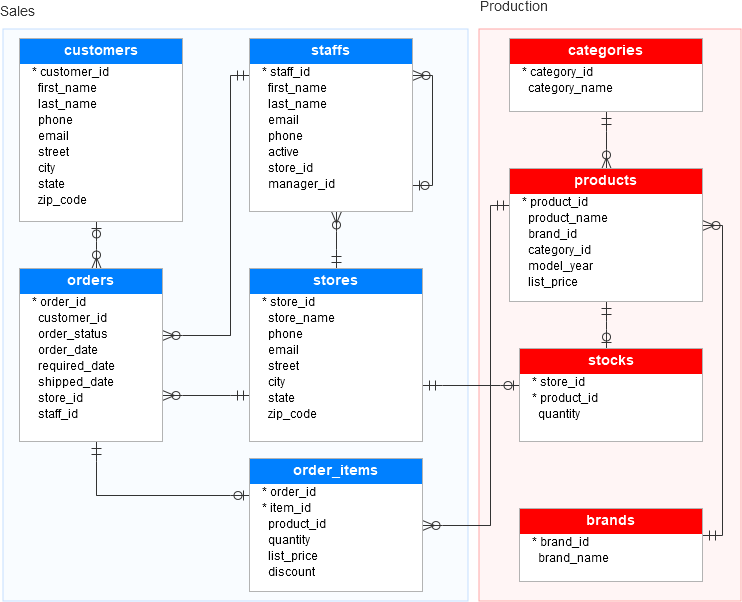
# KUIS 1 BASIS DATA LANJUT

**(Studi Kasus Toko Sepeda)**

Toko Fast merupakan toko yang menjual segala jenis sepeda. Gambar dibawah ini merupakan gambar desain database diagram Toko Fast



# DETAIL DATABASE

**Tabel sales.stores**

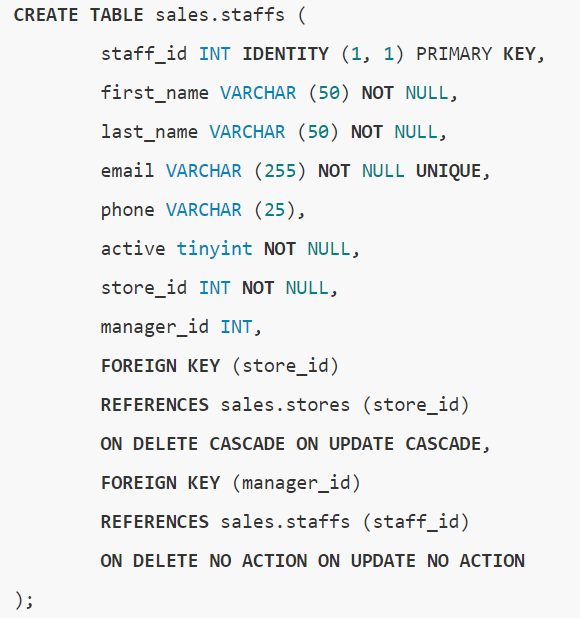
**Tabel sales.stores** berisi informasi toko. Setiap toko memiliki nama toko, informasi kontak seperti telepon dan email, dan alamat termasuk jalan, kota, negara bagian, dan kode pos. Berikut ini detail source code untuk mebuat tabel sales.stores



# Tabel sales.staffs

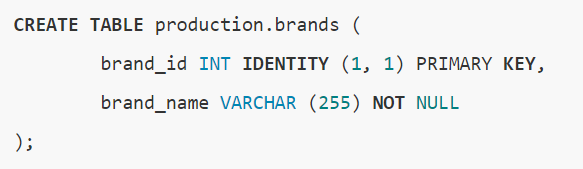
**Tabel sales.staffs** menyimpan informasi penting dari staf termasuk nama depan, nama belakang. Ini juga berisi informasi komunikasi seperti email dan telepon. Seorang staf bekerja di toko yang ditentukan oleh nilai di kolom store\_id. Sebuah toko dapat memiliki satu atau lebih staf. Seorang staf melapor kepada manajer toko yang ditentukan oleh nilai di kolom manager\_id. Jika nilai di manager\_id adalah null, maka staf adalah berperan sebagai top manajer. Jika staf tidak lagi bekerja untuk penyimpanan mana pun, nilai di kolom aktif di set ke nilai zero/nol.

