Client - Server Database Application Programming (Delphi + SQL server)

การเขียนโปรแกรมด้วย Delphi ขั้นพื้นฐาน

โปรแกรม delphi เป็นผลิตภัณฑ์ของบริษัทบอร์แลนด์ (ปัจจุบันเปลี่ยนชื่อเป็น inprise) เป็นบริษัทที่ชำนาญในการสร้างคอมไพเลอร์และเครื่องมือที่ใช้ในการพัฒนาระบบ เช่น Turbo Pascal, Turbo C, Borland C++ เป็นต้น สำหรับโปรแกรม Delphi มีรากฐาน มาจากภาษาปาสคาลที่ได้เพิ่มความ สามารถทางด้านการเขียนโปรแกรมเชิ่งวัตถุ หรือ Object Oriented Programming เข้าไป อาจเรียกได้ว่าเป็น ภาษาแบบ Object Pascal นอกจากนี้ยังเพิ่มความสามารถในการเขียนโปรแกรมสำหรับงานแบบวินโดว์ (Visual Programming)

เวอร์ชันของ Delphi แบ่งตามการใช้งานแบ่งออกเป็น 3 เวอร์ชน ดังนี้

- Dephi Standard Version เป็นเวอร์ชันที่มีราคาต่ำสุดสำหรับโปรแกรมเมอร์ที่พัฒนางานสำหรับผู้ใช้งานคน เดียว (Stand Alone)
- Delphi Prifessional Version เป็นเวอร์ชันที่ใช้พัฒนาระบบงานบน LAN ที่มีผู้ใช้งานไม่มากนัก
- Delphi Enterprise Version เป็นเวอร์ชันที่เต็มรูปแบบ ใช้สำหรับพัฒนางานเพื่อใช้งานในระบบงานแบบ
 Client / Server ที่มีผู้ใช้งานจำนวนมากและพัฒนาระบบงานที่ใช้กับอินเตอร์เน็ต

ความต้องการด้านฮาร์ดแวร์ Delphi ต้องการระบบคอมพิวเตอร์ที่มีคุณสมบัติต่อไปนี้ Delphi 5.0

- โพรเซลเซอร์แบบ Pentium 90 MHz. หรือสูงกว่า (ไม่ควรต่ำกว่า 166 MHz.)
- หน่วยความจำอย่างต่ำ 32 MB (ควรใช้ 64 MHz.)
- พื้นที่ว่างในฮาร์ดดิสก์อย่างน้อย 80 MB (ถ้าติดตั้งเต็มใช้ประมาณ 220 MB)
- ระบบปฏิบัติการ Windows 95/98 หรือ Windows NT 4.0 ซึ่งติดตั้ง Service Pack 3 หรือสูงกว่า

ส่วนประกอบของ IDE (Integrated Development Environment) ของ Delphi

แบ่งออกเป็น 5 ส่วน



การทำงานของ Delhi เมื่อสั่งรันโปรแกรม

เมื่อเขียนโปรแกรมเสร็จสิ้นต้องการที่จะทดสอบโปรแกรม จะต้องใช้เมนู Run/Run เพื่อเรียกคอมไพเลอร์ มาแปลโปรแกรมต้นฉบับ (Source Programe) แล้วทำการ Link โปรแกรมเพื่อให้ได้โปรแกรมในรูปของภาษา เครื่อง (Executable Programe) ซึ่งโปรแกรมที่เป็น .EXE นั้นจะเป็นชื่อเดียวกับชื่อ Project โดยสรุปไฟล์ที่ใช้ใน โปร แกรม และขั้นตอนการแปลและLink โปรแกรมดังนี้



เมื่อได้ โปรแกรมในภาษาเครื่อง .EXE แล้วจึงจะนำไปทำงานเพื่อแสดงผลลัพธ์จอภาพที่ออกแบบไว้

เมนูของเดลไฟล์แต่ละกลุ่มมีรายละเอียดการทำงานดังนี้

- เมนู File
 - New เป็นการเรียกวินโดวส์ของการสร้าง Item ขึ้นมาเพื่อให้ระบุสิ่งที่ต้องการสร้าง ดังรูป

เราสามารถเลือกสร้างโดยใช้

- ตัวอย่างของโปรเจ็คเดิม (Template) มาใช้
- เรียกโปรแกรมผู้เชี่ยวชาญ (Expert) มาช่วยสร้างโปรเจ็ค
- เรียกโปรแกรมผู้เชี่ยวชาญ (Expert) มาช่วยสร้างฟอร์ม
- สร้างเป็นคอมโพเน้นท์ ฯลฯ
- New Application สร้างโปรเจ็คใหม่ซึ่งเป็นโปรเจ็คว่าง
- New Form ใช้ในการสร้างฟอร์มใหม่เพื่อเป็นสมาชิกในโปรเจ็ค
- New Datamodule ใช้สำหรับสร้างยูนิตที่ใช้เก็บคอนโทรลที่เกี่ยวกับการอ่านข้อมูลไว้แห่งเดียวกัน เพื่อให้ฟอร์ม อื่นเรียกใช้งาน
- Open เปิดโปรเจ็คเดิมที่มีอยู่แล้วหรือเปิดแฟ้มอื่นๆ
- Reopen เรียกแฟ้มเดิมที่เคยเปิดแล้วมาเปิดใหม่
- Save จัดเก็บแฟ้มที่กำลังเปิดใช้งาน
- Save As ใช้ในการทำสำเนาฟอร์ม/ยูนิต เพื่อเตรียมไปสร้างงานใหม่ที่คล้ายกับงานเดิม
- Save Project As ใช้ในการทำสำเนาโปรเจ็ค เพื่อเตรียมไปสร้างงานใหม่ที่คล้ายกับงานเดิม
- Save All ใช้สำหรับจัดเก็บแฟ้มทุกแฟ้มที่เปิดอยู่
- Close ใช้สำหรับปิดแฟ้มข้อมูลที่เปิดอยู่
- Close All ใช้สำหรับปิดแฟ้มข้อมูลที่เปิดอยู่ทุกแฟ้ม
- Use Unit ใช้ในการอ้างอิงกันระหว่างฟอร์มโดย ฟอร์มที่ต้องการเรียกใช้ฟอร์มอื่นจะต้องใช้คำสั่ง uses [ชื่อยูนิต ของฟอร์ม] คำสั่งนี้จะช่วยเพิ่มคำสั่ง uses ให้
- Add to Project ได้กล่าวมาแล้วในส่วนของ Speedbar
- Remove From Project ได้กล่าวมาแล้วในส่วนของ Speedbar
- Print ใช้สำหรับพิมพ์ฟอร์ม/ยูนิตออกมาที่เครื่องพิมพ์
- Exit ใช้สำหรับจบโปรแกรมเดลไฟล์
- เมนู Edit
 - Undo (ชื่อเมนูจะเปลี่ยนไปขึ้นกับการสั่งงานล่าสุด) ใช้สำหรับยกเลิกการทำงานล่าสุดในการเขียนโปรแกรมในยู
 นิต เราสามารถกำหนดให้ Undo ได้ตั้งแต่ 1 ถึง 32767 ครั้ง ข้อจำกัดของการ Undo คือถ้ามีการสร้าง Event
 เพื่อเขียนคำสั่งใหม่แล้วจะไม่สามารถ Undo สิ่งที่เกิดขึ้นสำหรับ Event เดิม

ไพฑูรย์ งิ้วทั่ง 2/12/022545

- Redo การใช้งานจะตรงข้ามกับ Undo คือทำให้สิ่งที่เคย Undo ไปแล้ว กลับมาทำใหม่
- Cut การตัดคำสั่งงาน/คอมโพเน้นท์ไปไว้ที่คลิบบอร์ด
- Copy การทำสำเนาคำสั่งงาน/คอมโพเน้นท์ไปไว้ที่คลิบบอร์ด
- Paste การนำเอาคำสั่งงาน/คอมโพเน้นท์ที่ได้ cut/copy ไว้มาวางไว้ที่จุดที่ต้องการ
- Delete การลบคำสั่งงาน/คอมโพเน้นท์โดยไม่ต้องการนำไปใส่ไว้ในคลิบบอร์ด
- Select All การเลือกคอมโพเน้นท์บนฟอร์ม ทุกอัน ใช้เพื่อกำหนดค่าให้กับ Property ของคอมโพเน้นท์ทุกคอม โพเน้นท์พร้อมกัน หรือใช้เพื่อต้องการสั่งงานเป็นกลุ่มเช่น จัดเรียงใหม่, cut/copy เป็นกลุ่มๆ
- Align to Grid เป็นการสั่งงานให้คอมโพเน้นท์ บนฟอร์มที่เลือกไว้ เลื่อนไปให้ตรงกับแนวกริดบนฟอร์ม คำสั่งนี้ จะใช้ไม่ได้ผลหากไม่ได้กำหนดลักษณะการวางแบบ ตรงแนวกริดที่ Tools/Options - Snap to Grid
- Bring to Front ในกรณีที่คอนโทรล (คอมโพเน้นท์) บนฟอร์มมีพื้นที่ช้อนทับกันอยู่ เราสามารถกำหนดให้ คอนโทรลใดอยู่ข้างหน้าได้โดยใช้เมนูนี้
- Send to Back ในกรณีที่ต้องการให้ คอนโทรลใดอยู่ข้างหลังคอนโทรลอื่นให้ใช้เมนูนี้
- Align เรียกหน้าต่าง Alignment มาเพื่อจัดเรียงคอนโทรลบนฟอร์ม ที่ได้เลือกไว้ ปกติจะเลือกเป็นกลุ่มที่ต้องการ เรียงไว้ก่อนแล้ว จึงเรียกเมนูนี้ นอกจากสั่งโดยผ่านเมนูนี้แล้วเรายังสามารถเรียกเมนูแบบ popup ขึ้นมาเพื่อ ทำงานแบบนี้ได้โดยการคลิกเมาส์ด้านขวาที่กลุ่มที่เลือกไว้ก็จะแสดงเมนูแบบ popup ให้เราเลือกหน้าต่าง Alignment จะมีลักษณะ
- Size ใช้เพื่อเรียกหน้าต่าง size เพื่อทำการจัดขนาดของคอนโทรลบนฟอร์ม หน้าต่าง size จะมีลักษณะ
- Scale ใช้เพื่อปรับขนาดและเลื่อนคอนโทรล บนฟอร์ม ในกรณีที่ต้องการขยายหรือย่อฟอร์ม ซึ่งมีหน่วยของการ ปรับเป็นเปอร์เซนต์ตั้งแต่ 25% ถึง 400% โดยที่ ค่าที่มากกว่า 100% จะเป็นการขยายขนาด ส่วนค่าที่น้อยกว่า 100% จะเป็นการย่อขนาด
- Tab Order เป็นการจัดเรียงลำดับตำแหน่ง การหยุดเมื่อกด Tab ขณะที่รันโปรแกรม
- Creation Order เป็นการกำหนดลำดับของ การสร้างคอนโทรลเมื่อฟอร์มถูกสร้างขึ้น การใช้งานในส่วนนี้จะเกิด ขึ้นในกรณีที่การออกแบบไม่ได้วางคอนโทรล ที่ควรจะเกิดขึ้นก่อนตามลำดับ ลักษณะของวินโดวส์ของ Creation Order เป็น
- Lock Control เป็นการยึดคอนโทรลบนฟอร์มไม่ให้ขยับหรือเปลี่ยนขนาดโดยไม่ตั้งใจ
- Object เป็นการแก้ไขเปลี่ยนแปลงคอนโทรลที่เป็น OLE
- เมนู Search
 - Find ใช้ค้นหาคำในยูนิต
 - Replace ใช้แทนที่คำในยูนิต
 - Search Again ใช้ค้นหาหรือแทนที่ซ้ำอีกครั้ง
 - Incremental Search ใช้ค้นหาแบบค่อยๆเพิ่มตัวอักษรที่ต้องการค้นไปทีละตัว
 - Go To Line Number ใช้เพื่อเลื่อนเคอร์เซอร์ไปที่บรรทัดที่ต้องการใน Editor Window
 - Find Error ใช้ในการค้นหาบรรทัดที่เกิดข้อผิดพลาด
 - Browse Symbol ใช้ในการแสดง Object หรือยูนิตต่างๆของโปรแกรม และค้นหา Object Symbol ภายในโปร เจ็ค

