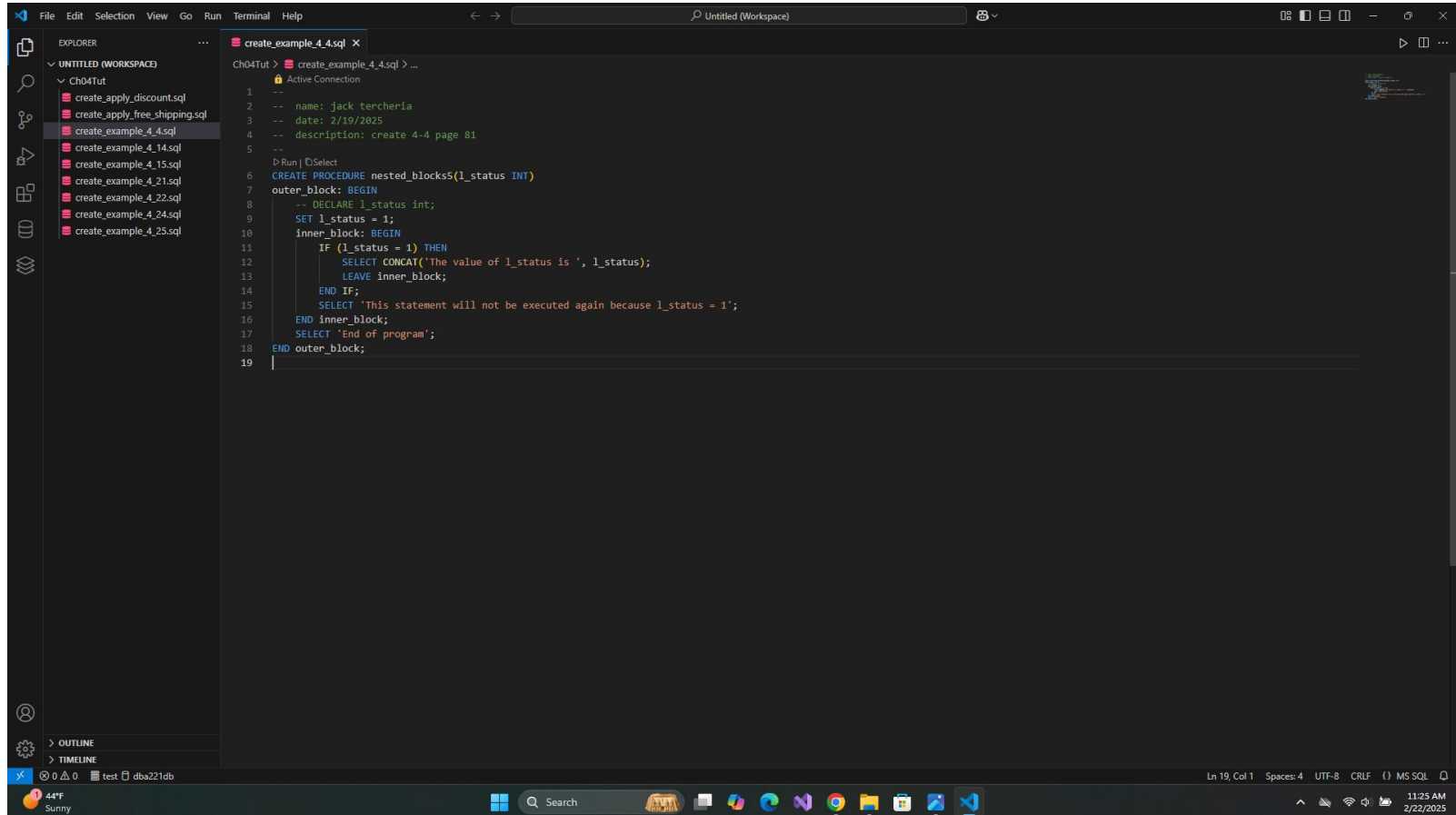


Ch 04 Tutorial

Jack Tercheria
2/22/2025

Create example 4-4



Call example 4-4

The screenshot displays a database IDE interface with a dark theme. On the left, a sidebar shows the database structure for 'dba221db'. The main area contains a SQL editor with a query that has been executed. Below the editor, a results pane shows the output of the query.

