

# การสร้างรายงานด้วย Report Designer ขั้นสูง

ระหว่างวันที่ ๑๘ - ๒๐ มกราคม ๒๕๕๕

จัดโดย

บ.บางกอก เมดิคอล ซอฟต์แวร์ จำกัด ๒ ชั้น ๒ ม.๘ ซ.สุขสวัสดิ์ ๓๓

แขวง/เขต ราษฎร์บูรณะ กรุงเทพฯ

---

เรียบเรียงโดย

นายสุชาติ จันทร์วงษ์ นักวิชาการสาธารณสุขชำนาญการ

สำนักงานสาธารณสุขอำเภอเชียงใหม่

## หลักสูตรอบรมผู้ดูแลระบบ

การสร้างรายงานด้วย Report Designer ขั้นสูง

วัตถุประสงค์

๑. เพื่อเพิ่มศักยภาพให้ผู้เข้าร่วมอบรมมีความรู้ ความเข้าใจ และสามารถเลือกใช้เครื่องมือ (Components) ในกำหนดรูปแบบการแสดงผลรายงานในรูปแบบต่างๆ ที่มีอยู่ใน Report Designer ได้อย่างเหมาะสม

๒. เพื่อเพิ่มศักยภาพให้ผู้เข้าร่วมอบรมมีความรู้ ความเข้าใจ สามารถสร้าง แบบฟอร์ม และรายงาน สำหรับหน่วยบริการได้อย่างมีประสิทธิภาพสูงสุด

เนื้อหา

๑. การใช้ Sub Report ในการสร้างรายงาน เพื่อให้รายงานแสดงผลได้ดีและสวยงามยิ่งขึ้น

๒. การใช้ Crosstab ในการสร้างรายงานที่ซับซ้อนขึ้น

๓. การสร้างรายงานโดย การเขียน Script ในโปรแกรม Report Designer

๔. การเปรียบเทียบความสามารถและเทคนิคการแสดงผลรายงาน ด้วย Components

๕. การเปรียบเทียบความสามารถและเทคนิคการแสดงผลรายงาน ด้วยการเขียน Script ในรายงาน

เหมาะสำหรับ: ผู้ที่ผ่านการอบรมการสร้างรายงานด้วย Report Designer ขั้นประยุกต์ หรือผู้ที่มีทักษะ

ความรู้เรื่องคำสั่ง SQL การใช้งาน Report Designer และการเขียนโปรแกรม

ระยะเวลาในการอบรม ๓ วัน / รุ่ง

วันที่ ๑๘-๒๐ มกราคม ๒๕๕๕

สถานที่อบรม

บริษัท บางกอก เมดิคอล ซอฟต์แวร์ จำกัด

๒ ชั้น ๒ ม.๘ ซ.สุขสวัสดิ์ ๓๓ ถ.สุขสวัสดิ์ แขวง/เขต ราษฎร์บูรณะ กรุงเทพฯ

ค่าลงทะเบียนท่านละ 7,000 บาท (เจ็ดพันบาทถ้วน) (รวมค่าเอกสาร อาหารกลางวัน 3 มื้อ อาหารว่าง 6 มื้อ)

ตารางการอบรม

วันที่ ๑

- การใช้ Sub Report ในการสร้างรายงาน
- แบบฝึกหัด

วันที่ ๒

- การใช้ Crosstab ในการสร้างรายงาน
- แบบฝึกหัด

วันที่ ๓

- การเขียน Script ในรายงาน
- การเลือกใช้วิธีสร้างรายงานให้เหมาะสมกับรายงาน
- แบบฝึกหัด

# การใช้ Sub Report ในการสร้างรายงาน เพื่อให้รายงานแสดงผลได้ดีและสวยงามยิ่งขึ้น

## ตัวอย่างรายงาน

### รายงานทะเบียนการตรวจรักษาผู้ป่วย

ตั้งแต่วันที่

ถึงวันที่

ชื่อโรงพยาบาล

ลำดับ	วันที่	HN	ชื่อ - สกุล	รวมค่าใช้จ่าย
การวินิจฉัยโรค				
หัตถการ				
รายการยา				

## วิธีการสร้างรายงาน

### ๑. สร้างคำสั่ง SQL คำสั่งหลัก

```
Select CONCAT(pname,fname," ",lname) AS pname,o.*
```

```
From ovst o,patient pt
```

```
Where o.hn = pt.hn and o.vstdate Between "๒๐๑๑-๑๐-๐๑" and "๒๐๑๑-๑๐-๓๑"
```

ในเมนู SQL Query ของโปรแกรม HOSxP และทดลอง Run

The screenshot shows the HOSxP SQL Query interface. The query entered is: `Select CONCAT(pname, fname, ' ', lname) AS pname, o.* From ovst o, patient pt Where o.hn = pt.hn and o.vstdate Between '2011-10-01' and '2011-10-31' LIMIT 10`. The results table displays patient information including name, ID, date, and various medical codes.

pname	hos_guid	vn	hn	an	vstdate	vstime	doctor	hospr	hospr	oqueue	ovst	ovst	ptyt	ptypeno	rfric	rfrict	rfror	
ค.ช.ธีรพันธ์ สิท้าว	{05F8F302-3BA4-435B-9736-541001081950}	0047323			1/10/2554	08:19:50	0098	11179	06539		12	01	98	71	๗7130434741			
ค.ช.พิทวัส พลนชัย	{06246210-2DBC-482C-9CBC-541001094752}	0053718			1/10/2554	09:47:52	0125	11179	06538		39	01	00	71	๗7171516599			
นางสาวชนกภาณี วงษา	{0AE910F8-8A26-44EE-A599-541001075146}	0025338			1/10/2554	07:51:46	0125	11179	06537		9	01	00	73	๗7360917646			
ค.ช.สิริวุฒิ ริมคำ	{0D3FA71B-C50E-486A-B346-541001144210}	0052672		540001453	1/10/2554	14:42:10	0125	10716	10716		83	01	05	19	๗7164984290			
ค.ช.รัชชานนท์ นันทฤทธิ	{1625BA0D-11D9-4782-8DE4-541001094237}	0050295			1/10/2554	09:42:37	0098	11179	11179		36	01	98	71	๗7153647877			
ค.ญ.อัญญา ปิตรี	{186DD255-2189-47C6-A593-541001100149}	0037953			1/10/2554	10:01:49	0125	11179	06539		44	01	00	73	๗7362657201			
นางงุชเกษ สุขานะ	{18991F14-C797-430B-8E5F-541001092532}	0017605			1/10/2554	09:25:32	0125	11179	11179		33	01	99	82	๗8266170819			
นางกชิตาง มะลิละ	{1E3B0EC6-24CE-476D-BB5-541001224451}	0024462			1/10/2554	22:44:51	0125	11179	06537		95	01	54	89	8963931147			4
นายณัฐพล ชันทะพิม	{1FAE8D3B-938C-46B0-A58-541001103323}	0021636			1/10/2554	10:33:23	0011	11179	06539		57	01	98	88	8975548687			
นางกนิช ธรรมจักร์	{2198D648-7E02-454E-B359-541001084831}	0003393			1/10/2554	08:48:31	0125	11179	06537		21	01	99	89	33210634			

## ๒. การสร้างช่วงวันที่ที่ต้องการระบุ

### ๒.๑ การประกาศตัวแปรวันที่

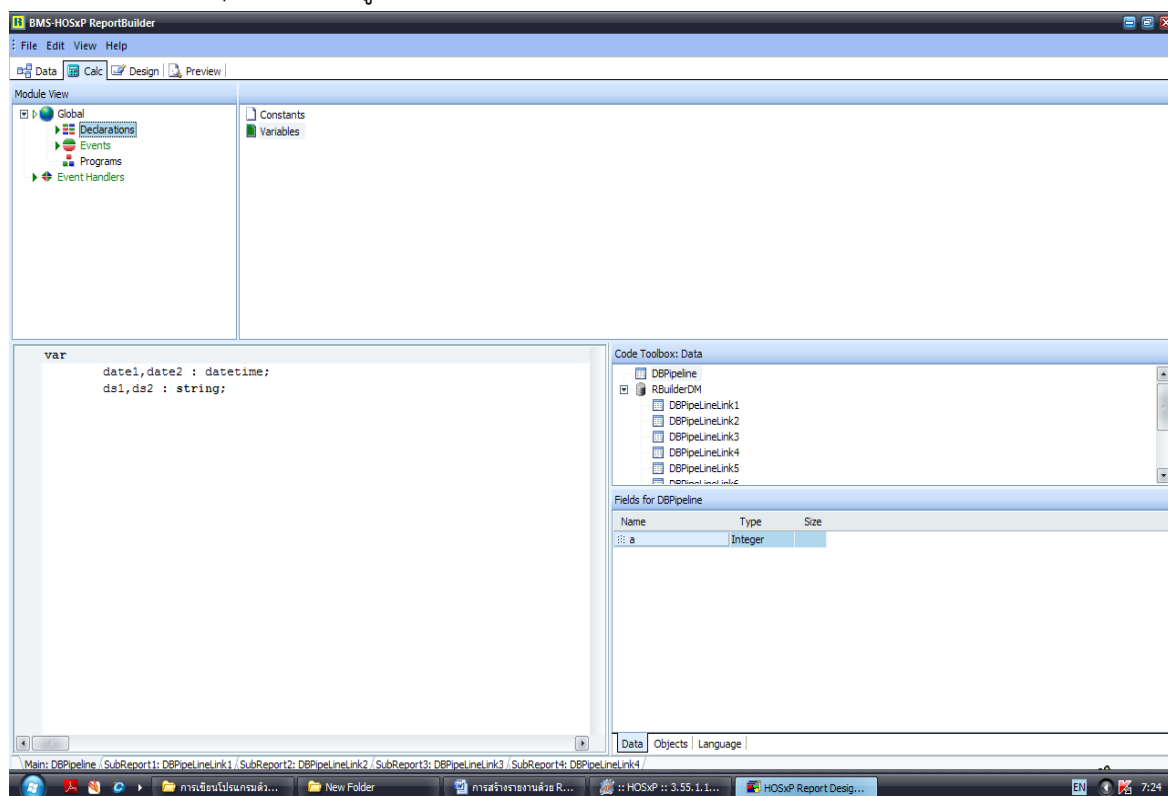
ที่แถบ Calc ประกาศตัวแปรวันที่โดยกำหนด และคลิก Declarations

