

เอกสารอ้างอิง



ภาษาไทย

ครรรชิต มาลัยวงศ์. วิศวกรรมซอฟต์แวร์ เคสและธุรกิจ. วารสาร บิซิเนส คอมพิวเตอร์,
ปีที่ 2 ฉบับที่ 15 เดือน พฤษภาคม, 2533.
สมพัฒน์ รุ่งตะวันเรืองศรี. เรียนรู้คอมพิวเตอร์กราฟิกส์ 2 มิติด้วยภาษา C
บริษัท ซีเอ็ดยูเคชั่น จำกัด, 2535

ภาษาอังกฤษ

Richard Baker. CASE* METHOD. Entity Relationship Modelling.
Addison-Wesley Publishing Company, 1990.
Ken S. Brathwaite. Analysis, Design and Implementation of Data Dictionary.
McGraw-Hill Book Company, 1988.
James A. Senn. Analysis & Design of Information Systems.
McGraw-Hill Publishing Company, 1989.
Fleming Candance C., Babara von Halle. Handbook of Relational Database Design.
Addison-Wesley Publishing Company, 1989.
Chris Gane. Computer-Aided Software Engineering, the methodologies, the product and
the future Prentice-Hall international, Inc, 1990.
Shan Lawrence Pfleeger. Software Engineering, the production of quality software.
USA: Macmillan Publishing Company, 1987.
James Martin. System Design From Provably Correct Construct.
Prentice-Hall, Inc, 1985.
George U. Hubbard. Computer Assisted Database Design.
Van Nostrand Reinhold Company, Inc, 1981.

ภาคผนวก

ภาคผนวก ก
ส่วนประกอบของรหัสต้นฉบับ

โปรแกรมช่วยออกแบบแบบจำลองข้อมูลที่สร้างขึ้นในการวิจัยนี้ เป็นโปรแกรมซึ่งทำงานภายใต้ระบบจัดการวินโดว์ จัดทำขึ้นโดยใช้รหัสภาษาซี โดยใช้ตัวแปลโปรแกรม Borland C++ for Windows และมีการเชื่อมโยงกับคลังโปรแกรม BRIDGE.DLL ซึ่งเป็นตัวกลางในการติดต่อกับโปรแกรมจัดการข้อมูล CodeBase

ส่วนประกอบของรหัสต้นฉบับ (source code) ของโปรแกรมในการวิจัยนี้ ประกอบด้วยเพิ่มข้อมูลดังต่อไปนี้ โดยที่ผู้วิจัยจะไม่ขอกล่าวในรายละเอียดเกี่ยวกับการเข้ารหัสโปรแกรมภาษา C และการเขียนโปรแกรมเพื่อให้ทำงานภายใต้ระบบจัดการวินโดว์ สำหรับผู้ที่สนใจและยังไม่มีความรู้ทางด้าน การเขียนโปรแกรมภาษา C เพื่อให้ทำงานภายใต้ระบบจัดการวินโดว์ สามารถศึกษาได้จากเอกสารอื่นๆ ที่เกี่ยวข้อง

โปรแกรมช่วยออกแบบแบบจำลองข้อมูล ประกอบด้วยเพิ่มรหัสต้นฉบับดังนี้

- WINMAIN.C

เป็นส่วนโปรแกรมหลักซึ่งทำหน้าที่ในการกำหนดค่าเริ่มต้นสร้างวินโดว์หลัก และทำหน้าที่วนรอบเพื่อดึงข้อความจากคิวข้อความ แล้วส่งต่อไปยังโปรแกรมประจำวินโดว์หลัก

- WININIT.C

เป็นส่วนโปรแกรมซึ่งทำหน้าที่ในการกำหนดค่าเริ่มต้นซึ่งถูกเรียกใช้จากโปรแกรมหลัก

- WINPROC.C

เป็นส่วนของโปรแกรมประจำวินโดว์หลัก ซึ่งทำหน้าที่ตรวจสอบข้อความที่ส่งมาจากโปรแกรมหลัก ก่อนแจกจ่ายไปยังส่วนของโปรแกรมอื่นๆ ซึ่งเตรียมไว้ในการทำงานตามเหตุการณ์ที่เกิดขึ้น รวมทั้งการสร้างวินโดว์ย่อยซึ่งทำหน้าที่เป็นบรรณาธิกรณ์แผนภาพด้วย

- INITBASE.C

เป็นส่วนโปรแกรมซึ่งใช้ในการติดต่อกับโปรแกรมจัดการข้อมูลโดยการเรียกผ่านฟังก์ชันซึ่งอยู่ในคลังโปรแกรม BRIDGE.DLL

- CLIENT.C

เป็นส่วนของโปรแกรมประจำวินโดว์ย่อย ซึ่งมีหน้าที่ในการสร้างวินโดว์ที่เป็นไม้บรรทัด และวินโดว์ที่เป็นพื้นที่แสดงแผนภาพ และทำหน้าที่ในการส่งผ่านข้อความไปให้ฟังก์ชันประจำวินโดว์ที่เป็นไม้บรรทัด และฟังก์ชันประจำพื้นที่แสดงภาพ

- RULERS.C
เป็นส่วนของฟังก์ชันประจำวินโดว์ที่เป็นไม้บรรทัด ซึ่งจะเป็นส่วนที่แสดงตำแหน่งภาพ และตำแหน่งของตัวชี้ตำแหน่งของอุปกรณ์เมาส์
- WORKAREA.C
เป็นส่วนของฟังก์ชันประจำวินโดว์ที่เป็นพื้นที่แสดงภาพ ซึ่งจะทำหน้าที่ในการสร้าง แสดง แก้ไข และลบวัตถุต่างๆ ที่ปรากฏในแผนภาพ
- OBJECT.C
เป็นส่วนของโปรแกรมที่ใช้ในการจัดการกับโครงสร้างข้อมูลที่ใช้ในการจัดเก็บข้อมูลของโปรแกรม
- DIALOG.C
เป็นส่วนของโปรแกรมที่ใช้ในการจัดการแบบฟอร์มสำหรับป้อนข้อมูลพจนานุกรมข้อมูล และกล่องคำโต้ตอบอื่นๆ ที่ใช้ในโปรแกรม
- DRAW.C
เป็นส่วนของโปรแกรมที่ใช้แสดงภาพต่างๆ บนหน้าต่างที่เป็นพื้นที่แสดงภาพ
- PRINT.C
เป็นส่วนของโปรแกรมที่ใช้ในการควบคุมการแสดงผลภาพออกทางเครื่องพิมพ์ การออกรายงาน และการสร้างโครงคำสั่งนิยามข้อมูล
- CALC.C
เป็นส่วนของโปรแกรมที่ใช้ในการคำนวณเนื้อหาของสื่อบันทึกข้อมูลในคำสั่ง CREATE TABLE
- UTILS.C
เป็นส่วนของโปรแกรมพื้นฐานซึ่งเตรียมไว้ให้โปรแกรมอื่นๆ เรียกใช้
- ERD.RC
เป็นส่วนของโปรแกรมที่ใช้บรรยายลักษณะของทรัพยากรของระบบจัดการวินโดว์ซึ่งโปรแกรมนี้เรียกใช้
- ERD.DEF
เป็นส่วนของโปรแกรมที่ใช้บรรยายลักษณะของหน่วยความจำ และฟังก์ชันภายนอกที่ใช้ในโปรแกรมนี้

