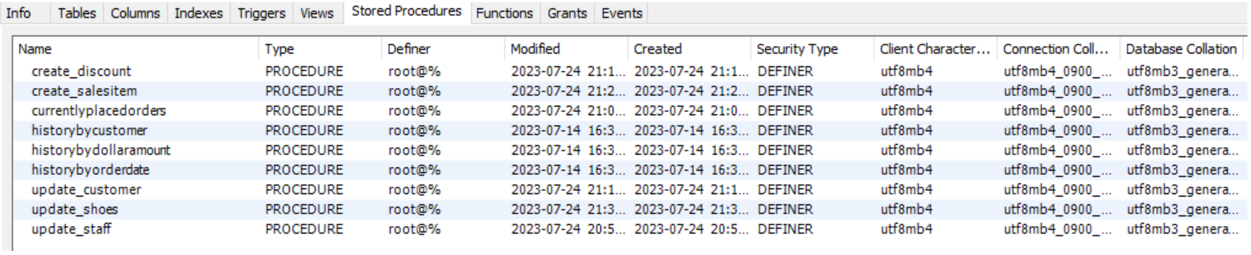
Shoe Shopping

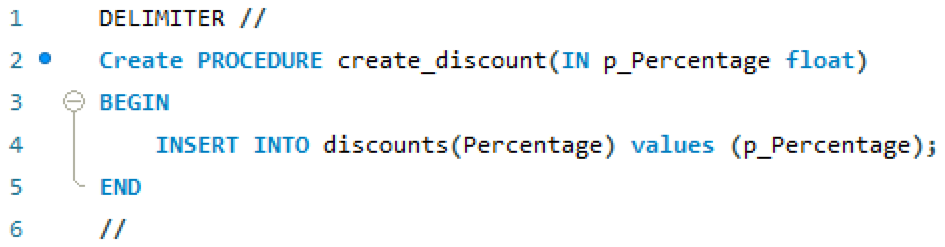
Stored Procedures



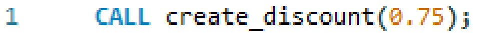
Name	Type	Definer	Modified	Created	Security Type	Client Character...	Connection Coll...	Database Collation
create_discount	PROCEDURE	root@%	2023-07-24 21:1...	2023-07-24 21:1...	DEFINER	utf8mb4	utf8mb4_0900_...	utf8mb3_genera...
create_salesitem	PROCEDURE	root@%	2023-07-24 21:2...	2023-07-24 21:2...	DEFINER	utf8mb4	utf8mb4_0900_...	utf8mb3_genera...
currentlyplacedorders	PROCEDURE	root@%	2023-07-24 21:0...	2023-07-24 21:0...	DEFINER	utf8mb4	utf8mb4_0900_...	utf8mb3_genera...
historybycustomer	PROCEDURE	root@%	2023-07-14 16:3...	2023-07-14 16:3...	DEFINER	utf8mb4	utf8mb4_0900_...	utf8mb3_genera...
historybydollaramount	PROCEDURE	root@%	2023-07-14 16:3...	2023-07-14 16:3...	DEFINER	utf8mb4	utf8mb4_0900_...	utf8mb3_genera...
historybyorderdate	PROCEDURE	root@%	2023-07-14 16:3...	2023-07-14 16:3...	DEFINER	utf8mb4	utf8mb4_0900_...	utf8mb3_genera...
update_customer	PROCEDURE	root@%	2023-07-24 21:1...	2023-07-24 21:1...	DEFINER	utf8mb4	utf8mb4_0900_...	utf8mb3_genera...
update_shoes	PROCEDURE	root@%	2023-07-24 21:3...	2023-07-24 21:3...	DEFINER	utf8mb4	utf8mb4_0900_...	utf8mb3_genera...
update_staff	PROCEDURE	root@%	2023-07-24 20:5...	2023-07-24 20:5...	DEFINER	utf8mb4	utf8mb4_0900_...	utf8mb3_genera...

- create_discount(Percentage)

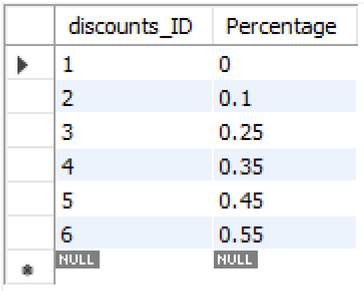
Below is the SQL code to create the stored procedure within the database followed by a call to that procedure and its before/after in the specified table.



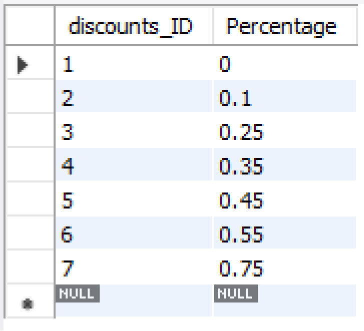
```
DELIMITER //
Create PROCEDURE create_discount(IN p_Percentage float)
BEGIN
    INSERT INTO discounts(Percentage) values (p_Percentage);
END
//
```



```
CALL create_discount(0.75);
```



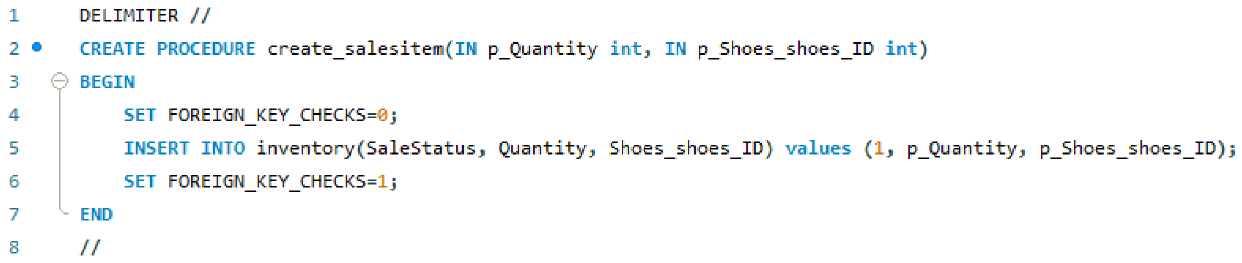
	discounts_ID	Percentage
▶	1	0
	2	0.1
	3	0.25
	4	0.35
	5	0.45
	6	0.55
*	NULL	NULL



