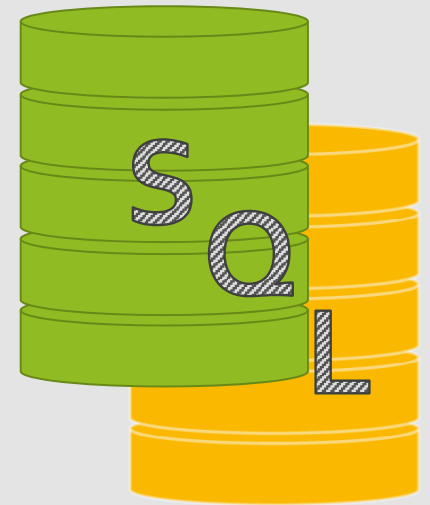


SQL

Structured Query Language



ตารางการอบรม



วันที่ และ เวลา	หัวข้อ
วันที่ 1 9.00-16.00	SQL คืออะไร ประโยชน์ของ SQL การออกแบบฐานข้อมูล ออกแบบตารางข้อมูลและความสัมพันธ์ ติดตั้งและทดลองใช้ Dbeaver
วันที่ 2 9.00-16.00	-คำสั่งพื้นฐานของ SQL -คำสั่ง CREATE DATABASE -คำสั่ง CREATE TABLE -คำสั่ง INSERT DELETE UPDATE -ฝึกปฏิบัติ
วันที่ 3 9.00-16.00	-คำสั่ง SELECT -คำสั่ง JOIN TABLE -คำสั่ง CREATE VIEW -ฝึกปฏิบัติ

SQL หรือ Structured Query Language



- SQL ย่อมาจาก structured query language คือภาษาที่ใช้ในการเขียนโปรแกรม เพื่อจัดการกับฐานข้อมูลโดยเฉพาะ เป็นภาษามาตรฐานบนระบบฐานข้อมูลเชิงสัมพันธ์(Relational Database) และเป็นระบบเปิด (open system) หมายถึงเราสามารถใช้คำสั่ง sql กับฐานข้อมูลชนิดใดก็ได้ และ คำสั่งงานเดียวกันเมื่อสั่งงานผ่าน ระบบฐานข้อมูลที่แตกต่างกันจะได้ผลลัพธ์เหมือนกัน ทำให้เราสามารถเลือกใช้ฐานข้อมูล ชนิดใดก็ได้โดยไม่ติดขัดกับฐานข้อมูลใดฐานข้อมูลหนึ่ง
- Relational Database คือ ระบบฐานข้อมูลเชิงสัมพันธ์ เป็นการเก็บข้อมูลในรูปแบบของตาราง (table) ในแต่ละตารางแบ่งออกเป็นแถวๆ และในแต่ละแถวจะแบ่งเป็นคอลัมน์ (Column) ซึ่งในการเชื่อมโยงกันระหว่างข้อมูลในตารางต่างๆ จะ เชื่อมโยงโดยใช้การอ้างอิงจากข้อมูลในคอลัมน์ที่กำหนดไว้

ประโยชน์ของ ภาษา SQL



ประโยชน์ของภาษา SQL

1. สร้างฐานข้อมูลและตาราง
2. สนับสนุนการจัดการฐานข้อมูล ซึ่งประกอบด้วย การเพิ่ม การปรับปรุง และการลบข้อมูล
3. สนับสนุนการเรียกใช้หรือ ค้นหาข้อมูล

ประเภทของข้อมูล (Data types) ใน SQL



SQL data types เป็นการกำหนดชนิดของข้อมูลในตารางว่าเป็นข้อมูลแบบใด เช่น

- ตัวเลข
- ตัวอักษร
- วันเวลา
- แบบไม่มีโครงสร้าง
- ซึ่งสิ่งเหล่านี้จำเป็นตั้งแต่เราเริ่มสร้าง database table เพื่อให้ข้อมูลที่เราจะใส่ลงสู่ table มีความถูกต้องตามที่วางเอาไว้ อีกทั้งยังช่วยให้ฐานข้อมูลหรือ database ของเราทำงานได้ง่ายขึ้นในการจัดเก็บ และการทำดัชนี (index) ได้เหมาะสมกับข้อมูลที่เราจะใช้งาน โดย data types บน database มีด้วยกันหลายชนิด ขึ้นอยู่กับชนิดของฐานข้อมูล หรือ database ที่เราใช้งาน

RDBMS (Relational Database Management System)



RDBMS (Relational Database Management System) คือ ระบบจัดการฐานข้อมูลเชิงสัมพันธ์ทำให้ผู้ใช้สามารถจัดการข้อมูลได้ง่ายขึ้น

RDBMS สามารถใช้คำสั่ง SQL พื้นฐานได้แต่จะมีบางคำสั่งที่เฉพาะกับ RDBMS รุ่นนั้นๆ ในปัจจุบันมีหลายค่ายได้แก่

- MS Access
- SQL Server
- MySQL
- SQLite
- PostGresSQL

SQLite



<https://www.sqlite.org/index.html>



Home About Documentation Download License Support Purchase

What Is SQLite?

SQLite is a C-language library that implements a [small](#), [fast](#), [self-contained](#), [high-reliability](#), [full-featured](#), SQL database engine. SQLite is the [most used](#) database engine in the world. SQLite is built into all mobile phones and most computers and comes bundled inside countless other applications that people use every day. [More Information...](#)

The SQLite [file format](#) is stable, cross-platform, and backwards compatible and the developers pledge to keep it that way through at least the year 2050. SQLite database files are commonly used as containers to transfer rich content between systems [\[1\]](#) [\[2\]](#) [\[3\]](#) and as a long-term archival format for data [\[4\]](#). There are over 1 trillion (1e12) SQLite databases in active use [\[5\]](#).

SQLite [source code](#) is in the [public-domain](#) and is free to everyone to use for any purpose.

Latest Release

[Version 3.32.3](#) (2020-06-18). [Download](#) [Prior Releases](#)

ฐานข้อมูล



คำศัพท์ที่เกี่ยวข้องกับฐานข้อมูล

- **เขตข้อมูล/ฟิลด์ (Field) หรือ คอลัมน์** คือ หน่วยของข้อมูลที่เกิดจากการนำตัวอักษรหลาย ๆ ตัวมารวมกัน เป็นคำที่มีความหมาย เช่น ฟิลด์ชื่อสินค้า ฟิลด์ราคาสินค้า ฟิลด์จำนวนสินค้า
- **ระเบียน/เรคอร์ด (Record) หรือ แถว** คือ การนำเขตข้อมูล หลายๆ เขตข้อมูล มาเรียงต่อกัน เรียกว่า ระเบียน (record) เช่น ระเบียนหรือแถวที่ 1 เก็บ ชื่อสินค้า ราคา จำนวนสินค้า ของ สินค้ารายการที่ 1
- **ตาราง (Table)** คือการนำเรคอร์ดหลาย ๆ เรคอร์ดมารวมกัน

Tables in SQL



ชื่อตาราง

ชื่อฟิลด์

Employee Enter a SQL expression to filter results (use Ctrl+Space)

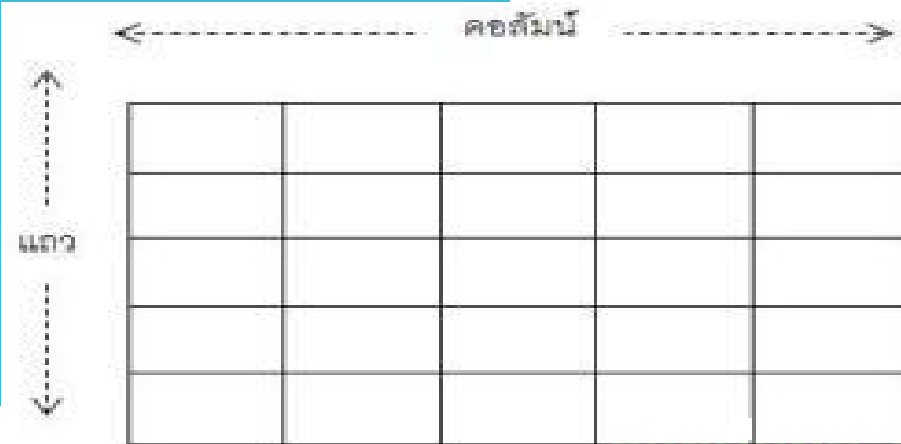
Grid	EmployeeId	LastName	FirstName	Title
1	1	Adams	Andrew	General Manager
2	2	Edwards	Nancy	Sales Manager
3	3	Peacock	Jane	Sales Support Agent
4	4	Park	Margaret	Sales Support Agent
5	5	Johnson	Steve	Sales Support Agent
6	6	Mitchell	Michael	IT Manager
7	7	King	Robert	IT Staff
8	8	Callahan	Laura	IT Staff

แถว หรือ records

RDBMS

- ความสัมพันธ์ (Relationships) ของระบบฐานข้อมูล

เป็นความสัมพันธ์ของ "ข้อมูลเชิงสัมพันธ์" ซึ่งข้อมูลเชิงสัมพันธ์นี้เกิดขึ้นจากตาราง 2 มิติ คือการประกอบกันของตารางด้วยแถว และคอลัมน์ (ดังรูปที่ 1 ตาราง 2 มิติ) เพื่อทำการเก็บข้อมูลต่างๆ โดยการออกแบบฐานข้อมูลจำเป็นต้องมีการกำหนดค่าความสัมพันธ์นั้นด้วย



Field

EmployeeId	LastName	FirstName	Title
1	Adams	Andrew	General Manager
2	Edwards	Nancy	Sales Manager
3	Peacock	Jane	Sales Support Agent
4	Park	Margaret	Sales Support Agent
5	Johnson	Steve	Sales Support Agent
6	Mitchell	Michael	IT Manager
7	King	Robert	IT Staff
8	Callahan	Laura	IT Staff

Record

Table

E-R Diagram



- E-R Diagram หรือ Entity Relationship Diagram คือ แผนภาพแสดงความสัมพันธ์ระหว่างข้อมูล ประกอบด้วย Entity และ Relationship
- เอนทิตี (Entity) คือวัตถุหรือสิ่งของหรือคนที่เราสนใจในระบบงานนั้นๆ เช่น เอนทิตีพนักงาน เอนทิตีลูกค้า
- แอททริบิว (Attribute) เป็นคุณสมบัติของวัตถุที่เราสนใจ เช่น เอนทิตีพนักงาน มี รหัสพนักงาน ชื่อ นามสกุล ตำแหน่ง
- ความสัมพันธ์ (Relationship) คือ ความสัมพันธ์ระหว่างเอนทิตี

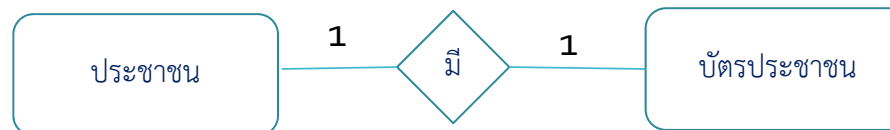
ความสัมพันธ์



- ชนิดความสัมพันธ์ของตาราง (Relationship)

1. ความสัมพันธ์แบบหนึ่งต่อหนึ่ง (one to one Relationships) เป็นความสัมพันธ์ของตารางหนึ่งกับอีกตารางหนึ่งเท่านั้นเช่น

อธิบายความสัมพันธ์ คือ ประชาชน 1 คน มีบัตรประชาชน 1 ใบ ถ้าอ่านสลับกันก็คือ บัตรประชาชน 1 ใบ สามารถมีประชาชนได้ 1 คน



ความสัมพันธ์แบบ One-to-One



- เมื่ออยู่ในตาราง (เมื่อหนึ่งแถวในหนึ่งตารางมีความสัมพันธ์กับหนึ่งแถวในอีกหนึ่งตาราง)



ประชากร

รหัสประชากร	ชื่อ	อายุ	เพศ	...
P1001	บันทกร บางเซน	45	ชาย	...
P1002	สมชาย สุขศรี	30	ชาย	...
P1003	ศิริกาญจน์ ดิศริ	22	หญิง	...
P1004	บุญมล พงษ์ศรี	54	หญิง	...

ใบขับขี่

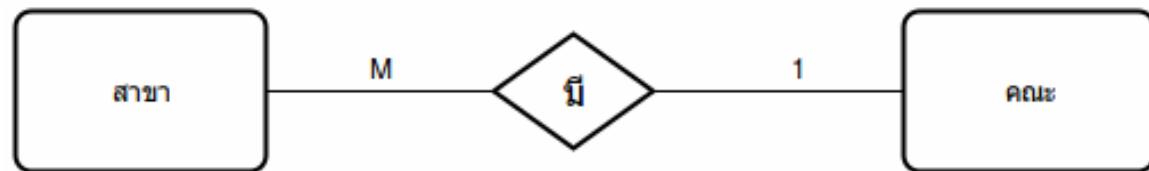
รหัสประชากร	วันที่ออกบัตร	วันที่หมดอายุ	เลขที่ใบขับขี่
1001	01/12/2547	01/12/2557	34246734
1002	15/01/2550	01/01/2560	39480037
1003	06/04/2550	06/04/2560	49002778
1004	02/06/2552	02/06/2562	73827234

mindphp.com

ความสัมพันธ์
แบบหนึ่งต่อกลุ่ม
(one to many
Relationships)



- เป็นความสัมพันธ์ของตารางหนึ่ง ที่มีข้อมูลอยู่ได้หลายตาราง แต่อีกตารางหนึ่งสามารถมีข้อมูลได้แค่ตารางเดียว ความสัมพันธ์แบบนี้เกิดขึ้นมากที่สุดในการสร้างฐานข้อมูล เป็นความสัมพันธ์ระหว่างหนึ่งสิ่งต่อหลาย ๆ สิ่ง เช่น



- อธิบาย สาขาหนึ่งสาขา สังกัดคณะได้เพียงหนึ่งคณะ อ่านกลับกันจะ
ได้ คณะหนึ่งคณะจะมีสาขาสังกัดได้หลายสาขา

ความสัมพันธ์แบบ One-to-Many



- ความสัมพันธ์แบบ One-to-Many เมื่ออยู่ในตาราง (เมื่อหนึ่งแถวในตารางสามารถสัมพันธ์กับหนึ่งหรือหลายแถวในอีกตารางหนึ่ง)



ทีมฟุตบอล

รหัสทีมฟุตบอล	ชื่อทีมฟุตบอล	ชื่อผู้จัดการทีม	...
FTH001	Bangkok Glass FC	Attaphol Buspakom	...
FTH002	TOT S.C.	Somchai Subpherm	...
FTH003	Muangthong United F.C.	René Desaeyere	...

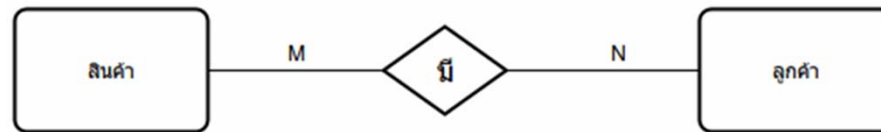
นักฟุตบอล

รหัสนักฟุตบอล	รหัสทีมฟุตบอล	รหัสนักฟุตบอล	...
1001	FTH001	Ekaphan Inthasen	...
1002	FTH002	Punnarat Klinsukon	...
1003	FTH003	Chainarong Tathong	...
1004	FTH001	Kritsana Klanklin	...

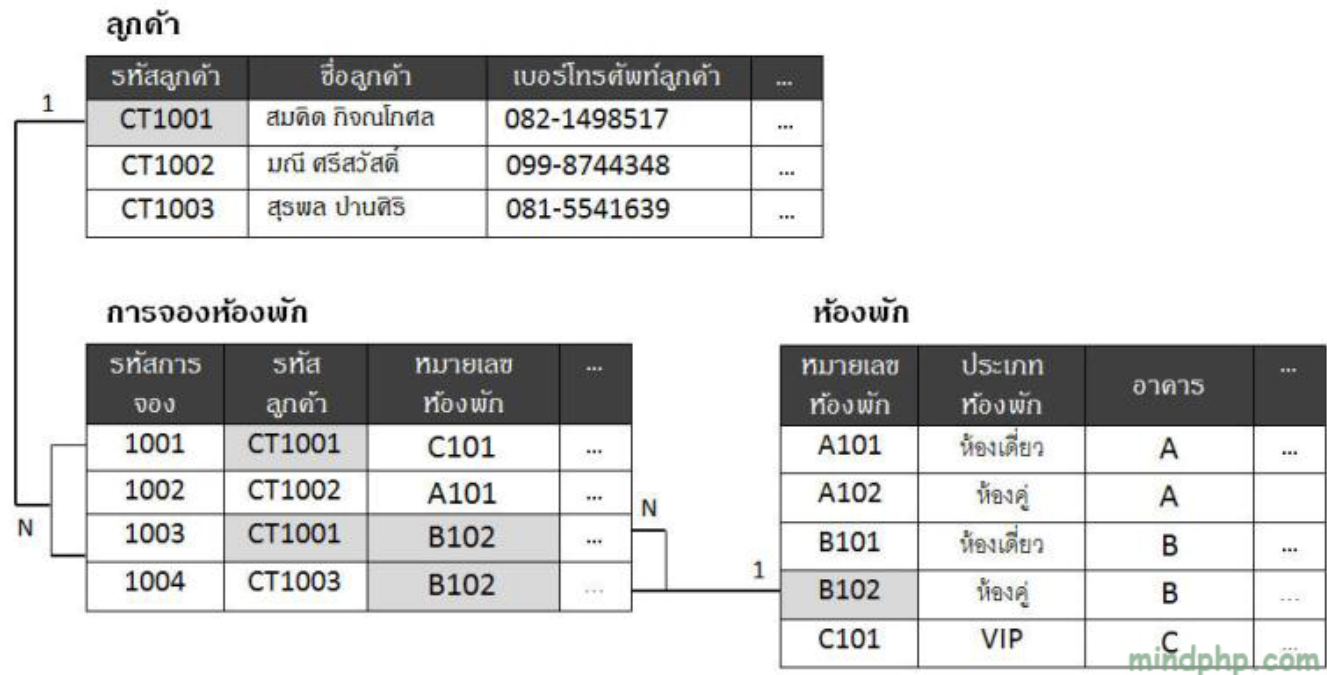
ความสัมพันธ์
แบบกลุ่มต่อกลุ่ม
(Many to Many)



- เป็นความสัมพันธ์ของตารางหนึ่งที่มีข้อมูลได้หลายตาราง และอีกตารางหนึ่งที่มีข้อมูลได้หลายตารางเช่นเดียวกันเช่น
- อธิบาย สินค้าหนึ่งชิ้น สามารถถูกซื้อได้โดยคนหลายคน อ่านกลับกันจะได้ คนหนึ่งคนสามารถซื้อสินค้าได้หลายชิ้น



ตัวอย่าง ความสัมพันธ์
แบบกลุ่มต่อกลุ่ม
(Many to Many)



ประเภทของคีย์ในระบบฐานข้อมูล



- การกำหนดคีย์ในระบบฐานข้อมูลนั้นเป็นการช่วยเพิ่มประสิทธิภาพในการทำงานของฐานข้อมูล เพราะเป็นการกำหนดการอ้างอิงของข้อมูล การค้นหาจึงทำให้การประมวลผลของระบบมีความรวดเร็วยิ่ง ซึ่งคีย์ในระบบฐานข้อมูลสามารถแบ่งออกได้ 5 ชนิดดังนี้
 1. Primary Key (คีย์หลัก)
 2. Secondary Key (คีย์รอง)
 3. Compound Key (คีย์รวม)
 4. Candidate Key (คีย์คู่แข่ง)
 5. Foreign Key (คีย์นอก)

Primary Key (คีย์หลัก)



- ข้อมูลคอลัมน์หนึ่งในตารางที่ไม่มีค่าซ้ำกับข้อมูลคอลัมน์อื่น และต้องไม่เป็นค่าว่าง (Null)

นักศึกษา

รหัสประจำตัวนักศึกษา	ชื่อ	นามสกุล	เพศ	อายุ	...
1501000011	สมคิด	ดำเนนศิริ	ชาย	21	...
1501000012	วัลภา	ตั้งประเสริฐ	หญิง	21	...
1501000013	ศิริโกศล	มีสุข	หญิง	20	...
1501000014	ดำรง	ดิติดี	ชาย	21	...

↑
"Primary Key"

- จากรูปจะเห็นว่าคอลัมน์ที่เป็น Primary Key คือ รหัสประจำตัวนักศึกษา เพราะไม่มีค่าซ้ำกับคอลัมน์อื่น และไม่เป็นค่าว่าง
#อาจมีข้อสงสัยว่าทำไมเราไม่ใช่ ชื่อ หรือ นามสกุล ในการกำหนด Primary Key เพราะชื่อและนามสกุลของคนเราอาจมีเหมือนกันได้ ดังนั้น หากนักศึกษามีชื่อซ้ำกันมากก็จะไม่สามารถระบุการอ้างอิงได้

Secondary Key (คีย์รอง)



- ในบางครั้งเรียกคีย์ชนิดนี้ว่า อินเด็กซ์ (Index) คีย์ชนิดนี้เปรียบเสมือนเป็นคีย์รองจากคีย์หลัก กล่าวคือเมื่อเรากำหนดคีย์หลักแล้ว DBMS ก็จะสามารถค้นหาข้อมูล แต่เมื่อไหร่ที่มีข้อมูลเป็นจำนวนมาก DBMS ก็จะต้องทำการค้นหาตั้งแต่ต้นจนกว่าจะเจอ ซึ่งทำให้เกิดการล่าช้า แต่หากมีคีย์รองเป็นชื่อและนามสกุลก็จะสามารถช่วยให้ DBMS ทำการค้นหาได้สะดวกรวดเร็วยิ่งขึ้น

นักศึกษา

รหัสประจำตัวนักศึกษา	ชื่อ	นามสกุล	เพศ	อายุ	...
1501000011	สมคิด	ดำเนนศิริ	ชาย	21	...
1501000012	วัลภา	ตั้งประเสริฐ	หญิง	21	...
1501000013	ศิริโกศล	มีสุข	หญิง	20	...
1501000014	ดำรง	ดีดี	ชาย	21	...

mindphp.com

↑
"Primary Key"

↑ ↑
"Secondary Key"

- #คีย์รองยอมให้ข้อมูลซ้ำกันได้ แต่หากข้อมูลซ้ำกันมากก็ไม่เป็นผลดี

Compound Key (คีย์รวม)



- ในบางครั้งเรียกคีย์ชนิดนี้ว่า Composite Key เป็นคีย์ที่ใช้คอลัมน์หลายคอลัมน์มารวมกันเป็นคีย์หลัก

นักศึกษา

ชื่อ	นามสกุล	เพศ	อายุ	...
สมคิด	ดำเนนศิริ	ชาย	21	...
วัลภา	ตั้งประเสริฐ	หญิง	21	...
ศิริโกศล	มีสุข	หญิง	20	...
ดำรง	คิตติ	ชาย	21	...



