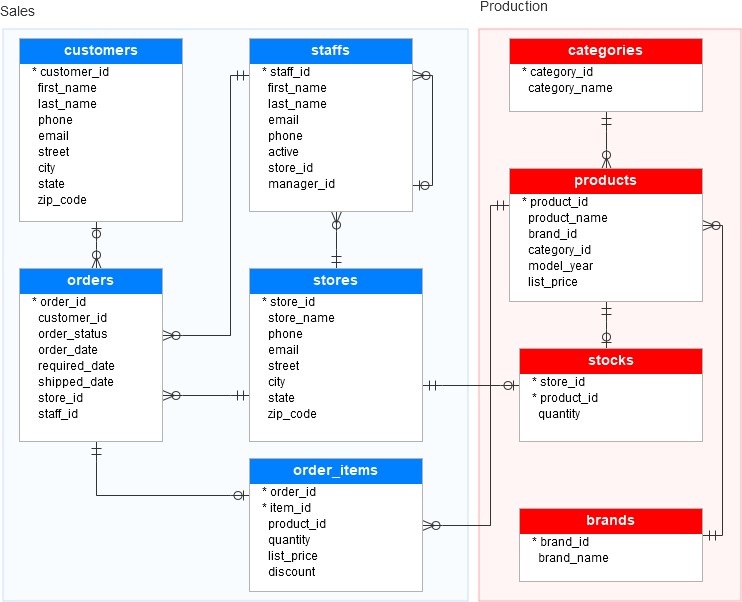
SAFRIZAL RAHMAN 19 SIB 2 G  
LINK GDRIVE  
<https://drive.google.com/drive/folders/1LKec1dypfwYexlBAIO4yUEiJZt8dkjw6?usp=sharing>

**QUIZ 1 DATABASE CARRY ON**

**(Case study Shop Bicycle)**

Shop Fast is shop Which sell all type bicycle. Picture under This is design drawing database diagram Shop Fast



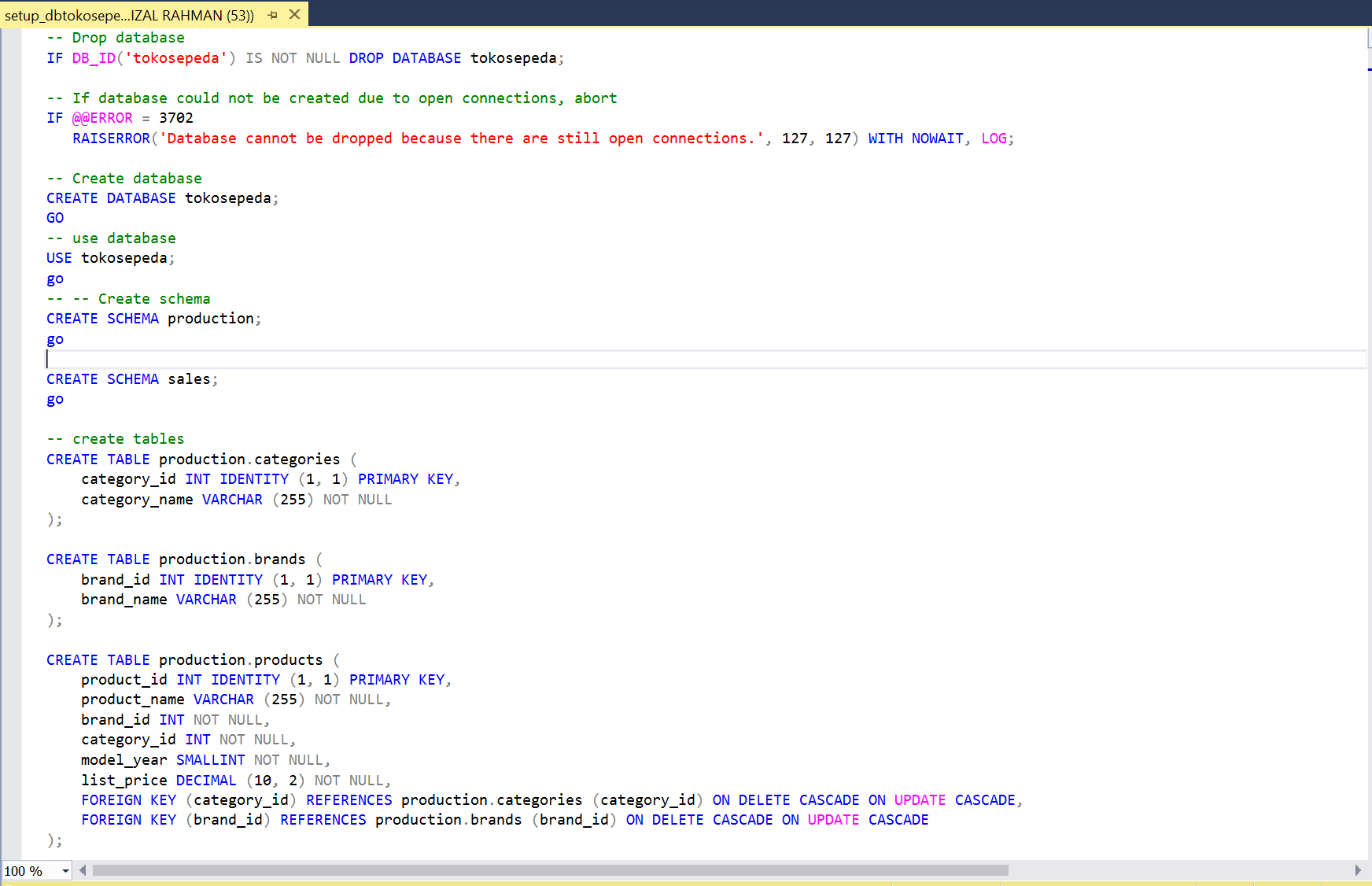
Laman://[www.polinema.ac.id E](http://www.polinema.ac.id/)mail:cs@polinema.ac.id