var

```
date๑,date๒ : datetime;
```

```
ds๑,ds๒ : string;
```

เสร็จแล้วลอง Compile ตัวแปรดู



### ๒.๒ การสร้างกล่องวันที่ Dialog

ที่แถบ Calc เลือก Events เลือก On Create เขียนคำสั่งโดยดำเนินการดังนี้

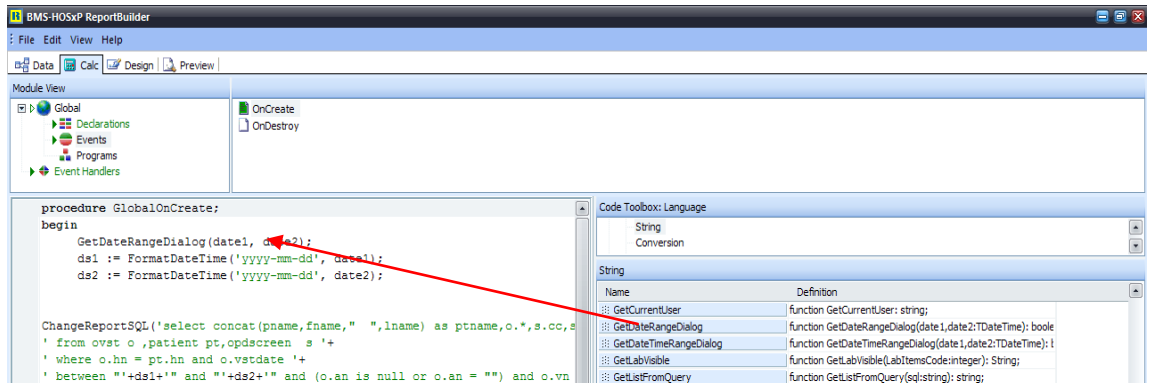
- ที่ Code Toolbox เลือก String
- เลือกแถบ Language เลือกฟังก์ชัน GetDateRangeDialog(date๑, date๒); หากมาวางใต้ begin และตัวแปร date๑ และdate๒ จะต้องตรงกับชื่อตัวแปร datetime ที่ประกาศใน var ในที่นี้เขียนได้ดังนี้  
GetDateRangeDialog(date๑, date๒);

### ๒.๓ เปลี่ยนรูปแบบวันที่ให้กลายเป็นรูปแบบ พ.ศ. โดยใช้ฟังก์ชัน FormatDateTime(Format, aDateTime);

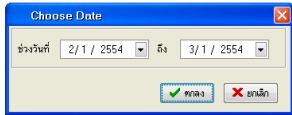
ที่ Code Toolbox เลือก Format เลือกแถบ Language เลือกฟังก์ชัน FormatDateTime(Format, aDateTime); หากไปวางใต้คำสั่ง ฟังก์ชัน GetDateRangeDialog(date๑, date๒); และสร้างตัวแปรมารับค่า โดยใช้ตัวแปรที่ได้ประกาศตัวแปรไว้ที่ Declarations ได้แก่ ds๑,ds๒ : string; ไปวางหน้าฟังก์ชัน ในที่นี้เขียนคำสั่งฟังก์ชันได้ดังนี้

```
ds๑ := FormatDateTime('yyyy-mm-dd', date๑);
```

```
ds๒ := FormatDateTime('yyyy-mm-dd', date๒);
```



- เสร็จแล้วลอง compile และ preview



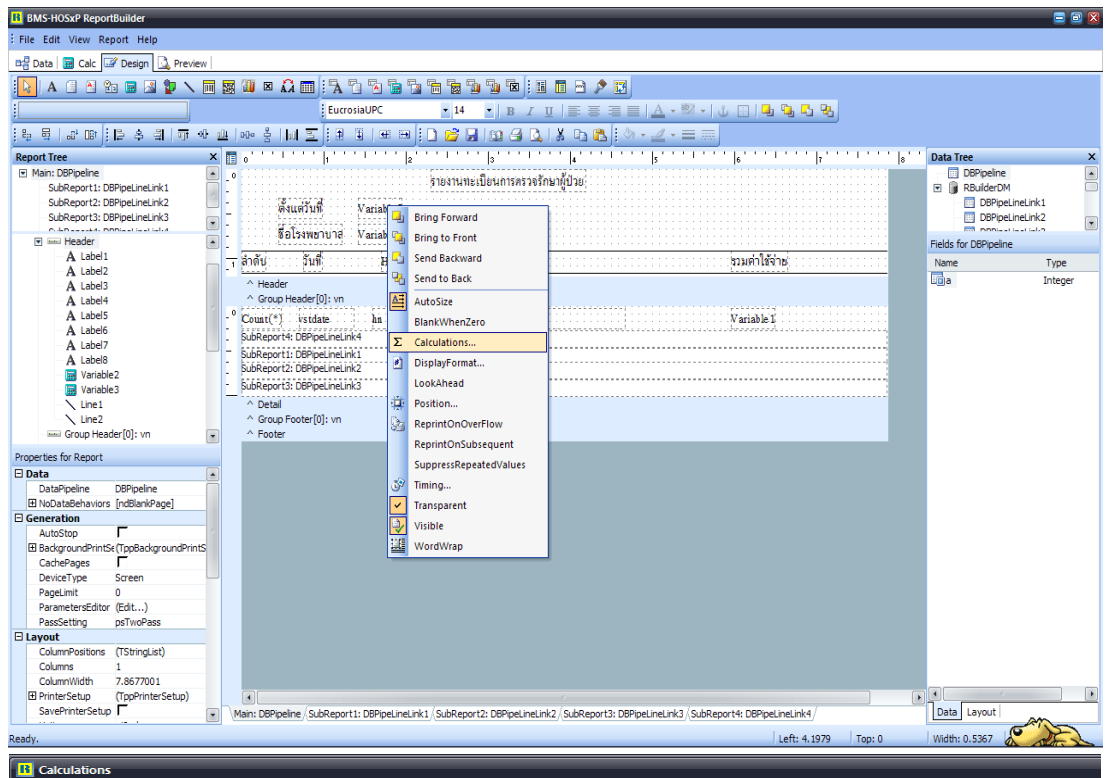
## ๒.๔ สร้างกล่องวันที่ในตัวรายงานที่หน้า Design

ใช้กล่องเครื่องมือ Variable มาวางตรงที่ต้องการช่วงวันที่ แล้วคลิกขวาที่ variable เลือก calculations... แล้วกำหนดค่าวันที่ให้ตรงตัวแปรที่เราต้องการ

เช่น ค่าแรกกำหนดเป็น Value := date๑; ค่าที่สองกำหนดเป็น Value := date๒; เสร็จแล้ว กำหนดชนิดข้อมูลเป็น date time ที่บริเวณมุมขวาบน

หรืออีกวิธีการสร้างวันที่โดยใช้รูปแบบการเชื่อมค่าโดยใช้ + ในการสร้างกล่อง variable เพียงกล่องเดียวแล้วกำหนดเป็น

Value := FormatThaiDate('D MMMM EEEE', date๑)+' ' +'ถึง'+ ' ' + FormatThaiDate('D MMMM EEEE', date๒);



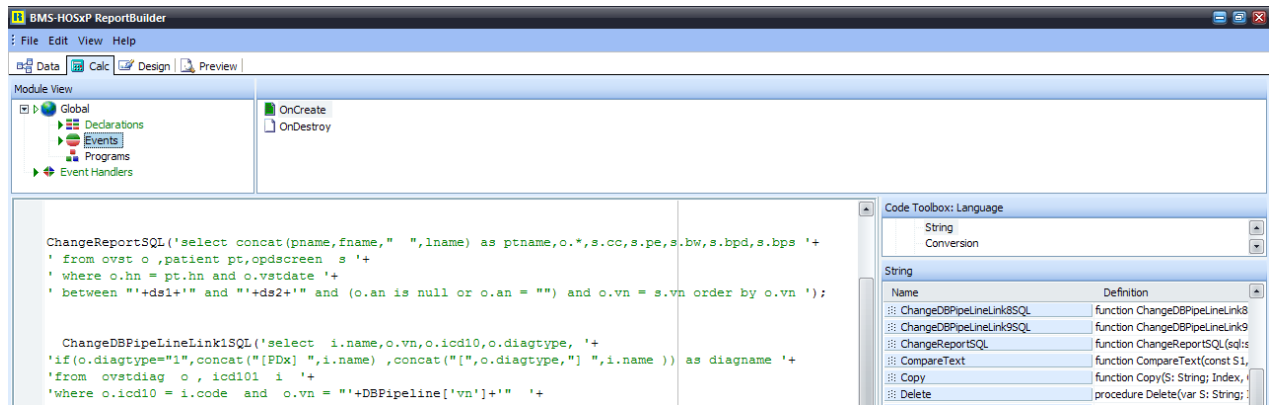
```
Value := FormatThaiDate('D MMMM EEEE', date1)+' ' +'ถึง'+ ' ' + FormatThaiDate('D MMMM EEEE', date2);
```