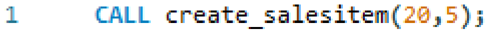
	discounts_ID	Percentage
▶	1	0
	2	0.1
	3	0.25
	4	0.35
	5	0.45
	6	0.55
	7	0.75
*	NULL	NULL

- create_salesitem(Quantity, shoes_ID);

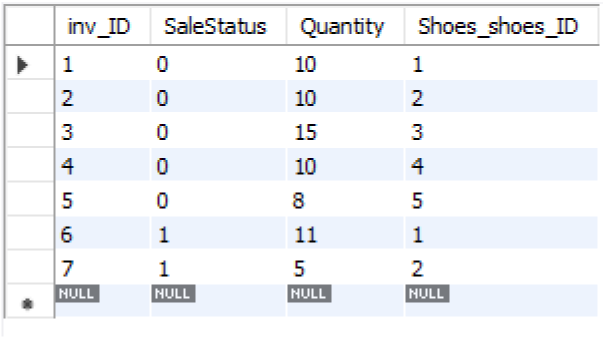
Below is the SQL code to create the stored procedure within the database followed by a call to that procedure and its before/after in the specified table.



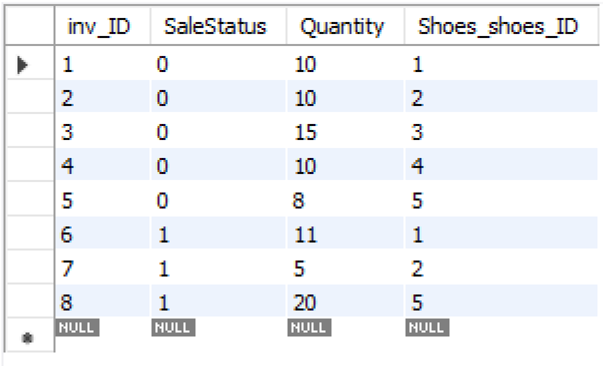
```
DELIMITER //
CREATE PROCEDURE create_salesitem(IN p_Quantity int, IN p_Shoes_shoes_ID int)
BEGIN
    SET FOREIGN_KEY_CHECKS=0;
    INSERT INTO inventory(SaleStatus, Quantity, Shoes_shoes_ID) values (1, p_Quantity, p_Shoes_shoes_ID);
    SET FOREIGN_KEY_CHECKS=1;
END
//
```



```
CALL create_salesitem(20,5);
```



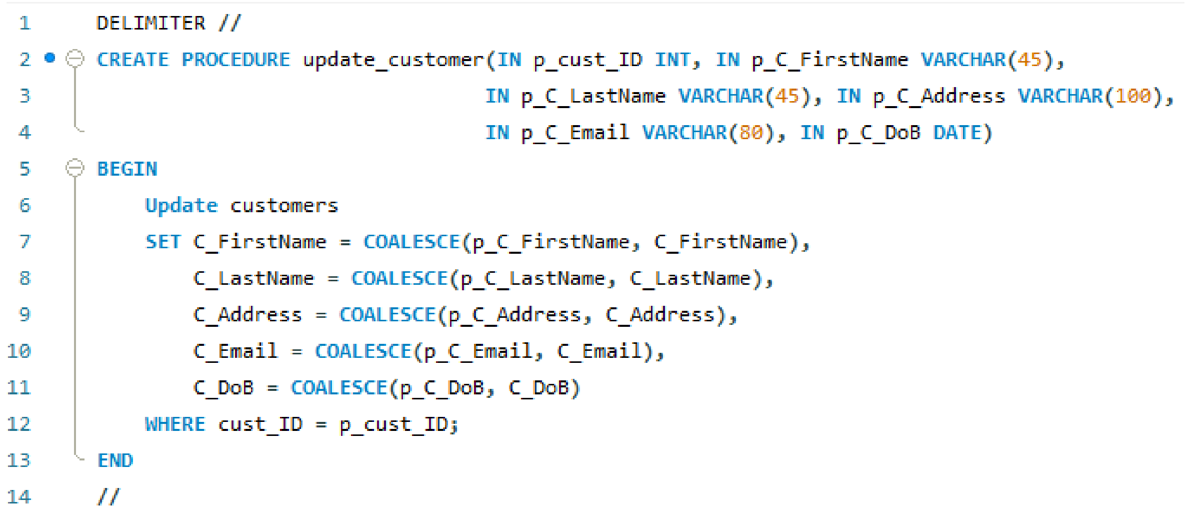
inv_ID	SaleStatus	Quantity	Shoes_shoes_ID
1	0	10	1
2	0	10	2
3	0	15	3
4	0	10	4
5	0	8	5
6	1	11	1
7	1	5	2
NULL	NULL	NULL	NULL



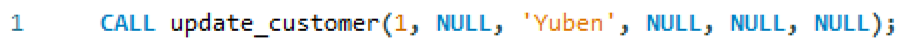
inv_ID	SaleStatus	Quantity	Shoes_shoes_ID
1	0	10	1
2	0	10	2
3	0	15	3
4	0	10	4
5	0	8	5
6	1	11	1
7	1	5	2
8	1	20	5
NULL	NULL	NULL	NULL

- update_customer(cust_ID, C_FirstName, C_LastName, C_Address, C_Email, C_DoB)

