



TECNOLÓGICO  
NACIONAL DE MÉXICO



# Instituto Tecnológico Superior de Jerez – ITSJ

Estudiante: Miguel Angel Bazán garduño.

[mabg211299@hotmail.com](mailto:mabg211299@hotmail.com)

5to Semestre Carrera: Ingeniería en sistemas  
computacionales (ISC).

Materia: Taller de Base De Datos.

Actividad: Tutorial Stored Procedures.

Docente: I.S.C. Salvador Acevedo Sandoval

Jerez de García Salinas Zacatecas

## Introducción to MySQL Stored Procedures:

```
mysql> CREATE PROCEDURE GetCustomers()
-> BEGIN
-> SELECT
-> customerName,
-> city,
-> state,
-> postalCode,
-> country
-> FROM
-> customers
-> ORDER BY customerName;
-> END$$
Query OK, 0 rows affected (0.33 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
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

## Delimiter

```
mysql> DELIMITER //
mysql> DELIMITER //
mysql>
mysql> SELECT * FROM customers //
```

customerNumber	customerName	city	state	postalCode
103	Atelier graphique	Nantes	NULL	44000
112	Signal Gift Stores	Las Vegas	NV	89001
114	Australian Collectors, Co.	Melbourne	Victoria	3004
119	La Rochelle Gifts	Nantes	NULL	44000
121	Baane Mini Imports	Stavern	NULL	4116
124	Mini Gifts Distributors Ltd.			

## Creando nuevo stored procedures:

```
mysql> DELIMITER ;
mysql> DELIMITER //
mysql>
mysql> CREATE PROCEDURE GetAllProducts()
-> BEGIN
-> SELECT * FROM products;
-> END //
Query OK, 0 rows affected (0.13 sec)

mysql>
mysql> DELIMITER ;
```

```
-----+-----+-----+
110 rows in set (0.00 sec)

Query OK, 0 rows affected (3.89 sec)

mysql>
```

## Borrando storeProcedures:

```
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 (0.12 sec)

mysql>
mysql> DELIMITER ;
mysql> DROP PROCEDURE GetEmployees;
Query OK, 0 rows affected (0.21 sec)

mysql>
```

## Parametros

```
mysql> CREATE PROCEDURE GetOfficeByCountry(
-> IN countryName VARCHAR(255)
-> )
-> BEGIN
-> SELECT *
-> FROM offices
-> WHERE country = countryName;
-> END //
Query OK, 0 rows affected (0.13 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 (0.26 sec)

Query OK, 0 rows affected (0.28 sec)
```

Show Procedures:

```
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	Security_type	Comment	character_set_client	collat
ion_connection	Database Collation								
classicmodels	GetAllProducts	PROCEDURE	root@localhost	2020-12-10 21:58:14	2020-12-10 21:58:14	DEFINER		cp850	cp850_
general_ci	latin1_swedish_ci								
classicmodels	GetCustomers	PROCEDURE	root@localhost	2020-12-10 21:46:48	2020-12-10 21:46:48	DEFINER		cp850	cp850_
general_ci	latin1_swedish_ci								
classicmodels	GetOfficeByCountry	PROCEDURE	root@localhost	2020-12-10 22:05:38	2020-12-10 22:05:38	DEFINER		cp850	cp850_
general_ci	latin1_swedish_ci								

```
3 rows in set (0.00 sec)

mysql>
```

## Section 2. Conditional Statements:

### MySQL simple IF-THEN statement

```
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 (0.13 sec)
```

```
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
119	118200.00
114	117300.00
448	116400.00
131	114900.00
363	114200.00
249	113000.00
484	65700.00
350	65000.00
129	64600.00
242	61100.00
216	60300.00
471	60300.00
487	60300.00
128	59700.00
344	59600.00
211	58600.00
347	57700.00
475	55400.00
209	53800.00
144	53100.00
333	51600.00

85 rows in set (0.00 sec)

```
mysql> CALL GetCustomerLevel(141, @level);  
Query OK, 1 row affected (0.03 sec)
```

```
mysql> SELECT @level;
```

@level
PLATINUM

1 row in set (0.00 sec)

## MySQL IF-THEN-ELSE statement

