

Virtual Internship Experience

**Data Joining dan
Union**



kimia farma

Disclaimer

“Dokumen ini memiliki hak cipta. Barang siapa yang menyebarluaskan atau menduplikasi tanpa izin dari instansi terkait dapat diproses sesuai dengan ketentuan hukum yang berlaku.”

Outline

- 1 **Proses Data Joining**
- 2 **Union**

Data Source

Table name: Order

OrderID	CustomerID	OrderDate
10308	2	1996-09-18
10309	37	1996-09-19
10310	77	1996-09-20

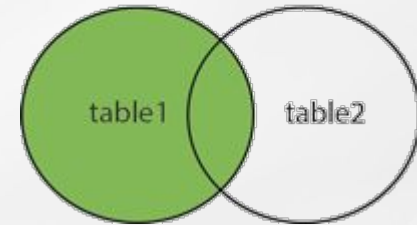
Table name: Customer

CustomerID	CustomerName	ContactName	Country
1	Alfreds Futterkiste	Maria Anders	Germany
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Mexico
3	Antonio Moreno Taquería	Antonio Moreno	Mexico

Left Join

```
SELECT
    order.orderid, customer.customername, order.orderdate
FROM order
LEFT JOIN customer ON order.customerid = customer.customerid;
```

LEFT JOIN

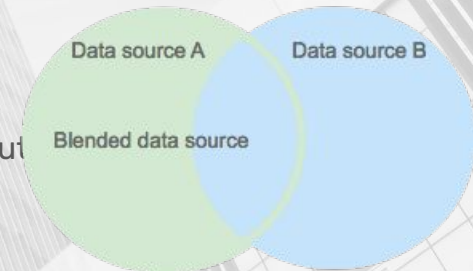


Dari query tersebut, akan menghasilkan:

- 3 row sesuai jumlah row pada table order
- customername yang muncul hanya yang customerid = 2, karena customerid 37 dan 77 tidak ada di table customer

Konsep left join ini juga sama dengan **blended data source**.

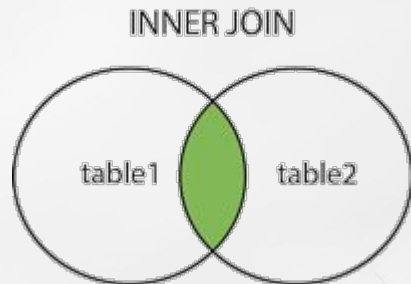
Left join biasa digunakan untuk join data fact atau transaksi ke mendapatkan informasi dari column dimension yang tidak ada di data transaksi tersebut





Inner Join

```
SELECT
    order.orderid, customer.customername, order.orderdate
FROM order
INNER JOIN customer ON order.customerid = customer.customerid;
```

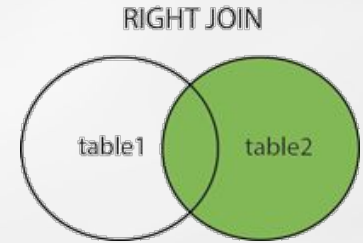


Dari query tersebut, akan menghasilkan: 1 row karena customerID yang match di table order dan customer hanya 1.

inner join biasanya digunakan untuk tujuan yang spesifik, yaitu untuk mendapatkan data yang ada di antara 2 table. contohnya, untuk mendapatkan data customer yang tinggal di kota tertentu, kita perlu inner join data dari tabel customer dengan tabel region sehingga mendapatkan informasi total customer pada suatu kota yang kita inginkan.

Right Join

```
SELECT
    order.orderid, customer.customername, order.orderdate
FROM order
RIGHT JOIN customer ON order.customerid = customer.customerid;
```



Dari query tersebut, akan menghasilkan:

- 3 row sesuai jumlah row pada table customer
- orderid dan orderdate yang muncul hanya yang customerid = 2, karena customerid 1 dan 3 tidak ada di table order

Right join pada dasarnya sangat jarang digunakan, karena kita bisa menggunakan left join dengan menukar posisi tabel yang digunakan sebagai main table. pada contoh query di atas, kita bisa menggunakan cara:

select ... from customer left join order on ...

query tersebut akan menghasilkan result yang sama dengan query right join di atas

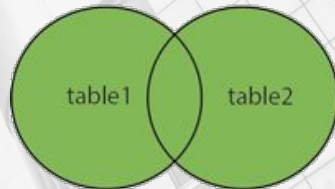
Full Outer Join

```
SELECT  
    COALESCE(order.customerid, customer.customerid) customerid, order.orderid, customer.customername, order.orderdate  
FROM order  
FULL OUTER JOIN customer ON order.customerid = customer.customerid;
```

Dari query tersebut, akan menghasilkan: 5 row, yaitu semua customerid: 2,3,7,7,1,3

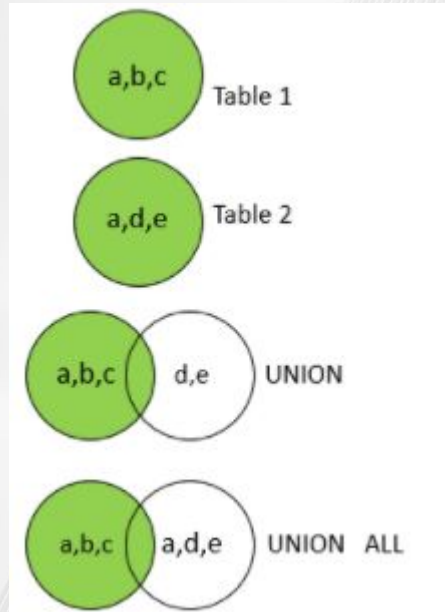
Biasanya full outer join akan digunakan untuk mendapatkan seluruh populasi dari 2 tabel dengan key tertentu. contoh use case nya adalah data customer yang melakukan transaksi pembelian obat kita full outer join dengan data transaksi customer yang melakukan pemeriksaan. Jika kita ingin menggabungkan data customer pada 2 tabel transaksi tersebut, kita bisa mendapatkan list dari customer yang melakukan pembelian obat dan atau pemeriksaan, juga memungkinkan jika ada data transaksi yang tidak tercatat customer nya bisa kita lihat untuk keperluan analisis lebih lanjut.

FULL OUTER JOIN





Union



```
#UNION
SELECT customerid FROM order
UNION
SELECT customerid FROM customer
```

```
#UNION ALL
SELECT customerid FROM order
UNION
SELECT customerid FROM customer
```

- Query UNION, menghasilkan 5 row, yaitu customerid: 1,2,3,37,77
- Query UNION ALL, menghasilkan 6 row, yaitu customerid: 1,2,3,2,37,77.

Perbedaannya adalah, jika UNION akan menghasilkan data yang unik atau distinct, sedangkan union all akan menghasilkan keseluruhan data pada tabel-tabel yang di UNION, dan memungkinkan ada duplicate.

Contoh implementasi UNION adalah, misalkan kita ingin menggabungkan data pelanggan dengan data vendor sebagai sebuah identity yang sama, dengan menambahkan column identity_type yang berisi customer dan vendor.



Thank You!