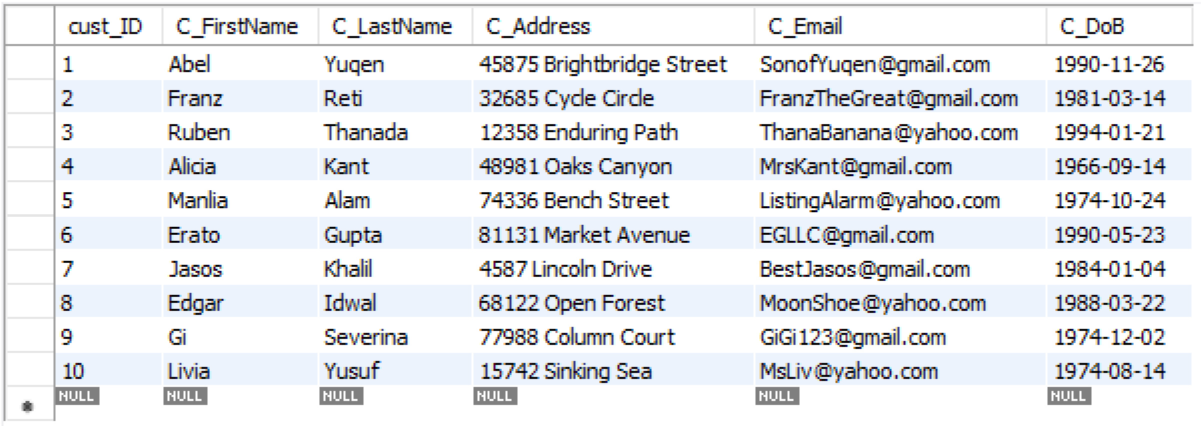
Below is the SQL code to create the stored procedure within the database followed by a call to that procedure and its before/after in the specified table.



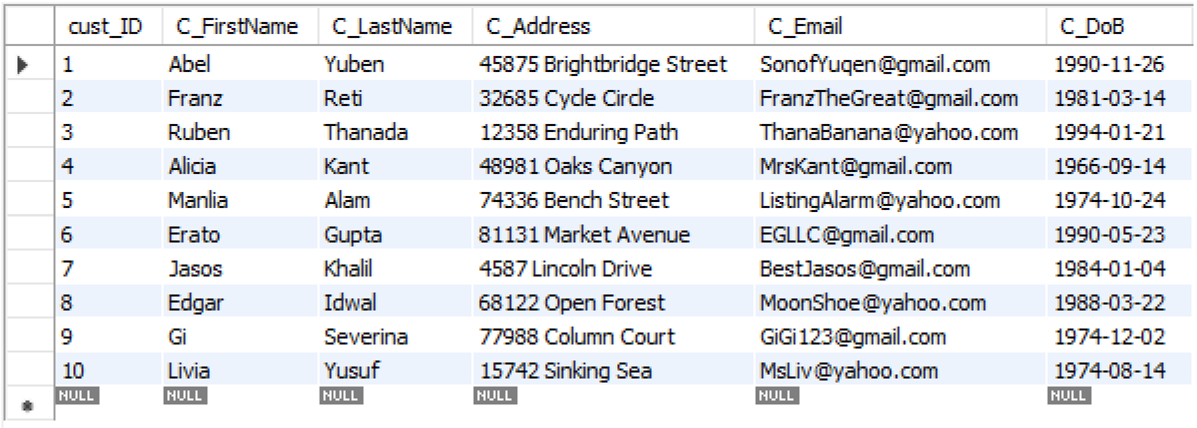
```sql
DELIMITER //
CREATE PROCEDURE update_customer(IN p_cust_ID INT, IN p_C_FirstName VARCHAR(45),
                                IN p_C_LastName VARCHAR(45), IN p_C_Address VARCHAR(100),
                                IN p_C_Email VARCHAR(80), IN p_C_DoB DATE)
BEGIN
    Update customers
    SET C_FirstName = COALESCE(p_C_FirstName, C_FirstName),
        C_LastName = COALESCE(p_C_LastName, C_LastName),
        C_Address = COALESCE(p_C_Address, C_Address),
        C_Email = COALESCE(p_C_Email, C_Email),
        C_DoB = COALESCE(p_C_DoB, C_DoB)
    WHERE cust_ID = p_cust_ID;
END
//
```



```sql
CALL update_customer(1, NULL, 'Yuben', NULL, NULL, NULL);
```



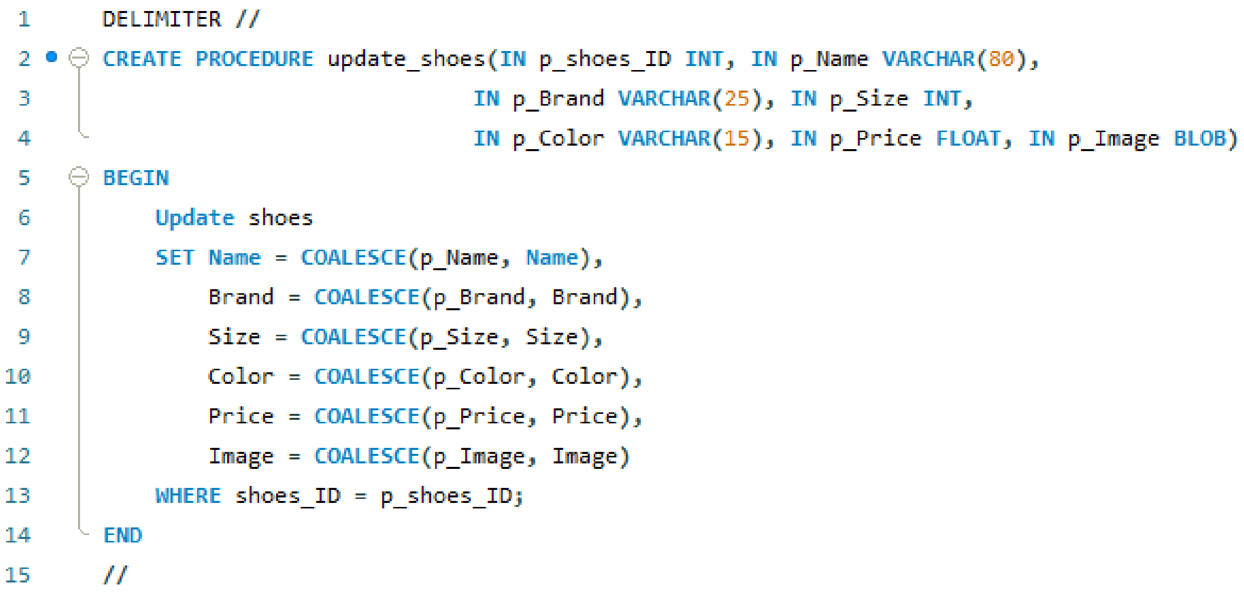
cust_ID	C_FirstName	C_LastName	C_Address	C_Email	C_DoB
1	Abel	Yuqen	45875 Brightbridge Street	SonofYuqen@gmail.com	1990-11-26
2	Franz	Reti	32685 Cycle Circle	FranzTheGreat@gmail.com	1981-03-14
3	Ruben	Thanada	12358 Enduring Path	ThanaBanana@yahoo.com	1994-01-21
4	Alicia	Kant	48981 Oaks Canyon	MrsKant@gmail.com	1966-09-14
5	Manlia	Alam	74336 Bench Street	ListingAlarm@yahoo.com	1974-10-24
6	Erato	Gupta	81131 Market Avenue	EGLLC@gmail.com	1990-05-23
7	Jasos	Khalil	4587 Lincoln Drive	BestJasos@gmail.com	1984-01-04
8	Edgar	Idwal	68122 Open Forest	MoonShoe@yahoo.com	1988-03-22
9	Gi	Severina	77988 Column Court	GiGi123@gmail.com	1974-12-02
10	Livia	Yusuf	15742 Sinking Sea	MsLiv@yahoo.com	1974-08-14
NULL	NULL	NULL	NULL	NULL	NULL



