

Pertemuan 07

Relationship dalam Mysql

Pendahuluan

Pada suatu saat nanti ketika kita membuat website kita pasti memiliki beberapa tabel yang memiliki relasi, seperti user memiliki sebuah postingan atau postingan dimiliki user, hal ini dapat di selesaikan dengan relationship, sistem relationship ini sudah ada di Mysql tinggal kita pakai.

Kegunaan penggunaan sistem relationship ini kita dapat memiliki konsistensi database, misal user yang memiliki banyak postingan dihapus jadi-nya secara otomatis postingan tersebut tidak ada induk/pemilik-nya nah untuk menyelesaikan masalah ini kita bisa menggunakan fitur relationship agar otomatis dihapus.

Membuat Relationship

Sebelum kita membuat relationship kita terlebih dahulu membuat tabel, tabel pertama merupakan induk-nya dan tabel kedua adalah anak, relationship ini sering disebut hasMany karena si induk memiliki banyak anak dan anak hanya memiliki satu induk, di setiap data di anak harus ada kolom foreign key, foreign key ini digunakan untuk data tersebut induk-nya siapa (biasanya id induk-nya).

Misal kita memiliki tabel users yang memiliki kolom

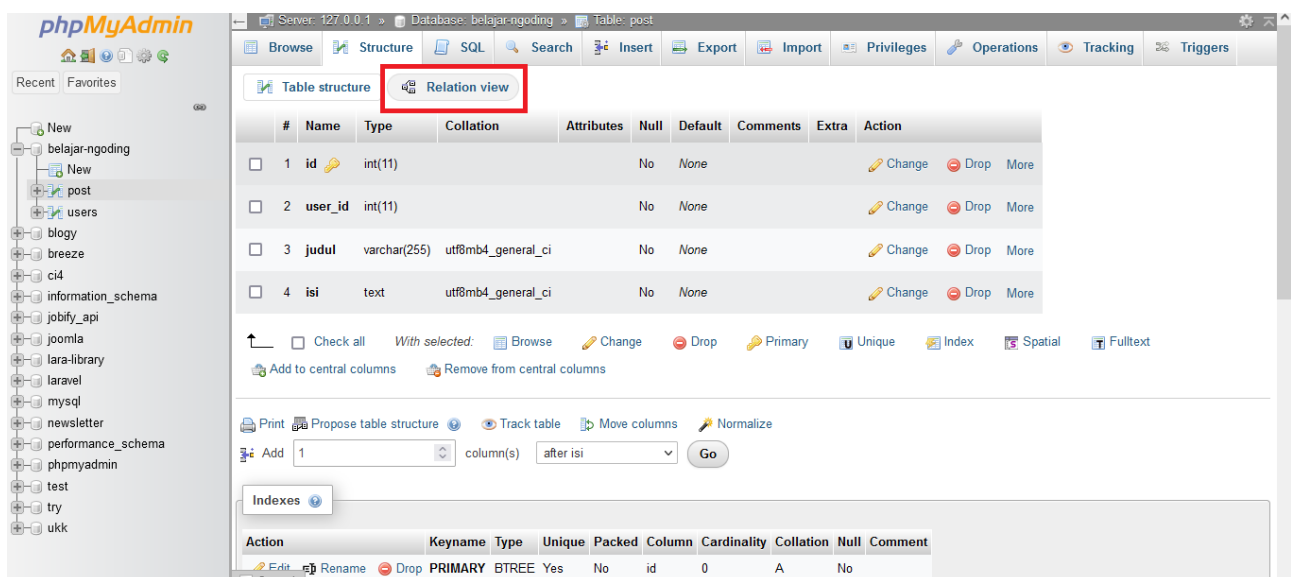
id	Integer	PRIMARY AUTO INCREMENT
nama	Varchar(255)	UNIQUE
email	Varchar(255)	UNIQUE
password	Varchar(255)	

lalu kita juga memiliki tabel posts yang memiliki kolom

id	Integer	PRIMARY AUTO INCREMENT
user_id	Integer	
judul	Varchar(255)	
isi	Text	

Kolom user_id ini digunakan sebagai foreign key, karena post memiliki relasi dengan user kolom user_id ini menjadi integer karena id-nya tabel user juga integer.

