INSTITUTO TECNOLÓGICO SUPERIOR DE JEREZ

INGENIERÍA EN SISTEMAS COMPUTACIONALES

TALLER DE BASES DE DATOS

9° SEMESTRE

I.S.C. SALVADOR ACEVEDO SANDOVAL

"STORED PROCEDURES & FUNCTIONS"

ALBAR DE LA TORRE GARCÍA

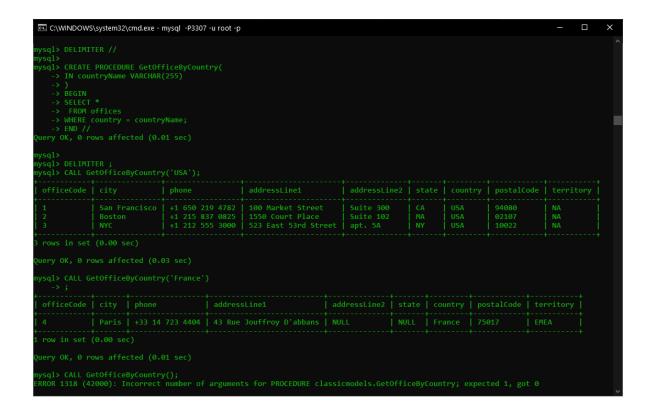No. Control: 16070122

Correo: albar00@hotmail.com

JEREZ ZACATECAS

10 DE DICIEMBRE DEL 2020

```
C:\WINDOWS\system32\cmd.exe - mysql  -P3307 -u root -p                    —    □    ×

mysql> USE classicmodels;
Database changed
mysql> SELECT
    ->     customerName,
    ->     city,
    ->     state,
    ->     postalCode,
    ->     country
    -> FROM
    ->     customers
    -> ORDER BY customerName;
+----------------------------------+-----------------+------------+------------+-------------+
| customerName                     | city            | state      | postalCode | country     |
+----------------------------------+-----------------+------------+------------+-------------+
| Alpha Cognac                     | Toulouse        | NULL       | 31000      | France      |
| American Souvenirs Inc           | New Haven       | CT         | 97823      | USA         |
| Amica Models & Co.               | Torino          | NULL       | 10100      | Italy       |
| ANG Resellers                    | Madrid          | NULL       | 28001      | Spain       |
| Anna's Decorations, Ltd          | North Sydney    | NSW        | 2060       | Australia   |
| Anton Designs, Ltd.              | Madrid          | NULL       | 28023      | Spain       |
| Asian Shopping Network, Co       | Singapore       | NULL       | 038988     | Singapore   |
| Asian Treasures, Inc.            | Cork            | Co. Cork   | NULL       | Ireland     |
| Atelier graphique                | Nantes          | NULL       | 44000      | France      |
| Australian Collectables, Ltd     | Glen Waverly    | Victoria   | 3150       | Australia   |
| Australian Collectors, Co.       | Melbourne       | Victoria   | 3004       | Australia   |
| Australian Gift Network, Co      | South Brisbane  | Queensland | 4101       | Australia   |
```

```
C:\WINDOWS\system32\cmd.exe - mysql  -P3307 -u root -p                    —    □    ×

mysql> DELIMITER $$
mysql>
mysql> CREATE PROCEDURE GetCustomers()
    -> BEGIN
    -> SELECT
    -> customerName,
    -> city,
    -> state,
    -> postalCode,
    -> country
    -> FROM
    -> customers
    -> ORDER BY customerName;
    -> END$$
Query OK, 0 rows affected (0.09 sec)

mysql> DELIMITER ;
mysql> CALL GetCustomers();
+----------------------------------+-----------------+------------+------------+-------------+
| customerName                     | city            | state      | postalCode | country     |
+----------------------------------+-----------------+------------+------------+-------------+
| Alpha Cognac                     | Toulouse        | NULL       | 31000      | France      |
| American Souvenirs Inc           | New Haven       | CT         | 97823      | USA         |
| Amica Models & Co.               | Torino          | NULL       | 10100      | Italy       |
| ANG Resellers                    | Madrid          | NULL       | 28001      | Spain       |
| Anna's Decorations, Ltd          | North Sydney    | NSW        | 2060       | Australia   |
```

```
C:\WINDOWS\system32\cmd.exe - mysql  -P3307 -u root -p                    —    □    ×


mysql> DELIMITER ;
mysql> DELIMITER $$
mysql>
mysql> CREATE PROCEDURE sp_name()
    -> BEGIN
    ->   -- statements
    -> END $$
Query OK, 0 rows affected (0.00 sec)

mysql>
mysql> DELIMITER ;
mysql>
```

```
Seleccionar C:\WINDOWS\system32\cmd.exe - mysql  -P3307 -u root -p

mysql> DELIMITER ;
mysql> DELIMITER //
mysql>
mysql> CREATE PROCEDURE GetAllProducts()
    -> BEGIN
    -> SELECT *  FROM products;
    -> END //
Query OK, 0 rows affected (0.00 sec)

mysql>
mysql> DELIMITER ;
mysql> CALL GetAllProducts();
+--------------+-------------------+------------------+-------------+-----------
------------------+----------
----------------------------------------------------------------------
----------------------------------------------------------------------
----------------------------------------------------------------------
-------------------------------------------------------+-------------
--+-----------+-------+
| productCode | productName                          | productLine    | productScale | productVen
dor             | productDescription
```

```
C:\WINDOWS\system32\cmd.exe - mysql  -P3307 -u root -p

mysql>
mysql>
mysql> DELIMITER $$
mysql>
mysql> CREATE PROCEDURE GetEmployees()
    -> BEGIN
    ->     SELECT
    ->         firstName,
    ->         lastName,
    ->         city,
    ->         state,
    ->         country
    ->     FROM employees
    ->     INNER JOIN offices using (officeCode);
    ->
    -> END$$
Query OK, 0 rows affected (0.00 sec)

mysql>
mysql> DELIMITER ;
mysql> DROP PROCEDURE GetEmployees;
Query OK, 0 rows affected (0.01 sec)

mysql>
```

```
C:\WINDOWS\system32\cmd.exe - mysql  -P3307 -u root -p

mysql> DROP PROCEDURE abc;
ERROR 1305 (42000): PROCEDURE classicmodels.abc does not exist
mysql> DROP PROCEDURE IF EXISTS abc;
Query OK, 0 rows affected, 1 warning (0.00 sec)

mysql> SHOW WARNINGS;
+-------+------+--------------------------------------------+
| Level | Code | Message                                    |
+-------+------+--------------------------------------------+
| Note  | 1305 | PROCEDURE classicmodels.abc does not exist |
+-------+------+--------------------------------------------+
1 row in set (0.00 sec)

mysql>
```

```
C:\WINDOWS\system32\cmd.exe - mysql  -P3307 -u root -p                      —    □    ×

mysql> DELIMITER $$
mysql>
mysql> CREATE PROCEDURE GetPayments()
    -> BEGIN
    ->     SELECT
    ->         customerName,
    ->         checkNumber,
    ->         paymentDate,
    ->         amount
    ->     FROM payments
    ->     INNER JOIN customers
    ->         using (customerNumber);
    -> END$$
Query OK, 0 rows affected (0.00 sec)

mysql>
mysql> DELIMITER ;
mysql> DROP PROCEDURE IF EXISTS GetPayments;
Query OK, 0 rows affected (0.01 sec)

mysql>
```

```
C:\WINDOWS\system32\cmd.exe - mysql  -P3307 -u root -p                      —    □    ×