- #เนื่องจากในบางครั้งการกำหนด Primary Key เพียงคอลัมน์เดียวอาจเกิดการซ้ำของข้อมูลได้

Candidate Key (คีย์คู่แข่ง)



- เมื่อมีคอลัมน์ที่มีคุณสมบัติครบถ้วนในการเป็น Primary Key คือ ไม่มีค่าซ้ำ ไม่เป็นค่าว่าง

นักศึกษา

รหัสประจำตัวนักศึกษา	ชื่อ	นามสกุล	เพศ	อายุ	...
1501000011	สมคิด	ดำเนนศิริ	ชาย	21	...
1501000012	วัลภา	ตั้งประเสริฐ	หญิง	21	...
1501000013	ศิริโกศล	มีสุข	หญิง	20	...
1501000014	ดำรง	คิดดี	ชาย	21	...

↑
↑
"Candidate Key"

- จะเห็นว่า มี Candidate Key อยู่ 2 ตัวคือ รหัสประจำตัวนักศึกษาและชื่อ (ถ้ามั่นใจแล้วว่าชื่อนักศึกษาจะไม่มีค่าซ้ำกันเลย)
#ในกรณีนี้หากผู้ออกแบบฐานข้อมูลกำหนดให้คอลัมน์รหัสประจำตัวนักศึกษาเป็น Primary Key ดังนั้นคอลัมน์ชื่อก็จะกลายเป็น Secondary Key อัตโนมัติ

ตัวอย่าง Candidate Key



- เพื่อให้เข้าใจยิ่งขึ้นให้ดูอีกตัวอย่างหนึ่งดังรูป

นักศึกษา

รหัสประจำตัวนักศึกษา	รหัสบัตรประชาชน	ชื่อ	นามสกุล	เพศ	อายุ	...
1501000011	1604500123653	สมคิด	ดำเนนศิริ	ชาย	21	...
1501000012	1803775500909	วัลภา	ตั้งประเสริฐ	หญิง	21	...
1501000013	1200450086598	ศิริโกศล	มีสุข	หญิง	20	...
1501000014	1699000897033	ดำรง	ดิติดี	ชาย	21	...

mindphp.com

↑
↑
"Candidate Key"

- จะเห็นได้อย่างชัดเจนว่า รหัสประจำตัวนักศึกษาและรหัสบัตรประชาชน จะไม่เป็นค่าซ้ำและค่าว่างอย่างแน่นอน ดังนั้นรหัสประจำตัวนักศึกษาและรหัสบัตรประชาชน คือ Candidate Key

Foreign Key (คีย์นอก)



- เป็นคีย์ที่ใช้เชื่อมความสัมพันธ์กับตารางอื่นๆ
- ตารางนักศึกษามีคอลัมน์รหัสประจำตัวนักศึกษาเป็น Primary Key
- ในตารางการลงทะเบียนมีคอลัมน์รหัสวิชาเป็น Primary Key แต่เมื่อ 2 ตารางมีความสัมพันธ์กัน คอลัมน์รหัสประจำตัวนักศึกษาจะเป็น Foreign Key ของตารางการลงทะเบียน ซึ่งความสัมพันธ์เป็นแบบ One-to-Many



mindphp.com

คำศัพท์ที่เกี่ยวข้องกับ SQL



- Data Definition Language (DDL) การสร้างหรือปรับปรุงโครงสร้างของตารางในฐานข้อมูล
 - Create/alter/delete tables and their attributes
- Data Manipulation Language (DML) การสืบค้น การแทรก การปรับปรุง การลบข้อมูลในตาราง
 - Query one or more tables
 - Insert/delete/modify tuples in tables

คำสั่งพื้นฐานของ SQL



- CREATE DATABASE
- CREATE TABLE
- CREATE VIEW
- INSERT
- SELECT
- UPDATE
- DELETE
- DROP

คำสั่งสร้างฐานข้อมูล CREATE DATABASE



- โดยชื่อ database ในระบบจะต้องไม่ซ้ำกันภายใน RDBMS ที่ใช้งาน

```
CREATE DATABASE dbname;
```

คำสั่งสร้างตาราง CREATE TABLE



- การสร้างตารางเพื่อเอาไว้เก็บข้อมูลที่ต้องการ เรียกคำสั่งนี้ว่า SQL Create table statement
- ข้อมูลแบบตารางต้องกำหนดค่าของ column รวมถึง data type ของแต่ละ column ว่าจะใช้เก็บข้อมูลชนิดใด

```
CREATE TABLE table_name (
    column1 datatype,
    column2 datatype,
    column3 datatype,
    ....
    PRIMARY KEY (one or more columns)
);
```



```
CREATE TABLE Persons (
    ID int NOT NULL,
    FirstName varchar(255),
    LastName varchar(255),
    Age int
    PRIMARY KEY (ID)
);
```

Name	Value
Updated Rows	0
Query	CREATE TABLE Persons2 (ID int NOT NULL, LastName varchar(255) NOT NULL, FirstName varchar(255), Age int, PRIMARY KEY (ID))
Finish time	Fri Aug 14 20:53:44 ICT 2020

ตัวอย่าง data types บน database ส่วนใหญ่



Data type	Description
CHARACTER(n)	Character string จำกัดจำนวนตัวอักษรเท่ากับ n ตัว
VARCHAR(n) or CHARACTER VARYING(n)	Character string จำกัดจำนวนตัวอักษรไม่มากกว่า n ตัว
BINARY(n)	Binary string จำกัดจำนวนตัวอักษรเท่ากับ n bytes
VARBINARY(n) or BINARY VARYING(n)	Binary string จำกัดจำนวนตัวอักษรไม่มากกว่า n bytes
BOOLEAN	TRUE หรือ FALSE คือข้อมูลแบบตรรกศาสตร์
INTEGER(p)	ตัวเลขจำนวนเต็ม p หลัก
SMALLINT	ตัวเลขจำนวนเต็ม 5 หลัก
INTEGER	ตัวเลขจำนวนเต็ม 10 หลัก
BIGINT	ตัวเลขจำนวนเต็ม 19 หลัก
DECIMAL(p,s)	ตัวเลขที่มีจำนวนทั้งหมด p หลัก เป็นทศนิยม s หลัก และ จำนวนเต็ม p-s หลัก
NUMERIC(p,s)	ตัวเลขที่มีจำนวนทั้งหมด p หลัก เป็นทศนิยม s หลัก และ จำนวนเต็ม p-s หลัก
FLOAT(p)	ตัวเลขค่าประมาณรูปทศนิยมจำนวน p หลัก
REAL	ตัวเลขค่าประมาณรูปทศนิยมจำนวน 7 หลัก
FLOAT	ตัวเลขค่าประมาณรูปทศนิยมจำนวน 16 หลัก
DOUBLE PRECISION	ตัวเลขค่าประมาณรูปทศนิยมจำนวน 16 หลัก
DATE	ข้อมูล วัน,เดือน,ปี
TIME	ข้อมูล ชั่วโมง นาที วินาที
TIMESTAMP	ข้อมูล วัน เดือน ปี ชั่วโมง นาที วินาที
INTERVAL	ข้อมูลความต่างในทางเวลา
ARRAY	ชุดของข้อมูลที่มีการจัดเรียง
MULTISET	ชุดของข้อมูลที่ไม่มีการจัดเรียง อาจมีข้อมูลซ้ำได้
XML	ข้อมูลชนิด XML

จัดกลุ่มของ
SQL Data Types
ที่มีลักษณะเดียวกัน
ได้ 6 กลุ่มคือ



- ตัวเลข
- ตัวเลขค่าประมาณ
- วันและเวลา
- ตัวอักษร (Non Unicode)
- ตัวอักษร (Unicode)
- ตัวอักษร (Binary)

กลุ่มของ SQL Data Types : ตัวเลข



DATA TYPE	FROM	TO
bigint	-9,223,372,036,854,775,808	9,223,372,036,854,775,807
int	-2,147,483,648	2,147,483,647
smallint	-32,768	32,767
tinyint	0	255
bit	0	1
decimal	$-10^{38} + 1$	$10^{38} - 1$
numeric	$-10^{38} + 1$	$10^{38} - 1$
money	-922,337,203,685,477.5808	+922,337,203,685,477.5807
smallmoney	-214,748.3648	+214,748.3647

กลุ่มของ SQL
Data Types :
ตัวเลขค่าประมาณ



DATA TYPE	FROM	TO
float	-1.79E + 308	1.79E + 308
real	-3.40E + 38	3.40E + 38

กลุ่มของ SQL Data Types : วันและเวลา



DATA TYPE	FROM	TO
datetime	Jan 1, 1753	Dec 31, 9999
smalldatetime	Jan 1, 1900	Jun 6, 2079
date	รูปแบบวัน June 30, 1991	
time	รูปแบบเวลา 12:30 P.M.	

กลุ่มของ SQL Data Types : ตัวอักษร (Non Unicode)



DATA TYPE	Description
char	สูงสุด 8,000 ตัวอักษร (จำนวนตัวอักษรต้องเท่ากัน และ ไม่ใช่ Unicode)
varchar	สูงสุด 8,000 ตัวอักษร (จำนวนตัวอักษรต้องไม่เกิน และ ไม่ใช่ Unicode)
text	สูงสุด 2,147,483,647 ตัวอักษร (จำนวนตัวอักษรต้องไม่เกิน และ ไม่ใช่ Unicode)

กลุ่มของ SQL Data Types : ตัวอักษร (Unicode)



DATA TYPE	Description
nchar	สูงสุด 4,000 ตัวอักษร (จำนวนตัวอักษรต้องเท่ากัน และเป็น Unicode)
nvarchar	สูงสุด 4,000 ตัวอักษร (จำนวนตัวอักษรต้องไม่เกิน และเป็น Unicode)
ntext	สูงสุด 1,073,741,823 ตัวอักษร (จำนวนตัวอักษรต้องไม่เกิน และเป็น Unicode)

กลุ่มของ SQL Data Types : ตัวอักษร (Binary)



DATA TYPE	Description
binary	สูงสุด 8,000 bytes (Fixed-length binary data)
varbinary	สูงสุด 8,000 bytes (Variable length binary data)
image	สูงสุด 2,147,483,647 bytes (Variable length Binary Data)

Workshop



Workshop 1

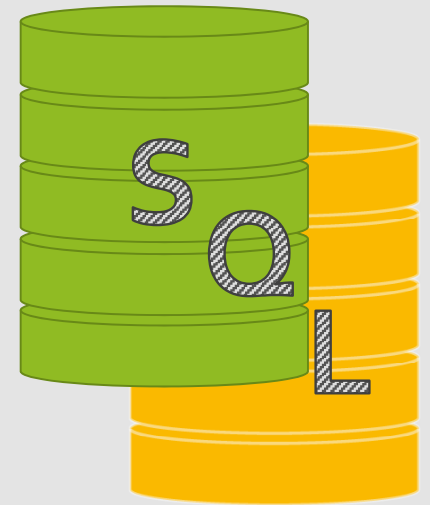
ดาวน์โหลดและติดตั้ง Dbeaver

Workshop 2

สร้างฐานข้อมูล ตาราง และกำหนดประเภทของข้อมูล

Dbeaver


ขั้นตอนการติดตั้ง



<https://dbeaver.io>

Dbeaver





DBEaver Community

Free Universal Database Tool

Star 21,904
Follow @dbeaver_news

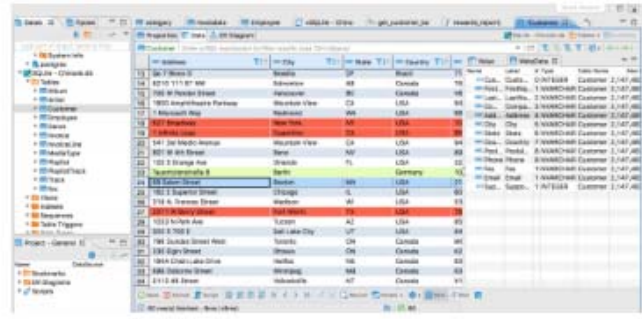
Go

Home
About
Download
Sources
Documentation
News
Support
Enterprise Edition
CloudBeaver

Universal Database Tool

Free multi-platform database tool for developers, database administrators, analysts and all people who need to work with databases. Supports all popular databases: MySQL, PostgreSQL, SQLite, Oracle, DB2, SQL Server, Sybase, MS Access, Teradata, Firebird, Apache Hive, Phoenix, Presto, etc.

Download



DBEaver 21.1.5

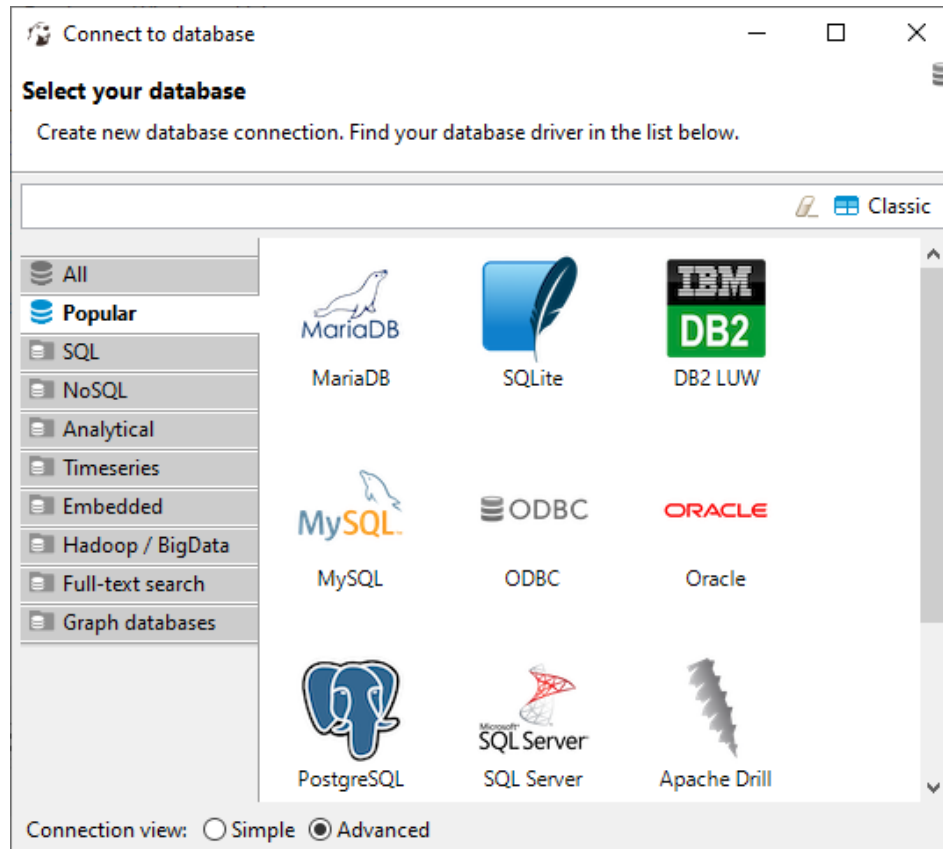
August 16th, 2021

- ERD editor:
 - Properties panel was added for diagram elements
 - Diagram drag tool was added
 - Entity search UI was improved

Dbeaver

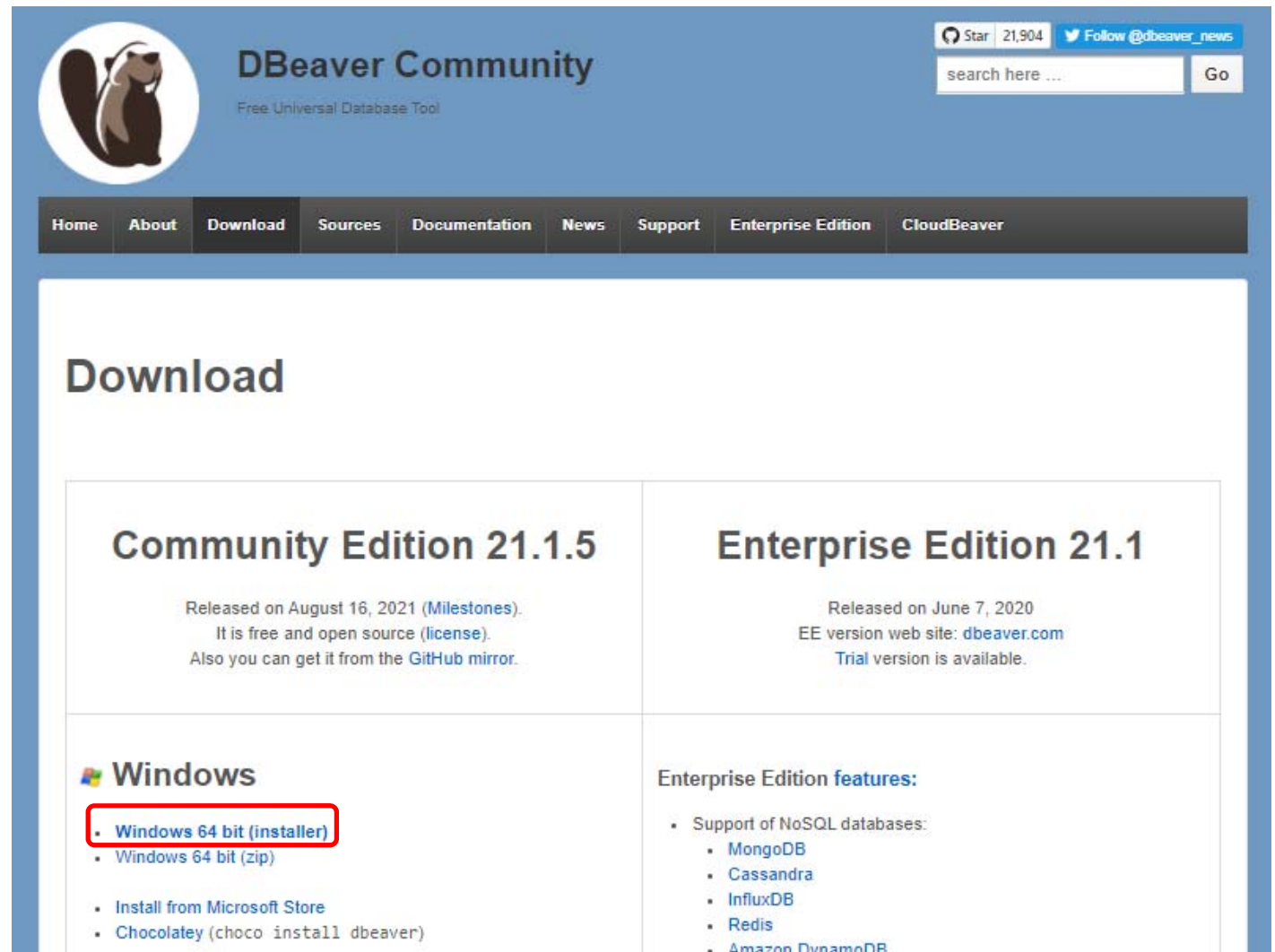


เครื่องมือฐานข้อมูลแบบหลายแพลตฟอร์มฟรี สำหรับนักพัฒนาระบบ ผู้ดูแลระบบฐานข้อมูล นักวิเคราะห์และทุกคนที่ต้องการทำงานกับฐานข้อมูล รองรับฐานข้อมูลยอดนิยมทั้งหมด: MySQL, PostgreSQL, SQLite, Oracle, DB2, SQL Server, Sybase, MS Access, Teradata, Firebird, Apache Hive, Phoenix, Presto และอื่น ๆ



<https://dbeaver.io/download/>

Dbeaver



The screenshot shows the DBeaver Community website. At the top, there is a navigation bar with links for Home, About, Download, Sources, Documentation, News, Support, Enterprise Edition, and CloudBeaver. The main content area is titled "Download" and is divided into two columns. The left column is for "Community Edition 21.1.5", released on August 16, 2021, and is free and open source. The right column is for "Enterprise Edition 21.1", released on June 7, 2020, and has a trial version available. Under the "Community Edition 21.1.5" section, there is a "Windows" sub-section with a list of download options: "Windows 64 bit (installer)", "Windows 64 bit (zip)", "Install from Microsoft Store", and "Chocolatey (choco install dbeaver)". The "Windows 64 bit (installer)" option is highlighted with a red box. The "Enterprise Edition 21.1" section lists features such as support for NoSQL databases: MongoDB, Cassandra, InfluxDB, Redis, and Amazon DynamoDB.

