

Praktikum – Bagian 1: Menyiapkan Database untuk Percobaan

Langkah	Keterangan
1	<p>Bersama jobsheet ini disertakan sebuah file bernama percobaan_join.sql, eksekusilah file tersebut pada server MySQL Anda dengan cara apapun yang Anda bisa.</p> <p>Pada contoh di bawah ini, SQL dieksekusi dengan menggunakan perintah SOURCE melalui MySQL Shell. Anda juga dapat mengimpor SQL tersebut melalui PHPMyAdmin atau MySQL Workbench, atau tools yang lain yang Anda sukai.</p>
	<pre>MariaDB [mysql]> source D:/percobaan_join.sql; Query OK, 0 rows affected, 1 warning (0.001 sec) Query OK, 1 row affected (0.002 sec) Database changed Query OK, 0 rows affected (0.054 sec) Query OK, 0 rows affected (0.024 sec) Query OK, 0 rows affected (0.015 sec) Query OK, 0 rows affected (0.013 sec) Query OK, 0 rows affected (0.012 sec) Query OK, 0 rows affected (0.012 sec) Query OK, 0 rows affected (0.011 sec) Query OK, 0 rows affected (0.015 sec) Query OK, 0 rows affected (0.042 sec) Records: 0 Duplicates: 0 Warnings: 0 Query OK, 0 rows affected (0.023 sec) Records: 0 Duplicates: 0 Warnings: 0 Query OK, 0 rows affected (0.025 sec) Records: 0 Duplicates: 0 Warnings: 0 Query OK, 0 rows affected (0.024 sec) Records: 0 Duplicates: 0 Warnings: 0 Query OK, 0 rows affected (0.022 sec) Records: 0 Duplicates: 0 Warnings: 0</pre>
2	<p>Jika Anda telah berhasil mengimpor/mengeksekusi/menjalankan file percobaan_join.sql tersebut, maka di server MySQL Anda akan dibuatkan database dengan struktur seperti berikut.</p>

	<pre> erDiagram dosen --o{ jadwal : "primary key" mk --o{ jadwal : "primary key" ruang --o{ jadwal : "primary key" prodi --o{ kelas : "primary key" kelas --o{ jadwal : "primary key" hari --o{ jadwal : "primary key" jp --o{ jadwal : "primary key" dosen { VARCHAR(4) kode_dosen PK VARCHAR(100) nama_dosen } mk { VARCHAR(5) kode_mk PK VARCHAR(100) nama_mk } ruang { VARCHAR(5) kode_ruang PK VARCHAR(20) nama_ruang VARCHAR(100) deskripsi_ruang } jadwal { INT(10) kode_jadwal PK VARCHAR(10) kode_kelas FK VARCHAR(4) kode_dosen FK VARCHAR(5) kode_mk FK VARCHAR(5) kode_ruang FK VARCHAR(3) kode_hari FK INT(3) jp_mulai FK INT(3) jp_selesai FK } prodi { VARCHAR(3) kode_prodi PK VARCHAR(100) nama_prodi } kelas { VARCHAR(10) kode_kelas PK VARCHAR(3) kode_prodi FK VARCHAR(5) nama_kelas } hari { VARCHAR(3) kode_hari PK VARCHAR(10) nama_hari } jp { INT(3) kode_jp PK TIME jp_mulai TIME jp_selesai } </pre>
3	<p>Pastikan pada database Anda terdapat tabel-tabel seperti di bawah ini dengan menjalankan SQL dibawah. Jika tabel-tabelnya sudah sesuai, lanjutkan ke Praktikum – Bagian 2.</p>
	<pre> MariaDB [percobaan_join]> USE percobaan_join; Database changed MariaDB [percobaan_join]> SHOW TABLES; +-----+ Tables_in_percobaan_join +-----+ dosen hari jadwal jp kelas mk prodi ruang +-----+ 8 rows in set (0.001 sec) MariaDB [percobaan_join]> </pre>