Database Structure (Left Sidebar):

- test 10.4.28-MariaDB
 - dba221db 3.08M
 - Query
 - Tables (14)
 - bind_example
 - books
 - creditcards 2000
 - customers 2534
 - departments 30
 - employees 2000
 - enum_test 4
 - limit_test 2
 - locations 30
 - max_sales_by_custom...
 - products 100
 - product_codes 6
 - sales 8256
 - timestamp_check 1001
 - Views
 - Functions (2)
 - f_discount_price (nomi...
 - f_title (in_gender char, in...
 - Procedures (10)
 - CustomerSales (in_cust...
 - GreetWorld
 - nested_blocks5 (l_statu...
 - operators
 - p1
 - p2
 - sp_demo_inout_param...
 - sp_demo_in_parameter...
 - sp_demo_out_paramet...
 - sp_emps_in_dept (in_e...

SQL Editor (Top):

```
create_example_4_4.sql dba221db.sql example_3_11.sql procedure-ddl.sql  
C:\Users\Jack>.dbclient>storage>1739290035056@@127.0.0.1@3306@dba221db> dba221db.sql > ...  
Run  
1 CALL nested_blocks5(0); 5ms
```

Results Pane (Bottom):

Result1 Result2 X

Search Results

End of program
varchar

End of program

Cost: 5ms < 1 > Total 1

System tray at the bottom shows the date and time: 9:35 AM 2/19/2025.

Create example apply free shipping

The screenshot displays the SQL Server Enterprise Manager interface. On the left, the 'DATABASE' tree shows the 'dba221db' database with various tables and procedures. The 'Procedures' folder is expanded, showing a list of procedures including 'apply_free_shipping'. The main window shows the SQL script for creating the 'apply_free_shipping' procedure. The script is as follows:

```
1 --  
2 -- name: jack tercheria  
3 -- date: 2/19/2025  
4 -- description apply_free_shipping  
5 --  
6 CREATE PROCEDURE apply_free_shipping(IN sale_id INT)  
7 BEGIN  
8     DECLARE free_shipping VARCHAR(50)  
9     DEFAULT 'Free shipping for sales id ';  
10  
11     SELECT free_shipping, sale_id;  
12 END
```

The script is executed, and the results are shown in the 'Results' pane. The results show a single row with the following values:

free_shipping	sale_id
Free shipping for sales id 2	2

The status bar at the bottom indicates the current line is 12, column 4, with 4 spaces, UTF-8 encoding, CRLF line endings, and MS SQL server.

Call example apply free shipping

The screenshot displays the SQL Server Enterprise Manager interface. On the left, the 'DATABASE' tree shows the 'test' database selected, with various tables and procedures listed. The 'Procedures (11)' folder is expanded, showing 'apply_free_shipping (s...)'. The main pane shows the execution of the 'CALL apply_free_shipping(2);' statement. The results pane at the bottom shows a single row with 'free_shipping' set to 'varchar' and 'sale_id' set to 'int', with a value of '2' for 'Free shipping for sales id'.

File Edit Selection View Go Run Terminal Help

create_apply_free_shipping.sql dba221db.sql create_variable_demo.sql example_3_3.sql procedure-ddl.sql create_example_4_4.sql

C:\Users\Jack> .dbclient > storage > 1739290035056@127.0.0.1@3306@dba221db > dba221db.sql > CALL apply_free_shipping(2);

```
1 CALL apply_free_shipping(2);
```

