



Sprint 02 - SQL for Data Analyst 102

☰ Tools	SQL
📅 Date	@November 8, 2023
⚙️ Status	Done

▼ EP01 - Filter Data Part 1

```
-- customers ที่ ชื่อ John
SELECT * FROM customers
where firstname = 'John';

-- customers ใน USA
SELECT * FROM customers
where country = 'USA';

-- customers ที่อยู่ใน USA and รัฐ(state):CA
SELECT * FROM customers
where country = 'USA' and state = 'CA';

-- customers ที่อยู่ใน USA or Canada
SELECT * FROM customers
where country = 'USA' or country = 'Canada';

-- customers ที่ ไม่อยู่ NOT() ใน USA or Canada or France
SELECT * FROM customers
where NOT(country = 'USA' or
           country = 'Canada' or
           country = 'France');
```

▼ EP02 - Filter Data Part 2

```

-- customers ที่ ไม่อยู่ NOT IN() ใน 3 ประเทศ นี้
SELECT * FROM customers
WHERE country NOT IN ('Brazil' , 'Germany' , 'Norway');

-- customers ที่ customerid ตั้งแต่ 5 - 10
วิธีที่ 1
SELECT * FROM customers
WHERE customerid BETWEEN 5 and 10;
วิธีที่ 2
SELECT * FROM customers
WHERE customerid >= 5 and customerid <=10;

-- ดู transaction ตั้งแต่ 2009-01-01 ถึงจนถึง - 2009-02-20 2กลุ่ม
SELECT invoicedate FROM invoices
WHERE invoicedate BETWEEN '2009-01-01 00:00:00' AND '2009-02-20 00:00:00';

-- B2C
SELECT * FROM customers
WHERE COMPANY IS NULL;

-- CORPREATE
SELECT * FROM customers
WHERE COMPANY IS not NULL;

```

▼ EP03 - Filter Data Part 3

```

--pattern matching
-- email ลูกค้า ที่ ลงท้ายด้วย @hotmail.com
select firstname, lastname, country, email, phone from cus
where email like '%@hotmail.com'; -- wildcard %*/

-- email ลูกค้า ที่ ไม่ ลงท้ายด้วย @hotmail.com
select firstname, lastname, country, email, phone from cus
where email NOT like '%@hotmail.com';

-- เบอร์โทร (phone) ที่มีเลข 9 : 2 ตัวติดกัน
select firstname, lastname, country, email, phone from cus
where phone like '%99%';

```

```
-- ใช้ _ matching ชื่อลูกค้า Leonie
select firstname, lastname, country, email, phone from cus
where firstname like 'Leon__'; -- %, _
```



Wildcard (%, _)

% match character ที่ตัวก็ได้ หรือ ไม่matchเลยก็ได้

_ match ได้เพียง 1 character เท่านั้น

▼ EP04 - COALESCE (การจัดการค่า Null)

COALESCE() = คำสั่ง replace ค่า Null

CASE WHEN = การสร้างเงื่อนไข

```
-- NULL/ missing valuesalbums
-- corporate
select * from customers
where company is not NULL;

select
    company,
    COALESCE(company, 'End Customer') as 'Company Clean',
    case when company is NULL then 'End Customer'
        else 'Corporate'
    end as 'Company Clean 2'
from customers;
```

Company	Company Clean	Company Clean 2
Embraer - Empresa Brasileira de Ae...	Embraer - Empresa Brasileira de Ae...	Corporate
NULL	End Customer	End Customer
NULL	End Customer	End Customer
NULL	End Customer	End Customer

▼ EP05 - JOIN Data using WHERE

```
-- join table with where clause
select * from artists, albums
where artists.artistid = albums.artistid;

                                VS

select * from artists join albums
ON artists.artistid = albums.artistid;

-- join table artists, albums, tracks
select
    artists.artistid,
    artists.name as artists_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
    AND artists.artistid IN (8, 100, 120);
    /*and artists.artistid = 100;*/
```

Artistid	artists_name	album_name	song_name
8	Audioslave	Audioslave	Cochise
8	Audioslave	Audioslave	Show Me How to Live
8	Audioslave	Audioslave	Gasoline
8	Audioslave	Audioslave	What You Are
8	Audioslave	Audioslave	Like a Stone
8	Audioslave	Audioslave	Set It Off

▼ EP06 - Aggregate Functions

```
-- Aggregate Functions
select
    ROUND(AVG(milliseconds), 2) as avg_mill,
```

```
SUM(milliseconds) as sum_mill,
MIN(milliseconds) as min_mill,
MAX(milliseconds) as max_mill,
COUNT(milliseconds) as count_mill
from tracks
```



avg sum min max count ไม่สนใจค่า Null

▼ EP07 - COUNT Distinct (การนับแบบไม่ซ้ำ)

```
select
    COUNT(DISTINCT country), -- ลูกค้ามีทั้งหมด 24 ประเทศ
    count(*) -- มีลูกค้าทั้งหมด 59 คน
from customers;
```

COUNT(DISTINCT country)	count(*)
24	59

▼ EP08 - GROUP BY

ใช้ **GROUP BY** ร่วมกับ **Aggregate Functions** ⇒ ในการแบ่งข้อมูลตามกลุ่มที่ต้องการ

```
-- GROUP BY ร่วมกับ Aggregate Functions

-- นับจำนวนลูกค้าแต่ละประเทศ
select
    country,
    count(*) as count_country
from customers
GROUP by country;

-- นับจำนวน songs ในแต่ละ genres
-- โดยต้อง join table genres, tracks ด้วย where
select
    genres.name,
```

```

        count(*) as count_songs
    from genres, tracks
    where genres.genreid = tracks.genreid
    group by genres.name;

```

Name	count_songs
Alternative	40
Alternative & Punk	332
Blues	81
Bossa Nova	15
Classical	74

▼ EP09 - HAVING

HAVING = ใช้ filter กลุ่ม ที่ผ่านการ group by มาแล้ว และมีความต่างกับ where clause ตรงที่ having เกิดหลัง group by ส่วน where เกิดขึ้นก่อน group by

WHERE = ใช้ filter table

```

-- Having
select
    genres.name,
    count(*) as count_songs
from genres, tracks
where genres.genreid = tracks.genreid
    and genres.name <> 'Rock'
    -- filter Rock song ออกไปก่อนแล้ว
group by genres.name
having count(*) >= 100;

```

▼ EP10 - ORDER BY

```

-- Order By + Limit
SELECT
    genres.name,

```

```
count(*)
from genres
join tracks on genres.genreid = tracks.genreid
GROUP by genres.name
ORDER by COUNT(*) desc
limit 5;
```

VS

```
-- au Comma + เปลี่ยน where เป็น on
SELECT
    genres.name,
    count(*)
from genres , tracks
where genres.genreid = tracks.genreid
GROUP by genres.name
ORDER by COUNT(*) desc -- descending order
limit 5;
```

ผลลัพธ์เหมือนกัน

Name	count(*)
Rock	1297
Latin	579
Metal	374
Alternative & Punk	332
Jazz	130