Praktikum – Bagian 2: INNER JOIN

Langkah	Keterangan																																								
1	<p>Apabila kita ingin menampilkan data yang kolom-kolomnya terdapat pada tabel yang berbeda, maka kita dapat menggunakan sintaksis INNER JOIN. Sintaksis ini akan menampilkan nilai kolom pada 2 atau lebih tabel yang saling bersesuaian dalam 1 baris.</p> <p>Contoh: “Tampilkan nama kelas berikut nama prodinya!”</p> <p>Solusi: Jalankan query berikut ini, dan akan ditampilkan 52 baris nama kelas berikut nama prodinya yang bersesuaian (kolom kode_prodi di tabel kelas sama nilainya dengan kolom kode_prodi di tabel prodi).</p> <pre> MariaDB [percobaan_join]> SELECT kelas.nama_kelas, prodi.nama_prodi -> FROM kelas -> INNER JOIN prodi ON kelas.kode_prodi = prodi.kode_prodi; </pre> <table border="1"> <thead> <tr> <th>nama_kelas</th><th>nama_prodi</th></tr> </thead> <tbody> <tr><td>MI-1A</td><td>D3 Manajemen Informatika</td></tr> <tr><td>MI-1B</td><td>D3 Manajemen Informatika</td></tr> <tr><td>MI-1C</td><td>D3 Manajemen Informatika</td></tr> <tr><td>MI-1D</td><td>D3 Manajemen Informatika</td></tr> <tr><td>MI-1E</td><td>D3 Manajemen Informatika</td></tr> <tr><td>MI-1F</td><td>D3 Manajemen Informatika</td></tr> <tr><td>MI-1H</td><td>D3 Manajemen Informatika</td></tr> <tr><td>MI-2A</td><td>D3 Manajemen Informatika</td></tr> <tr><td>MI-2B</td><td>D3 Manajemen Informatika</td></tr> <tr><td>MI-2C</td><td>D3 Manajemen Informatika</td></tr> <tr><td>MI-2D</td><td>D3 Manajemen Informatika</td></tr> <tr><td>MI-2E</td><td>D3 Manajemen Informatika</td></tr> <tr><td>MI-2F</td><td>D3 Manajemen Informatika</td></tr> <tr><td>MI-3A</td><td>D3 Manajemen Informatika</td></tr> <tr><td>MI-3B</td><td>D3 Manajemen Informatika</td></tr> <tr><td>MI-3C</td><td>D3 Manajemen Informatika</td></tr> <tr><td>MI-3D</td><td>D3 Manajemen Informatika</td></tr> <tr><td>MI-3E</td><td>D3 Manajemen Informatika</td></tr> <tr><td>MI-3F</td><td>D3 Manajemen Informatika</td></tr> </tbody> </table>	nama_kelas	nama_prodi	MI-1A	D3 Manajemen Informatika	MI-1B	D3 Manajemen Informatika	MI-1C	D3 Manajemen Informatika	MI-1D	D3 Manajemen Informatika	MI-1E	D3 Manajemen Informatika	MI-1F	D3 Manajemen Informatika	MI-1H	D3 Manajemen Informatika	MI-2A	D3 Manajemen Informatika	MI-2B	D3 Manajemen Informatika	MI-2C	D3 Manajemen Informatika	MI-2D	D3 Manajemen Informatika	MI-2E	D3 Manajemen Informatika	MI-2F	D3 Manajemen Informatika	MI-3A	D3 Manajemen Informatika	MI-3B	D3 Manajemen Informatika	MI-3C	D3 Manajemen Informatika	MI-3D	D3 Manajemen Informatika	MI-3E	D3 Manajemen Informatika	MI-3F	D3 Manajemen Informatika
nama_kelas	nama_prodi																																								
MI-1A	D3 Manajemen Informatika																																								
MI-1B	D3 Manajemen Informatika																																								
MI-1C	D3 Manajemen Informatika																																								
MI-1D	D3 Manajemen Informatika																																								
MI-1E	D3 Manajemen Informatika																																								
MI-1F	D3 Manajemen Informatika																																								
MI-1H	D3 Manajemen Informatika																																								
MI-2A	D3 Manajemen Informatika																																								
MI-2B	D3 Manajemen Informatika																																								
MI-2C	D3 Manajemen Informatika																																								
MI-2D	D3 Manajemen Informatika																																								
MI-2E	D3 Manajemen Informatika																																								
MI-2F	D3 Manajemen Informatika																																								
MI-3A	D3 Manajemen Informatika																																								
MI-3B	D3 Manajemen Informatika																																								
MI-3C	D3 Manajemen Informatika																																								
MI-3D	D3 Manajemen Informatika																																								
MI-3E	D3 Manajemen Informatika																																								
MI-3F	D3 Manajemen Informatika																																								
2	<p>Pada contoh sebelumnya, jika diperhatikan baik-baik, setiap nama kolom yang ingin ditampilkan harus disebutkan nama tabel asalnya dengan notasi dot/titik (nama_tabel.nama_kolom). Hal ini masuk akal karena hal tersebut memiliki tujuan untuk menghilangkan ambiguitas karena bisa saja 2 tabel yang berbeda memiliki kolom dengan nama yang sama.</p> <p>Namun demikian akan sedikit merepotkan apabila kita secara berulang-ulang menuliskan nama tabel di sebelah nama kolom, lagi dan lagi. Apalagi jika nama tabelnya Panjang.</p> <p>Solusinya adalah dengan menggunakan alias yaitu sintaksis AS. Dengan menggunakan sintaksis ini, query kita akan menjadi lebih singkat namun dengan hasil yang sama. Jalankan SQL berikut untuk mengetahui hasilnya!</p>																																								

```
MariaDB [percobaan_join]> SELECT k.nama_kelas, p.nama_prodi
-> FROM kelas k
-> INNER JOIN prodi p ON p.kode_prodi = k.kode_prodi;
```

nama_kelas	nama_prodi
MI-1A	D3 Manajemen Informatika
MI-1B	D3 Manajemen Informatika
MI-1C	D3 Manajemen Informatika
MI-1D	D3 Manajemen Informatika
MI-1E	D3 Manajemen Informatika
MI-1F	D3 Manajemen Informatika
MI-1H	D3 Manajemen Informatika
MI-2A	D3 Manajemen Informatika
MI-2B	D3 Manajemen Informatika
MI-2C	D3 Manajemen Informatika
MI-2D	D3 Manajemen Informatika
MI-2E	D3 Manajemen Informatika
MI-2F	D3 Manajemen Informatika
MI-3A	D3 Manajemen Informatika
MI-3B	D3 Manajemen Informatika
MI-3C	D3 Manajemen Informatika
MI-3D	D3 Manajemen Informatika
MI-3E	D3 Manajemen Informatika
MI-3F	D3 Manajemen Informatika

Pada contoh sebelumnya kita telah menampilkan 2 kolom yang terletak pada 2 tabel yang berbeda namun penggunaan INNER JOIN tidaklah terbatas pada 2 tabel saja. Kita juga dapat menampilkan data yang lebih banyak dari beberapa tabel sekaligus.