- เมนู view
 - Project Manager ใช้เพื่อเรียกหน้าต่างการ จัดการโปรเจ็คขึ้นมา เพื่อใช้ในการบริหารโปรเจ็ค เช่นการเพิ่ม สมาชิก, การถอดโปรแกรมออกจากสมาชิกในโปรเจ็ค
 - Project Source ใช้เพื่อเรียกดูซอร์สโปรแกรมของโปรเจ็คปัจจุบัน
 - Object Inspector ใช้เพื่อเรียกหน้าต่างของ Object Inspector เพื่อกำหนดค่า Property
 - Alignment Pallet ใช้เพื่อเรียกหน้าต่างของ Alignment เพื่อเรียงคอนโทรลบนฟอร์ม
 - Browser ใช้เพื่อเรียกหน้าต่างเพื่อดู Object และยูนิตในโปรเจ็คปัจจุบัน
 - Break Point ใช้เพื่อเรียกดู Break Point ที่ได้กำหนดไว้ในหน้าต่างที่แสดง Break Point เราสามารถเรียก
 Popup Menu เพื่อทำการแก้ไข/ลบ Break Point ได้
 - Call Stack ใช้เรียกดูรายการเรียกใช้ โพรซีเดอร์ในโปรเจ็คปัจจุบันในขณะที่กำลังรันโปรแกรม การเรียกดูนี้จะใช้ ในกรณีที่มีการเรียกโพรซีเดอร์ซ้อนกันเป็นจำนวนมาก
 - Watch ใช้ในการดูค่าตัวแปรหรือ Property ขณะที่หยุดโปรแกรมชั่วคราว เพื่อหาข้อผิดพลาดในโปรแกรม ซึ่งมี ลักษณะ
 - Threads เพื่อเรียกดูงานย่อยๆซึ่งกำลังทำงานอยู่ การเขียนโปรแกรมซึ่งมีการแยก Threads ซึ่งเป็นการทำงาน แบบขนานจะช่วยให้โปรแกรมทำงานได้รวดเร็วมากขึ้น
 - Component List ดูรายชื่อคอมโพเน้นท์ ซึ่งก็คือคอมโพเน้นท์ที่ได้ติดตั้งไว้บน Componnet Pallete นั่นเอง
 - Windows List ดูรายชื่อของหน้าต่างที่กำลังเปิดใช้อยู่ ซึ่งอาจเป็นฟอร์ม, ยูนิต, โปรเจ็ค, Watch, Breakpoint เป็นต้น
 - Toggle Form/Unit สลับหน้าต่างฟอร์มกับยูนิต
 - Units แสดงรายชื่อยูนิตที่อยู่ในโปรเจ็คให้เลือก
 - Forms แสดงรายชื่อฟอร์มที่อยู่ในโปรเจ็คให้เลือก
 - New Edit Window ใช้ในกรณีที่ต้องการเปิด Edit Window มากกว่า 1 Window
 - Speedbar แสดงหรือช่อน Speedbar
 - Component Pallete แสดงหรือช่อน Component Pallete
- เมนู Project
 - Add to Project ได้กล่าวมาแล้วในส่วนของ Speedbar
 - Remove from Project ได้กล่าวมาแล้วในส่วนของ Speedbar
 - Add to Repository ใช้ในการนำเอาโปรเจ็คปัจจุบันไปเก็บไว้เป็นต้นแบบใน Repository (เรียกใช้โดยเมนู new)
 - Compile แปลโปรแกรม การแปลจะแปลเฉพาะยูนิตที่มีการแก้ไข โดยการแปลยูนิตต่างๆออกเป็น .dcu (Delphi Compile Unit) จากนั้นจึงนำไปลิงค์
 - Build All คล้ายกับ Compile แต่จะแปลทุกยูนิต ใช้ในกรณีที่ไม่แน่ใจว่าวัน/เวลาของซอร์สโปรแกรมถูกต้อง
 - Syntax Check แปลเพื่อหาข้อผิดพลาดทางภาษา แต่จะไม่สร้าง dcu
 - Information ใช้ในการเรียกดูข้อมูลการแปล
 - Options ใช้สำหรับกำหนดรายละเอียดต่างๆของโปรเจ็ค โดยมีกลุ่มต่างๆดังนี้

6

Forms ใช้กำหนดฟอร์มที่จะเปิดขึ้นเป็นฟอร์มแรกเมื่อเรียกโปรแกรมมาทำงาน นอกจากนี้ยังใช้ กำหนดการสร้างฟอร์มโดยอัตโนมัติหรือไม่ และยังสามารถกำหนดลำดับของฟอร์มที่จะ สร้างขึ้นมา โดยอัตโนมัติ ในกรณีที่มีฟอร์มหลายๆฟอร์มในโปรเจ็ค การกำหนดให้สร้างโดยอัตโนมัติจะ ต้องนำ เอาฟอร์มที่ต้องการนไปไว้ในส่วนของ Auto Create Form เมื่อเราเปลี่ยนที่จุดนี้ เดลไฟล์จะเขียนคำสั่ง เพื่อทำให้เกิดการสร้างฟอร์มโดยอัตโนมัติ ในคำสั่งของ โปรแกรมในส่วนของโปรเจ็คต่อไปนี้จะเป็น การสร้างฟอร์ม Form1 และ Form2 โดยอัตโนมัติ

กรณี Auto Create

program project1; uses forms, unit1 in 'unit1.pas' {form1} unit2 in 'unit2.pas' {form2} {\$r *.res} begin application.createform(tform1,form1); application.createform(tform2,form2); application.run;

end.

กรณีที่ไม่ได้ใช้ Auto Create

ในกรณีนี้คำสั่ง Application.createform(tform2,form2); จะถูกตัดออก หากต้องการสร้างฟอร์มนี้ขึ้น มาขณะรันโปรแกรม ผู้เขียนโปรแกรมจะต้องสั่งสร้างฟอร์มขึ้นมาเอง

- Application ใช้สำหรับกำหนดรายละเอียดของ Application Object ได้แก่
 - 🔲 Title กำหนดชื่อของโปรแกรมซึ่งจะปรากฏอยู่ได้ icon เมื่อโปรแกรมถูก minimize
 - Help File ชื่อ Help File ที่จะถูกเรียกใช้เมื่อผู้ใช้กด F1
 - 🔲 Icon รูป Icon สัญลักษณ์ของโปรแกรม
- Compiler ใช้กำหนดรายละเอียดวิธีการแปลโปรแกรม
- Linker ใช้กำหนดรายละเอียดวิธีการลิงค์โปรแกรม
- Directory/Conditional ใช้กำหนดไดเร็คทอรี่ที่ใช้เก็บผลลัพธ์และการกำหนดการค้นหา object ในไลบ รารี่

• เมนู Run

- Run เรียกโปรแกรมมาทำงาน ก่อนเรียกมา ทำงานหากพบว่าต้องทำการแปลก็จะเรียกคอมไพเลอร์มาแปลใหม่
 จนได้ Executable File แล้วจึงเรียกมาทำงาน (Executable File จะมีชื่อตรงกับชื่อโปรเจ็ค)
- Parameters เป็นการกำหนดค่า Command Line Parameter ให้กับโปรแกรม โดยที่โปรแกรมสามารถรับค่า และตรวจสอบจำนวนค่าที่รับเข้ามาได้ โดยใช้ฟังก์ชัน Paramcount และ Paramstr()
- Step Over ได้อธิบายแล้วใน Speedbar
- Trace Into ได้อธิบายแล้วใน Speedbar

ไพฑูรย์ งิ้วทั่ง 2/12/022545

- Trace to Next Source Line ให้โปรแกรมทำงานต่อแล้วไปหยุดที่บรรทัดคำสั่งถัดไป โดยไม่สนใจ Flow ของการ
 Execute
- Run to Cursor ให้โปรแกรมทำงานจนกระทั่งพบตำแหน่งของเคอร์เซอร์จึงหยุด
- Show Execution Point แสดงจุดที่โปรแกรมกำลังหยุดทำงานอยู่
- Program Pause หยุดการทำงานของโปรแกรมชั่วคราว
- Program Reset ใช้ในกรณีที่ต้องการยกเลิกการทำงานของโปรแกรม
- Add Watch เพิ่มข้อมูลเกี่ยวกับตัวแปรที่ต้องการตรวจสอบค่าขณะที่ดีบักโปรแกรม
- Add Breakpoint เพิ่มจุดของคำสั่งที่ต้องการให้หยุดการทำงานชั่วคราว
- Evaluate/Modify ใช้แสดงค่าตัวแปร Object Property หรือนิพจน์ที่ต้องการตรวจสอบค่า นอกจากนี้ยัง สามารถเปลี่ยน แปลงค่าตัวแปรเพื่อให้โปรแกรมนำกลับไปใช้ได้ด้วย (การเปลี่ยนค่าทำได้เฉพาะกรณีที่เป็นตัว แปรเท่านั้น) ลักษณะจอภาพเป็น
- เมนู Component

7

- New ใช้เพื่อสร้างคอมโพเน้นท์ใหม่ โดยจะแสดงหน้าต่างเพื่อให้เลือกคอมโพเน้นท์ต้นแบบที่จะทำการ Inherit
- Install Component ใช้สำหรับติดตั้ง/ตัดออก คอมโพเน้นท์ต่างๆ ซึ่งจะมีผลตอการแสดงรายการบน
 Component Pallete โดยคอมโพเน้นท์ที่ติดตั้งเข้าไปใหม่จะแสดงตัวอยู่บน Pallete Page ที่กำหนดไว้ ถ้ายังไม่ มี page นั้นเดลไฟล์ก็จะสร้างให้
- Open Library ใช้กำหนดไลบรารี่ที่ต้องการ ใช้งาน โดยปกติจะเป็นไลบรารี่ชื่อ complib.dcl ซึ่งจะถูกเปิดโดย อัตโนมัติเมื่อเรียก โปรแกรมเดลไฟล์ อย่างไรก็ตามถ้ามีไลบรารี่อื่นที่ต้องการใช้แทนก็สามารถเปลี่ยนได้
- Rebuild Library ใช้สำหรับสร้างไลบรารี่ ขึ้นมาใหม่ ใช้ในกรณีที่ติดตั้งคอมโพเน้นท์ใหม่ไม่เรียบร้อย เพื่อทำให้ ไลบรารี่ถูกต้อง
- Configure Pallete ใช้ติดตั้ง/เปลี่ยนแปลง คอมโพเน้นท์ที่จะติดตั้งบน Pallete
- เมนู Database
 - Explore ใช้ในการเรียก Database Explorere
 - SQL Monitor ใช้ในการเรียก SQL Monitor เพื่อใช้ในการตรวจสอบการส่งหรือการรับข้อมูลกับ DBMS
 - Form Expert ใช้เพื่อเรียกผู้เชี่ยวชาญทำการสร้างฟอร์มให้
- เมนู Workgroup ใช้สำหรับการบริหารการเขียนโปรแกรมในสภาวะที่มีผู้เขียนโปรแกรมร่วมกันเป็นทีม
- เมนู Tools ยกเว้น Options, Repository และ Tools แล้วเมนูนี้อาจเปลี่ยนแปลงได้เนื่องจากผู้ใช้ สามารถกำหนดเมนู
 เพื่อเรียกโปรแกรมมาใช้งานเพิ่มเติมได้ เช่น Report Smith, Crystal Report, BDE Config เป็นต้น
 - Options(Environment Option) ใช้กำหนดรายละเอียดสภาพแวดล้อม IDE ประกอบด้วยกลุ่มต่างๆดังนี้
 - Preference ใช้ในการกำหนดสภาวะของ desktop, autosave, form designer, debuging, compiling, gallery
 - Library ใช้กำหนดไลบรารี่ในการค้นหา object เพื่อนำไปลิงค์
 - Editor Option ใช้กำหนดลักษณะของ Editor ที่ใช้งาน
 - Editor Display ใช้กำหนดลักษณะการแสดงผลของจอภาพ Editor รวมทั้งการใช้คีย์บอร์ด และการ กำหนดฟอนต์ (การใช้ภาษาไทยใน Editor จะต้องกำหนดฟอนต์เป็น Fixedsys)