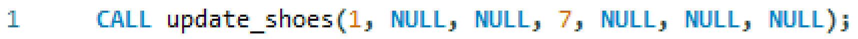
cust_ID	C_FirstName	C_LastName	C_Address	C_Email	C_DoB
1	Abel	Yuben	45875 Brightbridge Street	SonofYuqen@gmail.com	1990-11-26
2	Franz	Reti	32685 Cycle Circle	FranzTheGreat@gmail.com	1981-03-14
3	Ruben	Thanada	12358 Enduring Path	ThanaBanana@yahoo.com	1994-01-21
4	Alicia	Kant	48981 Oaks Canyon	MrsKant@gmail.com	1966-09-14
5	Manlia	Alam	74336 Bench Street	ListingAlarm@yahoo.com	1974-10-24
6	Erato	Gupta	81131 Market Avenue	EGLLC@gmail.com	1990-05-23
7	Jasos	Khalil	4587 Lincoln Drive	BestJasos@gmail.com	1984-01-04
8	Edgar	Idwal	68122 Open Forest	MoonShoe@yahoo.com	1988-03-22
9	Gi	Severina	77988 Column Court	GiGi123@gmail.com	1974-12-02
10	Livia	Yusuf	15742 Sinking Sea	MsLiv@yahoo.com	1974-08-14
NULL	NULL	NULL	NULL	NULL	NULL

- update_shoes(shoes_ID, Name, Brand, Size, Color, Price, Image)

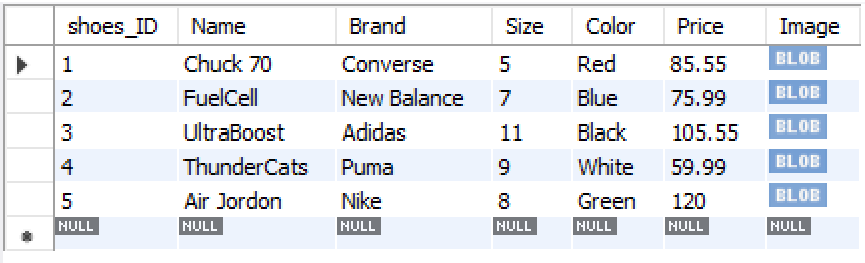
Below is the SQL code to create the stored procedure within the database followed by a call to that procedure and its before/after in the specified table.



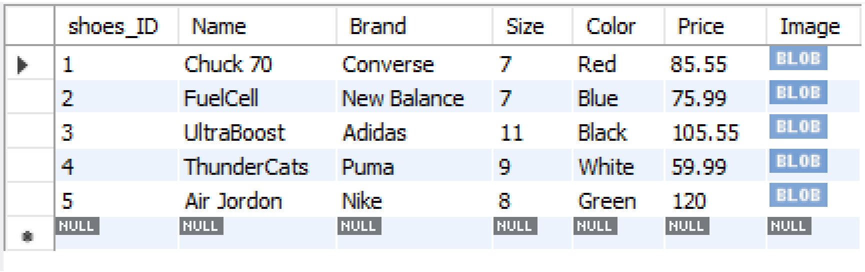
```
DELIMITER //
CREATE PROCEDURE update_shoes(IN p_shoes_ID INT, IN p_Name VARCHAR(80),
                             IN p_Brand VARCHAR(25), IN p_Size INT,
                             IN p_Color VARCHAR(15), IN p_Price FLOAT, IN p_Image BLOB)
BEGIN
    Update shoes
    SET Name = COALESCE(p_Name, Name),
        Brand = COALESCE(p_Brand, Brand),
        Size = COALESCE(p_Size, Size),
        Color = COALESCE(p_Color, Color),
        Price = COALESCE(p_Price, Price),
        Image = COALESCE(p_Image, Image)
    WHERE shoes_ID = p_shoes_ID;
END
//
```



```
CALL update_shoes(1, NULL, NULL, 7, NULL, NULL, NULL);
```



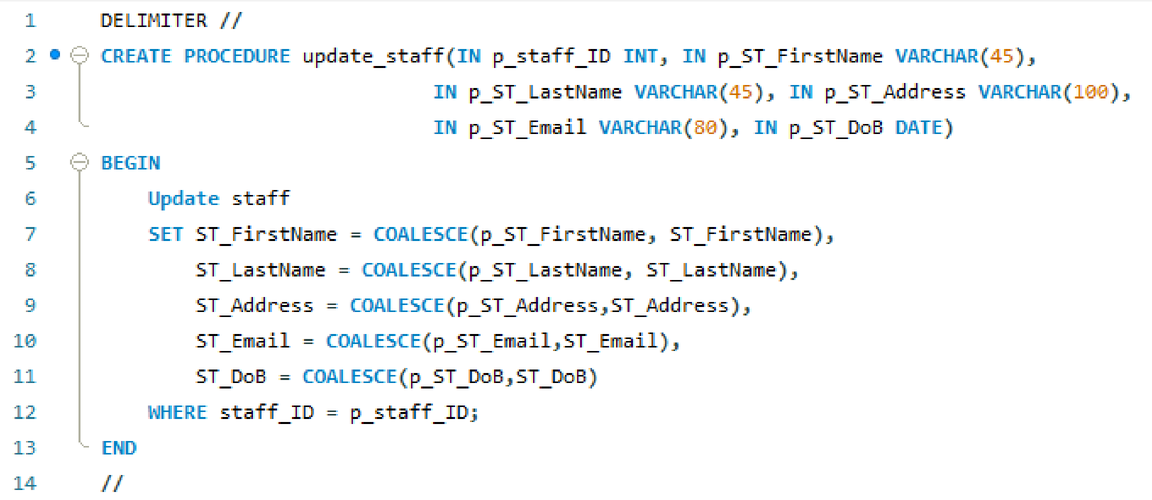
	shoes_ID	Name	Brand	Size	Color	Price	Image
▶	1	Chuck 70	Converse	5	Red	85.55	BLOB
	2	FuelCell	New Balance	7	Blue	75.99	BLOB
	3	UltraBoost	Adidas	11	Black	105.55	BLOB
	4	ThunderCats	Puma	9	White	59.99	BLOB
	5	Air Jordon	Nike	8	Green	120	BLOB
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL



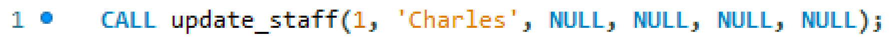
	shoes_ID	Name	Brand	Size	Color	Price	Image
▶	1	Chuck 70	Converse	7	Red	85.55	BLOB
	2	FuelCell	New Balance	7	Blue	75.99	BLOB
	3	UltraBoost	Adidas	11	Black	105.55	BLOB
	4	ThunderCats	Puma	9	White	59.99	BLOB
	5	Air Jordon	Nike	8	Green	120	BLOB
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

- update_staff(staff_ID, ST_FirstName, ST_LastName, ST_Address, ST_Email, ST_DoB)

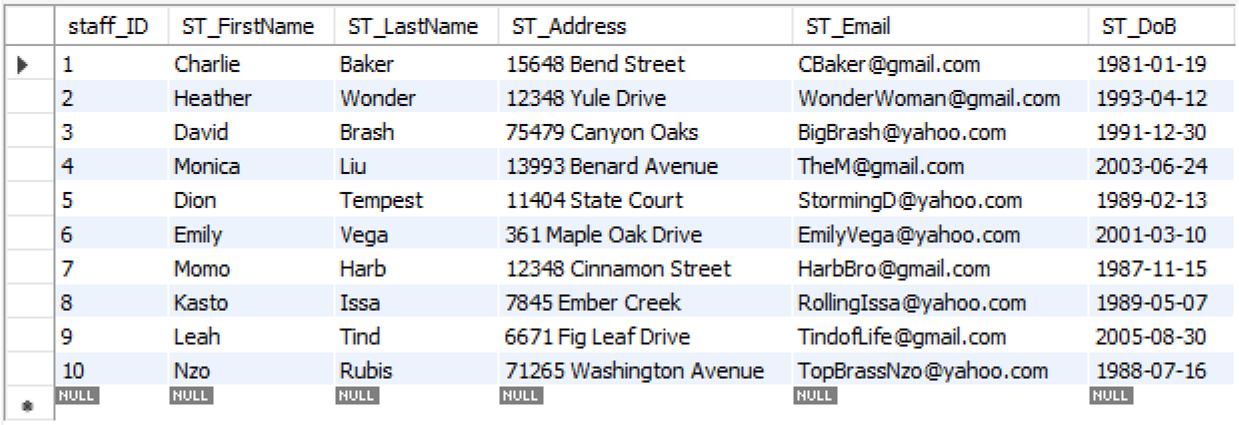
Below is the SQL code to create the stored procedure within the database followed by a call to that procedure and its before/after in the specified table.



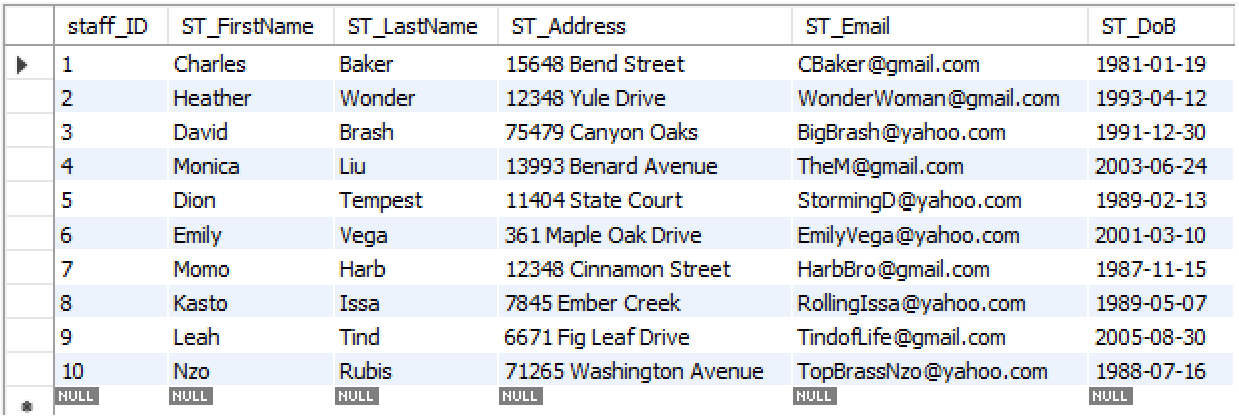
```sql
DELIMITER //
CREATE PROCEDURE update_staff(IN p_staff_ID INT, IN p_ST_FirstName VARCHAR(45),
                            IN p_ST_LastName VARCHAR(45), IN p_ST_Address VARCHAR(100),
                            IN p_ST_Email VARCHAR(80), IN p_ST_DoB DATE)
BEGIN
    Update staff
    SET ST_FirstName = COALESCE(p_ST_FirstName, ST_FirstName),
        ST_LastName = COALESCE(p_ST_LastName, ST_LastName),
        ST_Address = COALESCE(p_ST_Address,ST_Address),
        ST_Email = COALESCE(p_ST_Email,ST_Email),
        ST_DoB = COALESCE(p_ST_DoB,ST_DoB)
    WHERE staff_ID = p_staff_ID;
END
//
```



```sql
CALL update_staff(1, 'Charles', NULL, NULL, NULL, NULL);
```



