Manahil Alqarni

Nelly Sanchez

Edgar Morales

Isaac Nuno

The Plug Electronic Store

1)Explain business rules, assumptions and constraints in detail.

In our store, we have many employees that we need to communicate with so we need to keep track of their first and last name, address, phone number, and email. Our store stocks many electronics and we need to keep track of our inventory/products such as the price, qty, and description of the item. During operating hours, many customers will make purchases in store or online. If a purchase was made in store, we would like to keep track of the purchase as an order that includes items bought, qty, price of item, date purchased and when possible, the customer who made the purchase. If an order was made online, we'd like to keep track of items bought, qty, date purchased, shipping cost, shipment tracking number, the customer who made the online purchase and the employee who filled the order. To keep track of customers we'd like to have their first and last name, phone number, email, and address. A customer's address, name and email are required for online purchases. We also offer our discounts to customers who sign up for our membership program so we would like to keep track of the members first and last name, address, phone number and email. Memberships may have one or more members.

2)Explain your entities and their relationships based on the business rules for each table in detail. (Example: entity A has "one-to-many" relationship to entity B, why?)

Entities: Employees, products, orders, customer, members

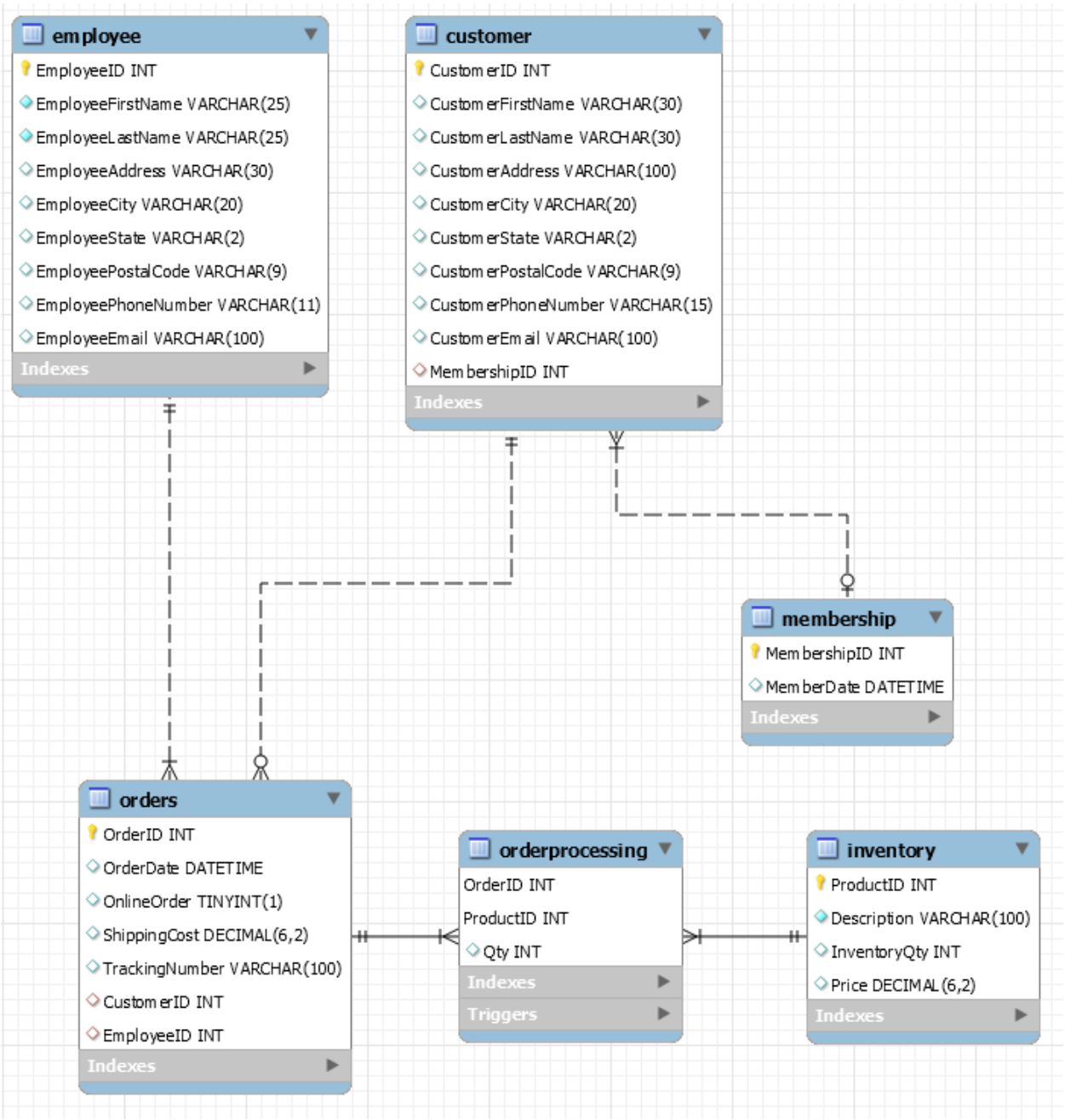
order (M:N) inventory since a single customer may purchase many different items in one transaction. One order may contain many products, One product could belong to many orders

customer(1:M) order since a customer may place one or more orders in store or online

membership (1: M) customers since a membership may have one or more members

employee (1:M) order since employees fill online orders and ring up customers in store.

3) ERD:


employee
EmployeeID INT
EmployeeFirstName VARCHAR(25)
EmployeeLastName VARCHAR(25)
EmployeeAddress VARCHAR(30)
EmployeeCity VARCHAR(20)
EmployeeState VARCHAR(2)
EmployeePostalCode VARCHAR(9)
EmployeePhoneNumber VARCHAR(11)
EmployeeEmail VARCHAR(100)
Indexes
customer
CustomerID INT
CustomerFirstName VARCHAR(30)
CustomerLastName VARCHAR(30)
CustomerAddress VARCHAR(100)
CustomerCity VARCHAR(20)
CustomerState VARCHAR(2)
CustomerPostalCode VARCHAR(9)
CustomerPhoneNumber VARCHAR(15)
CustomerEmail VARCHAR(100)
MembershipID INT
Indexes
membership
MembershipID INT
MemberDate DATETIME
Indexes
orders
OrderID INT
OrderDate DATETIME
OnlineOrder TINYINT(1)
ShippingCost DECIMAL(6,2)
TrackingNumber VARCHAR(100)
CustomerID INT
EmployeeID INT
Indexes
orderprocessing
OrderID INT
ProductID INT
Qty INT
Indexes
Triggers
inventory
ProductID INT
Description VARCHAR(100)
InventoryQty INT
Price DECIMAL(6,2)
Indexes

