OMAR ISMAIL ABDJALEEL ALOMORY

Matric No: S63955

Date: 12/01/2023

Lab: MP3

Lecturer: Dr. ROSAIDA ROSLY

MUHAMMAD NASYRUL ADLY BIN MOHD AFANDY

Lab 7

DATABASE (CSF3123) K2

Activity 1

Task 1:

Step 1; List record in tublar form:

0 NF

0 NF or known as "pre-normalization" or “raw data” refers to the initial state of data, before it has been organized into tables and implemented with any normalization rules. It is essentially an unstructured collection of data. In 0NF, there is no defined primary key, no relationships between tables and data is not split into multiple tables. It is just a collection of rows and columns, data is repeated for every record.

Order No	Order Type	Order Date	Delivery Date	Code item	Item Description	Order Quantity	Unit	Selling Price (RM)	Amount (RM)
1000021	Company Order	10/10/2016	10/17/2016	BL-2010	Lining B2010 Racket.	10	piece	200	2000
				YN-20	Yonex - Carbonex 20 Racket.	10	piece	200	2200

Customer Code	Name	Address	Post-code	City	State	HP Telephone	Email	Salesman ID	Salesman Name
CS21578	Hakim Adlin	No. 12, Lorong Kenari 1AU/5, Seksyen 7	41200	Shah Alam	Selangor	133873030	hakim@gmail.vom	SLS200	Abdullah Razali

*Record in tabular form (**0NF**)*

Notation of order table:

TBL Order (Order No, Order Type, Order Date, Delivery Date, (Item Code, Item Desc, Order Qty, Unit, SellingPrice, Amount), Customer Code, Name, Address, Postcode, City, State, HP No, Email, Sales Id, Saleman Name)

Step 2; Perform 1st Normal Form

A table in 1NF if it attributes do not contain repeating groups.

Step 2.1; Then, need to remove repeating groups.

<u>Order No</u>	Order Type	Order Date	Delivery Date	**<u>Code item</u>**	Item Description	Order Quantity	Unit	Selling Price (RM)	Amount (RM)
1000021	Company Order	10/10/2016	10/17/2016	BL-2010	Lining B2010 Racket	10	piece	200	2000
1000021	Company Order	10/10/2016	10/17/2016	YN-20	Yonex - Carbonex 20 Racket	10	piece	200	2200

Customer Code	Name	Address	Post-code	City	State	HP Telephone	Email	Salesman ID	Salesman Name
CS21578	Hakim Adlin	No. 12, Lorong Kenari 1AU/5, Seksyen 7	41200	Shah Alam	Selangor	133873030	hakim@gmail.vom	SLS200	Abdullah Razali
CS21578	Hakim Adlin	No. 12, Lorong Kenari 1AU/5, Seksyen 7	41200	Shah Alam	Selangor	133873030	hakim@gmail.vom	SLS200	Abdullah Razali

1st NF

Step 2.2; Identify a list of primary keys.

Order No -> Order Type, Order Date, Delivery Date, Customer Code, Name, Address, Postcode, City, State, HP No, Email, Sales Id, Saleman Name

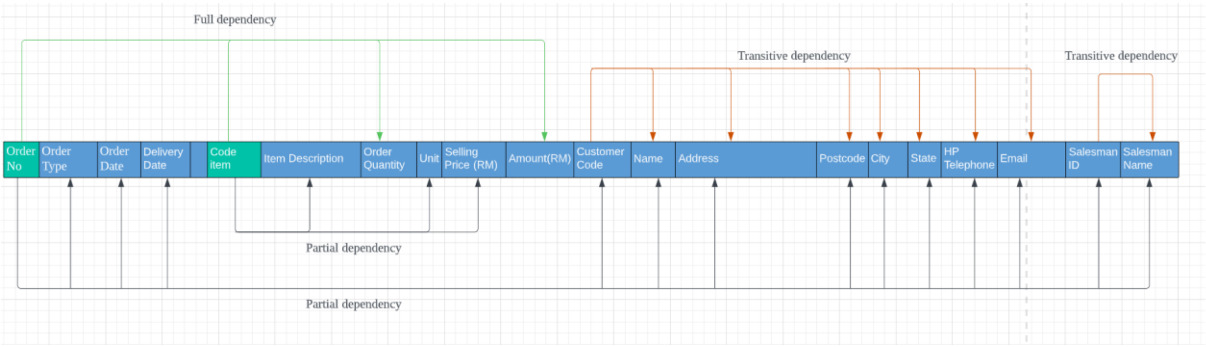
Item Code -> Item Desc, Unit, Selling Price

Order No, Item Code -> Order Qty, Amount

Step 2.3; Identify a primary key for 1NF (i.e; each tuple/row is unique based on this primary key)

Order No, Item Code -> Order Qty, Amount

Figure 2.3.1 Dependency Diagram



Step 3; Perform Second Normal Form (2NF)

Step 3.1; Identify the functional dependencies.

FD1 -> Order No, Item Code -> Order Qty, Amount (Full Dependency)

FD2 -> Order No -> Order Type, Order Date, Delivery Date, Customer Code, Name, Address, Postcode, City,

State, HP No, Email, Sales Id, Saleman Name(Partial Dependency)

FD3 -> Item Code -> Item Desc, Unit, Selling Price (Partial Dependency)

FD4 -> Customer Code -> Name, Address, Postcode, City, State, HPNo, Email (Transitive Dependency)

FD5 -> SalesmanId -> SalesmanName (Transitive Dependency)

Step 3.2; Create a relation for FD1 since its already identify as full dependency.

TBL Order_Item (Order No, Item Code, Order Qty, Amount)

Step 3.3; Find from a list of functional dependencies that identify as partial dependency

- Based on a list from functional dependencies, FD2 and FD3 is a partial dependency.
- A table in 2NF if it fulfil 1NF and non-key attribute is dependent on only a portion of primary key (partial dependencies).

Step 3.4; Remove partial dependency

- Based on a list from functional dependencies, remove partial dependency for FD2 and FD3 by creating TWO (2) new relation; **Order** and **Item**.

  TBL Order (Order No, Order Type, Order Date, Delivery Date, Customer Code, Name, Address, Postcode, City, State, HP No, Email, Salesman Id, Saleman Name)

  TBL Item (Item Code , Item Desc, Unit, Selling Price)

- At 2NF, THREE (3) relations created; **Order** , **Order_item** and **Item**. Shown below

TBL Order

Order No	Order Type	Order Date	Delivery Date	Customer Code	Name	Address	PostCode	City	State
1000021	Company Order	10/10/2016	10/17/2016	CS21578	Hakim Adlin	No. 12, Lorong Kenari 1AU/5, Seksyen 7	41200	Shah Alam	Selangor

HP No	Email	Sales id	Salesman name
133873030	hakim@gmail.vom	SLS200	Abdullah Razali

TBL Order_item

Order No	Item Code	Order Quantity	Amount
1000021	BL-2010	10	2000.00
1000021	YN-20	10	2200.00

TBL Item

Item Code	Item Description	Unit	Selling Price
BL-2010	Lining B2010 Racket	price	200.00
YN-20	Yonex-Carbonex 20 racket	price	220.00

Step 4; Perform Third Normal Form (3NF)

A table in 3NF if it fulfils 2NF and any column (or collection of columns) that determines another column is called a determinant but it is not a candidate key (primary key) for that table.

Step 4.1; Find from a list of functional dependencies that identify as transitive dependency.

- In 2NF form, we can assume that there is still a problem with a relation **Order** that can lead to data anomalies.

  Order (Order No, Order Type, Order Date, Delivery Date, Customer Code, Name, Address, Postcode, City, State, HP No, Email, Sales Id, Saleman Name)

TBL Order

Order No	Order Type	Order Date	Delivery Date	Customer Code	Name	Address	PostCode	City	State
1000021	Company Order	10/10/2016	10/17/2016	CS21578	Hakim Adlin	No. 12, Lorong Kenari 1AU/5, Seksyen 7	41200	Shah Alam	Selangor

HP No	Email	Sales id	Salesman name
133873030	hakim@gmail.vom	SLS200	Abdullah Razali

- **Customer code** is a determinant for other attributes that related to customer information but not a candidate key for relation **Order**.
- **Sales Id** is a determinant for **Salesman Name** but not a candidate key for relation **Order**.
  FD2 -> **Order No** -> Order Type, Order Date, Delivery Date, Customer Code, Name, Address, Postcode, City, State, HP No, Email, Sales Id, Saleman Name.
- **FD4** and **FD5** is a transitive dependency.
  FD4 -> **CustCode** -> Name, Address, Postcode, City, State, HPNo, Email (Transitive Dependency)
  FD5 -> **SalesmanId** -> SalesmanName (Transitive Dependency)

Step 4.2; Remove transitive dependency in FD4 and FD5 by creating a relation known as Customer and Salesman

- **Customer** (CustCode, Name, Address, Postcode, City, State, HPNo, Email)
- **Salesman**(SalesmanId, SalesmanName)

Step 4.3; Add a primary key for Customer and Salesman relation as a foreign key to an existing relation of Order

Order (Order No, Order Type, Order Date, Delivery Date, Cust Code, Salesman Id)

Finally, we have final relations for the case study after **3NF**; **Order**, **Order_item**, **Item**, **Customer**, **Salesman**

TBL Order

Order No	Order Type	Order Date	Delivery Date
1000021	Company Order	10/10/2016	10/17/2016

TBL Order_item

Order No	Item Code	Order Quantity	Amount
1000021	BL-2010	10	2000.00
1000021	YN-20	10	2200.00

TBL Item

Item Code	Item Description	Unit	Selling Price
BL-2010	Lining B2010 Racket	price	200.00
YN-20	Yonex-Carbonex 20 racket	price	220.00

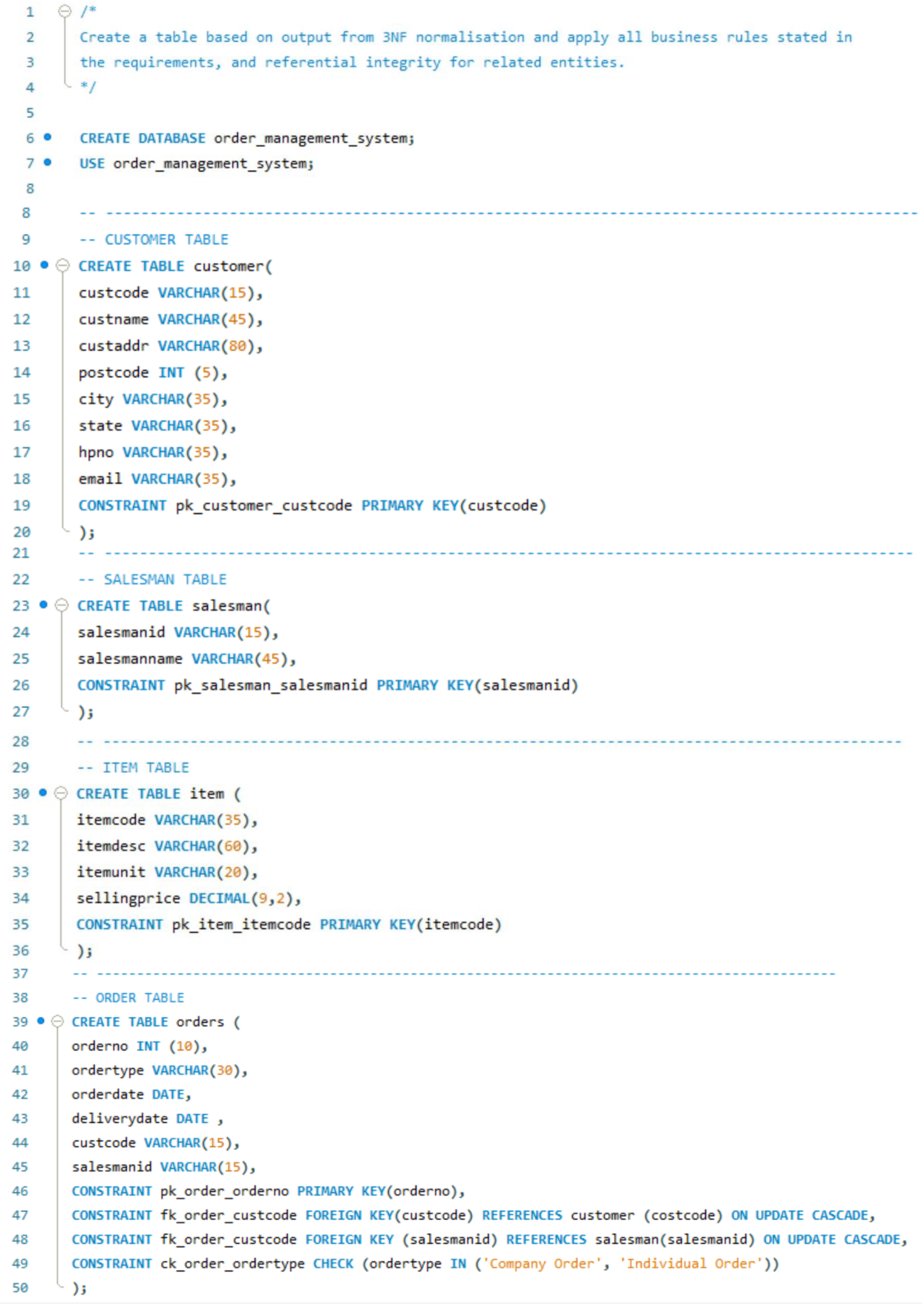
TBL Customer

Customer Code	Name	Address	PostCode	City	State	HP No	Email
CS21578	Hakim Adlin	No. 12, Lorong Kenari 1AU/5, Seksyen 7	41200	Shah Alam	Selangor	133873030	hakim@gmail.vom

