SQL Database

Basic SQL knowledge

- SQL : Structured Query Language
- ER Diagram : Entity Relational Diagram
    - Primary key ห้ามซ้ำกันเลยในตารางของมันเอง แต่ถ้าชื่อมันไปโผล่ในตารางอื่น จะเรียกว่า Forien Key หน้าที่ของมันคือการเชื่อมตารางเข้าด้วยกัน
    - ตารางหนึ่งตารางสามารถมี key สองตัวได้ไหม คำตอบคือได้ จะถูกเรียกว่าตารางกลาง (Bridge table) และจะเรียก key ว่า Composite Key
    - ความสัมพันธ์ one-many จะใช้งานกันบ่อยที่สุด
- Conclude SQL function table

Clauses	What it does?
SELECT	Select columns
FROM	From table
JOIN	Join multiple tables
WHERE	Filter data
Aggregate Functions	AVG SUM MIN MAX COUNT
GROUP BY	Group by statistics
HAVING	Filter groups
ORDER BY	Sort data

Create Table

ซึ่งการสร้าง Table ใน SQL เราจะต้องกำหนด data type ด้วยซึ่งมีดังนี้

- INT ตัวเลขจำนวนเต็ม
- REAL (real number หรือ float ตัวเลขทศนิยม)
- TEXT
- DATE (วันที่ใน SQLite จะมองเป็น TEXT หมดเลย)

```
CREATE TABLE employee (
        id INT UNIQUE,  -- unique คือคอลัมน์นี้ห้ามซ้ำ หรือกำหนดเป็น PRIMA
    name TEXT,
    department TEXT,
    salary REAL
);
```

Insert Data

ระวังดี ๆ คอลัมน์ที่เป็น unique หรือ primary key เราจะไม่สามารถใส่ข้อมูลซ้ำได้ แปลว่า `INSERT INTO` จะรันได้แค่ครั้งเดียว ถ้ากดรันมากกว่า 1 ครั้งมันจะแจ้ง errors กลับมา

```
INSERT INTO employee VALUES
      (1, "David", "Marketing", "CEO", 100000),
    (2, "John", "Marketing", "VP", 85000),
    (3, "Marry", "Sales", "Manager", 60000) ;
```

การเปลี่ยนชื่อ Table และเพิ่มคอลัมน์ใหม่ ในข้อมูลที่เรามีอยู่แล้ว

- Rename table

```
ALTER TABLE employee RENAME TO Myemployee;

SELECT * FROM Myemployee
```

- Add column and update data ก็จะได้คอลัมน์ใหม่ที่มีชื่อว่า email และ id ที่ 1 มีค่าใน column email = ceo@company.com

```
ALTER TABLE Myemployee
ADD email TEXT;

UPDATE Myemployee
SET email = "ceo@compamy.com"
WHERE id = 1;

SELECT * FROM Myemployee
```

Common SQL error

- `Table xxx already exists` มี table name ซ้ำ ให้เช็คก่อนว่ามี database เราที่มีชื่อนั้นอยู่แล้วหรือยัง
- `UNIQUE constraint failed: table.name` ถ้าเรากำหนดบางคอลัมน์เป็น unique หรือ primary key เราจะไม่สามารถ insert ข้อมูลซ้ำเข้าไปในตารางได้ วิธีแก้ให้ลอง `SELECT * FROM`

`employee` ว่ามีข้อมูลชื่อ ... อยู่ในตารางแล้วหรือยัง หรือ เราจะสร้าง `INSERT INTO` ใหม่แและคอมเมนต์ตัวที่เคยรันแล้วเพื่อไม่ให้มันแจ้ง error

- `No such table : emp` เขียนชื่อตารางผิดตอนรัน `SELECT` ให้ตรวจสอบชื่อตารางก่อน ซึ่งส่วนมากแล้วจะตั้งเป็นตัวพิมพ์เล็ก
- `near “FROM” : syntex errors` คือใส่ , comma เกิน อย่าลืมตรวจสอบ comma ก่อนรันน้า

การใช้งาน Random

เราใช้เพื่อที่จะสุ่มค่าออกมา จากตัวอย่าง สุ่มชื่อในคอลัมน์ name ออกมา 5 ตัว

```
SELECT name
FROM tracks
ORDER BY random() DESC
LIMIT 5
```

การใช้งานคำสั่ง SELECT

#เครื่องหมาย (;) เอาไว้ใช้ในการที่เราจะแบ่ง query เช่นเราต้องการเขียนหลายๆ query

#เครื่องหมาย (- -) คือการคอมเมนต์แค่หนึ่งบรรทัด(single line comment)