Berikut ini detail source code untuk mebuat tabel sales.staffs



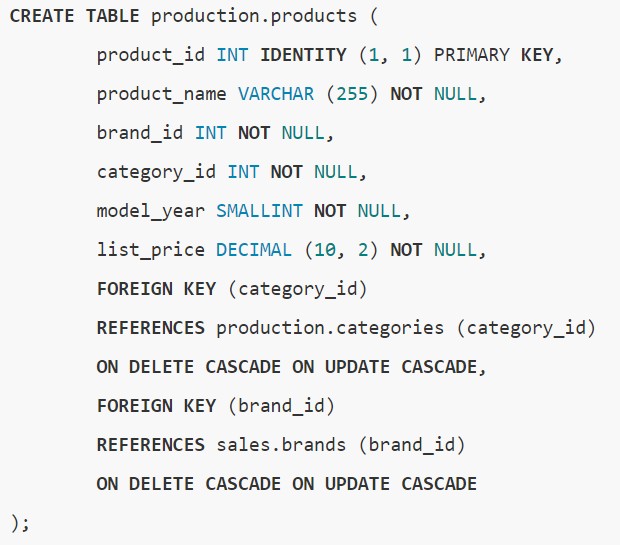
# Tabel production.brands

**Tabel production.brands** menyimpan data tentang informasi merk sepeda contohnya Electra, Haro, and Heller. Berikut ini detail source code untuk mebuat tabel production.brands



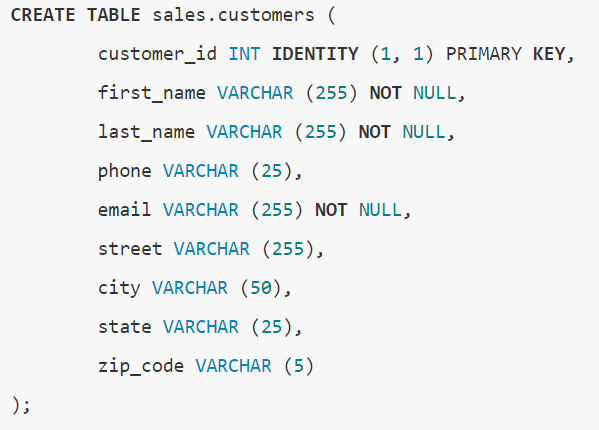
# Tabel production.products

**Tabel production.products** menyimpan informasi produk seperti nama, merek, kategori, model tahun, dan daftar harga. Setiap produk milik merek yang ditentukan oleh kolom brand\_id. Karenanya, sebuah merek mungkin memiliki nol atau banyak produk. Setiap produk juga termasuk dalam kategori yang ditentukan oleh kolom category\_id. Selain itu, setiap kategori mungkin memiliki nol atau banyak produk. Berikut ini detail source code untuk mebuat tabel production.products



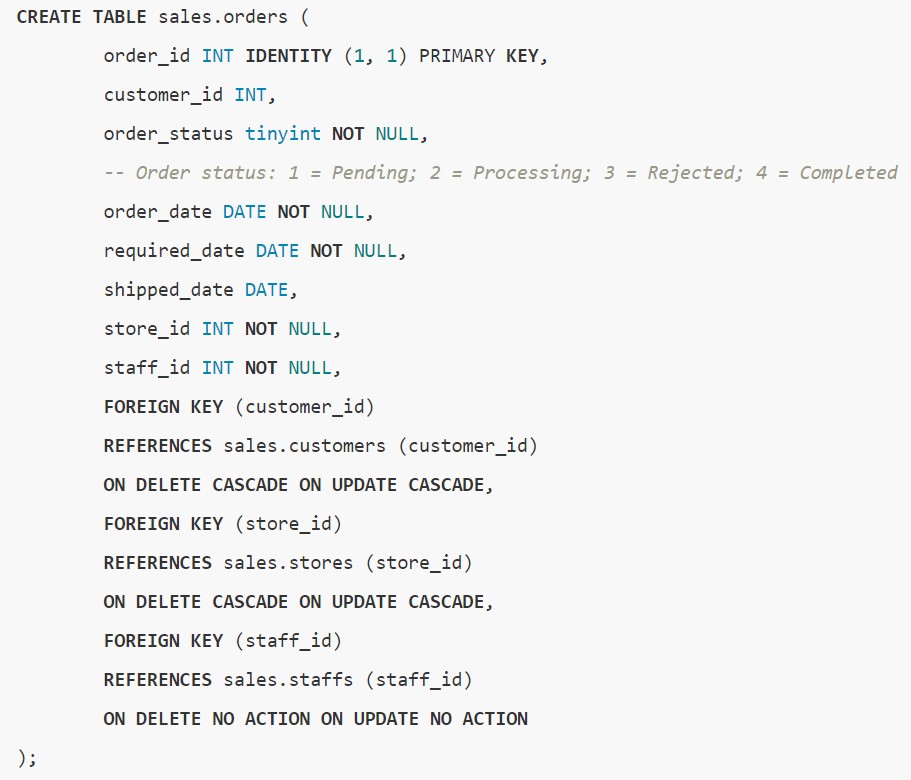
# Tabel sales.customers

**Tabel sales.customers** menyimpan informasi tabel pelanggan termasuk nama depan, nama belakang, telepon, email, jalan, kota, negara dan kode pos. Berikut ini detail source code untuk mebuat tabel sales.customers



