

# AoL Database Technology - Documentation

## Anggota Kelompok:

1. Aviel Aquino - 2802393711
2. Joshua Christian Rudy Thiopelus - 2802418323
3. Marissa Angelina Santoso - 2802418342

## Query 1: New Product

Add a new, never-before-seen product to the database.

```
1 • INSERT INTO Item (StockCode, StockDescription, StockPrice, StockQuantity)
2   VALUES ('12345', 'LIMITED EDITION LAPTOP', 17.50, 100);
3
```

Output		
#	Time	Action
4	12:04:32	UPDATE OrderedItem SET Quantity = Quantity - 1 WHERE InvoiceID = '123456' AND StockCode = '85048'
5	12:05:18	UPDATE Item SET StockQuantity = StockQuantity + 1 WHERE StockCode = '85048'
6	12:05:38	DELETE FROM OrderedItem WHERE InvoiceID = '123456' AND StockCode = '85048' AND Quantity = 0
7	12:16:08	INSERT INTO Item (StockCode, StockDescription, StockPrice, StockQuantity) VALUES ('12345', 'LIMITED EDITION LAPTOP', 17.50, 100)
8	12:16:57	delete from item where StockCode = '12345'
9	12:17:02	INSERT INTO Item (StockCode, StockDescription, StockPrice, StockQuantity) VALUES ('12345', 'LIMITED EDITION LAPTOP', 17.50, 100)

## Query 2: Customer Order

Write the series of statements required for an existing customer to order two different products in a single transaction.

```
1  INSERT INTO Invoice (InvoiceID, InvoiceDate, CustomerID)
2    VALUES ('123456', NOW(), '13085.0');
3
```

Output ::

Action	Time	Message
# 17	12:19:25	Action
17	12:19:25	start transaction
18	12:19:32	INSERT INTO Invoice (InvoiceID, InvoiceDate, CustomerID) VALUES ('123456', NOW(), 13085)
19	12:21:10	SELECT * FROM customers WHERE CustomerID = 13085 LIMIT 0, 1000
20	12:22:29	commit
21	12:22:56	start transaction
22	12:23:03	INSERT INTO Invoice (InvoiceID, InvoiceDate, CustomerID) VALUES ('123456', NOW(), 13085.0)

Output ::

Action	Time	Message
# 1	12:19:22	Action
1	12:19:22	start transaction
2	12:19:32	INSERT INTO Invoice (InvoiceID, InvoiceDate, CustomerID) VALUES ('123456', NOW(), 13085)
3	12:21:10	SELECT * FROM customers WHERE CustomerID = 13085 LIMIT 0, 1000
4	12:22:29	commit
5	12:22:56	start transaction
6	12:23:03	INSERT INTO Invoice (InvoiceID, InvoiceDate, CustomerID) VALUES ('123456', NOW(), 13085.0)

Output ::

Action	Time	Message
# 1	12:19:22	Action
1	12:19:22	start transaction
2	12:19:32	INSERT INTO Invoice (InvoiceID, InvoiceDate, CustomerID) VALUES ('123456', NOW(), 13085)
3	12:21:10	SELECT * FROM customers WHERE CustomerID = 13085 LIMIT 0, 1000
4	12:22:29	commit
5	12:22:56	start transaction
6	12:23:03	INSERT INTO Invoice (InvoiceID, InvoiceDate, CustomerID) VALUES ('123456', NOW(), 13085.0)
7	12:24:57	INSERT INTO OrderItem (InvoiceID, StockCode, Quantity) VALUES ('123456', '859487', 5), ('123456', '220411', 8)

```
1 • UPDATE Item AS i
2   JOIN OrderLine AS ol
3     ON i.StockCode = ol.StockCode
4   SET i.StockQuantity = i.StockQuantity - ol.Quantity
5   WHERE i.StockCode = '95948'
6     AND i.StockCode = '22841';
```

