



Pada modul praktikum kedua mempunyai tujuan yaitu mempelajari bagaimana penggunaan perintah lanjut SQL yaitu perintah *join table* untuk mengelola data dengan **Database MySQL**. Database masih menggunakan dbtransaksi pada modul praktikum sebelumnya.

A. SQL Lanjut

Joining tables

- **Table & Column Aliases** - memperkenalkan Anda ke tabel dan kolom alias.

Perintah

```
SELECT
    [column_1 | expression] AS descriptive_name
FROM table_name;
```

```
SELECT
    CONCAT_WS(' ', lastName, firstname)
FROM
    employees;
```

```
SELECT
    CONCAT_WS(' ', lastName, firstname) AS `Full name`
FROM
    employees;
```

```
SELECT
    CONCAT_WS(' ', lastName, firstname) `Full name`
FROM
    employees
ORDER BY
    `Full name`;
```

```
SELECT
    orderNumber `Order no.`,
    SUM(priceEach * quantityOrdered) total
FROM
    orderDetails
GROUP BY
    `Order no.`
HAVING
    total > 60000;
```

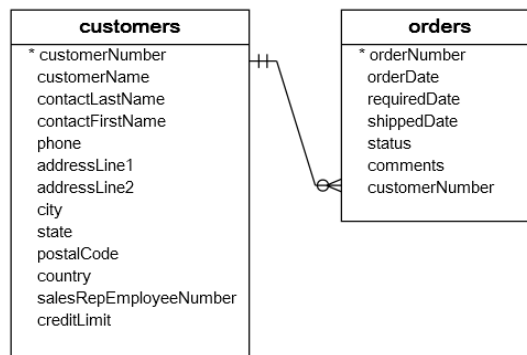
MySQL alias for tables

```
table_name AS table_alias
```

```
SELECT * FROM employees e;
```

```
table_alias.column_name
```

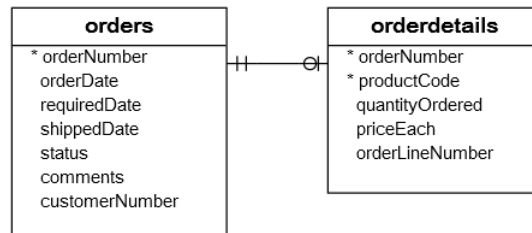
```
SELECT
    e.firstName,
    e.lastName
FROM
    employees e
ORDER BY e.firstName;
```



```
SELECT
    customerName,
    COUNT(o.orderNumber) total
FROM
    customers c
INNER JOIN orders o ON c.customerNumber = o.customerNumber
GROUP BY
    customerName
ORDER BY
    total DESC;
```

```
SELECT
    customers.customerName,
    COUNT(orders.orderNumber) total
FROM
    customers
INNER JOIN orders ON customers.customerNumber = orders.customerNumber
GROUP BY
    customerName
ORDER BY
    total DESC
```

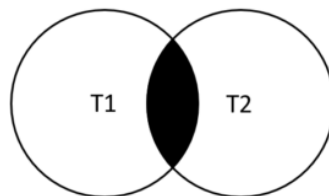
- **JOIN** – memberi Anda gambaran umum tentang gabungan yang didukung di MySQL termasuk inner join, left join, dan right join.



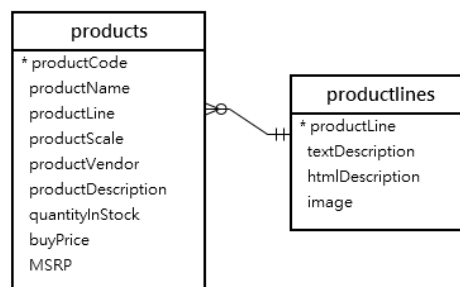
- **INNER JOIN** – kueri baris dari tabel yang memiliki baris yang cocok di tabel lain.

```
SELECT
    select_list
FROM t1
INNER JOIN t2 ON join_condition1
INNER JOIN t3 ON join_condition2
...;
```

Diagram Venn berikut menggambarkan cara kerja klausa INNER JOIN :



Mari kita lihat tabel produk dan lini produk di database sampel.