Contoh: “Tampilkan nama dosen berikut kelas yang diajar dan harinya!”

Solusi: Jalankan SQL berikut. Jika benar akan ditampilkan **320 baris**. Pada query tersebut melibatkan 4 tabel yaitu tabel jadwal, dosen, kelas, dan hari.

```
MariaDB [percobaan_join]> SELECT
-> j.kode_jadwal,
-> d.nama_dosen,
-> k.nama_kelas,
-> h.nama_hari
-> FROM
-> jadwal j
-> INNER JOIN dosen d ON d.kode_dosen = j.kode_dosen
-> INNER JOIN kelas k ON k.kode_kelas = j.kode_kelas
-> INNER JOIN hari h ON h.kode_hari = j.kode_hari;
```

kode_jadwal	nama_dosen	nama_kelas	nama_hari
1	Abdul Chalim, SAg., MPd.I	TI-1A	Senin
8	Ade Ismail	MI-2F	Senin
17	Ahmadi Yuli Ananta ST., MM.	MI-1F	Senin
18	Ahmadi Yuli Ananta ST., MM.	TI-2C	Senin
21	Ane Fany Novitasari, SH.MKn.	TI-1D	Senin
22	Ane Fany Novitasari, SH.MKn.	TI-1H	Senin
23	Ane Fany Novitasari, SH.MKn.	TI-1F	Senin
38	Anugrah Nur Rahmanto SSn., MDs.	MI-3F	Senin
48	Arie Rachmad Syulistyo SKom., MKom.	MI-1C	Senin
49	Arie Rachmad Syulistyo SKom., MKom.	MI-2B	Senin
57	Arwin Sumari ST., MT., DR.	TI-2D	Senin
75	Budi Harijanto ST., MMKom.	TI-1C	Senin
76	Budi Harijanto ST., MMKom.	TI-2D	Senin
84	Candra Bella Vista SKom., MT.	TI-3B	Senin
85	Candra Bella Vista SKom., MT.	MI-2C	Senin
86	Candra Bella Vista SKom., MT.	MI-2D	Senin
87	Candrasena Setiadi ST., MMT.	TI-3A	Senin
91	Deddy Kusbianto PA Ir. MMKom.	TI-2G	Senin
98	Dian Hanifudin Subhi SKom., MT.	MI-2E	Senin

SELECT JOIN juga bisa difilter. Tentu saja dengan menggunakan klausa WHERE.

Contoh: “Tampilkan nama kelas berikut nama prodinya, hanya untuk kelas yang A saja!”

Solusi: Jalankan SQL berikut ini. Jika benar akan ditampilkan 7 baris yaitu semua kelas dari kelas 1-4 di masing-masing prodi yang namanya diakhiri huruf “A”

```
MariaDB [percobaan_join]> SELECT k.nama_kelas, p.nama_prodi
-> FROM kelas k
-> INNER JOIN prodi p ON p.kode_prodi = k.kode_prodi
-> WHERE k.nama_kelas LIKE '%A';
+-----+-----+
| nama_kelas | nama_prodi |
+-----+-----+
| MI-1A      | D3 Manajemen Informatika |
| MI-2A      | D3 Manajemen Informatika |
| MI-3A      | D3 Manajemen Informatika |
| TI-1A      | D4 Teknik Informatika    |
| TI-2A      | D4 Teknik Informatika    |
| TI-3A      | D4 Teknik Informatika    |
| TI-4A      | D4 Teknik Informatika    |
+-----+-----+
7 rows in set (0.001 sec)
```