Result1 Result2

Search Results

free_shipping varchar sale_id int

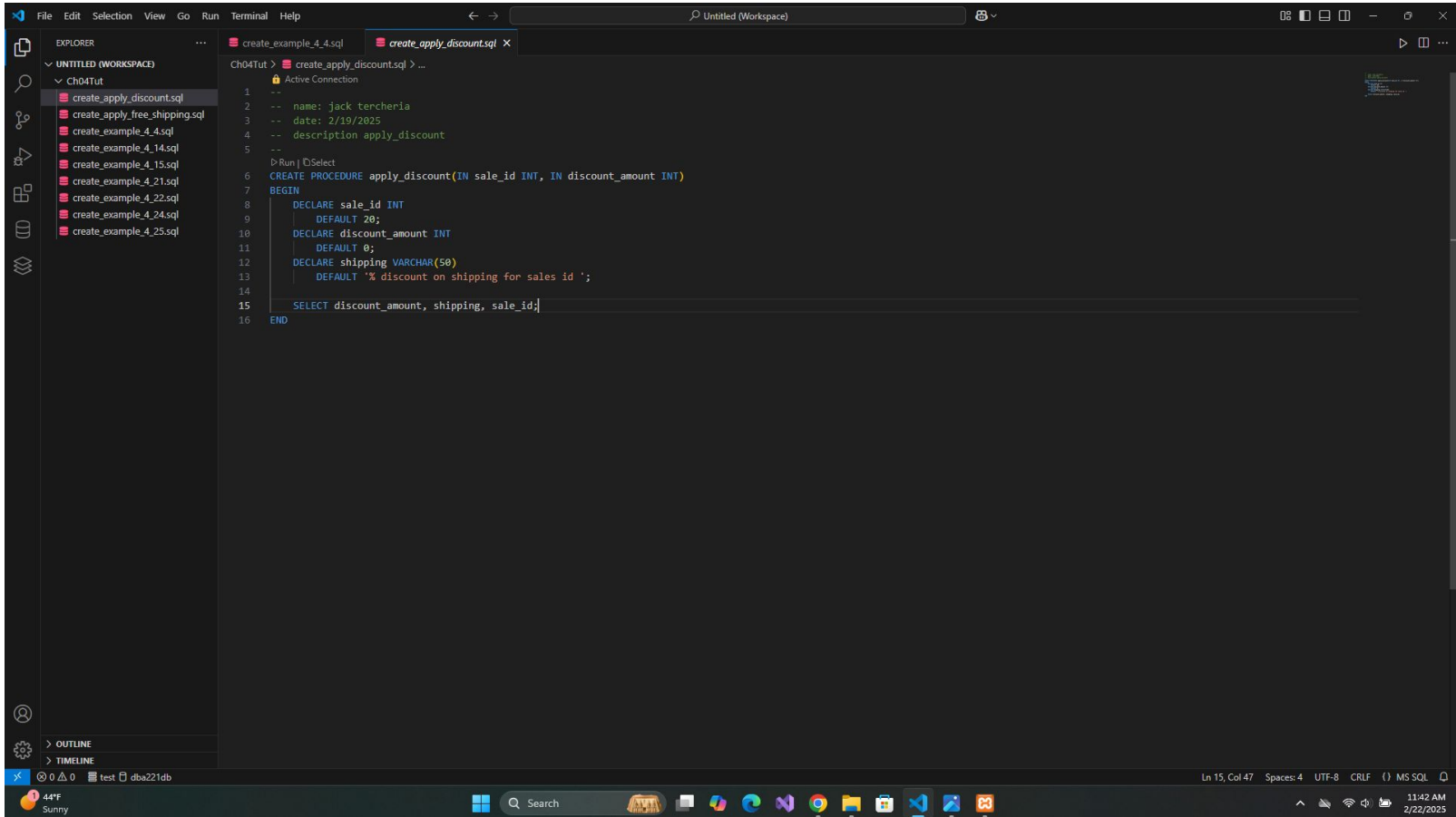
Free shipping for sales id 2

Cost: 2ms < 1 > Total 1

Ln 1, Col 1 Spaces: 4 UTF-8 CRLF {} MS SQL

33°F Light snow 10:01 AM 2/19/2025

Create example apply discount



Call example apply discount

The screenshot shows a database IDE with a dark theme. The left sidebar displays a database structure for 'test 104.28-MariaDB'. The main editor window shows a SQL script for a procedure named 'create_example_4_14.sql'. The script includes comments and a 'CREATE PROCEDURE' statement with nested IF logic to call 'apply_free_shipping' and 'apply_discount' based on customer status and sale value. The bottom panel shows the execution results for 'Result2', displaying a table with columns 'discount amount', 'shipping', and 'sale id'.

```
1 --
2 -- name: jack tercheria
3 -- date: 2/21/2025
4 -- description: create chapter 4 ex 14
5 --
6 CREATE PROCEDURE using_nested_if(IN sale_value INT)
7 BEGIN
8     DECLARE customer_status VARCHAR(50)
9         DEFAULT 'GOLD';
10    DECLARE discount_amount INT
11        DEFAULT 15;
12    DECLARE sale_id INT
13        DEFAULT 20;
14
15    IF (sale_value > 200) THEN
16        CALL apply_free_shipping(2);
17        IF (customer_status = 'PLATINUM') THEN
18            CALL apply_discount(sale_id,20);
19        ELSEIF (customer_status = 'GOLD') THEN
20            CALL apply_discount(sale_id,15);
21        ELSEIF (customer_status = 'SILVER') THEN
```

Result2

discount amount	shipping	sale id
15	% discount on shipping for	20

Create & Call example 4-14

The screenshot shows a database IDE with a dark theme. The left sidebar displays a database schema for 'test 10.4.28-MariaDB'. The main editor window shows the SQL script for creating and calling a stored procedure. The script is as follows:

```
1 --
2 -- name: jack tercheria
3 -- date: 2/21/2025
4 -- description: create chapter 4 ex 14
5 --
6 CREATE PROCEDURE using_nested_if(IN sale_value INT)
7 BEGIN
8     DECLARE customer_status VARCHAR(50)
9     DEFAULT 'GOLD';
10    DECLARE discount_amount INT
11    DEFAULT 15;
12    DECLARE sale_id INT
