

หลักสูตร

SQL เบื้องต้น

สำหรับเจ้าหน้าที่สาธารณสุข



โดย

นายจิรชาย นานบุญมี นักวิชาการสาธารณสุขชำนาญการ

สำนักงานสาธารณสุขอำเภอร่องขวาง

อำเภอร่องขวาง จังหวัดแพร่

หลักสูตร SQL เบื้องต้น สำหรับงานสาธารณสุข

ในปัจจุบัน การเก็บข้อมูลสุขภาพต่างๆในงานสาธารณสุข จะถูกเก็บไว้ด้วยโปรแกรมคอมพิวเตอร์ สำเร็จรูป ซึ่งจะถูส่งออกไปอยู่ในรูปของ 43แฟ้ม และถูกส่งไปยัง HDC (Health Data Center) ของจังหวัด ซึ่งเจ้าหน้าที่สาธารณสุขในปัจจุบันต้องดำเนินการในทุกๆเดือน แต่ข้อมูลสุขภาพเหล่านี้กลับไม่ถูกนำมาใช้ ประโยชน์ได้อย่างเต็มที่ ทั้งๆที่ข้อมูลสุขภาพเหล่านี้มีคุณค่าและประโยชน์อย่างยิ่ง หากเจ้าหน้าที่สาธารณสุข สามารถนำมาใช้ในการวางแผน, เฝ้าระวัง, แก้ไขปัญหาสุขภาพในพื้นที่ จะทำให้การทำงานสาธารณสุขในพื้นที่ ทำได้ง่ายยิ่งขึ้น ส่วนสาเหตุที่เจ้าหน้าที่สาธารณสุขไม่สามารถนำข้อมูลเหล่านี้มาใช้ได้อย่างเต็มที่ เนื่องจากยังขาดทักษะการนำข้อมูลต่างๆมาใช้งาน สำหรับโปรแกรมฐานข้อมูลสุขภาพที่เจ้าหน้าที่สาธารณสุขใช้งานเป็นประจำ ได้แก่ โปรแกรม Hosxp PCU, โปรแกรม JHCIS ซึ่งโปรแกรมเหล่านี้ใช้ฐานข้อมูลชนิด MySQL ดังนั้น ถ้าต้องการนำข้อมูลสุขภาพออกมาใช้งาน เราจำเป็นจะต้องเรียนรู้การเขียนคำสั่งในภาษา SQL นั่นเอง

สำหรับเนื้อหาในหลักสูตร SQL เบื้องต้นนี้ จะครอบคลุมตั้งแต่การทำความรู้จักกับฐานข้อมูล, MySQL และโปรแกรมที่ใช้จัดการฐานข้อมูล โดยเน้นการใช้งานจากสถานการณ์ปัจจุบันจริง ดังนั้นเราจะมาเรียนรู้จาก โปรแกรม Hosxp PCU ที่เราใช้งานกันเป็นประจำ

สำหรับคำสั่งที่จะใช้ใน SQL ในการศึกษาครั้งนี้ จะเน้นไปที่คำสั่ง SELECT เพื่อใช้สอบถามข้อมูล โดยเริ่มตั้งแต่แบบพื้นฐาน, แบบมีเงื่อนไข, แบบการเชื่อมโยงหลายๆตาราง รวมถึงการอธิบายถึงรายละเอียดของ ตารางในเบื้องต้น, เรื่องของตัวดำเนินการ(Operators) โดยเน้นเฉพาะการใช้งานเบื้องต้น เช่น and, or, in, between, <, >, = เป็นต้น

เอกสารประกอบการศึกษาภาษา SQL เบื้องต้นเล่มนี้ ผู้จัดทำได้รวบรวมเรียบเรียงตามความรู้และ ประสบการณ์ของผู้จัดทำ ซึ่งอาจจะไม่ถูกต้องตามหลักการเรียนภาษา SQL อย่างเป็นระบบ แต่ผู้จัดทำเชื่อว่า หากผู้เรียนได้ศึกษาเอกสารตามลำดับ จะสามารถทำความเข้าใจในภาษา SQL ในเบื้องต้นได้อย่างแน่นอน แต่ทั้งนี้ทั้งนั้น หากเอกสารเล่มนี้มีข้อผิดพลาดประการใด ผู้จัดทำก็ขออภัยมา ณ ที่นี้ด้วย

หลักสูตรนี้แบ่งการเรียนรู้ออกเป็น 2 วัน ดังนี้

การอบรมวันแรก

* ให้ทำบททดสอบความรู้ก่อนเรียน

** ชี้แจงวัตถุประสงค์การอบรมตามโครงการฯ

1. การเตรียมเครื่องคอมพิวเตอร์ให้พร้อมใช้งาน (ติดตั้งโปรแกรม Hosxp PCU พร้อมข้อมูลของ รพ.สต.)
2. ทำความรู้จักกับฐานข้อมูล คืออะไร, โปรแกรมที่มีฐานข้อมูลและไม่มี, ฐานข้อมูลยี่ห้อต่างๆ
3. ฐานข้อมูล MySQL คืออะไร ภาษา SQL คืออะไร ใช้ทำอะไร
4. การติดตั้งโปรแกรมจัดการฐานข้อมูล และการใช้งานเบื้องต้น (Navicat), การใช้งานเบื้องต้น
5. เครื่องมือจัดการฐานข้อมูลของ Hosxp PCU

6. เรียนรู้โครงสร้างของตารางข้อมูลเบื้องต้น เช่น Primary Key, ชนิดของข้อมูลที่ปรากฏในตาราง อะไรคือ Row และ Column
7. การดูรายละเอียดของตารางใน Navicat
8. ชนิดข้อมูลต่างๆที่สำคัญ
9. การดูรายละเอียดของตารางในคู่มือของ 43แพ้ม
10. สรุปสิ่งที่ควรรู้ก่อนที่จะเริ่มเขียนคำสั่ง SQL ทั้ง 5 ข้อ
11. เรียนรู้การใช้คำสั่ง SELECT เบื้องต้น
12. การเพิ่มคำสั่งในการปรับการแสดงผล LIMIT
13. การเพิ่มคำสั่งในการปรับการแสดงผล ORDER BY

การอบรมวันที่สอง

1. การใช้คำสั่ง SELECT แบบมีเงื่อนไข WHERE
2. การใช้คำสั่ง SELECT แบบมีเงื่อนไข WHERE พร้อมตัวดำเนินการ and, or, between, in
3. การใช้คำสั่ง SELECT แบบมีเงื่อนไข WHERE และเรียนรู้เครื่องหมาย <, >, =, <> ฯลฯ
4. การใช้คำสั่ง SELECT แบบมีเงื่อนไข WHERE และเรียนรู้ IS NULL, NOT NULL
5. เรียนรู้การใช้คำสั่งเชื่อมตาราง JOIN ประกอบด้วย LEFT JOIN, RIGHT JOIN, INNER JOIN
6. การใช้คำสั่ง SELECT โดยการเชื่อมตาราง 2 ตาราง ด้วย JOIN โดยเน้นการใช้งาน LEFT JOIN
7. การใช้คำสั่ง SELECT แบบมีเงื่อนไข WHERE + SUM + COUNT + GROUP BY

* ให้ทำแบบทดสอบความรู้หลังเรียน รวมถึงการทดสอบเชิงปฏิบัติ

หลักสูตรการอบรม SQL เบื้องต้น

เรียบเรียงและจัดทำโดย

นายจิรชาย นานบุญมี นักวิชาการสาธารณสุขชำนาญการ
สำนักงานสาธารณสุขอำเภอร่องกวาง จังหวัดแพร่

อนุญาตให้เผยแพร่ได้ แต่ไม่อนุญาตให้จำหน่ายหรือเพื่อหาประโยชน์ทางใดทางหนึ่งแก่บุคคลหนึ่งบุคคลใด



ฐานข้อมูล (Database)

ฐานข้อมูล คือ การเก็บข้อมูลที่มีความสัมพันธ์กันให้อยู่ในหัวเรื่องหรือจุดประสงค์อย่างใด อย่างหนึ่ง เช่น ชื่อ-นามสกุล ที่อยู่ หมายเลขโทรศัพท์เพื่อน สินค้าที่มีอยู่ในคลังสินค้า เป็นต้น ซึ่งเมื่อได้รับการจัดหมวดหมู่แล้วจะทำให้สามารถนำเอาส่วนประกอบนั้นๆ นำออกมาใช้ประโยชน์ได้

โปรแกรมที่มีฐานข้อมูลและไม่มี

ก่อนอื่นท่านทั้งหลายลองนึกถึงโปรแกรมต่างๆที่ท่านเคยใช้งานมาก่อน เช่น MS Word, โปรแกรมรูปภาพ, โปรแกรมดูหนัง, ฟังเพลง, โปรแกรมเครื่องคิดเลข และโปรแกรม Hosxp PCU ที่เราใช้กันอยู่ทุกวัน ท่านก็จะพบความแตกต่างถึงการใช้งาน ซึ่งโปรแกรมที่มีฐานข้อมูลชัดเจนจะเป็นโปรแกรมที่ใช้สำหรับเก็บข้อมูลที่เราต้องการและยังสามารถดูข้อมูลที่เราเก็บไว้ได้อีกด้วย

ฐานข้อมูลค่าย/แบบต่างๆ

ฐานข้อมูลที่มีให้ใช้ มีมากมายหลายแบบ หลายค่าย มีทั้งเสียเงินและฟรี ซึ่งแน่นอนว่าของฟรีแต่ดีก็มีในโลกนี้ ซึ่งก็คือฐานข้อมูลที่ใช้ในโปรแกรม Hosxp PCU คือ MySQL ซึ่งเป็นฐานข้อมูลแบบฟรีแวร์ จึงมีคนใช้งานอย่างแพร่หลาย และสามารถใช้งานในเชิงธุรกิจได้ด้วย(ไม่ต้องเสียค่าลิขสิทธิ์) และนี่คือตัวอย่างฐานข้อมูลค่ายต่างๆ



ฐานข้อมูล MySQL คืออะไร

MySQL (มายเอสคิวแอล) เป็นระบบจัดการฐานข้อมูล (Relational Database Management System) โดยใช้ภาษา SQL. แม้ว่า MySQL เป็นซอฟต์แวร์โอเพนซอร์ส แต่แตกต่างจากซอฟต์แวร์โอเพนซอร์สทั่วไป โดยมีการพัฒนาภายใต้บริษัท MySQL AB ในประเทศสวีเดน โดยจัดการ MySQL ทั้งในแบบที่ให้ใช้ฟรี และแบบที่ใช้ในเชิงธุรกิจ

MySQL สร้างขึ้นโดยชาวสวีเดน 2 คน และชาวฟินแลนด์ ชื่อ David Axmark, Allan Larsson และ Michael "Monty" Widenius. ปัจจุบันถูกซื้อสิทธิ์โดยบริษัทซันไมโครซิสเต็มส์ (Sun Microsystems, Inc.)

MySQL คืออะไร

MySQL คือ ระบบจัดการฐานข้อมูลที่มีหน้าที่เก็บข้อมูล เพื่อใช้ในการเก็บข้อมูลที่ได้จากระบบงานที่โปรแกรมเมอร์ได้สร้างขึ้น โดยใช้ภาษา SQL ซึ่ง MySQL จะทำหน้าที่เป็นตัวกลางระหว่างผู้ใช้กับฐานข้อมูล โดยเราสามารถติดต่อกับฐานข้อมูล MySQL โดยการเขียนโปรแกรมภาษาต่าง ๆ ได้เช่น PHP, VB เป็นต้น

ภาษา SQL คืออะไร

ภาษา SQL คือ ภาษาคอมพิวเตอร์ ที่ใช้ในการจัดการกับฐานข้อมูลโดยเฉพาะ เช่น สร้างฐานข้อมูล เพิ่มข้อมูล แก้ไขข้อมูล ลบข้อมูล เป็นต้น

ภาษา SQL เป็นภาษามาตรฐานสำหรับระบบฐานข้อมูล เป็นภาษาที่พัฒนาขึ้นมาโดยบริษัท IBM ภาษา SQL นี้ได้รับความนิยมอย่างสูง เพราะง่ายต่อความเข้าใจ โดยใช้ภาษาอังกฤษในการสื่อในเวลาเขียนคำสั่ง ดังนั้นหากเราแปลภาษาอังกฤษไปด้วยเวลาเขียนคำสั่ง ก็ทำให้เราเข้าใจภาษา SQL ได้ไม่ยาก

เรียนรู้โปรแกรมจัดการฐานข้อมูลสุดฮิต Navicat

โปรแกรม Navicat เป็นโปรแกรมจัดการฐานข้อมูลที่ถูกใช้งานกันอย่างแพร่หลาย เพราะด้วยหน้าที่ที่เรียบง่าย และการใช้งานที่เข้าใจได้ไม่ยาก

สามารถดาวน์โหลดตัวทดลองใช้งานได้ 30 วัน <http://www.navicat.com/download>



Navicat Premium

Navicat Premium is a multi-connections Database Administration Tool which allows you to bridge up to 6 databases within a single application: MySQL, MariaDB, SQL Server, SQLite, Oracle and PostgreSQL, create a quick and easy access to your databases all at once.
[Learn more >](#)

[Download Trial](#)

Navicat for MySQL (Support MariaDB)
[Learn more >](#)

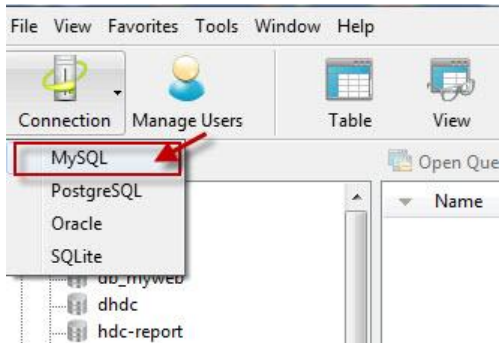
[Download Trial](#)