**DETAIL DATABASE**

# Table sales.stores

**The sales.stores table** contains store information. Each store has a store name, store information, contact like telephone And e-mail, And address including road, city, country part, And code post. The following details source code to make sales.stores table



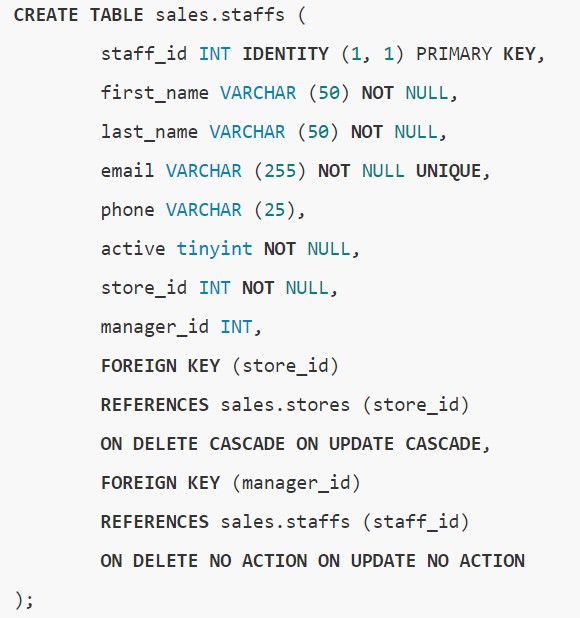


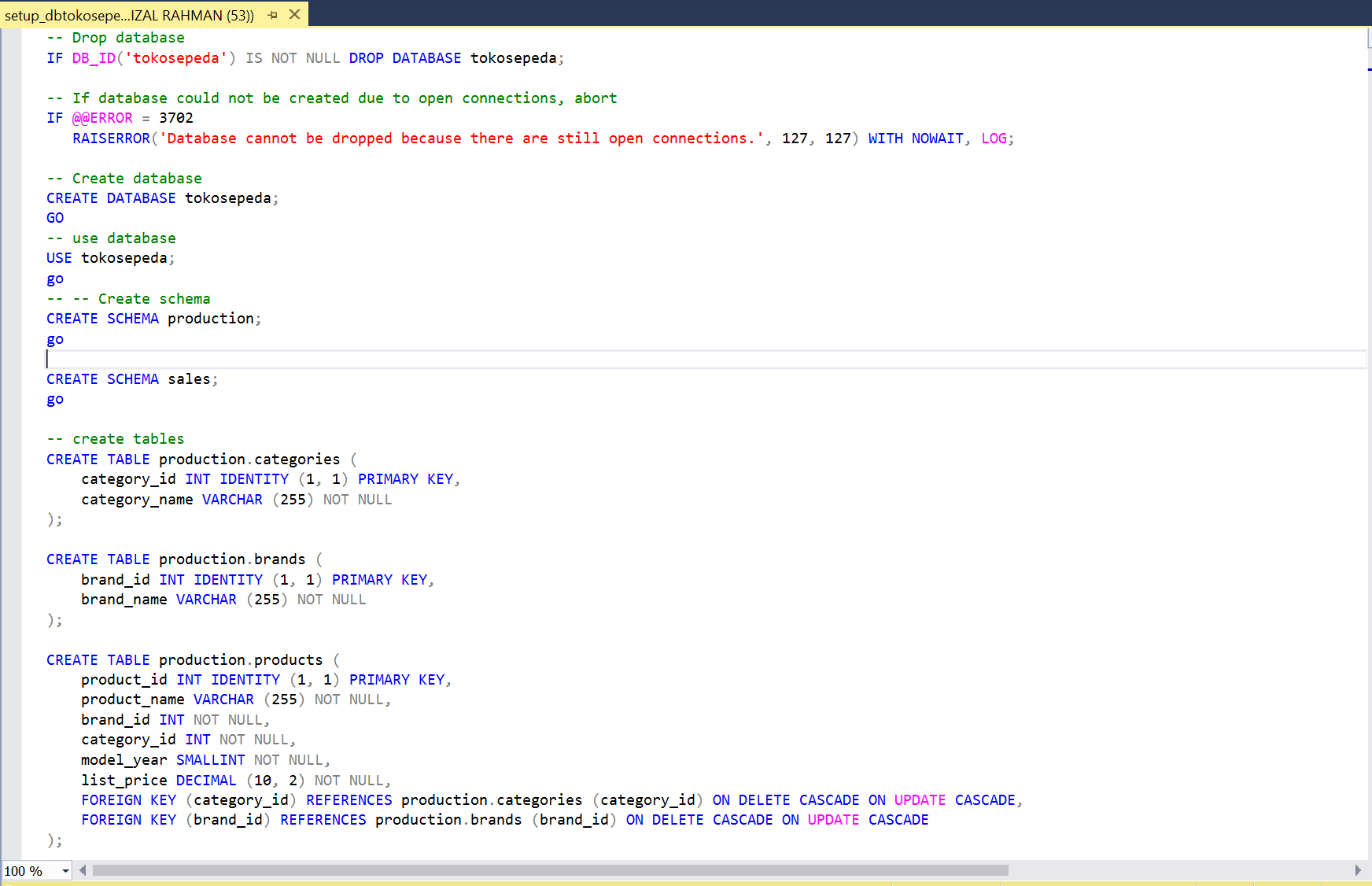
|  |  |
| --- | --- |
| **Table** | **sales.staffs** |
| **sales.staffs table** | |

store important information of staff including first name, Name

back. This Also containing information communication like e-mail And telephone. A staff Work in shop Which determined by mark in column store\_id . A shop can have one or more staff. A staff reports to the store manager who determined by mark in column manager\_id . If mark in manager\_id is null, so staff is play a role as top manager. If staff No Again Work For storage wherever, mark in column active in set to mark zero/zero.

