

การออกแบบและพัฒนาโปรแกรม

งานวิจัยนี้เป็นการพัฒนาเครื่องมือทางซอฟต์แวร์สำหรับแปลงผันเพิ่มข้อมูล จากวัตถุประสงค์ และขอบเขตของการวิจัยที่ได้กล่าวไปแล้วในบทที่ 1 จึงแบ่งการออกแบบและการพัฒนาโปรแกรมออกเป็น 4 ขั้นตอนดังนี้

3.1 การออกแบบและพัฒนาเครื่องมือทางซอฟต์แวร์สำหรับแปลงผันเพิ่มข้อมูลรายงาน ให้อยู่ในรูปแบบที่สามารถนำไปใช้งานต่อไปได้

3.2 การออกแบบและพัฒนาเครื่องมือทางซอฟต์แวร์สำหรับแปลงผันเพิ่มข้อมูลซี, เพิ่มข้อมูลโคบอล, เพิ่มข้อมูลปาสกาล, เพิ่มข้อมูลเบสิก ให้เป็นเพิ่มข้อความ

3.3 การประยุกต์ใช้ซอฟต์แวร์สำหรับแปลงผันเพิ่มข้อมูลบีทรีฟและฐานข้อมูลอวาเคลให้เป็นเพิ่มข้อความ

3.4 การออกแบบและพัฒนาตัวเชื่อมประสานกับผู้ใช้ (User Interface)

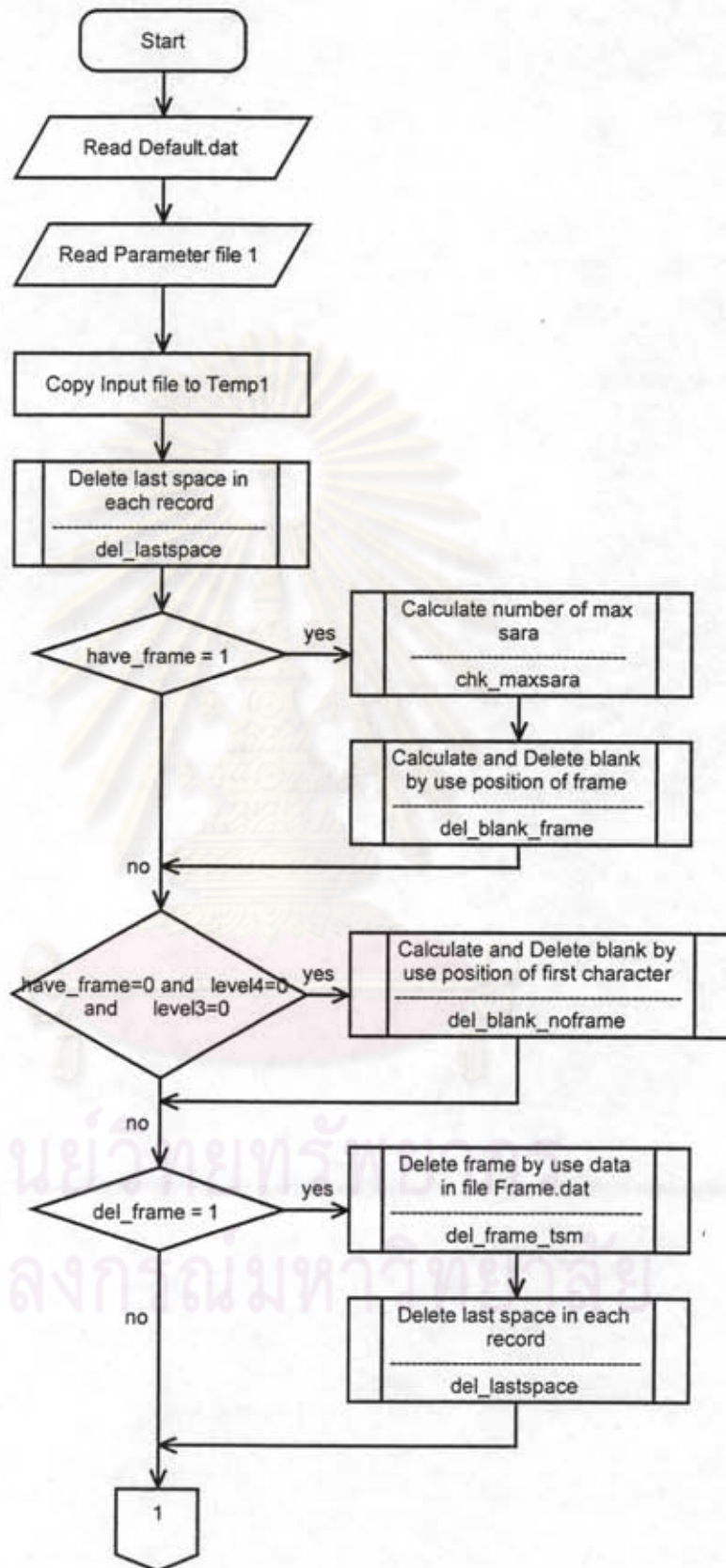
3.1 การออกแบบและพัฒนาเครื่องมือทางซอฟต์แวร์สำหรับแปลงผันเพิ่มข้อมูลรายงาน ให้อยู่ในรูปแบบที่สามารถนำไปใช้งานต่อไปได้

จากการศึกษา, วิเคราะห์ และรวบรวม เกี่ยวกับรูปแบบของเพิ่มข้อมูลรายงานที่ได้มาจากเครื่องคอมพิวเตอร์ขนาดใหญ่, เครื่องคอมพิวเตอร์ขนาดกลาง และเครื่องคอมพิวเตอร์ขนาดเล็ก และเพื่อเป็นการแก้ไขปัญหาต่าง ๆ ที่ได้รับไว้ในขอบเขตของการวิจัย จึงแบ่งลักษณะของเครื่องมือทางซอฟต์แวร์ออกเป็น 2 ส่วน โดยแยกตามหน้าที่ของการทำงาน ได้ดังนี้

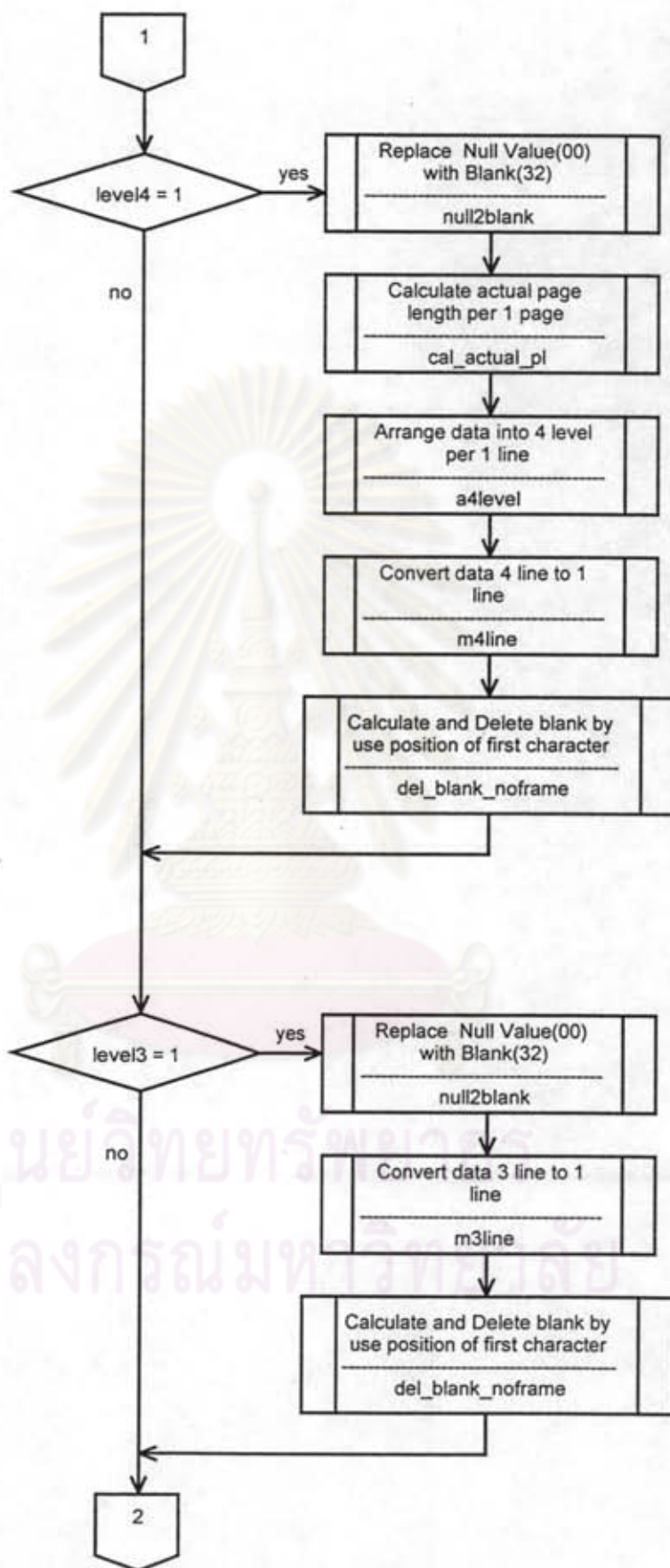
3.1.1 เครื่องมือทางซอฟต์แวร์ตัวที่ 1 (SWTOOL1) มีหน้าที่

- ตรวจสอบและตัดข้อมูลที่เป็นการรอบตารางออก
- ตรวจสอบและทำการแปลงข้อมูลภาษาไทยที่เก็บแบบ 4 ระดับหรือ 3 ระดับให้เหลือเพียง 1 ระดับ โดยนำมาเก็บในแนวเดียวกัน
- จัดการข้อมูลภาษาไทยให้อยู่ในแนวเดียวกัน ในกรณีที่ข้อมูลในแต่ละคอลัมน์ไม่อยู่ในแนวเดียวกัน
- ลบส่วนหัวของเพิ่มข้อมูลรายงาน, ลบส่วนสรุปของเพิ่มข้อมูลรายงาน, ลบขีดเส้นใต้
- สามารถเลือกดึงข้อมูลเฉพาะบางคอลัมน์ได้

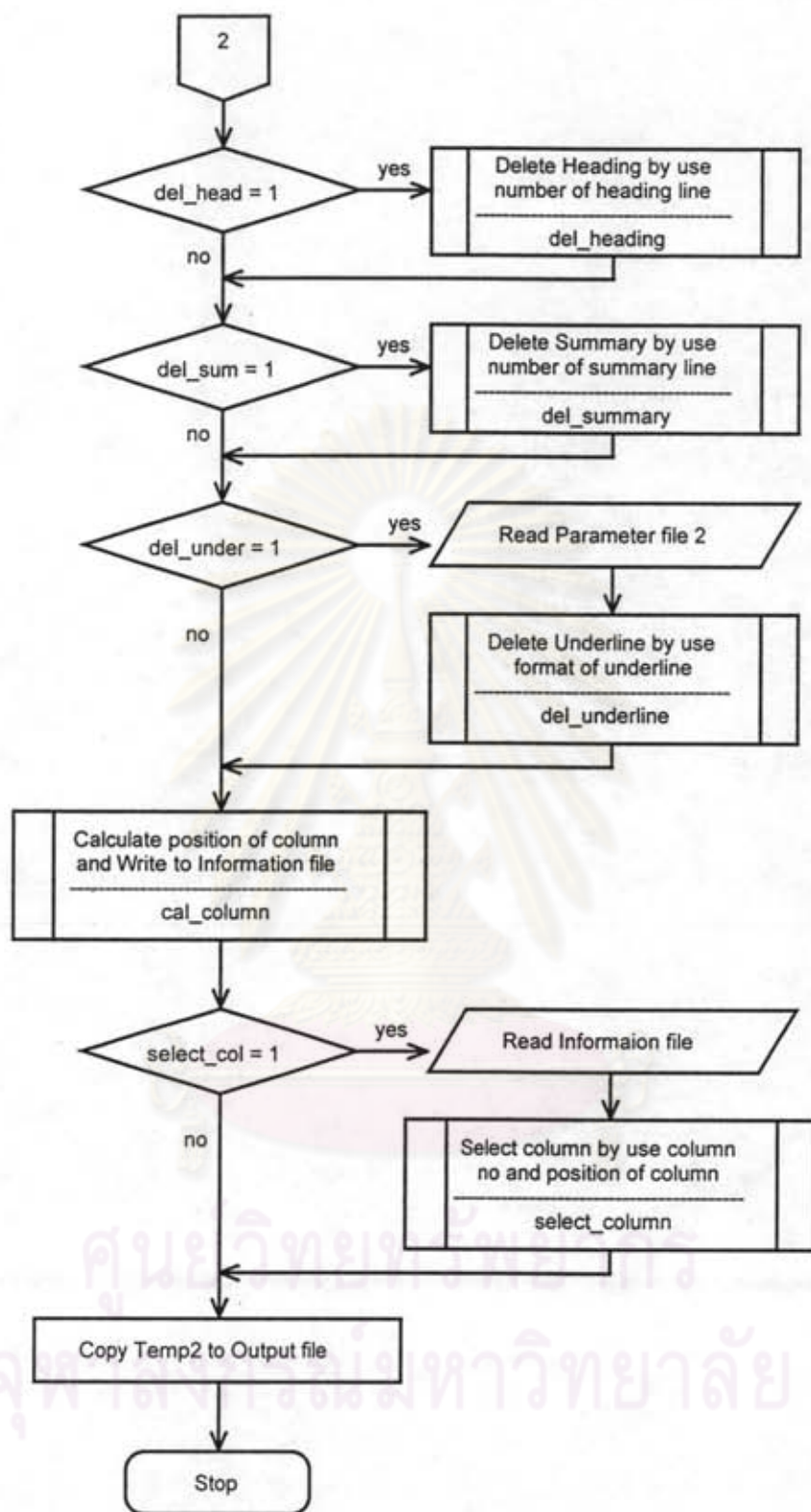
3.1.2 รูปแบบการทำงานของโปรแกรม SWTOOL1



รูปที่ 3.1 แสดงรูปแบบการทำงานของโปรแกรม SWTOOL1



รูปที่ 3.1 แสดงรูปแบบการทำงานของโปรแกรม SWTOOL1 (ต่อ)



รูปที่ 3.1 แสดงรูปแบบการทำงานของโปรแกรม SWTOOL1 (ต่อ)

3.1.3 คำอธิบายขั้นตอนการทำงานของโปรแกรม SWTOOL1

ขั้นตอนที่ 1 โปรแกรมทำการอ่านค่าจากแฟ้มข้อมูล Default.dat ซึ่งจะเก็บข้อมูลเป็นตัวเลขเรียงกันดังนี้

70 2 33 127 161 207 222 230

ความหมายของข้อมูลอธิบายดังตารางที่ 3.1

ข้อมูล	ความหมาย
70	ความยาวของตัวอักษรที่ใช้ในการเปรียบเทียบส่วนหัวของรายงานที่ต้องการลบ โดยใช้ข้อมูลบรรทัดแรกในแฟ้มข้อมูลรับเข้าเป็นหลักในการเปรียบเทียบซึ่งในกรณีนี้จะเปรียบเทียบเพียง 70 ตัวอักษร
2	จำนวนช่องว่างที่ใช้ในการแบ่งคอลัมน์ ในกรณีนี้หมายความว่าถ้าเจอช่องว่างมากกว่า 2 ช่องติดกัน จะถือเป็นอีกคอลัมน์หนึ่ง
33 127 161 207 222 230	ช่วงของตัวอักษรที่ใช้ในการเปรียบเทียบว่าอยู่ในระดับเดียวกับพยัญชนะ หรือไม่ โดยแบ่งออกเป็น 3 ช่วงคือ ช่วงที่ 1 - แอสกี 33 (I) ถึง แอสกี 127 (A) ช่วงที่ 2 - แอสกี 161 () ถึง แอสกี 207 (๗) ช่วงที่ 3 - แอสกี 222 (ไ) ถึง แอสกี 230 (๓)

ตารางที่ 3.1 แสดงคำอธิบายข้อมูลในแฟ้มข้อมูล Default.dat

ขั้นตอนที่ 2 โปรแกรมทำการอ่านค่าจากแฟ้มข้อมูลพารามิเตอร์ตัวที่ 1 (Filename.pr1) ซึ่งจะเก็บข้อมูลจากการที่ผู้ใช้ป้อนทางจอภาพ ตัวอย่าง เช่น

0 0 1 0 1 7 1 3 1 1 2 1 3 5

ความหมายของข้อมูลแต่ละตัวอธิบายดังตารางที่ 3.2

ข้อมูล	ความหมาย	ค่าที่เป็นไปได้
0	มีรอบตาราง ? (y/n)	0-ไม่มี, 1-มี
0	ลบรอบตาราง ? (y/n)	0-ไม่ลบ, 1-ลบ

ตารางที่ 3.2 แสดงคำอธิบายข้อมูลในแฟ้มข้อมูลพารามิเตอร์ตัวที่ 1 (SWTOOL1)

ข้อมูล	ความหมาย	ค่าที่เป็นไปได้
1	ภาษาไทยแบบ 4 ระดับ ? (y/n)	0-ไม่เป็น 4 ระดับ 1-เป็น 4 ระดับ
0	ภาษาไทยแบบ 3 ระดับ ? (y/n)	0-ไม่เป็น 3 ระดับ 1-เป็น 3 ระดับ
1	ลบส่วนหัวของรายงาน ? (y/n)	0-ไม่ลบ, 1-ลบ
7	จำนวนบรรทัดของส่วนหัวรายงานที่ต้องการลบ	ตัวเลข
1	ลบส่วนสรุปของรายงาน ? (y/n)	0-ไม่ลบ, 1-ลบ
3	จำนวนบรรทัดของส่วนสรุปรายงานที่ต้องการลบ	ตัวเลข
1	ลบขีดเส้นใต้ ? (y/n)	0-ไม่ลบ, 1-ลบ
1	เลือกเฉพาะบางคอลัมน์ ? (y/n)	0-ไม่เลือก, 1-เลือก
2	จำนวนช่องว่างที่คั่นระหว่างคอลัมน์ เช่นระหว่างชื่อกับนามสกุลมีช่องว่างคั่น 1 ช่อง ถ้าต้องการให้ชื่อกับ นามสกุลอยู่ในคอลัมน์เดียวกันจะต้องกำหนดตัวเลข 2 ที่ช่องนี้	ตัวเลข
1	คอลัมน์ที่ต้องการ	ตัวเลข
3	คอลัมน์ที่ต้องการ	ตัวเลข
5	คอลัมน์ที่ต้องการ	ตัวเลข
...	คอลัมน์ที่ต้องการ	ตัวเลข

