

# JOIN

## Praktikum Basis Data

Diyah Utami Kusumaning Putri, S.Kom., M.Sc., M.Cs.

# JOIN

- ❑ *Primary key* dan *foreign key* digunakan untuk menghubungkan *record* atau data antara beberapa tabel.
- ❑ Data dari seluruh tabel yang ada perlu dilihat seluruhnya untuk membuat informasi menjadi lengkap.
- ❑ Klausa *JOIN* digunakan untuk menggabungkan data dari dua atau beberapa tabel, berdasarkan kolom/*key* yang berhubungan di antara tabel tersebut.

# Tipe JOIN

- ❑ **(Inner) Join**: mengembalikan data yang memiliki nilai *key* yang sama/cocok di kedua tabel
- ❑ **Left (Outer) Join**: mengembalikan semua data dari tabel sebelah kiri dan menampilkan data dari tabel sebelah kanan yang cocok dengan kondisi *join*
- ❑ **Right (Outer) Join**: mengembalikan semua data dari tabel sebelah kanan dan menampilkan data dari tabel sebelah kiri yang cocok dengan kondisi *join*
- ❑ **Full (Outer) Join**: mengembalikan seluruh data/*record* dari tabel kiri dan tabel kanan yang dikenai ON (termasuk data-data yang bernilai NULL)
- ❑ **Self Join**: *Inner join* dari suatu tabel dengan dirinya sendiri
- ❑ **Union**: mengembalikan data yang gabungan dari dua atau lebih *statement* SELECT

# Table

employeeID	lastName	firstName	email	reportsTo	jobID
1	Carter	John	carterj@mail.co		1
2	Travolta	John	travj@mail.co		2
3	Carter	Peggy	carterp@mail.co		3
4	Connor	John	connj@mail.co		2
5	Johnson	Rian	jrian@mail.co		4
6	Rogers	Steve	cap@mail.co	3	2
7	Evans	Chris	evanc@mail.co	3	1
8	Pratt	Chris	prc@mail.co	1	5
9	Hemsworth	Chris	lebowski@mail.co	2	3
10	Hemsworth	Liam	hlam@mail.co		4
11	Feige	Kevin	kfeige@mail.co		6
12	Miller	George	travj@mail.co		

Tabel *Employees*

jobID	jobTitle
1	Producer
2	Director
3	Writer
4	Actor
5	Fans
6	Executive Producer
7	Stunt Organizer
8	VFX Artist
9	Stuntman

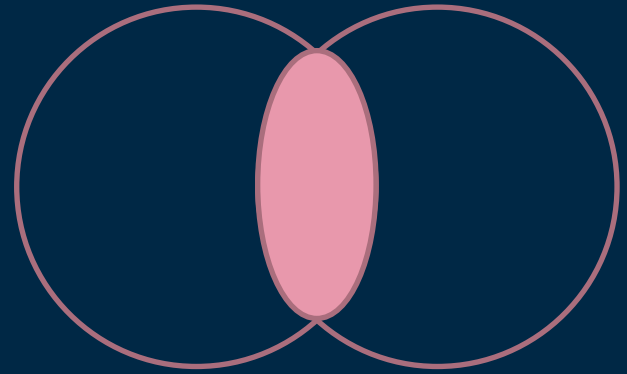
Tabel *Jobs*

# Inner Join

Mengembalikan data yang memiliki nilai *key* yang sama/cocok di kedua tabel

## Sintaks

```
1  -- INNER JOIN
2 • SELECT column1, column2
3  FROM table1
4  INNER JOIN table2
5  ON table1.column_name = table2.column_name;
```