<https://softfamous.com/dbeaver/download/>

Dbeaver



SOFT FAMOUS

HAVAL H6 Hybrid SUV

H6 Hybrid SUV รถ SUV ดีไซน์ล้ำสมัย มาพร้อมเทคโนโลยีอัจฉริยะ All New

[Home](#) ▶ [Windows](#) ▶ [Web Development](#) ▶ [Database Utils](#) ▶ [DBeaver](#)

Download DBeaver for Windows

Choose a mirror

Download



64 bit

Download



32 bit

DBeaver free download. Always available from the SoftFamous servers

https://softfamous.com/postdownload-file/dbeaver/10618/4242/

Dbeaver




SOFTFAMOUS

[Home](#) ▶ [Windows](#) ▶ [Web Development](#) ▶ [Database Utils](#) ▶ [DBeaver](#)

Downloading ...

The download started.
Do not close the tab before the download is completed.

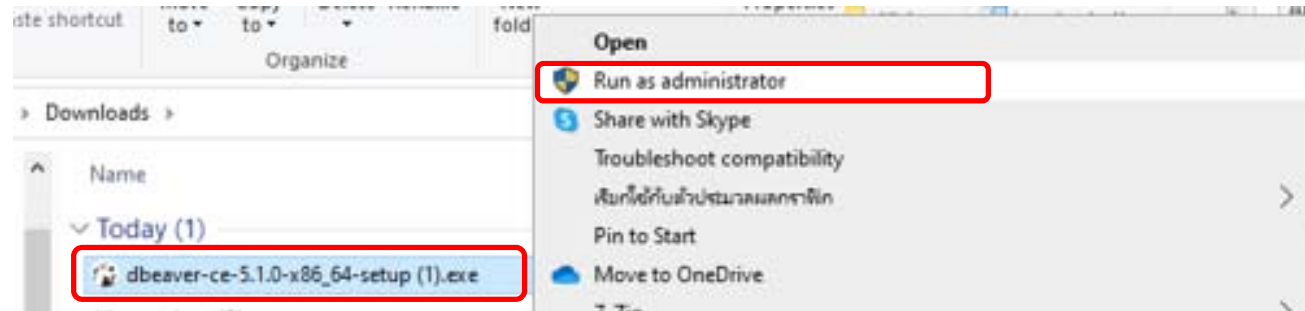
How to install your download?

 dbeaver-ce-5.1.0-x...exe ^

Dbeaver



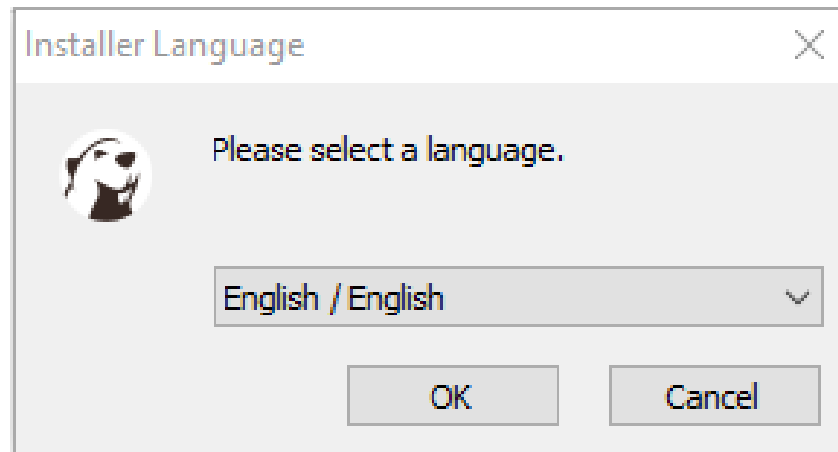
คลิกขวาที่ dbeaver-ce-5.10-x86_64setup.exe เลือก Run as administrator



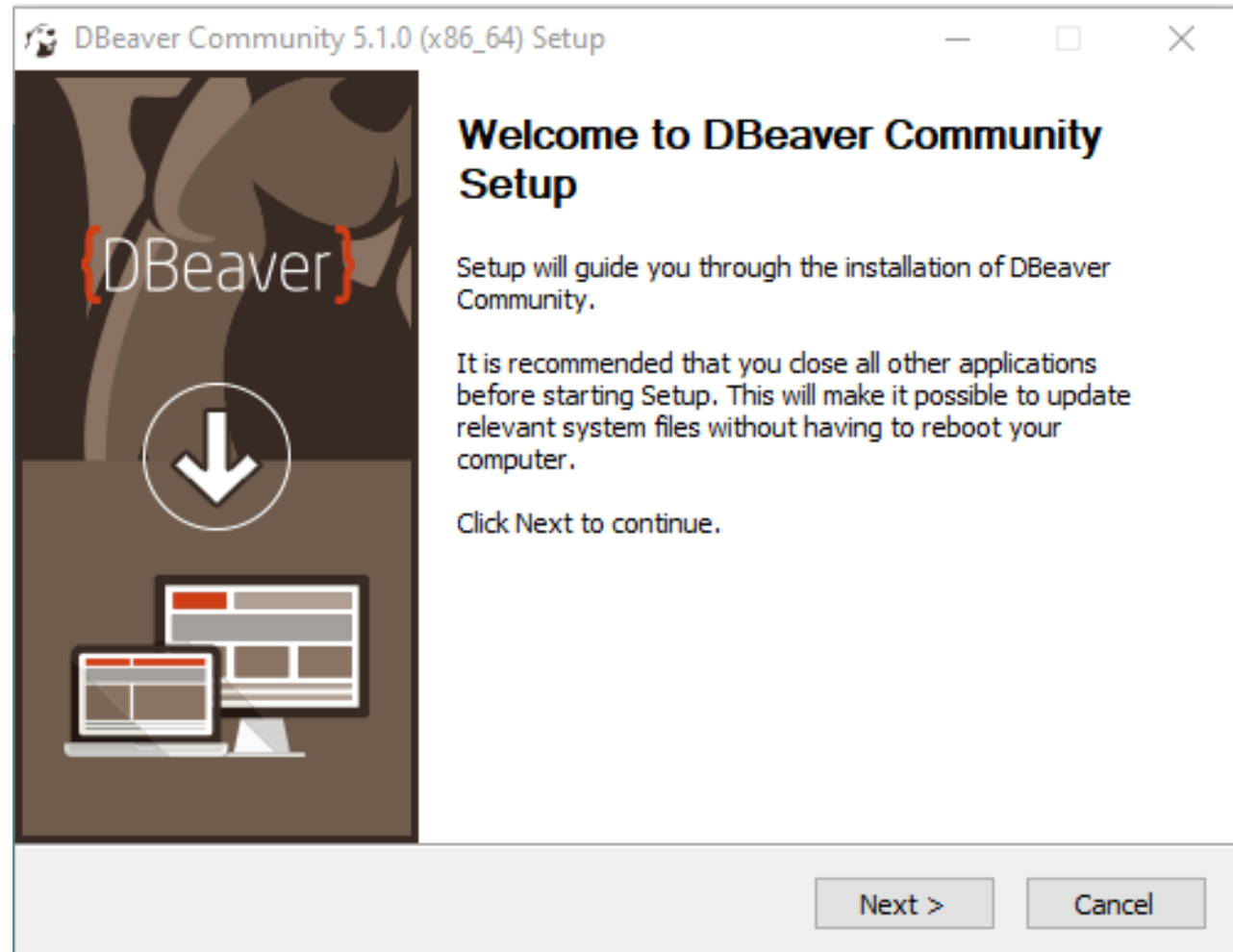
Dbeaver



เลือกภาษา English แล้วกด OK



Dbeaver



Dbeaver



DBeaver Community 5.1.0 (x86_64) Setup

License Agreement

Please review the license terms before installing DBeaver Community.

Press Page Down to see the rest of the agreement.

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

If you accept the terms of the agreement, click I Agree to continue. You must accept the agreement to install DBeaver Community.

Universal Database Manager

< Back I Agree Cancel

Dbeaver



DBeaver Community 5.1.0 (x86_64) Setup

Choose Installation Options

Who should this application be installed for?



Please select whether you wish to make this software available to all users or just yourself

- Anyone who uses this computer (all users)
- Only for me (Guitar)

There is already a per-machine installation. (C:\Program Files\DBeaver)
Will reinstall/upgrade.