Untuk membuat relasi kita klik tabel post lalu klik *Relation view*

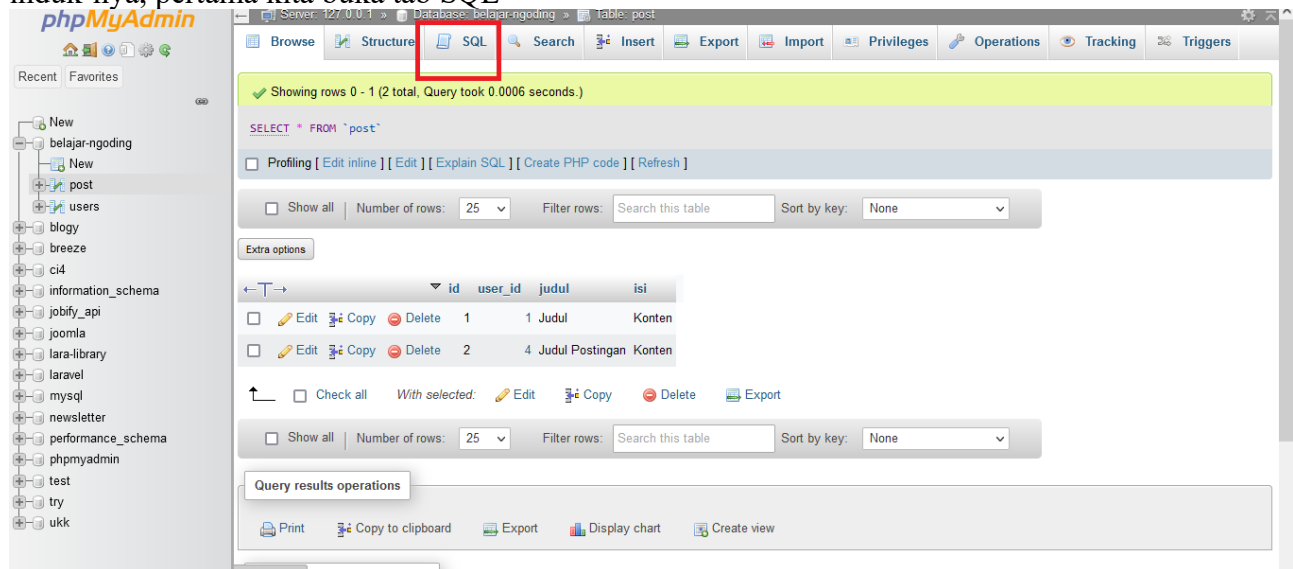


Lalu muncul seperti ini, kita dapat mengisi properti foreign key nya

- A) Nama relasinya (beri nama yang mudah dipahami)
- B) apa yang harus dilakukan ketika si foreign key yang dituju berubah data-nya, kalau ON DELETE CASCADE berarti ketika foreign key yang dituju dihapus data yang mengarah kesana juga ikut kehapus
- C) kolom yang digunakan foreign key
- D) nama foreign key yang dituju (pastikan tabel yang dituju sudah benar dan kolom-nya harus ngarah ke id-nya)

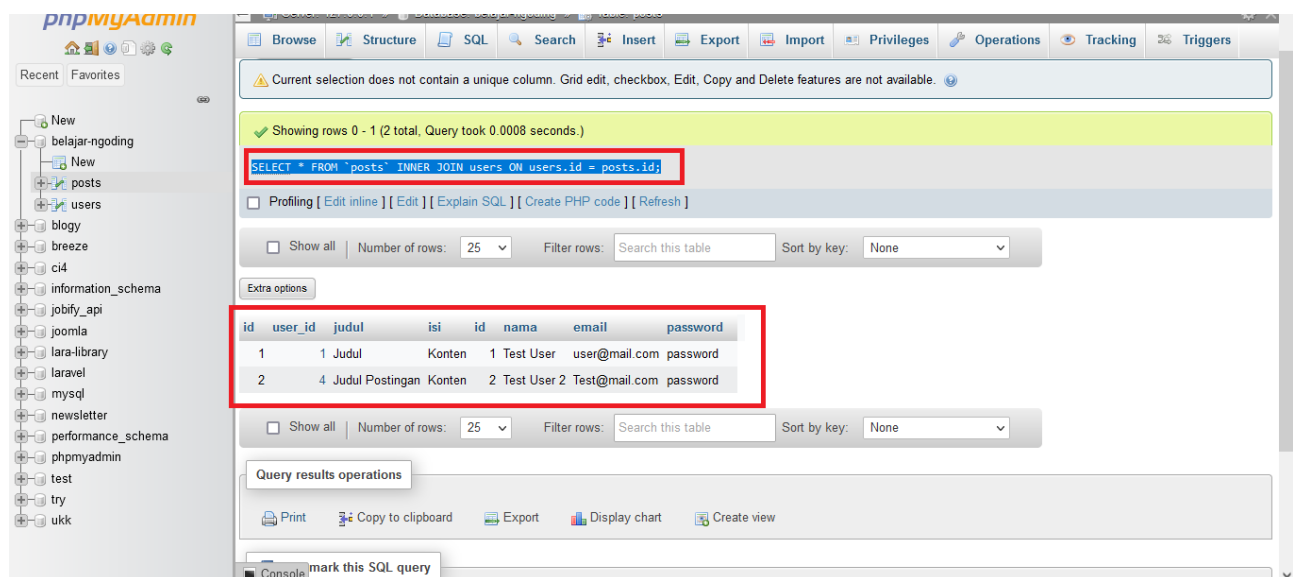
Lalu kita tinggal klik save, ketika kita pergi ke tab insert lalu kita insert data nanti muncul kolom user_id dengan dropdown yang berisi id user yang sudah ada di tabel user

