Rakamin Academy

SQL

Complex Queries

Rakamin Academy

Outline

1. Subquery
2. Join
3. Union
4. Common Table Expression (CTE)

Rakamin Academy

Subquery

Query di dalam query.

Subquery 1 - Definisi

Kita dapat membuat sebuah query didalam sebuah query, yang dinamakan subquery. Biasanya digunakan untuk mendapatkan value dari tabel lain yang dibutuhkan untuk proses selanjutnya

Perintah dalam SQL
SELECT [kolom_1], [kolom_2] FROM (SELECT [kolom_1], [kolom_2] FROM <tabel_a>) alias

Subquery 2 - Tabel Virtual

Subquery bisa digunakan untuk membuat sebuah tabel ‘virtual’. Setiap subquery harus memiliki **Alias**.

Query Editor | Query History

```
1  select *
2  from
3  (
4      select
5          customer_id,
6          customer_name,
7          city,
8          region
9      from superstore_customer
10 ) subq
```

Data Output | Explain | Messages | Notifications

	customer_id character varying	customer_name character varying	city character varying	region character varying
1	CG-12520	Claire Gute	Henderson	South
2	DV-13045	Darrin Van Huff	Los Angeles	West
3	SO-20335	Sean O'Donnell	Fort Lauderdale	South
4	BH-11710	Brosina Hoffman	Los Angeles	West
5	AA-10480	Andrew Allen	Concord	South

Subquery 2 - Tabel Virtual

Dari subquery tersebut, kita juga bisa melakukan filtering.

Contoh:
Bagaimana persebaran total customer pada masing-masing region yang jumlah customernya lebih dari 200?

Query Editor | Query History

```
1  select *
2  from
3  (
4      select
5          region,
6          count(1) as total_cust
7      from
8          superstore_customer
9      where
10         region is not null
11     group by 1
12 ) subq
13 where total_cust > 200
```

Data Output | Explain | Messages | Notifications

	region character varying	total_cust bigint
1	West	255
2	East	220

Subquery 3 - Filtering

Hasil dari subquery digunakan sebagai filter query utama.

Perintah dalam SQL
SELECT [kolom_1], [kolom_2] FROM <table_a> WHERE [kolom_1] IN (SELECT [kolom_1]   FROM <tabel_a>)

Subquery 3 - Filtering

Hasil dari subquery digunakan sebagai filter query utama.

Contoh:
Ambil data customer yang pernah melakukan order dengan metode shipping `Same Day`

Query Editor | Query History

```
1  select *
2  from superstore_customer
3  where
4      customer_id in (
5          select distinct customer_id
6          from superstore_order
7          where ship_mode = 'Same Day'
8      )
```

Data Output | Explain | Messages | Notifications

	customer_id character varying	customer_name character varying	segment character varying	country character varying	city character varying	state character varying	postal_code character varying	region character varying
1	CG-12520	Claire Gute	Consumer	United States	Henderson	Kentucky	42420	South
2	DV-13045	Darrin Van Huff	Corporate	United States	Los Angeles	California	90036	West
3	AA-10480	Andrew Allen	Consumer	United States	Concord	North Carolina	28027	South
4	PK-19075	Pete Kriz	Consumer	United States	Madison	Wisconsin	53711	Central
5	ZD-21925	Zuschuss Donatelli	Consumer	United States	San Francisco	California	94109	West

Rakamin Academy

Join

Menggabungkan tabel.

JOIN

Proses join menggunakan **kolom** dengan **nilai baris** yang **sama** antara 2 tabel.

