

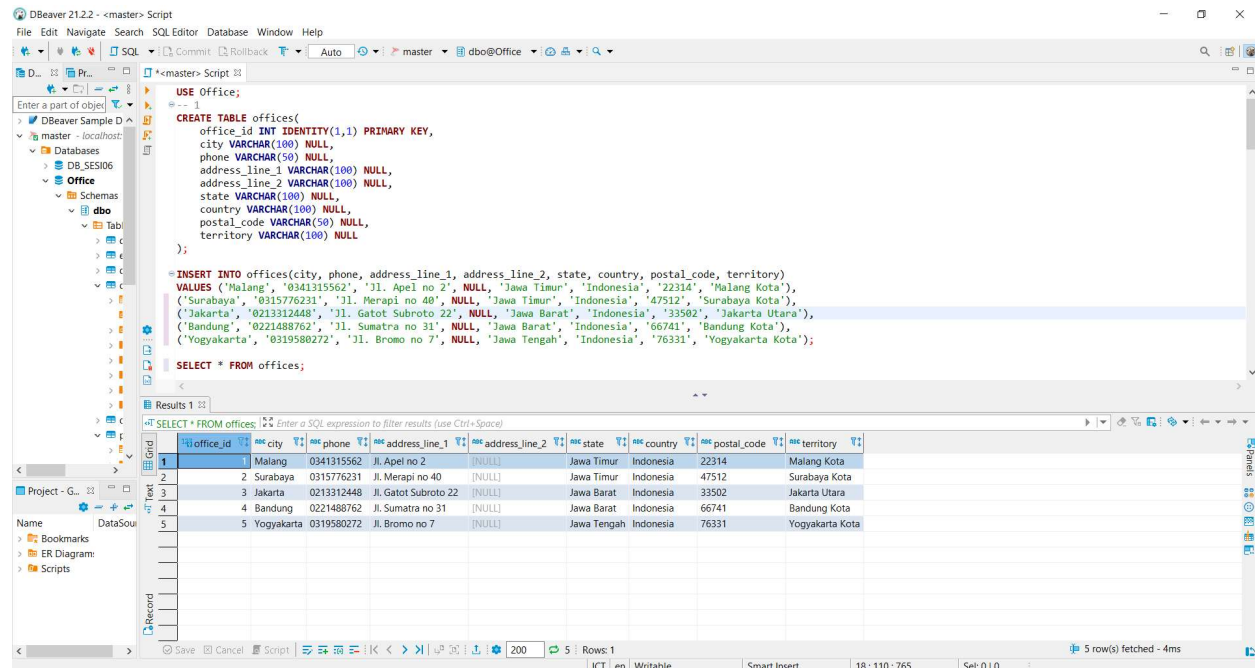
Nama Peserta: Indra Gunawan Budiharjo

Kode Peserta : FSDO001ONL027

Tugas : Assignment 2

Screenshot

1. Table offices



Query:

```
CREATE TABLE offices(
    office_id INT IDENTITY(1,1) PRIMARY KEY,
    city VARCHAR(100) NULL,
    phone VARCHAR(50) NULL,
    address_line_1 VARCHAR(100) NULL,
    address_line_2 VARCHAR(100) NULL,
    state VARCHAR(100) NULL,
    country VARCHAR(100) NULL,
    postal_code VARCHAR(50) NULL,
    territory VARCHAR(100) NULL
);
```

```
INSERT INTO offices(city, phone, address_line_1, address_line_2, state, country,
postal_code, territory)
VALUES ('Malang', '0341315562', 'Jl. Apel no 2', NULL, 'Jawa Timur', 'Indonesia',
'22314', 'Malang Kota'),
('Surabaya', '0315776231', 'Jl. Merapi no 40', NULL, 'Jawa Timur', 'Indonesia',
'47512', 'Surabaya Kota'),
('Jakarta', '0213312448', 'Jl. Gatot Subroto 22', NULL, 'Jawa Barat', 'Indonesia',
'33502', 'Jakarta Utara'),
```

```
('Bandung', '0221488762', 'Jl. Sumatra no 31', NULL, 'Jawa Barat', 'Indonesia',
'66741', 'Bandung Kota'),
('Yogyakarta', '0319580272', 'Jl. Bromo no 7', NULL, 'Jawa Tengah', 'Indonesia',
'76331', 'Yogyakarta Kota');
```

2. Table employees

The screenshot shows the DBeaver SQL Editor with the following SQL script:

```
CREATE TABLE employees(
    employee_id INT IDENTITY(1,1) PRIMARY KEY,
    first_name VARCHAR(100) NULL,
    last_name VARCHAR(100) NULL,
    extension VARCHAR(100) NULL,
    email VARCHAR(100) NULL,
    office_id INT NULL,
    reports_to INT NULL,
    job_title VARCHAR(100) NULL,
    CONSTRAINT FK_employees_office FOREIGN KEY (office_id)
    REFERENCES offices (office_id),
    CONSTRAINT FK_employees_report FOREIGN KEY (reports_to)
    REFERENCES employees (employee_id)
);

INSERT INTO employees(first_name, last_name, extension, email, office_id, reports_to, job_title)
VALUES ('Hadi', 'Kusuma', '1 year', 'hadi@xyz.com', 1, 2, 'Sales'),
('Sandi', 'Mukti', '5 year', 'sandi@xyz.com', 1, NULL, 'Manager'),
('Rahmat', 'Wijaya', '2 year', 'rahmat@xyz.com', 2, 4, 'Officer'),
('Fikri', NULL, '5 year', 'fikri@xyz.com', 2, NULL, 'Executive'),
('Andri', 'Suryajaya', '2 year', 'andri@xyz.com', 3, 6, 'Sales'),
('Bryan', 'Jonathan', '5 year', 'bryan@xyz.com', 3, NULL, 'Manager'),
('Septi', 'Muliawati', '1 year', 'septi@xyz.com', 4, 7, 'Sales'),
('Silvi', 'Ajeng', '2 year', 'silvi@xyz.com', 5, NULL, 'Executive');
```