5 Lanjutkan ke **Praktikum – Bagian 3.**

Praktikum - Bagian 3: OUTER JOIN

Langkah	Keterangan																																								
1	<p>Jika INNER JOIN hanya menampilkan baris-baris pada 2 atau lebih tabel yang saling bersesuaian. Maka untuk menampilkan data pada tabel yang saling bersesuaian ditambah yang tidak bersesuaian, kita dapat menggunakan sintaksis OUTER JOIN.</p> <p>OUTER JOIN dibagi menjadi 2:</p> <ul style="list-style-type: none"> - LEFT OUTER JOIN dan; - RIGHT OUTER JOIN <p>OUTER JOIN pada umumnya berguna untuk mengecek data yang tidak ada pasangannya di tabel yang di-JOIN-kan.</p> <p>Contoh: “Tampilkan data semua kelas berikut nama prodinya, beserta kelas yang <u>tidak ada prodinya!</u>”</p> <p>Solusi: Jalankan query berikut. Jika benar akan ditampilkan 57 baris data dimana 52 baris adalah nama kelas yang ada prodinya, dan 5 baris sisanya adalah nama kelas yang tidak terdaftar di prodi manapun.</p> <pre> MariaDB [percobaan_join]> SELECT -> k.nama_kelas, -> p.nama_prodi -> FROM kelas k -> LEFT OUTER JOIN prodi p ON k.kode_prodi = p.kode_prodi; </pre> <table border="1"> <thead> <tr> <th>nama_kelas</th><th>nama_prodi</th></tr> </thead> <tbody> <tr><td>MI-1A</td><td>D3 Manajemen Informatika</td></tr> <tr><td>MI-1B</td><td>D3 Manajemen Informatika</td></tr> <tr><td>MI-1C</td><td>D3 Manajemen Informatika</td></tr> <tr><td>MI-1D</td><td>D3 Manajemen Informatika</td></tr> <tr><td>MI-1E</td><td>D3 Manajemen Informatika</td></tr> <tr><td>MI-1F</td><td>D3 Manajemen Informatika</td></tr> <tr><td>MI-1H</td><td>D3 Manajemen Informatika</td></tr> <tr><td>MI-2A</td><td>D3 Manajemen Informatika</td></tr> <tr><td>MI-2B</td><td>D3 Manajemen Informatika</td></tr> <tr><td>MI-2C</td><td>D3 Manajemen Informatika</td></tr> <tr><td>MI-2D</td><td>D3 Manajemen Informatika</td></tr> <tr><td>MI-2E</td><td>D3 Manajemen Informatika</td></tr> <tr><td>MI-2F</td><td>D3 Manajemen Informatika</td></tr> <tr><td>MI-3A</td><td>D3 Manajemen Informatika</td></tr> <tr><td>MI-3B</td><td>D3 Manajemen Informatika</td></tr> <tr><td>MI-3C</td><td>D3 Manajemen Informatika</td></tr> <tr><td>MI-3D</td><td>D3 Manajemen Informatika</td></tr> <tr><td>MI-3E</td><td>D3 Manajemen Informatika</td></tr> <tr><td>MI-3F</td><td>D3 Manajemen Informatika</td></tr> </tbody> </table>	nama_kelas	nama_prodi	MI-1A	D3 Manajemen Informatika	MI-1B	D3 Manajemen Informatika	MI-1C	D3 Manajemen Informatika	MI-1D	D3 Manajemen Informatika	MI-1E	D3 Manajemen Informatika	MI-1F	D3 Manajemen Informatika	MI-1H	D3 Manajemen Informatika	MI-2A	D3 Manajemen Informatika	MI-2B	D3 Manajemen Informatika	MI-2C	D3 Manajemen Informatika	MI-2D	D3 Manajemen Informatika	MI-2E	D3 Manajemen Informatika	MI-2F	D3 Manajemen Informatika	MI-3A	D3 Manajemen Informatika	MI-3B	D3 Manajemen Informatika	MI-3C	D3 Manajemen Informatika	MI-3D	D3 Manajemen Informatika	MI-3E	D3 Manajemen Informatika	MI-3F	D3 Manajemen Informatika
nama_kelas	nama_prodi																																								
MI-1A	D3 Manajemen Informatika																																								
MI-1B	D3 Manajemen Informatika																																								
MI-1C	D3 Manajemen Informatika																																								
MI-1D	D3 Manajemen Informatika																																								
MI-1E	D3 Manajemen Informatika																																								
MI-1F	D3 Manajemen Informatika																																								
MI-1H	D3 Manajemen Informatika																																								
MI-2A	D3 Manajemen Informatika																																								
MI-2B	D3 Manajemen Informatika																																								
MI-2C	D3 Manajemen Informatika																																								
MI-2D	D3 Manajemen Informatika																																								
MI-2E	D3 Manajemen Informatika																																								
MI-2F	D3 Manajemen Informatika																																								
MI-3A	D3 Manajemen Informatika																																								
MI-3B	D3 Manajemen Informatika																																								
MI-3C	D3 Manajemen Informatika																																								
MI-3D	D3 Manajemen Informatika																																								
MI-3E	D3 Manajemen Informatika																																								
MI-3F	D3 Manajemen Informatika																																								
2	<p>Apabila hanya ingin menampilkan data yang tidak ada pasangannya saja, maka kita bisa menggunakan filter melalui penambahan klausa WHERE.</p> <p>Query berikut akan menampilkan data seperti sebelumnya, namun hanya yang tidak ada pasangannya saja.</p>																																								

```
MariaDB [percobaan_join]> SELECT
-> k.nama_kelas,
-> p.nama_prodi
-> FROM
-> kelas k LEFT OUTER JOIN prodi p ON k.kode_prodi = p.kode_prodi
-> WHERE p.kode_prodi IS NULL;
```

```
+-----+-----+
| nama_kelas | nama_prodi |
+-----+-----+
| TRM-1A     | NULL       |
| TRM-1B     | NULL       |
| TRM-1C     | NULL       |
| S2TI-A     | NULL       |
| S2TI-B     | NULL       |
+-----+-----+
5 rows in set (0.001 sec)
```

Selain **LEFT OUTER JOIN** juga ada **RIGHT OUTER JOIN**. Keduanya sama-sama menampilkan data yang bersesuaian ditambah yang tidak bersesuaian. Bedanya adalah **letak data yang TIDAK NULL-nya di sebelah mana**. Tabel kiri atau tabel kanan.

Bagaimana menentukan tabel kiri dan tabel kanan?

- Tabel kiri adalah yang ditulis di sebelah **KIRI** (SEBELUM) kata-kata JOIN.
- Tabel kanan adalah yang ditulis di sebelah **KANAN** (SESUDAH) kata-kata JOIN.

LEFT OUTER JOIN → NULL-nya di tabel kanan, data yang lengkap di tabel **KIRI (LEFT)**

RIGHT OUTER JOIN → NULL-nya di tabel kiri, data yang lengkap di tabel **KANAN (RIGHT)**

Contoh: “Terdapat prodi baru yang belum ada kelasnya, tampilkan nama kelas berikut nama prodinya serta nama prodi-prodi baru yang belum ada kelasnya tersebut!”