### ๓. การสร้างตัวแปรของข้อมูลที่ต้องการ

๓.๑ คัดลอกคำสั่งไปวางใน Calc ตามหลังฟังก์ชัน ChangeReportSQL(sql); โดยวางแทน sql โดยใส่เครื่องหมาย '' ก่อน แล้วดำเนินการเชื่อมค่า ผลที่ได้

```
ChangeReportSQL('select concat(pname,fname," ",lname) as pname,o.* '+  
' from ovst o ,patient pt '+  
' where o.hn = pt.hn and o.vstdate '+  
' between '"+ds๑+"' and '"+ds๒+"' and (o.an is null or o.an = '') order by o.vn ');
```

โดยวางถัดจากฟังก์ชันของวันที่ เสร็จแล้ว ลอง compile และ preview ดู จะได้ ที่แถบ data จะมี จะปรากฏชื่อ DBPipeline



๓.๒ ออกแบบรายงานในหน้า Design โดยดำเนินการดังนี้

- การสร้างลำดับที่ให้ RUN อัตโนมัติ โดยใช้เครื่องมือ DBcalc คลิกได้ชื่อลำดับในแถบ Detail แล้วคลิกขวาที่ variable เลือก calculations... Calc Type เลือก Count คลิก OK
- การสร้างวันที่ โดยใช้เครื่องมือที่ชื่อว่า DBtext คลิกวางในแถบ Detail ที่แถบ DBPipeline ข้างบน เลือก vstdate
- การสร้าง HN โดยใช้เครื่องมือที่ชื่อว่า DBtext คลิกวางในแถบ Detail ที่แถบ DBPipeline ข้างบนเลือก hn
- การสร้าง ชื่อ - สกุล โดยใช้เครื่องมือที่ชื่อว่า DBtext คลิกวางในแถบ Detail ที่แถบ DBPipeline ข้างบน เลือก pname
- การสร้าง รวมค่าใช้จ่าย โดยใช้เครื่องมือ Variable วาง แล้วคลิกขวา variable เลือก calculations... กำหนดค่าตัวแปร variable โดยที่ Code Toolbox เลือก Math เลือกฟังก์ชัน GetSQLDoubleData(sql); มาวางต่อจาก Value := และเขียนคำสั่ง sql เป็น Value := GetSQLDoubleData('select sum(sum\_price) from opitemrece where vn = '"+DBPipeline['vn']+"'"); อย่าลืมการเชื่อมค่าด้วย + และการกำหนดค่า vn = ให้ใช้ DBPipeline['vn'] จากแถบ data ลากมาวางแทนการพิมพ์ใน ""++"
- การเปลี่ยนค่า รวมค่าใช้จ่าย ให้เป็นรูปตัวเลขที่มีทศนิยม ๒ ตำแหน่ง โดยเปลี่ยนลักษณะข้อมูลเป็น Double คลิกขวาที่ Variable ของรวมค่าใช้จ่าย เลือก Display Format... เลือกลักษณะข้อมูลที่เป็นตัวเลขที่มีทศนิยม จำนวนทศนิยมสามารถพิมพ์เพิ่มได้

## การใช้ Sub Report

### ๔. การออกแบบการวินิจฉัยโรค

- นำข้อมูลที่ต้องการจากตาราง ovstdiag

- รูปแบบที่ต้องการแสดงผล

๑. รหัสโรคหลัก : [PDx] ชื่อโรค

๒. รหัสโรครอง : [ตัวเลข] ชื่อโรค หมายเลข ตัวเลขจะแสดงDiagType

- คำสั่งที่ต้องการ

```
Select i.name,o.vn,o.icd๑๐,o.diagtype,
```


```
IF (o.diagtype = "๑",CONCAT("[PDx] ",i.name),CONCAT("[",o.diagtype,"]",i.name)) AS
```

```
diagname
```

```
From ovstdiag o,icd๑๐๑ i
```

```
Where o.icd๑๐ = i.code and o.vn = "๕๒๑๑๑๔๐๓/๑๓/๕๓"
```

```
Order by o.diagtype
```

- การใช้แถบเครื่องมือ Sub Report  มาวางที่แถบ Detail ได้ข้อมูลที่ออกแบบไว้ก่อนแล้ว เสร็จแล้วไปกำหนดคุณสมบัติของข้อมูลโดยไปเลือกที่แถบ Properties For SubReport๑ ที่แถบ Data เลือก DataPipeline คลิกเลือก DBPipelineLink๑

- ไปที่ Calc เลือก Events เลือก On Create เขียนคำสั่งโดยดำเนินการดังนี้

๑. ที่ Code Toolbox เลือก String เลือกแถบ Language เลือกฟังก์ชัน

ChangeDBPipeLineLink๑SQL(sql); เสร็จแล้วนำคำสั่งที่ต้องการไปวางแทน sql (‘) และแก้ไขค่าตัวแปร vn โดยใช้ค่าของ "+DBPipeline['vn']+" มาวางแทน จะได้คำสั่งเป็น

```
ChangeDBPipeLineLink๑SQL('select i.name,o.vn,o.icd๑๐,o.diagtype, '+
```

```
'if(o.diagtype="๑",concat("[PDx] ",i.name) ,concat("[",o.diagtype,"]",i.name) ) as diagname '+
```

```
'from ovstdiag o , icd๑๐๑ i '+
```

```
'where o.icd๑๐ = i.code and o.vn = "+DBPipeline['vn']+" '+
```

```
'order by o.diagtype');
```

๒. เสร็จแล้ว Compile และ Preview ๑ รอบ สังเกตที่ Code Toolbox : Data เลือก

DBPipeLineLink๑ ที่แถบ Data บน Fields For DBPipeLineLink๑ จะมีข้อมูลอยู่

๓. ดำเนินการ Copy คำสั่งของ DBPipeLineLink๑ ไปวางที่ Calc คลิกขวาบริเวณที่ว่าง เลือก Event ที่แถบ Detail ของ Main : DBPipeLine คลิก BeforeGenerate เสร็จแล้วลอง compile และ preview ดู

- ดำเนินการออกแบบรายงานที่หน้า Design ที่ Sub Report๑: DBPipeLineLink๑ โดยใช้ DBtext และเลือกข้อมูลใน DBPipeLineLink๑ ให้ตรงตามค่าข้อมูลที่ต้องการ

๕. การออกแบบหัตถการ

- รูปแบบที่ต้องการแสดง

๑. หัตถการที่มาจาก ER ให้ขึ้นด้วยคำว่า ลำดับ [ER] ชื่อหัตถการ  
มาจากตาราง ER\_regist\_oper

๒. หัตถการที่มาจาก OPD ให้ขึ้นด้วยคำว่า ลำดับ [OPD] ชื่อหัตถการ  
มาจากตาราง doctor\_operation

- คำสั่งที่ต้องการ


๑. เขียนคำสั่งที่หัตถการมาจาก ER

```
select e.icd๙cm,o.er_oper_code,e.name,o.vn,concat("[ER] ",e.name) as o_name
from er_oper_code e
left outer join er_regist_oper o on e.er_oper_code = o.er_oper_code
where o.vn = "๕๔๑๑๑๑๑๑๑"
```

๒. เขียนคำสั่งที่หัตถการมาจาก OPD

```
select e.icd๙cm,o.er_oper_code,e.name,o.vn,concat("[OPD] ",e.name) as o_name from
er_oper_code e
left outer join doctor_operation o on e.er_oper_code = o.er_oper_code
where o.vn = "๕๔๑๑๑๑๑๑๑"
```

๓. นำคำสั่งที่ได้ทั้งสองมาเชื่อมกันโดยใช้ตัวเชื่อมเป็น Union

- การใช้แถบเครื่องมือ Sub Report  มาวางที่แถบ Detail ได้ข้อมูลที่ออกแบบไว้ก่อนแล้ว เสร็จแล้วไปกำหนดคุณสมบัติของข้อมูลโดยไปเลือกที่แถบ Properties For SubReport๒ ที่แถบ Data เลือก DataPipeline คลิกเลือก DBPipelineLink๒

- ไปที่ Calc เลือก Events เลือก On Create เขียนคำสั่งโดยดำเนินการดังนี้

๑. ที่ Code Toolbox เลือก String เลือกแถบ Language เลือกฟังก์ชัน ChangeDBPipeLineLink๒ SQL(sql); เสร็จแล้วนำคำสั่งที่ต้องการไปวางแทน sql ('') และแก้ไขค่าตัวแปร vn โดยใช้ค่าของ "+DBPipeline['vn']+" มาวางแทน จะได้คำสั่งเป็น

```
ChangeDBPipeLineLink๒SQL('select e.icd๙cm,o.er_oper_code,e.name,o.vn,concat("[ER] ",e.name)
as o_name '+
'from er_oper_code e '+
'left outer join er_regist_oper o on e.er_oper_code = o.er_oper_code '+
```



```
'where o.vn = ""+DBPipeline['vn']+"" '+
'union '+
'select e.icd๙cm,o.er_oper_code,e.name,o.vn,concat("[OPD] ",e.name) as o_name from
er_oper_code e '+
'left outer join doctor_operation o on e.er_oper_code = o.er_oper_code '+
'where o.vn = ""+DBPipeline['vn']+"" ');
```

