



SQL for Data Analyst 101

➤ Parent item	👤 OnDemand Class
⚙️ Status	Success

ไฟล์ Database ที่ใช้เรียน

chinook.db

เว็บ SQLite ใช้สำหรับเรียน

<https://sqliteonline.com/>

กับ DB Browser for SQLite

💡 Table of Contents 💡

💡 Table of Contents 💡

Database and ER Diagram

[DB Browser Installation](#)

[Select columns](#)

[Create new columns](#)

[Create new columns with CASE](#)

[Working with date](#)

[Get data from multiple tables](#)

[Inner vs. Left Join](#)

[More join examples](#)

[Random sampling](#)

Database and ER Diagram

Database คือฐานข้อมูล ใน 1 Database ประกอบไปด้วยหลาย Table ซึ่งเป็น Structured Data และในแต่ละ Table จะมีหลายคอลัมน์ และในแต่ละคอลัมน์จะประกอบไปด้วยข้อมูลและประเภทของข้อมูล(Data Type)

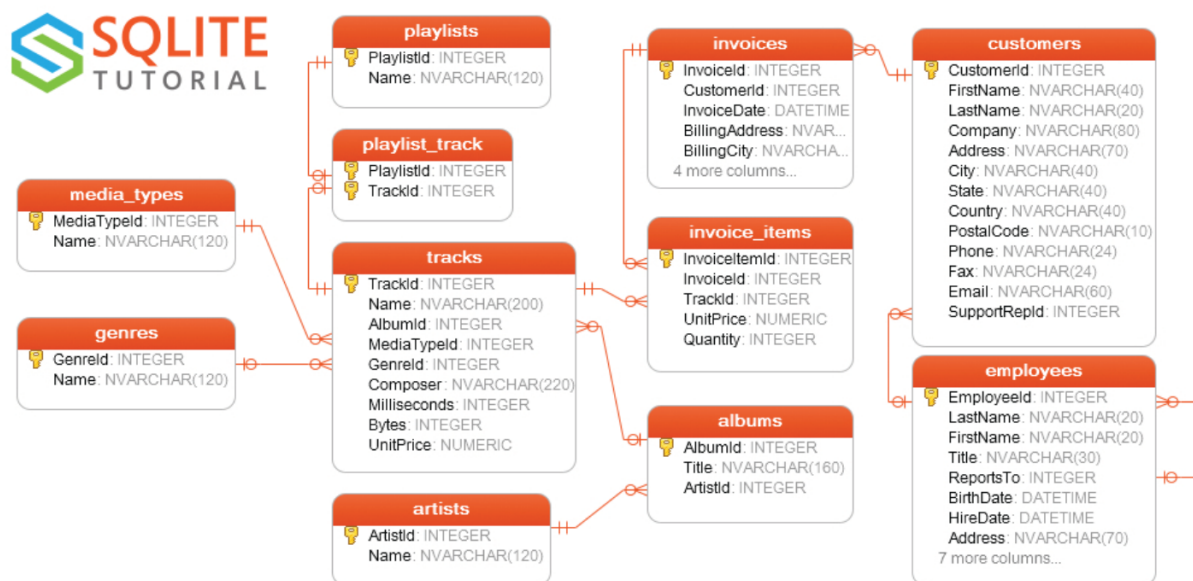
Primary Key (PK) คือ Unique Identifier หรือข้อมูลที่ไม่ซ้ำกัน

Foreign Key (FK) คือ คีย์ที่ใช้ PK ในการเชื่อมกับ Table อื่น

ER Diagram ย่อมาจาก Entity-Relationship Diagram เป็นเหมือนแผนที่ ที่บอกเราว่าข้อมูลใน database มีอะไรบ้าง และความสัมพันธ์ของแต่ละ table เป็นอย่างไร

รูปแบบความสัมพันธ์ที่เราใช้ใน database มีสามแบบหลักๆคือ

1. One-to-One 1:1
2. One-to-Many 1:M (อันนี้ใช้เยอะสุดเลย)
3. Many-to-Many M:N



DB Browser Installation

สามารถดาวน์โหลดโปรแกรมลงคอมได้ <https://sqlitebrowser.org/dl/>

หรือจะเรียนผ่านเว็บก็ได้เหมือนกัน <https://sqliteonline.com/>

Select columns

SELECT คือคำสั่งดึงข้อมูลจาก Table

```
SELECT * FROM tablename;
-- ถ้าต้องการดึงข้อมูลทุกคอลัมน์ใน Table ให้ใช้ *
```

```
SELECT firstname,lastname FROM customers;
-- ถ้าต้องการดึงแบบเฉพาะเจาะจงให้ใช้ชื่อคอลัมน์นั้นๆ
```

