**Sesi 12 Latihan 2**

**Database Company**

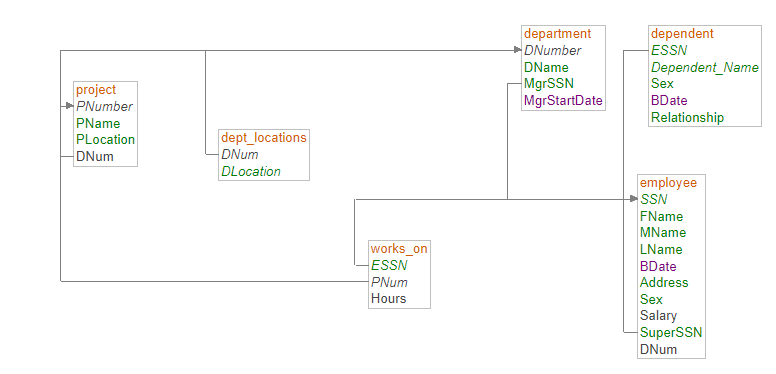
1. **Langkah ke-1**

Membuat Tabel dan insert data pada database Company menggunakan import pada file Latihan2.sql pada Folder ini. Lalu execute

****

1. **Langkah ke dua**

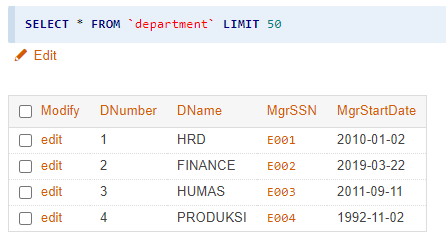
Lihat hubungan antar relasi pada database Company, lalu periksa apakah keseluruhan database telah terhubung