```
mysql> DROP PROCEDURE GetCustomerLevel;
Query OK, 0 rows affected (0.15 sec)

mysql> DELIMITER $$
mysql> CREATE PROCEDURE GetCustomerLevel(
-> IN pCustomerNumber INT,
-> OUT pCustomerLevel VARCHAR(20))
-> BEGIN
-> DECLARE credit DECIMAL DEFAULT 0;
->
-> SELECT creditLimit
-> INTO credit
-> FROM customers
-> WHERE customerNumber = pCustomerNumber;
->
-> IF credit > 50000 THEN
-> SET pCustomerLevel = 'PLATINUM';
-> ELSE
-> SET pCustomerLevel = 'NOT PLATINUM';
-> END IF;
-> END$$
Query OK, 0 rows affected (0.14 sec)

mysql>
mysql> DELIMITER ;
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
168	0.00
169	0.00
206	0.00
307	0.00
409	0.00
443	0.00
303	0.00
293	0.00
273	0.00
459	0.00
465	0.00
247	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>
```

MySQL IF-THEN-ELSEIF-ELSE statement:

```
mysql> DROP PROCEDURE GetCustomerLevel;
Query OK, 0 rows affected (0.15 sec)

mysql> DELIMITER $$
mysql>
mysql> CREATE PROCEDURE GetCustomerLevel(
->     IN  pCustomerNumber INT,
->     OUT pCustomerLevel  VARCHAR(20))
-> BEGIN
->     DECLARE credit DECIMAL DEFAULT 0;
->
->     SELECT creditLimit
->     INTO credit
->     FROM customers
->     WHERE customerNumber = pCustomerNumber;
->
->     IF credit > 50000 THEN
->         SET pCustomerLevel = 'PLATINUM';
->     ELSEIF credit <= 50000 AND credit > 10000 THEN
->         SET pCustomerLevel = 'GOLD';
->     ELSE
->         SET pCustomerLevel = 'SILVER';
->     END IF;
-> END $$
Query OK, 0 rows affected (0.09 sec)

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

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

## Simple CASE statement

```
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 (0.12 sec)

```
mysql>
mysql> DELIMITER ;
mysql> CALL GetCustomerShipping(112,@shipping);
Query OK, 1 row affected (0.00 sec)
```

```
mysql> SELECT @shipping;
```

```
+-----+
| @shipping |
+-----+
| 2-day Shipping |
+-----+
1 row in set (0.00 sec)
```

```
mysql>
```

Searched CASE statement:

```
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 (0.20 sec)

mysql> DELIMITER ;
mysql> CALL GetDeliveryStatus(10100,@delivery);
Query OK, 1 row affected (0.21 sec)

mysql> 
```

### Sección 3. Loops:

#### LOOP statement

```
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 (0.19 sec)

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



WHILE loop statement:

```
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 (0.11 sec)

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 (0.12 sec)

mysql>
mysql> DELIMITER ;
mysql> CALL LoadCalendars('2019-01-01',31);
Query OK, 1 row affected (3.83 sec)
```

## REPEAT loop:

```
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 (0.15 sec)

mysql>
mysql> DELIMITER ;
mysql> CALL RepeatDemo();
+-----+
| result |
+-----+
| 1,2,3,4,5,6,7,8,9, |
+-----+
1 row in set (0.01 sec)

Query OK, 0 rows affected (0.01 sec)

mysql>
```

## LEAVE statement:

```
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 (0.28 sec)

mysql>
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 (0.18 sec)

mysql>
mysql> DELIMITER ;
mysql> CALL LeaveDemo(@result);
Query OK, 0 rows affected (0.02 sec)

mysql> SELECT @result;
+-----+
| @result |
+-----+
| 1,2,3,4,5,6,7,8,9,10, |
+-----+
1 row in set (0.00 sec)
```

## Sección 4. Error Handling

Handler example in stored procedures:

```
mysql> CREATE PROCEDURE InsertSupplierProduct(
->   IN inSupplierId INT,
->   IN inProductId INT
-> )
-> BEGIN
->   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 the 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 the right syntax to use near 'END' at line
1
mysql>
mysql>   INSERT INTO SupplierProducts(supplierId,productId)
mysql>   VALUES(inSupplierId,inProductId);
ERROR 1054 (42S22): Unknown column 'inSupplierId' in 'field list'
mysql>
mysql>   SELECT COUNT(*)
mysql>   FROM SupplierProducts
mysql>   WHERE supplierId = inSupplierId;
ERROR 1054 (42S22): Unknown column 'inSupplierId' in 'where clause'
mysql>
mysql> END$$
-> DELIMITER ;
```

## SIGNAL statement

```
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 (0.26 sec)

mysql> DELIMITER ;
mysql> CALL AddOrderItem(10,'S10_1678',1,95.7,1);
ERROR 1644 (45000): Order No not found in orders table
```