4) Tables:

Customer Table

```
1   CREATE TABLE customer
2   (CustomerID INT NOT NULL AUTO_INCREMENT,
3   CustomerFirstName VARCHAR(30),
4   CustomerLastName VARCHAR(30),
5   CustomerAddress VARCHAR(100),
6   CustomerCity VARCHAR(20),
7   CustomerState VARCHAR(2),
8   CustomerPostalCode VARCHAR(9),
9   CustomerPhoneNumber VARCHAR(15),
10  CustomerEmail VARCHAR(100),
11  MembershipID INT,
12  CONSTRAINT customer_PK PRIMARY KEY (CustomerID),
13  CONSTRAINT customer_FK FOREIGN KEY (MembershipID)REFERENCES membership(MembershipID))
14  AUTO_INCREMENT = 1000;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

CustomerID	CustomerFirstName	CustomerLastName	CustomerAddress	CustomerCity	CustomerState	CustomerP
1000	MARY	SMITH	23 Workhaven Lane	Sacramento	CA	95811
1001	PATRICIA	JOHNSON	1411 Lillydale Drive	Discovery Bay	CA	94505
1002	LINDA	WILLIAMS	1913 Hanoi Way	Oceanside	CA	92058
1003	BARBARA	JONES	1121 Loja Avenue	Los Angeles	CA	90006
1004	ELIZABETH	JONES	692 Joliet Street	Reno	NV	89508
1005	JENNIFER	DAVIS	1566 Inegl Manor	Sloan	NV	89054
1006	MARIA	MILLER	53 Idfu Parkway	Mesquite	NV	89034
1007	SUSAN	WILSON	1795 Santiago de Compostela Way	Oceanside	CA	92049
1008	MARGARET	MOORE	900 Santiago de Compostela Parkway	Oceanside	CA	92049

customer 2 | Apply

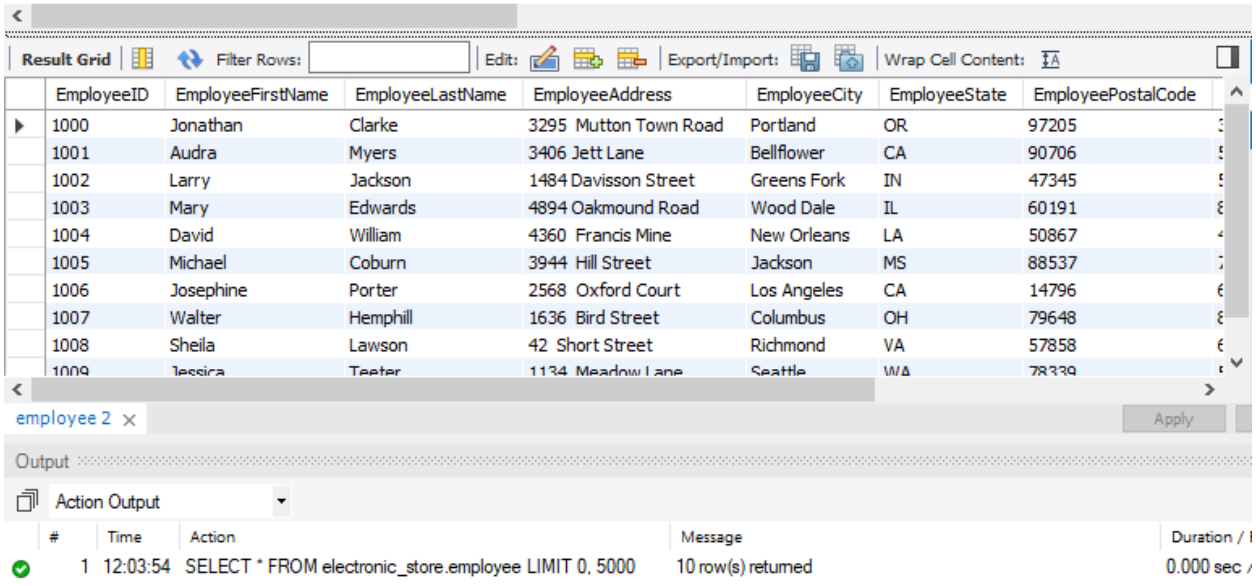
Output

Action Output

#	Time	Action	Message	Duration
1	12:09:53	SELECT * FROM electronic_store.customer LIMIT 0, 5000	58 row(s) returned	0.000 se

Employee Table

```
1   CREATE TABLE employee
2   (EmployeeID INT NOT NULL AUTO_INCREMENT,
3   EmployeeFirstName VARCHAR(25) NOT NULL,
4   EmployeeLastName VARCHAR(25) NOT NULL,
5   EmployeeAddress VARCHAR(30),
6   EmployeeCity VARCHAR(20),
7   EmployeeState VARCHAR(2),
8   EmployeePostalCode VARCHAR(9),
9   EmployeePhoneNumber VARCHAR(11),
10  EmployeeEmail VARCHAR(100),
11  CONSTRAINT Employee_PK PRIMARY KEY (EmployeeID))
12  AUTO_INCREMENT = 1000;
```



Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

EmployeeID	EmployeeFirstName	EmployeeLastName	EmployeeAddress	EmployeeCity	EmployeeState	EmployeePostalCode
1000	Jonathan	Clarke	3295 Mutton Town Road	Portland	OR	97205
1001	Audra	Myers	3406 Jett Lane	Bellflower	CA	90706
1002	Larry	Jackson	1484 Davisson Street	Greens Fork	IN	47345
1003	Mary	Edwards	4894 Oakmound Road	Wood Dale	IL	60191
1004	David	William	4360 Francis Mine	New Orleans	LA	50867
1005	Michael	Coburn	3944 Hill Street	Jackson	MS	88537
1006	Josephine	Porter	2568 Oxford Court	Los Angeles	CA	14796
1007	Walter	Hemphill	1636 Bird Street	Columbus	OH	79648
1008	Sheila	Lawson	42 Short Street	Richmond	VA	57858
1009	Jessica	Teeter	1134 Meadow Lane	Seattle	WA	78339

employee 2 | Apply

Output

Action Output

#	Time	Action	Message	Duration /
1	12:03:54	SELECT * FROM electronic_store.employee LIMIT 0, 5000	10 row(s) returned	0.000 sec

Inventory Table