ภาคผนวก ข

ฟังก์ชันพื้นฐานในการติดต่อระหว่างโปรแกรมส่วนหน้ากับโปรแกรมจัดการข้อมูล

ฟังก์ชันพื้นฐานที่ใช้ในการติดต่อระหว่างโปรแกรมส่วนหน้ากับโปรแกรมจัดการข้อมูลจัดทำไว้ในรูปแบบของคลังโปรแกรมแบบพลวัต โดยจัดเก็บไว้ในแฟ้มข้อมูลชื่อ BRIDGE.DLL ซึ่งรหัสต้นฉบับจัดเก็บในแฟ้มข้อมูล BRIDGE.C โดยมีโปรแกรม Borland C++ for Windows เป็นตัวแปลภาษา มีรูปแบบของโครงสร้างข้อมูลและฟังก์ชันเป็นดังนี้

โครงสร้างข้อมูล

โครงสร้างข้อมูลซึ่งใช้ระบุลักษณะของเขตข้อมูลในการสร้างตารางรีเลชันนัล โดยเขียนเป็นคำสั่งภาษาซี ดังนี้

1) ชนิดข้อมูลซึ่งใช้ระบุลักษณะของเขตข้อมูล

```
typedef struct {
    LPSTR name;
    char type;
    short len ;
    short dec ;
} FIELD_INFO ;
```

มีรายละเอียด ดังนี้

name	ชื่อเขตข้อมูลของตารางที่ต้องการสร้าง
type	ชนิดข้อมูลของเขตข้อมูล เช่น 'C' สำหรับระบุชนิดข้อมูลเป็นชนิดตัวอักษร 'N' สำหรับระบุชนิดข้อมูลเป็นชนิดตัวเลข 'D' สำหรับระบุชนิดข้อมูลเป็นชนิดวันที่
len	ความยาวสูงสุดที่เป็นได้ของข้อมูลในเขตข้อมูล
dec	จำนวนหลักหลังจุดทศนิยมในกรณีที่เขตข้อมูลมีชนิดข้อมูลเป็นตัวเลข

2) ชนิดข้อมูลซึ่งใช้ระบุถึงตัวชี้ (pointer) ซึ่งชี้ไปยังข้อมูลที่มีชนิดข้อมูลเป็น FIELD_INFO

```
typedef FIELD_INFO FAR *LPFIELD;
```

ฟังก์ชัน

1) ฟังก์ชันซึ่งใช้ในการเชื่อมต่อเข้ากับโปรแกรมจัดการข้อมูล

```
int ConnectDBS(HINSTANCE hInstance, HWND hMainWindow);
```

เป็นฟังก์ชันที่ใช้ในการเชื่อมต่อโปรแกรมประยุกต์เข้ากับโปรแกรมจัดการข้อมูล ก่อนที่จะเรียกใช้ฟังก์ชันในการจัดการข้อมูลอื่นๆ มีรายละเอียดของพารามิเตอร์ ดังนี้

hInstance	ระบุถึงโปรแกรมซึ่งต้องการเชื่อมต่อกับโปรแกรมจัดการข้อมูล
hMainWindow	ระบุถึงวินโดวหลักของโปรแกรมซึ่งต้องการเชื่อมต่อกับโปรแกรมจัดการข้อมูล

ค่าที่ส่งกลับ	1 - การเชื่อมต่อสำเร็จ
	0 - การเชื่อมต่อไม่สำเร็จ

2) คำสั่งในการสร้างตารางรีเลชันนัล

```
int CreateTable( LPSTR table_name, LPFIELD fields );
```

เป็นคำสั่งที่ใช้ในการสร้างตารางรีเลชันนัล ในฐานข้อมูลซึ่งโปรแกรมกำลังเชื่อมต่ออยู่ มีรายละเอียดของพารามิเตอร์ ดังนี้

table_name	ใช้ระบุชื่อตารางที่ต้องการสร้าง
fields	เป็นตัวชี้ซึ่งชี้ไปยังแถวลำดับที่มีชนิดข้อมูลเป็น FIELD_INFO ซึ่งจะบอกรายละเอียดเกี่ยวกับเขตข้อมูลแต่ละตัวในตารางที่ต้องการสร้าง โดยฟังก์ชันจะสร้างเขตข้อมูลในตาราง ตามลำดับของเขตข้อมูลทีระบุไว้ในตัวแปรแถวลำดับ และสิ้นสุดการสร้างเขตข้อมูลเมื่อ name ซึ่งเป็นตัวชี้ไปยังชื่อของเขตข้อมูล มีค่าเป็น 0 ดังตัวอย่างดังนี้

```
FIELD_INFO fields[3] = { {"name", 'C', 10, 0},
                          {"weigh", 'N', 5, 2},
                          {0, 0, 0, 0}
                        };
```

ค่าที่ส่งกลับ	1 - การสร้างตารางสำเร็จ
	0 - การสร้างตารางไม่สำเร็จ

3) คำสั่งในการแทรกรายการลงในตารางรีเลชันนัล

```
int InsertRow( LPSTR table_name, LPSTR *fname, LPSTR *value );
```

เป็นคำสั่งที่ใช้ในการแทรกรายการ 1 รายการ ลงในตารางรีเลชันนัลในฐานข้อมูลซึ่งโปรแกรมกำลังเชื่อมต่ออยู่ มีรายละเอียดของพารามิเตอร์ ดังนี้

table_name	ใช้ระบุชื่อตารางที่ต้องการแทรกรายการ
fname	เป็นตัวชี้ซึ่งชี้ไปยังแถวลำดับที่มีชนิดข้อมูลเป็น LPSTR ซึ่งเป็นตัวชี้ไปยังชื่อของเขตข้อมูลซึ่งต้องการจะแทรกค่าข้อมูล โดยฟังก์ชันจะแทรกค่าของเขตข้อมูลลงในตาราง ตามลำดับของชื่อเขตข้อมูลที่ระบุไว้ในตัวแปรแถวลำดับและสิ้นสุดการทำงานเมื่อค่าของตัวชี้ในตัวแปรแถวลำดับมีค่าเป็น 0 ดังตัวอย่างดังนี้ LPSTR fname[3] = { "name", "weight", 0 };
value	เป็นตัวชี้ซึ่งชี้ไปยังแถวลำดับที่มีชนิดข้อมูลเป็น LPSTR ซึ่งเป็นตัวชี้ไปยังค่าของเขตข้อมูลซึ่งต้องการจะแทรก ดังตัวอย่างดังนี้ LPSTR value[2] = { "ARNUPHAB", "67.5" };
ค่าที่ส่งกลับ	1 - การแทรกรายการลงในตารางสำเร็จ 0 - การแทรกรายการลงในตารางไม่สำเร็จ

4) คำสั่งในการเปลี่ยนแปลงค่าเขตข้อมูลของรายการลงในตารางรีเลชันนัล

```
int UpdateCol( LPSTR table_name, LPSTR column_name,  
LPSTR lpval, LPSTR condition );
```

