การเขียน SQL เพื่อจัดการฐานข้อมูล MySQL

ด้วยโปแกรม Navicat

โดย นายไพบูลย์ ไวกยี

สสจ.พระนครศรีอยุธยา

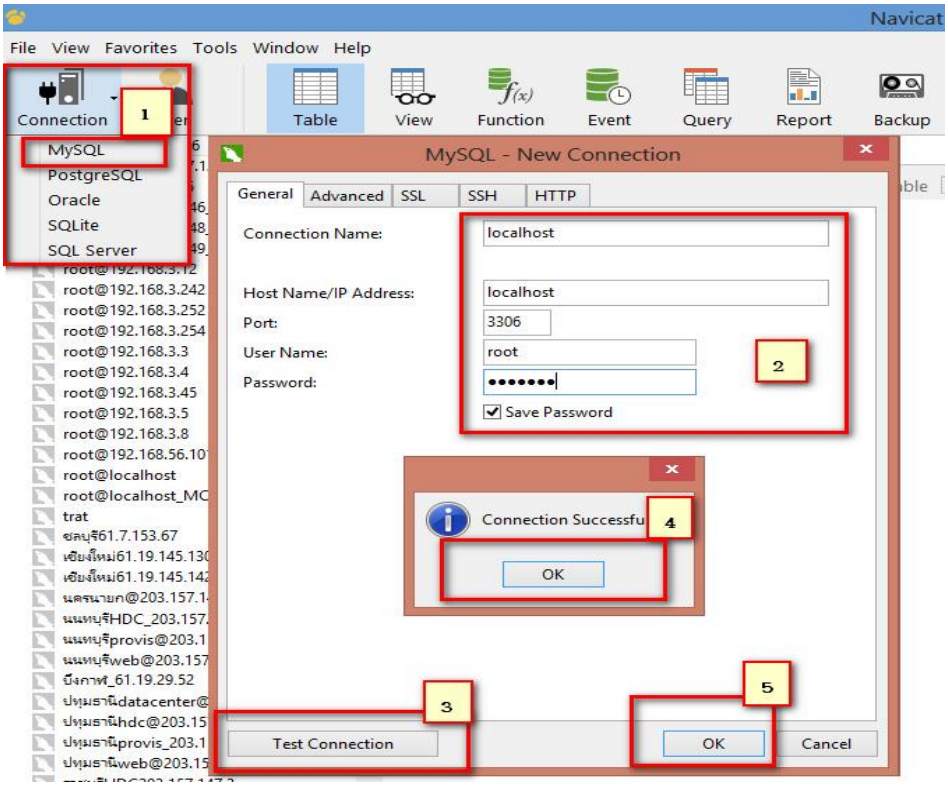
การเขียน SQL เบื้องต้นด้วยโปแกรม Navicat

SQL ย่อมาจาก structured query language คือ ภาษาที่ใช้ในการเขียนโปรแกรม เพื่อจัดการกับฐานข้อมูลโดยเฉพาะ ซึ่งองค์ประกอบของการจัดทำ SQL คือ ฐานข้อมูล (Database) , โปแกรม SQL Client และชุดคำสั่ง SQL ที่ใช้ในการประมวลผลต่างๆ ซึ่งเกิดจาก User สำหรับโปรแกรมและฐานข้อมูลที่จะกล่าวถึงต่อไปนี้คือ ฐานข้อมูล MySQL และโปรแกรม Navicat

ฐานข้อมูล MySQL มีหลากหลายเวอร์ชั่น แต่คำสั่งพื้นฐานจะมีเหมือนๆกัน

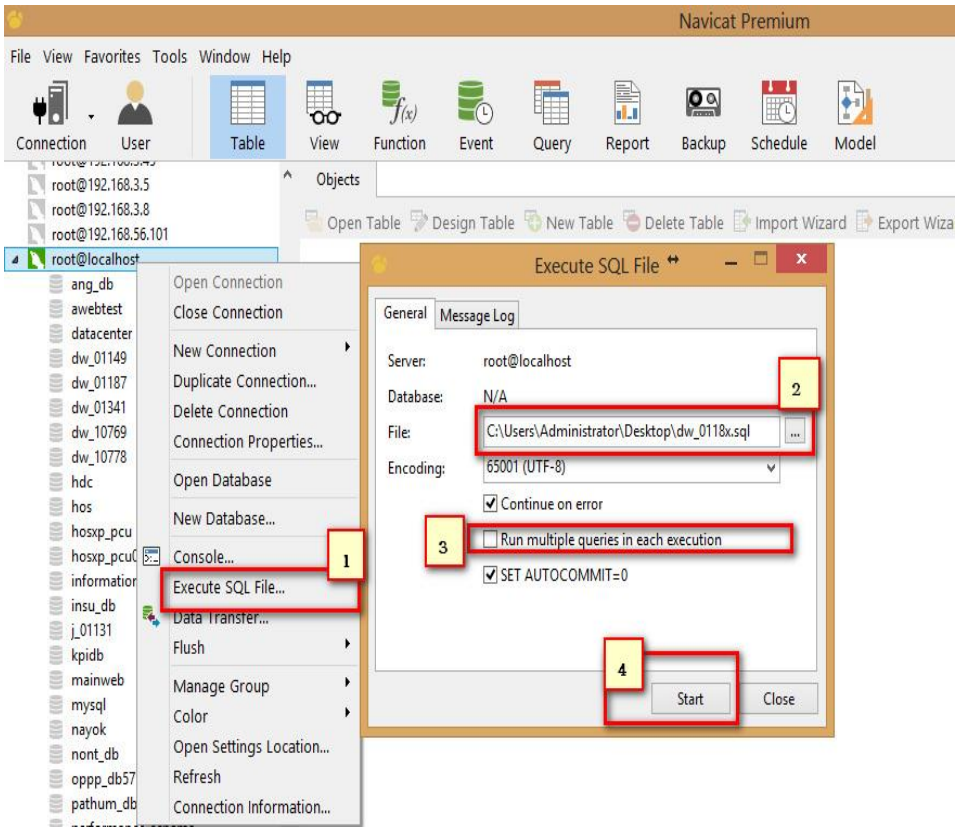
โปรแกรม Navicat ก็มีหลายเวอร์ชั่น สำหรับที่แนะนำให้ใช้งานตั้งแต่เวอร์ชั่น ๙ ขึ้นไป เพราะมีตัวช่วยในการเขียนคำสั่งที่มีประสิทธิภาพมากกว่าเวอร์ชั่นอื่นๆ

หลังจากติดตั้งโปรแกรม Navicat และโปแกรมฐานข้อมูล MySQLสำหรับการฝึกปฏิบัติเรียบร้อยแล้ว ให้สร้างการเชื่อมต่อฐานข้อมูล โดยให้คลิกที่ Connection และระบุแหล่งเชื่อมต่อ โดยให้พิมพ์ แหล่งเชื่อมต่อในการฝึกปฏิบัติต่อดังนี้

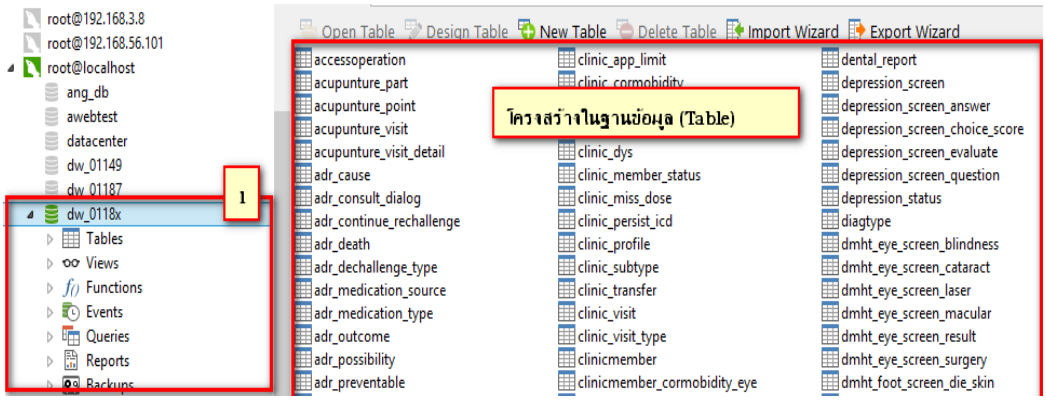


Connection Name	แล้วแต่จะตั้ง แต่ควรสื่อถึงฐานข้อมูล
Host Name/IP Address	เครื่องที่มีฐานข้อมูลที่ต้องการติดต่อ
Port	ปกติคือ ๓๓๐๖
Username	กำหนดโดยผู้ดูแลระบบ
Password	กำหนดโดยผู้ดูแลระบบ

ขั้นตอนนำเข้า Restore ฐานข้อมูลการฝึกปฏิบัติ คลิกขวาที่ Connection ที่สร้างไว้แล้วเลือก Execute SQL File เลือกไฟล์ SQL และนำเข้าโดย คลิก Checkbox Run multiple.....ออก