The Results grid shows the following data:

employee_id	first_name	last_name	extension	email	office_id	reports_to	job_title
1	Hadi	Kusuma	1 year	hadi@xyz.com	1	2	Sales
2	Sandi	Mukti	5 year	sandi@xyz.com	1	[NULL]	Manager
3	Rahmat	Wijaya	2 year	rahmat@xyz.com	2	4	Officer
4	Fikri	[NULL]	5 year	fikri@xyz.com	2	[NULL]	Executive
5	Andri	Suryajaya	2 year	andri@xyz.com	3	6	Sales
6	Bryan	Jonathan	5 year	bryan@xyz.com	3	[NULL]	Manager
7	Septi	Muliawati	1 year	septi@xyz.com	4	7	Sales
8	Silvi	Ajeng	2 year	silvi@xyz.com	5	[NULL]	Executive

Query:

```
CREATE TABLE employees(
    employee_id INT IDENTITY(1,1) PRIMARY KEY,
    first_name VARCHAR(100) NULL,
    last_name VARCHAR(100) NULL,
    extension VARCHAR(100) NULL,
    email VARCHAR(100) NULL,
    office_id INT NULL,
    reports_to INT NULL,
    job_title VARCHAR(100) NULL,
    CONSTRAINT FK_employees_office FOREIGN KEY (office_id)
    REFERENCES offices (office_id),
    CONSTRAINT FK_employees_report FOREIGN KEY (reports_to)
    REFERENCES employees (employee_id)
);
```

```
INSERT INTO employees(first_name, last_name, extension, email, office_id, reports_to,
job_title)
VALUES ('Hadi', 'Kusuma', '1 year', 'hadi@xyz.com', 1, 2, 'Sales'),
('Sandi', 'Mukti', '5 year', 'sandi@xyz.com', 1, NULL, 'Manager'),
('Rahmat', 'Wijaya', '2 year', 'rahmat@xyz.com', 2, 4, 'Officer'),
('Fikri', NULL, '5 year', 'fikri@xyz.com', 2, NULL, 'Executive'),
('Andri', 'Suryajaya', '2 year', 'andri@xyz.com', 3, 6, 'Sales'),
('Bryan', 'Jonathan', '5 year', 'bryan@xyz.com', 3, NULL, 'Manager'),
('Septi', 'Muliawati', '1 year', 'septi@xyz.com', 4, 7, 'Sales'),
('Silvi', 'Ajeng', '2 year', 'silvi@xyz.com', 5, NULL, 'Executive');
```

The screenshot shows the DBeaver 21.2.2 interface. The top menu bar includes File, Edit, Navigate, Search, SQL Editor, Database, and Help. The left sidebar displays the Database Navigator with a tree view showing the project 'master-Script', databases, and a schema 'dbo' containing a table 'customers'. The main SQL Editor window shows a SQL script that creates a 'customers' table and inserts data. The script is as follows:

```

CREATE TABLE customers(
customer_id INT IDENTITY(1,1) PRIMARY KEY,
customer_name VARCHAR(100) NULL,
contact_last_name VARCHAR(100) NULL,
contact_first_name VARCHAR(100) NULL,
phone VARCHAR(50) NULL,
address_line_1 VARCHAR(100) NULL,
address_line_2 VARCHAR(100) NULL,
city VARCHAR(100) NULL,
state VARCHAR(100) NULL,
postal_code VARCHAR(50) NULL,
country VARCHAR(100) NULL,
sales_rep_employee_number INT NULL,
credit_limit DECIMAL(18,2) NULL,
CONSTRAINT FK_customers_employee FOREIGN KEY (sales_rep_employee_number)
REFERENCES employees (employee_id)
);

INSERT INTO customers(customer_name, contact_last_name, contact_first_name, phone, address_line_1, address_line_2, city, state, postal_code, country, sales_rep_employee_number, credit_limit)
VALUES ('Stevi', 'Halim', 'Stevi', '089522113344', 'Jl. Muni no 2', 'NULL', 'Malang', 'Jawa Timur', '65543', 'Indonesia', 1, 200000000),
('Jordy', 'Jordy', 'Jordy', '089233177622', 'Jl. Mawar no 5', 'NULL', 'Malang', 'Jawa Timur', '65541', 'Indonesia', 1, 200000000),
('Steve', 'Pratama', 'Steve', '08124417680', 'Jl. Manggis no 10', 'NULL', 'Malang', 'Jawa Timur', '63527', 'Indonesia', 1, 150000000),
('Putri', 'Elena', 'Putri', '081325513301', 'Jl. Dieng no 25', 'NULL', 'Jakarta', 'DKI Jakarta', '65543', 'Indonesia', 5, 200000000),
('Hansen', 'Salim', 'Hansen', '089577922145', 'Jl. Anggur no 1', 'NULL', 'Jakarta', 'DKI Jakarta', '65411', 'Indonesia', 5, 150000000),

```

Below the script, the 'Results' tab shows the data inserted into the 'customers' table. The results are displayed in a table with columns: customer_id, customer_name, contact_last_name, contact_first_name, phone, address_line_1, address_line_2, city, state, postal_code, country, sales_rep_employee_number, and credit_limit. The data is as follows:

customer_id	customer_name	contact_last_name	contact_first_name	phone	address_line_1	address_line_2	city	state	postal_code	country	sales_rep_employee_number	credit_limit
1	Stevi	Halim	Jordy	089522113344	Jl. Muni no 2	NULL	Malang	Jawa Timur	65543	Indonesia	1	200000000
2	Jordy	Jordy	Jordy	089233177622	Jl. Mawar no 5	NULL	Malang	Jawa Timur	65541	Indonesia	1	200000000
3	Steve	Pratama	Steve	08124417680	Jl. Manggis no 10	NULL	Malang	Jawa Timur	63527	Indonesia	1	150000000
4	Putri	Elena	Putri	081325513301	Jl. Dieng no 25	NULL	Jakarta	DKI Jakarta	65543	Indonesia	5	200000000
5	Hansen	Salim	Hansen	089577922145	Jl. Anggur no 1	NULL	Jakarta	DKI Jakarta	65411	Indonesia	5	150000000
6	Devin	Devin	Devin	082133661980	Jl. Semeru no 21	NULL	Bandung	Jawa Barat	41543	Indonesia	1	200000000
7	Ratna	Sari	Ratna	082331382111	Jl. Agung no 12	NULL	Bandung	Jawa Barat	41123	Indonesia	1	200000000

