



RELATIONSHIP

ONE TO ONE RELATIONSHIP

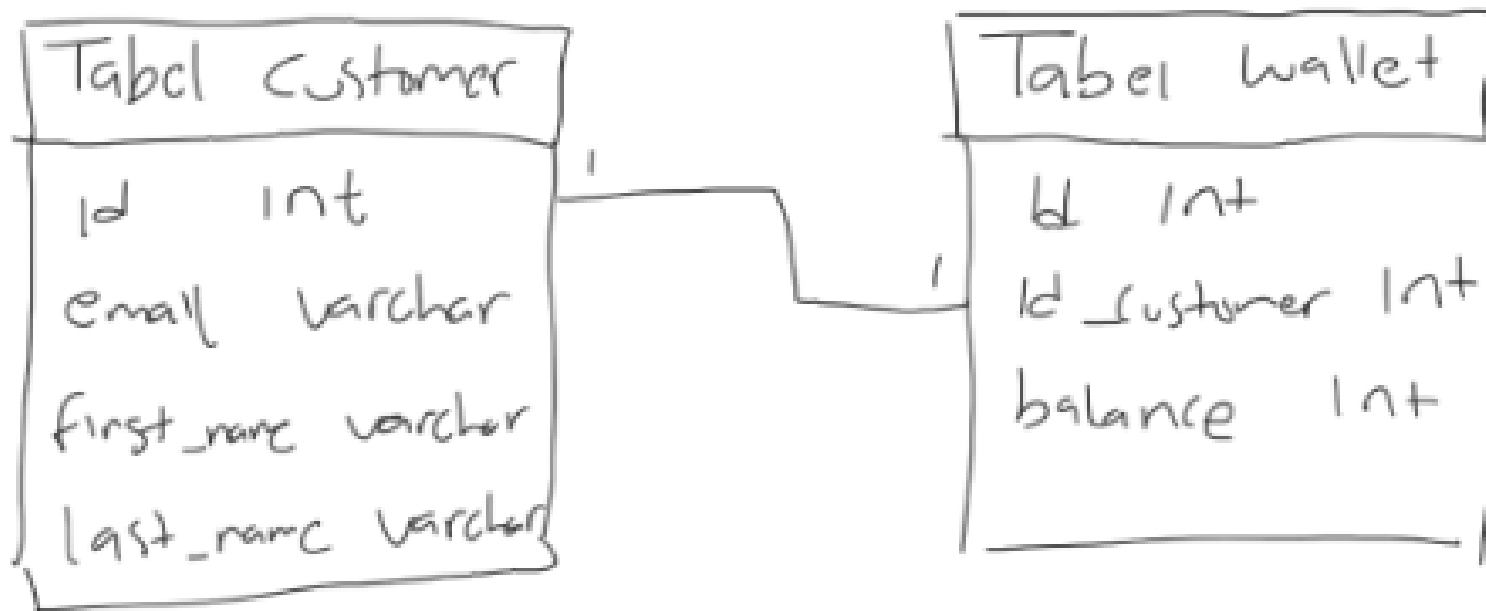
JENIS-JENIS RELASI TABEL

- Sekarang kita sudah tau untuk melakukan relasi antar tabel, kita bisa menggunakan FOREIGN KEY
- Dan untuk melakukan SELECT beberapa tabel, kita bisa menggunakan JOIN
- Dalam konsep relasi, ada banyak jenis-jenis relasi antar tabel
- Sekarang kita akan bahas dari yang pertama yaitu One to One relationship

ONE TO ONE RELATIONSHIP

- One to One relationship adalah relasi antar tabel yang paling sederhana
- Artinya tiap data di sebuah tabel hanya boleh berelasi ke maksimal 1 data di tabel lainnya
- Tidak boleh ada relasi lebih dari 1 data
- Contoh misal, kita membuat aplikasi toko online yang terdapat fitur wallet, dan 1 customer, cuma boleh punya 1 wallet

DIAGRAM ONE TO ONE RELATIONSHIP



MEMBUAT ONE TO ONE RELATIONSHIP

- Cara membuat One to One relationship cukup mudah
- Kita bisa membuat kolom foreign key, lalu set kolom tersebut menggunakan UNIQUE KEY, hal ini dapat mencegah terjadi data di kolom tersebut agar tidak duplikat
- Atau cara lainnya, kita bisa membuat tabel dengan primary key yang sama, sehingga tidak butuh lagi kolom untuk FOREIGN KEY