```
1   CREATE TABLE inventory
2   (ProductID INT NOT NULL AUTO_INCREMENT,
3   Description VARCHAR(100) NOT NULL,
4   InventoryQty INT DEFAULT(0),
5   Price DECIMAL(6,2),
6   CONSTRAINT Price CHECK (Price > 0),
7   CONSTRAINT product_PK PRIMARY KEY (ProductID));
```

Result Grid | Filter Rows: | Edit: | Export/Import:

ProductID	Description	InventoryQty	Price
1	Playstation 5	2	499.99
2	Nintendo Switch	12	299.99
3	iPhone 12 Pro Max	36	1099.99
4	X Box Series X	13	499.99
5	Samsung Galaxy S21	16	799.99
6	Samsung - 55" Class 7	4	449.99
7	AirPods Pro	35	199.99
8	Beats by Dr Dre	26	169.99
9	Samsung - Galaxy Buds	20	179.99
10	Apple Watch Series 6 (GPS)	25	429.00
11	iMac 27"-inch - Apple	11	1999.99
12	iPad Pro - 12.9"-inch	27	1099.99
13	Drone - DJI	5	799.99
14	Samsung Galaxy Watch	16	199.99
NULL	NULL	NULL	NULL

inventory 2

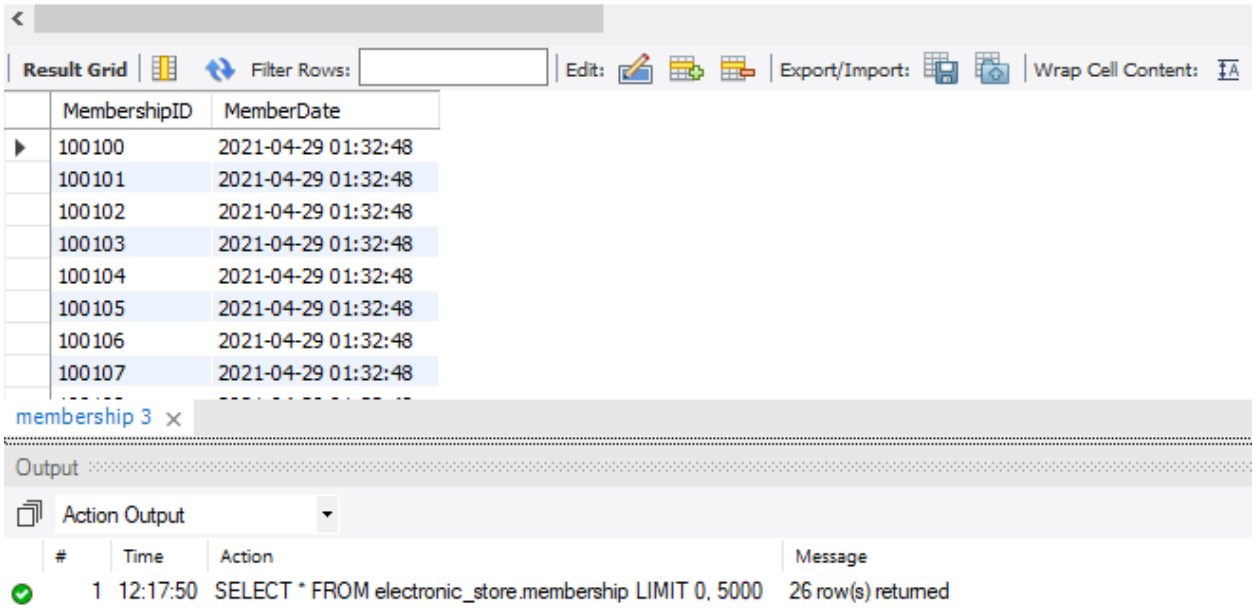
Output

Action Output

#	Time	Action	Message
1	12:13:33	SELECT * FROM electronic_store.inventory LIMIT 0, 5000	14 row(s) returned

Membership Table

```
1    CREATE TABLE membership
2   (MembershipID INT NOT NULL AUTO_INCREMENT,
3    MemberDate DATETIME DEFAULT CURRENT_TIMESTAMP,
4    CONSTRAINT membership_PK PRIMARY KEY (MembershipID))
5    AUTO_INCREMENT = 100100;
```



Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

MembershipID	MemberDate
100100	2021-04-29 01:32:48
100101	2021-04-29 01:32:48
100102	2021-04-29 01:32:48
100103	2021-04-29 01:32:48
100104	2021-04-29 01:32:48
100105	2021-04-29 01:32:48
100106	2021-04-29 01:32:48
100107	2021-04-29 01:32:48

membership 3

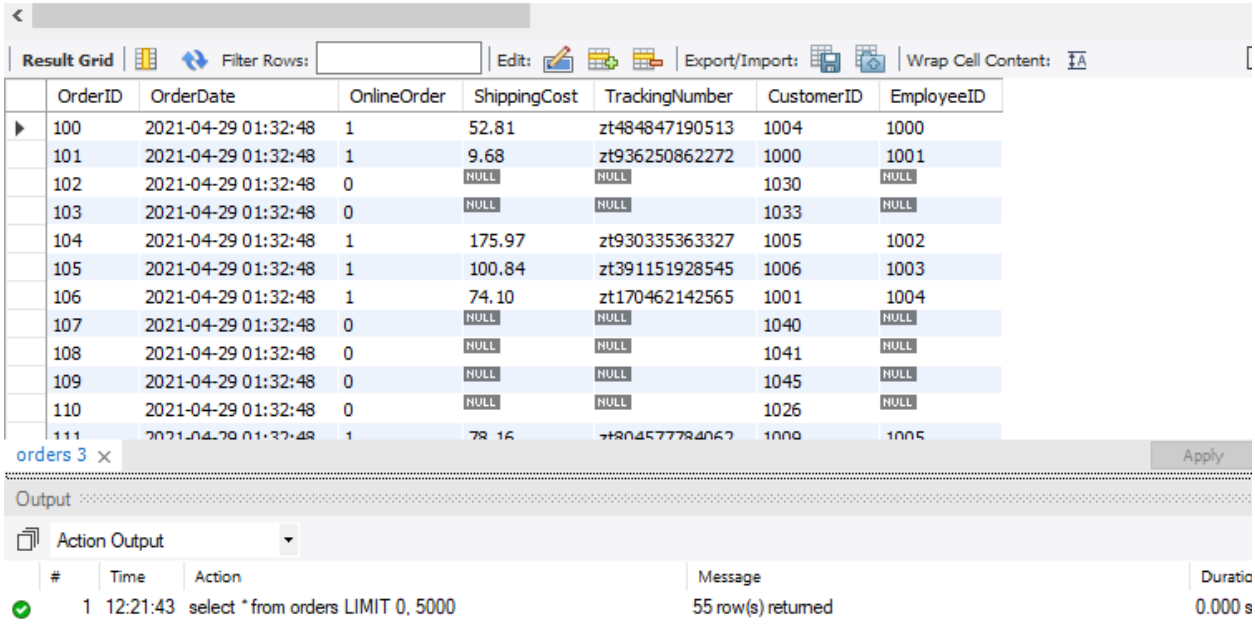
Output

Action Output

#	Time	Action	Message
1	12:17:50	SELECT * FROM electronic_store.membership LIMIT 0, 5000	26 row(s) returned

Orders Table