เป็นคำสั่งที่ใช้ในการเปลี่ยนแปลงค่าเขตข้อมูลของรายการในตารางรีเลชันนัลในฐานข้อมูลซึ่งโปรแกรมกำลังเชื่อมต่ออยู่ มีรายละเอียดของพารามิเตอร์ ดังนี้

table_name	ใช้ระบุชื่อตารางที่ต้องการเปลี่ยนแปลงค่าของเขตข้อมูล
column_name	ระบุชื่อเขตข้อมูลที่ต้องการเปลี่ยนแปลงค่าข้อมูล
lpvalue	ค่าข้อมูล
condition	เงื่อนไขที่ใช้ในการค้นหารายการในตารางที่ต้องการเปลี่ยนแปลงค่า มีลักษณะเป็นนิพจน์ตรรกะ ซึ่งประกอบด้วยนิพจน์เปรียบเทียบและอาจมีการเชื่อมต่อนิพจน์ด้วย AND. และ OR. เช่น LPSTR condition = "weight > 40 .AND. weight < 70";

ค่าที่ส่งกลับ	1 - การเปลี่ยนแปลงค่าสำเร็จ
	0 - การเปลี่ยนแปลงค่าไม่สำเร็จ
	-1 - การเปลี่ยนแปลงค่าเกิดความผิดพลาด

5) คำสั่งในการลบรายการในตารางรีเลชันนัล

```
int DeleteRow( LPSTR table_name, LPSTR condition )
```

เป็นคำสั่งที่ใช้ในการลบรายการออกจากตารางรีเลชันนัลในฐานข้อมูลซึ่งโปรแกรมกำลังเชื่อมต่ออยู่ มีรายละเอียดของพารามิเตอร์ ดังนี้

table_name	.ใช้ระบุชื่อตารางที่ต้องการลบรายการ
condition	เงื่อนไขที่ใช้ในการค้นหารายการในตารางที่ต้องการลบออกจากตาราง มีลักษณะเป็นนิพจน์ตรรกะ ซึ่งประกอบด้วยนิพจน์เปรียบเทียบและอาจมีการเชื่อมต่อนิพจน์ด้วย .AND. และ .OR. เช่น LPSTR condition = "weight > 40 .AND. weight < 70"

ค่าที่ส่งกลับ	1 - การลบรายการสำเร็จ
	0 - การลบรายการไม่สำเร็จ
	-1 - การลบรายการเกิดความผิดพลาด

6) คำสั่งในการค้นคืนรายการข้อมูลรายการแรกที่มีค่าข้อมูลตรงตามเงื่อนไข

```
int SelectRow( LPSTR table_name, LPSTR *fname,  
              LPSTR *value, LPSTR condition );
```

เป็นคำสั่งที่ใช้ในการค้นคืนรายการข้อมูลรายการแรกซึ่งมีค่าข้อมูลตรงตามเงื่อนไขที่ระบุไว้ในนิพจน์ condition จากตารางรีเลชันนัลในฐานข้อมูลซึ่งโปรแกรมกำลังเชื่อมต่ออยู่ มีรายละเอียดของพารามิเตอร์ ดังนี้

table_name	ใช้ระบุชื่อตารางที่ต้องการค้นคืนรายการข้อมูล
fname	เป็นตัวชี้ซึ่งชี้ไปยังแถวลำดับที่มีชนิดข้อมูลเป็น LPSTR ซึ่งเป็นตัวชี้ไปยังชื่อของเขตข้อมูลซึ่งต้องการจะค้นคืนค่าข้อมูล โดยฟังก์ชันจะค้นคืนค่าของเขตข้อมูลจากตาราง ตามลำดับของชื่อเขตข้อมูลที่ระบุไว้ในตัวแปรแถวลำดับและสิ้นสุดการทำงานเมื่อค่าของตัวชี้ในตัวแปรแถวลำดับมีค่าเป็น 0 ดังตัวอย่างดังนี้

value	LPSTR fname[3] = { "name", "weight", 0 }; เป็นตัวชี้ซึ่งชี้ไปยังแถวลำดับที่มีชนิดข้อมูลเป็น LPSTR ซึ่งเป็นตัวชี้ไปยังที่พัก (buffer) ซึ่งใช้เก็บค่าของเขตข้อมูลซึ่งต้องการจะค้นคืน
condition	เงื่อนไขที่ใช้ในการค้นหารายการในตารางที่ต้องการค้นคืนจากตาราง มีลักษณะเป็นนิพจน์ตรรกะ ซึ่งประกอบด้วยนิพจน์เปรียบเทียบและอาจมีการเชื่อมต่อนิพจน์ด้วย .AND. และ .OR.. เช่น LPSTR condition = "weight > 40 .AND. weight < 70"
ค่าที่ส่งกลับ	1 - การค้นคืนรายการสำเร็จ 0 - การค้นคืนไม่พบรายการที่ตรงตามเงื่อนไขที่ระบุ -1 - การค้นคืนรายการเกิดความผิดพลาด

7) คำสั่งในการค้นคืนรายการข้อมูลรายการถัดไปที่มีค่าข้อมูลตรงตามเงื่อนไข

```
int FetchNextRow( LPSTR table_name, LPSTR *fname,
                 LPSTR *value, LPSTR condition );
```

เป็นคำสั่งที่ใช้ในการค้นคืนรายการข้อมูลรายการถัดไป ซึ่งมีค่าข้อมูลตรงตามเงื่อนไขที่ระบุไว้ในนิพจน์ condition จากตารางวิเลชันนัลในฐานข้อมูลซึ่งโปรแกรมกำลังเชื่อมต่ออยู่ โดยจะต้องเรียกใช้งานฟังก์ชันนี้ต่อจากฟังก์ชัน SelectRow() ที่มีเงื่อนไขในการค้นคืนเดียวกัน มีรายละเอียดของพารามิเตอร์ ดังนี้

table_name	ใช้ระบุชื่อตารางที่ต้องการค้นคืนรายการข้อมูล
fname	เป็นตัวชี้ซึ่งชี้ไปยังแถวลำดับที่มีชนิดข้อมูลเป็น LPSTR ซึ่งเป็นตัวชี้ไปยังชื่อของเขตข้อมูลซึ่งต้องการจะค้นคืนค่าข้อมูล โดยฟังก์ชันจะค้นคืนค่าของเขตข้อมูลจากตาราง ตามลำดับของชื่อเขตข้อมูลที่ระบุไว้ในตัวแปรแถวลำดับและสิ้นสุดการทำงานเมื่อค่าของตัวชี้ในตัวแปรแถวลำดับมีค่าเป็น 0 ดังตัวอย่างดังนี้ LPSTR fname[3] = { "name", "weight", 0 };
value	เป็นตัวชี้ซึ่งชี้ไปยังแถวลำดับที่มีชนิดข้อมูลเป็น LPSTR ซึ่งเป็นตัวชี้ไปยังที่พัก (buffer) ซึ่งใช้เก็บค่าของเขตข้อมูลซึ่งต้องการจะค้นคืน
condition	เงื่อนไขที่ใช้ในการค้นหารายการในตารางที่ต้องการค้นคืนจากตาราง มีลักษณะเป็นนิพจน์ตรรกะ ซึ่งประกอบด้วยนิพจน์เปรียบเทียบและอาจมีการเชื่อมต่อนิพจน์ด้วย .AND. และ .OR.. เช่น LPSTR condition = "weight > 40 .AND. weight < 70"