Inner Join

# Inner Join

Tabel *Employees*

employeeID	lastName	firstName	email	reportsTo	jobID
1	Carter	John	carterj@mail.co		1
2	Travolta	John	travj@mail.co		2
3	Carter	Peggy	carterp@mail.co		3
4	Connor	John	connj@mail.co		2
5	Johnson	Rian	jrian@mail.co		4
6	Rogers	Steve	cap@mail.co	3	2
7	Evans	Chris	evanc@mail.co	3	1
8	Pratt	Chris	prc@mail.co	1	5
9	Hemsworth	Chris	lebowski@mail.co	2	3
10	Hemsworth	Liam	hlham@mail.co		4
11	Feige	Kevin	kfeige@mail.co		6
12	Miller	George	travj@mail.co		

Tabel *Jobs*

jobID	jobTitle
1	Producer
2	Director
3	Writer
4	Actor
5	Fans
6	Executive Producer
7	Stunt Organizer
8	VFX Artist
9	Stuntman

## Query

```
1  -- INNER JOIN
2 •  SELECT em.firstName, em.lastName, jo.jobTitle
3     FROM employees as em
4     INNER JOIN jobs as jo
5     ON em.jobID = jo.jobID;
```

## Hasil

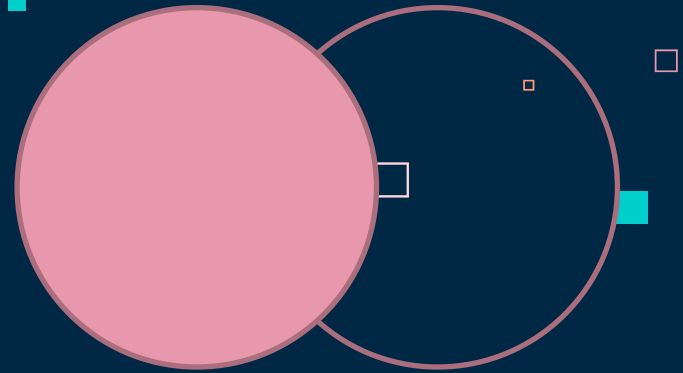
firstName	lastName	jobTitle
John	Carter	Producer
John	Travolta	Director
Peggy	Carter	Writer
John	Connor	Director
Rian	Johnson	Actor
Steve	Rogers	Director
Chris	Evans	Producer
Chris	Pratt	Fans
Chris	Hemsworth	Writer
Liam	Hemsworth	Actor
Kevin	Feige	Executi...

# Left (Outer) Join

mengembalikan semua data dari tabel sebelah kiri (tabel pertama) dan menampilkan data dari tabel sebelah kanan (tabel kedua) yang cocok dengan kondisi *join*

## Sintaks

```
1  -- LEFT JOIN
2  SELECT column1, column2
3  FROM table1
4  LEFT JOIN table2
5  ON table1.column_name = table2.column_name;
```



Left Join

# Left Join

Tabel *Employees*

employeeID	lastName	firstName	email	reportsTo	jobID
1	Carter	John	carterj@mail.co		1
2	Travolta	John	travj@mail.co		2
3	Carter	Peggy	carterp@mail.co		3
4	Connor	John	connj@mail.co		2
5	Johnson	Rian	jrian@mail.co		4
6	Rogers	Steve	cap@mail.co	3	2
7	Evans	Chris	evanc@mail.co	3	1
8	Pratt	Chris	prc@mail.co	1	5
9	Hemsworth	Chris	lebowski@mail.co	2	3
10	Hemsworth	Liam	hlham@mail.co		4
11	Feige	Kevin	kfeige@mail.co		6
12	Miller	George	travj@mail.co		

Tabel *Jobs*

jobID	jobTitle
1	Producer
2	Director
3	Writer
4	Actor
5	Fans
6	Executive Producer
7	Stunt Organizer
8	VFX Artist
9	Stuntman

## Query

```
1  -- LEFT JOIN
2  • SELECT em.firstName, em.lastName, jo.jobTitle
3  FROM employees as em
4  LEFT JOIN jobs as jo
5  ON em.jobID = jo.jobID;
```