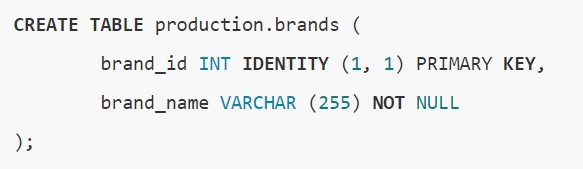
Following This details source code For make table sales.staffs





|  |
| --- |
| **Table production.brands** |
| **Table production.brands** |

keep data about information brand bicycle for example Electra, Hello, and Heller. Following This details source code For make table production.brands

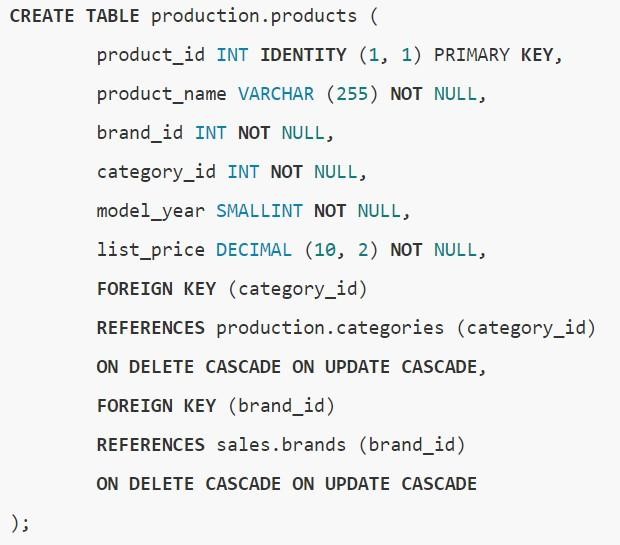




|  |
| --- |
| **Table production.products** |
| **production.products table** |

store product information such as name, brand,

category, model year, and price list. Each product belongs to a specified brand. by the brand\_id column . Therefore, a brand may have zero or many products. Each product also falls into a category defined by the column category\_id . Additionally, each category may have zero or many products. Following This details source code for make table production.products