ตารางที่ 3.2 แสดงคำอธิบายข้อมูลในแฟ้มข้อมูลพารามิเตอร์ตัวที่ 1 (SWTOOL1) (ต่อ)

ขั้นตอนที่ 3 เรียกใช้โปรแกรมย่อย "del_lastname" ซึ่งทำหน้าที่ ลบจำนวนช่องว่างที่เกินมาหลังข้อมูลตัวสุดท้ายในเรคอร์ดนั้น ๆ จนถึง ODOA

ขั้นตอนที่ 4 ตรวจสอบข้อมูลตัวที่หนึ่งแฟ้มข้อมูลพารามิเตอร์ตัวที่หนึ่งว่ามีกรอบตารางหรือไม่ ในกรณีที่มีกรอบตาราง โปรแกรมจะเรียกใช้โปรแกรมย่อย "chk_maxsara" ซึ่งทำหน้าที่ นับจำนวนสระ+วรรณยุกต์ที่มากที่สุดที่ในคอลัมน์หนึ่ง ๆ เพื่อใช้ในการบวกจำนวนช่องว่างกลับออกไป ทำให้ข้อมูลในคอลัมน์หลัง ๆ ไม่มากับข้อมูลในคอลัมน์ข้างหน้า โดยใช้ข้อมูลสระและวรรณยุกต์ในแฟ้มข้อมูล SARA.DAT และข้อมูลเส้นกรอบตารางในแฟ้มข้อมูล FRAME.DAT โดยเพิ่มข้อมูล SARA.DAT ใช้เก็บตัวอักษรสระและวรรณยุกต์ ในกรณีนี้ใช้รหัสภาษาไทยของสมอเป็นหลักและใช้ไทยไดเวอร์ที่แอสเอ็ม (Thai Driver TSM) ซึ่งจะมีข้อมูลดังนี้

บรรทัดที่อยู่ระดับเดียวกับพยัญชนะ หลังจากนั้นจึงเรียกใช้โปรแกรมย่อย "m4line" ซึ่งทำหน้าที่อ่านข้อมูลภาษาไทยทีละ 4 บรรทัดมาใส่ในแถวลำดับและทำการแปลงข้อมูลในแถวลำดับที่เก็บแบบ 4 ระดับให้เหลือเพียง 1 ระดับ

ขั้นตอนที่ 8 ตรวจสอบข้อมูลตัวที่สี่ในเพิ่มข้อมูลพารามิเตอร์ตัวที่หนึ่งว่ามีการจัดระดับภาษาไทยแบบ 3 ระดับหรือไม่ ในกรณีเป็นการจัดระดับภาษาไทยแบบ 3 ระดับ โปรแกรมจะเรียกใช้โปรแกรมย่อย "m3line" ซึ่งทำหน้าที่อ่านข้อมูลภาษาไทยทีละ 3 บรรทัดมาใส่ในแถวลำดับ และทำการแปลงข้อมูลในแถวลำดับที่เก็บแบบ 3 ระดับให้เหลือเพียง 1 ระดับ

ขั้นตอนที่ 9 ตรวจสอบข้อมูลตัวที่ห้า,หก,เจ็ด,แปด,เก้าในเพิ่มข้อมูลพารามิเตอร์ตัวที่หนึ่งว่ามีการลบส่วนหัวของรายงาน, ลบส่วนสรุปของรายงาน, ลบขีดเส้นใต้หรือไม่ ถ้ามีโปรแกรมจะเรียกใช้โปรแกรมย่อย "del_heading", "del_summary" และ "del_underline" ตามลำดับ ซึ่งในกรณีที่มีการลบขีดเส้นใต้ โปรแกรมจะทำการอ่านข้อมูลรูปแบบของขีดเส้นใต้จากเพิ่มข้อมูลพารามิเตอร์ตัวที่สอง และทำการลบขีดเส้นใต้ตามรูปแบบที่ผู้ใช้กำหนดไว้ ตัวอย่างเพิ่มข้อมูลพารามิเตอร์ตัวที่สอง อธิบายดังตารางที่ 3.3

ข้อมูล	ความหมาย	ค่าที่เป็นไปได้
+-----	รูปแบบของขีดเส้นใต้	ตัวอักษร 10 หลัก

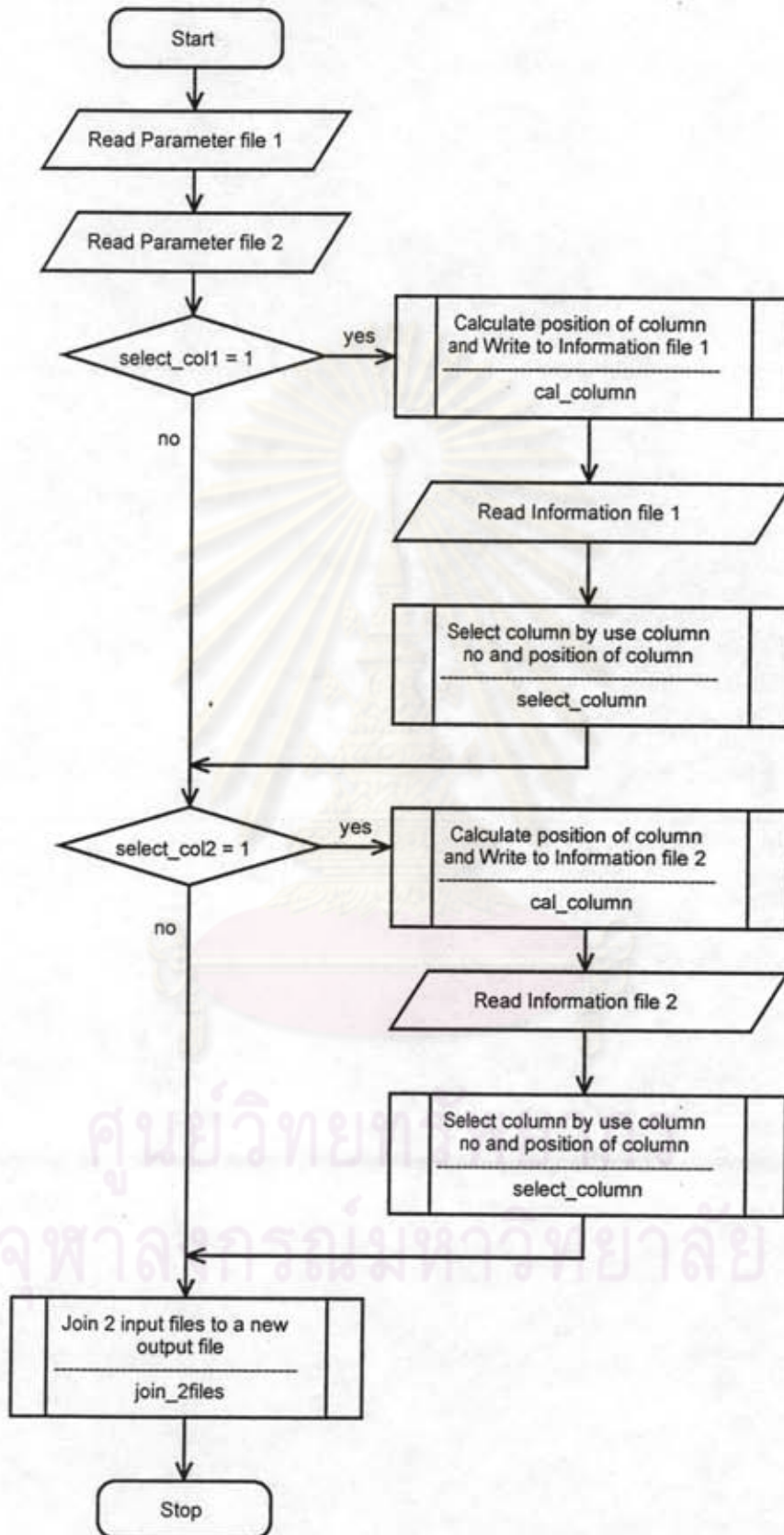
ตารางที่ 3.3 แสดงคำอธิบายข้อมูลในเพิ่มข้อมูลพารามิเตอร์ตัวที่ 2 (SWTOOL1)

ขั้นตอนที่ 10 ตรวจสอบข้อมูลตัวที่สิบในเพิ่มข้อมูลพารามิเตอร์ตัวที่หนึ่ง ว่ามีการเลือกเฉพาะบางคอลัมน์หรือไม่ ในกรณีที่มีการเลือกเฉพาะบางคอลัมน์ โปรแกรมจะเรียกใช้โปรแกรมย่อย "cal_column" ซึ่งทำหน้าที่คำนวณตำแหน่งของคอลัมน์จากเพิ่มข้อมูลที่อ่านเข้ามาในขณะนั้น และแสดงผลลัพธ์ที่ได้จากการคำนวณออกมาในรูปของเพิ่มข้อมูลอินฟอร์เมชัน หลังจากนั้นจึงเรียกใช้โปรแกรมย่อย "select_column" ซึ่งทำหน้าที่เลือกเฉพาะคอลัมน์ที่ต้องการ โดยผู้ใช้จะกำหนดว่าต้องการคอลัมน์ที่เท่าไรมาในเพิ่มข้อมูลพารามิเตอร์ตัวที่หนึ่ง และใช้ข้อมูลตำแหน่งคอลัมน์จากเพิ่มข้อมูลอินฟอร์เมชัน ในการดึงข้อมูลคอลัมน์ต่าง ๆ โดยข้อมูลแต่ละคอลัมน์ที่เลือกออกมาจะกันด้วยเครื่องหมายคอมม่า (,)

3.1.4 เครื่องมือทางซอฟต์แวร์ตัวที่ 2 (SWTOOL2) มีหน้าที่

- ทำการเชื่อมเพิ่มข้อมูล 2 เพิ่มข้อมูลเข้าด้วยกัน
- เลือกข้อมูลจากเพิ่มข้อมูลทั้งสอง เฉพาะบางคอลัมน์ที่ต้องการได้

3.1.5 รูปแบบการทำงานของโปรแกรม SWTOOL2



รูปที่ 3.2 แสดงรูปแบบการทำงานของโปรแกรม SWTOOL2

3.1.6 คำอธิบายขั้นตอนการทำงานของโปรแกรม SWTOOL2

ขั้นตอนที่ 1 โปรแกรมทำการอ่านค่าจากแฟ้มข้อมูลพารามิเตอร์ตัวที่ 1 (Filename .pr1) ซึ่งจะเก็บข้อมูลจากการที่ผู้ใช้ป้อนทางจอภาพ ตัวอย่างเช่น

1 2 1 2

ความหมายของข้อมูลแต่ละตัวอธิบายดังตารางที่ 3.4

ข้อมูล	ความหมาย	ค่าที่เป็นไปได้
1	เลือกเฉพาะบางคอลัมน์ ? (y/n)	0-ไม่เลือก, 1-เลือก
2	จำนวนช่องว่างที่คั่นระหว่างคอลัมน์	ตัวเลข
1	คอลัมน์ที่ต้องการ	ตัวเลข
2	คอลัมน์ที่ต้องการ	ตัวเลข
...	คอลัมน์ที่ต้องการ	ตัวเลข

ตารางที่ 3.4 แสดงคำอธิบายข้อมูลในแฟ้มข้อมูลพารามิเตอร์ตัวที่ 1 (SWTOOL2)

ขั้นตอนที่ 2 โปรแกรมทำการอ่านค่าจากแฟ้มข้อมูลพารามิเตอร์ตัวที่ 2 (Filename .pr2) ซึ่งจะเก็บข้อมูลจากการที่ผู้ใช้ป้อนทางจอภาพ ตัวอย่างเช่น

1 2 3 4

ความหมายของข้อมูลแต่ละตัวอธิบายดังตารางที่ 3.5

ข้อมูล	ความหมาย	ค่าที่เป็นไปได้
1	เลือกเฉพาะบางคอลัมน์ ? (y/n)	0-ไม่เลือก, 1-เลือก
2	จำนวนช่องว่างที่คั่นระหว่างคอลัมน์	ตัวเลข
3	คอลัมน์ที่ต้องการ	ตัวเลข
4	คอลัมน์ที่ต้องการ	ตัวเลข
...	คอลัมน์ที่ต้องการ	ตัวเลข

ตารางที่ 3.5 แสดงคำอธิบายข้อมูลในแฟ้มข้อมูลพารามิเตอร์ตัวที่ 2 (SWTOOL2)

ข้อมูลในส่วนของ คอลัมน์ที่ต้องการ ผู้ใช้สามารถใส่หมายเลขคอลัมน์ไปได้ถึง 50 คอลัมน์โดยจะเรียงลำดับคอลัมน์ไหนก่อน คอลัมน์ไหนหลังก็ได้

ขั้นตอนที่ 3 ตรวจสอบข้อมูลตัวที่หนึ่งในแฟ้มข้อมูลพารามิเตอร์ตัวที่หนึ่งว่าต้องการเลือกเฉพาะบางคอลัมน์หรือไม่ ในกรณีที่มีการเลือกเฉพาะบางคอลัมน์ โปรแกรมจะเรียกใช้โปรแกรมย่อย "cal_column" ซึ่งทำหน้าที่คำนวณตำแหน่งของคอลัมน์จากแฟ้มข้อมูลรับเข้า (Input File) ที่กำหนดให้และแสดงผลลัพธ์ที่ได้จากการคำนวณออกมาในรูปของแฟ้มข้อมูลอินฟอร์เมชัน (Information file) ตัวอย่างข้อมูลในแฟ้มข้อมูลอินฟอร์เมชัน เช่น

12 20

21 32

33 68

69 76

หมายความว่า ข้อมูลในแฟ้มข้อมูลรับเข้า (Input file) ถูกแบ่งออกเป็น 4 คอลัมน์ โดยที่

คอลัมน์ที่ 1 จะเริ่มต้นจากตัวอักษรที่ 12 ถึงตัวอักษรที่ 20

คอลัมน์ที่ 2 จะเริ่มต้นจากตัวอักษรที่ 21 ถึงตัวอักษรที่ 32

คอลัมน์ที่ 3 จะเริ่มต้นจากตัวอักษรที่ 33 ถึงตัวอักษรที่ 68

คอลัมน์ที่ 4 จะเริ่มต้นจากตัวอักษรที่ 69 ถึงตัวอักษรที่ 76

หลังจากนั้นโปรแกรมจะทำการเรียกใช้โปรแกรมย่อย "select_column" ซึ่งทำหน้าที่เลือกเฉพาะคอลัมน์ที่ต้องการ โดยผู้ใช้จะกำหนดว่าต้องการคอลัมน์ที่เท่าไรมาในแฟ้มข้อมูลพารามิเตอร์ตัวที่หนึ่ง และใช้ข้อมูลตำแหน่งคอลัมน์จากแฟ้มข้อมูลอินฟอร์เมชัน ในการดึงข้อมูลคอลัมน์ต่าง ๆ โดยข้อมูลแต่ละคอลัมน์ที่เลือกออกมาจะค้นด้วยเครื่องหมายคอมม่า (,)

ขั้นตอนที่ 4 ตรวจสอบข้อมูลตัวที่หนึ่งในแฟ้มข้อมูลพารามิเตอร์ตัวที่สองว่าต้องการเลือกเฉพาะบางคอลัมน์หรือไม่ ในกรณีที่มีการเลือกเฉพาะบางคอลัมน์ โปรแกรมจะเรียกใช้โปรแกรมย่อย "cal_column" ซึ่งทำหน้าที่คำนวณตำแหน่งของคอลัมน์จากแฟ้มข้อมูลรับเข้า (Input file) ที่กำหนดให้และแสดงผลลัพธ์ที่ได้จากการคำนวณออกมาในรูปของแฟ้มข้อมูลอินฟอร์เมชัน (Information file)

หลังจากนั้นโปรแกรมจะทำการเรียกใช้โปรแกรมย่อย "select_column" ซึ่งทำหน้าที่เลือกเฉพาะคอลัมน์ที่ต้องการ โดยผู้ใช้จะกำหนดว่าต้องการคอลัมน์ที่เท่าไรมาในแฟ้มข้อมูลพารามิเตอร์ตัวที่สอง และใช้ข้อมูลตำแหน่งคอลัมน์จากแฟ้มข้อมูลอินฟอร์เมชัน ในการดึงข้อมูลคอลัมน์ต่าง ๆ โดยข้อมูลแต่ละคอลัมน์ที่เลือกออกมาจะค้นด้วยเครื่องหมายคอมม่า (,)

ขั้นตอนที่ 5 หลังจากผ่านขั้นตอนที่ 3 และขั้นตอนที่ 4 แล้ว จะได้แฟ้มข้อมูล 2 แฟ้มข้อมูลที่ทำการเลือกเฉพาะบางคอลัมน์ออกมาเรียบร้อยแล้ว จากนั้นโปรแกรมจะทำการเรียกใช้โปรแกรมย่อย "join_2_files" ซึ่งทำหน้าที่เชื่อมแฟ้มข้อมูลทั้งสองเข้าด้วยกัน โดยนำข้อมูลในแต่ละแถวมาต่อเข้าด้วยกัน

3.2 การออกแบบและพัฒนาเครื่องมือทางซอฟต์แวร์สำหรับแปลงผันเพิ่มข้อมูลซี, เพิ่มข้อมูลโคบอล, เพิ่มข้อมูลปาสกาล, เพิ่มข้อมูลเบสิก ให้เป็นเพิ่มข้อความ

จากการศึกษา, วิเคราะห์ และรวบรวม เกี่ยวกับประเภทของข้อมูลในเพิ่มข้อมูลซี, เพิ่มข้อมูลโคบอล, เพิ่มข้อมูลปาสกาลและเพิ่มข้อมูลเบสิก ซึ่งมีลักษณะการเก็บข้อมูลในเพิ่มข้อมูลแต่ละประเภทแตกต่างกันไป ตามชนิดของตัวแปรและภาษาที่ใช้ในการสร้างเพิ่มข้อมูลประเภทนั้น ๆ งานวิจัยในส่วนนี้จะทำการแปลงผันข้อมูลในเพิ่มข้อมูล (Data File) ที่ได้รับไว้ข้างต้น ให้เป็นเพิ่มข้อความ โดยมีหน้าที่และรูปแบบการทำงานดังนี้