เปิดฐานข้อมูลและตรวจสอบโครงสร้าง (Table , Field ,PK, Index เป็นต้น)



การเขียน SQL ต้องเลือกฐานข้อมูลก่อนเสมอ

สิ่งสำคัญของการเขียน SQL คือ ผู้เขียนต้องมีความรู้เกี่ยวกับฐานข้อมูลที่จะเขียน มีความรู้ในด้านโครงสร้าง ชนิดการจัดเก็บ ความกว้าง ฯลฯ ทุกรายละเอียดของฐานข้อมูลนั้นๆ เพื่อการประมวลผลที่ถูกต้อง รายละเอียดโครงสร้างของฐานข้อมูลนี้ เรียกว่า “Data Dictionary”

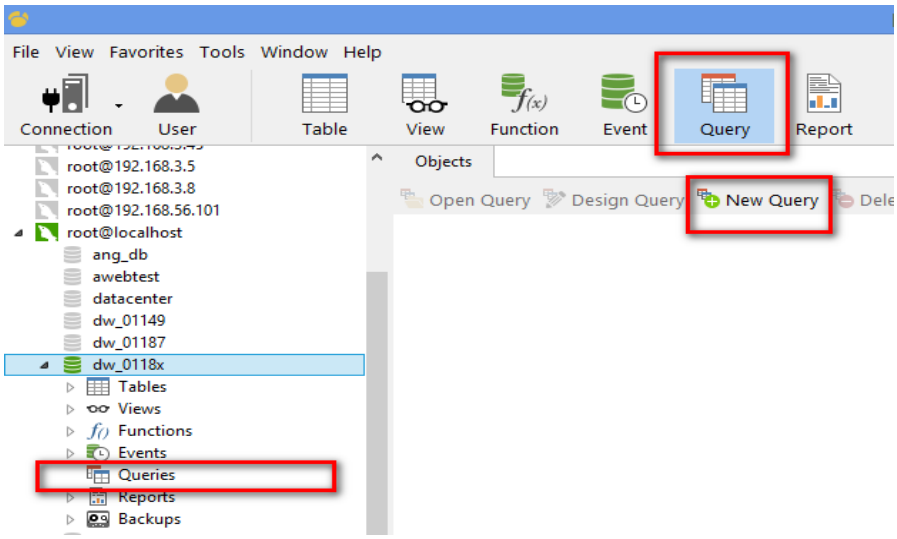
<u>คำสั่ง SQL หลักๆที่นิยมใช้</u>

ที่เกี่ยวกับ Database และ Table : CREATE , DROP

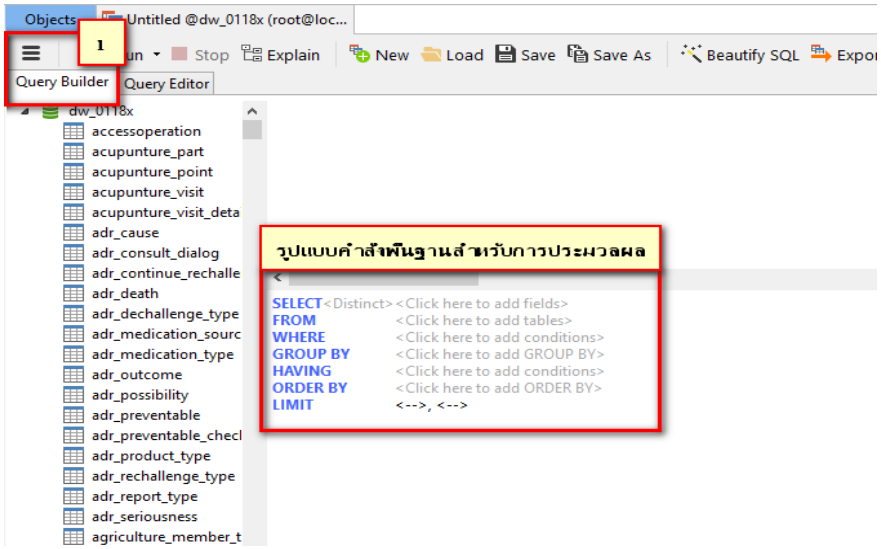
ที่เกี่ยวกับเนื้อข้อมูลใน Table : SELECT , INSERT , DELETE , UPDATE , ALTER

การเขียน SQL สำหรับโปรแกรม Navicat นั้นมี ๒ โหมด คือ โหมด Query Builder เป็นโหมดสำหรับ SQL แบบพื้นฐาน และโหมด Query Editor ซึ่งสามารถเขียน SQL แบบ Advance ได้

การเขียน SQL แบบพื้นฐานด้วย Query Builder คลิกที่ Query ได้จากทั้งสองที่ แล้วเลือก New Query



เลือก Query Builder ท่านจะเห็นรูปแบบคำสั่ง SELECT อยู่ด้านล่างขวามือ



คำสั่ง SELECT

SELECT เป็นการเลือกดูข้อมูลโดยกำหนดเงื่อนไขต่างๆ คล้ายกับหลักการทางระบาดวิทยา เช่น ข้อมูลอะไร จากที่ไหน ต้องการอย่างไร เมื่อไร ช่วงไหน ต้องการข้อมูลของใคร เป็นต้น

การใช้คำสั่ง SELECT ไม่ทำให้เกิดการเสียหายของข้อมูล ดังนั้นถือว่าเป็นคำสั่งที่ปลอดภัย แต่อาจทำให้ฐานข้อมูลนั้นๆ ประมวลผลงานอื่นๆช้าลง หากการเขียนคำสั่ง SELECT ไม่มีประสิทธิภาพ เช่น สับสน วนไปมา หรือกับข้อมูลขนาดใหญ่ที่ไม่ได้มีการจัดทำดัชนีการค้นหาไว้

ดังนั้น คำสั่ง SELECT จะมีผลกระทบกับข้อมูลที่ใช้ร่วมกัน หากเขียนไม่รัดกุม แต่ไม่ทำให้ฐานข้อมูลเสียหาย

รูปแบบคำสั่ง SELECT แบบฉบับเต็ม

SELECT [ALL | DISTINCT | DISTINCTROW] [HIGH_PRIORITY] [STRAIGHT_JOIN]

[SQL_SMALL_RESULT] [SQL_BIG_RESULT] [SQL_BUFFER_RESULT]

[SQL_CACHE | SQL_NO_CACHE] [SQL_CALC_FOUND_ROWS]

select_expr [, select_expr ...]

[FROM table_references

[WHERE where_condition]

[GROUP BY {col_name | expr | position} [ASC | DESC], ... [WITH ROLLUP]]

[HAVING where_condition]

[ORDER BY {col_name | expr | position} [ASC | DESC], ...]

[LIMIT {[offset,] row_count | row_count OFFSET offset}]

[PROCEDURE procedure_name(argument_list)]

[INTO OUTFILE 'file_name' export_options | INTO DUMPFILE 'file_name' | INTO @var_name [, @var_name]] [FOR UPDATE | LOCK IN SHARE MODE]]

รูปแบบคำสั่ง SELECT แบบฉบับย่อ

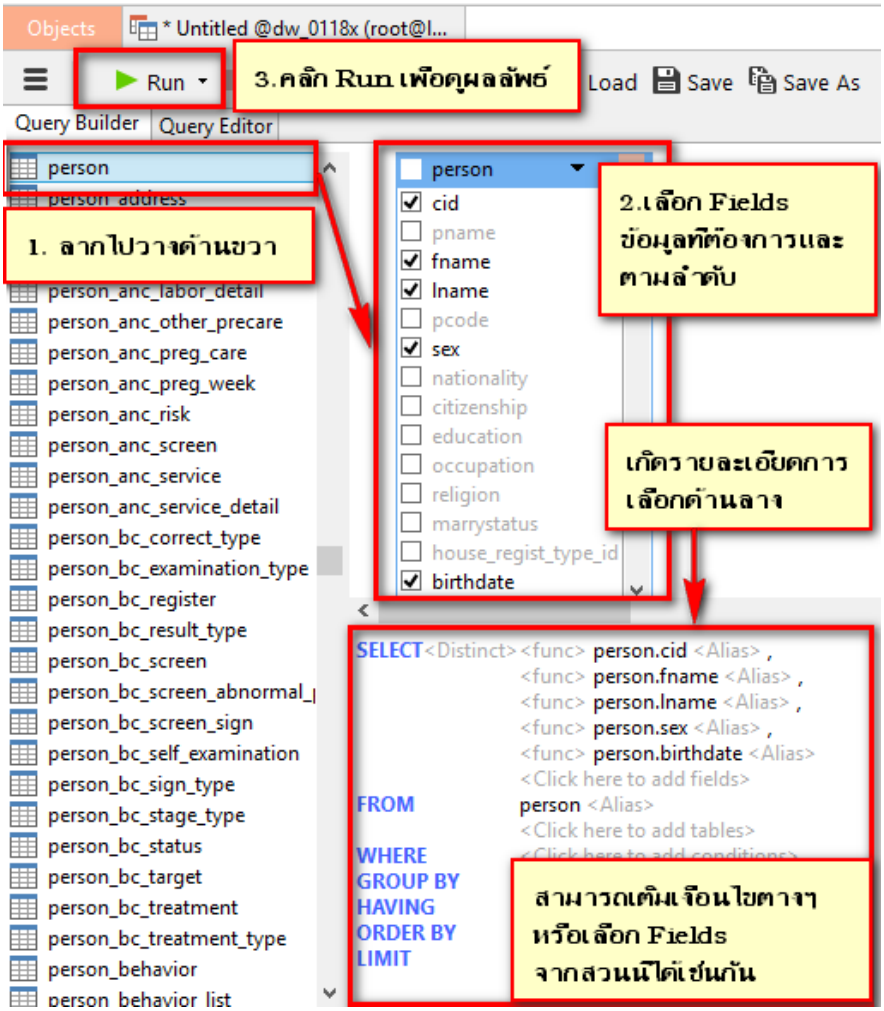
SELECT select_expr , select_expr [FROM table_references]