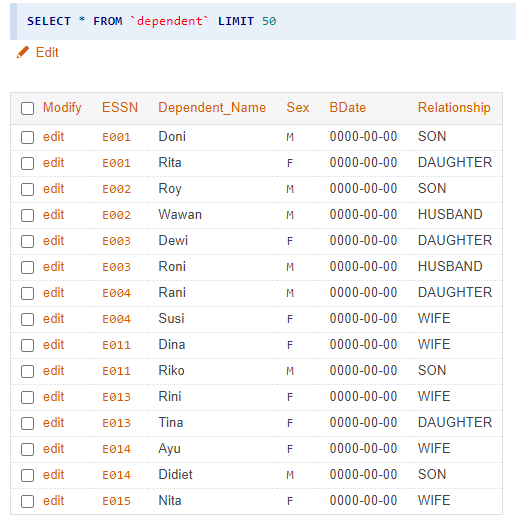


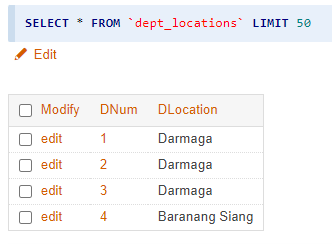
1. **Langkah ke 3**

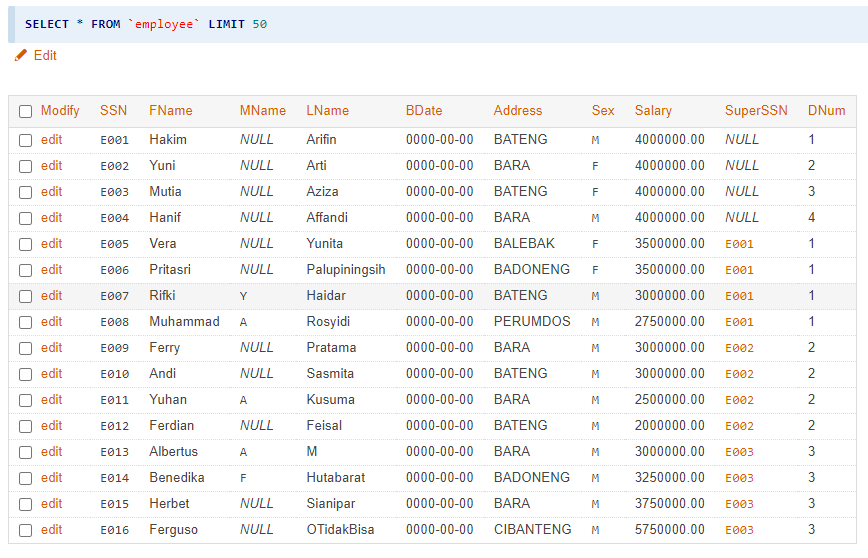
Database telah dibuat dengan baik

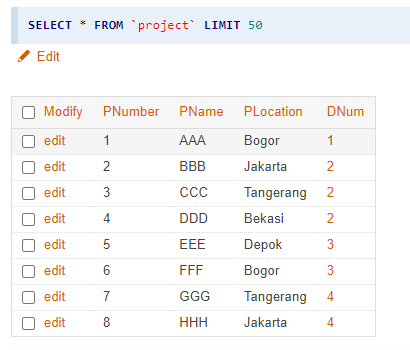
Dan berikut merupakan data yang telah di create

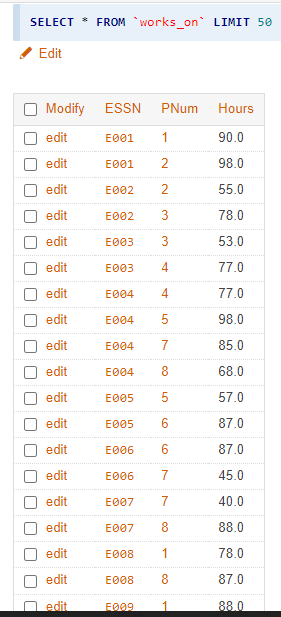












**Requirement 2 - Syntax SQL**

1. **Tampilkan dependent\_name dan relationship dengan employee yang namanya diawali huruf R ?**

**Syntax :**

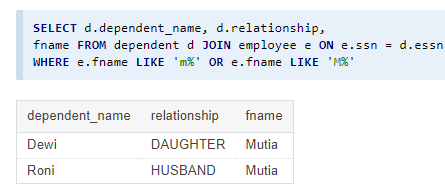
[**SELECT**](https://mariadb.com/kb/en/library/select/) d.dependent\_name, d.relationship,   
fname FROM [dependent](http://localhost/adminer/?server=localhost&username=root&db=company&table=dependent) d [**JOIN**](https://mariadb.com/kb/en/library/join/) [employee](http://localhost/adminer/?server=localhost&username=root&db=company&table=employee) e ON e.ssn = d.essn   
WHERE e.fname [**LIKE**](https://mariadb.com/kb/en/library/string-comparison-functions/#operator_like) 'r%' [**OR**](https://mariadb.com/kb/en/library/logical-operators/#operator_or) e.fname [**LIKE**](https://mariadb.com/kb/en/library/string-comparison-functions/#operator_like) 'R%';

****

Karena pada database kita tidak terdapat relationship yang memiliki nama depan diawali “r” atau “R”, maka tidak ada database yang muncul. Kita coba memanggil relationship yang memiliki nama depan diawali “m” atau “M” maka akan muncul data seperti ini

**Syntax :**

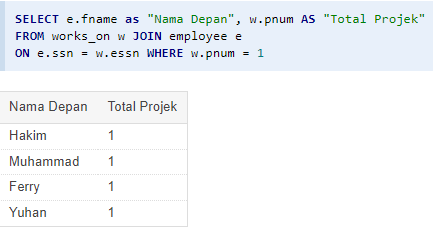
[**SELECT**](https://mariadb.com/kb/en/library/select/) d.dependent\_name, d.relationship,   
fname FROM [dependent](http://localhost/adminer/?server=localhost&username=root&db=company&table=dependent) d [**JOIN**](https://mariadb.com/kb/en/library/join/) [employee](http://localhost/adminer/?server=localhost&username=root&db=company&table=employee) e ON e.ssn = d.essn   
WHERE e.fname [**LIKE**](https://mariadb.com/kb/en/library/string-comparison-functions/#operator_like) 'm%' [**OR**](https://mariadb.com/kb/en/library/logical-operators/#operator_or) e.fname [**LIKE**](https://mariadb.com/kb/en/library/string-comparison-functions/#operator_like) 'M%';