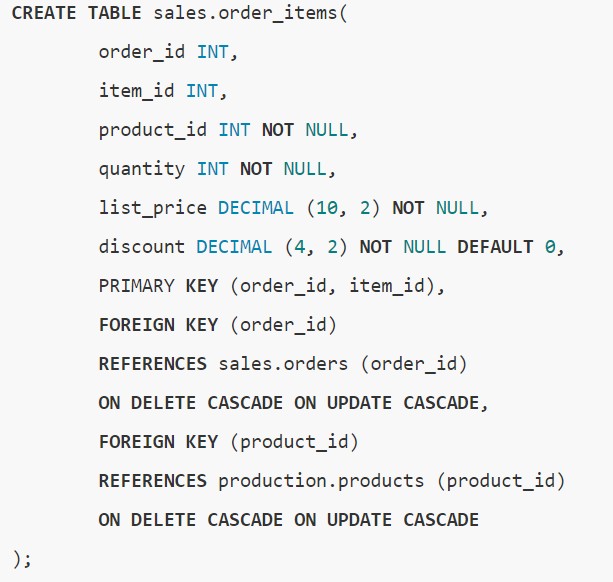
# Tabel sales.orders

Tabel sales.orders menyimpan informasi header order penjualan ini termasuk pelanggan, status order, tanggal order, tanggal yang dibutuhkan, tanggal dikirim. Ini juga menyimpan informasi di mana transaksi penjualan dibuat (toko) dan siapa yang membuatnya (staf). Setiap pesanan penjualan memiliki baris di tabel sales\_orders. Pesanan penjualan memiliki satu atau banyak item baris yang disimpan di tabel sales.order\_items. Berikut ini detail source code untuk mebuat tabel sales.orders



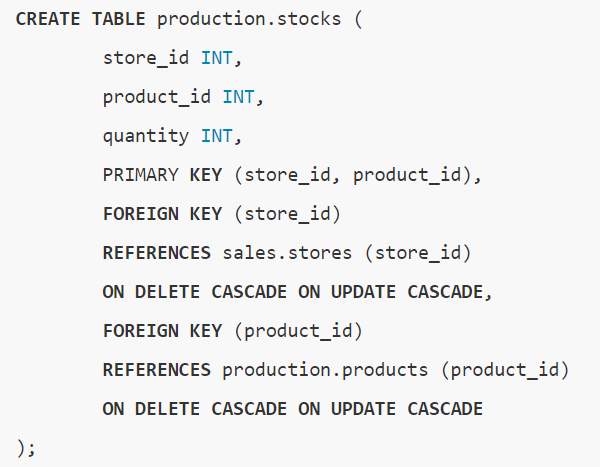
# Tabel sales.orders\_items

Tabel sales.order\_items menyimpan item baris dari order penjualan. Setiap item baris milik pesanan penjualan yang ditentukan oleh kolom order\_id. Item baris pesanan penjualan mencakup produk, kuantitas pesanan, harga jual, dan diskon. Berikut ini detail source code untuk mebuat tabel sales.orders\_items



# Tabel production.stocks

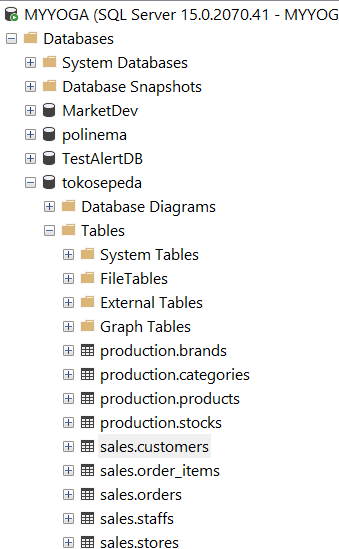
production.stocks menyimpan informasi persediaan yaitu kuantitas produk tertentu di toko tertentu. Berikut ini detail source code untuk mebuat tabel production.stocks