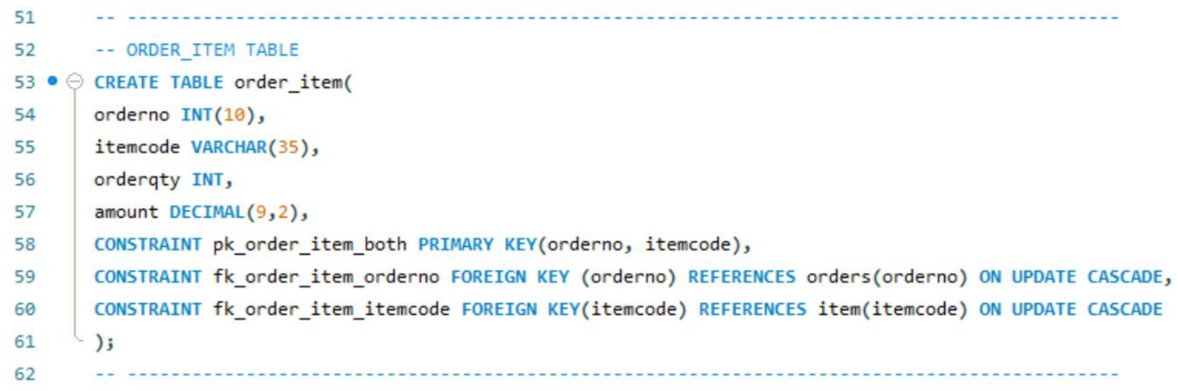
TBL Salesman

Sales id	Salesman name
SLS200	Abdullah Razali

Task 2:

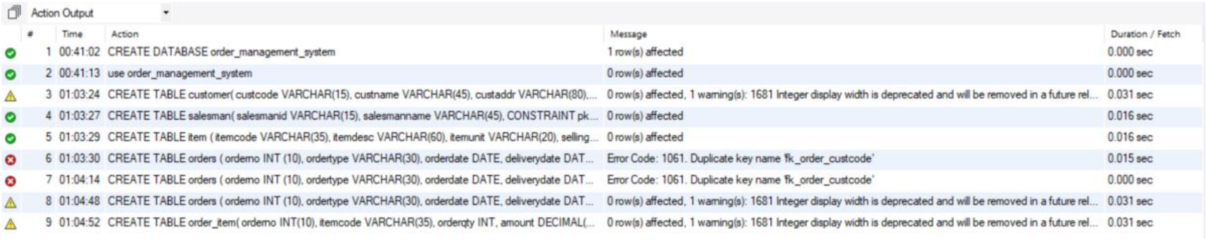


```
/*
Create a table based on output from 3NF normalisation and apply all business rules stated in
the requirements, and referential integrity for related entities.
*/

CREATE DATABASE order_management_system;
USE order_management_system;

-- -------------------------------------------------------------------------------------------
-- CUSTOMER TABLE
CREATE TABLE customer(
custcode VARCHAR(15),
custname VARCHAR(45),
custaddr VARCHAR(80),
postcode INT (5),
city VARCHAR(35),
state VARCHAR(35),
hpno VARCHAR(35),
email VARCHAR(35),
CONSTRAINT pk_customer_custcode PRIMARY KEY(custcode)
);
-- -------------------------------------------------------------------------------------------
-- SALESMAN TABLE
CREATE TABLE salesman(
salesmanid VARCHAR(15),
salesmanname VARCHAR(45),
CONSTRAINT pk_salesman_salesmanid PRIMARY KEY(salesmanid)
);
-- -------------------------------------------------------------------------------------------
-- ITEM TABLE
CREATE TABLE item (
itemcode VARCHAR(35),
itemdesc VARCHAR(60),
itemunit VARCHAR(20),
sellingprice DECIMAL(9,2),
CONSTRAINT pk_item_itemcode PRIMARY KEY(itemcode)
);
-- -------------------------------------------------------------------------------------------
-- ORDER TABLE
CREATE TABLE orders (
orderno INT (10),
ordertype VARCHAR(30),
orderdate DATE,
deliverydate DATE ,
custcode VARCHAR(15),
salesmanid VARCHAR(15),
CONSTRAINT pk_order_orderno PRIMARY KEY(orderno),
CONSTRAINT fk_order_custcode FOREIGN KEY(custcode) REFERENCES customer (costcode) ON UPDATE CASCADE,
CONSTRAINT fk_order_custcode FOREIGN KEY (salesmanid) REFERENCES salesman(salesmanid) ON UPDATE CASCADE,
CONSTRAINT ck_order_ordertype CHECK (ordertype IN ('Company Order', 'Individual Order'))
);
```



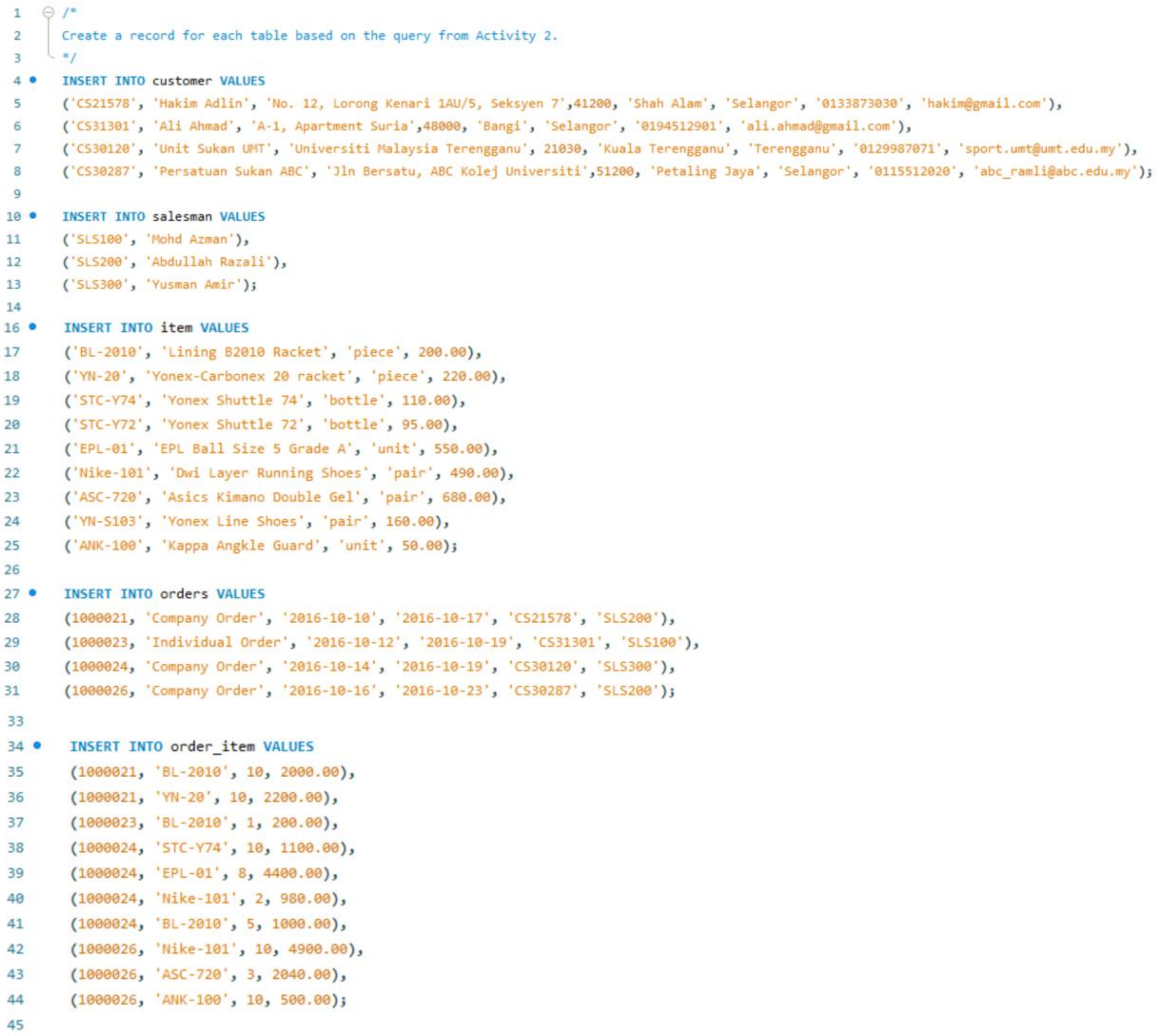
```sql
-- ---------------------------------------------------------------------------------------------
-- ORDER_ITEM TABLE
CREATE TABLE order_item(
orderno INT(10),
itemcode VARCHAR(35),
orderqty INT,
amount DECIMAL(9,2),
CONSTRAINT pk_order_item_both PRIMARY KEY(orderno, itemcode),
CONSTRAINT fk_order_item_orderno FOREIGN KEY (orderno) REFERENCES orders(orderno) ON UPDATE CASCADE,
CONSTRAINT fk_order_item_itemcode FOREIGN KEY(itemcode) REFERENCES item(itemcode) ON UPDATE CASCADE
);
-- ---------------------------------------------------------------------------------------------
```

Output



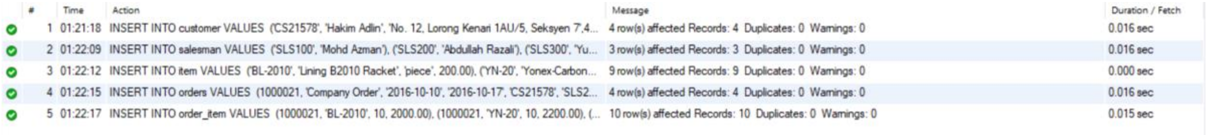
	#	Time	Action	Message	Duration / Fetch
✓	1	00:41:02	CREATE DATABASE order_management_system	1 row(s) affected	0.000 sec
✓	2	00:41:13	use order_management_system	0 row(s) affected	0.000 sec
⚠	3	01:03:24	CREATE TABLE customer(custcode VARCHAR(15), custname VARCHAR(45), custaddr VARCHAR(80),...	0 row(s) affected, 1 warning(s): 1681 Integer display width is deprecated and will be removed in a future rel...	0.031 sec
✓	4	01:03:27	CREATE TABLE salesman(salesmanid VARCHAR(15), salesmanname VARCHAR(45), CONSTRAINT pk...	0 row(s) affected	0.016 sec
✓	5	01:03:29	CREATE TABLE item (itemcode VARCHAR(35), itemdesc VARCHAR(60), itemunit VARCHAR(20), selling...	0 row(s) affected	0.016 sec
⊗	6	01:03:30	CREATE TABLE orders (orderno INT (10), ordertype VARCHAR(30), orderdate DATE, deliverydate DAT...	Error Code: 1061. Duplicate key name 'fk_order_custcode'	0.015 sec
⊗	7	01:04:14	CREATE TABLE orders (orderno INT (10), ordertype VARCHAR(30), orderdate DATE, deliverydate DAT...	Error Code: 1061. Duplicate key name 'fk_order_custcode'	0.000 sec
⚠	8	01:04:48	CREATE TABLE orders (orderno INT (10), ordertype VARCHAR(30), orderdate DATE, deliverydate DAT...	0 row(s) affected, 1 warning(s): 1681 Integer display width is deprecated and will be removed in a future rel...	0.031 sec
⚠	9	01:04:52	CREATE TABLE order_item(orderno INT(10), itemcode VARCHAR(35), orderqty INT, amount DECIMAL(...	0 row(s) affected, 1 warning(s): 1681 Integer display width is deprecated and will be removed in a future rel...	0.031 sec