๒. เสร็จแล้ว Compile และ Preview ๑ รอบ สังเกตที่ Code Toolbox : Data เลือก DBPipeLineLink ๒ ที่แถบ Data บน Fields For DBPipeLineLink๒ จะมีข้อมูลอยู่

๓. ดำเนินการ Copy คำสั่งของ DBPipeLineLink๒ ไปวางที่ Calc คลิกขวาบริเวณที่ว่าง เลือก Event ที่แถบ Detail ของ Main : DBPipeLine คลิก BeforeGenerate เสร็จแล้วลอง compile และ preview ดู

- ดำเนินการออกแบบรายงานที่หน้า Design ที่ Sub Report๑: DBPipeLineLink๒ โดยใช้ DBtext และ เลือกข้อมูลใน DBPipeLineLink๒ ให้ตรงตามค่าข้อมูลที่ต้องการ
- การดำเนินการให้ Sub Report๒ ไปต่อท้ายกับข้อมูลของ Sub Report๑ โดยคลิกขวาที่ Sub Report๒ เลือก ShiftRelativeTo... แล้วเลือกให้ต่อท้าย Sub Report๑

๖. การออกแบบรายการยา

- รูปแบบที่ต้องการแสดง

ลำดับ ชื่อ + ความแรง # จำนวนที่ใช้

มาจากตาราง drugitems และ opitemrece


- คำสั่งที่ต้องการ

```
select opitemrece.vn,concat(drugitems.name,"+",drugitems.strength,"#",
```

```
opitemrece.qty,drugitems.units) as d_name
```

```
from drugitems,opitemrece
```

```
where drugitems.icode = opitemrece.icode and opitemrece.vn = "๕๔๑๑๑๑๑๑๑"
```

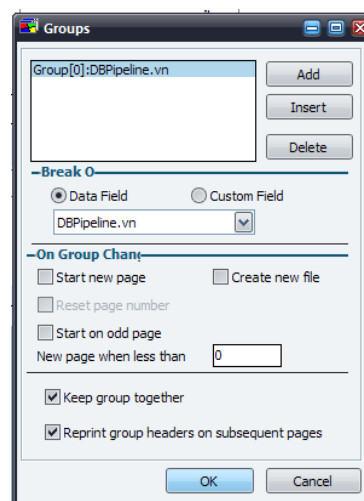
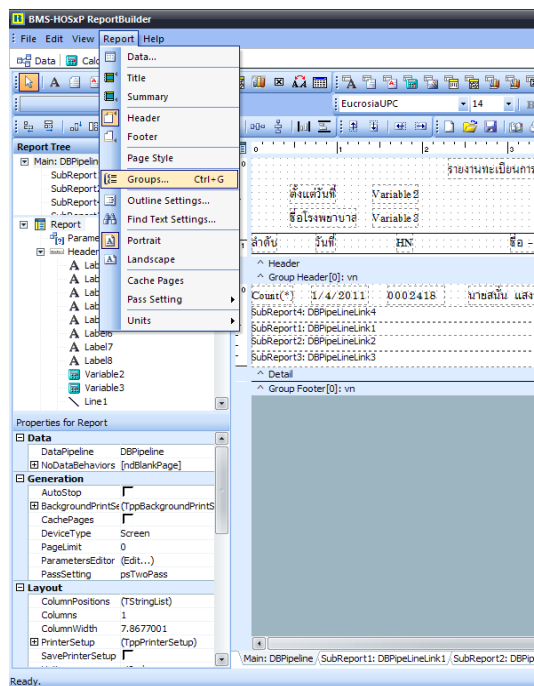
- การใช้แถบเครื่องมือ Sub Report  มาวางที่แถบ Detail ได้ข้อมูลที่ออกแบบไว้ก่อนแล้ว เสร็จแล้วไปกำหนดคุณสมบัติของข้อมูลโดยไปเลือกที่แถบ Properties For SubReport๓ ที่แถบ Data เลือก DataPipeline คลิกเลือก DBPipeLineLink๓

- ไปที่ Calc เลือก Events เลือก On Create เขียนคำสั่งโดยดำเนินการดังนี้

๑. ที่ Code Toolbox เลือก String เลือกแถบ Language เลือกฟังก์ชัน ChangeDBPipeLineLink๓ SQL(sql); เสร็จแล้วนำคำสั่งที่ต้องการไปวางแทน sql ('') และแก้ไขค่าตัวแปร vn โดยใช้ค่าของ ""+DBPipeline['vn']+"" มาวางแทน จะได้คำสั่งเป็น

ChangeDBPipeLineLink๓SQL('select opitemrece.vn,concat(drugitems.name,"+",  
drugitems.strength,"#",opitemrece.qty,drugitems.units) as d\_name '+  
'from drugitems,opitemrece '+  
' where drugitems.icode = opitemrece.icode and opitemrece.vn = '"+DBPipeLine['vn']+" '');

๒. เสร็จแล้ว Compile และ Preview ๑ รอบ สังเกตที่ Code Toolbox : Data เลือก DBPipeLineLink ๓ ที่แถบ Data บน Fields For DBPipeLineLink๓ จะมีข้อมูลอยู่
๓. ดำเนินการ Copy คำสั่งของ DBPipeLineLink๓ ไปวางที่ Calc คลิกขวาบริเวณที่วาง เลือก Event ที่ แถบ Detail ของ Main : DBPipeLine คลิก BeforeGenerate เสร็จแล้วลอง compile และ preview ดู
- ดำเนินการออกแบบรายงานที่หน้า Design ที่ Sub Report๓: DBPipeLineLink๑ โดยใช้ DBtext และ เลือกข้อมูลใน DBPipeLineLink๓ ให้ตรงตามค่าข้อมูลที่ต้องการ
๓๗. การสร้างเส้นให้แสดงโดยใช้การสร้างกรุปของข้อมูล ดำเนินการไปคลิกที่ Report บนแถบเมนูบาร์ เลือก Groups... ที่ปล๊อกของเมนู Groups ที่แถบ Break ๐ เลือก DBPipeLine:vn คลิก Add แล้ว Ok



จะได้ Group Footer [๐] : vn ให้สร้างเส้นใน Group นี้ เส้นจะปรากฏที่รายการสุดท้ายของเลขที่ VN นั้นๆ

๘. การสร้างกรอบของข้อมูล โดยใช้เครื่องมือ Region  ไปวางที่ SubReport ข้อมูลที่นำมาแสดงในกรอบนี้ได้แก่ อากาศสำคัญ อากาศแสดง ค่าความดันโลหิตสูง ค่าน้ำหนัก ส่วนสูง ฯลฯ โดยดำเนินการดังนี้

๑. สร้าง Sub Report ที่จะแสดงข้อมูล
๒. ออกแบบรายการข้อมูลที่ต้องการ โดยใช้เครื่องมือ DBtext และเลือกข้อมูลที่ DBPipeLineL

๓. คลิกที่เครื่องมือ Region มาวางทับข้อมูลที่ได้ออกแบบไว้ และคลิกขวาที่กรอบของ Region เลือก sent to black เพื่อให้แสดงข้อมูล

๔. ถ้าอยากให้ กรอบ Region ไม่แสดงในกรณีที่ไม่มีข้อมูล ดำเนินการคลิกเครื่องมือ Variable มาวางใกล้ๆของกรอบ Region คลิกขวาที่ Variable เลือก calculations... กำหนดค่าตัวแปร variable เช่น กรณีค่าของ CC ใช้ฟังก์ชัน

```
If DBText๙.text <>' ' then
    Region๒.visible :=True
else
    Region๒.visible :=False;
```

### ต้องการรายงาน

รายงานแสดงค่าใช้จ่ายในการรักษาพยาบาลโดยแยกตามสิทธิการรักษา

รหัส	สิทธิ	ค่าใช้จ่าย						
		ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.	เม.ย.

๑. เขียนคำสั่งหลัก

```
select o.pttype,pttype.name,sum(o.sum_price ) from opitemrece o, pttype ptt
where o.pttype = ptt.pttype and o.vstdate between "๒๐๐๙-๑๐-๐๑" and "๒๐๐๙-๑๐-๐๑"
group by o.pttype
```

๒. หรือจะใช้รูปแบบการเขียน script ดึงข้อมูลที่ต้องการมาเก็บไว้ชั่วคราวที่ TempReport แล้วค่อยดึงข้อมูลมาประมวลผลอีกที ดังนี้

```
unit Myptt; ตั้งชื่อ Scrip
procedure main; รูปแบบหลักที่ต้องมี
var ประกาศตัวแปร
mycds :TClientDataset;
Mycds๒:TClientDataset;
date๑, date๒ :Tdatetime;
ds๑,ds๒ :string;
```

```

begin
zquery.close; ให้ปิดรายงานเดิมที่เปิดไว้ก่อน
zquery.sql.text:='delete from tempreport where reportname="Myptt"; ให้ลบข้อมูลในรายงาน
เดิมก่อนเพื่อประหยัดหน่วยความจำ มีผลต่อความเร็วในการประมวลผล
zquery.execsql;
if not getdaterange() then exit
date๑ :=date_result๑();
date๒ :=date_result๒();
ds๑:=formatdatetime('YYY-MM-DD',date๑);
ds๒:=formatdatetime('YYY-MM-DD',date๒);
Mycds:=TClientDataset.Create(nil);
Mycds.data:=hosxp_getdataset('select o.pttype,ptt.name,o.sum_price,o.vstdate '+
' from opitemrece o,pttype ptt '+
' where o.pttype = ptt.pttype '+
' and o.vstdate between '"+ds๑+"' and '"+ds๒+"' ');
Mycds๒:= TClientDataset.Create(nil);
Mycds๒.data:=hosxp_getdataset('select * from tempreport '+
' where reportname="Myptt"');
Mycds.First;
Mycds๒.First;
while not Mycgs.EOF do
begin
Mycds๒.Insert;
Mycds๒.FieldName('reportname').AsString :='Myptt';
Mycds๒.FieldName('id').AsString :='Myptt';
Mycds๒.FieldName('name').AsString :=Mycds.FieldName('pttype').AsString;
Mycds๒.FieldName('name๑').AsString :=Mycds.FieldName('name').AsString;
Mycds๒.FieldName('mon').Asinteger :=Mycds.FieldName('sum_price').Asinteger;
Mycds๒.FieldName('date๑').Asdatetime :=Mycds.FieldName('vstdate').Asdatetime;
Mycds.Next;

```

```

Mycds๒.Next;

End;

If Mycds๒.changecount>๐ then

hosxp_updatedelta(Mycds๒.delta,'select * from tempreport where reportname="Myptt"');

Showmessage('Success');

end;

end.