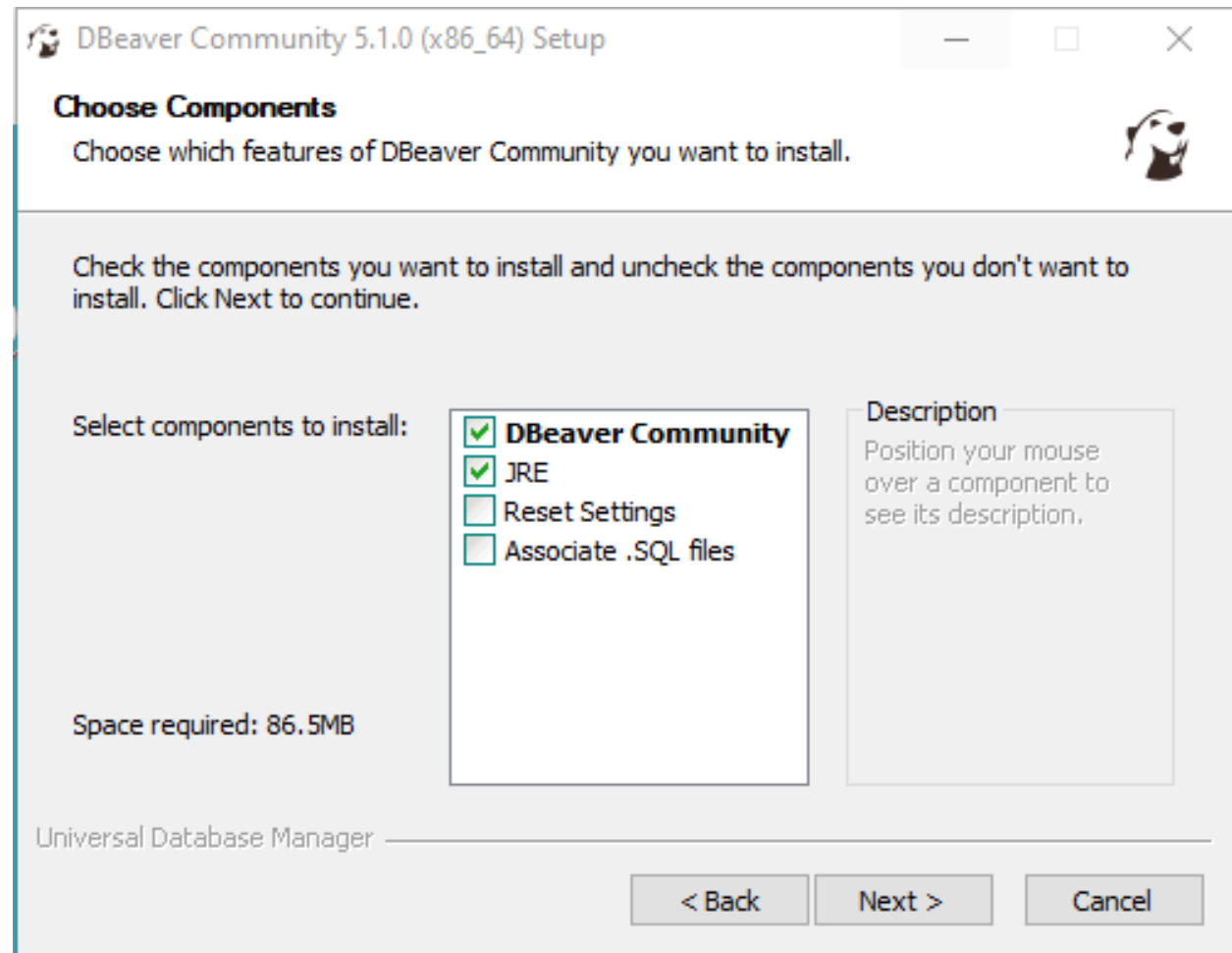
Universal Database Manager

< Back

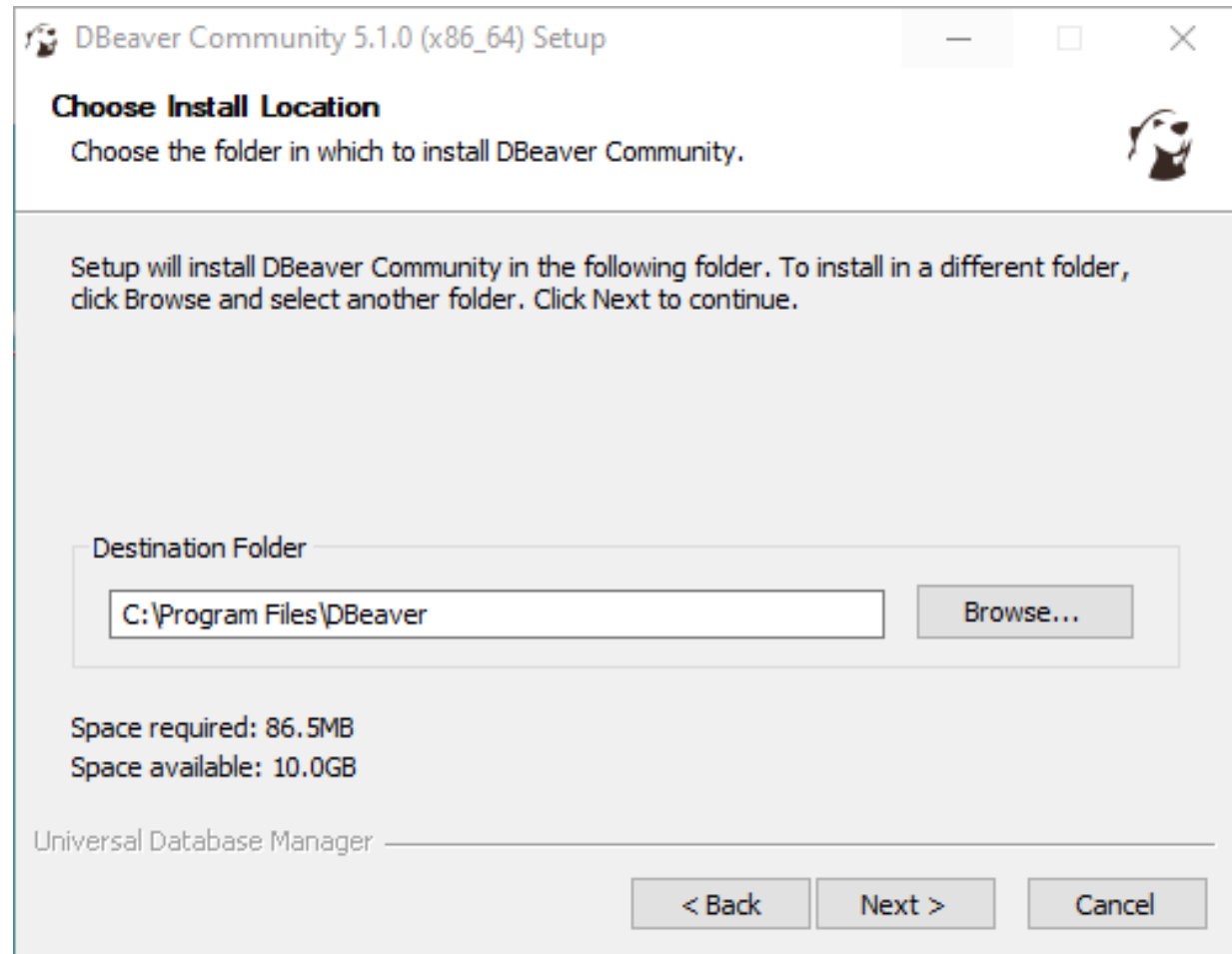
Next >

Cancel

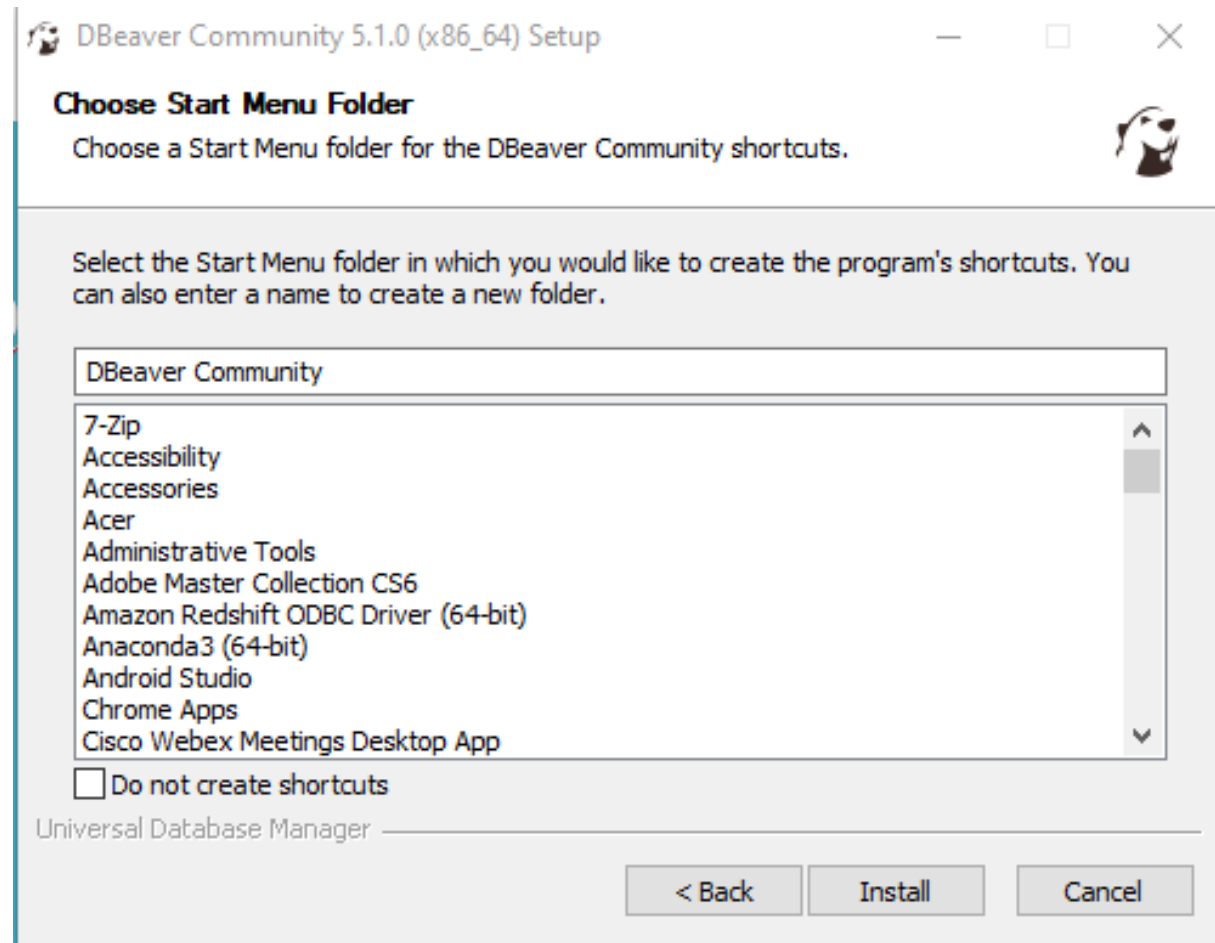
Dbeaver



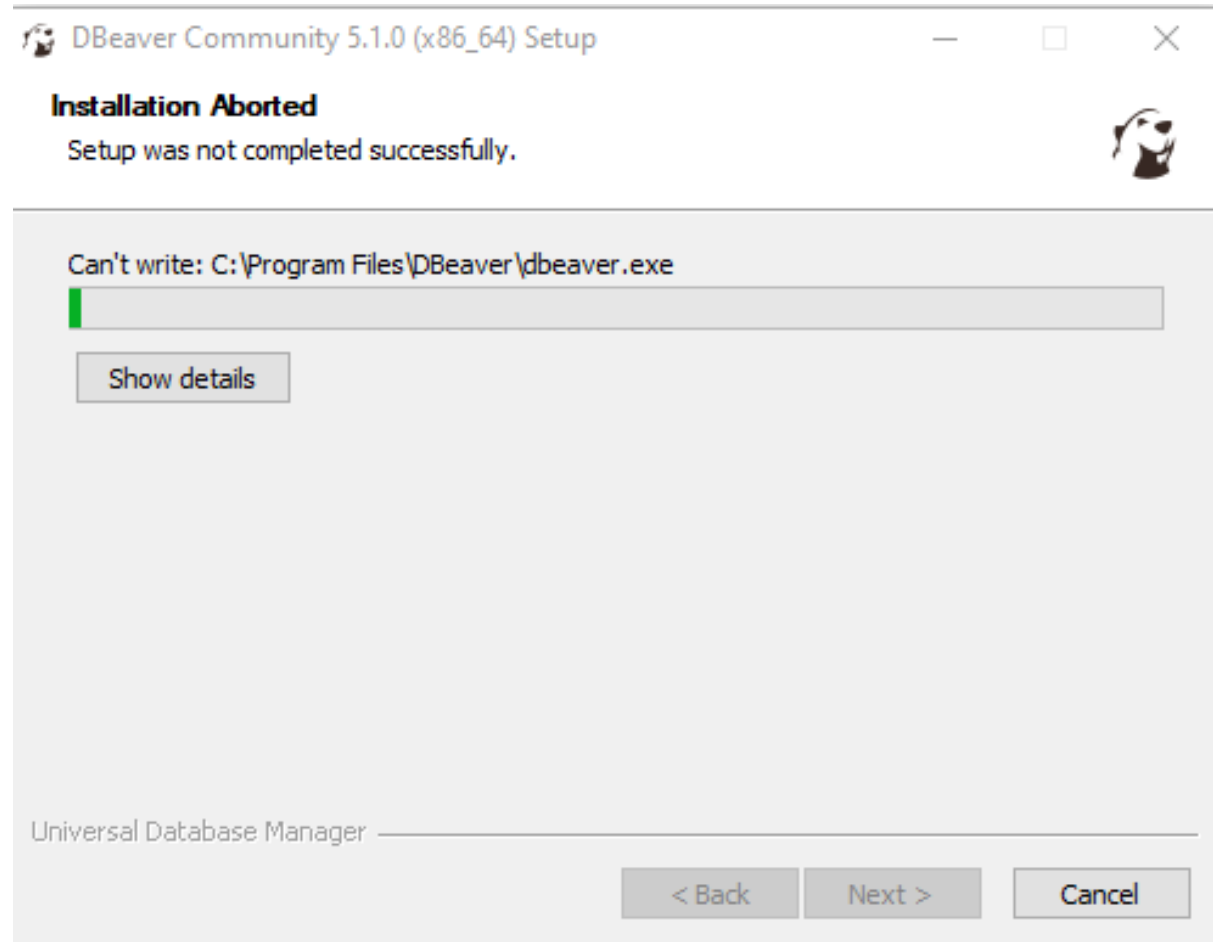
Dbeaver



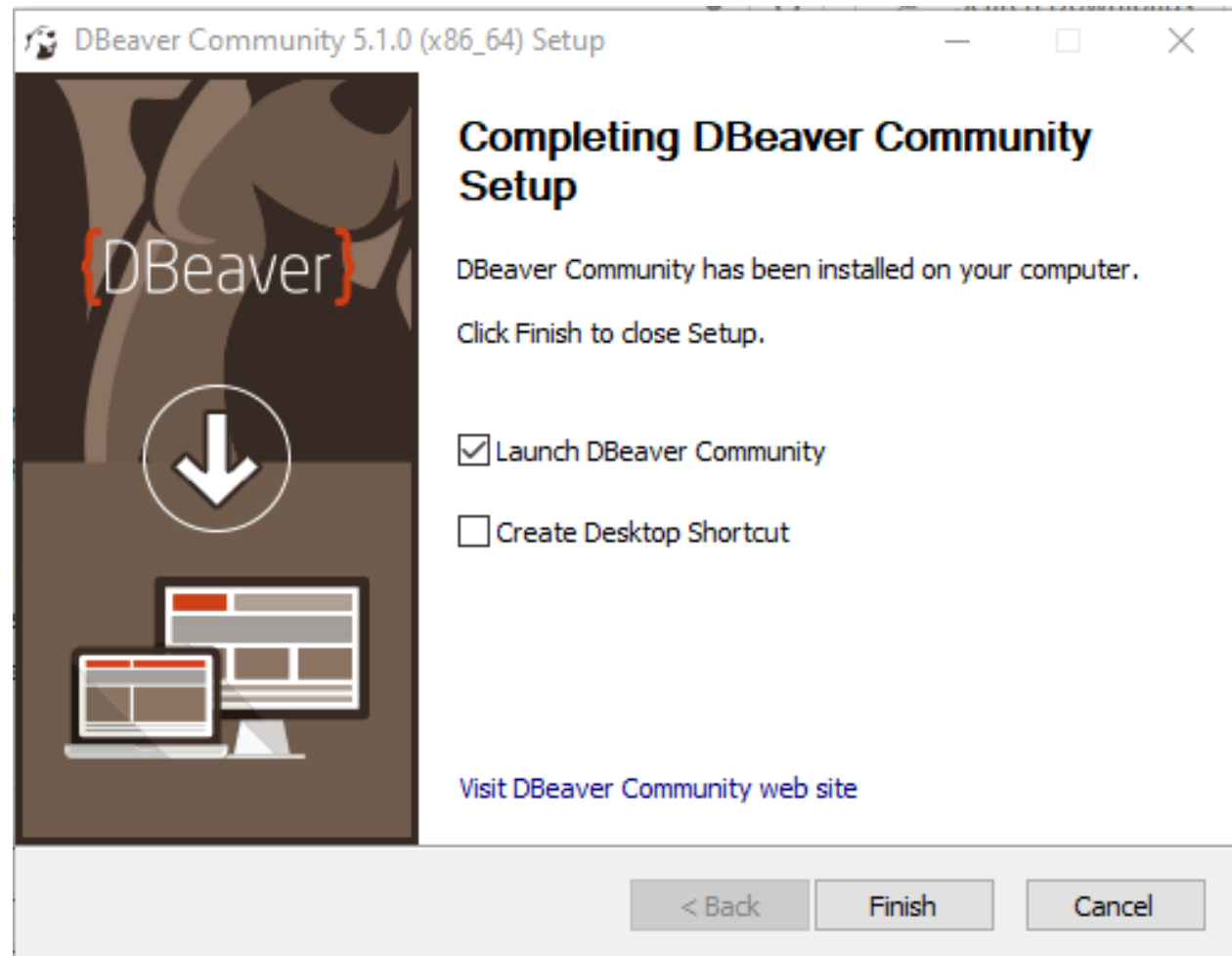
Dbeaver



Dbeaver



Dbeaver



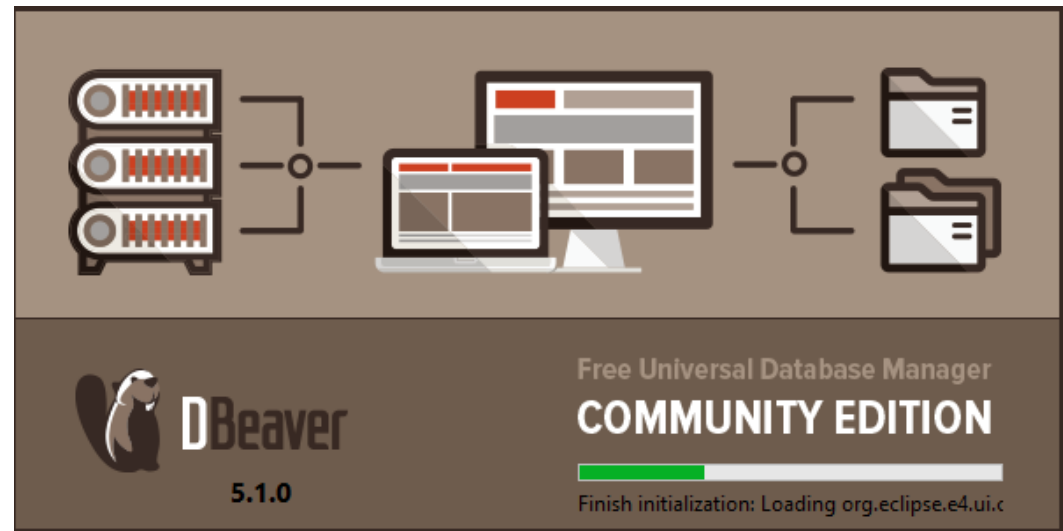
Dbeaver



ที่หน้าจอ Desktop มี Short cut ชื่อ Dbeaver ดังรูป
แล้วดับเบิลคลิกที่ Dbeaver



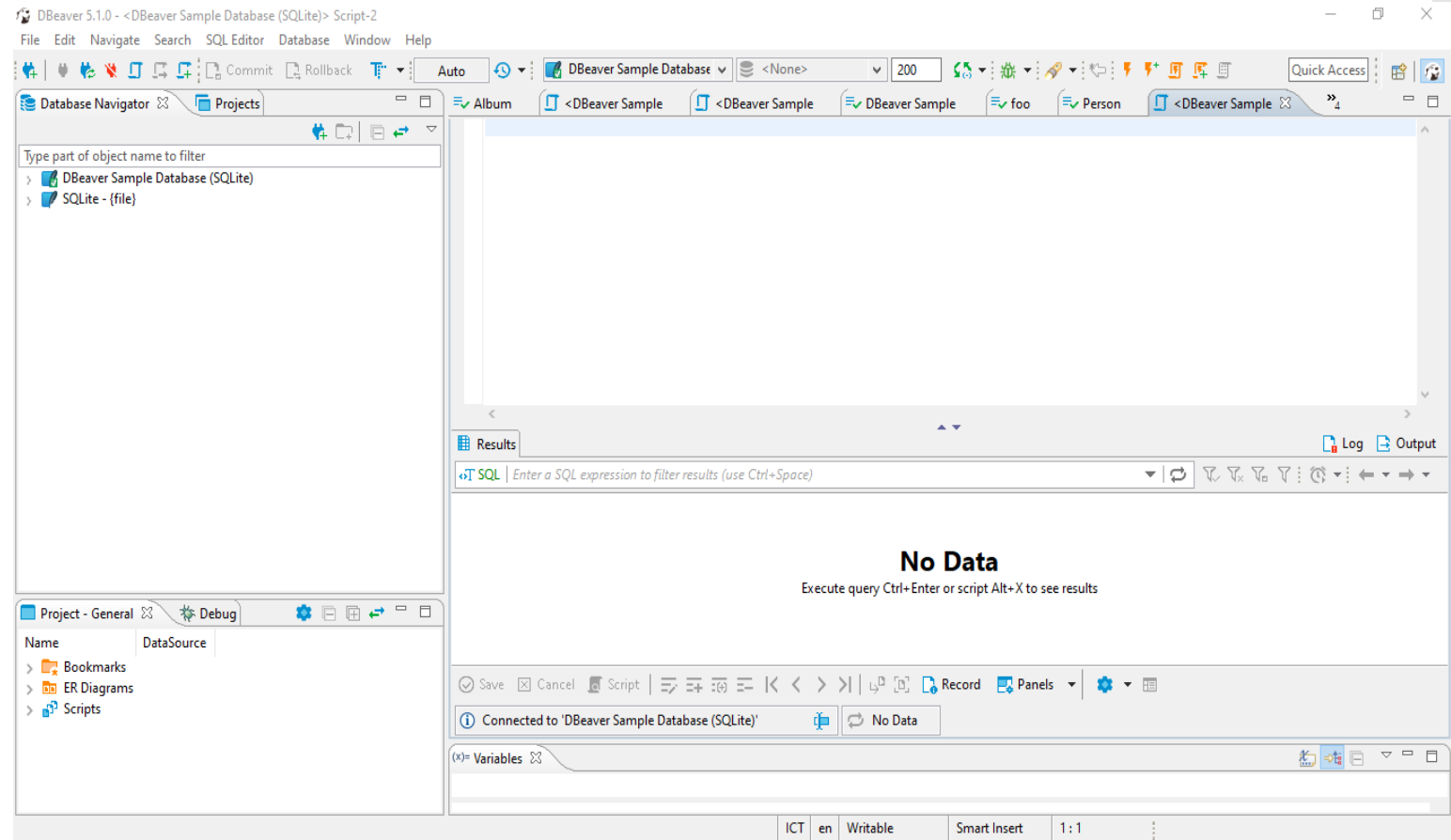
จะขึ้น



Dbeaver



คลิก Dbeaver Sample Database (SQLite)



Dbeaver



DBEaver 5.1.0 - <DBEaver Sample Database (SQLite)> Script-2

File Edit Navigate Search SQL Editor Database Window Help

Commit Rollback Auto DBEaver Sample Database <None> 200

Database Navigator Projects Album <DBEaver Sample <DBEaver Sample DBEaver Sample foo Person

Type part of object name to filter

- DBEaver Sample Database (SQLite)
 - Tables
 - Album
 - Artist
 - Customer
 - Employee3
 - Genre
 - Invoice
 - InvoiceLine
 - MediaType
 - Person
 - Playlist
 - PlaylistTrack
 - Track
 - Tracks_new
 - foo
 - Views
 - Indexes
 - Sequences
 - Table Triggers

Results

SQL | Enter a SQL expression to filter results (use Ctrl+Space)

No Data
Execute query Ctrl+Enter or script Alt+X to see results

Project - General Debug

Name	DataSource
Bookmarks	
ER Diagrams	
Scripts	

Save Cancel Script Record Panels

Connected to 'DBEaver Sample Database (SQLite)' No Data

Dbeaver



DBeaver 5.1.0 - Album

File Edit Navigate Search SQL Editor Database Window Help

Auto DBEaver Sample Database <None> 200

Database Navigator Projects

Type part of object name to filter

- DBEaver Sample Database (SQLite)
 - Tables
 - Album
 - Artist
 - Customer
 - Employee3
 - Genre
 - Invoice
 - InvoiceLine
 - MediaType
 - Person
 - Playlist
 - PlaylistTrack
 - Track
 - Tracks_new
 - foo
 - Views
 - Indexes
 - Sequences
 - Table Triggers

Properties Data ER Diagram

Album | Enter a SQL expression to filter results (use Ctrl+Space)

	123 AlbumId	ABC Title	123 ArtistId	Column1
1	1	For Those About To Rock We Salute You	1	oPNG IHDR
2	2	Balls to the Wall	2	oPNG IHDR n n
3	3	Restless and Wild	2	[NULL]
4	4	Let There Be Rock	1	ewqewq
5	5	Big Ones	3	[NULL]
6	6	Jagged Little Pill	4	[NULL]
7	7	Facelift	5	[NULL]
8	8	Warner 25 Anos	6	[NULL]
9	9	Plays Metallica By Four Cellos	7	[NULL]
10	10	Audioslave	8	[NULL]
11	11	Out Of Exile	8	[NULL]
12	12	BackBeat Soundtrack	9	[NULL]
13	13	The Best Of Billy Cobham	10	[NULL]
14	14	Alcohol Fueled Brewtality Live! [Disc 1]	11	[NULL]
15	15	Alcohol Fueled Brewtality Live! [Disc 2]	11	[NULL]
16	16	Black Sabbath	12	[NULL]
17	17	Black Sabbath Vol. 4 (Remaster)	12	[NULL]
18	18	Body Count	13	[NULL]
19	19	Chemical Wedding	14	[NULL]

Project - General Debug

Name DataSource

- Bookmarks
- ER Diagrams
- Scripts

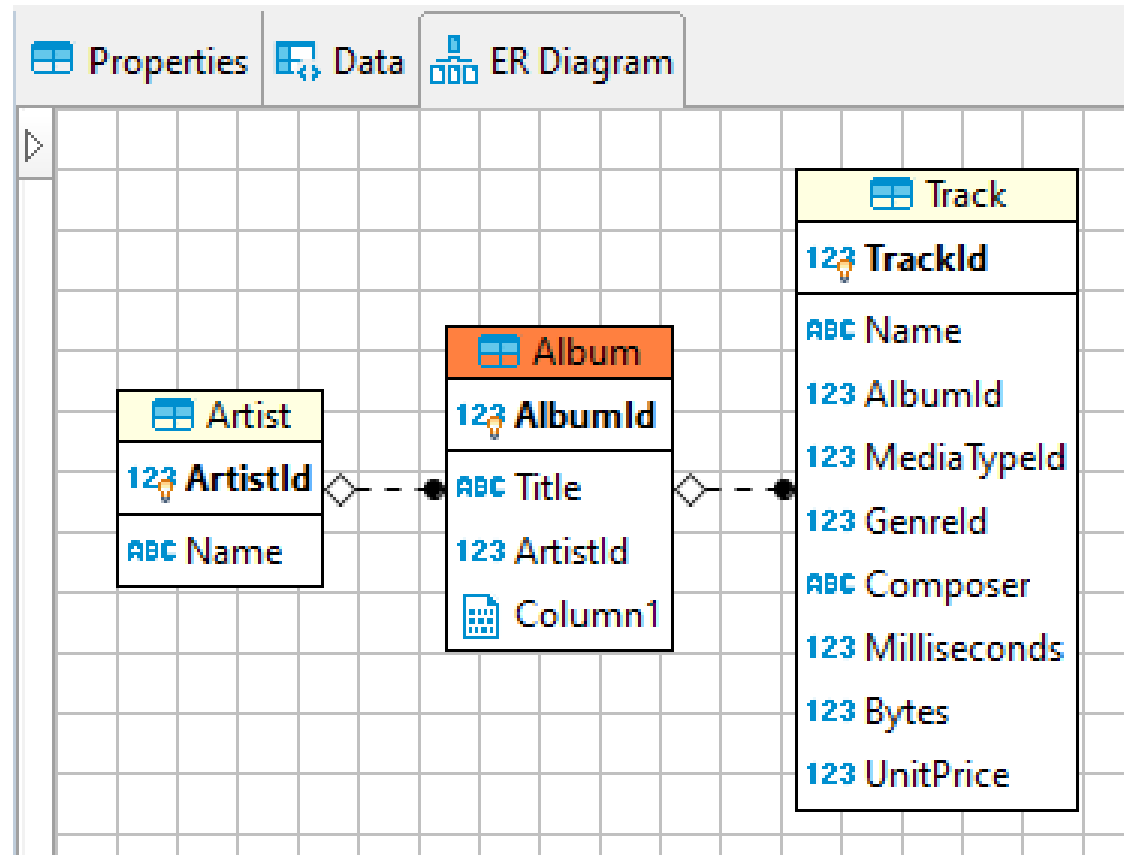
Save Cancel Script | Record Panels

200 row(s) fetched - 9ms (+5ms) 200+

Dbeaver



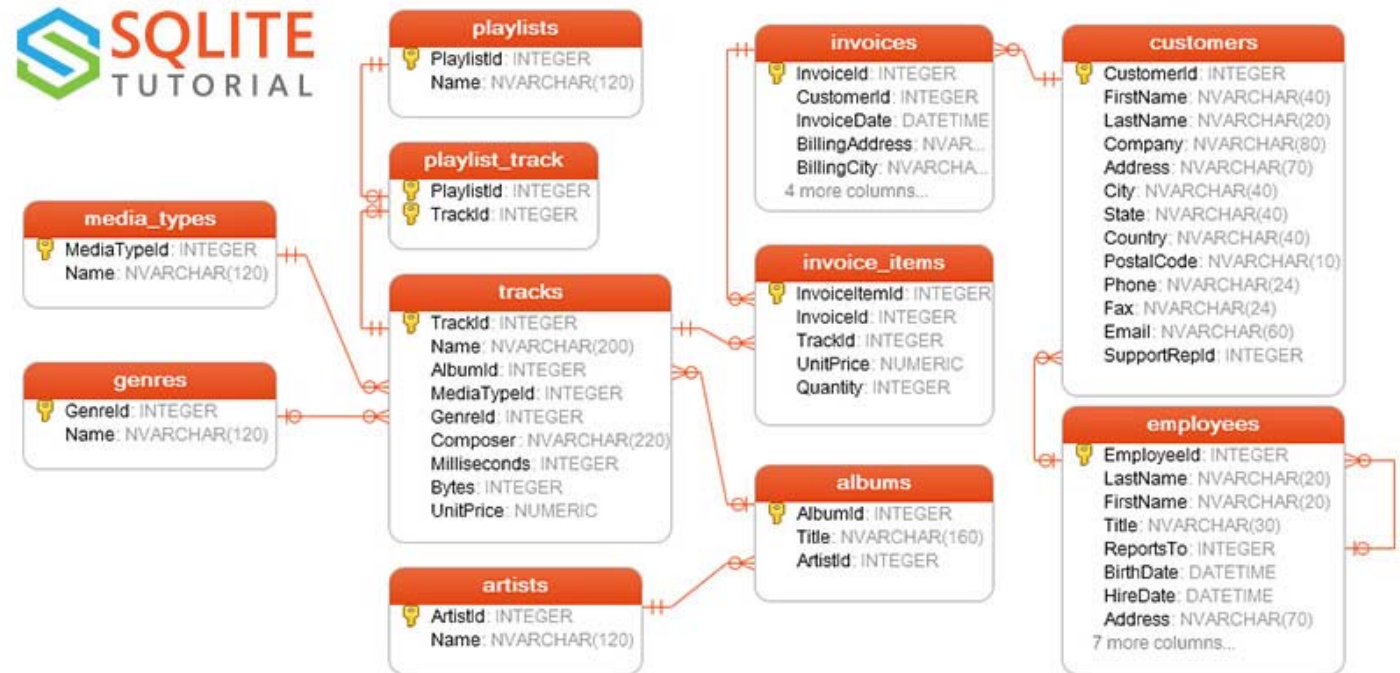
คลิกแท็บ ER Diagram



SQLite



- Chinook sample database tables



การปรับปรุงฐานข้อมูล



การปรับปรุงฐานข้อมูลมีอยู่ 3 ลักษณะ

- Insertions การแทรกข้อมูลเข้าสู่ตาราง
- Deletions การลบข้อมูลในตาราง
- Updates การปรับปรุงข้อมูลในตาราง

บางครั้งทั้งหมดนี้ถูกเรียกว่า “update”

- รูปแบบ

```
INSERT INTO table(A1,...., An) VALUES (v1,...., vn)
```

- ตัวอย่าง: การแทรกข้อมูลเข้าสู่ตาราง Tracks_new:

```
CREATE TABLE Tracks_new (
    Name NVARCHAR(200),
    Albumid INTEGER,
    Composer NVARCHAR(220)
);
```

```
INSERT INTO Tracks_new (Name,Albumid,Composer)
VALUES ('Fast As a Shark',3,'F. Baltes, S. Kaufman, U.
Dirksneider & W. Hoffman');
```

Insertions การแทรก
ข้อมูลเข้าสู่ตาราง



Insertions การแทรก
ข้อมูลจากราง 1
ทุกคอลัมน์เข้าสู่
ตาราง 2
โดยใช้ร่วมกับคำสั่ง
select



```
INSERT INTO table2
SELECT * FROM table1
WHERE condition;
```

- ตัวอย่าง

```
INSERT INTO Tracks_new(Name,Albumid,Composer )
SELECT Name,Albumid,Composer FROM Track
WHERE Composer LIKE '%Smith%'
ORDER BY Albumid
```

Name	Value
Updated Rows	97
Query	INSERT INTO Tracks_new(Name,Albumid,Composer) SELECT Name,Albumid,Composer FROM Track WHERE Composer LIKE '%Smith%' ORDER BY Albumid
Finish time	Fri Aug 14 21:35:00 ICT 2020