Output

Action	Time	Action	Message
•	19 12:21:10	SELECT * FROM customers WHERE CustomerID = '13085' LIMIT 0, 1000	0 rows returned
•	20 12:22:29	commit	0 rows affected
•	21 12:22:56	start transaction	0 rows affected
•	22 12:23:03	INSERT INTO Invoice (InvoiceID, InvoiceDate, CustomerID) VALUES ('123456', NOW(), '13085.0')	1 rows affected
•	23 12:24:57	INSERT INTO OrderLineItem (InvoiceID, StockCode, Quantity) VALUES ('123456', '95948', 5), ('123456', '22841', 8)	2 rows affected Records: 2 Duplicates: 0 Warnings: 0
•	24 12:26:37	UPDATE Item AS i JOIN OrderLine AS ol       ON i.StockCode = ol.StockCode SET i.StockQuantity = i.StockQuantity - ol.Quantity WHERE i.StockCode = '95948' AND i.StockCode = '22841'	0 rows affected Rows matched: 0 Changed: 0 Warnings: 0

## Query 3: Customer Return

Write the statements required to process a return for one of the items from the order you created above.

```
1 UPDATE OrderedItem
2 SET Quantity = Quantity - 1
3 WHERE InvoiceID = '123456'
4 AND StockCode = '85048';
```

Output		
#	Time	Action
1	12:01:28	SELECT cs.CustomerID, cs.CustomerName, SUM(o.Quantity * t.StockPrice) AS TotalSpent FROM Customers cs JOIN Invoice inv ON cs.CustomerID = inv.CustomerID JOIN OrderedItem o ON inv.InvoiceID = o.InvoiceID JOIN Item t ON o.StockCode = t.ItemID WHERE o.InvoiceID = '123456' AND o.StockCode = '85048'
2	12:02:45	SELECT MONTHNAME(inv.InvoiceDate) AS Month, SUM(o.Quantity * t.StockPrice) AS Revenue FROM Invoice inv JOIN OrderedItem o ON inv.InvoiceID = o.InvoiceID JOIN Item t ON o.StockCode = t.ItemID WHERE o.InvoiceID = '123456' AND o.StockCode = '85048'
3	12:04:23	START TRANSACTION
4	12:04:32	UPDATE OrderedItem SET Quantity = Quantity - 1 WHERE InvoiceID = '123456' AND StockCode = '85048'
5	12:05:18	UPDATE Item SET StockQuantity = StockQuantity + 1 WHERE StockCode = '85048'

```
1 ● DELETE FROM OrderedItem
2   WHERE InvoiceID = '123456'
3     AND StockCode = '85048'
4     AND Quantity = 0;
```

Output			
#	Time	Action	Message
1	12:01:29	SELECT cs.CustomerID, cs.CustomerName, SUM(ai.Quantity * it.StockPrice) AS TotalSpent FROM Customers cs JOIN Invoice inv ON cs.CustomerID = inv.CustomerID JOIN OrderedItem oi ON inv.InvoiceID = oi.InvoiceID JOIN Item it ON oi.StockCode ...	10 row(s) returned
2	12:02:45	SELECT MONTHNAME(inv.InvoiceDate) AS Month, SUM(ai.Quantity * it.StockPrice) AS Revenue FROM Invoice inv JOIN OrderedItem oi ON inv.InvoiceID = oi.InvoiceID JOIN Item it ON oi.StockCode ...	12 row(s) returned
3	12:04:23	START TRANSACTION	0 row(s) affected
4	12:04:32	UPDATE OrderedItem SET Quantity = Quantity - 1 WHERE InvoiceID = '123456' AND StockCode = '85048'	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0
5	12:05:19	UPDATE Item SET StockQuantity = StockQuantity + 1 WHERE StockCode = '85048'	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0
6	12:05:38	DELETE FROM OrderedItem WHERE InvoiceID = '123456' AND StockCode = '85048' AND Quantity = 0	0 row(s) affected

## Query 4: Analytical Report

Write a query to find the top 10 customers by total money spent.

```
1 •  SELECT
2      cs.CustomerID,
3      cs.CustomerName,
4      SUM(oi.Quantity * it.StockPrice) AS TotalSpent
5  FROM Customers cs
6  JOIN Invoice inv ON cs.CustomerID = inv.CustomerID
7  JOIN OrderedItem oi ON inv.InvoiceID = oi.InvoiceID
8  JOIN Item it ON oi.StockCode = it.StockCode
9  GROUP BY cs.CustomerID, cs.CustomerName
10 ORDER BY TotalSpent DESC
11 LIMIT 10;
```

