Instituto Tecnológico Superior de Jerez

Jerez, Zacatecas

08 de diciembre de 2020

Felipe de Jesús Márquez Olague

flype360360@gmail.com

S18070182

Ingeniería en Sistemas Computacionales

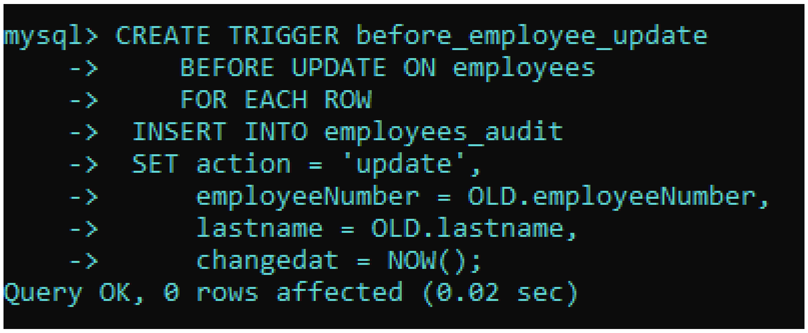
Quinto Semestre

Taller de Bases de Datos

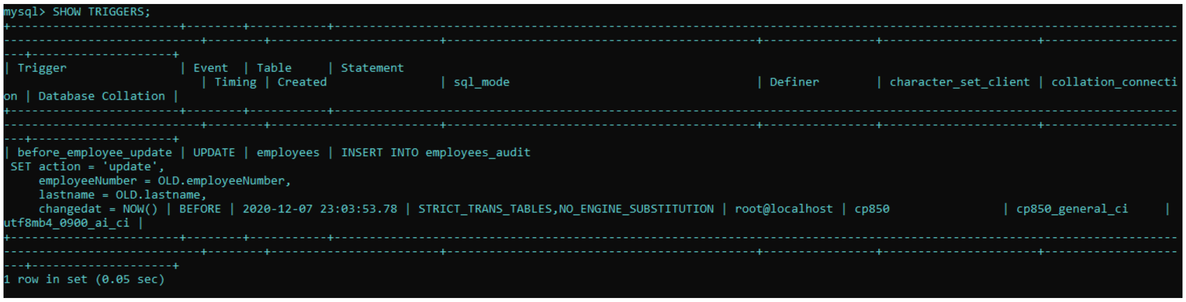
Capturas de triggers

ISC Salvador Acevedo Sandoval

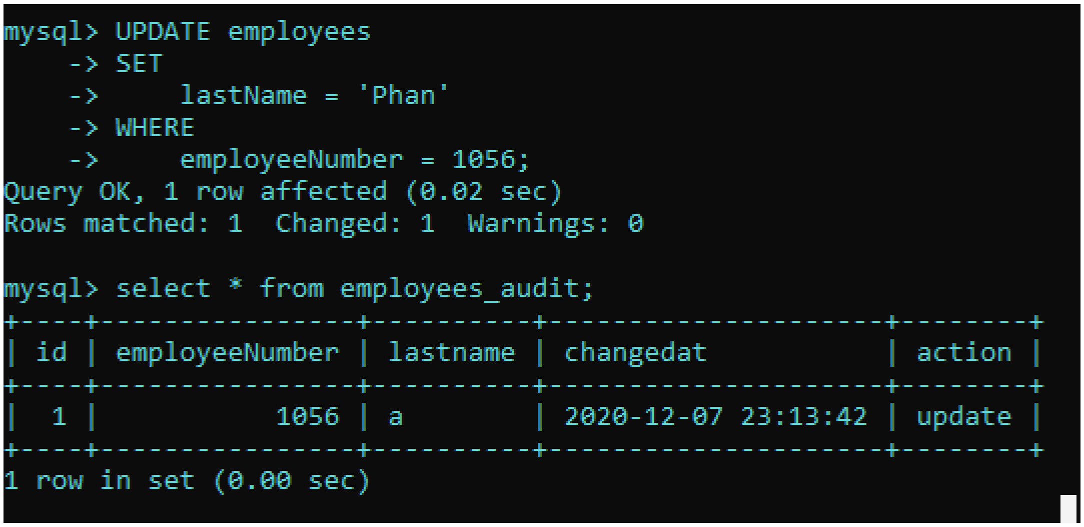
Create triggers



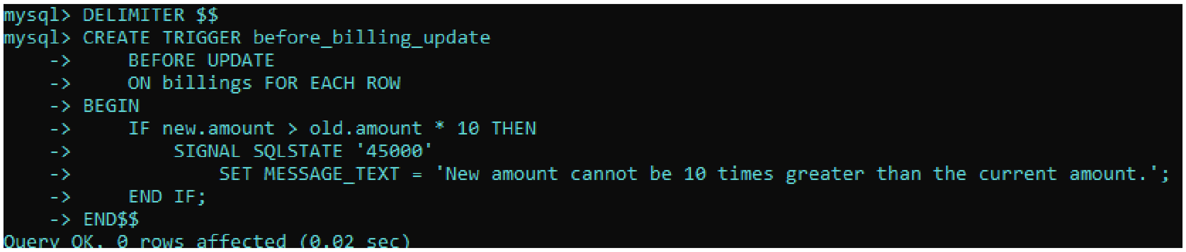
```
mysql> CREATE TRIGGER before_employee_update
    ->     BEFORE UPDATE ON employees
    ->     FOR EACH ROW
    ->  INSERT INTO employees_audit
    ->  SET action = 'update',
    ->      employeeNumber = OLD.employeeNumber,
    ->      lastname = OLD.lastname,
    ->      changedat = NOW();
Query OK, 0 rows affected (0.02 sec)
```



```
mysql> SHOW TRIGGERS;
+------------------------+--------+-----------+----------------------------------------------------------------------------------------------------------------------
-------------------------+--------+------------------------+------------------------------------------------------+-----------------+----------------------+----------------
---+--------------------+
| Trigger                | Event  | Table     | Statement
                          | Timing | Created                | sql_mode                                             | Definer         | character_set_client | collation_connecti
on | Database Collation |
+------------------------+--------+-----------+----------------------------------------------------------------------------------------------------------------------
-------------------------+--------+------------------------+------------------------------------------------------+-----------------+----------------------+----------------
---+--------------------+
| before_employee_update | UPDATE | employees | INSERT INTO employees_audit
 SET action = 'update',
     employeeNumber = OLD.employeeNumber,
     lastname = OLD.lastname,
     changedat = NOW() | BEFORE | 2020-12-07 23:03:53.78 | STRICT_TRANS_TABLES,NO_ENGINE_SUBSTITUTION | root@localhost | cp850                | cp850_general_ci     |
utf8mb4_0900_ai_ci |
+------------------------+--------+-----------+----------------------------------------------------------------------------------------------------------------------
-------------------------+--------+------------------------+------------------------------------------------------+-----------------+----------------------+----------------
---+--------------------+
1 row in set (0.05 sec)
```

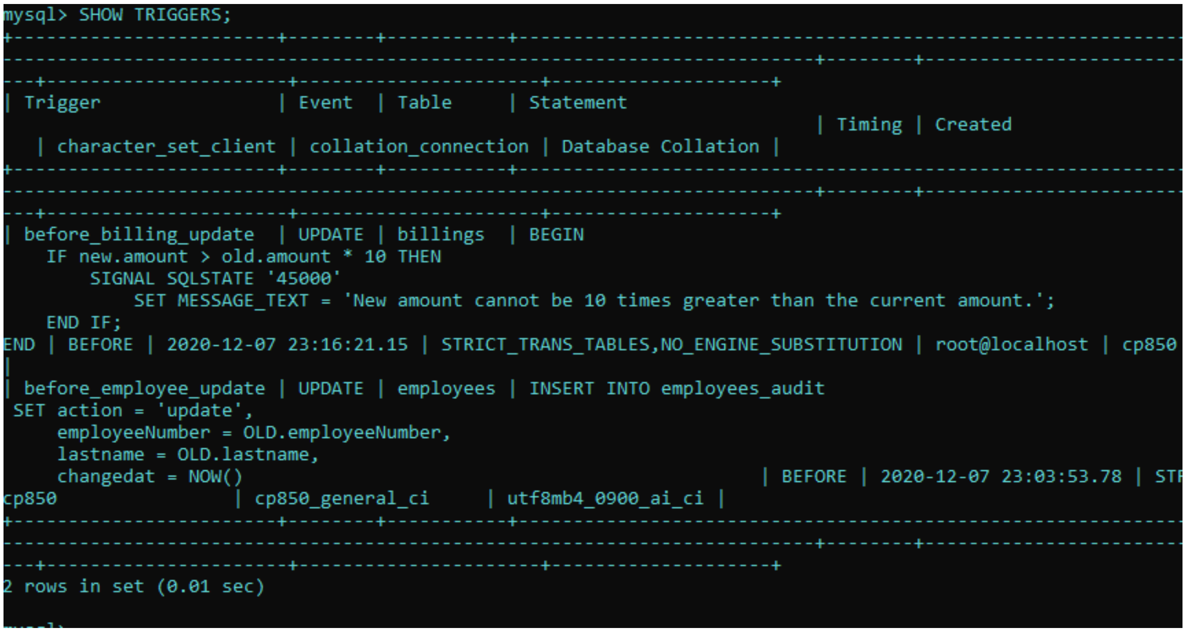


```
mysql> UPDATE employees
    -> SET
    ->     lastName = 'Phan'
    -> WHERE
    ->     employeeNumber = 1056;
Query OK, 1 row affected (0.02 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from employees_audit;
+----+----------------+----------+---------------------+--------+
| id | employeeNumber | lastname | changedat           | action |
+----+----------------+----------+---------------------+--------+
|  1 |           1056 | a        | 2020-12-07 23:13:42 | update |
+----+----------------+----------+---------------------+--------+
1 row in set (0.00 sec)
```