Task 3



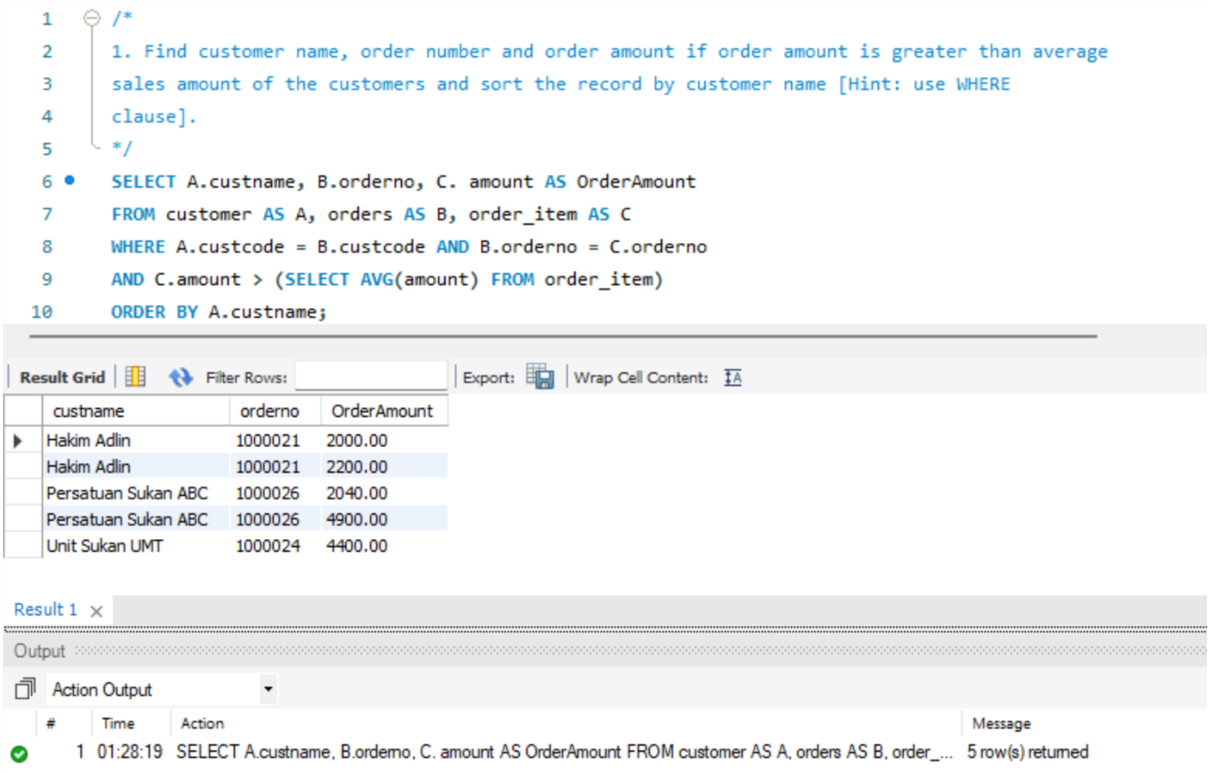
```sql
/*
Create a record for each table based on the query from Activity 2.
*/
INSERT INTO customer VALUES
('CS21578', 'Hakim Adlin', 'No. 12, Lorong Kenari 1AU/5, Seksyen 7',41200, 'Shah Alam', 'Selangor', '0133873030', 'hakim@gmail.com'),
('CS31301', 'Ali Ahmad', 'A-1, Apartment Suria',48000, 'Bangi', 'Selangor', '0194512901', 'ali.ahmad@gmail.com'),
('CS30120', 'Unit Sukan UMT', 'Universiti Malaysia Terengganu', 21030, 'Kuala Terengganu', 'Terengganu', '0129987071', 'sport.umt@umt.edu.my'),
('CS30287', 'Persatuan Sukan ABC', 'Jln Bersatu, ABC Kolej Universiti',51200, 'Petaling Jaya', 'Selangor', '0115512020', 'abc_ramli@abc.edu.my');

INSERT INTO salesman VALUES
('SLS100', 'Mohd Azman'),
('SLS200', 'Abdullah Razali'),
('SLS300', 'Yusman Amir');

INSERT INTO item VALUES
('BL-2010', 'Lining B2010 Racket', 'piece', 200.00),
('YN-20', 'Yonex-Carbonex 20 racket', 'piece', 220.00),
('STC-Y74', 'Yonex Shuttle 74', 'bottle', 110.00),
('STC-Y72', 'Yonex Shuttle 72', 'bottle', 95.00),
('EPL-01', 'EPL Ball Size 5 Grade A', 'unit', 550.00),
('Nike-101', 'Dwi Layer Running Shoes', 'pair', 490.00),
('ASC-720', 'Asics Kimano Double Gel', 'pair', 680.00),
('YN-S103', 'Yonex Line Shoes', 'pair', 160.00),
('ANK-100', 'Kappa Angkle Guard', 'unit', 50.00);

INSERT INTO orders VALUES
(1000021, 'Company Order', '2016-10-10', '2016-10-17', 'CS21578', 'SLS200'),
(1000023, 'Individual Order', '2016-10-12', '2016-10-19', 'CS31301', 'SLS100'),
(1000024, 'Company Order', '2016-10-14', '2016-10-19', 'CS30120', 'SLS300'),
(1000026, 'Company Order', '2016-10-16', '2016-10-23', 'CS30287', 'SLS200');

 INSERT INTO order_item VALUES
 (1000021, 'BL-2010', 10, 2000.00),
 (1000021, 'YN-20', 10, 2200.00),
 (1000023, 'BL-2010', 1, 200.00),
 (1000024, 'STC-Y74', 10, 1100.00),
 (1000024, 'EPL-01', 8, 4400.00),
 (1000024, 'Nike-101', 2, 980.00),
 (1000024, 'BL-2010', 5, 1000.00),
 (1000026, 'Nike-101', 10, 4900.00),
 (1000026, 'ASC-720', 3, 2040.00),
 (1000026, 'ANK-100', 10, 500.00);
```

Output



```
   #  Time      Action                                                                                      Message                                                      Duration / Fetch
✓  1  01:21:18  INSERT INTO customer VALUES  ('CS21578', 'Hakim Adlin', 'No. 12, Lorong Kenari 1AU/5, Seksyen 7',4...  4 row(s) affected Records: 4  Duplicates: 0  Warnings: 0     0.016 sec
✓  2  01:22:09  INSERT INTO salesman VALUES  ('SLS100', 'Mohd Azman'), ('SLS200', 'Abdullah Razali'), ('SLS300', 'Yu...  3 row(s) affected Records: 3  Duplicates: 0  Warnings: 0     0.016 sec
✓  3  01:22:12  INSERT INTO item VALUES  ('BL-2010', 'Lining B2010 Racket', 'piece', 200.00), ('YN-20', 'Yonex-Carbon...  9 row(s) affected Records: 9  Duplicates: 0  Warnings: 0     0.000 sec
✓  4  01:22:15  INSERT INTO orders VALUES  (1000021, 'Company Order', '2016-10-10', '2016-10-17', 'CS21578', 'SLS2...  4 row(s) affected Records: 4  Duplicates: 0  Warnings: 0     0.016 sec
✓  5  01:22:17  INSERT INTO order_item VALUES  (1000021, 'BL-2010', 10, 2000.00), (1000021, 'YN-20', 10, 2200.00), (...  10 row(s) affected Records: 10  Duplicates: 0  Warnings: 0  0.015 sec
```

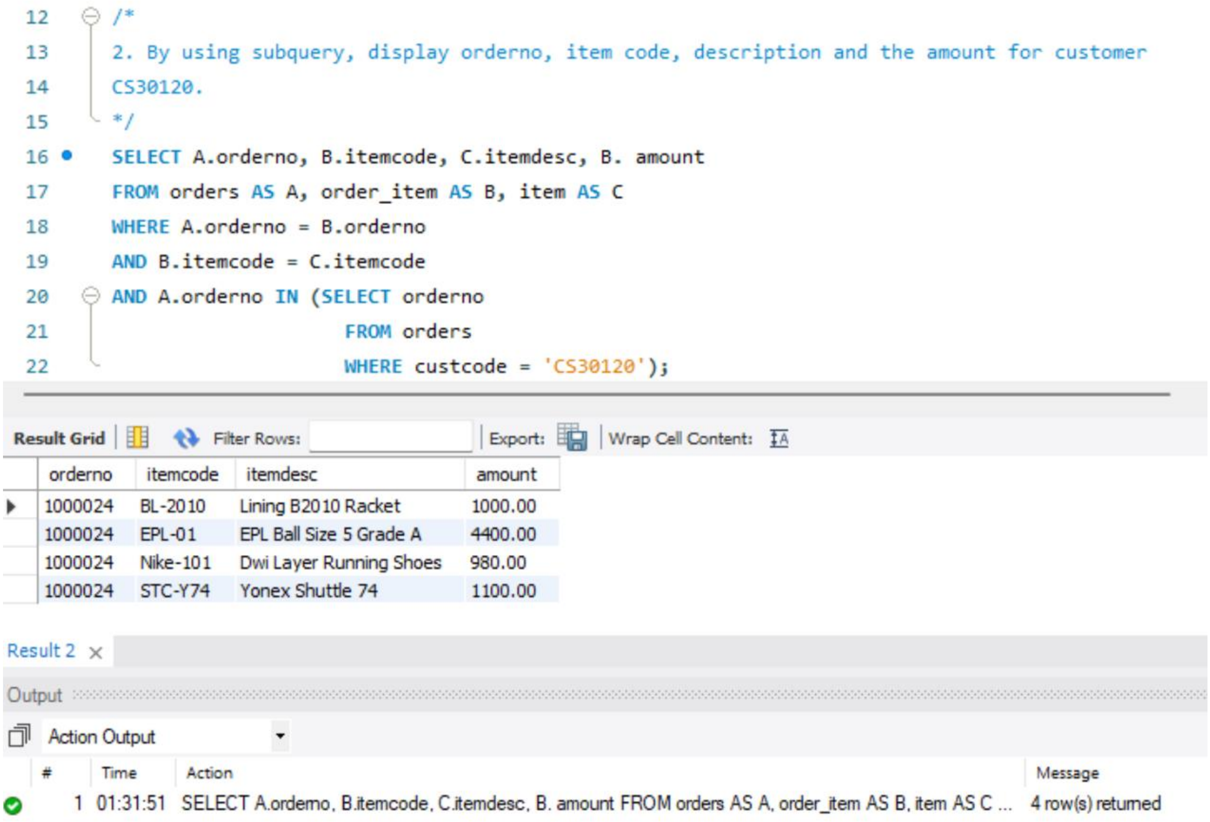
Activity 2



```
/*
1. Find customer name, order number and order amount if order amount is greater than average
sales amount of the customers and sort the record by customer name [Hint: use WHERE
clause].
*/
SELECT A.custname, B.orderno, C. amount AS OrderAmount
FROM customer AS A, orders AS B, order_item AS C
WHERE A.custcode = B.custcode AND B.orderno = C.orderno
AND C.amount > (SELECT AVG(amount) FROM order_item)
ORDER BY A.custname;
```

custname	orderno	OrderAmount
Hakim Adlin	1000021	2000.00
Hakim Adlin	1000021	2200.00
Persatuan Sukan ABC	1000026	2040.00
Persatuan Sukan ABC	1000026	4900.00
Unit Sukan UMT	1000024	4400.00

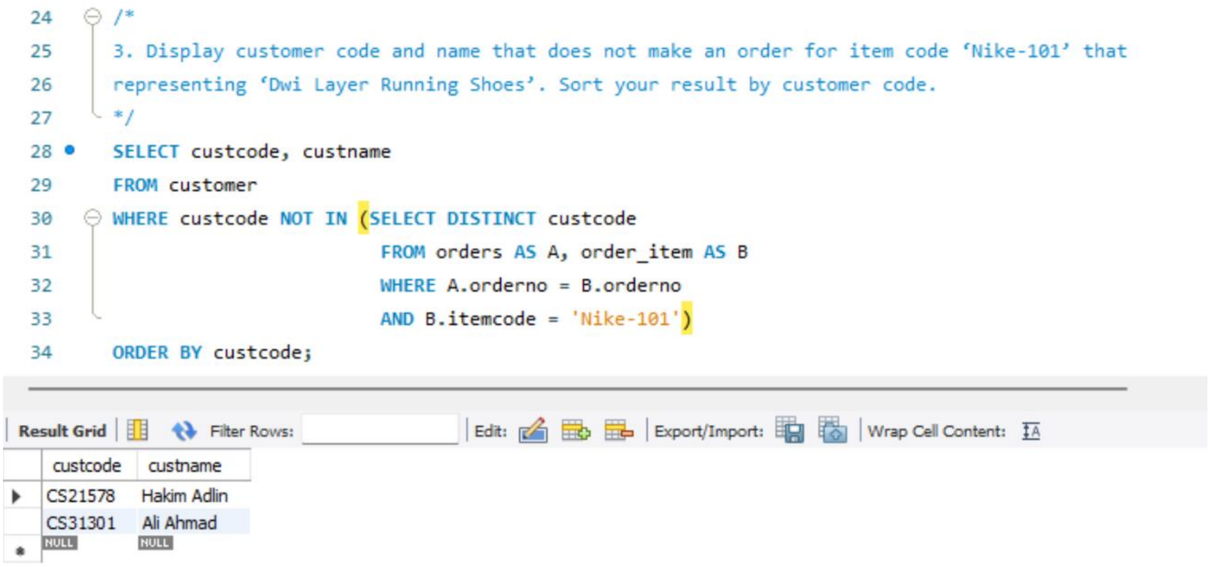
```
#  Time      Action                                                                                Message
1  01:28:19  SELECT A.custname, B.ordemo, C. amount AS OrderAmount FROM customer AS A, orders AS B, order_...  5 row(s) retumed
```



```
/*
2. By using subquery, display orderno, item code, description and the amount for customer
CS30120.
*/
SELECT A.orderno, B.itemcode, C.itemdesc, B. amount
FROM orders AS A, order_item AS B, item AS C
WHERE A.orderno = B.orderno
AND B.itemcode = C.itemcode
AND A.orderno IN (SELECT orderno
                    FROM orders
                    WHERE custcode = 'CS30120');
```