MEMBUAT TABLE WALLET

```
CREATE TABLE wallet
(
    id          INT NOT NULL AUTO_INCREMENT,
    id_customer INT NOT NULL,
    balance     INT NOT NULL DEFAULT 0,
    PRIMARY KEY (id),
    UNIQUE KEY fk_id_customer_unique (id_customer),
    CONSTRAINT fk_wallet_customer
        FOREIGN KEY (id_customer) REFERENCES customers (id)
) ENGINE = InnoDB;
```



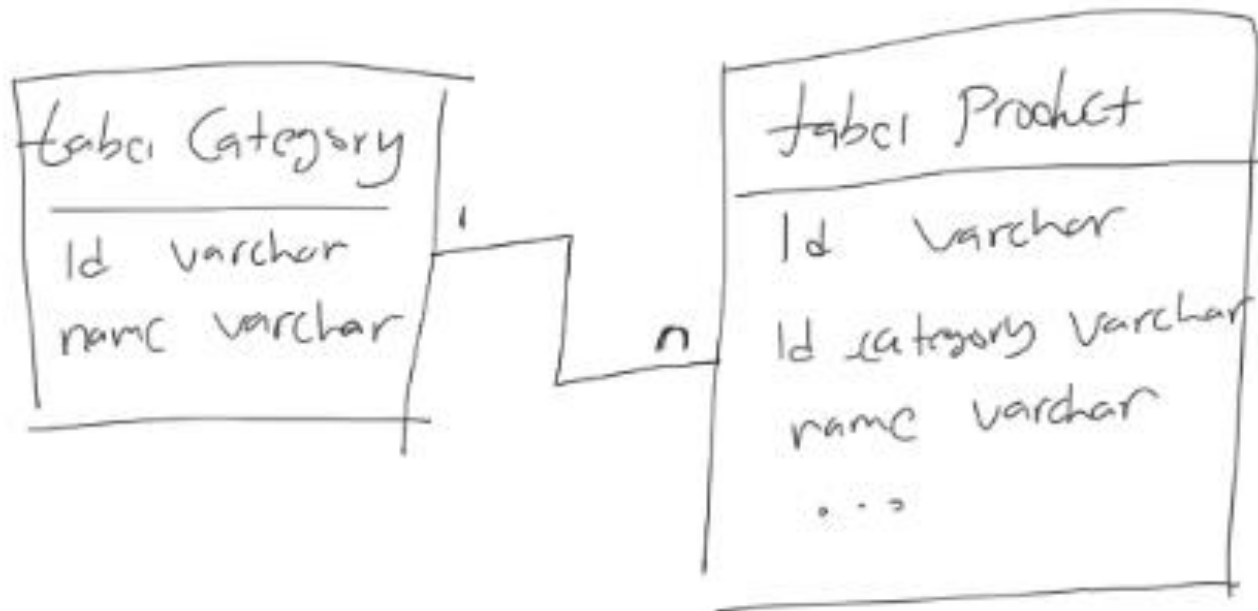

RELATIONSHIP

ONE TO MANY RELATIONSHIP

ONE TO MANY RELATIONSHIP

- One to many relationship adalah relasi antar tabel dimana satu data bisa digunakan lebih dari satu kali di tabel relasinya
- Berbeda dengan one to one yang cuma bisa digunakan maksimal 1 kali di tabel relasinya, one to many tidak ada batasan berapa banyak data digunakan
- Contoh relasi antar tabel categories dan products, dimana satu category bisa digunakan oleh lebih dari satu product, yang artinya relasinya nya one category to many products
- Pembuatan relasi one to many sebenarnya sama dengan one to one, yang membedakan adalah, kita tidak perlu menggunakan UNIQUE KEY, karena datanya memang bisa berkali-kali ditambahkan di tabel relasi nya

DIAGRAM ONE TO MANY RELATIONSHIP



MEMBUAT TABLE CATEGORY

```
CREATE TABLE categories
(
    id    VARCHAR(10) NOT NULL,
    name  VARCHAR(100) NOT NULL,
    PRIMARY KEY (id)
) ENGINE = InnoDB;
```