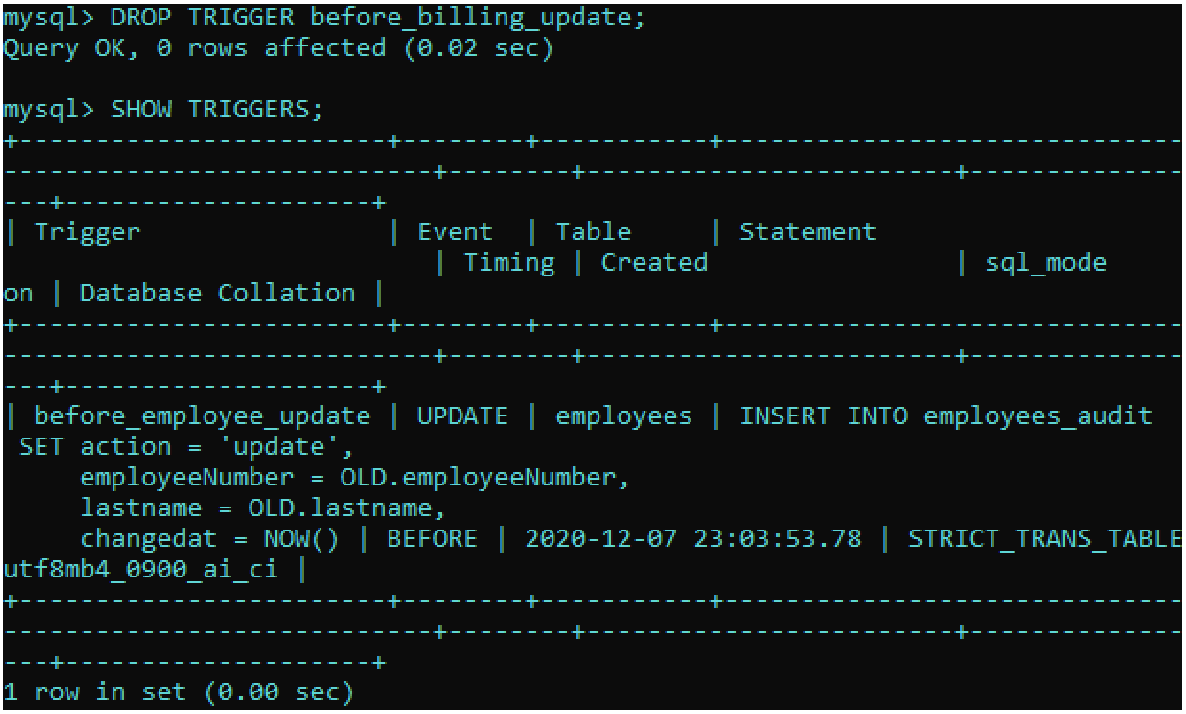


```
mysql> DELIMITER $$
mysql> CREATE TRIGGER before_billing_update
    ->     BEFORE UPDATE
    ->     ON billings FOR EACH ROW
    -> BEGIN
    ->     IF new.amount > old.amount * 10 THEN
    ->         SIGNAL SQLSTATE '45000'
    ->             SET MESSAGE_TEXT = 'New amount cannot be 10 times greater than the current amount.';
    ->     END IF;
    -> END$$
Query OK, 0 rows affected (0.02 sec)
```

Drop triggers

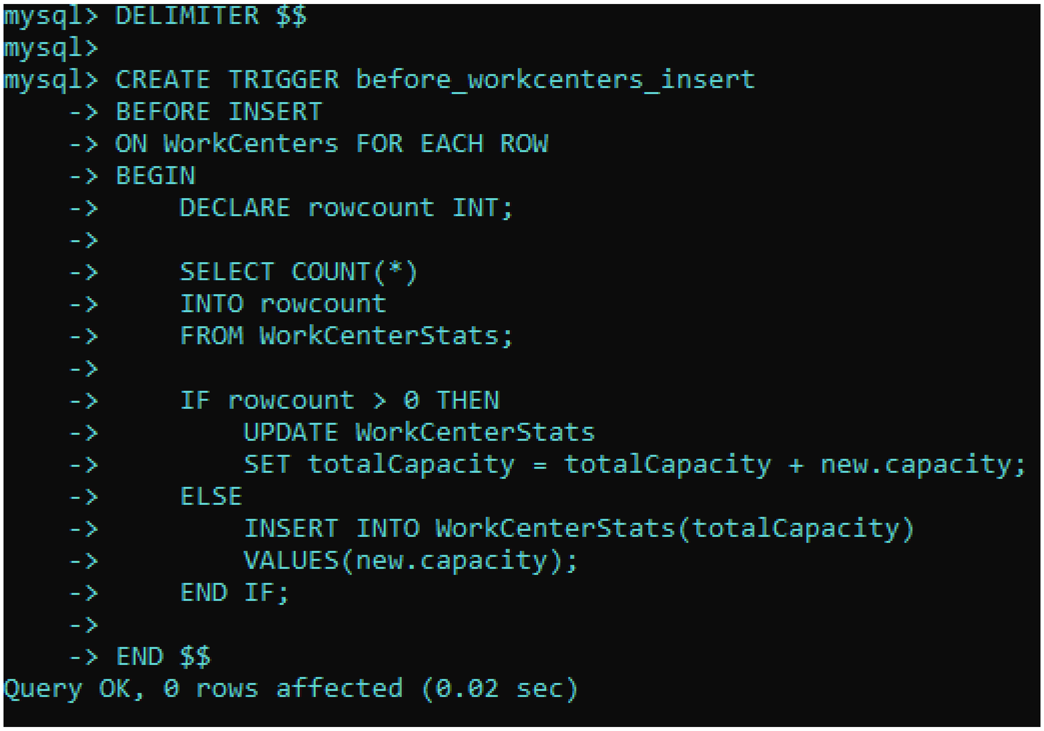


```
mysql> SHOW TRIGGERS;
+-----------------------+--------+-----------+-------------------------------------------------------------
---------------------------------------------------------------------------+--------+-----------------------
---+----------------------+----------------------+---------------------+
| Trigger               | Event  | Table     | Statement
                                                                           | Timing | Created
   | character_set_client | collation_connection | Database Collation |
+-----------------------+--------+-----------+-------------------------------------------------------------
---------------------------------------------------------------------------+--------+-----------------------
---+----------------------+----------------------+---------------------+
| before_billing_update  | UPDATE | billings  | BEGIN
    IF new.amount > old.amount * 10 THEN
        SIGNAL SQLSTATE '45000'
            SET MESSAGE_TEXT = 'New amount cannot be 10 times greater than the current amount.';
    END IF;
END | BEFORE | 2020-12-07 23:16:21.15 | STRICT_TRANS_TABLES,NO_ENGINE_SUBSTITUTION | root@localhost | cp850
|
| before_employee_update | UPDATE | employees | INSERT INTO employees_audit
 SET action = 'update',
     employeeNumber = OLD.employeeNumber,
     lastname = OLD.lastname,
     changedat = NOW()                                                     | BEFORE | 2020-12-07 23:03:53.78 | STR
cp850                | cp850_general_ci     | utf8mb4_0900_ai_ci |
+-----------------------+--------+-----------+-------------------------------------------------------------
---------------------------------------------------------------------------+--------+-----------------------
---+----------------------+----------------------+---------------------+
2 rows in set (0.01 sec)
```

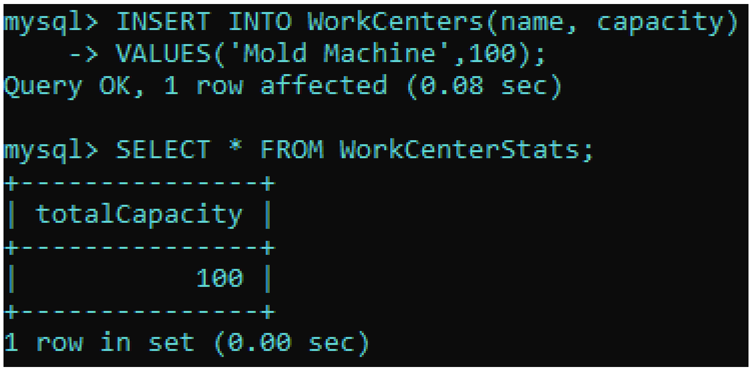


```
mysql> DROP TRIGGER before_billing_update;
Query OK, 0 rows affected (0.02 sec)

mysql> SHOW TRIGGERS;
+------------------------+--------+-----------+------------------------------
----------------------------+--------+------------------------+-------------
---+--------------------+
| Trigger                | Event  | Table     | Statement
                            | Timing | Created                  | sql_mode
on | Database Collation |
+------------------------+--------+-----------+------------------------------
----------------------------+--------+------------------------+-------------
---+--------------------+
| before_employee_update | UPDATE | employees | INSERT INTO employees_audit
 SET action = 'update',
     employeeNumber = OLD.employeeNumber,
     lastname = OLD.lastname,
     changedat = NOW() | BEFORE | 2020-12-07 23:03:53.78 | STRICT_TRANS_TABLE
utf8mb4_0900_ai_ci |
+------------------------+--------+-----------+------------------------------
----------------------------+--------+------------------------+-------------
---+--------------------+
1 row in set (0.00 sec)
```

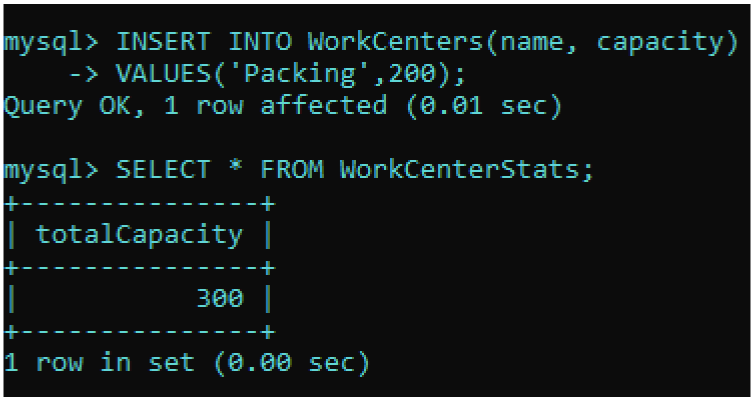
Create a BEFORE INSERT trigger



```
mysql> DELIMITER $$
mysql>
mysql> CREATE TRIGGER before_workcenters_insert
    -> BEFORE INSERT
    -> ON WorkCenters FOR EACH ROW
    -> BEGIN
    ->     DECLARE rowcount INT;
    ->
    ->     SELECT COUNT(*)
    ->     INTO rowcount
    ->     FROM WorkCenterStats;
    ->
    ->     IF rowcount > 0 THEN
    ->         UPDATE WorkCenterStats
    ->         SET totalCapacity = totalCapacity + new.capacity;
    ->     ELSE
    ->         INSERT INTO WorkCenterStats(totalCapacity)
    ->         VALUES(new.capacity);
    ->     END IF;
    ->
    -> END $$
Query OK, 0 rows affected (0.02 sec)
```