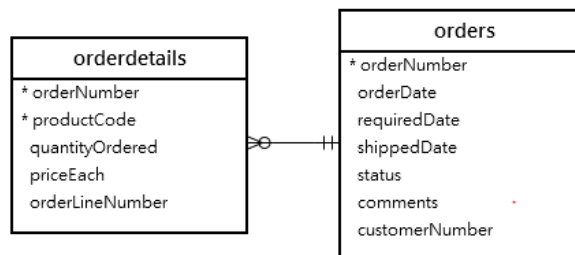


```
SELECT
    productCode,
    productName,
    textDescription
FROM
    products t1
INNER JOIN productlines t2
    ON t1.productline = t2.productline;
```

```
SELECT
    productCode,
    productName,
    textDescription
FROM
    products
INNER JOIN productlines USING (productline);
```

MySQL INNER JOIN dengan contoh klausa GROUP BY

Lihat tabel *orders* dan *orderdetails* berikut:

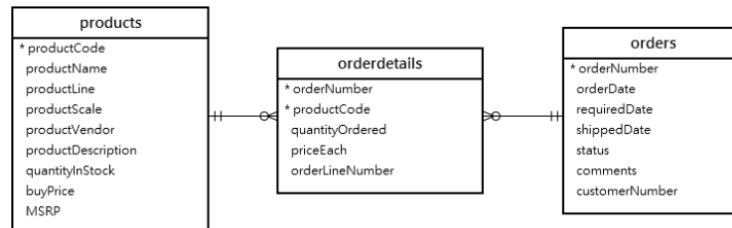


```
SELECT
    t1.orderNumber,
    t1.status,
    SUM(quantityOrdered * priceEach) total
FROM
    orders t1
INNER JOIN orderdetails t2
    ON t1.orderNumber = t2.orderNumber
GROUP BY orderNumber;
```

```
SELECT
    orderNumber,
    status,
    SUM(quantityOrdered * priceEach) total
FROM
    orders
INNER JOIN orderdetails USING (orderNumber)
GROUP BY orderNumber;
```

MySQL INNER JOIN – gabungkan tiga tabel contoh

Lihat tabel produk, pesanan, dan detail pesanan berikut:



```

SELECT
    orderNumber,
    orderDate,
    orderLineNumber,
    productName,
    quantityOrdered,
    priceEach
FROM
    orders
INNER JOIN
    orderdetails USING (orderNumber)
INNER JOIN
    products USING (productCode)
ORDER BY
    orderNumber,
    orderLineNumber;
    
```

MySQL INNER JOIN – gabungkan empat tabel contoh

Lihat pesanan, detail pesanan, pelanggan, dan tabel produk berikut:



```
SELECT
    orderNumber,
    orderDate,
    customerName,
    orderLineNumber,
    productName,
    quantityOrdered,
    priceEach
FROM
    orders
INNER JOIN orderdetails
    USING (orderNumber)
INNER JOIN products
    USING (productCode)
INNER JOIN customers
    USING (customerNumber)
ORDER BY
    orderNumber,
    orderLineNumber;
```

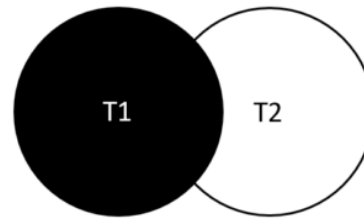
MySQL INNER JOIN menggunakan operator lain

```
SELECT
    orderNumber,
    productName,
    msrp,
    priceEach
FROM
    products p
INNER JOIN orderdetails o
    ON p.productcode = o.productcode
    AND p.msrp > o.priceEach
WHERE
    p.productcode = 'S10_1678';
```

- **LEFT JOIN** – mengembalikan semua baris dari tabel kiri dan baris yang cocok dari tabel kanan atau null jika tidak ada baris yang cocok ditemukan di tabel kanan.