Setelah kita isi beberapa data kedalam tabel post kita bisa meng-query data tersebut sekalian dengan induk-nya, pertama kita buka tab SQL



Lalu kita bisa meng-eksekusi query yang mengambil semua posts dan pemilik-nya

`SELECT * FROM `posts` INNER JOIN users ON users.id = posts.id;`



Nah kalau dilihat lihat itu id-nya ada dua, sekarang gimana cara kita bisa memisah id tersebut, hal ini bisa di hilangkan dengan cara aliasing, aliasing ini seperti mengganti nama kolom dengan nama yang lain, misal kolom nama kita ganti dengan kolom namaPeserta kita tinggal menjalankan query `SELECT users.nama as namaPeserta FROM users;`

Aliasing ini memiliki urutan jadi jangan pernah kebalik
nama_tabel.nama_kolom AS nama_yang_lain

Sekarang kita bisa ganti query yang tadi kita buat untuk mengambil data post-nya agar tidak memiliki data yang bertabrakan

`SELECT users.id as idUser, users.nama, users.email, posts.id as idPost, posts.judul, posts.isi FROM `posts` INNER JOIN users ON users.id = posts.id;`

dengan hasil

phpMyAdmin

Recent Favorites

New belajar-ngoding New posts users blogy breeze ci4 information_schema jobify_api joomla lara-library laravel mysql newsletter performance_schema phpmyadmin test try ukk

Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available.