ไพฑูรย์ งิ้วทั่ง 2/12/022545

- Editor Color ใช้ในการกำหนดสีในจอภาพ Editor ซึ่งมี Style ต่างๆ
- Browser ใช้กำหนดรายละเอียดการแสดงหน้าต่างของ Object Browser ซึ่งอยู่ในเมนู view/object browse หรือ search/browse symbol
- Repository ใช้ในการติดตั้ง/เปลี่ยนแปลงข้อมูลใน Repository
- Tools ใช้ในการติดตั้ง/เปลี่ยนแปลงเมนูของเครื่องมือ/โปรแกรมช่วยงาน
- Report Smith (Delphi 1.0) เรียก Report Smith ซึ่งเป็น Report Writer มาช่วยในการออกแบบรายงาน เมื่อ ออกแบบเสร็จก็จะเก็บ โครงสร้างรายงานไว้ รายงานนี้สามารถนำมาประกอบการเขียนโปรแกรมเดลไฟล์ได้ โดย นำเอา คอมโพเน้นท์ Treport ในกลุ่ม Data Access มาระบุชื่อรายงาน ที่สร้างไว้แล้วในโปรแกรมสามารถสั่งให้ พิมพ์รายงานนั้นออกมาได้ นอกจาก Report Smith แล้วผู้เขียนโปรแกรมอาจเลือก Report Writer ตัวอื่นมาทำ รายงานก็ได้ซึ่งวิธีการ เรียกมาใช้ในโปรแกรมก็จะใกล้เคียงกัน ตัวอย่าง Report Writer ที่สำคัญๆ นอกเหนือจาก Report Smith แล้วก็มี Crystal Report, r & r เป็นต้น
- Image Editor ใช้เรียกโปรแกรมที่ช่วยในการทำ/แก้ไข Image ซึ่งสามารถน้ำ Image ไปใช้ใน Toolbar หรือ ปุ่ม กด
- Database Desktop ใช้เรียกโปรแกรม Database Desktop มาจัดการข้อมูล ซึ่งได้อธิบายมาแล้วในบทก่อน
- BDE Config ใช้เรียกโปรแกรม Borlan Database Engine Config ซึ่งได้อธิบายมาแล้วในบทก่อน

ตัวแปรและประเภทตัวแปรแบบต่างๆ

(variables and types)

object pascal แบ่งประเภทของตัวแปรออกเป็นหลายประเภท แต่ประเภทที่เราจะศึกษาและมีการใช้งานกันทั่วไป ๆ ได้แก่

1. simple types ตัวแปรประเภทที่เก็บข้อมูลที่เป็นตัวเลขทั้งจำนวนเต็มและที่มีทศนิยม ได้แก่ integer types, charater types และ real types

- 2. string types ตัวแปรที่เก็บข้อความแบ่งออกเป็น short string และ long string
- 3. structured types ตัวแปรที่เก็บเป็นโครงสร้างข้อมูลเช่น record class เป็นต้น

4. variant types ตัวแปรที่สามารถเก็บค่าประเภทใดก็ได้ โดยที่ขณะที่สร้างขึ้นยังไม่ทราบประเภทแน่นอน และจะทราบ ประเภทเมื่อมีการกำหนดค่าให้ตัวแปรแล้ว

้นอกจากนี้ยังมีตัวแปรประเภท pointer และ procedure เนื่องจากเป็นเรื่องที่ ค่อนข้างซับซ้อนจึงไม่ขอกล่าวในที่นี้

• *ตัวแปรประเภท simple* แบ่งออกเป็น

- ordinal type ซึ่งแบ่งย่อยออกเป็น
 - integertype คือประเภทตัวแปรจำนวนเต็มซึ่งประเภทที่ไม่ขึ้นกับ cpu และ os (fundamental) ได้แก่ shortint : signed 8-bit เก็บค่าได้ตั้งแต่ -128 ถึง 127 smallint : signed 16-bit เก็บค่าได้ตั้งแต่ -32768 ถึง 32767
 - longint : signed 32-bit เก็บค่าได้ตั้งแต่ -2147483648 ถึง 2147483647
 - byte : unsigned 8-bit เก็บค่าได้ตั้งแต่ 0 ถึง 255
 - word : unsigned 8-bit เก็บค่าได้ตั้งแต 0 ถึง 65535
 - ประเภทที่ขึ้นกับ cpu และ os (generic) คือ

integer 16, 32 bit

cardinal 16,32 bit

character types ได้แก่ ansichar, wide char ซึ่งเป็น fundamental type และเก็บตัวอักษรขนาด 1
 byte และ 1 word ตามลำดับ ส่วนที่เป็น generic type คือ char ปกติจะเก็บ 1 byte เช่นเดียวกับ
 ansichar

ประเภท	ด่าที่เก็บ	ความแม่นยำ	ขนาดหน่วยความจำ	
		(หลัก)	(bytes)	
real	2.9x10 ⁻³⁹ ถึง 1.7x10 ³⁸	11-12	6	
single	1.5x10 ⁻⁴⁵ ถึง 3.4x10 ³⁸	7-8	4	
double	5.0x10 ⁻³²⁴ ถึง 1.7x10 ³⁰⁸	15-16	8	
extended	3.4x10 ⁻⁴⁹³² ถึง 1.1x10 ⁴⁹³²	19-20	10	
	-922337203685477.5807 ถึง	10.20	0	
currency	922337203685477.5807	19-20	2	

• real type เก็บค่าตัวเลขที่มีทศนิยมแบ่งออกเป็น 5 ประเภทคือ

10

- *ตัวแปรประเภท string* แบ่งออกเป็น 2 ประเภท
 - short string เก็บตัวแปรประเภทข้อความที่มีความยาวไม่เกิน 255 ตัวอักษร โดยที่การประกาศตัวแปรจะต้อง ระบุขนาดด้วย เช่น name: string [25];
 - long string เก็บตัวแปรข้อความที่มีความยาวมากถึง 2 gb โดยการประกาศตัวแปรไม่ต้องระบุขนาดเช่น name
 : string:

สำหรับตัวแปรประเภท structured type และ variant type จะได้กล่าวถึงรายละเอียด ในบทต่อๆ ไป เนื่องจากเดลไฟล์มีรากฐานจาก object pascal ซึ่งเป็นภาษาที่เป็นแบบคอมไพเลอร์ (ต่างจาก intepreter เช่น visual basic) ดังนั้น

การตรวจสอบประเภทข้อมูลจึงเป็นสิ่งสำคัญ ซึ่งจะต้องตรงกันในกรณีที่มีการกำหนดค่าให้กับตัวแปร เช่น

label1.caption := inttostr(strtoint(edit1.text) + strtoint(edit2.text));

เนื่องจากประเภทข้อมูลของ caption และ text ซึ่งเป็นพร๊อบเพอร์ตี้ของ lable และ edit ตามลำดับนั้นเป็นแบบ string ดังนั้นการที่ จะนำไปคำนวณจะต้องเปลี่ยนประเภทให้เป็น ประเภทที่คำนวณได้(เช่น integer)เสียก่อน

ในขณะที่ถ้าต้องการกำหนดค่าให้กับมันจะต้องแปลงกลับมาเป็น string โดยใช้ฟังก์ชั่นที่ช่วยในการแปลงประเภทข้อมูลซึ่งมีอยู่มาก มายโดยที่ในตัวอย่างจะใช้ strtoint และ inttostr

โดยทั่วไปการใช้ตัวแปรจะต้องประกาศเสียก่อนโดยใช้คำสั่ง var ดังนี้

ver v.v.: lengint:	var x : longint;
var x,y . ionginit,	y : longint;
	begin
x y + 3,	x := y + 3;
ena,	end;

ในตัวอย่างจะประกาศตัวแปรสองตัวคือ x และ y เป็นตัวแปรประเภท longint ซึ่งแสดงวิธีประกาศพร้อมกันบรรทัดเดียวและการ ประกาศแยก

ตัวแปรประเภท integer type & character type

ฟังก์ชั่น chr และ ord

ประเภทข้อมูลแบบcharจะใช้มากในกรณีที่ต้องการตรวจสอบการคีย์ที่แป้นพิมพ์ โดยทั่วไปประเภทข้อมูลแบบ char จะมีความ สัมพันธ์โดยตรงกับประเภทข้อมูลแบบ byte ซึ่งเป็น unsigned integer 8-bit โดยที่ byte จะใช้ในการระบุตำแหน่งของตัวอักษรใน ตารางเช่น

```
var
ch : char;
x : byte;
begin
x := 65;
ch= char(x); { ch มีค่าเท่ากับ 'a'}
x := x + 2;
```

```
11 ใพฑูรย์ งิ้วทั้ง 2/12/022545
ch := chr(x); { ch มีค่าเท่ากับ 'c'}
end:
```

จากตัวอย่าง ตำแหน่งของตัวอักษร a จะอยู่ที่ลำดับ 65 ในตาราง ascii การเปลี่ยนตำแหน่ง เป็นตัวอักษรจะใช้ฟังก์ชั่น chr ในทางกลับกันถ้าต้องการทราบตำแหน่งของตัวอักษรใดก็ให้ใช้ฟังก์ชั่น ord (ย่อมาจาก ordinal) ดังตัวอย่าง

```
var
ch : char;
x : byte;
begin
ch := 'c',
x : = ord(ch); {x = 67}
ch : = chr(x+2); {ch = 'e'}
```

```
end;
```

้นอกจากใช้ฟังก์ชั้น chr เพื่อเปลี่ยนตำแหน่งเป็นตัวอักษรแล้ว เรายังสามารถใช้ เครื่องหมาย '#' แล้วตามด้วยตัวเลขตำแหน่ง เพื่อ ทำงานเช่นเดียวกับ chr ได้ดังนี้

```
var
ch : char;
begin
ch := #65; {มีค่าเท่ากับ ch := chr(65); หรือ ch := 'a';}
end;
```

้ ตัวอย่าง ถ้าต้องการตรวจสอบคีย์ที่กดผ่านแป้นพิมพ์ว่าเป็น enter หรือไม่ ทำได้ดังนี้

```
procedure tform1. formkeypress(sender : tobject; key : char);
begin
if key = #13 then {enter was pressed}
begin
key := #0;
```

postmessage(handlle, wm_nextdlgctl, 0,0); {send tab}

```
end;
```

หมายเหตุ ในตัวอย่างนี้ต้องกำหนดพร๊อบเพอร์ตี้ keypreview ของฟอร์มเป็น true เพื่อดักจับคีย์บอร์ดทุกตัวแม้ว่าจะกดที่คอม โพเน้นท์ใดให้มาผ่านที่ฟอร์มก่อน การกำหนดอาจกำหนดโดยใช้ object inspector หรือเขียนคำสั่งไว้ใน formcreate ดังนี้ form1.keypreview := true;

real types

การเก็บข้อมูลประเภท real type จะเก็บโดยใช้รูปแบบของ floating point การเปลี่ยนประเภทข้อมูลจาก float ไปเป็น string สามารถทำได้หลายวิธีดังนี้

• <u>ฟังก์ชั้น floattostr(<ค่า floating point>)</u> เช่น

floattostr(r); {r:real, r:=2}

<u>ฟังก์ชั่น format(<format string>,[array ของค่าตัวแปร])</u> เช่น

t := format('value is :%f', [r]);

โดยที่

r เป็น array ของค่าที่ต้องการระบุให้กับฟังก์ชั่น format เพื่อจัดการโดย format string การระบุ array จะใช้เครื่องหมาย "[" และ "]" ล้อมรอบสมาชิกทั้งหมด ส่วนสมาชิกแต่ละตัวจะคั่นด้วยเครื่องหมาย "," เช่น [r,4,p,q] มีสมาชิก 4 ตัว <format string> มีสัญญลักษณ์พิเศษ ที่จะนำหน้ารูปแบบตัวอักษรที่สำคัญมีดังนี้

%f	แสดงค่า floating point	
%d	แสดงผลลัพธ์ของค่า integer	
%m	แสดงผลลัพธ์ที่เป็นจำนวนเงิน ซึ่งมี type เป็นแบบ currency ซึ่งกำหนดรูปแบบจาก window control panel	
%n	แสดงค่า floating point คล้ายกับ %f แต่จะมี ',' ที่หลักพัน และ หลักล้านได้	
%s	แสดงค่าของ string	

ระหว่างเครื่องหมาย '%' กับรูปแบบ(f,d,m,n,s) นั้นเราสามารถระบุ ความกว้างของผลลัพธ์และจุดทศนิยมได้ เช่น '%6.2f' หมายถึง ความกว้างของผลลัพธ์ทั้งหมด 6 ตัวอักษรและมีจุดทศนิยม 2 จุด

