หอสมุดกลาง สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 2
แนวความคิดและทฤษฎีที่เกี่ยวข้อง

2.1 ระบบคัมบัง (Kanban System)

ระบบคัมบังมิใช่สิ่งที่เกิดขึ้นจากการคิดค้นขึ้นเอง แต่มาจากแนวคิดของวิธีการควบคุมคงคลังแบบสถิติ ซึ่งเป็นระบบสั่งเมื่อถึงจุดกำหนด

ระบบคัมบังก็มีลักษณะเช่นเดียวกัน กล่าวคือในภาวะที่อุปสงค์มีการเปลี่ยนแปลงที่ไม่แน่นอนแล้ว ระบบคัมบังก็จะเกิดภาวะขาดแคลนชิ้นส่วนหรือเกิดภาวะคงคลังมากเกินไปเช่นเดียวกันวิธีการควบคุมแบบจุดสั่งผลิต

ดังนั้น ในขั้นตอนของการวางแผนการผลิตจึงต้องทำการเฉลี่ยชนิดสินค้า และปริมาณสินค้าให้เท่า ๆ กัน สร้างอุปสงค์ที่คงที่ขึ้นมาเราเรียกว่า การปรับเรียบการผลิต (การสร้างความสม่ำเสมอในการผลิต หรือการเฉลี่ยการผลิต)

ในสภาพเช่นนี้ ทำให้สามารถขจัดความสูญเปล่าได้อย่างเด็ดขาด แต่ทำให้ไม่สามารถใช้วิธีผลิตแบบเป็นกลุ่ม ในโรงงานที่มีการผลิตเป็นแบบล็อต หรือมีการผลิตเป็นแบบ BATCH PRODUCTION จึงจำเป็นจะต้องเปลี่ยนแปลงวิธีการผลิตเสียใหม่

ซึ่งในแง่ปฏิบัติแล้ว จะใช้วิธีควบคุมการสั่งเป็นแบบช่วงเวลาสั่งซื้อตายตัว เช่น ทุก ๆ วัน หรือ ทุก ๆ ครึ่งวัน เป็นต้น

นอกจากในความหมายในเชิงการควบคุมพัสดุคงคลังแล้ว คัมบังยังมีอีกหลายบทบาทดังนี้

บทบาทที่ 1 เป็นตัวแสดงข้อมูลการใช้ของและออกคำสั่งในการปฏิบัติงาน

นอกจากจะใช้เป็นเครื่องมือในการแสดงผลหรือข้อมูลในการผลิตเพื่อบอกให้รู้ว่า มีการใช้ชิ้นส่วนที่ผลิตไปประมาณเท่าไรแล้ว ยังใช้เป็นเครื่องมือในการกำหนดวิธีการ และจำนวนชิ้นส่วนที่ต้องการจะให้ผลิตเพิ่มได้อีกด้วย

บทบาทที่ 2 ขจัดความสูญเปล่าที่เกิดขึ้นจากการผลิตเกิน

โครงสร้างการผลิตโดยใช้คัมบังจะเป็นการบังคับไม่ให้มีการผลิตเกินความต้องการ โดยผู้ผลิตในกระบวนการก่อนหน้า จะตรวจดูและผลิตเสริมจากจำนวนคัมบังที่ขาดหายไป เนื่องจากมีการดึงไปใช้โดยกระบวนการถัดไป กล่าวอีกนัยหนึ่งคือเป็นการผลิตเพื่อให้สอดคล้องกับความต้องการ หรือที่เรียกกันว่า “ระบบการผลิตแบบดึง” นั่นเอง

บทบาทที่ 3 เป็นเครื่องมือส่งเสริมการปรับปรุงงาน

เราจะไม่มีทางทราบถึงปัญหาที่เกิดขึ้นได้เลย ตราบเท่าที่ยังคงมีการจัดเก็บสต๊อคอยู่ ฉันใดก็ฉันนั้นหากมีการเพิ่มจำนวนคัมบังมากจนเกินไปแล้ว ก็อาจจะไม่รู้ตัวเลยว่าในขณะนี้มีความสูญเปล่าเกิดขึ้นดังนั้น โดยการค่อย ๆ ลดจำนวนคัมบังที่ใช้อยู่ออกไปทีละแผ่น ๆ ก็จะช่วยทำให้เราทราบได้ว่าต้นตอของปัญหาคืออะไร เพื่อเป็นแนวทางในการปรับปรุงต่อไป เพราะการปรับปรุงจะเกิดขึ้นได้ก็ต่อเมื่อมีการคิดหาหนทางเพื่อแก้ไขปัญหาที่ตนรู้สึกว่าเดือนร้อนเท่านั้น

บทบาทที่ 4 เป็นเครื่องมือควบคุมความสูญเปล่า

โดยทั่วไปแล้วจะมีช่องว่างเกิดขึ้นระหว่างการไหลของวัตถุและข้อมูลในกระบวนการผลิตซึ่งอาจก่อให้เกิดความไขว้เขวเข้าใจผิด เนื่องจากข้อมูลที่ได้รับไม่ตรงกับความเป็นจริงที่เกิดขึ้น คัมบังจึงเป็นเครื่องมือที่ใช้สำหรับแก้ไขปัญหาความไม่ถูกต้องของข้อมูล ในการที่จะคอยรายงานความคืบหน้าทุกระยะของการผลิต ทำให้สามารถจะทำการตรวจสอบและควบคุมได้ด้วยตาเปล่านั่นเอง

อนึ่งในการใช้คัมบังนั้นยังมีกฎเกณฑ์อีก 6 ประการ ที่จำเป็นต้องพิจารณาประกอบด้วยดังนี้คือ

กฎข้อที่ 1 ดึงจากกระบวนการถัดไป

จำนวนคัมบังที่ขาดหายไปก็เพราะถูกดึงไปใช้หรือรับไปโดยกระบวนการถัดไป เท่าที่จำเป็นและในเวลาที่จำเป็นต้องใช้เท่านั้น

กฎข้อที่ 2 กระบวนการก่อนหน้าจะผลิตเท่าส่วนที่ถูกดึงออกไป

กระบวนการที่อยู่ก่อนหน้า จะต้องผลิตทดแทนในส่วนที่ถูกดึงไปใช้โดยกระบวนการที่อยู่ถัดไปเป็นจำนวนเท่า ๆ กับส่วนที่ขาดหายไปเท่านั้นด้วยวิธีนี้ก็จะเป็นการควบคุม "ไม่ให้เกิดความสูญเปล่าจากการผลิตเกิน"

กฎข้อที่ 3 ของดี 100%

ต้องสร้างคุณภาพในกระบวนการ เพื่อป้องกันไม่ให้มีของเสียที่จะส่งต่อไปยังกระบวนการถัดไป

กฎข้อที่ 4 ทำให้เกิดความสม่ำเสมอ หรือเฉลี่ยการผลิต

ปรับหรือควบคุมให้เกิดความสม่ำเสมอและมีเสถียรภาพ ไม่ให้เกิดการเบี่ยงเบนหรือความแปรปรวนในกระบวนการผลิต

กฎข้อที่ 5 ไปพร้อมกับของ

ดำเนินการด้านการควบคุมด้วยตาเปล่าอย่างจริงจัง กล่าวคือ คัมบังจะต้องมีการเปลี่ยนแปลงและตรงกับสภาพความเป็นจริงในกระบวนการเท่านั้น

กฎข้อที่ 6 เป็นเครื่องมือตรวจหาจุดซึ่งจำเป็นจะต้องมีการปรับปรุง

นำมาเป็นเครื่องมือ หรือใช้ประโยชน์ในการตรวจสอบว่ามีปัญหาเกิดขึ้นหรือไม่ ด้วยการค่อย ๆ ลดจำนวนคัมบังที่มีอยู่