## Hasil

firstName	lastName	jobTitle
John	Carter	Producer
John	Travolta	Director
Peggy	Carter	Writer
John	Connor	Director
Rian	Johnson	Actor
Steve	Rogers	Director
Chris	Evans	Producer
Chris	Pratt	Fans
Chris	Hemsworth	Writer
Liam	Hemsworth	Actor
Kevin	Feige	Executi...
George	Miller	NULL

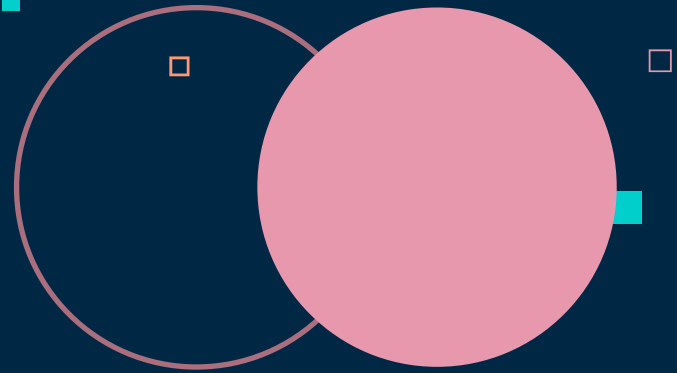


# Right (Outer) Join

mengembalikan semua data dari tabel sebelah kanan (tabel kedua) dan menampilkan data dari tabel sebelah kiri (tabel pertama) yang cocok dengan kondisi *join*

## Sintaks

```
-- RIGHT JOIN
SELECT column1, column2,...
FROM table1
RIGHT JOIN table2
ON table1.column_name = table2.column_name;
```



Right Join

# Right Join

Tabel *Employees*

employeeID	lastName	firstName	email	reportsTo	jobID
1	Carter	John	carterj@mail.co		1
2	Travolta	John	travj@mail.co		2
3	Carter	Peggy	carterp@mail.co		3
4	Connor	John	connj@mail.co		2
5	Johnson	Rian	jrian@mail.co		4
6	Rogers	Steve	cap@mail.co	3	2
7	Evans	Chris	evanc@mail.co	3	1
8	Pratt	Chris	prc@mail.co	1	5
9	Hemsworth	Chris	lebowski@mail.co	2	3
10	Hemsworth	Liam	hlham@mail.co		4
11	Feige	Kevin	kfeige@mail.co		6
12	Miller	George	travj@mail.co		

Tabel *Jobs*

jobID	jobTitle
1	Producer
2	Director
3	Writer
4	Actor
5	Fans
6	Executive Producer
7	Stunt Organizer
8	VFX Artist
9	Stuntman

## Query

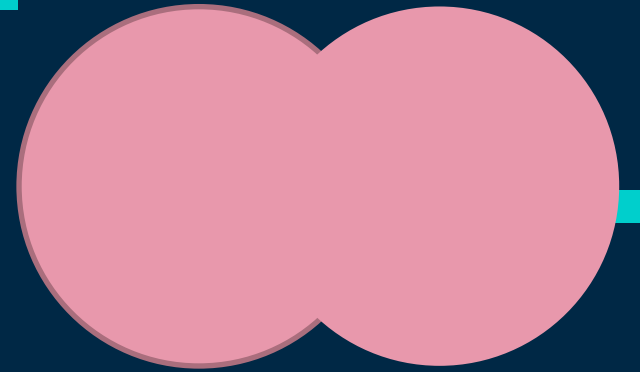
```
1  -- RIGHT JOIN
2 • SELECT em.firstName, em.lastName, jo.jobTitle
3  FROM employees as em
4  RIGHT JOIN jobs as jo
5  ON em.jobID = jo.jobID;
```

## Hasil

firstName	lastName	jobTitle
John	Carter	Producer
Chris	Evans	Producer
John	Travolta	Director
John	Connor	Director
Steve	Rogers	Director
Peggy	Carter	Writer
Chris	Hemsworth	Writer
Rian	Johnson	Actor
Liam	Hemsworth	Actor
Chris	Pratt	Fans
Kevin	Feige	Executi...
NULL	NULL	Stunt O...
NULL	NULL	VFX Artist
NULL	NULL	Stuntman