ค่าที่ส่งกลับ 1 - การค้นคืนรายการสำเร็จ
 0 - การค้นคืนไม่พบรายการที่ตรงตามเงื่อนไขที่ระบุ
 -1 - การค้นคืนรายการเกิดความผิดพลาด

8) คำสั่งในการยกเลิกการเชื่อมต่อโปรแกรมจัดการข้อมูล

```
int CloseDBS( void );
```

เป็นฟังก์ชันที่ใช้ในการยกเลิกการเชื่อมต่อโปรแกรมประยุกต์กับโปรแกรมจัดการข้อมูล



ภาคผนวก ค
การใช้งานโปรแกรม

เพิ่มข้อมูลของโปรแกรม

เพิ่มข้อมูลที่จะต้องใช้ในการทำงานของโปรแกรม ประกอบด้วยเพิ่มต่างๆ ดังนี้

- 1) โปรแกรมช่วยออกแบบแบบจำลองข้อมูล
ERD.EXE
- 2) คลังโปรแกรมกราฟิกของชุดโปรแกรม Borland C++ for Windows
BWCC.DLL
- 3) คลังโปรแกรมที่ใช้ติดต่อกับโปรแกรมจัดการข้อมูล
BRIDGE.DLL
- 4) คลังโปรแกรมจัดการข้อมูล CodeBase
T4DEBUG.DLL
- 5) เพิ่มข้อมูลของโปรแกรมจัดการข้อมูล CodeBase ได้แก่
RELSHIP.DBF DATA.DBF SYSTEM.DBF ENTITY.DBF
ATTRIB.DBF DOMAIN.DBF ALLWVAL.DBF UNQKEY.DBF
KEYCOMP.DBF SYSARC.DBF RELEND.DBF

การเรียกใช้งานโปรแกรม

การเรียกใช้โปรแกรม ทำได้โดยการดำเนินการ (execute) โปรแกรม ERD.EXE บนจอภาพ จะปรากฏวินโดว์หลักของโปรแกรม

การเริ่มต้นสร้างโมเดลข้อมูลของระบบงานใหม่

- 1) เลือกรายการเลือก NEW จากรายการเลือกแบบดิ่งลง SYSTEM
- 2) บนจอภาพจะปรากฏกล่องคำตอบ สำหรับให้ป้อนข้อมูลที่เกี่ยวข้องกับระบบงานที่ต้องการสร้าง จากนั้นให้กดปุ่ม OK
- 3) บนจอภาพก็จะปรากฏวินโดว์ย่อยผุดขึ้นในวินโดว์หลัก แสดงส่วนพื้นที่ที่ใช้ในการวาดแผนภาพโมเดลข้อมูล และพร้อมที่จะรับข้อมูลจาก ผู้ใช้ในการสร้างโมเดลข้อมูลต่อไป

การสร้างเอนติตี้

- 1) เลื่อนตัวชี้ไปที่ปุ่มกดที่มีข้อความว่า ENTITY แล้วกดเลือกด้วยการกดปุ่มเมาส์ซ้าย
- 2) เลื่อนตัวชี้มาที่บริเวณพื้นที่วาดภาพในวินโดว์ย่อย ตัวชี้จะเปลี่ยนไปเป็นรูปดินสอ
- 3) เลื่อนตัวชี้ไปยังตำแหน่งที่ต้องการ แล้วกดปุ่มเมาส์ซ้ายค้างไว้ บนจอภาพจะปรากฏโครงร่างของภาพสี่เหลี่ยมซึ่งใช้แทนเอนติตี้ ผู้ใช้ยังคงสามารถเลื่อนตัวชี้ไปมาเพื่อวางตำแหน่งตามต้องการแล้วจึงปล่อยปุ่มเมาส์
- 4) หากผู้ใช้กดปุ่ม Ctrl ที่แปงเป็นอักขระร่วมกับการกดปุ่มเมาส์ซ้ายในขณะที่เลื่อนตัวชี้ โปรแกรมจะอนุญาตให้ผู้ใช้เลื่อน ตำแหน่งภาพในทิศทาง แนวนอน แนวตั้ง และแนวทแยงเท่านั้น
- 5) บนจอภาพจะปรากฏกล่องคำโต้ตอบสำหรับให้ผู้ใช้ ป้อนข้อมูลเกี่ยวกับเอนติตี้ที่สร้างขึ้น แล้วกดปุ่ม COMMIT
- 6) บนจอภาพจะปรากฏภาพเอนติตี้เป็นรูปสี่เหลี่ยมมุมมน โดยมีชื่อเอนติตี้อยู่ที่มุมบนซ้าย

การสร้างเอนติตี้ในลักษณะที่เป็นซิปไทป์ของเอนติตี้อื่นที่สร้างไว้ก่อนหน้า

- 1) เลื่อนตัวชี้ไปที่ปุ่มกดที่มีข้อความว่า ENTITY แล้วกดเลือกด้วยการกดปุ่มเมาส์ซ้าย
- 2) เลื่อนตัวชี้มาที่บริเวณพื้นที่วาดภาพ ในวินโดว์ย่อย ตัวชี้จะเปลี่ยนไปเป็นรูปดินสอ
- 3) เลื่อนตัวชี้ไปยังภายในภาพของเอนติตี้ที่ต้องการให้เป็นซูปเปอร์ไทป์ของซิปไทป์ที่กำลังสร้าง แล้วกดปุ่มเมาส์ซ้าย บนจอภาพจะปรากฏโครงร่างของภาพสี่เหลี่ยมซึ่งใช้แทนเอนติตี้ อยู่เหนือภาพของซูปเปอร์ไทป์แล้วจึงปล่อยปุ่มเมาส์
- 4) บนจอภาพจะปรากฏกล่องคำโต้ตอบสำหรับให้ผู้ใช้ ป้อนข้อมูลเกี่ยวกับซิปไทป์ที่สร้างขึ้น โดยที่ชื่อของซูปเปอร์ไทป์จะปรากฏอยู่ในกล่องคำโต้ตอบโดยอัตโนมัติ จากนั้นให้กดปุ่ม COMMIT
- 5) บนจอภาพจะปรากฏภาพซิปไทป์อยู่ภายในกรอบภาพของซูปเปอร์ไทป์ เป็นรูปสี่เหลี่ยมมุมมน โดยมีชื่อซิปไทป์อยู่ที่มุมบนซ้าย

การเปลี่ยนแปลงแก้ไขขนาดของภาพเอนติตี้

- 1) เลื่อนตัวชี้ไปที่ปุ่มกดที่มีข้อความว่า EDIT แล้วกดเลือกด้วยการกดปุ่มเมาส์ซ้าย

2) เลื่อนตัวชี้ไปชี้ภายในภาพของแอนิเมชันที่ต้องการแก้ไข แล้วกดปุ่มเมาส์ซ้าย บนจอภาพจะปรากฏรูปสี่เหลี่ยมเล็กๆ 8 รูป ล้อมรอบรูปซึ่งชี้แทนแอนิเมชัน