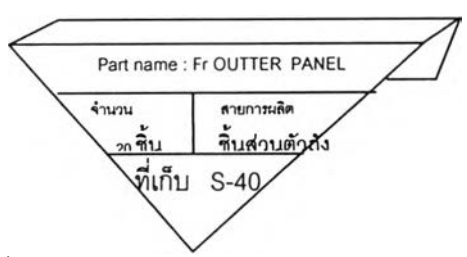
ชนิดของคัมบัง

1. **คัมบังสั่งซื้อ** โดยทั่วไปเรียกว่า **คัมบังรับมอบจากภายนอก** ใช้ในกรณีมีการรับมอบชิ้นส่วนจากภายนอกมายังสายการผลิต เช่น รับมอบมายังฝ่ายประกอบเป็นต้น โดยเฉพาะอย่างยิ่งคัมบังสั่งซื้อจะใช้กับชิ้นส่วนที่รับมอบจากการสั่งซื้อจากผู้ผลิตภายนอก จะเป็นกระดาษอยู่ในซองพลาสติกอีกที เพื่อป้องกันรอยเปื้อนต่าง ๆ

หมายเลขกล่องเก็บคัมบัง 15	ตำแหน่ง หรือจุดทำงานที่เก็บชิ้นส่วน WH101-15	ผู้ผลิตชิ้นส่วน KKP
Part no. 12345-45678 Part name : RR OUTTER MIRROR		
ตำแหน่ง หรือจุดทำงานที่ส่งชิ้นส่วน ตัวถัง 16	จำนวน 50 ชิ้น	1 : 1 : 1

ภาพที่ 2.1 ตัวอย่างใบคัมบังสั่งซื้อ หรือใบคัมบังรับมอบจากภายนอก

2. **คัมบังระหว่างกระบวนการผลิต** ในสายการประกอบจะมีการใช้ชิ้นส่วนที่ผลิตในกระบวนการภายในโรงงาน ซึ่งการรับมอบชิ้นส่วนเหล่านี้จะใช้คัมบังระหว่างกระบวนการ โดยปกติจะเรียกสั้น ๆ ว่า คัมบังรับมอบ และบางครั้งยังใช้ในความหมายต่อเนื่องเป็นสัญญาณให้เริ่มทำการผลิตชิ้นส่วนอีกครั้ง ซึ่งจะมีลักษณะเป็นแผ่นเหล็กหรือพลาสติก เพื่อใช้แขวนเป็นสัญญาณ



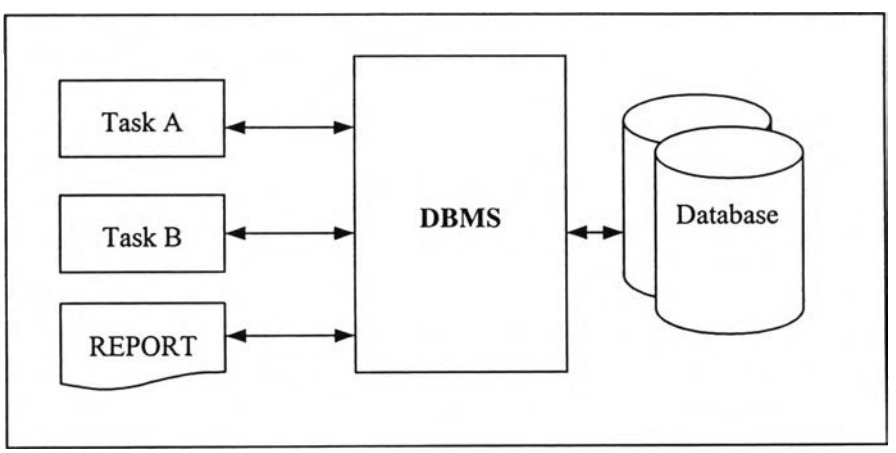
ภาพที่ 2.2 ตัวอย่างคัมบังระหว่างกระบวนการผลิต

2.2 ฐานข้อมูลแบบรีเลชั่น

2.2.1 ระบบจัดการฐานข้อมูล (Database Management System)

ก่อนที่จะทำความเข้าใจกับระบบจัดการฐานข้อมูล ควรจะทำความเข้าใจกับคำว่า ฐานข้อมูล (Database) เสียก่อน หากจะอธิบายอย่างง่าย ๆ ฐานข้อมูลจะเป็นการรวบรวมเอาข้อมูลที่เกี่ยวข้องกัน มีความสัมพันธ์กัน เข้าไว้ด้วยกัน เพื่อให้สะดวกต่อการจัดเก็บและเรียกใช้งาน หากเป็นเมื่อก่อนที่ยังไม่มีระบบจัดการฐานข้อมูลให้ชัดอย่างแพร่หลาย การจัดการกับข้อมูลระบบบุคลากร จะต้องมีแฟ้มข้อมูลที่จัดเก็บข้อมูลส่วนตัวบุคลากร เช่น ชื่อ ที่อยู่ แฟ้มหนึ่ง และมีแฟ้มข้อมูลที่ใช้สำหรับเก็บข้อมูลเกี่ยวกับโครงสร้างของหน่วยงานหรือองค์กรอีกแฟ้มหนึ่ง อาจจะมีข้อมูลอื่น ๆ อยู่ในอีกหลาย ๆ แฟ้ม การจะให้ได้ข้อมูลของบุคลากรคนหนึ่งที่ครบถ้วน จะต้องไปเปิด และ อ่านข้อมูลในแฟ้มต่าง ๆ ที่เกี่ยวข้อง ทำให้โปรแกรมยุ่งยากขึ้น เกิดการผิดพลาดง่ายขึ้นแต่เมื่อเป็นฐานข้อมูลทำให้ข้อมูลต่าง ๆ ที่มีความสัมพันธ์ของข้อมูลต่าง ๆ ในฐานข้อมูลนั้นด้วย ซึ่งจะช่วยให้เราเรียกใช้ข้อมูลต่าง ๆ ในฐานข้อมูลนั้นได้ง่ายขึ้น

เนื่องจากข้อมูลต่าง ๆ ถูกจัดเก็บรวมกันไว้ในฐานข้อมูล ทำให้เราไม่สมารถนำข้อมูลเข้าเก็บหรือเรียกข้อมูลเข้าไปเก็บหรือเรียกข้อมูลจากฐานข้อมูลขึ้นมาใช้ได้โดยตรง การนำเอาข้อมูลเข้าไปเก็บหรือดึงออกมาใช้นั้นจะต้องทำผ่านระบบจัดการฐานข้อมูลของฐานข้อมูลนั้น



ภาพที่ 2.3 แสดงความสัมพันธ์ของระบบจัดการฐานข้อมูล (DBMS)

ดังนั้นพอจะกล่าวได้อย่างคร่าว ๆ ว่าระบบจัดการฐานข้อมูลก็คือ โปรแกรม ๆ หนึ่ง ที่มีหน้าที่ในการจัดการ จัดเก็บบันทึก ค้นหาและนำข้อมูลออกมาใช้ และดูแลรักษาข้อมูลให้มีความถูกต้องอยู่เสมอ ซึ่งหน้าที่หลัก ๆ ของระบบจัดการฐานข้อมูลมีดังนี้

ดูแลรักษาความถูกต้องของข้อมูลในฐานข้อมูล

ระบบจัดการฐานข้อมูล จะทำหน้าที่ในการตรวจสอบความถูกต้องของข้อมูลทุก ๆ ครั้งที่มีการเปลี่ยนแปลง หรือแก้ไขข้อมูลในฐานข้อมูล หากข้อมูลนั้นมีการกำหนดเงื่อนไขไว้ เช่น ค่าที่เป็นยอมรับได้สำหรับข้อมูลนั้น (Valid Range) เป็นต้น รวมถึงความสามารถในการแก้ไขความผิดพลาดของข้อมูล (Error Recovery) เมื่อความผิดพลาดเกิดขึ้นในระบบได้