1. **Banyaknya employee yang mengerjakan project PNum = 1 ?**

**Syntax :**

[**SELECT**](https://mariadb.com/kb/en/library/select/) e.fname as "Nama Depan", w.pnum AS "Total Projek"   
FROM [works\_on](http://localhost/adminer/?server=localhost&username=root&db=company&table=works_on) w [**JOIN**](https://mariadb.com/kb/en/library/join/) [employee](http://localhost/adminer/?server=localhost&username=root&db=company&table=employee) e   
ON e.ssn = w.essn WHERE w.pnum = 1;

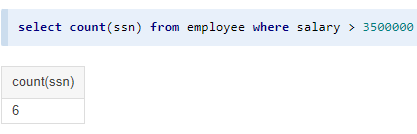
****

Logika nya, ini akan menSelect kolom fName dan pNum dari table Works\_on lalu melakukan join table pada employee dimana nilai pNum = 1. Jadi yang ditampilkan ialah karyawan yang memiliki pNum = 1.

1. **Banyaknya employee yang memiliki salary lebih dari 3500000 ?**

**Syntax :**

[**select**](https://mariadb.com/kb/en/library/select/) [**count**](https://mariadb.com/kb/en/library/group-by-functions/#function_count)(ssn) from [employee](http://localhost/adminer/?server=localhost&username=root&db=company&table=employee) where salary > 3500000;

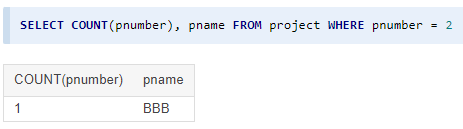
****

Pada Syntax kali ini, kita akan mencari banyaknya employee yang memiliki salary lebih dari 3.5 juta. Nah, untuk mencari banyaknya employee kita akan menggunakan fungsi Count. Count berfungsi untuk menampilkan banyaknya data yang diminta pada sebuah table

1. **Banyaknya project yang dikerjakan DNum =2 ?**

**Syntax :**

[**SELECT**](https://mariadb.com/kb/en/library/select/) [**COUNT**](https://mariadb.com/kb/en/library/group-by-functions/#function_count)(pnumber), pname FROM [project](http://localhost/adminer/?server=localhost&username=root&db=company&table=project) WHERE pnumber = 2;

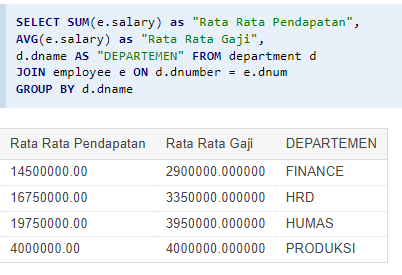
****

Logikanya sama seperti nomor 3 diatas, yaitu menggunakan fungsi count untuk menghitung banyaknya kolom yang diminta di mana pNumber = 2

1. **Hitung total dan rata-rata salary dari setiap departemen ?**

**Syntax :**

[**SELECT**](https://mariadb.com/kb/en/library/select/) [**SUM**](https://mariadb.com/kb/en/library/group-by-functions/#function_sum)(e.salary) as "Rata Rata Pendapatan",   
[**AVG**](https://mariadb.com/kb/en/library/group-by-functions/#function_avg)(e.salary) as "Rata Rata Gaji",   
d.dname AS "Departemen" FROM [department](http://localhost/adminer/?server=localhost&username=root&db=company&table=department) d   
[**JOIN**](https://mariadb.com/kb/en/library/join/) [employee](http://localhost/adminer/?server=localhost&username=root&db=company&table=employee) e ON d.dnumber = e.dnum   
GROUP BY d.dname;