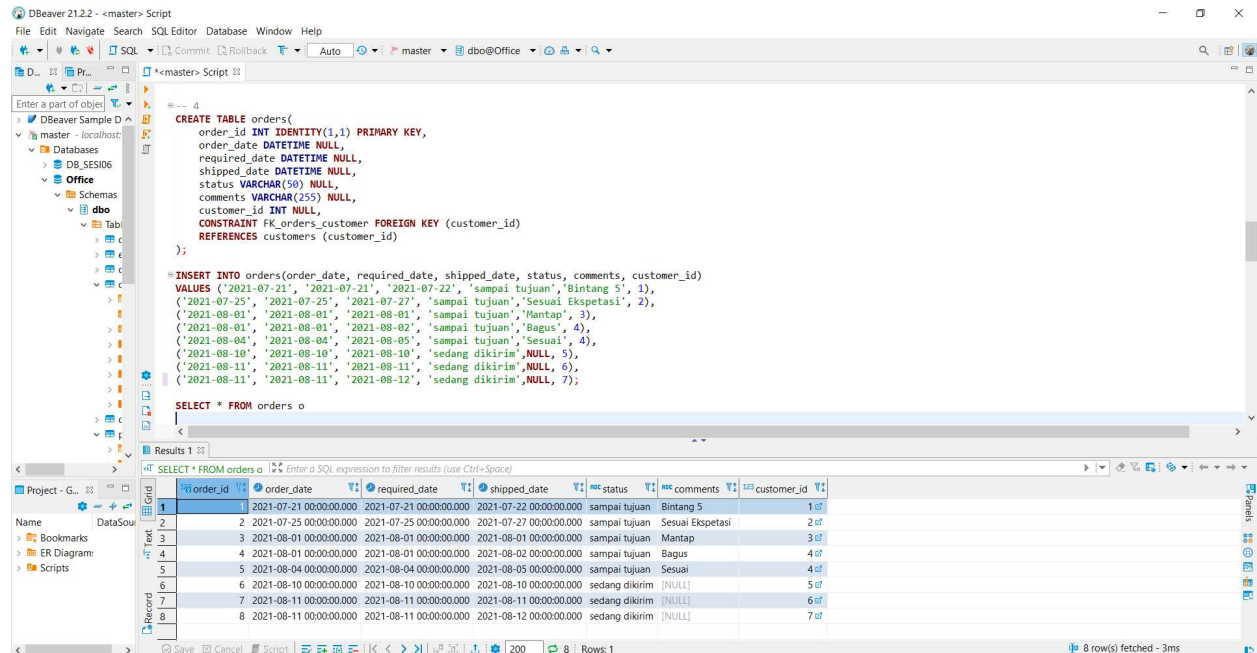
The bottom status bar indicates '7 row(s) fetched - 3ms (-1ms)'.

```
CREATE TABLE customers(  
    customer_id INT IDENTITY(1,1) PRIMARY KEY,  
    customer_name VARCHAR(100) NULL,  
    contact_last_name VARCHAR(100) NULL,  
    contact_first_name VARCHAR(100) NULL,  
    phone VARCHAR(50) NULL,  
    address_line_1 VARCHAR(100) NULL,  
    address_line_2 VARCHAR(100) NULL,  
    city VARCHAR(100) NULL,  
    state VARCHAR(100) NULL,  
    postal_code VARCHAR(50) NULL,  
    country VARCHAR(100) NULL,  
    sales_rep_employee_number INT NULL,  
    credit_limit DECIMAL(18,2) NULL,  
    CONSTRAINT FK_customers_employee FOREIGN KEY (sales_rep_employee_number)  
    REFERENCES employees (employee_id)  
);
```

```
INSERT INTO customers(customer_name, contact_last_name, contact_first_name, phone,
address_line_1, address_line_2, city, state, postal_code, country,
sales_rep_employee_number, credit_limit)
VALUES ('Stevi', 'Halim', 'Stevi', '089522113344', 'Jl. Wuni no 2', NULL, 'Malang',
'Jawa Timur', '65543', 'Indonesia', 1, 20000000),
('Jordy', 'Jordy', 'Jordy', '089233177622', 'Jl. Mawar no 5', NULL, 'Malang', 'Jawa
Timur', '65541', 'Indonesia', 1, 20000000),
('Steve', 'Pratama', 'Steve', '08124417680', 'Jl. Manggis no 10', NULL, 'Malang',
'Jawa Timur', '63527', 'Indonesia', 1, 15000000),
```

```
( 'Putri', 'Elena', 'Putri', '081325513301', 'Jl. Dieng no 25', NULL , 'Jakarta', 'DKI Jakarta', '65543', 'Indonesia', 5, 20000000),
( 'Hansen', 'Salim', 'Hansen', '089577922145', 'Jl. Anggur no 1', NULL , 'Jakarta', 'DKI Jakarta', '65411', 'Indonesia', 5, 15000000),
( 'Devin', 'Devin', 'Devin', '082133661980', 'Jl. Semeru no 21', NULL , 'Bandung', 'Jawa Barat', '41543', 'Indonesia', 7, 20000000),
( 'Ratna', 'Sari', 'Ratna', '082331382111', 'Jl. Agung no 12', NULL , 'Bandung', 'Jawa Barat', '41123', 'Indonesia', 7, 20000000);
```