```
1   CREATE TABLE orders
2   (OrderID INT NOT NULL AUTO_INCREMENT,
3   OrderDate DATETIME DEFAULT CURRENT_TIMESTAMP,
4   OnlineOrder BOOLEAN,
5   ShippingCost DECIMAL(6,2),
6   TrackingNumber VARCHAR(100),
7   CustomerID INT,
8   EmployeeID INT,
9   CONSTRAINT ShippingCost CHECK (ShippingCost > 0),
10  CONSTRAINT order_PK PRIMARY KEY (OrderID),
11  CONSTRAINT order_FK FOREIGN KEY (CustomerID) REFERENCES customer(CustomerID) on update cascade,
12  CONSTRAINT order_FK2 FOREIGN KEY (EmployeeID) REFERENCES employee(EmployeeID) on update cascade)
13  AUTO_INCREMENT = 100; #orders start at 100
14
15
```



Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

OrderID	OrderDate	OnlineOrder	ShippingCost	TrackingNumber	CustomerID	EmployeeID
100	2021-04-29 01:32:48	1	52.81	zt484847190513	1004	1000
101	2021-04-29 01:32:48	1	9.68	zt936250862272	1000	1001
102	2021-04-29 01:32:48	0	NULL	NULL	1030	NULL
103	2021-04-29 01:32:48	0	NULL	NULL	1033	NULL
104	2021-04-29 01:32:48	1	175.97	zt930335363327	1005	1002
105	2021-04-29 01:32:48	1	100.84	zt391151928545	1006	1003
106	2021-04-29 01:32:48	1	74.10	zt170462142565	1001	1004
107	2021-04-29 01:32:48	0	NULL	NULL	1040	NULL
108	2021-04-29 01:32:48	0	NULL	NULL	1041	NULL
109	2021-04-29 01:32:48	0	NULL	NULL	1045	NULL
110	2021-04-29 01:32:48	0	NULL	NULL	1026	NULL
111	2021-04-29 01:32:48	1	78.16	zt804577784062	1009	1005

orders 3 | Apply

Output

Action Output

#	Time	Action	Message	Duratio
1	12:21:43	select * from orders LIMIT 0, 5000	55 row(s) returned	0.000 s

OrderProcessing Table

```
1    CREATE TABLE orderProcessing
2    (OrderID INT NOT NULL,
3    ProductID INT NOT NULL,
4    Qty INT DEFAULT NULL,
5    CONSTRAINT Qty CHECK (Qty > 0),
6    CONSTRAINT orderProcessing_PK PRIMARY KEY (OrderID, ProductID),
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

OrderID	ProductID	Qty
100	2	1
101	5	1
101	9	1
102	1	1
103	7	2
103	10	1
104	3	1
104	7	1
104	10	1
104	12	1
105	13	3
106	1	1
107	6	1
108	2	1
108	8	3
109	5	3
110	11	1

orderprocessing 2

Output

Action Output

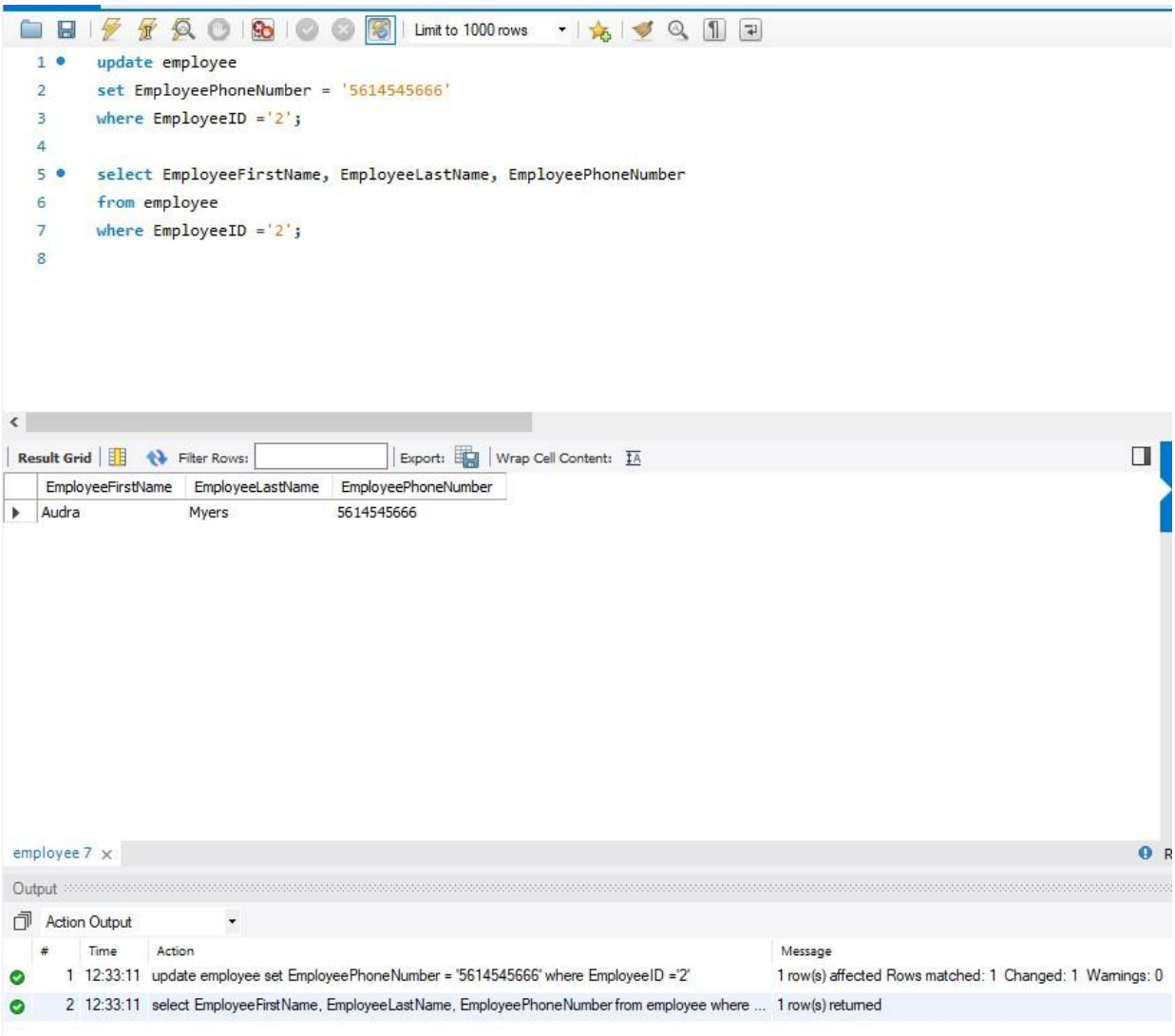
#	Time	Action	Message
1	12:21:43	select * from orders LIMIT 0, 5000	55 row(s) returned