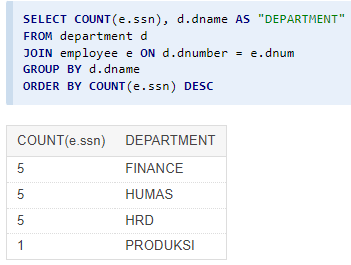
****

Logika pada soal kali ini adalah menggunakan 3 fungsi yakni **sum,** untuk menjumlahkan satu kolom yang dipilih, **AVG** untuk mencari rata-ratanya, dan **Join** untuk menggabungkan 2 tabel menjadi 1 tabel.

1. **Banyaknya employee dari setiap department dan urutkan berdasarkan employee terbanyak ?**

**Syntax :**

[**SELECT**](https://mariadb.com/kb/en/library/select/) [**COUNT**](https://mariadb.com/kb/en/library/group-by-functions/#function_count)(e.ssn) as "Jumlah Pegawai",  
d.dname AS "DEPARTMENT"   
FROM [department](http://localhost/adminer/?server=localhost&username=root&db=company&table=department) d   
[**JOIN**](https://mariadb.com/kb/en/library/join/) [employee](http://localhost/adminer/?server=localhost&username=root&db=company&table=employee) e ON d.dnumber = e.dnum   
GROUP BY d.dname   
ORDER BY [**COUNT**](https://mariadb.com/kb/en/library/group-by-functions/#function_count)(e.ssn) DESC;

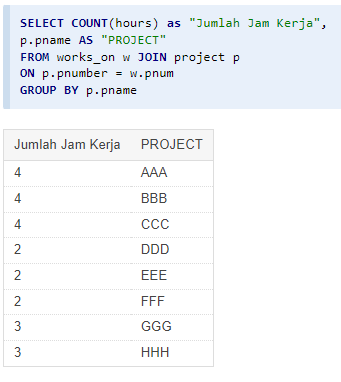
****

Fungsi yang digunakan ialah **DESC**, untuk mengurutkan data dari besar ke terkecil atau biasa di kenal dengan **Descending**. Jika kebalikannya menggunakan **ASC,** atau **Ascending** untuk mengurutkan data dari kecil ke terbesar. Selanjutnya ada Syntax **Group By** berfungsi untuk mengelompokkan data dalam sebuah kolom yang ditunjuk

1. **Total hours perweek dari semua employee untuk setiap project ?**

**Syntax :**

[**SELECT**](https://mariadb.com/kb/en/library/select/) [**COUNT**](https://mariadb.com/kb/en/library/group-by-functions/#function_count)(hours) as "Jumlah Jam Kerja",   
p.pname AS "PROJECT"   
FROM [works\_on](http://localhost/adminer/?server=localhost&username=root&db=company&table=works_on) w [**JOIN**](https://mariadb.com/kb/en/library/join/) [project](http://localhost/adminer/?server=localhost&username=root&db=company&table=project) p   
ON p.pnumber = w.pnum   
GROUP BY p.pname;

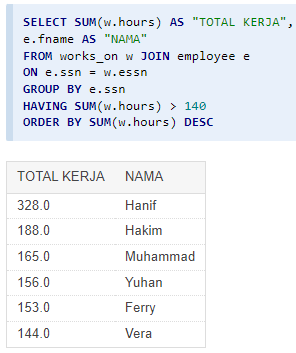
****

**Pada** soal kali ini kita akan menghitung banyaknya Jumlah jam kerja per Minggu dari seuruh karyawan untuk tiap projek. Logika yang digunakan ialah menjumlahkan kolom **HOURS** dengan memfilter pName dengan p.PName pada projek. Maka kita akan mendapatkan output jumlah jam kerja pada tiap projek

1. **Employee yang memiliki total hours perweek lebih besar dari 140 hours dan urutkan berdasarkan jumlah jam kerja terbanyak?**