จัดการดูแลการใช้งานข้อมูลร่วมกันของโปรแกรมหลาย ๆ โปรแกรม

หน้าที่หนึ่งของระบบจัดการฐานข้อมูล คือ พยายามเอื้ออำนวยให้สามารถเรียกใช้ข้อมูลเดียวกันจากหลาย ๆ โปรแกรมได้อย่างสะดวกและเหมาะสมต่อการใช้งานโดยที่ไม่ทำให้ความถูกต้องของข้อมูลเสียไป

จัดการและดูแลระบบรักษาความปลอดภัยหรือการกำจัดสิทธิในการเรียกใช้ข้อมูล

การยอมให้มีการใช้งานข้อมูลได้พร้อม ๆ กัน จากหลาย ๆ โปรแกรมหรือผู้ใช้หลาย ๆ คนนั้น ผู้ใช้แต่ละคนหรือโปรแกรมแต่ละโปรแกรมมีความต้องการในการใช้ข้อมูลในระดับที่แตกต่างกัน บางคนต้องการที่จะเรียกดูเท่านั้น แต่บางคนต้องการจะแก้ไขข้อมูลด้วย การกำหนดสิทธิในการใช้ข้อมูลให้เหมาะสมกับความต้องการ จะช่วยลดความผิดพลาดของข้อมูลได้เป็นอย่างดี

สร้างมาตรฐานในการเรียกใช้ข้อมูล

ทุก ๆ โปรแกรมที่ต้องการจะเรียกใช้ข้อมูลจากฐานข้อมูล จะต้องทำการร้องขอ (Request) จากระบบจัดการฐานข้อมูล ดังนั้นคำสั่งในการเรียกใช้ข้อมูลจากโปรแกรมต่าง ๆ จะเหมือนกันหมดไม่ว่าจะใช้ภาษาอะไรในการพัฒนาโปรแกรม เพราะจะต้องส่งคำสั่งในการเรียกใช้ข้อมูลตามที่ระบบการจัดการฐานข้อมูลจะเข้าใจได้เท่านั้น ซึ่งภาษาหรือคำสั่งมาตรฐานที่ใช้ในการเรียกใช้ และจัดการกับข้อมูลของระบบจัดการฐานข้อมูลต่างๆ ในปัจจุบัน คือ Structured Query Language หรือ SQL

2.2.2 ประโยชน์จากการประมวลผลด้วยฐานข้อมูล

ประโยชน์จากการใช้ฐานข้อมูลในการประมวลผลมีมากมาย (โดยเฉพาะอย่างยิ่ง ถ้าเราเลือกใช้ DBMS ที่มีคุณภาพ) ซึ่งส่วนใหญ่เราก็ได้กล่าวถึงไปแล้วแต่ในที่นี้จะสรุปไว้ให้เห็นเด่นชัดอีกครั้งหนึ่งดังต่อไปนี้

1. ลดความซ้ำซ้อนของข้อมูล
2. สามารถหลีกเลี่ยงความขัดแย้งของข้อมูลได้ในระดับหนึ่ง
3. สามารถใช้ข้อมูลร่วมกันได้

4. สามารถควบคุมความเป็นมาตรฐานได้
5. สามารถจัดหาระบบความปลอดภัยที่รัดกุมไว้
6. สามารถควบคุมความคงสภาพของข้อมูลได้
7. สามารถสร้างสมดุลในความขัดแย้งของความต้องการได้
8. เกิดความเป็นอิสระของข้อมูล

ซึ่งเราจะขอขยายความแต่ละหัวข้อ โดยเราได้ใส่ข้อความภาษาอังกฤษกำกับหัวข้อเอาไว้ เพราะคำศัพทบางคำค่อนข้างจะเป็นมาตรฐาน จึงอยากให้ท่านผู้อ่านได้เห็นศัพท์เหล่านี้ด้วย ดังต่อไปนี้

ลดความซ้ำซ้อนของข้อมูล (redundancy can be reduced) จากตัวอย่างที่ผ่าน ๆ มา จะเห็นว่าการประมวลผลโดยใช้ไฟล์ธรรมดานั้นจำเป็นที่ผู้ใช้แต่ละกลุ่มจะต้องมีไฟล์ส่วนตัวเอาไว้ ดังนั้นจึงเกิดเหตุการณ์ที่ข้อมูลทั้งหมดมาเก็บไว้ที่หลาย ๆ แห่ง หรือที่เราเรียกกันว่าความซ้ำซ้อน การนำข้อมูลทั้งหมดมาเก็บไว้ที่เดียวกันในฐานข้อมูลนี้เป็นการ " ลด " ความซ้ำซ้อนลงไปได้ ขอให้สังเกตว่า เราใช้คำว่า " ลด " แทนที่จะใช้คำว่า " ขจัด " ทั้งนี้ก็เพราะมีงานบางประเภทเหมือนกันที่เราอาจจะต้องเก็บข้อมูลชุดเดียวกันไว้มากกว่า 1 แห่ง อย่างไรก็ดีการใช้ระบบฐานข้อมูลจะทำให้เราสามารถควบคุมการเกิดความซ้ำซ้อน เพราะถึงแม้ว่าจะต้องเก็บข้อมูลชุดเดียวกันไว้มากกว่า 1 แห่ง DBMS ก็จะเป็นตัวที่ทราบอยู่ตลอดเวลาว่ามีความซ้ำซ้อนอยู่ที่ใดบ้าง

สามารถหลีกเลี่ยงความขัดแย้งของข้อมูลได้ในระดับหนึ่ง (inconsistency can be avoided... to some extent) ประโยชน์ในข้อนี้ก็สืบเนื่องมาจากข้อที่แล้ว เพราะอย่างที่เราได้กล่าวถึงไปแล้วว่าการเก็บข้อมูลไว้หลาย ๆ แห่งอาจจะก่อให้เกิดปัญหาว่า การแก้ไขข้อมูลเดียวกันนี้ทำไม่เหมือนกันในทุก ๆ แห่ง ทำให้เกิดปัญหาว่า ข้อมูลชุดเดียวกันอาจมีค่าในแต่ละแห่งไม่ตรงกัน ดังนั้นถ้าการใช้ระบบฐานข้อมูลทำให้เราสามารถลดความซ้ำซ้อนลงไปได้ โดยมี DBMS เป็นตัวควบคุมดูแลว่า เมื่อเกิดการแก้ไขข้อมูลขึ้นเมื่อใด จะต้องแก้ให้เหมือนกันครบทุกแห่ง

สามารถใช้ข้อมูลร่วมกันได้ (the data can be shared) การใช้ข้อมูลร่วมกันได้นี้ ไม่ได้จำกัดอยู่เฉพาะโปรแกรมที่ใช้ข้อมูลอยู่ในปัจจุบันเท่านั้น แต่กินความถึงโปรแกรมประยุกต์ที่จะพัฒนาขึ้นมาใหม่ด้วย ที่สามารถจะใช้ข้อมูลที่มีอยู่ได้เลย โดยไม่จำเป็นต้องเพิ่มเติมข้อมูลเข้าไปในระบบอีก

สามารถควบคุมความมาตรฐานได้ (standards can be enforced) จากการที่เรานำข้อมูลมาเก็บรวมกันไว้ในฐานข้อมูลเช่นนี้ ทำให้ผู้ที่มีหน้าที่ควบคุมดูแลการใช้ระบบฐานข้อมูลสามารถกำหนดมาตรฐานของข้อมูลขึ้นมาได้ เช่น ให้ใช้หน่วยมาตราการวัดที่เหมือนกัน รูปแบบในการเขียนวันที่ให้เหมือนกัน เป็นต้น ซึ่งการที่เหล่าข้อมูลล้วนใช้มาตรฐานเดียวกันเช่นนี้ ทำให้การแลกเปลี่ยนข้อมูล