```

๓. ดำเนินการออกแบบรายงานในหน้า Design

๔. การกำหนดปีงบประมาณ โดยดึงข้อมูลปีเฉพาะที่ต้องการออกมา ด้วยการใช้ฟังก์ชัน date\_format หรือ year รูปแบบ (date\_format(vstdate,"%Y")+๕๔๓ หรือ distinct(year(vstdate))+๕๔๓

๕. ดำเนินการเขียนคำสั่ง

```

begin
year:=GetPickupList('select distinct(year(date๑))+๕๔๓ as y '+
' from tempreport having y like "๒๕%" order by y desc');
showMessage(year);
byear :=StrToInt(year)-๕๔๔;
eyear :=StrToInt(year)-๕๔๓;
m๑๐๐ := encodedate(byear,๑๐,๐๑) ;
m๑๐๑ := encodedate(byear,๑๐,๓๑) ;
m๑๑๐ := encodedate(byear,๑๑,๐๑) ;
m๑๑๑ := encodedate(byear,๑๑,๓๐) ;
m๑๒๐ := encodedate(byear,๑๒,๐๑) ;
m๑๒๑ := encodedate(byear,๑๒,๓๑) ;
m๐๑๐ := encodedate(eyear,๐๑,๐๑) ;
m๐๑๑ := encodedate(eyear,๐๑,๓๑) ;
m๐๒๐ := encodedate(eyear,๐๒,๐๑) ;
m๐๒๑ := encodedate(eyear,๐๒,๒๙) ;
m๐๓๐ := encodedate(eyear,๐๓,๐๑) ;
m๐๓๑ := encodedate(eyear,๐๓,๓๑) ;
m๐๔๐ := encodedate(eyear,๐๔,๐๑) ;
m๐๔๑ := encodedate(eyear,๐๔,๓๐) ;
m๐๕๐ := encodedate(eyear,๐๕,๐๑) ;

```

```

m๐๕๑ := encodedate(eyear,๐๕,๑๑) ;
m๐๖๐ := encodedate(eyear,๐๖,๐๑) ;
m๐๖๑ := encodedate(eyear,๐๖,๑๐) ;
m๐๗๐ := encodedate(eyear,๐๗,๐๑) ;
m๐๗๑ := encodedate(eyear,๐๗,๑๑) ;
m๐๘๐ := encodedate(eyear,๐๘,๐๑) ;
m๐๘๑ := encodedate(eyear,๐๘,๑๑) ;
m๐๙๐ := encodedate(eyear,๐๙,๐๑) ;
m๐๙๑ := encodedate(eyear,๐๙,๑๐) ;
oct := '''+FormatDateTime('YYYY-MM-DD', m๑๐๐)''' and '''+FormatDateTime('YYYY-MM-DD', m๑๐๑)''';
nov := '''+FormatDateTime('YYYY-MM-DD', m๑๑๐)''' and '''+FormatDateTime('YYYY-MM-DD', m๑๑๑)''';
dec := '''+FormatDateTime('YYYY-MM-DD', m๑๒๐)''' and '''+FormatDateTime('YYYY-MM-DD', m๑๒๑)''';
jan := '''+FormatDateTime('YYYY-MM-DD', m๑๓๐)''' and '''+FormatDateTime('YYYY-MM-DD', m๑๓๑)''';
feb := '''+FormatDateTime('YYYY-MM-DD', m๑๔๐)''' and '''+FormatDateTime('YYYY-MM-DD', m๑๔๑)''';
mar := '''+FormatDateTime('YYYY-MM-DD', m๑๕๐)''' and '''+FormatDateTime('YYYY-MM-DD', m๑๕๑)''';
apr := '''+FormatDateTime('YYYY-MM-DD', m๑๖๐)''' and '''+FormatDateTime('YYYY-MM-DD', m๑๖๑)''';
may := '''+FormatDateTime('YYYY-MM-DD', m๑๗๐)''' and '''+FormatDateTime('YYYY-MM-DD', m๑๗๑)''';
jun := '''+FormatDateTime('YYYY-MM-DD', m๑๘๐)''' and '''+FormatDateTime('YYYY-MM-DD', m๑๘๑)''';
jul := '''+FormatDateTime('YYYY-MM-DD', m๑๙๐)''' and '''+FormatDateTime('YYYY-MM-DD', m๑๙๑)''';
aug := '''+FormatDateTime('YYYY-MM-DD', m๒๐๐)''' and '''+FormatDateTime('YYYY-MM-DD', m๒๐๑)''';
sep := '''+FormatDateTime('YYYY-MM-DD', m๒๑๐)''' and '''+FormatDateTime('YYYY-MM-DD', m๒๑๑)''';
octTosep := '''+FormatDateTime('YYYY-MM-DD', m๒๐๐)''' and '''+FormatDateTime('YYYY-MM-DD', m๒๑๑)''';
ChangeReportSQL('select name as pptype,name๑ as name,date๑ as vstdate,mon as sum_price '+
'from tempreport where reportname = "Myppt" and vstdate between '+octTosep+' group by pptype ');
end;

```

๖. ใช้ DBText ในการออกแบบรายงาน

## การใช้ Crosstab ในการสร้างรายงานที่ซับซ้อนขึ้น

### ตัวอย่างรายงาน

ต้องการรายงานแสดงจำนวนผู้มารับบริการแยกตามสิทธิการรักษา และแยกตามแพทย์ผู้รักษา

ดำเนินการดังนี้

๑. ข้อมูลที่ต้องการได้มาจากตาราง vn\_stat , opitemrece

๒. สร้างกล่องแสดงวันที่ที่ต้องการ

๓. เขียนคำสั่งหลักที่ต้องการ

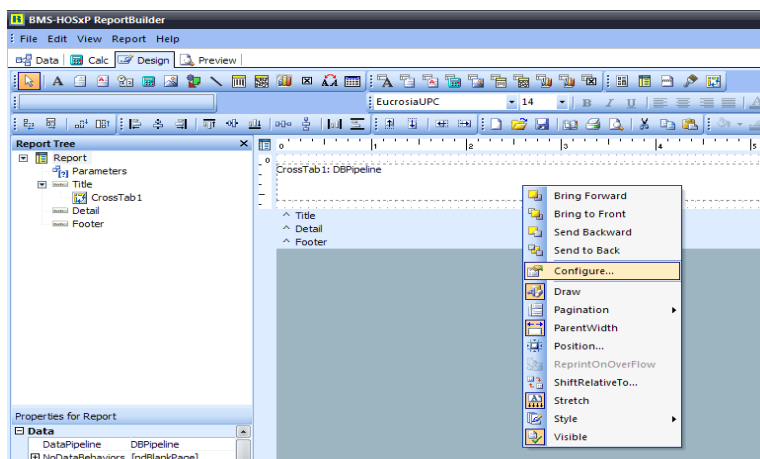
```
select v.pttype,v.dx_doctor,v.income,ptt.name as pttname,d.name as docname  
from vn_stat v,pttype ptt,doctor d  
where v.pttype=ptt.pttype and v.dx_doctor=d.code  
and v.vstdate between '"+ds๑+"' and '"+ds๒+"'  
order by d.code');
```

๔. ดำเนินการออกแบบรายงานในแถบ Design



คลิกที่ปุ่มเมนู Crosstab มาวางที่ Title โดยปกติรายงานที่สร้างจาก Crosstab จะนิยมสร้างในส่วน Title เพื่อให้แสดงผลครั้งเดียว ถ้าสร้างในส่วน Header หรือ Detail จะมีการวนลูปของรายงาน ทำให้มีการแสดงผลหลายครั้ง

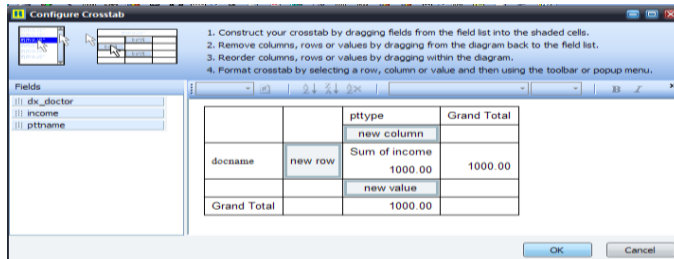
๕. คลิกขวาที่แถบ Cross Tab เลือก Configure...