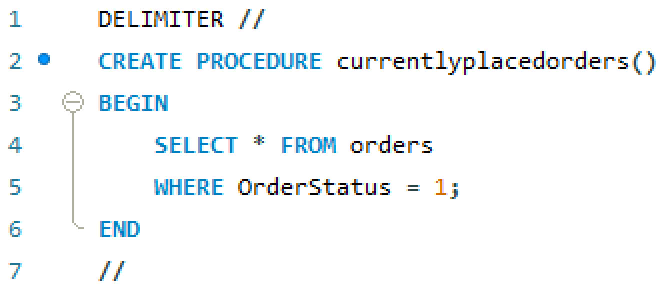
staff_ID	ST_FirstName	ST_LastName	ST_Address	ST_Email	ST_DoB
1	Charlie	Baker	15648 Bend Street	CBaker@gmail.com	1981-01-19
2	Heather	Wonder	12348 Yule Drive	WonderWoman@gmail.com	1993-04-12
3	David	Brash	75479 Canyon Oaks	BigBrash@yahoo.com	1991-12-30
4	Monica	Liu	13993 Benard Avenue	TheM@gmail.com	2003-06-24
5	Dion	Tempest	11404 State Court	StormingD@yahoo.com	1989-02-13
6	Emily	Vega	361 Maple Oak Drive	EmilyVega@yahoo.com	2001-03-10
7	Momo	Harb	12348 Cinnamon Street	HarbBro@gmail.com	1987-11-15
8	Kasto	Issa	7845 Ember Creek	RollingIssa@yahoo.com	1989-05-07
9	Leah	Tind	6671 Fig Leaf Drive	TindofLife@gmail.com	2005-08-30
10	Nzo	Rubis	71265 Washington Avenue	TopBrassNzo@yahoo.com	1988-07-16
NULL	NULL	NULL	NULL	NULL	NULL



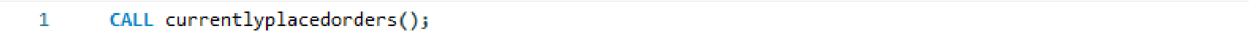
staff_ID	ST_FirstName	ST_LastName	ST_Address	ST_Email	ST_DoB
1	Charles	Baker	15648 Bend Street	CBaker@gmail.com	1981-01-19
2	Heather	Wonder	12348 Yule Drive	WonderWoman@gmail.com	1993-04-12
3	David	Brash	75479 Canyon Oaks	BigBrash@yahoo.com	1991-12-30
4	Monica	Liu	13993 Benard Avenue	TheM@gmail.com	2003-06-24
5	Dion	Tempest	11404 State Court	StormingD@yahoo.com	1989-02-13
6	Emily	Vega	361 Maple Oak Drive	EmilyVega@yahoo.com	2001-03-10
7	Momo	Harb	12348 Cinnamon Street	HarbBro@gmail.com	1987-11-15
8	Kasto	Issa	7845 Ember Creek	RollingIssa@yahoo.com	1989-05-07
9	Leah	Tind	6671 Fig Leaf Drive	TindofLife@gmail.com	2005-08-30
10	Nzo	Rubis	71265 Washington Avenue	TopBrassNzo@yahoo.com	1988-07-16
NULL	NULL	NULL	NULL	NULL	NULL

- currentlyplacedorders()

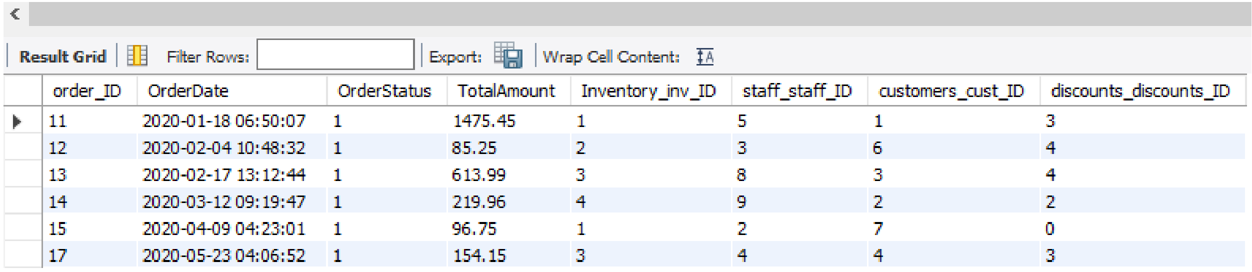
Below is the SQL code to create the stored procedure within the database followed by a call to that procedure and its output.



```
DELIMITER //
CREATE PROCEDURE currentlyplacedorders()
BEGIN
    SELECT * FROM orders
    WHERE OrderStatus = 1;
END
//
```