5) Queries:

1) Update employee phone number.

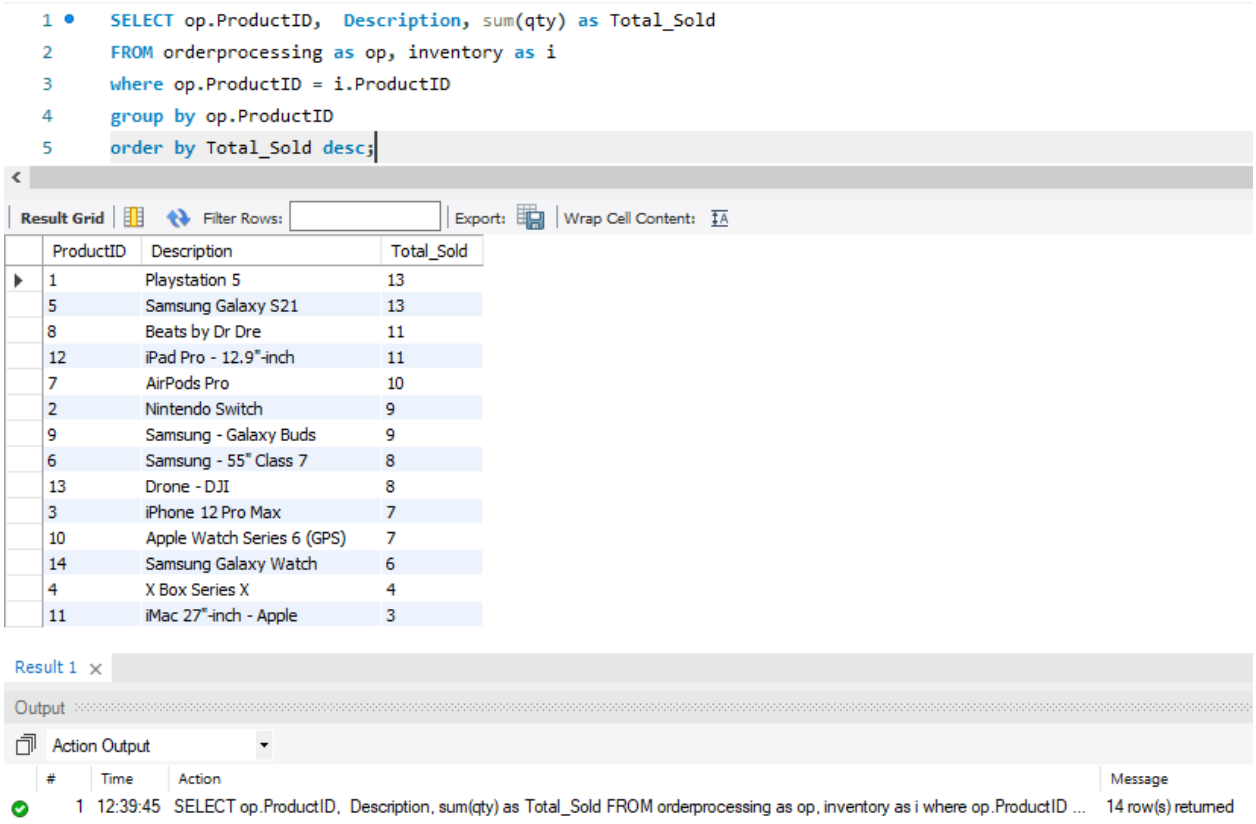
Update employee
Set EmployeePhoneNumber = ‘5614545666’
Where EmployeeID = ‘2’;

Select EmployeeFirstName, EmployeeLastName, EmployeePhoneNumber
From employee
Where EmployeeID = ‘2’;



