RELASI BASIS DATA

TABLE RELATIONSHIP

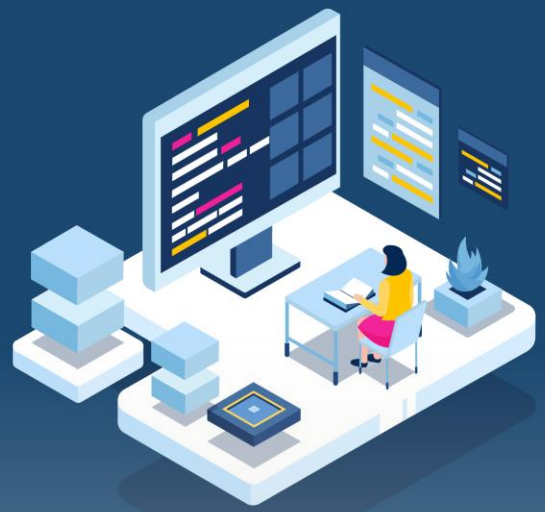


TABLE RELATIONSHIP

- Dalam Relational DBMS, salah satu fitur andalan nya adalah table relationship. Yaitu relasi antar tabel
- Kita bisa melakukan relasi dari satu tabel ke tabel lain.
- Dalam kehidupan nyata pun pasti kita akan sering membuat relasi antar tabel
- Misal, saat kita membuat aplikasi penjualan, di laporan penjualan pasti ada data barang. Jika di tabel artinya tabel penjualan akan berelasi dengan tabel barang
- Misal dalam aplikasi kampus, tabel mahasiswa akan berelasi dengan tabel mata kuliah, dan tabel dosen
- Dan lain-lain

FOREIGN KEY

- Saat membuat relasi tabel, biasanya kita akan membuat sebuah kolom sebagai referensi ke tabel lainnya
- Misal saat kita membuat tabel penjualan, di dalam tabel penjualan, kita akan menambahkan kolom id_produk sebagai referensi ke tabel produk, yang berisi primary key di tabel produk
- Kolom referensi ini di MySQL dinamakan Foreign Key
- Kita bisa menambah satu satu lebih foreign key ke dalam sebuah tabel
- Membuat foreign key sama seperti membuat kolom biasanya, hanya saja kita perlu memberi tahu MySQL bahwa itu adalah foreign key ke tabel lain

MEMBUAT TABLE DENGAN FOREIGN KEY

Active Connection | Execute | Run on active connection | Select block

```
1  CREATE TABLE country (
2   id int PRIMARY Key AUTO_INCREMENT,
3   name_country varchar(15)
4   )Engine=Innodb;
```

[124] ✓ 0.0s

fieldCount	affectedRows	insertId	info	serverStatus	warningStatus
0	0	0		2	0

Active Connection | Execute | Run on active connection | Select block

```
1  CREATE TABLE city (
2    id int(11) NOT NULL AUTO_INCREMENT,
3    name_city VARCHAR(15),
4    country_id int(11) NOT NULL,
5    PRIMARY KEY (`id`),
6    KEY country_id  (`country_id`),
7    CONSTRAINT country_id FOREIGN KEY (`country_id`) REFERENCES `country` (`id`)
8  )ENGINE=Innodb;
```

[125] ✓ 0.0s

fieldCount	affectedRows	insertId	info	serverStatus	warningStatus
0	0	0		2	0

KEUNTUNGAN MENGGUNAKAN FOREIGN KEY

- Foreign key memastikan bahwa data yang kita masukkan ke kolom tersebut harus tersedia di tabel reference nya
- Selain itu saat kita menghapus data di tabel reference, MySQL akan mengecek apakah id nya digunakan di foreign key di tabel lain, jika digunakan, maka secara otomatis MySQL akan menolak proses delete data di tabel reference tersebut

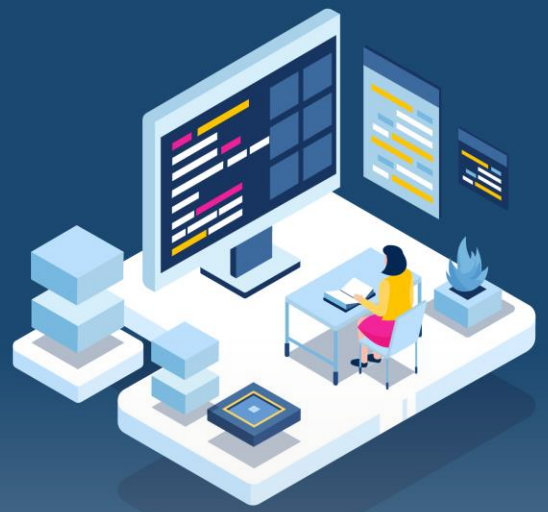
KETIKA MENGHAPUS DATA BERELASI

- Seperti yang sebelumnya dibahas, ketika kita menghapus data yang berelasi, maka secara otomatis MySQL akan menolak operasi delete tersebut
- Kita bisa mengubah fitur ini jika kita mau, ada banyak hal yang bisa dilakukan ketika data berelasi dihapus, defaultnya memang akan ditolak (RESTRICT)