#ถ้าจะคอมเมนต์หลาย บรรทัดให้ใช้ (/* ข้อความ */) เรียกว่า multiple line comment

- การเชื่อมคอลัมน์เข้าด้วยกันโดย || ‘ ‘ || หรือจะเป็น || ‘ข้อความที่ต้องการต่อหลังได้’

```
SELECT
    FirstName,
    LastName,
    FirstName || ' ' || LastName AS Fullname
FROM customers;
```

	FirstName	LastName	Fullname
1	Luís	Gonçalves	Luís Gonçalves
2	Leonie	Köhler	Leonie Köhler
3	François	Tremblay	François Tremblay
4	Bjørn	Hansen	Bjørn Hansen
5	František	Wichterlová	František Wichterlová
6	Helena	Holý	Helena Holý
7	Astrid	Gruber	Astrid Gruber
8	Daan	Peeters	Daan Peeters
9	Kara	Nielsen	Kara Nielsen
10	Eduardo	Martins	Eduardo Martins
11	Alexandre	Rocha	Alexandre Rocha

- สามารถสร้างคอลัมน์ใหม่ในคำสั่ง SELECT ได้เลย หรือจะใช้คำนวณคิดเลขได้ เช่น

```
SELECT
    InvoiceDate,
    BillingAddress,
    total,
    total + (total * 0.07) AS total_incl_vat,
    (100+2)*5 AS answer
FROM invoices;
```

	InvoiceDate	BillingAddress	Total	total_incl_vat	answer
1	2009-01-01 00:00:00	Theodor-Heuss-Straße 34	1.98	2.1186	510
2	2009-01-02 00:00:00	Ullevålsveien 14	3.96	4.2372	510
3	2009-01-03 00:00:00	Grétrystraat 63	5.94	6.3558	510
4	2009-01-06 00:00:00	8210 111 ST NW	8.91	9.5337	510
5	2009-01-11 00:00:00	69 Salem Street	13.86	14.8302	510
6	2009-01-19 00:00:00	Berger Straße 10	0.99	1.0593	510
7	2009-02-01 00:00:00	Barbarossastraße 19	1.98	2.1186	510
8	2009-02-01 00:00:00	8, Rue Hanovre	1.98	2.1186	510
9	2009-02-02 00:00:00	9, Place Louis Barthou	3.96	4.2372	510
10	2009-02-03 00:00:00	3 Chatham Street	5.94	6.3558	510
11	2009-02-06 00:00:00	202 Hoxton Street	8.91	9.5337	510

- ใช้คำสั่ง ROUND ใช้ในการกำหนดทศนิยม value funvtion to change format หรือ จะใช้ LOWER เปลี่ยนเป็นตัวเล็ก หรือ UPPER เปลี่ยนเป็นตัวใหญ่

```
SELECT
    InvoiceDate,
    BillingAddress,
```

```
    total,
    ROUND(total + (total * 0.07),2) AS total_incl_vat
FROM invoices;
```

	InvoiceDate	BillingAddress	Total	total_incl_vat
1	2009-01-01 00:00:00	Theodor-Heuss-Straße 34	1.98	2.12
2	2009-01-02 00:00:00	Ullevålsveien 14	3.96	4.24
3	2009-01-03 00:00:00	Grétrystraat 63	5.94	6.36
4	2009-01-06 00:00:00	8210 111 ST NW	8.91	9.53
5	2009-01-11 00:00:00	69 Salem Street	13.86	14.83
6	2009-01-19 00:00:00	Berger Straße 10	0.99	1.06
7	2009-02-01 00:00:00	Barbarossastraße 19	1.98	2.12
8	2009-02-01 00:00:00	8, Rue Hanovre	1.98	2.12
9	2009-02-02 00:00:00	9, Place Louis Barthou	3.96	4.24
10	2009-02-03 00:00:00	3 Chatham Street	5.94	6.36
11	2009-02-06 00:00:00	202 Hoxton Street	8.91	9.53

- การใช้ฟังก์ชัน STRFTIME เพื่อดึง ปี เดือน วัน ออกมาเป็นอีกคอลัมน์เพื่อใช้งานต่อ ส่วนบรรบัดสุดท้ายตามโค้ดด่านล่างเป็นเทคนิคเพื่อใช้งาน Where ได้งานขึ้นและตรงจุด เช่น อยากได้ข้อมูลแค่ เดือน 10 ปี 2009