Insertions การแทรก
ข้อมูลจากตารางบาง
คอลัมน์เข้าสู่ตาราง 2
โดยใช้ร่วมกับคำสั่ง
select แบบมีเงื่อนไข



```
INSERT INTO table2 (column1, column2, column3, ...)
SELECT column1, column2, column3, ...
FROM table1
WHERE condition;
```

```
INSERT INTO Tracks_new(Name, Albumid, Composer )
SELECT
FROM Track
WHERE Composer LIKE '%Smith%'
ORDER BY Albumid
```

Insertions การแทรก
ข้อมูลจากตารางบาง
คอลัมน์เข้าสู่ตาราง 2
โดยใช้ร่วมกับคำสั่ง
select แบบมีเงื่อนไข



• ตัวอย่าง

```
INSERT INTO Tracks_new(Name, Albumid, Composer )
SELECT
Name, Albumid, Composer
FROM Track
WHERE Composer LIKE '%Smith%'
ORDER BY Albumid
```

	ABC Name	123 Albumid	ABC Composer
1	Restless and W	3	F. Baltes, R.A. Smith-Diesel, S. I
2	Princess of the	3	Deaffy & R.A. Smith-Diesel
3	Killing Floor	19	Adrian Smith
4	Machine Men	19	Adrian Smith
5	2 Minutes To M	95	Adrian Smith/Bruce Dickinson
6	Can I Play With	96	Adrian Smith/Bruce Dickinson,

Deletions การลบ



- ตัวอย่าง :

```
DELETE FROM Tracks_new
WHERE Albumid = 3 AND Composer LIKE '%Smith%'
```

จะเห็นว่า ไม่สามารถลบโดยใช้เงื่อนไขเดียวได้

เนื่องจากมีความสัมพันธ์กันระหว่าง 2 เงื่อนไข

Updates การปรับปรุงแก้ไขข้อมูล

- ตัวอย่าง :

```
UPDATE Tracks_new
SET Albumid = 36
WHERE Composer LIKE 'Adrian Smith';
```

ก่อน

	ABC Name	123 Albumid	ABC Composer
1	Restless and Wi	3	F. Baltes, R.A. Smith-Diesel, S. I
2	Princess of the	3	Deaffy & R.A. Smith-Diesel
3	Killing Floor	19	Adrian Smith
4	Machine Men	19	Adrian Smith
5	2 Minutes To M	95	Adrian Smith/Bruce Dickinson



หลัง

	ABC Name	123 Albumid	ABC Composer
1	Restless and Wi	3	F. Baltes, R.A. Smith-Diesel, S. I
2	Princess of the	3	Deaffy & R.A. Smith-Diesel
3	Killing Floor	36	Adrian Smith
4	Machine Men	36	Adrian Smith
5	2 Minutes To M	95	Adrian Smith/Bruce Dickinson
6	Can I Play With	96	Adrian Smith/Bruce Dickinson

คำสั่ง SELECT



- คำสั่ง SELECT เป็นคำสั่งที่ใช้บ่อยที่สุด
- รูปแบบ

```
SELECT <attributes>  
FROM <one or more relations>  
WHERE <conditions>
```

คำสั่ง SELECT



```
SELECT DISTINCT column_list  
FROM table_list  
    JOIN table ON join_condition  
WHERE row_filter  
ORDER BY column  
GROUP BY column  
HAVING group_filter;
```

ตัวอย่างข้อมูลพนักงาน



- Employee3

	EmployeeId	LastName	FirstName	Title
1	1	Adams	Andrew	General Manager
2	2	Edwards	Nancy	Sales Manager
3	3	Peacock	Jane	Sales Support Agent
4	4	Park	Margaret	Sales Support Agent
5	5	Johnson	Steve	Sales Support Agent
6	6	Mitchell	Michael	IT Manager
7	7	King	Robert	IT Staff
8	8	Callahan	Laura	IT Staff

1 การ SELECT ข้อมูล ทั้งหมด



- เลือกทั้งหมด โดยระบุชื่อตัวแปรทั้งหมด ถ้ามีหลายตัวแปรให้คั่นด้วยเครื่องหมายจุลภาค หรือ comma หรือระบุสัญลักษณ์ * แทนชื่อตัวแปรทั้งหมดจะได้ผลลัพธ์แบบเดียวกัน

```
SELECT EmployeeId, LastName, FirstName, Title, ReportsTo,  
BirthDate, HireDate, Address, City, State, Country, PostalCode,  
Phone, Fax, Email  
FROM Employee3;
```

- หรือ

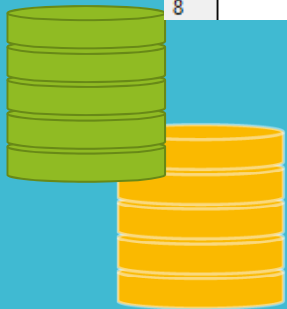
```
SELECT *  
FROM Employee3;
```

1 การ SELECT ข้อมูล ทั้งหมด

- การแสดงผล

Employee *Enter a SQL expression to filter results (use Ctrl+Space)*

	EmployeeId	LastName	FirstName	Title	ReportsTo	BirthDate	HireDate	Address
1	1	Adams	Andrew	General Manager	[NULL]	1962-02-18 00:00:00	2002-08-14 00:00:00	11120 Jasper Ave NW
2	2	Edwards	Nancy	Sales Manager	1	1958-12-08 00:00:00	2002-05-01 00:00:00	825 8 Ave SW
3	3	Peacock	Jane	Sales Support Agent	2	1973-08-29 00:00:00	2002-04-01 00:00:00	1111 6 Ave SW
4	4	Park	Margaret	Sales Support Agent	2	1947-09-19 00:00:00	2003-05-03 00:00:00	683 10 Street SW
5	5	Johnson	Steve	Sales Support Agent	2	1965-03-03 00:00:00	2003-10-17 00:00:00	7727B 41 Ave
6	6	Mitchell	Michael	IT Manager	1	1973-07-01 00:00:00	2003-10-17 00:00:00	5827 Bowness Road NW
7	7	King	Robert	IT Staff	6	1970-05-29 00:00:00	2004-01-02 00:00:00	590 Columbia Boulevard
8	8	Callahan	Laura	IT Staff	6	1968-01-09 00:00:00	2004-03-04 00:00:00	923 7 ST NW



2 การ SELECT ข้อมูล บางคอลัมน์



- ในกรณีที่ต้องการให้แสดงผลเพียงไม่กี่คอลัมน์ หรือ เฉพาะคอลัมน์ที่ต้องการเท่านั้น

```
SELECT EmployeeId, LastName, FirstName, Title, Email  
FROM Employee3;
```


- การแสดงผล

2 การ SELECT ข้อมูล บางคอลัมน์



	EmployeeId	ABC LastName	ABC FirstName	ABC Title	ABC Email
1	1	Adams	Andrew	General Manager	andrew@chinookcorp.com
2	2	Edwards	Nancy	Sales Manager	nancy@chinookcorp.com
3	3	Peacock	Jane	Sales Support Agent	jane@chinookcorp.com
4	4	Park	Margaret	Sales Support Agent	margaret@chinookcorp.com
5	5	Johnson	Steve	Sales Support Agent	steve@chinookcorp.com
6	6	Mitchell	Michael	IT Manager	michael@chinookcorp.com
7	7	King	Robert	IT Staff	robert@chinookcorp.com
8	8	Callahan	Laura	IT Staff	laura@chinookcorp.com

3 การ SELECT ข้อมูล โดยการจัดลำดับการ แสดงผล



- หากต้องการมีการสลับกันของคอลัมน์ในการแสดงผล **SELECT** ก็สามารถทำได้
- >> ต้องการข้อมูลรหัสพนักงาน ชื่อของพนักงาน นามสกุลของพนักงาน แต่อยาก
ให้ชื่อของพนักงาน นามสกุลของพนักงาน ขึ้นก่อนรหัสของพนักงาน

```
SELECT FirstName, LastName, EmployeeId  
FROM Employee3
```

3 การ SELECT ข้อมูล โดยการจัดลำดับการ แสดงผล



- การแสดงผล

	ABC FirstName	ABC LastName	123 EmployeeId
1	Andrew	Adams	1
2	Nancy	Edwards	2
3	Jane	Peacock	3
4	Margaret	Park	4
5	Steve	Johnson	5
6	Michael	Mitchell	6
7	Robert	King	7
8	Laura	Callahan	8

4 การ SELECT ข้อมูล ด้วย SELECT... AS (เปลี่ยนชื่อคอลัมน์ใน การแสดงผล)



- การแสดงผลไม่จำเป็นต้องให้ชื่อหัวข้อเป็นชื่อเดียวกับชื่อคอลัมน์ทุกครั้ง เพราะตั้งชื่อคอลัมน์อาจมีการสื่อความหมายที่ไม่ชัดเจนต่อการแสดงผล

SELECT ชื่อคอลัมน์ AS “ชื่อที่ต้องการเปลี่ยน”

4 การ SELECT ข้อมูลด้วย
SELECT... AS
(เปลี่ยนชื่อคอลัมน์ใน
การแสดงผล)



- ต้องการแสดงข้อมูลของพนักงานทั้งหมด โดยให้หัวข้อแต่ละหัวข้อชื่อ “Employee Id”, “First Name”, “Last Name”

```
SELECT EmployeeId AS "Employee Id", FirstName AS "First Name",  
LastName AS "Last Name"  
FROM Employee3
```

4 การ SELECT ข้อมูล
ด้วย SELECT... AS
(เปลี่ยนชื่อคอลัมน์ใน
การแสดงผล)



- การแสดงผล

	Employee Id	First Name	Last Name
1	1	Andrew	Adams
2	2	Nancy	Edwards
3	3	Jane	Peacock
4	4	Margaret	Park
5	5	Steve	Johnson
6	6	Michael	Mitchell
7	7	Robert	King
8	8	Laura	Callahan

5 การ SELECT ข้อมูล ด้วย Distinct (แสดง ข้อมูลไม่ซ้ำ)



- เมื่อข้อมูลมีค่าที่เหมือนกันอยู่ในตารางข้อมูล เมื่อเวลาแสดงผลออกมาก็จะเห็นข้อมูลซ้ำๆ หลายบรรทัด

ซึ่งทำให้ดูยาก

โค้ด: เลือกทั้งหมด

```
SELECT DISTINCT ชื่อคอสม์_1, ชื่อคอสม์_2, ชื่อคอสม์_3
FROM ชื่อตาราง;
```

5 การ SELECT ข้อมูล ด้วย Distinct (แสดง ข้อมูลไม่ซ้ำ)



- ต้องการทราบว่าพนักงานมีตำแหน่งใดบ้าง

```
SELECT DISTINCT Title
FROM Employee3;
```

- การแสดงผล

	ABC Title
1	General Manager
2	Sales Manager
3	Sales Support Age
4	IT Manager
5	IT Staff

6 การ SELECT ข้อมูล ด้วย Order By (จัดเรียง ข้อมูล)



- เป็นคำสั่งที่ใช้เรียงข้อมูลที่ไม่เป็นระเบียบในตาราง โดยจะเรียงลำดับจากมากไปหาน้อย หรือ น้อยไปหามากก็ได้

```
โค้ด: เลือกทั้งหมด
ORDER BY ชื่อคอสม์ [DESC];
```

คำอธิบาย

ตัว DESC คือตัวกำหนดว่าจะให้มีการเรียงลำดับจากมากไปหาน้อย หรือ ASC คือตัวกำหนดว่าจะให้มีการเรียงลำดับจากน้อยไปหามาก ดังนั้นหากมีการกำหนด DESC ไว้แสดงว่าต้องการให้ข้อมูลเรียงจากมากไปหาน้อย

6 การ SELECT ข้อมูล ด้วย Order By (จัดเรียงข้อมูล)



- ตัวอย่างที่ 1 >> ต้องการทราบรายชื่อพนักงาน โดยให้แสดงรายชื่อพนักงานตามลำดับชื่อจากน้อยไปมาก

```
SELECT FirstName, LastName
FROM Employee3
ORDER BY FirstName
```

	ABC FirstName	ABC LastName
1	Andrew	Adams
2	Jane	Peacock
3	Laura	Callahan
4	Margaret	Park
5	Michael	Mitchell
6	Nancy	Edwards
7	Robert	King
8	Steve	Johnson

6 การ SELECT ข้อมูล ด้วย Order By (จัดเรียงข้อมูล)



- ตัวอย่างที่ 2 >> ต้องการทราบรหัสพนักงาน และรายชื่อของพนักงานโดยให้แสดงรหัสพนักงานจากมากไปน้อย

```
SELECT EmployeeId, FirstName, LastName
FROM Employee3
ORDER BY EmployeeId DESC
```

EmployeeId	FirstName	LastName
8	Laura	Callahan
7	Robert	King
6	Michael	Mitchell
5	Steve	Johnson
4	Margaret	Park
3	Jane	Peacock
2	Nancy	Edwards
1	Andrew	Adams

6 การ SELECT ข้อมูล ด้วย Order By (จัดเรียง ข้อมูล)



- ตัวอย่างที่ 3 >> ต้องการทราบรายชื่อของพนักงานและเมือง โดยให้แสดงจากกลุ่มเมืองเรียงลำดับจากน้อยไปมาก และในแต่ละกลุ่มเมืองก็ให้เรียงตามรายชื่อพนักงานจากน้อยไปมาก

```
SELECT FirstName, LastName, City
FROM Employee3
ORDER BY City, FirstName
```

	ABC FirstName	ABC LastName	ABC City
1	Jane	Peacock	Calgary
2	Margaret	Park	Calgary
3	Michael	Mitchell	Calgary
4	Nancy	Edwards	Calgary
5	Steve	Johnson	Calgary
6	Andrew	Adams	Edmonton
7	Laura	Callahan	Lethbridge
8	Robert	King	Lethbridge

7 การ SELECT ข้อมูล ด้วย SELECT.. WHERE (แบบมีเงื่อนไข)



- เลือกข้อมูลบางแถวได้ โดยการใช้นเงื่อนไข WHERE ซึ่งมีรูปแบบดังนี้

โค้ด: เลือกทั้งหมด

```
SELECT ชื่อคอลัมน์_1, ชื่อคอลัมน์_2, ชื่อคอลัมน์_3;  
FROM ชื่อตาราง  
WHERE เงื่อนไข;
```

- คำอธิบาย

SELECT คือการเลือกคอลัมน์ที่เรากำหนดเอาไว้

FROM คือการเลือกตารางข้อมูล

WHERE คือการกำหนดเงื่อนไข ซึ่งจะอธิบายรายละเอียดเป็นหัวข้อย่อย ๆ ไปได้ดังต่อไปนี้

7 การ SELECT ข้อมูล ด้วย SELECT.. WHERE (แบบมีเงื่อนไข)



- เงื่อนไขในการ SELECT ข้อมูล มีประเภทของเงื่อนไขหลายเงื่อนไข ได้แก่
 - 1) การใช้เครื่องหมาย Operators (=, !=, >, <, >=, <=, AND, OR, NOT)
 - 2) การเลือกข้อมูลที่ต้องตามชุดข้อมูลด้วย IN
 - 3) การเลือกข้อมูลที่อยู่ในช่วงที่ต้องการด้วย BETWEEN...AND
 - 4) การเลือกข้อมูลที่ตรงกับรูปแบบด้วย LIKE (%,_)

7 การ SELECT ข้อมูล
ด้วย SELECT.. WHERE
(แบบมีเงื่อนไข)



7.1 เงื่อนไขในการ SELECT ด้วยการการใช้เครื่องหมาย Operators
(=, !=, >, <, >=, <=, AND, OR, NOT)

เครื่องหมายที่ใช้	ความหมาย
=	เท่ากับ
!=	ไม่เท่ากับ
>	มากกว่า
<	น้อยกว่า
<=	น้อยกว่า หรือ เท่ากับ
>=	มากกว่า หรือ เท่ากับ
AND	และ
OR	หรือ
NOT	ไม่ใช่

mindphp.com

7 การ SELECT ข้อมูล ด้วย SELECT... WHERE (แบบมีเงื่อนไข)

7.1 เงื่อนไขในการ SELECT ด้วยการการใช้เครื่องหมาย Operators

(=, !=, >, <, >=, <=, AND, OR, NOT)

ตัวอย่างที่ 1 >> ต้องการทราบ invoiceld ไດ มี total มากกว่า 15

```
SELECT * FROM Invoice WHERE Total > 15
```

123 Invoiceld	123 CustomerId	ABC InvoiceDate	ABC BillingAddress	ABC BillingCity	ABC BillingState	ABC BillingCountry	ABC BillingPostalCode	123 Total
88	57	2008-01-13 00:00:00	Calle Lira, 198	Santiago	[NULL]	Chile	[NULL]	17.91
89	7	2008-01-18 00:00:00	Rotenturmstraße 4, 1011	Vienne	[NULL]	Austria	1010	18.86
96	45	2008-02-18 00:00:00	Erzsébet krt. 58.	Budapest	[NULL]	Hungary	H-1073	21.86
103	24	2008-03-20 00:00:00	162 E Superior Street	Chicago	IL	USA	60611	15.86
194	46	2009-04-27 00:00:00	3 Chatham Street	Dublin	Dublin	Ireland	[NULL]	21.86
201	25	2009-05-28 00:00:00	319 N. Frances Street	Madison	WI	USA	53703	18.86
208	4	2009-06-28 00:00:00	Ullevålsveien 14	Oslo	[NULL]	Norway	0171	15.86
299	26	2010-08-05 00:00:00	2211 W Berry Street	Fort Worth	TX	USA	76110	23.86
306	5	2010-09-05 00:00:00	Klanova 9/506	Prague	[NULL]	Czech Republic	14700	16.86
313	43	2010-10-06 00:00:00	68, Rue Jouvence	Dijon	[NULL]	France	21000	16.86
404	6	2011-11-13 00:00:00	Rilská 3174/6	Prague	[NULL]	Czech Republic	14300	25.86

7 การ SELECT ข้อมูล ด้วย SELECT.. WHERE (แบบมีเงื่อนไข)

7.1 เงื่อนไขในการ SELECT ด้วยการใช้เครื่องหมาย Operators

(=, !=, >, <, >=, <=, AND, OR, NOT)

ตัวอย่างที่ 2 >> ต้องการทราบเมืองใดในสหรัฐที่มีผลรวมมากกว่าหรือเท่ากับ 15

```
SELECT *
FROM Invoice
WHERE (BillingCountry='USA') AND Total >= 15
```

	Invoiceld	123 CustomerId	ABC InvoiceDate	ABC BillingAddress	ABC BillingCity	ABC BillingState	ABC BillingCountry	ABC BillingPostalCode	123 Total
1	103	24	2008-03-20 00:00:00	162 E Superior Street	Chicago	IL	USA	60611	15.86
2	201	25	2009-05-28 00:00:00	319 N. Frances Street	Madison	WI	USA	53703	18.86
3	299	26	2010-08-05 00:00:00	2211 W Berry Street	Fort Worth	TX	USA	76110	23.86



7 การ SELECT ข้อมูล
ด้วย SELECT.. WHERE
(แบบมีเงื่อนไข)



7.1 เงื่อนไขในการ SELECT ด้วยการการใช้เครื่องหมาย Operators

(=, !=, >, <, >=, <=, AND, OR, NOT)

ตัวอย่างที่ 3 >> ต้องการทราบว่าพนักงานที่ไม่ได้มาจากเมือง Calgary มีใครบ้าง

```
SELECT FirstName, LastName, City
FROM Employee3
WHERE City != 'Calgary'
```

ABC FirstName	ABC LastName	ABC City
Andrew	Adams	Edmonton
Robert	King	Lethbridge
Laura	Callahan	Lethbridge

7 การ SELECT ข้อมูล ด้วย SELECT.. WHERE (แบบมีเงื่อนไข)



7.2 การเลือกข้อมูลที่ต้องตามชุดข้อมูลด้วย IN

เป็นการเลือกการแสดงผลตามชุดข้อมูลที่เลือกซึ่งหากไม่ใช่ IN ก็ยังสามารถใช้งานได้ โดยใช้เครื่องหมายทาง Operator

โค้ด: เลือกทั้งหมด

```
SELECT ชื่อคอสม์น
FROM ชื่อตาราง
WHERE ชื่อคอสม์น='ข้อมูลที่ต้องการให้แสดง' ;
```

แต่หากมีใช้ IN เข้ามาช่วยจะทำให้สามารถใช้งานได้สะดวก และเข้าใจง่ายมากขึ้น โดยมีรูปแบบการเรียกใช้งานดังนี้

โค้ด: เลือกทั้งหมด

```
SELECT ชื่อคอสม์น
FROM ชื่อตาราง
WHERE ชื่อคอสม์น IN (ข้อมูลที่ต้องการให้แสดง) ;
```

7 การ SELECT ข้อมูล
ด้วย SELECT.. WHERE
(แบบมีเงื่อนไข)



7.2 การเลือกข้อมูลที่ต้องตามชุดข้อมูลด้วย IN

ตารางข้อมูลพนักงาน

EmployeeId	ABC LastName	ABC FirstName	ABC Title	123 ReportsTo	ABC BirthDate	ABC HireDate	ABC Address	ABC City
1	Adams	Andrew	General Manager	[NULL]	1962-02-18 00:00:00	2002-08-14 00:00:00	11120 Jasper Ave NW	Edmonton
2	Edwards	Nancy	Sales Manager	1	1958-12-08 00:00:00	2002-05-01 00:00:00	825 8 Ave SW	Calgary
3	Peacock	Jane	Sales Support Agent	2	1973-08-29 00:00:00	2002-04-01 00:00:00	1111 6 Ave SW	Calgary
4	Park	Margaret	Sales Support Agent	2	1947-09-19 00:00:00	2003-05-03 00:00:00	683 10 Street SW	Calgary
5	Johnson	Steve	Sales Support Agent	2	1965-03-03 00:00:00	2003-10-17 00:00:00	7727B 41 Ave	Calgary
6	Mitchell	Michael	IT Manager	1	1973-07-01 00:00:00	2003-10-17 00:00:00	5827 Bowness Road NW	Calgary
7	King	Robert	IT Staff	6	1970-05-29 00:00:00	2004-01-02 00:00:00	590 Columbia Boulevard West	Lethbridge
8	Callahan	Laura	IT Staff	6	1968-01-09 00:00:00	2004-03-04 00:00:00	923 7 ST NW	Lethbridge