Navicat for MariaDB
[Learn more >](#)

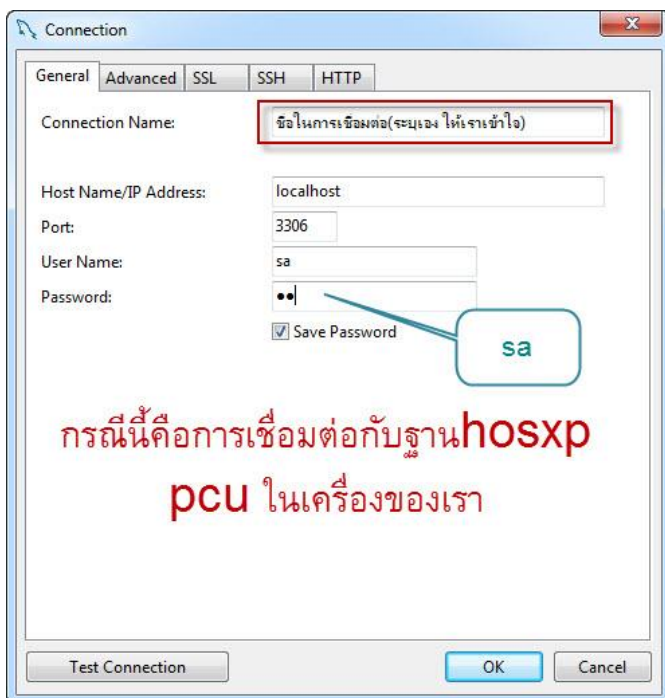
[Download Trial](#)

เมื่อติดตั้งเรียบร้อยแล้ว เราก็จะเข้าสู่ขั้นตอนต่อไป คือ วิธีใช้งานเบื้องต้น

การใช้งานโปรแกรม Navicat เบื้องต้น

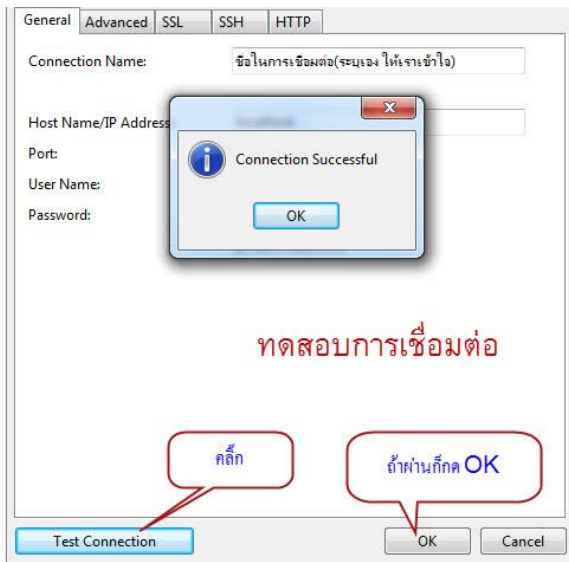


เมื่อเปิดโปรแกรม ให้เราเลือก Connection >> MySQL



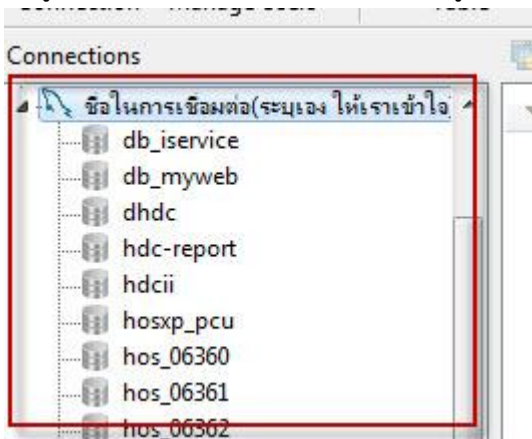
ก็จะขึ้นหน้าต่างเกี่ยวกับการเชื่อมต่อฐานข้อมูล เราต้องกรอกค่าให้ถูกต้อง เพราะฐานข้อมูลต่างๆมักจะมี Username และ Password เสมอ รวมถึง Port ที่ใช้งาน ซึ่งสำหรับฐานข้อมูล Hosxp PCU ในเครื่องของเรา ให้เราตั้งค่าตามรูปครับ

***สิ่งน่ารู้** คำว่า localhost ในช่อง Host Name/IP Address เราสามารถแทนด้วยคำว่า 127.0.0.1 ได้ด้วยนะ ซึ่งทั้ง 2 คำหมายถึง เครื่องที่เราใช้งานอยู่ในขณะนั้นนั่นเอง

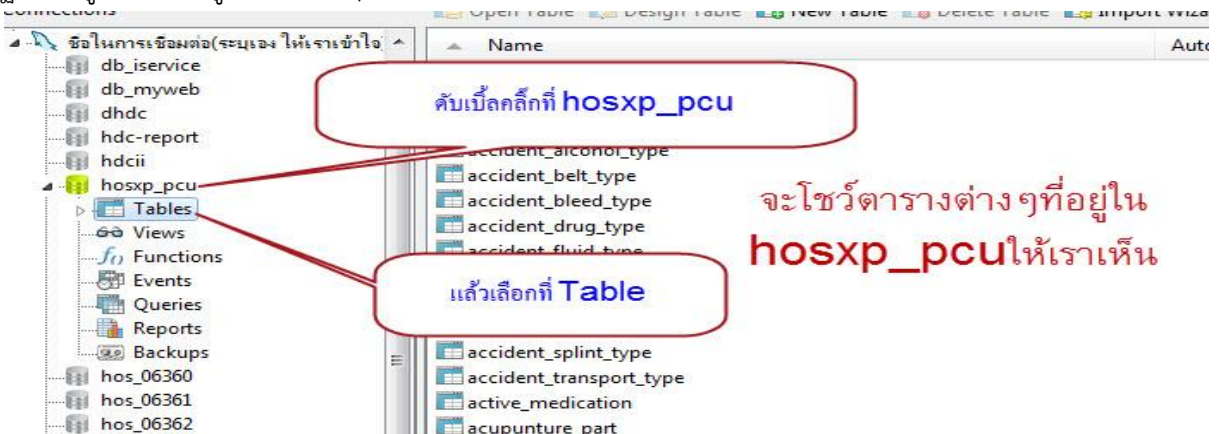


เมื่อเราตั้งค่าเรียบร้อยแล้ว ให้เรากด Test Connection เพื่อทดสอบการเชื่อมต่อว่า ได้หรือไม่ ถ้าสามารถเชื่อมต่อได้ ก็จะมีขึ้น Connection Successful ดังภาพ แล้วเราก็กด OK ได้เลย

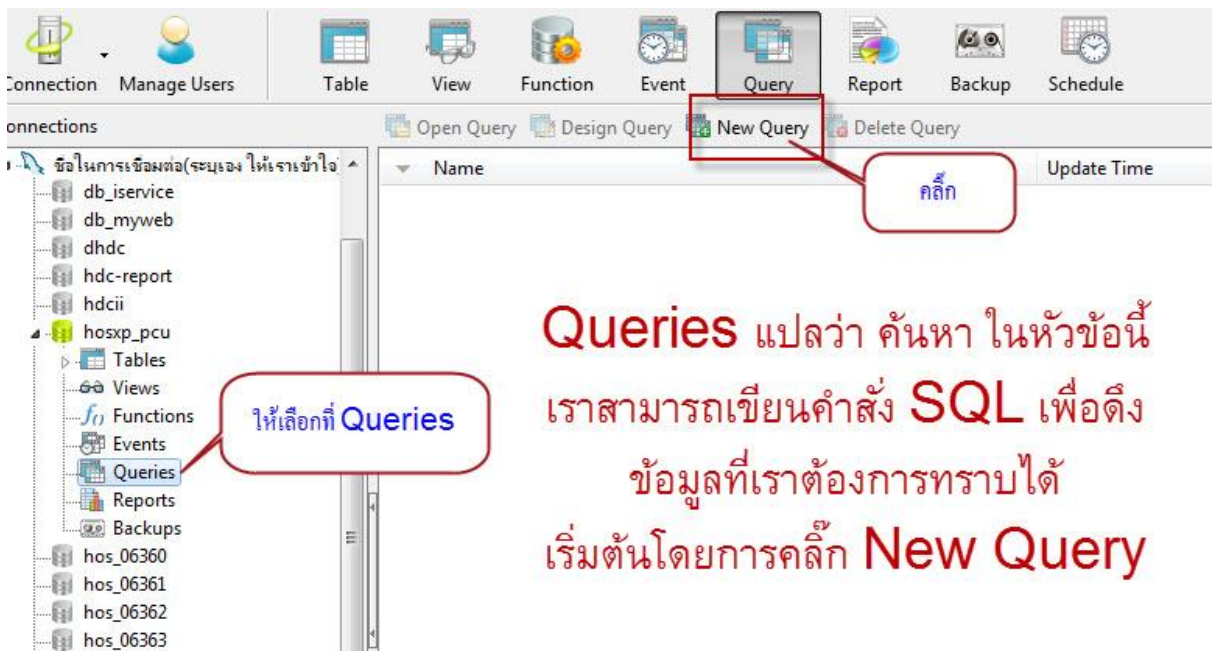
เมื่อเชื่อมต่อฐานข้อมูลได้แล้ว จะเห็นว่าชื่อในการเชื่อมต่อของเราโผล่ขึ้นมาดังภาพ และมีชื่อฐานข้อมูลต่างๆ ที่อยู่ในเครื่องเราโผล่มา จะมาก-น้อย ก็อยู่ที่เครื่องของเรามีฐานข้อมูลกี่ฐานครับ



ถึงตอนนี้เรายังไม่ได้เข้าไปดูในฐานข้อมูลใดๆ ดังนั้นเราต้องเข้าไปดูในฐานข้อมูลชื่อ hosxp_pcu ซึ่งเป็นฐานข้อมูลที่เก็บข้อมูลของ Hosxp PCU ของเราเอาไว้นั่นเอง ทำตามภาพเลยครับ



หัวข้อ Tables จะเก็บตารางต่างๆในฐานข้อมูลนี้ครับ เมื่อเลือกแล้วจะโชว์ตารางด้านขวามือ และอีกหัวข้อหนึ่งที่เราต้องใช้งานก็คือ หัวข้อ Queries แปลว่า ค้นหา ดังนั้นเราอยากได้ข้อมูลอะไร แบบไหน เราสามารถหาได้ที่นี้ครับ แต่มีข้อแม้ว่าเราต้องเขียนคำสั่ง SQL เป็นเสียก่อนนะครับ



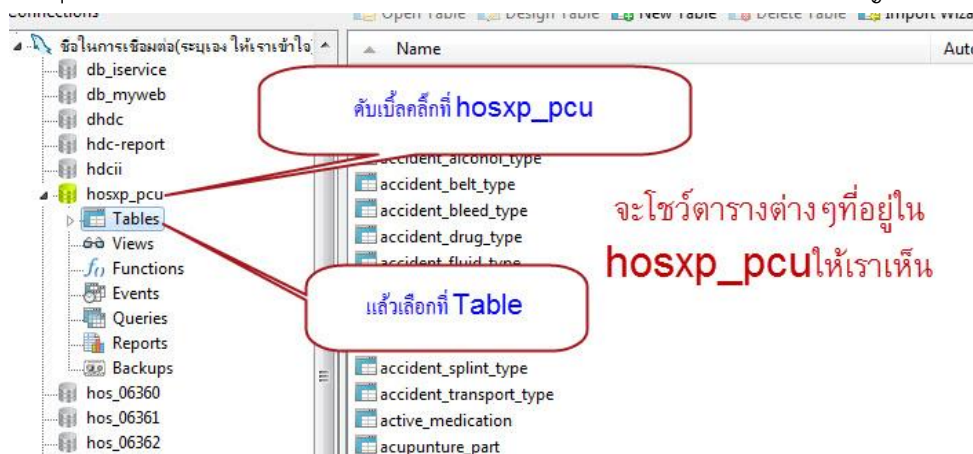
ก่อนที่จะไปในเรื่องอื่นๆ เรามาดูเครื่องมือจัดการฐานข้อมูลที่ Hosxp PCU มีให้มาแล้วกันก่อนครับ