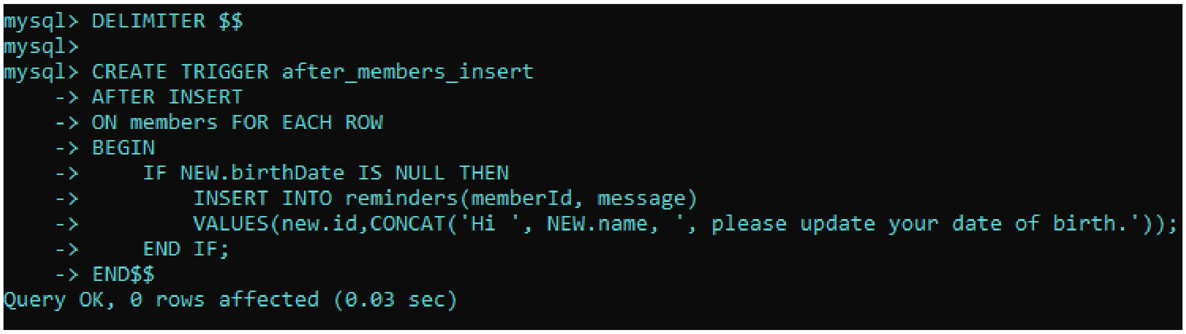


```
mysql> INSERT INTO WorkCenters(name, capacity)
    -> VALUES('Mold Machine',100);
Query OK, 1 row affected (0.08 sec)

mysql> SELECT * FROM WorkCenterStats;
+---------------+
| totalCapacity |
+---------------+
|           100 |
+---------------+
1 row in set (0.00 sec)
```

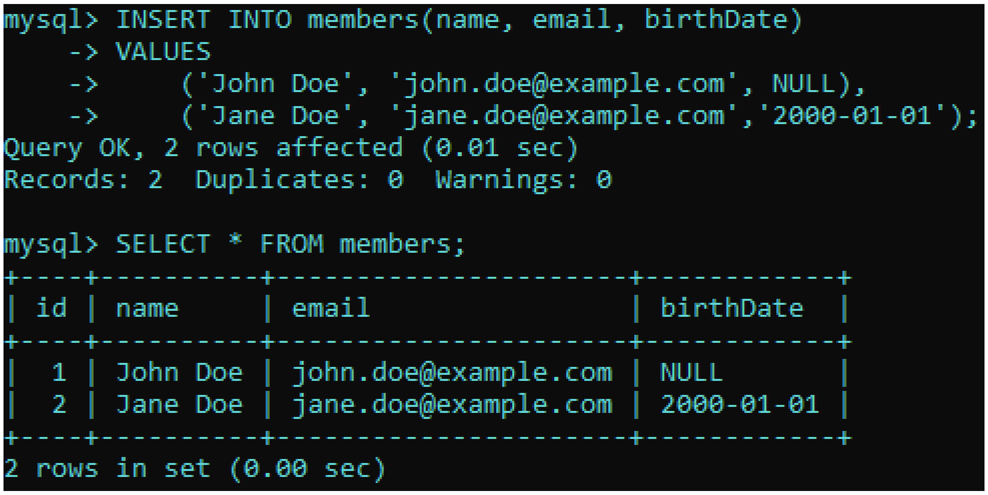


```
mysql> INSERT INTO WorkCenters(name, capacity)
    -> VALUES('Packing',200);
Query OK, 1 row affected (0.01 sec)

mysql> SELECT * FROM WorkCenterStats;
+---------------+
| totalCapacity |
+---------------+
|           300 |
+---------------+
1 row in set (0.00 sec)
```

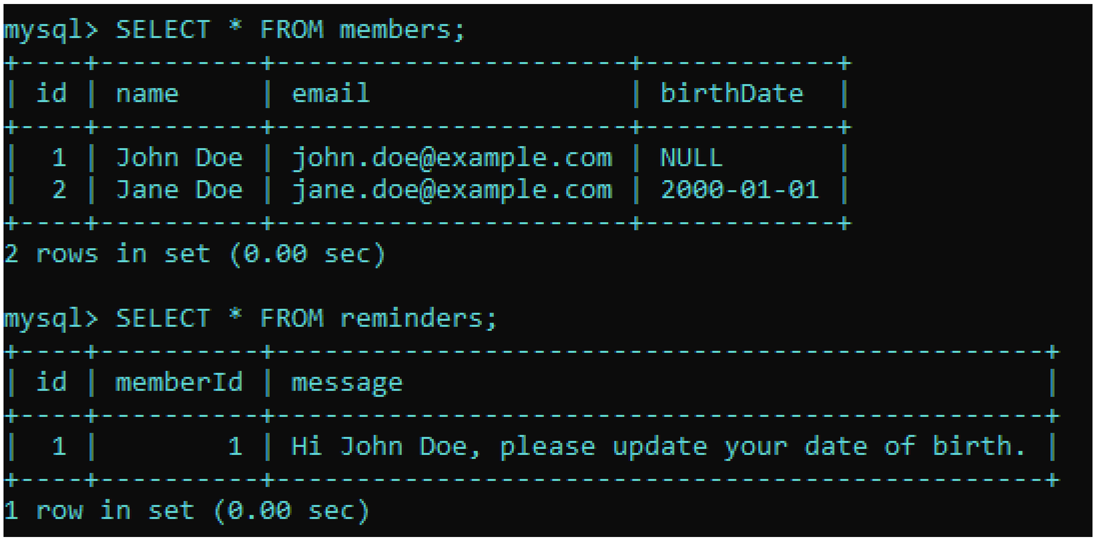
Create an AFTER INSERT trigger



```
mysql> DELIMITER $$
mysql>
mysql> CREATE TRIGGER after_members_insert
    -> AFTER INSERT
    -> ON members FOR EACH ROW
    -> BEGIN
    ->     IF NEW.birthDate IS NULL THEN
    ->         INSERT INTO reminders(memberId, message)
    ->         VALUES(new.id,CONCAT('Hi ', NEW.name, ', please update your date of birth.'));
    ->     END IF;
    -> END$$
Query OK, 0 rows affected (0.03 sec)
```

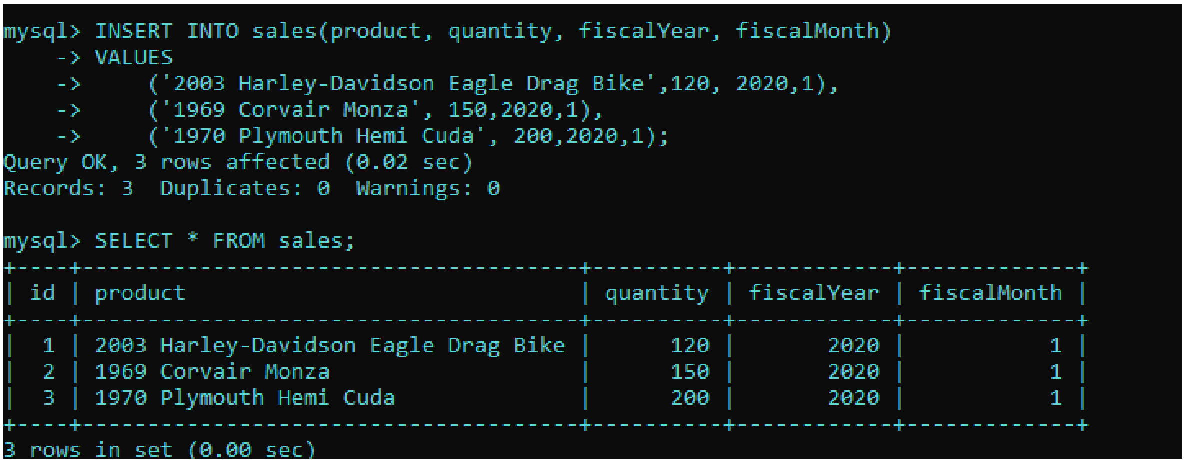


```
mysql> INSERT INTO members(name, email, birthDate)
    -> VALUES
    ->     ('John Doe', 'john.doe@example.com', NULL),
    ->     ('Jane Doe', 'jane.doe@example.com','2000-01-01');
Query OK, 2 rows affected (0.01 sec)
Records: 2  Duplicates: 0  Warnings: 0

mysql> SELECT * FROM members;
+----+----------+----------------------+------------+
| id | name     | email                | birthDate  |
+----+----------+----------------------+------------+
|  1 | John Doe | john.doe@example.com | NULL       |
|  2 | Jane Doe | jane.doe@example.com | 2000-01-01 |
+----+----------+----------------------+------------+
2 rows in set (0.00 sec)
```