Result Grid			
	CustomerID	CustomerName	TotalSpent
▶	17857.0	Robert Morgan	172868718.37870026
	ANON	noname	91203179.52262183
	14607.0	Alexander Wilson	40317363.469430685
	12748.0	Kenneth Lucas	11600511.419283636
	15311.0	Joy Hill	9754918.029238194
	17290.0	Michael Ford	5616419.28001833
	17949.0	Tony Perkins	4170834.7409992814
	15867.0	Lauren Pierce	3842776.5900555775
	14667.0	Caitlin Barker	3019387.910188541
	13268.0	Brandon Graham	2978634.559908062

## Query 5: Analytical Report

Write a query to identify the month with the highest total sales revenue in the year 2011.

```
1 •  SELECT
2      MONTHNAME(inv.InvoiceDate) AS Month,
3      SUM(oi.Quantity * it.StockPrice) AS Revenue
4  FROM Invoice inv
5  JOIN OrderedItem oi ON inv.InvoiceID = oi.InvoiceID
6  JOIN Item it ON oi.StockCode = it.StockCode
7  WHERE YEAR(inv.InvoiceDate) = 2011
8  GROUP BY MONTH(inv.InvoiceDate), MONTHNAME(inv.InvoiceDate)
9  ORDER BY Revenue DESC
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
Month	Revenue			
► November	166087626.55331153			
October	55461821.23895882			
December	34425369.74858913			
September	32016682.993845277			
January	14267531.520145878			
March	12710647.840513445			
May	12319745.448097922			
July	8191317.85469228			
April	7142728.565564804			
February	6375232.402011961			
June	6364236.5323526785			
August	5626708.643974885			

## Trigger

Create a trigger that automatically updates the inventory level of a product whenever a transaction (sale or return) involving that product is recorded.

```
1  DELIMITER $$  
2  
3  •  CREATE TRIGGER CustomerOrder  
4    AFTER INSERT ON OrderedItem  
5    FOR EACH ROW  
6    BEGIN  
7      UPDATE Item  
8      SET StockQuantity = StockQuantity - NEW.Quantity  
9      WHERE StockCode = NEW.StockCode;  
10   END $$  
11  
12  DELIMITER ;
```

Output		
#	Time	Action
5	20:37:48	CREATE DATABASE aofinaldb
6	20:37:51	CREATE DATABASE aofinaldb
7	20:45:07	SELECT COUNT(*) FROM OrderedItem LIMIT 0, 1000
8	20:45:23	SELECT COUNT(*) FROM customers LIMIT 0, 1000
9	20:45:24	SELECT COUNT(*) FROM customers LIMIT 0, 1000
10	21:10:25	CREATE TRIGGER CustomerOrder AFTER INSERT ON OrderedItem FOR EACH ROW BEGIN UPDATE Item SET StockQuantity = StockQuantity - NEW.Q... 0 row(s) affected

```
1  DELIMITER $$  
2  
3  CREATE TRIGGER CustomerReturn  
4    AFTER UPDATE ON OrderedItem  
5    FOR EACH ROW  
6    BEGIN  
7      DECLARE diff INT;  
8  
9      SET diff = OLD.Quantity - NEW.Quantity;  
10  
11     UPDATE Item  
12     SET StockQuantity = StockQuantity + diff;  
13     WHERE StockCode = NEW.StockCode;  
14  
15   END $$  
16  
17  DELIMITER ;
```

Output		
#	Time	Action
6	20:37:51	CREATE DATABASE aofinaldb
7	20:45:07	SELECT COUNT(*) FROM OrderedItem LIMIT 0, 1000
8	20:45:23	SELECT COUNT(*) FROM customers LIMIT 0, 1000
9	20:45:24	SELECT COUNT(*) FROM customers LIMIT 0, 1000
10	21:10:25	CREATE TRIGGER CustomerOrder AFTER INSERT ON OrderedItem FOR EACH ROW BEGIN UPDATE Item SET StockQuantity = StockQuantity - NEW.Q... 0 row(s) affected
11	21:13:03	CREATE TRIGGER CustomerReturn AFTER UPDATE ON OrderedItem FOR EACH ROW BEGIN DECLARE diff INT; SET diff = OLD.Quantity - NEW.Q... 0 row(s) affected

## Stored Procedure

Create a stored procedure named GetCustomerInvoiceHistory that accepts a CustomerID as input and returns a complete list of all invoices (including the date and total value) belonging to that customer.