13    DEFAULT 20;
14
15    IF (sale_value > 200) THEN
16        CALL apply_free_shipping(2); -- free shipping
17        IF (customer_status = 'PLATINUM') THEN
18            CALL apply_discount(sale_id, 20); -- 20% discount
19        ELSEIF (customer_status = 'GOLD') THEN
20            CALL apply_discount(sale_id, 15); -- 15% discount
21        ELSEIF (customer_status = 'SILVER') THEN
```

Below the script, the 'Result' tab shows the output of the procedure call. The output is a table with three columns: 'discount_amount', 'shipping', and 'sale_id'. The first row shows a discount of 15, shipping of 20, and a sale ID of 20.

discount_amount	shipping	sale_id
15	% discount on shipping for	20

The bottom status bar indicates the current line is 27, column 51, with 4 spaces, UTF-8 encoding, and CRLF line endings. The system tray shows the date and time as 9:36 AM on 2/22/2025.

Create & Call example 4-15

The screenshot shows a SQL IDE interface with a dark theme. The left sidebar displays a database schema for 'dba221db'. The main editor window shows the SQL script for creating and calling a stored procedure. The script is as follows:

```
1 --
2 -- name: jack tercheria
3 -- date: 2/21/2025
4 -- description: create chapter 4 ex 15
5 --
6 CREATE PROCEDURE using_case(IN sale_value INT)
7 BEGIN
8     DECLARE customer_status VARCHAR(50)
9         DEFAULT 'GOLD';
10    DECLARE discount_amount INT
11        DEFAULT 15;
12    DECLARE sale_id INT
13        DEFAULT 20;
14
15    CASE customer_status
16        WHEN 'PLATINUM' THEN
17            CALL apply_discount(2,20); -- 20% discount
18        WHEN 'GOLD' THEN
19            CALL apply_discount(2,15); -- 15% discount
20        WHEN 'SILVER' THEN
21            CALL apply_discount(2,10); -- 10% discount
```

The script is executed, and the results are displayed in a table with the following columns: discount_amount, shipping, and sale_id. The results show a 15% discount on shipping for a sale ID of 20.

discount_amount	shipping	sale_id
15	% discount on shipping for	20

The bottom status bar indicates the current position in the script: Ln 13, Col 20, Spaces: 4, UTF-8, CRLF, MS SQL.