2) “List store products sold from most popular to least popular”

```
SELECT op.ProductID,  Description, sum(qty) as Total_Sold
FROM orderprocessing as op, inventory as i
where op.ProductID = i.ProductID
group by op.ProductID
order by Total_Sold desc;
```



```
1  SELECT op.ProductID,  Description, sum(qty) as Total_Sold
2  FROM orderprocessing as op, inventory as i
3  where op.ProductID = i.ProductID
4  group by op.ProductID
5  order by Total_Sold desc;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

ProductID	Description	Total_Sold
1	Playstation 5	13
5	Samsung Galaxy S21	13
8	Beats by Dr Dre	11
12	iPad Pro - 12.9"-inch	11
7	AirPods Pro	10
2	Nintendo Switch	9
9	Samsung - Galaxy Buds	9
6	Samsung - 55" Class 7	8
13	Drone - DJI	8
3	iPhone 12 Pro Max	7
10	Apple Watch Series 6 (GPS)	7
14	Samsung Galaxy Watch	6
4	X Box Series X	4
11	iMac 27"-inch - Apple	3

Result 1

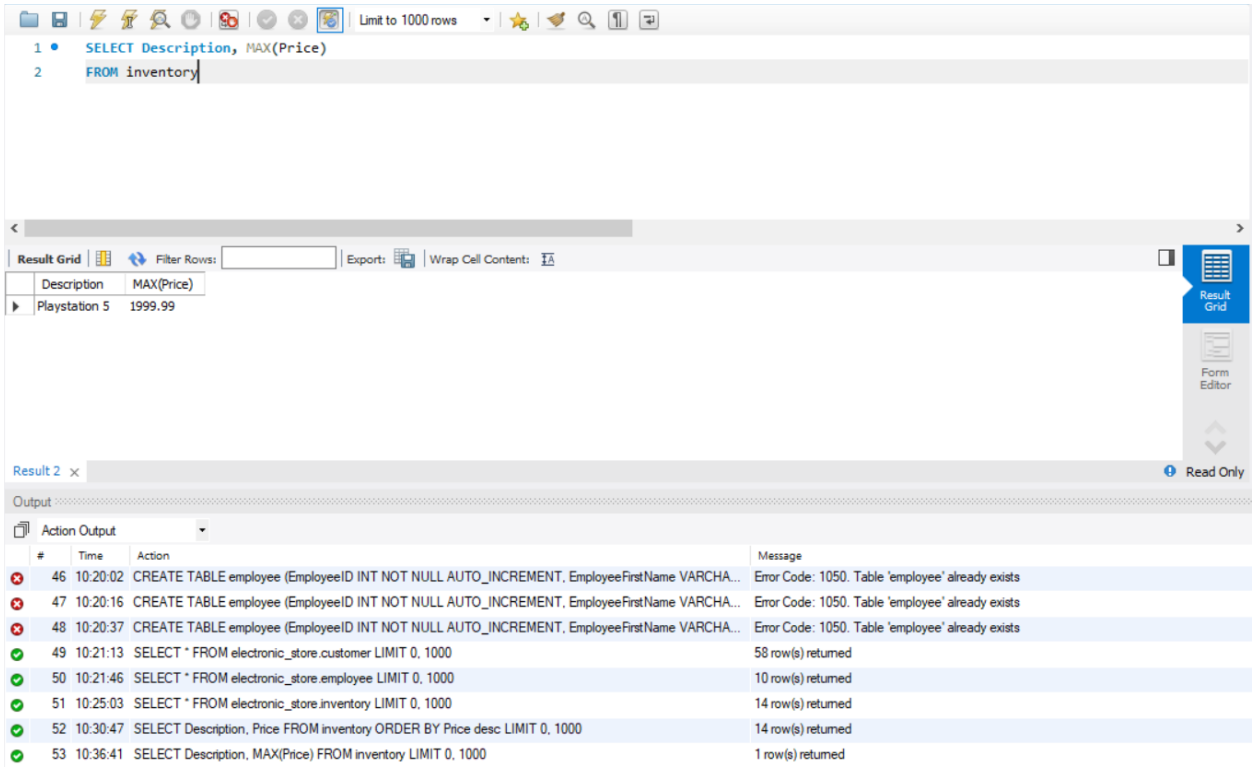
Output

Action Output

#	Time	Action	Message
1	12:39:45	SELECT op.ProductID, Description, sum(qty) as Total_Sold FROM orderprocessing as op, inventory as i where op.ProductID ...	14 row(s) returned

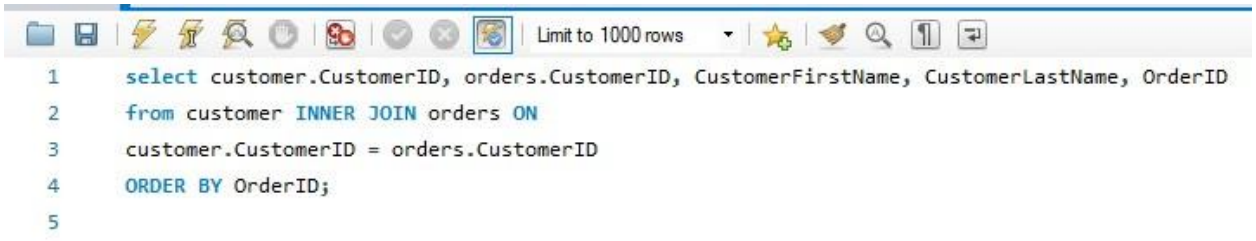
3) What's our most expensive product?

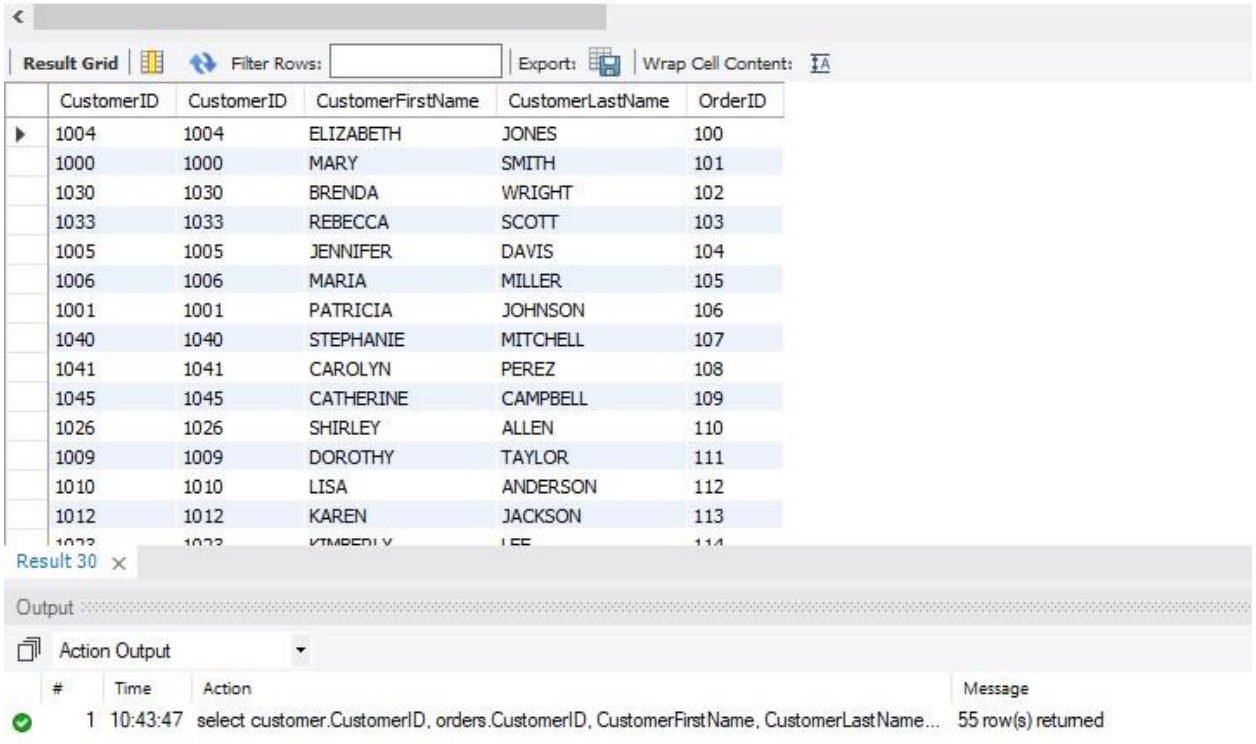
SELECT Description, MAX(Price)
FROM inventory



4) For each customer who placed an order, what is the customer's name and order number.

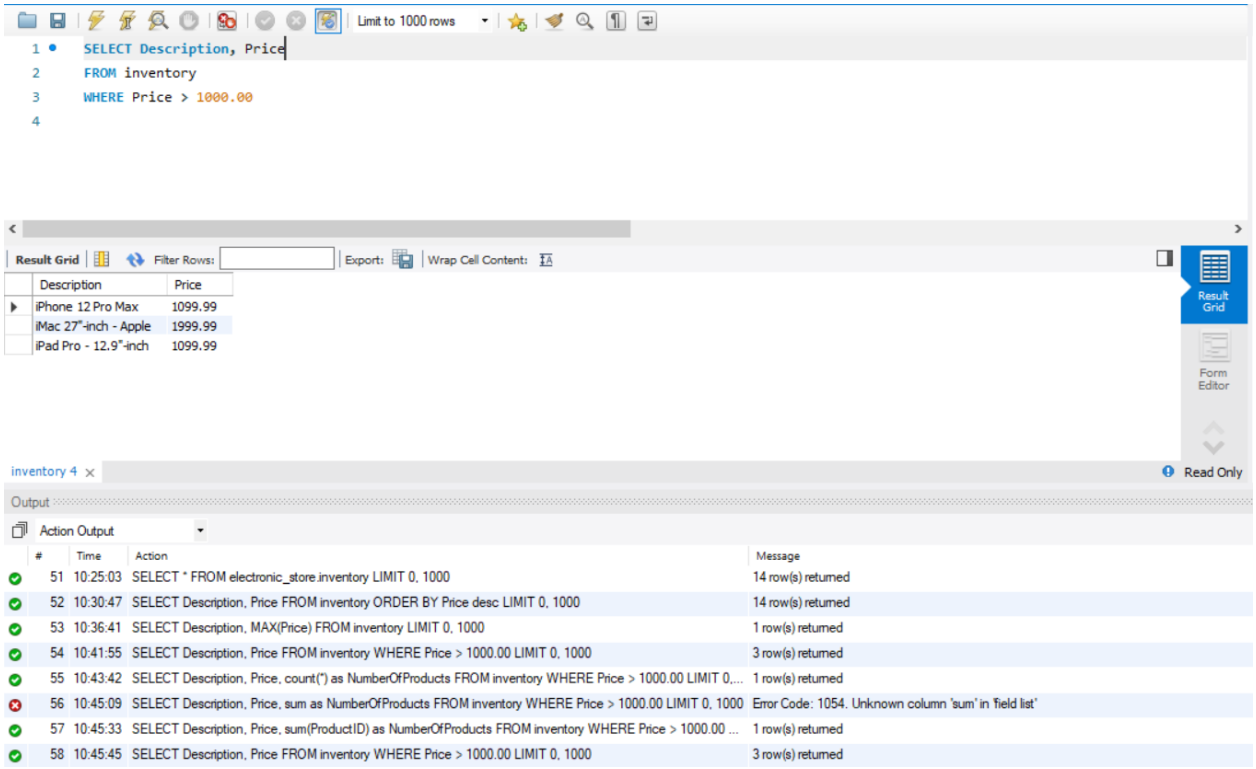
```
select customer.CustomerID, orders.CustomerID, CustomerFirstName,
CustomerLastName, OrderID
from customer INNER JOIN orders ON
customer.CustomerID = orders.CustomerID
ORDER BY OrderID;
```





5) How many products do we have that cost over $1000?