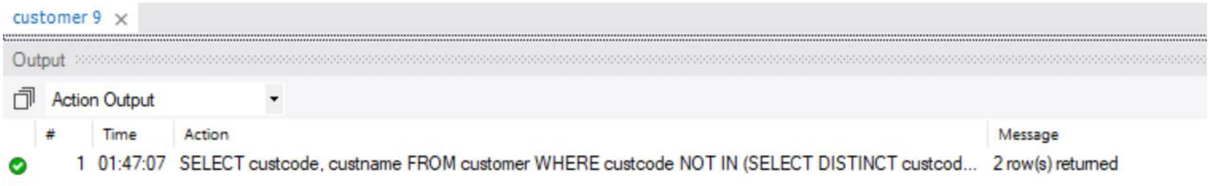
orderno	itemcode	itemdesc	amount
1000024	BL-2010	Lining B2010 Racket	1000.00
1000024	EPL-01	EPL Ball Size 5 Grade A	4400.00
1000024	Nike-101	Dwi Layer Running Shoes	980.00
1000024	STC-Y74	Yonex Shuttle 74	1100.00

```
#  Time      Action                                                                                  Message
1  01:31:51  SELECT A.ordemo, B.itemcode, C.itemdesc, B. amount FROM orders AS A, order_item AS B, item AS C ...  4 row(s) retumed
```

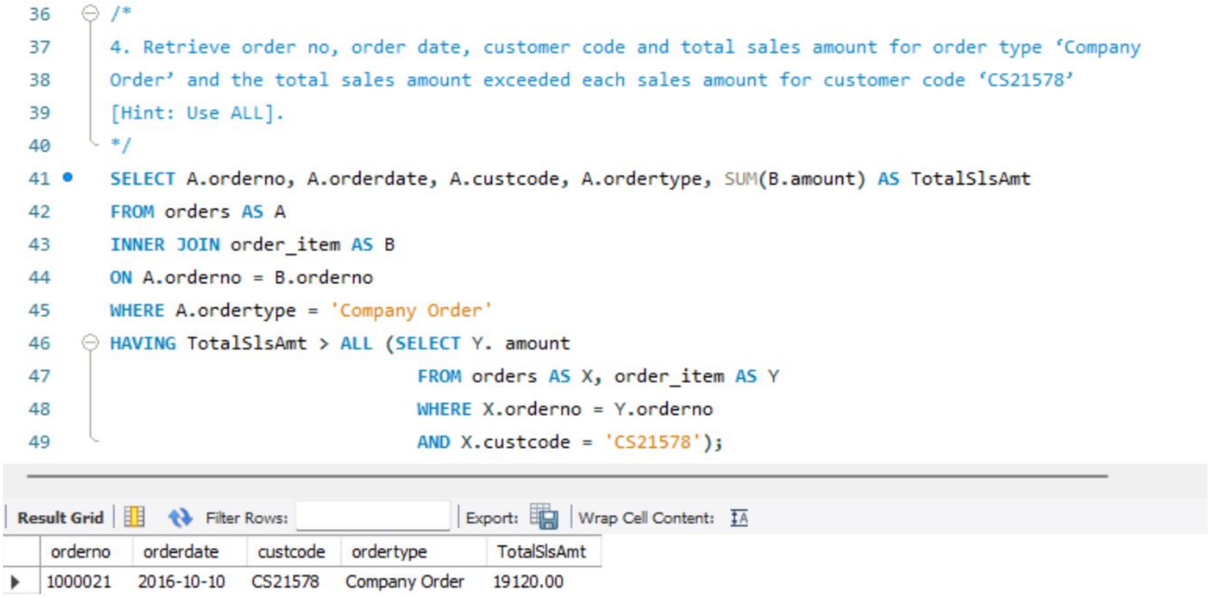


```sql
/*
3. Display customer code and name that does not make an order for item code ‘Nike-101’ that
representing ‘Dwi Layer Running Shoes’. Sort your result by customer code.
*/
SELECT custcode, custname
FROM customer
WHERE custcode NOT IN (SELECT DISTINCT custcode
                       FROM orders AS A, order_item AS B
                       WHERE A.orderno = B.orderno
                       AND B.itemcode = 'Nike-101')
ORDER BY custcode;
```

custcode	custname
CS21578	Hakim Adlin
CS31301	Ali Ahmad
NULL	NULL

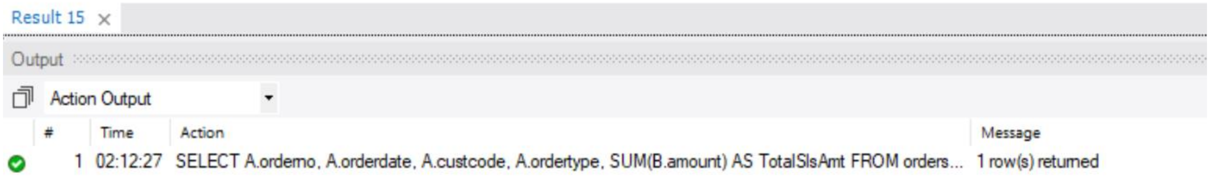


```
#  Time      Action                                                                              Message
1  01:47:07  SELECT custcode, custname FROM customer WHERE custcode NOT IN (SELECT DISTINCT custcod...  2 row(s) returned
```

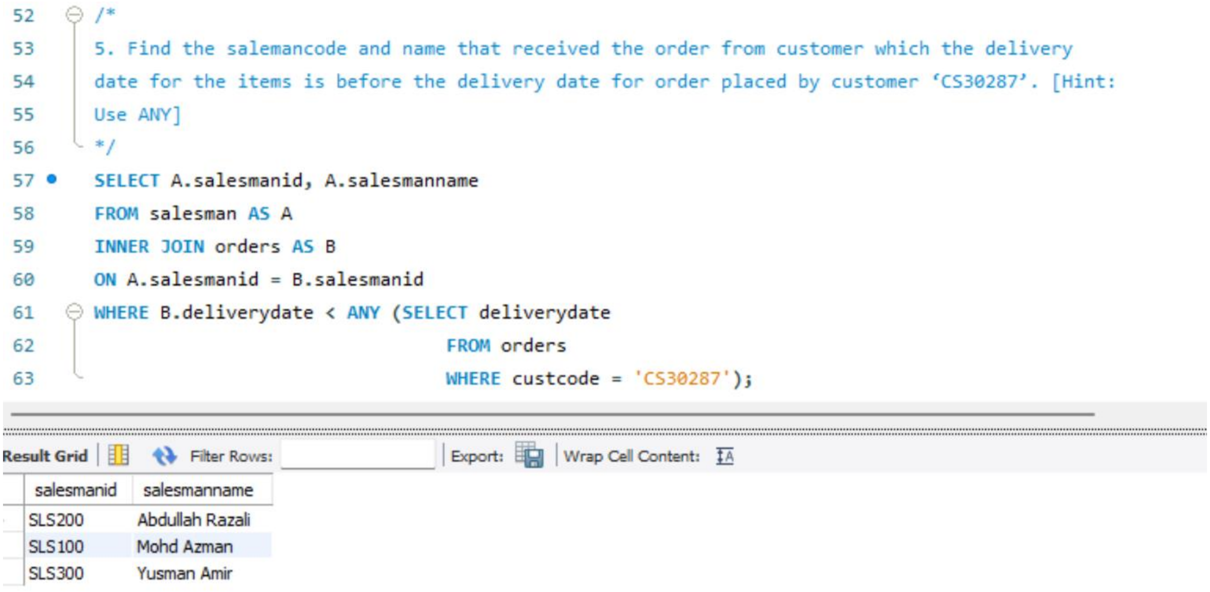


```sql
/*
4. Retrieve order no, order date, customer code and total sales amount for order type ‘Company
Order’ and the total sales amount exceeded each sales amount for customer code ‘CS21578’
[Hint: Use ALL].
*/
SELECT A.orderno, A.orderdate, A.custcode, A.ordertype, SUM(B.amount) AS TotalSlsAmt
FROM orders AS A
INNER JOIN order_item AS B
ON A.orderno = B.orderno
WHERE A.ordertype = 'Company Order'
HAVING TotalSlsAmt > ALL (SELECT Y. amount
                          FROM orders AS X, order_item AS Y
                          WHERE X.orderno = Y.orderno
                          AND X.custcode = 'CS21578');
```

orderno	orderdate	custcode	ordertype	TotalSlsAmt
1000021	2016-10-10	CS21578	Company Order	19120.00

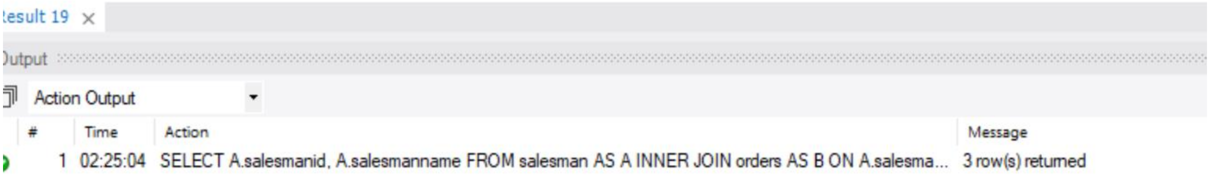


```
#  Time      Action                                                                                   Message
1  02:12:27  SELECT A.orderno, A.orderdate, A.custcode, A.ordertype, SUM(B.amount) AS TotalSlsAmt FROM orders...  1 row(s) returned
```

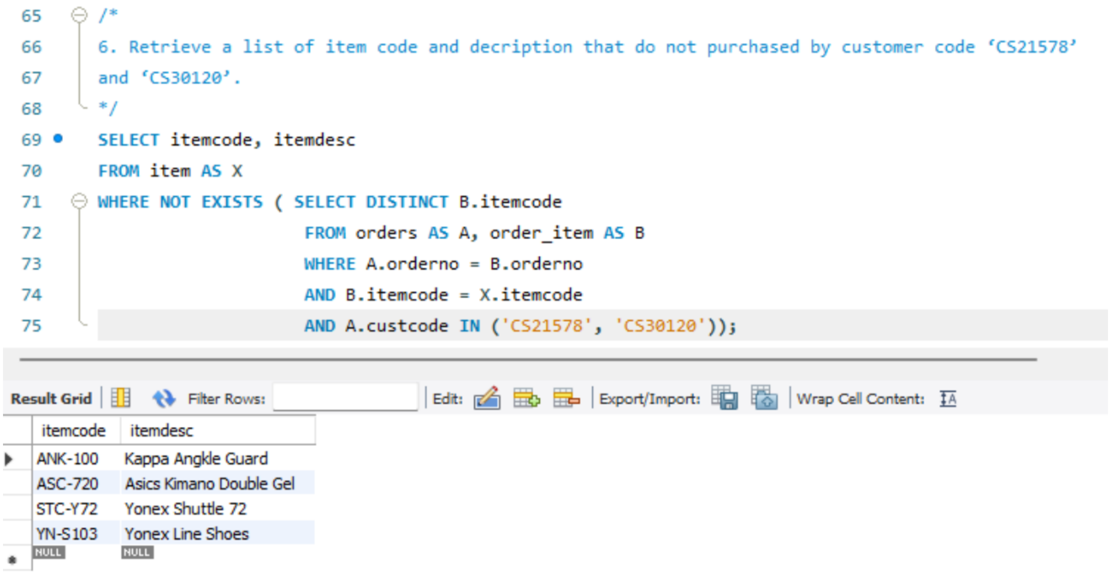


```sql
/*
5. Find the salemancode and name that received the order from customer which the delivery
date for the items is before the delivery date for order placed by customer ‘CS30287’. [Hint:
Use ANY]
*/
SELECT A.salesmanid, A.salesmanname
FROM salesman AS A
INNER JOIN orders AS B
ON A.salesmanid = B.salesmanid
WHERE B.deliverydate < ANY (SELECT deliverydate
                            FROM orders
                            WHERE custcode = 'CS30287');
```

salesmanid	salesmanname
SLS200	Abdullah Razali
SLS100	Mohd Azman
SLS300	Yusman Amir

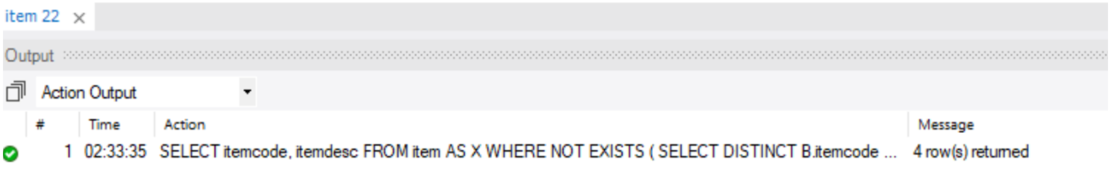


```
#   Time       Action                                                                                    Message
1   02:25:04   SELECT A.salesmanid, A.salesmanname FROM salesman AS A INNER JOIN orders AS B ON A.salesma...   3 row(s) returned
```



```sql
/*
6. Retrieve a list of item code and decription that do not purchased by customer code ‘CS21578’
and ‘CS30120’.
*/
SELECT itemcode, itemdesc
FROM item AS X
WHERE NOT EXISTS ( SELECT DISTINCT B.itemcode
                   FROM orders AS A, order_item AS B
                   WHERE A.orderno = B.orderno
                   AND B.itemcode = X.itemcode
                   AND A.custcode IN ('CS21578', 'CS30120'));
```

itemcode	itemdesc
ANK-100	Kappa Angkle Guard
ASC-720	Asics Kimano Double Gel
STC-Y72	Yonex Shuttle 72
YN-S103	Yonex Line Shoes
NULL	NULL