ที่แถบเมนู Configure Crosstab ดำเนินการออกแบบโดยจะให้ข้อมูลอะไรไว้ในคอลัมน์ หรือข้อมูลอะไรไว้ในช่องแถว เช่นให้ข้อมูลรหัสสิทธิการรักษา แสดงในคอลัมน์ และข้อมูลแพทย์ผู้รักษาแสดงในแถว จะดำเนินการดังนี้

- ลาก docname วางที่ new row
- ลาก pttype วางที่ new column

- ลาก income มาวางตรงกลาง ระหว่าง docname กับ pptype ที่ new value ซึ่งโปรแกรมจะกำหนดค่าเริ่มต้นที่ Sum และสามารถแก้ไขได้
- ดำเนินแก้ไข ปรับปรุงแบบรายงานตามที่ต้องการ
- ลอง Preview ดู



BMS-HOSxP ReportBuilder

File Edit View Report Help

Data Calc Design Preview

100%

Sum of income	pptype																	Grand Total
docname	10	22	24	34	35	71	72	73	74	75	76	77	80	82	87	88	89	Grand Total
พญ.จุฑารัตน์ ใจโรจน์ไฉนยิด		1340											50					1390
พญ.พงศกร สิริสิทธิกร		4450											350					4800
นพ.เอื้อยงกูรย์ พิธิริมงคล	100	8364	755	314		290	1525		4767		745	15932		485		807	15300	49384
นางวนิดา ชินาอศ								1850									1050	2900
นางศุภาวรรณ น้อยอินทร์			50			1510		2260					275					4095
นางสาววนาลี ต๊ะทองง			550					2150			800							3500
พญ.กิตติยา ณาน	400	2857		246	95	457	4896		4047	570		2035	1645	189	271	2657		20365
พญ.นาฏนลิน ชลิตพงศ์	300	3039	454	130		115	2215		925			7111	203	189		540	7268	22489
พญ.มัลลิกา ณ ราช			3950		762	120	56		2286			941	1101	112	290		2697	12315
Grand Total	800	24600	1209	1452	95	2492	8692	6260	12025	570	1545	26419	1579	2431	479	1618	28972	121238

๖. การแก้ไขชื่อข้อมูลต่างๆต้องไปแก้ไขที่คำสั่งหลัก As.....



## การสร้างรายงานโดย การเขียน Script ในโปรแกรม Report Designer

### ทบทวน Pascal

#### รูปแบบของโปรแกรม Pascal

Program

Var

Begin

End.

๑. ส่วนหัวของโปรแกรม : Program Heading

ลักษณะที่ใช้

ใช้ในการตั้งชื่อของโปรแกรม

รูปแบบ

Program ชื่อโปรแกรม;

ตัวอย่าง

MyFirstProgram;

๒. ส่วนหัวการประกาศ : Declaration Part

ลักษณะที่ใช้

๑. ประกาศตัวแปรที่ใช้ในโปรแกรม (Variable Declaration)

รูปแบบ Var ชื่อตัวแปร : ชนิดตัวแปร;

ตัวอย่าง Var date๑,date๒ : datetme; ds๑,ds๒ : string;

๒. ประกาศค่าคงที่ (Constant Declaration)

รูปแบบ Const ชื่อค่าคงที่ = ค่าที่กำหนด;

ตัวอย่าง Const Addon = ๘๐;

๓. ประกาศชนิดของข้อมูลที่สร้างขึ้นใหม่ (Type Declaration)

ตัวอย่าง Type Color = (red,green,blue);

๔. ประกาศตัวแปรย่อยในรูปของ Procedure และ Function

๓. ส่วนคำสั่ง : Statement

ลักษณะที่ใช้

ใช้ในการเขียนคำสั่งประมวลผล

รูปแบบ

Begin

Statement\_๑

Statement\_๒

.....

Statement\_n

End.

ตัวอย่าง

```
Program MyFirstProgram
```

```
Begin
```

```
    ShowMessage('This is a Pascal Program');
```

```
End.
```

```
Program MyFirstProgram
```

```
Var IntegerVar : Integer;
```

```
Begin
```

```
    IntegerVar := ๓;
```

```
    ShowMessage(IntToStr(IntegerVar));
```

```
End.
```

### ชนิดของข้อมูล

๑. Integer เก็บข้อมูลแบบตัวเลขจำนวนเต็ม มีค่าระหว่าง -๓๒๗๖๘ ถึง ๓๒๗๖๗
๒. Real เก็บข้อมูลแบบตัวเลขจำนวนจริง ที่ประกอบด้วยตัวเลขจำนวนเต็มและทศนิยม
๓. Char เก็บข้อมูลเป็นตัวอักษร หรือตัวอักษร เพียง ๑ ตัว อาจจะเป็นตัวเลข, ตัวอักษร, สัญลักษณ์พิเศษ
๔. String เก็บข้อมูลที่เป็นข้อความ
๕. Boolean เก็บข้อมูลที่เป็นแบบตรรกศาสตร์ ที่แสดงถึงการตัดสินใจว่าข้อมูลนั้นจริง(True) หรือเท็จ(False)

### เครื่องหมายทางคณิตศาสตร์ (Arithmetic Operator)

- |   |           |         |
|---|-----------|---------|
| ๑. เครื่องหมายบวก                       | สัญลักษณ์ | +       |
| ๒. เครื่องหมายลบ                        | สัญลักษณ์ | -       |
| ๓. เครื่องหมายคูณ                       | สัญลักษณ์ | *       |
| ๔. เครื่องหมายหาร                       | สัญลักษณ์ | /       |
| ๕. เครื่องหมายหารตัดเศษ                 | สัญลักษณ์ | div     |
| ๖. เครื่องหมายหารเอาแต่เศษ              | สัญลักษณ์ | mod     |
| ๗. ฟังก์ชันเพื่อคำนวณการยกกำลังสองของ X | สัญลักษณ์ | sqr(x)  |
| ๘. ฟังก์ชันการหารากที่สองของ x          | สัญลักษณ์ | sqrt(x) |

## Compound Statement

รูปแบบ

```
IF Comdition๑ Then
    Begin
        ....
    End // end ตัวแรกไม่ต้องมีเครื่องหมาย;
Else
    Begin
        ....
    End;
```

## การทำงานแบบวนรอบ หรือทำซ้ำ (LOOP)

เป็นการเขียนคำสั่งเพื่อให้โปรแกรมวนทำงานซ้ำคำสั่งเดิม โดยการกำหนดเงื่อนไขในการวนรอบทำงานจนกว่าเงื่อนไขไม่เป็นจริง จึงจะหยุด

คำสั่ง WHILE

รูปแบบ

```
While Condition do
    Statement;
หรือ
While Condition do
    Begin
        Statement๑;
        Statement๒;
        .....
        Increment/Decrement;
    End;
```

## Data set Component

คือชุดเครื่องมือที่ใช้จัดการชุดข้อมูล เช่น

- DBMS : Database Management System คือ โปรแกรมที่ทำหน้าเป็นตัวกลางระหว่างผู้ใช้ (user) กับฐานข้อมูลเพื่อจัดการและควบคุมความถูกต้อง ความซ้ำซ้อน และความสัมพันธ์ระหว่างข้อมูลต่างๆ ภายในฐานข้อมูล  
เช่น MySQL, Postgresql
- Connection Component  
เช่น  
TMy Connection จะใช้บ่อย  
TSQL Connection จะใช้ติดต่อกับฐานข้อมูล

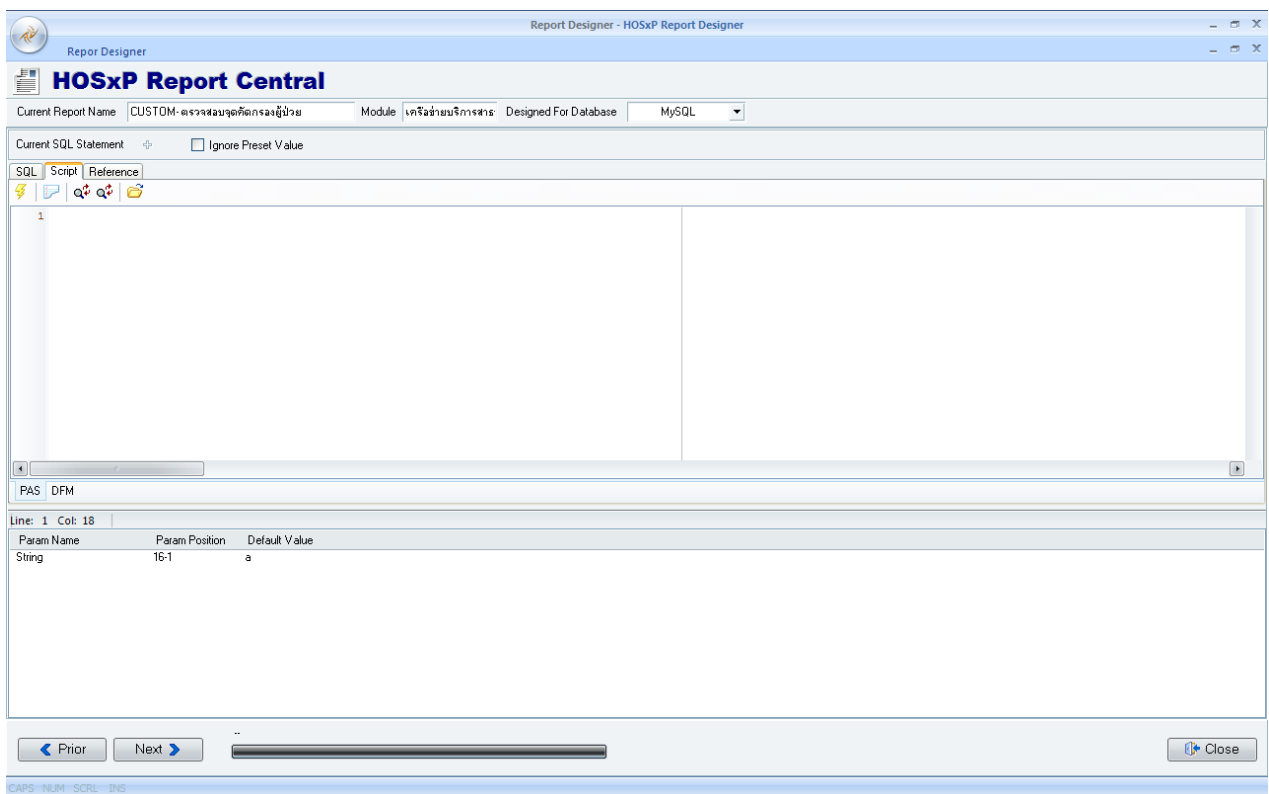
- Dataset Component จะใช้ในการสั่งงาน เช่น TMy Query , TSQL Dataset