  สังเกตว่าแถว `WHERE` ต้องใช้ “” เพราะด้านบนที่สร้างมันเป็น `text` เราสามารถเปลี่ยนประเภทข้อมูลได้ เช่น `CAST(STRFTIME("%Y",  InvoiceDate) AS INT) AS year` ถ้าเป็นแบบนี้ตรง WHERE ไม่ต้องใส่ “” แล้ว

```
SELECT
    InvoiceDate,
    STRFTIME("%Y",  InvoiceDate) AS year,
    STRFTIME("%m",  InvoiceDate) AS month,
    STRFTIME("%d",  InvoiceDate) AS day,
    STRFTIME("%Y%m",  InvoiceDate) AS monthid
FROM invoices
WHERE monthid = "200910"
```

	InvoiceDate	year	month	day	monthid
1	2009-10-07 00:00:00	2009	10	07	200910
2	2009-10-07 00:00:00	2009	10	07	200910
3	2009-10-08 00:00:00	2009	10	08	200910
4	2009-10-09 00:00:00	2009	10	09	200910
5	2009-10-12 00:00:00	2009	10	12	200910
6	2009-10-17 00:00:00	2009	10	17	200910
7	2009-10-25 00:00:00	2009	10	25	200910

การใช้งานคำสั่ง WHERE

คือการ Filter table เงื่อนไขตามที่เราต้องการ ดึงได้มากกว่าหนึ่งเงื่อนไข

- WHERE เป็นฟังช์กันที่เวลาค้นหาคำที่อยู่ในตารางจะต้องพิมพ์ชื่อให้ตรง ไม่อย่างนั้นจะไม่มีข้อมูลขึ้นมาแสดง ทริคการแก้ง่ายๆคือ ใส่ฟังก์ชัน LOWER และ UPPER ตรงชื่อคอลัมน์

```
SELECT
    FirstName,
    Country,
    Email
FROM customers
WHERE  LOWER(Country) = 'united kingdom' ;
```

	FirstName	Country	Email
1	Emma	United Kingdom	emma_jones@hotmail.com
2	Phil	United Kingdom	phil.hughes@gmail.com
3	Steve	United Kingdom	steve.murray@yahoo.uk

- ถ้าต้องการดึงหลายๆ เงื่อนไขให้ใช้คำสั่ง OR คั่นกลางระหว่างเงื่อนไข เช่น `WHERE Country = 'United Kingdom' OR Country = 'Brazil' OR Country = 'Belgium' ;` จะเห็นว่าโค้ดยาวมาก มีฟังก์ชันที่ทำให้โค้ดสั้นลงคือ `IN operetor หรือ NOT IN`

```
SELECT
    FirstName,
```

```
    Country,
    Email
FROM customers
WHERE  Country IN ('USA' , 'Brazil', 'Belgium', 'France') ;
```

	FirstName	Country	Email
1	Luís	Brazil	luisg@embraer.com.br
2	Daan	Belgium	daan_peeters@apple.be
3	Eduardo	Brazil	eduardo@woodstock.com.br
4	Alexandre	Brazil	alero@uol.com.br
5	Roberto	Brazil	roberto.almeida@riotur.gov.br
6	Fernanda	Brazil	fernadaramos4@uol.com.br
7	Frank	USA	fharris@google.com

- หรือถ้าเติม Not หลัง Where คือเราจะเอาหมด **ยกเว้น** USA และ CANADA

```
SELECT * FROM customers
WHERE NOT (Country = "USA" OR Country = "Canada")
```

- การสร้างตารางใหม่ แค่ใช้ฟังก์ชัน CREATE TABLE บนสุดของโค้ด อย่าลืมตั้งชื่อตารางด้วยน้า ส่วนมากใช้เพื่อสร้างตาราง backup หรือ copy นั้นเอง

```
CREATE TABLE eu_customers AS
    SELECT
        FirstName,
        Country,
        Email
    FROM customers
    WHERE  Country IN ('USA' , 'Brazil', 'Belgium', 'France') ;
```

และถ้าต้องการลบตาราง ให้พิมพ์ `DROP TABLE ตามด้วยชื่อ ;`

- ปกติแล้วเนี่ย Where จะรันก่อน Select เพราะฉนั้นเราสามารถ filter คอลัมน์ที่ไม่ได้อยู่ใน select ก่อนได้ โดยการจะทำแบบนี้ มีในเฉพาะ SQLite ดังนั้น ถ้าจะให้เซฟๆ ให้เขียนสูตรตรง WHERE ด้วย (จะใช้ชื่อย่อใน select มาใส่ใน where ไม่ได้ ต้องใช้ operetion เต็มๆ)

```
SELECT
    name,
    Composer,
    Bytes/(1024*1024) AS MB
FROM tracks
WHERE  Bytes/(1024*1024) >= 8
```