ตัวอย่างโปรแกรม

- 1. ใช้เมนู file/new application เพื่อสร้างโปรเจ็คใหม่
- 2. นำคอนโทรล listbox และ button วางบนฟอร์มโดยให้ชื่อ listbox1 และ button1
- 3. เขียนคำสั่งที่ onclick ของ button1 ดังนี้

procedure tform1.button1click (sender : tobject);

var r : real;

t : string;

begin

listbox1.clear;

r := 1234567.3456;

listbox1.items.add(floattostr(r));

t := format ('%f', [r]);

listbox1.items.add(t);

t := format ('%n', [r]);

listbox1.items.add(t);

t := format ('%12.3f', [r]);

listbox1.items.add(t);

end;

4. ทดลองรันโปรแกรม

โดยที่

x เป็นค่า real หรือ integer

width ระบะความกว้างของผลลัพธ์ ส่วน decimal ระบะจำนวนทศนิยม

s เป็น string ที่เก็บผลลัพธ์

ตัวอย่าง

function tform1.real2str (n : real; width, place : integer): string;

var

tempstr : string; begin str(n : width : place, tempstr); real2str := tempstr; end;

้สังเกตว่า real2str เป็นฟังก์ชั่นที่ส่งค่ากลับเป็น string ดังนั้นจึงเรียกใช้ได้ดังตัวอย่าง

listbox1.items.add(real2str(5.762,10,2));

edit1.text := real2str (5.762,10,2);

ในตอนท้ายของฟังก์ชั่น real2str จะมีการกำหนดค่าให้กับชื่อฟังก์ชั่นเพื่อส่งกลับไป

string types

้ถ้าพิจารณา string ข้อความให้ละเอียดแล้วจะพบว่า string ก็คือ array (แถวลำดับ) ของ character นั่นเอง

type

tmystring = array [0..255] of char;

var

mystring : tmytring;

thestring : string;

จากตัวอย่าง ได้กำหนดประเภทข้อมูลชื่อ tmysting ขึ้น ซึ่งเป็น array ของตัวอักษรที่มีความยาว 256 ตัวอักษรโดยคำสั่งที่อยู่ใน type คือ tmystring = array [0.255] of char; ส่วนตัวแปร thestring จะมีลักษณะคล้ายกับ mystring ความแตกต่างจะอยู่ที่ string ได้ถูกกำหนดเป็นประเภทไว้ก่อนแล้ว และจะมีความยาวไม่เกิน 255 ตัวอักษร

การจัดการข้อมูลใน string สามารถอ้างข้อมูลทั้งหมดโดยใช้ชื่อ เช่น

listbox1.items.add(thestring);

หรืออ้างตัวอักษรตัวใดที่อยู่ใน string ในลักษณะเดียวกับการอ้างสมาชิกใน array ดังนี้

listbox1.items.add(thestring[2]);

ตัวอักษรตัวแรกจะเป็นสมาชิกหมายเลข 1 และตัวสุดท้ายจะเป็น n โดยที่ n เป็นความยาวของ string

ฟังก์ชั่นที่ใช้ในการเปลี่ยนประเภทข้อมูลของ string ได้แก่

- inttostr : เปลี่ยน integer เป็น string
- strtoint : เปลี่ยน string เป็น integer
- floattostr : เปลี่ยน floating point เป็น string
- strtofloat : เปลี่ยน string เป็น floating point
- val : เปลี่ยน string เป็นตัวเลข
- str : เปลี่ยนตัวเลขเป็น string

ฟังก์ชั่นที่ใช้ในการจัดการข้อมูลใน string

- length : ความยาวของ string
- uppercase : เปลี่ยนเป็นตัวอักษรใหญ่
- Iowercase : เปลี่ยนเป็นตัวอักษรเล็ก
- trim : ตัดตัวอักษรที่เป็น blak ด้านซ้ายและขวา
- trimleft : ตัดตัวอักษรที่เป็น blak ด้านซ้าย
- trimright : ตัดตัวอักษรที่เป็น blak ด้านขวา
- pos : หาตำแหน่งของข้อความใน string
- concat : เชื่อม string 2 sting
- insert : แทรก string ด้วย string
- copy : copy substing ออกจาก string
- delete : ลบ substing ใน string

คำสั่งที่ใช้ในการควบคุมลำดับการทำงาน

(control flow)

คำสั่งที่ใช้ในการควบคุมลำดับการทำงานแบ่งออกเป็น 2 ประเภท ประเภทแรกเป็นคำสั่งที่ใช้ในการตัดสินใจ

้ ได้แก่ คำสั่ง if และคำสั่ง case ส่วนประเภทที่สองเป็นคำสั่งที่เป็นคำสั่งที่เป็นวงวน(loop) ได้แก่ คำสั่ง while, คำสั่ง repeat และคำ สั่ง for

คำสั่ง if

รูปแบบ

if <นิพจน์> then <คำสั่ง>	
[else <คำสั่ง>]	

โดยที่

้นิพจน์ : เป็นเงื่อนไขในการตรวจสอบ ถ้าผลลัพธ์เป็น true ก็จะทำคำสั่งหลัง then ถ้าเป็น false ก็จะทำคำสั่งหลัง else

คำสั่ง : อาจจะเป็นคำสั่งเดียวหรือหลายคำสั่ง ถ้าเป็นหลายคำสั่งต้องอยู่ภายใต้ block ของ begin..end เช่น

if a = 1 then b := 2	
else begin	
c := 3;	
d := 4;	
end;	

โปรดสังเกตคำสั่งหลัง then เป็นคำสั่งเดียวส่วน คำสั่งหลัง else เป็นคำสั่งหลายๆ คำสั่งซึ่งจะอยู่ภายใต้ begin...end และคำสั่งที่ อยู่ก่อน else จะไม่มีเครื่องหมายจบคำสั่งหรือ ';' ในกรณีที่ไม่มีส่วน else คำสั่งหลัง then สุดท้ายต้องมีเครื่องหมาย ';' เช่น if a = 1 then b:= 2;

ตัวอย่าง กรณีที่คำสั่งหลัง then มีหลายคำสั่ง

if a = 1	then		
begin			
b := 2;			
x := 3;			
end			
else			
begin			
c := 3;			
d := 4;			
end;			

เครื่องหมายที่ใช้ในการเปรียบเทียบ(relational operator) มีดังนี้

=	เท่ากับ
<>	ไม่เท่ากับ
<	น้อยกว่า
>	มากกว่า
>=	มากกว่าหรือเท่ากับ
<=	น้อยกว่าหรือเท่ากับ

ไพฑูรย์ จิ้วทั้ง 2/12/022545

เครื่องหมายที่ใช้ในการเชื่อมนิพจน์เปรียบเทียบ(logical operator) ได้แก่

not	ใม่
and	และ
or	หรือ

ตัวอย่าง การใช้งาน operator ทั้งสองประเภท

if (a = b) and (not tabel1.eof) then

{same group of record}

หมายถึง ค่า a เท่ากับ b และ ยังไม่สิ้นสุด(eof)แฟ้มข้อมูล table1

้ลำดับการคำนวณของเครื่องหมายที่ใช้ในการคำนวณ (operator precedence)

ชนิด	ลำดับ	หมายเหตุ
not, @	1	@ คือ pointer
*, /, div, mod	2	div=หารจำนวนเต็ม mod=หารเพื่อหาเศษ
and, +, -, or	3	
=, <>, <, >, <=, >=	4	

แลลัพธ์ที่ได้จากการประเมินค่านิพจน์ในคำสั่ง if จะมีค่าเป็นแบบ boolean โดยที่เดลไฟล์ จะใช้ค่าคงที่สำหรับข้อมูลประเภทนี้คือ true และ false สามารถนำไปใช้ได้ เช่น

var found : boolean;
begin
found := false;
while not table1.eof do
begin
if (condition is meet) then
begin
found := true;
table1.next;
end;
end;
end;

คำสั่ง case รูปแบบ

```
case <นิพจน์> of
<ค่าคงที่ 1> : <คำสั่ง >;
[<ค่าคงที่ 2> : <คำสั่ง>;]
:
[else <คำสั่ง>;]
end;
```

ตัวอย่าง

case myselector of

5 : listbox1.add('special case');

1...4,b..10 : listbox1.add('general case');

else listbox1.add('missing case');

end;

ในตัวอย่างนี้ จะประเมินค่าของ myselector ซึ่งถ้ามีค่าเท่ากับ 5 ก็จะให้ค่า 'special case' ถ้ามีค่า 1 ถึง 10 ก็จะไร้ค่า 'general case' ถ้าไม่อยู่ในช่วงดังกล่าวจะมีค่าเป็น 'missing case'

ปกติคำสั่ง case จะนิยมใช้ในกรณีที่มีเงื่อนไขเดียวแต่มีทางเลือกจำนวนมาก เพราะจะทำให้คำสั่งที่เขียนง่ายกว่าการใช้คำสั่ง if ซึ่ง ต้องนำมาซ้อนกันหลาย ๆ ชั้น ในกรณีที่ใช้คำสั่ง case และเพื่อให้สามารถดูแลโปรแกรมได้ง่ายจึงควรจะนำเอาประเภทข้อมูลแบบ enumerated มาใช้

ตัวอย่างการใช้ enumerated

สมมติว่ามีประเภทของรูปร่างให้เลือกดังนี้

- 1. สี่เหลี่ยมผืนผ้า
- 2. สี่เหลี่ยมจตุรัส
- 3. สี่เหลี่ยมผืนผ้าขอบมน
- 4. สี่เหลี่ยมจตุรัสขอบมน
- 5. วงกลม
- 6. วงรี

ถ้าเรากำหนดค่าคงที่ที่เป็นตัวเลขตามลำดับข้างต้น คำสั่ง case จะมีลักษณะดังนี้

case shape of

- 1:
- 2 :
- 3:
- 4:
- 5:
- 6:

end;

```
18 ไพฑูรย์ งิ้วทั่ง 2/12/022545
หากเราใช้ตัวแปรประเภท enumerated เข้าช่วยคำสั่งจะมีลักษณะดังนี้
type
tshapetype = (rectangle, sguare, roundrect,
roundsquare, ellipse, circle);
var shape1 : tshapetype;
begin
case shape1 of
rectangle : <statements>;
square : <statements>;
roundrect : <statements>;
ellipse : <statements>;
circle : <statements>;
```

end;

end;

การตรวจสอบโปรแกรมจะสะดวกมากกว่า สำหรับคอมไพเลอร์แล้ว การประกาศตัวแปรแบบ enumerated (ประกาศเป็น list โดยมี เครื่องหมายวงเล็บล้อมรอบ) จะทำให้คอมไพเลอร์นำเอาค่าคงที่ตั้งแต่ 0,1,2..ไปแทนในตำแหน่งใน list แล้วเก็บตารางของค่าไว้ เมื่อพบชื่อที่ตรงกันก็จะนำค่าตัวเลขมาแทนให้โดยอัตโนมัติ โดยที่โปรแกรมเมอร์ไม่ต้องสนใจในส่วนนั้น เพียงแต่การเรียกใช้ต้อง อ้างชื่อให้ตรงกันที่อยู่ใน list เท่านั้น

คำสั่ง for

รูปแบบ

for <control-var> := <init> [to | downto] <final> do

<statement>;

<control-var> คือ ตัวแปรที่เก็บค่าตัวเลขได้แก่ integer type, real type หรือ character type

<init> ค่าเริ่มต้นของตัวแปร ก่อนจะเข้าในวงวน (loop)

<final> ค่าสุดท้ายที่จะจบการทำงานออกจากวงวน

้นอกจากค่า <final> ที่จะทำให้จบการทำงานแล้วในวงวนของ for, while และ report เรายังสามารถใช้ break; เพื่อจบการทำวงวน ได้ และอาจใช้ continue; เพื่อข้ามคำสั่ง

ส่วนต่อไปของวงวนเพื่อเริ่มต้นการทำงานในรอบถัดไปของวงวน

้สำหรับกรณีที่ต้องการจบ procedure นั้นก็ให้ใช้คำสั่ง exit, และถ้าจบโปรแกรมก็สามารถใช้ halt; หรือ application.terminate;