Solusi: Jalankan query berikut. Apabila benar akan ditampilkan 55 baris dimana 52 baris adalah data yang bersesuaian (prodi dan nama kelasnya masing-masing), sedangkan 3 baris sisanya adalah prodi baru yang belum ada kelasnya. Perhatikan **data yang lengkap ada di kolom nama_prodi yang merupakan kolom dari tabel prodi yang ditulis di sebelah KANAN (RIGHT) dari kata JOIN** pada sintaksis SQL-nya.

```
MariaDB [percobaan_join]> SELECT
-> k.nama_kelas,
-> p.nama_prodi
-> FROM kelas k
-> RIGHT OUTER JOIN prodi p ON p.kode_prodi = k.kode_prodi;
```

nama_kelas	nama_prodi
MI-1A	D3 Manajemen Informatika
MI-1B	D3 Manajemen Informatika
MI-1C	D3 Manajemen Informatika
MI-1D	D3 Manajemen Informatika
MI-1E	D3 Manajemen Informatika
MI-1F	D3 Manajemen Informatika
MI-1H	D3 Manajemen Informatika
MI-2A	D3 Manajemen Informatika
MI-2B	D3 Manajemen Informatika
MI-2C	D3 Manajemen Informatika
MI-2D	D3 Manajemen Informatika
MI-2E	D3 Manajemen Informatika
MI-2F	D3 Manajemen Informatika
MI-3A	D3 Manajemen Informatika
MI-3B	D3 Manajemen Informatika
MI-3C	D3 Manajemen Informatika
MI-3D	D3 Manajemen Informatika
MI-3E	D3 Manajemen Informatika
MI-3F	D3 Manajemen Informatika
TI-4F	D4 Teknik Informatika
TI-4G	D4 Teknik Informatika
NULL	D4 Sistem Integritas Tinggi
NULL	D4 Kecerdasan Buatan dan Robotika
NULL	D4 Sistem Informasi Bisnis

Jika kita ingin menampilkan semua data yang bersesuaian, ditambah dengan data yang tidak bersesuaian di tabel KANAN dan KIRI sekaligus, maka kita dapat menggunakan sintaksis FULL JOIN.

Pada beberapa DBMS tertentu, sintaksis eksplisit FULL JOIN telah didukung, namun pada MySQL, sintaksis ini belum didukung.

Untuk mengakalinya kita dapat menggunakan sintaksis UNION ALL yang akan menggabungkan 2 buah himpunan hasil SELECT yang berbeda.

Namun jangan lupa bahwa:

- pada sintaksis ini, kedua buah hasil SELECT harus memiliki jumlah kolom yang sama. Jika tidak, maka datanya tidak akan dapat ditampilkan.
- Semikolon (;) harus diletakkan sekali saja di akhir statement SELECT yang paling belakang.
- Jangan gunakan UNION saja karena data yang sama (duplikat) akan dihilangkan.

Jalankan query berikut ini untuk menampilkan hasil FULL join terhadap tabel kelas dan tabel prodi. Apabila benar, maka akan ditampilkan sebanyak 112 baris data dengan rincian:

- 104 data yang lengkap ada prodi dan kelasnya.
- 5 baris data kelas yang tidak ada prodinya
- 3 baris nama prodi yang tidak ada kelasnya


```

MariaDB [percobaan_join]> SELECT k.nama_kelas, p.nama_prodi
-> FROM kelas k
-> LEFT OUTER JOIN prodi p ON k.kode_prodi = p.kode_prodi
-> UNION ALL
-> SELECT k.nama_kelas, p.nama_prodi
-> FROM kelas k
-> RIGHT OUTER JOIN prodi p ON p.kode_prodi = k.kode_prodi;

```

TI-4G	D4 Teknik Informatika
TRM-1A	NULL
TRM-1B	NULL
TRM-1C	NULL
S2TI-A	NULL
S2TI-B	NULL

TI-4G	D4 Teknik Informatika
NULL	D4 Sistem Integritas Tinggi
NULL	D4 Kecerdasan Buatan dan Robotika
NULL	D4 Sistem Informasi Bisnis

112 rows in set (0.001 sec)