# Full (Outer) Join

mengembalikan seluruh data/record dari tabel kiri dan tabel kanan yang dikenai ON (termasuk data-data yang bernilai NULL) ketika ada kecocokan antara data dari kedua tabel tersebut



## Sintaks

```
1 -- FULL OUTER JOIN
2 SELECT column1, column2, ...
3 FROM table1
4 FULL OUTER JOIN table2
5 ON table1.column_name = table2.column_name;
```

## Sintaks (MySQL)

```
1 -- FULL OUTER JOIN
2 SELECT column1, column2, ...
3 FROM table1
4 LEFT JOIN table2
5 ON table1.column_name = table2.column_name
6 UNION
7 SELECT column1, column2, ...
8 FROM table1
9 RIGHT JOIN table2
10 ON table1.column_name = table2.column_name
```

## Full Join

# Full Join

Tabel *Employees*

employeeID	lastName	firstName	email	reportsTo	jobID
1	Carter	John	carterj@mail.co		1
2	Travolta	John	travj@mail.co		2
3	Carter	Peggy	carterp@mail.co		3
4	Connor	John	connj@mail.co		2
5	Johnson	Rian	jrian@mail.co		4
6	Rogers	Steve	cap@mail.co	3	2
7	Evans	Chris	evanc@mail.co	3	1
8	Pratt	Chris	prc@mail.co	1	5
9	Hemsworth	Chris	lebowski@mail.co	2	3
10	Hemsworth	Liam	hlam@mail.co		4
11	Feige	Kevin	kfeige@mail.co		6
12	Miller	George	travj@mail.co		

Tabel *Jobs*

jobID	jobTitle
1	Producer
2	Director
3	Writer
4	Actor
5	Fans
6	Executive Producer
7	Stunt Organizer
8	VFX Artist
9	Stuntman

## Query

```
1  -- FULL OUTER JOIN
2  SELECT em.firstName, em.lastName, jo.jobTitle
3  FROM employees AS em
4  LEFT JOIN jobs AS jo
5  ON em.jobID = jo.jobID
6  UNION
7  SELECT em.firstName, em.lastName, jo.jobTitle
8  FROM employees AS em
9  RIGHT JOIN jobs AS jo
10 ON em.jobID = jo.jobID
```

## Hasil

firstName	lastName	jobTitle
John	Carter	Producer
John	Travolta	Director
Peggy	Carter	Writer
John	Connor	Director
Rian	Johnson	Actor
Steve	Rogers	Director
Chris	Evans	Producer
Chris	Pratt	Fans
Chris	Hemsworth	Writer
Liam	Hemsworth	Actor
Kevin	Feige	Executive Producer
George	Miller	NULL
NULL	NULL	Stunt Organizer
NULL	NULL	VFX Artist
NULL	NULL	Stuntman

# Self Join

mengembalikan seluruh *record* dari tabel yang digabungkan dengan dirinya sendiri

## Sintaks

```
1  -- SELF JOIN
2 •  SELECT column1, column2
3     FROM table1 as T1, table2 as T2
4     WHERE conditions;
```

# Self Join

Tabel *Employees*

employeeID	lastName	firstName	email	reportsTo	jobID
1	Carter	John	carterj@mail.co		1
2	Travolta	John	travj@mail.co		2
3	Carter	Peggy	carterp@mail.co		3
4	Connor	John	connj@mail.co		2
5	Johnson	Rian	jrian@mail.co		4
6	Rogers	Steve	cap@mail.co	3	2
7	Evans	Chris	evanc@mail.co	3	1
8	Pratt	Chris	prc@mail.co	1	5
9	Hemsworth	Chris	lebowski@mail.co	2	3
10	Hemsworth	Liam	hlam@mail.co		4
11	Feige	Kevin	kfeige@mail.co		6
12	Miller	George	travj@mail.co		