[WHERE where_condition] [GROUP BY col_name | expr]

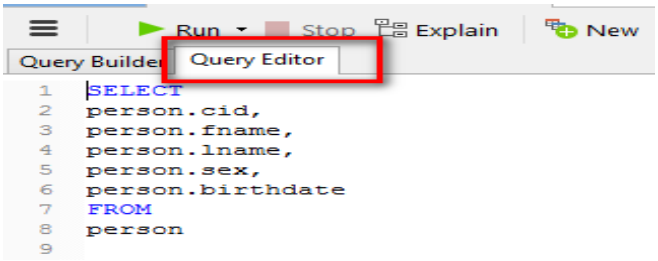
[HAVING where_condition] [ORDER BY BY col_name | expr [ASC | DESC]]

ตัวอย่างที่ ๑ แสดงข้อมูล จงแสดงข้อมูล เลขประชาชน,ชื่อ,นามสกุล,เพศ,วันเดือนปีเกิด ของทุก Records ที่มีอยู่ใน Table PERSON

รายละเอียดข้อมูลที่เกี่ยวข้อง Table PERSON , Fields cid,fname,lname,sex,birthdate



จากภาพจะเห็นได้ว่าในส่วนขวาของด้านล่างจอภาพมีรูปแบบคำสั่งของการทำ SQL เกิดขึ้นในลักษณะคล้ายกับคำสั่งในรูปแบบ Command mode และในส่วนนี้เองจะเห็นว่าสามารถมีการใช้งาน Function เพิ่มเติมต่างๆ ได้ Function ประกอบด้วย SUM,MAX,MIN,AVG,COUNT... ซึ่งเป็น Function พื้นฐานทางคณิตศาสตร์



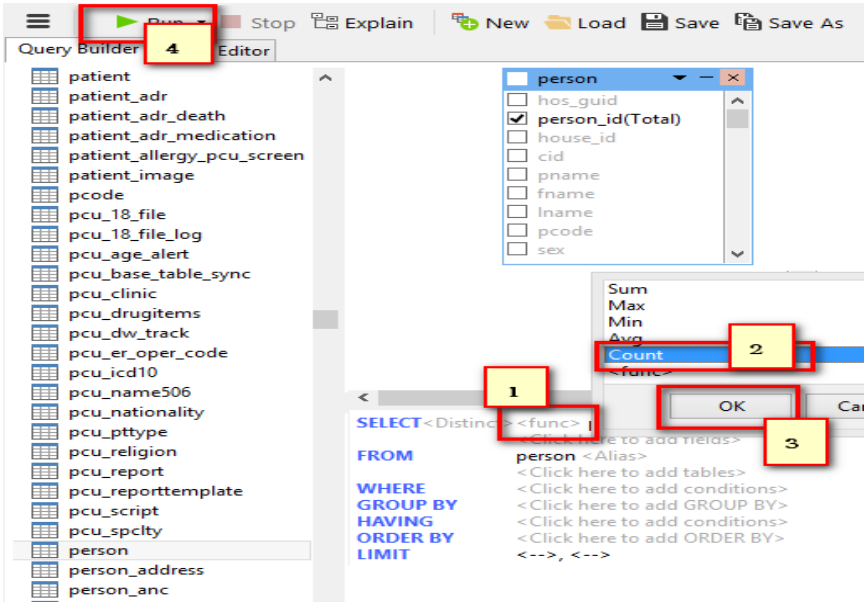
ภาพตัวอย่าง คำสั่งในรูปแบบ Command mode

ตัวอย่างที่ ๒ นับจำนวนข้อมูล จงนับจำนวนข้อมูลทุก Records ที่มีอยู่ใน Table PERSON

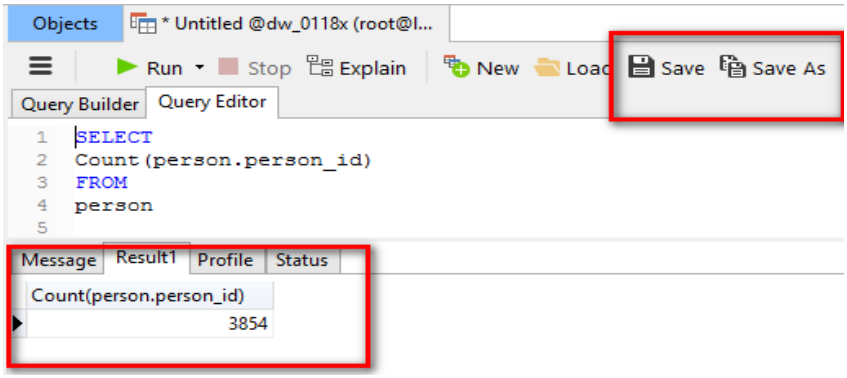
รายละเอียดข้อมูลที่เกี่ยวข้อง

Table: PERSON

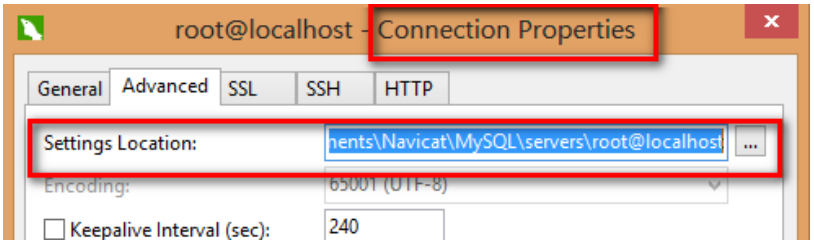
Function: Count



ผลลัพธ์จากการประมวลผล



จากภาพจะเห็นว่า SQL สามารถ Save เก็บไว้ได้ โดยคลิกที่ Save หรือ Save As ซึ่งไฟล์ที่เก็บจะมีที่ตั้งตามการกำหนดค่าพื้นฐานของ Connection



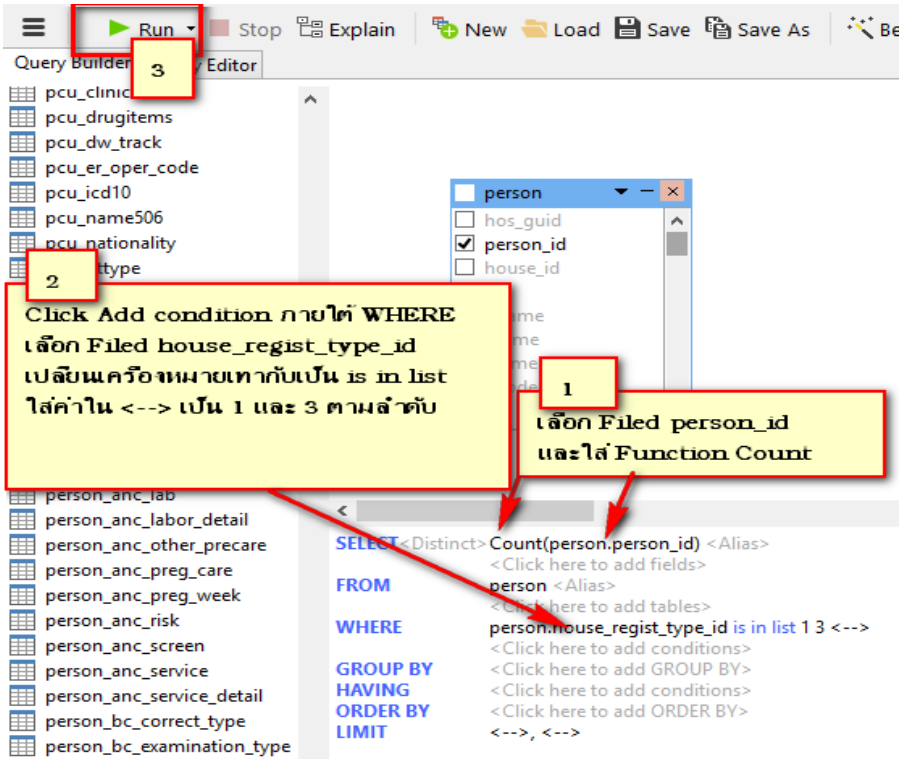
ตัวอย่างที่ ๓ นับจำนวนข้อมูลแบบมีเงื่อนไข จงนับจำนวน Records ใน Table PERSON ที่ Filed house_regist_type_id เป็น ๑ และ ๓