```
2  DELIMITER $$  
3  
4  • CREATE PROCEDURE GetCustomerInvoiceHistory  
5      (IN CustomerID VARCHAR(10))  
6  BEGIN  
7      SELECT  
8          inv.InvoiceID,  
9          inv.InvoiceDate,  
10         SUM(oi.Quantity * it.StockPrice) AS TotalValue  
11     FROM Invoice AS inv  
12     JOIN OrderedItem AS oi ON inv.InvoiceID = oi.InvoiceID  
13     JOIN Item AS it ON oi.StockCode = it.StockCode  
14     WHERE inv.CustomerID = CustomerID  
15     GROUP BY inv.InvoiceID, inv.InvoiceDate  
16     ORDER BY inv.InvoiceDate DESC;  
17  END $$  
18  
19  DELIMITER ;
```

Output			Cor
#	Time	Action	Message
12	23:53:20	SHOW PROCEDURE STATUS WHERE Db = DATABASE()	1 row(s) returned
13	23:54:13	CALL GetCustomerInvoiceHistory('13085.0')	Error Code: 1054. Unknown column '1.StockPrice' in field list'
14	23:56:38	DROP PROCEDURE GetCustomerInvoiceHistory	0 row(s) affected
15	23:56:55	SHOW PROCEDURE STATUS	26 row(s) returned
16	23:57:05	SHOW PROCEDURE STATUS WHERE Db = DATABASE()	0 row(s) returned
17	23:58:09	CREATE PROCEDURE GetCustomerInvoiceHistory (IN CustomerID VARCHAR(10)) BEGIN SELECT inv.InvoiceID, inv.InvoiceDate, SUM(oi.Q...)	0 row(s) affected

## Test Trigger

```
1 •   SELECT StockCode, StockQuantity
2     FROM Item
3   WHERE StockCode = '85123A';
```

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

	StockCode	StockQuantity
▶	85123A	100749
*	HULL	HULL

Item 1 ×

Output :::::

Action Output

#	Time	Action
1	22:53:00	SELECT StockCode, StockQuantity FROM Item WHERE StockCode = '85123A' LIMIT 0, 1000

```
1 • INSERT INTO Invoice (InvoiceID, InvoiceDate, CustomerID)
2   VALUES ('TEST_02', NOW(), '13085.0');
```

Output

#	Time	Action	Message
1	22:53:00	SELECT StockCode, StockQuantity FROM Item WHERE StockCode = '85123A' LIMIT 0, 1000	1 row(s) returned
2	22:55:24	INSERT INTO Invoice (InvoiceID, InvoiceDate, CustomerID) VALUES ('TEST_INV_02', NOW(), '13085.0')	Error Code: 1406. Data too long for column 'InvoiceID' at row 1 1 row(s) affected
3	22:55:59	INSERT INTO Invoice (InvoiceID, InvoiceDate, CustomerID) VALUES ('TEST_02', NOW(), '13085.0')	1 row(s) affected

```
1 • INSERT INTO OrderedItem (InvoiceID, StockCode, Quantity)
2   VALUES ('TEST_02', '85123A', 5);
```

Output

#	Time	Action	Message
1	22:53:00	SELECT StockCode, StockQuantity FROM Item WHERE StockCode = '85123A' LIMIT 0, 1000	1 row(s) returned
2	22:55:24	INSERT INTO Invoice (InvoiceID, InvoiceDate, CustomerID) VALUES ('TEST_INV_02', NOW(), '13085.0')	Error Code: 1406. Data too long for column 'InvoiceID' at row 1 1 row(s) affected
3	22:55:59	INSERT INTO Invoice (InvoiceID, InvoiceDate, CustomerID) VALUES ('TEST_02', NOW(), '13085.0')	1 row(s) affected
4	22:57:38	INSERT INTO OrderedItem (InvoiceID, StockCode, Quantity) VALUES ('TEST_02', '85123A', 5)	1 row(s) affected