MENGUBAH TABEL PRODUCT

```
ALTER TABLE products
  DROP column category;

ALTER TABLE products
  ADD COLUMN id_category VARCHAR(100);

ALTER TABLE products
  ADD CONSTRAINT fk_product_category
  FOREIGN KEY (id_category) REFERENCES categories (id);
```



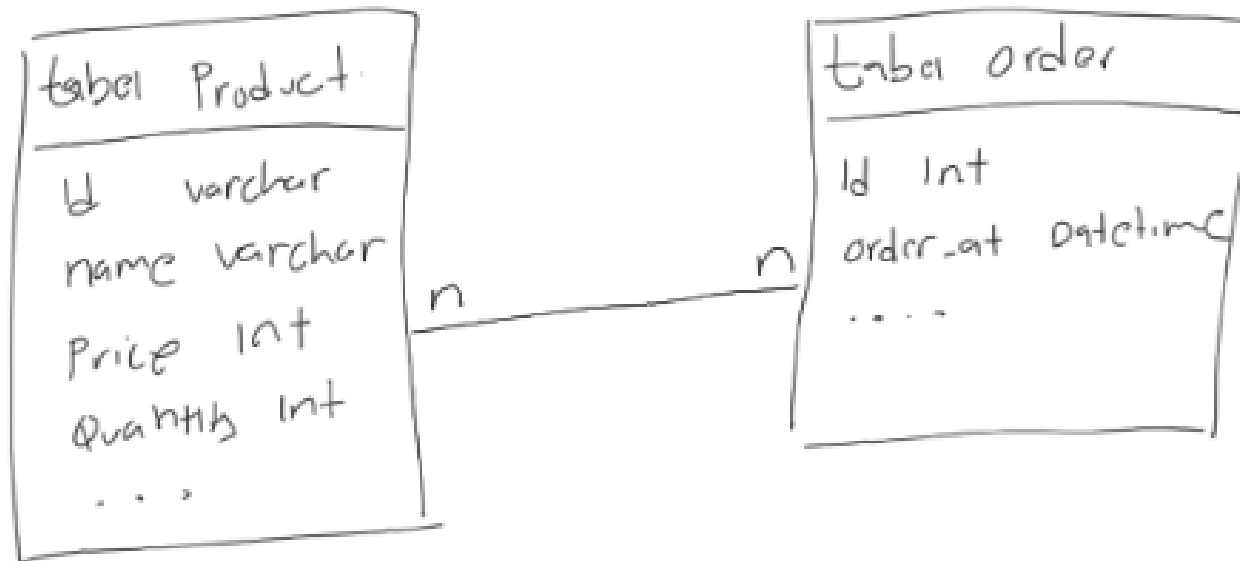
RELATIONSHIP

MANY TO MANY RELATIONSHIP

MANY TO MANY RELATIONSHIP

- Many to Many adalah table relationship yang paling kompleks, dan kadang membingungkan untuk pemula
- Many to Many adalah relasi dimana ada relasi antara 2 tabel dimana table pertama bisa punya banyak relasi di table kedua, dan table kedua pun punya banyak relasi di table pertama
- Ini memang sedikit membingungkan, bagaimana caranya bisa relasi kebanyakan secara bolak balik, sedangkan di table kita cuma punya 1 kolom?
- Contoh relasi many to many adalah relasi antara produk dan penjualan, dimana setiap produk bisa dijual berkali kali, dan setiap penjualan bisa untuk lebih dari satu produk

DIAGRAM MANY TO MANY RELATIONSHIP



BAGAIMANA IMPLEMENTASI MANY TO MANY?

- Sekarang pertanyaannya, bagaimana implementasi many to many?
- Apakah kita harus menambahkan id_order di table products? atau id_product di table orders?