JOIN

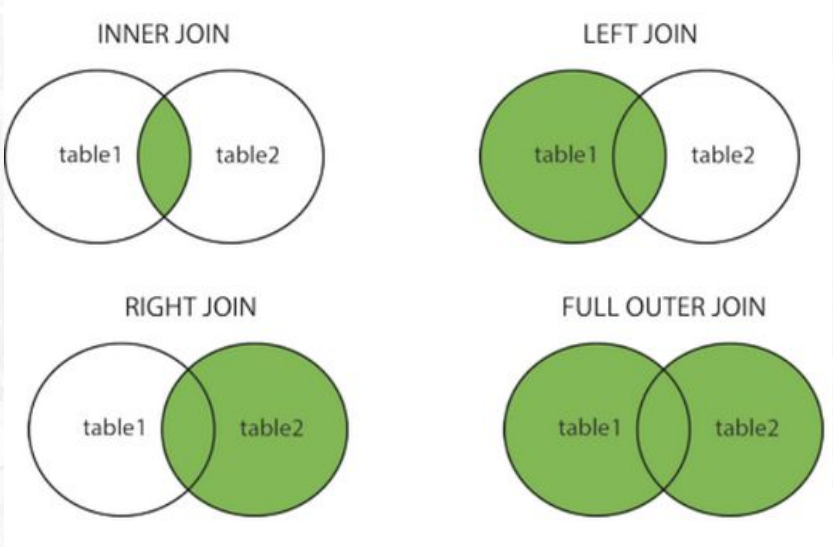
Proses join menggunakan **kolom** dengan **nilai baris** yang **sama** antara 2 tabel.

Perintah dalam SQL
select [kolom_1], [kolom_2] from <table_a> [tipe_join] <table_b> on table_a.[kolom] = table_b.[kolom]

keyword **ON** mencocokkan value yang sama pada kondisi **JOIN**

Tipe-tipe Join Table

Terdapat berbagai macam tipe tergantung kebutuhan menggabungkan tabel



Tipe-tipe Join Table

Terdapat berbagai macam tipe tergantung kebutuhan menggabungkan tabel

Tipe	Definisi
INNER JOIN	Hanya mencocokkan dan menampilkan data yang sama antara 2 tabel
LEFT JOIN	Menampilkan semua data di tabel sebelah kiri, dan data yang cocok di tabel sebelah kanan
RIGHT JOIN	Menampilkan semua data di tabel sebelah kanan, dan data yang cocok di tabel sebelah kiri
FULL OUTER JOIN	Menampilkan semua data baik ketika ada yang cocok antara kedua tabel ataupun tidak

Ilustrasi

Tabel A

Id	Usia
1	18
2	50
4	24
6	45

Tabel B

Id	Negara
1	ID
2	SG
3	JP
5	KR

INNER JOIN

Id	Usia	Negara
1	18	ID
2	50	SG

FULL OUTER JOIN

Id	Usia	Negara
1	18	ID
2	50	SG
4	24	*NULL*
6	45	*NULL*
3	*NULL*	JP
5	*NULL*	KR

LEFT JOIN

Id	Usia	Negara
1	18	ID
2	50	SG
4	24	*NULL*
6	45	*NULL*

RIGHT JOIN

Id	Usia	Negara
1	18	ID
2	50	SG
3	*NULL*	JP
5	*NULL*	KR

INNER JOIN / JOIN

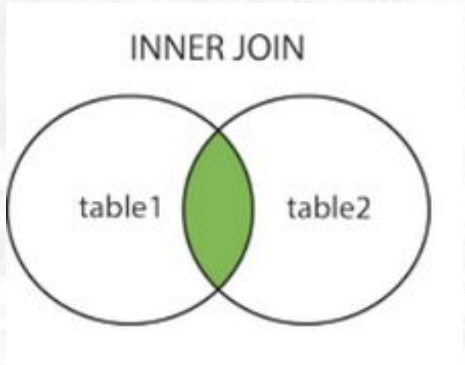
Contoh : Mengambil order ID, nama produk dan quantity dari setiap order.