7 การ SELECT ข้อมูล ด้วย SELECT.. WHERE (แบบมีเงื่อนไข)

7.2 การเลือกข้อมูลที่ต้องตามชุดข้อมูลด้วย IN

ตัวอย่างที่ 1 >> ต้องการทราบข้อมูลของพนักงานที่อยู่ใน Calgary และ Edmonton

```
SELECT *
FROM Employee3
WHERE City IN ('Calgary', 'Edmonton')
```

EmployeeId	LastName	FirstName	Title	ReportsTo	BirthDate	HireDate	Address	City
1	Adams	Andrew	General Manager	[NULL]	1962-02-18 00:00:00	2002-08-14 00:00:00	11120 Jasper Ave NW	Edmonton
2	Edwards	Nancy	Sales Manager	1	1958-12-08 00:00:00	2002-05-01 00:00:00	825 8 Ave SW	Calgary
3	Peacock	Jane	Sales Support Agent	2	1973-08-29 00:00:00	2002-04-01 00:00:00	1111 6 Ave SW	Calgary
4	Park	Margaret	Sales Support Agent	2	1947-09-19 00:00:00	2003-05-03 00:00:00	683 10 Street SW	Calgary
5	Johnson	Steve	Sales Support Agent	2	1965-03-03 00:00:00	2003-10-17 00:00:00	7727B 41 Ave	Calgary
6	Mitchell	Michael	IT Manager	1	1973-07-01 00:00:00	2003-10-17 00:00:00	5827 Bowness Road NW	Calgary

7 การ SELECT ข้อมูล ด้วย SELECT... WHERE (แบบมีเงื่อนไข)

7.2 การเลือกข้อมูลที่ต้องตามชุดข้อมูลด้วย IN

ตัวอย่างที่ 2 >> ต้องการทราบข้อมูลของพนักงานเฉพาะที่ไม่ได้อยู่ใน
Calgary

```
SELECT *
FROM Employee3
WHERE City NOT IN ('Calgary')
```

EmployeeId	LastName	FirstName	Title	ReportsTo	BirthDate	HireDate	Address	City
1	Adams	Andrew	General Manager	[NULL]	1962-02-18 00:00:00	2002-08-14 00:00:00	11120 Jasper Ave NW	Edmonton
7	King	Robert	IT Staff	6	1970-05-29 00:00:00	2004-01-02 00:00:00	590 Columbia Boulevard	Lethbridge
8	Callahan	Laura	IT Staff	6	1968-01-09 00:00:00	2004-03-04 00:00:00	923 7 ST NW	Lethbridge



7 การ SELECT ข้อมูล ด้วย SELECT.. WHERE (แบบมีเงื่อนไข)



7.3 การเลือกข้อมูลที่อยู่ในช่วงที่ต้องการด้วย BETWEEN...AND

การเลือกข้อมูลที่อยู่ในช่วงระหว่างจะมีรูปแบบดังนี้

โค้ด: เลือกทั้งหมด

```
SELECT ชื่อคอลัมน์
FROM ชื่อตาราง
WHERE ชื่อคอลัมน์ [NOT] BETWEEN <ค่าที่1> AND <ค่าที่2>;
```

7 การ SELECT ข้อมูล
ด้วย SELECT.. WHERE
(แบบมีเงื่อนไข)

7.3 การเลือกข้อมูลที่อยู่ในช่วงที่ต้องการด้วย BETWEEN...AND

ตัวอย่างที่ 1 >> แบบ BETWEEN

ตาราง Invoice

Invoiceld	CustomerId	InvoiceDate	BillingAddress	BillingCity	BillingState	BillingCountry	BillingPostalCode	Total
1	2	2007-01-01 00:00:00	Theodor-Heuss-Straße 34	Stuttgart	[NULL]	Germany	70174	1.98
2	4	2007-01-02 00:00:00	Ullevålsveien 14	Oslo	[NULL]	Norway	0171	3.96
3	8	2007-01-03 00:00:00	Grétrystraat 63	Brussels	[NULL]	Belgium	1000	5.94
4	14	2007-01-06 00:00:00	8210 111 ST NW	Edmonton	AB	Canada	T6G 2C7	8.91
5	23	2007-01-11 00:00:00	69 Salem Street	Boston	MA	USA	2113	13.86
6	37	2007-01-19 00:00:00	Berger Straße 10	Frankfurt	[NULL]	Germany	60316	0.99
7	38	2007-02-01 00:00:00	Barbarossastraße 19	Berlin	[NULL]	Germany	10779	1.98
8	40	2007-02-01 00:00:00	8, Rue Hanovre	Paris	[NULL]	France	75002	1.98
9	42	2007-02-02 00:00:00	9, Place Louis Barthou	Bordeaux	[NULL]	France	33000	3.96
10	46	2007-02-03 00:00:00	3 Chatham Street	Dublin	Dublin	Ireland	[NULL]	5.94
11	52	2007-02-06 00:00:00	202 Hoxton Street	London	[NULL]	United Kingdom	N1 5LH	8.91
12	2	2007-02-11 00:00:00	Theodor-Heuss-Straße 34	Stuttgart	[NULL]	Germany	70174	13.86

7 การ SELECT ข้อมูล
ด้วย SELECT.. WHERE
(แบบมีเงื่อนไข)



7.3 การเลือกข้อมูลที่อยู่ในช่วงที่ต้องการด้วย BETWEEN...AND

ตัวอย่างที่ 1 >> แบบ BETWEEN

>> ต้องการทราบ รหัส Invoice ที่มีผลรวมตั้งแต่ 15 และ 20

```
SELECT Invoiceld, Total
FROM Invoice
WHERE Total BETWEEN 15 and 20
```

	123 Invoiceld	123 Total
1	88	17.91
2	89	18.86
3	103	15.86
4	201	18.86
5	208	15.86
6	306	16.86
7	313	16.86

7 การ SELECT ข้อมูล
ด้วย SELECT.. WHERE
(แบบมีเงื่อนไข)



7.3 การเลือกข้อมูลที่อยู่ในช่วงที่ต้องการด้วย BETWEEN...AND

ตัวอย่างที่ 2 >> แบบ NOT BETWEEN

>> ต้องการทราบรหัส Invoice และผลรวมที่ไม่ถึง 1 และ 20

```
SELECT Invoiceld, Total
FROM Invoice
WHERE Total NOT BETWEEN 1 and 20
```

123 Invoiceld	123 Total
6	0.99
13	0.99
20	0.99
27	0.99
34	0.99
41	0.99
48	0.99
55	0.99
62	0.99
69	0.99
76	0.99
83	0.99
90	0.99
96	21.86
104	0.99
111	0.99
118	0.99
125	0.99
132	0.99
139	0.99

7 การ SELECT ข้อมูล ด้วย SELECT... WHERE (แบบมีเงื่อนไข)



7.4 การเลือกข้อมูลที่ตรงกับรูปแบบด้วย LIKE (%,_)

การ SELECT ข้อมูลยังสามารถเลือกข้อมูลที่ตรงกับคำที่เราต้องได้ การ SELECT ข้อมูลแบบนี้ใช้รูปแบบ LIKE ซึ่งตัวอย่างที่เราเห็นกันได้ชัดเจน คือ การค้นหาคำใน Google ที่เราพิมพ์แค่บางคำ แต่สามารถค้นหาเจอได้มากมาย วิธีการ SELECT ด้วย LIKE มีอยู่ 2 รูปแบบดังนี้

7.4.1 การ SELECT ข้อมูล ด้วย LIKE และ %

คำอธิบาย

% คือสัญลักษณ์ที่แทนตัวอักษรอะไรก็ได้ ก็ตัวก็ได้ เช่น เราต้องการหาคนที่คนที่มีชื่อขึ้นต้นด้วย 'สม' เวลาที่เราเขียนคำสั่งจะได้เป็น 'สม%' เป็นต้น นอกจากนั้น % ยังสามารถอยู่ส่วนท้าย หรือกลางก็ได้

```
SELECT ชื่อคอสมน์
FROM ชื่อตาราง
WHERE ชื่อคอสมน์
LIKE 'ข้อความที่ต้องการค้นหา';
```

7.4.2 การ SELECT ข้อมูล ด้วย LIKE และ _

คำอธิบาย

คือสัญลักษณ์ที่แทนตัวอักษรอะไรก็ได้ 1 ตัว ต่างกับ % ที่แทนตัวอักษรก็ตัวก็ได้

```
SELECT ชื่อคอสมน์
FROM ชื่อตาราง
WHERE ชื่อคอสมน์
LIKE 'ข้อความที่ต้องการค้นหา_';
```

7 การ SELECT ข้อมูล
ด้วย SELECT.. WHERE
(แบบมีเงื่อนไข)



7.4 การเลือกข้อมูลที่ตรงกับรูปแบบด้วย LIKE (%,_)

ตัวอย่างที่ 1 >> ต้องการทราบชื่อลูกค้าที่ขึ้นต้นด้วยตัวอักษร 'J'

```
Select * from Customer where FirstName like 'J%'
```

	123 CustomerId	ABC FirstName	ABC LastName
1	15	Jennifer	Peterson
2	17	Jack	Smith
3	23	John	Gordon
4	28	Julia	Barnett
5	34	João	Fernandes
6	48	Johannes	Van der Berg
7	51	Joakim	Johansson

7 การ SELECT ข้อมูล
ด้วย SELECT... WHERE
(แบบมีเงื่อนไข)



7.4 การเลือกข้อมูลที่ตรงกับรูปแบบด้วย LIKE (%,_)

ตัวอย่างที่ 2 >> ต้องการทราบชื่อพนักงานที่ขึ้นต้นด้วยตัวอักษร 'M' และตามด้วยตัวอักษรอะไรก็ได้ 6 ตัว

```
SELECT FirstName, LastName
FROM Employee
WHERE FirstName Like ('M_____')
```

	ABC FirstName	ABC LastName
1	Michael	Mitchell

8 การ SELECT ข้อมูล โดยการคำนวณทาง คณิตศาสตร์

- การ SELECT นอกจากการค้นหาข้อมูลแล้วนำมาแสดงผลแล้ว ยังสามารถนำข้อมูลที่ค่าเป็นตัวเลข มาทำการคำนวณก่อนจะแสดงผลได้ด้วย โดยใช้เครื่องหมายคณิตศาสตร์ เช่น +, -, *, /
- ตาราง Invoice

Invoiceld	123 CustomerId	ABC InvoiceDate	ABC BillingAddress	ABC BillingCity	ABC BillingState	ABC BillingCountry	ABC BillingPostalCode	123 Total
1	2	2007-01-01 00:00:00	Theodor-Heuss-Straße 34	Stuttgart	[NULL]	Germany	70174	1.98
2	4	2007-01-02 00:00:00	Ullevålsveien 14	Oslo	[NULL]	Norway	0171	3.96
3	8	2007-01-03 00:00:00	Grétrystraat 63	Brussels	[NULL]	Belgium	1000	5.94
4	14	2007-01-06 00:00:00	8210 111 ST NW	Edmonton	AB	Canada	T6G 2C7	8.91
5	23	2007-01-11 00:00:00	69 Salem Street	Boston	MA	USA	2113	13.86
6	37	2007-01-19 00:00:00	Berger Straße 10	Frankfurt	[NULL]	Germany	60316	0.99
7	38	2007-02-01 00:00:00	Barbarossastraße 19	Berlin	[NULL]	Germany	10779	1.98
8	40	2007-02-01 00:00:00	8, Rue Hanovre	Paris	[NULL]	France	75002	1.98
9	42	2007-02-02 00:00:00	9, Place Louis Barthou	Bordeaux	[NULL]	France	33000	3.96
10	46	2007-02-03 00:00:00	3 Chatham Street	Dublin	Dublin	Ireland	[NULL]	5.94
11	52	2007-02-06 00:00:00	202 Hoxton Street	London	[NULL]	United Kingdom	N1 5LH	8.91
12	2	2007-02-11 00:00:00	Theodor-Heuss-Straße 34	Stuttgart	[NULL]	Germany	70174	13.86
13	16	2007-02-19 00:00:00	1600 Amphitheatre Parkway	Mountain View	CA	USA	94043-1351	0.99

8 การ SELECT ข้อมูล
โดยการคำนวณทาง
คณิตศาสตร์



>> ต้องการทราบส่วนลด 5% ของผลรวมทั้งหมด โดยให้มีการแสดงผล
รหัส Invoice และผลรวมที่ลด

```
SELECT Invoiceld, (Total-(Total*5/100)) As "Total"
FROM Invoice
```

	123 Invoiceld	123 Total
1	1	1.98
2	2	3.96
3	3	5.94
4	4	8.91
5	5	13.86
6	6	0.99
7	7	1.98
8	8	1.98
9	9	3.96
10	10	5.94
11	11	8.91
12	12	13.86
13	13	0.99
14	14	1.98
15	15	1.98
16	16	3.96
17	17	5.94
18	18	8.91
19	19	13.86
20	20	0.99



	123 Invoiceld	123 Total
1	1	1.881
2	2	3.762
3	3	5.643
4	4	8.4645
5	5	13.167
6	6	0.9405
7	7	1.881
8	8	1.881
9	9	3.762
10	10	5.643
11	11	8.4645
12	12	13.167
13	13	0.9405
14	14	1.881
15	15	1.881
16	16	3.762
17	17	5.643
18	18	8.4645
19	19	13.167
20	20	0.9405

การ SELECT ข้อมูลด้วย Functions (ฟังก์ชัน)



การ SELECT ข้อมูลด้วยฟังก์ชัน (Function) ในฐานข้อมูลมีอยู่มากมายหลายคำสั่ง ซึ่งสามารถแบ่งออกเป็นฟังก์ชันใหญ่ๆได้เป็น 2 กลุ่ม

1. Aggregate Functions (ฟังก์ชันการรวม) 2. Scalar functions (ฟังก์ชันการคำนวณ)

ฟังก์ชัน	ความหมาย
AVG()	หาค่าเฉลี่ย
COUNT()	นับจำนวน
MAX()	หาค่าที่มากที่สุด
MIN()	หาค่าที่น้อยที่สุด
SUM()	หาผลรวม

ฟังก์ชัน	ความหมาย	ตัวอย่าง
ABS(x)	ทำให้เป็นจำนวนสัมบูรณ์	ABS(-78) = 78
MOD(x,y)	ตัวเศษของผลหาร	MOD(11,3) = 2
ROUND(x,y)	การปัดเศษ หากมากกว่า 5 ขึ้นไปให้ปัดขึ้นและตัว y คือ ตำแหน่งทศนิยมที่ต้องแสดง	ROUND(4.67,1) = 4.7 หรือ ROUND(6.214,2) = 6.21
TRUNC(x,y)	คล้ายกับ ROUND แต่ให้ตัดเศษทิ้งไปเลย โดยไม่มีเงื่อนไขใดๆ	TRUNC(4.67,1) = 4.6

9 การ SELECT ข้อมูล
ด้วย
Aggregate Functions
(ฟังก์ชันการรวม)



Aggregate Functions (ฟังก์ชันการรวม)

ฟังก์ชัน	ความหมาย
AVG()	หาค่าเฉลี่ย
COUNT()	นับจำนวน
MAX()	หาค่าที่มากที่สุด
MIN()	หาค่าที่น้อยที่สุด
SUM()	หาผลรวม

9 การ SELECT ข้อมูล
ด้วย
Aggregate Functions
(ฟังก์ชันการรวม)



ตัวอย่าง >> ต้องการทราบผลรวมโดยเฉลี่ย ผลรวมที่สูงที่สุด และ
ผลรวมที่ต่ำที่สุด

```
SELECT AVG(Total), MAX(Total), MIN(Total)
FROM Invoice
```

	123 AVG(Total) ↑↓	123 MAX(Total) ↑↓	123 MIN(Total) ↑↓
1	5.6519417476	25.86	0.99

10 การ SELECT ข้อมูล ด้วย Scalar functions (ฟังก์ชันการคำนวณ)



ตัวอย่าง >> ต้องการแสดงรหัส Invoice และผลรวมโดยที่ไม่มีจุดทศนิยม และให้เรียงผลรวมจากสูงที่สุดไปหาผลรวมน้อยที่สุด

```
SELECT Invoiceld, round(Total,-1) AS "Total"
FROM Invoice
ORDER BY Total DESC
```

คำสั่ง round หากเป็นเลขติดลบ เช่น -2 ,-1 หรือ 0 จะแสดงจำนวนเต็ม หากเป็นเลขบวก จะแสดงทศนิยม ตามจำนวนเลขบวกนั้น เช่น 2 แสดงทศนิยม 2 ตำแหน่ง

select round(5.6621,2) =5.66
select round(5.6621,-2)=6
select round(5.6621,0)=6

	123 Invoiceld	123 Total
1	404	26
2	299	24
3	96	22
4	194	22
5	89	19
6	201	19
7	88	18
8	306	17
9	313	17
10	103	16

- ใช้เพื่อเลือกข้อความ หรือตัวอักษร

11 การ SELECT ข้อมูล ด้วย String functions (ฟังก์ชันตัวอักษร)



ฟังก์ชัน	ความหมาย	ตัวอย่าง
substr(x,y,z)	แสดงข้อความจากข้อความ x เริ่มต้นตำแหน่งที่ y มา z ตัวอักษร x คือข้อความ y คือ ตำแหน่งที่เริ่มต้นในข้อความ x z คือ จำนวน z ตัวอักษร หาก + จะดึงจากซ้ายสุด มา z ตัว หาก - จะดึงจากขวาสุด มา z ตัว	select substr("T6G 2C7",-2)= C7 select substr("T6G 2C7",1,2)=T6 select substr("T6G 2C7",2,1)=6 select substr("T6G 2C7",3,4)=G 2C
upper(x)	แสดงข้อความ x เป็นตัวพิมพ์ใหญ่	select UPPER("tar") =TAR
lower(x)	แสดงข้อความ x เป็นตัวพิมพ์เล็ก	select LOWER("TAR") =tar
length(x)	แสดงจำนวนตัวอักษรของข้อความ x	select LENGTH ("enough")=6

11 การ SELECT ข้อมูล ด้วย String functions (ฟังก์ชันตัวอักษร)



- ตัวอย่าง >> ต้องการทราบรหัสไปรษณีย์ โดยให้แสดงเพียง 2 ตัวท้ายของรหัสไปรษณีย์ รายชื่อประเทศให้แสดงเป็นตัวพิมพ์ใหญ่ทั้งหมด

```
SELECT substr(BillingPostalCode,-2) as postalcode,  
upper(BillingCountry) FROM Invoice
```

	ABC postalcode	ABC upper(BillingCountry)
1	74	GERMANY
2	71	NORWAY
3	00	BELGIUM
4	C7	CANADA
5	13	USA
6	16	GERMANY
7	79	GERMANY
8	02	FRANCE

12 การ SELECT ข้อมูล ด้วย Date and time



`SELECT DATE('now', 'start of month', '+1 month', '-1 day');`

No	Format
1	YYYY-MM-DD
2	YYYY-MM-DD HH:MM
3	YYYY-MM-DD HH:MM:SS
4	YYYY-MM-DD HH:MM:SS.SSS
5	YYYY-MM-DDTHH:MM
6	YYYY-MM-DDTHH:MM:SS
7	YYYY-MM-DDTHH:MM:SS.SSS
8	HH:MM
9	HH:MM:SS
10	HH:MM:SS.SSS
11	now
12	DDDDDDDDDD

12 การ SELECT ข้อมูล ด้วย Date and time



No	Modifier	Description
1	NNN days	Add \pm NNN days to a date and time
2	NNN hours	Add \pm NNN hours to a date and time
3	NNN minutes	Add \pm NNN minutes to a date and time
4	NNN.NNNN seconds	Add \pm NNN seconds to/from a date and time
5	NNN months	Add \pm NNN months to a date and time
6	NNN years	Add \pm NNN year to a date and time
7	start of month	Backward to the beginning of the month
8	start of year	backward to the beginning of the year
9	start of day	Backward to the beginning of the day
10	weekday N	Advance a date forward to the next date where the weekday number is N
11	unixepoch	Unix time
12	localtime	Return local time
13	utc	Return UTC time

12 การ SELECT ข้อมูล ด้วย Date and time



SELECT DATETIME('now');

	ABC DATETIME('now') ↑↓
1	2020-08-15 04:43:39

SELECT DATETIME('now','localtime');

	ABC DATETIME('now','localtime') ↑↓
1	2020-08-15 11:45:18

SELECT DATE('2018-11-01','-1 day'); = 2018-10-31

SELECT DATE('2018-11-01','-1 month'); = 2018-10-01

SELECT DATE('2018-11-01','-1 year'); = 2017-11-01

SELECT DATE('2018-11-01','+1 day'); = 2018-11-02

SELECT DATE('2018-11-01','+1 month'); = 2018-12-01

SELECT DATE('2018-11-01','+1 year'); = 2019-11-01

13 การ SELECT ข้อมูล ด้วย SELECT ซ้อน SELECT



- เป็นการ SELECT ข้อมูลแบบซับซ้อนมากขึ้น หรือ ที่เรียกว่า SELECT ซ้อน SELECT โดยจะมี Main Query เป็นตัว SELECT หลัก แล้วมี Subquery อยู่ภายใต้ Main Query

*** การทำงานของ SELECT ซ้อน SELECT จะทำการประมวลผลจาก SELECT ย่อยก่อน

```
SELECT ชื่อคอสม์
FROM ชื่อตารางข้อมูล
WHERE เงื่อนไข
      (SELECT ชื่อคอสม์
       FROM ชื่อตารางข้อมูล
       WHERE เงื่อนไข);
```