Create new columns

```
SELECT
    firstname,
    lastname,
    firstname || ' ' || lastname AS Fullname,
    LOWER(FirstName) || '@company.com' AS Email
FROM customers ;
/*
ดึงคอลัมน์ firstname,lastname,
firstname มาเชื่อมกับ ' ' เชื่อมกับ lastname แล้วตั้งว่า Fullname,
ทำfirstnameให้เป็นตัวพิมพ์เล็กแล้วนำมาเชื่อมกับคำว่า '@company.com' แล้วตั้งชื่อว่า
ดึงมาจาก Table customers
*/

SELECT
    name,
    ROUND(milliseconds/60000.0,2) AS minute,
    ROUND(bytes/(1024*1024.0),2) AS mb
FROM tracks;
/*
ดึงคอลัมน์ name,
นำ milliseconds หาร 60000.0 แล้วปัดทศนิยมเป็น 2 ตำแหน่ง ตั้งชื่อว่า minute
นำ bytes หาร (1024*1024) แล้วปัดทศนิยมเป็น 2 ตำแหน่ง ตั้งชื่อว่า mb
ดึงมาจาก table tracks;
*/
```

หากต้องการตั้งชื่อคอลัมน์ใหม่ให้ใช้ **AS** ตามด้วยชื่อใหม่

LOWER() ทำให้ text กลายเป็นตัวพิมพ์ใหญ่ทั้งหมด

UPPER() ทำให้ text กลายเป็นตัวพิมพ์เล็กทั้งหมด

`ROUND()` เป็นคำสั่งปัดเลขทศนิยมเหมือนpythonเลย

Create new columns with CASE

`CASE WHEN` ใช้ในการเขียนเงื่อนไข สร้างคอลัมน์ใหม่ เราจะเขียนใน `SELECT`

ทุกครั้งที่เราเขียนเสร็จอย่าลืม `END` ปิดท้ายเสมอ

เราสามารถเขียนได้มากกว่าหนึ่งเงื่อนไข ได้ผลลัพธ์เหมือนกับฟังก์ชัน `=IFS()` ของ Google Sheets/ Excel หรือ if-else ในภาษาpython

```
SELECT
    company,
    CASE
        WHEN company IS NOT NULL THEN 'Corporate'
        ELSE 'End Customer'
    END AS segment
FROM customers;
/*
เลือกคอลัมน์ company,
เงื่อนไขถ้า company ไม่ใช่ค่า null ให้แสดง 'Corporate'
ถ้าเป็นค่า null ให้แสดง 'End Customer'
END แล้วตั้งชื่อคอลัมน์ว่า segment
ดึงจากตาราง customers;
*/
```

Working with date

มาตรฐานในการเก็บ DATE คือ YYYY-MM-DD เช่น 2023-01-20

```
SELECT
    Invoicedate,
    CAST(STRFTIME('%Y',invoicedate)AS INT) AS year,
    STRFTIME('%m',invoicedate) AS month,
    STRFTIME('%d',invoicedate) AS day,
```

```

        STRFTIME( '%Y-%m-%d', invoicedate) AS date
FROM invoices
WHERE year = 2010;
/*
เลือกคอลัมน์ invoicedate,
ใช้ฟังก์ชัน STRFTIME ถึง ปี ออกมาจาก invoicedate แล้วใช้ CAST แปลงจาก STR:
ใช้ฟังก์ชัน STRFTIME ถึง เดือน แล้วตั้งชื่อคอลัมน์ว่า month,
ใช้ฟังก์ชัน STRFTIME ถึง วัน แล้วตั้งชื่อคอลัมน์ว่า day,
ใช้ฟังก์ชัน STRFTIME ถึง ปี-เดือน-วัน แล้วตั้งชื่อคอลัมน์ว่า date
จากตาราง invoices
เงื่อนไข year ต้องเท่ากับ 2010
*/

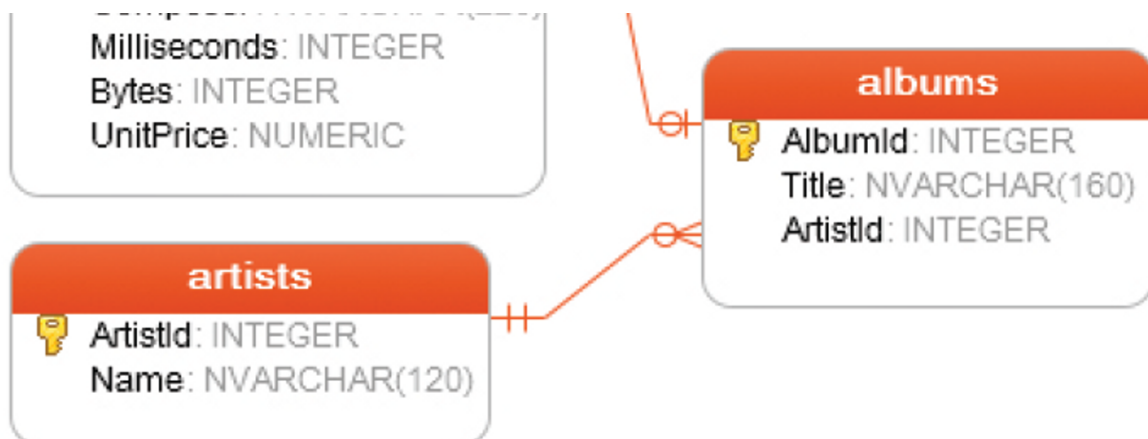
```