เครื่องมือจัดการฐานข้อมูลของ Hosxp PCU

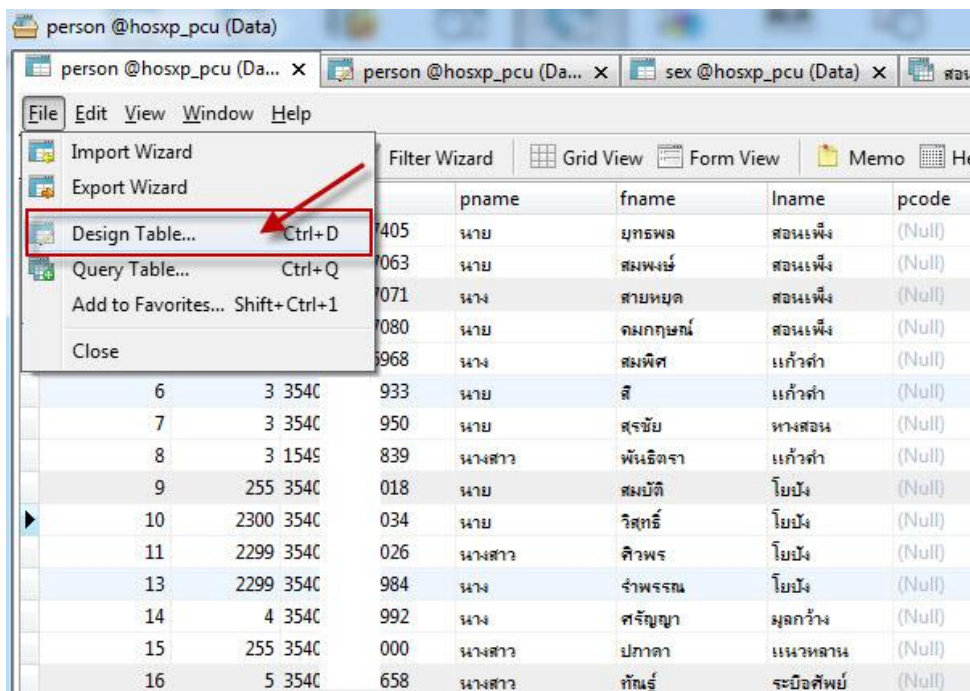
จริงๆแล้วในโปรแกรม Hosxp PCU นั้นมีโปรแกรมจัดการฐานข้อมูลของ Hosxp PCU อยู่แล้ว โดยที่เราไม่ต้องใช้โปรแกรมจัดการฐานข้อมูลอื่นๆเลย แต่ทั้งนี้ทั้งนั้น ขึ้นอยู่กับผู้ใช้งานว่าจะถนัดแบบไหน สะดวกแบบไหนมากกว่า

house_regist_type_id	house_regist_type_name	hos_guid	export_code
1	มีชื่อในทะเบียนบ้าน และอยู่อาศัยจริง	{E17A8F67-238D-11E0-9E8E-F2033505784D}	1
2	มีชื่อในทะเบียนบ้านแต่ไม่ได้อยู่อาศัย	{E17A9230-238D-11E0-9E8E-F2033505784D}	2
3	ไม่มีชื่อในทะเบียนบ้านแต่มาอยู่อาศัย	{E17A9365-238D-11E0-9E8E-F2033505784D}	3
4	บุคคลนอกเขต		4
5	คนร่อนไม่มีที่พำอาศัย(ในเขต)	{E17A948B-238D-11E0-9E8E-F2033505784D}	5

เมื่อได้ทราบถึงเครื่องมือสำหรับจัดการฐานข้อมูลกันไปแล้ว เราก็จะมาเจาะลึกถึงรายละเอียดของตารางต่างๆ กันเสียก่อนว่า ตาราง(Table)แต่ละตารางนั้นประกอบด้วยอะไรบ้าง สำคัญอย่างไร

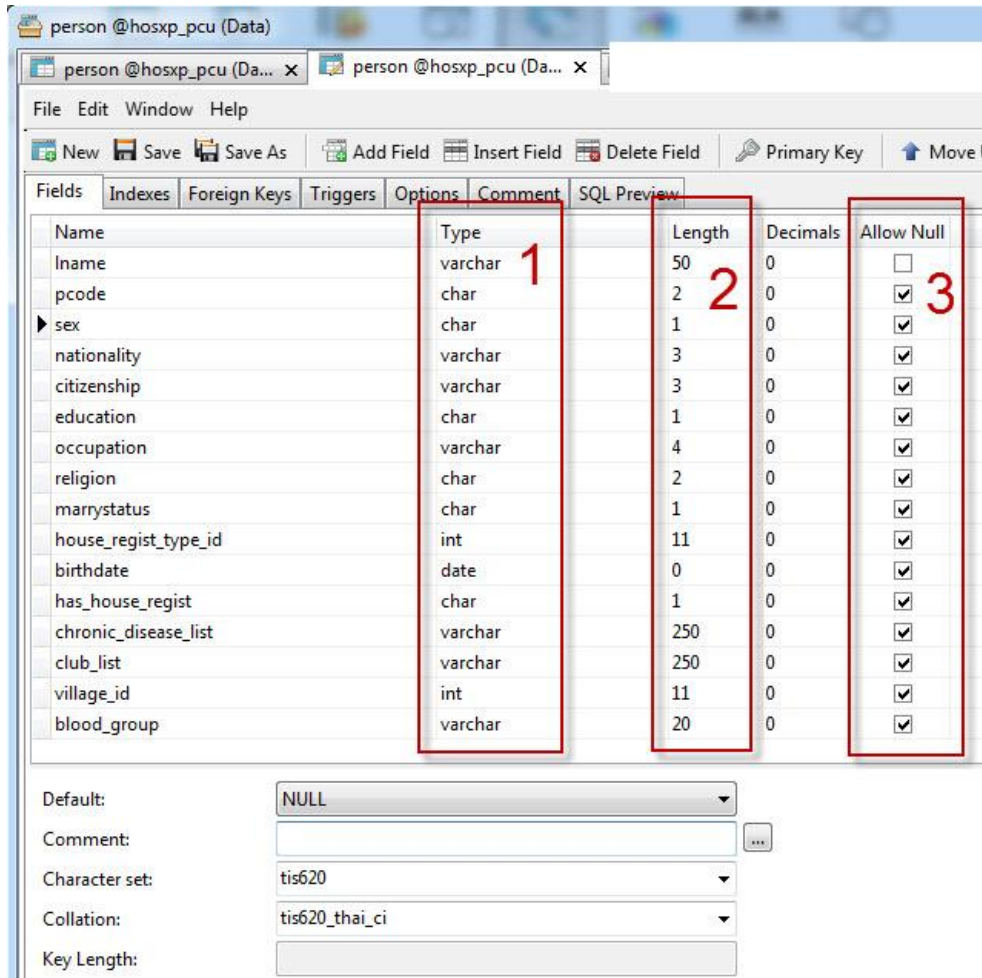


ให้เราไปที่หัวข้อ Tables แล้วเลือกตารางด้านขวามือ ชื่อว่า person ก็จะมีปรากฏข้อมูลของตารางออกมาให้เรา เห็นว่ามีข้อมูลอะไรบ้าง



ต่อจากนี้ให้เราเลือกที่ File >> Design Table เพื่อดูรายละเอียดการตั้งค่าของตารางนี้ ว่าคนที่เค้าสร้างตาราง person เค้าได้กำหนดให้แต่ละ Field มีค่าอย่างไร เป็นข้อมูลประเภทไหนบ้าง ซึ่งเราก็ต้องเรียนรู้จุดนี้เป็นพื้นฐาน เพื่อให้เราสามารถดูรายละเอียดของแต่ละตารางได้ โดยจะมีผลต่อการใช้คำสั่ง SQL ในการเรียนรู้ลำดับต่อไป

เรามาดูรายละเอียดของตารางกันครับ



ในหลักสูตรนี้ เราจะดูแค่ใน Tab ชื่อ Fields เท่านั้น ซึ่งประกอบไปด้วยหัวตาราง ดังนี้

- **Name** คือ ชื่อของ Column ต่างๆที่ปรากฏในตารางนี้
- **Type** คือ ชนิดของข้อมูล ได้แก่
 - varchar, char คือ ข้อมูลที่ถูกเก็บเป็นตัวอักษร
 - int คือ ข้อมูลที่ถูกเก็บเป็นค่าตัวเลข
 - date คือ ข้อมูลที่เก็บเป็นรูปแบบของวันที่ ซึ่งจะเป็นลักษณะ '2015-08-16' ประมาณนี้

*หลักสูตรนี้ ยังไม่ต้องใส่ใจอะไรกับชนิดของข้อมูลมาก ให้เข้าใจตามนี้ก็พอครับ

- **Length** คือ แปลตรงตัวว่าความยาว ซึ่งก็คือ ความยาวของข้อมูล เช่น ถ้าเป็นข้อมูลชนิด varchar แล้วมี length = 50 หมายถึง เราสามารถพิมพ์ตัวอักษรได้ติดกัน 50 ตัวในช่องนี้ แต่ถ้าเป็น int ที่มี length = 2 ก็หมายถึง ช่องนี้เก็บค่าตัวเลขที่มีค่าสูงสุดแค่หลักสิบ นั่นก็คือสูงสุดได้แค่ 99 เท่านั้น

- **Decimals** คือ แปลตรงตัวว่า ทศนิยม นั่นก็คือ สามารถใส่จุดทศนิยมได้ที่จุดนั่นเอง

- **Allow Null** คือ แปลตรงตัว(อีกแล้ว) ว่า ยอมให้ช่องนี้เป็นค่าว่างได้ไหม? ถ้าไม่มีติ๊กถูก เช่น ช่อง lname แสดงว่าช่อง lname นั้นไม่สามารถปล่อยว่างได้ แต่ถ้ามีการติ๊กถูก แสดงว่าช่องนั้นๆสามารถปล่อยว่างได้

ต่อไปเรามาดูในเรื่องของ Column กับ Row เมื่อใหม่มักจะจำผิดจำถูก จำสลับกันเป็นประจำ

Column หรือ ข้อมูลในแนวตั้ง จะมีจุดสังเกต* โดยจะมีชื่อของ Column เสมอ

แนวตั้ง คือ Colum ให้สังเกตว่า คอลัมน์ จะมีชื่อเสมอ

person_id	house_id	cid	pname	fname	lname	pcode	sex	
1	2	5	05	นาย	ยุทธพล	สอนเพ็ง	(Null)	1
2	2	3	63	นาย	สมพงษ์	สอนเพ็ง	(Null)	1
3	2	3	71	นาง	สาย	สอนเพ็ง	(Null)	2
4	2	3	80	นาย	คณ	สอนเพ็ง	(Null)	1
5	3	3	68	นาง	สน	สอนเพ็ง	(Null)	1
6	3	3	33	นาย	สี	สอนเพ็ง	(Null)	1
7	3	3	50	นาย	ศร	สอนเพ็ง	(Null)	1
8	3	1	39	นางสาว	พิ	สอนเพ็ง	(Null)	1
9	255	3	18	นาย	สน	สอนเพ็ง	(Null)	1
10	2300	3	34	นาย	วิ	สอนเพ็ง	(Null)	1
11	2299	3	26	นางสาว	ศิริ	สอนเพ็ง	(Null)	1
13	2299	3	84	นาง	จำพรจ	สอนเพ็ง	(Null)	1
14	4	3	92	นาง	ศรัญญา	มุลกว้าง	(Null)	2

Column คือ ชนิดข้อมูลต่างๆที่เราเก็บไว้ในตาราง เช่น คำนามหน้า pname, ชื่อ fname, นามสกุล lname เป็นต้น

Row หรือ แถว สามารถเรียกได้อีกอย่างว่า ระเบียบข้อมูลหรือเรคคอร์ด (Record) หมายถึงกลุ่มของเขตข้อมูลที่มีความสัมพันธ์กัน แต่ละข้อมูลสามารถแสดงคุณลักษณะและคุณสมบัติของเรื่องเดียวกัน

แนวอน เรียกว่า ROW หรือแถว มักจะมีจำนวนไม่จำกัด ไม่มีชื่อแถว ถ้ามีข้อมูลมาก แถวก็จะมามาก ให้สังเกตง่ายกว่า 1 แถว เก็บข้อมูล 1 คน โดยแต่ละคนก็มีข้อมูลหลายอย่าง เช่น ชื่อ, เพศ, ที่อยู่, เลขประจำตัว ฯลฯ

person_id	house_id	cid	pname	fname	lname	pcode	sex
1	2	405	นาย	ยุทธพล	สอนเพ็ง	(Null)	1
2	2	063	นาย	สมพงษ์	สอนเพ็ง	(Null)	1
3	2	071	นาง	สายหยุด	สอนเพ็ง	(Null)	2
11	2299	020	นางสาว	ศรพร	เยมิง	(Null)	2
13	2299	984	นาง	จำพรจณ	โยมิง	(Null)	2

สิ่งน่ารู้ * เขตข้อมูลหรือฟิลด์ (Field) หมายถึง ข้อมูลใดข้อมูลหนึ่งที่ต้องการเก็บเพื่อนำมาใช้ประโยชน์ได้

	Column 1	Column 2
Row 1	Field	Field
Row 2	Field	Field
Row 3	Field	Field

สรุปสิ่งที่ควรรู้ก่อนที่จะเริ่มเขียนคำสั่ง SQL ทั้ง 5 ข้อ

1. รู้จักชื่อของตาราง

เช่น ตาราง person, ตาราง sex

2. รู้จักว่าตารางไหน เก็บข้อมูลอะไร

เช่น ตาราง person เก็บข้อมูลประชากรแต่ละคน ซึ่งมีรายละเอียดต่างๆ เช่น เพศ, สถานะสมรส ฯลฯ

3. รู้จักชื่อ Column ของตาราง

เช่น ตาราง person มี Column ชื่อ fname เก็บข้อมูลชื่อของประชากร, lname เก็บข้อมูลนามสกุลของประชากร, sex เก็บข้อมูลเพศของประชากร เป็นต้น

4. รู้จักชนิดของข้อมูลที่ถูกเก็บในแต่ละ Column

เช่น Column ชื่อ fname เก็บข้อมูลเป็นแบบตัวอักษร, sex เก็บข้อมูลเป็นตัวเลข 1 กับ 2 , birthdate เก็บข้อมูลเป็นวันที่ '2013-08-25' เป็นต้น

5. รู้จักใช้เครื่องมือ/โปรแกรม ที่เราจะป้อนคำสั่ง SQL

เช่น Navicat, SQL Query ใน Hosxp PCU เป็นต้น



มาเริ่มการเขียนคำสั่ง SQL สำหรับผู้เริ่มต้นกันเถอะ

คำสั่ง SQL มีมากมายหลายคำสั่ง ล้วนเป็นคำสั่งสำหรับจัดการกับฐานข้อมูล MySQL ทั้งสิ้น ไม่ว่าจะ เป็น CREATE DATABASE, CREATE TABLE, DROP DATABASE, DROP TABLE, DELETE, UPDATE, SELECT เป็นต้น แต่สำหรับหลักสูตรนี้ เราเน้นที่จะนำข้อมูลที่เรามีอยู่ ออกมาแสดงตามที่เราต้องการด้วยคำสั่ง SQL ที่มีชื่อว่า SELECT (แปลว่าเลือก) ถึงแม้ว่าจะใช้เพียงคำสั่งหลักคำสั่งเดียว แต่ก็ยังมีรายละเอียดปลีกย่อยอีกไม่น้อยที่เราต้องศึกษา