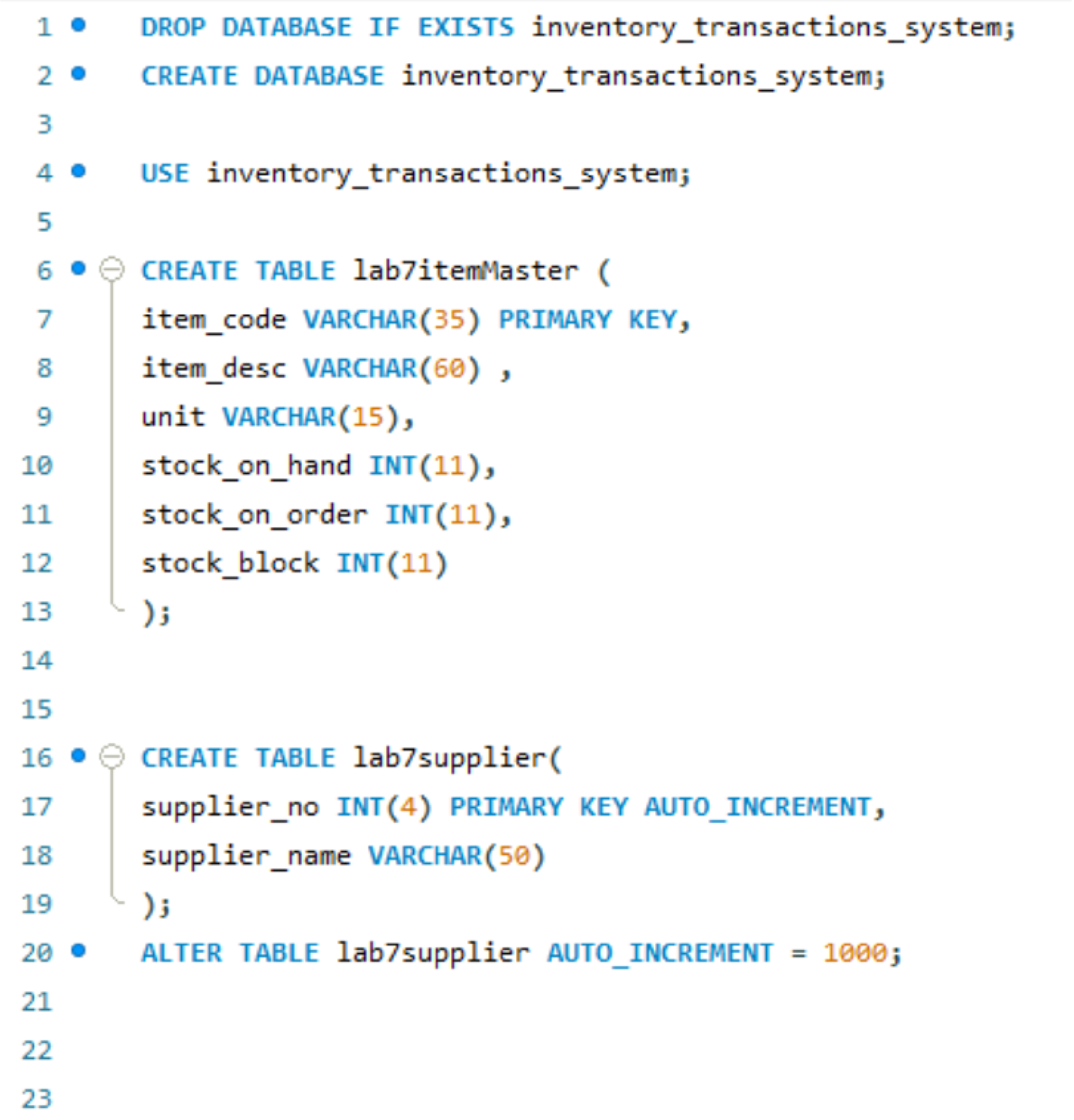


```
#   Time       Action                                                                        Message
1   02:33:35   SELECT itemcode, itemdesc FROM item AS X WHERE NOT EXISTS ( SELECT DISTINCT B.itemcode ...   4 row(s) returned
```

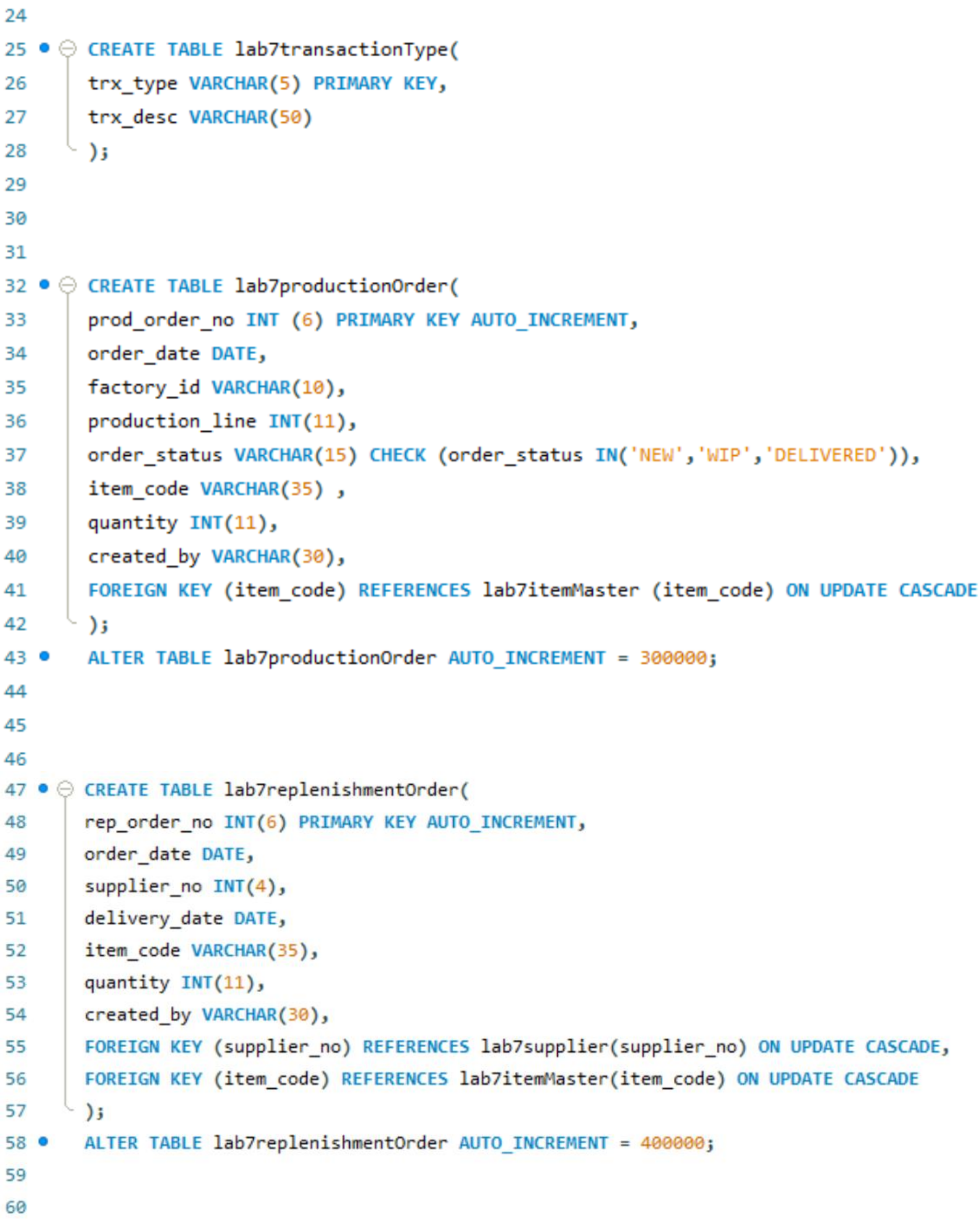
Lab Exercise

Task 1

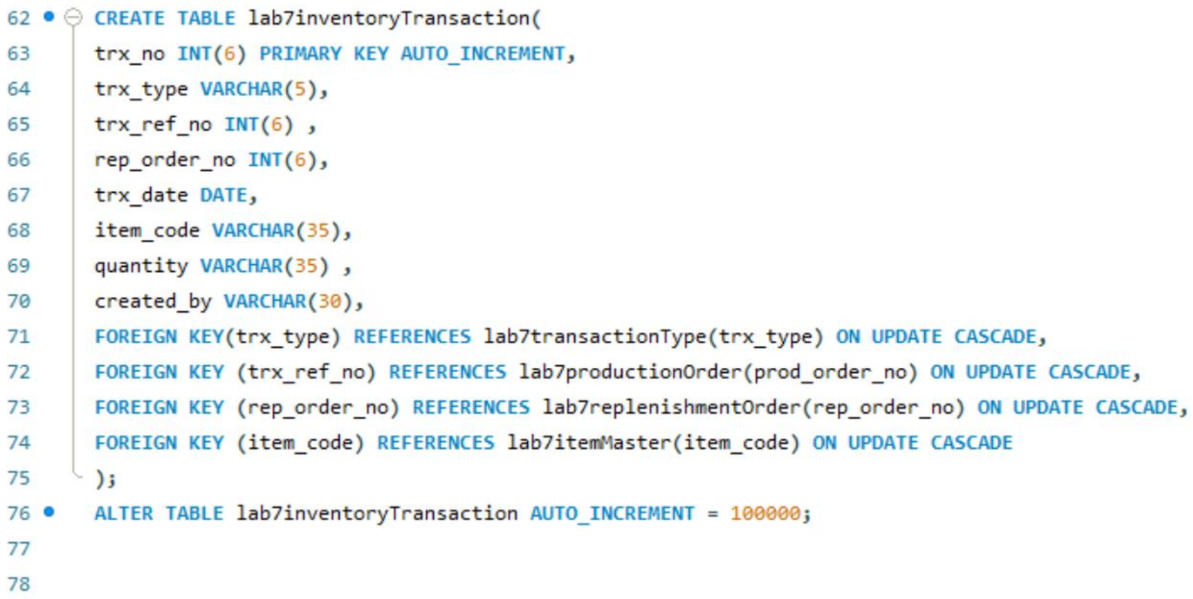
Creating the tables



```sql
DROP DATABASE IF EXISTS inventory_transactions_system;
CREATE DATABASE inventory_transactions_system;

USE inventory_transactions_system;

CREATE TABLE lab7itemMaster (
item_code VARCHAR(35) PRIMARY KEY,
item_desc VARCHAR(60) ,
unit VARCHAR(15),
stock_on_hand INT(11),
stock_on_order INT(11),
stock_block INT(11)
);


CREATE TABLE lab7supplier(
supplier_no INT(4) PRIMARY KEY AUTO_INCREMENT,
supplier_name VARCHAR(50)
);
ALTER TABLE lab7supplier AUTO_INCREMENT = 1000;
```

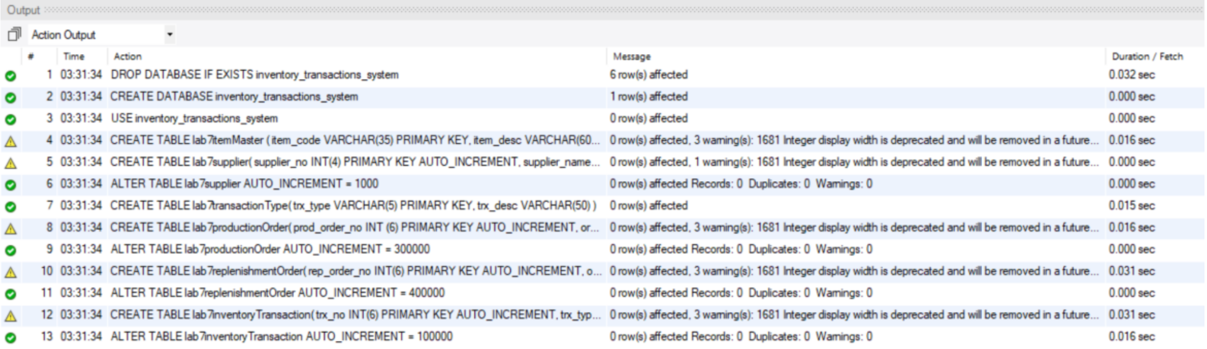


```sql

CREATE TABLE lab7transactionType(
trx_type VARCHAR(5) PRIMARY KEY,
trx_desc VARCHAR(50)
);



CREATE TABLE lab7productionOrder(
prod_order_no INT (6) PRIMARY KEY AUTO_INCREMENT,
order_date DATE,
factory_id VARCHAR(10),
production_line INT(11),
order_status VARCHAR(15) CHECK (order_status IN('NEW','WIP','DELIVERED')),
item_code VARCHAR(35) ,
quantity INT(11),
created_by VARCHAR(30),
FOREIGN KEY (item_code) REFERENCES lab7itemMaster (item_code) ON UPDATE CASCADE
);
ALTER TABLE lab7productionOrder AUTO_INCREMENT = 300000;



CREATE TABLE lab7replenishmentOrder(
rep_order_no INT(6) PRIMARY KEY AUTO_INCREMENT,
order_date DATE,
supplier_no INT(4),
delivery_date DATE,
item_code VARCHAR(35),
quantity INT(11),
created_by VARCHAR(30),
FOREIGN KEY (supplier_no) REFERENCES lab7supplier(supplier_no) ON UPDATE CASCADE,
FOREIGN KEY (item_code) REFERENCES lab7itemMaster(item_code) ON UPDATE CASCADE
);
ALTER TABLE lab7replenishmentOrder AUTO_INCREMENT = 400000;