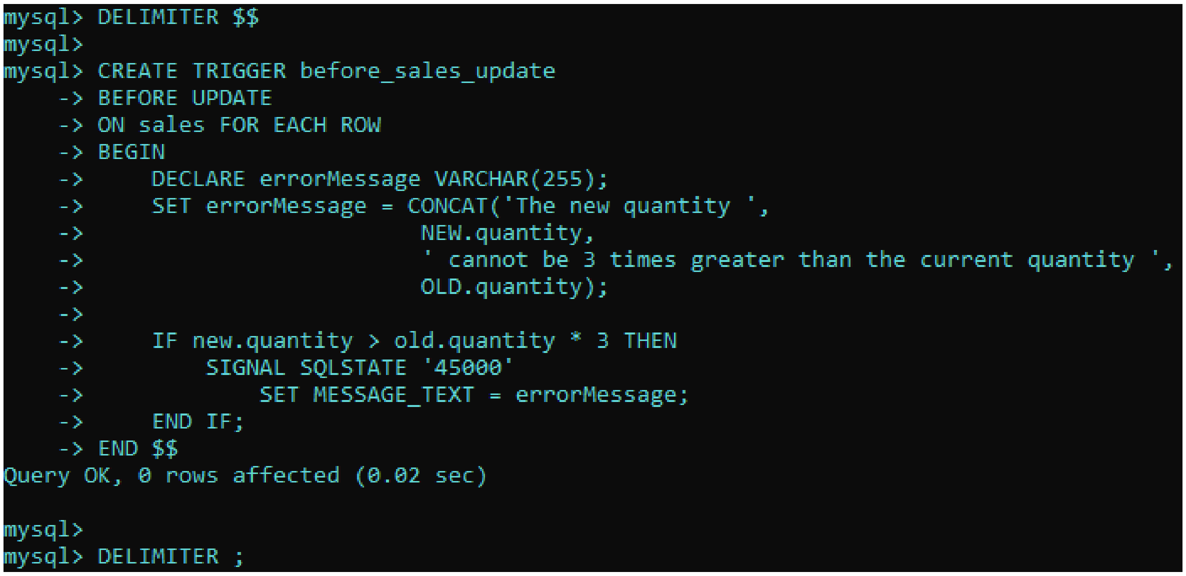


```
mysql> SELECT * FROM members;
+----+----------+----------------------+------------+
| id | name     | email                | birthDate  |
+----+----------+----------------------+------------+
|  1 | John Doe | john.doe@example.com | NULL       |
|  2 | Jane Doe | jane.doe@example.com | 2000-01-01 |
+----+----------+----------------------+------------+
2 rows in set (0.00 sec)

mysql> SELECT * FROM reminders;
+----+----------+-------------------------------------------------+
| id | memberId | message                                         |
+----+----------+-------------------------------------------------+
|  1 |        1 | Hi John Doe, please update your date of birth. |
+----+----------+-------------------------------------------------+
1 row in set (0.00 sec)
```

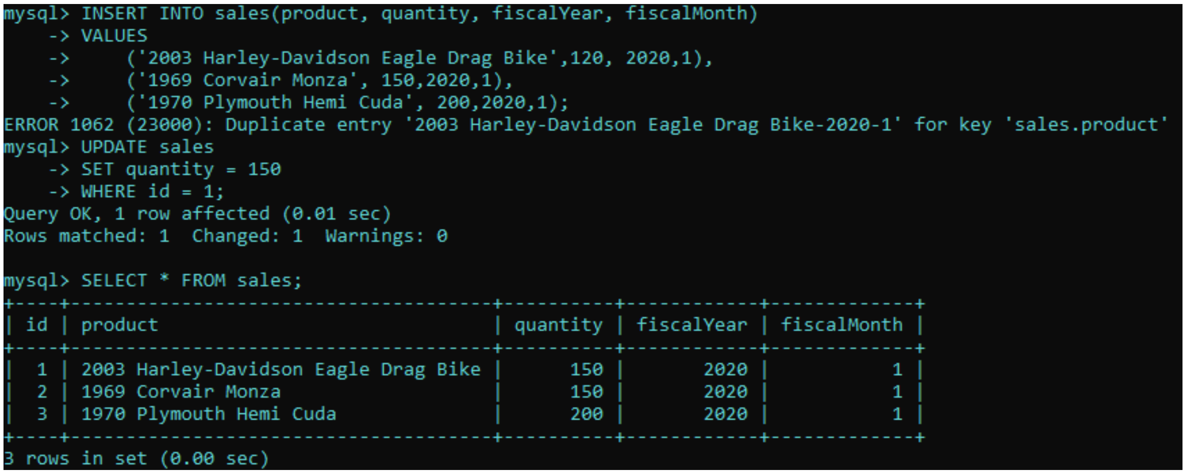
MySQL BEFORE UPDATE Trigger



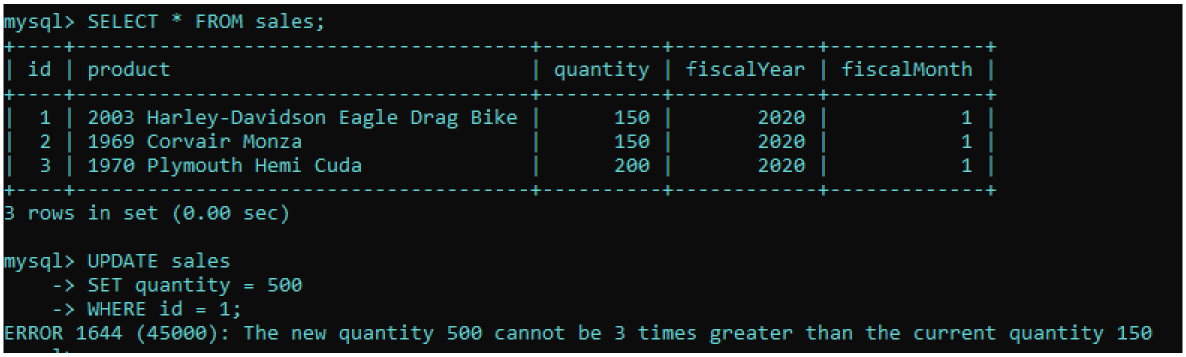
```
mysql> INSERT INTO sales(product, quantity, fiscalYear, fiscalMonth)
    -> VALUES
    ->     ('2003 Harley-Davidson Eagle Drag Bike',120, 2020,1),
    ->     ('1969 Corvair Monza', 150,2020,1),
    ->     ('1970 Plymouth Hemi Cuda', 200,2020,1);
Query OK, 3 rows affected (0.02 sec)
Records: 3  Duplicates: 0  Warnings: 0

mysql> SELECT * FROM sales;
+----+--------------------------------------+----------+------------+-------------+
| id | product                              | quantity | fiscalYear | fiscalMonth |
+----+--------------------------------------+----------+------------+-------------+
|  1 | 2003 Harley-Davidson Eagle Drag Bike |      120 |       2020 |           1 |
|  2 | 1969 Corvair Monza                   |      150 |       2020 |           1 |
|  3 | 1970 Plymouth Hemi Cuda              |      200 |       2020 |           1 |
+----+--------------------------------------+----------+------------+-------------+
3 rows in set (0.00 sec)
```



```
mysql> DELIMITER $$
mysql>
mysql> CREATE TRIGGER before_sales_update
    -> BEFORE UPDATE
    -> ON sales FOR EACH ROW
    -> BEGIN
    ->     DECLARE errorMessage VARCHAR(255);
    ->     SET errorMessage = CONCAT('The new quantity ',
    ->                         NEW.quantity,
    ->                         ' cannot be 3 times greater than the current quantity ',
    ->                         OLD.quantity);
    ->
    ->     IF new.quantity > old.quantity * 3 THEN
    ->         SIGNAL SQLSTATE '45000'
    ->             SET MESSAGE_TEXT = errorMessage;
    ->     END IF;
    -> END $$
Query OK, 0 rows affected (0.02 sec)

mysql>
mysql> DELIMITER ;
```

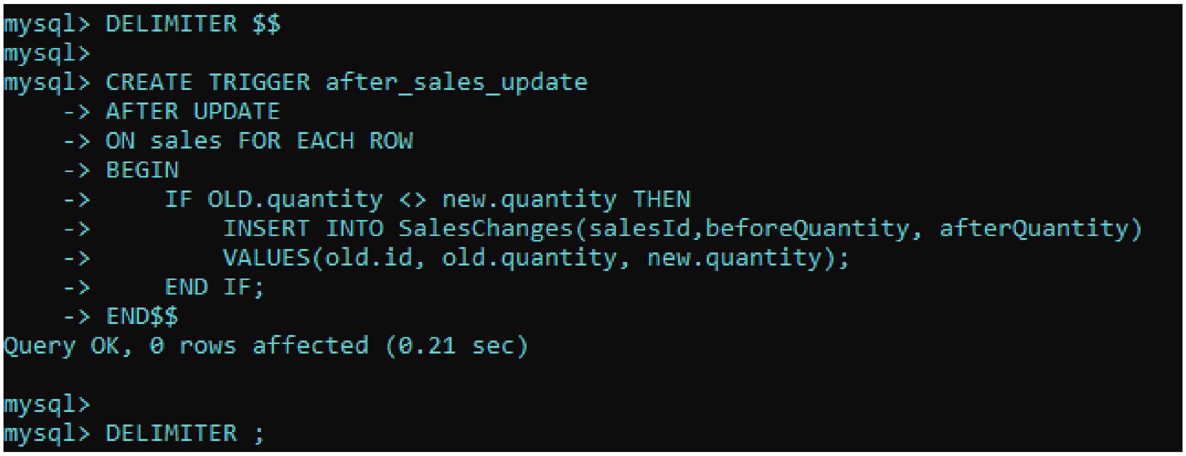


```
mysql> INSERT INTO sales(product, quantity, fiscalYear, fiscalMonth)
    -> VALUES
    ->     ('2003 Harley-Davidson Eagle Drag Bike',120, 2020,1),
    ->     ('1969 Corvair Monza', 150,2020,1),
    ->     ('1970 Plymouth Hemi Cuda', 200,2020,1);
ERROR 1062 (23000): Duplicate entry '2003 Harley-Davidson Eagle Drag Bike-2020-1' for key 'sales.product'
mysql> UPDATE sales
    -> SET quantity = 150
    -> WHERE id = 1;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> SELECT * FROM sales;
+----+--------------------------------------+----------+------------+-------------+
| id | product                              | quantity | fiscalYear | fiscalMonth |
+----+--------------------------------------+----------+------------+-------------+
|  1 | 2003 Harley-Davidson Eagle Drag Bike |      150 |       2020 |           1 |
|  2 | 1969 Corvair Monza                   |      150 |       2020 |           1 |
|  3 | 1970 Plymouth Hemi Cuda              |      200 |       2020 |           1 |
+----+--------------------------------------+----------+------------+-------------+
3 rows in set (0.00 sec)
```