Showing rows 0 - 1 (2 total, Query took 0.0006 seconds.)

```
SELECT users.id as idUser, users.nama, users.email, posts.id as idPost, posts.judul, posts.isi FROM `posts` INNER JOIN users ON users.id = posts.id;
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all Number of rows: 25 Filter rows: Search this table Sort by key: None

Extra options

idUser	nama	email	idPost	judul	isi
1	Test User	user@mail.com	1	Judul	Konten
2	Test User 2	Test@mail.com	2	Judul Postingan	Konten

Show all Number of rows: 25 Filter rows: Search this table Sort by key: None

Query results operations

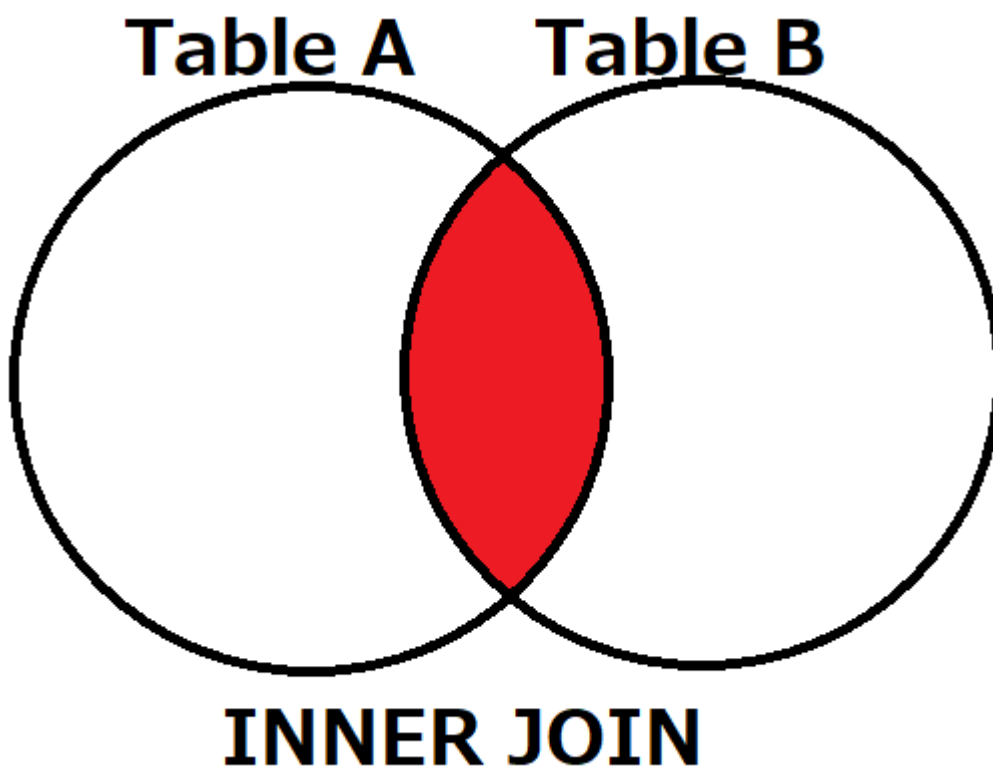
Print Copy to clipboard Export Display chart Create view

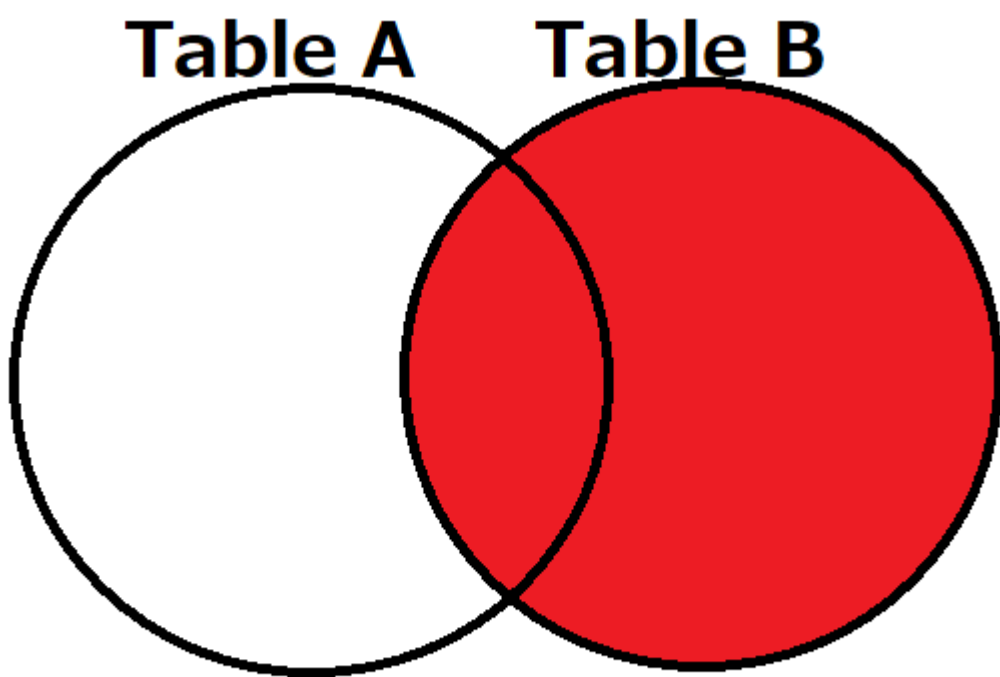
mark this SQL query