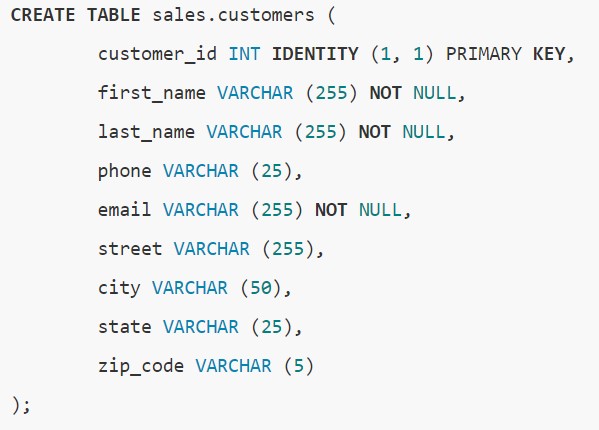




Laman://[www.polinema.ac.id E](http://www.polinema.ac.id/)mail:cs@polinema.ac.id

# Table sales.customers

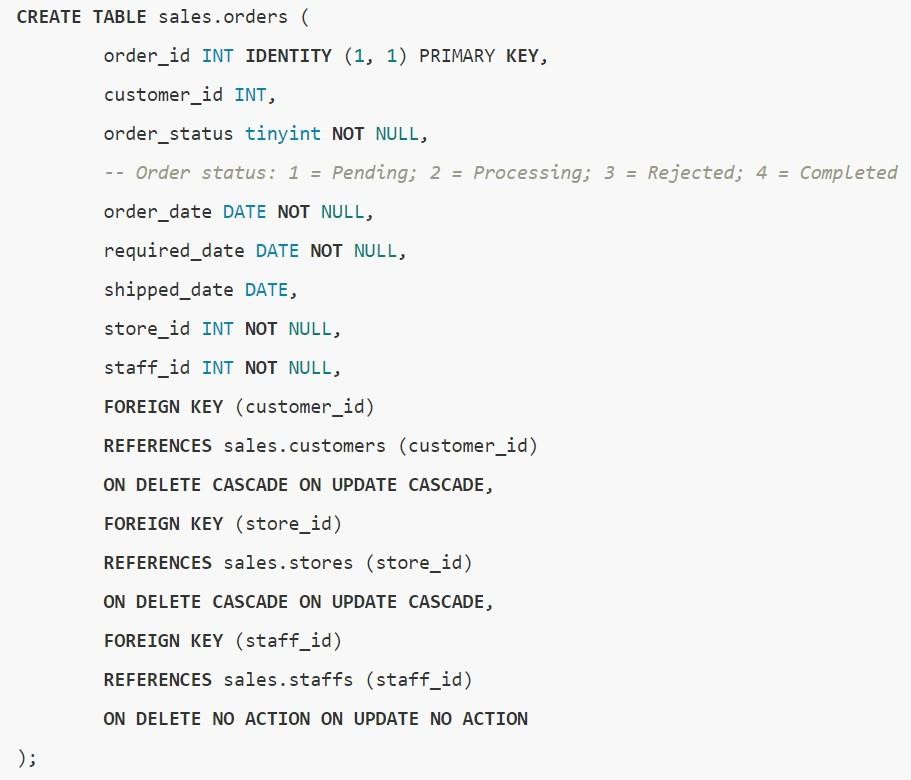
**Table sales.customers** keep information table customer including Name first, last name, phone, email, street, city, country and zip code. Here are details source code to make table sales.customers





**Table sales.orders**

Table sales.orders keep information Header order sale This including customer, order status, order date, required date, shipped date. This is also stores information on where the sales transaction was made (store) and who made it. make it (staff). Every order sale own line in table sales\_orders. Order sale own One or Lots Items line Which saved in table sales.order\_items . Following This details source code For make table sales.orders

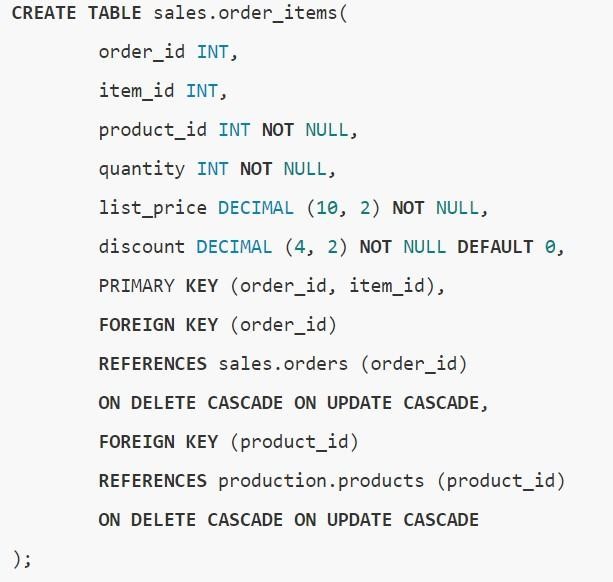




|  |  |
| --- | --- |
| **Table sales.orders\_items** | |
| Table sales.order\_items |  |