Create & Call example 4-21

The screenshot displays the SQL Server Enterprise Manager interface. On the left, the 'DATABASE' pane shows a tree view of the 'test' database, including tables like 'bind_example', 'books', 'creditcards', 'customers', 'departments', 'employees', 'enum_test', 'limit_test', 'locations', 'max_sales_by_custome...', 'products', 'product_codes', 'sales', and 'timestamp_check'. The 'Functions' pane shows 'f_discount_price' and 'f_title'. The 'Procedures' pane shows 'apply_discount', 'apply_free_shipping', 'CustomerSales', 'GreetWorld', 'nested_blocks5', 'operators', 'p1', 'p2', 'sp_demo_inout_param...', 'sp_demo_in_parameter...', 'sp_demo_out_paramet...', 'sp_emps_in_dept', 'using_case', 'using_iterate', and 'using_nested_if'. The main editor window shows the SQL script for 'create_example_4_21.sql'.

```
CREATE PROCEDURE using_iterate()
BEGIN
    DECLARE i INT
    DEFAULT 0;
    SET i=0;
    loop1: LOOP
        SET i = i + 1;
        IF i >= 10 THEN
            LEAVE loop1;
        ELSEIF MOD(i, 2) = 0 THEN
            ITERATE loop1;
        END IF;
        SELECT CONCAT(i, ' is an odd number');
    END LOOP loop1;
END
```

The script is executed, and the results pane shows the output of the stored procedure call:

```
CALL using_iterate(); 2ms
```

The results pane displays the output of the stored procedure call, showing the concatenation of the loop variable 'i' and the string ' is an odd number' for each iteration of the loop.

Result1
CONCAT(0, ' is an odd number')
CONCAT(1, ' is an odd number')
CONCAT(2, ' is an odd number')
CONCAT(3, ' is an odd number')
CONCAT(4, ' is an odd number')
CONCAT(5, ' is an odd number')
CONCAT(6, ' is an odd number')
CONCAT(7, ' is an odd number')
CONCAT(8, ' is an odd number')
CONCAT(9, ' is an odd number')

The status bar at the bottom indicates the current file is 'test dba221db' and the editor is in 'MS SQL' mode.

Create & Call example 4-22

The screenshot shows a database IDE with a dark theme. On the left, a sidebar displays a database structure for 'test 104.28-MariaDB'. Under 'Tables (14)', several tables are listed, including 'books', 'creditcards', 'customers', 'departments', 'employees', 'enum_test', 'limit_test', 'locations', 'max_sales_by_custome...', 'products', 'product_codes', 'sales', and 'timestamp_check'. Under 'Procedures (18)', various stored procedures are listed, including 'apply_discount', 'apply_free_shipping', 'CustomerSales', 'GreetWorld', 'nested_blocks5', 'nested_loops', 'operators', 'p1', 'p2', 'sp_demo_inout_param...', 'sp_demo_in_parameter...', 'sp_demo_out_parameter...', 'sp_emps_in_dept', 'using_case', 'using_iterate', 'using_nested_if', 'using_repeat', and 'using_while_less'.