Nah sebenarnya keyword JOIN sebenar-nya digunakan untuk menggabungkan suatu tabel, ada beberapa macam JOIN

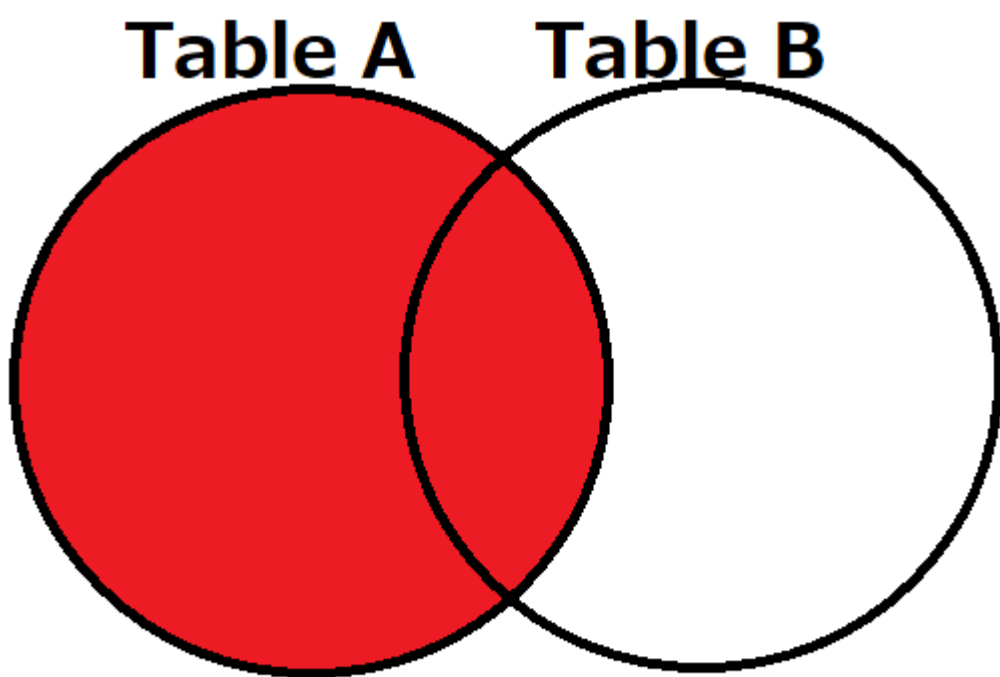
- LEFT JOIN
- RIGHT JOIN
- INNER JOIN
- OUTER JOIN

Penjelasan dengan cara ilustrasi

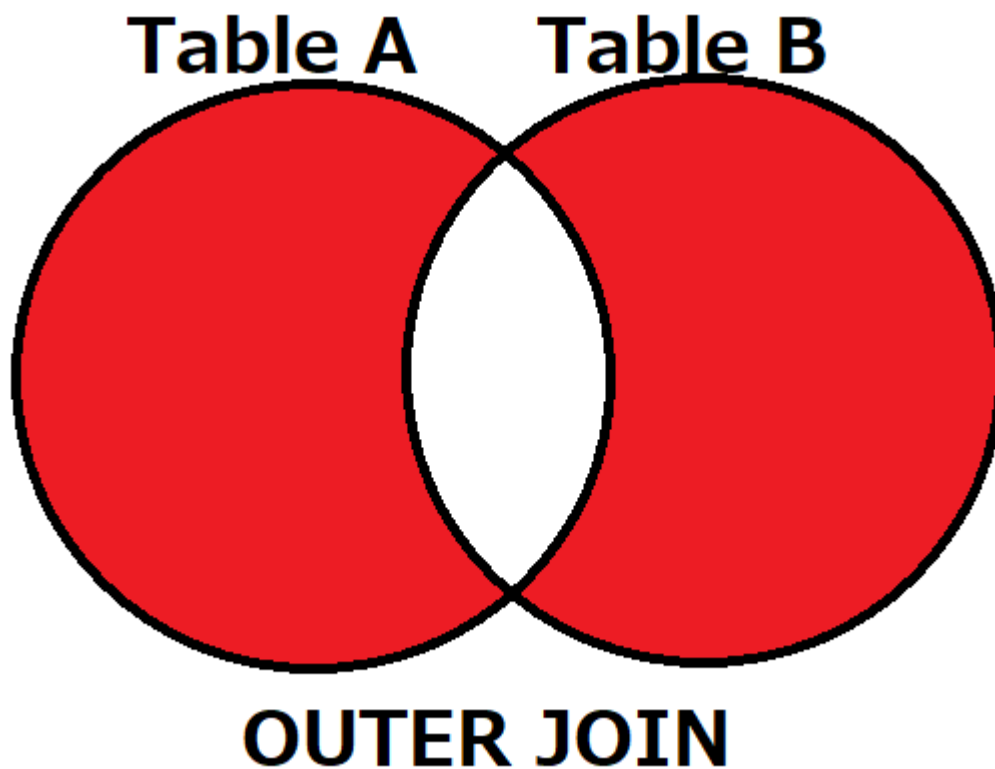




RIGHT JOIN



LEFT JOIN



Tugas :

Membuat website blog yang memiliki tabel users dan posts, setiap user memiliki beberapa posts
Dapat registrasi users, dapat login user, membuat posts, menghapus posts, meng-edit posts.

Ingat setiap user memiliki post yang berbeda jadi setiap perintah hapus dan edit harus di cek apakah post yang dimaksud milik dia.

Tips : Untuk urusan login bisa menggunakan variabel super global \$_SESSION untuk menyimpan data user yang login (misal id user atau nama user)

Batas waktu : 1 minggu