keep Items line from order sale. Every Items line belongs to

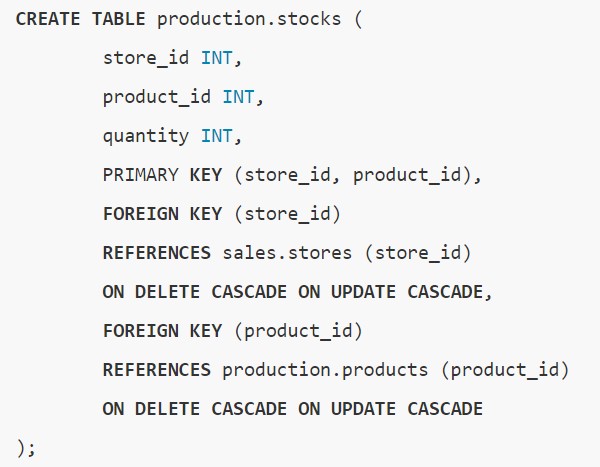
the sales order specified by the order\_id column . The order line item sales include products, order quantities, selling prices, and discounts. The following details source code For create table sales.orders\_items



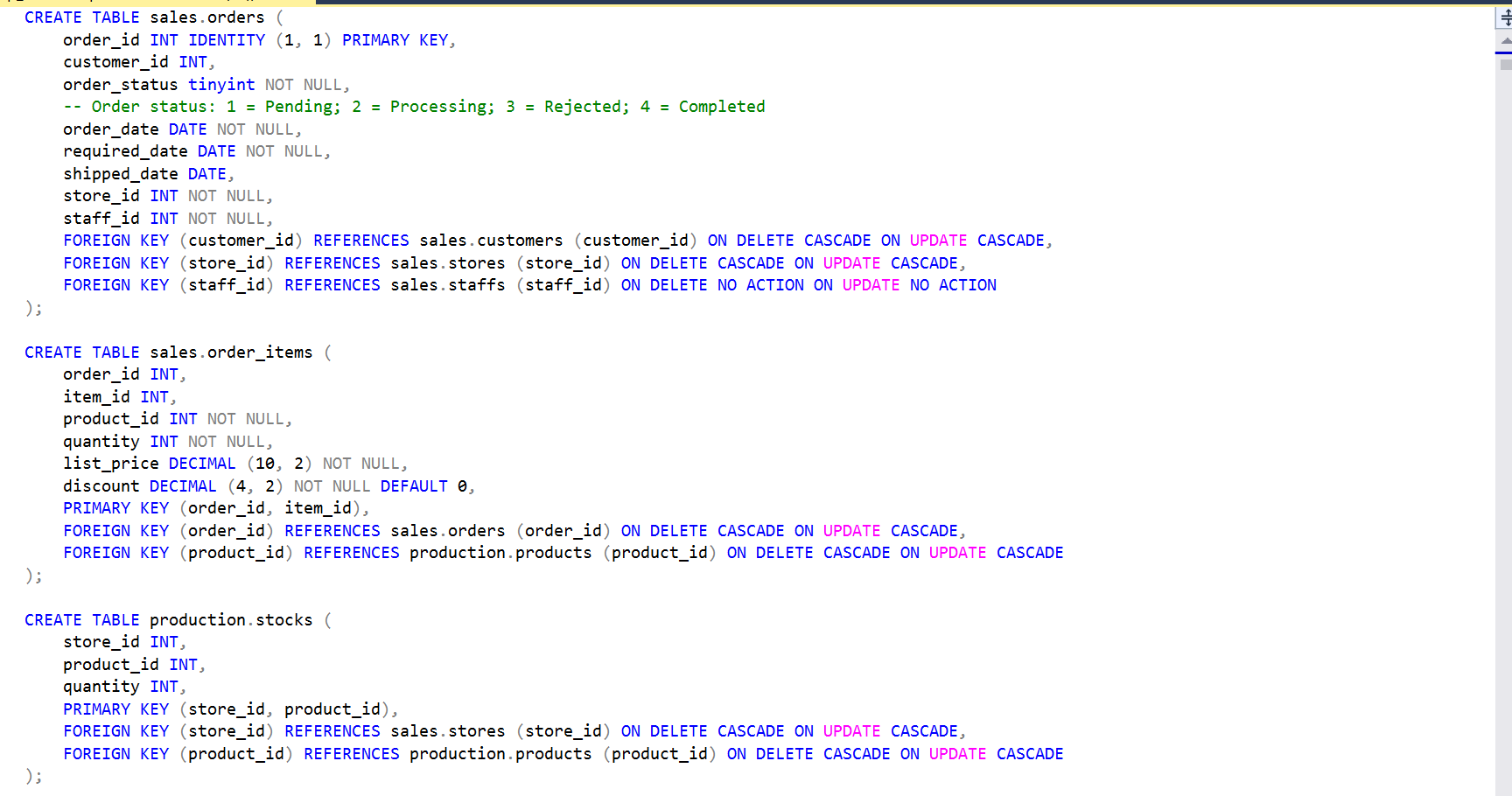
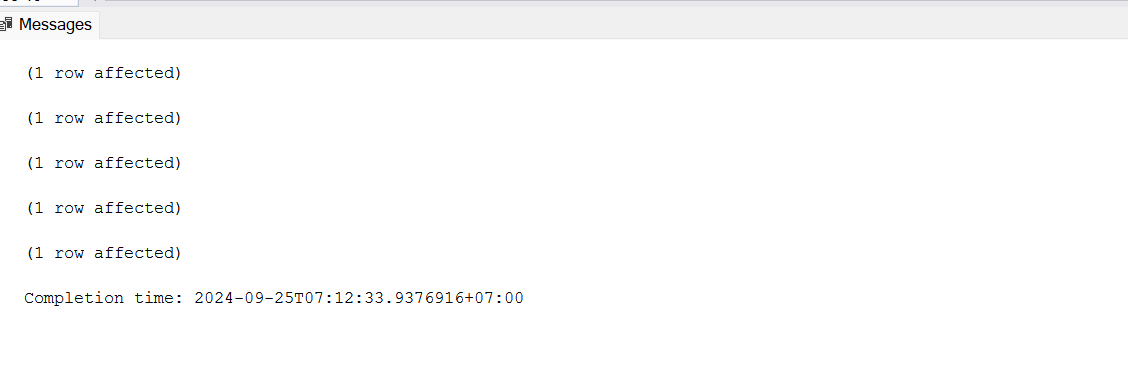