5 Lanjutkan ke **Praktikum – Bagian 5**.

Praktikum - Bagian 4: INNER JOIN Implisit dan CROSS JOIN

Langkah	Keterangan																																						
1	<p>Sintaksis JOIN yang kita pelajari sebelumnya merupakan sintaksis ANSI SQL yang lebih baru. Sekedar pengetahuan saja, bahwa sebelum distandarkannya format sintaksis tersebut, sebelumnya JOIN dilakukan dengan menggunakan sintaksis yang tidak ada kata JOIN-nya. Format ini disebut sebagai IMPLICIT JOIN. Sedangkan format sintaksis kita sebelumnya disebut sebagai EXPLICIT JOIN.</p> <p>Kita dianjurkan untuk menggunakan format yang baru, yaitu yang ada kata JOIN-nya karena cenderung lebih jelas dan menghindari terjadinya kesalahan maksud pada SQL yang kita tulis. Dengan menuliskan kata-kata JOIN, berarti kita secara sadar memang ingin menampilkan data dari 2 tabel atau lebih.</p> <p>Namun demikian untuk sekedar pengetahuan Anda, jalankan sintaksis SQL berikut, hasilnya akan sama dengan hasil pada Praktikum – Bagian 1 Langkah 1.</p> <p>Perhatikan pada sintaksis ini tidak ada kata-kata “JOIN”-nya.</p> <pre> MariaDB [percobaan_join]> SELECT kelas.nama_kelas, prodi.nama_prodi -> FROM kelas, prodi -> WHERE kelas.kode_prodi = prodi.kode_prodi; </pre> <table border="1"> <thead> <tr> <th>nama_kelas</th><th>nama_prodi</th></tr> </thead> <tbody> <tr><td>MI-1A</td><td>D3 Manajemen Informatika</td></tr> <tr><td>MI-1B</td><td>D3 Manajemen Informatika</td></tr> <tr><td>MI-1C</td><td>D3 Manajemen Informatika</td></tr> <tr><td>MI-1D</td><td>D3 Manajemen Informatika</td></tr> <tr><td>MI-1E</td><td>D3 Manajemen Informatika</td></tr> <tr><td>MI-1F</td><td>D3 Manajemen Informatika</td></tr> <tr><td>MI-1H</td><td>D3 Manajemen Informatika</td></tr> <tr><td>MI-2A</td><td>D3 Manajemen Informatika</td></tr> <tr><td>MI-2B</td><td>D3 Manajemen Informatika</td></tr> <tr><td>MI-2C</td><td>D3 Manajemen Informatika</td></tr> <tr><td>MI-2D</td><td>D3 Manajemen Informatika</td></tr> <tr><td>MI-2E</td><td>D3 Manajemen Informatika</td></tr> <tr><td>MI-2F</td><td>D3 Manajemen Informatika</td></tr> <tr><td>MI-3A</td><td>D3 Manajemen Informatika</td></tr> <tr><td>MI-3B</td><td>D3 Manajemen Informatika</td></tr> <tr><td>MI-3C</td><td>D3 Manajemen Informatika</td></tr> <tr><td>MI-3D</td><td>D3 Manajemen Informatika</td></tr> <tr><td>MI-3E</td><td>D3 Manajemen Informatika</td></tr> </tbody> </table>	nama_kelas	nama_prodi	MI-1A	D3 Manajemen Informatika	MI-1B	D3 Manajemen Informatika	MI-1C	D3 Manajemen Informatika	MI-1D	D3 Manajemen Informatika	MI-1E	D3 Manajemen Informatika	MI-1F	D3 Manajemen Informatika	MI-1H	D3 Manajemen Informatika	MI-2A	D3 Manajemen Informatika	MI-2B	D3 Manajemen Informatika	MI-2C	D3 Manajemen Informatika	MI-2D	D3 Manajemen Informatika	MI-2E	D3 Manajemen Informatika	MI-2F	D3 Manajemen Informatika	MI-3A	D3 Manajemen Informatika	MI-3B	D3 Manajemen Informatika	MI-3C	D3 Manajemen Informatika	MI-3D	D3 Manajemen Informatika	MI-3E	D3 Manajemen Informatika
nama_kelas	nama_prodi																																						
MI-1A	D3 Manajemen Informatika																																						
MI-1B	D3 Manajemen Informatika																																						
MI-1C	D3 Manajemen Informatika																																						
MI-1D	D3 Manajemen Informatika																																						
MI-1E	D3 Manajemen Informatika																																						
MI-1F	D3 Manajemen Informatika																																						
MI-1H	D3 Manajemen Informatika																																						
MI-2A	D3 Manajemen Informatika																																						
MI-2B	D3 Manajemen Informatika																																						
MI-2C	D3 Manajemen Informatika																																						
MI-2D	D3 Manajemen Informatika																																						
MI-2E	D3 Manajemen Informatika																																						
MI-2F	D3 Manajemen Informatika																																						
MI-3A	D3 Manajemen Informatika																																						
MI-3B	D3 Manajemen Informatika																																						
MI-3C	D3 Manajemen Informatika																																						
MI-3D	D3 Manajemen Informatika																																						
MI-3E	D3 Manajemen Informatika																																						
2	<p>INNER JOIN Implisit juga bisa diberikan alias nama tabel agar tidak terlalu Panjang.</p> <p>Alias nama tabel dituliskan pada klausa FROM, sama seperti pada Explicit JOIN.</p> <p>Jalankan query berikut. Hasilnya akan sama dengan langkah sebelumnya, namun dengan penulisan SQL yang lebih singkat.</p>																																						