mysql> DELIMITER //
mysql>
mysql> CREATE PROCEDURE GetOfficeByCountry(
    -> IN countryName VARCHAR(255)
    -> )
    -> BEGIN
    -> SELECT *
    ->  FROM offices
    -> WHERE country = countryName;
    -> END //
Query OK, 0 rows affected (0.01 sec)

mysql>
mysql> DELIMITER ;
mysql> CALL GetOfficeByCountry('USA');
+------------+---------------+-----------------+------------------+-------------+-------+---------+------------+-----------+
| officeCode | city          | phone           | addressLine1     | addressLine2 | state | country | postalCode | territory |
+------------+---------------+-----------------+------------------+-------------+-------+---------+------------+-----------+
| 1          | San Francisco | +1 650 219 4782 | 100 Market Street | Suite 300   | CA    | USA     | 94080      | NA        |
| 2          | Boston        | +1 215 837 0825 | 1550 Court Place | Suite 102   | MA    | USA     | 02107      | NA        |
| 3          | NYC           | +1 212 555 3000 | 523 East 53rd Street | apt. 5A  | NY    | USA     | 10022      | NA        |
+------------+---------------+-----------------+------------------+-------------+-------+---------+------------+-----------+
3 rows in set (0.00 sec)

Query OK, 0 rows affected (0.03 sec)

mysql> CALL GetOfficeByCountry('France')
    -> ;
+------------+-------+-----------------+---------------------------+-------------+-------+---------+------------+-----------+
| officeCode | city  | phone           | addressLine1              | addressLine2 | state | country | postalCode | territory |
+------------+-------+-----------------+---------------------------+-------------+-------+---------+------------+-----------+
| 4          | Paris | +33 14 723 4404 | 43 Rue Jouffroy D'abbans  | NULL        | NULL  | France  | 75017      | EMEA      |
+------------+-------+-----------------+---------------------------+-------------+-------+---------+------------+-----------+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.01 sec)

mysql> CALL GetOfficeByCountry();
ERROR 1318 (42000): Incorrect number of arguments for PROCEDURE classicmodels.GetOfficeByCountry; expected 1, got 0
```

```
C:\WINDOWS\system32\cmd.exe - mysql  -P3307 -u root -p                                      —  □  ×

mysql> DELIMITER $$
mysql>
mysql> CREATE PROCEDURE GetOrderCountByStatus (
    -> IN  orderStatus VARCHAR(25),
    -> OUT total INT
    -> )
    -> BEGIN
    -> SELECT COUNT(orderNumber)
    -> INTO total
    -> FROM orders
    -> WHERE status = orderStatus;
    -> END$$
Query OK, 0 rows affected (0.01 sec)

mysql>
mysql> DELIMITER ;
mysql> CALL GetOrderCountByStatus('Shipped',@total);
Query OK, 1 row affected (0.01 sec)

mysql> SELECT @total;
+--------+
| @total |
+--------+
|    303 |
+--------+
1 row in set (0.00 sec)

mysql>
```

```
C:\WINDOWS\system32\cmd.exe - mysql  -P3307 -u root -p                                      —  □  ×

    -> END$$
Query OK, 0 rows affected (0.01 sec)

mysql>
mysql> DELIMITER ;
mysql> CALL GetOrderCountByStatus('Shipped',@total);
Query OK, 1 row affected (0.01 sec)

mysql> SELECT @total;
+--------+
| @total |
+--------+
|    303 |
+--------+
1 row in set (0.00 sec)

mysql> CALL GetOrderCountByStatus('in process',@total);
Query OK, 1 row affected (0.00 sec)

mysql> SELECT @total AS  total_in_process;
+------------------+
| total_in_process |
+------------------+
|                6 |
+------------------+
1 row in set (0.00 sec)

mysql>
```

```
C:\WINDOWS\system32\cmd.exe - mysql  -P3307 -u root -p                        —    □    ×

mysql> DELIMITER $$
mysql>
mysql> CREATE PROCEDURE SetCounter(
    -> INOUT counter INT,
    ->    IN inc INT
    -> )
    -> BEGIN
    -> SET counter = counter + inc;
    -> END$$
Query OK, 0 rows affected (0.01 sec)

mysql>
mysql> DELIMITER ;
mysql> SET @counter = 1;
Query OK, 0 rows affected (0.00 sec)

mysql> CALL SetCounter(@counter,1); -- 2
Query OK, 0 rows affected (0.00 sec)

mysql> CALL SetCounter(@counter,1); -- 3
Query OK, 0 rows affected (0.00 sec)

mysql> CALL SetCounter(@counter,5); -- 8
Query OK, 0 rows affected (0.00 sec)

mysql> SELECT @counter; -- 8
+----------+
| @counter |
+----------+
|        8 |
+----------+
1 row in set (0.00 sec)

mysql>
```

```
C:\WINDOWS\system32\cmd.exe - mysql  -P3307 -u root -p                        —    □    ×

mysql> DELIMITER $$
mysql>
mysql> CREATE PROCEDURE GetOrderAmount(
    -> IN pOrderNumber INT
    -> )
    -> BEGIN
    -> SELECT
    ->  SUM(quantityOrdered * priceEach)
    -> FROM orderDetails
    -> WHERE orderNumber = pOrderNumber;
    -> END$$
Query OK, 0 rows affected (0.00 sec)

mysql>
mysql> DELIMITER ;
```

```
line 1
mysql>
mysql> SELECT COUNT(*)
    -> INTO productCount
    -> FROM products;
ERROR 1327 (42000): Undeclared variable: productCount
mysql> DELIMITER $$
mysql>
mysql> CREATE PROCEDURE GetTotalOrder()
    -> BEGIN
    -> DECLARE totalOrder INT DEFAULT 0;
    ->
    ->     SELECT COUNT(*)
    ->     INTO totalOrder
    ->     FROM orders;
    ->
    ->     SELECT totalOrder;
    -> END$$
Query OK, 0 rows affected (0.01 sec)

mysql>
mysql> DELIMITER ;
mysql> DECLARE totalOrder INT DEFAULT 0;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to yo
ur MySQL server version for the right syntax to use near 'DECLARE totalOrder INT DEFAULT 0' at li
ne 1
mysql> SELECT COUNT(*)
    -> INTO totalOrder
    -> FROM orders;
ERROR 1327 (42000): Undeclared variable: totalOrder
mysql>
```

```
mysql> CALL GetTotalOrder();
+------------+
| totalOrder |
+------------+
|        326 |
+------------+
1 row in set (0.01 sec)

Query OK, 0 rows affected (0.01 sec)

mysql>
```

```
C:\WINDOWS\system32\cmd.exe - mysql  -P3307 -u root -p

mysql> SHOW PROCEDURE STATUS WHERE search_condition;
ERROR 1054 (42S22): Unknown column 'search_condition' in 'where clause'
mysql> SHOW PROCEDURE STATUS WHERE db = 'classicmodels';
+---------------+----------------------+-----------+----------------+---------------------+---------------------+---------
--------+---------+----------------------+----------------------+--------------------+
| Db            | Name                 | Type      | Definer        | Modified            | Created             | Securi
