



SQL Statement Fundamentals Lecture

SELECT

SELECT  **FROM** table_1
Database

Table 1	Table 2	Table 3																																				
<table><tr><th>c1</th><th>c2</th><th>c3</th></tr><tr><td>x</td><td>23</td><td>a</td></tr><tr><td>y</td><td>18</td><td>b</td></tr><tr><td>z</td><td>46</td><td>c</td></tr></table>	c1	c2	c3	x	23	a	y	18	b	z	46	c	<table><tr><th>c1</th><th>c2</th><th>c3</th></tr><tr><td>1</td><td>Q</td><td>13</td></tr><tr><td>2</td><td>R</td><td>34</td></tr><tr><td>3</td><td>S</td><td>56</td></tr></table>	c1	c2	c3	1	Q	13	2	R	34	3	S	56	<table><tr><th>c1</th><th>c2</th><th>c3</th></tr><tr><td>c</td><td>0</td><td>12</td></tr><tr><td>b</td><td>0</td><td>24</td></tr><tr><td>b</td><td>1</td><td>45</td></tr></table>	c1	c2	c3	c	0	12	b	0	24	b	1	45
c1	c2	c3																																				
x	23	a																																				
y	18	b																																				
z	46	c																																				
c1	c2	c3																																				
1	Q	13																																				
2	R	34																																				
3	S	56																																				
c1	c2	c3																																				
c	0	12																																				
b	0	24																																				
b	1	45																																				

SELECT c1, c3 **FROM** table_1

Database

Table 1	Table 2	Table 3																																				
<table><tr><th>c1</th><th>c2</th><th>c3</th></tr><tr><td>x</td><td>23</td><td>a</td></tr><tr><td>y</td><td>18</td><td>b</td></tr><tr><td>z</td><td>46</td><td>c</td></tr></table>	c1	c2	c3	x	23	a	y	18	b	z	46	c	<table><tr><th>c1</th><th>c2</th><th>c3</th></tr><tr><td>1</td><td>Q</td><td>13</td></tr><tr><td>2</td><td>R</td><td>34</td></tr><tr><td>3</td><td>S</td><td>56</td></tr></table>	c1	c2	c3	1	Q	13	2	R	34	3	S	56	<table><tr><th>c1</th><th>c2</th><th>c3</th></tr><tr><td>c</td><td>0</td><td>12</td></tr><tr><td>b</td><td>0</td><td>24</td></tr><tr><td>b</td><td>1</td><td>45</td></tr></table>	c1	c2	c3	c	0	12	b	0	24	b	1	45
c1	c2	c3																																				
x	23	a																																				
y	18	b																																				
z	46	c																																				
c1	c2	c3																																				
1	Q	13																																				
2	R	34																																				
3	S	56																																				
c1	c2	c3																																				
c	0	12																																				
b	0	24																																				
b	1	45																																				

Challenge → SELECT

Situation

- เราต้องการส่งอีเมลส่งเสริมการขายให้กับลูกค้าปัจจุบันของเรา!

Challenge

- ใช้คำสั่ง SELECT เพื่อดึงชื่อและนามสกุลของลูกค้าทุกรายและที่อยู่อีเมลของพวกเขา

Solution

```
SELECT first_name, last_name, email FROM customer;
```

	first_name character varying (45) 🔒	last_name character varying (45) 🔒	email character varying (50) 🔒
1	Jared	Ely	jared.ely@sakilacustomer.org
2	Mary	Smith	mary.smith@sakilacustomer.org
3	Patricia	Johnson	patricia.johnson@sakilacustomer.org
4	Linda	Williams	linda.williams@sakilacustomer.org
5	Barbara	Jones	barbara.jones@sakilacustomer.org
6	Elizabeth	Brown	elizabeth.brown@sakilacustomer.org
7	Jennifer	Davis	jennifer.davis@sakilacustomer.org
8	Maria	Miller	maria.miller@sakilacustomer.org
9	Susan	Wilson	susan.wilson@sakilacustomer.org
Total rows: 599 of 599		Query complete 00:00:00.043	

SELECT DISTINCT

- SELECT DISTINCT name FROM color_table

Name	Choice
Zach	Green
David	Green
Claire	Yellow
David	Red

```
#SELECT DISTINCT การเลือกข้อมูลที่ไม่ซ้ำกัน คือดึงตัวขึ้นมาแค่ตัวเดียว
SELECT DISTINCT (release_year) FROM film;
SELECT DISTINCT (rental_rate) FROM film;
SELECT DISTINCT (rating) FROM film ORDER BY rating DESC;
```

Challenge → SELECT DISTINCT

Situation

- นักท่องเที่ยวชาวออสเตรเลียไม่คุ้นเคยกับการจัดเรตภาพยนตร์ของ MPAA (p.g. PG , PG-13, R, etc...)
- เราต้องการทราบประเภทของการให้คะแนนที่เรามีในฐานข้อมูลของเรา
- เรามีคะแนนอะไรบ้าง?

Challenge

- ใช้สิ่งที่คุณได้เรียนรู้เกี่ยวกับ SELECT DISTINCT เพื่อดึงข้อมูลประเภทการให้คะแนนที่แตกต่างกันซึ่งภาพยนตร์ของเราอาจมีในฐานข้อมูลของเรา

Solution

```
SELECT DISTINCT (rating) FROM film ORDER BY rating DESC;
```

Data Output		Messages	Graph Visualiser	×	Notifications
	rating mpaa_rating				
1	NC-17				
2	R				
3	PG-13				
4	PG				
5	G				

COUNT

- `SELECT COUNT(name) FROM table;`

Name	Choice
Zach	Green
David	Green
Claire	Yellow
David	Red

```
# โดยปกติการนับแถวในคอลัมน์มักจะใช้ COUNT(*)
SELECT COUNT(*) FROM payment;
# หากมี Query ที่เฉพาะเจาะจงไปที่ amount ก็สามารถเขียนได้
SELECT COUNT(amount) FROM payment;
# แต่ผลลัพธ์จะเท่ากับ COUNT(*) เนื่องจากตารางต้องมีแถวเท่ากันในทุกคอลัมน์
# ปัญหาคือหากเราต้องการนับตัวที่ไม่ซ้ำกันเลย เราต้องใช้ DISTINCT เข้ามาช่วย
```

- `SELECT COUNT(DISTINCT name) FROM table;`

Name	Choice
Zach	Green
David	Green
Claire	Yellow
David	Red

```
# หากต้องการนับตัวที่ไม่ซ้ำกันเลย DISTINCT เข้ามาช่วย
SELECT COUNT(DISTINCT(amount)) FROM payment;
```

SELECT WHERE

- Comparison Operators

Operator	Description
=	Equal
>	Greater than
<	Less Than
>=	Greater than or equal to
<=	Less than or equal to
<> or !=	Not equal to

- Logical Operators
 - Allow us to combine multiple comparison operators
 - **AND**
 - **OR**
 - **NOT**

- **SELECT** name,choice **FROM** table
WHERE name = 'David'

Name	Choice
David	Green
David	Red

- **SELECT** name,choice **FROM** table
WHERE name = 'David' **AND** choice= 'Red'

Name	Choice
David	Red

```
SELECT * FROM customer
WHERE first_name = 'Jared';
```

```
SELECT * FROM film
WHERE rental_rate > 4
      AND replacement_cost >= 19.99;
```

```
SELECT title FROM film
WHERE rental_rate > 4
      AND replacement_cost >= 19.99
      AND rating = 'R';
```

```
# เราสามารถใช้ COUNT นับหลังจากทำการ filter แล้วได้
SELECT COUNT(title) FROM film -- ใช้ COUNT(*) ได้ผลลัพธ์เหมือนกัน
WHERE rental_rate > 4
      AND replacement_cost >= 19.99
      AND rating = 'R';
```

Challenge → SELECT WHERE

Challenge No. 1

- ### Solution

The screenshot shows the 'Data Output' tab selected in the top navigation bar. Below the navigation bar is a toolbar with icons for file operations (add, save, expand, clipboard, trash, database, download, and line graph). Below the toolbar is a table with two rows. The first row has an empty index column, a cell containing 'email' and 'character varying (50)' with a lock icon, and an empty action column. The second row has an index of '1' and a cell containing the email address 'nancy.thomas@sakilacustomer.org'.

- ลูกคำต้องการทราบว่าภาพยนตร์เรื่อง “Outlaw Hanky” เป็นเรื่องเกี่ยวกับอะไร
- คุณช่วยอธิบายเกี่ยวกับภาพยนตร์เรื่อง “Outlaw Hanky” ให้พวกเขาฟังได้ไหม?

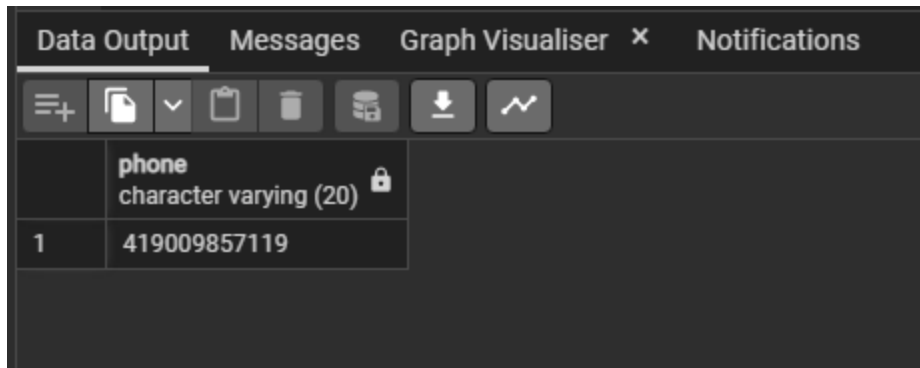
Solution

Challenge No. 3

- ลูกค้าส่งคืนมาสายและเราได้ส่งจดหมายไปยังที่อยู่ของพวกเขาที่ '259 Ipoh Drive'
- เราควรโทรหาพวกเขาทางโทรศัพท์เพื่อแจ้งให้ทราบ
- หมายเลขโทรศัพท์ของลูกค้าที่อาศัยอยู่ที่ '259 Ipoh Drive' ได้ไหม?

Solution

```
SELECT phone FROM address  
WHERE address = '259 Ipoh Drive';
```



	phone character varying (20) 🔒
1	419009857119

ORDER BY

- **SELECT** company,name,sales **FROM** table
ORDER BY company,sales

Company	Name	Sales
Apple	Andrew	100
Apple	Zach	300
Google	Claire	200
Google	David	500
Xerox	Steven	100



ASC น้อยไปมาก , **DESC** มากไปน้อย

```
SELECT store_id, first_name, last_name FROM customer
ORDER BY store_id DESC , first_name ASC;
```

LIMIT

```
SELECT store_id, first_name, last_name FROM customer
ORDER BY store_id DESC
LIMIT 5;
```

Challenge → ODER BY

Challenge Task 1

- เราต้องการให้รางวัลลูกค้าที่ชำระเงิน 10 คนแรกของเรา
- รหัสลูกค้าของลูกค้า 10 รายแรกที่ชำระเงินคืออะไร?

Solution

```
SELECT customer_id FROM payment
ORDER BY payment_date ASC
LIMIT 10;
```

Data Output			Messages	Graph Visualiser	×	Notifications
	customer_id					
	smallint					
1		416				
2		516				
3		239				
4		592				
5		49				
6		264				
7		46				
8		481				
9		139				
10		595				

Challenge Task 2

- ลูกค้าต้องการเช่าวิดีโออย่างรวดเร็วเพื่อดูในช่วงพักเที่ยงสั้นๆ
- ภาพยนตร์ 5 เรื่องที่สั้นที่สุด (ตามความยาวรันไทม์) มีชื่อว่าอะไร

Solution

```
SELECT title, length FROM film
ORDER BY length ASC
LIMIT 5;
```

	title	length
	character varying (255)	smallint
1	Labyrinth League	46
2	Alien Center	46
3	Iron Moon	46
4	Kwai Homeward	46
5	Ridgemont Submarine	46

Quick Bonus Question

- หากลูกค้าคนก่อนสามารถชมภาพยนตร์ที่มีความยาวไม่เกิน 50 นาที เธอมีตัวเลือกกี่ตัวเลือก

Solution

```
SELECT COUNT(title) FROM film
WHERE length <= 50;
```

	count
	bigint
1	37

BETWEEN

- The **BETWEEN** operator is the same as:
 - value \geq low AND value \leq high
 - value BETWEEN low AND high
- You can also combine **BETWEEN** with the **NOT** logical operator:
 - value NOT BETWEEN low AND high
- The **NOT BETWEEN** operator is the same as:
 - value $<$ low OR value $>$ high
 - value NOT BETWEEN low AND high
- The **BETWEEN** operator can also be used with dates. Note that you need to format dates in the ISO 8601 standard format, which is YYYY-MM-DD
 - date BETWEEN '2007-01-01' AND '2007-02-01'

```
# BETWEEN
SELECT COUNT(*) FROM payment
WHERE amount BETWEEN 8 AND 9;
```

```
# NOT BETWEEN
SELECT COUNT(*) FROM payment
WHERE amount NOT BETWEEN 8 AND 9;

SELECT * FROM payment
WHERE payment_date BETWEEN '2007-02-01' AND '2007-02-15';
```

IN

- The general syntax is:
 - value **IN** (option1,option2,...,option_n)
- Example query:
 - SELECT color FROM table
WHERE color IN ('red','blue','green')
- Example query:
 - SELECT color FROM table
WHERE color NOT IN ('red','blue')

```
SELECT * FROM payment
WHERE amount IN (0.99, 1.98, 1.99);

SELECT * FROM customer
WHERE first_name IN ('John', 'Jake', 'Julie');

SELECT * FROM customer
WHERE first_name NOT IN ('John', 'Jake', 'Julie');
```

LIKE and ILIKE

ตัวดำเนินการ LIKE ช่วยให้เราสามารถจับคู่รูปแบบกับข้อมูลสตริงโดยใช้อักขระตัวแทน:

- Percent % จับคู่ลำดับของอักขระใดก็ได้

ชื่อทั้งหมดที่ขึ้นต้นด้วย 'A'

→ WHERE name LIKE 'A%'

ชื่อทั้งหมดที่ลงท้ายด้วย 'a'

→ WHERE name LIKE '%a'

- Underscore _ จับคู่อักขระตัวเดียว

การใช้เครื่องหมายขีดล่างช่วยให้เราสามารถแทนที่อักขระเพียงตัวเดียวได้

รับภาพยนตร์ Mission Impossible ทั้งหมด

→ WHERE title LIKE 'Mission Impossible _'

คุณสามารถใช้ขีดล่างได้หลายอัน

ลองนึกภาพว่าเรามีรหัสสตริงเวอร์ชันในรูปแบบ 'Version#A4' , 'Version#B7' ฯลฯ...

→ WHERE value LIKE 'Version#__'

- เรายังสามารถรวมตัวดำเนินการจับคู่รูปแบบเพื่อสร้างรูปแบบที่ซับซ้อนมากขึ้น

→ WHERE name LIKE '_her%'

- Cheryl
- Theresa
- Sherri

- WHERE name LIKE '_her%'
 - Cheryl
 - Theresa
 - Sherri




ขอให้สังเกตว่า LIKE คำนึงถึงตัวพิมพ์เล็กและใหญ่ เราสามารถใช้ ILIKE ซึ่งไม่คำนึงถึงตัวพิมพ์เล็กและใหญ่



ในตอนนี้เรามุ่งเน้นไปที่ LIKE และ ILIKE ในตอนนี้ แต่โปรดทราบว่า PostgreSQL รองรับความสามารถของ regex อย่างเต็มรูปแบบ:

9.7. Pattern Matching

There are three separate approaches to pattern matching provided by PostgreSQL: the traditional SQL LIKE operator, the more recent SIMILAR TO operator

 <https://www.postgresql.org/docs/12/functions-matching.html>



```
SELECT * FROM customer
WHERE first_name LIKE 'J%';

#ILIKE พิมพ์เล็กพิมพ์ใหญ่ไม่มีผล
SELECT * FROM customer
WHERE first_name ILIKE 'J%' AND last_name ILIKE 's%';

SELECT * FROM customer
WHERE first_name NOT LIKE '_her%';

SELECT * FROM customer
WHERE first_name LIKE 'A%' AND last_name NOT LIKE 'B%'
ORDER BY last_name;
```

Challenge!

- มีธุรกรรมการชำระเงินที่มากกว่า \$5.00 ที่รายการ

```
SELECT COUNT(amount) FROM payment
WHERE amount > 5.00;
--> 3618
```

- มีนักแสดงกี่คนที่มียี่ห้อขึ้นต้นด้วยตัวอักษร P?

```
SELECT COUNT(*) FROM actor
WHERE first_name LIKE 'P%';
--> 5
```


- ลูกค้าของเรามาจากกี่อำเภอที่ไม่ซ้ำกัน?

```
SELECT COUNT(DISTINCT (district)) FROM address;  
--> 378
```

- ดึงรายชื่อสำหรับเขตที่ชัดเจนเหล่านั้นจากคำถามก่อนหน้านี้

```
SELECT DISTINCT (district) FROM address;  
--> 378
```

- มีภาพยนตร์กี่เรื่องที่มีเรต R และ replacement cost ระหว่าง 5 ถึง 15 ดอลลาร์

```
SELECT replacement_cost, rating FROM film  
WHERE AND rating = 'R'  
      replacement_cost BETWEEN 5 AND 15;  
--> 52
```

- มีภาพยนตร์กี่เรื่องที่มีคำว่า Truman อยู่ในชื่อเรื่อง?

```
SELECT * FROM film  
WHERE title ILIKE '%truman%';  
--> 5
```