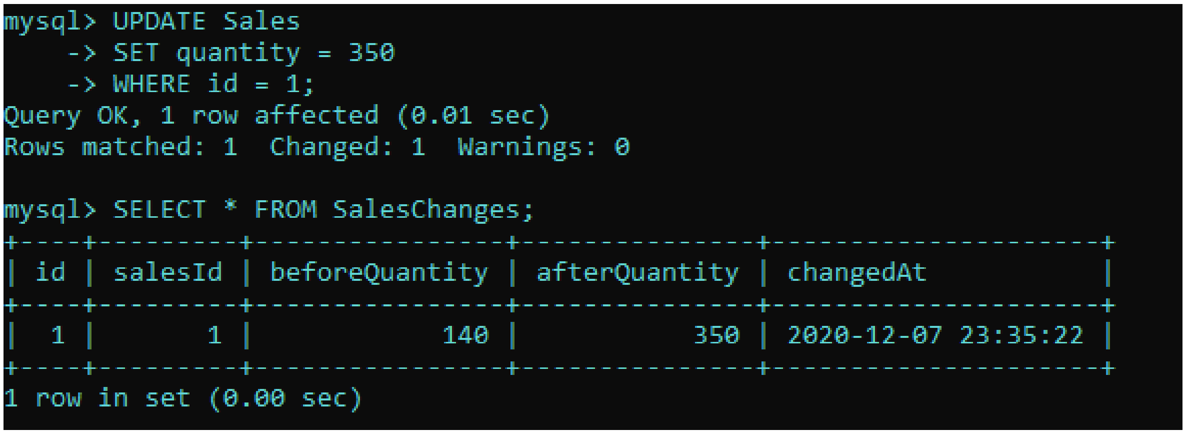


```
mysql> SELECT * FROM sales;
+----+---------------------------------------+----------+------------+-------------+
| id | product                               | quantity | fiscalYear | fiscalMonth |
+----+---------------------------------------+----------+------------+-------------+
|  1 | 2003 Harley-Davidson Eagle Drag Bike |      150 |       2020 |           1 |
|  2 | 1969 Corvair Monza                   |      150 |       2020 |           1 |
|  3 | 1970 Plymouth Hemi Cuda              |      200 |       2020 |           1 |
+----+---------------------------------------+----------+------------+-------------+
3 rows in set (0.00 sec)

mysql> UPDATE sales
    -> SET quantity = 500
    -> WHERE id = 1;
ERROR 1644 (45000): The new quantity 500 cannot be 3 times greater than the current quantity 150
```

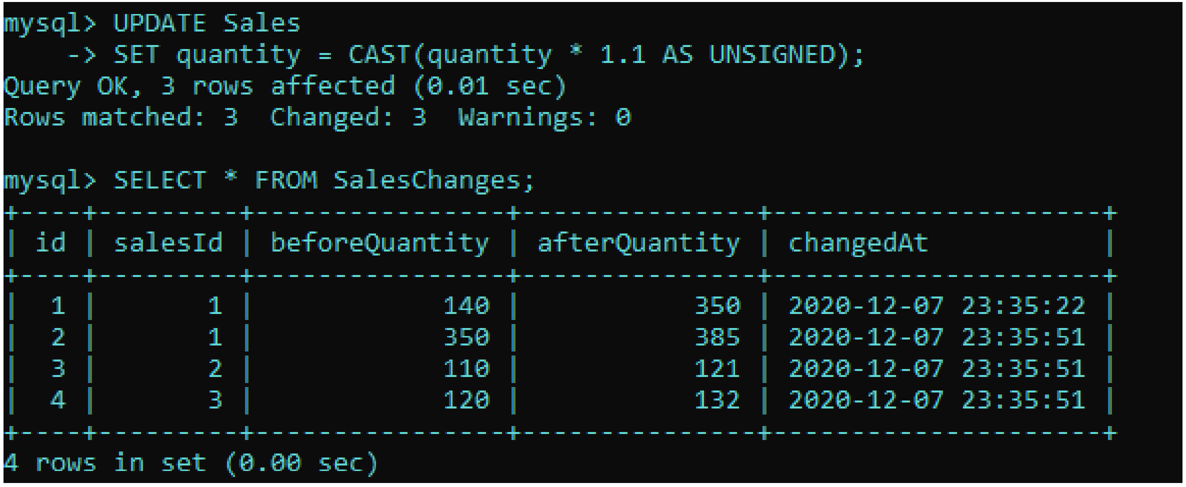
MySQL AFTER UPDATE Trigger



```
mysql> DELIMITER $$
mysql>
mysql> CREATE TRIGGER after_sales_update
    -> AFTER UPDATE
    -> ON sales FOR EACH ROW
    -> BEGIN
    ->     IF OLD.quantity <> new.quantity THEN
    ->         INSERT INTO SalesChanges(salesId,beforeQuantity, afterQuantity)
    ->         VALUES(old.id, old.quantity, new.quantity);
    ->     END IF;
    -> END$$
Query OK, 0 rows affected (0.21 sec)

mysql>
mysql> DELIMITER ;
```

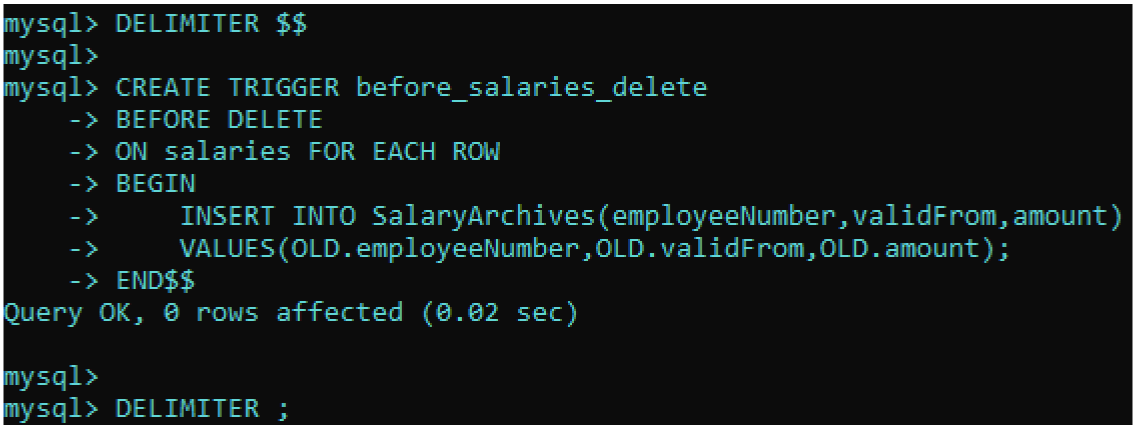


```
mysql> UPDATE Sales
    -> SET quantity = 350
    -> WHERE id = 1;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> SELECT * FROM SalesChanges;
+----+---------+----------------+---------------+---------------------+
| id | salesId | beforeQuantity | afterQuantity | changedAt           |
+----+---------+----------------+---------------+---------------------+
|  1 |       1 |            140 |           350 | 2020-12-07 23:35:22 |
+----+---------+----------------+---------------+---------------------+
1 row in set (0.00 sec)
```

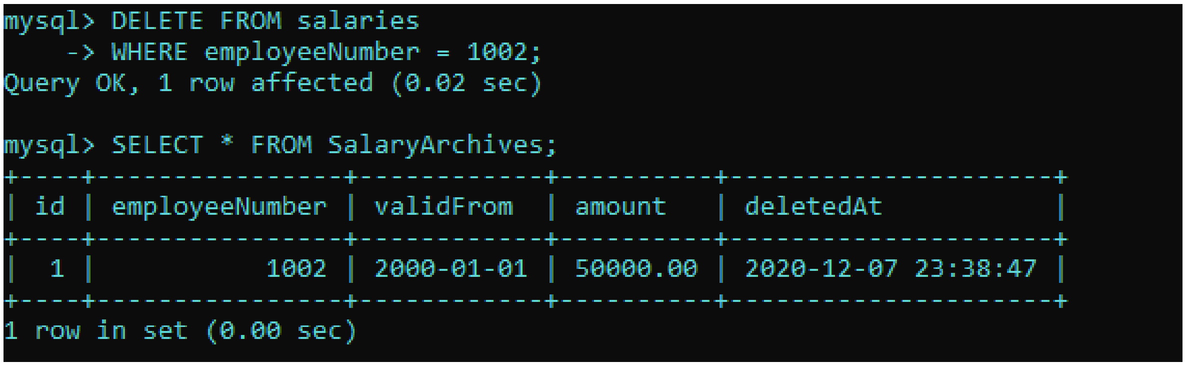


```
mysql> UPDATE Sales
    -> SET quantity = CAST(quantity * 1.1 AS UNSIGNED);
Query OK, 3 rows affected (0.01 sec)
Rows matched: 3  Changed: 3  Warnings: 0

mysql> SELECT * FROM SalesChanges;
+----+---------+----------------+---------------+---------------------+
| id | salesId | beforeQuantity | afterQuantity | changedAt           |
+----+---------+----------------+---------------+---------------------+
|  1 |       1 |            140 |           350 | 2020-12-07 23:35:22 |
|  2 |       1 |            350 |           385 | 2020-12-07 23:35:51 |
|  3 |       2 |            110 |           121 | 2020-12-07 23:35:51 |
|  4 |       3 |            120 |           132 | 2020-12-07 23:35:51 |
+----+---------+----------------+---------------+---------------------+
4 rows in set (0.00 sec)
```