Query Editor | Query History

```
1  select
2      o.order_id,
3      p.product_name,
4      o.quantity
5  from
6      superstore_order o
7  join
8      superstore_product p
9  on o.product_id = p.product_id
```

Data Output | Explain | Messages | Notifications

	order_id character varying	product_name character varying	quantity integer
1	CA-2016-152156	Bush Somerset Collection Bookcase	2
2	CA-2016-152156	Hon Deluxe Fabric Upholstered Stacking Chairs, Rounded Back	3
3	CA-2016-138688	Self-Adhesive Address Labels for Typewriters by Universal	2
4	US-2015-108966	Bretford CR4500 Series Slim Rectangular Table	5
5	US-2015-108966	Eldon Fold 'N Roll Cart System	2



LEFT JOIN

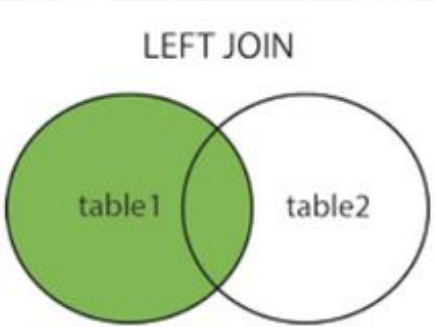
Contoh : Ambil customer_id dan order_date dari masing-masing customer. Kita ingin melihat siapa yang sudah pernah dan belum pernah melakukan order.

Query Editor | Query History

```
1  select
2      c.customer_id,
3      o.order_date
4  from superstore_customer c
5  left join superstore_order o
6  on c.customer_id = o.customer_id
7  order by o.order_date desc
```

Data Output | Explain | Messages | Notifications

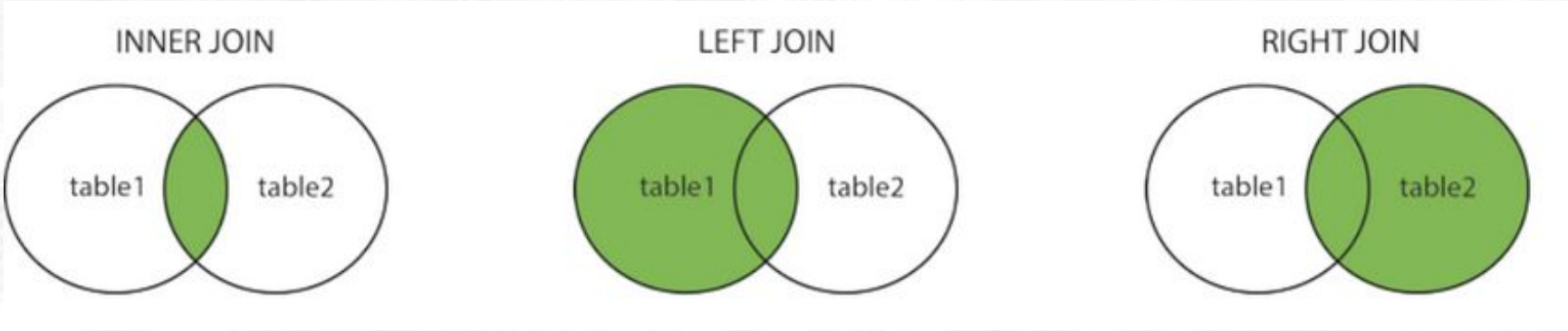
	customer_id character varying	order_date date
1	IZ-94832	[null]
2	NT-32326	[null]
3	FR-65423	[null]
4	LF-12123	[null]
5	OB-48472	[null]
6	CC-12430	2017-12-30
7	PO-18865	2017-12-30
8	EB-13975	2017-12-30
9	FB-13975	2017-12-30



LEFT JOIN disebut juga dengan LEFT OUTER JOIN

Jadi,
Apa bedanya ya JOIN dan LEFT JOIN?

