรหัส 4 - 5 บิต ในชุดของตัวอักษรที่มีการใช้งานมาก ซึ่งจะเป็นชุดของตัวอักษรที่ใช้โดยปกตินั้น จะมีรหัสสำหรับเปลี่ยนชุดของตัวอักษรเป็นชุดที่มีการใช้น้อยได้ 1 หรือ 2 รหัส ในกรณีที่ต้องการใช้รหัสจากชุดที่มีการใช้งานน้อย จะทำได้โดยใส่รหัสเปลี่ยนชุดตัวอักษรที่ต้องการลงไปข้างหน้าก่อนแล้วใส่รหัสของตัวอักษรที่ต้องการ ดังนั้นตัวอักษรในชุดตัวอักษรที่มีการใช้น้อยจะใช้ถูกแทนที่ด้วยรหัส 2 รหัส คือ รหัสเปลี่ยนชุดตัวอักษรและรหัสของตัวอักษรนั้น (Salton, 1989)

4.3 เทคนิคการใช้รหัสที่มีขนาดไม่เท่ากันแบบจำกัด

เทคนิคแบบนี้จะใช้รหัสที่มีขนาดความยาวไม่เท่ากันในการแทนตัวอักษร โดยที่จะใช้รหัสที่ความยาวน้อยในการแทนตัวอักษรที่มีความถี่ในการใช้งานสูง และใช้รหัสที่มีความยาวมากกว่าในการแทนตัวอักษรที่มีความถี่ในการใช้งานน้อย ตัวอย่างเช่น ใช้รหัสขนาด 4 บิต แทนกลุ่มตัวอักษรที่มีการใช้งานมาก โดยบิตแรกเป็น 0 จะเป็นตัวบอกว่าเป็นรหัสชุดที่มีความถี่ในการใช้งานมากมีขนาดของรหัสเป็น 4 บิต สามารถใช้แทนตัวอักษรที่มีความถี่การใช้สูงได้ 8 ตัว และใช้รหัสขนาด 8 บิต แทนกลุ่มของตัวอักษรที่มีความถี่การใช้ไม่บ่อยนัก โดยบิตแรกของรหัสจะเป็น 1 เป็นตัวบอกว่าเป็นรหัสที่มีความยาว 8 บิต สามารถใช้แทนตัวอักษรอื่นที่มีความถี่การใช้น้อยกว่าชุดแรกได้ 128 ตัว เป็นต้น (Salton, 1989)

4.4 เทคนิคการใช้รหัสที่มีขนาดไม่เท่ากัน

เทคนิคการอัดข้อมูลโดยใช้รหัสที่มีความยาวไม่เท่ากันนี้ จะใช้รหัสที่มีความยาวแตกต่างกันในการแทนตัวอักษรที่มีความถี่ในการใช้งานแตกต่างกัน เช่น ใช้รหัสขนาด 1 บิต แทนตัวอักษรที่มีความถี่ในการใช้งานสูงสุด ใช้รหัสขนาด 2 บิตแทนตัวอักษรที่มีความถี่ในการใช้เป็นอันดับที่ 2 และใช้รหัสขนาด 256 บิต แทนตัวอักษรแทบจะไม่ได้ใช้เลยเป็นต้น (Salton, 1989)

เทคนิคการอัดข้อมูลแบบนี้แบ่งได้เป็น 2 แบบ คือ แบบแรกโปรแกรมจะทำการวิเคราะห์และทำสถิติความถี่ของการใช้งานตัวอักษรทั้งหมด แล้วเรียงจากมากไปน้อย การเข้ารหัสก็ทำโดยใช้รหัสขนาด 1 บิตแทนตัวอักษรที่มีความถี่ในการใช้งานมากเป็นอันดับที่ 1 ใช้รหัสขนาด 2 บิต แทนตัวอักษรที่มีความถี่ในการใช้งานเป็นอันดับ 2 ใช้รหัสขนาด 3 บิต แทนตัวอักษรที่มีความถี่การใช้งานมากเป็นอันดับ 3 ทำอย่างนี้จนกระทั่งถึงตัวอักษรที่มีความถี่ในการใช้งานน้อยที่สุดก็ใช้รหัสขนาด 256 บิต เป็นต้น แบบที่ 2 เป็นแบบที่ทำการปรับปรุงจากแบบแรกโดยจะมีการทำสถิติความถี่ของการใช้งานของตัวอักษรต่างๆ เช่นเดียวกับแบบแรก แต่การแทนรหัสจะใช้จำนวนบิตที่เหมาะสมกว่าในการแทนที่ตัวอักษรต่างๆ โดยที่ตัวอักษรที่มีความถี่ในการใช้งานใกล้เคียงหรือเท่ากันจะใช้รหัสที่มีขนาดเท่ากัน เช่น ตัวอักษรที่มีความถี่ในการใช้งานสูงอันดับที่ 1 ใช้รหัสขนาด 1 บิต ส่วนตัวอักษรอันดับถัดไปอีก 4 ตัวมีความถี่ในการใช้งานใกล้เคียงกันมาก ใช้รหัสขนาด 3 บิต ในการแทนตัวอักษรเป็นต้น เทคนิคแบบที่ 2 นี้จำเป็นต้องมีวิธีสำหรับคำนวนหาจำนวนบิตที่เหมาะสมในการใช้แทนตัวอักษรต่างๆ หลังการทำสถิติความถี่ของตัวอักษรต่างๆ แล้ว (Salton, 1989)