**Table production.stocks**

production.stocks stores inventory information, namely the quantity of a particular product in stock. shop certain. Following this is the detail source code for make table production.stocks

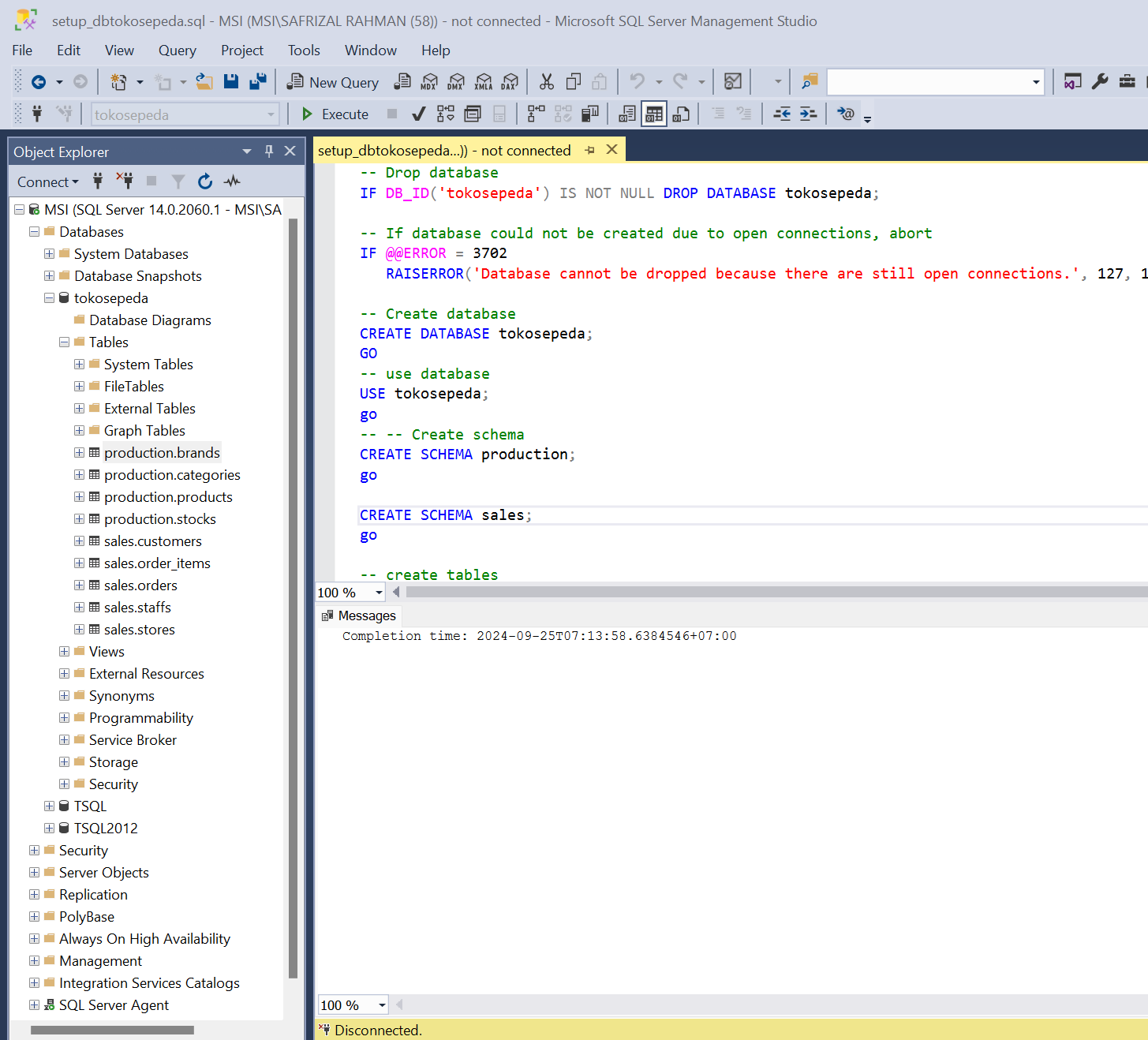


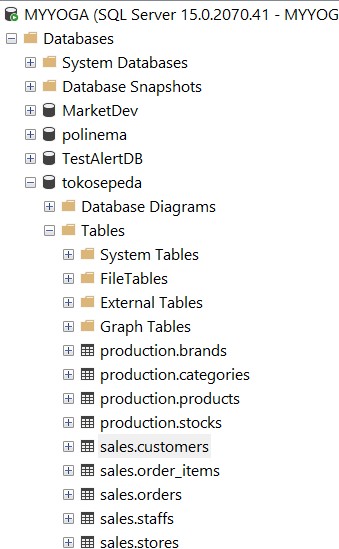


# Task

1. Please run the query source code on file number two to your SQL Server. This source code functioning For make database shop bicycle with details Which Already explained in the previous explanation.