3.2.1 เครื่องมือทางซอฟต์แวร์ตัวที่ 3 (SWTOOL3) มีหน้าที่

- แปลงข้อมูลในเพิ่มข้อมูลซีที่สร้างจากโปรแกรมเทอร์โบซี โดยมีชนิดของตัวแปรดังนี้ ตัวอักษร (character), เลขจำนวนเต็ม (integer), เลขจำนวนเต็มแบบสั้น (short interger), เลขจำนวนเต็มแบบยาว (long integer), เลขจำนวนเต็มแบบไม่คิดเครื่องหมาย (unsigned integer), เลขจำนวนจริง (floating point), เลขจำนวนจริงที่มีความแม่นยำเป็นสองเท่า (double-precision floating point), เลขจำนวนจริงที่มีความแม่นยำเป็นสองเท่าแบบขยายความยาว (long double-precision floating point) ให้เป็นเพิ่มข้อความ

- แปลงข้อมูลในเพิ่มข้อมูลโคบอลที่สร้างจากโปรแกรมไมโครซอฟต์โคบอล โดยมีชนิดของข้อมูลดังนี้ ตัวอักษร (alphabetic), ตัวอักษรปนตัวเลข (alphanumeric), ตัวเลขจำนวนเต็ม (numeric), ตัวเลขที่มีทศนิยม ให้เป็นเพิ่มข้อความ

- แปลงข้อมูลในเพิ่มข้อมูลปาสกาลที่สร้างจากโปรแกรมเทอร์โบปาสกาล โดยมีชนิดของตัวแปรดังนี้ ไบต์ (byte), เลขจำนวนเต็มแบบสั้น (short interger), เลขจำนวนเต็ม (integer), เลขจำนวนเต็มบวก (word), เลขจำนวนเต็มแบบยาว (long integer), เลขจำนวนจริง (real number), เลขจำนวนจริงที่มีความแม่นยำเป็นหนึ่งเท่า (single), เลขจำนวนจริงที่มีความแม่นยำเป็นสองเท่า (double), เลขจำนวนจริงแบบขยาย (extended), ตัวอักษร (char), สตริง (string) ให้เป็นเพิ่มข้อความ

- แปลงข้อมูลในเพิ่มข้อมูลเบสิกที่สร้างจากโปรแกรมไมโครซอฟต์จัดับลิวเบสิก โดยมีชนิดของตัวแปรดังนี้ ตัวอักษร (string), เลขจำนวนเต็ม (integer), เลขจำนวนจริงที่มีความแม่นยำเป็นหนึ่งเท่า (single), เลขจำนวนจริงที่มีความแม่นยำเป็นสองเท่า (double) ให้เป็นเพิ่มข้อความ

การทำงานของเครื่องมือทางซอฟต์แวร์ตัวที่ 3 แบ่งออกเป็น 4 โปรแกรมคือ

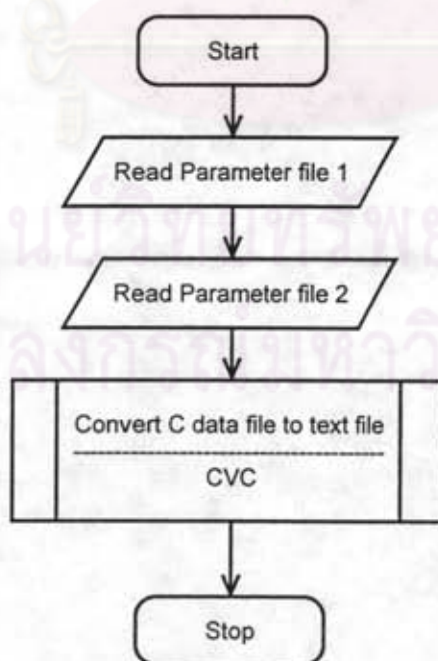
1. โปรแกรม CVC.C พัฒนาโดยใช้ภาษาซี เนื่องจากเพิ่มข้อมูลรับเข้า (Input file) ของเครื่องมือทางซอฟต์แวร์ตัวที่ 3 เป็นเพิ่มข้อมูลที่เกิดจากภาษาซีและการแปลงผันชนิดตัวแปร (Data type) จากเพิ่มข้อมูลซีโดยใช้ภาษาซีจะทำได้ง่ายและสะดวกกว่าพัฒนาโดยภาษาอื่น โปรแกรมนี้ทำหน้าที่แปลงผันเพิ่มข้อมูลซีให้เป็นเพิ่มข้อความ

2. โปรแกรม CVCOBOL.BAS พัฒนาโดยใช้ภาษาเบสิก เนื่องจากเพิ่มข้อมูลรับเข้า (Input file) ของเครื่องมือทางซอฟต์แวร์ตัวที่ 3 เป็นเพิ่มข้อมูลที่เกิดจากภาษาโคบอล ซึ่งโดยส่วนใหญ่นิยมเก็บในรูปของเพิ่มข้อความ ทำให้ไม่เกิดการแปลงผันชนิดตัวแปร (Data type) ดังนั้นจึงเลือกใช้ภาษาเบสิกจัดการในเรื่องของตัวเลขที่มีจุดทศนิยม

3. โปรแกรม CVPASCAL.PAS พัฒนาโดยใช้ภาษาปาสกาล เนื่องจากเพิ่มข้อมูลรับเข้า (Input file) ของเครื่องมือทางซอฟต์แวร์ตัวที่ 3 เป็นเพิ่มข้อมูลที่เกิดจากภาษาปาสกาลและการแปลงผันชนิดตัวแปร (Data type) จากเพิ่มข้อมูลปาสกาลโดยใช้ภาษาปาสกาลจะทำได้ง่ายและสะดวกกว่าพัฒนาโดยภาษาอื่น โปรแกรมนี้ทำหน้าที่แปลงผันเพิ่มข้อมูลปาสกาลให้เป็นเพิ่มข้อความ

4. โปรแกรม CVBASIC.BAS พัฒนาโดยใช้ภาษาเบสิก เนื่องจากเพิ่มข้อมูลรับเข้า (Input file) ของเครื่องมือทางซอฟต์แวร์ตัวที่ 3 เป็นเพิ่มข้อมูลที่เกิดจากภาษาเบสิกและการแปลงผันชนิดตัวแปร (Data type) จากเพิ่มข้อมูลเบสิกโดยใช้ภาษาเบสิกจะทำได้ง่ายและสะดวกกว่าพัฒนาโดยภาษาอื่น โปรแกรมนี้ทำหน้าที่แปลงผันเพิ่มข้อมูลเบสิกให้เป็นเพิ่มข้อความ

3.2.2 รูปแบบการทำงานของโปรแกรม CVC



รูปที่ 3.3 แสดงรูปแบบการทำงานของโปรแกรม CVC

3.2.3 คำอธิบายขั้นตอนการทำงานของโปรแกรม CVC

ขั้นตอนที่ 1 โปรแกรมทำการอ่านค่าจากแฟ้มข้อมูลพารามิเตอร์ตัวที่ 1 (Filename .pr1) ซึ่งจะเก็บข้อมูลจากการที่ผู้ใช้ป้อนทางจอภาพ ตัวอย่างเช่น

C:\DATA\CDATA.DAT , C:\DATA\CDATA.OUT

ความหมายของข้อมูลอธิบายดังตารางที่ 3.6

ข้อมูล	ความหมาย
C:\DATA\CDATA.DAT	ชื่อแฟ้มข้อมูลรับเข้า
C:\DATA\CDATA.OUT	ชื่อแฟ้มข้อมูลผลลัพธ์

ตารางที่ 3.6 แสดงคำอธิบายข้อมูลในแฟ้มข้อมูลพารามิเตอร์ตัวที่ 1 (CVC)

ขั้นตอนที่ 2 โปรแกรมทำการอ่านค่าจากแฟ้มข้อมูลพารามิเตอร์ตัวที่ 2 (Filename .pr2) ซึ่งจะเก็บข้อมูลจากการที่ผู้ใช้ป้อนทางจอภาพ ตัวอย่างเช่น

- 2 , Integer
- 30 , String
- 1 , Char
- 8 , Double , 10.3
- 4 , Float , 5.2
- 10 , Long Double , 15.2
- 2 , ShortInt
- 4 , LongInt

ความหมายของข้อมูลอธิบายดังตารางที่ 3.7

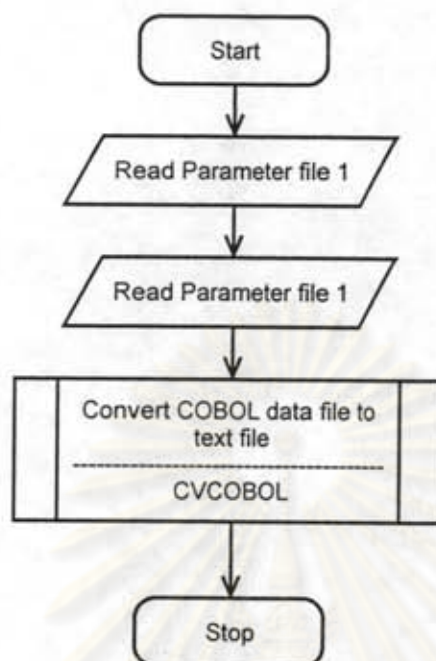
ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ข้อมูล	ความหมาย
2	ความยาวของฟิลด์ตัวที่ 1
Integer	ประเภทข้อมูลของฟิลด์ตัวที่ 1 (Empno)
30	ความยาวของฟิลด์ตัวที่ 2
String	ประเภทข้อมูลของฟิลด์ตัวที่ 2 (Ename)
1	ความยาวของฟิลด์ตัวที่ 3
Char	ประเภทข้อมูลของฟิลด์ตัวที่ 3 (Sex)
8	ความยาวของฟิลด์ตัวที่ 4
Double	ประเภทข้อมูลของฟิลด์ตัวที่ 4 (Salary)
10.3	รูปแบบผลลัพธ์ของฟิลด์ตัวที่ 4
4	ความยาวของฟิลด์ตัวที่ 5
Float	ประเภทข้อมูลของฟิลด์ตัวที่ 5 (OT)
5.2	รูปแบบผลลัพธ์ของฟิลด์ตัวที่ 5
10	ความยาวของฟิลด์ตัวที่ 6
LongDouble	ประเภทข้อมูลของฟิลด์ตัวที่ 6 (SumSalary)
15.2	รูปแบบผลลัพธ์ของฟิลด์ตัวที่ 6
2	ความยาวของฟิลด์ตัวที่ 7
ShortInt	ประเภทข้อมูลของฟิลด์ตัวที่ 7 (Deptno)
4	ความยาวของฟิลด์ตัวที่ 8
LongInt	ประเภทข้อมูลของฟิลด์ตัวที่ 8 (Idno)

ตารางที่ 3.7 แสดงคำอธิบายข้อมูลในแฟ้มข้อมูลพารามิเตอร์ตัวที่ 2 (CVC)

ขั้นตอนที่ 3 ทำการเปิดแฟ้มข้อมูลรับเข้าตามชื่อที่ระบุในแฟ้มข้อมูลพารามิเตอร์ตัวที่ 1 หลังจากนั้นนำโครงสร้างแฟ้มข้อมูล (File Structure) ที่ได้จากแฟ้มข้อมูลพารามิเตอร์ตัวที่ 2 มาใช้ในการแปลงข้อมูลที่เก็บในแฟ้มข้อมูลรับเข้าตามประเภทข้อมูลที่กำหนด โดยอ่านเท่ากับจำนวนไบต์ที่กำหนด แล้วเก็บลงในตัวแปรตามประเภทของข้อมูล หลังจากนั้นจึงเก็บบันทึกลงในแฟ้มข้อมูลผลลัพธ์ในรูปแบบของแฟ้มข้อความ

3.2.4 รูปแบบการทำงานของโปรแกรม CVCOBOL



รูปที่ 3.4 แสดงรูปแบบการทำงานของโปรแกรม CVCOBOL

3.2.5 คำอธิบายขั้นตอนการทำงานของโปรแกรม CVCOBOL

ขั้นตอนที่ 1 โปรแกรมทำการอ่านค่าจากแฟ้มข้อมูลพารามิเตอร์ตัวที่ 1 (Filename .pr1) ซึ่งจะเก็บข้อมูลจากการที่ผู้ใช้ป้อนทางจอภาพ ตัวอย่างเช่น

C:\DATA\COBOL.DAT, C:\DATA\COBOL.OUT

ความหมายของข้อมูลอธิบายดังตารางที่ 3.8

ข้อมูล	ความหมาย
C:\DATA\COBOL.DAT	ชื่อแฟ้มข้อมูลรับเข้า
C:\DATA\COBOL.OUT	ชื่อแฟ้มข้อมูลผลลัพธ์

ตารางที่ 3.8 แสดงคำอธิบายข้อมูลในแฟ้มข้อมูลพารามิเตอร์ตัวที่ 1 (CVCOBOL)

ขั้นตอนที่ 2 โปรแกรมทำการอ่านค่าจากแฟ้มข้อมูลพารามิเตอร์ตัวที่ 2 (Filename .pr2) ซึ่งจะเก็บข้อมูลจากการที่ผู้ใช้ป้อนทางจอภาพ ตัวอย่างเช่น

2 , Numeric

30 , Alphanumeric

10.2 , Numeric decimal

6.2 , Numeric decimal

ความหมายของข้อมูลอธิบายดังตารางที่ 3.9

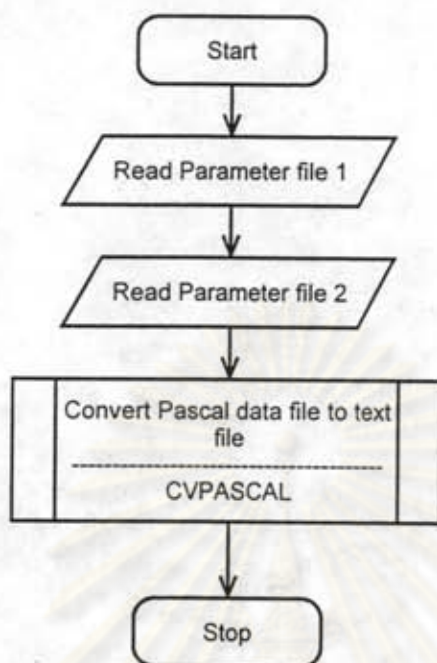
ข้อมูล	ความหมาย
2	ความยาวของฟิลด์ตัวที่ 1
Numeric	ประเภทข้อมูลของฟิลด์ตัวที่ 1 (Empno)
30	ความยาวของฟิลด์ตัวที่ 2
Alphanumeric	ประเภทข้อมูลของฟิลด์ตัวที่ 2 (Ename)
10.2	ความยาวของฟิลด์ตัวที่ 3
Numeric decimal	ประเภทข้อมูลของฟิลด์ตัวที่ 3 (Salary)
6.2	ความยาวของฟิลด์ตัวที่ 4
Numeric decimal	ประเภทข้อมูลของฟิลด์ตัวที่ 4 (OT)

ตารางที่ 3.9 แสดงคำอธิบายข้อมูลในแฟ้มข้อมูลพารามิเตอร์ตัวที่ 2 (CVCOBOL)

ขั้นตอนที่ 3 ทำการเปิดแฟ้มข้อมูลรับเข้าตามชื่อที่ระบุในแฟ้มข้อมูลพารามิเตอร์ตัวที่ 1 หลังจากนั้นนำโครงสร้างแฟ้มข้อมูล (File Structure) ที่ได้จากแฟ้มข้อมูลพารามิเตอร์ตัวที่ 2 มาใช้ในการแปลงข้อมูลที่เก็บในแฟ้มข้อมูลรับเข้าตามประเภทข้อมูลที่กำหนด โดยอ่านเท่ากับจำนวนไบต์ที่กำหนด แล้วเก็บลงในตัวแปรตามประเภทของข้อมูล หลังจากนั้นจึงเก็บบันทึกลงในแฟ้มข้อมูลผลลัพธ์ในรูปแบบของแฟ้มข้อมูล

จุฬาลงกรณ์มหาวิทยาลัย

3.2.6 รูปแบบการทำงานของโปรแกรม CVPASCAL



รูปที่ 3.5 แสดงรูปแบบการทำงานของโปรแกรม CVPASCAL

3.2.7 คำอธิบายขั้นตอนการทำงานของโปรแกรม CVPASCAL

ขั้นตอนที่ 1 โปรแกรมทำการอ่านค่าจากแฟ้มข้อมูลพารามิเตอร์ตัวที่ 1 (Filename .pr1) ซึ่งจะเก็บข้อมูลจากการที่ผู้ใช้ป้อนทางจอภาพ ตัวอย่างเช่น

C:\DATA\PASCAL.DAT, C:\DATA\PASCAL.OUT

ความหมายของข้อมูลอธิบายดังตารางที่ 3.10

ข้อมูล	ความหมาย
C:\DATA\PASCAL.DAT	ชื่อแฟ้มข้อมูลรับเข้า
C:\DATA\PASCAL.OUT	ชื่อแฟ้มข้อมูลผลลัพธ์

ตารางที่ 3.10 แสดงคำอธิบายข้อมูลในแฟ้มข้อมูลพารามิเตอร์ตัวที่ 1 (CVPASCAL)

ขั้นตอนที่ 2 โปรแกรมทำการอ่านค่าจากแฟ้มข้อมูลพารามิเตอร์ตัวที่ 2 (Filename .pr2) ซึ่งจะเก็บข้อมูลจากการที่ผู้ใช้ป้อนทางจอภาพ ตัวอย่างเช่น

2 , Integer
 30 , String
 1 , Char
 8 , Double
 4 , Single
 6 , Real
 10 , Extended
 1 , Shortint
 4 , Longint