- ต่อมาถ้าอยาก filter ชื่อหรือข้อความ เราจะใช้คำสั่ง LIKE ซึ่งเป็นฟังก์ชัน Insensitive `1. %` `(wildcars แมตกี่ charactor ก็ได้ หรือจะไม่แมทเลยก็ได้) 2. _ แมตแค่ตัวเดียว`

```
SELECT
    name,
    Composer,
    Bytes/(1024*1024) AS MB
FROM tracks
WHERE  Bytes/(1024*1024)  >= 8 AND Composer LIKE 'S%' ;
```

Name	Composer	MB
Walk On Water	Steven Tyler, Joe Perry, Jack Blades, Tommy Shaw	9
Love In An Elevator	Steven Tyler, Joe Perry	10
Rag Doll	Steven Tyler, Joe Perry, Jim Vallance, Holly Knight	8
What It Takes	Steven Tyler, Joe Perry, Desmond Child	9
Dude (Looks Like A Lady)	Steven Tyler, Joe Perry, Desmond Child	8
Janie's Got A Gun	Steven Tyler, Tom Hamilton	10
Cryin'	Steven Tyler, Joe Perry, Taylor Rhodes	9

หรือจะค้นหา NULL โดยเขียนดังนี้ `WHERE Composer IS NULL ;`

หรือ `WHERE Composer IS NOT NULL ;` - - missing value

- คำสั่ง BETWEEN ดึงค่าระหว่างที่เราต้องการ หรือใช้ในการ **fillter date**

```
SELECT
    name,
    Composer,
    Bytes/(1024*1024) AS MB
FROM tracks
WHERE  Bytes/(1024*1024)  BETWEEN 9 AND 10 ; -- inclusive
```

```
SELECT  InvoiceDate FROM invoices
WHERE  InvoiceDate BETWEEN '2009-01-01 00:00:00' AND '2009-01-1
```

Name	Composer	MB
For Those About To Rock (We Salute You)	Angus Young, Malcolm Young, Brian Johnson	10
Go Down	AC/DC	10
Problem Child	AC/DC	10
Whole Lotta Rosie	AC/DC	10
Walk On Water	Steven Tyler, Joe Perry, Jack Blades, Tommy Shaw	9
Love In An Elevator	Steven Tyler, Joe Perry	10
What It Takes	Steven Tyler, Joe Perry, Desmond Child	9

- การ UPDATE ข้อมูล จากโค้ดด้านล่าง เราจะอัปเดต ตาราง employee ค่าที่ต้องการเปลี่ยน เปลี่ยนที่ id = 1 และลองรันดูผลลัพธ์

```
UPDATE employee
SET salary = 99000
WHERE id = 1;

SELECT * FROM employee
```

- การ DELETE ข้อมูล คือมันจะลบแถวที่มีชื่อ Walker ออก **ระวังดีๆ** ถ้าเราไม่ใส่ `WHERE` มันจะลบทุกแถวออกไปหมดเลย เพราะฉนั้นต้อง `INSERT INTO` ใหม่

```
DELETE FROM employee
WHERE name = 'Walker';

SELECT * FROM employee
```

- `join table` โดยใช้ `WHERE` โดยมีสามตาราง คือ artists, albums, tracks เทียบเท่ากับ `INNER JOIN`

```
SELECT
    artists.ArtistId,
    artists.name AS artist_name,
    albums.Title AS album_name,
    tracks.name AS song_name
FROM artists, albums, tracks
```

```
WHERE artists.ArtistId = albums.ArtistId -- PK == FK
    AND albums.AlbumId = tracks.AlbumId
```

การแทนที่ ค่า null ด้วย coalesce

```
SELECT
    Company,
    coalesce(Company, 'End customers') AS 'company clean',
    CASE WHEN Company IS NULL THEN 'End customers'
               ELSE 'Corporate'
    END AS 'segment'
 FROM customers
```

	Company	company clean	segment
1	Embraer - Empresa Brasileira de Aeronáutica S.A.	Embraer - Empresa Brasileira de Aeronáutica S.A.	Corporate
2	*NULL*	End customers	End customers
3	*NULL*	End customers	End customers
4	*NULL*	End customers	End customers
5	JetBrains s.r.o.	JetBrains s.r.o.	Corporate
6	*NULL*	End customers	End customers
7	*NULL*	End customers	End customers
8	*NULL*	End customers	End customers

การใช้งาน LIMIT

คือฟังก์ชันที่กำหนดจำนวนแถวตามที่เราต้องการ เช่น

```
SELECT
    name,
    Composer,
    Bytes/(1024*1024) AS MB
FROM tracks
LIMIT 3
```

Name	Composer	MB
For Those About To Rock (We Salute You)	Angus Young, Malcolm Young, Brian Johnson	10
Balls to the Wall	*NULL*	5
Fast As a Shark	F. Baltes, S. Kaufman, U. Dirkscneider & W. Hoffman	3