3) ผู้ใช้สามารถเปลี่ยนขนาดของภาพแอนิเมชัน โดยการเลื่อนตัวชี้ไปที่สี่เหลี่ยมเล็กๆ รูปใดรูปหนึ่ง แล้วกดปุ่มเมาส์ซ้ายค้างไว้ เลื่อนตัวชี้ไปมาจนได้ขนาดของภาพที่พอใจแล้วจึงปล่อยปุ่มเมาส์ ขณะที่เลื่อนตัวชี้ บนจอภาพจะปรากฏโครงร่างของภาพแอนิเมชันเลื่อนตามตัวชี้

4) หากผู้ใช้กดปุ่ม Ctrl ที่แผงแป้นอักขระร่วมกับการกดปุ่มเมาส์ซ้ายในขณะที่เลื่อนตัวชี้ โปรแกรมจะอนุญาตให้ผู้ใช้เปลี่ยนขนาดภาพในทิศทาง แนวนอน แนวตั้ง และแนวทแยงเท่านั้น

5) ในกรณีที่แอนิเมชันเป็นซับไทป์ของแอนิเมชันอื่น โปรแกรมจะอนุญาตให้ผู้ใช้เปลี่ยนขนาดของแอนิเมชันอยู่ภายในกรอบภาพของแอนิเมชันที่เป็นซูเปอร์ไพบ์ของแอนิเมชันนั้นเท่านั้น

การเปลี่ยนแปลงแก้ไขตำแหน่งของภาพแอนิเมชัน

1) เลื่อนตัวชี้ไปที่ปุ่มกดที่มีข้อความว่า EDIT แล้วกดเลือกด้วยการกดปุ่มเมาส์ซ้าย

2) เลื่อนตัวชี้ไปชี้ภายในภาพของแอนิเมชันที่ต้องการแก้ไข แล้วกดปุ่มเมาส์ซ้ายค้างไว้ บนจอภาพจะปรากฏรูปสี่เหลี่ยมเล็กๆ 8 รูป ล้อมรอบรูปซึ่งชี้แทนแอนิเมชัน

3) ผู้ใช้สามารถเปลี่ยนตำแหน่งของภาพแอนิเมชัน โดยการเลื่อนตัวชี้ไปยังตำแหน่งของภาพที่ต้องการ แล้วจึงปล่อยปุ่มเมาส์ ขณะที่เลื่อนตัวชี้ บนจอภาพจะปรากฏโครงร่างของภาพแอนิเมชันเลื่อนตามตัวชี้

4) หากผู้ใช้กดปุ่ม Ctrl ที่แผงแป้นอักขระร่วมกับการกดปุ่มเมาส์ซ้ายในขณะที่เลื่อนตัวชี้ โปรแกรมจะอนุญาตให้ผู้ใช้เลื่อนตำแหน่งภาพในทิศทาง แนวนอน แนวตั้ง และแนวทแยงเท่านั้น

5) ในกรณีที่แอนิเมชันเป็นซับไทป์ของแอนิเมชันอื่น โปรแกรมจะอนุญาตให้ผู้ใช้เปลี่ยนตำแหน่งของแอนิเมชันอยู่ภายในกรอบภาพของแอนิเมชันที่เป็นซูเปอร์ไพบ์ของแอนิเมชันนั้นเท่านั้น

การยกเลิกการเป็นซับไทป์ของแอนิเมชัน

1) เลื่อนตัวชี้ไปที่ปุ่มกดที่มีข้อความว่า EDIT แล้วกดเลือกด้วยการกดปุ่มเมาส์ซ้าย

2) เลื่อนตัวชี้ไปชี้ภายในภาพของแอนิเมชันที่เป็นซับไทป์ แล้วกดปุ่มเมาส์ซ้ายค้างไว้พร้อมกับกดปุ่ม

Shift ที่แฉงเป็นอักขระ ตัวชี้จะเปลี่ยนไปเป็นภาพมือ และบนจอภาพจะปรากฏภาพโครงร่างของเอนิติตินั้น

3) ผู้ใช้สามารถยกเลิกการเป็นซัพไทป์ของเอนิติตินั้น โดยการเลื่อนตัวชี้ออกมาจากรอบของซูเปอร์ไทป์ของเอนิติติดังกล่าว ไปยังบริเวณซึ่งต้องการให้เป็นตำแหน่งใหม่ของเอนิติตินั้น แล้วปล่อยปุ่มเมาส์ ขณะที่เลื่อนตัวชี้ บนจอภาพจะปรากฏโครงร่างของภาพเอนิติตีเลื่อนตามตัวชี้

การเปลี่ยนเอนิติตีใดๆ ให้เป็นซัพไทป์ของเอนิติตีอื่น

- 1) เลื่อนตัวชี้ไปที่ปุ่มกดที่มีข้อความว่า EDIT แล้วกดเลือกด้วยการกดปุ่มเมาส์ซ้าย
- 2) เลื่อนตัวชี้ไปที่ภายในภาพของเอนิติตีที่ต้องการ แล้วกดปุ่มเมาส์ซ้ายค้างไว้พร้อมกับกดปุ่ม Shift ที่แฉงเป็นอักขระ ตัวชี้จะเปลี่ยนไปเป็นภาพมือและบนจอภาพจะปรากฏภาพโครงร่างของเอนิติตินั้น
- 3) ผู้ใช้สามารถเปลี่ยนเอนิติตินั้นให้เป็นซัพไทป์ของเอนิติตีอื่น โดยการเลื่อนตัวชี้เข้ามาชี้ภายในกรอบของเอนิติตีที่ต้องการให้เป็นซูเปอร์ไทป์ของเอนิติติดังกล่าว แล้วปล่อยปุ่มเมาส์ ขณะที่เลื่อนตัวชี้ บนจอภาพจะปรากฏโครงร่างของภาพเอนิติตีเลื่อนตามตัวชี้

การเปลี่ยนแปลงข้อมูลในรายละเอียดที่เกี่ยวกับเอนิติตี

- 1) เลื่อนตัวชี้ไปที่ปุ่มกดที่มีข้อความว่า EDIT แล้วกดเลือกด้วยการกดปุ่มเมาส์ซ้าย
- 2) เลื่อนตัวชี้ไปที่ภายในภาพของเอนิติตีที่ต้องการ แล้วกดปุ่มเมาส์ขวา บนจอภาพจะปรากฏกล่องคำโต้ตอบสำหรับให้ผู้ใช้ป้อนข้อมูลเกี่ยวกับเอนิติตินั้น
- 3) เมื่อแก้ไขข้อมูลเป็นที่ต้องการแล้ว ให้กดปุ่ม COMMIT

การลบเอนิติตี

- 1) เลื่อนตัวชี้ไปที่ปุ่มกดที่มีข้อความว่า EDIT แล้วกดเลือกด้วยการกดปุ่มเมาส์ซ้าย
- 2) เลื่อนตัวชี้ไปที่ภายในภาพของเอนิติตีที่ต้องการลบ แล้วกดปุ่มเมาส์ซ้าย บนจอภาพจะปรากฏรูปสี่เหลี่ยมเล็กๆ 8 รูป ล้อมรอบรูปซึ่งใช้แทนเอนิติตี
- 3) เลือกรายการเลือก Delete จากรายการเลือกแบบดิ่งลง Edit เพื่อลบเอนิติตินั้น