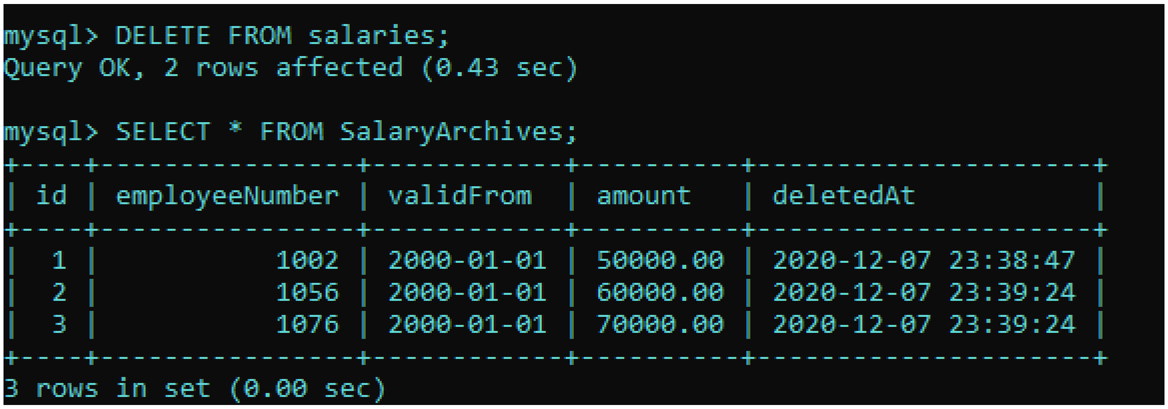
MySQL BEFORE DELETE Trigger



```
mysql> DELIMITER $$
mysql>
mysql> CREATE TRIGGER before_salaries_delete
    -> BEFORE DELETE
    -> ON salaries FOR EACH ROW
    -> BEGIN
    ->     INSERT INTO SalaryArchives(employeeNumber,validFrom,amount)
    ->     VALUES(OLD.employeeNumber,OLD.validFrom,OLD.amount);
    -> END$$
Query OK, 0 rows affected (0.02 sec)

mysql>
mysql> DELIMITER ;
```

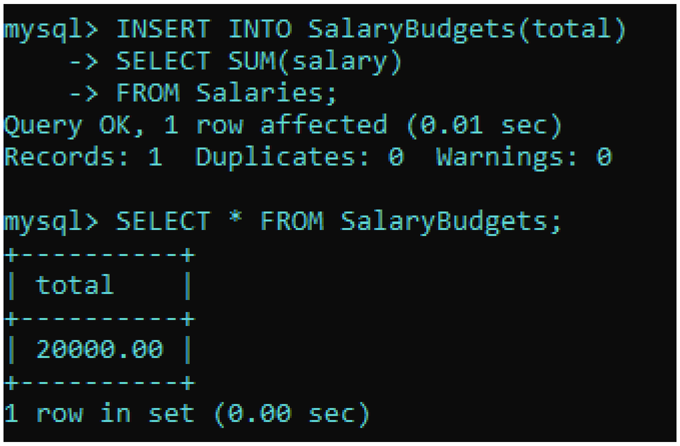


```
mysql> DELETE FROM salaries
    -> WHERE employeeNumber = 1002;
Query OK, 1 row affected (0.02 sec)

mysql> SELECT * FROM SalaryArchives;
+----+----------------+------------+----------+---------------------+
| id | employeeNumber | validFrom  | amount   | deletedAt           |
+----+----------------+------------+----------+---------------------+
|  1 |           1002 | 2000-01-01 | 50000.00 | 2020-12-07 23:38:47 |
+----+----------------+------------+----------+---------------------+
1 row in set (0.00 sec)
```

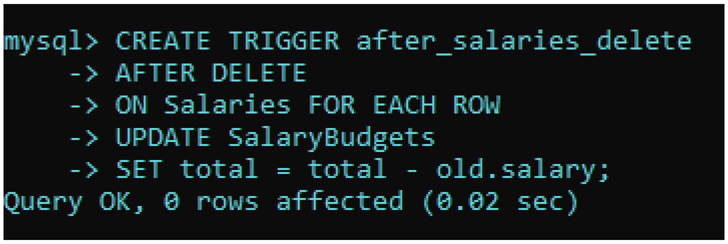


```
mysql> DELETE FROM salaries;
Query OK, 2 rows affected (0.43 sec)

mysql> SELECT * FROM SalaryArchives;
+----+----------------+------------+----------+---------------------+
| id | employeeNumber | validFrom  | amount   | deletedAt           |
+----+----------------+------------+----------+---------------------+
|  1 |           1002 | 2000-01-01 | 50000.00 | 2020-12-07 23:38:47 |
|  2 |           1056 | 2000-01-01 | 60000.00 | 2020-12-07 23:39:24 |
|  3 |           1076 | 2000-01-01 | 70000.00 | 2020-12-07 23:39:24 |
+----+----------------+------------+----------+---------------------+
3 rows in set (0.00 sec)
```

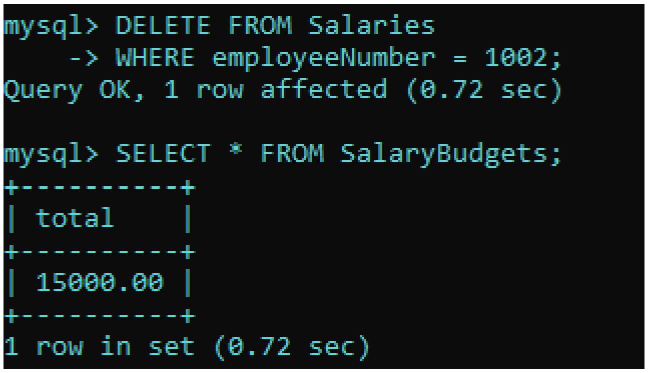
MySQL AFTER DELETE Trigger



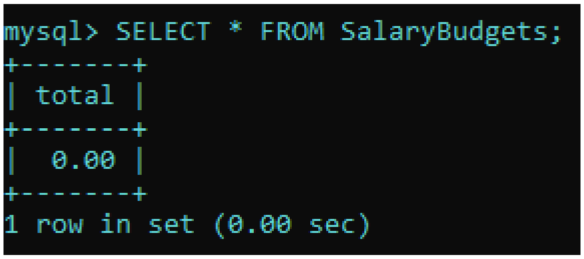
```
mysql> INSERT INTO SalaryBudgets(total)
    -> SELECT SUM(salary)
    -> FROM Salaries;
Query OK, 1 row affected (0.01 sec)
Records: 1  Duplicates: 0  Warnings: 0

mysql> SELECT * FROM SalaryBudgets;
+----------+
| total    |
+----------+
| 20000.00 |
+----------+
1 row in set (0.00 sec)
```



```
mysql> CREATE TRIGGER after_salaries_delete
    -> AFTER DELETE
    -> ON Salaries FOR EACH ROW
    -> UPDATE SalaryBudgets
    -> SET total = total - old.salary;
Query OK, 0 rows affected (0.02 sec)
```

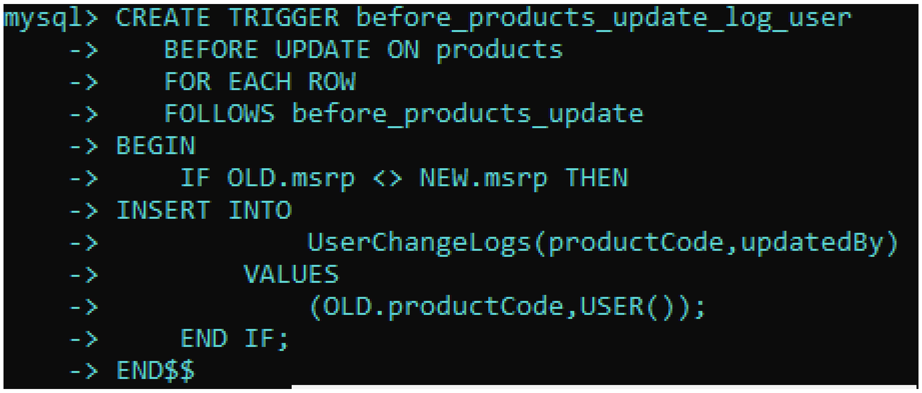


```
mysql> DELETE FROM Salaries
    -> WHERE employeeNumber = 1002;
Query OK, 1 row affected (0.72 sec)

mysql> SELECT * FROM SalaryBudgets;
+----------+
| total    |
+----------+
| 15000.00 |
+----------+
1 row in set (0.72 sec)
```

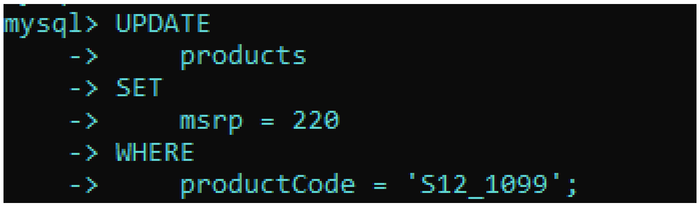


```
mysql> SELECT * FROM SalaryBudgets;
+-------+
| total |
+-------+
|  0.00 |
+-------+
1 row in set (0.00 sec)
```

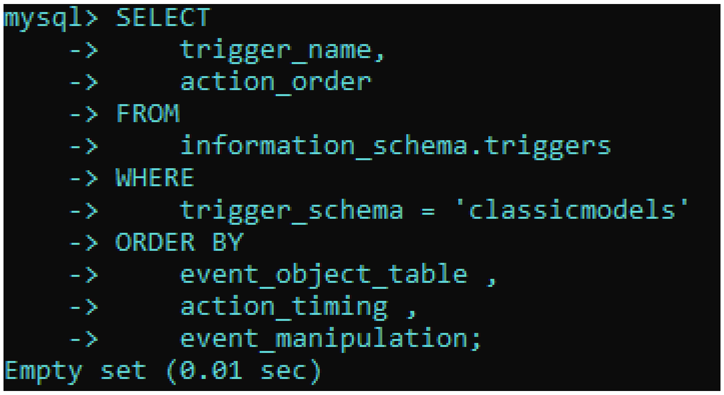
Create Multiple Triggers



```
mysql> CREATE TRIGGER before_products_update_log_user
    ->    BEFORE UPDATE ON products
    ->    FOR EACH ROW
    ->    FOLLOWS before_products_update
    -> BEGIN
    ->     IF OLD.msrp <> NEW.msrp THEN
    -> INSERT INTO
    ->             UserChangeLogs(productCode,updatedBy)
    ->         VALUES
    ->             (OLD.productCode,USER());
    ->     END IF;
    -> END$$
```