4. Table orders



```
CREATE TABLE orders(
  order_id INT IDENTITY(1,1) PRIMARY KEY,
  order_date DATETIME NULL,
  required_date DATETIME NULL,
  shipped_date DATETIME NULL,
  status VARCHAR(50) NULL,
  comments VARCHAR(255) NULL,
  customer_id INT NULL,
  CONSTRAINT FK_orders_customer FOREIGN KEY (customer_id)
  REFERENCES customers (customer_id)
);

INSERT INTO orders(order_date, required_date, shipped_date, status, comments, customer_id)
VALUES ('2021-07-21', '2021-07-21', '2021-07-22', 'sampai tujuan', 'Bintang 5', 1),
('2021-07-25', '2021-07-25', '2021-07-27', 'sampai tujuan', 'Sesuai Ekspetasi', 2),
('2021-08-01', '2021-08-01', '2021-08-01', 'sampai tujuan', 'Mantap', 3),
('2021-08-01', '2021-08-01', '2021-08-02', 'sampai tujuan', 'Bagus', 4),
('2021-08-04', '2021-08-04', '2021-08-05', 'sampai tujuan', 'Sesuai', 4),
('2021-08-10', '2021-08-10', '2021-08-10', 'sedang dikirim', NULL, 5),
('2021-08-11', '2021-08-11', '2021-08-11', 'sedang dikirim', NULL, 6),
('2021-08-11', '2021-08-11', '2021-08-12', 'sedang dikirim', NULL, 7);

SELECT * FROM orders o
```

order_id	order_date	required_date	shipped_date	status	comments	customer_id
1	2021-07-21 00:00:00.000	2021-07-21 00:00:00.000	2021-07-22 00:00:00.000	sampai tujuan	Bintang 5	1
2	2021-07-25 00:00:00.000	2021-07-25 00:00:00.000	2021-07-27 00:00:00.000	sampai tujuan	Sesuai Ekspetasi	2
3	2021-08-01 00:00:00.000	2021-08-01 00:00:00.000	2021-08-01 00:00:00.000	sampai tujuan	Mantap	3
4	2021-08-01 00:00:00.000	2021-08-01 00:00:00.000	2021-08-02 00:00:00.000	sampai tujuan	Bagus	4
5	2021-08-04 00:00:00.000	2021-08-04 00:00:00.000	2021-08-05 00:00:00.000	sampai tujuan	Sesuai	4
6	2021-08-10 00:00:00.000	2021-08-10 00:00:00.000	2021-08-10 00:00:00.000	sedang dikirim	[NULL]	5
7	2021-08-11 00:00:00.000	2021-08-11 00:00:00.000	2021-08-11 00:00:00.000	sedang dikirim	[NULL]	6
8	2021-08-11 00:00:00.000	2021-08-11 00:00:00.000	2021-08-12 00:00:00.000	sedang dikirim	[NULL]	7

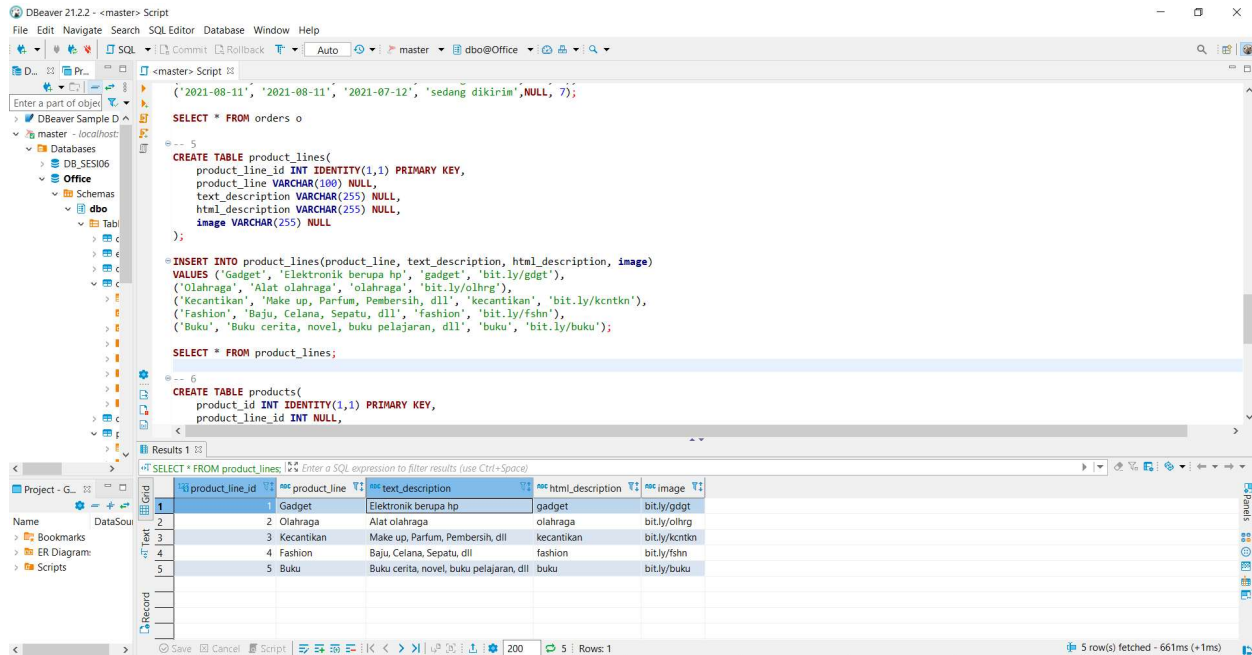
Query:

```
CREATE TABLE orders(
  order_id INT IDENTITY(1,1) PRIMARY KEY,
  order_date DATETIME NULL,
  required_date DATETIME NULL,
  shipped_date DATETIME NULL,
  status VARCHAR(50) NULL,
  comments VARCHAR(255) NULL,
  customer_id INT NULL,
  CONSTRAINT FK_orders_customer FOREIGN KEY (customer_id)
  REFERENCES customers (customer_id)
);
```

```
INSERT INTO orders(order_date, required_date, shipped_date, status, comments,
customer_id)
VALUES ('2021-07-21', '2021-07-21', '2021-07-22', 'sampai tujuan', 'Bintang 5', 1),
('2021-07-25', '2021-07-25', '2021-07-27', 'sampai tujuan', 'Sesuai Ekspetasi', 2),
('2021-08-01', '2021-08-01', '2021-08-01', 'sampai tujuan', 'Mantap', 3),
('2021-08-01', '2021-08-01', '2021-08-02', 'sampai tujuan', 'Bagus', 4),
('2021-08-04', '2021-08-04', '2021-08-05', 'sampai tujuan', 'Sesuai', 4),
('2021-08-10', '2021-08-10', '2021-08-10', 'sedang dikirim', NULL, 5),
```

```
( '2021-08-11', '2021-08-11', '2021-08-11', 'sedang dikirim', NULL, 6),
( '2021-08-11', '2021-08-11', '2021-08-12', 'sedang dikirim', NULL, 7);
```