4.5 เทคนิคการใช้รหัสแทนกลุ่มของตัวอักษร

เทคนิคต่างๆที่ได้กล่าวไปแล้วทั้งหมดเป็นเทคนิคที่ใช้ในการเข้ารหัสตัวอักษรเพียง 1 ตัวเท่านั้น แต่การเข้ารหัสข้อความนั้น การเข้ารหัสในแต่ละครั้งอาจจะเข้ารหัสทีละมากกว่า 1 ตัวอักษรก็ได้ เช่น ครั้งละ 2 - 4 ตัวอักษร การเข้ารหัสโดยวิธีนี้ ใช้หลักการที่ว่ากลุ่มของตัวอักษรที่มาจับกลุ่มเป็นคำนั้นมีการใช้ซ้ำกันค่อนข้างมาก และมีแบบการจับกลุ่มของตัวอักษรเป็นจำนวนมากที่จะไม่มีการพบในคำศัพท์เลย การใช้รหัสแทนกลุ่มของตัวอักษรนี้จำเป็นที่จะต้องสร้างพจนานุกรมของชุดตัวอักษรที่จะทำการเข้ารหัสทั้งหมด ซึ่งวิธีการนี้จะมีประสิทธิภาพในการอัดขนาดข้อมูลมากยิ่งขึ้นเมื่อจำนวนตัวอักษรที่จะเข้ารหัสในแต่ละครั้งเพิ่มขึ้น แต่ในทางปฏิบัติ การเข้ารหัสตัวอักษรครั้งละมากๆ เช่นมากกว่า 4 ตัวอักษรขึ้นไปขนาดของพจนานุกรมที่ใช้จะมีขนาดใหญ่ขึ้นมาก ทำให้เวลาที่ต้องใช้ในค้นหารหัสและปรับปรุงพจนานุกรมมากยิ่งขึ้นทำให้ไม่เหมาะสมในการนำมาใช้งาน เนื่องจากใช้เวลาในการอัดข้อมูลและขยายข้อมูลในแต่ละครั้งนานเกินไป (Salton, 1989)

5. การค้นคืนสารสนเทศโดยใช้แฟ้มดัชนีแบบผกผัน (Inverted Indexing Methods)

โดยปกติฐานข้อมูลสารสนเทศต่างๆ จะถูกเก็บไว้ในลักษณะของระเบียน เป็นรายการ แต่ละรายการประกอบด้วยเลขที่รายการและเขตของข้อมูลต่างๆ ซึ่งในกรณีของข้อมูลแบบข้อความที่มีขนาดความยาวและรูปแบบไม่แน่นอน (Unstructured Text) ก็อาจจะมีการรวบรวมหรือแยกคำสำคัญ (Keywords) ออกมาเก็บไว้ในคนละเขตข้อมูล ดังนั้นในแต่ละรายการข้อมูลจึงอาจประกอบด้วยเลขที่ระเบียน ชุดของคำสำคัญ และข้อความในลักษณะของเอกสารที่ไม่มีรูปแบบแน่นอน (Unstructured Text) การเก็บข้อมูลในลักษณะปกติที่กล่าวมาแล้วนั้นจะเกิดปัญหาในกรณีที่ต้องการสอบถามว่าคำสำคัญที่ต้องการทราบนั้นมีพบอยู่ในเอกสารใดบ้าง เพราะว่าจะต้องทำการตรวจสอบทุกๆ ระเบียนในแฟ้มข้อมูล ยิ่งถ้าข้อมูลที่เก็บไว้ไม่ได้มีการรวบรวมคำสำคัญแยกออกมาเก็บไว้เป็นเขตข้อมูลต่างหาก การตรวจสอบจำเป็นต้องเข้าไปกราดตรวจในเอกสารต้นฉบับทุกๆ ครั้งแล้ว ก็จะยิ่งทำให้การค้นข้อมูลแต่ละครั้งต้องใช้เวลามาก (Salton, 1989)

การสร้างแฟ้มดัชนีแบบผกผัน (Inverted Index File) คือ การสร้างแฟ้มดัชนีของคำสำคัญ โดยจะทำการเก็บข้อมูลว่าคำสำคัญแต่ละคำมีพบในระเบียนข้อมูลใดบ้างในลักษณะที่ผกผันกับแฟ้มข้อมูลหลักที่กล่าวไปแล้ว แฟ้มดัชนีแบบผกผันที่ทำการสร้างขึ้นใหม่นี้จะช่วยเพิ่มความสะดวกรวดเร็วในการสืบค้นว่าคำสำคัญที่ต้องการทราบนั้นมีพบในข้อความเอกสารใดบ้าง โดยคำสำคัญที่เก็บไว้ในแฟ้มดัชนีแบบผกผัน จะเรียงข้อมูลตามตัวอักษรหรืออาจจะมีการจัดโครงสร้างเป็นพิเศษเพื่อเพิ่มประสิทธิภาพในการสืบค้นข้อความได้ โดยไม่จำเป็นต้องทำการค้นแบบลำดับ (Salton, 1989)