Untuk statement JOIN pada SQL, **hanya mencocokkan dan menampilkan data yang sama antara 2 tabel,** sedangkan LEFT JOIN, mengambil semua data pada tabel sebelah kiri, dan **data yang cocok saja di sebelah kanan**



Rakamin Academy

Query Editor | Query History

```
1 select *
2 from superstore_customer c
3 join superstore_order o
4 on c.customer_id = o.customer_id
```

Data Output | Explain | Messages | Notifications

```
Successfully run. Total query runtime: 129 msec.
9994 rows affected.
```

INNER JOIN

Query Editor | Query History

```
1 select *
2 from superstore_customer c
3 left join superstore_order o
4 on c.customer_id = o.customer_id
```

Data Output | Explain | Messages | Notifications

```
Successfully run. Total query runtime: 112 msec.
9999 rows affected.
```

LEFT JOIN

Query Editor | Query History

```
1  select *
2  from superstore_customer c
3  left join superstore_order o
4  on c.customer_id = o.customer_id
```

Data Output | Explain | Messages | Notifications

```
Successfully run. Total query runtime: 112 msec.
9999 rows affected.
```

LEFT JOIN

Query Editor | Query History

```
1  select *
2  from superstore_order o
3  right join superstore_customer c
4  on c.customer_id = o.customer_id
```

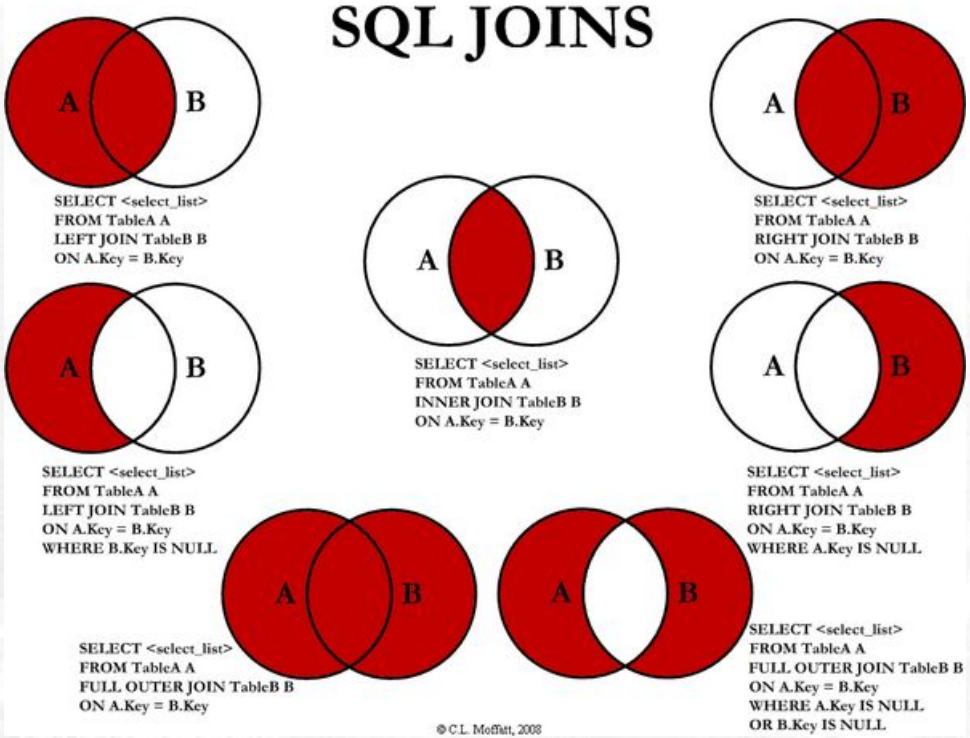
Data Output | Explain | Messages | Notifications

```
Successfully run. Total query runtime: 114 msec.
9999 rows affected.
```

RIGHT JOIN

Penggunaan join lainnya

Perintah *joins* bisa dimodifikasi untuk penggunaan yang lebih kompleks.



Contoh - Menggunakan Subquery pada JOIN

Contoh:
Ambil data customer yang pernah melakukan transaksi yang total sales lebih dari 15000.