```



```sql
CREATE TABLE lab7inventoryTransaction(
trx_no INT(6) PRIMARY KEY AUTO_INCREMENT,
trx_type VARCHAR(5),
trx_ref_no INT(6) ,
rep_order_no INT(6),
trx_date DATE,
item_code VARCHAR(35),
quantity VARCHAR(35) ,
created_by VARCHAR(30),
FOREIGN KEY(trx_type) REFERENCES lab7transactionType(trx_type) ON UPDATE CASCADE,
FOREIGN KEY (trx_ref_no) REFERENCES lab7productionOrder(prod_order_no) ON UPDATE CASCADE,
FOREIGN KEY (rep_order_no) REFERENCES lab7replenishmentOrder(rep_order_no) ON UPDATE CASCADE,
FOREIGN KEY (item_code) REFERENCES lab7itemMaster(item_code) ON UPDATE CASCADE
);
ALTER TABLE lab7inventoryTransaction AUTO_INCREMENT = 100000;
```

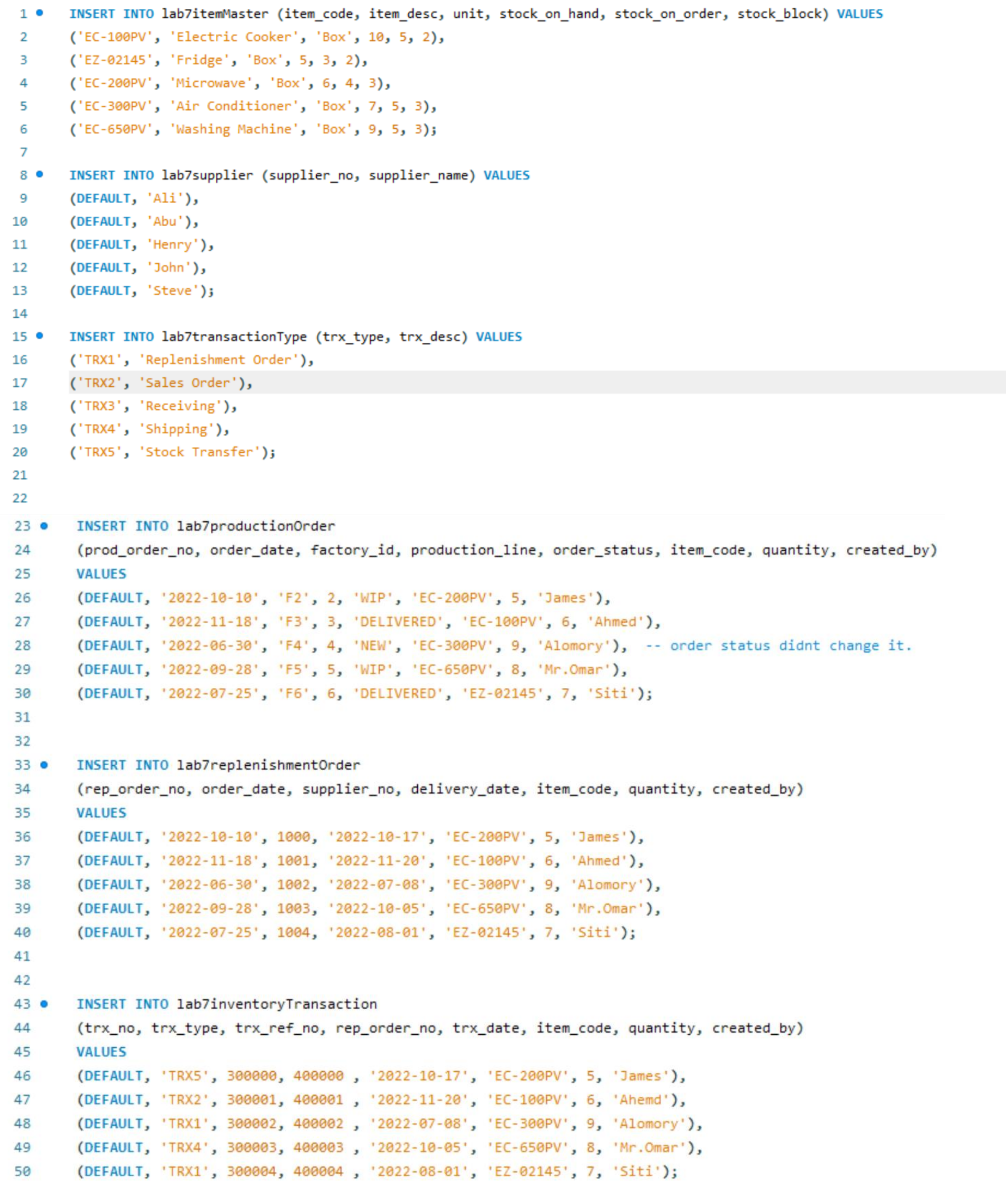
Output



Output

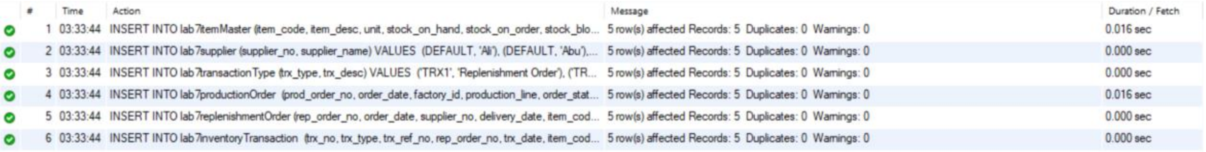
	#	Time	Action	Message	Duration / Fetch
✓	1	03:31:34	DROP DATABASE IF EXISTS inventory_transactions_system	6 row(s) affected	0.032 sec
✓	2	03:31:34	CREATE DATABASE inventory_transactions_system	1 row(s) affected	0.000 sec
✓	3	03:31:34	USE inventory_transactions_system	0 row(s) affected	0.000 sec
⚠	4	03:31:34	CREATE TABLE lab7itemMaster (item_code VARCHAR(35) PRIMARY KEY, item_desc VARCHAR(60...	0 row(s) affected, 3 warning(s): 1681 Integer display width is deprecated and will be removed in a future...	0.016 sec
⚠	5	03:31:34	CREATE TABLE lab7supplier(supplier_no INT(4) PRIMARY KEY AUTO_INCREMENT, supplier_name...	0 row(s) affected, 1 warning(s): 1681 Integer display width is deprecated and will be removed in a future...	0.000 sec
✓	6	03:31:34	ALTER TABLE lab7supplier AUTO_INCREMENT = 1000	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.000 sec
✓	7	03:31:34	CREATE TABLE lab7transactionType(trx_type VARCHAR(5) PRIMARY KEY, trx_desc VARCHAR(50))	0 row(s) affected	0.015 sec
⚠	8	03:31:34	CREATE TABLE lab7productionOrder(prod_order_no INT (6) PRIMARY KEY AUTO_INCREMENT, or...	0 row(s) affected, 3 warning(s): 1681 Integer display width is deprecated and will be removed in a future...	0.016 sec
✓	9	03:31:34	ALTER TABLE lab7productionOrder AUTO_INCREMENT = 300000	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.000 sec
⚠	10	03:31:34	CREATE TABLE lab7replenishmentOrder(rep_order_no INT(6) PRIMARY KEY AUTO_INCREMENT, o...	0 row(s) affected, 3 warning(s): 1681 Integer display width is deprecated and will be removed in a future...	0.031 sec
✓	11	03:31:34	ALTER TABLE lab7replenishmentOrder AUTO_INCREMENT = 400000	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.000 sec
⚠	12	03:31:34	CREATE TABLE lab7inventoryTransaction(trx_no INT(6) PRIMARY KEY AUTO_INCREMENT, trx_typ...	0 row(s) affected, 3 warning(s): 1681 Integer display width is deprecated and will be removed in a future...	0.031 sec
✓	13	03:31:34	ALTER TABLE lab7inventoryTransaction AUTO_INCREMENT = 100000	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.016 sec

Task 2



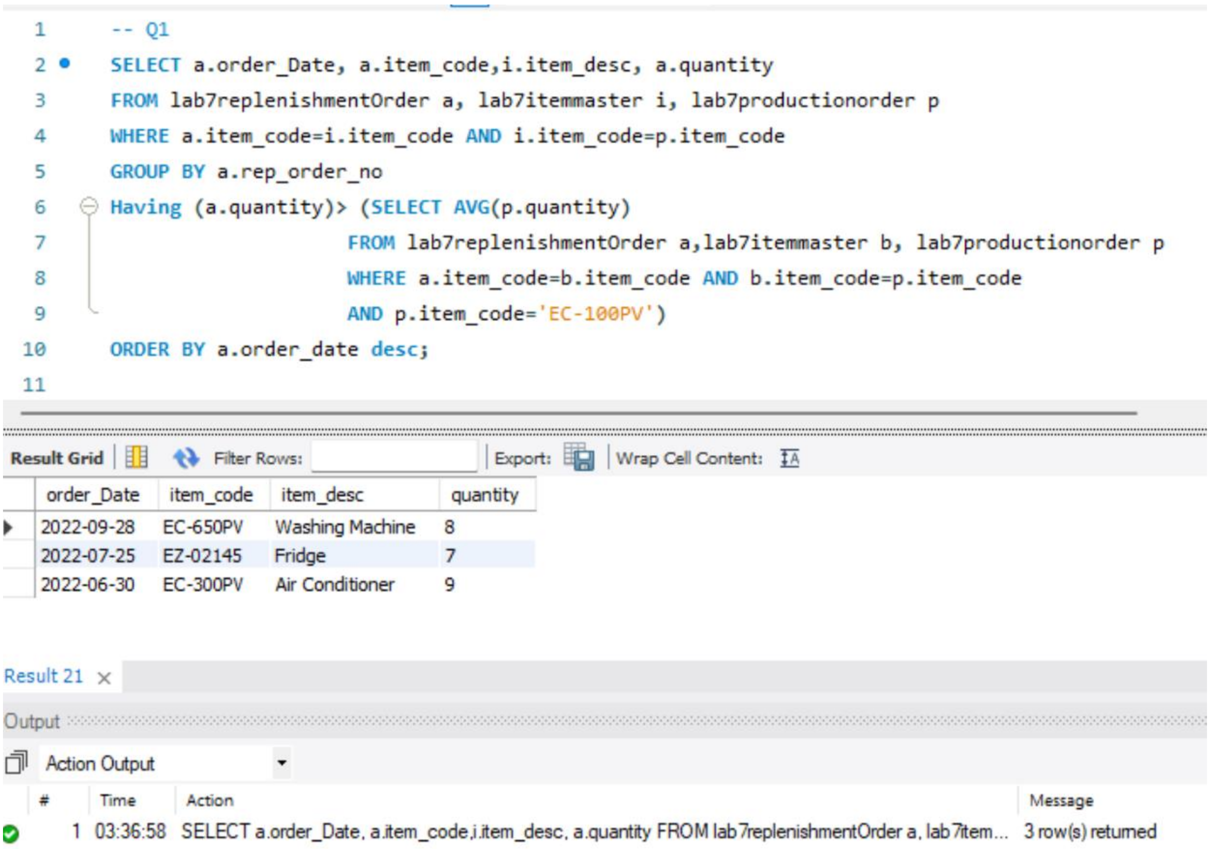
```sql
INSERT INTO lab7itemMaster (item_code, item_desc, unit, stock_on_hand, stock_on_order, stock_block) VALUES
('EC-100PV', 'Electric Cooker', 'Box', 10, 5, 2),
('EZ-02145', 'Fridge', 'Box', 5, 3, 2),
('EC-200PV', 'Microwave', 'Box', 6, 4, 3),
('EC-300PV', 'Air Conditioner', 'Box', 7, 5, 3),
('EC-650PV', 'Washing Machine', 'Box', 9, 5, 3);

INSERT INTO lab7supplier (supplier_no, supplier_name) VALUES
(DEFAULT, 'Ali'),
(DEFAULT, 'Abu'),
(DEFAULT, 'Henry'),
(DEFAULT, 'John'),
(DEFAULT, 'Steve');

INSERT INTO lab7transactionType (trx_type, trx_desc) VALUES
('TRX1', 'Replenishment Order'),
('TRX2', 'Sales Order'),
('TRX3', 'Receiving'),
('TRX4', 'Shipping'),
('TRX5', 'Stock Transfer');


INSERT INTO lab7productionOrder
(prod_order_no, order_date, factory_id, production_line, order_status, item_code, quantity, created_by)
VALUES
(DEFAULT, '2022-10-10', 'F2', 2, 'WIP', 'EC-200PV', 5, 'James'),
(DEFAULT, '2022-11-18', 'F3', 3, 'DELIVERED', 'EC-100PV', 6, 'Ahmed'),
(DEFAULT, '2022-06-30', 'F4', 4, 'NEW', 'EC-300PV', 9, 'Alomory'),  -- order status didnt change it.
(DEFAULT, '2022-09-28', 'F5', 5, 'WIP', 'EC-650PV', 8, 'Mr.Omar'),
(DEFAULT, '2022-07-25', 'F6', 6, 'DELIVERED', 'EZ-02145', 7, 'Siti');


INSERT INTO lab7replenishmentOrder
(rep_order_no, order_date, supplier_no, delivery_date, item_code, quantity, created_by)
VALUES
(DEFAULT, '2022-10-10', 1000, '2022-10-17', 'EC-200PV', 5, 'James'),
(DEFAULT, '2022-11-18', 1001, '2022-11-20', 'EC-100PV', 6, 'Ahmed'),
(DEFAULT, '2022-06-30', 1002, '2022-07-08', 'EC-300PV', 9, 'Alomory'),
(DEFAULT, '2022-09-28', 1003, '2022-10-05', 'EC-650PV', 8, 'Mr.Omar'),
(DEFAULT, '2022-07-25', 1004, '2022-08-01', 'EZ-02145', 7, 'Siti');


INSERT INTO lab7inventoryTransaction
(trx_no, trx_type, trx_ref_no, rep_order_no, trx_date, item_code, quantity, created_by)
VALUES
(DEFAULT, 'TRX5', 300000, 400000 , '2022-10-17', 'EC-200PV', 5, 'James'),
(DEFAULT, 'TRX2', 300001, 400001 , '2022-11-20', 'EC-100PV', 6, 'Ahemd'),
(DEFAULT, 'TRX1', 300002, 400002 , '2022-07-08', 'EC-300PV', 9, 'Alomory'),
(DEFAULT, 'TRX4', 300003, 400003 , '2022-10-05', 'EC-650PV', 8, 'Mr.Omar'),
(DEFAULT, 'TRX1', 300004, 400004 , '2022-08-01', 'EZ-02145', 7, 'Siti');
```