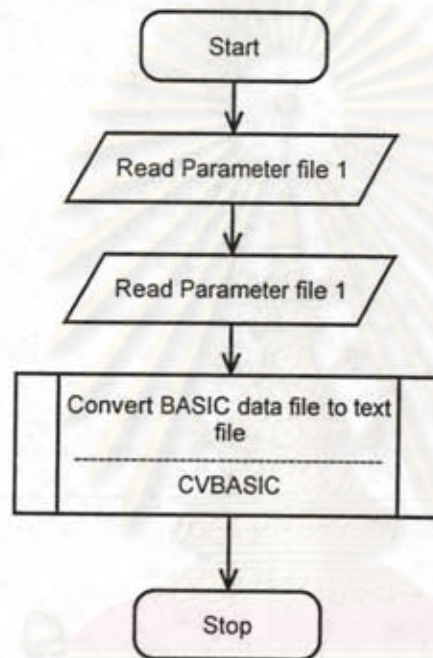
ความหมายของข้อมูลอธิบายดังตารางที่ 3.11

ข้อมูล	ความหมาย
2	ความยาวของฟิลด์ตัวที่ 1
Integer	ประเภทข้อมูลของฟิลด์ตัวที่ 1 (Empno)
30	ความยาวของฟิลด์ตัวที่ 2
String	ประเภทข้อมูลของฟิลด์ตัวที่ 2 (Ename)
1	ความยาวของฟิลด์ตัวที่ 3
Char	ประเภทข้อมูลของฟิลด์ตัวที่ 3 (Sex)
8	ความยาวของฟิลด์ตัวที่ 4
Double	ประเภทข้อมูลของฟิลด์ตัวที่ 4 (Salary)
4	ความยาวของฟิลด์ตัวที่ 5
Single	ประเภทข้อมูลของฟิลด์ตัวที่ 5 (OT)
6	ความยาวของฟิลด์ตัวที่ 6
Real	ประเภทข้อมูลของฟิลด์ตัวที่ 6 (Comm)
10	ความยาวของฟิลด์ตัวที่ 7
Extended	ประเภทข้อมูลของฟิลด์ตัวที่ 7 (SumSalary)
1	ความยาวของฟิลด์ตัวที่ 8
Shortint	ประเภทข้อมูลของฟิลด์ตัวที่ 8 (Deptno)
4	ความยาวของฟิลด์ตัวที่ 9
Longint	ประเภทข้อมูลของฟิลด์ตัวที่ 9 (Idno)

ตารางที่ 3.11 แสดงคำอธิบายข้อมูลในแฟ้มข้อมูลพหามิติเตอร์ตัวที่ 2 (CVPASCAL)

ขั้นตอนที่ 3 ทำการเปิดแฟ้มข้อมูลรับเข้าตามชื่อที่ระบุในแฟ้มข้อมูลพารามิเตอร์ตัวที่ 1 หลังจากนั้นนำโครงสร้างแฟ้มข้อมูล (File Structure) ที่ได้จากแฟ้มข้อมูลพารามิเตอร์ตัวที่ 2 มาใช้ในการแปลงข้อมูลที่เก็บในแฟ้มข้อมูลรับเข้าตามประเภทข้อมูลที่กำหนด โดยอ่านเท่ากับจำนวนไบต์ที่กำหนด แล้วเก็บลงในตัวแปรตามประเภทของข้อมูล หลังจากนั้นจึงเก็บบันทึกลงในแฟ้มข้อมูลผลลัพธ์ในรูปแบบของแฟ้มข้อความ

3.2.8 รูปแบบการทำงานของโปรแกรม CVBASIC



รูปที่ 3.6 แสดงรูปแบบการทำงานของโปรแกรม CVBASIC

3.2.9 คำอธิบายขั้นตอนการทำงานของโปรแกรม CVBASIC

ขั้นตอนที่ 1 โปรแกรมทำการอ่านค่าจากแฟ้มข้อมูลพารามิเตอร์ตัวที่ 1 (Filename .pr1) ซึ่งจะเก็บข้อมูลจากการที่ผู้ใช้ป้อนทางจอภาพ ตัวอย่างเช่น

C:\DATA\BASIC.DAT, C:\DATA\BASIC.OUT

ความหมายของข้อมูลอธิบายดังตารางที่ 3.12

ข้อมูล	ความหมาย
C:\DATA\BASIC.DAT	ชื่อแฟ้มข้อมูลรับเข้า
C:\DATA\BASIC.OUT	ชื่อแฟ้มข้อมูลผลลัพธ์

ตารางที่ 3.12 แสดงคำอธิบายข้อมูลในแฟ้มข้อมูลพารามิเตอร์ตัวที่ 1 (CVBASIC)

ขั้นตอนที่ 2 โปรแกรมทำการอ่านค่าจากแฟ้มข้อมูลพารามิเตอร์ตัวที่ 2 (Filename.pr2) ซึ่งจะเก็บข้อมูลจากการที่ผู้ใช้ป้อนทางจอภาพ ตัวอย่างเช่น

2 , Integer

30 , String

8 , Double

4 , Single

ความหมายของข้อมูลอธิบายดังตารางที่ 3.13

ข้อมูล	ความหมาย
2	ความยาวของฟิลด์ตัวที่ 1
Integer	ประเภทข้อมูลของฟิลด์ตัวที่ 1 (Empno)
30	ความยาวของฟิลด์ตัวที่ 2
String	ประเภทข้อมูลของฟิลด์ตัวที่ 2 (Ename)
8	ความยาวของฟิลด์ตัวที่ 3
Double	ประเภทข้อมูลของฟิลด์ตัวที่ 3 (Salary)
4	ความยาวของฟิลด์ตัวที่ 4
Single	ประเภทข้อมูลของฟิลด์ตัวที่ 4 (OT)

ตารางที่ 3.13 แสดงคำอธิบายข้อมูลในแฟ้มข้อมูลพารามิเตอร์ตัวที่ 2 (CVBASIC)

ขั้นตอนที่ 3 ทำการเปิดแฟ้มข้อมูลรับเข้าตามชื่อที่ระบุในแฟ้มข้อมูลพารามิเตอร์ตัวที่ 1 หลังจากนั้นนำโครงสร้างแฟ้มข้อมูล (File Structure) ที่ได้จากแฟ้มข้อมูลพารามิเตอร์ตัวที่ 2 มาใช้ในการแปลงข้อมูลที่เก็บในแฟ้มข้อมูลรับเข้าตามประเภทข้อมูลที่กำหนด โดยอ่านเท่ากับจำนวนไบต์ที่กำหนด แล้วเก็บลงในตัวแปรตามประเภทของข้อมูล หลังจากนั้นจึงเก็บบันทึกลงในแฟ้มข้อมูลผลลัพธ์ในรูปแบบของแฟ้มข้อความ

3.3 การประยุกต์ใช้ซอฟต์แวร์สำหรับแปลงผันเพิ่มข้อมูลบีทรีฟและฐานข้อมูลออราเคิลให้เป็นเพิ่มข้อความ

จากการศึกษา, วิเคราะห์ โครงสร้างของเพิ่มข้อมูลบีทรีฟและข้อมูลในฐานข้อมูลออราเคิล สามารถนำซอฟต์แวร์สำเร็จรูปมาประยุกต์ใช้ในการแปลงผันให้เป็นเพิ่มข้อความได้ ซึ่งซอฟต์แวร์สำเร็จรูปที่มีความสามารถในการทำงานลักษณะนี้ ในปัจจุบันมีด้วยกันหลายโปรแกรม โดยจะทำการเปรียบเทียบความสามารถในการเข้าถึง (Access) เพิ่มข้อมูลบีทรีฟและฐานข้อมูลออราเคิล ของโปรแกรมดังต่อไปนี้

1. ฟอรัลเนสแอนด์ทรีเวอร์ชัน 3.0 (Forest & Tree Version 3.0)
2. ไมโครซอฟต์แอ็กเซสเวอร์ชัน 2.0 (Microsoft Access Version 2.0)
3. วิซวลเบสิกเวอร์ชัน 3.0 (Visual Basic Version 3.0)
4. เดลไฟเวอร์ชัน 1.0 (Delphi Version 1.0)
5. โลตัสโน้ตเวอร์ชัน 3.0 (Lotus Notes Version 3.0)

3.3.1 การเปรียบเทียบคุณสมบัติของโปรแกรม ซึ่งจะทำการเปรียบเทียบโปรแกรมฟอรัลเนสแอนด์ทรี, ไมโครซอฟต์แอ็กเซส, วิซวลเบสิก, เดลไฟ และโลตัสโน้ต ในการแปลงผันเพิ่มข้อมูลบีทรีฟและฐานข้อมูลออราเคิลให้เป็นเพิ่มข้อความ โดยเปรียบเทียบในลักษณะตาราง ดังต่อไปนี้

โปรแกรม	คุณสมบัติทั่วไป
1. ฟอรัลเนสแอนด์ทรี	เป็นโปรแกรมที่ช่วยในการทำงานทางด้านวิธีการเข้าถึงข้อมูลและเป็นเครื่องมือในการจัดทำรายงาน (Data Access and Reporting Tool) มีความสามารถในการเก็บและรวบรวมข้อมูลจากแหล่งต่าง ๆ เข้าด้วยกัน เช่น แผ่นตารางทำการ (spreadsheets), ฐานข้อมูล (database) และฐานข้อมูลที่ติดตั้งที่เครื่องบริการ (Database Server) สามารถนำไปใช้ในการสร้างระบบสารสนเทศเพื่อการจัดการ (Management Information System)
2. ไมโครซอฟต์แอ็กเซส	เป็นโปรแกรมจัดการฐานข้อมูลเชิงสัมพันธ์ (Relational Database Management System) โดยมีส่วนที่เรียกว่าดาต้าเบสเอนจิน (Database Engine) เป็นตัวที่จะเข้าถึงฐานข้อมูล มีตัวเชื่อมประสานกับผู้ใช้ (User Interface) และโปรแกรมบิวเดอร์ (Program Builder) ที่จะช่วยให้ผู้ใช้สามารถใช้เป็นทางผ่านที่จะเข้าถึงฐานข้อมูลและดึงข้อมูลออกมาใช้งาน นอกจากนี้ยังมียูทิลิตี้ (Utility) ช่วยในการสร้างรายงานในรูปแบบต่าง ๆ ได้โดยไม่ต้องเขียนโปรแกรม

ตารางที่ 3.14 แสดงการเปรียบเทียบคุณสมบัติทั่วไปของโปรแกรม

โปรแกรม	คุณสมบัติทั่วไป
3. วิชาการเบสิก	เป็นเครื่องมือที่ช่วยในการพัฒนาระบบงานบนวินโดว์ (Windows Programmer Development Tool) และมียูทิลิตี้ (Utility) ในการสร้างเอ็กคิวทีเบิลไฟล์ (Executable File .EXE) ซึ่งเวลาทำงานจะต้องมีไฟล์ ดีเอลเอล (.DLL) ชื่อ VBRUN300.DLL จึงจะสามารถใช้งานได้ เหมาะสำหรับพัฒนาระบบงานที่ต้องการกราฟฟิกตัวเชื่อมประสานกับผู้ใช้ (Graphic User Interface), เหมาะสำหรับนำมาใช้เป็นเครื่องมือพัฒนาระบบงาน (Front End Tool) ที่ใช้กับฐานข้อมูลภายนอก (Access External Database) และระบบงานสารสนเทศ (Information System)
4. เดลไฟ	เป็นเครื่องมือที่ช่วยในการพัฒนาระบบงานบนวินโดว์ (Windows Programmer Development Tool) และมีเดลไฟคอมไพเลอร์ (Delphi Compiler) แปลคำสั่งเป็นภาษาเครื่อง (.EXE) ทำให้โปรแกรมทำงานได้เร็วกว่าโปรแกรมที่ต้องการตัวแปลคำสั่ง (Interpreter) และเวลาทำงานก็ไม่ต้องมีไฟล์ดีเอลเอล (.DLL) ผู้เขียนโปรแกรมสามารถสร้างคอมโพเนนต์ (Component) ขึ้นใหม่ได้ และสามารถใช้ไฟล์ดีเอลเอล (.DLL) จากภาษาอื่น ๆ นอกจากนั้นสามารถเข้าถึง (Access) ฐานข้อมูลต่าง ๆ ได้โดยผ่านทางบีดีอี (BDE) ย่อมาจาก Borland Database Engine ซึ่งสามารถเข้าถึงข้อมูลในฐานข้อมูลทั้งที่เป็นแบบโลคอล (Local) และแบบเซิร์ฟเวอร์ (Server)
5. โลตัสโน้ต	เป็นกรุปแวร์ (Groupware) หมายถึงซอฟต์แวร์ที่จะช่วยให้ทีมงานสามารถเชื่อมโยงความรู้ของทีมงาน กระบวนการทำงานและแอปพลิเคชันต่าง ๆ เพื่อเพิ่มประสิทธิภาพให้กับหน่วยงานนั้น ๆ โดยอาศัยการใช้ระบบข้อความ, ระบบจัดเก็บเอกสารอิเล็กทรอนิกส์ และระบบที่สามารถพัฒนาแอปพลิเคชันของตัวเองได้ นอกจากนั้นยังสามารถใช้กับงานเฉพาะด้าน เช่น งานด้านจัดเอกสารเป็นภาพ (Document Imaging), ระบบจัดการกระบวนการงาน (Workflow Management) และระบบติดต่อข้อมูลภายนอก

ตารางที่ 3.14 แสดงการเปรียบเทียบคุณสมบัติทั่วไปของโปรแกรม (ต่อ)

โปรแกรม	การเข้าถึง (Access) เพิ่มข้อมูลบิทรีฟ
1. ฟอ์เรสแอนด์ทรี	สามารถเข้าถึงเพิ่มข้อมูลบิทรีฟได้ โดยใช้โปรแกรมที่ชื่อว่าฟลิก (FLIC) ย่อมาจาก Flat File Interface Construction kit เป็นโปรแกรมที่ช่วยในการกำหนดโครงสร้างของเพิ่มข้อมูล (File Structure) และสร้างคำนิยามเพิ่มข้อมูล (Definition File) ให้ ซึ่งเป็นโปรแกรมที่มีมาพร้อมกับโปรแกรมฟอ์เรสแอนด์ทรี หลังจากนั้นจึงใช้โปรแกรมฟอ์เรสแอนด์ทรีดึงข้อมูล (Retrieve) ในเพิ่มข้อมูลบิทรีฟได้ตามความต้องการโดยใช้ภาษาเอสคิวเอล (SQL) และมียูทิลิตี้ (Utility) ในการถ่ายข้อมูลออก (Export) ให้เป็นเพิ่มข้อความ (Text File)
2. ไมโครซอฟต์แอ็คเซส	สามารถเข้าถึงเพิ่มข้อมูลบิทรีฟได้ โดยใช้โปรแกรมเอ็ทรีฟ (Xtrieve) ในการกำหนดโครงสร้างของเพิ่มข้อมูล (File Structure) และสร้างคำนิยามเพิ่มข้อมูล (Definition File) โดยจะเก็บในเพิ่มข้อมูลชื่อ File.DDF ซึ่งโปรแกรมเอ็ทรีฟเป็นโปรแกรมที่จะต้องจัดซื้อเพิ่มเติม หลังจากนั้นจึงใช้โปรแกรมไมโครซอฟต์แอ็คเซสทำการติดต่อกับเพิ่มข้อมูล File.DDF จึงจะสามารถดึงข้อมูล (Retrieve) ในเพิ่มข้อมูลบิทรีฟได้ โดยใช้ภาษาเอสคิวเอล (SQL) และมียูทิลิตี้ (Utility) ในการถ่ายข้อมูลออก (Export) ให้เป็นเพิ่มข้อความ (Text File)
3. วิซวลเบสิก	เหมือนกับไมโครซอฟต์แอ็คเซส แต่ไม่มียูทิลิตี้ (Utility) ในการถ่ายข้อมูลออก (Export) ให้เป็นเพิ่มข้อความ (Text File) ซึ่งจะต้องใช้วิธีการเขียนโปรแกรมในการถ่ายข้อมูลออก ทำให้ผู้ใช้ไม่สะดวกในการทำงาน
4. เดลไฟ	เหมือนกับไมโครซอฟต์แอ็คเซส แต่ไม่มียูทิลิตี้ (Utility) ในการถ่ายข้อมูลออก (Export) ให้เป็นเพิ่มข้อความ (Text File) ซึ่งจะต้องใช้วิธีการเขียนโปรแกรมในการถ่ายข้อมูลออก ทำให้ผู้ใช้ไม่สะดวกในการทำงาน
5. โลตัสโน้ต	เวอร์ชันนี้ไม่สามารถเข้าถึงเพิ่มข้อมูลบิทรีฟได้

ตารางที่ 3.15 แสดงการเปรียบเทียบการเข้าถึง (Access) เพิ่มข้อมูลบิทรีฟ

โปรแกรม	การเข้าถึง (Access) ฐานข้อมูลออรากเคิล
1. พอร์เรสแอนด์ทรี	สามารถเข้าถึงฐานข้อมูลออรากเคิลได้ โดยผ่านทางโปรแกรมเอสคิวเอลเน็ต (SQL*Net driver) ซึ่งมีหน้าที่ทำให้ไคลเอ็นต์ (Client) สามารถติดต่อกับออรากเคิลเซิร์ฟเวอร์ (Oracle Server) โดยผ่านทางเครือข่ายได้ โปรแกรมเอสคิวเอลเน็ต (SQL*Net) เป็นโปรแกรมที่มีมากับชุดโปรแกรมออรากเคิล ซึ่งหลังจากเรียกโปรแกรมเอสคิวเอลเน็ตแล้ว โปรแกรมพอร์เรสแอนด์ทรีจะสามารถเข้าถึงฐานข้อมูลออรากเคิลได้โดยระบุชื่อฐานข้อมูลให้ถูกต้อง หลังจากนั้นสามารถเรียกดึงข้อมูล (Retrieve) ในฐานข้อมูลออรากเคิลได้ตามความต้องการ โดยใช้ภาษาเอสคิวเอล (SQL) และมียูทิลิตี้ (Utility) ในการถ่ายข้อมูลออก (Export) ให้เป็นแฟ้มข้อความ (Text File)
2. ไมโครซอฟต์แอ็กเซส	สามารถเข้าถึงฐานข้อมูลออรากเคิลได้ โดยผ่านทางโอดีบีซีไดเวอร์ (ODBC driver) ย่อมาจาก Open Database Connectivity หลังจากติดตั้งโอดีบีซีไดเวอร์และกำหนดโครงแบบ (Configuration) ในกาที่จะติดต่อกับฐานข้อมูลออรากเคิลเรียบร้อยแล้ว โปรแกรมไมโครซอฟต์แอ็กเซสจึงสามารถติดต่อกับฐานข้อมูลออรากเคิลและสามารถดึงข้อมูล (Retrieve) ในตาราง (Table) ได้ โดยใช้ภาษาเอสคิวเอล (SQL) และมียูทิลิตี้ (Utility) ในการถ่ายข้อมูลออก (Export) ให้เป็นแฟ้มข้อความ (Text File)
3. วิซวลเบสิก	เหมือนกับไมโครซอฟต์แอ็กเซส แต่ไม่มียูทิลิตี้ (Utility) ในการถ่ายข้อมูลออก (Export) ให้เป็นแฟ้มข้อความ (Text File) ซึ่งจะต้องใช้วิธีการเขียนโปรแกรมในการถ่ายข้อมูลออก ทำให้ผู้ใช้ไม่สะดวกในการทำงาน
4. เดลไฟ	เหมือนกับไมโครซอฟต์แอ็กเซส แต่ไม่มียูทิลิตี้ (Utility) ในการถ่ายข้อมูลออก (Export) ให้เป็นแฟ้มข้อความ (Text File) ซึ่งจะต้องใช้วิธีการเขียนโปรแกรมในการถ่ายข้อมูลออก ทำให้ผู้ใช้ไม่สะดวกในการทำงาน
5. โลตัสโน้ต	เหมือนกับไมโครซอฟต์แอ็กเซส และมียูทิลิตี้ (Utility) ในการถ่ายข้อมูลออก (Export) ให้เป็นแฟ้มข้อความ (Text File)