Pernyataan berikut menunjukkan cara menggunakan klausa LEFT JOIN untuk menggabungkan dua tabel :

```
SELECT
    select_list
FROM
    t1
LEFT JOIN t2 ON
    join_condition;
```



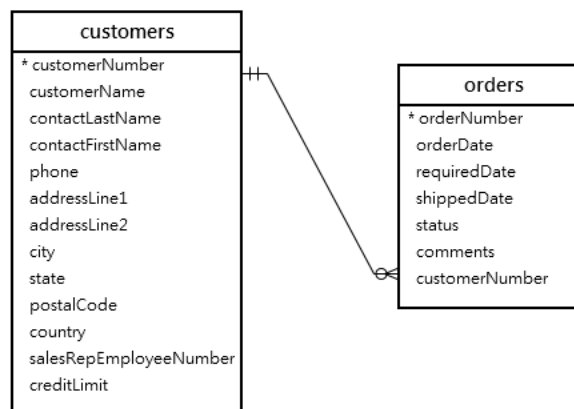
MySQL LEFT JOIN – Venn Diagram

Contoh klausa MySQL LEFT JOIN

Mari kita ambil beberapa contoh penggunaan klausa LEFT JOIN.

1) Menggunakan klausa MySQL LEFT JOIN untuk menggabungkan dua tabel

Lihat tabel berikut pelanggan dan pesanan di database sampel.



```
SELECT
    customers.customerNumber,
    customerName,
    orderNumber,
    status
FROM
    customers
LEFT JOIN orders ON
    orders.customerNumber = customers.customerNumber;
```

```
SELECT
    c.customerNumber,
    customerName,
    orderNumber,
    status
FROM
    customers c
LEFT JOIN orders o
    ON c.customerNumber = o.customerNumber;
```

Dalam contoh ini:

- ✓ Pelanggan adalah meja kiri dan pesanan adalah meja kanan.
- ✓ Klausula LEFT JOIN mengembalikan semua pelanggan termasuk pelanggan yang tidak memiliki pesanan. Jika pelanggan tidak memiliki pesanan, nilai pada kolom orderNumber dan status adalah NULL.

Karena tabel pelanggan dan pesanan memiliki nama kolom yang sama (customerNumber) dalam kondisi join dengan operator yang sama, Anda dapat menggunakan sintaks USING sebagai berikut:

```
SELECT
    customerNumber,
    customerName,
    orderNumber,
    status
FROM
    customers
LEFT JOIN orders USING (customerNumber);
```

The following clauses are equivalent:

```
USING (customerNumber)
```

And

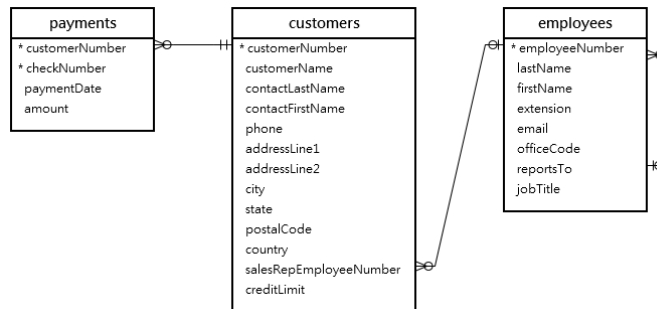
```
ON c.customerNumber = o.customerNumber
```

2) Menggunakan klausa MySQL LEFT JOIN untuk menemukan

```
SELECT
    c.customerNumber,
    c.customerName,
    o.orderNumber,
    o.status
FROM
    customers c
LEFT JOIN orders o
    ON c.customerNumber = o.customerNumber
WHERE
    orderNumber IS NULL;
```


3) Menggunakan MySQL LEFT JOIN untuk menggabungkan tiga tabel

Lihat tiga tabel berikut karyawan, pelanggan, dan pembayaran:



```
SELECT
    lastName,
    firstName,
    customerName,
    checkNumber,
    amount
FROM
    employees
LEFT JOIN customers ON
    employeeNumber = salesRepEmployeeNumber
LEFT JOIN payments ON
    payments.customerNumber = customers.customerNumber
ORDER BY
    customerName,
    checkNumber;
```