	<pre> MariaDB [percobaan_join]> SELECT k.nama_kelas, p.nama_prodi -> FROM kelas k, prodi p -> WHERE k.kode_prodi = p.kode_prodi; -----+----- nama_kelas nama_prodi -----+----- MI-1A D3 Manajemen Informatika MI-1B D3 Manajemen Informatika MI-1C D3 Manajemen Informatika MI-1D D3 Manajemen Informatika MI-1E D3 Manajemen Informatika MI-1F D3 Manajemen Informatika MI-1H D3 Manajemen Informatika MI-2A D3 Manajemen Informatika MI-2B D3 Manajemen Informatika MI-2C D3 Manajemen Informatika MI-2D D3 Manajemen Informatika MI-2E D3 Manajemen Informatika MI-2F D3 Manajemen Informatika MI-3A D3 Manajemen Informatika MI-3B D3 Manajemen Informatika MI-3C D3 Manajemen Informatika MI-3D D3 Manajemen Informatika MI-3E D3 Manajemen Informatika MI-3F D3 Manajemen Informatika </pre>
	<p>Selain INNER JOIN, OUTER JOIN, dan FULL JOIN, terdapat satu jenis JOIN lagi yaitu CROSS JOIN.</p> <p>CROSS JOIN akan menampilkan kombinasi satu-satu dari setiap kolom pada semua tabel tanpa mempertimbangkan kesesuaian antar tabel atau tidak.</p> <p>Jumlah baris yang dihasilkan adalah perkalian jumlah baris pada kedua buah tabel yang di CROSS-kan.</p>
3	<pre> MariaDB [percobaan_join]> SELECT k.nama_kelas, p.nama_prodi -> FROM kelas k CROSS JOIN prodi p; -----+----- nama_kelas nama_prodi -----+----- MI-1A D3 Manajemen Informatika MI-1A D4 Teknik Informatika MI-1A D4 Sistem Integritas Tinggi MI-1A D4 Kecerdasan Buatan dan Robotika MI-1A D4 Sistem Informasi Bisnis MI-1B D3 Manajemen Informatika MI-1B D4 Teknik Informatika MI-1B D4 Sistem Integritas Tinggi MI-1B D4 Kecerdasan Buatan dan Robotika MI-1B D4 Sistem Informasi Bisnis MI-1C D3 Manajemen Informatika MI-1C D4 Teknik Informatika MI-1C D4 Sistem Integritas Tinggi MI-1C D4 Kecerdasan Buatan dan Robotika MI-1C D4 Sistem Informasi Bisnis </pre>
4	<p>CROSS JOIN juga dapat dilakukan secara IMPLICIT yaitu dengan tanpa memberikan WHERE pada sintaksis JOIN IMPLICIT.</p> <p>Jalankan query di bawah ini, hasilnya akan sama dengan langkah sebelumnya.</p>

```
MariaDB [percobaan_join]> SELECT k.nama_kelas, p.nama_prodi  
-> FROM kelas k, prodi p;
```

nama_kelas	nama_prodi
MI-1A	D3 Manajemen Informatika
MI-1A	D4 Teknik Informatika
MI-1A	D4 Sistem Integritas Tinggi
MI-1A	D4 Kecerdasan Buatan dan Robotika
MI-1A	D4 Sistem Informasi Bisnis
MI-1B	D3 Manajemen Informatika
MI-1B	D4 Teknik Informatika
MI-1B	D4 Sistem Integritas Tinggi
MI-1B	D4 Kecerdasan Buatan dan Robotika
MI-1B	D4 Sistem Informasi Bisnis
MI-1C	D3 Manajemen Informatika
MI-1C	D4 Teknik Informatika
MI-1C	D4 Sistem Integritas Tinggi
MI-1C	D4 Kecerdasan Buatan dan Robotika
MI-1C	D4 Sistem Informasi Bisnis

Tugas

1. Jalankan semua SQL pada praktikum-praktikum di atas, pahami maksudnya dan *Screenshot*-lah hasilnya!
2. Tampilkan nama dosen berikut mata kuliah yang mereka ampu (186 baris) dengan ketentuan:
 - a. Tidak ada data yang duplikat
 - b. Urut berdasarkan nama dosen dari A-Z.
 - c. Nama kolom yang ditampilkan harus sesuai dengan contoh di bawah.

```
MariaDB [percobaan_join]> SELECT DISTINCT d.nama_dosen, m.nama_mk
-> FROM dosen d
-> INNER JOIN jadwal j ON d.kode_dosen = j.kode_dosen
-> INNER JOIN mk m ON j.kode_mk = m.kode_mk
-> ORDER BY d.nama_dosen ASC;
```

nama_dosen	nama_mk
Abdul Chalim, SAg., MPd.I	Agama
Ade Ismail	Sistem Manajemen Basis Data
Ade Ismail	Sistem Informasi
Ade Ismail	Pemrograman Web Lanjut
Agung Nugroho Pramudhita ST., MT.	Digital Entrepreneurship
Agung Nugroho Pramudhita ST., MT.	Komputasi Multimedia
Ahmadi Yuli Ananta ST., MM.	Proyek 1_P1
Ahmadi Yuli Ananta ST., MM.	Proyek 2_P2
Ahmadi Yuli Ananta ST., MM.	Penulisan Ilmiah
Ane Fany Novitasari, SH.MKn.	Kewarganegaraan
Annisa Puspa Kirana MKom.	Basis Data
Annisa Puspa Kirana MKom.	Praktikum Basis Data
Annisa Puspa Kirana MKom.	Sistem Manajemen Basis Data
Annisa Taufika Firdausi ST., MT.	Sistem Operasi
Annisa Taufika Firdausi ST., MT.	Desain Pemrograman Web
Anugrah Nur Rahmanto SSn., MDs.	Digital Entrepreneurship
Anugrah Nur Rahmanto SSn., MDs.	Sistem Informasi
Anugrah Nur Rahmanto SSn., MDs.	Sistem Operasi
Ariadi Retno Ririd SKom., MKom.	Rekayasa Perangkat Lunak
Ariadi Retno Ririd SKom., MKom.	Analisis Dan Desan Berorientasi Objek

3. Tampilkan ruang yang digunakan untuk perkuliahan pada hari 'Selasa' berikut jam-nya (79 baris) dengan ketentuan:
 - a. Harus ditampilkan ruangan tersebut dipakai mulai dari jam berapa sampai dengan jam berapa.
 - b. Nama kolom yang ditampilkan harus sesuai contoh di bawah.

```

MariaDB [percobaan_join]> SELECT j.kode_ruang, r.nama_ruang, h.nama_hari, l.jp_mulai, m.jp_selesai
-> FROM jadwal j
-> INNER JOIN ruang r ON j.kode_ruang = r.kode_ruang
-> INNER JOIN hari h ON j.kode_hari = h.kode_hari
-> INNER JOIN jp l ON j.jp_mulai = l.kode_jp
-> INNER JOIN jp m ON j.jp_selesai = m.kode_jp
-> WHERE h.nama_hari = 'Selasa';