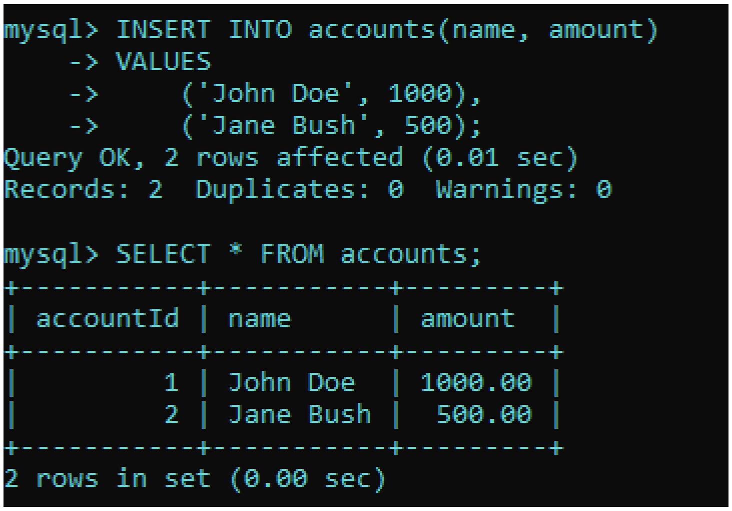


```
mysql> UPDATE
    ->     products
    -> SET
    ->     msrp = 220
    -> WHERE
    ->     productCode = 'S12_1099';
```

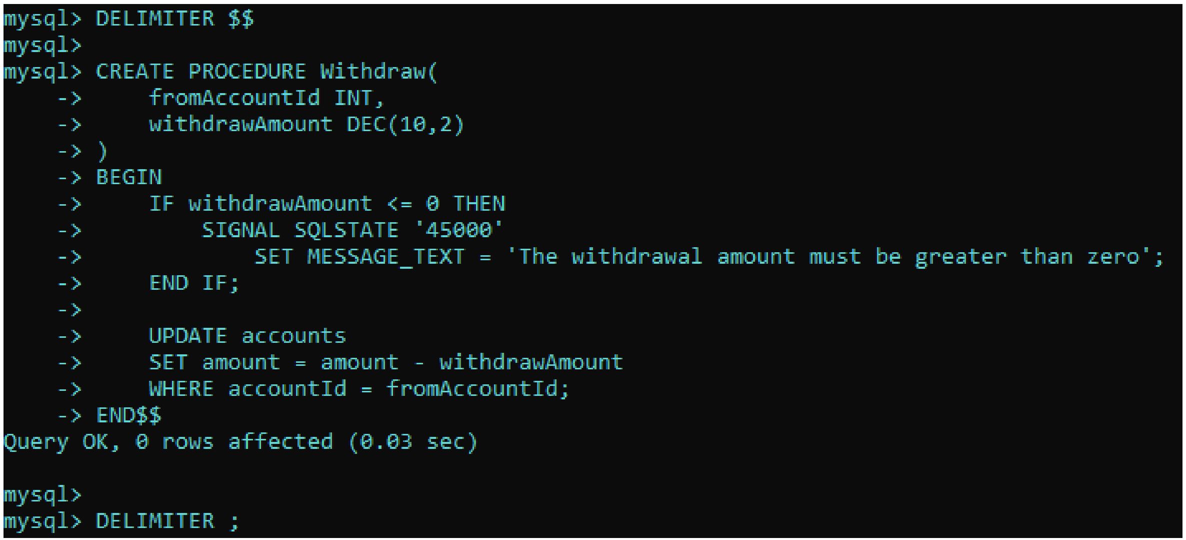


```
mysql> SELECT
    ->     trigger_name,
    ->     action_order
    -> FROM
    ->     information_schema.triggers
    -> WHERE
    ->     trigger_schema = 'classicmodels'
    -> ORDER BY
    ->     event_object_table ,
    ->     action_timing ,
    ->     event_manipulation;
Empty set (0.01 sec)
```

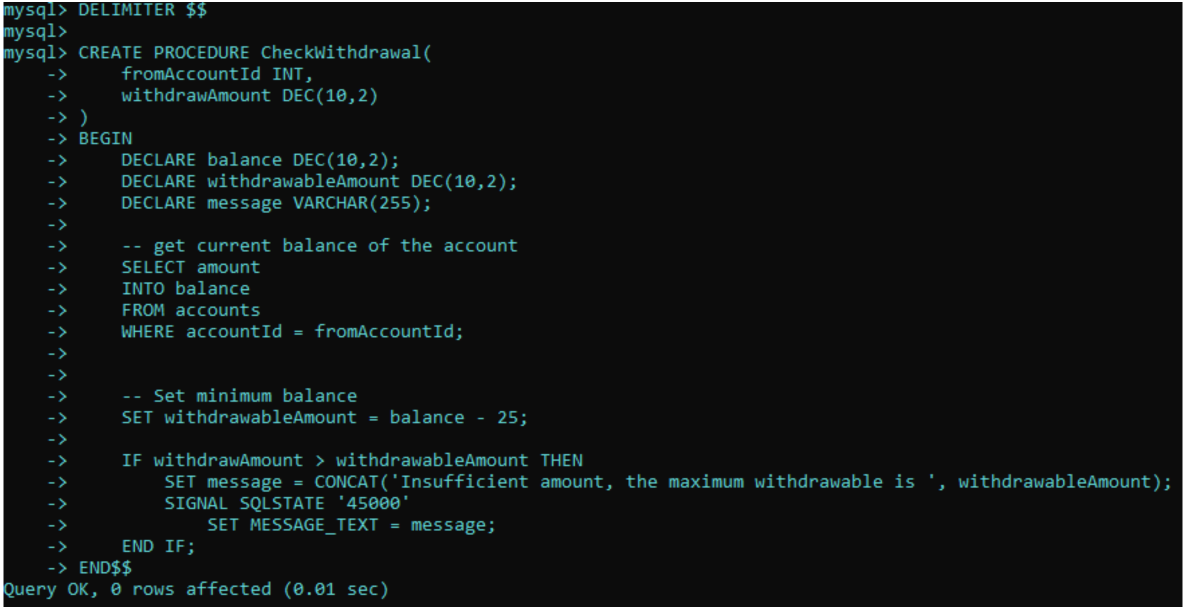
How To Call a Stored Procedure From a Trigger in MySQL.



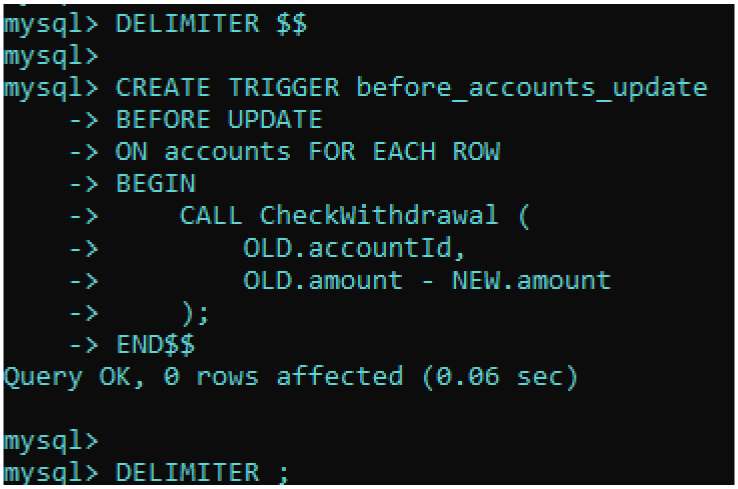
```
mysql> INSERT INTO accounts(name, amount)
    -> VALUES
    ->     ('John Doe', 1000),
    ->     ('Jane Bush', 500);
Query OK, 2 rows affected (0.01 sec)
Records: 2  Duplicates: 0  Warnings: 0

mysql> SELECT * FROM accounts;
+-----------+-----------+---------+
| accountId | name      | amount  |
+-----------+-----------+---------+
|         1 | John Doe  | 1000.00 |
|         2 | Jane Bush |  500.00 |
+-----------+-----------+---------+
2 rows in set (0.00 sec)
```



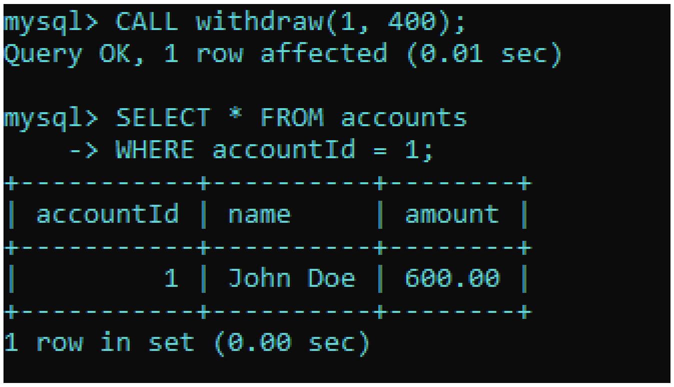
```
mysql> DELIMITER $$
mysql>
mysql> CREATE PROCEDURE Withdraw(
    ->     fromAccountId INT,
    ->     withdrawAmount DEC(10,2)
    -> )
    -> BEGIN
    ->     IF withdrawAmount <= 0 THEN
    ->         SIGNAL SQLSTATE '45000'
    ->             SET MESSAGE_TEXT = 'The withdrawal amount must be greater than zero';
    ->     END IF;
    ->
    ->     UPDATE accounts
    ->     SET amount = amount - withdrawAmount
    ->     WHERE accountId = fromAccountId;
    -> END$$
Query OK, 0 rows affected (0.03 sec)

mysql>
mysql> DELIMITER ;
```