ty_type | Comment | character_set_client | collation_connection | Database Collation |
+---------------+----------------------+-----------+----------------+---------------------+---------------------+---------
--------+---------+----------------------+----------------------+--------------------+
| classicmodels | GetAllProducts       | PROCEDURE | root@localhost | 2020-12-09 16:05:28 | 2020-12-09 16:05:28 | DEFINE
R       |         | cp850                | cp850_general_ci     | latin1_swedish_ci  |
| classicmodels | GetCustomers         | PROCEDURE | root@localhost | 2020-12-09 15:59:02 | 2020-12-09 15:59:02 | DEFINE
R       |         | utf8                 | utf8_general_ci      | latin1_swedish_ci  |
| classicmodels | GetOfficeByCountry   | PROCEDURE | root@localhost | 2020-12-09 16:15:34 | 2020-12-09 16:15:34 | DEFINE
R       |         | cp850                | cp850_general_ci     | latin1_swedish_ci  |
| classicmodels | GetOrderAmount       | PROCEDURE | root@localhost | 2020-12-09 16:36:21 | 2020-12-09 16:36:21 | DEFINE
R       |         | cp850                | cp850_general_ci     | latin1_swedish_ci  |
| classicmodels | GetOrderCountByStatus | PROCEDURE | root@localhost | 2020-12-09 16:26:11 | 2020-12-09 16:26:11 | DEFINE
R       |         | cp850                | cp850_general_ci     | latin1_swedish_ci  |
| classicmodels | GetTotalOrder        | PROCEDURE | root@localhost | 2020-12-09 16:38:08 | 2020-12-09 16:38:08 | DEFINE
R       |         | cp850                | cp850_general_ci     | latin1_swedish_ci  |
| classicmodels | SetCounter           | PROCEDURE | root@localhost | 2020-12-09 16:27:56 | 2020-12-09 16:27:56 | DEFINE
R       |         | cp850                | cp850_general_ci     | latin1_swedish_ci  |
| classicmodels | sp_name              | PROCEDURE | root@localhost | 2020-12-09 16:03:05 | 2020-12-09 16:03:05 | DEFINE
R       |         | cp850                | cp850_general_ci     | latin1_swedish_ci  |
+---------------+----------------------+-----------+----------------+---------------------+---------------------+---------
--------+---------+----------------------+----------------------+--------------------+
8 rows in set (0.00 sec)

mysql>
```

```
C:\WINDOWS\system32\cmd.exe - mysql  -P3307 -u root -p

8 rows in set (0.00 sec)

mysql> SHOW PROCEDURE STATUS LIKE '%pattern%'
    -> ;
Empty set (0.00 sec)

mysql> SHOW PROCEDURE STATUS LIKE '%Order%';
+---------------+----------------------+-----------+----------------+---------------------+---------------------+---------
--------+---------+----------------------+----------------------+--------------------+
| Db            | Name                 | Type      | Definer        | Modified            | Created             | Securi
ty_type | Comment | character_set_client | collation_connection | Database Collation |
+---------------+----------------------+-----------+----------------+---------------------+---------------------+---------
--------+---------+----------------------+----------------------+--------------------+
| classicmodels | GetOrderAmount       | PROCEDURE | root@localhost | 2020-12-09 16:36:21 | 2020-12-09 16:36:21 | DEFINE
R       |         | cp850                | cp850_general_ci     | latin1_swedish_ci  |
| classicmodels | GetOrderCountByStatus | PROCEDURE | root@localhost | 2020-12-09 16:26:11 | 2020-12-09 16:26:11 | DEFINE
R       |         | cp850                | cp850_general_ci     | latin1_swedish_ci  |
| classicmodels | GetTotalOrder        | PROCEDURE | root@localhost | 2020-12-09 16:38:08 | 2020-12-09 16:38:08 | DEFINE
R       |         | cp850                | cp850_general_ci     | latin1_swedish_ci  |
+---------------+----------------------+-----------+----------------+---------------------+---------------------+---------
--------+---------+----------------------+----------------------+--------------------+
3 rows in set (0.00 sec)

mysql>
```

```
3 rows in set (0.00 sec)

mysql> SELECT
    ->      routine_name
    -> FROM
    ->      information_schema.routines
    -> WHERE
    ->      routine_type = 'PROCEDURE'
    ->          AND routine_schema = '<database_name>';
Empty set (0.00 sec)

mysql> SELECT
    ->      routine_name
    -> FROM
    ->      information_schema.routines
    -> WHERE
    ->      routine_type = 'PROCEDURE'
    ->          AND routine_schema = 'classicmodels';
+----------------------+
| ROUTINE_NAME         |
+----------------------+
| GetAllProducts       |
| GetCustomers         |
| GetOfficeByCountry   |
| GetOrderAmount       |
| GetOrderCountByStatus|
| GetTotalOrder        |
| SetCounter           |
| sp_name              |
+----------------------+
8 rows in set (0.00 sec)

mysql>
```

```
mysql>
mysql> CREATE PROCEDURE GetCustomerLevel(
    ->     IN  pCustomerNumber INT,
    ->     OUT pCustomerLevel  VARCHAR(20))
    -> BEGIN
    ->     DECLARE credit DECIMAL(10,2) DEFAULT 0;
    ->
    ->     SELECT creditLimit
    ->     INTO credit
    ->     FROM customers
    ->     WHERE customerNumber = pCustomerNumber;
    ->
    ->     IF credit > 50000 THEN
    ->         SET pCustomerLevel = 'PLATINUM';
    ->     END IF;
    -> END$$
Query OK, 0 rows affected (0.08 sec)

mysql>
mysql> DELIMITER ;
mysql> SELECT
    ->      customerNumber,
    ->      creditLimit
    -> FROM
    ->      customers
    -> WHERE
    ->      creditLimit > 50000
    -> ORDER BY
    ->      creditLimit DESC;
+----------------+-------------+
| customerNumber | creditLimit |
+----------------+-------------+
|            141 |   227600.00 |
|            124 |   210500.00 |
|            298 |   141300.00 |
|            151 |   138500.00 |
|            187 |   136800.00 |
|            146 |   123900.00 |
|            286 |   123700.00 |
|            386 |   121400.00 |
|            227 |   120800.00 |
|            259 |   120400.00 |
|            278 |   119600.00 |
```

```
C:\WINDOWS\system32\cmd.exe - mysql  -P3307 -u root -p                    —    □    ✕

mysql> CALL GetCustomerLevel(141, @level);
Query OK, 1 row affected (0.00 sec)

mysql> SELECT @level;
+----------+
| @level   |
+----------+
| PLATINUM |
+----------+
1 row in set (0.00 sec)

mysql>
```

```
C:\WINDOWS\system32\cmd.exe - mysql  -P3307 -u root -p                    —    □    ✕