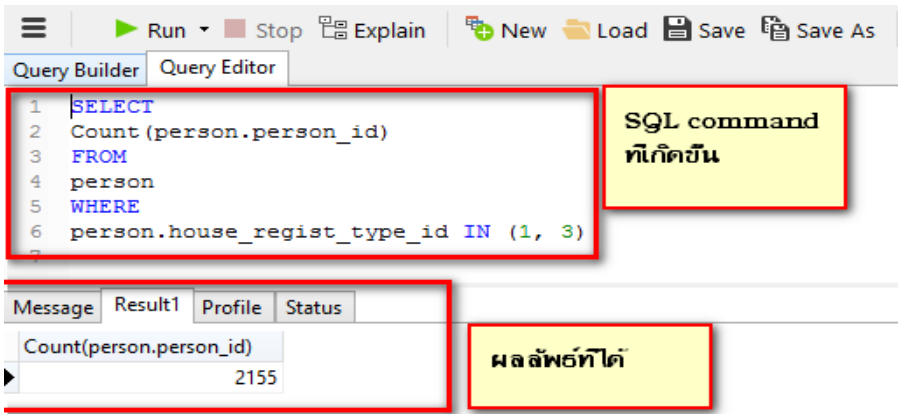
รายละเอียดข้อมูลที่เกี่ยวข้อง

Table: PERSON

Function: Count ,IN

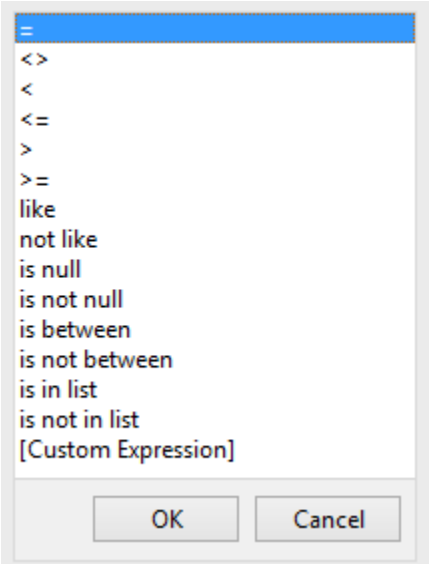
Field: peson_id,house_regist_type_id





โปรดสังเกตและจดจำ SQL Command ที่เกิดขึ้นจากการใช้ Function ต่างๆด้วย เพราะจะมีประโยชน์ในการเขียน SQL แบบ Advance และบางครั้งของการนับข้อมูลในระบบสาธารณสุข ต้องใช้ Function เพิ่มเติมด้วย เช่น is in list ถูกเปลี่ยนเป็น Function IN() ใน Command mode

เครื่องหมายทางคณิตศาสตร์ กับ Condition ที่มีให้ใช้งาน เป็นการแปลงไปสู่ Function บางส่วน เช่น is between , is in list ส่วนเครื่องหมายอื่น ที่น่าสนใจ <> มีความหมาย ไม่เท่ากับ



การเชื่อม Condition ในแต่ละ Condition จะต้องเชื่อมด้วย AND หรือ OR

การใส่วงเล็บ จะทำให้ SQL ประมวลผลข้อมูลในวงเล็บก่อนเสมอ

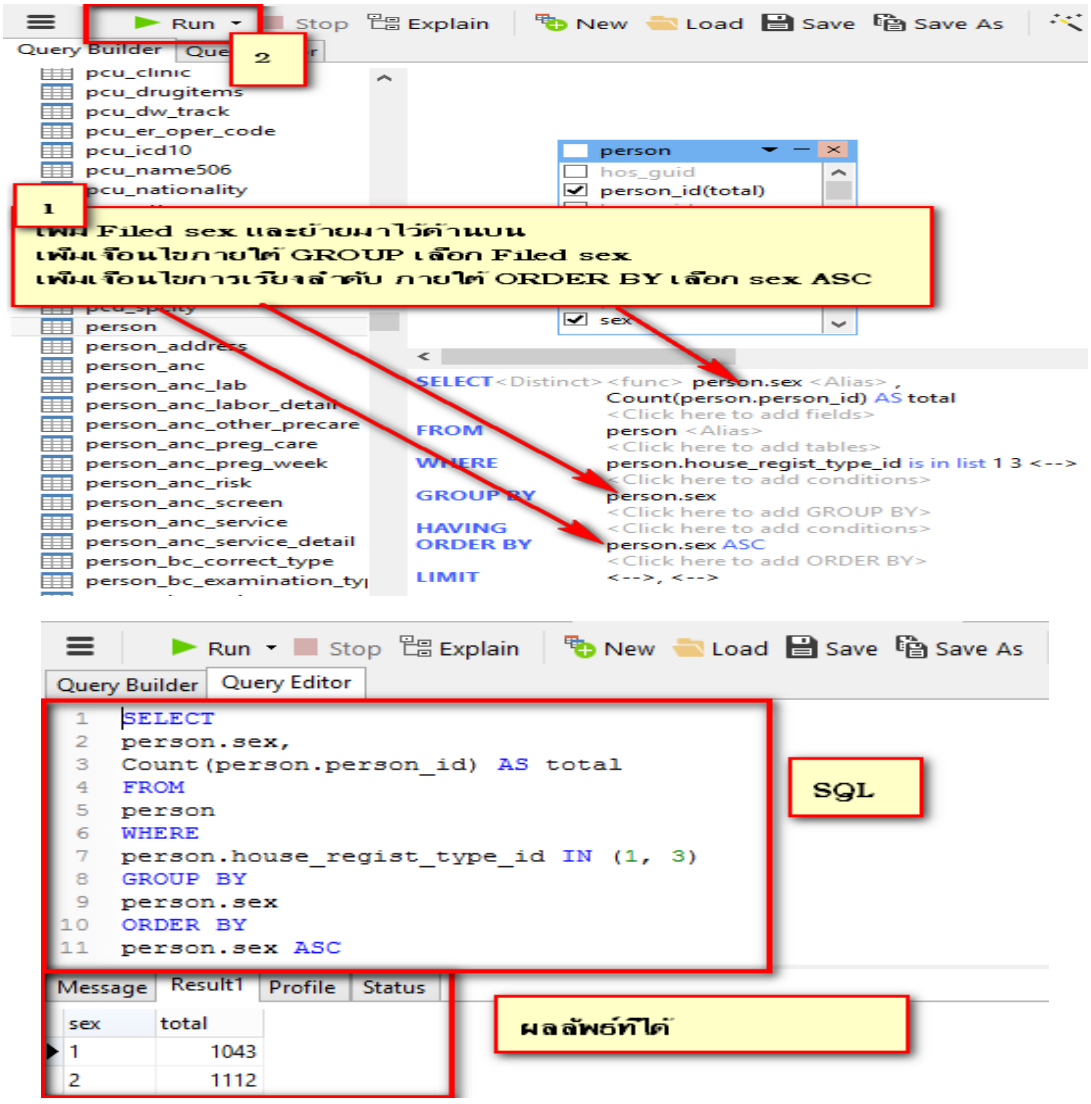
ตัวอย่างที่ ๔ นับจำนวนข้อมูลแบบมีเงื่อนไข การจัดกลุ่ม และการเรียงลำดับข้อมูล จงนับจำนวน Records ใน Table PERSON ที่ Filed house_regist_type_id เป็น ๑ และ ๓ และจัดกลุ่มตาม เพศ

รายละเอียดข้อมูลที่เกี่ยวข้อง

Table: PERSON

Function: Count , IN ,

Field: peson_id,house_regist_type_id,sex



การใช้งาน Query Builder และ Query Editor แบบผสมผสาน

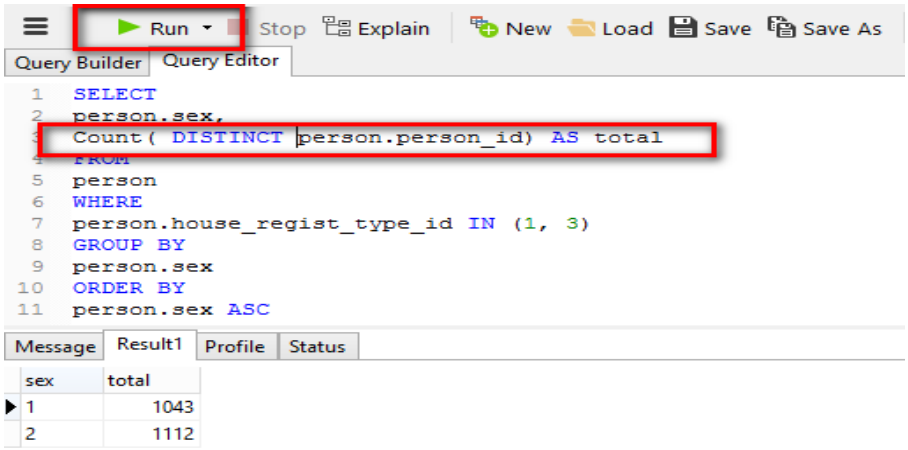
ตัวอย่างที่ ๕ นับจำนวนข้อมูลแบบมีเงื่อนไข การจัดกลุ่ม และการเรียงลำดับข้อมูล จงนับจำนวน Records ใน Table PERSON ที่ Filed house_regist_type_id เป็น ๑ และ ๓ และจัดกลุ่มตาม เพศ โดยที่ Field person_id ไม่ซ้ำกัน