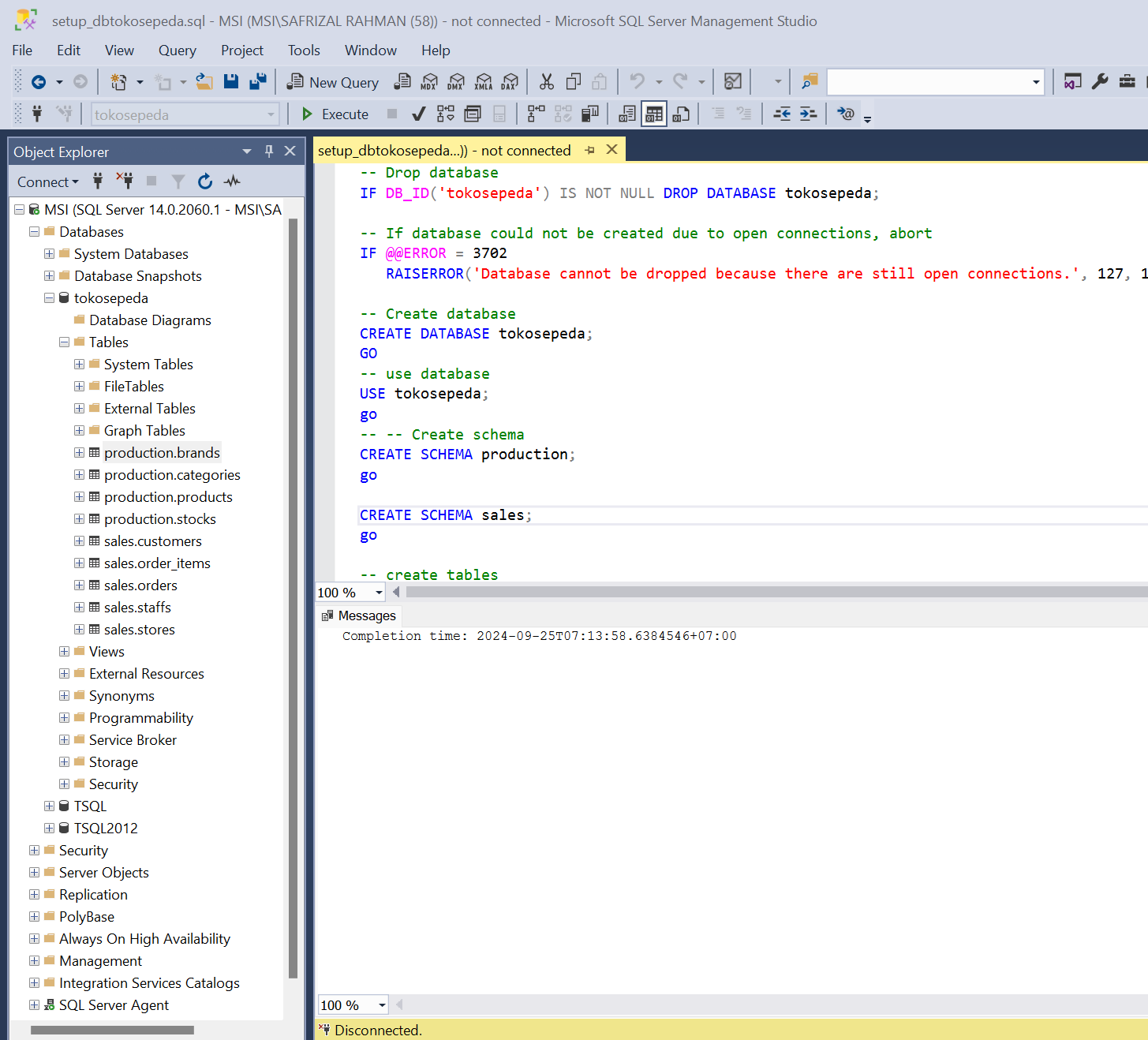
*Capture*

*hasil*

*running source setup*

*yang*

*terbentuk contoh*



1. Please insert 1 new data in **the sales.customers table** where the details are Name, number telephone, And e-mail equated with data self You. Capture source code And results insert You

|  |
| --- |
| ***Copy/paste source TSQL YINSERT INTO sales.customers (first\_name, last\_name , phone, email)***  ***VALUES ('SAFRIZAL', 'RAHMAN','082213286139', 'safrrizalrahman46@gmail.com');***    *ou here* |
| *Capture data results insert You here* |

1. Show all over column from **table sales.customers** with provision letter beginning in Name front they The same with Name front You Then sort based on the largest customer\_id first and display only 10 top data. Capture source code and results SELECT You

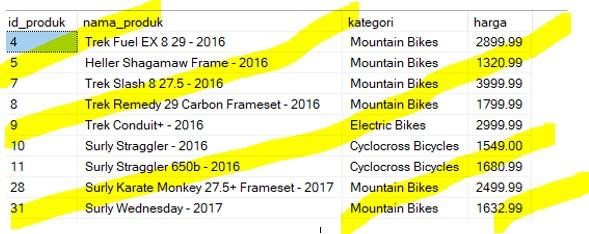
Example results :



|  |
| --- |
| *Copy/paste source TSQL You here*  SELECT TOP 10 \*  FROM sales.customers  WHERE first\_name LIKE 'S%'  ORDER BY customer\_id DESC; |
| *Capture results running TSQL You* |

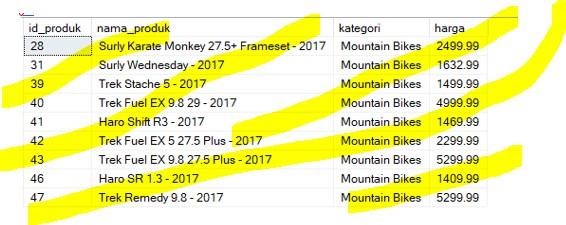