```

1 •  SELECT StockCode, StockQuantity
2   FROM Item
3   WHERE StockCode = '85123A';
4

```

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

StockCode	StockQuantity
85123A	100744
85123A	1000

Result Grid  
Form Editor  
Field Types  
Query Stats  
Execution Plan

**Item 2** x

Output:

Action Output

#	Time	Action	Message
1	22:53:00	SELECT StockCode, StockQuantity FROM Item WHERE StockCode = '85123A' LIMIT 0, 1000	1 row(s) returned
2	22:55:24	INSERT INTO Invoice (InvoiceID, InvoiceDate, CustomerID) VALUES ('TEST_INV_02', NOW(), '13085.0')	Error Code: 1406. Data too long for column 'InvoiceID' at row 1
3	22:55:59	INSERT INTO Invoice (InvoiceID, InvoiceDate, CustomerID) VALUES ('TEST_02', NOW(), '13085.0')	1 row(s) affected
4	22:57:38	INSERT INTO OrderedItem (InvoiceID, StockCode, Quantity) VALUES ('TEST_02', '85123A', 5)	1 row(s) affected
5	22:59:09	SELECT StockCode, StockQuantity FROM Item WHERE StockCode = '85123A' LIMIT 0, 1000	1 row(s) returned

Apply Revert Cont

1 • UPDATE OrderedItem

2 SET Quantity = 2

3 WHERE InvoiceID = 'TEST\_02'

4 AND StockCode = '85123A';

Aub h

Output:

Action Output

#	Time	Action	Message
1	22:53:00	SELECT StockCode, StockQuantity FROM Item WHERE StockCode = '85123A' LIMIT 0, 1000	1 row(s) returned
2	22:55:24	INSERT INTO Invoice (InvoiceID, InvoiceDate, CustomerID) VALUES ('TEST_INV_02', NOW(), '13085.0')	Error Code: 1406. Data too long for column 'InvoiceID' at row 1
3	22:55:59	INSERT INTO Invoice (InvoiceID, InvoiceDate, CustomerID) VALUES ('TEST_02', NOW(), '13085.0')	1 row(s) affected
4	22:57:38	INSERT INTO OrderedItem (InvoiceID, StockCode, Quantity) VALUES ('TEST_02', '85123A', 5)	1 row(s) affected
5	22:59:09	SELECT StockCode, StockQuantity FROM Item WHERE StockCode = '85123A' LIMIT 0, 1000	1 row(s) returned
6	23:00:59	UPDATE OrderedItem SET Quantity = 2 WHERE InvoiceID = 'TEST_02' AND StockCode = '85123A'	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0

Context H

1 •   SELECT StockCode, StockQuantity  
 2   FROM Item  
 3   WHERE StockCode = '85123A';  
 4

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

StockCode	StockQuantity
85123A	100747
NULL	NULL

Form Editor  
Field Types  
Query Stats  
Execution Plan

Item 3 × Apply Revert

Output

Action Output

#	Time	Action	Message
1	22:55:24	DELETE FROM OrderedItem WHERE InvoiceID = 'TEST_02' AND StockCode = '85123A';	Error Code: 1406. Data too long for column 'InvoiceID' at row 1
2	22:55:24	INSERT INTO Invoice (InvoiceID, InvoiceDate, CustomerID) VALUES ('TEST_INV_02', NOW(), '13085.0')	1 row(s) affected
3	22:55:59	INSERT INTO Invoice (InvoiceID, InvoiceDate, CustomerID) VALUES ('TEST_02', NOW(), '13085.0')	1 row(s) affected
4	22:57:38	INSERT INTO OrderedItem (InvoiceID, StockCode, Quantity) VALUES ('TEST_02', '85123A', 5)	1 row(s) returned
5	22:59:09	SELECT StockCode, StockQuantity FROM Item WHERE StockCode = '85123A' LIMIT 0, 1000	1 row(s) returned
6	23:00:59	UPDATE OrderedItem SET Quantity = 2 WHERE InvoiceID = 'TEST_02' AND StockCode = '85123A'	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0
7	23:06:05	SELECT StockCode, StockQuantity FROM Item WHERE StockCode = '85123A' LIMIT 0, 1000	1 row(s) returned
8	23:06:05	SELECT StockCode, StockQuantity FROM Item WHERE StockCode = '85123A' LIMIT 0, 1000	1 row(s) returned

Output

Action Output

#	Time	Action	Message
1	22:55:59	INSERT INTO Invoice (InvoiceID, InvoiceDate, CustomerID) VALUES ('TEST_02', NOW(), '13085.0')	1 row(s) affected
2	22:57:38	INSERT INTO OrderedItem (InvoiceID, StockCode, Quantity) VALUES ('TEST_02', '85123A', 5)	1 row(s) affected
3	22:59:09	SELECT StockCode, StockQuantity FROM Item WHERE StockCode = '85123A' LIMIT 0, 1000	1 row(s) returned
4	23:00:59	UPDATE OrderedItem SET Quantity = 2 WHERE InvoiceID = 'TEST_02' AND StockCode = '85123A'	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0
5	23:06:05	SELECT StockCode, StockQuantity FROM Item WHERE StockCode = '85123A' LIMIT 0, 1000	1 row(s) returned
6	23:07:07	DELETE FROM OrderedItem WHERE InvoiceID = 'TEST_02' AND StockCode = '85123A'	1 row(s) affected