Query Editor | Query History

```
1  select
2      c.customer_id,
3      subq.total_sales,
4      c.customer_name
5  from
6  (
7      select
8          customer_id,
9          sum(sales) as total_sales
10     from superstore_order
11     group by 1
12     having sum(sales) > 15000
13 ) subq
14 left join superstore_customer c
15 on subq.customer_id = c.customer_id
```

Data Output | Explain | Messages | Notifications

	customer_id character varying	total_sales double precision	customer_name character varying
1	RB-19360	15117.339	Raymond Buch
2	TC-20980	19052.217999999993	Tamara Chand
3	SM-20320	25043.05	Sean Miller

Union

Menggabungkan query.

Union

Untuk menggabungkan dua atau lebih hasil query dari sebuah tabel, dapat digunakan UNION.

Perintah dalam SQL
select [kolom_1], [kolom_2] from <table_a> [UNION] select [kolom_1], [kolom_2] from <table_b>

Syarat:

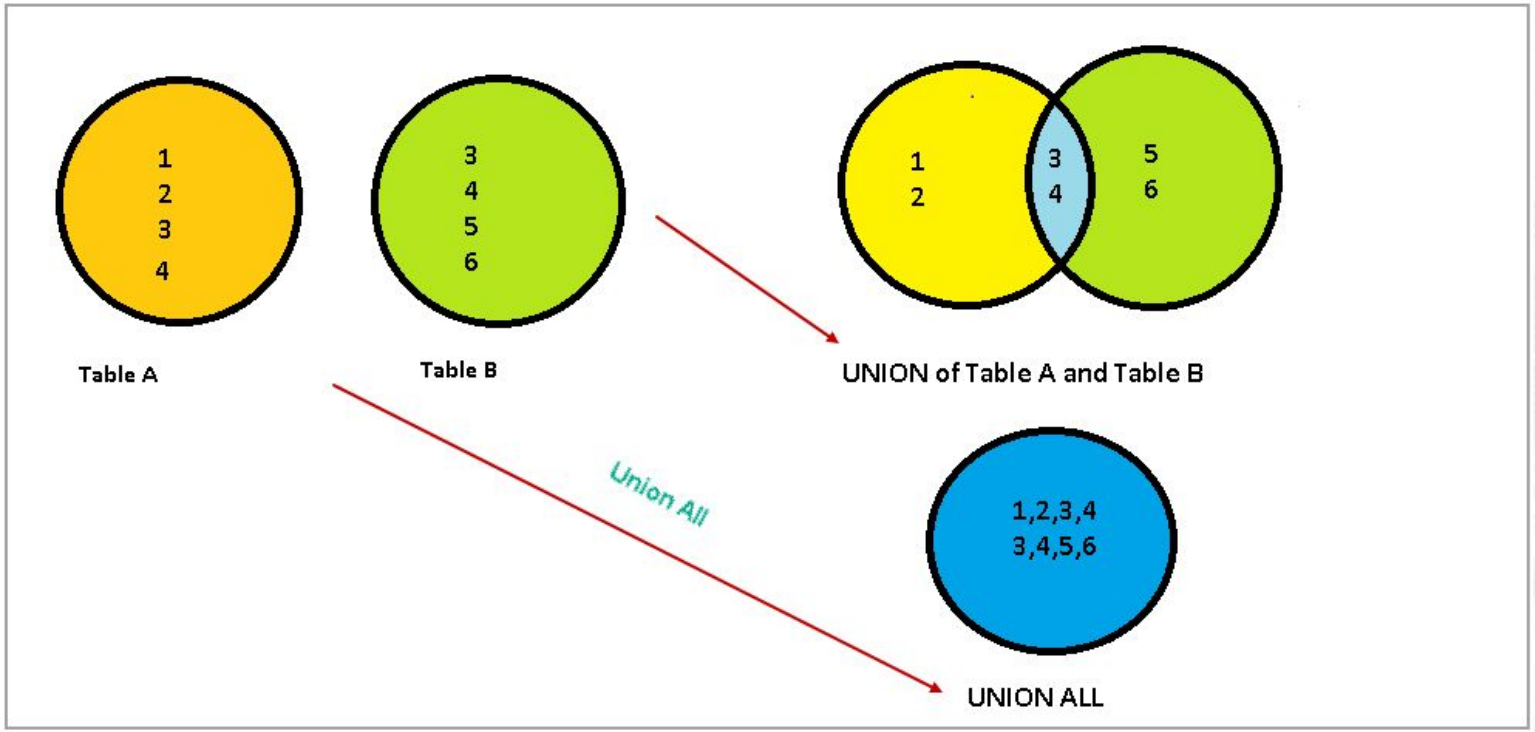
- Semua kolom yang digunakan antara tabel a dan tabel b harus sama
- Tipe data dari kolomnya juga harus sama
- Urutan dari kolom pun juga harus sama