ตารางที่ 3.16 แสดงการเปรียบเทียบการเข้าถึง (Access) ฐานข้อมูลออรากเคิล

จากการเปรียบเทียบคุณสมบัติของโปรแกรมข้างต้น จะเห็นได้ว่าโปรแกรมพอร์เรสแอนด์ทรีเป็นโปรแกรมที่เหมาะสมในการนำมาใช้ในการแปลงผันแฟ้มข้อมูลบิทรีฟและข้อมูลในฐานข้อมูลออรากเคิลให้เป็นแฟ้มข้อความ เนื่องจากเป็นโปรแกรมที่สามารถทำงานได้ทันทีโดยไม่ต้องจัดซื้อโปรแกรมอื่นเพิ่มเติมและมีการใช้งานที่สะดวกและง่ายกว่าโปรแกรมอื่น ๆ เนื่องจากมียูทิลิตี้ (Utility) ช่วยในการถ่ายข้อมูลออก (Export) ให้เป็นแฟ้มข้อความ (Text File)

ในงานวิจัยชิ้นนี้ขอเสนอโปรแกรมฟอร์เรสแอนด์ทรี (Forest & Trees) มาใช้ในการทำงานในส่วนนี้ โดยจะอธิบายถึงคุณสมบัติของโปรแกรมฟอร์เรสแอนด์ทรี, ขั้นตอนการทำงานในการใช้โปรแกรม พร้อมยกตัวอย่างประกอบ

3.3.2 คุณสมบัติของโปรแกรมฟอร์เรสแอนด์ทรี

โปรแกรมฟอร์เรสแอนด์ทรี เป็นโปรแกรมที่ช่วยในการทำงานทางด้านวิธีการเข้าถึงข้อมูลและเป็นเครื่องมือในการจัดทำรายงาน (Data Access and Reporting Tool) มีความสามารถในการเก็บและรวบรวมข้อมูลจากแหล่งต่าง ๆ เข้าด้วยกัน เช่น แผ่นตารางทำการ (spreadsheets), เพิ่มข้อมูลฐานข้อมูล (database files) และฐานข้อมูลที่ติดตั้งที่เครื่องบริการ (Database Server) โดยสร้างเป็นระบบงาน เพื่อให้ผู้บริหารใช้ในการวิเคราะห์ตัดสินใจและติดตามผลได้ทุกระดับตั้งแต่ข้อมูลที่สรุปเรียบร้อยแล้วจนถึงรายละเอียดภายใน โดยมีฟังก์ชันต่าง ๆ ดังนี้

- รวบรวมข้อมูล (Data collection) ใช้ภาษาที่ได้รับความนิยมมากที่สุดในการเข้าถึงข้อมูลคือ ภาษาสอบถามเชิงโครงสร้าง (structured query language)
- เชื่อมโยงข้อมูล (Combining data) ผู้ใช้สามารถสร้างการสอบถาม (queries) และสูตร (formulas) เพื่อเชื่อมโยงข้อมูลจากฐานข้อมูลชนิดเดียวกันหรือต่างชนิดกัน สามารถจัดการกับข้อมูลได้หลายประเภทเช่น ข้อมูลตัวเลข, ข้อมูลตัวอักษร, ข้อมูลวันที่, ข้อมูลเวลา และฟังก์ชันที่ใช้ในการแปลงค่า (conversion functions)
- การนำเสนอ (Presentation) ผู้ใช้สามารถสร้างการนำเสนอแบบง่าย ๆ หรือแบบยุ่งยากซับซ้อนก็ได้ ในกรณีที่ต้องการสร้างระบบงานแบบผู้ใช้หลายคน (multi-user application) ผู้ใช้สามารถปรับแต่งระบบโดยการสร้างเมนู, ปุ่ม, เซตข้อมูล และกราฟฟิก ได้ด้วยตนเอง
- กำหนดการทำงาน (Action) ผู้ใช้สามารถทำทุกสิ่งทุกอย่างให้เป็นไปโดยอัตโนมัติได้โดยผ่านทางชุดคำสั่งในการควบคุมฟังก์ชัน (set of control functions) หรือใช้วิธีกดปุ่ม (button-clicking) ตัวอย่างเช่น มีการกำหนดหมายกำหนดการไว้ล่วงหน้า ให้โปรแกรมทำการคำนวณตัวชี้ทางการเงินที่สำคัญ (important financial indicator) และตรวจดูว่าตกอยู่ในช่วงที่ต้องตัดสินใจล่วงหน้า (predetermined alarm range) หรือไม่ ถ้าตกอยู่ในช่วงนั้น โปรแกรมจะทำการส่งจดหมายเพื่อเป็นการเตือนผู้ใช้และจะทำการสอบถามข้อมูลในส่วนของรายละเอียดขึ้นมาให้, ทำการพิมพ์รายงานให้ ทุกอย่างสามารถทำได้โดยอัตโนมัติ
- การสร้างรายงาน (Report) ฟอร์เรสแอนด์ทรีมีโปรแกรมการสร้างรายงาน (report writer) ที่มีความสามารถมาก เพื่อให้ผู้ใช้สามารถออกแบบรายงานได้โดยง่าย
- ระบบเฝ้าคุม (Monitoring) ผู้ใช้สามารถสร้างหมายกำหนดการล่วงหน้าในการแก้ไขสารสนเทศได้โดยอัตโนมัติ และสามารถติดตั้งการเตือนล่วงหน้าในกรณีที่มีข้อมูลผิดปกติเกิดขึ้น การเตือน

ในลักษณะเช่นนี้สามารถแสดงเฉพาะผลลัพธ์และแสดงข้อความแบบผุดขึ้น (pop-up messages) หรือสั่งให้ทริกเกอร์ (trigger) ทำการส่งไปรษณีย์อิเล็กทรอนิกส์ (electronic mail) ได้

- การวิเคราะห์ (Analysis) ผู้ใช้สามารถวิเคราะห์ข้อมูลได้โดยการสร้างกราฟหรือสร้างตาราง (cross-tabulation) ได้อย่างง่าย ๆ นอกจากนี้ยังมีการเก็บประวัติ เก็บรายการที่เปลี่ยนแปลงเพื่อให้ผู้ใช้สามารถนำไปวิเคราะห์แนวโน้มในภายหลังได้

- การตกแต่ง (Customizing) ผู้ใช้สามารถกำหนดตกแต่งรูปร่างหน้าตาของวินโดว์ รวมทั้งการควบคุมวินโดว์และทางเลือกต่าง ๆ บนเมนู

- การใช้ข้อมูลร่วม (Sharing Data) ผู้ใช้สามารถนำข้อมูลออกไปยังเพิ่มข้อมูลหรือฐานข้อมูลได้โดยตรงหรือผ่านทาง การแลกเปลี่ยนข้อมูลแบบพลวัต (Dynamic Data Exchange) หรือวินโดว์คลิปบอร์ด (Window clipboard) รวมทั้งสามารถเอ็กพอร์ต (export) กราฟและรายงานได้เช่นเดียวกัน

- ความมั่นคง (Security) ผู้ใช้สามารถใช้ระบบความมั่นคงของฐานข้อมูลเดิมที่มีอยู่แล้ว รวมทั้งระบบความมั่นคงของโปรแกรมฟอร์เรสแอนด์ทรีที่มีเพิ่มให้ ซึ่งจะอนุญาตให้ผู้ใช้เข้าถึงวิวไฟล์ หรือข้อมูลภายในไฟล์

นอกจากคุณสมบัติโดยทั่วไปที่กล่าวไปแล้วข้างต้น โปรแกรมฟอร์เรสแอนด์ทรียังมีความสามารถในการเข้าถึงเพิ่มข้อมูลต่าง ๆ ได้มากมาย โดยแบ่งออกเป็น 2 ประเภทดังนี้

3.3.2.1 การจัดเตรียมแบบมาตรฐาน (Standard Provider) ซึ่งสามารถเข้าถึงเพิ่มข้อมูลต่อไปนี้

- File/Server Providers: DataEase, dBASE, Paradox, Q&A, R:BASE
- Lotus Notes Provider
- Spreadsheet Providers: Lotus 1-2-3, Microsoft Excel
- Flat File Providers: Btrieve, ASCII fixed-length, ASCII variable-length, Binary fixed-length
- F&T/W View Providers: Active Views, View File (V3 and V2)
- F&T/W V3 Statistics Provider
- F&T/W V2 Statistics Provider

3.3.2.2 การจัดเตรียมแบบไคลเอ็นต์เซิร์ฟเวอร์ (Client/Server Provider) ซึ่งสามารถเข้าถึงฐานข้อมูลต่อไปนี้

- ORACLE Provider: ORACLE
- Netware SQL Provider: Netware SQL
- SQLBase Provider: SQLBase
- SQL Server Provider: SQL Server
- Ingres Provider: Ingres

3.3.3 การแปลงเพิ่มข้อมูลบีทรีฟให้เป็นเพิ่มข้อความ

ในกรณีนี้จะใช้โปรแกรมที่มีชื่อว่าฟลิก (FLIC) ย่อมาจาก Flat File Interface Construction Kit เป็นโปรแกรมที่ช่วยในการกำหนดโครงสร้างของเพิ่มข้อมูล (file structure) และสร้างค่านิยามเพิ่มข้อมูล (definition file) ให้ ซึ่งการแปลงเพิ่มข้อมูลบีทรีฟ จะต้องมีการติดตั้งการเตรียมเพิ่มข้อมูลข้อความ (Flat File Provider) ไว้ก่อน จึงจะสามารถเข้าถึงข้อมูลในเพิ่มข้อมูลบีทรีฟได้ โดยมีขั้นตอนการทำงานดังนี้

3.3.3.1 สร้างค่านิยามเพิ่มข้อมูล (definition file) โดยคลิกที่ปุ่มฟลิก (FLIC) จะปรากฏจอภาพดังรูปที่ 3.7

Forest & Trees Definition File Builder DOS Version 3.02 (c) Copyright 1990 - 1992, Channel Computing, Inc. All Rights Reserved			
File Name : File Type : ASCII VARIABLE LENGTH FIELDS Number Of Header Lines : 0 Record Separator : \n Column Separator : , Text Delimiter Character..... : * NULL Character(s) : -0- Date Format : MM/DD/YY Number of Records To Sample ... : 25			
F1	F2	F3	F4
Help	File	Quit	Columns

รูปที่ 3.7 แสดงจอภาพแรกในการสร้างนิยามสำหรับเพิ่มข้อมูลบีทรีฟ

หลังจากนั้นกำหนดค่าต่าง ๆ ดังนี้

File Name : ให้ใส่ชื่อเพิ่มข้อมูลบีทรีฟ เช่น workyear.dat

File Type : ให้เลือก Btrieve RECORD MANAGER

Record Separator : ให้ใส่ความยาวของเรคอร์ด เช่น 71

(ถ้าไม่ใส่โปรแกรมจะคำนวณให้เอง หลังจากกำหนดชื่อคอลัมน์เรียบร้อยแล้ว)

จะปรากฏจอภาพดังรูปที่ 3.8

Forest & Trees Definition File Builder
DOS Version 3.02
(c) Copyright 1990 - 1992, Channel Computing, Inc. All Rights Reserved

File Name : c:\ftw3\train\workyear.dat
File Type : Btrieve RECORD MANAGER
Number Of Header Lines : 0
Record Separator : (LEN=71)
Column Separator :
Text Delimiter Character..... :
NULL Character(s) : -0-
Date Format : MM/DD/YY
Number of Records To Sample . : 25

F1	F2	F3	F4
Help	File	Quit	Columns

รูปที่ 3.8 แสดงจอภาพที่สองในการสร้างนิยามสำหรับเพิ่มข้อมูลบีทรีฟ

หลังจากนั้นให้กดปุ่ม F4 จะปรากฏจอภาพดังรูปที่ 3.9

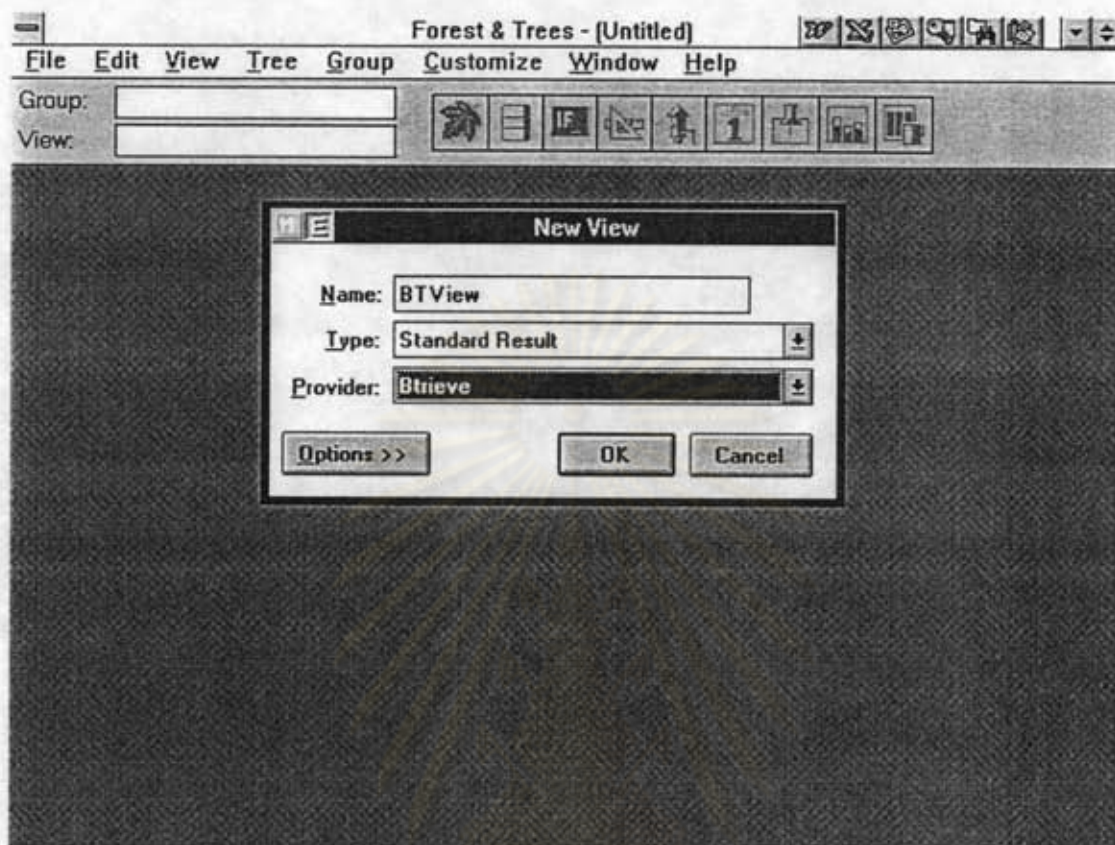
Forest & Trees Definition File Builder
DOS Version 3.02
(c) Copyright 1990 - 1992, Channel Computing, Inc. All Rights Reserved

File Name : c:\ftw3\train\workyear.dat

Binary File Columns			
COLUMN NAME	TYPE	WIDTH	VALUE
projectid	L String	7	OAA103
adjustplan	Unsigned Int.	2	0
estyear	Unsigned Int.	2	34
estmanmonth	IEEE Double	8	9
estfinished	IEEE Double	8	100
	F2	F7	F8
Record:1	Next	Insert	Delete

รูปที่ 3.9 แสดงจอภาพที่สามในการสร้างนิยามสำหรับเพิ่มข้อมูลบีทรีฟ

3.3.3.2.1 สร้างวิว (Views) โดยใช้คำสั่งวิว (View) และเลือกนิว (New) เมื่อเป็นการสร้างวิวในครั้งแรก จะปรากฏจอภาพดังรูปที่ 3.11