หลักสูตรนี้สร้างขึ้นเพื่อสำหรับผู้เริ่มต้นศึกษา SQL เป็นสำคัญ ดังนั้นจะพยายามอธิบายตามสเต็ปอย่างละเอียด จากง่ายที่สุดแล้วค่อยๆเพิ่มระดับความยากขึ้นไป

คำสั่งเลือกดูข้อมูล หรือ คำสั่ง SELECT

มีโครงสร้างพื้นฐานการเขียน ดังนี้

SELECT ชื่อ Column หรือ *
FROM ชื่อตาราง

หมายเหตุ * หมายถึงทุกๆ Column

กค Run

```

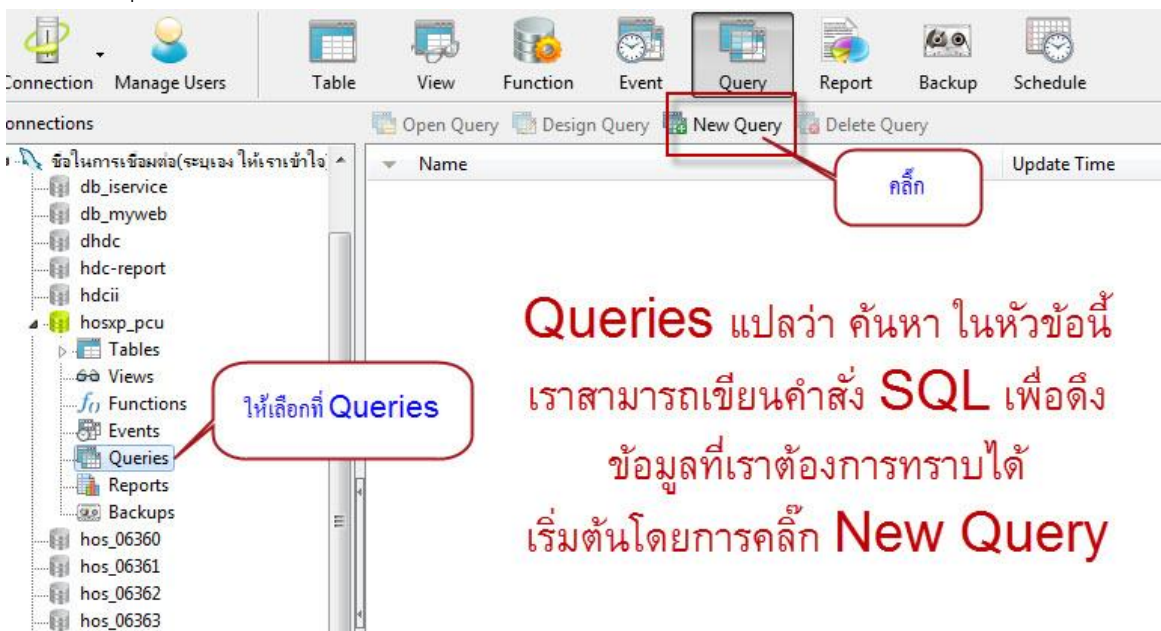
1 #Exp ตัวอย่างที่:1
2
3 SELECT *
4 FROM person
5
6
7

```

person_id	house_id	cid	pname	fname	lname
1	2	5540200027405	นาย	ยุทธพล	สอนเพ็ญ
2	2	3540200017063	นาย	สมพงษ์	สอนเพ็ญ
3	2	3540200017071	นาง	สายหยุด	สอนเพ็ญ
4	2	3540200017080	นาย	คมกฤษณ์	สอนเพ็ญ
5	3	3540200026968	นาง	สมจิต	แก้วฟ้า

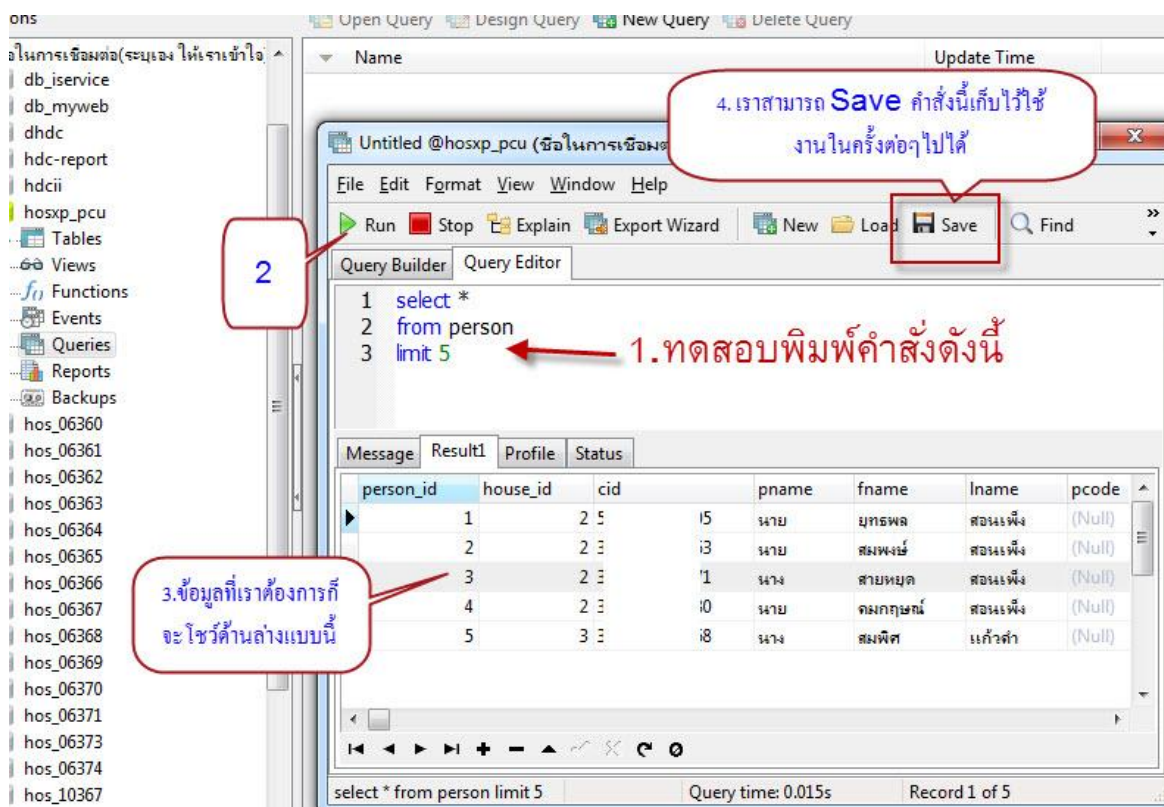
ต่อไปนี้จะสอนถึงการเขียนคำสั่ง SQL เพื่อดึงข้อมูล ด้วยโปรแกรม Navicat

ก่อนอื่นให้ทุกท่านไปที่ โปรแกรม Navicat >> Queries ดังภาพ



Queries แปลว่า ค้นหา ในหัวข้อนี้ เราสามารถเขียนคำสั่ง SQL เพื่อดึงข้อมูลที่เรากำลังต้องการทราบได้ เริ่มต้นโดยการคลิก **New Query**

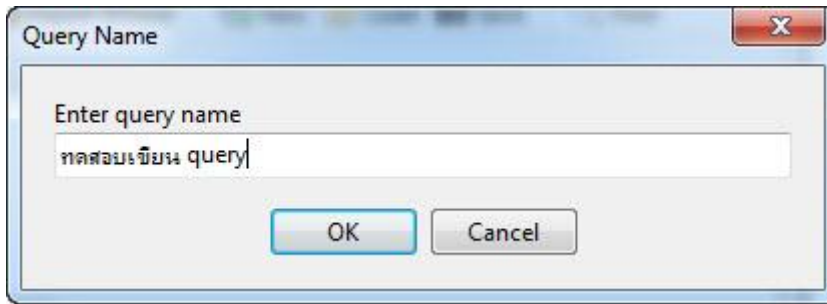
ให้ลองพิมพ์คำสั่ง ตามภาพ



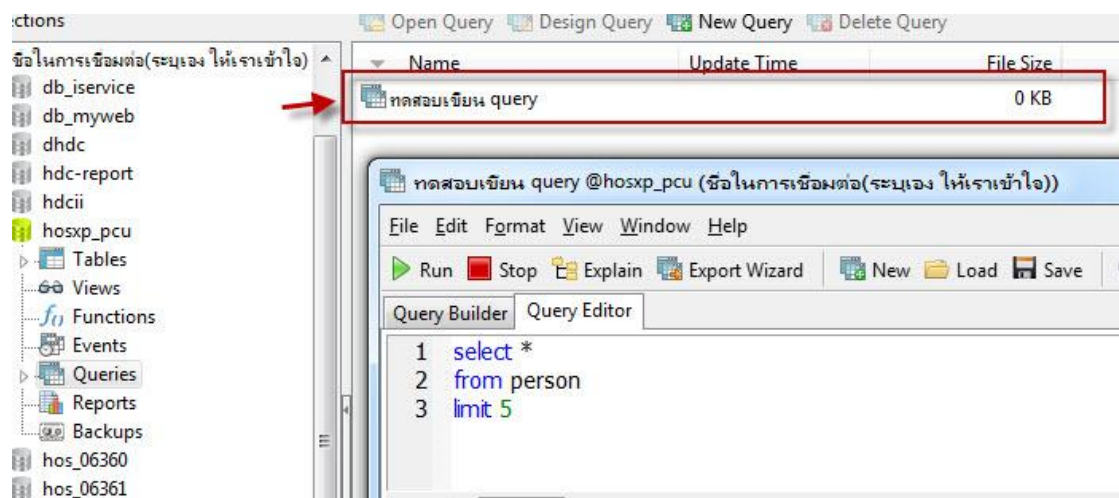
หมายเหตุ *

คำสั่งเพิ่มเติม Limit (แปลว่าจำกัด) เอาไว้จำกัดการเรียกดูข้อมูล ว่าต้องการดูกี่แถว ดังนั้นคำสั่ง Limit เป็นคำสั่งประเภทปรับการแสดงผล นั่นเอง

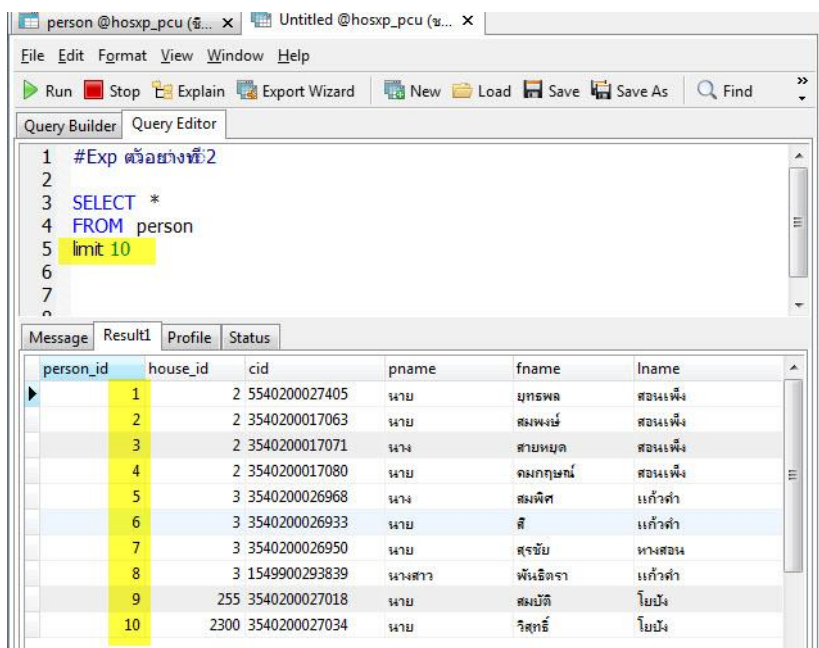
การเขียนคำสั่ง SQL ผ่านโปรแกรม Navicat มีข้อดีคือ สามารถเก็บคำสั่งไว้ใช้งานในครั้งต่อไป หรือ สามารถส่งคำสั่งที่เรา SAVE ไว้ให้คนอื่นได้อีกด้วย เพียงเรากด SAVE ดังภาพเท่านั้น



ให้เราระบุชื่อคำสั่ง (ควรเขียนให้สื่อถึงสิ่งที่เราต้องการ) แล้วก็กด OK ได้เลย จะเห็นว่ามีไฟล์ไฟล์ขึ้นมาแล้วครับ เวลาใช้งานก็แสนง่าย แค่ดับเบิลคลิกขึ้นมาแล้วกด Run เท่านั้นเองครับผม



✓ การเพิ่มคำสั่งในการปรับการแสดงผล LIMIT



จากตัวอย่างจะพบว่า แสดงเพียง 10 อันดับแรกเท่านั้นครับ

✓ การเพิ่มคำสั่งในการปรับการแสดงผล ORDER BY

ORDER BY คือ การทำให้ผลการแสดงถูกเรียงลำดับตามที่เรต้องการนั่นเอง โดยมี 2 แบบ คือ

- แบบ ASC คือ เรียงจากน้อยไปหามาก
- แบบ DESC คือ เรียงจากมากไปหาน้อย

รูปแบบคำสั่งคือ ORDER BY [ชื่อColumn] [รูปแบบ ASC, DESC]