### ชนิดของ Dataset

1. รูปแบบที่ข้อมูลจะดำเนินการเปลี่ยนแปลงทันทีเมื่อใช้งานและจำเป็นต้องติดต่อ(Connect) ฐานข้อมูลตลอดที่ใช้งาน ได้แก่
  - TTable สามารถสั่งงานได้เลย โดยจะอ้างถึง Tablename
  - TQuery เช่น SQL Text
2. Manual apply update after post ข้อมูลจะถูกจำค่าไว้ที่ฐานจำลองชั่วคราว และจะถูกเปลี่ยนที่ฐานข้อมูลเมื่อถูกสั่งอีกครั้ง ได้แก่
  - TClient Dataset

### Working with dataset

1. Dataset.Open เปิด
2. Dataset.Insert / Append นำเข้าข้อมูล
3. Dataset.Fields[x].Asxx := จะเก็บค่าข้อมูล [x] คือลำดับของคอลัมน์
4. Dataset.Post
5. Dataset.Close



เครื่องมือที่มีใน Script

- ปุ่มแรกลักษณะคล้ายสายฟ้าสีเหลือง ทำหน้าที่คล้ายกับ Run หรือ Append ใน SQL Query ใน Hosxp
- ปุ่มที่สอง คล้ายหน้าเอกสารสีครึ่ง ทำหน้าที่ Load Template ซึ่งเป็นรูปแบบในการเขียนคำสั่ง
- ปุ่มที่สาม และปุ่มที่สี่ คล้ายแว่นขยาย ทำหน้าที่สวิตช์ไปมาระหว่างตัวอักษร
- ปุ่มที่ห้า ทำหน้าที่โหลดไฟล์ที่เป็นภาษา Pascal จากภายนอกเข้ามา

### ขั้นตอนการเขียน Script ในโปรแกรม Report Designer

๑. ติดต่อกับฐานข้อมูล HOSXP

การใช้งานในโปรแกรม Report Designer ปกติจะเชื่อมต่อกับฐานข้อมูล HOSXP ตั้งแต่เข้าสู่โปรแกรมแล้ว จึงสามารถข้ามขั้นตอนนี้ได้

๒. สร้างคำสั่ง SQL เพื่อดึงข้อมูลที่ต้องการออกมาเก็บไว้ที่ Dataset

๓. แสดงข้อมูลใน Dataset

### ตัวอย่างการเขียน Script

#### วิธีการสร้างตัวแปรที่มีชนิดเป็น Dataset (TClient Dataset)

๐. ตั้งชื่อโปรแกรม ประกาศตัวแปร และสร้างตัวแปรที่ใช้เก็บข้อมูล

Unit MyScript; ชื่อโปรแกรม MyScript

Procedure main;

var Mycds:TClientDataset; ประกาศตัวแปร

begin

Mycds:=TClientDataset.Create(nil); สร้างข้อมูล Mycds ที่มีชนิดเป็น dataset (TClient Dataset) : nil ไม่มีใคร ลอยๆ วัตถุไม่ขึ้นกับใครจะถูกทำลายเมื่อปิด Report

๑. ส่งคำสั่ง SQL เข้าไปและดูข้อมูลมาไว้ใน Dataset

ฟังก์ชันที่ใช้ เช่น hosxp\_getdataset จะใช้ดูข้อมูล และเก็บข้อมูลเป็นนามสกุล .data

Mycds.data:=hosxp\_getdataset('select \* from ptttype order by ptttype limit ๑๐');

Mycds.First;

While not Mycds.EOF do

begin

๒. แสดงข้อมูลใน Dataset จะต้องระบุให้แสดงข้อมูลในคอลัมน์ไหน เช่น

ShowMessage(Mycds.fields[๐].AsString+' '+

Mycds.fieldByName('name').AsString);

Mycds.Next; ให้แสดงแถวแรก

end;

end;

## ขั้นตอนการนำข้อมูลไปเก็บใน Temp Report

๑. ตรวจสอบก่อนว่ามีข้อมูลของ Report Name นี้หรือยังถ้ามีแล้วให้ดำเนินการลบทิ้ง

เช่น `select * from tempreport where reportname='RB๑T๐๒'`

`Delete from tempreport where reportname='RB๑T๐๒'`

### ฟังก์ชัน

`zquery.close;` เพื่อให้ปิดฟอร์มที่เปิดทิ้งไว้ ก่อนจะสั่งให้ดำเนินการตามคำสั่งใหม่

`zquery.open;` อาจจะไม่ใช้ก็ได้

`zquery.sql.text:='delete from tempreport where reportname="Myscript";` ใช้ลบข้อมูล

`zquery.execsql;`

๒. ส่งคำสั่ง SQL ไปดึงข้อมูลมาเก็บใน Dataset

ประกาศตัวแปรเพิ่ม

`Mycds๒:TClientDataset;`

สร้างตัวแปรขึ้นมา

`Mycds๒:= TClientDataset.Create(nil);`

`Mycds๒.data:=hosxp_getdataset('select * from tempreport '+  
' where reportname="Myscript");`

๓. นำข้อมูลไปเก็บ

สั่งให้ไปที่คอลัมน์แรก

`Mycds.First;`

`Mycds๒.First;`

นำข้อมูลจากmycds ไปเก็บ Mycds๒

`while not Mycds.EOF do`

`begin`

`Mycds๒.Insert;` เพื่อบอกให้รู้ว่าเป็นการใส่ค่าให้ เหมือนการสั่งให้ อัปเดตข้อมูล

`Mycds๒.FieldName('reportname').AsString := 'Myscript';`

`Mycds๒.FieldName('id').AsString := 'Myscript';`

`Mycds๒.FieldName('name').AsString :=Mycds.FieldName('ptype').AsString;`

`Mycds๒.FieldName('name๑').AsString :=Mycds.FieldName('name').AsString;`

`Mycds.Next;`

`Mycds๒.Next;`

`End;`

คำสั่งการอัปเดตข้อมูลลงในฐานข้อมูล

If Mycds๒.changecount>๐ then เป็นการตรวจสอบว่ามีข้อมูลในตารางหรือไม่ มากกว่าศูนย์  
hosxp\_updatedelta(Mycds๒.delta,'select \* from tempreport where reportname="Myscript"');  
เป็นฟังก์ชันที่มีอยู่แล้ว  
Showmessage('Success'); เพื่อบอกให้รู้

### คำสั่งที่ได้

```
unit myscript;
procedure main;
var
  Mycds:TClientDataset;
  Mycds๒:TClientDataset;
  onlineuser : string;
begin
  onlineuser := report_value(๒);
  zquery.close;
  zquery.sql.text:='delete from tempreport where reportname="Myscript" and id
 ="'+onlineuser+'";
  zquery.execsql;
  Mycds:=TClientDataset.Create(nil);
  Mycds.data:=hosxp_getdataset('select * from pptype order by pptype limit ๑๐');
  Mycds๒:= TClientDataset.Create(nil);
  Mycds๒.data:=hosxp_getdataset('select * from tempreport '+
  ' where reportname="Myscript" and id="'+onlineuser+'");
  Mycds.First;
  Mycds๒.First;
  while not Mycds.EOF do
  begin
    Mycds๒.Insert;
    Mycds๒.FieldName('reportname').AsString := 'Myscript';
    Mycds๒.FieldName('id').AsString := onlineuser;
    Mycds๒.FieldName('name').AsString :=Mycds.FieldName('pptype').AsString;
    Mycds๒.FieldName('name๑').AsString :=Mycds.FieldName('name').AsString;
    Mycds.Next;
    Mycds๒.Next;
  end;
End;
```

```

If Mycds๒.changeCount>0 then
    hosxp_updatedelta(Mycds๒.delta,'select * from tempreport where reportname="Myscript" and
    id="' +onlineuser+'");
    Showmessage('Success');
    mycds.free;
    mycds๒.free;
end;
end.