13 การ SELECT ข้อมูล ด้วย SELECT ซ้อน SELECT



- ตัวอย่าง >> ต้องการทราบรหัสลูกค้าที่อยู่เมืองเดียวกับรหัสลูกค้าเท่ากับ 16
ตัวอย่างการใช้งาน (แบบยังไม่ซ้อน SELECT)

	123 Invoiceld	123 CustomerId	ABC BillingCity
1	1	2	Stuttgart
2	2	4	Oslo
3	3	8	Brussels
4	4	14	Edmonton
5	5	23	Boston
6	6	37	Frankfurt
7	7	38	Berlin
8	8	40	Paris
9	9	42	Bordeaux
10	10	46	Dublin
11	11	52	London
12	12	2	Stuttgart
13	13	16	Mountain View

13 การ SELECT ข้อมูล ด้วย SELECT ซ้อน SELECT



- ตัวอย่าง >> ต้องการทราบรหัสลูกค้าที่อยู่เมืองเดียวกับรหัสลูกค้าเท่ากับ 16 ตัวอย่างการใช้งาน (แบบยังไม่ซ้อน SELECT)
 - ต้องทราบเมืองของลูกค้าที่มีรหัส 16 ก่อน - เมื่อทราบผลแล้วว่าลูกค้าที่มีรหัส 16 อยู่เมือง Mountain View

```
SELECT
CustomerId,InvoiceId,
BillingCity FROM Invoice
where CustomerId = 16
```

	123 CustomerId	123 InvoiceId	ABC BillingCity
1	16	13	Mountain View
2	16	134	Mountain View
3	16	145	Mountain View
4	16	200	Mountain View
5	16	329	Mountain View
6	16	352	Mountain View
7	16	374	Mountain View

13 การ SELECT ข้อมูล ด้วย SELECT ซ้อน SELECT



- ตัวอย่าง >> ต้องการทราบรหัสลูกค้าที่อยู่เมืองเดียวกับรหัสลูกค้าเท่ากับ 16
ตัวอย่างการใช้งาน (แบบยังไม่ซ้อน SELECT)

```
SELECT BillingCity FROM Invoice
WHERE CustomerId = 16
```

```
SELECT CustomerId FROM Invoice
WHERE BillingCity = 'Mountain View'
```

- นำมาเขียนคำสั่งให้อยู่ในรูป SELECT ซ้อน SELECT ได้ดังนี้

```
SELECT CustomerId
FROM Invoice
WHERE BillingCity =( SELECT BillingCity
FROM Invoice WHERE CustomerId = 16)
```

	123 CustomerId	
1	16	↗
2	20	↗
3	20	↗
4	16	↗
5	16	↗
6	20	↗
7	16	↗
8	20	↗
9	16	↗
10	20	↗
11	16	↗
12	20	↗
13	16	↗
14	20	↗

14 การ SELECT ข้อมูล ด้วย SELECT ... GROUP BY (แบบจัดกลุ่ม)



- คือการจัดกลุ่มค่าที่ซ้ำกันให้แสดงแค่ค่าเดียว หรือการจัดกลุ่มของค่าซ้ำกันให้รวมกลุ่มกัน โดยจะมีการเรียงลำดับตามตัวอักษร เพื่อใช้ในการวิเคราะห์หรือทำรายงานสรุปข้อมูล
- อธิบายไค้ด GROUP BY
 - table_name คือ ชื่อตารางข้อมูล
 - column_name คือ ชื่อคอลัมน์

```
SELECT column_name1, column_name2
FROM table_name
GROUP BY column_name1, column_name2;
```

14 การ SELECT ข้อมูล ด้วย SELECT ... GROUP BY (แบบจัดกลุ่ม)

- ตัวอย่างข้อมูลในตาราง INVOICE

123 Invoiceld	123 CustomerId	ABC InvoiceDate	ABC BillingAddress	ABC BillingCity	ABC BillingState	ABC BillingCountry	ABC BillingPostalCode	123 Total
1	2	2007-01-01 00:00:00	Theodor-Heuss-Straße 34	Stuttgart	[NULL]	Germany	70174	1.98
2	4	2007-01-02 00:00:00	Ullevålsveien 14	Oslo	[NULL]	Norway	0171	3.96
3	8	2007-01-03 00:00:00	Grétrystraat 63	Brussels	[NULL]	Belgium	1000	5.94
4	14	2007-01-06 00:00:00	8210 111 ST NW	Edmonton	AB	Canada	T6G 2C7	8.91
5	23	2007-01-11 00:00:00	69 Salem Street	Boston	MA	USA	2113	13.86
6	37	2007-01-19 00:00:00	Berger Straße 10	Frankfurt	[NULL]	Germany	60316	0.99
7	38	2007-02-01 00:00:00	Barbarossastraße 19	Berlin	[NULL]	Germany	10779	1.98
8	40	2007-02-01 00:00:00	8, Rue Hanovre	Paris	[NULL]	France	75002	1.98



14 การ SELECT ข้อมูล ด้วย SELECT ... GROUP BY (แบบจัดกลุ่ม)



ตัวอย่าง >> การดึงข้อมูลจากราย INVOICE เพื่อแสดงคอลัมน์ BillingCountry รวม Total ด้วย GROUP BY

```
SELECT BillingCountry, SUM(Total)
FROM Invoice
GROUP BY BillingCountry
```

	ABC BillingCountry	123 SUM(Total)
1	Argentina	37.62
2	Australia	37.62
3	Austria	42.62
4	Belgium	37.62
5	Brazil	190.1
6	Canada	303.96

- จากผลลัพธ์จะเห็นได้ว่า การดึงข้อมูลจากราย INVOICE เพื่อแสดงคอลัมน์ BillingCountry จะมีการจัดกลุ่มค่าซ้ำให้รวมอยู่เป็นกลุ่มเดียวกันและมีการเรียงลำดับตัว
- อักษรนำหน้าภาษาอังกฤษตัวที่มาก่อน ซึ่งหลังเรียงตามลำดับ รวมทั้งผลรวมของ Total ตามลำดับของคอลัมน์ BillingCountry ด้วย

15 การ SELECT ข้อมูล
ด้วย HAVING
(เงื่อนไขของการ
จัดกลุ่ม) และ WHERE



```
SELECT column_1, column_2, aggregate_function (column_3)  
FROM table  
GROUP BY column_1, column_2  
HAVING search_condition;
```


15 ตัวอย่าง การ SELECT ข้อมูลด้วย HAVING

ตัวอย่าง ตารางข้อมูล track

▼ **Track**

▼ **Columns**

- 123 TrackId (INTEGER)
- ABC Name (NVARCHAR(200))
- 123 AlbumId (INTEGER)
- 123 MediaTypeId (INTEGER)
- 123 GenreId (INTEGER)
- ABC Composer (NVARCHAR(220))
- 123 Milliseconds (INTEGER)
- 123 Bytes (INTEGER)
- 123 UnitPrice (NUMERIC(10,2))

Track | Enter a SQL expression to filter results (use Ctrl+Space)

	123 TrackId	ABC Name	123 AlbumId	123 MediaTypeId	123 GenreId	ABC Composer	123 Milliseconds	123 Bytes	123 UnitPrice
1	1	For Those About To Rock (We Salute You)	1	1	1	Angus Young, Malc	343,719	11,170,334	0.99
2	2	Balls to the Wall	2	2	1	[NULL]	342,562	5,510,424	0.99
3	3	Fast As a Shark	3	2	1	F. Baltes, S. Kaufma	230,619	3,990,994	0.99
4	4	Restless and Wild	3	2	1	F. Baltes, R.A. Smith	252,051	4,331,779	0.99
5	5	Princess of the Dawn	3	2	1	Deaffy & R.A. Smith	375,418	6,290,521	0.99

ค้นหา : ให้แสดง albumid และจำนวน Trackid ของแต่ละ AlbumId โดยเลือกแสดงเฉพาะที่จำนวน trackid มีค่าตั้งแต่ 18-20 และให้เรียงตาม albumid

15 ตัวอย่าง
การ SELECT ข้อมูลด้วย
HAVING (ตามด้วยเงื่อนไขในการค้นหา)



- ตัวอย่าง

```
SELECT
albumid,
COUNT(trackid)
FROM
track
GROUP BY
albumid
HAVING
COUNT(trackid) BETWEEN 18 AND 20
ORDER BY albumid;
```

Track ☒

SELECT albumid, COUNT(trackid) FROM trac | Er

	123 AlbumId	123 COUNT(trackid)
1	21	18
2	37	20
3	54	20
4	55	20
5	72	18
6	102	18
7	115	20
8	145	18
9	146	18
10	202	18
11	211	18
12	213	18
13	221	20
14	227	19
15	248	19
16	258	19

NULLS in SQL



- เมื่อใดก็ตามที่ข้อมูลไม่มีค่าใดๆ สามารถระบุเป็นค่าให้เป็น NULL ได้
- หมายถึง :
 - ไม่มีค่า
 - มีค่า แต่เราไม่ทราบ
 - ค่านั้นไม่สามารถนำมาใช้ได้
 - อื่น ๆ
- โครงสร้างจะกำหนดไว้สำหรับแต่ละ attribute ให้เป็น null หรือไม่
- SQL จัดการกับตารางที่มีค่า NULL อย่างไร ?

Null Values



ทดสอบค่า null โดยใช้ IS NULL

{ column | expression } IS NULL;

หา tracks ที่ composers เป็น NULL

**SELECT Name, Composer FROM track
WHERE Composer IS NULL
ORDER BY Name;**

Grid	ABC Name	ABC Composer
1	"?"	[NULL]
2	#9 Dream	[NULL]
3	(I Can't Help) F	[NULL]
4	...And Found	[NULL]
5	...In Translation	[NULL]
6	.07%	[NULL]
7	100% Hardcore	[NULL]

มี 978 แถว

ทดสอบไม่เป็นค่า null โดยใช้ IS NOT NULL

{ column | expression } IS NOT NULL;

หา tracks ที่ composers ไม่เป็น NULL

**SELECT Name, Composer FROM track
WHERE Composer IS NOT NULL
ORDER BY Name;**

Grid	ABC Name	ABC Composer
2500	You're Gonna Break My Hai	Sykes
2501	You're My Best Friend	Deacon, John
2502	You're What's Happening (I	Allen Story/George Gord
2503	You've Been A Long Time C	Brian Holland/Eddie Holl
2504	You've Got It	Mick Hucknall and Lamc
2505	Your Blue Room	Brian Eno, Bono, Adam C
2506	Your Mirror	Mick Hucknall

มี 2525 แถว

ประเภทการเชื่อมโยง ข้อมูลด้วยการ Join Tables



- การใช้คำสั่ง SELECT ทั่วๆ ไปเป็นการดึงข้อมูลจากตารางเดียว แต่ในการปฏิบัติงานจริงแล้ว อาจต้องมีการดึงข้อมูลจากหลายตาราง โดยอาศัยคอลัมน์ที่ตรงกันของ 2 ตารางเป็นตัวเชื่อม (Foreign Key) วิธีการนี้เรียกว่า "Join Table" ซึ่งการ Join Table แบ่งออกได้เป็น 2 ประเภท ได้แก่
 1. Inner Join ได้แก่ Equi join และ Self join
 2. Outer Join ได้แก่ Left Outer Join และ Right Outer Join

ประเภทการเชื่อมโยง ข้อมูลด้วยการ Join Tables



- ตัวอย่างการใช้งาน (แบบยังไม่ Join Table)

ตาราง Customer

123 CustomerId	ABC FirstName	ABC LastName
1	Luís	Gonçalves
2	Leonie	Köhler
3	François	Tremblay
4	Bjørn	Hansen
5	František	Wichterlová
6	Helena	Holý
7	Astrid	Gruber
8	Daan	Peeters
9	Kara	Nielsen

ตาราง Invoice

123 InvoiceId	123 CustomerId	ABC InvoiceDate	123 Total
1	2	2007-01-01 00:00:00	1.98
2	4	2007-01-02 00:00:00	3.96
3	8	2007-01-03 00:00:00	5.94
4	14	2007-01-06 00:00:00	8.91
5	23	2007-01-11 00:00:00	13.86
6	37	2007-01-19 00:00:00	0.99
7	38	2007-02-01 00:00:00	1.98
8	40	2007-02-01 00:00:00	1.98
9	42	2007-02-02 00:00:00	3.96
10	46	2007-02-03 00:00:00	5.94
11	52	2007-02-06 00:00:00	8.91
12	2	2007-02-11 00:00:00	13.86
13	16	2007-02-19 00:00:00	0.99
14	17	2007-03-04 00:00:00	1.98



ประเภทการเชื่อมโยง ข้อมูลด้วยการ Join Tables



- ตัวอย่าง >> ต้องการทราบรหัสลูกค้าที่ทำการสั่งซื้อสินค้าในรหัสการสั่งซื้อ 05

```
SELECT CustomerId, Invoiceld
FROM Invoice WHERE Invoiceld = 5
```

	123 CustomerId	123 Invoiceld
1	23	5

- >> จากการแสดงผลข้างต้นต้องการหาชื่อลูกค้าที่มีรหัสดังกล่าว

```
SELECT CustomerId, FirstName,
LastName FROM Customer
WHERE CustomerId = 23
```

123 CustomerId	ABC FirstName	ABC LastName
23	John	Gordon

- *** จะเห็นว่าตัวอย่างข้างต้นเป็นการ SELECT ทีละตาราง และต้องทำการ SELECT ถึง 2 ครั้งกว่าจะได้ข้อมูลที่ต้องการ

ประเภทการเชื่อมโยง ข้อมูลด้วยการ Join Tables



- ตัวอย่างการใช้งาน (แบบ Join Table) >> ดั่งนั้นการ Join Table สามารถลดขั้นตอนได้ดังนี้

```
SELECT Invoice.InvoiceId, Invoice.CustomerId, Customer.FirstName,
Customer.LastName
FROM Invoice, Customer
WHERE Customer.CustomerId = Invoice.CustomerId
AND Invoice.InvoiceId = 5
```

123 InvoiceId	123 CustomerId	ABC FirstName	ABC LastName
5	23	John	Gordon

JOIN TABLE แบบ Inner Join



- เป็นการ Join Table ตั้งแต่ 2 ตารางขึ้นไปที่มี Key หรือ คอลัมน์ตรงกันมาแสดง
แบ่ง Inner Join ได้เป็น 2 ประเภทคือ Equi Join และ Self Join

JOIN TABLE แบบ Inner Join



- รูปแบบการใช้งาน INNER JOIN

```
SELECT ชื่อคอลัมน์1, ชื่อคอลัมน์2, ชื่อคอลัมน์3,...
FROM ชื่อตาราง1 INNER JOIN ชื่อตาราง2
ON ชื่อตาราง1.ชื่อคอลัมน์ = ชื่อตาราง2.ชื่อคอลัมน์;
```

JOIN TABLE แบบ Inner Join



- รูปแบบการใช้งาน INNER JOIN

```
SELECT ชื่อคอลัมน์1, ชื่อคอลัมน์2, ชื่อคอลัมน์3,...
FROM ชื่อตาราง1 INNER JOIN ชื่อตาราง2
ON ชื่อตาราง1.ชื่อคอลัมน์ = ชื่อตาราง2.ชื่อคอลัมน์;
```

Customer

123 CustomerId	ABC FirstName	ABC LastName	ABC Phone
1	Luís	Gonçalves	+55 (12) 3923-5555
2	Leonie	Köhler	+49 0711 2842222
3	François	Tremblay	+1 (514) 721-4711
4	Bjørn	Hansen	+47 22 44 22 22
5	František	Wichterlová	+420 2 4172 5555
6	Helena	Holý	+420 2 4177 0449

Invoice

123 InvoiceId	123 CustomerId	ABC InvoiceDate	123 Total
1	2	2007-01-01 00:00:00	1.98
2	4	2007-01-02 00:00:00	3.96
3	8	2007-01-03 00:00:00	5.94
4	14	2007-01-06 00:00:00	8.91
5	23	2007-01-11 00:00:00	13.86
6	37	2007-01-19 00:00:00	0.99

JOIN TABLE แบบ Inner Join



```
SELECT ชื่อคอลัมน์1, ชื่อคอลัมน์2, ชื่อคอลัมน์3,...
FROM ชื่อตาราง1 INNER JOIN ชื่อตาราง2
ON ชื่อตาราง1.ชื่อคอลัมน์ = ชื่อตาราง2.ชื่อคอลัมน์;
```

- ตัวอย่างการใช้งาน INNER JOIN

>> ต้องการทราบรหัสInvoice รายชื่อลูกค้า และเบอร์โทรศัพท์ต่อ

```
SELECT Invoice.InvoiceId, Customer.CustomerId,
Customer.FirstName, Customer.LastName, Customer.Phone
FROM Invoice INNER JOIN Customer
ON Invoice.CustomerId = Customer.CustomerId
ORDER BY Invoice.CustomerId
```

Invoiceld	CustomerId	FirstName	LastName	Phone
98	1	Luís	Gonçalves	+55 (12) 3923-5555
121	1	Luís	Gonçalves	+55 (12) 3923-5555
143	1	Luís	Gonçalves	+55 (12) 3923-5555
195	1	Luís	Gonçalves	+55 (12) 3923-5555
316	1	Luís	Gonçalves	+55 (12) 3923-5555
327	1	Luís	Gonçalves	+55 (12) 3923-5555
382	1	Luís	Gonçalves	+55 (12) 3923-5555
1	2	Leonie	Köhler	+49 0711 2842222
12	2	Leonie	Köhler	+49 0711 2842222
67	2	Leonie	Köhler	+49 0711 2842222

JOIN TABLE แบบ Inner Join (แบบ Equi Join)



- เป็นการ Join Table ตั้งแต่ 2 ตารางขึ้นไปโดยมีเงื่อนไขว่าข้อมูลของทั้ง 2 ตารางจะต้องตรงกัน มีค่าเท่ากัน หรือ ใช้เครื่องหมาย "=" ในการ Join ตาราง
- รูปแบบคำสั่ง

```
SELECT ชื่อตารางa.ชื่อคอลัมน์, ชื่อตารางb.ชื่อคอลัมน์
FROM ชื่อตารางa JOIN ชื่อตารางb
ON ชื่อตารางa.ชื่อคอลัมน์ = ชื่อตารางb.ชื่อคอลัมน์;
```

JOIN TABLE แบบ Inner Join (แบบ Equi Join)



- ตัวอย่าง >>

```
SELECT ชื่อตารางa.ชื่อคอลัมน์, ชื่อตารางb.ชื่อคอลัมน์
FROM ชื่อตารางa JOIN ชื่อตารางb
ON ชื่อตารางa.ชื่อคอลัมน์ = ชื่อตารางb.ชื่อคอลัมน์;
```

Customer

123 CustomerId	ABC FirstName	ABC LastName	ABC Phone
1	Luís	Gonçalves	+55 (12) 3923-5555
2	Leonie	Köhler	+49 0711 2842222
3	François	Tremblay	+1 (514) 721-4711
4	Bjørn	Hansen	+47 22 44 22 22
5	František	Wichterlová	+420 2 4172 5555
6	Helena	Holý	+420 2 4177 0449

Invoice

123 InvoiceId	123 CustomerId	ABC InvoiceDate	123 Total
1	2	2007-01-01 00:00:00	1.98
2	4	2007-01-02 00:00:00	3.96
3	8	2007-01-03 00:00:00	5.94
4	14	2007-01-06 00:00:00	8.91
5	23	2007-01-11 00:00:00	13.86
6	37	2007-01-19 00:00:00	0.99



JOIN TABLE แบบ Inner Join (แบบ Equi Join)



- ตัวอย่าง >> ต้องการทราบชื่อลูกค้า และจำนวนเงินที่สั่งซื้อสินค้า

```
SELECT Customer.FirstName, Invoice.total
FROM Customer JOIN Invoice
ON Customer.CustomerId = Invoice.CustomerId
```

ABC FirstName	123 Total
Leonie	1.98
Bjørn	3.96
Daan	5.94
Mark	8.91
John	13.86
Fynn	0.99
Niklas	1.98

```
SELECT ชื่อตารางa.ชื่อคอลัมน์, ชื่อตารางb.ชื่อคอลัมน์
FROM ชื่อตารางa JOIN ชื่อตารางb
ON ชื่อตารางa.ชื่อคอลัมน์ = ชื่อตารางb.ชื่อคอลัมน์;
```

JOIN TABLE แบบ Inner Join (แบบ Self Join)



- เป็นการ Join กันเองภายในตารางเดียว เมื่อต้องการเปรียบเทียบข้อมูลจากคอลัมน์เดียวกันในตารางเดียวกัน แต่เนื่องจากต้องการข้อมูลในตารางเดียวกัน จึงต้องใช้นามแฝง (Aliases Name) เพื่อความถูกต้องในการดึงข้อมูล
- รูปแบบการใช้งาน

```
SELECT ชื่อคอลัมน์
FROM ชื่อตาราง AS ชื่อนามแฝง;
```


JOIN TABLE แบบ Inner Join (แบบ Self Join)



```
SELECT ชื่อคอลัมน์
FROM ชื่อตาราง AS ชื่อนามแฝง;
```

```
SELECT m.firstname || ' ' || m.lastname AS 'Manager',
       e.firstname || ' ' || e.lastname AS 'Direct report'
FROM employee3 e
INNER JOIN employee3 m ON m.employeeid = e.reportsto
ORDER BY manager
```