mysql> DROP PROCEDURE GetCustomerLevel;
Query OK, 0 rows affected (0.01 sec)

mysql> DELIMITER $$
mysql>
mysql> CREATE PROCEDURE GetCustomerLevel(
    ->     IN  pCustomerNumber INT,
    ->     OUT pCustomerLevel  VARCHAR(20))
    -> BEGIN
    ->     DECLARE credit DECIMAL DEFAULT 0;
    ->
    ->     SELECT creditLimit
    ->     INTO credit
    ->     FROM customers
    ->     WHERE customerNumber = pCustomerNumber;
    ->
    ->     IF credit > 50000 THEN
    ->         SET pCustomerLevel = 'PLATINUM';
    ->     ELSE
    ->         SET pCustomerLevel = 'NOT PLATINUM';
    ->     END IF;
    -> END$$
Query OK, 0 rows affected (0.08 sec)

mysql>
mysql> DELIMITER ;
mysql>
```

```
mysql>
mysql> DELIMITER ;
mysql> SELECT
    ->     customerNumber,
    ->     creditLimit
    -> FROM
    ->     customers
    -> WHERE
    ->     creditLimit <= 50000
    -> ORDER BY
    ->     creditLimit DESC;
+----------------+-------------+
| customerNumber | creditLimit |
+----------------+-------------+
|            447 |    49700.00 |
|            233 |    48700.00 |
|            452 |    45300.00 |
|            173 |    43400.00 |
|            489 |    43300.00 |
|            328 |    43000.00 |
|            362 |    41900.00 |
|            456 |    39800.00 |
|            473 |    34800.00 |
|            381 |    23500.00 |
|            198 |    23000.00 |
|            103 |    21000.00 |
|            219 |    11000.00 |
|            125 |        0.00 |
```

```
|            477 |        0.00 |
|            480 |        0.00 |
|            481 |        0.00 |
|            348 |        0.00 |
+----------------+-------------+
37 rows in set (0.00 sec)

mysql> CALL GetCustomerLevel(447, @level);
Query OK, 1 row affected (0.00 sec)

mysql> SELECT @level;
+-------------+
| @level      |
+-------------+
| NOT PLATINUM |
+-------------+
1 row in set (0.00 sec)

mysql>
```

```
C:\WINDOWS\system32\cmd.exe - mysql  -P3307 -u root -p                    —   □   ✕

mysql> DELIMITER $$
mysql>
mysql> CREATE PROCEDURE GetCustomerLevel(
    ->     IN  pCustomerNumber INT,
    ->     OUT pCustomerLevel  VARCHAR(20))
    -> BEGIN
    ->     DECLARE credit DECIMAL DEFAULT 0;
    ->
    ->     SELECT creditLimit
    ->     INTO credit
    ->     FROM customers
    ->     WHERE customerNumber = pCustomerNumber;
    ->
    ->     IF credit > 50000 THEN
    ->         SET pCustomerLevel = 'PLATINUM';
    ->     ELSEIF credit <= 50000 AND credit > 10000 THEN
    ->         SET pCustomerLevel = 'GOLD';
    ->     ELSE
    ->         SET pCustomerLevel = 'SILVER';
    ->     END IF;
    -> END $$
ERROR 1304 (42000): PROCEDURE GetCustomerLevel already exists
mysql>
mysql> DELIMITER ;
mysql>
```

```
C:\WINDOWS\system32\cmd.exe - mysql  -P3307 -u root -p                    —   □   ✕

mysql> DELIMITER ;
mysql> CALL GetCustomerLevel(447, @level);
Query OK, 1 row affected (0.00 sec)

mysql> SELECT @level;
+--------------+
| @level       |
+--------------+
| NOT PLATINUM |
+--------------+
1 row in set (0.00 sec)

mysql>
```

```
C:\WINDOWS\system32\cmd.exe - mysql -P3307 -u root -p                    —    □    ✕

mysql>
mysql> DELIMITER $$
mysql>
mysql> CREATE PROCEDURE GetCustomerShipping(
    -> IN  pCustomerNUmber INT,
    -> OUT pShipping       VARCHAR(50)
    -> )
    -> BEGIN
    ->     DECLARE customerCountry VARCHAR(100);
    ->
    -> SELECT
    ->     country
    -> INTO customerCountry FROM
    ->     customers
    -> WHERE
    ->     customerNumber = pCustomerNUmber;
    ->
    ->     CASE customerCountry
    -> WHEN  'USA' THEN
    ->     SET pShipping = '2-day Shipping';
    -> WHEN 'Canada' THEN
    ->     SET pShipping = '3-day Shipping';
    -> ELSE
    ->     SET pShipping = '5-day Shipping';
    -> END CASE;
    -> END$$
Query OK, 0 rows affected (0.00 sec)

mysql>
mysql> DELIMITER ;
mysql>
```

```
C:\WINDOWS\system32\cmd.exe - mysql -P3307 -u root -p                    —    □    ✕

mysql>
mysql> DELIMITER ;
mysql> CALL GetCustomerShipping(112,@shipping);
Query OK, 1 row affected (0.01 sec)

mysql> SELECT @shipping;
+----------------+
| @shipping      |
+----------------+
| 2-day Shipping |
+----------------+
1 row in set (0.00 sec)

mysql>
```

```
C:\WINDOWS\system32\cmd.exe - mysql  -P3307 -u root -p

mysql> DELIMITER $$
mysql>
mysql> CREATE PROCEDURE GetDeliveryStatus(
    -> IN pOrderNumber INT,
    ->    OUT pDeliveryStatus VARCHAR(100)
    -> )
    -> BEGIN
    -> DECLARE waitingDay INT DEFAULT 0;
    ->    SELECT
    -> DATEDIFF(requiredDate, shippedDate)
    -> INTO waitingDay
    -> FROM orders
    ->    WHERE orderNumber = pOrderNumber;
    ->
    ->    CASE
    -> WHEN waitingDay = 0 THEN
    -> SET pDeliveryStatus = 'On Time';
    ->        WHEN waitingDay >= 1 AND waitingDay < 5 THEN
    -> SET pDeliveryStatus = 'Late';
    -> WHEN waitingDay >= 5 THEN
    -> SET pDeliveryStatus = 'Very Late';
    -> ELSE
    -> SET pDeliveryStatus = 'No Information';
    -> END CASE;
    -> END$$
Query OK, 0 rows affected (0.01 sec)

mysql> DELIMITER ;
mysql> CALL GetDeliveryStatus(10100,@delivery);
Query OK, 1 row affected (0.08 sec)

mysql>
```

```
C:\WINDOWS\system32\cmd.exe - mysql  -P3307 -u root -p

mysql>
mysql> DELIMITER $$
mysql> CREATE PROCEDURE LoopDemo()
    -> BEGIN
    -> DECLARE x  INT;
    -> DECLARE str  VARCHAR(255);
    ->
    -> SET x = 1;
    -> SET str =  '';
    ->
    -> loop_label:  LOOP
    -> IF  x > 10 THEN
    -> LEAVE  loop_label;
    -> END  IF;
    ->
    -> SET  x = x + 1;
    -> IF  (x mod 2) THEN
    -> ITERATE  loop_label;
    -> ELSE
    -> SET  str = CONCAT(str,x,',');
    -> END  IF;
    -> END LOOP;
    -> SELECT str;
    -> END$$
Query OK, 0 rows affected (0.01 sec)

mysql>
mysql> DELIMITER ;
mysql>
```

```
    -> END$$
Query OK, 0 rows affected (0.01 sec)

mysql>
mysql> DELIMITER ;
mysql> CALL LoopDemo();
+-------------+
| str         |
+-------------+
| 2,4,6,8,10, |
+-------------+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

mysql>
```

```
Query OK, 0 rows affected (0.00 sec)

mysql> CREATE TABLE calendars(
    ->     id INT AUTO_INCREMENT,
    ->     fulldate DATE UNIQUE,
    ->     day TINYINT NOT NULL,
    ->     month TINYINT NOT NULL,
    ->     quarter TINYINT NOT NULL,