## Query

```
1 • SELECT
2     em1.employeeID, em1.firstName, em1.lastName,
3     em1.reportsTo, em2.firstName, em2.lastName
4 FROM
5     employees as em1, employees as em2
6 WHERE
7     em1.reportsTo = em2.employeeID;
```

## Hasil

	employeeID	firstName	lastName	reportsTo	firstName	lastName
▶	6	Steve	Rogers	3	Peggy	Carter
	7	Chris	Evans	3	Peggy	Carter
	8	Chris	Pratt	1	John	Carter
	9	Chris	Hemsworth	2	John	Travolta

# Union

mengembalikan data yang gabungan dari dua atau lebih *statement SELECT*

## Sintaks

```
1  -- UNION
2 • SELECT column1, column2 FROM table1
3  UNION
4  SELECT column1, column2 FROM table2;
```

## Catatan

- ❑ Setiap statement SELECT dalam UNION harus memiliki jumlah kolom yang sama
- ❑ Kolom-kolom tersebut juga harus memiliki tipe data yang serupa
- ❑ Kolom-kolom pada setiap statement SELECT juga harus berada dalam urutan yang sama

# Union

Tabel *Employees*

employeeID	lastName	firstName	email	reportsTo	jobID
1	Carter	John	carterj@mail.co		1
2	Travolta	John	travj@mail.co		2
3	Carter	Peggy	carterp@mail.co		3
4	Connor	John	connj@mail.co		2
5	Johnson	Rian	jrian@mail.co		4
6	Rogers	Steve	cap@mail.co	3	2
7	Evans	Chris	evanc@mail.co	3	1
8	Pratt	Chris	prc@mail.co	1	5
9	Hemsworth	Chris	lebowski@mail.co	2	3
10	Hemsworth	Liam	hlam@mail.co		4
11	Feige	Kevin	kfeige@mail.co		6
12	Miller	George	travj@mail.co		

Tabel *Jobs*

jobID	jobTitle
1	Producer
2	Director
3	Writer
4	Actor
5	Fans
6	Executive Producer
7	Stunt Organizer
8	VFX Artist
9	Stuntman

Query

```
1  -- UNION
2 •  SELECT jobID
3    FROM employees
4    WHERE jobID IS NOT NULL
5    UNION
6    SELECT jobID FROM jobs;
```

Hasil

jobID
1
2
3
4
5
6
7
8
9



# Union All

Tabel *Employees*

employeeID	lastName	firstName	email	reportsTo	jobID
1	Carter	John	carterj@mail.co		1
2	Travolta	John	travj@mail.co		2
3	Carter	Peggy	carterp@mail.co		3
4	Connor	John	connj@mail.co		2
5	Johnson	Rian	jrian@mail.co		4
6	Rogers	Steve	cap@mail.co	3	2
7	Evans	Chris	evanc@mail.co	3	1
8	Pratt	Chris	prc@mail.co	1	5
9	Hemsworth	Chris	lebowski@mail.co	2	3
10	Hemsworth	Liam	hlam@mail.co		4
11	Feige	Kevin	kfeige@mail.co		6
12	Miller	George	travj@mail.co		

Tabel *Jobs*

jobID	jobTitle
1	Producer
2	Director
3	Writer
4	Actor
5	Fans
6	Executive Producer
7	Stunt Organizer
8	VFX Artist
9	Stuntman

Query

```
1  -- UNION ALL
2  • SELECT jobID
3  FROM employees
4  WHERE jobID IS NOT NULL AND jobID < 6
5  UNION ALL
6  SELECT jobID FROM jobs;
```

Hasil

jobID
1
1
2
2
2
3
3
4
4
5
6
7
8
9

The background is a dark blue gradient. It is decorated with various geometric elements: thin white vertical lines of varying lengths, small squares in teal, orange, and pink, and larger squares in teal and orange. Some squares are solid, while others are outlined in white. The overall aesthetic is modern and minimalist.

# THANK YOU