การใช้งาน AGGREGATE

#COUNT ฟังก์ชัน จะไม่นับค่า NULL

```
SELECT
    COUNT(*)      AS total_songs,
    AVG(Bytes)    AS avg_bytes,
    SUM(Bytes/(1024*1024))   AS sum_mb,
    MIN(Bytes)   AS min_bytes,
    MAX(Bytes)  AS max_bytes
FROM tracks ;
```

total_songs	avg_bytes	sum_mb	min_bytes	max_bytes
3503	33510207.0653725	110227	38747	1059546140

avg, sum, min ,max, count จำๆ

Count Distinct

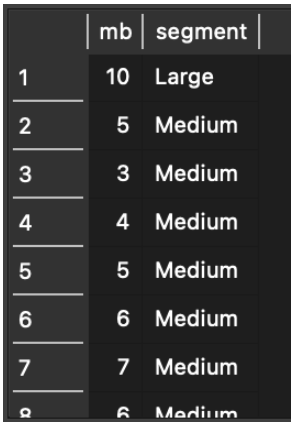
คือการทำค่าที่อยู่ในคอลัมน์เป็นค่าที่ไม่ซ้ำ หรือ Unique

```
SELECT DISTINCT Country FROM customers
```

การสร้างคอลัมน์ใหม่ด้วยใช้เงื่อนไข CASE WHEN

คือการเขียนเงื่อนไข if-else โดยการใช้ case when then else end ท่องๆนะ จำให้ได้ สามารถนำมาแทนค่า `null` ได้ เช่น `WHEN company IS NOT NULL THEN 'Corporate'` หรือจะใช้กับ `text` ได้ เช่น `WHEN country IN ('Canada', 'USA') THEN 'America'`

```
SELECT
    Bytes/(1024*1024) AS mb,
    CASE
        WHEN Bytes/(1024*1024) >= 8 THEN "Large"
        WHEN Bytes/(1024*1024) >= 3 THEN "Medium"
        ELSE "Small"
    END AS segment
FROM tracks ;
```



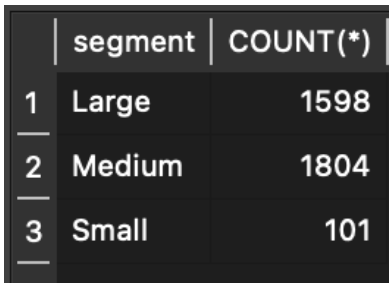
	mb	segment
1	10	Large
2	5	Medium
3	3	Medium
4	4	Medium
5	5	Medium
6	6	Medium
7	7	Medium
8	6	Medium

ฟังก์ชัน GROUP BY

คือการจัดกลุ่มตามคอลัมน์ที่เราต้องการจะจัดกลุ่ม

```
SELECT
    CASE
        WHEN Bytes/(1024*1024) >= 8 THEN "Large"
```

```
        WHEN Bytes/(1024*1024) >= 3 THEN "Medium"
        ELSE "Small"
    END AS segment,
    COUNT(*)
FROM tracks
GROUP BY 1 ;
```



	segment	COUNT(*)
1	Large	1598
2	Medium	1804
3	Small	101

และ group by สามารถรวมกลุ่มได้มากกว่า 1 กลุ่มได้ด้วย

แต่ มีอีก trick หนึ่ง คือการแทนที่ค่า NULL ด้วยค่าอื่นตามที่เรากำหนด โดยฟังก์ชัน COALESCE เช่น

```
SELECT
    Company,
    COALESCE(Company, "B2C") AS clean_company
FROM customers
```

	Company	clean_company
1	Embraer - Empresa Brasileira de Aeronáutica S.A.	Embraer - Empresa Brasileira de Aeronáutica S.A.
2	*NULL*	B2C
3	*NULL*	B2C
4	*NULL*	B2C
5	JetBrains s.r.o.	JetBrains s.r.o.
6	*NULL*	B2C
7	*NULL*	B2C
8	*NULL*	B2C

ถ้าเจอค่า NULL ให้แทนด้วย B2C หรือจะใช้ CASE WHEN ในหัวข้อก่อนหน้านี้ก็ได้ ตามโค้ดด้านล่าง

```
SELECT
    Company,
    COALESCE(Company, "B2C") AS clean_company,
    CASE
        WHEN Company IS NULL THEN "B2C"
        ELSE "B2B"
    END AS segment
FROM customers;
```