รูปที่ 3.11 แสดงจอภาพการสร้างวิวสำหรับเพิ่มข้อมูลบรีฟ

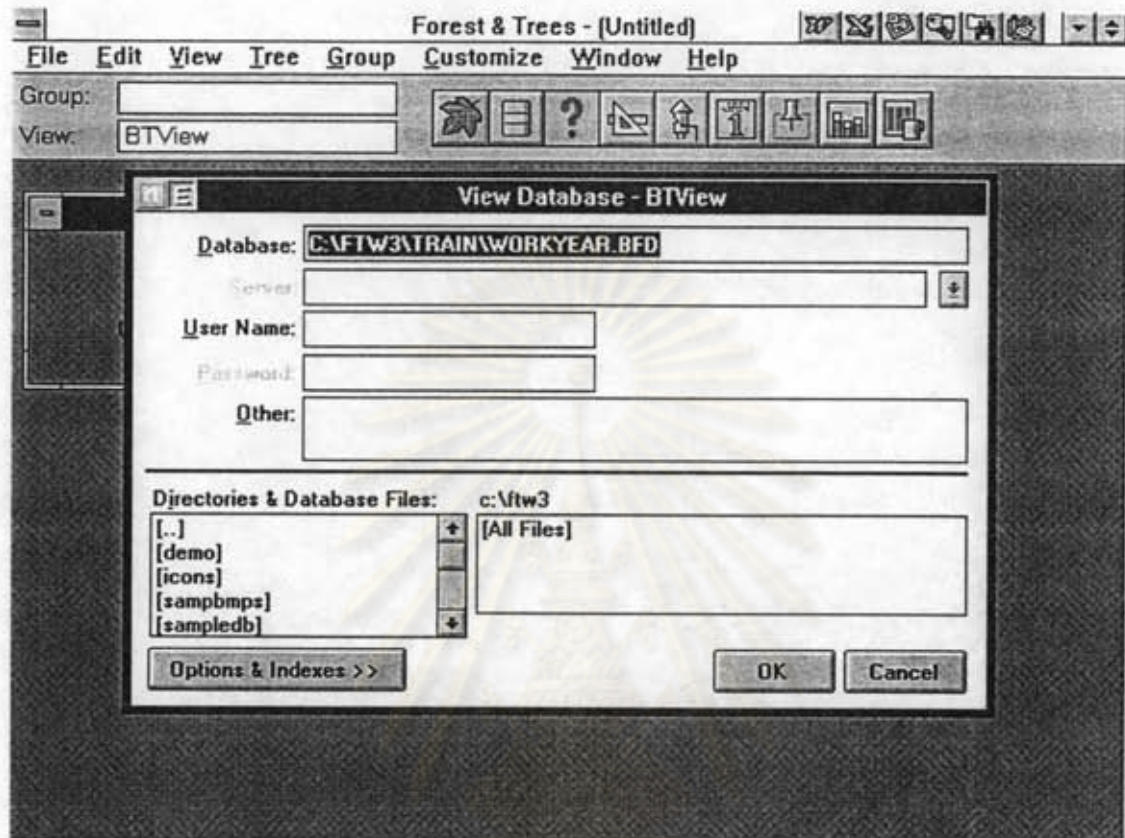
หลังจากนั้นกำหนดค่าต่าง ๆ ดังนี้

Name : ให้กำหนดชื่อวิว เช่น BTView

Type : กำหนดประเภทของวิว เช่น Standard Result

Provider : ในกรณีนี้จะต้องกำหนด Btrieve

3.3.3.2.2 สร้างดาต้าเบส (Database) โดยคลิกที่ปุ่มดาต้าเบส  จะปรากฏจอภาพดังรูปที่ 3.12



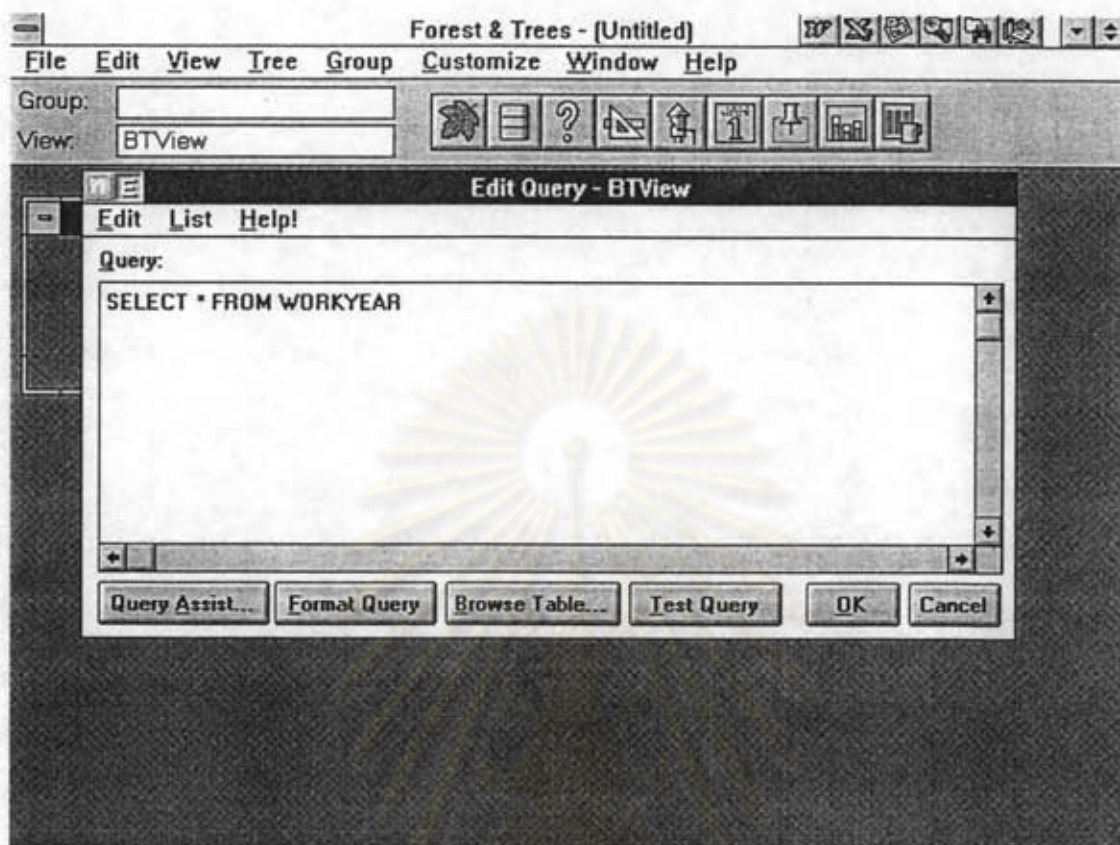
รูปที่ 3.12 แสดงจอภาพการกำหนดดาต้าเบสสำหรับเพิ่มข้อมูลบิทรีฟ

หลังจากนั้นกำหนดค่าต่าง ๆ ดังนี้

Database : ให้กำหนดชื่อแฟ้มข้อมูลที่สร้างด้วยโปรแกรมฟลิค เช่น Workyear.BFD

จุฬาลงกรณ์มหาวิทยาลัย


3.3.3.2.3 สร้างคิวรี (Query) โดยคลิกที่ปุ่มคิวรี จะปรากฏจอภาพดังรูปที่ 3.13

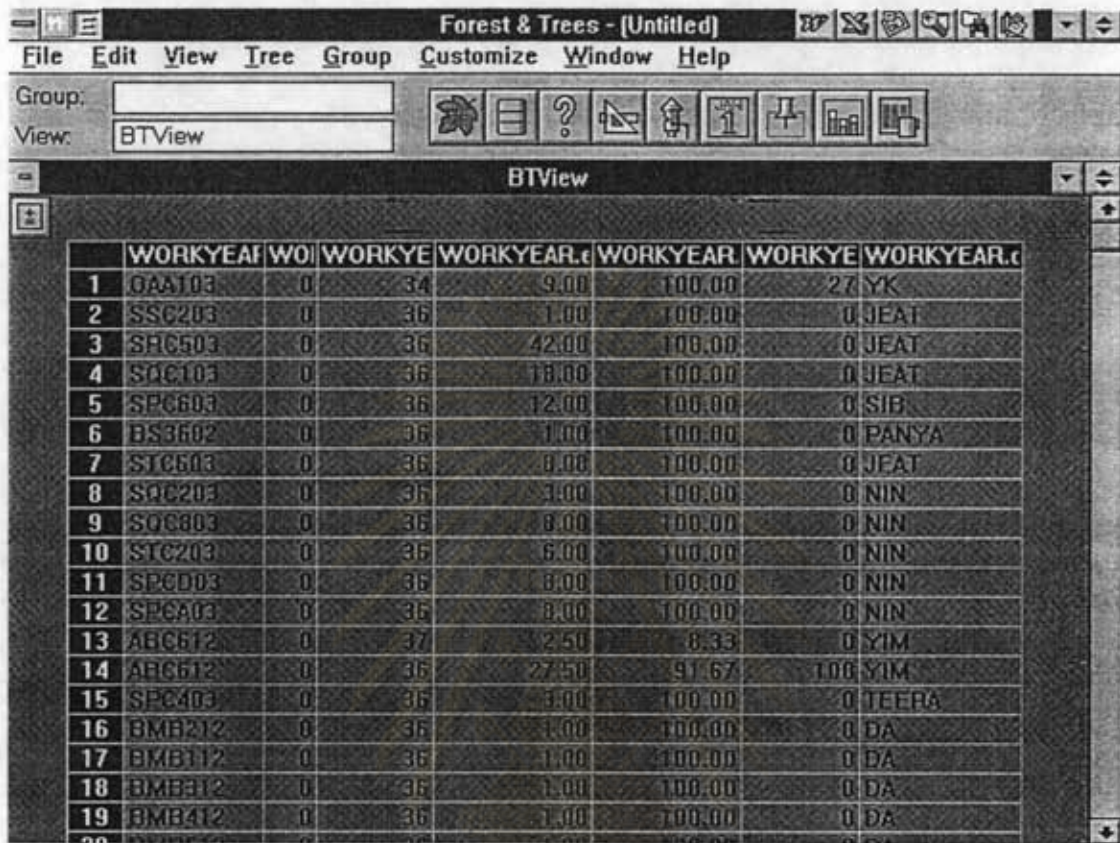


รูปที่ 3.13 แสดงจอภาพการสร้างคิวรีสำหรับเพิ่มข้อมูลบี่ทรีฟ

หลังจากนั้นสามารถป้อนคำสั่งที่ใช้ในการสอบถามได้ตามต้องการ ตัวอย่างเช่น `select * from workyear`

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

3.3.3.2.4 การเรียกดูข้อมูล (Calculate) โดยคลิกที่ปุ่มคำนวณ  จะได้ข้อมูลจากแฟ้มข้อมูลบิทรืฟปรากฏขึ้นมาในรูปของตาราง ดังรูปที่ 3.14

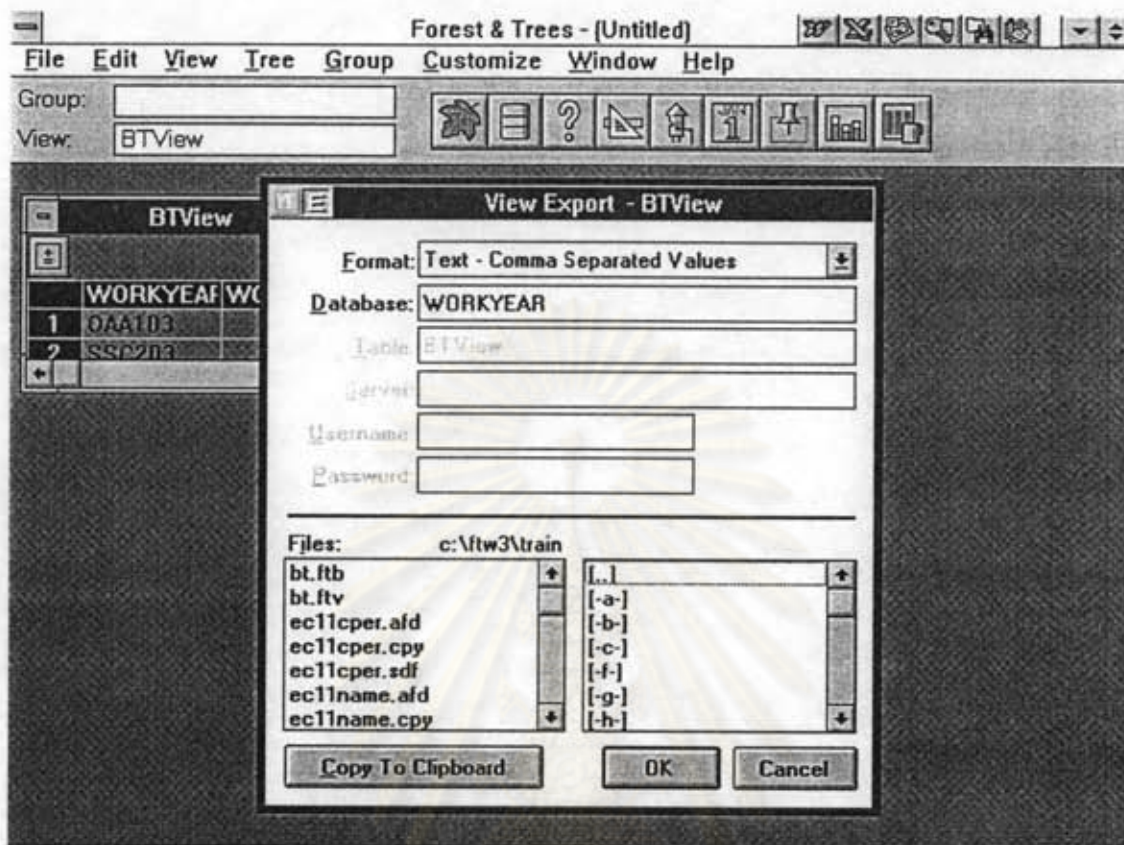


	WORKYEAF	WO	WORKYE	WORKYEAR.€	WORKYEAR	WORKYE	WORKYEAR.¢
1	DAA103	0	34	9.00	100.00	27	YK
2	SSC203	0	36	1.00	100.00	0	JEAT
3	SPC503	0	36	42.00	100.00	0	JEAT
4	SQC103	0	36	18.00	100.00	0	JEAT
5	SPC603	0	36	12.00	100.00	0	SIB
6	BS3602	0	36	1.00	100.00	0	PANYA
7	STC603	0	36	0.00	100.00	0	JEAT
8	SQC203	0	36	3.00	100.00	0	NIN
9	SQC803	0	36	0.00	100.00	0	NIN
10	STC203	0	36	6.00	100.00	0	NIN
11	SPCDD3	0	36	8.00	100.00	0	NIN
12	SPCA03	0	36	8.00	100.00	0	NIN
13	ABC612	0	37	2.50	8.33	0	YIM
14	ABC612	0	36	27.50	91.67	100	YIM
15	SPC403	0	36	3.00	100.00	0	TEEPA
16	BMB212	0	36	1.00	100.00	0	DA
17	BMB112	0	36	1.00	100.00	0	DA
18	BMB312	0	36	1.00	100.00	0	DA
19	BMB412	0	36	1.00	100.00	0	DA

รูปที่ 3.14 แสดงจอภาพการเรียกดูข้อมูลในแฟ้มข้อมูลบิทรืฟ

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

3.3.3.2.5 การถ่ายข้อมูลออก (Export) โดยใช้คำสั่งวิว (View) แล้วเลือกเอ็กพอร์ต (Export) จะปรากฏจอภาพดังรูปที่ 3.15



รูปที่ 3.15 แสดงจอภาพการถ่ายข้อมูลออกจากแฟ้มข้อมูลบีทีวีพี

ให้กำหนดค่าต่าง ๆ ดังนี้

Format : ให้เลือก Text - Comma Separated Values

Database : ให้ใส่ชื่อแฟ้มข้อความที่ต้องการ เช่น workyear จะได้นามสกุล .CSV

3.3.3.2.6 ตัวอย่างเพิ่มข้อความ หลังจากทำการถ่ายข้อมูลออก (Export) เรียบร้อยแล้ว จะได้ตัวอย่างเพิ่มข้อความ workyear.csv ดังรูปที่ 3.16

```
workyear.projectid,workyear.adjustplan,workyear.estyear,workyear.estmanmonth,
workyear.estfinished,workyear.estprogram,workyear.operatorid
"OAA103",0,34,9.0,100.0,27,"YK"
"SSC203",0,36,1.0,100.0,0,"JEAT"
"SRC503",0,36,42.0,100.0,0,"JEAT"
"SOC103",0,36,18.0,100.0,0,"JEAT"
"SPC603",0,36,12.0,100.0,0,"SIB"
```

รูปที่ 3.16 แสดงเพิ่มข้อความที่ได้จากแฟ้มข้อมูลบีทีวีพี

3.3.4 การแปลงข้อมูลในฐานข้อมูลออร์เคิลให้เป็นแฟ้มข้อความ

การแปลงข้อมูลในฐานข้อมูลออร์เคิลให้เป็นแฟ้มข้อความนั้น สามารถทำได้หลายวิธี โดยงานวิจัยนี้จะขอเสนอแนะไว้ 2 วิธีคือ

3.3.4.1 โดยใช้โปรแกรมเอสคิวแอลพลัส (SQLPLUS) เป็นโปรแกรมที่มีมากับชุดโปรแกรมออร์เคิล หลังจากเรียกคำสั่ง SQLPLUS แล้วจะปรากฏข้อความพร้อมรับ (prompt) ดังนี้

```
SQLPLUS> _
```

หลังจากนั้นให้ใช้คำสั่ง SPOOL แล้วตามด้วยชื่อแฟ้มข้อมูลข้อความที่ต้องการเก็บข้อมูล เช่นต้องการเพิ่มข้อมูลข้อความชื่อ EMP.TXT ให้ป้อนคำสั่งดังนี้

```
SQLPLUS> SPOOL EMP.TXT
```

หลังจากนั้นให้ใช้ภาษาสอบถามเชิงโครงสร้าง (SQL) ในการเรียกใช้ข้อมูล เช่นต้องการเรียกดูข้อมูลในตาราง (table) ที่ชื่อว่า EMP โดยระบุทุกคอลัมน์ในตารางนี้ ให้ป้อนคำสั่งดังนี้

```
SQLPLUS> SELECT * FROM EMP;
```

จะปรากฏข้อมูลในตาราง EMP ขึ้นมาบนจอภาพพร้อมกับบันทึกลงไปแฟ้มข้อมูล EMP.TXT ดังรูปที่ 3.17 หลังจากนั้นให้ปิดแฟ้มข้อมูล EMP.TXT โดยใช้คำสั่ง SPOOL OFF ดังนี้

```
SQLPLUS> SPOOL OFF
```

```
SQL> select * from emp;
```