```

### จงเก็บข้อมูลต่อไปนี้ ลงในตาราง TempReport

HN,ชื่อ – สกุล,อายุ(ปี),สิทธิประจำตัว,วันเกิด

โดยตั้งชื่อ reportname ว่า MyPatient ๒๐ คนแรก

```
unit Mypatient;
```

```
procedure main;
```

```
    Mycds:TClientDataset;
```

```
    Mycds:=TClientDataset.Create(nil);
```

```
    Mycds๒:TClientDataset;
```

```
    Mycds๒:= TClientDataset.Create(nil);
```

```
    Mycds.data:=hosxp_getdataset('select p.hn,concat(p.pname,p.fname," ",p.lname) as
        name,v.age_y,ptt.name as nameptt,p.birthday '+
        ' from patient p,vn_stat v,pttype ptt '+
        ' where p.hn=v.hn and p.pttype=ptt.pttype '+
        ' LIMIT ๒๐');
```

```
    Mycds๒.data:=hosxp_getdataset('select * from tempreport '+
        ' where reportname="Mypatient");
```

```
    Mycds.First;
```

```
    Mycds๒.First;
```

```
While not Mycds.EOF do
```

```
Begin
```

```
    Mycds๒.Insert;
```

```
    Mycds๒.FieldName('reportname').AsString :='Mypatient';
```

```
    Mycds๒.FieldName('id').AsString :='Mypatient';
```

```
    Mycds๒.FieldName('name').AsString :=Mycds.FieldName('hn').AsString;
```

```
    Mycds๒.FieldName('name๑').AsString :=Mycds.FieldName('name').AsString;
```



```

Mycds๒.FieldByName('name๒').AsString :=Mycds.FieldByName('age_y').AsString;
Mycds๒.FieldByName('name๓').AsString :=Mycds.FieldByName('nameptt').AsString;
Mycds๒.FieldByName('name๔').AsString :=Mycds.FieldByName('birthday').AsString;
Mycds.Next;
Mycds๒.Next;
End;
If Mycds๒.changeCount>0 then
    hosxp_updatedelta(Mycds๒.delta,'select * from tempreport where reportname="Mypatient");
    Showmessage('Success');
End;
End.

```

### การแสดงข้อมูลในตารางRefer

lhn | ชื่อ - สกุล | เลขที่refer | สถานที่ส่งมา | วันที่

- เขียน Script ดึงข้อมูลของคนไข้ที่มารับบริการจากตาราง ovst ของคนไข้ที่มารับบริการในช่วงวันที่  
เลือกไว้ก่อน

คำสั่งที่ต้องการ

```

select o.vn,o.hn,o.vstdate
from ovst o
left outer join patient p on o.hn=p.hn
where o.vstdate between "๒๐๑๑-๐๑-๐๑" and "๒๐๑๑-๐๑-๓๑"

```

- ดำเนินการเขียน Script

```

Unit myscript๑;
Procedure main;
var
    mycgs :TClientDataset;
    Mycgs๒:TClientDataset;
Begin
    zquery.close;
    zquery.sql.text:='delete from tempreport where reportname="Myscript๑"';
    zquery.execsql;
    Mycgs:=TClientDataset.Create(nil);
    Mycgs.data:=hosxp_getdataset('select o.vn,o.hn,concat(p.pname,p.fname," ",p.lname) as

```

```

name,o.vstdate from ovst o left outer join patient p on o.hn=p.hn '+
' where o.vstdate between "๒๐๑๑-๐๑-๐๑" and "๒๐๑๑-๐๑-๓๑" LIMIT ๑๐');
Mycds๒:= TClientDataset.Create(nil);
Mycds๒.data:=hosxp_getdataset('select * from tempreport '+
' where reportname="Myscript๑");
Mycds.First;
Mycds๒.First;
While not Mycgs.EOF do
Begin
Mycds๒.Insert;
Mycds๒.FieldName('reportname').AsString :='Myscript๑';
Mycds๒.FieldName('id').AsString :='Myscript๑';
Mycds๒.FieldName('name').AsString :=Mycds.FieldName('vn').AsString;
Mycds๒.FieldName('name๑').AsString :=Mycds.FieldName('hn').AsString;
Mycds๒.FieldName('name๒').AsString :=Mycds.FieldName('name').AsString;
Mycds๒.FieldName('name๓').AsString :=Mycds.FieldName('vstdate').AsString;
Mycds.Next;
Mycds๒.Next;
End;
If Mycgs๒.changecount>๐ then
hosxp_updatedelta(Mycds๒.delta,'select * from tempreport where reportname="Myscript๑");
Showmessage('Success');
End;
End.

```

### การเขียนฟังก์ชันที่ใช้รูปแบบที่มีอยู่แล้วในโปรแกรมช่วยเขียนให้ง่ายขึ้น

#### - ฟังก์ชัน Inputquery(Title,Label,ตัวแปรเก็บค่า)

๑. ใช้สำหรับบอกให้ผู้ใช้ดำเนินการอย่างไร
๒. ใช้ค่าที่ผู้ใช้ไม่กรอกแต่ทราบค่าที่แน่นอน (ค่าDeflur)

เช่น –

```
var min,max : string เป็นการประกาศตัวแปร
```

```
min := '๘๐'
```

```
max := '๙๙'
```

```
Inputquery('ค่าเริ่มต้น','กรุณาใส่ค่าเริ่มต้นของระดับน้ำตาลที่ต้องการ?','๘๐')
```

```
Inputquery('ค่าสิ้นสุด', 'กรุณาใส่ค่าสิ้นสุดของระดับน้ำตาลที่ต้องการ?', '๙๙')
```

```
zquery.close; ลบอันเดิมออกก่อน
```

```
zquery.sql.text := 'delete from tempreport where reportname = "testscript"'; เป็นการ
```

ลบฟอร์ม

```
zquery.execsql;
```

```
fcds.close;
```

```
fcds.datarequest('select * from tempreport where reportname = "testscript"');
```

```
fcds.open;
```

```
fcds.insert;
```

```
fcds['reportname'] := 'testscript';
```

```
fcds['name๑'] := min;
```

```
fcds['name๒'] := max;
```

```
fcds.post;
```

```
fcds.datarequest('select * from tempreport where reportname = "testscript"') ต้องเขียน
```

อีกรอบ

```
applyupdate_fcds; เป็นการสั่งให้อัพเดทข้อมูล
```

หมายเหตุ

- fcds เป็นตัวแปรที่มีอยู่แล้วไม่จำเป็นต้องประกาศตัวแปรอีก
- .datarequest เป็นค่าที่ดึงข้อมูลออกมาจากHosxp ซึ่งมีอยู่แล้วในโปรแกรม
- Mycds๒.FieldByName('reportname').AsString สามารถแทนได้ fcds['reportname']
- ฟังก์ชัน **GetUserParameter** เป็นฟังก์ชันไว้ตรวจสอบมีการส่งตัวแปรหรือพารามิเตอร์เข้าfrom นี้หรือยัง ถ้ายังจะได้เรียกฟังก์ชันให้ใส่ตัวแปรต่อไป

เช่น

```
Unit myscript๑;
```

```
Procedure main;
```

```
var
```

```
mycds :TClientDataset;
```

```
Mycds๒:TClientDataset;
```

```
date๑, date๒ :Tdatetime;
```

```
ds๑, ds๒ :string;
```

```
Begin
```

```
zquery.close;
```

```
zquery.sql.text := 'delete from tempreport where reportname = "Myscript๑";
```

```
zquery.execsql;
```

If not getdaterange() then exit

date<sub>o</sub> :=date\_result<sub>o</sub>();

date<sub>h</sub> :=date\_result<sub>h</sub>();

ds<sub>o</sub>:=formatdatetime('YYY-MM-DD',date<sub>o</sub>);

ds<sub>h</sub>:=formatdatetime('YYY-MM-DD',date<sub>h</sub>);

Mycds:=TClientDataset.Create(nil);

Mycds.data:=hosxp\_getdataset('select o.vn,o.hn,concat(p.pname,p.fname," ",p.lname) as  
name,o.vstdate from ovst o left outer join patient p on o.hn=p.hn '+  
' where o.vstdate between '"+ds<sub>o</sub>+"' and '"+ds<sub>h</sub>+"'');

Mycds<sub>h</sub>:= TClientDataset.Create(nil);

Mycds<sub>h</sub>.data:=hosxp\_getdataset('select \* from tempreport '+  
' where reportname="Myscript<sub>o</sub>");

Mycds.First;

Mycds<sub>h</sub>.First;

While not Mycds.EOF do

Begin

Mycds<sub>h</sub>.Insert;

Mycds<sub>h</sub>.FieldByName('reportname').AsString :='Myscript<sub>o</sub>';

Mycds<sub>h</sub>.FieldByName('id').AsString :='Myscript<sub>o</sub>';

Mycds<sub>h</sub>.FieldByName('name').AsString :=Mycds.FieldByName('vn').AsString;

Mycds<sub>h</sub>.FieldByName('name<sub>o</sub>').AsString :=Mycds.FieldByName('hn').AsString;

Mycds<sub>h</sub>.FieldByName('name<sub>h</sub>').AsString :=Mycds.FieldByName('name').AsString;

Mycds<sub>h</sub>.FieldByName('name<sub>o</sub>h').AsString :=Mycds.FieldByName('vstdate').AsString;

Mycds.Next;

Mycds<sub>h</sub>.Next;

End;

If Mycds<sub>h</sub>.changecount>0 then

hosxp\_updatedelta(Mycds<sub>h</sub>.delta,'select \* from tempreport where reportname="Myscript<sub>o</sub>");

Showmessage('Success');

End;

End.

## การเปรียบเทียบความสามารถและเทคนิคการแสดงผลรายงาน ด้วยการเขียน Script ในรายงาน

๑. มีความยืดหยุ่นในการเขียนรายงาน
๒. สามารถนำข้อมูลมาเก็บไว้ใน Dataset แล้วค่อยนำมาแสดงผลที่หลังทำให้มีผลต่อความไวในการประมวลผลที่มีข้อมูลจำนวนมาก