ตัวอย่าง แบบ ASC

The screenshot shows a SQL query editor with the following code:

```

1 #Exp ตัวอย่างที่3.1
2
3 SELECT *
4 FROM person
5 ORDER BY person_id ASC
6 limit 10
7
8 #ASC คือ เรียงจากน้อยไปหามาก
9 #DESC คือ เรียงจากมากไปหาน้อย
10

```

The results table is as follows:

person_id	house_id	cid	pname	fname	lname
1	2	5540200027405	นาย	ยุทธพล	สองเฒ่า
2	2	3540200017063	นาย	สมพงษ์	สองเฒ่า
3	2	3540200017071	นาง	สายหยุด	สองเฒ่า
4	2	3540200017080	นาย	คมกฤษณ์	สองเฒ่า
5	3	3540200026968	นาง	สมพิศ	แก้วดำ
6	3	3540200026933	นาย	สี	แก้วดำ
7	3	3540200026950	นาย	สุรชัย	ทางสอง
8	3	1549900293839	นางสาว	พันธิตรา	แก้วดำ
9	255	3540200027018	นาย	สมบัติ	โยปิง
10	2300	3540200027034	นาย	วิสิทธิ์	โยปิง

ตัวอย่าง แบบ DESC

The screenshot shows a SQL query editor with the following code:

```

1 #Exp ตัวอย่างที่3.2
2
3 SELECT *
4 FROM person
5 ORDER BY person_id DESC
6 limit 10
7
8 #ASC คือ เรียงจากน้อยไปหามาก
9 #DESC คือ เรียงจากมากไปหาน้อย
10

```

The results table is as follows:

person_id	house_id	cid	pname	fname	lname
13506	2216	1909800930671	นาย	ศิวกร	บุญส่ง
13505	2216	1800900151332	นางสาว	อลิตา	บุญชูวงศ์
13504	2204	3540200457691	นาง	เสาวภา	คำภีระ
13501	2216	1639900207613	นาย	สุทธิพงศ์	แช่เกอ
13500	976	1549900936977	เด็กชาย	ปภัทกร	มูลเมือง
13499	2204	1549900736561	เด็กหญิง	ณัชชา	ผลสุด
13498	2204	3749900224441	นางสาว	จกาทัดน์	ชะจิงจุ
13497	2204	3510400495295	นาย	บุญ	พิมพ์ย
13496	2004	1549900932068	เด็กชาย	ธนกฤต	กัมภฤษณ์
13495	2204	1660600126237	นางสาว	วิภาจัตน์	เพียรดี



การดึงข้อมูลแบบมีเงื่อนไข ด้วยคำสั่ง WHERE

การที่เราเขียนคำสั่ง SQL เพื่อดึงข้อมูลนั้น การกำหนดเงื่อนไขจำเป็นอย่างยิ่ง เพราะเวลาเราอยากทราบข้อมูลต่างๆในฐานข้อมูล เราคงจะไม่ดูข้อมูลทั้งหมดอย่างแน่นอน แต่เราจะเลือกดูข้อมูลที่เราอยากจะดูเท่านั้น การเขียนคำสั่ง SQL เพื่อเลือกดูเฉพาะเงื่อนไขที่ต้องการ เราสามารถใช้คำสั่ง WHERE พร้อมกับตัวดำเนินการ or, and, between และใช้เครื่องหมายเปรียบเทียบเช่น <, >, = ประกอบ จะทำให้เราสามารถดึงข้อมูลที่ต้องการได้อย่างถูกต้อง, ตรงใจ, ตรงตามความต้องการ

เรามารู้จักการใช้คำสั่ง SELECT แบบมีเงื่อนไข WHERE

มีโครงสร้างพื้นฐานการเขียน ดังนี้

```
SELECT ชื่อ Column หรือ *
FROM ชื่อตาราง
WHERE เงื่อนไขที่ต้องการ
```

หมายเหตุ * หมายถึงทุกๆ Column

ตัวอย่างการใช้ WHERE เพื่อดูข้อมูลประชากรที่มีค่า sex = 1 เท่านั้น

* sex = 1 หมายถึง ผู้ชาย , = 2 หมายถึง ผู้หญิง

The screenshot shows a SQL query editor with the following query:

```
1 #Exp ตัวอย่างที่4 การใช้where
2
3 SELECT pname, fname, lname , sex
4 FROM person
5 WHERE sex = '1'
6
7 limit 10
8 #WHERE ใช้กำหนดเงื่อนไขที่ต้องการ
9
10
```

The results table shows the following data:

pname	fname	lname	sex
นาย	ยุทธพล	สอนเพ็ง	1
นาย	สมพงษ์	สอนเพ็ง	1
นาย	คมกฤษณ์	สอนเพ็ง	1
นาย	สี	แก้วคำ	1
นาย	ศุภชัย	ทางสอน	1
นาย	สมบัติ	โยบึง	1
นาย	วิสุทธิ์	โยบึง	1
นาย	พิไชย	วงศ์กา	1
นาย	เนตร	ยานะวีน	1
นาย	ประวิทย์	สมพงษ์	1

✓ เรียนรู้ตัว Operators แบบต่างๆ

ตัว Operators ใน SQL มีหลักๆ 3 ประเภท ที่เราสามารถใ้กำหนดรายละเอียดในเงื่อนไขใน WHERE มีดังนี้

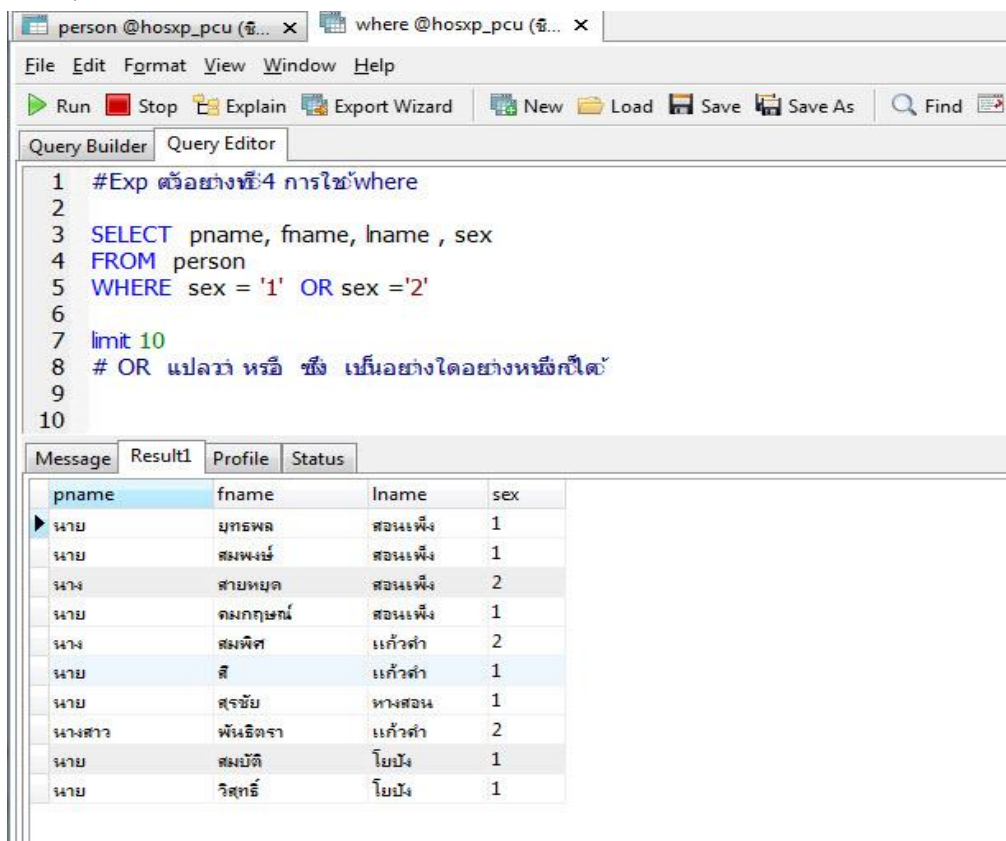
- Arithmetic Operators: คือ เครื่องหมายทางคณิตศาสตร์ +, -, *, / (**หลักสูตรนี้ยังไม่เรียน)
- Logical Operators: คือ คำภาษาอังกฤษ เช่น or, and, between, in
- Comparison Operators: คือ เครื่องหมายเปรียบเทียบต่างๆ เช่น <, >, =, <>, !=

➤ Logical Operators

- OR แปลว่า "หรือ" ซึ่งก็หมายถึง อย่างใดอย่างหนึ่ง เช่น เป็นเพศชาย หรือ เพศหญิง ซึ่งสามารถเขียนเป็นเงื่อนไขหลัง WHERE ได้ดังนี้

WHERE sex = '1' OR sex = '2'

Example.



```

1 #Exp ตัวอย่างที่ 4 การใช้ where
2
3 SELECT pname, fname, lname , sex
4 FROM person
5 WHERE sex = '1' OR sex = '2'
6
7 limit 10
8 # OR แปลว่า หรือ ซึ่ง เ็นอย่างใดอย่างหนึ่งก็ได้
9
10

```

pname	fname	lname	sex
นาย	ยุทธพล	สองแฉ่ง	1
นาย	สมพงษ์	สองแฉ่ง	1
นาง	สายหยุด	สองแฉ่ง	2
นาย	คมกฤษณ์	สองแฉ่ง	1
นาง	สมพิศ	แก้วคำ	2
นาย	สี	แก้วคำ	1
นาย	สุชัย	ทางสอง	1
นางสาว	พันธิตจา	แก้วคำ	2
นาย	สมบัติ	โยม้ง	1
นาย	วิสุทธิ์	โยม้ง	1

*ข้อควรรู้ จากการเรียนรู้ที่ผ่านมา จะเห็นว่าโครงสร้างที่เราสามารถเขียนได้ในตอนนี้ มีดังนี้

SELECT ชื่อ Column หรือ *
 FROM ชื่อตาราง
 WHERE เงื่อนไขที่ต้องการ
 ORDER BY [ชื่อColumn] [รูปแบบ ASC, DESC]
 LIMIT จำนวนแถวที่ต้องการแสดง

- AND แปลว่า "และ" ซึ่งก็หมายถึง ต้องเป็นทั้ง 2 อย่าง เช่น เป็นเพศชาย และ เพศหญิง ซึ่งสามารถเขียนเป็นเงื่อนไขหลัง WHERE ได้ดังนี้

WHERE sex = '1' AND sex = '2'

Example.

The screenshot shows a SQL query editor with the following code:

```

1 #Exp ตัวอย่างที่4 การใช้where
2
3 SELECT pname, fname, lname , sex
4 FROM person
5 WHERE sex = '1' AND sex = '2'
6 |
7 limit 10
8 # AND แปลว่า และ ซึ่ง ต้องเป็นทั้ง 2 อย่าง
9
10

```

The result set is empty, showing a table with columns pname, fname, lname, and sex, and a single row with (Null) for all values.

ไม่มีผลลัพธ์ เพราะคงไม่มีใครเป็นทั้งผู้หญิงและผู้ชาย

- BETWEEN แปลว่า "ระหว่าง" ซึ่งก็หมายถึง กำหนดเงื่อนไขให้มีค่าอยู่ระหว่าง xx และ yy ซึ่งสามารถเขียนเป็นเงื่อนไขหลัง WHERE ได้ดังนี้

WHERE sex BETWEEN '1' AND '3'

Example.

The screenshot shows a SQL query editor with the following code:

```

1 #Exp ตัวอย่างที่4 การใช้where
2
3 SELECT pname, fname, lname , sex
4 FROM person
5 WHERE sex BETWEEN '1' and '3'
6 |
7 limit 10
8 # BETWEEN แปลว่า ระหว่าง จากตัวอย่างด้านบน จะนับตั้งแต่ 1,2,3
9
10

```

The result set contains 10 records with columns pname, fname, lname, and sex.

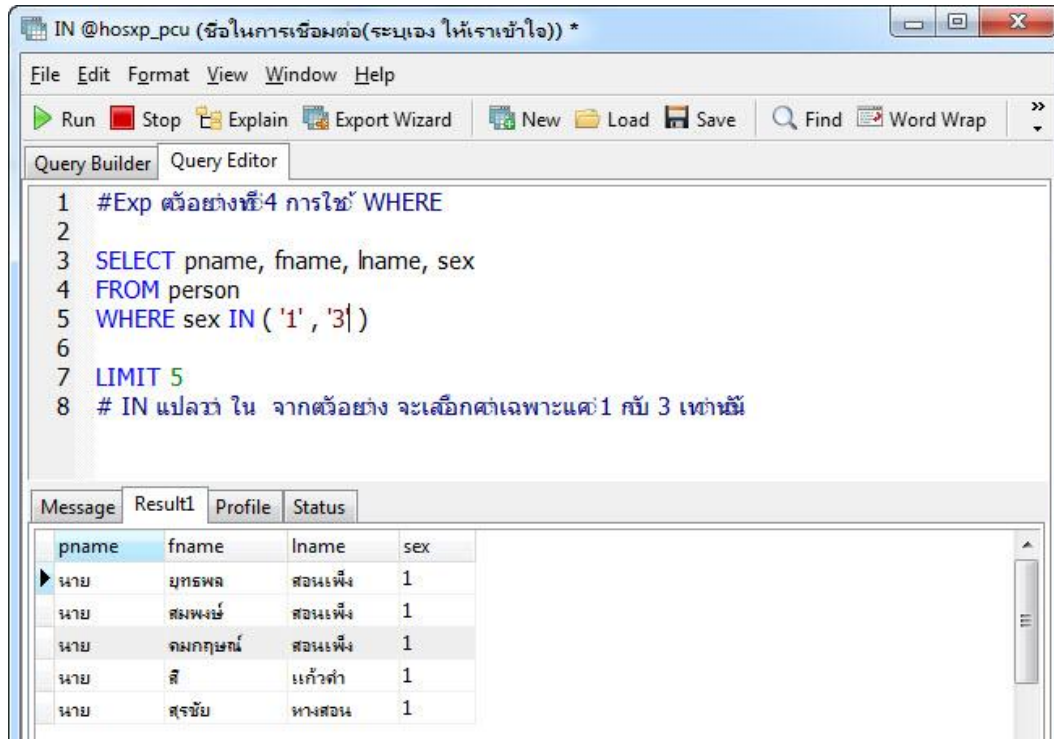
pname	fname	lname	sex
นาย	ยุทธพล	สอนแห่ง	1
นาย	สมพงษ์	สอนแห่ง	1
นาง	สายหยุด	สอนแห่ง	2
นาย	คมกฤษณ์	สอนแห่ง	1
นาง	สมพิศ	แก้วคำ	2
นาย	สี	แก้วคำ	1
นาย	สุชัย	หางสอน	1
นางสาว	พินิจตรา	แก้วคำ	2
นาย	สมบัติ	โยมั่ง	1
นาย	วิศุทธิ์	โยมั่ง	1

** ข้อควรจำ ** BETWEEN จะต้องใช้ **AND** ด้วยเสมอ เพราะเป็นโครงสร้างของ BETWEEN และจากตัวอย่างด้านบน จะถูกนับค่าที่เป็นได้ คือ 1, 2, 3

- IN แปลว่า "ใน" ซึ่งก็หมายถึง กำหนดเงื่อนไขให้มีค่าอยู่ในที่กำหนด/ ในวงเล็บที่กำหนด ซึ่งสามารถเขียนเป็นเงื่อนไขหลัง WHERE ได้ดังนี้

WHERE sex IN ('1', '3')

Example.