**Syntax :**

[**SELECT**](https://mariadb.com/kb/en/library/select/) [**SUM**](https://mariadb.com/kb/en/library/group-by-functions/#function_sum)(w.hours) AS "TOTAL JAM KERJA",   
e.fname AS "NAMA"   
FROM [works\_on](http://localhost/adminer/?server=localhost&username=root&db=company&table=works_on) w [**JOIN**](https://mariadb.com/kb/en/library/join/) [employee](http://localhost/adminer/?server=localhost&username=root&db=company&table=employee) e   
ON e.ssn = w.essn   
GROUP BY e.ssn   
HAVING [**SUM**](https://mariadb.com/kb/en/library/group-by-functions/#function_sum)(w.hours) > 140   
ORDER BY [**SUM**](https://mariadb.com/kb/en/library/group-by-functions/#function_sum)(w.hours) DESC;

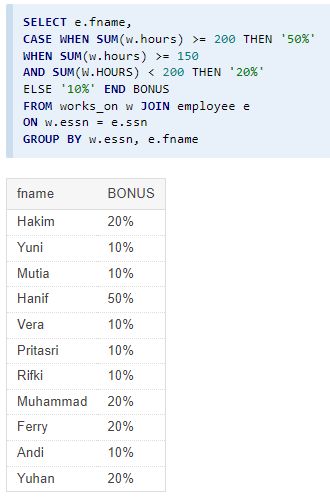
****

Fungsi **MySQL HAVING** digunakan dalam kombinasi dengan fungsi MySQL **GROUP BY** untuk membatasi kelompok baris yang dikembalikan hanya kepada mereka yang kondisinya **BENAR**. Jadi pada syntax kali ini bisa dilihat bahwa e.ssn telah di **GROUP BY,** data yang telah di grupkan akan diberikan suatu kondisi kembali dengan menggunakan **Having Sum (w.hours) > 140**. Artinya jika jam kerja kurang dari 140 atau data **False** maka data tidak akan di munculkan.

1. **Kelompokkan bonus employee berdasarkan jumlah jam kerjanya ? (Jika >= 200 hours, maka bonus = 50%; Jika >= 150 hours, maka bonus = 25%, Selainnya bonus = 10%)**

**Syntax :**

[**SELECT**](https://mariadb.com/kb/en/library/select/) e.fname,   
[**CASE**](https://mariadb.com/kb/en/library/control-flow-functions/#operator_case) WHEN [**SUM**](https://mariadb.com/kb/en/library/group-by-functions/#function_sum)(w.hours) >= 200 THEN '50%'   
WHEN [**SUM**](https://mariadb.com/kb/en/library/group-by-functions/#function_sum)(w.hours) >= 150   
[**AND**](https://mariadb.com/kb/en/library/logical-operators/#operator_and) [**SUM**](https://mariadb.com/kb/en/library/group-by-functions/#function_sum)(W.HOURS) < 200 THEN '20%'   
ELSE '10%' [**END**](https://mariadb.com/kb/en/library/begin-end/) BONUS   
FROM [works\_on](http://localhost/adminer/?server=localhost&username=root&db=company&table=works_on) w [**JOIN**](https://mariadb.com/kb/en/library/join/) [employee](http://localhost/adminer/?server=localhost&username=root&db=company&table=employee) e   
ON w.essn = e.ssn   
GROUP BY w.essn, e.fname;

****

Nah pada soal kali ini kita mendapatkan syntax baru yakni **CASE, WHEN, ELSE**

**CASE,** digunakan untuk melewati kondisi dan mengembalikan nilai saat kondisi pertama **terpenuhi** (seperti pernyataan I**F-THEN-ELSE**). Jadi, begitu suatu kondisi **benar**, itu akan berhenti membaca dan mengembalikan hasilnya. Jika tidak ada kondisi yang benar, ini mengembalikan nilai dalam klausa **ELSE**.