ตัวอย่าง

for i := 1 to 25 do

listbox1.items.add(inttostr(i));

เป็นการเพิ่มรายการใน listbox1 25 รายการ ค่าของรายการ เริ่มจาก 1 ถึง 25

k := 1; for i := 1 to j do begin k := k * 2; listbox1.items.add(inttostr(k)); end; เป็นการหาค่าของ 2 ยกกำลัง j โดยค่าที่ได้ในแต่ละรอบของการคำนวณ จะนำไปใส่ไว้ใน listbox1 ในตัวอย่างนี้แสดงให้เห็นถึง block ของคำลั่ง for ซึ่งอยู่ระหว่างคำสั่ง begin....end

คำสั่ง while

รูปแบบ

```
while <นิพจน์> do
<คำสั่ง>
```

นิพจน์ คือ ค่าที่จะประเมินและตัดสินใจในการทำวงวน ถ้าผลลัพธ์เป็น true ก็จะทำคำสั่งในวงวน ถ้าเป็น false ก็จะจบคำสั่งในวง วน

ตัวอย่าง

```
while (s = 5) and (not table1.eof) do
```

begin

```
tot := tot + table1salary.asfloat;
```

table1.next;

end;

การทำวงวนนี้จะทำงานเมื่อ s มีค่าเท่ากับ 5 และ ยังไม่หมดข้อมูลในแฟ้ม table1 ขณะทำงานในวงวนก็จะทำการบวกสะสม ยอดในฟิลด์จาก table1 และการอ่านรายการถัดไปของ table1

i := 0;

while i < 10 do

begin

listbox1.items.add (inttostr(i));

i := i + 1;

end;

น้ำค่า 0-9 ไปใส่ใน listbox1 ทั้งหมด 10 รายการ

คำสั่ง repeat
รูปแบบ repeat
<คำสั่ง>
until <นิพจน์>;

นิพจน์ คือ เงื่อนไขซึ่งจะใช้ในการตัดสินใจว่าจะทำวงวนต่อหรือไม่ ซึ่งจะตรงข้ามกับคำสั่ง while โดยที่ถ้านิพจน์มีค่าเป็น true ก็จะ หยุดการทำวงวน แต่ถ้าเป็น false ก็จะทำวงวนต่อไป

ตัวอย่าง

repeat

write('enter value: ');

readln(i);

until (i >=0) and (i <=9);

ทำการอ่านข้อมูลโดยใช้ standard i/o(screen) จนกระทั่ง i มีค่า 0-9 จึงจะจบการทำงาน (ต้องกำหนด project/option generate dos console)

<u>โปรแกรมจัดการฐานข้อมูล และ SQL</u>

โปรแกรมจัดการฐานข้อมูล

โปรแกรมจัดการฐานข้อมูล (Database Management Systems) หรือเรียกย่อๆ ว่า DBMS นั้นเป็นซอฟต์แวร์ที่ช่วยในการจัด เก็บ, บริหาร, และเรียกใช้ข้อมูลอย่างมีประสิทธิภาพ โดย DBMS จะเป็นตัวกลางระหว่างผู้ใช้งานกับ ข้อมูล

ระบบการเก็บข้อมูลปัจจุบัน อาจแบ่งออกได้เป็น 2 ประเภทคือ

ระบบข้อมูลธรรมดาที่ใช้โดยผู้ใช้คนเดียวทั่วๆไป (Desktop Database) ได้แก่

ระบบข้อมูลแฟ้ม DBF	ระบบแฟ้มข้อมูลของโปรแกรมดีเบส,ฟ๊อกโปร
ระบบแฟ้ม BTrieve	ระบบแฟ้มข้อมูลแบบ Indexed Sequential
Paradox	แฟ้มข้อมูลคล้ายกับ DBF แต่เป็นของบริษัทบอร์แลนด์
Access	แฟ้มข้อมูลใช้งานบนระบบปฏิบัติการวินโดว์ของบริษัท
100000	ไมโครซอฟต์

 ระบบฐานข้อมูลที่ใช้ร่วมกันหลายๆผู้ใช้ (Server Database) เป็นระบบฐานข้อมูลเชิงสัมพันธ์ที่แท้จริงที่สามารถนำไปใช้ ในลักษณะ Client/Server ได้ ระบบฐานข้อมูลประเภทนี้ ได้แก่

Oracle	ปัจจุบันพัฒนาถึงเวอร์ชั่น 8 เป็นของบริษัทออเรเคิล	
Informix	ปัจจุบันพัฒนาถึงเวอร์ชั่น 9 เช่นเดียวกัน เป็นของบริษัทอินฟอร์มิกซ์	
Sybase	เป็นของบริษัทพาวเวอร์ซอฟต์	
MS SQL	ป็นโปรแกรมของไมโครซอฟต์ที่เดิมใช้เทคโนโลยีของ Sybase ในการพัฒนา	
Server	ปัจจุบันไมโครซอฟต์ได้แยกการ พัฒนาออกจาก Sybase แล้ว	
Ingres	เป็นของบริษัทคอมพิวเตอร์แอสโซซิเอท หรือ CA	
InterBase	เป็นดาต้าเบสของบริษัทบอร์แลนด์	

21 ไพตูรย์ งิ้วทั่ง 2/12/022545

บริษัทบอร์แลนด์ ได้สร้างมาตรฐานการเชื่อมต่อฐานข้อมูลทั้งสองแบบข้างต้น โดยผ่านโครงสร้างที่ชื่อ IDAPI (Integrated Database API) เพื่อให้ระบบงานที่สร้างขึ้น สามารถเรียกใช้ข้อมูลภายในฐานข้อมูล ผ่านระบบการเชื่อมต่อแบบนี้ ทั้งนี้ไม่ต้องเรียก ใช้ฟังก์ชั่นของฐานข้อมูลโดยตรง ทำให้ผู้พัฒนาระบบสามารถโยกย้าย ระบบงานไปใช้กับฐานข้อมูลต่างๆ ได้หลายชนิดและมีความ สะดวกในการโยกย้าย ส่วนบริษัทไมโครซอฟต์ได้สร้างโครงสร้างเชื่อมต่อที่เรียกว่า ODBC (Open DataBase Connectivity) เพื่อ ให้ระบบงานที่สร้างขึ้นเชื่อมต่อกับฐานข้อมูลคล้ายกับ IDAPI แต่โครงสร้างชนิดนี้เป็นโครงสร้างที่ผู้พัฒนาระบบนิยมใช้กันมาก และ

ยอมรับเป็นมาตรฐานการ เชื่อมต่อ อย่างไรก็ตามบริษัทบอร์แลนด์ก็ได้สนับสนุนการใช้ ODBC ผ่านโครงสร้าง IDAPI ด้วย การกำหนดการเรียกใช้ฐานข้อมูลของเดลไฟล์ต้องใช้โปรแกรม BDE (Borland Database Engine) รูปต่อไปนี้แสดงกับโครงสร้าง ของการ เชื่อมต่อกับฐานข้อมูลของบริษัทบอร์แลนด์



การกำหนดฐานข้อมูลที่โปรแกรมต้องการเชื่อมต่อ จะใช้โปรแกรม BDE Configuration ซึ่ง โปรแกรม BDE Configuration จะช่วย ในการระบุฐานข้อมูลที่จะใช้งาน โดยการตั้งชื่อ Alias พร้อมทั้งระบุรายละเอียดของการต่อเชื่อม หลังจากกำหนดเรียบร้อยแล้ว Alias สามารถไปอ้างอิงได้ในโปรแกรม *(ในใช้งานในที่นี้จะติดต่อฐานข้อมูล โดยใช้ ODBC แทนการใช้ BDE)*

<u>ภาษา SQL(Structured Query Language)</u>

ฐาน ข้อมูลเชิงสัมพันธ์ทุกยี่ห้อ สามารถรองรับการเรียกใช้ฐานข้อมูลโดยผ่านภาษากลางที่เป็นมาตรฐานเดียวกัน คือ

SQL(Structured Query Languaqe) ในการบริหารจัดการและปรับปรุงข้อมูลในฐานข้อมูล ทั้งนี้เพื่อให้โปรแกรมที่เขียนขึ้นมา สามารถย้ายไปทำงานกับระบบฐานข้อมูลอีกยี่ห้อหนึ่งได้โดยไม่ต้องทิ้งโปรแกรมชุดเดิมแล้วเขียนโปรแกมรขึ้นมาใหม่นอกจากนี้ยัง ช่วยลอเวลาการเรียนรู้กำสั่งในการจัดการข้อมูลของแต่ละยี่ห้อ

้คำสั่ง ใน SQL ที่มีอยู่อาจแบ่งออกเป็นกลุ่มหลัก ๆ ได้ดังนี้

- คำสังที่ในในการจัดการกับโครงสร้างข้อมูล (DD/D) หรือ Meta data เรียกว่า Data Definition Language (DDL) เป็นคำสั่ง ที่ใช้สำหรับ การสร้าง / แก้ไข / ลบตาราง ข้อมูลซึ่งมีผลทำให้โครงสร้างข้อมูลในตารางเปลี่ยนแปลงไปการทำงานแก้ไข ข้อมูลส่วนของ Meta-data นอกจากนี้ยังรวมถึงคำสั่งที่ใช้ในการสร้าง / แก้ไข / ลบ View, คัชนี, Constrain, Trigger, Stored Procedure และ คำสั่งที่ใช้ในการกำหนดสิทธิ์ ให้ผู้ใช้งาน คำสั่งในกลุ่มนี้ได้แก่ คำสั่ง CREATE, ALTER, DROP ต่างๆ เช่น CREATE TABLE, CREATE INDEX, CREATE TRIGGER คำสั่งที่ใช้ในการกำหนดสิทธิ์ได้แก่ GRANT, REVOKE
- คำสั่งที่ใช้ในการเรียกใช้, เพิ่มข้อมูล, แก้ไขข้อมูล, ลบข้อมูลในตาราง เรียกว่า Data Manipulation Language (DML)
 เป็นคำสั่งที่ใช้ในการจัดการข้อมูลที่อยู่ในตารางเพื่อให้ข้อมูลในตารางทันสมัย (Updated) ตรงกับความเป็นจริง คำสั่งใน กลุ่มนี้ได้แก่ SELECT, INSERT, UPDATE, DELETE และ COMMITE ตามลำดับ

โครงสร้างคำสั่ง Select

 Select [Distinct] / * / รายชื่อฟิลด์

 From ชื่อตาราง1, ชื่อตาราง2,...

 [Where เงื่อนไข]
 //เงื่อนไข where

 [Order By]
 //การเรียงลำดับจากน้อยไปมา

 [Group By]
 //การจัดกลุ่ม

 [Having]
 //เงื่อนไข having

 [Union]
 // คำสั่ง Select ที่ช้อนเข้ามา

<u>คำสั่ง SQL กับเดลไฟ</u>

การสร้างชุดคำสั่ง SQL ในขณะ Design Time

้ ก่อนที่จะใช้คำสั่ง SQL ในเดลไฟนั้นให้คลิก Query ในแท็บของ Data Access ก่อนแล้วทำตามขั้นตอนต่อไปนี้

- 1. ที่คุณสมบัติ DataBaseName ให้เลือกชื่อ ODBC ที่เรากำหนดไว้ในขั้นตอนการเชื่อมฐานข้อมูลด้วย ODBC
- ที่คุณสมบันติด SQL ให้ดับเบิ้ลคลิกก็จะเห็นไดอเล็อกสำหรับป้อนคำสั่ง SQL ให้ป้อนคำสั่ง Select From Where หรืออีกวิธี หนึ่งจะใช้ คลิกขวาที่ Query แล้วเลือก SQL Bullder ก็ได้ เป็นเครื่องมือในการสร้างภาษา SQL ให้ แล้วก็เลือก Table และ flelds ที่ต้องการ จากนั้นก็ Save
- 3. คุณสมบัติ Active ให้ดับเบิ้ลคลิก เป็น True ถ้าไม่เปลี่ยนเป็น Trueให้ให้ตรวจสอบขั้นตอนที่ผ่านมาถูกหรือไม่
- 4. ตั้งชื่อ qNameให้สอดคล้อง และกำหนด DataSource ให้สอดคล้องกับ Query ด้วย