```
SELECT Description, Price
FROM inventory
WHERE Price > 1000.00
```



6) List all customers with memberships. Show the customer ID, name, membershipID(Inner join..on)

select customer.MembershipID, membership.MembershipID,CustomerID, CustomerFirstName, CustomerLastName
from customer INNER JOIN membership ON
customer.MembershipID = membership.MembershipID
ORDER BY membership.MembershipID;

```
1   select customer.MembershipID, membership.MembershipID,CustomerID, CustomerFirstName, CustomerLastName
2   from customer INNER JOIN membership ON
3   customer.MembershipID = membership.MembershipID
4   ORDER BY membership.MembershipID;
5
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

MembershipID	MembershipID	CustomerID	CustomerFirstName	CustomerLastName
100100	100100	1003	BARBARA	JONES
100100	100100	1004	ELIZABETH	JONES
100101	100101	1008	MARGARET	MOORE
100102	100102	1009	DOROTHY	TAYLOR
100103	100103	1014	HELEN	HARRIS
100103	100103	1015	SANDRA	HARRIS
100103	100103	1016	DONNA	HARRIS
100103	100103	1017	CAROL	HARRIS
100104	100104	1019	SHARON	ROBINSON
100105	100105	1020	MICHELLE	CLARK
100106	100106	1023	KIMBERLY	LEE
100106	100106	1024	DEBORAH	WALKER
100107	100107	1028	ANGELA	HERNANDEZ
100108	100108	1029	MELISSA	KING
100109	100109	1030	BRENDA	WRIGHT

Result 42

Output

Action Output

#	Time	Action	Message
1	11:09:02	select customer.MembershipID, membership.MembershipID,CustomerID, CustomerFirstNam...	32 row(s) returned