```

kode_ruang	nama_ruang	nama_hari	jp_mulai	jp_selesai
0506	RT06	Selasa	14:30:00	17:10:00
0508	LPY1	Selasa	07:00:00	10:30:00
0508	LPY1	Selasa	07:00:00	10:30:00
0617	LSI2	Selasa	12:50:00	18:00:00
0502	RT02	Selasa	07:00:00	09:30:00
0806	RT10	Selasa	09:40:00	12:10:00
0701	LPR1	Selasa	07:00:00	09:30:00
0705	LPR5	Selasa	07:50:00	12:10:00
0505	RT05	Selasa	12:50:00	15:30:00
0505	RT05	Selasa	12:50:00	15:30:00
0713	LKJ2	Selasa	12:50:00	15:30:00
0718	LIG2	Selasa	07:00:00	11:20:00
0619	LPY2	Selasa	07:00:00	11:20:00
0704	LPR4	Selasa	14:30:00	17:10:00
0805	RT09	Selasa	09:40:00	12:10:00
0805	RT09	Selasa	12:50:00	15:20:00
0720	LAI1	Selasa	08:40:00	11:20:00
0619	LPY2	Selasa	12:50:00	18:00:00
0707	LKJ1	Selasa	07:00:00	09:30:00
0713	LKJ2	Selasa	07:00:00	12:10:00
0501	RT01	Selasa	09:40:00	12:10:00
0501	RT01	Selasa	09:40:00	12:10:00
0504	RT04	Selasa	07:00:00	10:30:00
0503	RT03	Selasa	13:40:00	17:10:00
0503	RT03	Selasa	13:40:00	17:10:00

4. Tampilkan dosen yang tidak mendapatkan jadwal mengajar! Catatan: Nama kolom harus sesuai contoh.

```

MariaDB [percobaan_join]> SELECT d.nama_dosen, j.kode_jadwal
-> FROM dosen d
-> LEFT OUTER JOIN jadwal j ON d.kode_dosen = j.kode_dosen
-> WHERE j.kode_jadwal IS NULL;

```

nama_dosen	kode_jadwal
Kamado Tanjiro, S.Kom., M.Kom.	NULL
Rei Ayanami, S.ST., M.Sc.	NULL
Soryu Asuka Langley, M.Eng, Ph.D.	NULL

3 rows in set (0.004 sec)

```

MariaDB [percobaan_join]>

```

5. Tampilkan nama mata kuliah berikut dosen pengampunya berikut (192 baris) yang tidak ada dosen pengampunya! Ketentuan:
- Nama Kolom harus sesuai contoh
 - Anda bisa menggunakan RIGHT OUTER JOIN agar lebih mudah

```

MariaDB [percobaan_join]> SELECT m.nama_mk, d.nama_dosen
  -> FROM mk m
  -> LEFT OUTER JOIN jadwal j ON m.kode_mk = j.kode_mk
  -> LEFT OUTER JOIN dosen d ON j.kode_dosen = d.kode_dosen;

```

nama_mk	nama_dosen
Agama	Abdul Chalim, SAg., MPd.I
Agama	Abdul Chalim, SAg., MPd.I
Agama	Abdul Chalim, SAg., MPd.I
Agama	Abdul Chalim, SAg., MPd.I
Agama	Abdul Chalim, SAg., MPd.I
Agama	Abdul Chalim, SAg., MPd.I
Agama	Abdul Chalim, SAg., MPd.I
Agama	Moh. Amin
Agama	Moh. Amin
Agama	Moh. Amin
Agama	Moh. Amin
Agama	Moh. Amin
Agama	Moh. Amin
Agama	Moh. Amin
Agama	Moh. Amin
Teknologi Data	Yoppy Yunhasnawa, SST., MSc.
Teknologi Data	Yoppy Yunhasnawa, SST., MSc.
Cyber Physical System	NULL
Komputasi Awan	NULL
Komputasi Kuantum	NULL
Swarm Robotics	NULL
Swarm Robotics	NULL
Collaborative Thought	NULL
Matematika Transendental	NULL

377 rows in set (0.001 sec)

6. Tampilkan nama-nama dosen (88 baris) berikut jumlah jadwal mereka! Ketentuan:
- Nama kolom harus sesuai contoh
 - Apabila ada dosen yang tidak mendapatkan jadwal, jumlah_jadwal_mengajarnya haruslah = 0

```

MariaDB [percobaan_join]> SELECT d.nama_dosen, COUNT(j.kode_jadwal) AS jumlah_jadwal_mengajar
  -> FROM dosen d
  -> LEFT OUTER JOIN jadwal j
  -> ON d.kode_dosen = j.kode_dosen
  -> GROUP BY d.nama_dosen
  -> ORDER BY jumlah_jadwal_mengajar DESC;

```

nama_dosen	jumlah_jadwal_mengajar
Ane Fany Novitasari, SH.MKn.	8
Moh. Amin	8
Siti Komariah, Dra., M.H.	1
Soryu Asuka Langley, M.Eng, Ph.D.	0
Rei Ayanami, S.ST., M.Sc.	0
Kamado Tanjiro, S.Kom., M.Kom.	0

88 rows in set (0.006 sec)

-- Selamat Mengerjakan --