การสร้างชุดคำสั่ง SQL ในขณะ Run Time

1. ปิดการทำงานของชิ้นส่วนโปรแกรม แบบ Query ชุดที่จะบรรจุคำสั่งใหม่ลงไปด้วยคำสั่ง Close

ไพฑูรย์ งิ้วทั่ง 2/12/022545

- 2. ทำการลบชุดคำสั่ง SQL เดิม ออกโดยใช้คำสั่ง Clear
- 3. บรรจุคำสั่ง SQL ใหม่ ลงในคุณสมบัติ SQL ด้วยคำสั่ง Add
- 4. ใช้คำสั่ง Open หรือ ExecSQL กระตุ้นให้ Query ที่ดำเนินการอยู่ทำงาน

ตัวอย่าง

Query1.Close; Query1.SQL.Clear; Query1.SQL.Add(' Select * '); Query1.SQL.Add(' From Employee '); Query1.SQL.Add(' Where monny >= 500 '); Query1.Open;

การส่งผ่านพารามิเตอร์ในคำสั่ง SQL

การเขียนโปรแกรมฐานข้อมูลนั้นจะต้องมีการเพิ่ม ลบ ปรับปรุงข้อมูล ฉนั้นจึงจำเป็นที่ต้องทราบว่าในการกำหนดตัวแปรที่ เป็นพารามิเตอร์นั้นมีการหนด อย่างไร การกำหนดตัวแปรที่ใช้เป็นพารามิเตอร์มีรูปแบบดังนี้

ชื่อฟิลด์ในตาราง =: ชื่อตัวเปรทีเป็นพารามิเตอร์

ตัวอย่าง

Query1.Close; Query1.SQL.Clear; Query1.SQL.Add(' Select * '); Query1.SQL.Add(' From Employee '); Query1.SQL.Add(' Where Employee_id=: now_id '); Query1.ParamByname(' now_id').AsString :=Edit1.Text ; Query1.Open;

รูปแบบคำสั่ง

ชื่อ Query.ParamByName('ชื่อตัวแปร').ชนิดของตัวแปร :=

24

โปแกรม SQL SERVER

้โปรแกรม SQL Server เป็นโปรแกรมจัดการฐานข้อมูลเชิงสัมพันธ์ ของบริษัท Microsoft

วิธีการใช้งาน SQL Server

- โปรแกรม Delphi และ SQL Server อยู่บนเครื่องเดียวกัน ในกรณีนี้ให้ลง โปรแกรม SQL Server ไว้ในเครื่อง Client ด้วย และการใช้งานไม่จำเป็นต้องมีการ์ดเครือข่ายเพราะการติดต่อจะเป็นลักษณะแบบ Local ดังนั้นจะเรียก โปรแกรมแบบนี้ว่า SQL Server Service Manager
- โปรแกรม Delphi และ SQL Server อยู่ต่างเครื่องกัน ในกรณีนี้ให้ลง โปรแกรม SQL Server เฉพาะที่เครื่อง Server เท่านั้น และการใช้วิธีนี้ จะต้องมีการเชื่อมโยงระหว่างเครื่องทั้งสองด้วยเครือข่าย และต้องติดตั้งโปรโตคอลที่จะ สนับสนุนการเชื่อมต่อแบบเดียวกัน โดยที่ SQL Server สนับสนุนโปรโตคอลมาตรฐานทั้งสามประเภทคือ NetBEUI, TCP/IP, IPX/SPX

บรรยายก็มานานแล้วเรามาเริ่มสร้าง Client- Server Database Application ตัวแรกกันเลย

- สร้าง ฐานข้อมูลใน SQL Server ชื่อว่า tadatest
- โดยเริ่มที่ Start -> Programs -> Microsoft SQL Server7 -> Enterprise Manager
- จะปรากฎหน้าจอดังรูป



คลิกขวาที่โฟเดอร์ ที่ชื่อว่า Databases -> New Database จากนั้นก็ตั้งชื่อฐานข้อมูลชื่อ datatest ดังรูป แล้วกด OK

ไพฑูรย์ จิ้วทั้ง	2/12/022545					
Database Properties - d	atatest					×
General Transaction Log]					
<u>N</u> ame: da	tatest		_			
Database files						-
File name	Location		Initial si	ze (MB)	File gro	up
娕 datatest_Data	C:\MSSQL7\d	ata\datat	1		PRIMAP	IY
						-11
File properties						-1
-V Automatically grow	file					
File growth		- Maximum	file size –			
C. In mercabutes:		Intest	tricted file	arowth		
in megabytes.		- <u>o</u> nics	anotod nie	giomai		
By percent:		C <u>R</u> estri	ct filegrow	vth (MB):	2	-
			_		1	
		01	K	Cancel	He	lp

- จากนั้นให้สร้าง Table โดยการคลิกที่เครื่องหมาย + หน้า โฟลเดอร์ DataBases -> คลิกเครื่องหมาย +หน้า ฐานข้อมที่
 ชื่อ datatest
- จากนั้นให้คลิกขาวที่ Tables -> New Table ดังรูป

26



ไพฑูรย์ งิ้วทั่ง 2/12/022545

- ตั้งชื่อ และ ออกแบบเหมือนกับดังนี้
- การกำหนด key หลัก ให้คลิกจองที่ ฟิลด์ ที่กำหนดจากนั้นก็คลิกที่ Tools bar รูปกุญแจ

เอนทิตี้สินค้า (tgoods.db)

27

ชื่อ Flild	Data Type	Width	Decription	Key
Goods_id	Char	10	รหัสสินค้า	*
Company_id	Char	10	รหัสบริษัท	
Goods_name	Char	25	ชื่อสินค้า	
Onhand	Integer	4	จำนวนคงเหลือ	

เอนทิตี้ บริษัทจัดจ่ำหน่าย(tcompany.db)

ชื่อ Flild	Data Type	Width	Decription	Key
Copany_id	Char	10	รหัสบริษัทจัดจำหน่าย	*
Company_name	Char	20	ชื่อบริษัท	
Company_add	Char	20	ที่อยู่	

เอนทิตี้ พนักงาน(tem.db)

ชื่อ Flild	Data Type	Width	Decription	Key
Em_id	Char	5	รหัสพนักงาน	*
Em_name	Char	20	ชื่อ -สกุล	
Em_add	Char	30	ที่อยู่	
Em_us	Char	10	User Name	
Em_pass	Char	10	Pass word	
Em_write	Char	5	กำหนดสิทธิ์ Write	

เอนทิตี้ สั่งสินค้า(torder.db)

ชื่อ Flild	Data Type	Width	Decription	Key
Or_id	Integer	4	ใบสั่งชื่อเลขที่	*
Or_date	DateTime	8	วันที่สั่งชื้อ	
Em_id	Char	5	รหัสพนักงาน	
Company_id	Char	10	รหัสบริษัท	
Price_tatal	Integer	4	ราคารวมทั้งสิน	

เอนทิตี้ รายละเอียดการสั่งชื้อ(torderdetail.db)

ชื่อ Flild	Data Type	Width	Decription	Key
Or_id	Integer	4	ใบสั่งซื้อเลขที่	
Or_no	Integer	4	ลำดับที่	
Goods_Id	Char	10	รหัสสินค้า	
Goods_name	Char	25	ชื่อสินค้า	
Or_num	Integer	4	จำนวนที่สั่ง	
Price	Integer	4	ราคา/หน่วย	

การกำหนด Security Logins

• คลิก เครื่องหมาย + หน้าคำว่า Security จากนั้นให้คลิกขวาที่คำว่า Login -> New Login.. ดังรูป

SQL Server Login Properties - New Login	×
General Server Rolez Database Access	
None: Toglest	
Authentication	- 1
C Windows NT sufferitation	
Degwin. 💌	
Density assess	
🕫 jij ant access	
C Dergannas	
Eastword	
Defaults	
Specity the default language and database for this login.	
Qafabase:	
Language: <delsuit></delsuit>	
OK Cancel He	Þ

Name ให้ใส่ว่า logtest , Password ให้ใส่ว่า 258584 ,เลือก Database ชื่อ datatase

SQL Serve	r Login Properties - New Login	×
General	Server Roles Database Access	
		1
Server R	loles -	
3	Server roles are used to grant server-wide security privileges to a login.	
	Server Role -	
	System Administrators	
	Security Administrators	
	Server Administrations	
	Setup Administrators	
	Process Administrators	
	Disk. Administrators	
	Database Cleators	
	Description	
	Can perform any activity in the SQL Server installation.	
	Experies	
	OK Cancel Help	1

จากนั้นให้เลือก Page Server Roles ให้คลิกหน้าคำว่า System Adminstator เป็นเครื่องหมายถูกตามรูป ด้านบน

29		ไพฑูรย์ งิ้วทั่ง 2/12/022545	
	SQL Serve	er Login Properties - New Login	×
	General	Server Roles Database Access	
	Databas	IR BOORSS	- 1
		Specily which databases can be accessed by this login. Pemit Database User	
		dalatest logtest	- 11
			- 111
		- rote	- 88
			- 111
		C C zdata	- 110
		- E H	- 11
		🗆 🗧 tempdb	-
		Database roles for 'datatest':	
		Permit in database role	
		🗹 🌔 public	
		to do_owner	
		C C db_eccessadmin	
		C do_tecuipeann	
		Experimenter Experimenter	117
		OK Cancel	ielp
	Ψ.	الم م ال	

้จานนั้นให้คลิก Page Database Access เลือกฐานข้อมูลที่เข้าถึง โดยเลือกที่ datatest แล้วกด OK

ขั้นตอนการเชื่อมโยงฐานข้อมูล (ODBC)

- เมื่อเข้าสู่ Control panel แล้ว
- 2. ให้ดับเบิ้ลคลิกที่ ODBC Data Source(32bit)
- 3. ที่ไดอะล็อก ODBC Data Source Adminstator ให้คลิกที่ปุ่ม Add
- 4. ให้เลือกชนิดของฐานข้อมูล ให้เลือกที่ SQL Server
- 5. แล้วคลิกปุ่ม Finish
- 6. ให้ตั้งชื่อ Data Source ว่า odbdatatest และ Server เป็น [Local] ดังรูปด้านล่าง

Create a New Data Sou	rce to SQL Server 🔀
Select a driver we	This wizard will help you create an ODBC data source that you can use to connect to SQL Server. What name do you want to use to refer to the data source? Neme: odbdatatest How do you want to describe the data source? Description: Which SQL Server do you want to connect to? Server: local
	Finish Next > Cancel Help

จากนั้นคลิก Next

ไพฑูรย์ งิ้วทั้ง 2/12/022545

30

Create a New Data Sou	rce to SQL Server
Salard a davie vor	How should SQL. Server verify the authenticity of the login ID?
	<back next=""> Cancel Help</back>

ใส่ login ID : logtest ,Password: 258584 ดังรูป คลิก Next จะพบหน้าจอด้านล่าง

Create a New Data Sou	irce to SQL Server	×
Select a denter un entre Accession of Based (Change the default database to datatest Attach database Rename:	
	 Desite temporary atoms procedures for prepared PDL statements end dop the stored procedures: Chi when you disconnect. When you disconnect and as appropriate while you are connected. 	
	V Lice ANSI quoted identifiers.	
2025 - CENE	Use ANSI nulls, paddings and warnings.	
	 Use the jalover GOL Servent the primary GOL Genvenis not everytable. 	
	<back next=""> Cancel Help</back>	

คลิก Next

ไพฑูรย์ จิ้วทั่ง 2/12/022545	
Create a New Data Source to SQL Server	×
Change the language of SQL Server system messages to English Change the language of SQL Server system messages to English Change the language of SQL Server system messages to English Change the language of SQL Server system messages to English Change the language of SQL Server system messages to English Change the language of SQL Server system messages to English Change the language of SQL Server system messages to English Change the language of SQL Server system messages to English Change the language of SQL Server system messages to English Change the language of SQL Server system messages to English Change the language of SQL Server system messages to English Change the language of SQL Server system messages to English Change the language of SQL Server system English E]
< Back Finish Cancel Help	J