Kondisi dalam klausa WHERE vs. klausa ON
Lihat contoh berikut.

```
SELECT
    o.orderNumber,
    customerNumber,
    productCode
FROM
    orders o
LEFT JOIN orderDetails
    USING (orderNumber)
WHERE
    orderNumber = 10123;
```

```
SELECT
    o.orderNumber,
    customerNumber,
    productCode
FROM
    orders o
LEFT JOIN orderDetails d
    ON o.orderNumber = d.orderNumber AND
       o.orderNumber = 10123;
```

- **RIGHT JOIN** – mengembalikan semua baris dari tabel kanan dan baris yang cocok dari tabel kiri atau null jika tidak ada baris yang cocok ditemukan di tabel kiri.

Berikut sintaks RIGHT JOIN dari dua tabel t1 dan t2:

```
SELECT
    select_list
FROM t1
RIGHT JOIN t2 ON
    join_condition;
```

Dalam sintaks ini:

- ✓ t1 adalah tabel kiri dan t2 adalah tabel kanan.
- ✓ Join_condition menentukan aturan untuk mencocokkan baris dari kedua tabel.

Jika join_condition menggunakan operator yang sama (=) dan kolom gabungan dari kedua tabel memiliki nama yang sama, dan Anda dapat menggunakan sintaks USING seperti ini:

```
SELECT
    select_list
FROM t1
RIGHT JOIN t2 USING(column_name);
```

Therefore, the following join conditions are equivalent:

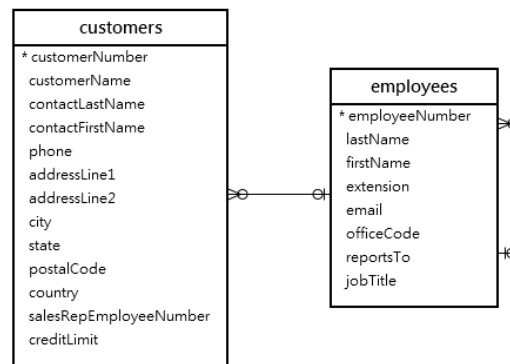
```
ON t1.column_name = t2.column_name
```

and

```
USING (column_name);
```

Contoh klausa MySQL RIGHT JOIN

Disini menggunakan tabel karyawan dan pelanggan dari database sampel untuk demonstrasi :



1) Contoh MySQL RIGHT JOIN sederhana

```
SELECT
    employeeNumber,
    customerNumber
FROM
    customers
RIGHT JOIN employees
    ON salesRepEmployeeNumber = employeeNumber
ORDER BY
    employeeNumber;
```

2) Menggunakan MySQL RIGHT JOIN untuk menemukan baris yang tidak cocok

```
SELECT
    employeeNumber,
    customerNumber
FROM
    customers
RIGHT JOIN employees ON
    salesRepEmployeeNumber = employeeNumber
WHERE customerNumber is NULL
ORDER BY employeeNumber;
```

- **CROSS JOIN** – membuat produk Cartesian dari baris dari beberapa tabel.

```
SELECT * FROM t1  
CROSS JOIN t2;
```

```
SELECT * FROM t1  
CROSS JOIN t2  
WHERE t1.id = t2.id;
```

Contoh klausa MySQL CROSS JOIN

Mari kita siapkan beberapa tabel untuk mendemonstrasikan klausa CROSS JOIN.