	Company	clean_company	segment
1	Embraer - Empresa ...	Embraer - Empresa ...	B2B
2	*NULL*	B2C	B2C
3	*NULL*	B2C	B2C
4	*NULL*	B2C	B2C
5	JetBrains s.r.o.	JetBrains s.r.o.	B2B
6	*NULL*	B2C	B2C
7	*NULL*	B2C	B2C
8	*NULL*	B2C	B2C

ต่อมาเราจะมารวมกลุ่มจริงๆ กันแล้ว

ส่วนมากแล้วในคำสั่ง GROUP BY จะดึงค่ามาใส่โดยเป็นค่าทั่วไปที่ไม่ใช่ค่า aggregate function เช่น

```
SELECT
    CASE
        WHEN Company IS NULL THEN "B2C"
        ELSE "B2B"
    END AS segment,
    Country,
    COUNT(*) AS num_customers
FROM customers
GROUP BY 1,2
```

	segment	Country	m_custom
1	B2B	Brazil	4
2	B2B	Canada	2
3	B2B	Czech Republic	1
4	B2B	USA	3
5	B2C	Argentina	1
6	B2C	Australia	1
7	B2C	Austria	1
8	B2C	Belgium	1

ต่อมาถ้าเราอยากจะดึงเงื่อนไขบางอย่างจากการรวมกลุ่ม เราจะใช้ฟังก์ชันที่มีชื่อว่า HAVING จะเห็นว่าลักษณะการทำงานเหมือนกับ WHERE แต่ HAVING รันหลัง GROUP BY ส่วน WHERE รันก่อน ทั้งสองสามารถใช้แทนกันได้ ได้ผลลัพธ์เหมือนกัน แต่ WHERE จะรันออกเร็วกว่า (ลำดับการรันต่างกัน) **พูดง่ายๆอีกอย่างคือ WHERE กรองตาราง HAVING กรองข้อมูลที่ GROUP BY**

```
SELECT
    CASE
        WHEN Company IS NULL THEN "B2C"
        ELSE "B2B"
    END AS segment,
    Country,
    COUNT(*) AS num_customers
FROM customers
GROUP BY 1,2
HAVING Country IN ('Brazil', 'USA', 'France')
```

	segment	Country	num_customers
1	B2B	Brazil	4
2	B2B	USA	3
3	B2C	Brazil	1
4	B2C	France	5
5	B2C	USA	10

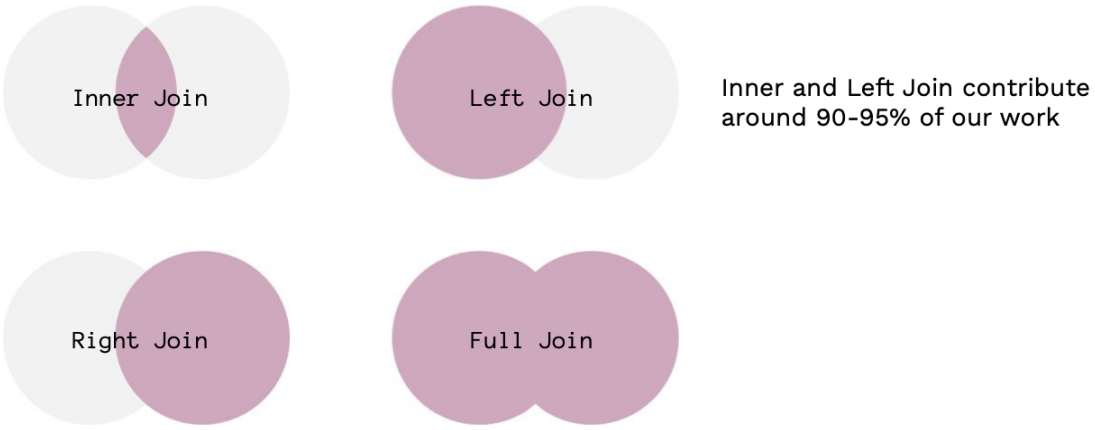
การใช้งาน ORDER BY

คือการเรียงค่า มาก > น้อย หรือ น้อย > มาก

```
SELECT
    name,
    ROUND(Milliseconds/60000.0,2) AS minute
FROM tracks
ORDER BY minute DESC
LIMIT 5
```

	Name	minute
1	Occupation / Precipice	88.12
2	Through a Looking Glass	84.81
3	Greetings from Earth, Pt. 1	49.34
4	The Man With Nine Lives	49.28
5	Battlestar Galactica, Pt. 2	49.27

Join Types



- Inner join ดึงมาส่วนที่ key มันแมตได้ 100%
- Left join ดึงตารางซ้ายมารอเลย โรลไหนแมตค่าได้ให้ดึงมา ถ้าโรลไหนไม่ได้จะติดค่า Null
- Right join แค่สลับตารางกับ Left join
- Full join เอาทุกแถวเลยของทั้งสองตาราง ตารางมันจะเยอะขึ้น

มาเริ่มเขียนกัน....