5. Table product_lines

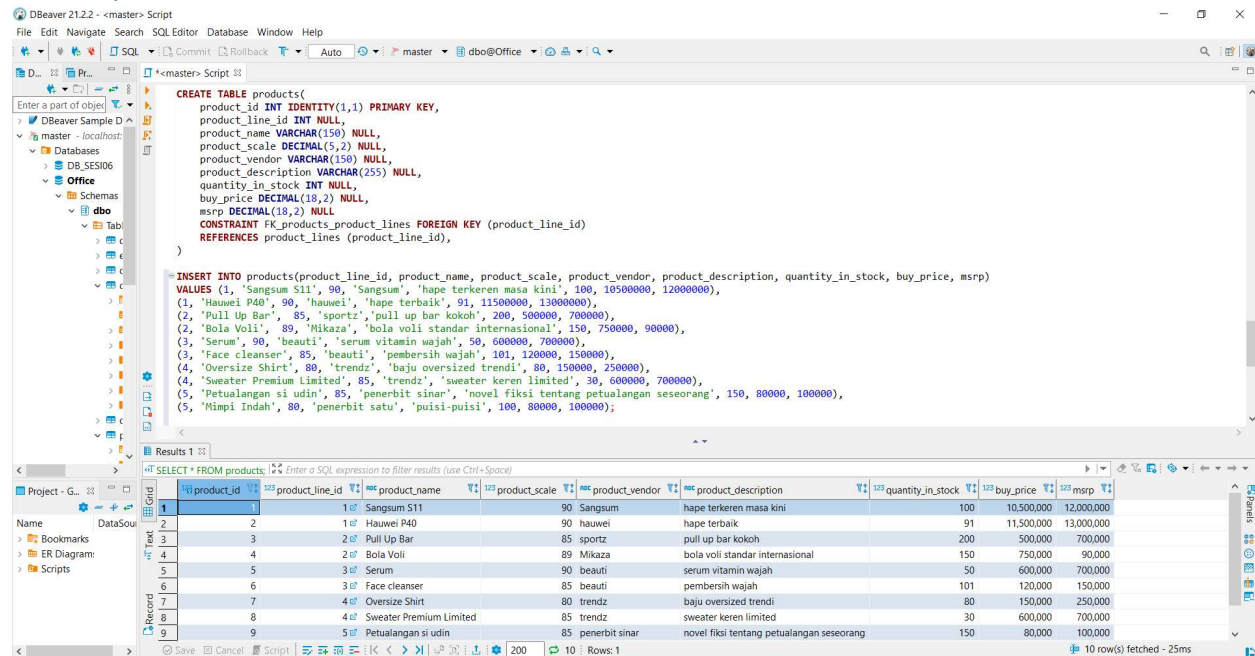


Query:

```
CREATE TABLE product_lines(
  product_line_id INT IDENTITY(1,1) PRIMARY KEY,
  product_line VARCHAR(100) NULL,
  text_description VARCHAR(255) NULL,
  html_description VARCHAR(255) NULL,
  image VARCHAR(255) NULL
);
```

```
INSERT INTO product_lines(product_line, text_description, html_description, image)
VALUES ('Gadget', 'Elektronik berupa hp', 'gadget', 'bit.ly/gdgt'),
('Olahraga', 'Alat olahraga', 'olahraga', 'bit.ly/olhrg'),
('Kecantikan', 'Make up, Parfum, Pembersih, dll', 'kecantikan', 'bit.ly/kcntkn'),
('Fashion', 'Baju, Celana, Sepatu, dll', 'fashion', 'bit.ly/fshn'),
('Buku', 'Buku cerita, novel, buku pelajaran, dll', 'buku', 'bit.ly/buku');
```