## RESIGNAL statement

```
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 (0.26 sec)

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

## Sección 5. Cursors

```
-> SELECT email FROM employees;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'DECLARE curE
mail
CURSOR FOR
SELECT email FROM employees' at line 1
mysql>
mysql> -- declare NOT FOUND handler
mysql> DECLARE CONTINUE HANDLER
->      FOR NOT FOUND SET finished = 1;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'DECLARE CONT
INUE HANDLER
      FOR NOT FOUND SET finished = 1' at line 1
mysql> DECLARE finished INTEGER DEFAULT 0;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'DECLARE finis
hed INTEGER DEFAULT 0' at line 1
mysql> DECLARE emailAddress varchar(100) DEFAULT "";
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'DECLARE email
lAddress varchar(100) DEFAULT ""' at line 1
mysql>
mysql>
mysql> DECLARE curEmail
-> CURSOR FOR
-> SELECT email FROM employees;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'DECLARE curE
mail
CURSOR FOR
SELECT email FROM employees' at line 1
mysql>
mysql>
mysql> DECLARE CONTINUE HANDLER
->      FOR NOT FOUND SET finished = 1;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'DECLARE CONT
INUE HANDLER
      FOR NOT FOUND SET finished = 1' at line 1
mysql>
mysql>
mysql> DECLARE finished INTEGER DEFAULT 0;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'DECLARE finis
hed INTEGER DEFAULT 0' at line 1
mysql> DECLARE emailAddress varchar(100) DEFAULT "";
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'DECLARE email
lAddress varchar(100) DEFAULT ""' at line 1
mysql>
```

## Sección 6. Stored Functions:

### CREATE FUNCTION

```
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 (0.24 sec)

mysql> DELIMITER ;
mysql> SHOW FUNCTION STATUS
-> WHERE db = 'classicmodels';
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| Db      | Name      | Type      | Definer      | Modified      | Created      | Security_type | Comment | character_set_client | collation_co
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| classicmodels | CustomerLevel | FUNCTION | root@localhost | 2020-12-11 00:21:51 | 2020-12-11 00:21:51 | DEFINER      |        | cp850                | cp850_genera
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| classicmodels | latin1_swedish_ci |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.07 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 Associés & 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 |
| Collectable Mini Designs Co. | PLATINUM |
| Collectables For Less Inc. | PLATINUM |
| Corporate Gift Ideas Co. | PLATINUM |
| Corrida Auto Replicas, Ltd | PLATINUM |
| Cramer Spezialitäten, Ltd | SILVER |
| Cruz & Sons Co. | PLATINUM |
| Daedalus Designs Imports | PLATINUM |
| Danish Wholesale Imports | PLATINUM |
| Der Hund Imports | SILVER |
+-----+-----+
```

**DROP FUNCTION** statement:

```
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 (0.17 sec)

mysql>
mysql> DELIMITER ;
mysql> DROP FUNCTION OrderLeadTime;
Query OK, 0 rows affected (0.22 sec)

mysql> DROP FUNCTION IF EXISTS NonExistingFunction;
Query OK, 0 rows affected, 1 warning (0.04 sec)

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

mysql>
```

## Listing Stored Functions

[illegible]

## Sección 7. Stored Program Security:

```
mysql> CREATE DATABASE testdb;
Query OK, 1 row affected (0.15 sec)

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 (0.83 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 (0.12 sec)

mysql>
mysql> DELIMITER ;
mysql>
```

```
C:\Users\Bazán>mysql -u dev@localhost -p
Enter password: *****
ERROR 1045 (28000): Access denied for user 'dev@localhost'@'localhost' (using password: YES)

C:\Users\Bazán>mysql -u dev@localhost -p
Enter password: *****
ERROR 1045 (28000): Access denied for user 'dev@localhost'@'localhost' (using password: YES)

C:\Users\Bazán>mysql -u dev@localhost -p
Enter password: *****
ERROR 1045 (28000): Access denied for user 'dev@localhost'@'localhost' (using password: YES)

C:\Users\Bazán>mysql -u dev@localhost -p
Enter password: *****
ERROR 1045 (28000): Access denied for user 'dev@localhost'@'localhost' (using password: YES)

C:\Users\Bazán>mysql -u dev@localhost -p
Enter password: *****
ERROR 1045 (28000): Access denied for user 'dev@localhost'@'localhost' (using password: YES)

C:\Users\Bazán>
```