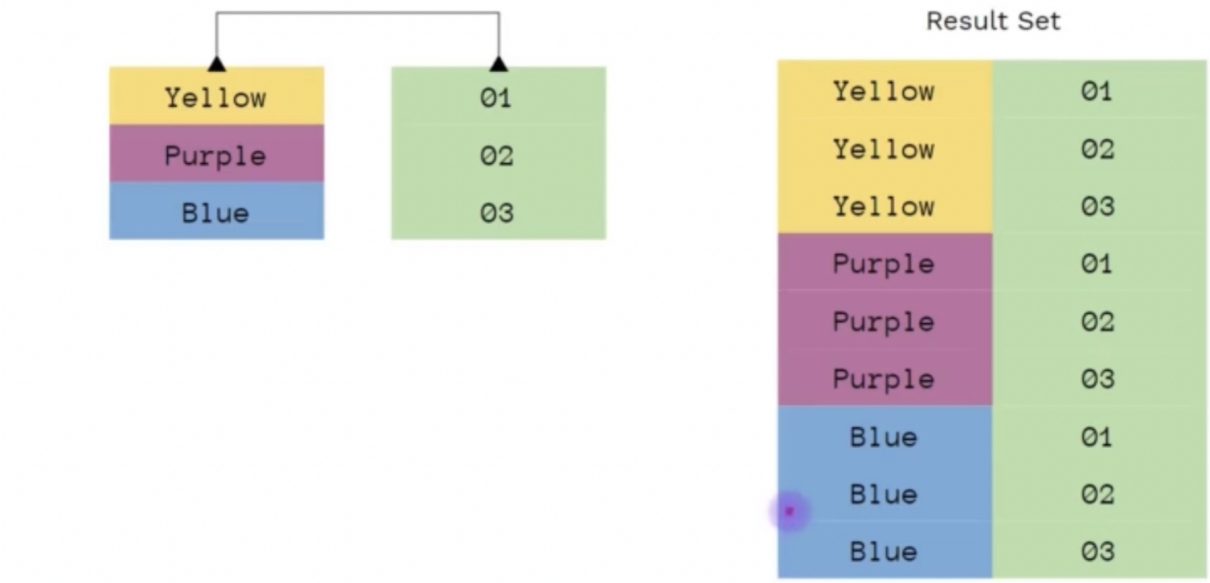
ถ้าจะเขียน `WHERE` ต่อก็ `WHERE ar.name = 'Aerosmith';`

```
SELECT
    ar.name     AS artist_name, -- table name.column name
    al.title    AS album_name,
    tr.name     AS track_name
FROM artists        AS ar
INNER JOIN albums   AS al
    ON ar.ArtistId = al.ArtistId -- pk=fk
```

```
INNER JOIN tracks   AS tr
    ON tr.AlbumId = al.AlbumId
```

	artist_name	album_name	track_name
1	AC/DC	For Those About To Rock We Salute You	For Those About To Rock (We Salute You)
2	AC/DC	For Those About To Rock We Salute You	Put The Finger On You
3	AC/DC	For Those About To Rock We Salute You	Let's Get It Up
4	AC/DC	For Those About To Rock We Salute You	Inject The Venom
5	AC/DC	For Those About To Rock We Salute You	Snowballed
6	AC/DC	For Those About To Rock We Salute You	Evil Walks
7	AC/DC	For Those About To Rock We Salute You	C.O.D.
8	AC/DC	For Those About To Rock We Salute You	Breaking The Rules

Cross join (aka. cartesian) คือการคูณกันของตาราง เช่น table ซ้ายมี 3 ค่า table ขวามี 3 ค่า resuit set คือ 9 แถว ไม่จำเป็นต้องมี primary key เพราะมัน cross กันหมดเลย เช่น เลขไพ่ กับ สัญลักษณ์ไพ่




```
SELECT * FROM ranks CROSS JOIN suits; --ranks and suits is table
```

Self Join คือการดึงข้อมูลในตารางเดียวกัน เช่นการดึงชื่อพนักงานกับ ตำแหน่ง

```
SELECT
    e1.name staff,
    e1.level staff_level,
    e2.name manager,
    e2.level manger_level,
    e1.name || 'reports to' || e2.name AS comment
FROM employee e1, employee e2
WHERE e1.manager_id = e2.id;
```