6. Table products



The screenshot shows the DBeaver 21.2.2 interface. The SQL editor on the right contains the following SQL script:

```
CREATE TABLE products(  
    product_id INT IDENTITY(1,1) PRIMARY KEY,  
    product_line_id INT NULL,  
    product_name VARCHAR(150) NULL,  
    product_scale DECIMAL(5,2) NULL,  
    product_vendor VARCHAR(150) NULL,  
    product_description VARCHAR(255) NULL,  
    quantity_in_stock INT NULL,  
    buy_price DECIMAL(18,2) NULL,  
    msrp DECIMAL(18,2) NULL  
    CONSTRAINT FK_products_product_lines FOREIGN KEY (product_line_id)  
    REFERENCES product_lines (product_line_id),  
)  
  
INSERT INTO products(product_line_id, product_name, product_scale, product_vendor, product_description, quantity_in_stock, buy_price, msrp)  
VALUES (1, 'Sangsum S11', 90, 'Sangsum', 'hape terkeren masa kini', 100, 10500000, 12000000),  
(1, 'Hauwei P40', 90, 'Hauwei', 'hape terbaik', 91, 11500000, 13000000),  
(2, 'Pull Up Bar', 85, 'sportz', 'pull up bar kokoh', 200, 500000, 700000),  
(2, 'Bola Voli', 89, 'Mikaza', 'bola voli standar internasional', 150, 750000, 900000),  
(3, 'Serum', 90, 'beauti', 'serum vitamin wajah', 50, 600000, 700000),  
(3, 'Face cleanser', 85, 'beauti', 'pembersih wajah', 101, 120000, 150000),  
(4, 'Oversize Shirt', 80, 'trendz', 'baju oversized trendi', 80, 150000, 250000),  
(4, 'Sweater Premium Limited', 85, 'trendz', 'sweater keren limited', 30, 600000, 700000),  
(5, 'Petualangan si udin', 85, 'penerbit sinar', 'novel fiksi tentang petualangan seseorang', 150, 80000, 100000),  
(5, 'Mimpi Indah', 80, 'penerbit satu', 'puisi-puisi', 100, 80000, 100000);
```

The Results 1 tab at the bottom shows a grid of 10 rows of data:

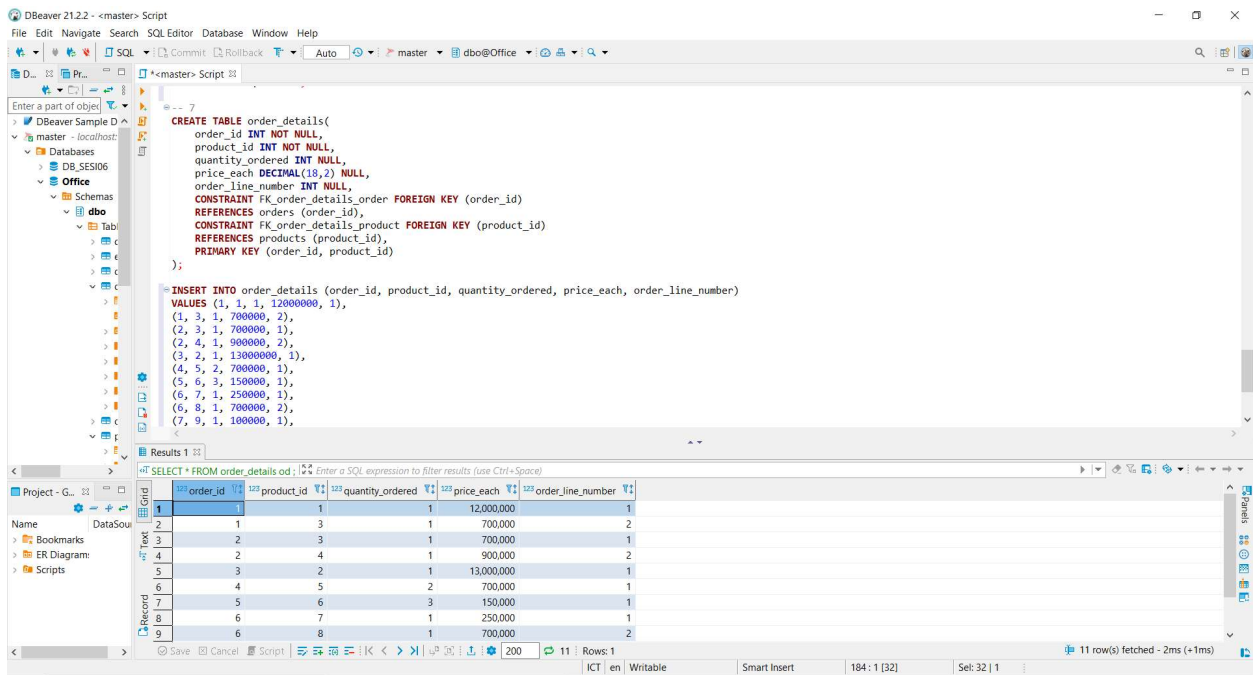
product_line_id	product_name	product_scale	product_vendor	product_description	quantity_in_stock	buy_price	msrp
1	Sangsum S11	90	Sangsum	hape terkeren masa kini	100	10,500,000	12,000,000
1	Hauwei P40	90	Hauwei	hape terbaik	91	11,500,000	13,000,000
2	Pull Up Bar	85	sportz	pull up bar kokoh	200	500,000	700,000
2	Bola Voli	89	Mikaza	bola voli standar internasional	150	750,000	900,000
3	Serum	90	beauti	serum vitamin wajah	50	600,000	700,000
3	Face cleanser	85	beauti	pembersih wajah	101	120,000	150,000
4	Oversize Shirt	80	trendz	baju oversized trendi	80	150,000	250,000
4	Sweater Premium Limited	85	trendz	sweater keren limited	30	600,000	700,000
5	Petualangan si udin	85	penerbit sinar	novel fiksi tentang petualangan seseorang	150	80,000	100,000
5	Mimpi Indah	80	penerbit satu	puisi-puisi	100	80,000	100,000