ระหว่างระบบเป็นไปอย่างสะดวกและถูกต้อง เรามีศัพท์ที่ใช้เรียกผู้ควบคุมระบบว่า ผู้บริหารฐานข้อมูล (Data Base Administrator) หรือ DBA โดยที่ DBA นี้อาจจะเป็นบุคคลผู้เดียว หรือ กลุ่มบุคคลก็ได้

สามารถจัดหาระบบความปลอดภัยที่รัดกุมได้ (security restrictions can be applied) คำว่าระบบความปลอดภัยในที่นี้หมายถึง การป้องกันไม่ให้ผู้ใช้ที่ไม่มีสิทธิ์มาใช้ข้อมูลในระบบได้ เนื่องจาก DBA เป็นผู้ที่ควบคุมการใช้ข้อมูล เขาจึงสามารถกำหนดสิทธิการใช้ให้แก่ผู้ใช้คนใด ๆ ก็ได้ตามความเหมาะสม และผู้ใช้แต่ละคนก็อาจจะใช้ข้อมูลได้ในระดับที่ต่างกัน หรือ พูดอีกนัยหนึ่งคือ ผู้ใช้แต่ละคนจะมองฐานข้อมูลด้วยวิวที่ต่างกัน โดยที่ถ้า DBA ไม่ได้รวมข้อมูลใดไว้ในวิวของผู้ใช้แล้ว ผู้ใช้คนนั้นก็จะไม่มีสิทธิเรียกใช้ข้อมูลส่วนนั้น นอกจากนี้ DBA ยังสามารถกำหนดรหัสลับในการเรียกใช้ข้อมูลบางส่วนได้อีกด้วย

สามารถควบคุมความคงสภาพของข้อมูลได้ (integrity can be maintained) ตัวอย่างอันหนึ่งของความไม่คงสภาพของข้อมูลคือ การที่เกิดความขัดแย้งของข้อมูลดังที่ได้กล่าวไปแล้ว ซึ่งกรณีของความขัดแย้งนี้จะเกิดขึ้นได้ก็เมื่อข้อมูลมีความซ้ำซ้อนเท่านั้น แต่ในอีกแง่หนึ่งของความคงสภาพที่เราจะศึกษากันในที่นี้อาจเกิดขึ้นได้ แม้ว่าจะไม่มีความซ้ำซ้อน ยกตัวอย่างเช่น ข้อมูลเกี่ยวกับอายุของพนักงานในบริษัทอาจจะมีค่า 300 แทนที่จะเป็น 30 ซึ่งความผิดพลาดแบบนี้เกิดขึ้นได้ง่าย ๆ จากความสะเพร่าในการพิมพ์ข้อมูลก็ได้ ในลักษณะของความไม่ถูกต้องเช่นนี้ ผู้ที่ออกแบบระบบฐานข้อมูลสามารถใส่กฎเกณฑ์เพื่อควบคุมความคงสภาพไว้ เช่น ตามตัวอย่างนี้ก็อาจจะใส่กฎว่า ค่าของอายุจะต้องเป็นตัวเลขระหว่าง 16 ถึง 60 เป็นต้น ดังนั้น เมื่อมีการใส่ข้อมูลใหม่หรือแก้ไขข้อมูล DBMS ก็จะควบคุมดูแลให้ข้อมูลดังกล่าวถูกต้องตามกฎเกณฑ์

เรื่องของความคงสภาพของข้อมูลกับการที่ผู้ใช้หลายคนใช้ข้อมูลร่วมกันนี้มีความสำคัญมากเมื่อเทียบกับการใช้ไฟล์ข้อมูลส่วนตัวอยู่คนเดียว เพราะการที่มีผู้ใช้หลายคนนั้น ทำให้โอกาสที่ผู้ใช้คนใดคนหนึ่งจะพลั้งเผลอแก้ไขข้อมูลผิดพลาดไปมีมากขึ้น และความผิดพลาดดังกล่าวยังกระทบกระเทือนการใช้ข้อมูลของผู้อื่นทั้งหมด ดังนั้นความสามารถของ DBMS ในเรื่องนี้จึงถือเป็นเรื่องที่มีความหมายไม่น้อยทีเดียว

สามารถสร้างสมดุลในความขัดแย้งของความต้องการได้ (conflicting requirements can be balanced) การที่ผู้ใช้ทั้งหมดขององค์กรใช้ข้อมูลจากฐานข้อมูลร่วมกันเช่นนี้ ทำให้ DBA ทราบถึงความต้องการและความสำคัญของผู้ใช้งานทั้งหมด จึงสามารถกำหนดโครงสร้างของฐานข้อมูลเพื่อให้บริการที่ดีที่สุดได้ เช่น เลือกเก็บข้อมูลที่จะต้องใช้บ่อย ๆ ไว้ในสื่อข้อมูลที่มีความเร็วเป็นพิเศษ เป็นต้น เป็นการสร้างสมดุลของความต้องการไม่ให้เกิดความขัดแย้งในหมู่ผู้ใช้ เพราะการออกแบบนั้นกระทำบนแนวทางที่มุ่งจะให้ประโยชน์ส่วนรวมดีที่สุดแล้ว

เกิดความเป็นอิสระของข้อมูล (data independence) วิธีที่จะทำความเข้าใจว่า ความเป็นอิสระของข้อมูลนั้นคืออะไร ก็โดยการลองดูในด้านตรงข้ามกันก่อนว่า ข้อมูลที่ไม่เป็นอิสระนั้นเป็นอย่างไร ลักษณะของข้อมูลที่ไม่เป็นอิสระคือ ข้อมูลที่ถูกนำมาประยุกต์ใช้ยังมีความผูกพันอยู่กับวิธีการจัดเก็บและเรียกใช้ข้อมูล ซึ่งในลักษณะการเขียนโปรแกรมประยุกต์บางประเภท เราอาจจำเป็นต้องใส่เทคนิคการจัดเก็บ และเรียกใช้ข้อมูลไว้ในตัวโปรแกรมเสียด้วย นั่นก็หมายความว่า ถ้าเกิดต้องมีการเปลี่ยนแปลงวิธีการจัดเก็บ หรือ การเรียกใช้ข้อมูลแล้ว ผู้ใช้ก็จำเป็นที่จะต้องสร้างวิธีการประยุกต์ใช้ขึ้นมาใหม่ ซึ่งเป็นความไม่สะดวกอย่างยิ่ง และทำให้เราหมดโอกาสที่จะปรับปรุงโครงสร้างของข้อมูล เพื่อให้ใช้งานได้อย่างมีประสิทธิภาพมากขึ้น

โดยสรุปแล้ว การใช้ระบบฐานข้อมูล จะทำให้เกิดความเป็นอิสระระหว่างการจัดเก็บข้อมูลและการประยุกต์ใช้ ทั้งนี้ ก็เพราะส่วนของการจัดเก็บข้อมูลจริง ๆ ถูก " ซ่อน " ออกจากวิวของการใช้งานระบบการจัดการฐานข้อมูล

จะเห็นว่า DBMS ที่ปรากฏอยู่ในท้องตลาดปัจจุบัน จะมีหลาย ๆ ตัวที่ไม่มีคุณสมบัติครบถ้วนตามนี้ เช่น ถ้าเป็น DBMS ที่ใช้กับเครื่อง PC ที่มีผู้ใช้เพียงคนเดียวก็ไม่จำเป็นที่จะต้องมีคุณสมบัติตามข้อ 5 เป็นต้น ดังนั้นจึงเป็นสิ่งที่ผู้จัดหา DBMS จะต้องพิจารณาว่าฟังก์ชันการใช้งานที่ตนต้องการนั้นมีอะไรบ้าง และตรวจสอบให้แน่ใจว่า DBMS ที่กำลังจะใช้นั้นมีความสามารถครบถ้วนตามที่ต้องการหรือไม่