```
1 •  SELECT StockCode, StockQuantity
2   FROM Item
3 WHERE StockCode = '85123A';
```

Result Grid	
StockCode	StockQuantity
85123A	100747
85123A	100747

Item 4 ×

Output : Action Output

#	Time	Action	Message
4	22:57:38	INSERT INTO OrderedItem (InvoiceID, StockCode, Quantity) VALUES ('TEST_02', '85123A', 5)	1 row(s) affected
5	22:59:09	SELECT StockCode, StockQuantity FROM Item WHERE StockCode = '85123A' LIMIT 0, 1000	1 row(s) returned
6	23:00:59	UPDATE OrderedItem SET Quantity = 2 WHERE InvoiceID = 'TEST_02' AND StockCode = '85123A'	1 row(s) affected Rows matched: 1 Changed: 1 1
7	23:06:05	SELECT StockCode, StockQuantity FROM Item WHERE StockCode = '85123A' LIMIT 0, 1000	1 row(s) returned
8	23:07:07	DELETE FROM OrderedItem WHERE InvoiceID = 'TEST_02' AND StockCode = '85123A'	1 row(s) affected
9	23:07:41	SELECT StockCode, StockQuantity FROM Item WHERE StockCode = '85123A' LIMIT 0, 1000	1 row(s) returned

## Test Procedure

```
1 • CALL GetCustomerInvoiceHistory('13085.0');
```

The screenshot shows a database management interface with two main panes. On the left is a 'Result Grid' displaying a table of invoice history. On the right is a 'Result 8' pane showing a log of actions and their messages.

**Result Grid:**

InvoiceID	InvoiceDate	TotalValue
123456	2025-12-17 09:20	115.72000122070312
558996	2011-07-05 12:11:00	311.93000316619873
544306	2011-02-17 13:57:00	658.7499980926514
496166	2010-01-29 11:42:00	1101.7999954223633
496092	2010-01-29 10:06:00	1000.0399980549044
490069	2009-12-03 14:07:00	364.79999828338623
490068	2009-12-03 14:06:00	664.8799996240234
489435	2009-12-01 07:46:00	332.5199947357178
489434	2009-12-01 07:45:00	1246.5599956512451

**Result 8:**

#	Time	Action	Message
13	23:54:13	CALL GetCustomerInvoiceHistory('13085.0')	Error Code: 1054. Unknown column '1.StockPrice' in field list'
14	23:56:38	DROP PROCEDURE GetCustomerInvoiceHistory	0 row(s) affected
15	23:56:55	SHOW PROCEDURE STATUS	26 row(s) returned
16	23:57:05	SHOW PROCEDURE STATUS WHERE Db = DATABASE()	0 row(s) returned
17	23:58:09	CREATE PROCEDURE GetCustomerInvoiceHistory (IN CustomerID VARCHAR(10)) BEGIN SELECT inv.InvoiceID, inv.InvoiceDate, SUM(inv.Q...)	0 row(s) affected
18	00:00:17	CALL GetCustomerInvoiceHistory('13085.0')	9 row(s) returned

```
4      SUM(oi.Quantity * it.StockPrice) AS TotalValue
5  FROM Invoice AS inv
6  JOIN OrderedItem AS oi
7    ON inv.InvoiceID = oi.InvoiceID
8  JOIN Item AS it
9    ON oi.StockCode = it.StockCode
10 WHERE inv.CustomerID = '13085.0'
--
```

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

	InvoiceID	InvoiceDate	TotalValue
▶	123456	2025-12-17 09:09:20	115.72000122070312
	558996	2011-07-05 12:11:00	311.93000316619873
	544306	2011-02-17 13:57:00	658.7499980926514
	496166	2010-01-29 11:42:00	1101.7999954223633
	496092	2010-01-29 10:06:00	1000.0399980545044
	490069	2009-12-03 14:07:00	364.79999828338623
	490068	2009-12-03 14:06:00	664.8799896240234
	489435	2009-12-01 07:46:00	332.5199947357178
	489434	2009-12-01 07:45:00	1246.5599956512451