Query:

```
CREATE TABLE products(  
    product_id INT IDENTITY(1,1) PRIMARY KEY,  
    product_line_id INT NULL,  
    product_name VARCHAR(150) NULL,  
    product_scale DECIMAL(5,2) NULL,  
    product_vendor VARCHAR(150) NULL,  
    product_description VARCHAR(255) NULL,  
    quantity_in_stock INT NULL,  
    buy_price DECIMAL(18,2) NULL,  
    msrp DECIMAL(18,2) NULL  
    CONSTRAINT FK_products_product_lines FOREIGN KEY (product_line_id)  
    REFERENCES product_lines (product_line_id),  
)
```

```
INSERT INTO products(product_line_id, product_name, product_scale, product_vendor,  
product_description, quantity_in_stock, buy_price, msrp)  
VALUES (1, 'Sangsum S11', 90, 'Sangsum', 'hape terkeren masa kini', 100, 10500000,  
12000000),  
(1, 'Hauwei P40', 90, 'Hauwei', 'hape terbaik', 91, 11500000, 13000000),  
(2, 'Pull Up Bar', 85, 'sportz', 'pull up bar kokoh', 200, 500000, 700000),  
(2, 'Bola Voli', 89, 'Mikaza', 'bola voli standar internasional', 150, 750000,  
900000),  
(3, 'Serum', 90, 'beauti', 'serum vitamin wajah', 50, 600000, 700000),  
(3, 'Face cleanser', 85, 'beauti', 'pembersih wajah', 101, 120000, 150000),  
(4, 'Oversize Shirt', 80, 'trendz', 'baju oversized trendi', 80, 150000, 250000),  
(4, 'Sweater Premium Limited', 85, 'trendz', 'sweater keren limited', 30, 600000,  
700000),  
(5, 'Petualangan si udin', 85, 'penerbit sinar', 'novel fiksi tentang petualangan  
seseorang', 150, 80000, 100000),  
(5, 'Mimpi Indah', 80, 'penerbit satu', 'puisi-puisi', 100, 80000, 100000);
```

7. Table order_details



The screenshot shows the DBeaver 21.2.2 interface. The SQL Editor contains the following SQL script:

```
CREATE TABLE order_details(  
    order_id INT NOT NULL,  
    product_id INT NOT NULL,  
    quantity_ordered INT NULL,  
    price_each DECIMAL(18,2) NULL,  
    order_line_number INT NULL,  
    CONSTRAINT FK_order_details_order FOREIGN KEY (order_id)  
        REFERENCES orders (order_id),  
    CONSTRAINT FK_order_details_product FOREIGN KEY (product_id)  
        REFERENCES products (product_id),  
    PRIMARY KEY (order_id, product_id)  
);  
  
INSERT INTO order_details (order_id, product_id, quantity_ordered, price_each, order_line_number)  
VALUES (1, 1, 1, 12000000, 1),  
(1, 3, 1, 700000, 2),  
(2, 3, 1, 700000, 1),  
(2, 4, 1, 900000, 2),  
(3, 2, 1, 13000000, 1),  
(4, 5, 2, 700000, 1),  
(5, 6, 3, 150000, 1),  
(6, 7, 1, 250000, 1),  
(6, 8, 1, 700000, 2),  
(7, 9, 1, 100000, 1),  
(7, 9, 1, 100000, 1);
```

The Results pane shows the following data:

order_id	product_id	quantity_ordered	price_each	order_line_number
1	1	1	12000000	1
1	3	1	700000	2
2	3	1	700000	1
2	4	1	900000	2
3	2	1	13000000	1
4	5	2	700000	1
5	6	3	150000	1
6	7	1	250000	1
6	8	1	700000	2
7	9	1	100000	1
7	9	1	100000	1

Query:

```
CREATE TABLE order_details(  
    order_id INT NOT NULL,  
    product_id INT NOT NULL,  
    quantity_ordered INT NULL,  
    price_each DECIMAL(18,2) NULL,  
    order_line_number INT NULL,  
    CONSTRAINT FK_order_details_order FOREIGN KEY (order_id)  
        REFERENCES orders (order_id),  
    CONSTRAINT FK_order_details_product FOREIGN KEY (product_id)  
        REFERENCES products (product_id),  
    PRIMARY KEY (order_id, product_id)  
);
```

```
INSERT INTO order_details (order_id, product_id, quantity_ordered, price_each,  
order_line_number)  
VALUES (1, 1, 1, 12000000, 1),  
(1, 3, 1, 700000, 2),  
(2, 3, 1, 700000, 1),  
(2, 4, 1, 900000, 2),  
(3, 2, 1, 13000000, 1),  
(4, 5, 2, 700000, 1),  
(5, 6, 3, 150000, 1),  
(6, 7, 1, 250000, 1),  
(6, 8, 1, 700000, 2),  
(7, 9, 1, 100000, 1),  
(8, 10, 1, 100000, 1);
```

8. Table Payment

The screenshot shows the DBeaver 21.2.2 interface. The SQL Editor on the right contains the following SQL script:

```
(7, 9, 1, 100000, 1);  
(8, 10, 1, 100000, 1);  
  
SELECT * FROM order_details od ;  
  
-- 8  
CREATE TABLE payments(  
    payment_id INT IDENTITY(1,1) PRIMARY KEY,  
    customer_id INT NULL,  
    payment_date DATE NULL,  
    amount DECIMAL(18,2) NULL,  
    CONSTRAINT FK_payments_customer FOREIGN KEY (customer_id)  
        REFERENCES customers (customer_id)  
);  
  
INSERT INTO payments (customer_id, payment_date, amount)  
VALUES (1, '2021-07-30', 12700000),  
(2, '2021-07-30', 1600000),  
(3, '2021-08-10', 13000000),  
(4, '2021-08-20', 850000),  
(5, '2021-08-20', 950000),  
(6, '2021-08-30', 100000),  
(7, '2021-08-30', 100000);  
  
SELECT * FROM payments;
```

The Results grid at the bottom displays the data from the 'payments' table:

	payment_id	customer_id	payment_date	amount
1	1	1	2021-07-30	12,700,000
2	2	2	2021-07-30	1,600,000
3	3	3	2021-08-10	13,000,000
4	4	4	2021-08-20	850,000
5	5	5	2021-08-20	950,000
6	6	6	2021-08-30	100,000
7	7	7	2021-08-30	100,000