Output



	#	Time	Action	Message	Duration / Fetch
✓	1	03:33:44	INSERT INTO lab7itemMaster (item_code, item_desc, unit, stock_on_hand, stock_on_order, stock_blo...	5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0	0.016 sec
✓	2	03:33:44	INSERT INTO lab7supplier (supplier_no, supplier_name) VALUES (DEFAULT, 'Ali'), (DEFAULT, 'Abu'),...	5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0	0.000 sec
✓	3	03:33:44	INSERT INTO lab7transactionType (trx_type, trx_desc) VALUES ('TRX1', 'Replenishment Order'), ('TR...	5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0	0.000 sec
✓	4	03:33:44	INSERT INTO lab7productionOrder (prod_order_no, order_date, factory_id, production_line, order_stat...	5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0	0.016 sec
✓	5	03:33:44	INSERT INTO lab7replenishmentOrder (rep_order_no, order_date, supplier_no, delivery_date, item_cod...	5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0	0.000 sec
✓	6	03:33:44	INSERT INTO lab7inventoryTransaction (trx_no, trx_type, trx_ref_no, rep_order_no, trx_date, item_cod...	5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0	0.000 sec

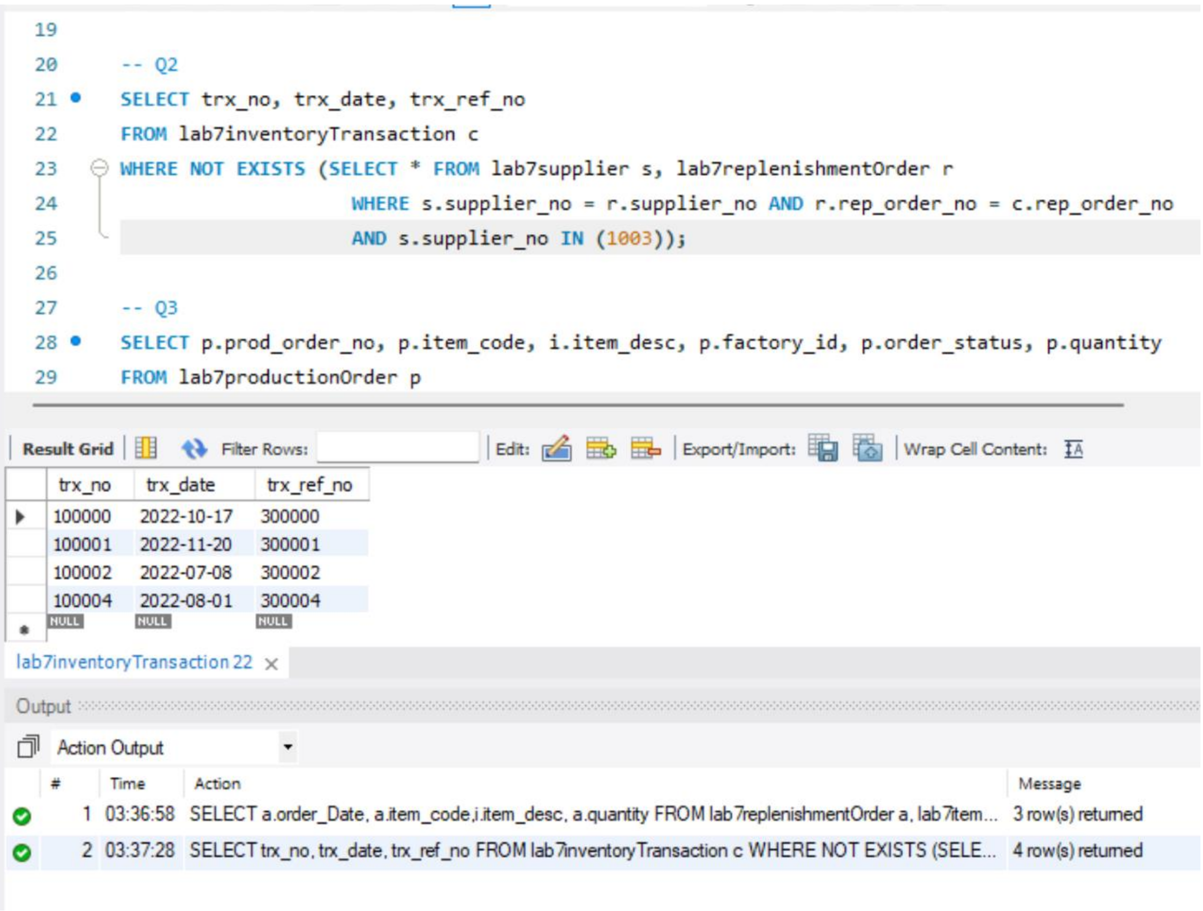
Task 3



```sql
-- Q1
SELECT a.order_Date, a.item_code,i.item_desc, a.quantity
FROM lab7replenishmentOrder a, lab7itemmaster i, lab7productionorder p
WHERE a.item_code=i.item_code AND i.item_code=p.item_code
GROUP BY a.rep_order_no
Having (a.quantity)> (SELECT AVG(p.quantity)
                        FROM lab7replenishmentOrder a,lab7itemmaster b, lab7productionorder p
                        WHERE a.item_code=b.item_code AND b.item_code=p.item_code
                        AND p.item_code='EC-100PV')
ORDER BY a.order_date desc;
```

order_Date	item_code	item_desc	quantity
2022-09-28	EC-650PV	Washing Machine	8
2022-07-25	EZ-02145	Fridge	7
2022-06-30	EC-300PV	Air Conditioner	9

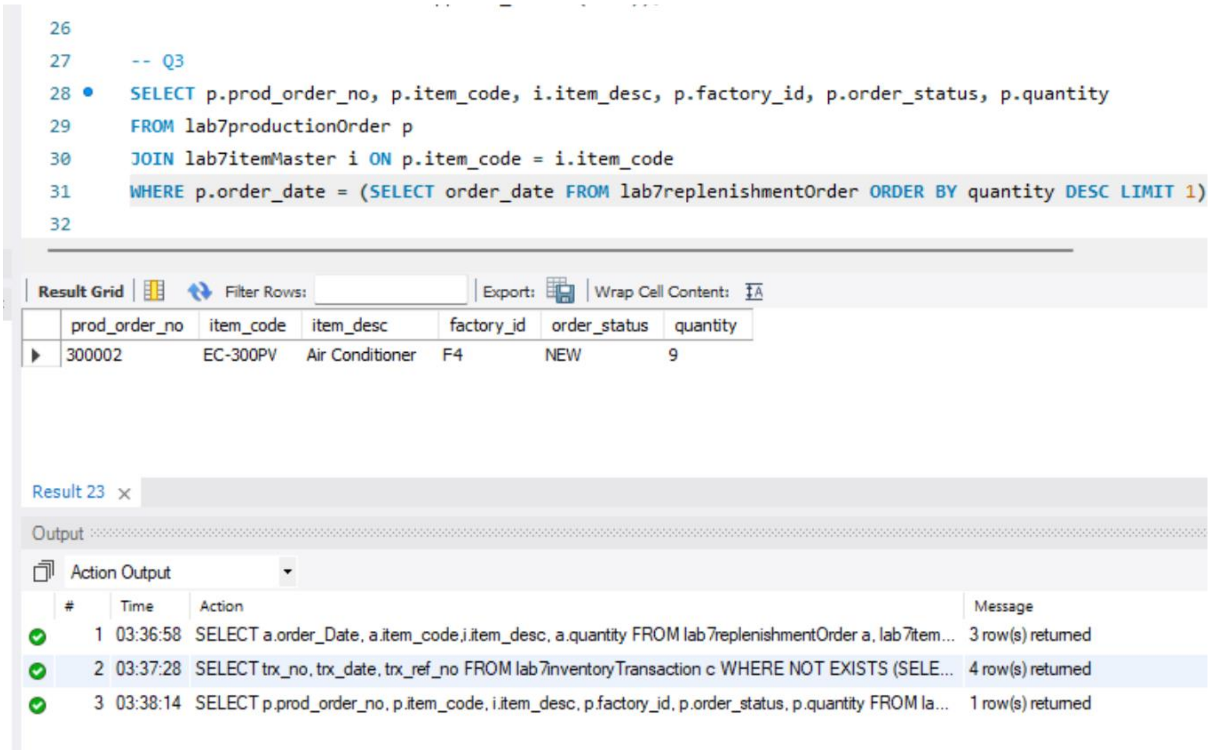
#	Time	Action	Message
1	03:36:58	SELECT a.order_Date, a.item_code,i.item_desc, a.quantity FROM lab7replenishmentOrder a, lab7item...	3 row(s) returned



```sql

-- Q2
SELECT trx_no, trx_date, trx_ref_no
FROM lab7inventoryTransaction c
WHERE NOT EXISTS (SELECT * FROM lab7supplier s, lab7replenishmentOrder r
                        WHERE s.supplier_no = r.supplier_no AND r.rep_order_no = c.rep_order_no
                        AND s.supplier_no IN (1003));

-- Q3
SELECT p.prod_order_no, p.item_code, i.item_desc, p.factory_id, p.order_status, p.quantity
FROM lab7productionOrder p
```

trx_no	trx_date	trx_ref_no
100000	2022-10-17	300000
100001	2022-11-20	300001
100002	2022-07-08	300002
100004	2022-08-01	300004
NULL	NULL	NULL