2.3 Microsoft Access 97

Microsoft Access 97 เป็นโปรแกรมฐานข้อมูลหนึ่งที่นิยมใช้งานกันอย่างแพร่หลายบนระบบปฏิบัติการ Windows 95 หรือ Windows NT เวอร์ชั่น 3.51 ขึ้นไป เนื่องจากเป็นโปรแกรมฐานข้อมูลที่มีความสามารถสูง ใช้งานง่าย และสามารถช่วยเราในการสร้างแอพพลิเคชั่นฐานข้อมูลแบบใช้งานเอง หรือ แอพพลิเคชั่นฐานข้อมูลบนระบบเครือข่ายก็ได้

ถ้าเราเคยได้ยินว่าโปรแกรมฐานข้อมูลนั้นใช้งานยาก และต้องมีความเข้าใจในการเขียนโปรแกรมด้วย คำกล่าวนี้ไม่เป็นจริงสำหรับ Access 97 เนื่องจากโปรแกรมนี้สามารถสร้างแอพพลิเคชั่นฐานข้อมูลมาได้ โดยที่เราไม่จำเป็นต้องไปศึกษาการเขียนโปรแกรมให้ยุ่งยาก และ Access 97 ยังมีเครื่องมือต่าง ๆ ที่เรียกว่า วิซาร์ด (Wizard) ที่ช่วยในการทำงานต่าง ๆ ให้รวดเร็วยิ่งขึ้นด้วย สรุปก็คือ

เราสามารถสร้างแอพพลิเคชั่นฐานข้อมูลขึ้นมาได้อย่างรวดเร็ว และไม่ต้องเสียเวลาศึกษาวิธีการเขียนโปรแกรมให้ยุ่งยากด้วย Access 97

สำหรับนักพัฒนาโปรแกรมมืออาชีพแล้ว Access 97 นั้น ยังมีความสามารถต่าง ๆ ที่ตอบสนองความต้องการในระดับสูง เช่น เชื่อมต่อกับระบบฐานข้อมูลอื่น ๆ ได้ง่าย การสร้างโปรแกรมฐานข้อมูลบนระบบเครือข่าย การนำข้อมูลในฐานข้อมูลออกเผยแพร่บนอินเตอร์เน็ตซึ่งทำได้อย่างง่ายดาย และยังมีภาษาโปรแกรมให้ใช้ ถ้าต้องการสร้างแอพพลิเคชั่นที่มีความซับซ้อน เป็นต้น

โปรแกรม Microsoft Access 97 เป็นส่วนหนึ่งในชุดโปรแกรม Mircrosoft Office 97 Professional Edition ซึ่งโปรแกรม Access 97 ได้เปลี่ยนแปลงหน้าตา และรูปแบบการติดต่อกับผู้ใช้ใหม่ เช่นเดียวกับโปรแกรมในชุด Microsoft Ofiice 97 นอกจากนี้ Access 97 ยังสนับสนุนความสามารถด้านอินเตอร์เน็ตด้วย ดังนั้น ถ้าเราวางแผนที่จะเผยแพร่ข้อมูลของเราผ่านอินเตอร์เน็ต หรืออินทราเน็ต โปรแกรม Access 97 สามารถช่วยเราได้

Access 97 นั้นสามารถทำงานต่าง ๆ ต่อไปนี้ได้

- ใช้สร้างแอพพลิเคชั่นฐานข้อมูล เช่น โปรแกรมควบคุมสินค้าคงคลัง โปรแกรมบันทึกเวลาเข้าออกของพนักงาน เป็นต้น โดยที่ใน Access 97 นั้น มีเครื่องมือต่าง ๆ ในการสร้างแอพพลิเคชั่นได้อย่างรวดเร็ว และใช้งานง่าย ซึ่งเราอาจจะไม่ต้องเขียนโปรแกรมเลยก็ได้
- มีเครื่องมือในการสอบถามข้อมูลต่าง ๆ จากฐานข้อมูล เพื่อนำผลลัพธ์ไปทำงานบางอย่าง เช่น เราอาจจะต้องการทราบว่ายอดขายสินค้าแต่ละอย่างเป็นเท่าไร เป็นต้น
- สามารถสร้างเครื่องมือในการติดต่อกับผู้ใช้ได้อย่างเหมาะสม เช่น การแสดงข้อมูลลูกค้าให้ผู้ใช้งานแก้ไขข้อมูลได้ เป็นต้น
- ช่วยเราในการสร้างรายงานจากฐานข้อมูลได้ เพื่อใช้ในการทำงานบางอย่าง เช่น พิมพ์ชื่อและที่อยู่ลูกค้า เพื่อทำฉลากติดซองจดหมายส่งข้อมูลไปยังลูกค้า เป็นต้น
- ช่วยให้เราสามารถเผยแพร่ข้อมูลขององค์กรที่อยู่ในฐานข้อมูล ผ่านทางอินเตอร์เน็ตและอินทราเน็ตได้อย่างง่ายดาย เนื่องจากใน Access 97 มีเครื่องมือที่ช่วยในการทำงานต่าง ๆ เหล่านี้อย่างครบถ้วน

ความต้องการของระบบในการใช้งาน Access 97

โปรแกรม MS Access 97 ต้องการระบบคอมพิวเตอร์ขั้นต่ำที่มีรายละเอียดต่าง ๆ ดังต่อไปนี้ในการใช้งาน

- CPU 486DX-33 ขึ้นไป

- RAM 12 MB สำหรับระบบปฏิบัติการ Windows 95
  RAM 16 MB สำหรับระบบปฏิบัติการ Windows NT
- ฮาร์ดดิสก์ที่มีเนื้อที่ว่าง 70 MB สำหรับการลง Access 97 แบบสมบูรณ์
- ระบบปฏิบัติการ Windows 95 หรือระบบปฏิบัติการ Windows NT เวอร์ชั่น 3.51 ขึ้นไป

2.4 บาร์โค้ด (Bar code)

การใช้บาร์โค้ดได้เพิ่มขึ้นอย่างมากเมื่อ 15 ปีที่ผ่านมา ด้วยการยอมรับจากจากระบบการค้าปลีกของสหรัฐอเมริกาในปลายทศวรรต 1970 โดยใช้บาร์โค้ดแบบ UPC (Universal Product Code) บาร์โค้ดนั้นช่วยให้การป้อนข้อมูลได้รวดเร็ว ง่าย และเที่ยงตรง ความถูกต้องที่ได้จากการใช้บาร์โค้ดสามารถลดเวลาการทำงานของพนักงานและเพิ่มประสิทธิภาพขององค์กร สิ่งหนึ่งที่ควรรับรู้เกี่ยวกับบาร์โค้ด คือ โปรแกรมประยุกต์ที่รับข้อมูลผ่านระบบบาร์โค้ดนั้นมีโอกาสที่จะใช้งานได้ผลถึง 95% มากกว่าที่จะล้มเหลว อย่างไรก็ตามบาร์โค้ดเป็นเพียงวิธีการป้อนข้อมูลแบบหนึ่งเท่านั้น สิ่งที่สำคัญคือจะนำข้อมูลไปใช้งานอะไร ด้วยการเริ่มใช้งานคอมพิวเตอร์ส่วนบุคคลของบริษัท IBM (IBM Personal Computer) ในช่วงต้นทศวรรต 1980 ทำให้มีการขยายตัวการใช้บาร์โค้ดไปพร้อม ๆ กับการขยายตัวของการใช้คอมพิวเตอร์ส่วนบุคคล