Query:

```
CREATE TABLE payments(  
    payment_id INT IDENTITY(1,1) PRIMARY KEY,  
    customer_id INT NULL,  
    payment_date DATE NULL,  
    amount DECIMAL(18,2) NULL,  
    CONSTRAINT FK_payments_customer FOREIGN KEY (customer_id)  
        REFERENCES customers (customer_id)  
);
```

```
INSERT INTO payments (customer_id, payment_date, amount)  
VALUES (1, '2021-07-30', 12700000),  
(2, '2021-07-30', 1600000),  
(3, '2021-08-10', 13000000),  
(4, '2021-08-20', 850000),  
(5, '2021-08-20', 950000),  
(6, '2021-08-30', 100000),  
(7, '2021-08-30', 100000);
```


-- Query data

-- 1. tampilkan nama customer, order_id, nama produk, harga tiap produk dari customer_id 4

```
SELECT c.customer_name, o.order_id, p.product_name, od.price_each
FROM customers c
INNER JOIN orders o ON c.customer_id = o.customer_id
INNER JOIN order_details od ON o.order_id = od.order_id
INNER JOIN products p ON od.product_id = p.product_id
WHERE c.customer_id = 4;
```

-- 2. tampilkan nama barang dengan jumlah stok terbanyak
SELECT product_name, quantity_in_stock FROM products
WHERE quantity_in_stock = (SELECT max(quantity_in_stock) FROM products);

-- 3. tampilkan product yang paling banyak yang diorder

Results 1

	customer_name	order_id	product_name	price_each
1	Putri	4	Serum	700,000
2	Putri	5	Face cleanser	150,000

Query:

```
SELECT c.customer_name, o.order_id, p.product_name, od.price_each
FROM customers c
INNER JOIN orders o ON c.customer_id = o.customer_id
INNER JOIN order_details od ON o.order_id = od.order_id
INNER JOIN products p ON od.product_id = p.product_id
WHERE c.customer_id = 4;
```

-- 2. tampilkan nama barang dengan jumlah stok terbanyak

```
INNER JOIN order_details od ON o.order_id = od.order_id
INNER JOIN products p ON od.product_id = p.product_id
WHERE c.customer_id = 4;
```

-- 2. tampilkan nama barang dengan jumlah stok terbanyak
SELECT product_name, quantity_in_stock FROM products
WHERE quantity_in_stock = (SELECT max(quantity_in_stock) FROM products);

-- 3. tampilkan product yang paling banyak yang diorder

```
SELECT a.product_name
FROM (SELECT p.product_name, SUM(od.quantity_ordered) as total_ordered
FROM order_details od
INNER JOIN products p ON od.product_id = p.product_id
GROUP BY p.product_name) a
ORDER BY total_ordered DESC;
```

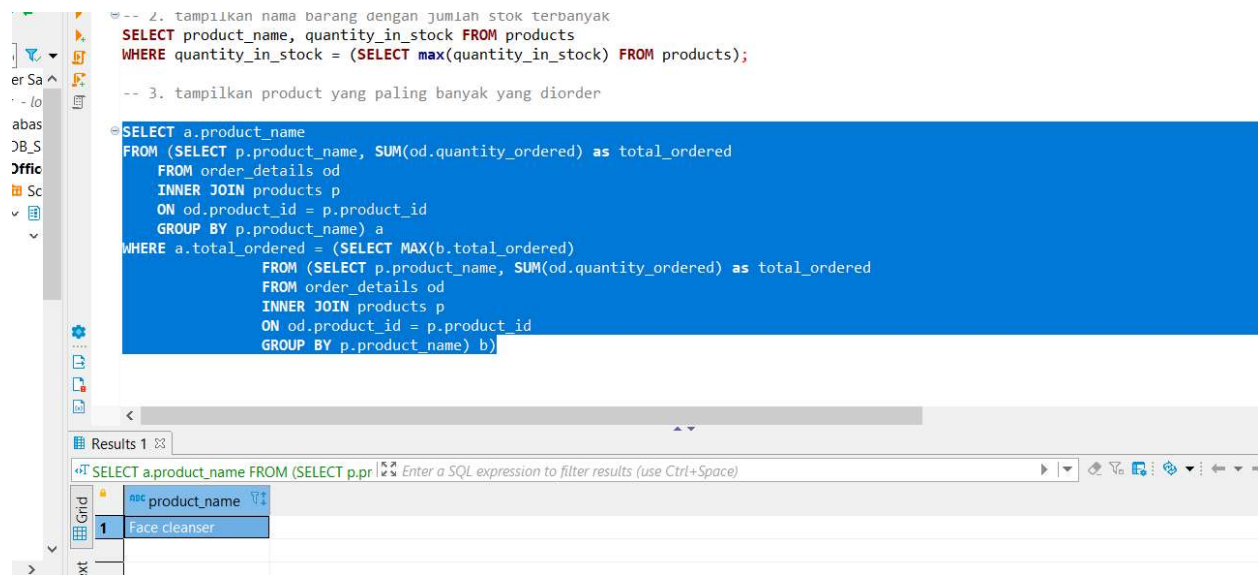
Results 1

	product_name	quantity_in_stock
1	Pull Up Bar	200

Query:

```
SELECT product_name, quantity_in_stock FROM products
WHERE quantity_in_stock = (SELECT max(quantity_in_stock) FROM products);
```

-- 3. tampilkan product yang paling banyak yang diorder



Query:

```
SELECT a.product_name
FROM (SELECT p.product_name, SUM(od.quantity_ordered) as total_ordered
      FROM order_details od
      INNER JOIN products p
      ON od.product_id = p.product_id
      GROUP BY p.product_name) a
WHERE a.total_ordered = (SELECT MAX(b.total_ordered)
                        FROM (SELECT p.product_name, SUM(od.quantity_ordered) as
total_ordered
                        FROM order_details od
                        INNER JOIN products p
                        ON od.product_id = p.product_id
                        GROUP BY p.product_name) b)
```