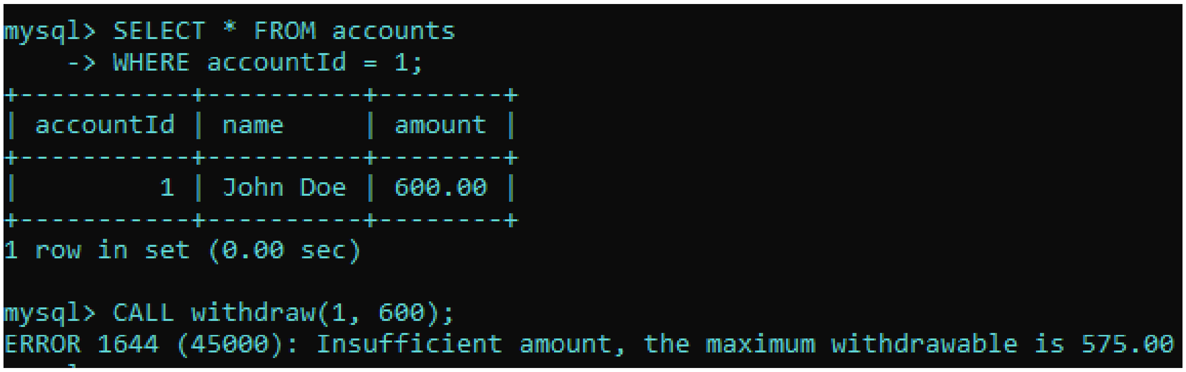
```
mysql> DELIMITER $$
mysql>
mysql> CREATE PROCEDURE CheckWithdrawal(
    ->     fromAccountId INT,
    ->     withdrawAmount DEC(10,2)
    -> )
    -> BEGIN
    ->     DECLARE balance DEC(10,2);
    ->     DECLARE withdrawableAmount DEC(10,2);
    ->     DECLARE message VARCHAR(255);
    ->
    ->     -- get current balance of the account
    ->     SELECT amount
    ->     INTO balance
    ->     FROM accounts
    ->     WHERE accountId = fromAccountId;
    ->
    ->
    ->     -- Set minimum balance
    ->     SET withdrawableAmount = balance - 25;
    ->
    ->     IF withdrawAmount > withdrawableAmount THEN
    ->         SET message = CONCAT('Insufficient amount, the maximum withdrawable is ', withdrawableAmount);
    ->         SIGNAL SQLSTATE '45000'
    ->             SET MESSAGE_TEXT = message;
    ->     END IF;
    -> END$$
Query OK, 0 rows affected (0.01 sec)
```



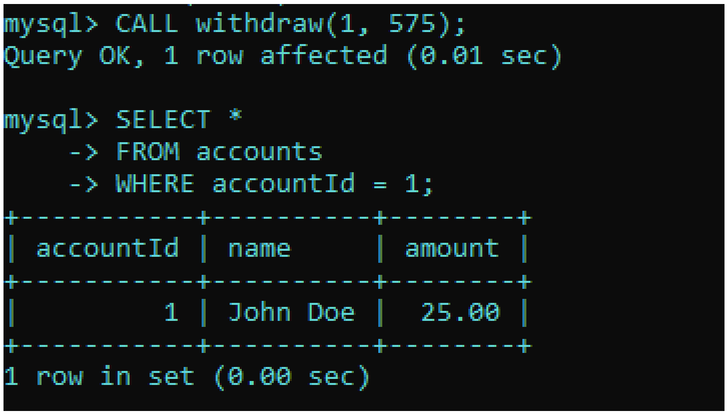
```
mysql> DELIMITER $$
mysql>
mysql> CREATE TRIGGER before_accounts_update
    -> BEFORE UPDATE
    -> ON accounts FOR EACH ROW
    -> BEGIN
    ->     CALL CheckWithdrawal (
    ->         OLD.accountId,
    ->         OLD.amount - NEW.amount
    ->     );
    -> END$$
Query OK, 0 rows affected (0.06 sec)

mysql>
mysql> DELIMITER ;
```



```
mysql> CALL withdraw(1, 400);
Query OK, 1 row affected (0.01 sec)

mysql> SELECT * FROM accounts
    -> WHERE accountId = 1;
+-----------+----------+--------+
| accountId | name     | amount |
+-----------+----------+--------+
|         1 | John Doe | 600.00 |
+-----------+----------+--------+
1 row in set (0.00 sec)
```

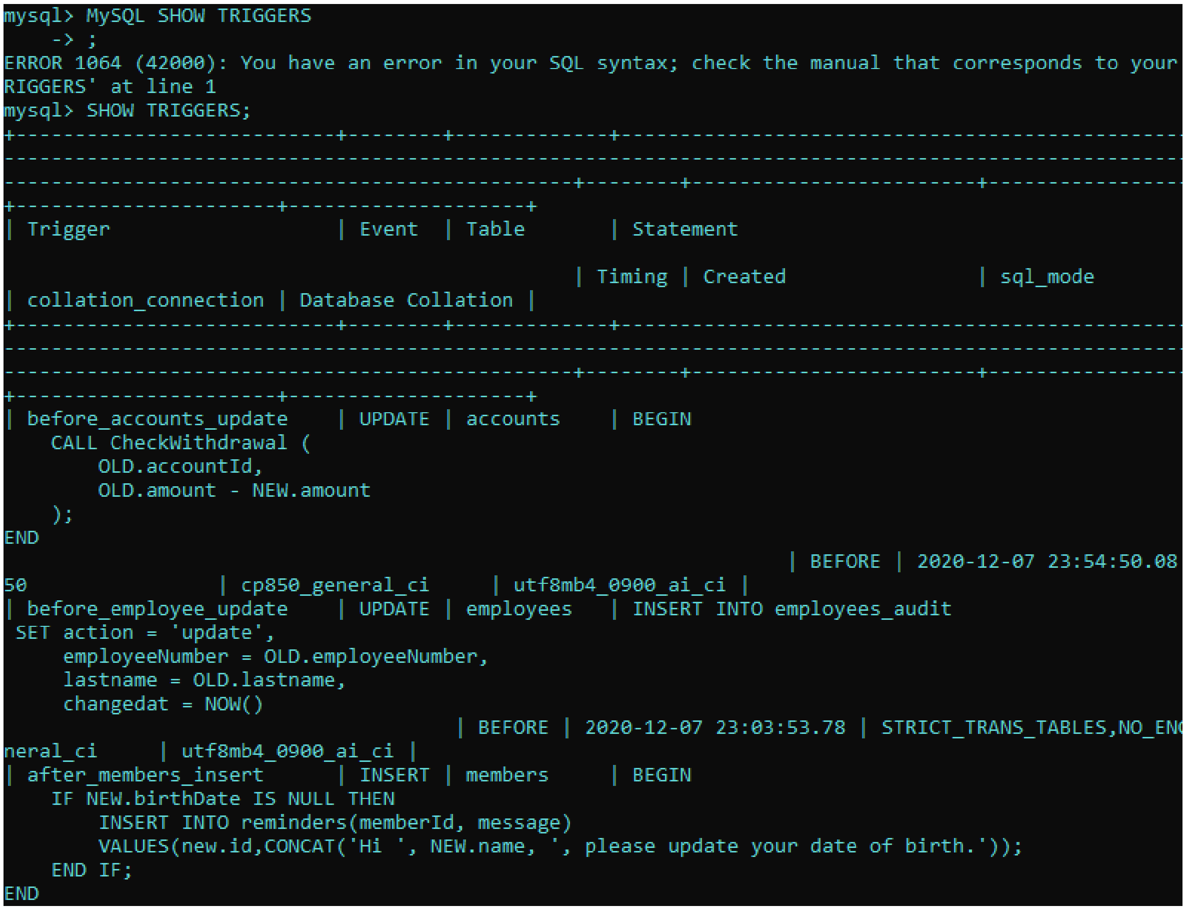


```
mysql> SELECT * FROM accounts
    -> WHERE accountId = 1;
+-----------+----------+--------+
| accountId | name     | amount |
+-----------+----------+--------+
|         1 | John Doe | 600.00 |
+-----------+----------+--------+
1 row in set (0.00 sec)

mysql> CALL withdraw(1, 600);
ERROR 1644 (45000): Insufficient amount, the maximum withdrawable is 575.00
```



```
mysql> CALL withdraw(1, 575);
Query OK, 1 row affected (0.01 sec)

mysql> SELECT *
    -> FROM accounts
    -> WHERE accountId = 1;
+-----------+----------+--------+
| accountId | name     | amount |
+-----------+----------+--------+
|         1 | John Doe |  25.00 |
+-----------+----------+--------+
1 row in set (0.00 sec)
```

MySQL SHOW TRIGGERS



```
mysql> MySQL SHOW TRIGGERS
    -> ;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your
RIGGERS' at line 1
mysql> SHOW TRIGGERS;
+---------------------------+--------+-------------+-----------------------------------------------
---------------------------------------------------------------------------------------------------
------------------------------------------------+--------+------------------------+----------------
+----------------------+--------------------+
| Trigger                   | Event  | Table       | Statement

                                                | Timing | Created                | sql_mode
| collation_connection | Database Collation |
+---------------------------+--------+-------------+-----------------------------------------------
---------------------------------------------------------------------------------------------------
------------------------------------------------+--------+------------------------+----------------
+----------------------+--------------------+
| before_accounts_update    | UPDATE | accounts    | BEGIN
    CALL CheckWithdrawal (
        OLD.accountId,
        OLD.amount - NEW.amount
    );
END
                                                                  | BEFORE | 2020-12-07 23:54:50.08
50                | cp850_general_ci     | utf8mb4_0900_ai_ci |
| before_employee_update    | UPDATE | employees   | INSERT INTO employees_audit
 SET action = 'update',
     employeeNumber = OLD.employeeNumber,
     lastname = OLD.lastname,
     changedat = NOW()
                                      | BEFORE | 2020-12-07 23:03:53.78 | STRICT_TRANS_TABLES,NO_ENG
neral_ci     | utf8mb4_0900_ai_ci |
| after_members_insert      | INSERT | members     | BEGIN
    IF NEW.birthDate IS NULL THEN
        INSERT INTO reminders(memberId, message)
        VALUES(new.id,CONCAT('Hi ', NEW.name, ', please update your date of birth.'));
    END IF;
END
```