บาร์โค้ดเป็นรูปแบบของการสลับกันระหว่างส่วนที่มืดและสว่างซึ่งจะถูกถอดรหัสเป็นความหมายในรูปแบบที่เราเข้าใจ เช่น อยู่ในรูปของตัวอักษร เมื่อเครื่องอ่านบาร์โค้ด (barcode scanner)ทำการกวาดภาพของสัญลักษณ์บาร์โค้ดก็จะถูกถอดรหัสแปลงกลับสู่ข้อความเดิมที่เป็นตัวอักษร บาร์โค้ดส่วนมากประกอบด้วยรูปแบบของแท่งสี่เหลี่ยม แต่บาร์โค้ดระบบใหม่ ๆ บางชนิดก็จะใช้รูปทรงแบบอื่น ๆ บาร์โค้ดเป็นส่วนหนึ่งของการใช้งานในระบบการแยกแยะอัตโนมัติในอุตสาหกรรม ซึ่งบางครั้งก็ใช้เป็นระบบการป้อนข้อมูลโดยไม่ใช้แป้นพิมพ์

บาร์โค้ดสามารถอ่านข้อมูลได้ผิดพลาดน้อยมากหรือไม่ผิดพลาดเลยในสภาพการใช้งานในสถานที่ทำงานจริง ในบางสถานการณ์มันสามารถอ่านได้อย่างถูกต้องอย่างอัตโนมัติโดยไม่ต้องใช้คนเข้ามาเกี่ยวข้อง การอธิบายนี้จะมุ่งเน้นการสร้างและการพิมพ์บาร์โค้ดมากกว่าจะอธิบายถึงหลักการทำงานว่ามันอ่านค่าหรือถอดรหัสได้อย่างไร ตัวอ่านบาร์โค้ดจะอ่านบาร์โค้ดด้วยการปล่อยแสงเลเซอร์สีแดงของความถี่ที่กำหนดซึ่งจะตกกระทบภาพของแท่งบาร์โค้ด และช่วยแยกแยะขอบของแท่ง

บาร์โค้ดแต่ละแท่งเพื่อแยกแยะแต่ละแท่งได้ถูกต้อง และตีความหมายของระยะห่างของแต่ละแท่งออกมาเป็นตัวอักษรตามรูปแบบที่กำหนดไว้

บาร์โค้ดหลายชนิดมีการตรวจสอบความถูกต้องในการอ่านค่า ซึ่งใช้หลักการที่เรียกว่า การ checksum เพื่อเพิ่มความเที่ยงตรงในการอ่านค่า เมื่อพิมพ์บาร์โค้ดออกมา ข้อมูลของตัวอักษรทั้งหมดถูกรวมกันเข้า, คำนวนด้วยกฎเกณฑ์ที่กำหนด คิดแยกออกเป็นหมายเลขเฉพาะรหัสของของข้อความนั้นหมายเลข และถูกพิมพ์ออกมาที่ตัวท้ายของบาร์โค้ดก่อนตัวสุดท้ายที่บอกว่าสิ้นสุดบาร์โค้ด เมื่อตัวอ่านอ่านบาร์โค้ด มันจะคำนวณตัวเปรียบเทียบของค่าที่อ่านได้ เทียบกับค่าที่ควรจะเป็นตามหลักเกณฑ์ ที่กำหนดไว้ ซึ่งถ้ามันไม่ตรงกัน แสดงว่ามีข้อผิดพลาดบางอย่างเกิดขึ้น และมันจะยกเลิกการอ่านค่าครั้งนั้น

บาร์โค้ดมีหลายชนิด มีสัญลักษณ์และการถอดรหัสที่ต่างกัน ชนิดต่าง ๆ ของบาร์โค้ดมาจากต่าของสัญลักษณ์ที่แตกต่างกันเกี่ยวข้องกับการแก้ปัญหาที่แตกต่างกัน รหัสชนิด UPC ถูกพัฒนาขึ้นสำหรับความต้องการใช้งานด้านการค้าปลีก มันกำหนดค่า 12 หลักในพื้นที่ที่ค่อนข้างจำกัดได้อย่างเหมาะสม รหัสชนิด 39 (Code 39) ถูกพัฒนาเพราะความต้องการด้านอุตสาหกรรมที่ต้องการการถอดรหัสทั้งตัวอักษรและตัวเลข รหัสชนิด 128 (Code 128) มาจากความต้องการอักษรที่มากกว่ารหัสชนิด 39 ซึ่งตารางต่อไปนี้จะแสดงตัวอย่างข้อแตกต่างของบาร์โค้ดบางชนิด

ชนิดของบาร์โค้ด (แบบมิติเดียว)	แทนค่าได้เฉพาะค่าตัวเลข	แทนค่าเป็นตัวอักษร	จำนวนรูปแบบตัวอักษรที่แทนค่า	จำนวนตัวอักษรสูงสุดต่อนิ้ว	ปรับความยาวบาร์โค้ดต่อนิ้ว
Int 2 of 5	ได้	ไม่ได้	10	17.8	ไม่ได้
Code 39	ไม่ได้	ได้	43	9.4	ได้
Codabar	ได้	ไม่ได้	16	10	ได้

ตารางที่ 2.1 การเปรียบเทียบความแตกต่างของบาร์โค้ดชนิดต่าง ๆ บางชนิด

ซึ่งในการศึกษาครั้งนี้จะเลือกใช้บาร์โค้ดแบบ 39 ซึ่งมีจุดเด่นที่สำคัญคือ

1. แสดงผลได้ทั้งตัวเลขและตัวอักษร
2. สามารถปรับความยาวของแถบบาร์โค้ดได้
3. อุปกรณ์ที่ใช้อ่านค่า (Barcode Reader) หาซื้อได้ง่าย มีใช้แพร่หลาย
4. ปัจจุบันมีแบบของตัวอักษร (Font) ที่เป็นบาร์โค้ดแบบ 39 ซึ่งใช้บนระบบปฏิบัติการ Microsoft Windows ซึ่งสามารถนำไปประยุกต์ใช้งานได้หลากหลาย

บาร์โค้ดแบบ 39 มีรายละเอียดดังต่อไปนี้

2.4.1 บาร์โค้ดแบบ 39 (Code 39)

Code 39 (ย่อมาจาก Code 3 of 9) เป็นชนิดของบาร์โค้ดที่นิยมใช้กันมากในการระบุตัวบุคคล , ระบบวัสดุคงคลัง และเพื่อการติดตามอื่น ๆ ตามต้องการ ตัวอย่างใช้งานที่เห็นได้ในชีวิตประจำวัน เช่น ในร้านเช่าวีดีโอเทป ซึ่งมักจะใช้บาร์โค้ดแบบ Code 39 ในการติดตามการให้เช่าวีดีโอที่ถูกอ่านที่เคาน์เตอร์

ประโยชน์ทั่ว ๆ ไปของบาร์โค้ดชนิด Code 39

- สามารถใช้ตัวอ่านบาร์โค้ด (barcode scanner) ทั่ว ๆ ไปในการอ่านค่าได้
- สามารถใช้งานทั้งในด้านตัวอักษร (ตัวอักษรภาษาอังกฤษแบบตัวพิมพ์ใหญ่) หรือค่าตัวเลขได้
- ความยาวของข้อความที่แปลงเป็นบาร์โค้ดไม่บังคับตายตัว สามารถปรับความยาวของข้อความได้ (เพิ่มหรือลดจำนวนตัวอักษรในข้อความ)
- แทบไม่จำเป็นต้องใช้หลักการตรวจสอบตัวเอง แต่ก็สามารถใช้ได้ถ้าจำเป็น
- ความกว้างของแถบบาร์โค้ด สามารถปรับกว้างมาก หรือน้อยได้ โดยที่ยังสามารถอ่านค่าได้
- ตัวอักษรที่สามารถใช้งานได้สำหรับบาร์โค้ดชนิด Code 39 ได้แก่ อักษร A ~ Z (ตัวพิมพ์ใหญ่) , 0 ~ 9 , $, % , + , - , . , / , ช่องไฟ (space) และเครื่องหมายดอกจันทน์ (แต่จะสงวนไว้ไม่ใช้ในการสื่อความหมาย แต่ใช้เป็นตัวเริ่มต้นและปิดท้ายของข้อความที่แปลงเป็นบาร์โค้ด) สำหรับการใช้งานในอักษรระบบคอมพิวเตอร์ ASCII ซึ่งรวมอักษรตัวพิมพ์เล็กด้วย เป็น 128 ตัวอักษรนั้น สามารถทำได้โดยใช้ตัวอักษรร่วมกันสองตัว เช่นถ้าต้องการให้บาร์โค้ดแปลงค่าออกมาเป็น “a” ก็ต้องใช้บาร์โค้ดค่า “+A” สำหรับการใช้งานที่ต้องการความถูกต้องของการอ่านข้อมูลสูง ก็สามารถใช้ส่วนเสริมเกี่ยวกับการตรวจสอบตัวเองได้ (เช่นในกิจการเกี่ยวกับสุขภาพ)