ต่อมาคือการเขียน Aggregate + การ Join

```
CREATE VIEW genre_stats AS
    SELECT
        ge.name,
        COUNT(*)          AS count_tracks,
        AVG(Milliseconds) AS avg_millisecounds
    FROM artists      AS ar
        INNER JOIN albums AS al ON ar.ArtistId = al.ArtistId
        INNER JOIN tracks AS tr ON tr.AlbumId = al.AlbumId
        INNER JOIN genres AS ge ON ge.GenreId = tr.GenreId
    GROUP BY 1
    ORDER BY 3 DESC
    LIMIT 5
```

	Name	count_tracks	avg_millisecounds
1	Sci Fi & Fantasy	26	2911783.03846154
2	Science Fiction	13	2625549.07692308
3	Drama	64	2575283.78125
4	TV Shows	93	2145041.02150538
5	Comedy	17	1585263.70588235

ซึ่งตัวอย่างนี้มีรูปแบบการเขียนคือ เราดึงประเภทเพลงขึ้นมา นับจำนวนโรล(เพลงร็อค, เพลงป๊อป มีกี่เพลง...) หาค่าเฉลี่ย และจัดกลุ่มตาม GROUP BY

#ต่อมาสำคัญน่ะ

ในความเป็นจริงข้อมูลมันอัปเดตตลอดเวลา เพราะฉนั้นเราจะต้องสร้าง ตารางที่เป็นแบบ **virtual table ⇒ view**

ถ้าสังเกตดีๆ คือบรรทัดบนสุด `CREATE VIEW genre_stats AS`

ทุกครั้งที่มีการอัปเดตข้อมูล ผู้ใช้ที่ใช้ตารางนี้จะได้รับข้อมูลที่ Up-to-date

หรือจะลบตารางก็ได้ `DROP VIEW genre_stats;`

Intersect & Except & Onion

- `intersect` คือ การหาว่า id ในตารางที่ 1 และ id ในตารางที่สองมีตัวไหนที่เหมือนกันบ้าง คล้ายกับ Inner join
- `Except` คือ การหาว่า id ในตารางที่ 1 ตัวไหนที่ไม่ได้อยู่ใน ตารางที่ 2
- `Union` คือ มันนจะลบแถวที่ซ้ำกัน remove duplicate หรือถ้าเราต้องการเก็บค่า duplicate ไว้ก็ ใช้ `UNION ALL`

```
SELECT id FROM book_shop
INTERSECT --หรือเปลี่ยนเป็น EXCEPT, UNION, UNION ALL ตรงนี้
SELECT id FROM favorite_book
```

Advance SQL

- Subqueries คือ การ breakdown our long query into steps

```
SELECT FirstName, Country
FROM (SELECT * FROM customers) AS sub
WHERE Country = 'United Kingdom'
```

SELECT ในวงเล็บเรียกว่า INNER QUERY คือมันจะเริ่มทำงานตรงนี้ก่อนเป็นลำดับแรก

- WITH : common table expression

```
WITH sub AS (SELECT * FROM customers)

SELECT FirstName, Country
FROM sub
WHERE Country = 'United Kingdom'
```

หลักการเหมือนกับ Subqueries และคำตอบก็เหมือนกัน แค่ลักษณะการเขียนต่างกัน

แต่!

ข้อดีของ WITH มันเจ๋งตรงนี้... example

> 😊 Query American customers who purchase our products in 2009-10(invoices)

```
-- basic query
SELECT
    firstname,
    lastname,
    email,
    COUNT(*) count_order
FROM customers c
JOIN invoices i ON c.customerid = i.customerid
```

```
WHERE c.country = 'USA' AND STRFTIME("%Y-%m", i.InvoiceDate) = '
GROUP BY 1,2,3

-- With cuase
WITH usa_customers AS (
    SELECT * FROM customers
    WHERE country = 'USA'
), invoice_2009 AS(
    SELECT * FROM invoices
    WHERE STRFTIME("%Y-%m", InvoiceDate) = "2009-10"
)

SELECT FirstName, LastName, Email, COUNT(*)
FROM usa_customers t1
JOIN invoice_2009      t2
ON t1.CustomerId = t2.CustomerId
GROUP BY 1,2,3

-- sub query
SELECT FirstName, LastName, Email, COUNT(*)
FROM (
    SELECT * FROM customers
    WHERE country = 'USA'
) AS t1
JOIN (
    SELECT * FROM invoices
    WHERE STRFTIME("%Y-%m", InvoiceDate) = "2009-10"
) AS t2
ON t1.CustomerId = t2.CustomerId
GROUP BY 1,2,3
```

ทุกคำตอบจะเหมือนกันหมด....

	FirstName	LastName	Email	COUNT(*)
1	Victor	Stevens	vstevens@yahoo.com	1

Homework Project