# Tugas

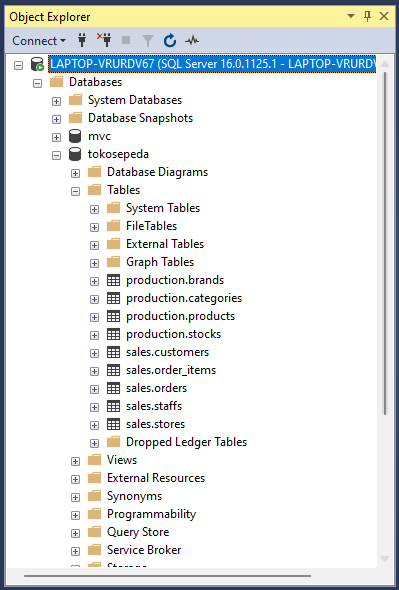
1. Silahkan jalankan source code berikut ini di SQL Server Anda. Source code ini berfungsi untuk membuat database toko sepeda dengan detail yang sudah dijelaskan di penjelasan sebelumnya

**Link download :** [**http://bit.ly/dbtokosepeda**](http://bit.ly/dbtokosepeda)



*Capture hasil running source setup yang terbentuk contoh*

Hasilnya :

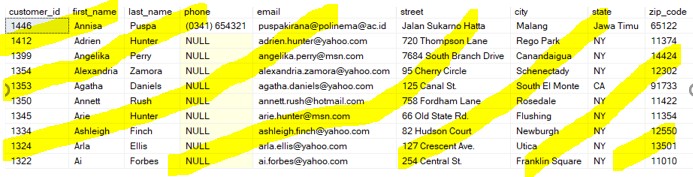
****

1. Silahkan lakukan insert 1 data baru pada **tabel sales.customers** dimana detail nama, nomor telepon, dan email disamakan dengan data diri Anda. Capture source code dan hasil insert Anda

|  |
| --- |
| INSERT INTO sales.customers(first\_name, last\_name, phone, email, street, city, state, zip\_code) VALUES('Lavina','Cantik',087765776316,'lavina@gmail.com','Jl. Perusahaan ','Malang','JT',12345); |
|  |

1. Tampilkan seluruh kolom dari **tabel sales.customers** dengan ketentuan huruf awal di nama depan mereka sama dengan nama depan Anda kemudian urutkan berdasarkan customer\_id yang paling besar terlebih dahulu dan tampilkan hanya 10 data teratas. Capture source code dan hasil SELECT Anda

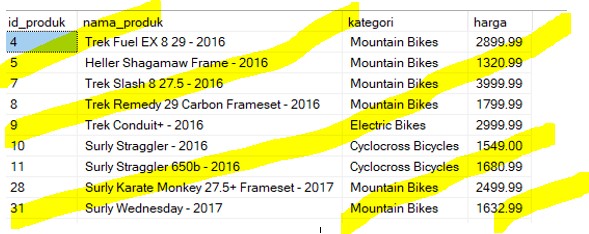
Contoh hasil :



|  |
| --- |
| SELECT TOP 10 \* FROM sales.customers  WHERE first\_name LIKE 'L%' ORDER BY customer\_id DESC; |
|  |

1. Tuliskan perintah SELECT antar 2 tabel. Tampilkan kolom product\_id, product\_name, list\_price dari tabel production.products dan kolom category name dari tabel production.categories. Ubahlah tampilan nama kolomnya sehingga kolom product\_id menjadi id\_produk, product\_name menjadi nama\_produk, category\_name menjadi kategori, list\_price menjadi harga. Data yang ditampilkan adalah produk yang memiliki harga lebih besar atau sama dengan 1000 dan urutkan berdasar product\_id terkecil. Silahkan lakukan dengan menggunakan perintah **JOIN**.

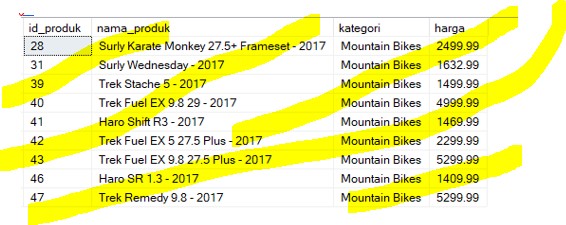
Contoh hasil :

…dst

|  |
| --- |
| SELECT p.product\_id AS id\_produk, p.product\_name AS nama\_produk, c.category\_name AS kategori, p.list\_price AS harga  FROM production.products AS p  JOIN production.categories AS c ON p.category\_id = c.category\_id  WHERE p.list\_price >= 1000 ORDER BY product\_id ASC; |
|  |

1. Berdasarkan soal nomor 4 silahkan tampilkan data setelah 7 baris pertama dan tampilkan hanya 9 data saja

Contoh hasil :



|  |
| --- |
| SELECT  p.product\_id AS id\_produk, p.product\_name AS nama\_produk, c.category\_name AS kategori, p.list\_price AS harga  FROM production.products AS p  JOIN production.categories AS c ON p.category\_id = c.category\_id  WHERE p.list\_price >= 1000 ORDER BY product\_id ASC  OFFSET 7 ROWS FETCH NEXT 9 ROWS ONLY; |
|  |