การนิยามแอตทริบิว

- 1) เลื่อนตัวชี้ไปชี้ภายในภาพของเอนิตีที่ต้องการนิยามแอตทริบิว แล้วกดปุ่มเมาส์ขวา บนจอภาพ จะปรากฏกล่องคำโต้ตอบสำหรับให้ผู้ใช้ป้อนข้อมูลเกี่ยวกับเอนิตีนั้น
- 2) เลือกปุ่มกดที่มีข้อความ Attribute บนจอภาพจะปรากฏกล่องคำโต้ตอบซึ่งจะแสดงรายชื่อแอตทริบิวทั้งหมดของเอนิตีนั้น
- 3) เมื่อต้องการสร้างแอตทริบิวใหม่ในเอนิตีนั้น ให้กดปุ่ม New
- 4) เมื่อต้องการแก้ไขแอตทริบิวใดในเอนิตีนั้น ให้เลื่อนตัวชี้ไปยังรายชื่อของแอตทริบิวที่ต้องการ แล้วกดเลือกด้วยการกดปุ่มเมาส์ซ้าย จากนั้นเลื่อนตัวชี้ไปยังปุ่มกดซึ่งมีข้อความว่า Change แล้วกดเลือก
- 5) เมื่อต้องการลบแอตทริบิวใดในเอนิตีนั้น ให้เลื่อนตัวชี้ไปยังรายชื่อของแอตทริบิวที่ต้องการ แล้วกดเลือกด้วยการกดปุ่มเมาส์ซ้าย จากนั้นเลื่อนตัวชี้ไปยังปุ่มกดซึ่งมีข้อความว่า Delete แล้วกดเลือก
- 6) เมื่อต้องการจบการแก้ไขรายการแอตทริบิว เลือกปุ่มกดที่มีข้อความว่า Done

การสร้างรีเลชันชิประหว่างเอนิตี

- 1) เลื่อนตัวชี้ไปที่ปุ่มกดที่มีข้อความว่า Relationship แล้วกดเลือกด้วยการกดปุ่มเมาส์ซ้าย
- 2) เลื่อนตัวชี้มาที่บริเวณพื้นที่วาดภาพ ในวินโดว์ย่อย ตัวชี้จะเปลี่ยนไปเป็นรูปดินสอ
- 3) เลื่อนตัวชี้ไปยังเอนิตีที่ต้องการ กดปุ่มเมาส์ซ้ายหนึ่งครั้ง เมื่อผู้ใช้เลื่อนตัวชี้ บนจอภาพ จะปรากฏโครงร่างของเส้นตรงซึ่งใช้แทนรีเลชันชิป ผู้ใช้สามารถจบการสร้างรีเลชันชิป โดยการเลื่อนตัวชี้ไปยังเอนิตีที่ต้องการให้มีรีเลชันชิปกับเอนิตีแรก แล้วกดปุ่มเมาส์ซ้ายอีกครั้ง
- 4) หากต้องการให้เส้นซึ่งแทนรีเลชันชิปเป็นเส้นหักมุม ก็สามารถทำได้โดยการกดปุ่มเมาส์ซ้าย ณ ตำแหน่งที่ต้องการให้เส้นตรงมีการหักมุม แล้วจึงค่อยจบการสร้างรีเลชันชิปโดยการกดปุ่มเมาส์ซ้ายบนเอนิตีที่ต้องการให้มีรีเลชันชิปกับเอนิตีแรก โปรแกรมจะอนุญาตให้ผู้ใช้สร้างจุดหักมุมบนเส้นรีเลชันชิปได้ไม่เกิน 3 จุด
- 5) หากผู้ใช้กดปุ่ม Ctrl ที่แฉงเป็นอัคระร่วมกับการกดปุ่มเมาส์ซ้ายในขณะที่เลื่อนตัวชี้ โปรแกรมจะอนุญาตให้ผู้ใช้ลากเส้นโครงร่างรีเลชันชิปในทิศทาง แนวนอน แนวตั้ง และแนวทแยงเท่านั้น

6) เมื่อผู้ใช้จบการสร้างรีเสชันชิป บนจอภาพจะปรากฏกล่องคำตอบสำหรับให้ผู้ใช้ ป้อนข้อมูลเกี่ยวกับรีเสชันชิปที่สร้างขึ้น

7) เมื่อผู้ใช้ป้อนข้อมูลที่เกี่ยวกับรีเสชันชิปแล้ว ให้กดปุ่ม OK

8) บนจอภาพจะปรากฏเส้นซึ่งแทนรีเสชันชิป โดยมีชื่อรีเสชันชิป ลักษณะเส้นอาจเป็นเส้นประ หรือมีเส้นดินกาศที่ปลายเส้น ขึ้นอยู่กับข้อมูลที่ผู้ใช้ได้ป้อนไว้ในข้อ 6

การเลื่อนตำแหน่งของจุดบนเส้นซึ่งแทนรีเสชันชิป

1) เลื่อนตัวชี้ไปชี้ที่ปุ่มกดที่มีข้อความว่า EDIT แล้วกดเลือกด้วยการกดปุ่มเมาส์ซ้าย

2) เลื่อนตัวชี้ไปชี้ตรงเส้นรีเสชันชิปที่ต้องการแก้ไข แล้วกดปุ่มเมาส์ซ้ายหนึ่งครั้ง บนจอภาพจะปรากฏรูปสี่เหลี่ยมเล็กๆ ที่จุดบนเส้นซึ่งแทนรีเสชันชิป

3) ผู้ใช้สามารถเปลี่ยนตำแหน่งของจุดบนเส้น โดยการเลื่อนตัวชี้ไปยังรูปสี่เหลี่ยมเล็กบนจุดที่ต้องการเลื่อน กดปุ่มเมาส์ซ้ายค้างไว้แล้วเลื่อนตัวชี้ไปยังตำแหน่งที่ต้องการ แล้วจึงปล่อยปุ่มเมาส์ขณะที่เลื่อนตัวชี้ บนจอภาพจะปรากฏโครงร่างของเส้นรีเสชันชิปเลื่อนตามตัวชี้

4) หากผู้ใช้กดปุ่ม C๗ ที่แผงแป้นอักขระร่วมกับการกดปุ่มเมาส์ซ้ายในขณะที่เลื่อนตัวชี้ โปรแกรมจะอนุญาตให้ผู้ใช้เลื่อนตำแหน่งจุดในทิศทาง แนวนอน แนวตั้ง และแนวทแยงเท่านั้น

5) โปรแกรมจะไม่อนุญาตให้ผู้ใช้เปลี่ยนตำแหน่งปลายรีเสชันชิปจากเอนติตีเดิม ไปที่เอนติตีอื่น

การเปลี่ยนแปลงแก้ไขข้อมูลที่เกี่ยวกับรีเสชันชิป

1) เลื่อนตัวชี้ไปชี้ที่ปุ่มกดที่มีข้อความว่า EDIT แล้วกดเลือกด้วยการกดปุ่มเมาส์ซ้าย