The main editor window is titled 'create_example_4_22.sql' and contains the following SQL code:

```
1 --
2 -- name: jack tercheria
3 -- date: 2/22/2025
4 -- description: create chapter 4 ex 22
5 --
6
7 CREATE PROCEDURE using_repeat()
8 BEGIN
9     DECLARE i INT
10     DEFAULT 0;
11     SET i=0;
12     loop1: REPEAT
13         SET i = i + 1;
14         IF MOD(i,2)<>0 THEN          -- Even number - try AGAINST
15             SELECT CONCAT(i, ' is an odd number.');

Below the code editor, the 'Run' button is highlighted. The 'Results' pane shows the output of the procedure execution:



```
Result1 | Result2 | Result3 | Result4 | Result5
+-----+
Search Results
Q CONCAT(i, ' is an odd
 varchar
> 9 is an odd number.
```



The status bar at the bottom indicates 'Ln 18, Col 16 Spaces: 4 UTF-8 CR LF MS SQL'.


```

Create & Call example 4-24

The screenshot displays a MySQL IDE interface with a dark theme. On the left, a sidebar shows a database structure for 'test 104.28-MariaDB'. Under 'Tables (14)', various tables are listed, including 'books', 'creditcards', 'customers', 'departments', 'employees', 'enum_test', 'limit_test', 'locations', 'max_sales_by_custome...', 'products', 'product_codes', 'sales', and 'timestamp_check'. Under 'Functions (2)', 'f_discount_price' and 'f_title' are listed. Under 'Procedures (18)', several procedures are listed, including 'apply_discount', 'apply_free_shipping', 'CustomerSales', 'GreetWorld', 'nested_blocks5', 'nested_loops', 'operators', 'p1', 'p2', 'sp_demo_inout_param...', 'sp_demo_in_parameter...', 'sp_demo_out_param...', 'sp_emps_in_dept', 'using_case', 'using_iterate', 'using_nested_if', 'using_repeat', and 'using_while_less'. The main editor window shows a SQL script in 'create_example_4_24.sql'. The script starts with a comment 'Ch04Tut > create_example_4_24.sql > --' and includes a date 'date: 2/22/2025' and a description 'description: create chapter 4 ex 24'. The script then declares a procedure 'using_while_less()' and defines its logic using a loop and an if statement. The procedure is called with 'CALL using_while_less();'. The bottom panel shows the results of the execution, with a single row containing the text '9 is an odd number.'.

```
Ch04Tut > create_example_4_24.sql > --
1 -- date: 2/22/2025
2 --
3 -- description: create chapter 4 ex 24
4 --
5 --
6
7 > Run | DSelect
8 CREATE PROCEDURE using_while_less()
9 BEGIN
10     DECLARE i INT
11     | DEFAULT 0;
12     SET i=0;
13     loop1: WHILE i <= 10 DO
14         IF MOD(i,2)<>0 THEN -- Even number- try again
15             SELECT CONCAT(i, ' is an odd number.');

Result1 Result2 Result3 Result4 Result5 X



Search Results



CONCAT(i, ' is an odd  
varchar



> 9 is an odd number.


```

Create & Call example 4-25

The screenshot shows a SQL IDE interface with a dark theme. On the left, a database explorer shows a connection to 'test 10.4.28-MariaDB' with a database 'dba221db'. The 'Tables (14)' folder is expanded, listing various tables like 'books', 'creditcards', 'customers', etc. The main editor window is titled 'create_example_4_25.sql' and contains the following SQL code:

```
CH04Tut > create_example_4_25.sql > --
1  --
2  -- name: jack tercheria
3  -- date: 2/22/2025
4  -- description: create chapter 4 ex 25
5  --
6
7  D:\Run | D:\Select
8  CREATE PROCEDURE nested_loops()
9  BEGIN
10     DECLARE i, j INT
11     DEFAULT 1;
12     SET j = 1;
13     loop1: WHILE i < 12 DO
14         loop2: WHILE j < 12 DO
15             SELECT CONCAT(i, ' times ', j, ' is ', i * j);
16             SET i = i + 1;
17             SET j = j + 1;
18         END WHILE loop2;
19     END WHILE loop1;
20 END
```

Below the code editor, the 'D:\Run' button is visible. The results pane shows the output of the procedure execution:

Result
11 times 11 is 121

The status bar at the bottom indicates 'Ln 1, Col 1 Spaces: 4 UTF-8 CRLF () MS SQL'.