Menyiapkan tabel sampel

Pertama, buat database baru salesdb:

```
CREATE DATABASE IF NOT EXISTS salesdb;
```

```
USE salesdb;
```

Ketiga, buat tabel baru di database salesdb:

```
CREATE TABLE products (  
  id INT PRIMARY KEY AUTO_INCREMENT,  
  product_name VARCHAR(100),  
  price DECIMAL(13,2 )  
);  
  
CREATE TABLE stores (  
  id INT PRIMARY KEY AUTO_INCREMENT,  
  store_name VARCHAR(100)  
);  
  
CREATE TABLE sales (  
  product_id INT,  
  store_id INT,  
  quantity DECIMAL(13 , 2 ) NOT NULL,  
  sales_date DATE NOT NULL,  
  PRIMARY KEY (product_id , store_id),  
  FOREIGN KEY (product_id)  
    REFERENCES products (id)  
    ON DELETE CASCADE ON UPDATE CASCADE,  
  FOREIGN KEY (store_id)  
    REFERENCES stores (id)  
    ON DELETE CASCADE ON UPDATE CASCADE  
);
```

```
INSERT INTO products(product_name, price)
VALUES('iPhone', 699),
      ('iPad', 599),
      ('Macbook Pro', 1299);

INSERT INTO stores(store_name)
VALUES('North'),
      ('South');

INSERT INTO sales(store_id, product_id, quantity, sales_date)
VALUES(1, 1, 20, '2017-01-02'),
      (1, 2, 15, '2017-01-05'),
      (1, 3, 25, '2017-01-05'),
      (2, 1, 30, '2017-01-02'),
      (2, 2, 35, '2017-01-05');
```

Contoh MySQL CROSS JOIN

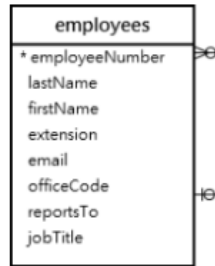
```
SELECT
    store_name,
    product_name,
    SUM(quantity * price) AS revenue
FROM
    sales
    INNER JOIN
    products ON products.id = sales.product_id
    INNER JOIN
    stores ON stores.id = sales.store_id
GROUP BY store_name, product_name;
```

```
SELECT
    store_name, product_name
FROM
    stores AS a
    CROSS JOIN
    products AS b;
```

- **Self-join** – gabungan tabel ke tabel itu sendiri menggunakan alias tabel dan hubungkan baris dalam tabel yang sama menggunakan inner join dan left join.

Contoh self join MySQL

Mari kita lihat tabel karyawan di database sampel.



1) MySQL self join menggunakan INNER JOIN clause

Untuk mendapatkan seluruh struktur organisasi, Anda dapat menggabungkan tabel karyawan ke dirinya sendiri menggunakan kolom employeeNumber dan reportTo. Tabel Employee memiliki dua peran: satu adalah Manajer dan yang lainnya adalah Direct Reports.

```
SELECT
    CONCAT(m.lastName, ', ', m.firstName) AS Manager,
    CONCAT(e.lastName, ', ', e.firstName) AS 'Direct report'
FROM
    employees e
INNER JOIN employees m ON
    m.employeeNumber = e.reportsTo
ORDER BY
    Manager;
```

2) MySQL self join menggunakan klausa LEFT JOIN

Presiden adalah karyawan yang tidak memiliki manajer atau nilai apa pun di kolom reportTo is NULL .

Pernyataan berikut menggunakan klausa LEFT JOIN daripada INNER JOIN untuk menyertakan President:

```
SELECT
    IFNULL(CONCAT(m.lastname, ', ', m.firstname),
        'Top Manager') AS 'Manager',
    CONCAT(e.lastname, ', ', e.firstname) AS 'Direct report'
FROM
    employees e
LEFT JOIN employees m ON
    m.employeeNumber = e.reportsto
ORDER BY
    manager DESC;
```

- 3) Menggunakan MySQL self join untuk membandingkan baris yang berurutan.

Dengan menggunakan MySQL self join, Anda dapat menampilkan daftar pelanggan yang berada di kota yang sama dengan menggabungkan tabel pelanggan ke dirinya sendiri.

```
SELECT
    c1.city,
    c1.customerName,
    c2.customerName
FROM
    customers c1
INNER JOIN customers c2 ON
    c1.city = c2.city
    AND c1.customername > c2.customerName
ORDER BY
    c1.city;
```

----- Sekian Terima Kasih -----

Referensi

- <https://www.mysql.com/>
- <https://www.mysqltutorial.org/>