2) เลื่อนตัวชี้ไปชี้ตรงเส้นรีเสชันชิปที่ต้องการแก้ไข แล้วกดปุ่มเมาส์ขวาหนึ่งครั้ง บนจอภาพจะปรากฏกล่องคำตอบสำหรับให้ผู้ใช้แก้ไขข้อมูลรายละเอียดเกี่ยวกับรีเสชันชิปนั้น

3) เมื่อแก้ไขข้อมูลเป็นที่ต้องการแล้ว ให้กดปุ่ม OK

การนิยามคีย์เอกลักษณ์

- 1) เลื่อนตัวชี้ไปชี้ภายในภาพของเอนติตี้ที่ต้องการนิยามคีย์เอกลักษณ์ แล้วกดปุ่มเมาส์ขวา บนจอภาพ จะปรากฏกล่องคำโต้ตอบสำหรับให้ผู้ใช้ป้อนข้อมูลเกี่ยวกับเอนติตี้นั้น
- 2) เลือกปุ่มกดที่มีข้อความ Unique Key บนจอภาพจะปรากฏกล่องคำโต้ตอบซึ่งจะแสดงรายการคีย์เอกลักษณ์ทั้งหมดของเอนติตี้นั้น
- 3) เมื่อต้องการสร้างคีย์เอกลักษณ์ใหม่ในเอนติตี้นั้น ให้กดปุ่ม New Key
- 4) เมื่อต้องการแก้ไขคีย์เอกลักษณ์ใดในเอนติตี้นั้น ให้เลื่อนตัวชี้ไปยังรายการของคีย์เอกลักษณ์ที่ต้องการ แล้วกดเลือกด้วยการกดปุ่มเมาส์ซ้าย จากนั้นเลื่อนตัวชี้ไปยังปุ่มกดซึ่งมีข้อความว่า Edit Key แล้วกดเลือก
- 5) เมื่อต้องการลบคีย์เอกลักษณ์ใดในเอนติตี้นั้น ให้เลื่อนตัวชี้ไปยังรายการของคีย์เอกลักษณ์ที่ต้องการ แล้วกดเลือกด้วยการกดปุ่มเมาส์ซ้าย จากนั้นเลื่อนตัวชี้ไปยังปุ่มกดซึ่งมีข้อความว่า Delete Key แล้วกดเลือก
- 6) เมื่อต้องการจบการแก้ไขรายการคีย์เอกลักษณ์ เลือกปุ่มกดที่มีข้อความว่า Done

การเก็บบันทึกโมเดลของระบบงาน

เลือกรายการเลือก Save จากรายการเลือกดังลง System

การเก็บบันทึกโมเดลของระบบงานโดยเปลี่ยนไปเป็นชื่อระบบงานอื่น

เลือกรายการเลือก Save as จากรายการเลือกดังลง System บนจอภาพจะปรากฏกล่องคำโต้ตอบให้ผู้ใช้ป้อนชื่อ และหมายเลขรุ่นที่ต้องการบันทึก แล้วกดปุ่ม OK

การเรียกโมเดลระบบงานที่จัดเก็บไว้มาใช้งาน

เลือกรายการเลือก Open จากรายการเลือกดังลง System บนจอภาพจะปรากฏกล่องคำโต้ตอบแสดงรายชื่อโมเดลระบบงานที่จัดเก็บไว้ ให้ผู้ใช้เลื่อนตัวชี้ไปยังชื่อโมเดลที่ต้องการแล้วกดปุ่มเมาส์ซ้าย จากนั้นจึงกดปุ่ม OK

การลบโมเดลระบบที่กำลังใช้งานออกจากฐานข้อมูล

เลือกรายการเลือก Remove จากรายการเลือกแบบดึงลง System

การสร้างตารางรีเส็นของพจนานุกรมข้อมูลในฐานข้อมูลที่โปรแกรมเชื่อมต่ออยู่

- 1) เลื่อนตัวชี้ไปยังปุ่มกดเมนูระบบที่มุมบนซ้ายของวินโดว์หลัก แล้วกดเลือก
- 2) เลือกรายการเลือก Init DD

การสร้างรายงานโมเดลข้อมูลในลักษณะคำบรรยาย

- 1) เลือกรายการเลือก Report จากเมนูแบบดึงลง Document บนจอภาพจะปรากฏกล่องคำ
โต้ตอบให้ผู้ใช้ป้อนชื่อเพิ่มข้อมูลที่ต้องการจัดเก็บรายงาน
- 2) เมื่อป้อนชื่อเพิ่มข้อมูลแล้ว ให้กดปุ่ม OK

การสร้างคำสั่งนิยามข้อมูลของโมเดลข้อมูล

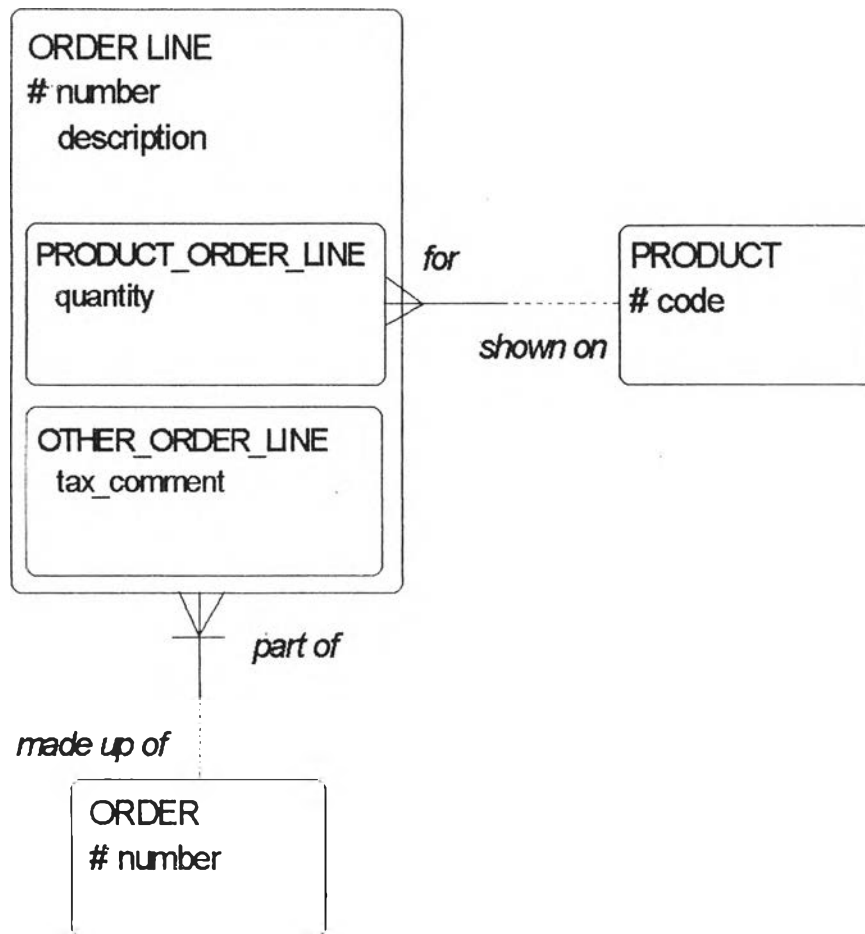
- 1) เลือกรายการเลือก Generate DDL จากเมนูแบบดึงลง Document บนจอภาพจะ
ปรากฏกล่องคำโต้ตอบให้ผู้ใช้ป้อนชื่อเพิ่มข้อมูลที่ต้องการจัดเก็บคำสั่งนิยามข้อมูล พร้อมทั้งให้ผู้ใช้เลือกทาง
เลือกในการสร้างคำสั่งในแบบ รวมทั้งทุกฉบับใหม่เป็นตารางเดียว หรือ แยกตารางสำหรับฉบับใหม่ในระดับบนสุด
- 2) เมื่อป้อนชื่อเพิ่มข้อมูลและเลือกทางเลือกแล้ว ให้กดปุ่ม OK