    ->     year INT NOT NULL,
    ->     PRIMARY KEY(id)
    -> );
Query OK, 0 rows affected (0.07 sec)

mysql> DELIMITER $$
mysql>
mysql> CREATE PROCEDURE InsertCalendar(dt DATE)
    -> BEGIN
    ->     INSERT INTO calendars(
    ->         fulldate,
    ->         day,
    ->         month,
    ->         quarter,
    ->         year
    ->     )
    ->     VALUES(
    ->         dt,
    ->         EXTRACT(DAY FROM dt),
    ->         EXTRACT(MONTH FROM dt),
    ->         EXTRACT(QUARTER FROM dt),
    ->         EXTRACT(YEAR FROM dt)
    ->     );
    -> END$$
Query OK, 0 rows affected (0.01 sec)

mysql>
mysql> DELIMITER ;
mysql>
```

```
mysql>
mysql> DELIMITER ;
mysql> DELIMITER $$
mysql>
mysql> CREATE PROCEDURE LoadCalendars(
    ->     startDate DATE,
    ->     day INT
    -> )
    -> BEGIN
    ->
    ->     DECLARE counter INT DEFAULT 1;
    ->     DECLARE dt DATE DEFAULT startDate;
    ->
    ->     WHILE counter <= day DO
    ->         CALL InsertCalendar(dt);
    ->         SET counter = counter + 1;
    ->         SET dt = DATE_ADD(dt,INTERVAL 1 day);
    ->     END WHILE;
    ->
    -> END$$
Query OK, 0 rows affected (0.00 sec)

mysql>
mysql> DELIMITER ;
mysql>
```

```
    ->         CALL InsertCalendar(dt);
    ->         SET counter = counter + 1;
    ->         SET dt = DATE_ADD(dt,INTERVAL 1 day);
    ->     END WHILE;
    ->
    -> END$$
Query OK, 0 rows affected (0.00 sec)

mysql>
mysql> DELIMITER ;
mysql> CALL LoadCalendars('2019-01-01',31);
Query OK, 1 row affected (0.06 sec)

mysql> CALL LoadCalendars('2019-01-01',31);
```

```
C:\WINDOWS\system32\cmd.exe - mysql -P3307 -u root -p                    —   □   ×

mysql> DELIMITER $$
mysql>
mysql> CREATE PROCEDURE RepeatDemo()
    -> BEGIN
    ->     DECLARE counter INT DEFAULT 1;
    ->     DECLARE result VARCHAR(100) DEFAULT '';
    ->
    ->     REPEAT
    ->         SET result = CONCAT(result,counter,',');
    ->         SET counter = counter + 1;
    ->     UNTIL counter >= 10
    ->     END REPEAT;
    ->
    ->     -- display result
    ->     SELECT result;
    -> END$$
Query OK, 0 rows affected (0.00 sec)

mysql>
mysql> DELIMITER ;
mysql> CALL RepeatDemo();
+--------------------+
| result             |
+--------------------+
| 1,2,3,4,5,6,7,8,9, |
+--------------------+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

mysql>
```

```
C:\WINDOWS\system32\cmd.exe - mysql -P3307 -u root -p                    —   □   ×

DELIMITER' at line 1
mysql> DELIMITER $$
mysql>
mysql> CREATE PROCEDURE CheckCredit(
    ->     inCustomerNumber int
    -> )
    -> sp: BEGIN
    ->
    ->     DECLARE customerCount INT;
    ->
    ->     -- check if the customer exists
    ->     SELECT
    ->         COUNT(*)
    ->     INTO customerCount
    ->     FROM
    ->         customers
    ->     WHERE
    ->         customerNumber = inCustomerNumber;
    ->
    ->     -- if the customer does not exist, terminate
    ->     -- the stored procedure
    ->     IF customerCount = 0 THEN
    ->         LEAVE sp;
    ->     END IF;
    ->
    ->     -- other logic
    ->     -- ...
    -> END$$
Query OK, 0 rows affected (0.01 sec)

mysql>
mysql> DELIMITER ;
mysql>
```

```
C:\WINDOWS\system32\cmd.exe - mysql  -P3307 -u root -p

mysql> DELIMITER $$
mysql>
mysql> CREATE PROCEDURE LeaveDemo(OUT result VARCHAR(100))
    -> BEGIN
    ->     DECLARE counter INT DEFAULT 1;
    ->     DECLARE times INT;
    ->     -- generate a random integer between 4 and 10
    ->     SET times  = FLOOR(RAND()*(10-4+1)+4);
    ->     SET result = '';
    ->     disp: LOOP
    ->         -- concatenate counters into the result
    ->         SET result = concat(result,counter,',');
    ->
    ->         -- exit the loop if counter equals times
    ->         IF counter = times THEN
    ->             LEAVE disp;
    ->         END IF;
    ->         SET counter = counter + 1;
    ->     END LOOP;
    -> END$$
Query OK, 0 rows affected (0.01 sec)

mysql>
mysql> DELIMITER ;
mysql>
```

```
C:\WINDOWS\system32\cmd.exe - mysql  -P3307 -u root -p

    ->         END IF;
    ->         SET counter = counter + 1;
    ->     END LOOP;
    -> END$$
Query OK, 0 rows affected (0.01 sec)

mysql>
mysql> DELIMITER ;
mysql> CALL LeaveDemo(@result);
Query OK, 0 rows affected (0.00 sec)

mysql> SELECT @result;
+----------------+
| @result        |
+----------------+
| 1,2,3,4,5,6,7, |
+----------------+
1 row in set (0.00 sec)

mysql>
```

```
mysql>
mysql>
mysql>
mysql> DELIMITER $$
mysql> CREATE PROCEDURE createEmailList (
    -> INOUT emailList varchar(4000)
    -> )
    -> BEGIN
    -> DECLARE finished INTEGER DEFAULT 0;
    -> DECLARE emailAddress varchar(100) DEFAULT "";
    ->
    -> -- declare cursor for employee email
    -> DEClARE curEmail
    -> CURSOR FOR
    -> SELECT email FROM employees;
    ->
    -> -- declare NOT FOUND handler
    -> DECLARE CONTINUE HANDLER
    ->        FOR NOT FOUND SET finished = 1;
    ->
    -> OPEN curEmail;
    ->
    -> getEmail: LOOP
    -> FETCH curEmail INTO emailAddress;
    -> IF finished = 1 THEN
    -> LEAVE getEmail;
    -> END IF;
    -> -- build email list
    -> SET emailList = CONCAT(emailAddress,";",emailList);
    -> END LOOP getEmail;
    -> CLOSE curEmail;
    ->
    -> END$$
Query OK, 0 rows affected (0.00 sec)

mysql> DELIMITER ;
mysql>
```

```
    -> CLOSE curEmail;
    ->
    -> END$$
Query OK, 0 rows affected (0.00 sec)

mysql> DELIMITER ;
mysql> SET @emailList = "";
Query OK, 0 rows affected (0.00 sec)

mysql> CALL createEmailList(@emailList);
Query OK, 0 rows affected (0.02 sec)

mysql> SELECT @emailList;
+--------------------------------------------------------------
--------------------------------------------------------------
--------------------------------------------------------------
--------------------------------------------------------------
--------------------------------------------------------------
--------------------------------------------------------------
--------------------------------------------------------+
| @emailList
```

```
C:\WINDOWS\system32\cmd.exe - mysql  -P3307 -u root -p                    —    □    ✕

mysql> CREATE TABLE SupplierProducts (
    ->       supplierId INT,
    ->       productId INT,