คลิก Finish

31



→คลิก Test Data Source ว่าODBC ผ่านหรือไม่ถ้าจะปรากฏตามรูป



ไพฑูรย์ จิ้วทั่ง 2/12/022545

เราก็จะได้ Data Source Name (DSN) ตามรูป

32

Name	Driver	▲ Add
dBASE Files	Microsoft dBase Driver (*.dbf)	
Excel Files	Microsoft Excel Driver (*.xlo)	Remove
Export	Microsoft Paradox Driver (*.db.)	
FaxPto Files	Microsoft FoxPro Driver (*.dbf)	Configure
MS Access 97 Database	Microsoft Access Driver (*.mdb)	
odbdatatest	SQL Server	
pcience	SQL Server	
s-science	SQL Server	
Text File:	Microsoft Test Driver (".txt; ".csv)	
Visual FoxPro Database	Microsoft Visual FoxPro Driver	
Visual FoxPro Tables	Microsoft Visual FoxPro Driver	*
An ODBC User the indicated da and can only be	data source stores information about ho sta provider. A User data source is only a used on the current machine.	w to connect to y visible to you,

คอมไพเลอร์ Object Pascal (Delphi)

คอมไพเลอร์ถูกเรียกใช้โดย IDE (หรือผ่าน Command Lineก็ได้) คอมไพเลอร์จะแปลงโปรแกรมออกมา เป็น ภาษาเครื่อง (Machin Language) ซึ่งในวินโคว์จะมีนามสกุล เป็น .EXE หรือแปลงออกมาเป็น .DLL หรือ Dynamic Link Library

****(การเขียนในครั้งนี้ในส่วนของ DELPHI 5.0 จะใช้คอมโปรแนน infopower2000 มาร่วมด้วยสามารถไป Dowload ได้ที่เวบไชต์www.woll2woll.com จากนั้นให้ ทำการ SETUPลงไปด้วย โดยใช้รหัสผ่านดังนี้

IP8250248391)******

1.การออกแบบหน้าจอ Design From หลัก (กำหนดชื่อว่าFmain)



2.การออกแบบหน้าจอ Design From บริษัทจัดจำหน่าย

กำหนดชื่อว่า Fcompany

🚟 ข้อมูลบริมีกรักร่างนี่	na i		
Add	งที่สมให้เขพิดห่างน่าง	Świtika	-
Delete	faij		-
Edit Save	Company_id Company_nam	we Company_add	-
Cancel			
			1
	search company code search Edv1 View All	_	
		•	

3.สร้างหน้าจอ Data Modul

AT DM1					. D X
DM1	Components	Data Diag	rain		
Comman to eccar) Comman to eccar) Comman to eccar) Comman to eccar) Comman to Comman to	Indefabere		Q company	Dompany	
- 😽 Doompany - 📩 Field: - 🍓 Parana	Gorder	Dorder			
	Qorderd	Dorderd			
	State				
		_			

- 1. กำหนดชื่อ Data modul ว่า DM1
- 2. ให้นำ Component ที่ชื่อว่า DataBase อยู่ในกลุ่ม Data Access หลังจากนั้นให้กำหนดคุณสมบัติดังนี้
 - กำหนดชื่อ Indatabase
 - กำหนดAliasName→ odbdatatest
 - กำหนด Loginpromt → False
 - กำหนด Params

USER NAME=logtest PASSWORD=258584

- กำหนด Connected > True
- 3. ให้น้ำ Component ที่ชื่อว่า Query และ Datasource อยู่ในกลุ่ม Data Access หลังจากนั้นให้กำหนดคุณสมบัติดังนี้
 - Query กำหนดชื่อ Qcompany
 - DataBaseName → datatest

- CachedUpdate → True
- RequestLive \rightarrow True
- SQL→ select*

34

from tcompany

- Active → True
- Datasource กำหนดชื่อ Dcompany
- Dataset → Qcompany
- 4. เปิด From Fcompany
 - ให้ทำการ Add UserUnit ของ From Datamodul เข้ามา
 - จากนั้น ให้ทำ การ Set ค่าของDBEdit และ DBGrid ไปที่ Qcompany และ Dcompany ตามลำดับ
 - แล้วก็เขียนคำสั่งในปุ่มต่างๆ ดังนี้

Tcompany ปุ่ม Add

procedure TFcompany.Button1Click(Sender: TObject);

begin

//เพิ่มคำสั่งSQL

Dm1.Qcompany.Close;

DM1.Qcompany.SQL.Clear;

DM1.Qcompany.SQL.Add('select * From tcompany');

DM1.Qcompany.Open;

//เพิ่มข้อมูล

DM1.Qcompany.Append;

end;

Tcompany ปุ่ม Delete

procedure TFcompany.Button2Click(Sender: TObject);

begin

//ยื่นยันการลบ

if messageDlg('ต้องการลบจริงหรือไม่',mtConfirmation,[mbYes, mbNo],0)= mrYes Then

DM1.Qcompany.Delete;

DM1.Qcompany.Edit;

DM1.Qcompany.post;

DM1.Qcompany.CommitUpdates;

DM1.Qcompany.Close;

35

ไพฑูรย์ งิ้วทั่ง 2/12/022545

DM1.Qcompany.Open;

end;

Tcompany ปุ่ม Edit

procedure TFcompany.Button3Click(Sender: TObject);

begin

DM1.Qcompany.Edit;

end;

Tcompany ปุ่ม Save

procedure TFcompany.Button4Click(Sender: TObject);

begin

DM1.Qcompany.Edit;

DM1.Qcompany.post;

DM1.Qcompany.CommitUpdates;

DM1.Qcompany.Close;

DM1.Qcompany.Open;

end;

Tcompany ปุ่ม Cancel

procedure TFcompany.Button5Click(Sender: TObject);

begin

DM1.Qcompany.Cancel;

end;

Tcompany ปุ่ม Search

procedure TFcompany.Button6Click(Sender: TObject);

begin

DM1.Qcompany.SQL.Clear;

DM1.Qcompany.SQL.Add('select * From tcompany');

DM1.Qcompany.SQL.Add('Where Company_id=:now');

DM1.Qcompany.ParamByName('now').Asstring:=edit1.Text;

DM1.Qcompany.Open;

ไพฑูรย์ จิ้วทั่ง 2/12/022545

if DM1.Qcompany.ParamByName('now').Asstring=edit1.Text Then

Begin

36

ShowMessage('พบข้อมูลที่ค้น');

End

Else

Begin

ShowMessage('ไม่พบข้อมูลที่ค้น');

End;

end;

Tcompany ปุ่ม View All

procedure TFcompany.Button7Click(Sender: TObject);

begin

Dm1.Qcompany.Close;

DM1.Qcompany.SQL.Clear;

DM1.Qcompany.SQL.Add('select * From tcompany');

DM1.Qcompany.Open;

end;

end.

4.การออกแบบหน้าจอ Design From พนักงาน

. Charles and a set			
	รพัสพณิกราม	นิจพนิกราม	
	D BE dirt	DBE dt2	
Add			
Datation	Tag		
088.5	D BE dk3		
5 a 1			
Save			
Cancel			

- 1. กำหนด ชื่อFrom -→ Ftem
- 2. เปิด From Data Modul ขึ้นมา

37	ไพฑูรย์ งิ้วทั้ง	2/12/0225	45			
	PDM1					
	B-H Default (Session)	Components	Data Diagra	am 🛛		
		Indelebace		Q company	Dcompany	
	datatest indatabasei Goompany Constraints Doompany	Corder	Dorder	Citem	Dtem	
	Sields Signature Si	Qorderd	Dorderd			
	Fields Paramo	Report				
		<u> </u>				

- 3. ให้น้ำ Component Query และ DataSource มาวาง อย่างละหนึ่ง Component
- 4. Query ให้เปลี่ยนชื่อ Qtem และกำหนด DatabaseName → datatest
- 5. CacheUpdate \rightarrow True
- 6. RequestLive \rightarrow True
- 7. SQ \rightarrow select *

From tem

- 8. Active →True
- 9. Datasource ให้เปลี่ยนชื่อ Dtem และกำหนด Dataset → Qtem
- 10. ให้เปิด From ของ พนักงานขึ้นมา แล้วทำการ Use Unit ของ From DataModul เข้ามา
- 11. และทำการ กำหนดค่าต่าง ของ Ccmponent เช่น Dbedit1→DatasourceName=Dtem Datafield= Em_id ฯ
- 12. **จากนั้น**ให้เขียนคำสั่งในปุ่มดังนี้

ปุ่ม Add

procedure TFtem.Button1Click(Sender: TObject);

begin

DM1.Qtem.Append;

end;

ปุ่ม Delete

procedure TFtem.Button2Click(Sender: TObject);

begin

//ยื่นยันการลบ

ไพฑูรย์ จิ้วทั่ง 2/12/022545 if messageDlg('ต้องการลบจริงหรือไม่',mtConfirmation,[mbYes, mbNo],0)= mrYes Then DM1.Qtem.Delete ; DM1.Qtem.Edit; DM1.Qtem.post ; DM1.Qtem.CommitUpdates ; DM1.Qtem.Close; DM1.Qtem.Open; end;

ปุ่ม Edit

38

procedure TFtem.Button3Click(Sender: TObject);

begin

DM1.Qtem.Edit;

end;

ปุ่ม Save

procedure TFtem.Button4Click(Sender: TObject);

begin

DM1.Qtem.Edit;

DM1.Qtem.post;

DM1.Qtem.CommitUpdates;

DM1.Qtem.Close;

DM1.Qtem.Open;

end;

ปุ่ม Cancel

procedure TFtem.Button5Click(Sender: TObject);

begin

Dm1.Qtem.Cancel;

end;

end.

5.การออกแบบหน้าจอ Design From คลังสินค้า

39	ไพฑูรย์ งิ้วทั้ง 2/12/	/022545	
SP 601	มุลสินกัง		
	Add Statuth DECdr1 Delete CBLostupComboBlor1	Saltvén DBEdit2 DBEdit3	
	Edit		-
	Cancel		
5 😒	SQL Server Enterprise Ma Wikeoust W	ord - ClientSec. 🖉 Delphi 5	- 8 54 3 9 >

- 1. ให้กำหนดชื่อ From → Fgoods
- ให้ทำการเปิด From DataModul ขึ้นมา และให้นำ Component Query มาวาง 2 Component กับ Datasourec มาวาง 2 Component



- 3. Query1 ให้กำหนด Name → Qgoods
- 4. DatabaseName →datatest
- 5. CacheUpdate → True
- 6. RequestList \rightarrow True
- 7. SQL \rightarrow select *

From tgoods

- 8. Active → True
- 9. Datasource1 ให้กำหนด Name → Dgoods
- 10. Dataset →Qgoods

40 ไพฑูรย์ งิ้วทั้ง 2/12/022545

- 11. Query2 ให้กำหนด Name → Qcompany_list_goods
- 12. DatabaseName →datatest
- 13. CacheUpdate → True
- 14. RequestList \rightarrow True
- 15. SQL → select *

From tcompany

- 16. Active → True
- 17. Datasource2 ให้กำหนด Name → Dcompany_list_goods
- 18. Dataset →Qcompany_list_goods
- ี่ 19. ต่อมาให้เปิด From ของ**สินค้า**ขึ้นมา
- 20. ให้ทำการ Use Unit ของ DataModul เข้ามา และทำการ กำหนดค่าให้กับ Component ต่างๆ
- 21. เมื่อกำหนดค่าให้กับ Component เสร็จเรียบร้อยแล้วก็ให้เขียนคำสั่งในปุ่มต่าง ๆดังนี้

ปุ่ม Add

procedure TFgoods.Button1Click(Sender: TObject);

begin

DM1.Qgoods.Append;

end;

ปุ่ม Delete

procedure TFgoods.Button2Click(Sender: TObject);

begin

//ยื่นยันการลบ

if messageDlg('ต้องการลบจริงหรือไม่',mtConfirmation,[mbYes, mbNo],0)= mrYes Then

DM1.Qgoods.Delete;

DM1.Qgoods.Edit;

DM1.Qgoods.post;

DM1.Qgoods.CommitUpdates;

DM1.Qgoods.Close;

DM1.Qgoods.Open;

end;

41 ไพฑูรย์ จิ้วทั่ง 2/12/022545

ปุ่ม Edit

procedure TFgoods.Button3Click(Sender: TObject);

begin

DM1.Qgoods.Edit;

end;

ปุ่ม Save

procedure TFgoods.Button4Click(Sender: TObject);

begin

DM1.Qgoods.Edit;

DM1.Qgoods.post;

DM1.Qgoods.CommitUpdates;

DM1.Qgoods.Close;

DM1.Qgoods.Open;

end;

ปุ่ม Cancel

procedure TFgoods.Button5Click(Sender: TObject);

begin

DM1.Qgoods.Cancel;

end;

6.การออกแบบหน้าจอ Design From การสั่งชื้อสินค้า

สารส่งใจฟ้า		
Add Delete Edit Save Cancel Print	uerňiuší Stuří D0Edri D0Edri uškinšké/málus Edri DELookupConi D8Text1 Stateušanu D8Text2 DELookupConi D8Text2 Stateušanu D8Text2 Stateušanu D8Text2 Stateušanu D8Text2 Stateušanu D8Text2 Stateušanu D8Text2	
	sumsed DBText3 sum	,

42 ไพตูรย์ จิ้วทั่ง 2/12/022545

- 1. ให้กำหนด Name → Forder
- 2. ให้เปิด From ของ Data Modul แล้วน้ำ Componen มาวาง.