รายละเอียดข้อมูลที่เกี่ยวข้อง

Table: PERSON

Function: Count , IN ,DISTINCT

Field: peson_id,house_regist_type_id,sex



เติม Function DISTINCT ใน Function Count เพื่อให้เกิดการนับแบบไม่ซ้ำกัน ซึ่งทำใน โหมด SQL EDITOR

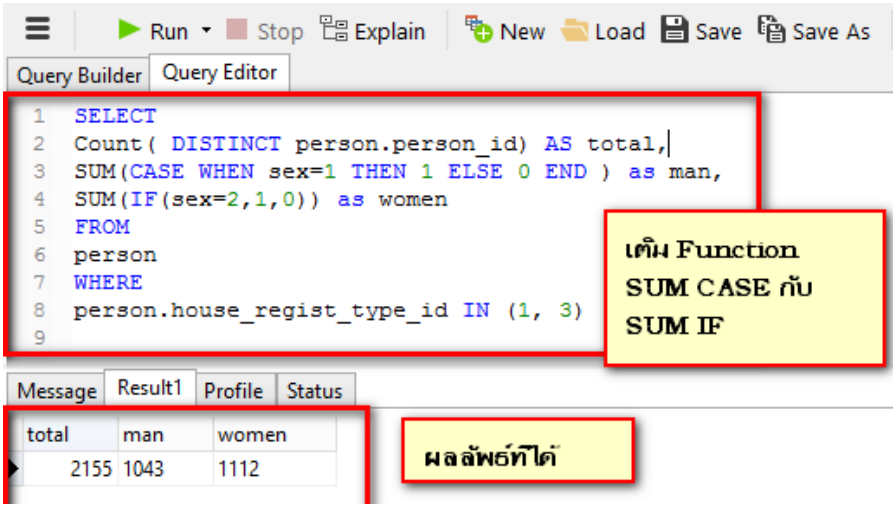
ตัวอย่างที่ ๖ นับจำนวนข้อมูลแบบมีเงื่อนไข การจัดกลุ่ม และการเรียงลำดับข้อมูล แบบใช้ Function SUM จงนับจำนวน Records ใน Table PERSON ที่ Filed house_regist_type_id เป็น ๑ และ ๓ และจัดกลุ่มตาม เพศ

รายละเอียดข้อมูลที่เกี่ยวข้อง

Table: PERSON

Function: Count , IN ,Distinct, SUM, Case ,IF

Field: peson_id,house_regist_type_id,sex



ข้อควรระวัง

การใช้ Function Count(Distinct expr) จะนับแบบไม่ซ้ำตาม expr แต่การใช้ SUM() จะนับตามเงื่อนไขโดยไม่สนใจการซ้ำซ้อนของข้อมูล ในตัวอย่างที่ได้เท่ากันเพราะ Field person_id ไม่ซ้ำกัน

การเขียน SQL แบบหลายๆ TABLE เชื่อมโยง SQL Relationship

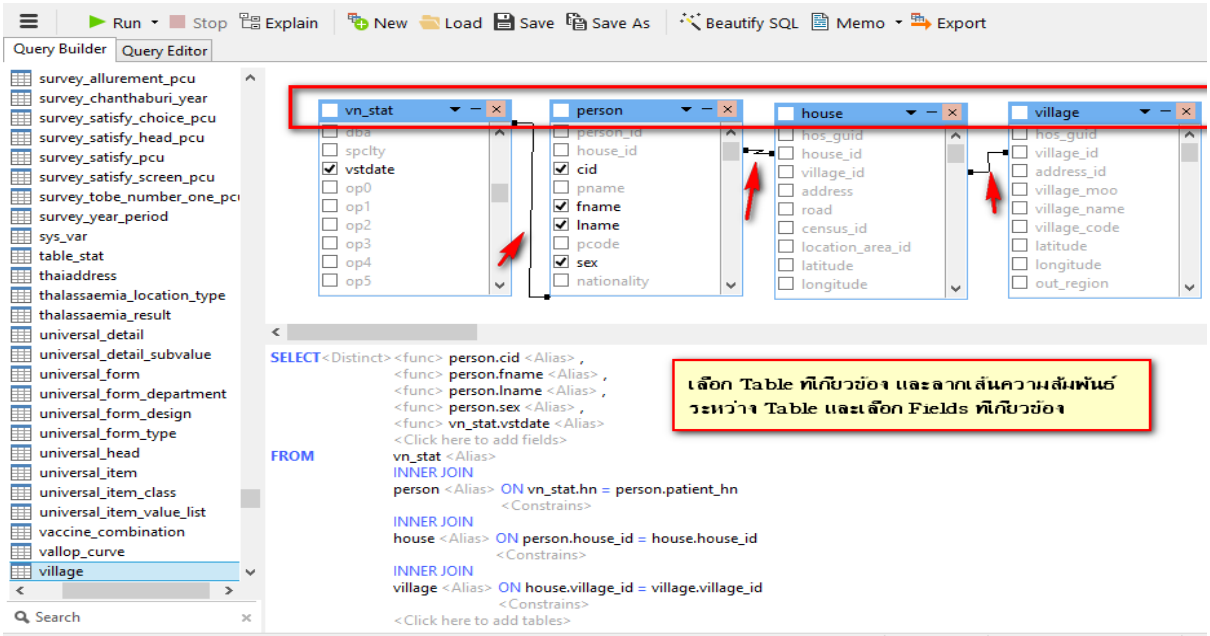
ตัวอย่างที่ ๖ แสดงข้อมูลแบบเชื่อมโยง จงแสดงข้อมูลผู้รับบริการผู้ป่วยนอกระหว่างวันที่ ๑ มกราคม ๒๕๕๕ ถึงวันที่ ๓๑ มกราคม ๒๕๕๕ โดยเป็นคนในเขตของหน่วยบริการ (นับคนที่อยู่ในหมู่บ้านในเขต) เลขประชาชน ชื่อ นามสกุล เพศ วันที่รับบริการ

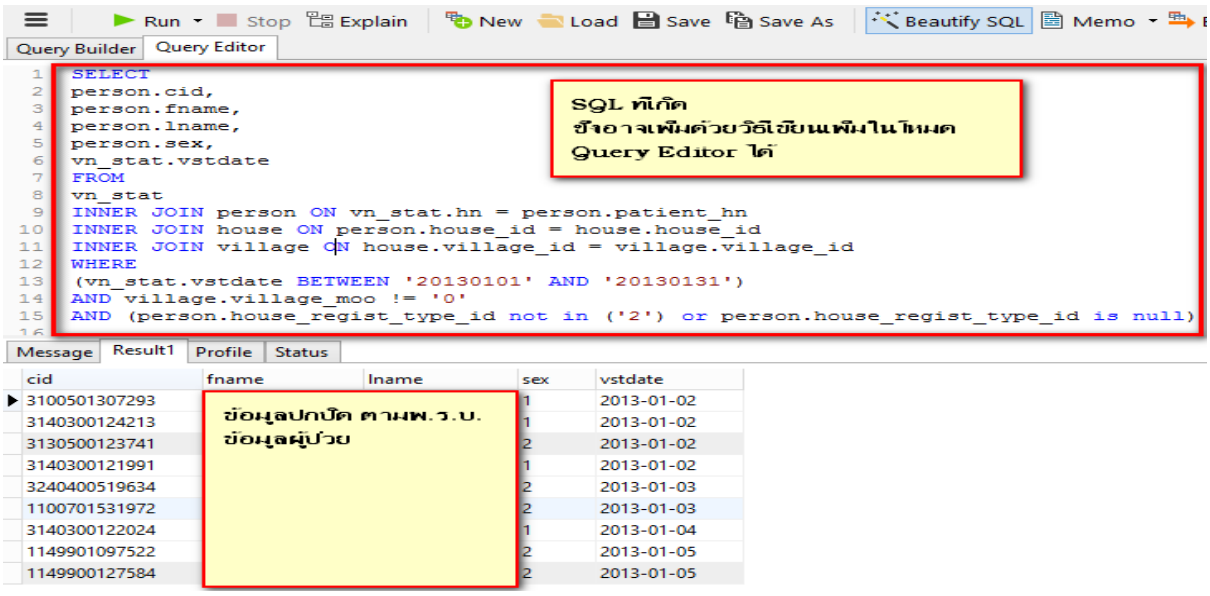
รายละเอียดข้อมูลที่เกี่ยวข้อง

Table: VN_STAT , PERSON , HOUSE , VILLAGE

Function: Count , IN ,DISTINCT

Field: cid , fname , lname , sex , vstdate , hn , patient_hn , house_id , village_id , house_regist_type_id , village_moo





cid	fname	lname	sex	vstdate
3100501307293			1	2013-01-02
3140300124213			1	2013-01-02
3130500123741			2	2013-01-02
3140300121991			1	2013-01-02
3240400519634			2	2013-01-03
1100701531972			2	2013-01-03
3140300122024			1	2013-01-04
1149901097522			2	2013-01-05
1149900127584			2	2013-01-05