Result

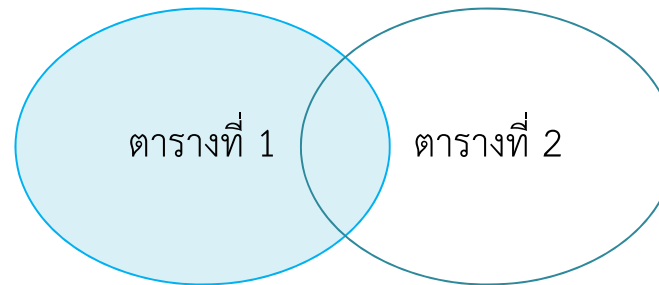
SELECT m.firstname || ' ' || m.lastnam

	ABC Manager	ABC Direct report
1	Andrew Adams	Nancy Edwards
2	Andrew Adams	Michael Mitchell
3	Michael Mitchell	Robert King
4	Michael Mitchell	Laura Callahan
5	Nancy Edwards	Jane Peacock
6	Nancy Edwards	Margaret Park
7	Nancy Edwards	Steve Johnson

JOIN TABLE แบบ Left Outer Join



- เป็นการ Join Table โดยยึดตารางทางซ้ายมือเป็นหลักดังภาพ



```
SELECT ชื่อคอลัมน์1, ชื่อคอลัมน์2, ชื่อคอลัมน์3, ..
FROM ชื่อตาราง1
LEFT OUTER JOIN ชื่อตาราง2
ON ชื่อตาราง1.ชื่อคอลัมน์=ชื่อตาราง2.ชื่อคอลัมน์;
```

JOIN TABLE แบบ Left Outer Join



- เป็นการ Join Table โดยยึดตารางทางซ้ายมือเป็นหลักดังภาพ

```
SELECT ชื่อคอสม์น1, ชื่อคอสม์น2, ชื่อคอสม์น3, ..
FROM ชื่อตาราง1
LEFT OUTER JOIN ชื่อตาราง2
ON ชื่อตาราง1.ชื่อคอสม์น=ชื่อตาราง2.ชื่อคอสม์น;
```

Customer

123 CustomerId	ABC FirstName	ABC LastName	ABC Phone
1	Luís	Gonçalves	+55 (12) 3923-5555
2	Leonie	Köhler	+49 0711 2842222
3	François	Tremblay	+1 (514) 721-4711
4	Bjørn	Hansen	+47 22 44 22 22
5	František	Wichterlová	+420 2 4172 5555
6	Helena	Holý	+420 2 4177 0449

Invoice

123 InvoiceId	123 CustomerId	ABC InvoiceDate	123 Total
1	2	2007-01-01 00:00:00	1.98
2	4	2007-01-02 00:00:00	3.96
3	8	2007-01-03 00:00:00	5.94
4	14	2007-01-06 00:00:00	8.91
5	23	2007-01-11 00:00:00	13.86
6	37	2007-01-19 00:00:00	0.99



JOIN TABLE แบบ Left Outer Join



- >> ต้องการทราบชื่อลูกค้า และ รหัส Invoice

```
SELECT Customer.FirstName, Invoice.InvoiceId
FROM Customer
LEFT OUTER JOIN Invoice
ON Customer.CustomerId = Invoice.CustomerId
ORDER By Customer.FirstName
```

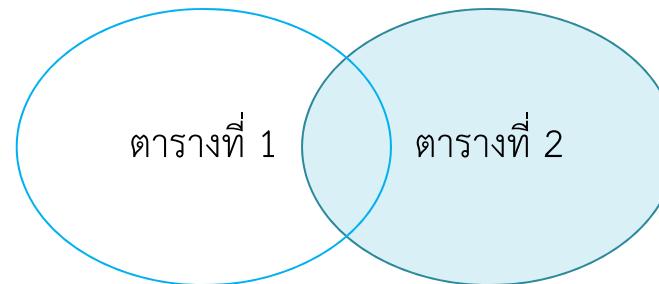
```
SELECT ชื่อคอลัมน์1, ชื่อคอลัมน์2, ชื่อคอลัมน์3, ..
FROM ชื่อตาราง1
LEFT OUTER JOIN ชื่อตาราง2
ON ชื่อตาราง1.ชื่อคอลัมน์=ชื่อตาราง2.ชื่อคอลัมน์;
```

ABC FirstName	123 InvoiceId
Aaron	50
Aaron	61
Aaron	116
Aaron	245
Aaron	268
Aaron	290
Aaron	342
Alexandre	57
Alexandre	68
Alexandre	123

JOIN TABLE แบบ Right Outer Join



- เป็นการ Join Table โดยยึดตารางทางขวามือเป็นหลักดังภาพ



```
SELECT ชื่อคอลัมน์1, ชื่อคอลัมน์2, ชื่อคอลัมน์3, ...
FROM ชื่อตาราง1
RIGHT OUTER JOIN ชื่อตาราง2
ON ชื่อตาราง1.ชื่อคอลัมน์=ชื่อตาราง2.ชื่อคอลัมน์;
```

JOIN TABLE แบบ Right Outer Join



- ตัวอย่าง

```
SELECT ชื่อคอลัมน์1, ชื่อคอลัมน์2, ชื่อคอลัมน์3, ...
FROM ชื่อตาราง1
RIGHT OUTER JOIN ชื่อตาราง2
ON ชื่อตาราง1.ชื่อคอลัมน์=ชื่อตาราง2.ชื่อคอลัมน์;
```

Customer

123 CustomerId	ABC FirstName	ABC LastName	ABC Phone
1	Luís	Gonçalves	+55 (12) 3923-5555
2	Leonie	Köhler	+49 0711 2842222
3	François	Tremblay	+1 (514) 721-4711
4	Bjørn	Hansen	+47 22 44 22 22
5	František	Wichterlová	+420 2 4172 5555
6	Helena	Holý	+420 2 4177 0449

Invoice

123 InvoiceId	123 CustomerId	ABC InvoiceDate	123 Total
1	2	2007-01-01 00:00:00	1.98
2	4	2007-01-02 00:00:00	3.96
3	8	2007-01-03 00:00:00	5.94
4	14	2007-01-06 00:00:00	8.91
5	23	2007-01-11 00:00:00	13.86
6	37	2007-01-19 00:00:00	0.99



- ตัวอย่าง >> ต้องการทราบชื่อลูกค้า และ รหัส Invoice

SQLite ไม่สนับสนุน RIGHT OUTER JOIN

```
SELECT Customer.FirstName, Invoice.InvoiceId
FROM Customer
RIGHT OUTER JOIN Invoice
ON Customer.CustomerId = Invoice.CustomerId
ORDER By Customer.FirstName
```

```
SELECT ชื่อคอลัมน์1, ชื่อคอลัมน์2, ชื่อคอลัมน์3, ...
FROM ชื่อตาราง1
RIGHT OUTER JOIN ชื่อตาราง2
ON ชื่อตาราง1.ชื่อคอลัมน์=ชื่อตาราง2.ชื่อคอลัมน์;
```

JOIN TABLE แบบ Right Outer Join



คำสั่งประยุกต์ Advanced SQLizing



1. UNION , UNION ALL
2. INTERSECT
3. EXCEPT หรือ SUBTRACT
4. Quantifiers
5. Aggregation v.s. subqueries

คำสั่งประยুক্ত UNION รวมข้อมูลตั้งแต่ 2 table ขึ้นไป



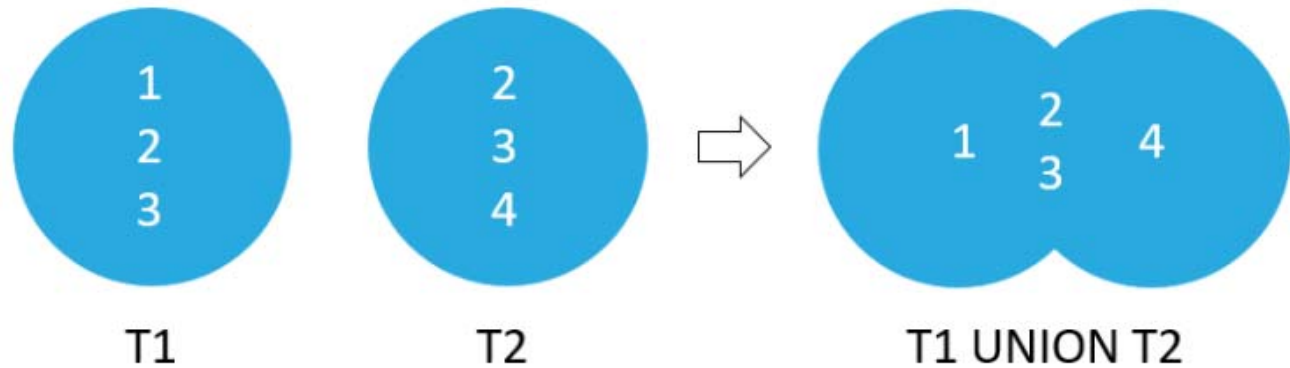
ในบางครั้งฐานข้อมูล หรือ database ของเรา อาจจะมี table ที่เก็บข้อมูล คล้ายๆกัน ผู้ใช้งานอาจต้องการรวมข้อมูลทั้ง 2 table ในรูปแบบ [SQL JOIN statement](#) หรือ การสร้างความสัมพันธ์ระหว่าง column แต่สำหรับข้อมูลที่มี column เหมือนกันรวมถึง data type แบบเดียวกัน เราสามารถนำข้อมูลมารวมกันในเชิงบรรทัด ด้วยคำสั่ง SQL UNION statement โดยข้อมูลที่เลือกมาจำเป็นต้องมีคุณสมบัติดังนี้

- * จำนวน column ที่เลือกมาเท่ากัน
- * มี data type ทั้งหมดชนิดเดียวกัน
- * จัดเรียง column ในรูปแบบเดียวกัน

คำสั่งประยุกต์ UNION
รวมข้อมูลตั้งแต่ 2
table ขึ้นไป



- รูปแบบ (Syntax)



```
SELECT v1 FROM t1 UNION
SELECT v2 FROM t2;
```



v1
1
2
3
4

ตัวอย่าง คำสั่งประยุกต์ UNION รวมข้อมูลตั้งแต่ 2 table ขึ้นไป



Employee

```
SELECT FirstName, LastName FROM Employee
```

	ABC FirstName	ABC LastName
1	Andrew	Adams
2	Nancy	Edwards
3	Jane	Peacock
4	Margaret	Park
5	Steve	Johnson
6	Michael	Mitchell
7	Robert	King
8	Laura	Callahan

มี 8 แถว



Customer

```
SELECT FirstName, LastName FROM Customer
```

	ABC FirstName	ABC LastName
1	Luis	Gonçalves
2	Leonie	Köhler
3	François	Tremblay
4	Bjørn	Hansen
5	František	Wichterlová
6	Helena	Holý
7	Astrid	Gruber

มี 59 แถว

```
SELECT FirstName, LastName, 'Employee' AS Type
FROM Employee
UNION
SELECT FirstName, LastName, 'Customer'
FROM Customer
```



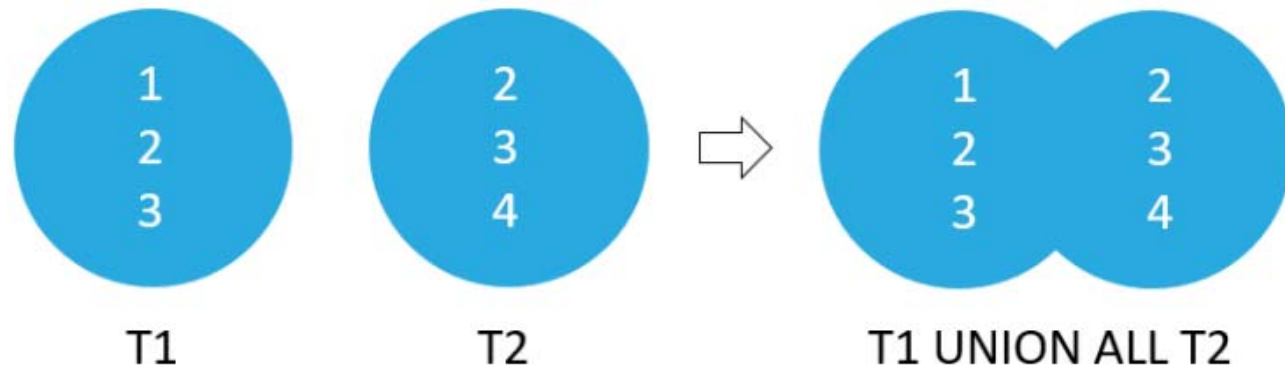
	ABC FirstName	ABC LastName	ABC Type
1	Aaron	Mitchell	Customer
2	Alexandre	Rocha	Customer
3	Andrew	Adams	Employee
4	Astrid	Gruber	Customer
5	Bjørn	Hansen	Customer
6	Camille	Bernard	Customer
7	Daan	Peeters	Customer

มี 67 แถว

คำสั่งประยุกต์
UNION ALL รวมข้อมูล
ตั้งแต่ 2 table ขึ้นไป



เอาข้อมูลทั้งหมดของ 2 table มารวมกันต่อให้ข้อมูลซ้ำกันก็ไม่มีเอาออกให้
รูปแบบ (Syntax)



```
SELECT v1 FROM t1
UNION ALL
SELECT v2 FROM t2;
```



v1
1
2
3
2
3
4

ตัวอย่าง คำสั่งประยุกต์
UNION ALL รวมข้อมูล
ตั้งแต่ 2 table ขึ้นไป



Employee

```
SELECT FirstName, LastName FROM Employee
```

	ABC FirstName	ABC LastName
1	Andrew	Adams
2	Nancy	Edwards
3	Jane	Peacock
4	Margaret	Park
5	Steve	Johnson
6	Michael	Mitchell
7	Robert	King
8	Laura	Callahan

มี 8 แถว



Customer

```
SELECT FirstName, LastName FROM Customer
```

	ABC FirstName	ABC LastName
1	Luís	Gonçalves
2	Leonie	Köhler
3	François	Tremblay
4	Bjørn	Hansen
5	František	Wichterlová
6	Helena	Holý
7	Astrid	Gruber

มี 59 แถว

```
SELECT FirstName, LastName,
'Employee' AS Type
FROM Employee
UNION ALL
SELECT FirstName, LastName,
'Customer'
FROM Customer
```



	ABC FirstName	ABC LastName	ABC Type
1	Andrew	Adams	Employee
2	Nancy	Edwards	Employee
3	Jane	Peacock	Employee
4	Margaret	Park	Employee
5	Steve	Johnson	Employee
6	Michael	Mitchell	Employee
7	Robert	King	Employee

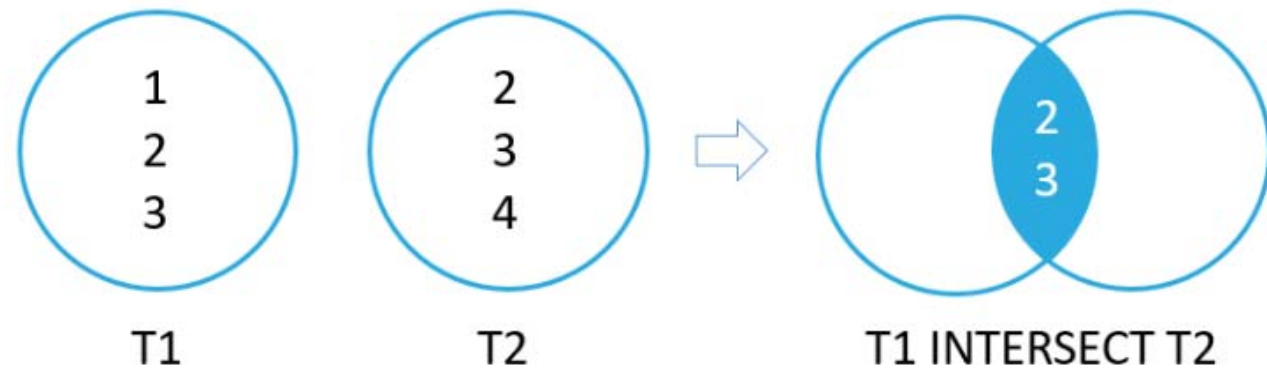
มี 67 แถว

INTERSECT and EXCEPT : not in SQL Server



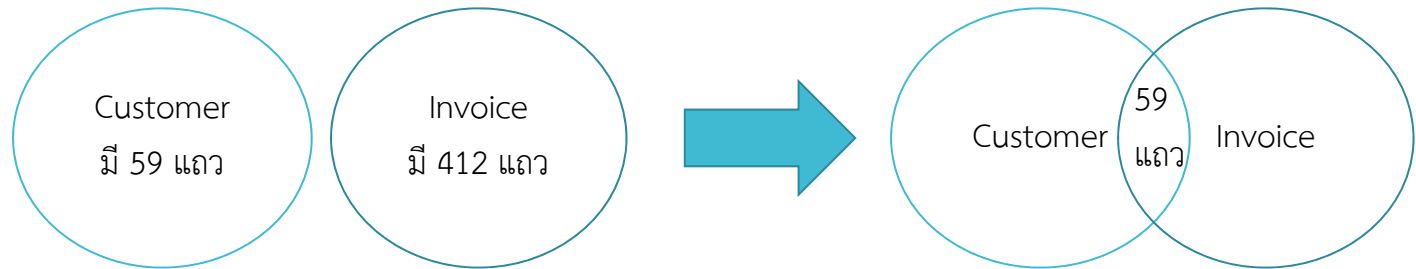
1. INTERSECT

เลือกเอาเฉพาะข้อมูลบรรทัดที่มีเหมือนกันเท่านั้นใน 2 table
รูปแบบ (Syntax)



```
SELECT select_list1 FROM table1
INTERSECT SELECT
select_list2 FROM table2
```

INTERSECT and
EXCEPT :
not in SQL Server



```
SELECT CustomerId
FROM customer
INTERSECT
SELECT CustomerId
FROM invoice
ORDER BY CustomerId
```



	123 CustomerId	
1	1	1
2		2
3		3
4		4
5		5
6		6
7		7

มี 59 แถว

2. Quantifiers



ค้นหาแทร็กที่มีรหัสอัลบั้ม = 1 และมีความยาวมากกว่า 250,000 มิลลิวินาที

```
SELECT name, milliseconds, bytes, albumid
FROM track
WHERE albumid = 1 AND milliseconds > 250000
```

Track ☒

SQL: `SELECT name, milliseconds, bytes, albumid FROM` *Enter a SQL expression to filter results (use Ctrl+Space)*

	ABC Name	123 Milliseconds	123 Bytes	123 AlbumId
1	For Those About To Rock (We Salute You)	343,719	11,170,334	1
2	Evil Walks	263,497	8,611,245	1
3	Breaking The Rules	263,288	8,596,840	1
4	Spellbound	270,863	8,817,038	1

3. Nested Query



```
SELECT column_1
FROM table_1
WHERE column_1 = (
    SELECT column_1
    FROM table_2
);
```

ต้องการหา trackid , ชื่อ track, และ albumid ที่มี albumid เดียวกันกับ ชื่ออัลบั้มว่า 'Let There Be Rock'

```
SELECT trackid,
       name,
       albumid
FROM track
WHERE albumid = (
    SELECT albumid
    FROM album
    WHERE title = 'Let There Be Rock'
)
```

Track

SELECT trackid, name, albumid FROM

	TrackId	Name	AlbumId
1	15	Go Down	4
2	16	Dog Eat Dog	4
3	17	Let There Be Ro	4
4	18	Bad Boy Boogie	4
5	19	Problem Child	4
6	20	Overdose	4
7	21	Hell Ain't A Bac	4
8	22	Whole Lotta Ro	4

3. Group-by



ค้นหา : ให้นำ Trackid ของ Albumid = 1 ว่ามีจำนวนเท่าไร

```
SELECT Albumid,
COUNT(Trackid)
FROM Track
GROUP BY Albumid
HAVING Albumid = 1;
```

	123 Albumid	123 COUNT(Trackid)
1	1	10

คำสั่งสร้างวิว CREATE VIEW



```
CREATE VIEW view_name[(column-  
name-list)] AS select-statement;
```

```
CREATE VIEW v_track  
AS  
SELECT  
    trackid,  
    track.name,  
    album.Title AS album,  
    mediatype.Name AS media,  
    genre.Name AS genre  
FROM  
    track  
INNER JOIN album ON Album.AlbumId = track.AlbumId  
INNER JOIN mediatype ON mediatype.MediaTypeId = track.MediaTypeId  
INNER JOIN genre ON genre.GenreId = track.GenreId;
```

```
SELECT * FROM v_track;
```

คำสั่งสร้างวิว CREATE VIEW



```
CREATE VIEW v_album ( AlbumTitle, Minutes ) AS
SELECT album.title, SUM(milliseconds) / 60000
FROM track INNER JOIN album USING ( AlbumId )
GROUP BY Album.Title;
```

```
SELECT * FROM v_album;
```

คำสั่งลบวิว
CREATE VIEW



```
DROP VIEW view_name;
```

```
DROP VIEW v_album;
```

อ้างอิงจากลิงค์



<https://mindphp.com/forums/viewtopic.php?f=74&t=18341&sid=875c58559f215f474cb77c2f07f5aae7>
<https://saixiii.com/database-sql-data-types/>
<https://www.sqlitetutorial.net/>
<https://www.sqlitetutorial.net/sqlite-where/>
<https://www.sqlitetutorial.net/sqlite-select/>
<https://www.sqlitetutorial.net/sqlite-having/>
<https://www.sqlitetutorial.net/sqlite-union/>
<https://www.sqlitetutorial.net/sqlite-intersect/>
<https://www.sqlitetutorial.net/sqlite-group-by/>
<https://www.sqlitetutorial.net/sqlite-inner-join/>
<https://www.sqlitetutorial.net/sqlite-is-null/>
<https://www.sqlitetutorial.net/sqlite-subquery/>
<https://www.sqlitetutorial.net/sqlite-self-join/>
<https://www.sqlitetutorial.net/sqlite-create-view/>
<https://www.sqlite.org/datatype3.html>
<https://www.sqlitetutorial.net/sqlite-date-functions/sqlite-date-function/>

SQLite



THANK YOU