➤ Comparison Operators

เป็นเครื่องหมายทางคณิตศาสตร์ที่ใช้ในเชิงเปรียบเทียบ ซึ่งเรามักจะใช้เปรียบเทียบค่า ดังต่อไปนี้

- ระหว่าง Column กับ Column

เช่น person.PID = service.PID

เปรียบเทียบว่าคอลัมน์PID ของตาราง Person มีค่าเท่ากับ(=) คอลัมน์PID ของตาราง service หรือไม่

- ระหว่าง Column กับค่าที่เรากำหนด

เช่น sex = '1' # เปรียบเทียบดูว่า คอลัมน์ sex มีค่าเท่ากับ(=) 1 หรือไม่

Comparison Operators ที่ใช้บ่อยๆ มีดังนี้

"=" หมายถึง เท่ากับ

"<>" หรือ "!=" หมายถึง ไม่เท่ากับ

"<" หมายถึง น้อยกว่า

">" หมายถึง มากกว่า

"<=" หมายถึง น้อยกว่าหรือเท่ากับ

">=" หมายถึง มากกว่าหรือเท่ากับ

**** หลักสูตรเบื้องต้นนี้ Comparison Operators ให้เราจำแค่นี้ก็เพียงพอแล้วครับ ****

ตัวอย่างการใช้ Comparison Operators

1 #Exp ตัวอย่างที่ 4 การใช้ where
2
3 SELECT pname, fname, lname, sex
4 FROM person
5 WHERE sex <> '1' จากตัวอย่าง <> คือ ไม่เท่ากับ
6 limit 10
7
8 # เครื่องหมาย =, !=, <>, <, <=, >, >= ใช้ตามหลักคณิตศาสตร์ทั่วไป
9
10

pname	fname	lname	sex
นาง	สายหยุด	สอนเพ็ญ	2
นาง	สมพิศ	แก้วดำ	2
นางสาว	พันธิตรา	แก้วดำ	2
นางสาว	ศิวพร	โยียง	2
นาง	จำพรจณ	โยียง	2
นาง	ศรัญญา	มุลกร่าง	2
นางสาว	ปภาดา	แนวหลาณ	2
นางสาว	กัทธ์	ระมิตชัย	2
นางสาว	มาลัย	ยานะริน	2
นางสาว	รุ่งระวี	วงศัทา	2

➤ Logical Operators (พิเศษ)

สำหรับตัวดำเนินการที่เป็นภาษาอังกฤษ ยังมีอีก 2 ตัวที่สำคัญ แต่การใช้งานจะเป็นแบบเฉพาะตัว จึงอยากยกมาอธิบายไว้ต่างหากเป็นพิเศษ เพื่อป้องกันความสับสน คือ IS NULL และ IS NOT NULL ซึ่งเป็นตัวดำเนินการที่ไม่ได้เปรียบเทียบกับใคร แต่เปรียบเทียบกับตัวของมันเอง ดังนี้

- IS NULL หมายถึง ฟิลด์(Field)นั้นเป็นค่าว่าง
- IS NOT NULL หมายถึง ฟิลด์(Field)นั้นไม่เป็นค่าว่าง

ประโยชน์ในการใช้งาน :

- IS NULL เอาไว้เช็คค่าข้อมูลที่เรากดรอกไว้เป็นค่าว่างหรือไม่ ซึ่งก็หมายถึง ถ้ามี แสดงว่าเราไม่ได้กรอกข้อมูลตรงนี้นั่นเอง
- IS NOT NULL เอาไว้เช็คค่าข้อมูลก่อนที่เราจะนำข้อมูลไปใช้ เช่น เวลาเรานับประชากร เราควรใส่เงื่อนไขว่า เลข 13 หลัก จะต้องไม่เป็นค่าว่าง เป็นต้น ซึ่งจะทำให้ข้อมูลที่เรานำไปใช้มีความถูกต้องยิ่งขึ้น

Tip.น่ารู้ เกี่ยวกับลำดับความสำคัญของตัวดำเนินการ ของคำสั่ง SQL ดังนี้

1. ()
2. NOT
3. AND
4. ALL, ANY, BETWEEN, IN, LIKE, OR, SOME

ตัวอย่างการใช้งาน IS NULL

```

1 #Exp ตัวอย่างที่4 การใช้where
2
3 SELECT pname, fname, lname , sex
4 FROM person
5 WHERE sex is NULL
6 limit 10
7
8 # is NULL (เป็นค่าว่าง) กับ Not NULL (มีค่าไม่ว่าง)
9
10

```

pname	fname	lname	sex
(Null)	(Null)	(Null)	(Null)

ไม่มีผลลัพธ์แสดงว่า ไม่มี Row ใดๆที่ช่อง sex เป็นค่าว่างเลย

* เมื่อผลลัพธ์ไม่มีตามเงื่อนไข จึงแสดงออกมาเป็นค่าว่าง

ตัวอย่างการใช้งาน IS NOT NULL

```

1 #Exp ตัวอย่างที่4 การใช้where
2
3 SELECT pname, fname, lname , sex
4 FROM person
5 WHERE sex is NOT NULL
6 limit 10
7
8 # is NULL (เป็นค่าว่าง) กับ is Not NULL (มีค่าไม่ว่าง)
9
10

```

pname	fname	lname	sex
นาย	ยุทธพล	สอนเพ็ง	1
นาย	สมพงษ์	สอนเพ็ง	1
นาง	สายหยุด	สอนเพ็ง	2
นาย	คมกฤษณ์	สอนเพ็ง	1
นาง	สมพิศ	แก้วคำ	2
นาย	สี	แก้วคำ	1
นาย	สุรชัย	หางสอน	1
นางสาว	พันธิตรา	แก้วคำ	2
นาย	สมบัติ	โยปง	1
นาย	วิสิทธิ์	โยปง	1

ถ้ามีค่าไม่ว่าง(มีข้อมูล)ก็โชว์ตามปกติ

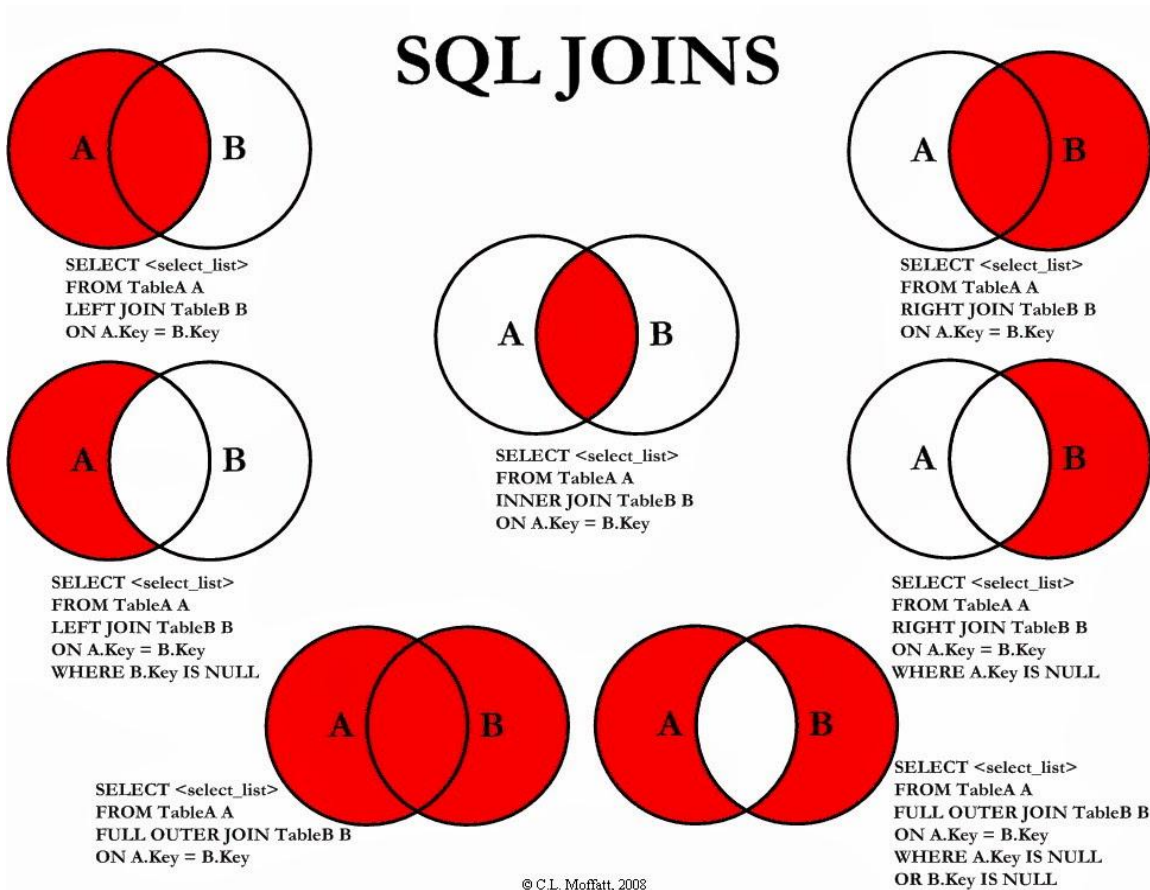
* เมื่อผลลัพธ์เป็นไปตามเงื่อนไข Record ใดที่ไม่ใช่ค่าว่าง ก็จะแสดงออกมา



การเชื่อมตารางด้วยคำสั่ง JOIN

เหตุใดเราถึงต้องเชื่อมตาราง ? ก็เพราะตารางแต่ละตารางนั้นเก็บข้อมูลเฉพาะเอาไว้ เช่น ตาราง person นั้นเก็บข้อมูลประชากร, ตาราง service เก็บข้อมูลการรับบริการ ดังนั้น ถ้าเราอยากทราบว่า นาย ก. มารับบริการอะไร, วันที่เท่าไร ซึ่งเราจะดูเพียงตารางใดตารางหนึ่งไม่ได้เลย จึงจำเป็นต้องเอาตารางทั้ง 2 มาเชื่อมกันเสียก่อน จึงจะสามารถทราบได้

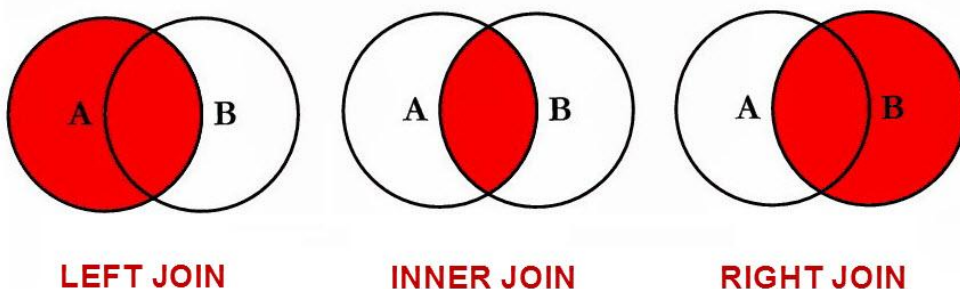
การเชื่อมตารางมีหลายรูปแบบ ดังนี้



ที่มาของภาพ <http://www.hightech-post.com/2011/07/self-join-in-sql-server.html>

เห็นแผนผังการ JOIN ด้านบนแล้วอย่าเพิ่งท้อใจไปนะครับ สำหรับมือใหม่ทุกท่าน ที่ผมนำมาใส่ไว้ในหลักสูตรการอบรมครั้งนี้ เพื่อที่จะเอาไว้อ้างอิง เผื่อว่าท่านใดอยากที่จะเรียนรู้กันไปว่าหลักสูตรนี้ก็สามารถทำได้ครับ โดยศึกษาจากแผนผังด้านบนนี้ได้เลย

แต่สำหรับหลักสูตรนี้ จะกล่าวถึงเพียง 3 แบบเท่านั้น คือ LEFT JOIN, RIGHT JOIN, INNER JOIN