สัดส่วนของขนาดของบาร์โค้ดชนิด Code 39 จริง ๆ แล้วปรับเปลี่ยนได้ซึ่งขึ้นอยู่กับพื้นที่ที่ต้องการพิมพ์ วิธีการพิมพ์ตัวบาร์โค้ดออกมา สิ่งที่สำคัญก็คือขนาดความที่ต้องการพิมพ์ ข้อจำกัดในการเปลี่ยนจากตัวอักษรหรือข้อความมาเป็นบาร์โค้ด คือตัวเริ่มต้นและตัวจบของข้อความที่พิมพ์ออกมาเป็นบาร์โค้ดต้องเป็นอักษร “*” เช่น ถ้าเราต้องการพิมพ์บาร์โค้ดเพื่อให้อ่านว่า “STORYHOUSE” เราต้องป้อนข้อมูลเป็น “*STORYHOUSE*” เมื่ออ่านด้วยเครื่องอ่านบาร์โค้ดแล้วจะได้ข้อความ “STOTYHOUSE” ตามที่ต้องการ โดยจะไม่เห็นตัวอักษร “*”

ภาพที่ 2.4 ตัวอย่างบาร์โค้ดแบบ39

แต่ละตัวอักษรที่แทนด้วยบาร์โค้ดจะต้องประกอบด้วยแท่งขาวสลับดำจำนวน 9 แท่ง (จริง ๆ แล้ว "แท่งขาว" ก็คือพื้นที่ส่วนที่เว้นไว้ หรือคั่นระหว่างแถบดำ และขนาดสัดส่วนของแต่ละแท่งจะมีอยู่สองขนาด คือช่องปกติมีค่าเป็น 1 และช่องขนาดที่สองจะมีค่าประมาณ 2 ~ 3 เท่า (สัดส่วน 2.1 ~ 3.0 : 1 บางกลุ่มอุตสาหกรรมจะระบุเจาะจงที่ขนาด 2.2 : 1 เพื่อให้ได้ขนาดที่เล็ก

Code 39 สามารถรับรู้จุดเริ่มต้นและจุดสิ้นสุดของบาร์โค้ดได้จากการใช้เครื่องหมายดอกจันทร์ (*) ซึ่งเครื่องหมายดอกจันทร์มีรูปแบบดังนี้ แท่งแคบ , ช่องกว้าง , แท่งแคบ , ช่องแคบ , แท่งกว้าง , ช่องแคบ , แท่งกว้าง , ช่องแคบ , แท่งกว้าง

ภาพที่ 2.5 บาร์โค้ดแบบ39 ของตัวอักษร "*" (ดอกจันทร์)

Code 39 (บางครั้งเรียก Code 3 from 9) มีลักษณะเป็นบาร์โค้ดแบบไม่ต่อเนื่อง (discrete barcode) ซึ่งหมายถึงว่าแต่ละตัวอักษรจะมีรูปแบบของการเรียงของแท่งแบบเฉพาะจงจง ซึ่งแต่ละตัวอักษรจะไม่ซ้ำกัน แต่ละตัวอักษรจะแทนด้วยแท่ง 9 แท่ง, 3 แท่งจะเป็นแท่งแบบกว้าง (แท่งในที่นี้หมายถึงแท่งสีดำ และช่องว่างระหว่างแท่งก็ถือว่าเป็นแท่ง เช่นเมื่อพิมพ์บาร์โค้ดลงบนกระดาษสีขาว เราจะเห็นช่องว่างระหว่างแท่งสีดำเหมือนเป็นแท่งสีขาว คือเมื่อมองบาร์โค้ดที่พิมพ์ออกมาทั้งแถบ จะเห็นเป็นแท่งสีดำสลับด้วยแท่งสีขาว) ซึ่งเมื่อมองดูแล้วหนึ่งตัวอักษรจะประกอบด้วยแท่งสีดำ 5 แท่ง และแท่งสีขาว 4 แท่งสลับกัน

สัดส่วนความกว้างของแต่ละแท่งจะมีขนาดสัดส่วนอยู่ระหว่าง 2.2 ~ 3 : 1 การอ่านบาร์โค้ดจะให้ได้ค่าที่ถูกต้องนั้น เครื่องอ่านบาร์โค้ดหรือเครื่องถอดรหัสต้องสามารถแยกยะว่าเป็นแท่งแบบแคบ หรือแบบกว้างได้ ในทางปฏิบัติเพื่อให้ได้การอ่านค่าที่แม่นยำ ควรใช้บาร์โค้ดที่มีค่าสัดส่วนความกว้างของแท่งใกล้เคียงค่า 3 : 1 ซึ่งจะสามารถลดโอกาสที่จะอ่านค่าผิดพลาดได้ถึง 50 %

ระหว่างบาร์โค้ดที่เรียงติดกันของแต่ละตัวอักษรจะมีช่องว่างที่เรียกว่า ช่องว่างระหว่างตัวอักษร (The inter-character gap) ซึ่งไม่มีข้อกำหนดเกี่ยวกับความกว้างของช่องว่างระหว่างตัวอักษร แต่โดยทั่ว ๆ ไปจะมีความกว้างใกล้เคียงกับแท่งขนาดแคบสีขาว โดยปกติแล้วการพิมพ์บาร์โค้ดควรจะได้ขนาดของแท่งแต่ละแบบคงที่ เช่น แท่งขนาดแคบสีดำทุก ๆ แท่งควรมีขนาดเท่ากัน แต่ในทางปฏิบัติเมื่อพิมพ์ออกมาแล้วจะไม่เท่ากันเนื่องจากการซึมของน้ำหมึกบนบริเวณขอบของแท่งสีดำ ซึ่งจะทำให้ขนาดของแท่งสีดำกว้างขึ้น และทำให้ขนาดของแท่งสีขาวแคบลง (แท่งสีขาวก็คือส่วนของช่องว่างที่ไม่ได้พิมพ์อะไร) ซึ่งทำให้ขนาดสัดส่วนของแท่งไม่อยู่ในช่วง 2.1 ~ 3 : 1 ตามที่กำหนด

ซึ่งมีผลอย่างมากในการอ่านค่าของบาร์โค้ดขนาดเล็ก (คือบาร์โค้ดที่บีบสัดส่วนทั้งหมดให้เล็กลงเพื่อพิมพ์ในพื้นที่จำกัด) เพราะในจำนวนอักษรเท่ากัน Code 39 สามารถพิมพ์ออกมาให้ความยาวโดยรวมมากหรือน้อยได้ เช่น อาจพิมพ์ออกมาในความกว้างรวม 1 ซ.ม. หรือ 2 ซ.ม. ก็ได้ โดยที่แต่ละแท่งยังคงสัดส่วน 2.2 ~ 3 : 1 แต่ถ้าพิมพ์ออกมาในขนาดเล็กการซึมของหมึกเพียงเล็กน้อยจะทำให้แท่งสีดำมีขนาดเพิ่มขึ้นเล็กน้อย แต่เมื่อเทียบเป็นสัดส่วนแล้วจะขยายขึ้นมาก ทำให้ไม่สามารถคงสัดส่วนขนาดของแท่งละแท่งได้ ทำให้การอ่านค่ามีโอกาสผิดพลาดได้มากขึ้นหรือไม่สามารถอ่านค่าได้ โดยเฉพาะถ้าเครื่องใช้กับอ่านบาร์โค้ดที่มีประสิทธิภาพต่ำ (แยกแยะขนาดได้ไม่ดีพอ)