การสร้างแผนภาพโมเดลข้อมูล

- 1) เลือกรายการเลือก Diagram จากเมนูแบบดึงลง Document บนจอภาพจะปรากฏกล่อง
คำโต้ตอบให้ผู้ใช้จัดเตรียม (set up) เครื่องพิมพ์ที่จะใช้พิมพ์ รวมทั้งขนาดของกระดาษที่จะพิมพ์
โดยผู้ใช้จะต้องระบุตำแหน่งเริ่มต้นที่จะพิมพ์ด้วย
- 2) เมื่อผู้ใช้จัดเตรียมเครื่องพิมพ์เสร็จแล้ว ให้กดปุ่ม OK

ภาคผนวก ง.
ตัวอย่างผลที่ได้จากโปรแกรม

ตัวอย่างแผนภาพที่ได้จากโปรแกรม



ตัวอย่างรายงานที่สร้างโดยโปรแกรม

Date : 21/ 4/1994

-----SYSTEM INFORMATION-----

Page : 1

System name : ORDER

Version : 1

Title : use for test

Date :

-----DESCRIPTION-----

use for test

-----OBJECTIVE-----

Remarks :

Date : 21/ 4/1994

ENTITY DESCRIPTION
System ORDER Version 1

Page : 2

Entity name : ORDER_LINE

Abbreviate : OL

Subtype of :

Initial volumn :

Average volumn :

Maximum volumn :

Annual growth (<) :

-----DESCRIPTION-----

-----ATTRIBUTES-----

* NUMBER

Format : CHAR

Length : 10

Domain :

Mandatory : Y

DESC

Format : CHAR

Length : 30

Domain :

Mandatory : N

-----RELATIONSHIPS-----

* MUST BE part of

ONLY ONE ORDER.

< "*" means "is a part of the primary unique key" >

Date : 21/ 4/1994

ENTITY DESCRIPTION
System ORDER Version 1

Page : 3

Entity name : PROD_ORDER_LINE

Abbreviate : POL

Subtype of : ORDER_LINE

Initial volumn :

Average volumn :

Maximum volumn :

Annual growth (<) :

-----DESCRIPTION-----

-----ATTRIBUTES-----

QUANTITY

Format : NUM

Length : 10

Domain :

Mandatory : N

-----RELATIONSHIPS-----

MUST BE for

ONLY ONE PRODUCT

< "*" means "is a part of the primary unique key" >

Date : 21/ 4/1994

Page : 4

ENTITY DESCRIPTION
System ORDER Version 1

Entity name : OTHR_ORDER_LINE	Initial volumn :
Abbreviate : OOL	Average volumn :
Subtype of : ORDER_LINE	Maximum volumn :
	Annual growth (<) :

-----DESCRIPTION-----

-----ATTRIBUTES-----

COMMENT		Domain :
Format : CHAR	Length : 30	Mandatory : N

-----RELATIONSHIPS-----
< "*" means "is a part of the primary unique key" >

Date : 21/ 4/1994

Page : 5

ENTITY DESCRIPTION
System ORDER Version 1

Entity name : PRODUCT	Initial volumn :
Abbreviate : PRD	Average volumn :
Subtype of :	Maximum volumn :
	Annual growth (<) :

-----DESCRIPTION-----

-----ATTRIBUTES-----

* CODE		Domain :
Format : CHAR	Length : 10	Mandatory : Y

-----RELATIONSHIPS-----
MAY BE shown on MANY PROD_ORDER_LINE
< "*" means "is a part of the primary unique key" >

Date : 21/ 4/1994

Page : 6

ENTITY DESCRIPTION
System ORDER Version 1

Entity name : ORDER.	Initial volumn :
Abbreviate : ORD	Average volumn :
Subtype of :	Maximum volumn :
	Annual growth (<) :

-----DESCRIPTION-----

-----ATTRIBUTES-----

* NUMBER		Domain :
Format : CHAR	Length : 10	Mandatory : Y

-----RELATIONSHIPS-----
MAY BE made up of MANY ORDER_LINE
< "*" means "is a part of the primary unique key" >

ตัวอย่างโครงสร้างฐานข้อมูลแอพลิเคชันที่สร้างโดยโปรแกรม

```

CREATE VIEW ORDER_LINE AS
  SELECT
    NUMBER,
    DESC,
    ORD_NUMBER
  FROM OTHR_ORDER_LINE
 UNION
  SELECT
    NUMBER,
    DESC,
    ORD_NUMBER
  FROM PROD_ORDER_LINE

CREATE TABLE PRODUCT (
  CODE                CHAR(10)      NOT NULL,
  PRIMARY KEY (CODE)
)

CREATE TABLE ORDER. (
  NUMBER              CHAR(10)      NOT NULL,
  PRIMARY KEY (NUMBER)
)

CREATE TABLE OTHR_ORDER_LINE (
  NUMBER              CHAR(10)      NOT NULL,
  DESC                CHAR(30),
  ORD_NUMBER          CHAR(10)      NOT NULL,
  OOL_COMMENT         CHAR(30),
  PRIMARY KEY (NUMBER,ORD_NUMBER),
  FOREIGN KEY (ORD_NUMBER) REFERENCES ORDER. ON DELETE RESTRICT
)

CREATE TABLE PROD_ORDER_LINE (
  NUMBER              CHAR(10)      NOT NULL,
  DESC                CHAR(30),
  ORD_NUMBER          CHAR(10)      NOT NULL,
  POL_QUANTITY        NUM(10,2),
  POL_PRD_CODE        CHAR(10)      NOT NULL,
  PRIMARY KEY (NUMBER,ORD_NUMBER),
  FOREIGN KEY (ORD_NUMBER) REFERENCES ORDER. ON DELETE RESTRICT,
  FOREIGN KEY (POL_PRD_CODE) REFERENCES PRODUCT ON DELETE
  RESTRICT
)

```



ประวัติผู้เขียนวิทยานิพนธ์

นาย อานุภาพ เดียวสกุล เกิดเมื่อวันที่ 2 มกราคม พ.ศ. 2509 ที่จังหวัดสตูล สำเร็จการศึกษาระดับปริญญาตรีวิทยาศาสตร์บัณฑิตจากภาควิชาคณิตศาสตร์ คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ในปีการศึกษา 2530 และเข้าศึกษาต่อในหลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิทยาศาสตร์คอมพิวเตอร์ ที่จุฬาลงกรณ์มหาวิทยาลัย เมื่อ พ.ศ. 2532