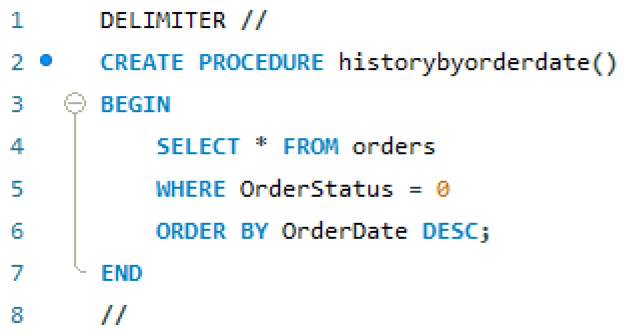
```
CALL currentlyplacedorders();
```



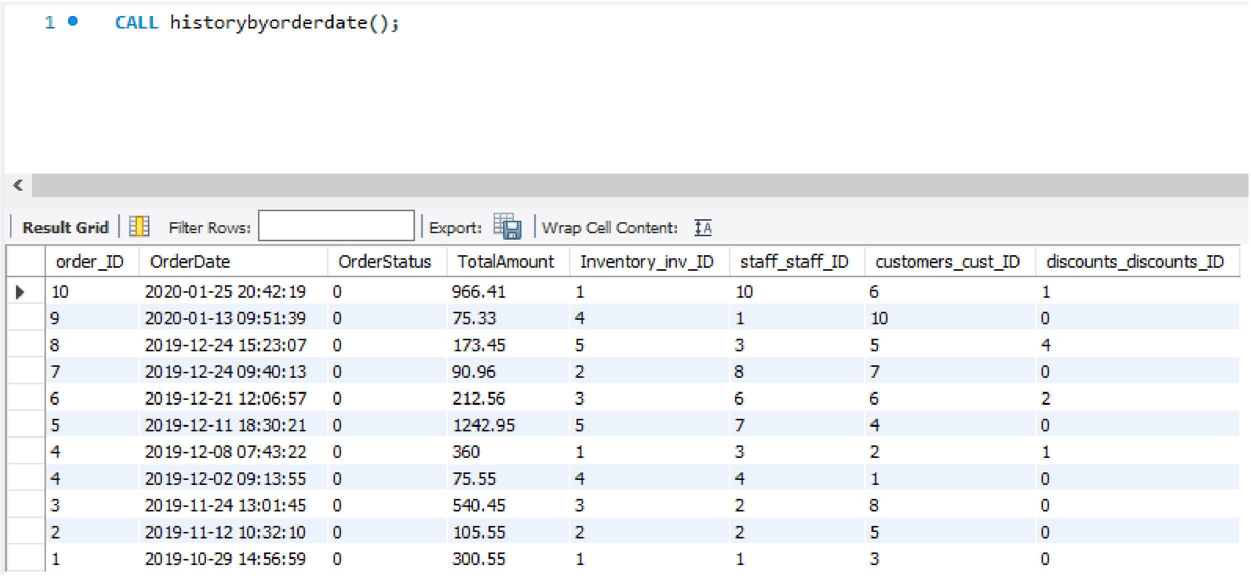
order_ID	OrderDate	OrderStatus	TotalAmount	Inventory_inv_ID	staff_staff_ID	customers_cust_ID	discounts_discounts_ID
11	2020-01-18 06:50:07	1	1475.45	1	5	1	3
12	2020-02-04 10:48:32	1	85.25	2	3	6	4
13	2020-02-17 13:12:44	1	613.99	3	8	3	4
14	2020-03-12 09:19:47	1	219.96	4	9	2	2
15	2020-04-09 04:23:01	1	96.75	1	2	7	0
17	2020-05-23 04:06:52	1	154.15	3	4	4	3

- historybyorderdate()

Below is the SQL code to create the stored procedure within the database followed by a call to that procedure and its output.



```sql
DELIMITER //
CREATE PROCEDURE historybyorderdate()
BEGIN
    SELECT * FROM orders
    WHERE OrderStatus = 0
    ORDER BY OrderDate DESC;
END
//
```

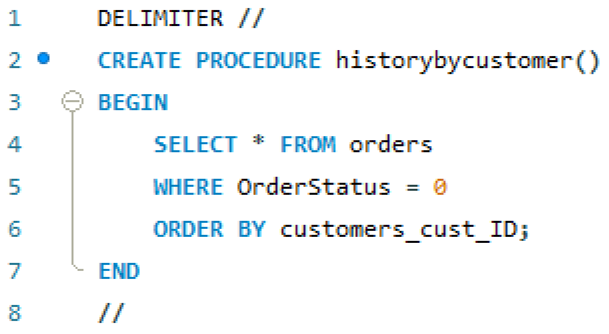


```sql
CALL historybyorderdate();
```

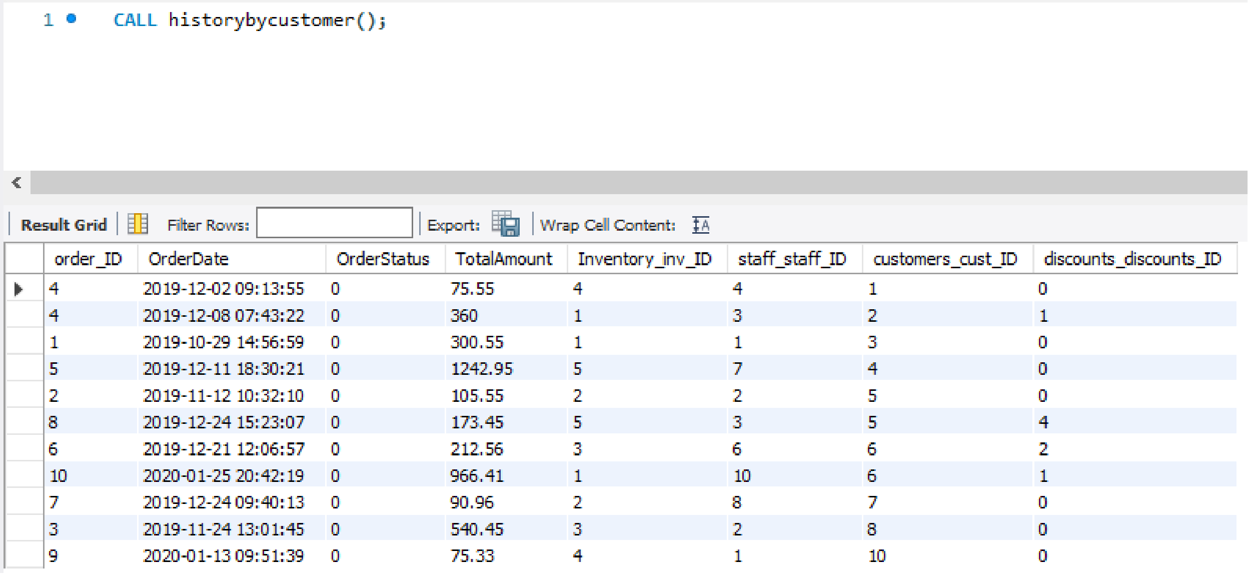
order_ID	OrderDate	OrderStatus	TotalAmount	Inventory_inv_ID	staff_staff_ID	customers_cust_ID	discounts_discounts_ID
10	2020-01-25 20:42:19	0	966.41	1	10	6	1
9	2020-01-13 09:51:39	0	75.33	4	1	10	0
8	2019-12-24 15:23:07	0	173.45	5	3	5	4
7	2019-12-24 09:40:13	0	90.96	2	8	7	0
6	2019-12-21 12:06:57	0	212.56	3	6	6	2
5	2019-12-11 18:30:21	0	1242.95	5	7	4	0
4	2019-12-08 07:43:22	0	360	1	3	2	1
4	2019-12-02 09:13:55	0	75.55	4	4	1	0
3	2019-11-24 13:01:45	0	540.45	3	2	8	0
2	2019-11-12 10:32:10	0	105.55	2	2	5	0
1	2019-10-29 14:56:59	0	300.55	1	1	3	0