Code 39 จะมีรูปแบบของแท่งที่แทนตัวอักษรได้ 43 ตัว (อย่างที่กล่าวมาแล้ว)และรวมถึงตัวที่ทำหน้าที่เป็นตัวบอกเริ่มต้นหรือสิ้นสุดของบาร์โค้ด (คือเครื่องหมายดอกจันทร์) นั่นหมายถึงเมื่อเราพิมพ์บาร์โค้ดของข้อความหรือกลุ่มของตัวอักษรที่ต้องการแล้ว ที่บาร์โค้ดต้องเริ่มต้นและปิดท้ายด้วยเครื่องหมายดอกจันทร์ (*) เสมอ แต่เมื่อใช้เครื่องอ่านบาร์โค้ดอ่านค่า จะได้แต่ค่าของข้อความหรือกลุ่มของตัวอักษรที่ต้องการ โดยไม่มีเครื่องหมายดอกจันทร์ออกมาด้วย

โดยสรุปแล้วบาร์โค้ดชนิด Code 39 มีจุดเด่นต่าง ๆ ดังนี้

- ด้วยคุณสมบัติที่สามารถอ่านค่าออกมาเป็นได้ทั้งค่าตัวเลขและตัวอักษร
- แสดงความยาวของข้อความได้ไม่จำกัดตายตัว (จำนวนตัวอักษรได้หลายตัว เช่น อาจเป็น 3 , 4 , 5 , ... ตัวอักษร ข้อจำกัดอยู่ที่ความสามารถในการอ่านของเครื่องอ่านว่าอ่านข้อความได้กว้างสุดเท่าไร กี่ ซ.ม.
- อ่านได้จากทั้งจากซ้ายไปขวา หรือขวาไปซ้าย (Bi-directional scanning จะได้ข้อความออกมาเหมือนการอ่านหนังสือปกติ คือจะได้ข้อความจากซ้ายไปขวา)
- สามารถบรรจุตัวอักษรได้ 3 – 9.4 ตัวอักษร / นิ้ว (รวมเครื่องหมายดอกจันทร์ในการบอกจุดเริ่มต้นกับจุดสิ้นสุดข้อความ)
- พื้นที่ว่างรอบ ๆ บาร์โค้ดประมาณ 10 เท่าของช่องแบบแคบ (Quiet zone คือถ้ามีขีดหรือแท่งอื่นใดอยู่ในขอบเขตดังกล่าว อาจทำให้การอ่านผิดพลาดเพราะเข้าใจว่าขีดหรือแท่งดังกล่าวเป็นส่วนหนึ่งของบาร์โค้ด)
- จุดเด่นในการทดสอบตัวเองป้องกันการอ่านผิดพลาด

ซึ่งทำให้ Code 39 เป็นที่นิยมใช้งานและยอมรับมากที่สุดในงานด้านอุตสาหกรรม

2.4.2 สีของบาร์โค้ด

เครื่องอ่านบาร์โค้ดส่วนมากจะใช้สีในย่านแสงเดียว (monochromatic)ในการแยกแยะแท่งบาร์โค้ดแต่ละแท่ง ซึ่งส่วนใหญ่จะใช้แสงในย่านสีแดงของสเปกตรัม บางครั้งก็ใช้แสงในย่านอินฟราเรด (infra red) ซึ่งมนุษย์มองไม่เห็น

การแยกความแตกต่างที่ดีที่สุดมาจากการใช้พื้นหลัง (background) ที่สะท้อนแสงได้ดี และสีที่พิมพ์แท่งบาร์โค้ดจะไม่สะท้อนแสง แม้ว่าในทางปฏิบัติแล้วจะไม่ให้ผลที่ดีที่สุดเสมอไป แต่ก็นับว่ามีผลช่วยให้การอ่านค่าน่าเชื่อถือมากขึ้น

โดยทั่วไปแล้วถ้าพิมพ์แท่งบาร์โค้ดด้วยสีดำลงบนพื้นหลังที่เป็นสีขาวจะให้การตัดกันที่เด่นชัดระหว่างแท่งบาร์โค้ดสีดำกับสีขาว ทำให้การแยกแยะแท่งบาร์โค้ดของเครื่องบาร์โค้ดอ่านค่าได้ถูกต้องมากขึ้น ในกรณีที่เครื่องอ่านบาร์โค้ดทำงานโดยอาศัยแสงในย่านอินฟราเรดต้องระวังเรื่องหมึกที่ใช้พิมพ์บาร์โค้ด เนื่องจากหมึกดำบางชนิดจะสะท้อนแสงได้ทำให้เครื่องแยกแยะความแตกต่างระหว่างแท่งบาร์โค้ดสีดำกับสีขาวได้ไม่ดี ซึ่งส่งผลให้เครื่องอ่านบาร์โค้ดอ่านค่าผิดพลาดถึงแม้ว่าจะเห็นว่ามีความแตกต่างกันชัดเจนเมื่อมองด้วยตาเปล่า หมึกที่มีปัญหาดังกล่าว เช่นหมึกที่ทำจากพืชผัก (vegetable dyes) ซึ่งมักใช้ในผ้าหมึกของเครื่องพิมพ์แบบเข็ม (dot matrix printer) ดังนั้นควรใช้หมึกที่ได้จากถ่าน (carbon pigment) จะให้ผลที่ดีกว่า

นอกจากนี้เครื่องอ่านที่ใช้แสงย่านอินฟราเรดยังใช้ได้ไม่ดีกับกระดาษความร้อน (thermal paper เช่นกระดาษเครื่องโทรสาร) ปัจจุบันนิยมเครื่องพิมพ์บาร์โค้ดแบบใช้ความร้อน (thermally printer) โดยใช้แถบความร้อน (thermal transfer ribbon) ซึ่งเป็นผงหมึกคาร์บอนเคลือบด้วยขี้ผึ้งติดอยู่แบนแผ่นเทป และใช้ความร้อนในการลอกลายแท่งบาร์โค้ดหรือหมึกลงบนกระดาษ ด้วยวิธีนี้ช่วยให้แยกแยะความแตกต่างระหว่างแท่งบาร์โค้ดได้ดีขึ้นสำหรับเครื่องอ่านแบบต่าง ๆ

อย่างไรก็ดีมีบางกรณีมีความจำเป็นต้องใช้หมึกสีอื่น ๆ ในการพิมพ์บาร์โค้ดลงบนพื้นหลังที่บางครั้งก็ไม่ใช่สีขาว ก็ควรใช้หลักการเกี่ยวกับการแยกความแตกต่างดังกล่าวในการเลือกสีทั้งหมึกและสีของพื้นหลัง โดยสีของพื้นหลังควรสะท้อนได้ดีในย่านแถบสีแดงและสีของหมึกสะท้อนได้ดีในย่านแถบสีน้ำเงิน (ไม่สะท้อนแสงในย่านสีแดง) และบางครั้งก็ควรคำนึงถึงสภาพการใช้งานจริงที่อาจมีวัสดุบางอย่างเคลือบอยู่บนบาร์โค้ดอีกที เช่น มีแผ่นใสหุ้มบริเวณบาร์โค้ดซึ่งอาจส่งผลให้การอ่านค่าผิดพลาดได้ อย่างไรก็ดีเราอาจใช้หลักการกลับกันในการพิมพ์บาร์โค้ด เช่น การพิมพ์สีขาวลงบนพื้นหลังสีดำ แต่ก็ไม่ควรทำถ้าไม่จำเป็นจริง ๆ