    ->       PRIMARY KEY (supplierId , productId)
    -> );
Query OK, 0 rows affected (0.03 sec)


mysql>
```

```
C:\WINDOWS\system32\cmd.exe - mysql  -P3307 -u root -p                    —    □    ✕

Query OK, 0 rows affected (0.03 sec)

mysql> CREATE PROCEDURE InsertSupplierProduct(
    ->       IN inSupplierId INT,
    ->       IN inProductId INT
    -> )
    -> BEGIN
    ->       -- exit if the duplicate key occurs
    ->       DECLARE EXIT HANDLER FOR 1062
    ->       BEGIN
    ->   SELECT CONCAT('Duplicate key (',inSupplierId,',',inProductId,') occurred') AS message;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for th
e right syntax to use near '' at line 9
mysql>        END;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for th
e right syntax to use near 'END' at line 1
mysql>
mysql>        -- insert a new row into the SupplierProducts
mysql>        INSERT INTO SupplierProducts(supplierId,productId)
    ->        VALUES(inSupplierId,inProductId);
ERROR 1054 (42S22): Unknown column 'inSupplierId' in 'field list'
mysql>
mysql>        -- return the products supplied by the supplier id
mysql>        SELECT COUNT(*)
    ->        FROM SupplierProducts
    ->        WHERE supplierId = inSupplierId;
ERROR 1054 (42S22): Unknown column 'inSupplierId' in 'where clause'
mysql>
mysql> END$$
    ->
    -> DELIMITER ;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for th
e right syntax to use near 'END$$

DELIMITER' at line 1
```

```
C:\WINDOWS\system32\cmd.exe - mysql  -P3307 -u root -p                    —    □    ✕

mysql> INSERT INTO SupplierProducts(supplierId,productId)
    -> VALUES(supplierId,productId);
Query OK, 1 row affected (0.07 sec)

mysql> CALL InsertSupplierProduct(1,1);
ERROR 1305 (42000): PROCEDURE classicmodels.InsertSupplierProduct does not exist
mysql> CALL InsertSupplierProduct(1,2);
ERROR 1305 (42000): PROCEDURE classicmodels.InsertSupplierProduct does not exist
mysql> CALL InsertSupplierProduct(1,3);
ERROR 1305 (42000): PROCEDURE classicmodels.InsertSupplierProduct does not exist
mysql> CALL InsertSupplierProduct(1,3);
ERROR 1305 (42000): PROCEDURE classicmodels.InsertSupplierProduct does not exist
mysql>
```

```
C:\WINDOWS\system32\cmd.exe - mysql  -P3307 -u root -p                    —    □    ✕

mysql>
mysql> DELIMITER $$
mysql>
mysql> CREATE PROCEDURE InsertSupplierProduct(
    ->     IN inSupplierId INT,
    ->     IN inProductId INT
    -> )
    -> BEGIN
    ->     -- exit if the duplicate key occurs
    ->     DECLARE CONTINUE HANDLER FOR 1062
    ->     BEGIN
    -> SELECT CONCAT('Duplicate key (',inSupplierId,',',inProductId,') occurred') AS message;
    ->     END;
    ->
    ->     -- insert a new row into the SupplierProducts
    ->     INSERT INTO SupplierProducts(supplierId,productId)
    ->     VALUES(inSupplierId,inProductId);
    ->
    ->     -- return the products supplied by the supplier id
    ->     SELECT COUNT(*)
    ->     FROM SupplierProducts
    ->     WHERE supplierId = inSupplierId;
    ->
    -> END$$
Query OK, 0 rows affected (0.00 sec)

mysql>
mysql> DELIMITER ;
mysql> CALL InsertSupplierProduct(1,3);
+----------+
| COUNT(*) |
+----------+
|        1 |
+----------+
1 row in set (0.01 sec)

Query OK, 0 rows affected (0.02 sec)

mysql>
```

```
C:\WINDOWS\system32\cmd.exe - mysql  -P3307 -u root -p                               —    □    ✕

mysql> DELIMITER $$
mysql>
mysql> CREATE PROCEDURE InsertSupplierProduct(
    ->      IN inSupplierId INT,
    ->      IN inProductId INT
    -> )
    -> BEGIN
    ->      -- exit if the duplicate key occurs
    ->      DECLARE EXIT HANDLER FOR 1062 SELECT 'Duplicate keys error encountered' Message;
    ->      DECLARE EXIT HANDLER FOR SQLEXCEPTION SELECT 'SQLException encountered' Message;
    ->      DECLARE EXIT HANDLER FOR SQLSTATE '23000' SELECT 'SQLSTATE 23000' ErrorCode;
    ->
    ->      -- insert a new row into the SupplierProducts
    ->      INSERT INTO SupplierProducts(supplierId,productId)
    ->      VALUES(inSupplierId,inProductId);
    ->
    ->      -- return the products supplied by the supplier id
    ->      SELECT COUNT(*)
    ->      FROM SupplierProducts
    ->      WHERE supplierId = inSupplierId;
    ->
    -> END$$
Query OK, 0 rows affected (0.00 sec)

mysql>
mysql> DELIMITER ;
mysql> CALL InsertSupplierProduct(1,3);
+----------------------------------+
| Message                          |
+----------------------------------+
| Duplicate keys error encountered |
+----------------------------------+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.01 sec)

mysql>
```

```
C:\WINDOWS\system32\cmd.exe - mysql  -P3307 -u root -p                               —    □    ✕

| Duplicate keys error encountered |
+----------------------------------+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.01 sec)

mysql> DELIMITER $$
mysql>
mysql> CREATE PROCEDURE TestProc()
    -> BEGIN
    ->
    ->      DECLARE EXIT HANDLER FOR 1146
    ->      SELECT 'Please create table abc first' Message;
    ->
    ->      SELECT * FROM abc;
    -> END$$
Query OK, 0 rows affected (0.01 sec)

mysql>
mysql> DELIMITER ;
```

```
C:\WINDOWS\system32\cmd.exe - mysql  -P3307 -u root -p

mysql>
mysql> DELIMITER $$
mysql>
mysql> CREATE PROCEDURE TestProc()
    -> BEGIN
    ->     DECLARE TableNotFound CONDITION for 1146 ;
    ->
    ->     DECLARE EXIT HANDLER FOR TableNotFound
    -> SELECT 'Please create table abc first' Message;
    ->     SELECT * FROM abc;
    -> END$$
Query OK, 0 rows affected (0.00 sec)

mysql>
mysql> DELIMITER ;
mysql>
```

```
C:\WINDOWS\system32\cmd.exe - mysql  -P3307 -u root -p

mysql> DELIMITER ;
mysql> DELIMITER $$
mysql>
mysql> CREATE PROCEDURE AddOrderItem(
    ->         in orderNo int,
    ->  in productCode varchar(45),
    ->  in qty int,
    ->                    in price double,
    ->                    in lineNo int )
    -> BEGIN
    -> DECLARE C INT;
    ->
    -> SELECT COUNT(orderNumber) INTO C
    -> FROM orders
    -> WHERE orderNumber = orderNo;
    ->
    -> -- check if orderNumber exists
    -> IF(C != 1) THEN
    -> SIGNAL SQLSTATE '45000'
    -> SET MESSAGE_TEXT = 'Order No not found in orders table';
    -> END IF;
    -> -- more code below
    -> -- ...