BEHAVIOR FOREIGN KEY

Behavior	ON DELETE	ON UPDATE
RESTRICT	Ditolak	Ditolak
CASCADE	Data akan dihapus	Data akan ikut diubah
NO ACTION	Data dibiarkan	Data dibiarkan
SET NULL	Diubah jadi NULL	Diubah jadi NULL

JOIN

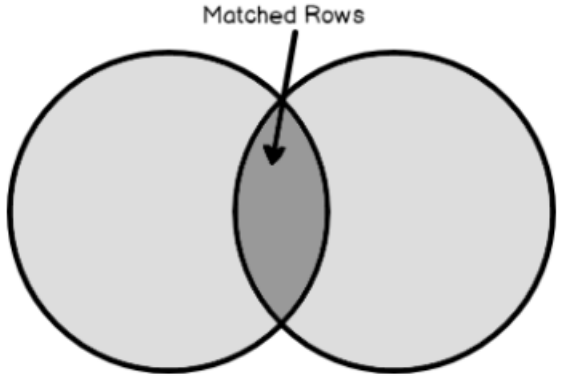


JOIN

- MySQL mendukung query SELECT langsung ke beberapa tabel secara sekaligus
- Namun untuk melakukan itu, kita perlu melakukan JOIN di SQL SELECT yang kita buat
- Untuk melakukan JOIN, kita perlu menentukan tabel mana yang merupakan referensi ke tabel lain
- Join cocok sekali dengan foreign key, walaupun di MySQL tidak ada aturan kalau JOIN harus ada foreign key
- Join di MySQL bisa dilakukan untuk lebih dari beberapa tabel
- Tapi ingat, semakin banyak JOIN, maka proses query akan semakin berat dan lambat, jadi harap bijak ketika melakukan JOIN
- Idealnya kita melakukan JOIN jangan lebih dari 5 tabel, karena itu bisa berdampak ke performa query yang lambat

INNER JOIN

Fungsi INNER JOIN



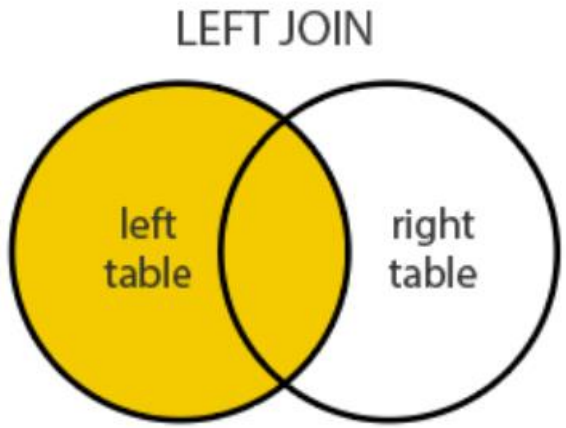
SQL Inner Join

INNER JOIN membandingkan record di setiap table untuk dicek apakah nilai sama atau tidak. Jika nilai kedua table sama, maka akan terbentuk table baru yang hanya menampilkan record yang sama dari kedua table.

```
SELECT column_name(s)
FROM table1
INNER JOIN table2
ON table1.column_name = table2.column_name;
```

LEFT JOIN

Fungsi LEFT JOIN

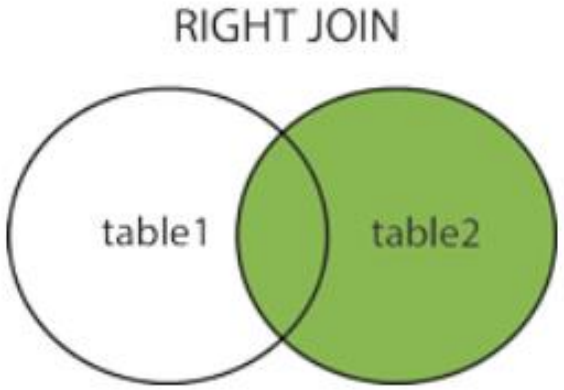


LEFT JOIN menghasilkan nilai berdasarkan table kiri (left table) dan nilai yang sama di table kanan (right table). Jika table kanan tidak sama nilainya dengan table kiri, maka akan diisi nilai NULL pada table kanan.

```
SELECT column_name(s)
FROM table1
LEFT JOIN table2
ON table1.column_name = table2.column_name;
```

RIGHT JOIN

Fungsi RIGHT JOIN



Konsep RIGHT JOIN hampir sama seperti LEFT JOIN hanya yang menjadi master adalah table kanan (right table). Jika table kiri tidak sama nilainya dengan table kanan, maka akan diisi nilai NULL pada table kiri.

```
SELECT column_name(s)
FROM table1
RIGHT JOIN table2
ON table1.column_name = table2.column_name;
```


PRACTICE....