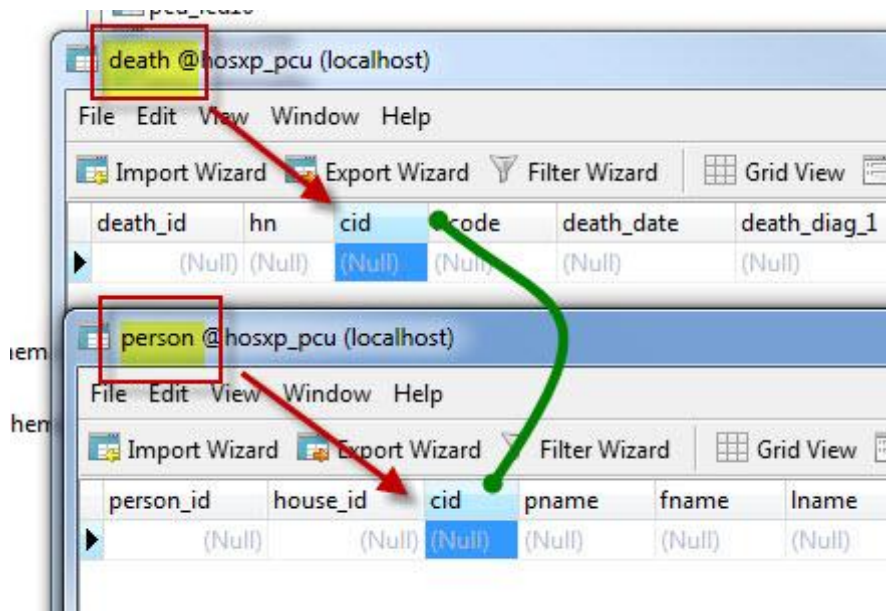
➤ เริ่มต้นการ JOIN ด้วย LEFT JOIN

การที่จะกล่าวถึงการเชื่อมตาราง หรือ JOIN แบบใดก็ตาม เรามาดูโครงสร้างการเขียนกันก่อนครับ โดยจะขอยกตัวอย่าง LEFT JOIN มาแสดงให้ดูครับ

```
SELECT ชื่อ Column หรือ *
FROM [ชื่อตาราง (ตาราง A)]
LEFT JOIN [ชื่อตาราง (ตาราง B)] ON ตารางA.KEY = ตารางB.KEY
WHERE เงื่อนไขที่ต้องการ
ORDER BY [ชื่อColumn] [รูปแบบ ASC, DESC]
LIMIT จำนวนแถวที่ต้องการแสดง
```

*อธิบายเพิ่มเติม

ตารางA.KEY หมายถึง การระบุถึง Column ชื่อ KEY ของตารางA การที่เราจะเขียนแบบนี้เพราะว่า ในชุดคำสั่ง SQL จำนวน 1ชุด ถ้ามีการอ้างอิงถึงตารางมากกว่า 1 ตาราง หากท่านดูรายละเอียดตารางต่างๆของ Hosxp PCU ท่านจะทราบว่า ตารางที่ต่างกันอาจจะมี Column ที่มีชื่อเหมือนกันก็ได้ เช่น ตาราง person จะมี Column ชื่อ cid, ตาราง death ก็มี Column ชื่อ cid เช่นกัน ดังนั้นเราจึงต้อง ระบุให้แน่ชัดว่า Column ชื่อ cid มาจากตารางไหนกันแน่ โดยเขียนว่า death.cid หรือ person.cid นั่นเอง



*ข้อควรระวัง การเชื่อมตารางด้วยชื่อ Column ที่เหมือนกันอาจจะไม่ถูกต้องเสมอไป เพราะบางครั้ง ข้อมูลอย่างเดียวกัน เช่น เลขบัตรประชาชนอาจจะอยู่ใน Column ที่ชื่อต่างกันได้ เช่น CID, pid เป็นต้น

** หลักสูตรนี้ จะพยายามหลีกเลี่ยงคำศัพท์เฉพาะ (เช่น Primary Key, Foreign Key) แต่จะเน้นให้เข้าใจ หลักการเชื่อมตาราง โดยการยกตัวอย่างให้เห็นภาพ เพื่อที่จะสามารถศึกษาต่อยอดได้ในอนาคต

ตัวอย่าง LEFT JOIN แบบง่ายๆ

ตาราง Person

person_id	house_id	cid	pname	fname	lname	pcode	sex	national
1	2	55	นาย	ยุทธพล	สอนเพ็ง	(Null)	1	99
2	2	35	นาย	สมพงษ์	สอนเพ็ง	(Null)	1	99
3	2	35	นาง	สายหยุด	สอนเพ็ง	(Null)	2	99
4	2	35	นาย	คมกฤษณ์	สอนเพ็ง	(Null)	1	99
5	3	35	นาง	สมพิศ	แก้วคำ	(Null)	2	99
6	3	35	นาย	สี	แก้วคำ	(Null)	1	99
7	3	35	นาย	สุรชัย	หางสอน	(Null)	1	99
8	3	15	นางสาว	พินิจตรา	แก้วคำ	(Null)	2	99
9	255	35	นาย	สมบัติ	โยบึง	(Null)	1	99
10	2300	35	นาย	วิสุทธิ์	โยบึง	(Null)	1	99
11	2299	35	นางสาว	สิวพร	โยบึง	(Null)	2	99
13	2299	354000026984	นาง	จำพรณ	โยบึง	(Null)	2	99

ตาราง sex

ตาราง sex ใน Hosxp PCU

code	name	hos_guid
1	ชาย	{681962F6-243E-11E0-B4D1-}
2	หญิง	{68196505-243E-11E0-B4D1-}

*จุดสังเกต ท่านเห็นอะไรใน 2 ตารางนี้ ระหว่างตาราง person กับ ตาราง sex ?

ใช่แล้วครับ !! ในตาราง person จะมี Column ชื่อว่า sex และในตาราง sex จะมี Column ชื่อว่า code เราสามารถนำทั้ง 2 ตารางนี้มาเชื่อมกันได้ครับ มาดูตัวอย่างกันเลย...

ตัวอย่างการ LEFT JOIN เพื่อดึง Column ชื่อ name ของตาราง sex ออกมา

Query Builder Query Editor

```

1 select pname, fname, lname, s.name as เพศ
2 from person p
3 left join sex s on p.sex = s.code
4 limit 10
5 # person p และ sex s เป็นการ สร้างตัวชื่อ ของชื่อตาราง เพื่อสะดวกในการเขียน
6 # กรณีนี้เราจะนำตาราง เช่น s.name หมายถึง field ชื่อ name ของตาราง sex
        
```

Message Result1 Profile Status

pname	fname	lname	เพศ
นาย	ยุทธพล	สอนเพ็ง	ชาย
นาย	สมพงษ์	สอนเพ็ง	ชาย
นาง	สายหยุด	สอนเพ็ง	หญิง
นาย	คมกฤษณ์	สอนเพ็ง	ชาย
นาง	สมพิศ	แก้วคำ	หญิง
นาย	สี	แก้วคำ	ชาย
นาย	สุรชัย	หางสอน	ชาย
นางสาว	พินิจตรา	แก้วคำ	หญิง
นาย	สมบัติ	โยบึง	ชาย
นาย	วิสุทธิ์	โยบึง	ชาย

การเชื่อมตารางหรือการ
Join, left Join

เก็บตกรายละเอียดการเขียน SQL เพิ่มเติม

Tip. การย่อชื่อตาราง

```

1 select pname, fname, lname, s.name as เพศ
2 from person p
3 left join sex s on p.sex = s.code
4 limit 10
5
6

```

ใช้ตัวย่อและการอ้างชื่อด้วยตัวย่อ

การเชื่อมต่อหลายๆตาราง อาจจะทำให้เราต้องอ้างอิงทั้งชื่อตารางและชื่อคอลัมน์ ทำให้พิมพ์เยอะ เสียเวลามากทีเดียว ดังนั้น SQL จึงให้เราสามารถย่อชื่อของตารางได้ด้วยครับ ด้วยการพิมพ์อักษรย่อหลังชื่อตาราง โดยเว้นระยะห่างอย่างน้อย 1 ตัวอักษร เช่น person p ซึ่งตัว p จะเป็นตัวย่อของตาราง person ดังนั้นการอ้างอิงถึง Column ในตาราง person ก็ไม่จำเป็นต้องเขียน person.sex อีกแล้ว แต่ให้เราเขียน p.sex แทน

***จุดที่ควรระวัง** ถ้าเราได้กำหนดไว้ว่า person p เรียบร้อยแล้ว เราจะกลับมาเขียนว่า person.sex ไม่ได้แล้วนะครับ เราจะต้องเขียน p.sex เท่านั้นครับ

Tip. การเปลี่ยนหัวชื่อคอลัมน์ตามใจเรา

```

1 select pname, fname, lname, s.name as เพศ
2 from person p
3 left join sex s on p.sex = s.code
4 limit 10
5
6

```

pname	fname	lname	เพศ
นาย	ยุทธพล	สอนเพ็ง	ชาย
นาย	สมพงษ์	สอนเพ็ง	ชาย

เราสามารถเปลี่ยนหัว Column ได้นะครับ ด้วยคำสั่ง **as** ดังภาพ ถ้าเราไม่ได้ใช้คำสั่ง as คำว่าเพศที่ด้านบนของคอลัมน์ จะเป็นคำว่า s.name แทน แต่ถ้าเราใช้ as แล้วพิมพ์ชื่อที่เราต้องการลงไป คำๆนั้นจะกลายเป็นหัวของคอลัมน์นั้นทันทีครับ

****หลักสูตรนี้จะไม่ขอลงลึกในรายละเอียดของ JOIN หรือ LEFT JOIN และการ JOIN แบบอื่นๆ เพราะมีเนื้อหาใหม่ๆท่านอาจจะงง หรือต้องการเวลาทบทวนอีกสักกระยะหนึ่ง แต่เอกสารนี้ก็เพื่อให้แนวทางในการพัฒนาต่อยอดด้วยรูปแบบการ JOIN แบบต่างๆ เหลือเพียงการฝึกฝนและหาความรู้เพิ่มเติมก็จะสามารถพัฒนาต่อได้อย่างแน่นอน**



การแสดงผลแบบจัดกลุ่มด้วย GROUP BY

หลายๆท่านอาจจะสงสัยว่าทำไมผมถึงเอาคำสั่ง GROUP BY หรือคำสั่งจัดกลุ่มมานำเสนอในช่วงท้ายๆ ทั้งๆที่การใช้งานฟังดูแล้วน่าจะทำได้ง่ายๆ แต่ผมขอบอกเลยว่า GROUP BY ไม่ง่ายอย่างที่คิดครับ เพราะการใช้งาน GROUP BY ให้ได้ประสิทธิภาพนั้น ต้องใช้ร่วมกับฟังก์ชันอื่นๆด้วย (ใช้เดี่ยวๆไม่ Work ครับ) ซึ่งในหลักสูตรนี้ขอแนะนำ ฟังก์ชันที่ใช้ร่วมกับ GROUP BY บ่อยๆ 2 ตัวครับ คือ SUM และ COUNT

GROUP BY = คำสั่งการจัดกลุ่ม

SUM = เป็นฟังก์ชันผลรวม รูปแบบที่ใช้คือ SUM(ชื่อ Column)

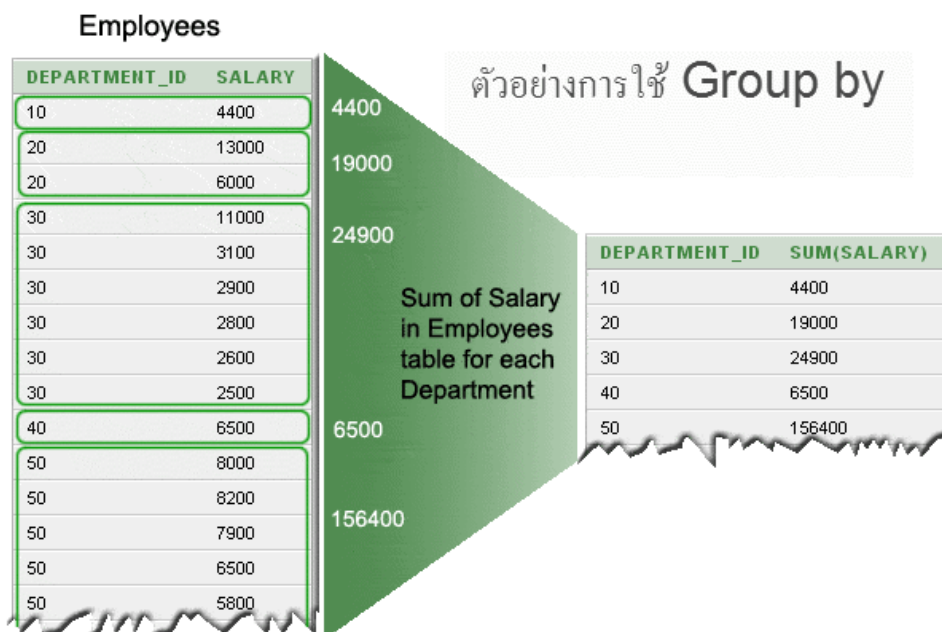
COUNT = เป็นฟังก์ชันนับ รูปแบบที่ใช้คือ COUNT(ชื่อ Column)

รูปแบบการใช้งาน

*การใช้ GROUP BY อาจจะมีรูปแบบที่ต่างกับ รูปแบบที่ผ่านมา ลองสังเกตดูครับ

```
SELECT [ชื่อ Column A] , SUM(Column B) หรือ COUNT(Column B)
FROM [ชื่อตาราง]
WHERE เงื่อนไขที่ต้องการ
GROUP BY [ชื่อColumn A]
```

*ในหลักสูตรนี้จะสอนใช้ฟังก์ชัน SUM และ COUNT เท่านั้น ซึ่งในความเป็นจริงยังมีฟังก์ชันอีกจำนวนมาก
มองภาพ GROUP BY ให้ชัด ด้วยตัวอย่างที่ใช้ SUM