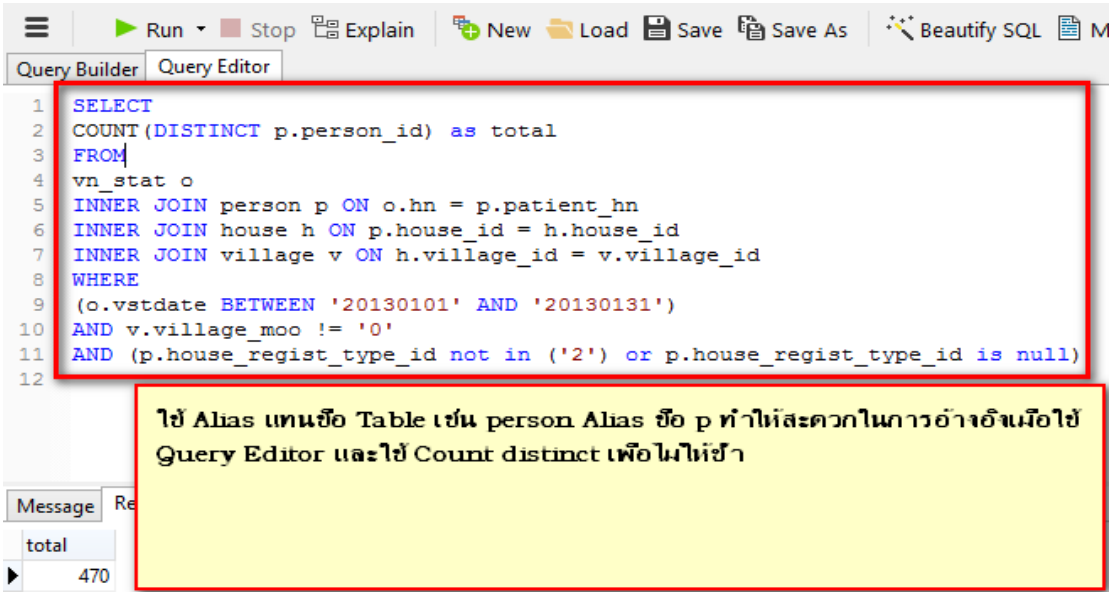
ตัวอย่างที่ ๗ นับข้อมูลแบบเชื่อมโยง จงนับจำนวนผู้รับบริการผู้ป่วยนอกระหว่างวันที่ ๑ มกราคม ๒๕๕๕ ถึงวันที่ ๓๑ มกราคม ๒๕๕๕ โดยเป็นคนในเขตของหน่วยบริการ (นับคนที่อยู่ในหมู่บ้านในเขต) โดยนับเป็นคน

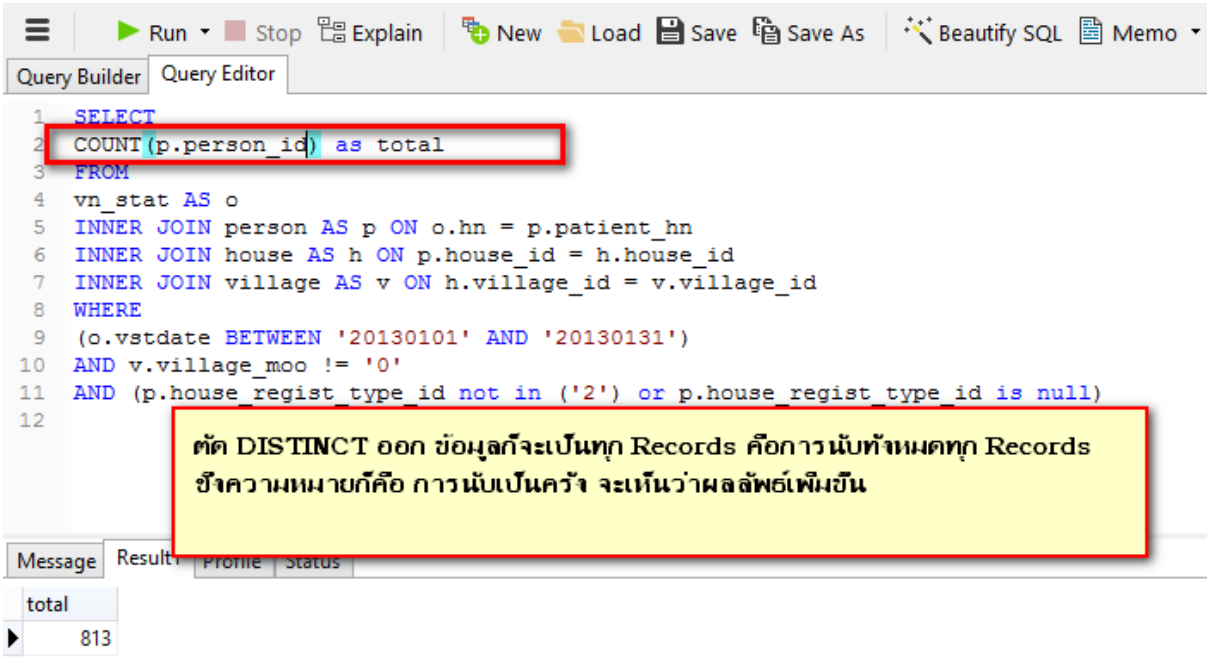
รายละเอียดข้อมูลที่เกี่ยวข้อง

Table: VN_STAT , PERSON , HOUSE , VILLAGE

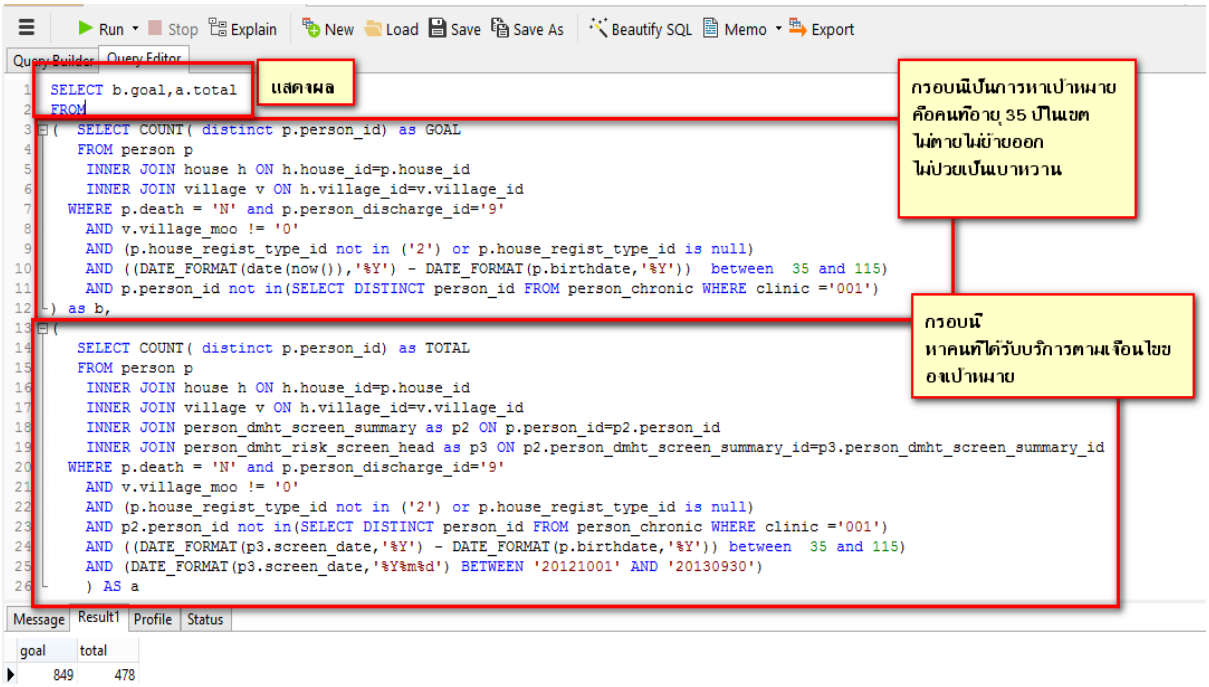
Function: Count , IN ,DISTINCT

Field: cid , fname , lname , sex , vstdate , hn , patient_hn , house_id , village_id , house_regist_type_id , village_moo