STRFTIME() เป็นฟังก์ชันดึงวันเดือนปีออกมาจาก text

- %Y ใช้ถึงปีให้ ใช้ตัวพิมพ์ใหญ่
- %m ใช้ถึงเดือน ใช้ตัวพิมพ์เล็กเท่านั้น
- %d ใช้ถึงวัน ใช้ตัวพิมพ์เล็กเท่านั้น

CAST เป็นฟังก์ชันแปลงชนิดข้อมูล เช่น CAST AS INT

Get data from multiple tables



JOIN Table artists กับ albums ด้วย Artistid (foreign key)

```

SELECT * FROM artists
JOIN albums
ON artists.artistid = albums.artistid
WHERE artists.artistid = 50;
/*
เลือกแสดงคอลัมน์ทั้งหมดของ artists นำไป JOIN กับ albums
โดย artists.artistid = albums.artistid เท่านั้น
เงื่อนไขแสดงเฉพาะ artists.artistid = 50;
*/

```

หรือ สามารถตั้งชื่อ table ใหม่ได้ ให้สั้นๆ โดยใช้ AS

```

SELECT
    A.artistid,
    B.albumid,
    A.name,
    B.title
FROM artists AS A
JOIN albums AS B
ON A.artistid = B.artistid;

```

วิธีการ JOIN table คือ เลือก table ที่ต้องการ แล้วนำไป JOIN กับ table ที่เลือก ON primary key กับ foreign key ที่เหมือนกัน

หรือจะเขียนอีกแบบ ก็ได้ โดยใช้ WHERE กำหนดเงื่อนไข

```

SELECT *
FROM artists,albums
WHERE artists.artistid = albums.artistid;

```

หรือ

```

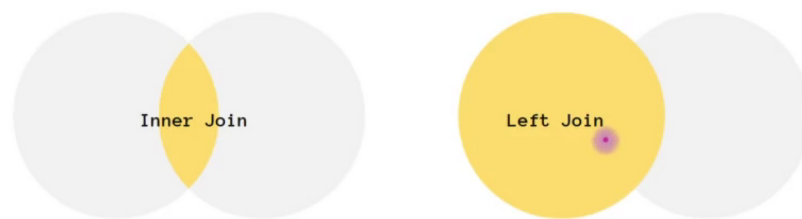
SELECT
    artists.artistid,
    albums.albumid,
    artists.name,
    albums.title

```

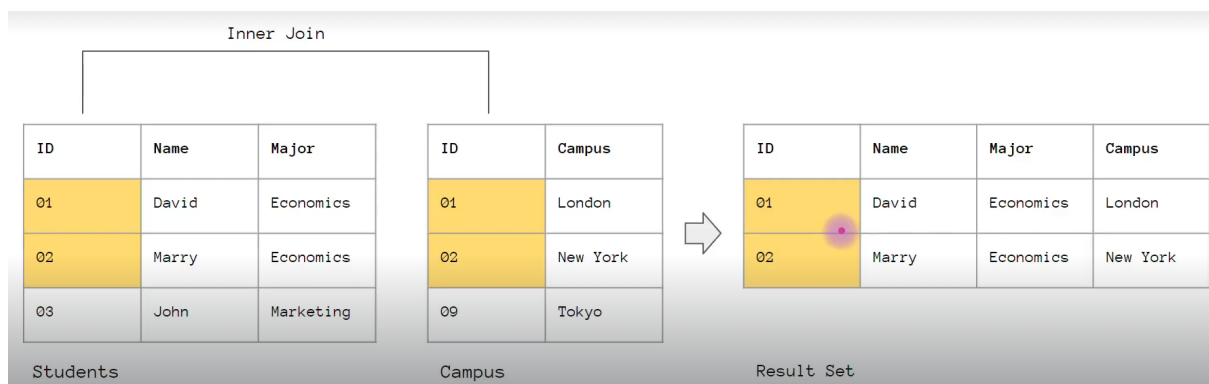
```
FROM artists,albums
WHERE artists.artistid = albums.artistid;
```