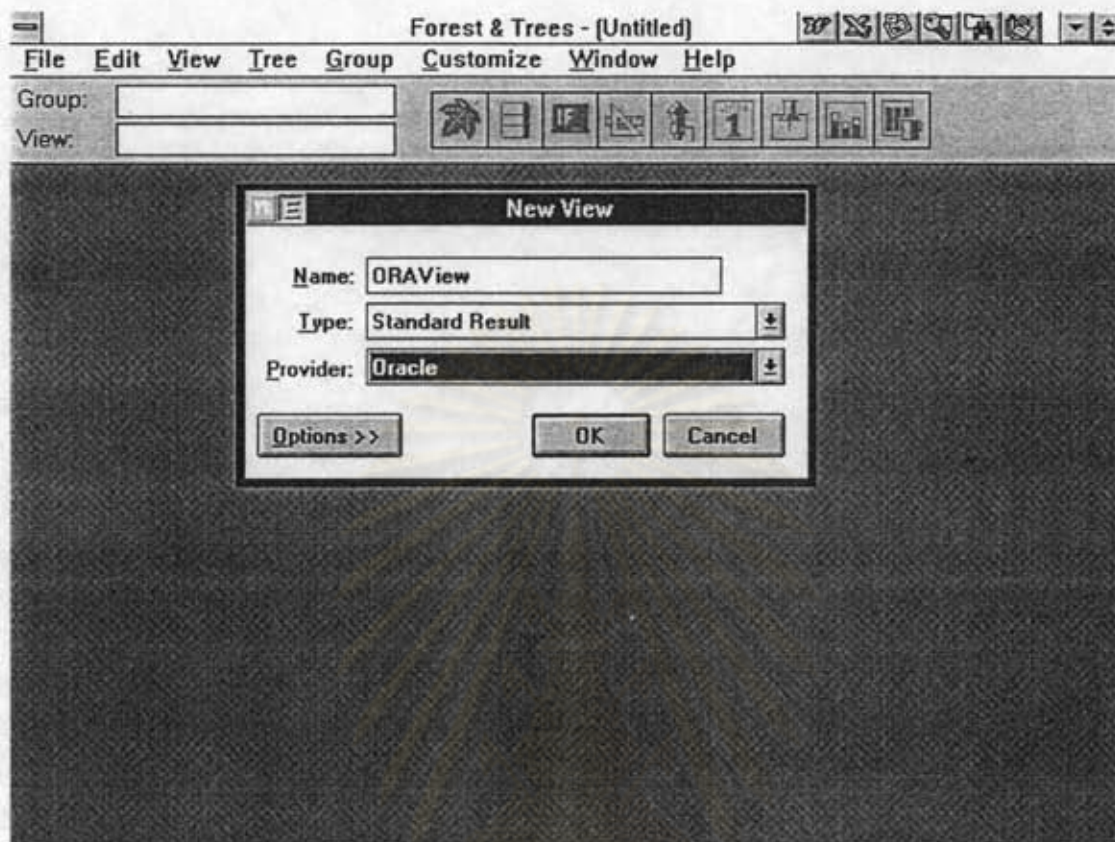
EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	7902	17-DEC-80	800		20
7499	ALLEN	SALESMAN	7698	20-FEB-81	1600	300	30
7521	WARD	SALESMAN	7698	22-FEB-81	1250	500	30
7566	JONES	MANAGER	7839	02-APR-81	2975		20
7654	MARTIN	SALESMAN	7698	28-SEP-81	1250	1400	30
7698	BLAKE	MANAGER	7839	01-MAY-81	2850		30
7782	CLARK	MANAGER	7839	09-JUN-81	2450		10
7788	SCOTT	ANALYST	7566	09-DEC-82	3000		20
7839	KING	PRESIDENT		17-NOV-81	5000		10
7844	TURNER	SALESMAN	7698	08-SEP-81	1500	0	30
7876	ADAMS	CLERK	7788	12-JAN-83	1100		20
7900	JAMES	CLERK	7698	03-DEC-81	950		30
7902	FORD	ANALYST	7566	03-DEC-81	3000		20
7934	MILLER	CLERK	7782	23-JAN-82	1300		10

14 rows selected.
SQL> spool off

รูปที่ 3.17 แสดงแฟ้มข้อความที่ได้จากฐานข้อมูลออราเคิลโดยวิธีที่ 1

3.3.4.2 โดยใช้โปรแกรมฟอร์เรสแอนด์ทรี ซึ่งจะต้องเรียกโปรแกรมเอสคิวแอลเน็ต (SQL*Net) ที่มีมากับชุดโปรแกรมออราเคิลก่อนเรียกโปรแกรมฟอร์เรสแอนด์ทรี โปรแกรมเอสคิวแอลเน็ตจะมีชื่อในการเรียกใช้งานแตกต่างกันออกไป ขึ้นอยู่กับโปรโตคอล (Protocol) ที่ใช้กับเครื่องนั้น ๆ เช่น โปรแกรมที่ชื่อว่า เอสคิวแอลเอสพีเอ็กซ์ (SQLSPX.EXE) จะใช้กับโปรโตคอลเอสพีเอ็กซ์ไอพีเอ็กซ์ (SPX/IPX) บนระบบปฏิบัติการโนเวลเน็ตแวร์ หรือ เอสคิวแอลทีซีพี (SQLTCP.EXE) จะใช้กับโปรโตคอลทีซีพีไอพี (TCP/IP) บนระบบปฏิบัติการยูนิกซ์ โดยที่โปรแกรมเอสคิวแอลเน็ตนี้จะทำหน้าที่ ทำให้ไคลเอ็นต์ (Client) สามารถติดต่อกับออราเคิลเซิร์ฟเวอร์ (Oracle Server) โดยผ่านทางเครือข่ายได้ หลังจากเรียกโปรแกรมเอสคิวแอลเน็ตแล้ว ให้กดดับเบิลคลิกที่ปุ่มฟอร์เรสแอนด์ทรี และทำตามขั้นตอนต่อไปนี้

3.3.4.2.1 สร้างวิว (Views) โดยใช้คำสั่งวิว (View) และเลือกนิว (New) เมื่อเป็นการสร้าง วิวในครั้งแรก จะปรากฏจอภาพดังรูปที่ 3.18



รูปที่ 3.18 แสดงจอภาพการสร้างวิวสำหรับฐานข้อมูลอราเคิล

หลังจากนั้นกำหนดค่าต่าง ๆ ดังนี้

Name : ให้กำหนดชื่อวิว เช่น ORAView

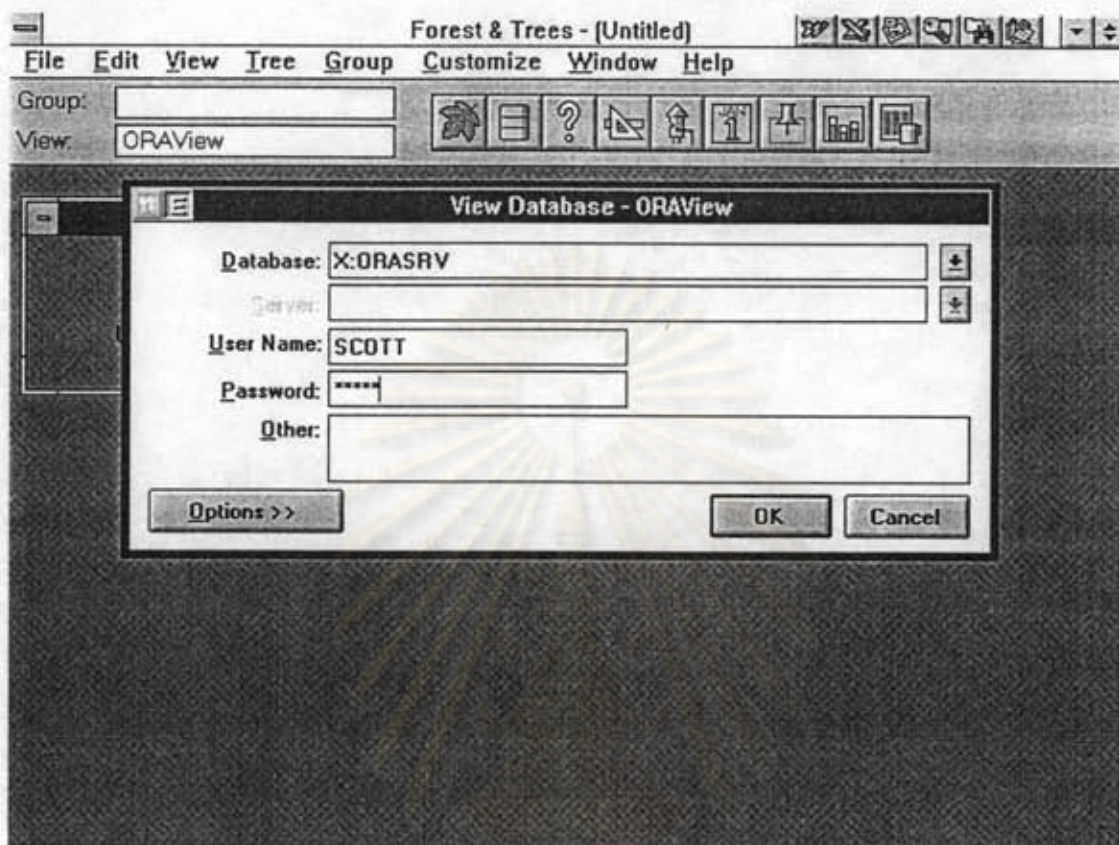
Type : กำหนดประเภทของวิว เช่น Standard Result

Provider : ในกรณีนี้จะต้องกำหนด Oracle

จุฬาลงกรณ์มหาวิทยาลัย

3.3.4.2.2 สร้างดาต้าเบส (Database) โดยคลิกที่ปุ่มดาต้าเบส  จะปรากฏจอภาพดัง

รูปที่ 3.19



รูปที่ 3.19 แสดงจอภาพการกำหนดดาต้าเบสสำหรับฐานข้อมูลอวาคีล

หลังจากนั้นกำหนดค่าต่าง ๆ ดังนี้


Database : กำหนดชื่อฐานข้อมูลที่ใช้ เช่น

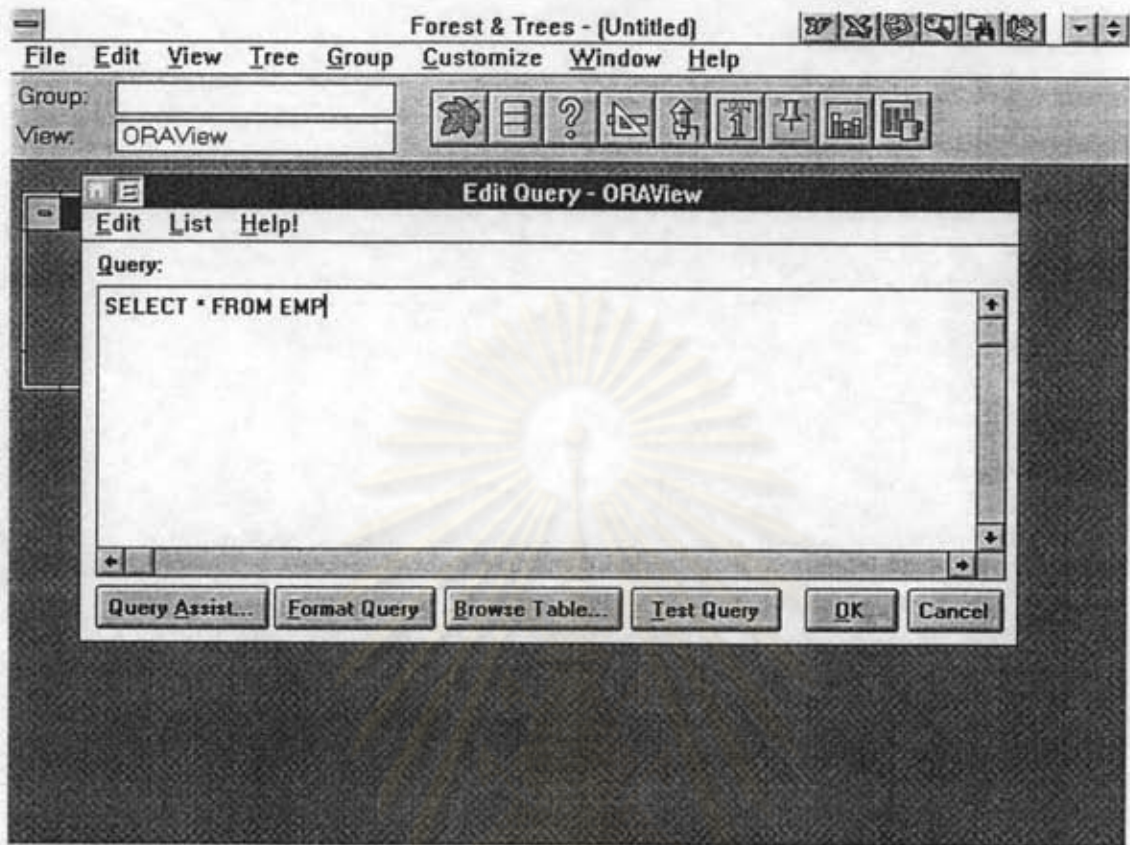
X:ORASRV กรณีที่ใช้โปรโตคอล SPX/IPX และฐานข้อมูลชื่อ ORASRV

T:ILFLW:LW กรณีที่ใช้โปรโตคอล TCP/IP เครื่องชื่อ ILFLW และฐานข้อมูลชื่อ LW

User Name : กำหนดชื่อผู้ใช้ที่มีสิทธิ์ในการใช้ฐานข้อมูลอวาคีล เช่น Scott

Password : กำหนดรหัสผ่านในการเข้าใช้ฐานข้อมูลอวาคีล เช่น Tiger


3.3.4.2.3 สร้างคิวรี (Query) โดยคลิกที่ปุ่มคิวรี  จะปรากฏจอภาพดังรูปที่ 3.20

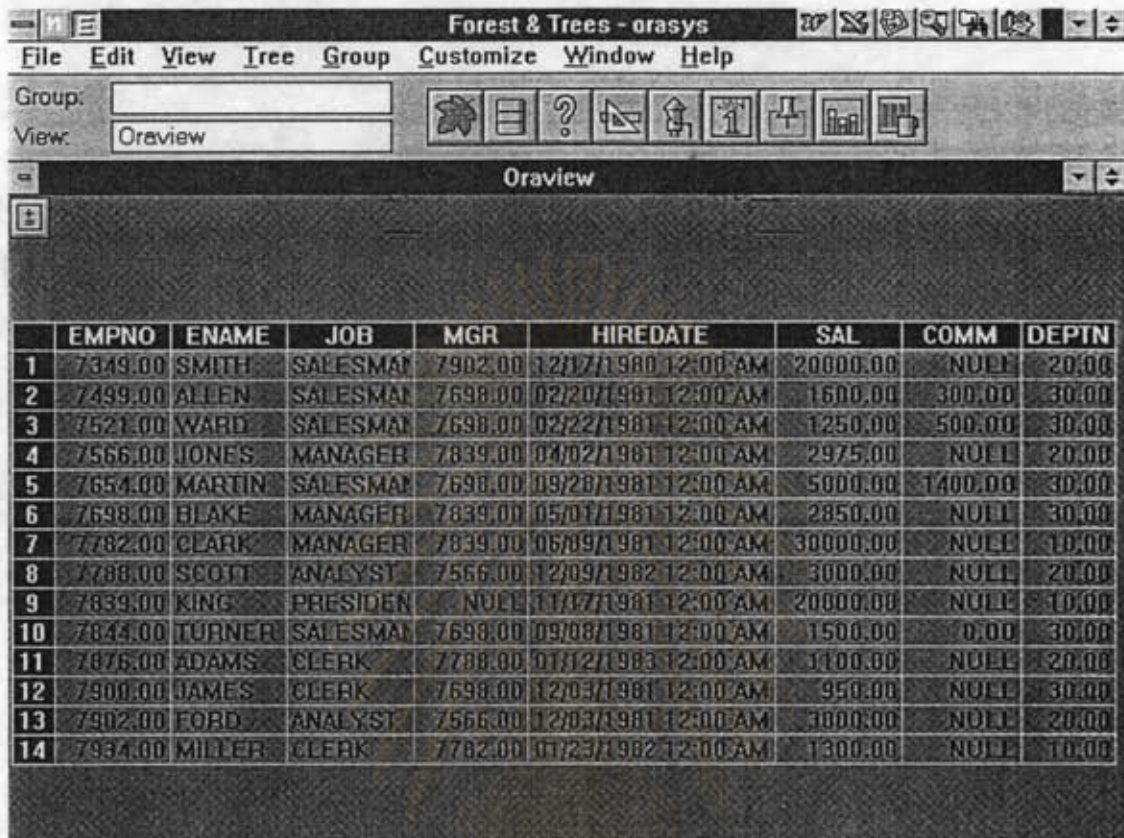


รูปที่ 3.20 จอภาพแสดงการสร้างคิวรีสำหรับฐานข้อมูลออราเคิล

หลังจากนั้นสามารถป้อนคำสั่งที่ใช้ในการสอบถามได้ตามต้องการ ตัวอย่างเช่น `select * from emp`

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

3.3.4.2.4 การเรียกดูข้อมูล (Calculate) โดยคลิกที่ปุ่มคำนวณ  จะได้ข้อมูลจากฐานข้อมูลออราเคิลปรากฏขึ้นมาในรูปของตาราง ดังจอภาพที่ 3.21