1. **Banyaknya project yang dikerjakan tiap employee dan urutkan dari yang terbanyak ?**

**Syntax :**

[**SELECT**](https://mariadb.com/kb/en/library/select/) [**COUNT**](https://mariadb.com/kb/en/library/group-by-functions/#function_count)(p.pnumber) AS "TOTAL PROJECT",  
 e.fname AS "NAMA" FROM [project](http://localhost/adminer/?server=localhost&username=root&db=company&table=project) p [**JOIN**](https://mariadb.com/kb/en/library/join/) [employee](http://localhost/adminer/?server=localhost&username=root&db=company&table=employee) e   
ON e.dnum = p.dnum   
GROUP BY e.ssn   
ORDER BY [**COUNT**](https://mariadb.com/kb/en/library/group-by-functions/#function_count)(p.pnumber) DESC;

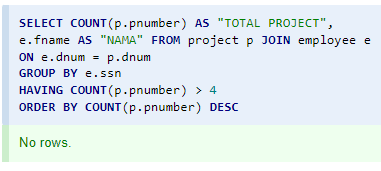
****

Sebenernya **mager** jelasin, tapi demi **reza** di masa depan yang selalu lupa sytax bakal dijelasin nih. Logikanya adalah mengelompokkan jumlah karyawan dengan banyak nya projek yang di kerjakaannya. Menggunakan table **e.fname** pada table **employee** lalu di join dengan **e.dnum** yang nilainya sama dengan **project dNumber**. Lalu di kelompokkan berdasarkan id **SSN** lalu di count (dijumlahkan) dan diurutkan **DESC** (dari besar ke kecil)

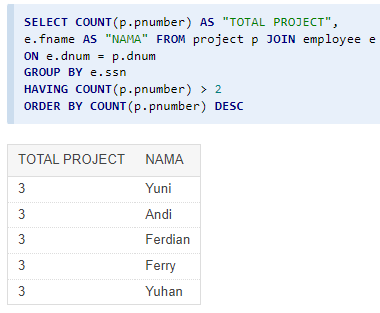
1. **Employee yang bekerja pada 4 project ?**

**Syntax :**

[**SELECT**](https://mariadb.com/kb/en/library/select/) [**COUNT**](https://mariadb.com/kb/en/library/group-by-functions/#function_count)(p.pnumber) AS "TOTAL PROJECT",   
e.fname AS "NAMA" FROM [project](http://localhost/adminer/?server=localhost&username=root&db=company&table=project) p [**JOIN**](https://mariadb.com/kb/en/library/join/) [employee](http://localhost/adminer/?server=localhost&username=root&db=company&table=employee) e   
ON e.dnum = p.dnum   
GROUP BY e.ssn   
HAVING [**COUNT**](https://mariadb.com/kb/en/library/group-by-functions/#function_count)(p.pnumber) > 4   
ORDER BY [**COUNT**](https://mariadb.com/kb/en/library/group-by-functions/#function_count)(p.pnumber) DESC;



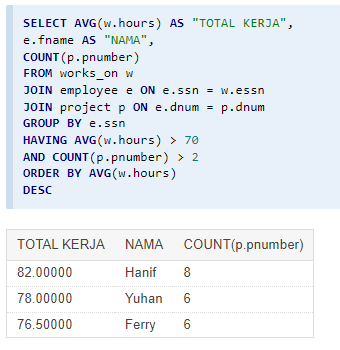
Mengapa nilainya **No Rows?** Ini dikarenakan datanya memang tidak ada yang memiliki lebih dari 4 projek pada waktu bersamaan. Jika kita coba menggunakan lebih dari 2 projek maka akan menampilkan :



1. **Employee yang memiliki rata-rata hours perweek = 70 jam dan bekerja pada 2 project ?**

