Projekt w ramach przedmiotu Bazy Danych

Skład zespołu:
Jan Boduch
Mariusz Rogucki
Sebastian Dąbek

Wykorzystane technologie:

- Docker
- Docker Compose
- MySQL
- PHP7
- Wrapper
- orphans/mysql-wrapper

Uruchomienie programu:

Opcja nr 1:
Uruchomienie za pomocą skryptu *„start.sh" (./start.sh, w razie potrzeby chmod + x start.sh)*

Opcja nr 2:
Wykonanie poleceń:
docker build -t my_php -f website/Dockerfile website
docker-compose up

Opis wykorzystanych technologii:

Docker Compose to narzędzie do uruchamiania wielokontenerowych aplikacji. Kontenery, które zostaną uruchomione są zdefiniowane w pliku docker-compose.yml

Stworzony przez nas plik docker-compose:

```
version: '3'

services:

  db:
    image: mysql:5.7
    environment:
      MYSQL_ROOT_PASSWORD: 123
      MYSQL_DATABASE: test_db
      MYSQL_USER: user1
      MYSQL_PASSWORD: 123
    volumes:
      - ./data:/var/lib/mysql
    ports:
      - 5001:3306
    restart: always

  phpmyadmin:
    depends_on:
      - db
    image: phpmyadmin/phpmyadmin
    restart: always
    ports:
      - '5002:80'
    environment:
      PMA_HOST: db
      MYSQL_ROOT_PASSWORD: 123

  website:
    image: my_php
    volumes:
      - ./website:/var/www/html
    ports:
      - 5000:80
    depends_on:
      - db
```

orphans/mysql-wrapper: „A database wrapper class to help reduce code and support rapid development”

Jak skonfigurować?

1.) Stworzyć plik connect.php

```
<?php

require_once('wrapper/mysql_wrapper.class.php');

$db = new MYSQL_WRAPPER();

$db->connect(array(
	'host' => 'db',
	'port' => '3306',
	'username' => 'user1',
	'password' => '123',
	'database' => 'test_db'
));
```

2.) Wczytać plik connect.php w module, w którym planujemy go użyć

```
include(„connect.php");
```

3.)Wykonać zapytanie

INSERT

```
$db->insert('users', [
	'name' => 'John Smith',
	'email' => 'john.smith@somedomain.com',
	'last_updated' => 'NOW()',
]);
```

UPDATE

```
$db->update("employees", $_POST, "`EmployeeID`=:employeeID", ['employeeID' => $employeeID]);
```

DELETE

```
$delete = $db->delete("employees", "`EmployeeID`=:employeeID",
	[
		'employeeID' => $employeeID
	]
);
```

SELECT

```
$db->select("SELECT * FROM `orders`");
```

Operacje CRUD są wykonywane na tabelach Employee, Customers, Orders, Products.

Mamy również możliwość skorzystania z modułu Stats.

Moduł Stats:

Jak widać poniżej mamy do wyboru 9 opcji odnośnie statystyk serwisu.

1. Ilość zamówionych produktów przez daną firmę
2. Ilość produktów w konkretnym zamówieniu
3. Ilość zamówień dokonanych przez daną firmę
4. Najczęściej zamawiane produkty
5. Ilość zamówień zrealizowanych przez danego pracownika
6. Ilość zamówień do konkretnych miast
7. Historia zamówień
8. 10 najdroższych zamówień
9. Zamówienia rok do roku

Ad1. W każdym zapytanie mamy wyświetlany czas w mikrosekundach.

Ilość zamówionych produktów przez daną firmę

Czas wykonywania zapytania = 9.0599060058594 mikrosekund

No	Company Name	TotalQuantitiy
1	Save-a-lot Markets	4958
2	Ernst Handel	4543
3	QUICK-Stop	3961
4	Hungry Owl All-Night Grocers	1684
5	Frankenversand	1525
6	Rattlesnake Canyon Grocery	1383
7	Folk och f? HB	1234
8	HILARI?N-Abastos	1096
9	Supr?mes d?lices	1079
10	White Clover Markets	1063
11	Queen Cozinha	1031
12	Berglunds snabbk?p	1001
13	Bon app'	980

Zapytanie:

```
SELECT CompanyName, sum(howMany) as TotalQuantitiy
FROM (SELECT CompanyName, OrderID, sum(Quantity) as howMany FROM `orders`
NATURAL JOIN `customers` NATURAL JOIN `order_details` GROUP BY OrderID
ORDER BY CompanyName, OrderID) AS table1 GROUP BY CompanyName ORDER BY
`TotalQuantitiy`  DESC
```

Ad2.

Ilość produktów w konkretnym zamówieniu

Czas wykonywania zapytania = 0 mikrosekund

No	OrderID	Ilość produktów
1	10895	346
2	11030	330
3	10847	288
4	10515	286
5	10678	280
6	10612	263
7	10990	256
8	10658	255
9	10845	245
10	10324	241
11	10451	238
12	10514	233
13	10595	215
14	10607	213
15	10440	208

Zapytanie:

```
SELECT OrderID, sum(Quantity) FROM `order_details` GROUP BY OrderID ORDER BY `sum(Quantity)`  DESC
```

Ad3.

Ilość zamówień dokonanych przez daną firmę

Czas wykonywania zapytania = 0 mikrosekund

No	CustomerID	CompanyName	Ilość zamówień
1	RATTC	Rattlesnake Canyon Grocery	25
2	SAVEA	Save-a-lot Markets	6
3	ERNSH	Ernst Handel	6
4	SAVEA	Save-a-lot Markets	6
5	QUEEN	Queen Cozinha	5
6	SUPRD	Supr?mes d?lices	5
7	ERNSH	Ernst Handel	5
8	AROUT	Around the Horn	5
9	FRANK	Frankenversand	5
10	ERNSH	Ernst Handel	5

Zapytanie:

```
SELECT CustomerID, CompanyName, count(OrderID) FROM `customers`
NATURAL JOIN `orders` NATURAL JOIN `order_details` GROUP BY OrderID ORDER BY
`count(OrderID)`  DESC LIMIT 10
```

Ad 4.

Najczęściej zamawiane produkty

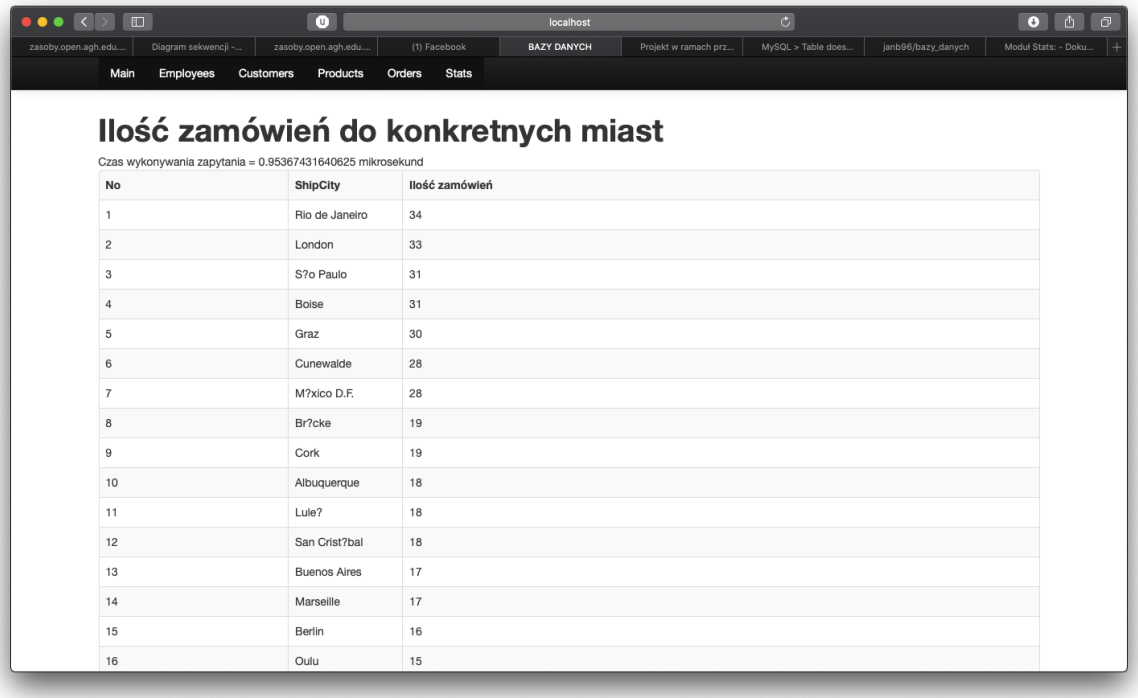
Czas wykonywania zapytania = 0.95367431640625 mikrosekund

No	ProductID	Ilość zamówień)
1	60	1577
2	59	1496
3	31	1397
4	56	1263
5	16	1158
6	75	1155
7	24	1125
8	40	1103
9	62	1083
10	2	1057
11	71	1057
12	21	1016
13	1	985
14	41	981
15	76	981

Zapytanie:

```
SELECT ProductID, sum(Quantity) as howMuch FROM `order_details` GROUP BY ProductID ORDER BY `howMuch` DESC
```

Ad5.



No	ShipCity	Ilość zamówień
1	Rio de Janeiro	34
2	London	33
3	S?o Paulo	31
4	Boise	31
5	Graz	30
6	Cunewalde	28
7	M?xico D.F.	28
8	Br?cke	19
9	Cork	19
10	Albuquerque	18
11	Lule?	18
12	San Crist?bal	18
13	Buenos Aires	17
14	Marseille	17
15	Berlin	16
16	Oulu	15

Zapytanie:

```
SELECT EmployeeID, count(OrderID) as howMuch
FROM `order_details` NATURAL JOIN `orders` NATURAL JOIN `employees` GROUP BY
EmployeeID ORDER BY `howMuch`  DESC
```

Ad 6.

Ilość zamówień do konkretnych miast

Czas wykonywania zapytania = 0 mikrosekund

No	ShipCity	Ilość zamówień
1	Rio de Janeiro	34
2	London	33
3	S?o Paulo	31
4	Boise	31
5	Graz	30
6	M?xico D.F.	28
7	Cunewalde	28
8	Br?cke	19
9	Cork	19
10	Albuquerque	18
11	Lule?	18
12	San Crist?bal	18
13	Buenos Aires	17
14	Marseille	17

Zapytanie:

```
SELECT ShipCity, count(OrderID) as howMuch FROM `orders` GROUP BY ShipCity
ORDER BY `howMuch` DESC
```

Opcje 7-9 przedstawiają widoki.
Ad 7.

Historia zamówień

Czas wykonywania zapytania = 0.95367431640625 mikrosekund

No	Rok zamówienia	Miesiąc zamówienia	Ilość zamówień
1	1996	7	22
2	1996	8	25
3	1996	9	23
4	1996	10	26
5	1996	11	25
6	1996	12	31
7	1997	1	33
8	1997	2	29
9	1997	3	30
10	1997	4	31
11	1997	5	32
12	1997	6	30
13	1997	7	33
14	1997	8	33

Zapytanie:
SELECT * FROM uvw_orders_historically

```
1. CREATE ALGORITHM=UNDEFINED DEFINER=`root`@`%` SQL SECURITY DEFINER VIEW
   `uvw_orders_historically` AS selectyear(`orders`.`OrderDate`) AS
   `year(OrderDate)`,month(`orders`.`OrderDate`) AS
   `month(OrderDate)`,count(`orders`.`OrderID`) AS`count(OrderID)` from `orders`
   group by year(`orders`.`OrderDate`),month(`orders`.`OrderDate`);
```

Ad 8.

10 najdroższych zamówień

Czas wykonywania zapytania = 0.95367431640625 mikrosekund

No	OrderID	Wartość zamówienia
1	11030	2070
2	10847	1727
3	10607	1558
4	10549	1517
5	10514	1482
6	10776	1429
7	10430	1405
8	10678	1400
9	11017	1385
10	10515	1282

Projekt na **Bazy Danych** by Jan Boduch, Sebastian Dąbek, Mariusz Rogucki. 2020.

Zapytanie:
SELECT * FROM uvw_orders_top10_expensive

```
1. CREATE ALGORITHM=UNDEFINED DEFINER=`root`@`%` SQL SECURITY DEFINER VIEW
   `uvw_orders_top10_expensive` AS select `od`.`OrderID`
   AS`OrderID`,sum((`p`.`UnitPrice` * `od`.`Quantity`)) AS `OrderAmount` from
   (`order_details` `od` left join `products` `p`on((`od`.`ProductID` =
   `p`.`ProductID`))) group by `od`.`OrderID` order by `OrderAmount` desc limit
   10;
```

Ad 9.

Zamówienia rok do roku

Czas wykonywania zapytania = 0.95367431640625 mikrosekund

No	Rok	Wartość zamówienia
1		58
2	1996	48289
3	1997	126340
4	1998	76473
5	2020	1757

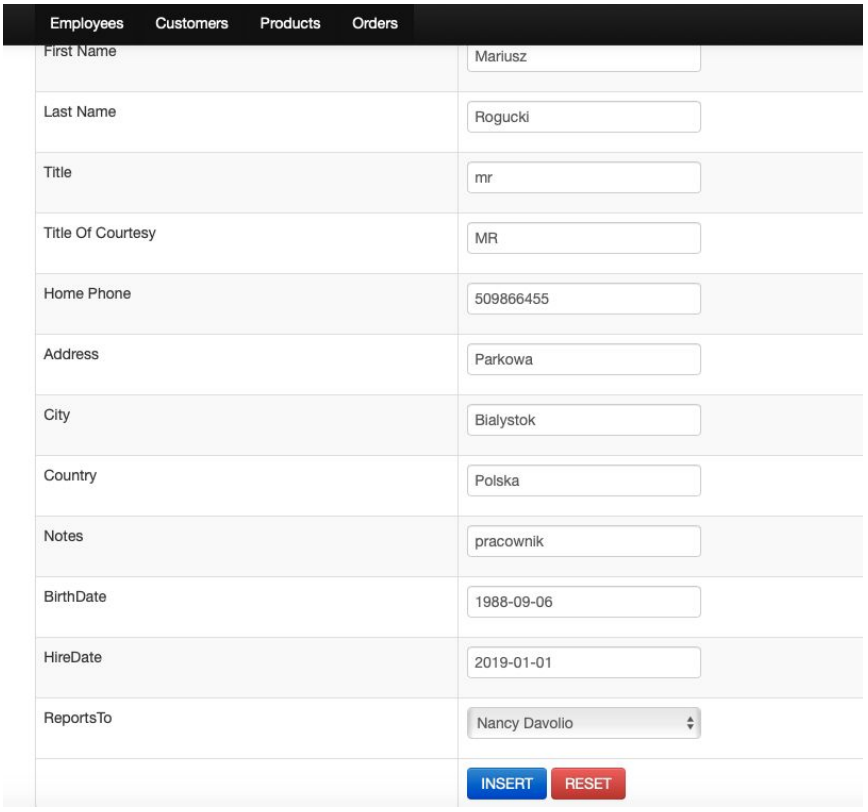
Zapytanie:

SELECT * FROM uvw_orders_y2y_summary

```
1. CREATE ALGORITHM=UNDEFINED DEFINER=`root`@`%` SQL SECURITY DEFINER VIEW
   `uvw_orders_y2y_summary` AS select year(`o`.`OrderDate`)AS
   `year(o.OrderDate)`,sum((`p`.`UnitPrice` * `od`.`Quantity`)) AS `OrderAmount`
   from ((`order_details` `od` left join `products``p` on((`od`.`ProductID` =
   `p`.`ProductID`))) left join `orders` `o` on((`od`.`OrderID` =
   `o`.`OrderID`))) group byyear(`o`.`OrderDate`);
```

Operacja CRUD na tabeli Employee.

1. Dodawanie pracownika.

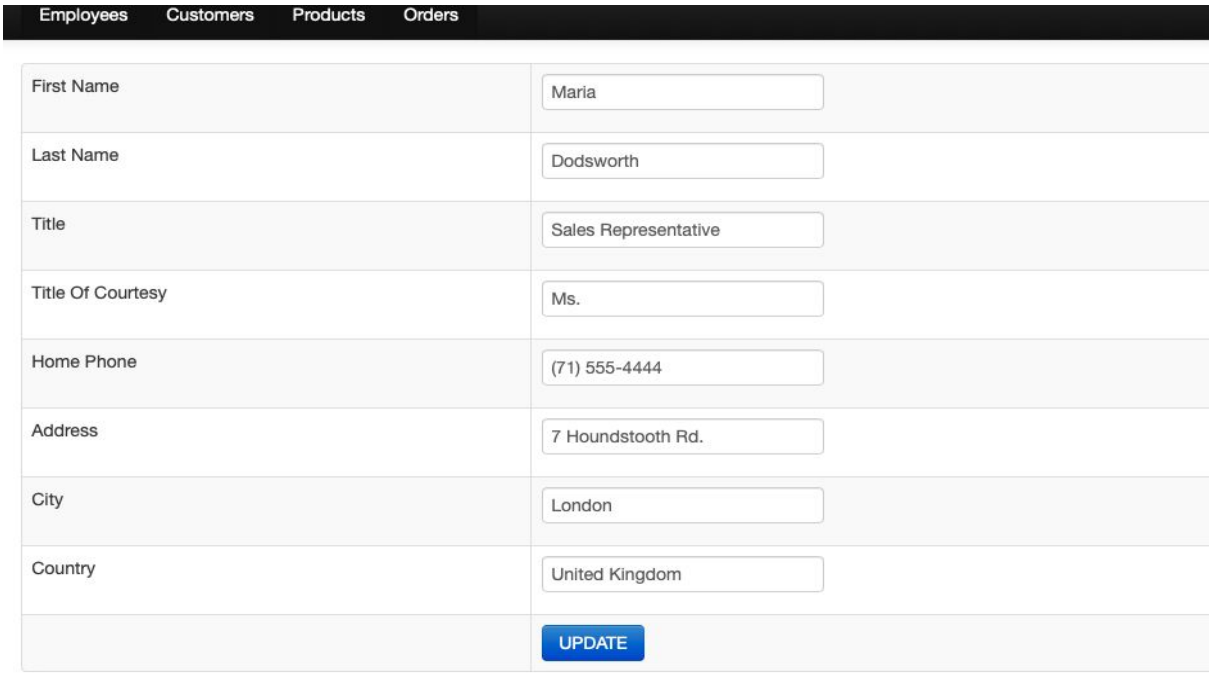


i po kliknięciu “INSERT” otrzymujemy taki wynik:

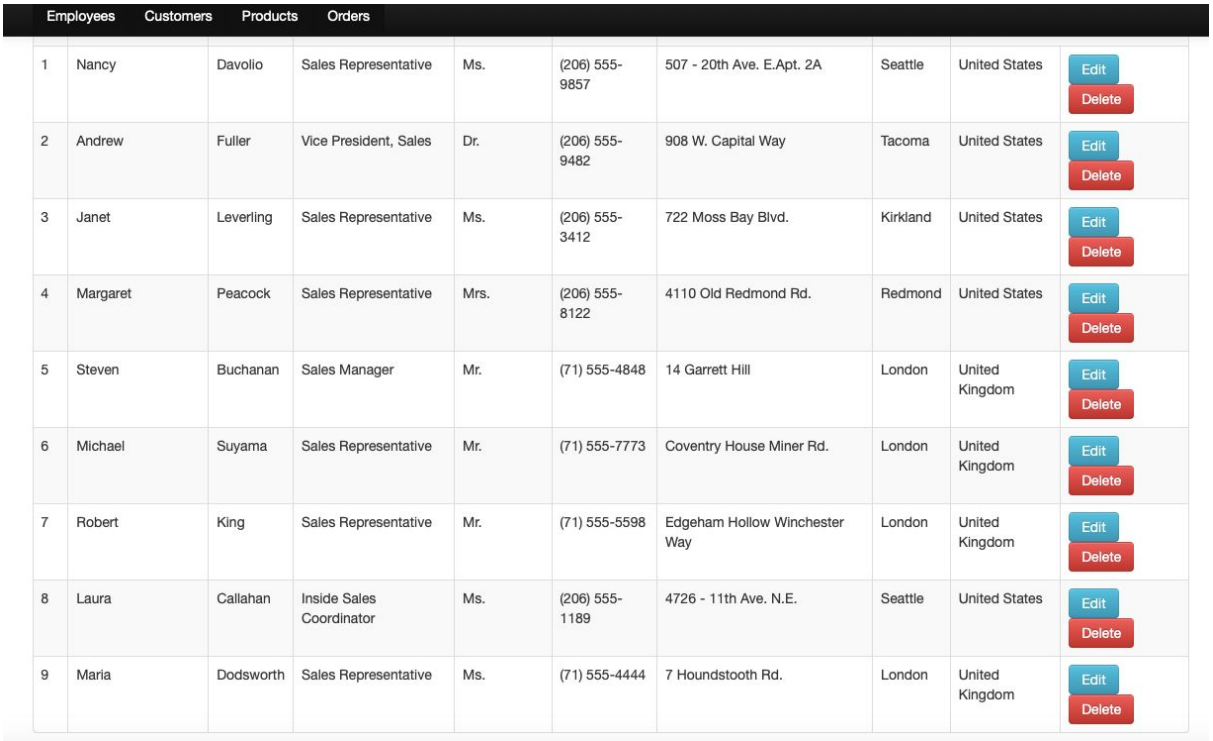
Następnie na liście wszystkich pracowników widzimy nowo dodanego pracownika:

9	Anne	Dodsworth	Sales Representative	Ms.	(71) 555-4444	7 Houndstooth Rd.	London	United Kingdom	Edit Delete
10	Mariusz	Rogucki	mr	MR	509866455	Parkowa	Bialystok	Polska	Edit Delete

Możemy również edytować pracownika. Po kliknięciu “Edit” otrzymujemy taki wynik:



Po zmianie danych (w tym przypadku imienia) i kliknięciu “UPDATE” widzimy na liście pracowników dokonane zmiany (zmiana w pozycji 9. imienia Anne na Maria):



1	Nancy	Davolio	Sales Representative	Ms.	(206) 555-9857	507 - 20th Ave. E.Apt. 2A	Seattle	United States	Edit Delete
2	Andrew	Fuller	Vice President, Sales	Dr.	(206) 555-9482	908 W. Capital Way	Tacoma	United States	Edit Delete
3	Janet	Leverling	Sales Representative	Ms.	(206) 555-3412	722 Moss Bay Blvd.	Kirkland	United States	Edit Delete
4	Margaret	Peacock	Sales Representative	Mrs.	(206) 555-8122	4110 Old Redmond Rd.	Redmond	United States	Edit Delete
5	Steven	Buchanan	Sales Manager	Mr.	(71) 555-4848	14 Garrett Hill	London	United Kingdom	Edit Delete
6	Michael	Suyama	Sales Representative	Mr.	(71) 555-7773	Coventry House Miner Rd.	London	United Kingdom	Edit Delete
7	Robert	King	Sales Representative	Mr.	(71) 555-5598	Edgeham Hollow Winchester Way	London	United Kingdom	Edit Delete
8	Laura	Callahan	Inside Sales Coordinator	Ms.	(206) 555-1189	4726 - 11th Ave. N.E.	Seattle	United States	Edit Delete
9	Maria	Dodsworth	Sales Representative	Ms.	(71) 555-4444	7 Houndstooth Rd.	London	United Kingdom	Edit Delete

Możemy również usunąć wybranego pracownika. W tym celu klikamy “Delete”.

1	Nancy	Davolio	Sales Representative	Ms.	(206) 555-9857	507 - 20th Ave. E. Apt. 2A	Seattle	United States	Edit Delete
2	Andrew	Fuller	Vice President, Sales	Dr.	(206) 555-9482	908 W. Capital Way	Tacoma	United States	Edit Delete
3	Janet	Leverling	Sales Representative	Ms.	(206) 555-3412	722 Moss Bay Blvd.	Kirkland	United States	Edit Delete
4	Margaret	Peacock	Sales Representative	Mrs.	(206) 555-8122	4110 Old Redmond Rd.	Redmond	United States	Edit Delete
5	Steven	Buchanan	Sales Manager	Mr.	(71) 555-4848	14 Garrett Hill	London	United Kingdom	Edit Delete
6	Michael	Suyama	Sales Representative	Mr.	(71) 555-7773	Coventry House Miner Rd.	London	United Kingdom	Edit Delete
7	Robert	King	Sales Representative	Mr.	(71) 555-5598	Edgeham Hollow Winchester Way	London	United Kingdom	Edit Delete
8	Laura	Callahan	Inside Sales Coordinator	Ms.	(206) 555-1189	4726 - 11th Ave. N.E.	Seattle	United States	Edit Delete

Powyżej widzimy tabelę po usunięciu pracownika z numerem porządkowym 9.

Procedura skladowalna

```
CREATE PROCEDURE newOrderWithParam
(
   quantityParameter int(11)
)
BEGIN
  insert into order_details (ProductID, UnitPrice, Quantity, Discount)
  select
   odp.ProductID
   ,p.UnitPrice
   ,quantityParameter as `Quantity`
   ,0 as `Discount`
   from order_data_processing odp
   left join products p on odp.ProductID = p.ProductID;


   insert into orders (CustomerID, EmployeeID, OrderDate, RequiredDate,
ShippedDate, ShipVia, Freight, ShipName, ShipAddress, ShipCity, ShipRegion,
ShipPostalCode, ShipCountry)
   select
   -- autoincrement OrderID,
   odp.CustomerID
   ,1 as EmployeeID
   ,curdate() as OrderDate
```

```
   ,DATE_ADD(curdate(), INTERVAL 30 DAY) as RequiredDate
   ,NULL as ShippedDate
   ,1 as ShipVia
   ,1 as Freight
   ,c.CompanyName as ShipName
   ,c.Address as ShipAddress
   ,c.City as ShipCity
   ,c.Region as ShipRegion
   ,c.PostalCode as ShipPostalCode
   ,c.Country as ShipCountry
   from order_data_processing odp
   left join customers c on odp.CustomerID = c.CustomerID;

   delete
   from order_data_processing;
END //
```