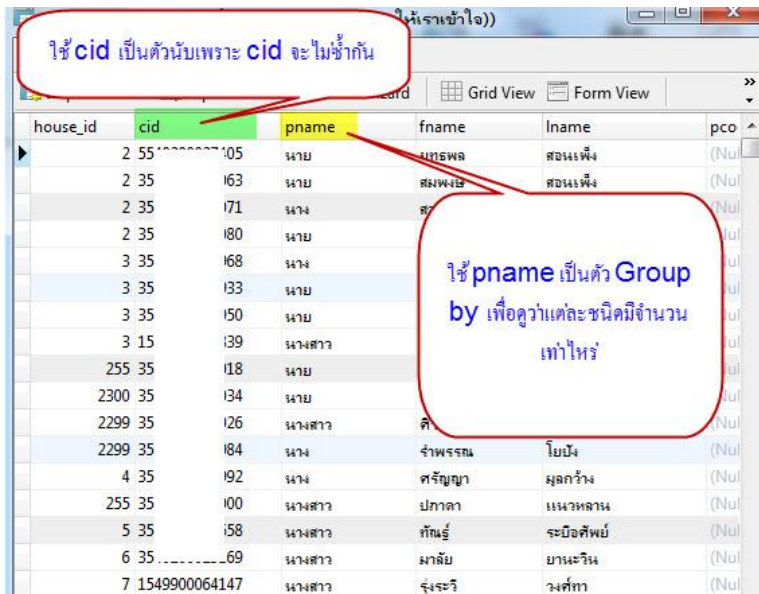
ที่มาของภาพ <http://www.w3resource.com/sql/group-by.php>

จากภาพ เราจะเห็นว่าตารางชื่อ Employees มี DEPARTMENT_ID ซ้ำๆกัน แต่ SALARY(เงินเดือน) ไม่ซ้ำกัน เราจึงสามารถจัดกลุ่มตาม DEPARTMENT_ID โดยเราจะทราบว่าแต่ละ DEPARTMENT_ID มีผลรวมของ SALARY เท่าไหร่ ด้วยฟังก์ชัน SUM

จากตาราง Employees เราสามารถเขียนคำสั่ง ได้ดังนี้

```
SELECT DEPARTMENT_ID, SUM(SALARY)
FROM Employees
GROUP BY DEPARTMENT_ID
```

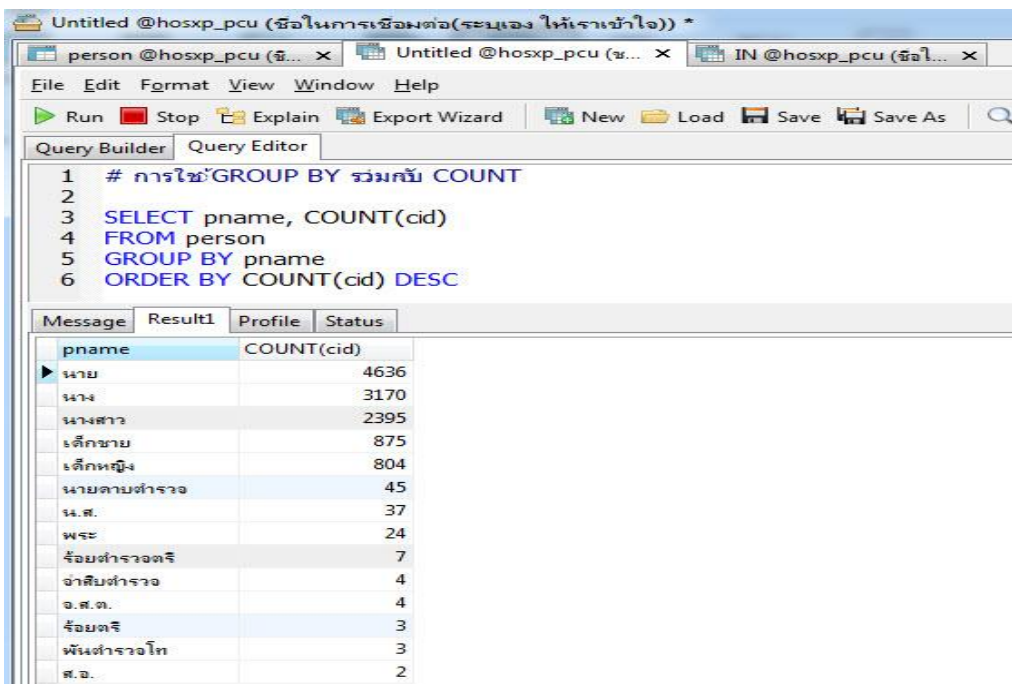
เราทดลองใช้ GROUP BY ร่วมกับฟังก์ชัน SUM กันไปแล้ว ต่อไปลองใช้ร่วมกับฟังก์ชัน COUNT กันดูบ้างครับ โดยตัวอย่างนี้ จะขอใช้ตาราง person ในฐานข้อมูล Hosxp PCU มาดูตัวอย่างกันเลยครับ



เมื่อเราเปิดเข้าไปดูรายละเอียดของตาราง person ก็พบว่า Column ชื่อ pname ที่เก็บข้อมูลค่านำหน้าของชื่อ มีการเก็บข้อมูลซ้ำๆกัน เช่น นาย, นาง, นางสาว เป็นต้น

ซึ่งเราอยากจะทราบว่าแต่ละค่านำหน้าที่ถูกใช้ มีจำนวนเท่าไรบ้าง เราสามารถใช้ GROUP BY ร่วมกับ COUNT ได้ครับ

*หมายเหตุ Column ที่จะนำมานับ ควรเป็นข้อมูลที่ไม่ซ้ำกันเลย เช่น เลขบัตรประชาชน เป็นต้น



เอกสารและแหล่งอ้างอิง

<https://th.wikipedia.org/wiki/มายเอสคิวแอล>

<http://www.hightech-post.com/2011/07/self-join-in-sql-server.html>

<http://www.w3resource.com/sql/group-by.php>

<http://www.w3resource.com/w3skills/sql-beginner/>

<http://www.tutorialspoint.com/sql/sql-operators.htm>

<http://www.w3schools.com/sql/>

http://code.function.in.th/sql/operator/priority_of_sql_operator



แบบทดสอบระหว่างเรียน

แบบทดสอบที่ 1 คำสั่ง SQL เบื้องต้น

- 1.1 จงแสดงข้อมูลทุก Column ในตาราง person
- 1.2 จงแสดงข้อมูลเฉพาะ Column ชื่อ, นามสกุล, เพศ ในตาราง person

แบบทดสอบที่ 2 การปรับการแสดงผลตามต้องการด้วย LIMIT, ORDER BY

- 2.1 จงแสดงข้อมูลทุก Column ในตาราง person โดยจำกัดเพียง 10 Record เท่านั้น
- 2.2 จงแสดงข้อมูลเฉพาะ Column house_id, ชื่อ, นามสกุล, เพศ ในตาราง person โดยเรียงลำดับจาก house_id จากน้อยไปหามาก
- 2.3 จงแสดงข้อมูลทุก Column ในตาราง person โดยจำกัดเพียง 10 Record เท่านั้น โดยเรียงลำดับจาก house_id จากมากไปหาน้อย

แบบทดสอบที่ 3 การแสดงข้อมูลตามเงื่อนไขที่กำหนดด้วย WHERE

- 3.1 จงแสดงข้อมูลทุก Column ในตาราง person โดยมีเงื่อนไขว่าจะต้องเป็นคนที่มีสถานะสมรสเป็น "ม่าย" (marry status = 3) โดยจำกัดเพียง 10 Record เท่านั้น
- 3.2 จงแสดงข้อมูลเฉพาะ Column person_id, ชื่อ, นามสกุล, เพศ, education ในตาราง person โดยมีเงื่อนไขว่าต้องมีสัญชาติเป็นคนไทย(nationality=99) และ มีระดับการศึกษา (education=9) โดยเรียงลำดับจาก person_id จากน้อยไปหามาก
- 3.3 จงแสดงข้อมูลเฉพาะ Column person_id, ชื่อ, นามสกุล, เพศ, education ในตาราง person โดยมีเงื่อนไขว่าต้องมีสัญชาติเป็นคนไทย(nationality=99) หรือ มีระดับการศึกษา (education=9)
- 3.4 จงแสดงข้อมูลเฉพาะ Column ชื่อ, นามสกุล, เพศ, age_y ในตาราง person โดยมีเงื่อนไขว่าต้องมีอายุ(ปี)อยู่ระหว่าง 20 และ 30 ปี
- 3.5 จงแสดงข้อมูลเฉพาะ Column ชื่อ, นามสกุล, เพศ, age_y ในตาราง person โดยมีเงื่อนไขว่าต้องมีอายุ(ปี)มากกว่าหรือเท่ากับ 60 ปี
- 3.6 จงแสดงข้อมูลเฉพาะ Column ชื่อ, นามสกุล, เพศ, age_y ในตาราง person โดยมีเงื่อนไขว่าต้องมีอายุ(ปี)อยู่ระหว่าง 0 และ 12 ปี และไม่เป็นเพศหญิง(sex=2) โดยเรียงลำดับจาก age_y จากมากไปหาน้อยและจำกัดเพียง 10 Record เท่านั้น
- 3.7 จงแสดงข้อมูลเฉพาะ Column ชื่อ, นามสกุล, เพศ, age_y, last_screen_dmht_bdg_year ในตาราง person โดยมีเงื่อนไขว่าต้องมีอายุ(ปี)มากกว่าหรือเท่ากับ 35 ปี และปีล่าสุดที่ได้รับการคัดกรองเบาหวาน-ความดัน(last_screen_dmht_bdg_year) ไม่เป็นค่าว่าง
- 3.8 จงแสดงข้อมูลเฉพาะ Column ชื่อ, นามสกุล, เพศ, father_name, father_cid ในตาราง person โดยมีเงื่อนไขว่าเลขบัตรประชาชนของบิดา(father_cid) เป็นค่าว่าง

แบบทดสอบที่ 4 การแสดงข้อมูลตามเงื่อนไขที่กำหนดด้วย WHERE และการเชื่อมตาราง JOIN

4.1 เมื่อเราดูข้อมูลในตาราง person เราพบว่า มี Column ชื่อ occupation ซึ่งได้เก็บรหัสอาชีพ 3 หลักเอาไว้ แต่เราอยากทราบว่ารหัสเหล่านั้น หมายถึงอาชีพอะไรบ้าง ?? ในเมื่อเราทราบว่า ตาราง occupation นั้น ได้เก็บรายละเอียดของอาชีพต่างๆเอาไว้ !!

- จงแสดงข้อมูลเฉพาะ Column ชื่อ, นามสกุล, เพศ, อาชีพ(ที่ไม่ใช่รหัส แต่เป็นชื่ออาชีพ) โดยให้แสดงจำนวน 20 Record เท่านั้น

4.2 เมื่อเราดูข้อมูลในตาราง person เราพบว่า มี Column ชื่อ education ซึ่งได้เก็บรหัสการศึกษา 1 หลักเอาไว้ แต่เราอยากทราบว่ารหัสเหล่านั้น หมายถึงอะไร ?? ในเมื่อเราทราบว่า ตาราง education นั้น ได้เก็บรายละเอียดของระดับการศึกษาต่างๆเอาไว้ !!

- จงแสดงข้อมูลเฉพาะ Column ชื่อ, นามสกุล, เพศ, ระดับการศึกษา(ที่ไม่ใช่รหัส แต่เป็นชื่อระดับการศึกษา) โดยให้แสดงจำนวน 20 Record เท่านั้น และยังให้เรียงลำดับจากน้อยไปหามากด้วย Column ชื่อ education

แบบทดสอบที่ 5 การใช้คำสั่ง GROUP BY ร่วมกับฟังก์ชัน SUM และ COUNT

5.1 เมื่อเราดูข้อมูลในตาราง ovst ซึ่งเป็นตารางที่เก็บข้อมูลผู้ที่มารับบริการกับเรา พบว่ามี Column ชื่อ ptype ซึ่งเก็บข้อมูลประเภทของบัตรทอง เช่น 71,77 เป็นต้น เราอยากทราบว่า ตั้งแต่วันที่ 1 มกราคม 2558 (vstdate='2015-01-01') ถึง 31 มกราคม 2558(vstdate='2015-01-31') มีผู้ใช้สิทธิ์ประเภทต่างๆ เป็นจำนวนเท่าไร โดยแยกตามสิทธิ์ แต่เราจะไม่นับ Record ที่ Column ptype เป็นค่าว่าง

- จากข้อมูลด้านบนจงแสดงข้อมูลเฉพาะ ptype และ จำนวนผู้ที่มารับบริการ ตามเงื่อนไขดังที่กล่าวไปแล้วข้างต้น

5.2 เมื่อเราดูข้อมูลในตาราง vn_stat ซึ่งเป็นตารางที่เก็บข้อมูลการวินิจฉัยโรคของผู้ป่วยที่มารับบริการกับเรา พบว่ามี Column ชื่อ pdx ซึ่งเก็บข้อมูลรหัสการวินิจฉัยโรค เช่น z34, j00 เป็นต้น เราอยากทราบว่า ตั้งแต่วันที่ 1 มกราคม 2558 (vstdate='2015-01-01') ถึง 31 มกราคม 2558(vstdate='2015-01-31') มีการวินิจฉัยโรคอะไรบ้าง โดยแต่ละโรคมีค่าใช้จ่ายรวมแล้วเท่าไร(income) แต่เราจะไม่นับ Record ที่ pdx เป็นค่าว่าง

- จากข้อมูลด้านบนจงแสดงข้อมูลเฉพาะ pdx และ จำนวนค่าใช้จ่ายรวมของแต่ละวินิจฉัย ตามเงื่อนไขดังที่กล่าวไปแล้วข้างต้น