Jenis UNION

Ada beberapa jenis untuk menggabungkan sebuah data dengan UNION

Tipe	Definisi
UNION	Mengembalikan data secara *distinct*
UNION ALL	Mengembalikan data secara duplikat (jika ada)

Ilustrasi


1
2
3
4
Table A
3
4
5
6
Table B
1
2
3
4
5
6
UNION of Table A and Table B
Union All
1,2,3,4
3,4,5,6
UNION ALL

Contoh - UNION & UNION ALL

Ambil data customer yang berada di region South. Ambil juga data customer yang merupakan segmen Consumer. (gunakan UNION & UNION ALL)

Query Editor | Query History

```
1  select *
2  from superstore_customer
3  where region = 'South'
4  union
5  select *
6  from superstore_customer
7  where segment = 'Consumer'
```

Data Output | Explain | Messages | Notifications

```
Successfully run. Total query runtime: 67 msec.
480 rows affected.
```

Query Editor | Query History

```
1  select *
2  from superstore_customer
3  where region = 'South'
4  union all
5  select *
6  from superstore_customer
7  where segment = 'Consumer'
```

Data Output | Explain | Messages | Notifications

```
Successfully run. Total query runtime: 59 msec.
548 rows affected.
```

Common Table Expression (CTE)

WITH ... AS ()

Kita dapat juga membentuk sebuah temporary table dengan perintah SQL ini

Perintah dalam SQL
WITH [nama_temporary_table] AS ([put your query here...]) SELECT * FROM [nama_temporary_table]

Contoh - WITH ... AS

Dengan menggunakan contoh sebelumnya (Ambil data customer yang pernah melakukan transaksi yang total sales nya lebih dari 15000)

Query Editor | Query History

```
1  with
2  temp as (
3      select
4          customer_id,
5          sum(sales) as total_sales
6      from superstore_order
7      group by 1
8      having sum(sales) > 15000
9  )
10
11 select
12     c.customer_id,
13     temp.total_sales,
14     c.customer_name
15 from temp
16 left join superstore_customer c
17 on temp.customer_id = c.customer_id
```

Data Output | Explain | Messages | Notifications

	customer_id character varying	total_sales double precision	customer_name character varying
1	RB-19360	15117.339	Raymond Buch
2	TC-20980	19052.217999999993	Tamara Chand
3	SM-20320	25043.05	Sean Miller

Contoh - WITH ... AS (2)

Variabel *with* bisa lebih dari satu.

Query Editor | Query History

```
1  with
2  big_sales as (
3      select
4          customer_id,
5          sum(sales) as total_sales
6      from superstore_order
7      group by 1
8      having sum(sales) > 15000
9  ),
10 south_region as (
11     select *
12     from superstore_customer
13     where region = 'South'
14 )
15
16 select *
17 from south_region sr
18 join big_sales bs
19 on sr.customer_id = bs.customer_id
```

Data Output | Explain | Messages | Notifications

	customer_id character varying	customer_name character varying	segment character varying	country character varying	city character va
1	SM-20320	Sean Miller	Home Office	United States	Monroe

Terimakasih