    -> END
    -> $$
Query OK, 0 rows affected (0.00 sec)
```

```
C:\WINDOWS\system32\cmd.exe - mysql  -P3307 -u root -p

mysql> CALL AddOrderItem(10,'S10_1678',1,95.7,1);
    ->
    -> ;
    -> $$
ERROR 1644 (45000): Order No not found in orders table
mysql> DELIMITER $$
mysql>
mysql> CREATE PROCEDURE Divide(IN numerator INT, IN denominator INT, OUT result double)
    -> BEGIN
    -> DECLARE division_by_zero CONDITION FOR SQLSTATE '22012';
    ->
    -> DECLARE CONTINUE HANDLER FOR division_by_zero
    -> RESIGNAL SET MESSAGE_TEXT = 'Division by zero / Denominator cannot be zero';
    -> --
    -> IF denominator = 0 THEN
    -> SIGNAL division_by_zero;
    -> ELSE
    -> SET result := numerator / denominator;
    -> END IF;
    -> END
    -> $$
Query OK, 0 rows affected (0.00 sec)

mysql> CALL Divide(10,0,@result);
    -> $$
ERROR 1644 (22012): Division by zero / Denominator cannot be zero
mysql>
```

```
C:\WINDOWS\system32\cmd.exe - mysql  -P3307 -u root -p

mysql>
mysql> CREATE FUNCTION CustomerLevel(
    -> credit DECIMAL(10,2)
    -> )
    -> RETURNS VARCHAR(20)
    -> DETERMINISTIC
    -> BEGIN
    ->     DECLARE customerLevel VARCHAR(20);
    ->
    ->     IF credit > 50000 THEN
    -> SET customerLevel = 'PLATINUM';
    ->     ELSEIF (credit >= 50000 AND
    -> credit <= 10000) THEN
    ->         SET customerLevel = 'GOLD';
    ->     ELSEIF credit < 10000 THEN
    ->         SET customerLevel = 'SILVER';
    ->     END IF;
    -> -- return the customer level
    -> RETURN (customerLevel);
    -> END$$
Query OK, 0 rows affected (0.01 sec)

mysql> DELIMITER ;
mysql> SHOW FUNCTION STATUS
    -> WHERE db = 'classicmodels';
+--------------+--------------+----------+---------------+---------------------+---------------------+---------------+---------+
--------------------+----------------------+--------------------+
| Db           | Name         | Type     | Definer       | Modified            | Created             | Security_type | Comment |
character_set_client | collation_connection | Database Collation |
+--------------+--------------+----------+---------------+---------------------+---------------------+---------------+---------+
--------------------+----------------------+--------------------+
| classicmodels | CustomerLevel | FUNCTION | root@localhost | 2020-12-09 17:44:22 | 2020-12-09 17:44:22 | DEFINER       |         |
cp850                | cp850_general_ci     | latin1_swedish_ci  |
+--------------+--------------+----------+---------------+---------------------+---------------------+---------------+---------+
--------------------+----------------------+--------------------+
1 row in set (0.01 sec)

mysql>
```

```
--------------------------------+--------------------+--------------------+
1 row in set (0.01 sec)

mysql> SELECT
    ->     customerName,
    ->     CustomerLevel(creditLimit)
    -> FROM
    ->     customers
    -> ORDER BY
    ->     customerName;
+--------------------------------+----------------------------+
| customerName                   | CustomerLevel(creditLimit) |
+--------------------------------+----------------------------+
| Alpha Cognac                   | PLATINUM                   |
| American Souvenirs Inc         | SILVER                     |
| Amica Models & Co.             | PLATINUM                   |
| ANG Resellers                  | SILVER                     |
| Anna's Decorations, Ltd        | PLATINUM                   |
| Anton Designs, Ltd.            | SILVER                     |
| Asian Shopping Network, Co     | SILVER                     |
| Asian Treasures, Inc.          | SILVER                     |
| Atelier graphique              | NULL                       |
| Australian Collectables, Ltd   | PLATINUM                   |
| Australian Collectors, Co.     | PLATINUM                   |
| Australian Gift Network, Co    | PLATINUM                   |
| Auto Associ?s & Cie.           | PLATINUM                   |
| Auto Canal+ Petit              | PLATINUM                   |
| Auto-Moto Classics Inc.        | NULL                       |
| AV Stores, Co.                 | PLATINUM                   |
| Baane Mini Imports             | PLATINUM                   |
| Bavarian Collectables Imports, Co. | PLATINUM               |
| BG&E Collectables              | SILVER                     |
| Blauer See Auto, Co.           | PLATINUM                   |
| Boards & Toys Co.              | NULL                       |
| CAF Imports                    | PLATINUM                   |
| Cambridge Collectables Co.     | NULL                       |
| Canadian Gift Exchange Network | PLATINUM                   |
| Classic Gift Ideas, Inc        | PLATINUM                   |
| Classic Legends Inc.           | PLATINUM                   |
| Clover Collections, Co.        | PLATINUM                   |
```

```
mysql> DELIMITER $$
mysql>
mysql> CREATE PROCEDURE GetCustomerLevel(
    ->     IN  customerNo INT,
    ->     OUT customerLevel VARCHAR(20)
    -> )
    -> BEGIN
    ->
    -> DECLARE credit DEC(10,2) DEFAULT 0;
    ->
    ->     -- get credit limit of a customer
    ->     SELECT
    -> creditLimit
    -> INTO credit
    ->     FROM customers
    ->     WHERE
    -> customerNumber = customerNo;
    ->
    ->     -- call the function
    ->     SET customerLevel = CustomerLevel(credit);
    -> END$$
ERROR 1304 (42000): PROCEDURE GetCustomerLevel already exists
mysql>
mysql> DELIMITER ;
mysql> CALL GetCustomerLevel(-131,@customerLevel);
Query OK, 0 rows affected (0.00 sec)

mysql> SELECT @customerLevel;
+----------------+
| @customerLevel |
+----------------+
| NOT PLATINUM   |
+----------------+
1 row in set (0.00 sec)

mysql>
```

```
mysql>
mysql> DELIMITER $$
mysql>
mysql> CREATE FUNCTION OrderLeadTime (
    ->     orderDate DATE,
    ->     requiredDate DATE
    -> )
    -> RETURNS INT
    -> DETERMINISTIC
    -> BEGIN
    ->     RETURN requiredDate - orderDate;
    -> END$$
Query OK, 0 rows affected (0.00 sec)

mysql>
mysql> DELIMITER ;
mysql> DROP FUNCTION OrderLeadTime;
Query OK, 0 rows affected (0.07 sec)

mysql> DROP FUNCTION IF EXISTS NonExistingFunction;
Query OK, 0 rows affected, 1 warning (0.07 sec)

mysql> SHOW WARNINGS;
+-------+------+-------------------------------------------------------------+
| Level | Code | Message                                                     |
+-------+------+-------------------------------------------------------------+
| Note  | 1305 | FUNCTION classicmodels.NonExistingFunction does not exist   |
+-------+------+-------------------------------------------------------------+
1 row in set (0.00 sec)

mysql>
```

```
+-------+------+-------------------------------------------------------------+
1 row in set (0.00 sec)

mysql> SELECT
    ->     routine_name
    -> FROM
    ->     information_schema.routines
    -> WHERE
    ->     routine_type = 'FUNCTION'
    ->         AND routine_schema = '<database_name>';
Empty set (0.00 sec)

mysql> SELECT
    ->     routine_name
    -> FROM
    ->     information_schema.routines
    -> WHERE
    ->     routine_type = 'FUNCTION'
    ->         AND routine_schema = 'classicmodels';
+---------------+
| ROUTINE_NAME  |
+---------------+
| CustomerLevel |
+---------------+