7) Create an Invoice for order number 128: (Multi-table join)

```
SELECT c.CustomerID, c.CustomerFirstName, c.CustomerLastName,
c.CustomerAddress, c.CustomerCity,
c.CustomerState, c.CustomerPostalCode, op.OrderID, i.ProductID, i.Description, op.Qty,
(op.Qty*i.Price) as Total
FROM   customer AS c JOIN orders AS o JOIN orderProcessing  AS op JOIN  inventory
AS i
USING (ProductID)
USING (OrderID)
USING (CustomerID)
WHERE op.OrderID = 128;
```

```
1  SELECT c.CustomerID, c.CustomerFirstName, c.CustomerLastName, c.CustomerAddress, c.CustomerCity,
2  c.CustomerState, c.CustomerPostalCode, op.OrderID, i.ProductID, i.Description, op.Qty, (op.Qty*i.Price) as Total
3  FROM   customer AS c JOIN orders AS o JOIN orderProcessing AS op JOIN  inventory AS i
4  USING (ProductID)
5  USING (OrderID)
6  USING (CustomerID)
7  WHERE op.OrderID = 128;
8
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

CustomerID	CustomerFirstName	CustomerLastName	CustomerAddress	CustomerCity	CustomerState	CustomerPostalCode	OrderID	ProductID	Description	Qty	Total
1050	ALICE	STEWART	226 Halifax Street	Los Angeles	CA	90013	128	3	iPhone 12 Pro Max	2	2199.98
1050	ALICE	STEWART	226 Halifax Street	Los Angeles	CA	90013	128	7	AirPods Pro	3	599.97
1050	ALICE	STEWART	226 Halifax Street	Los Angeles	CA	90013	128	10	Apple Watch Series 6 (GPS)	2	858.00

Result 12

Output

Action Output

#	Time	Action	Message
1	20:35:00	SELECT c.CustomerID, c.CustomerFirstName, c.CustomerLastName, c.CustomerAddress, c.CustomerCit...	3 row(s) returned

8.) List all customers that placed an online order. Show customerID, name, and orderID (Left join)

SELECT customer.CustomerID, customer.CustomerFirstName AS First_Name,
customer.CustomerLastName AS Last_Name, orders.OrderID
FROM electronic_store.customer
LEFT JOIN orders
ON customer.customerID = orders.customerID WHERE OnlineOrder = 1;

```
1  SELECT customer.CustomerID, customer.CustomerFirstName AS First_Name,
2  customer.CustomerLastName AS Last_Name, orders.OrderID
3  FROM electronic_store.customer
4  LEFT JOIN orders
5  ON customer.customerID = orders.customerID WHERE OnlineOrder = 1;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

CustomerID	First_Name	Last_Name	OrderID
1004	ELIZABETH	JONES	100
1000	MARY	SMITH	101
1005	JENNIFER	DAVIS	104
1006	MARIA	MILLER	105
1001	PATRICIA	JOHNSON	106

Result 9

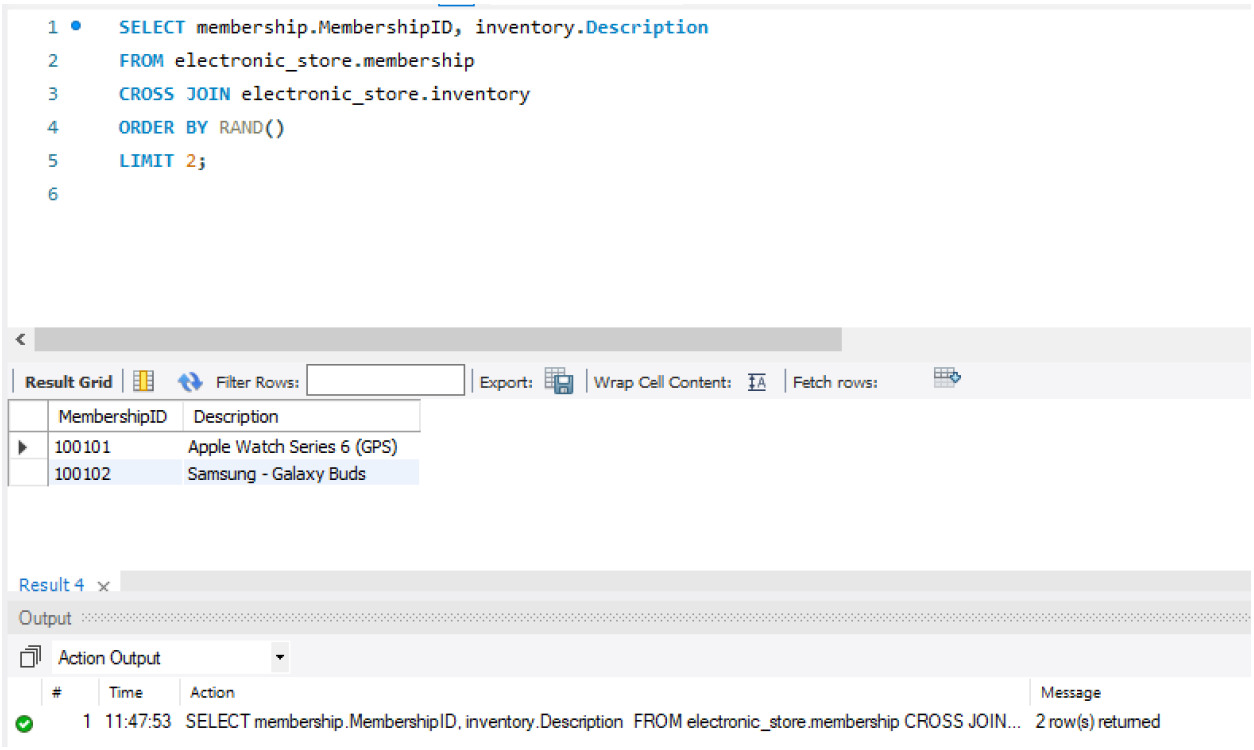
Output

Action Output

#	Time	Action	Message
1	06:34:37	SELECT customer.CustomerID, customer.CustomerFirstName AS First_Name, customer.CustomerLastName A...	32 row(s) returned

9.) Raffle held every two months for loyal members(Cross join)

```
SELECT membership.MembershipID, inventory.Description
FROM electronic_store.membership
CROSS JOIN electronic_store.inventory
ORDER BY RAND()
LIMIT 2;
```



10.) Employee in charged of what order(inner join...using)

SELECT employee.EmployeeID, employee.EmployeeFirstName,
employee.EmployeeLastName, orders.OrderID
FROM electronic_store.employee
INNER JOIN electronic_store.orders using(EmployeeID)
ORDER BY OrderID;

```
1 •  SELECT employee.EmployeeID, employee.EmployeeFirstName, employee.EmployeeLastName, orders.OrderID
2    FROM electronic_store.employee
3    INNER JOIN electronic_store.orders using(EmployeeID)
4    ORDER BY OrderID;
5
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

EmployeeID	EmployeeFirstName	EmployeeLastName	OrderID
1000	Jonathan	Clarke	100
1001	Audra	Myers	101
1002	Larry	Jackson	104
1003	Mary	Edwards	105
1004	David	William	106

Result 34

Output

Action Output

#	Time	Action	Message
1	03:41:15	SELECT employee.EmployeeID, employee.EmployeeFirstName, employee.EmployeeLastName, orders.OrderI...	32 row(s) returned

JDBC: