



JOINS Lecture

Welcome to the JOINS Section!

แหล่งข้อมูลเพิ่มเติมหลังจากทบทวนบทเรียนแล้ว:

[SQL JOINS Explained with Venn Diagrams](#)

[SQL JOIN Examples](#)

[Wikipedia Page on SQL JOINS](#)

AS → นามแฝง (Alias)

Query Editor		Query History	
1	SELECT	amount	AS rental_price
2	FROM	payment;	

Data Output		Explain	Messages	Notifications
	rental_price numeric (5,2)			
1	7.99			
2	1.99			

Query Editor		Query History	
1	SELECT	SUM(amount)	AS net_revenue
2	FROM	payment;	

Data Output		Explain	Messages	Notifications
	net_revenue numeric			
1	61312.04			

ตัวดำเนินการ AS ถูกดำเนินการที่ส่วนท้ายสุดของ Query หมายความว่าเราไม่สามารถใช้ ALIAS ภายใน WHERE หรือ GROUP BY ได้

INNER JOIN

JOIN คืออะไร?

JOIN ช่วยให้เราสามารถรวมตารางหลายตารางเข้าด้วยกัน

เหตุผลหลักที่ JOIN มีหลายประเภท คือการตัดสินใจว่าจะจัดการกับข้อมูลที่แสดงในตารางอย่างไร

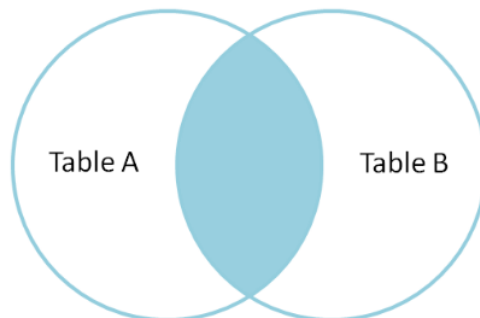
- **INNER JOIN** → ดึงเฉพาะตัวที่ match

An INNER JOIN will result with the set of records that **match in both** tables.

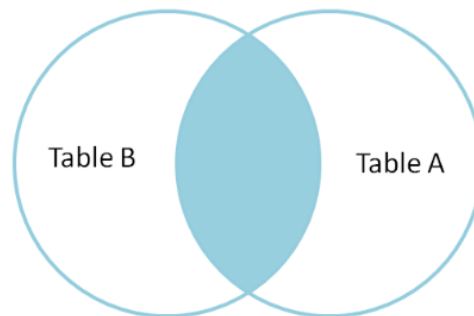
REGISTRATIONS	
reg_id	name
1	Andrew
2	Bob
3	Charlie
4	David

LOGINS	
log_id	name
1	Xavier
2	Andrew
3	Yolanda
4	Bob

SELECT * FROM **TableA**
INNER JOIN **TableB**
ON **TableA**.col_match = **TableB**.col_match

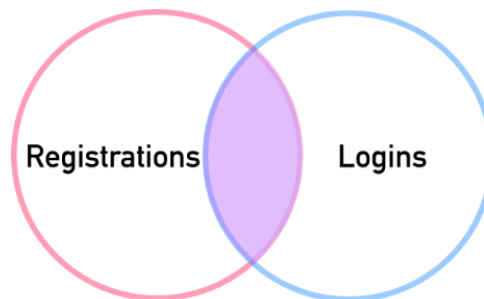


```
SELECT * FROM TableB
INNER JOIN TableA
ON TableA.col_match = TableB.col_match
```



- SELECT * FROM Registrations
INNER JOIN Logins
ON Registrations.name = Logins.name

REGISTRATIONS	
reg_id	name
1	Andrew
2	Bob
3	Charlie
4	David



LOGINS	
log_id	name
1	Xavier
2	Andrew
3	Yolanda
4	Bob

- `SELECT * FROM Registrations
INNER JOIN Logins
ON Registrations.name = Logins.name`

REGISTRATIONS	
reg_id	name
1	Andrew
2	Bob
3	Charlie
4	David

RESULTS			
reg_id	name	log_id	name
1	Andrew	2	Andrew
2	Bob	4	Bob

LOGINS	
log_id	name
1	Xavier
2	Andrew
3	Yolanda
4	Bob

PIERIAN DATA

`SELECT reg_id, Logins.name, log_id
FROM Registrations
INNER JOIN Logins
ON Registrations.name = Logins.name`

RESULTS		
reg_id	name	log_id
1	Andrew	2
2	Bob	4



โปรดจำไว้ว่าลำดับของตารางไม่สำคัญใน INNER JOIN
นอกจากนี้ หากคุณเห็นเพียง JOIN โดยไม่มีคำว่า **INNER** บน PostgreSQL จะถือว่าเป็น INNER JOIN

```
SELECT payment_id, payment.customer_id, first_name
FROM payment
INNER JOIN customer
ON payment.customer_id = customer.customer_id; -- PK = FK
```

OUTER JOINS

OUTER JOIN มีหลายประเภทที่แตกต่างกัน

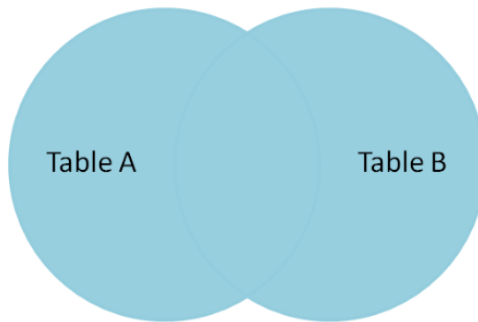
ซึ่งจะช่วยให้เราบริหารจัดการกับค่าที่มีอยู่ในตารางใดตารางหนึ่งที่กำลังเข้าร่วมเท่านั้น
นี่คือ JOIN ที่ซับซ้อนมากขึ้น ใช้เวลาของคุณในการพยายามทำความเข้าใจ!

In these lectures we will explain:

- FULL OUTER JOIN
 - Clarifying WHERE null
- LEFT OUTER JOIN
 - Clarifying WHERE null
- RIGHT OUTER JOIN
 - Clarifying WHERE null

FULL OUTER JOIN

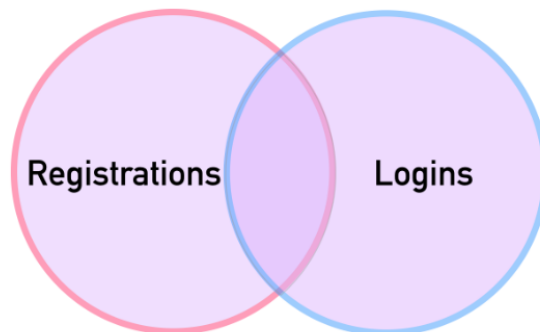
- SELECT * FROM **TableA**
FULL OUTER JOIN **TableB**
ON **TableA**.col_match = **TableB**.col_match



PIERIAN  DATA

- SELECT * FROM Registrations
FULL OUTER JOIN Logins
ON Registrations.name = Logins.name

REGISTRATIONS	
reg_id	name
1	Andrew
2	Bob
3	Charlie
4	David



LOGINS	
log_id	name
1	Xavier
2	Andrew
3	Yolanda
4	Bob

PIERIAN  DATA

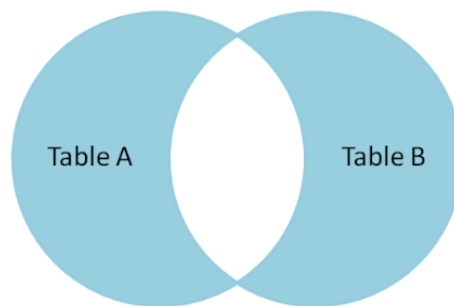
- SELECT * FROM Registrations FULL OUTER JOIN Logins
ON Registrations.name = Logins.name

REGISTRATIONS		RESULTS				LOGINS	
reg_id	name	reg_id	name	log_id	name	log_id	name
1	Andrew	1	Andrew	2	Andrew	1	Xavier
2	Bob	2	Bob	4	Bob	2	Andrew
3	Charlie	3	Charlie	null	null	3	Yolanda
4	David	4	David	null	null	4	Bob
		null	null	1	Xavier		
		null	null	3	Yolanda		

FULL OUTER JOIN with WHERE



- SELECT * FROM **TableA**
FULL OUTER JOIN **TableB**
ON **TableA**.col_match = **TableB**.col_match
WHERE **TableA**.id IS null OR
TableB.id IS null



- `SELECT * FROM Registrations FULL OUTER JOIN Logins
ON Registrations.name = Logins.name
WHERE Registrations.reg_id IS null OR
Logins.log_id IS null`

REGISTRATIONS		RESULTS				LOGINS	
reg_id	name	reg_id	name	log_id	name	log_id	name
1	Andrew	3	Charlie	null	null	1	Xavier
2	Bob	4	David	null	null	2	Andrew
3	Charlie	null	null	1	Xavier	3	Yolanda
4	David	null	null	3	Yolanda	4	Bob

- หมายถึงเรากำลังร่วมตารางแบบ Full JOIN โดยใช้ WHERE เพื่อมาหาแถวที่ไม่ซ้ำ

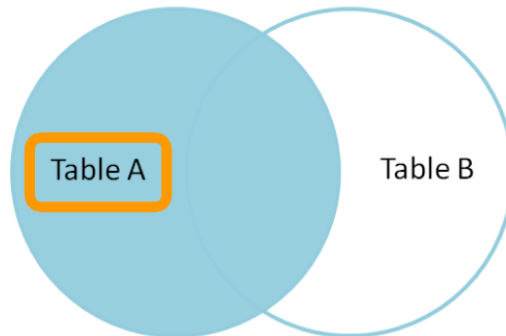
```
SELECT *
FROM customer
FULL OUTER JOIN payment
ON customer.customer_id = payment.customer_id
WHERE customer.customer_id IS null
OR payment.payment_id IS null;
```

LEFT OUTER JOIN



**ORDER
MATTERS FOR
LEFT OUTER JOIN!**

- `SELECT * FROM TableA
LEFT OUTER JOIN TableB
ON TableA.col_match = TableB.col_match`



PIERIAN  DATA

- `SELECT * FROM Registrations
LEFT OUTER JOIN Logins
ON Registrations.name = Logins.name`

REGISTRATIONS	
reg_id	name
1	Andrew
2	Bob
3	Charlie
4	David

RESULTS			
reg_id	name	log_id	name
1	Andrew	2	Andrew
2	Bob	4	Bob
3	Charlie	null	null
4	David	null	null

LOGINS	
log_id	name
1	Xavier
2	Andrew
3	Yolanda
4	Bob

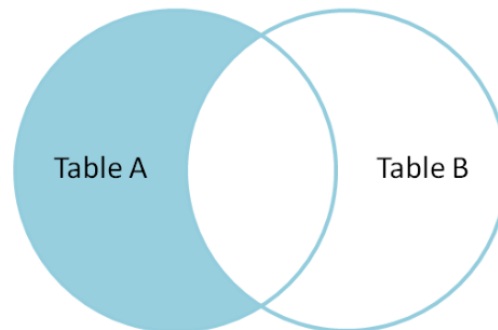
PIERIAN  DATA

LEFT OUTER JOIN With WHERE

- จะเป็นอย่างไรหากเราต้องการเฉพาะรายการที่ไม่ซ้ำกับตาราง A แถวเหล่านั้นที่พบในตาราง A และไม่พบในตาราง B



- SELECT * FROM **TableA**
LEFT OUTER JOIN **TableB**
ON **TableA**.col_match = **TableB**.col_match
WHERE **TableB**.id IS null



PIERIAN  DATA



- SELECT * FROM Registrations
LEFT OUTER JOIN Logins
ON Registrations.name = Logins.name

REGISTRATIONS	
reg_id	name
1	Andrew
2	Bob
3	Charlie
4	David

RESULTS			
reg_id	name	log_id	name
1	Andrew	2	Andrew
2	Bob	4	Bob
3	Charlie	null	null
4	David	null	null

LOGINS	
log_id	name
1	Xavier
2	Andrew
3	Yolanda
4	Bob

PIERIAN  DATA



```
SELECT * FROM Registrations  
LEFT OUTER JOIN Logins  
ON Registrations.name = Logins.name  
WHERE Logins.log_id IS null
```

REGISTRATIONS	
reg_id	name
1	Andrew
2	Bob
3	Charlie
4	David

RESULTS			
reg_id	name	log_id	name
3	Charlie	null	null
4	David	null	null

LOGINS	
log_id	name
1	Xavier
2	Andrew
3	Yolanda
4	Bob

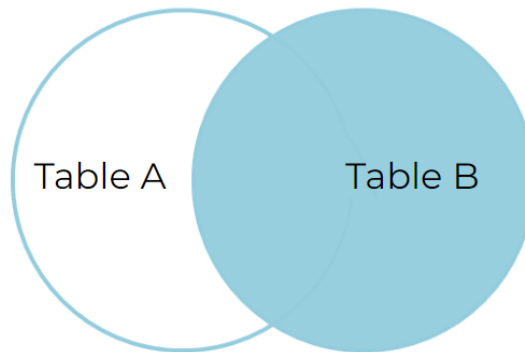
PIERIAN  DATA

```
SELECT film.film_id, title, inventory_id, store_id  
FROM film  
LEFT JOIN inventory  
ON inventory.film_id = film.film_id  
WHERE inventory.film_id IS NULL;
```

RIGHT JOINS

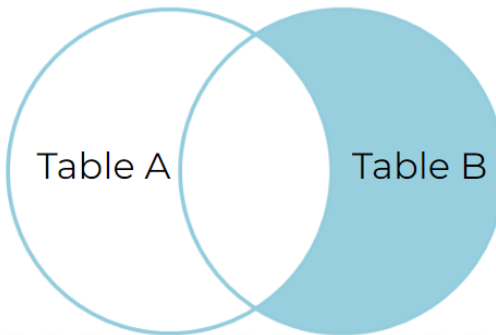
คล้าย LEFT JOIN แต่สลับตาราง

- SELECT * FROM **TableA**
RIGHT JOIN **TableB**
ON **TableA**.col_match = **TableB**.col_match



PIERIAN  DATA

- SELECT * FROM **TableA**
RIGHT OUTER JOIN **TableB**
ON **TableA**.col_match = **TableB**.col_match
WHERE **TableA**.id IS null



PIERIAN  DATA

UNIONS

การรวม 2 ตารางเข้าด้วยกัน

Let's explore a **UNION** with two example tables.

Sales2021_Q1	
name	amount
David	100
Claire	50

Sales2021_Q2	
name	amount
David	200
Claire	100

```
SELECT * FROM Sales2021_Q1  
UNION  
SELECT * FROM Sales2021_Q2;
```

name	amount
David	100
Claire	50
David	200
Claire	100

```
SELECT * FROM Sales2021_Q1
UNION
SELECT * FROM Sales2021_Q2
ORDER BY name;
```

name	amount
David	100
David	200
Claire	50
Claire	100

JOIN Challenges

- กฎหมายภาษีการขายของรัฐแคลิฟอร์เนียมีการเปลี่ยนแปลง และเราจำเป็นต้องแจ้งเตือนลูกค้าของเราเกี่ยวกับเรื่องนี้ผ่านทางอีเมล
- อีเมลของลูกค้าที่อาศัยอยู่ในแคลิฟอร์เนียคืออะไร

```
SELECT district, customer.email
FROM customer
INNER JOIN address
ON address.address_id = customer.address_id
WHERE district = 'California';
```

```
#Answer
SELECT district,email FROM address
INNER JOIN customer ON
address.address_id = customer.address_id
WHERE district = 'California';
```

Data Output		Messages	Notifications
<div> <div>≡ +</div> <div>📄</div> <div>▼</div> <div>📋</div> <div>🗑️</div> <div>🔄</div> <div>⬇️</div> <div>📈</div> </div>			
	district character varying (20) 🔒	email character varying (50) 🔒	
1	California	patricia.johnson@sakilacustomer.org	
2	California	betty.white@sakilacustomer.org	
3	California	alice.stewart@sakilacustomer.org	
4	California	rosa.reynolds@sakilacustomer.org	
5	California	renee.lane@sakilacustomer.org	
6	California	kristin.johnston@sakilacustomer.org	
Total rows: 9 of 9		Query complete 00:00:00.039	

- ลูกค้าเดินเข้ามาและเป็นแฟนตัวยงของนักแสดง “Nick Wahlberg” และอยากทราบว่าเขาอยู่ในภาพยนตร์เรื่องใด
- ดูรายชื่อภาพยนตร์ทั้งหมดที่ “Nick Wahlberg” เข้าฉาย

```
SELECT actor.first_name,actor.last_name,film.title
FROM film_actor
INNER JOIN actor
ON actor.actor_id = film_actor.actor_id
INNER JOIN film
ON film.film_id = film_actor.film_id
WHERE first_name = 'Nick'
AND last_name = 'Wahlberg';
```

```
#Answer
SELECT title,first_name,last_name
FROM film_actor
INNER JOIN actor
ON film_actor.actor_id = actor.actor_id
INNER JOIN film
ON film_actor.film_id = film.film_id
WHERE first_name = 'Nick'
AND last_name = 'Wahlberg';
```

Data Output Messages Notifications			
<div> <div>≡+</div> <div>📄</div> <div>▼</div> <div>📋</div> <div>🗑️</div> <div>🗄️</div> <div>⬇️</div> <div>📈</div> </div>			
	first_name character varying (45) 🔒	last_name character varying (45) 🔒	title character varying (255) 🔒
5	Nick	Wahlberg	Chainsaw Uptown
6	Nick	Wahlberg	Chisum Behavior
7	Nick	Wahlberg	Destiny Saturday
8	Nick	Wahlberg	Dracula Crystal
9	Nick	Wahlberg	Fight Jawbreaker
10	Nick	Wahlberg	Flash Wars
Total rows: 25 of 25		Query complete 00:00:00.039	