- histroybycustomer()

Below is the SQL code to create the stored procedure within the database followed by a call to that procedure and its output.



```sql
DELIMITER //
CREATE PROCEDURE historybycustomer()
BEGIN
    SELECT * FROM orders
    WHERE OrderStatus = 0
    ORDER BY customers_cust_ID;
END
//
```

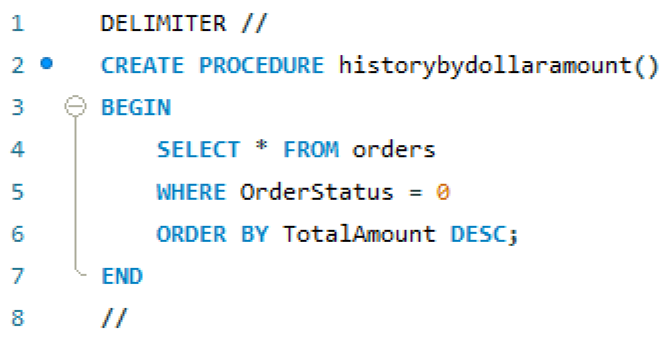


```sql
CALL historybycustomer();
```

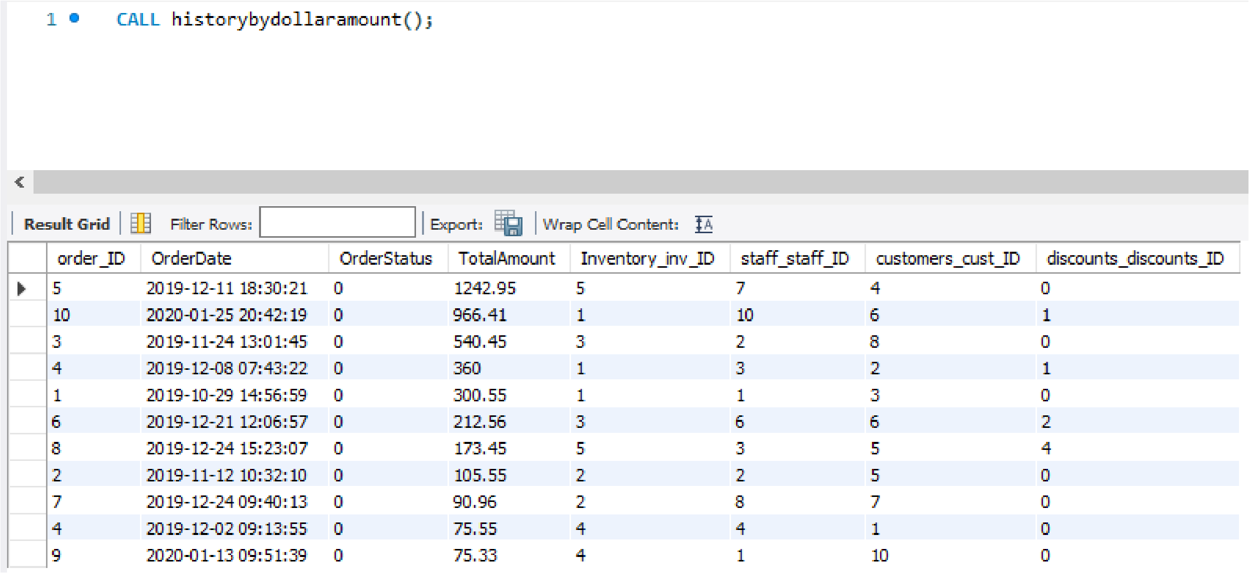
order_ID	OrderDate	OrderStatus	TotalAmount	Inventory_inv_ID	staff_staff_ID	customers_cust_ID	discounts_discounts_ID
4	2019-12-02 09:13:55	0	75.55	4	4	1	0
4	2019-12-08 07:43:22	0	360	1	3	2	1
1	2019-10-29 14:56:59	0	300.55	1	1	3	0
5	2019-12-11 18:30:21	0	1242.95	5	7	4	0
2	2019-11-12 10:32:10	0	105.55	2	2	5	0
8	2019-12-24 15:23:07	0	173.45	5	3	5	4
6	2019-12-21 12:06:57	0	212.56	3	6	6	2
10	2020-01-25 20:42:19	0	966.41	1	10	6	1
7	2019-12-24 09:40:13	0	90.96	2	8	7	0
3	2019-11-24 13:01:45	0	540.45	3	2	8	0
9	2020-01-13 09:51:39	0	75.33	4	1	10	0

- historybydollaramount()

Below is the SQL code to create the stored procedure within the database followed by a call to that procedure and its output.



```sql
DELIMITER //
CREATE PROCEDURE historybydollaramount()
BEGIN
    SELECT * FROM orders
    WHERE OrderStatus = 0
    ORDER BY TotalAmount DESC;
END
//
```



```sql
CALL historybydollaramount();
```

order_ID	OrderDate	OrderStatus	TotalAmount	Inventory_inv_ID	staff_staff_ID	customers_cust_ID	discounts_discounts_ID
5	2019-12-11 18:30:21	0	1242.95	5	7	4	0
10	2020-01-25 20:42:19	0	966.41	1	10	6	1
3	2019-11-24 13:01:45	0	540.45	3	2	8	0
4	2019-12-08 07:43:22	0	360	1	3	2	1
1	2019-10-29 14:56:59	0	300.55	1	1	3	0
6	2019-12-21 12:06:57	0	212.56	3	6	6	2
8	2019-12-24 15:23:07	0	173.45	5	3	5	4
2	2019-11-12 10:32:10	0	105.55	2	2	5	0
7	2019-12-24 09:40:13	0	90.96	2	8	7	0
4	2019-12-02 09:13:55	0	75.55	4	4	1	0
9	2020-01-13 09:51:39	0	75.33	4	1	10	0