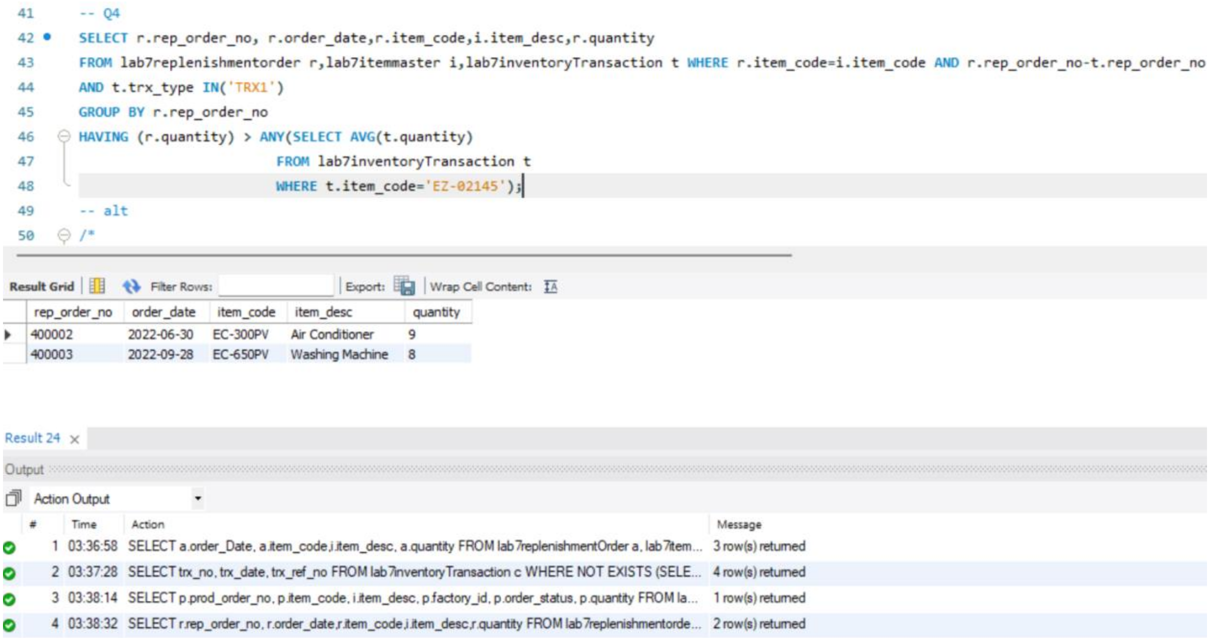
#	Time	Action	Message
1	03:36:58	SELECT a.order_Date, a.item_code,i.item_desc, a.quantity FROM lab7replenishmentOrder a, lab7item...	3 row(s) returned
2	03:37:28	SELECT trx_no, trx_date, trx_ref_no FROM lab7inventoryTransaction c WHERE NOT EXISTS (SELE...	4 row(s) returned



```sql
-- Q3
SELECT p.prod_order_no, p.item_code, i.item_desc, p.factory_id, p.order_status, p.quantity
FROM lab7productionOrder p
JOIN lab7itemMaster i ON p.item_code = i.item_code
WHERE p.order_date = (SELECT order_date FROM lab7replenishmentOrder ORDER BY quantity DESC LIMIT 1)
```

	prod_order_no	item_code	item_desc	factory_id	order_status	quantity
▶	300002	EC-300PV	Air Conditioner	F4	NEW	9

```
#  Time      Action                                                                                        Message
1  03:36:58  SELECT a.order_Date, a.item_code,i.item_desc, a.quantity FROM lab7replenishmentOrder a, lab7item...  3 row(s) returned
2  03:37:28  SELECT trx_no, trx_date, trx_ref_no FROM lab7inventoryTransaction c WHERE NOT EXISTS (SELE...        4 row(s) returned
3  03:38:14  SELECT p.prod_order_no, p.item_code, i.item_desc, p.factory_id, p.order_status, p.quantity FROM la...  1 row(s) returned
```

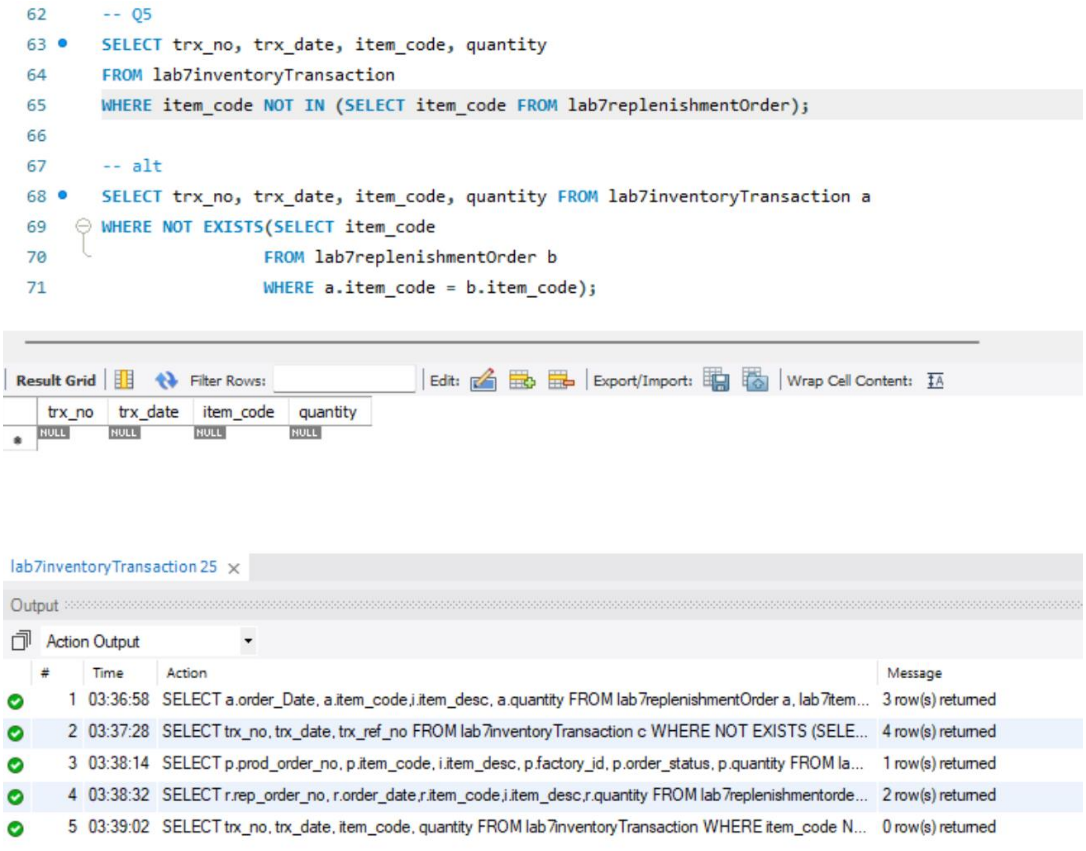


```sql
-- Q4
SELECT r.rep_order_no, r.order_date,r.item_code,i.item_desc,r.quantity
FROM lab7replenishmentorder r,lab7itemmaster i,lab7inventoryTransaction t WHERE r.item_code=i.item_code AND r.rep_order_no-t.rep_order_no
AND t.trx_type IN('TRX1')
GROUP BY r.rep_order_no
HAVING (r.quantity) > ANY(SELECT AVG(t.quantity)
                        FROM lab7inventoryTransaction t
                        WHERE t.item_code='EZ-02145');
-- alt
/*
```

	rep_order_no	order_date	item_code	item_desc	quantity
▶	400002	2022-06-30	EC-300PV	Air Conditioner	9
	400003	2022-09-28	EC-650PV	Washing Machine	8

```
#  Time      Action                                                                                        Message
1  03:36:58  SELECT a.order_Date, a.item_code,i.item_desc, a.quantity FROM lab7replenishmentOrder a, lab7item...  3 row(s) returned
2  03:37:28  SELECT trx_no, trx_date, trx_ref_no FROM lab7inventoryTransaction c WHERE NOT EXISTS (SELE...        4 row(s) returned
3  03:38:14  SELECT p.prod_order_no, p.item_code, i.item_desc, p.factory_id, p.order_status, p.quantity FROM la...  1 row(s) returned
4  03:38:32  SELECT r.rep_order_no, r.order_date,r.item_code,i.item_desc,r.quantity FROM lab7replenishmentorde...  2 row(s) returned
```

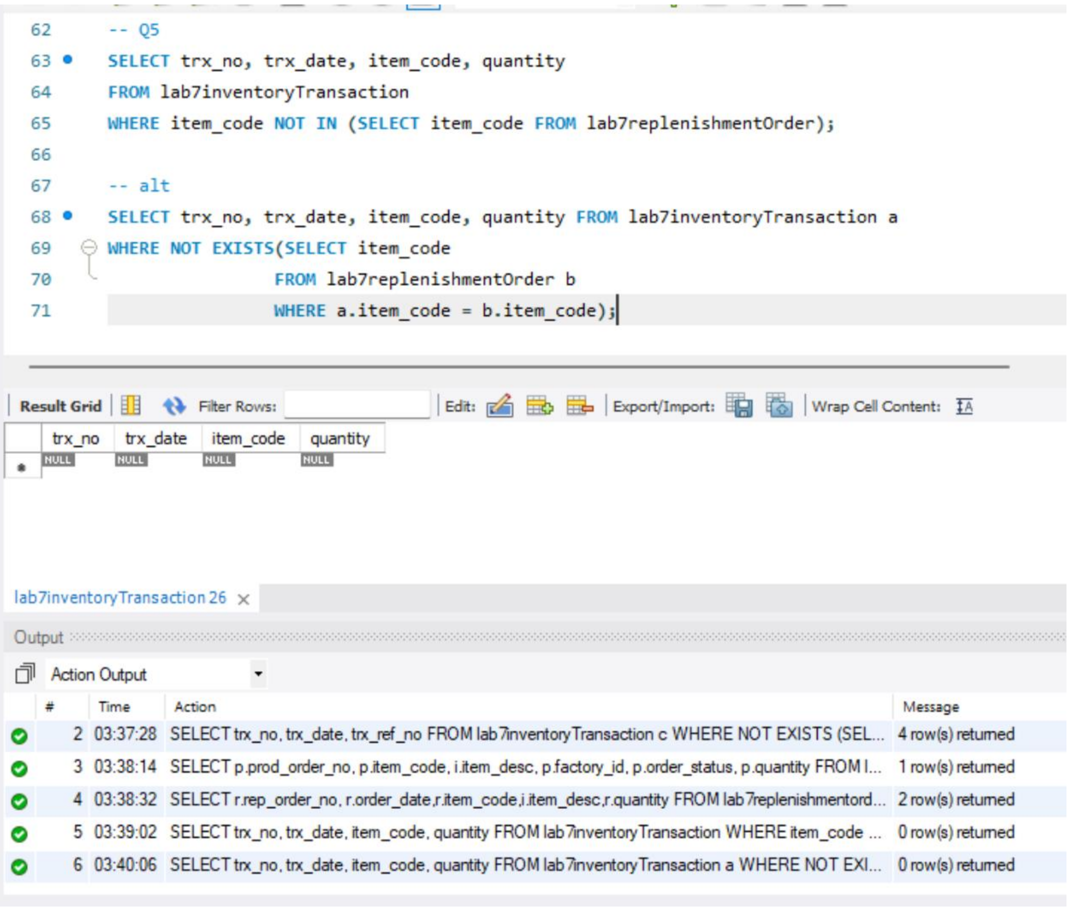
For the last question I could not find result(empty record).



```
-- Q5
SELECT trx_no, trx_date, item_code, quantity
FROM lab7inventoryTransaction
WHERE item_code NOT IN (SELECT item_code FROM lab7replenishmentOrder);

-- alt
SELECT trx_no, trx_date, item_code, quantity FROM lab7inventoryTransaction a
WHERE NOT EXISTS(SELECT item_code
                 FROM lab7replenishmentOrder b
                 WHERE a.item_code = b.item_code);
```

	trx_no	trx_date	item_code	quantity
*	NULL	NULL	NULL	NULL

#	Time	Action	Message
1	03:36:58	SELECT a.order_Date, a.item_code,i.item_desc, a.quantity FROM lab7replenishmentOrder a, lab7item...	3 row(s) returned
2	03:37:28	SELECT trx_no, trx_date, trx_ref_no FROM lab7inventoryTransaction c WHERE NOT EXISTS (SELE...	4 row(s) returned
3	03:38:14	SELECT p.prod_order_no, p.item_code, i.item_desc, p.factory_id, p.order_status, p.quantity FROM la...	1 row(s) returned
4	03:38:32	SELECT r.rep_order_no, r.order_date,r.item_code,i.item_desc,r.quantity FROM lab7replenishmentorde...	2 row(s) returned
5	03:39:02	SELECT trx_no, trx_date, item_code, quantity FROM lab7inventoryTransaction WHERE item_code N...	0 row(s) returned



```
-- Q5
SELECT trx_no, trx_date, item_code, quantity
FROM lab7inventoryTransaction
WHERE item_code NOT IN (SELECT item_code FROM lab7replenishmentOrder);

-- alt
SELECT trx_no, trx_date, item_code, quantity FROM lab7inventoryTransaction a
WHERE NOT EXISTS(SELECT item_code
                 FROM lab7replenishmentOrder b
                 WHERE a.item_code = b.item_code);
```

	trx_no	trx_date	item_code	quantity
*	NULL	NULL	NULL	NULL

#	Time	Action	Message
2	03:37:28	SELECT trx_no, trx_date, trx_ref_no FROM lab7inventoryTransaction c WHERE NOT EXISTS (SEL...	4 row(s) returned
3	03:38:14	SELECT p.prod_order_no, p.item_code, i.item_desc, p.factory_id, p.order_status, p.quantity FROM l...	1 row(s) returned
4	03:38:32	SELECT r.rep_order_no, r.order_date,r.item_code,i.item_desc,r.quantity FROM lab7replenishmentord...	2 row(s) returned
5	03:39:02	SELECT trx_no, trx_date, item_code, quantity FROM lab7inventoryTransaction WHERE item_code ...	0 row(s) returned
6	03:40:06	SELECT trx_no, trx_date, item_code, quantity FROM lab7inventoryTransaction a WHERE NOT EXI...	0 row(s) returned