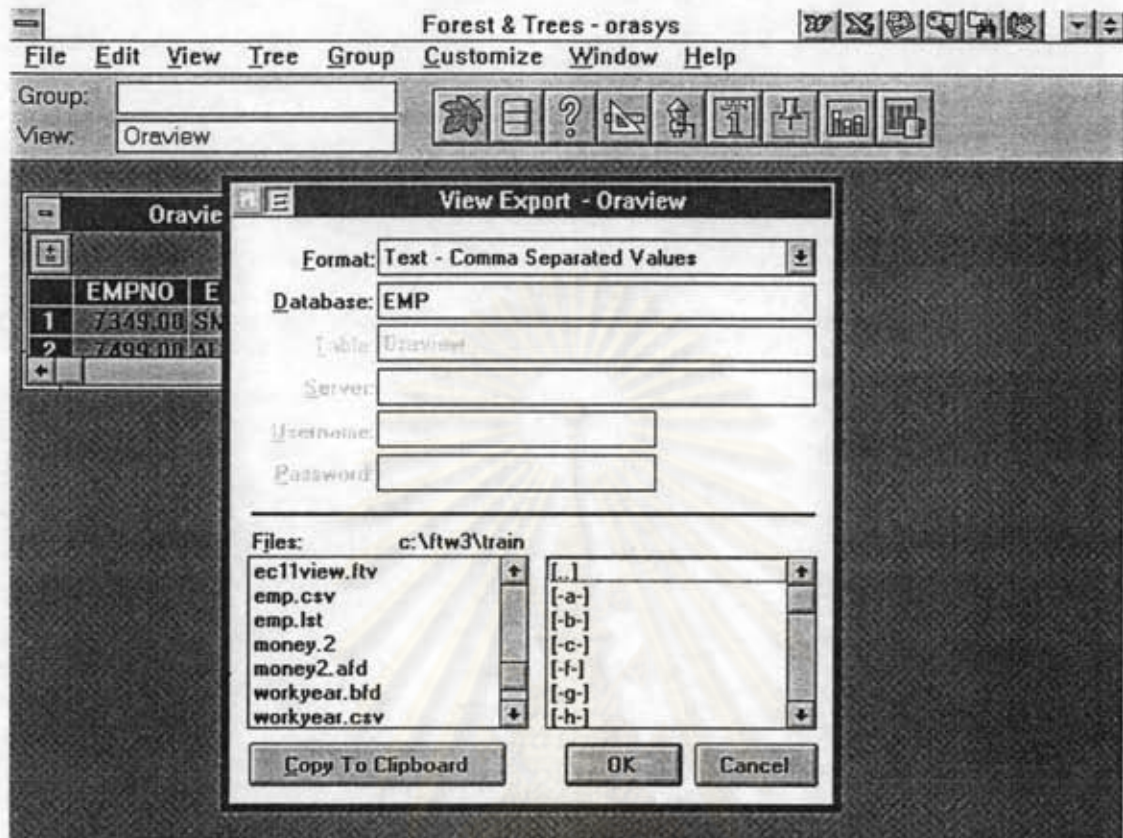
The screenshot shows a software interface titled "Forest & Trees - orasys". It has a menu bar with "File", "Edit", "View", "Tree", "Group", "Customize", "Window", and "Help". Below the menu bar are input fields for "Group:" and "View:" (set to "Oraview"). A toolbar contains various icons including a tree, a list, a question mark, a magnifying glass, a printer, and a refresh icon. The main area is titled "Oraview" and displays a table with 14 rows and 9 columns. The columns are labeled EMPNO, ENAME, JOB, MGR, HIREDATE, SAL, COMM, and DEPTN. The data in the table is as follows:

	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTN
1	7349.00	SMITH	SALESMAN	7902.00	12/17/1980 12:00 AM	20000.00	NULL	20.00
2	7499.00	ALLEN	SALESMAN	7698.00	02/20/1981 12:00 AM	1600.00	300.00	30.00
3	7521.00	WARD	SALESMAN	7698.00	02/22/1981 12:00 AM	1250.00	500.00	30.00
4	7566.00	JONES	MANAGER	7839.00	04/02/1981 12:00 AM	2975.00	NULL	20.00
5	7654.00	MARTIN	SALESMAN	7698.00	09/28/1981 12:00 AM	5000.00	1400.00	30.00
6	7698.00	BLAKE	MANAGER	7839.00	05/01/1981 12:00 AM	2850.00	NULL	30.00
7	7782.00	CLARK	MANAGER	7839.00	06/09/1981 12:00 AM	30000.00	NULL	10.00
8	7788.00	SCOTT	ANALYST	7566.00	12/09/1982 12:00 AM	3000.00	NULL	20.00
9	7839.00	KING	PRESIDENT	NULL	11/07/1981 12:00 AM	20000.00	NULL	10.00
10	7844.00	TURNER	SALESMAN	7698.00	09/08/1981 12:00 AM	1500.00	0.00	30.00
11	7876.00	ADAMS	CLERK	7788.00	01/12/1983 12:00 AM	1100.00	NULL	20.00
12	7900.00	JAMES	CLERK	7698.00	12/03/1981 12:00 AM	950.00	NULL	30.00
13	7902.00	FORD	ANALYST	7566.00	12/03/1981 12:00 AM	3000.00	NULL	20.00
14	7934.00	MILLER	CLERK	7782.00	01/23/1982 12:00 AM	1300.00	NULL	10.00

รูปที่ 3.21 แสดงจอภาพการเรียกดูข้อมูลในฐานข้อมูลออราเคิล

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

3.3.4.2.5 การถ่ายข้อมูลออก (Export) โดยใช้คำสั่งวิว (View) แล้วเลือกเอ็กพอร์ต (Export) จะปรากฏจอภาพดังรูปที่ 3.22



รูปที่ 3.22 แสดงจอภาพการถ่ายข้อมูลออกจากฐานข้อมูลอวาเคล

ให้กำหนดค่าต่าง ๆ ดังนี้

Format : ให้เลือก Text - Comma Separated Values

Database : ให้ใส่ชื่อเพิ่มข้อความที่ต้องการ เช่น emp จะได้นามสกุล .CSV

จุฬาลงกรณ์มหาวิทยาลัย

3.3.4.2.6 ตัวอย่างเพิ่มความถี่ หลังจากทำการถ่ายข้อมูลออก (Export) เรียบร้อยแล้ว จะ
ได้ตัวอย่างเพิ่มความถี่ emp.csv ดังรูปที่ 3.23

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7349.0	"SMITH	"SALESMAN	"	,7902.0,12/17/80 12:00 AM	,20000.0	,,20.0	
7499.0	" ALLEN	"SALESMAN	"	,7698.0,2/20/81 12:00 AM	,1600.0	,300.0,30.0	
7521.0	"WARD	"SALESMAN	"	,7698.0,2/22/81 12:00 AM	,1250.0	,500.0,30.0	
7566.0	"JONES	"MANAGER	"	,7839.0,4/2/81 12:00 AM	,2975.0	,,20.0	
7654.0	"MARTIN	"SALESMAN	"	,7698.0,9/28/81 12:00 AM	,5000.0	,1400.0,30.0	
7698.0	"BLAKE	"MANAGER	"	,7839.0,5/1/81 12:00 AM	,2850.0	,,30.0	
7782.0	"CLARK	"MANAGER	"	,7839.0,6/9/81 12:00 AM	,30000.0	,,10.0	
7788.0	"SCOTT	"ANALYST	"	,7566.0,12/9/82 12:00 AM	,3000.0	,,20.0	
7839.0	"KING	"PRESIDENT"	"	,,11/17/81 12:00 AM	,20000.0	,,10.0	
7844.0	"TURNER	"SALESMAN	"	,7698.0,9/8/81 12:00 AM	,1500.0	,,0,30.0	
7876.0	"ADAMS	"CLERK	"	,7788.0,1/12/83 12:00 AM	,1100.0	,,20.0	
7900.0	"JAMES	"CLERK	"	,7698.0,12/3/81 12:00 AM	,950.0	,,30.0	
7902.0	"FORD	"ANALYST	"	,7566.0,12/3/81 12:00 AM	,3000.0	,,20.0	
7934.0	"MILLER	"CLERK	"	,7782.0,1/23/82 12:00 AM	,1300.0	,,10.0	

รูปที่ 3.23 แสดงเพิ่มความถี่ที่ได้จากฐานข้อมูลออราเคิลโดยวิธีที่ 2

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

3.3.5 การแปลงเพิ่มข้อมูลซี,โคบอล,ปาสกาลและเบสิกให้เป็นเพิ่มข้อความ

การแปลงเพิ่มข้อมูลซี,โคบอล,ปาสกาลและเบสิกให้เป็นเพิ่มข้อความนั้น มีขั้นตอนการทำงาน เหมือนกับการแปลงเพิ่มข้อมูลบีทรีฟให้เป็นเพิ่มข้อความซึ่ง มี 2 ขั้นตอนดังนี้

3.3.5.1 สร้างคำนิยามเพิ่มข้อมูล (definition file) ในส่วนนี้จะอธิบายเฉพาะจุดที่แตกต่าง จากการแปลงเพิ่มข้อมูลบีทรีฟให้เป็นเพิ่มข้อความ โดยจะเปรียบเทียบในรูปของตาราง ดังตารางที่ 3.17

เพิ่มข้อมูลบีทรีฟ	เพิ่มข้อมูลซี,โคบอล,ปาสกาลและเบสิก
1. File Type : Btrieve RECORD MANAGER (ดูรายละเอียดได้จากรูปที่ 3.7)	1. File Type : BINARY FIXED-LENGTH FIELDS
2. Record Separator : โปรแกรมจะคำนวณให้เองหลังจากกำหนดชื่อ คอลัมน์เรียบร้อยแล้ว (ดูรายละเอียดได้จากรูปที่ 3.7)	2. Record Separator : ต้องคำนวณความยาวของเรคอร์ดล่วงหน้าแล้วใส่ ค่าที่คำนวณได้
3. Data Type : ตัวอย่างเช่น ในกรณีที่เพิ่มข้อมูลบีทรีฟมีประเภทข้อมูลเป็น Integer เลือก Integer - Unsigned Int. ความกว้าง 2 ไบต์ Lstring เลือก Text String - Length Descriptor String ความกว้างเท่ากับความยาวของ สตริงบวก 1 ไบต์ Double เลือก Float - IEEE Double ความกว้าง 8 ไบต์ (ดูรายละเอียดได้จากรูปที่ 3.9)	3. Data Type : ตัวอย่างเช่น ในกรณีที่เพิ่มข้อมูลซี,โคบอล,ปาสกาลและ เบสิกมีประเภทข้อมูลเป็น Integer . เพิ่มข้อมูลซี,โคบอล,ปาสกาล,เบสิก เลือก Integer - Unsigned Int ความกว้าง 2 ไบต์ String เพิ่มข้อมูลซี เลือก Text String - Null Terminated String ความกว้างเท่ากับความยาวของสตริง เพิ่มข้อมูลปาสกาล เลือก Text String - Length Descriptor String ความกว้างเท่ากับความยาวของสตริงบวก 1

ตารางที่ 3.17 แสดงข้อแตกต่างการสร้างคำนิยามระหว่างเพิ่มข้อมูลบีทรีฟและเพิ่มข้อมูลซี,โคบอล,ปาสกาล
และเบสิก

เพิ่มข้อมูลบีทรีฟ	เพิ่มข้อมูลซี,โคบอล,ปาสกาลและเบสิก
	เพิ่มข้อมูลโคบอล,เบสิก เลือก Text String - Fixed Length ความกว้างเท่ากับความยาวของสตริง Double เพิ่มข้อมูลซี,เบสิก เลือก Float - IEEE Double ความกว้าง 8 ไบต์ เพิ่มข้อมูลโคบอล Picture 9 เช่น S9(12) เลือก ASCII - Double ความกว้าง 12 ไบต์ เพิ่มข้อมูลปาสกาล เช่น Real เลือก Packed - BCD ความกว้าง 10 ไบต์
4. Save Definition File โปรแกรมจะทำการบันทึกข้อมูลลงในเพิ่มข้อมูล ชื่อเดียวกับที่กำหนดไว้ โดยให้นามสกุล BFD	4. Save Definition File โปรแกรมจะทำการบันทึกข้อมูลลงในเพิ่มข้อมูล ชื่อเดียวกับที่กำหนดไว้ โดยให้นามสกุล AFD

ตารางที่ 3.17 แสดงข้อแตกต่างการสร้างคำนิยามระหว่างเพิ่มข้อมูลบีทรีฟและเพิ่มข้อมูลซี,โคบอล,ปาสกาล และเบสิก (ต่อ)

3.3.5.2 การเรียกใช้โปรแกรมฟอร์เรสแอนด์ทรี (Forest & Tree) ทำนองเดียวกันกับการแปลงเพิ่มข้อมูลบีทรีฟให้เป็นเพิ่มข้อความ โดยจะเปรียบเทียบข้อแตกต่างในรูปของตารางดังนี้

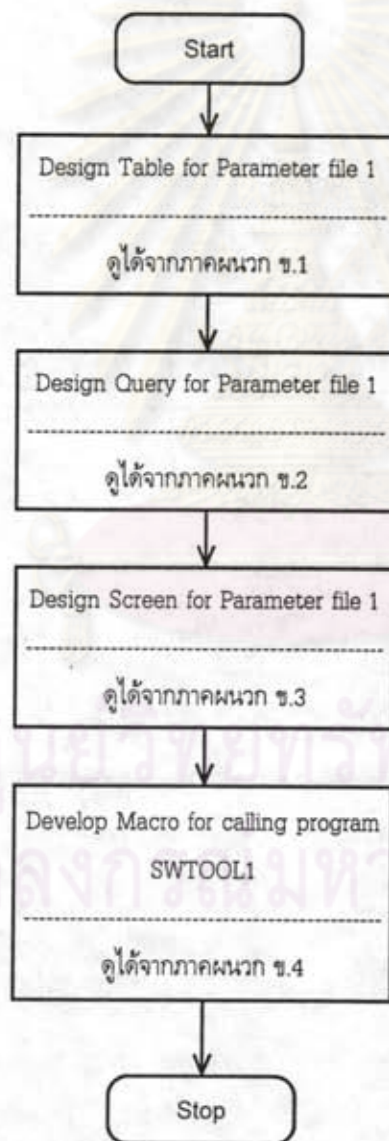
เพิ่มข้อมูลบีทรีฟ	เพิ่มข้อมูลซี,โคบอล,ปาสกาลและเบสิก
1. การสร้างวิว (Views) Provider : Btrieve (ดูรายละเอียดได้จากรูปที่ 3.11)	1. การสร้างวิว (Views) Provider : FLIC (Flat File)
2. การสร้างดาต้าเบส (Database) Database : Filename.BFD (ดูรายละเอียดได้จากรูปที่ 3.12)	2. การสร้างดาต้าเบส (Database) Database : Filename.AFD

ตารางที่ 3.18 แสดงข้อแตกต่างการกำหนดค่าในโปรแกรมฟอร์เรสแอนด์ทรีระหว่างเพิ่มข้อมูลบีทรีฟ และเพิ่มข้อมูลซี,โคบอล,ปาสกาลและเบสิก

3.4 การออกแบบและพัฒนาตัวเชื่อมประสานกับผู้ใช้ (User Interface)

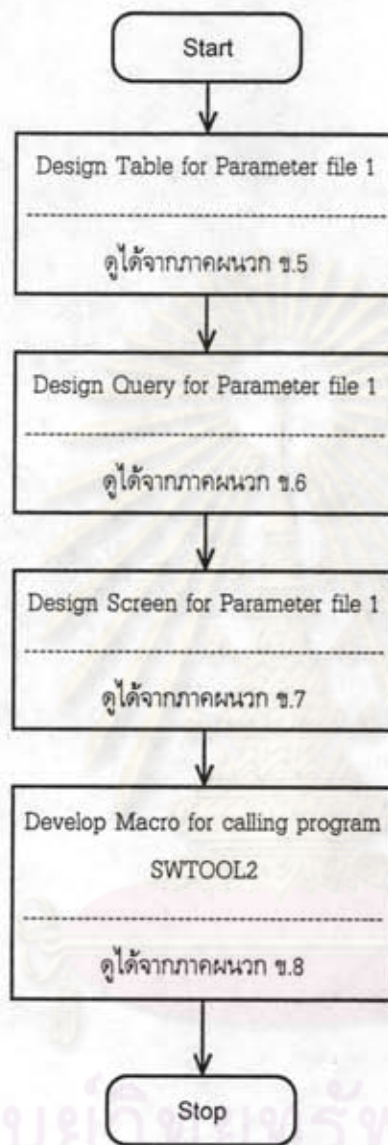
หลังจากการออกแบบและพัฒนาเครื่องมือทางซอฟต์แวร์สำหรับแปลงผันเพิ่มข้อมูลที่สามารถใช้งานบนระบบปฏิบัติการเอ็มเอสดอส (MS-DOS) ได้เรียบร้อยแล้ว เนื่องจากต้องการให้โปรแกรมมีความสะดวกและง่ายในการเรียกใช้ จึงได้ทำการออกแบบในส่วนของตัวเชื่อมประสานกับผู้ใช้ โดยใช้โปรแกรมไมโครซอฟต์แอ็กเซส (Microsoft Access)

3.4.1 การออกแบบและพัฒนาตัวเชื่อมประสานกับผู้ใช้สำหรับเครื่องมือทางซอฟต์แวร์ตัวที่ 1
แบ่งตามขั้นตอนการทำงานได้ดังต่อไปนี้



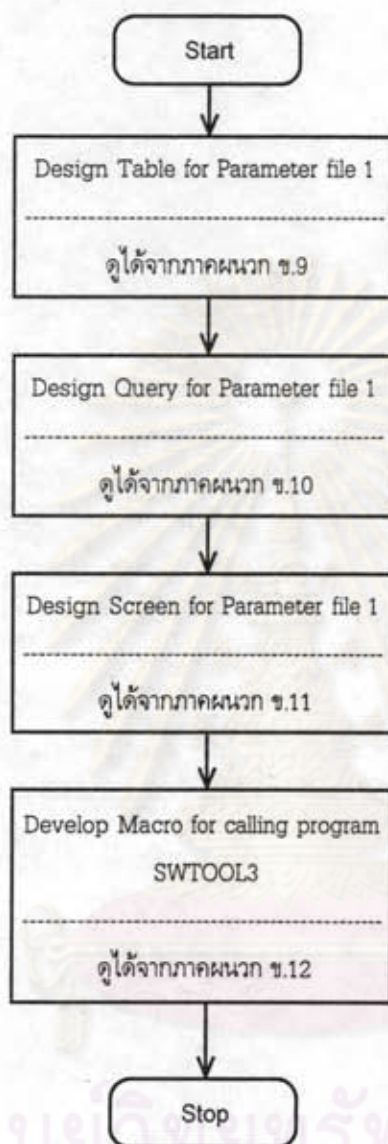
รูปที่ 3.24 แสดงรูปแบบการออกแบบและพัฒนาตัวเชื่อมประสานกับผู้ใช้ SWTOOL1

3.4.2 การออกแบบและพัฒนาตัวเชื่อมประสานกับผู้ใช้สำหรับเครื่องมือทางซอฟต์แวร์ตัวที่ 2
แบ่งตามขั้นตอนการทำงานได้ดังต่อไปนี้



รูปที่ 3.25 แสดงรูปแบบการออกแบบและพัฒนาตัวเชื่อมประสานกับผู้ใช้ SWTOOL2

3.4.3 การออกแบบและพัฒนาตัวเชื่อมประสานกับผู้ใช้สำหรับเครื่องมือทางซอฟต์แวร์ตัวที่ 3 แบ่งตามขั้นตอนการทำงานได้ดังต่อไปนี้



รูปที่ 3.26 แสดงรูปแบบการออกแบบและพัฒนาตัวเชื่อมประสานกับผู้ใช้ SWTOOL3