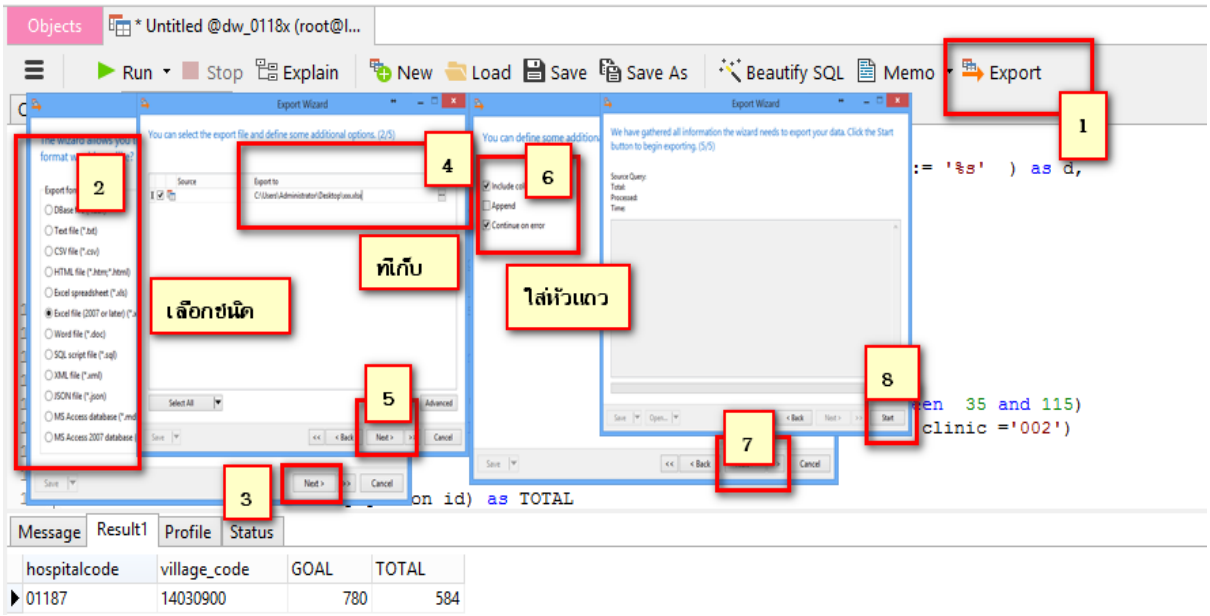




ตัวอย่างที่ ๘ นับข้อมูลแบบเชื่อมโยง (Advance) จงนับจำนวนคนในเขตของหน่วยบริการ ที่ได้รับการคัดกรองโรคเบาหวาน โดยที่อายุ ๓๕ ปีขึ้นไป และไม่ป่วยด้วยโรคเบาหวาน เปรียบเทียบกับจำนวนคนในเขตของหน่วยบริการที่ไม่ป่วยด้วยโรคเบาหวาน ข้อมูลระหว่างวันที่ ๑ ตุลาคม ๒๕๕๕ ถึงวันที่ ๓๐ กันยายน ๒๕๕๖ **รายละเอียดข้อมูลที่เกี่ยวข้อง** หลายตาราง และเป็นการเขียนแบบ SQL ซ้อน SQL ซึ่งต้องใช้ในโหมด Query Editor เป็นหลัก



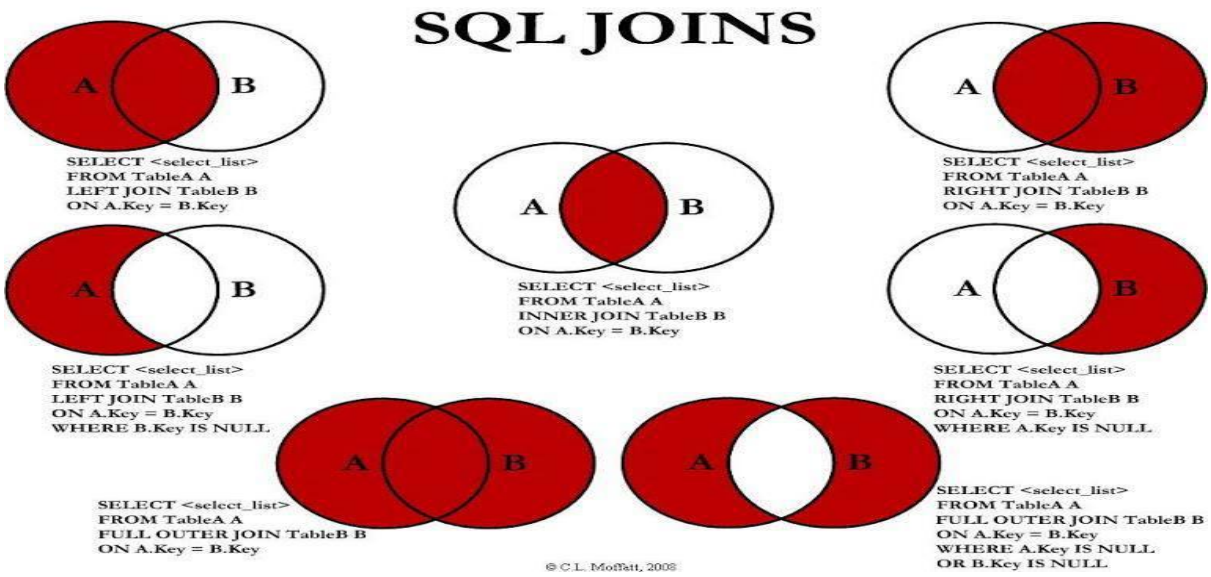
การส่งออกผลลัพธ์จากการประมวลผล เป็นไฟล์ต่างๆ เช่น Excel ,Text และอื่น



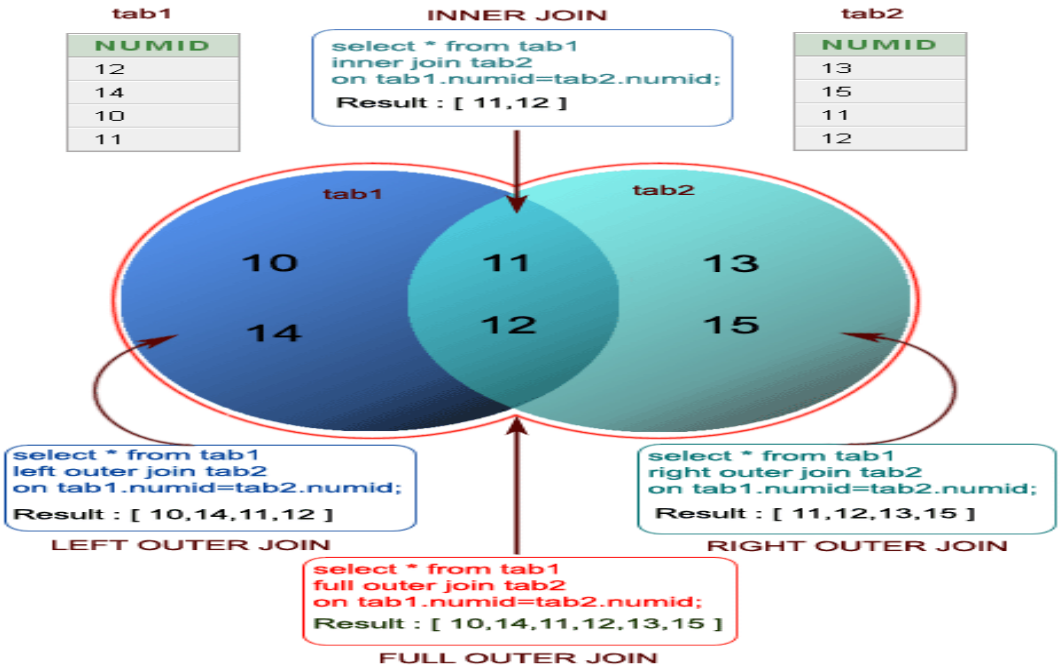
การ JOIN ใน SQL มีหลายรูปแบบดังนี้

LEFT JOIN , RIGHT JOIN , INNER JOIN , FULL JOIN

การ JOIN ต่างๆ จะต้องมีเงื่อนไขว่า ความเชื่อมโยงระหว่าง Table คืออะไร ซึ่งอยู่หลัง ON ซึ่งเป็น Key



ผลที่ได้จากการ JOIN ก็ได้ต่างกัน เมื่อดูจากรูป



การเขียน SQL JOIN อีกรูปแบบหนึ่ง คือ การใช้ comma(,) แทนการ JOIN แต่ทั้งนี้ก็ต้องมีการกำหนดเงื่อนไขในการเชื่อมโยงเช่นกัน แต่การกำหนดเงื่อนไข กำหนดใน WHERE แทน