1. Write it down order SELECT between 2 table. Show column product\_id, product\_name, list\_price from the production.products table and category name column from table production.categories. Change it appearance Name the column so that column product\_id become id\_product, product\_name become product name, category\_name becomes category, list\_price becomes price. The data displayed is a product that has a price greater than or equal to 1000 and sort by smallest product\_id. Please do it using order **JOIN**.

Example results :

…etc

|  |
| --- |
| *Copy/paste source TSQL You here*  SELECT p.product\_id AS id\_produk, p.product\_name AS nama\_produk,  p.list\_price AS harga, c.category\_name AS kategori  FROM production.products p  JOIN production.categories c ON p.category\_id = c.category\_id  WHERE p.list\_price >= 1000  ORDER BY p.product\_id ASC; |
| *Capture results running TSQL You* |

1. Based on question number 4 Please show data after 7 line First And show only 9 data only Example results :



|  |
| --- |
| *Copy/paste source TSQL You here*  SELECT p.product\_id AS id\_produk, p.product\_name AS nama\_produk,  p.list\_price AS harga, c.category\_name AS kategori  FROM production.products p  JOIN production.categories c ON p.category\_id = c.category\_id  WHERE p.list\_price >= 1000  ORDER BY p.product\_id ASC  OFFSET 7 ROWS FETCH NEXT 9 ROWS ONLY; |
| *Capture results running TSQL You* |