**Syntax :**

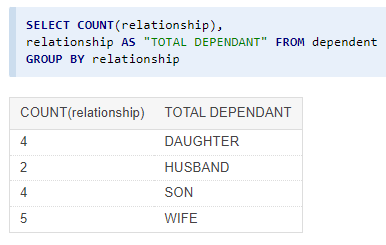
[**SELECT**](https://mariadb.com/kb/en/library/select/) [**AVG**](https://mariadb.com/kb/en/library/group-by-functions/#function_avg)(w.hours) AS "TOTAL JAM KERJA",   
e.fname AS "NAMA",   
[**COUNT**](https://mariadb.com/kb/en/library/group-by-functions/#function_count)(p.pnumber)   
FROM [works\_on](http://localhost/adminer/?server=localhost&username=root&db=company&table=works_on) w   
[**JOIN**](https://mariadb.com/kb/en/library/join/) [employee](http://localhost/adminer/?server=localhost&username=root&db=company&table=employee) e ON e.ssn = w.essn   
[**JOIN**](https://mariadb.com/kb/en/library/join/) [project](http://localhost/adminer/?server=localhost&username=root&db=company&table=project) p ON e.dnum = p.dnum   
GROUP BY e.ssn   
HAVING [**AVG**](https://mariadb.com/kb/en/library/group-by-functions/#function_avg)(w.hours) > 70   
[**AND**](https://mariadb.com/kb/en/library/logical-operators/#operator_and) [**COUNT**](https://mariadb.com/kb/en/library/group-by-functions/#function_count)(p.pnumber) > 2   
ORDER BY [**AVG**](https://mariadb.com/kb/en/library/group-by-functions/#function_avg)(w.hours)  
DESC;

****

1. **Banyaknya Dependent berdasarkan relationship dengan employee ?**

**Syntax :**

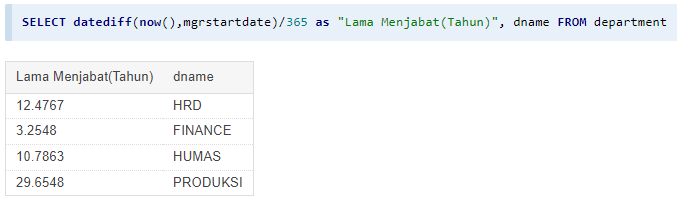
[**SELECT**](https://mariadb.com/kb/en/library/select/) [**COUNT**](https://mariadb.com/kb/en/library/group-by-functions/#function_count)(relationship),   
relationship AS "TOTAL DEPENDANT" FROM [dependent](http://localhost/adminer/?server=localhost&username=root&db=company&table=dependent)   
GROUP BY relationship;

****

1. **Berapa lama Manager tiap Department sudah menjabat ?**

**Syntax :**

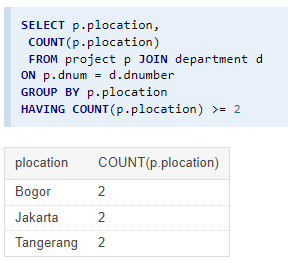
[**SELECT**](https://mariadb.com/kb/en/library/select/) [**datediff**](https://mariadb.com/kb/en/library/date-and-time-functions/#function_datediff)([**now**](https://mariadb.com/kb/en/library/date-and-time-functions/#function_now)(),mgrstartdate), dname FROM [department](http://localhost/adminer/?server=localhost&username=root&db=company&table=department);

****

1. **Lokasi project yang menjadi tempat lebih dari satu department?**

**Syntax :**

[**SELECT**](https://mariadb.com/kb/en/library/select/) p.plocation,  
 [**COUNT**](https://mariadb.com/kb/en/library/group-by-functions/#function_count)(p.plocation)  
 FROM [project](http://localhost/adminer/?server=localhost&username=root&db=company&table=project) p [**JOIN**](https://mariadb.com/kb/en/library/join/) [department](http://localhost/adminer/?server=localhost&username=root&db=company&table=department) d   
ON p.dnum = d.dnumber   
GROUP BY p.plocation   
HAVING [**COUNT**](https://mariadb.com/kb/en/library/group-by-functions/#function_count)(p.plocation) >= 2;

****