3. กำหนดค่าที่ Qorder

DatabaseName \rightarrow datatest

SQL \rightarrow select *

from torder

where or_id =:now

RequestLive -→ True

CachedUpdate \rightarrow True

Active → True

4. กำหนดค่า Dorder

Datasete \rightarrow Qorder

5. กำหนดค่าที่ Qorderd

DatabaseName → datatest

SQL \rightarrow select *

from torderdetail

where Or_id =:Or_id

order by or_no

RequestLive - \rightarrow True

CachedUpdate → True

ไพฑูรย์ งิ้วทั่ง 2/12/022545 43 Active → True 6. กำหนดค่า Dorderd Datasete \rightarrow Qorderd 7. กำหนดค่าที่ Qorder_tem DatabaseName \rightarrow datatest SQL → select * from tem where Em_id =: Em_id RequestLive -→ True CachedUpdate → True Active → True 8. กำหนดค่าที่ Dorder_tem Dataset →Qorder_tem 9. กำหนดค่าที่ Qorder_company DatabaseName \rightarrow datatest SQL → select * from tcompany where Company_id =:Company_id RequestLive -→ True CachedUpdate → True Active → True 10. กำหนดค่าที่ Dorder_company Dataset →Qorder_company 11. กำหนดค่าที่ Qorder_goods DatabaseName → datatest SQL → select * from tgoods RequestLive -→ True CachedUpdate → True Active \rightarrow True 11. กำหนดค่าที่ Dorder_goods

44 ไพตูรย์ งิ้วทั่ง 2/12/022545

Detaset \rightarrow Qorder_goods

- 12. จากนั้นให้ทำการ เปิด From ของ การสั่งซื้อ แล้วเรียกใช้ Use Unit ของ Data Modul และทำการกำหนดค่า ต่าง ของ Component ที่ได้ออกแบบมา
- 13. จากนั้นให้เขียนคำสั่งในปุ่มต่างๆ ดังนี้

ปุ่ม ADD

procedure TForder.Button2Click(Sender: TObject);

var i:integer;

begin

//เพิ่มคำสั่งSQL

Dm1.Qorder.Close;

DM1.Qorder.SQL.Clear;

DM1.Qorder.SQL.Add('select * from torder ');

DM1.Qorder.Open;

dm1.Qorder.Last;

i:= Dm1.Qorder.FieldByName('or_id').Asinteger;

if i > 0 Then

begin

i:=i+1;

End

Else

begin

i:= 1;

End;

dm1.Qorder.Append;

Dm1.Qorder.FieldByName('or_id').Asinteger := i;

Dm1.Qorder.FieldByName('or_date').AsdateTime:=date;

end;

ปุ่ม Delete

procedure TForder.Button3Click(Sender: TObject);

var count,i: integer;

begin

if messageDlg('ต้องการลบจริงหรือไม่',mtConfirmation,[mbYes, mbNo],0)= mrYes Then

Count:=DM1.Qorderd.RecordCount;

ไพฑูรย์ งิ้วทั่ง 2/12/022545 45 if count > 0 Then Begin Dm1.Qorderd.First; For i:=1 To Count do Begin Dm1.Qorderd.Delete; End; End; DM1.Qorder.Delete; DM1.Qorder.Edit; DM1.Qorder.Post; DM1.Qorder.CommitUpdates; DM1.Qorder.Close; DM1.Qorder.Open; end; ปุ่ม Edit procedure TForder.Button4Click(Sender: TObject); begin Dm1.Qorder.Edit; end; ปุ่ม Save procedure TForder.Button5Click(Sender: TObject); var price,sum,count,i:integer; begin // บันทึกรายละเอียด dm1.Qorderd.Edit; dm1.Qorderd.Post; dm1.Qorderd.CommitUpdates; dm1.Qorderd.Close; dm1.Qorderd.Open;

//บันทึกหลัก

Count:= Dm1.Qorderd.RecordCount ;

Dm1.Qorderd.First;

46 ไพฑูรย์ งิ้วทั้ง 2/12/022545

Price:=0;

for i:= 1 to Count do

begin

sum:=Dm1.Qorderd.fieldByname('or_num').Asinteger *

Dm1.Qorderd.fieldByname('price').Asinteger;

price:=price+sum;

end;

dm1.Qorder.Edit ;

Dm1.Qorder.FieldByName('price_tatal').Asinteger:=price;

dm1.Qorder.Edit;

dm1.Qorder.Post;

dm1.Qorder.CommitUpdates;

dm1.Qorder.Close;

dm1.Qorder.Open;

end;

ปุ่ม Cancel

procedure TForder.Button6Click(Sender: TObject);

begin

Dm1.Qorderd.Cancel;

Dm1.Qorder.Cancel;

end;

ปุ่ม Find

procedure TForder.Button7Click(Sender: TObject);

begin

//เพิ่มคำสั่งSQL

DM1.Qorder.SQL.Clear;

DM1.Qorder.SQL.Add('select * From torder');

DM1.Qorder.SQL.Add('Where or_id=:now');

DM1.Qorder.ParamByName('now').Asstring:=edit1.Text;

DM1.Qorder.Open;

end;

ปุ่ม Add (ในส่วนของรายละเอียด)

procedure TForder.Button1Click(Sender: TObject);

ไพฑูรย์ งิ้วทั่ง 2/12/022545 47 var i: integer; begin DM1.Qorderd.Last; i:=Dm1.Qorderd.FieldByName('or_no').Asinteger; if i<0 Then Begin i:=1 End; DM1.Qorderd.Append; Dm1.Qorderd.FieldByName('or_id').Asinteger := DM1.Qorder.fieldByname('or_id').Asinteger; Dm1.Qorderd.FieldByName('or_no').Asinteger:= i+1; end; คำสั่งในเหตุการ CloseUp ที่ใช้ List รายการของสินค้า procedure TForder.DBLookupComboBox1CloseUp(Sender: TObject); begin Dm1.Qorderd.Edit; Dm1.Qorderd.FieldByName('goods_name').ASString:= DM1.Qgoods.fieldByname('goods_name').Asstring; end; ้คำสั่งในปุ่มเห็นการคลิ๊กขวาของเมาส์บน Gridของรายละเอียด(Menu PopUp) เมนู เพิ่ม procedure TForder.N1Click(Sender: TObject);

var i: integer;

begin

DM1.Qorderd.Last;

i:=Dm1.Qorderd.FieldByName('or_no').Asinteger;

if i<0 Then

Begin

i:=1

End;

DM1.Qorderd.Append;

Dm1.Qorderd.FieldByName('or_id').Asinteger :=

48 ไพฑูรย์ งิ้วทั้ง 2/12/022545

DM1.Qorder.fieldByname('or_id').Asinteger;

Dm1.Qorderd.FieldByName('or_id').Asinteger:= i+1;

end;

เมนูลบ

procedure TForder.N2Click(Sender: TObject);

begin

if messageDlg('ต้องการลบจริงหรือไม่',mtConfirmation,[mbYes, mbNo],0)= mrYes Then

Dm1.Qorderd.Delete;

Dm1.Qorderd.Edit;

Dm1.Qorderd.Post;

Dm1.Qorderd.CommitUpdates ;

Dm1.Qorderd.Close;

Dm1.Qorderd.Open;

end;

เมนูแก้ไข

procedure TForder.N3Click(Sender: TObject);

begin

DM1.Qorderd.Edit;

end;

เมนูบันทึก

procedure TForder.N4Click(Sender: TObject);

begin

DM1.Qorderd.Edit;

Dm1.Qorderd.Post;

Dm1.Qorderd.CommitUpdates;

DM1.Qorderd.Close;

Dm1.Qorderd.Open;

end;

เมนูยกเลิก

procedure TForder.N5Click(Sender: TObject);

begin

DM1.Qorderd.Cancel;

end;

14. ต่อมาให้ทำการ สร้าง From พิมพ์ใบสั่งชื้อสั่งชื้อ

สร้างที่ New → เลือก Report

และทำการออกแบบ ตามรูปด้านล่าง และกำหนดชื่อ Qrreport



15. เปิด From DataModulขึ้นมา และนำ Query มาวาง 1 Component ดังภูป



50 ไพตูรย์ งิ้วทั่ง 2/12/022545

กำหนด ชื่อQuery -→ qreport

DatabaseName →datatest

CacheUpdate → True

 $\mathsf{RequestList} \rightarrow \mathsf{True}$

SQL → SELECT Torder.or_id, Torder.or_date, Torder.em_id, Torder.company_id, Torder.price_tatal, Torderdetail.or_no, Torderdetail.goods_id, Torderdetail.goods_name, Torderdetail.or_num, Torderdetail.price, Tcompany.company_name, Tcompany.company_add, Tem.Em_name, (Torderdetail.or_num*Torderdetail.price) FROM torder Torder, torderdetail Torderdetail, tcompany Tcompany, tem Tem WHERE (Torder.or_id =:now) AND (Torderdetail.or_id = Torderdetail.or_id) AND (Tcompany.Company_id = Torder.Company_id) AND (Tem.Em_id = Torder.em_id)

Active → True

 เปิด From Reprot ใบสั่งซื้อขึ้นมาแล้วทำการ Use Unit ของ Data Modul เข้ามา และกำหนดค่าต่างๆ ใน Component ที่ได้ออกแบบไว้

Despert																	ЗĽ,
				1.1												**	
																	÷
																	4
Page Reader																	÷
						1.											4
							A410.0										÷
Sect	1.2					- 6	4.4	- C									÷
	252						10.18	a									÷
1.2.2	*					-		-									÷
1.0	1.00											- 346	1.4.	lain .			÷
3.4	Townson of	ite m										1 4	11	4			÷
110		10,000										1940	- 528	w)			÷
5.00	11	r	4.5			T			777	1.1	-	11	r	1			ъ
204.8		- Lines	inum:	_			in the second			100			in the second				3
10.00	111	goods.				111			- P	10.10		12	104.18	ALIMAN .			1
14. A		5				1				6.1		4 14	C				1
									÷	5	÷	· [·	r				£.
and the second									2000	264348	n., .		122M)	heling:	OUT	1814Q	£.
	1.4.4											~~~					1
	10000	84				-11		644	Store B.								
		·	7					s					1.				1.
		20,000	-						<u> </u>				-0				1.
				_						r		7					ъ.
Top Fame		h.		1						6							٦.
																	1
	-			-	-	-	_			_		-		_			

17. เปิด From ของการสังซือขึ้นมา และทำการเขียน**คำสั่งที่ปุ่ม Print** ดังนี

procedure TForder.Button8Click(Sender: TObject);

begin

dm1.qreport.Close;

dm1.qreport.ParamByName('now').Asinteger := DBEdit1.Field.Value ;

dm1.qreport.Open;

Qrreport.Preview;

end;

18. คำสั่งในหน้าจอหลัก

เมนูข้อมูลพนักงาน

procedure TFmain.N2Click(Sender: TObject);

begin

Ftem.Show;

end;

เมนูข้อมูลสินค้า

procedure TFmain.N3Click(Sender: TObject);

begin

Fgoods.Show;

end;

เมนูบริษัทจัดจำหน่าย

procedure TFmain.N4Click(Sender: TObject);

begin

Fcompany.show;

end;

เมนูการสั่งซื้อสินค้า

procedure TFmain.N7Click(Sender: TObject);

begin

Forder.show;

end;

เมนูออกจากโปรแกรม

procedure TFmain.N5Click(Sender: TObject);

begin

Application.Terminate;

end;

end.

19. ทดสอบการRun