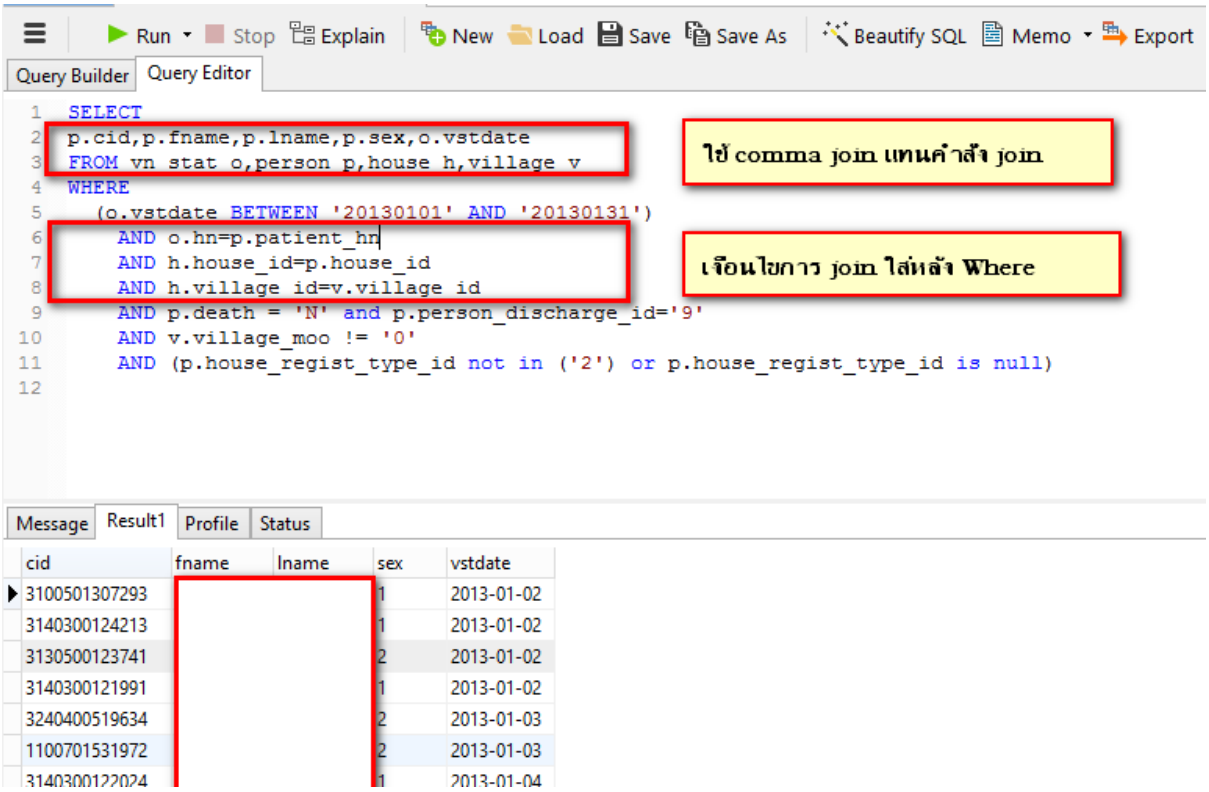
ตัวอย่างที่ ๙ แสดงข้อมูลแบบเชื่อมโยงแบบ Comma Join จงแสดงข้อมูลผู้รับบริการผู้ป่วยนอกระหว่างวันที่ ๑ มกราคม ๒๕๕๕ ถึงวันที่ ๓๑ มกราคม ๒๕๕๕ โดยเป็นคนในเขตของหน่วยบริการ (นับคนที่อยู่ในหมู่บ้านในเขต) เลขประชาชน ชื่อ นามสกุล เพศ วันที่รับบริการ

รายละเอียดข้อมูลที่เกี่ยวข้อง

Table: VN_STAT , PERSON , HOUSE , VILLAGE

Function: Count , IN ,DISTINCT

Field: cid , fname , lname , sex , vstdate , hn , patient_hn , house_id , village_id , house_regist_type_id , village_moo



แบบฝึกหัด

ให้แสดง SQL และ ผลลัพธ์ที่ได้

๑. จงหารายชื่อ (ชื่อ , นามสกุล , อายุ , เพศ และวันที่มารับบริการ) ผู้รับบริการผู้ป่วยนอกเป็นครั้งระหว่างวันที่ ๑ มกราคม ๒๕๕๕ ถึงวันที่ ๓๑ มกราคม ๒๕๕๕ ที่ป่วยด้วยโรคอุจาระร่วง (ICD A๐๙) โดยนับเป็นครั้ง โดยแสดง SQL และจำนวนผลลัพธ์ที่ได้

ตอบ

๒. จงหาจำนวนผู้รับบริการผู้ป่วยนอกเป็นครั้งระหว่างวันที่ ๑ มกราคม ๒๕๕๕ ถึงวันที่ ๓๑ มกราคม ๒๕๕๕ ที่ป่วยด้วยโรคอุจาระร่วง (ICD A๐๙) โดยนับเป็นครั้ง โดยแสดง SQL และจำนวนผลลัพธ์ที่ได้

ตอบ

๓. จงหารายชื่อ (ชื่อ , นามสกุล , อายุ , เพศ และวันที่มารับบริการ) ผู้รับบริการผู้ป่วยนอกเป็นครั้งระหว่างวันที่ ๑ มกราคม ๒๕๕๕ ถึงวันที่ ๓๑ มกราคม ๒๕๕๕ ที่ป่วยด้วยโรคอุจาระร่วง (ICD A๐๙) โดยนับเป็นคนและนับเฉพาะคนในเขตรับผิดชอบ(คนที่อยู่หมู่บ้านในเขต) โดยแสดง SQL และจำนวนผลลัพธ์ที่ได้

ตอบ

๔. จงหาจำนวนผู้รับบริการผู้ป่วยนอกเป็นครั้งระหว่างวันที่ ๑ มกราคม ๒๕๕๕ ถึงวันที่ ๓๑ มกราคม ๒๕๕๕ ที่ป่วยด้วยโรคอุจาระร่วง (ICD A๐๙) โดยนับเป็นคนและนับเฉพาะคนในเขตรับผิดชอบ(คนที่อยู่หมู่บ้านในเขต) โดยแสดง SQL และจำนวนผลลัพธ์ที่ได้

ตอบ

ภาคผนวก

ชื่อ Table และประเภทข้อมูลที่เก็บ ใน HosXp เฉพาะ Table ที่สำคัญๆ

Table	Description
Patient	เก็บข้อมูลพื้นฐานผู้ป่วย
Person	เก็บข้อมูลบุคคลตามทะเบียน ๑ ไม่ป่วยก็เก็บ ทุกคนในเขต
Vn_stat	เก็บข้อมูลการรับบริการผู้ป่วยนอกทุกราย(แบบสรุป)
Ovst	เก็บข้อมูลการรับบริการผู้ป่วยนอก
Ovstdiag	เก็บข้อมูลการให้รหัสการวินิจฉัยผู้ป่วยนอก
Opitemrece	เก็บรายละเอียดค่าใช้จ่ายผู้ป่วยนอกและผู้ป่วยใน
An_stat	เก็บข้อมูลการรับบริการผู้ป่วยในทุกราย(แบบสรุป)
Ipt	เก็บข้อมูลการรับบริการผู้ป่วยใน
Iptdiag	เก็บข้อมูลการให้รหัสการวินิจฉัยผู้ป่วยใน
Operation_list	เก็บข้อมูลการผ่าตัดผู้ป่วยนอกและผู้ป่วยใน
Operation_detail	เก็บรายละเอียดการผ่าตัดผู้ป่วยนอกและผู้ป่วยใน
Referin,Referout	เก็บข้อมูลการส่งต่อผู้ป่วย
Oapp	เก็บข้อมูลการนัดหมาย
Labor	เก็บข้อมูลการคลอด
Lab_order	เก็บข้อมูลการสั่งตรวจทางห้องปฏิบัติการ
Xray_order	เก็บข้อมูลการสั่ง X-RAY
Er_regist	เก็บข้อมูลการให้บริการที่ห้อง ER
Dental_care	เก็บข้อมูลการตรวจสุขภาพช่องปาก
Dtmain	เก็บข้อมูลการให้บริการเกี่ยวกับทันตกรรม
Health_med_service	เก็บข้อมูลการให้บริการแพทย์แผนไทย
House	เก็บข้อมูลหลังคาเรือน
Village	เก็บข้อมูลหมู่บ้าน
Person_anc	เก็บข้อมูลการฝากครรภ์
Person_epi	เก็บข้อมูลสร้างเสริมภูมิคุ้มกันโรค
Person_wbc	เก็บข้อมูลสร้างเสริมภูมิคุ้มกันโรค
Person_women	เก็บข้อมูลหญิงวัยเจริญพันธ์
Person_dmht_screen_summary	เก็บข้อมูลการคัดกรองเบาหวานความดัน
Person_chronic	เก็บข้อมูลผู้ป่วยโรคเรื้อรัง
Person_death,death	เก็บข้อมูลผู้เสียชีวิต
Village_school	เก็บข้อมูลโรงเรียน
Village_student	เก็บข้อมูลนักเรียน
Village_temple	เก็บข้อมูลวัด