1 row in set (0.00 sec)

mysql>
```

```
C:\WINDOWS\system32\cmd.exe - mysql  -P3307 -u root -p                        —    □    ×

mysql> DELIMITER $$
mysql>
mysql> CREATE PROCEDURE get_order_by_cust(
    -> IN cust_no INT,
    -> OUT shipped INT,
    -> OUT canceled INT,
    -> OUT resolved INT,
    -> OUT disputed INT)
    -> BEGIN
    -> -- shipped
    -> SELECT
    ->              count(*) INTO shipped
    ->         FROM
    ->             orders
    ->         WHERE
    ->             customerNumber = cust_no
    ->                 AND status = 'Shipped';
    ->
    -> -- canceled
    -> SELECT
    ->              count(*) INTO canceled
    ->         FROM
    ->             orders
    ->         WHERE
    ->             customerNumber = cust_no
    ->                 AND status = 'Canceled';
    ->
    -> -- resolved
    -> SELECT
    ->              count(*) INTO resolved
    ->         FROM
    ->             orders
    ->         WHERE
    ->             customerNumber = cust_no
    ->                 AND status = 'Resolved';
    ->
    -> -- disputed
    -> SELECT
    ->              count(*) INTO disputed
    ->         FROM
```

```
C:\WINDOWS\system32\cmd.exe - mysql  -P3307 -u root -p                        —    □    ×

    ->              orders
    ->         WHERE
    ->             customerNumber = cust_no
    ->                 AND status = 'Disputed';
    ->
    -> END
    -> $$
Query OK, 0 rows affected (0.07 sec)

mysql> CALL get_order_by_cust(141,@shipped,@canceled,@resolved,@disputed);
    -> SELECT @shipped,@canceled,@resolved,@disputed;
    -> $$
Query OK, 1 row affected (0.00 sec)

+----------+-----------+-----------+-----------+
| @shipped | @canceled | @resolved | @disputed |
+----------+-----------+-----------+-----------+
|       22 |         0 |         1 |         1 |
+----------+-----------+-----------+-----------+
1 row in set (0.01 sec)

mysql>
```

```
C:\WINDOWS\system32\cmd.exe - mysql  -P3307 -u root -p                         —    □    ×

ERROR:
No query specified

mysql> USE testdb;
Database changed
mysql> CREATE TABLE messages (
    ->      id INT AUTO_INCREMENT,
    ->      message VARCHAR(100) NOT NULL,
    ->      PRIMARY KEY (id)
    -> );
Query OK, 0 rows affected (0.04 sec)

mysql> DELIMITER $$
mysql>
mysql> CREATE DEFINER = root@localhost PROCEDURE InsertMessage(
    ->      msg VARCHAR(100)
    -> )
    -> SQL SECURITY DEFINER
    -> BEGIN
    ->      INSERT INTO messages(message)
    ->      VALUES(msg);
    -> END$$
Query OK, 0 rows affected (0.01 sec)

mysql>
mysql> DELIMITER ;
mysql> CREATE USER dev@localhost
    -> IDENTIFIED BY 'Abcd1234';
Query OK, 0 rows affected (0.01 sec)

mysql> GRANT EXECUTE ON testdb.*
    -> TO dev@localhost;
Query OK, 0 rows affected (0.09 sec)

mysql>
```

```
C:\WINDOWS\system32\cmd.exe - mysql  -P3307 -u root -p                         —    □    ×

mysql> DELIMITER $$
mysql>
mysql> CREATE DEFINER=root@localhost
    -> PROCEDURE UpdateMessage(
    ->      msgId INT,
    ->      msg VARCHAR(100)
    -> )
    -> SQL SECURITY INVOKER
    -> BEGIN
    ->      UPDATE messages
    ->      SET message = msg
    ->      WHERE id = msgId;
    -> END$$
Query OK, 0 rows affected (0.00 sec)

mysql>
```

```
C:\WINDOWS\system32\cmd.exe - mysql  -P3307 -u root -p                    —    □    ✕

mysql> CREATE DATABASE testSP;
Query OK, 1 row affected (0.08 sec)

mysql> use testSP;
Database changed
mysql> create table Product(
    -> ProductId int primary key,
    -> ProductName varchar(20) unique,
    ->    ProductQty int,
    ->     ProductPrice float
    -> );
Query OK, 0 rows affected (0.14 sec)

mysql>
```

```
C:\WINDOWS\system32\cmd.exe - mysql  -P3307 -u root -p                    —    □    ✕

    ->       ProductPrice float
    -> );
Query OK, 0 rows affected (0.14 sec)

mysql> insert product values(1,'Printer',10,4500);
Query OK, 1 row affected (0.01 sec)

mysql> insert product values(2,'Scanner',15,3500);
Query OK, 1 row affected (0.00 sec)

mysql> insert product values(3,'Mouse',45,500);
Query OK, 1 row affected (0.00 sec)

mysql> SELECT * FROM Product;
+-----------+-------------+------------+--------------+
| ProductId | ProductName | ProductQty | ProductPrice |
+-----------+-------------+------------+--------------+
|         1 | Printer     |         10 |         4500 |
|         2 | Scanner     |         15 |         3500 |
|         3 | Mouse       |         45 |          500 |
+-----------+-------------+------------+--------------+
3 rows in set (0.00 sec)

mysql>
```

```
C:\WINDOWS\system32\cmd.exe - mysql  -P3307 -u root -p                    —    □    ✕

    -> DELIMITER$$
    -> create procedure prcInsert(id int, name varchar(40), qty int, price float)
    -> BEGIN
    -> declare cnt int;
    -> declare p float;
    -> select cnt=COUNT(ProductId) from Product where pname=name;
    -> if (p<price) THEN
    -> update Product set ProductQty = ProductQty + qty where ProductName =name;
    -> select p=ProductPrice from Product where ProductName=name;
    -> if(p<price) THEN
    -> update Product set ProductPrice=price where ProductName=name;
    -> END IF;
    -> ELSE
    -> insert Product values(id, name, qty, price);
    -> END IF;
    -> END$$
```

```
C:\WINDOWS\system32\cmd.exe - mysql  -P3307 -u root -p                          —   □   ×

mysql> SET autocommit = 1;
Query OK, 0 rows affected (0.00 sec)

mysql> START TRANSACTION;
Query OK, 0 rows affected (0.00 sec)

mysql>
mysql> -- 2. Get the highest income
mysql>
mysql> SELECT @income:= MAX(income) FROM employees;
ERROR 1054 (42S22): Unknown column 'income' in 'field list'
mysql>
mysql> -- 3. Insert a new record into the employee table
mysql>
mysql> INSERT INTO employees(emp_id, emp_name, emp_age, city, income)
    -> VALUES (111, 'Alexander', 45, 'California', 70000);
ERROR 1054 (42S22): Unknown column 'emp_id' in 'field list'
mysql>
mysql> -- 4. Insert a new record into the order table
mysql>
mysql> INSERT INTO Orders(order_id, prod_name, order_num, order_date)
    -> VALUES (6, 'Printer', 5654, '2020-01-10');
ERROR 1146 (42S02): Table 'employees.orders' doesn't exist
mysql>
mysql> -- 5. Commit changes
mysql> COMMIT;
Query OK, 0 rows affected (0.00 sec)

mysql>
```

```
C:\WINDOWS\system32\cmd.exe - mysql  -P3307 -u root -p                          —   □   ×

mysql>
mysql>      -- 1. Start a new transaction
mysql>
mysql>      START TRANSACTION;
Query OK, 0 rows affected (0.00 sec)

mysql>
mysql>      -- 2. Get the highest income
mysql>
mysql>      SELECT @income:= MAX(income) FROM employees;
ERROR 1054 (42S22): Unknown column 'income' in 'field list'
mysql>
mysql>      -- 3. Insert a new record into the employee table
mysql>
mysql>      INSERT INTO employees(emp_id, emp_name, emp_age, city, income)
    ->      VALUES (111, 'Alexander', 45, 'California', 70000);
ERROR 1054 (42S22): Unknown column 'emp_id' in 'field list'
mysql>
mysql>      -- 4. Insert a new record into the order table
mysql>
mysql>      INSERT INTO Orders(order_id, prod_name, order_num, order_date)
    ->      VALUES (6, 'Printer', 5654, '2020-01-10');
ERROR 1146 (42S02): Table 'employees.orders' doesn't exist
mysql>
mysql>      -- 5. Commit changes
mysql>      COMMIT;
Query OK, 0 rows affected (0.00 sec)

mysql>
```