Inner vs. Left Join

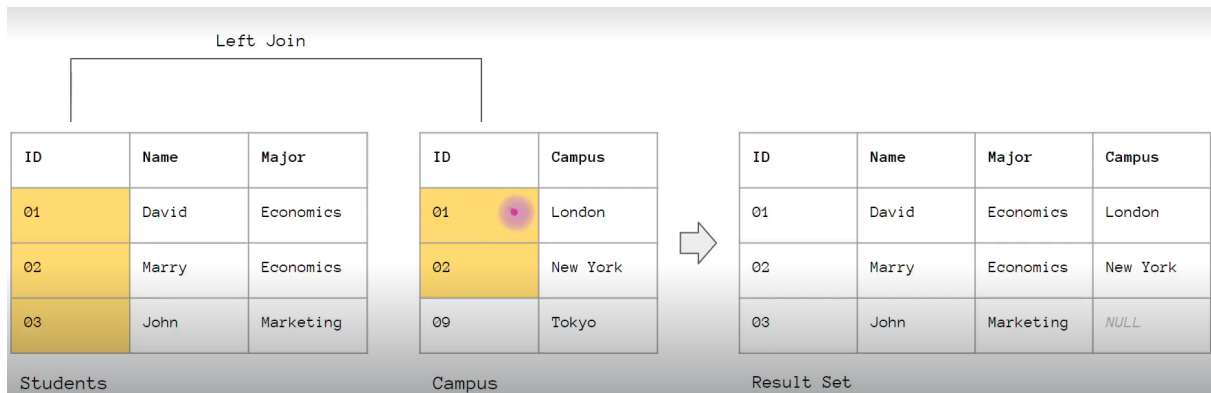
Inner vs. Left Join



INNER JOIN คือ การ join กันแบบปกติเหมือน join ทุกอย่าง เช่น `id = id` ต้องเหมือนกันเท่านั้น ถึงจะ join กันได้



LEFT JOIN คือ การยก Table ซ้ายมาตั้งรอไว้ก่อน row ไหน match ค่ากันได้ก็ดึงมา แต่ row ที่ไม่ match กันก็จะแสดงเป็น NULL (Missing Value)



#INNER JOIN: Returns only row with overlapping values

SELECT

A.artistid,

A.name,

B.title

FROM artists as A

INNER JOIN albums as B

ON A.artistid = B.artistid;

#LEFT JOIN: Returns all rows from the left table, shows NULL

SELECT

A.artistid,

A.name,

B.title

FROM artists as A

LEFT JOIN albums as B

ON A.artistid = B.artistid;

More join examples

เราสามารถ JOIN Table ได้มากกว่า 1 ตาราง

SELECT

artists.artistid,

artists.name,

albums.title,


```
        tracks.name,  
        tracks.composer  
FROM artists  
JOIN albums  
ON artists.artistid = albums.artistid  
JOIN tracks  
ON albums.albumid = tracks.albumid  
WHERE artists.name = 'Aerosmith';  
/*  
ใช้ artists.artistid = albums.artistid  
และ albums.albumid = tracks.albumid  
เท่านั้นตารางก็จะเชื่อมกัน  
*/
```

Random sampling

RANDOM() คือ ฟังก์ชันสุ่มค่า ทุกครั้งที่สุ่มค่าก็จะเปลี่ยนไปเรื่อยๆ

```
SELECT name, RANDOM() FROM tracks  
ORDER BY RANDOM()DESC  
LIMIT 10;
```

หรือ

```
SELECT name  
FROM tracks  
ORDER BY RANDOM()DESC  
LIMIT 10;
```

ในตัวอย่างเราสุ่มค่า name ใน Table tracks ให้แสดงผลแค่ 10 ชื่อ