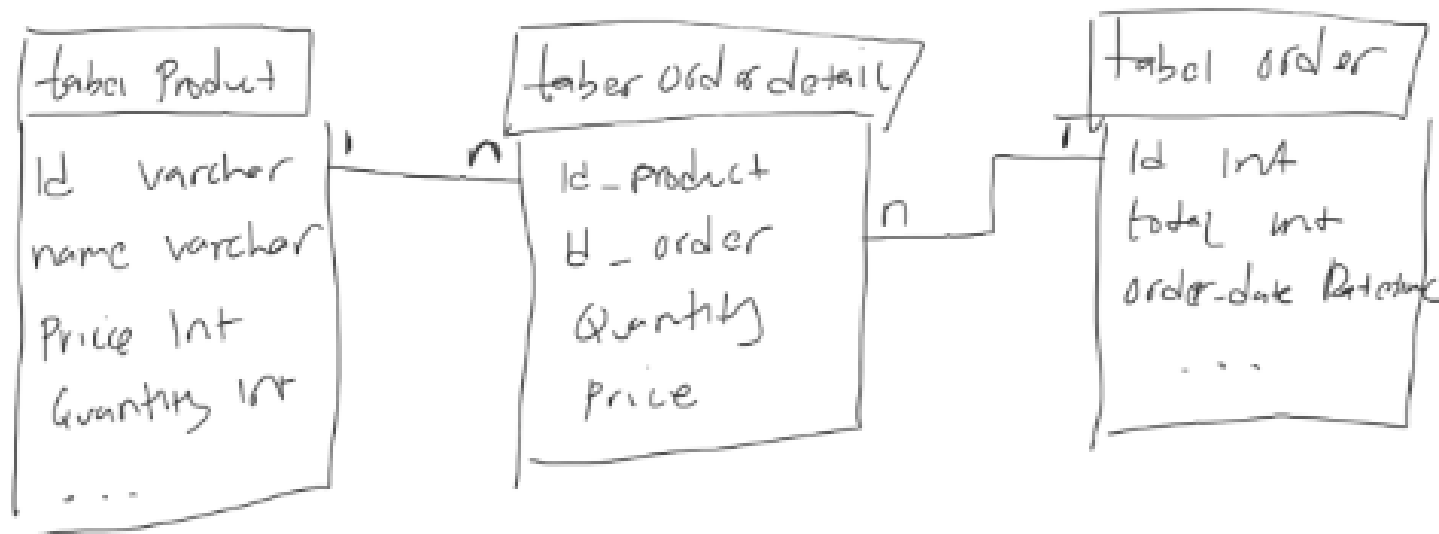
ID PRODUCT DI TABLE ORDER

- Jika kita menambahkan id_product di table orders, artinya sekarang sudah benar, bahwa 1 product bisa dijual berkali-kali
- Namun masalahnya adalah, berarti 1 order hanya bisa membeli 1 product, karena cuma ada 1 kolom untuk id_product
- Oke kalo gitu kita tambahkan id_product1, id_product2, dan seterusnya. Solusi ini bisa dilakukan, tapi tidak baik, artinya akan selalu ada maksimal barang yang bisa kita beli dalam satu order

ID ORDER DI TABLE PRODUCT

- Jika kita tambahkan id_order di table products, artinya sekarang 1 order bisa membeli lebih dari 1 product, oke sudah benar
- Tapi sayangnya masalahnya terbalik sekarang, 1 product cuma bisa dijual satu kali, tidak bisa dijual berkali-kali, karena kolom id_order nya cuma 1
- Kalupun kita tambah id_order1, id_order2 dan seterusnya di table product, tetap ada batasan maksimal nya
- Lantai bagaimana solusinya untuk relasi many to many?

DIAGRAM MANY TO MANY RELATIONSHIP



MEMBUAT TABLE ORDER

```
CREATE TABLE orders
(
    id          INT          NOT NULL AUTO_INCREMENT,
    total       INT          NOT NULL,
    order_date  DATETIME     NOT NULL DEFAULT CURRENT_TIMESTAMP,
    PRIMARY KEY (id)
) ENGINE = InnoDB;
```

MEMBUAT TABLE ORDER DETAIL

```
CREATE TABLE orders_detail
(
    id_product VARCHAR(10) NOT NULL,
    id_order    INT         NOT NULL,
    price       INT         NOT NULL,
    quantity    INT         NOT NULL,
    PRIMARY KEY (id_product, id_order)
) ENGINE = InnoDB;
```

MEMBUAT FOREIGN KEY

```
ALTER TABLE orders_detail
  ADD CONSTRAINT fk_orders_detail_product
    FOREIGN KEY (id_product) REFERENCES products (id);

ALTER TABLE orders_detail
  ADD CONSTRAINT fk_orders_detail_order
    FOREIGN KEY (id_order) REFERENCES orders (id);
```

MELIHAT DATA ORDER, DETAIL DAN PRODUCT-NYA

```
SELECT *  
FROM orders  
    JOIN orders_detail ON (orders_detail.id_order = orders.id)  
    JOIN products ON (products.id = orders_detail.id_product);
```




THANK YOU