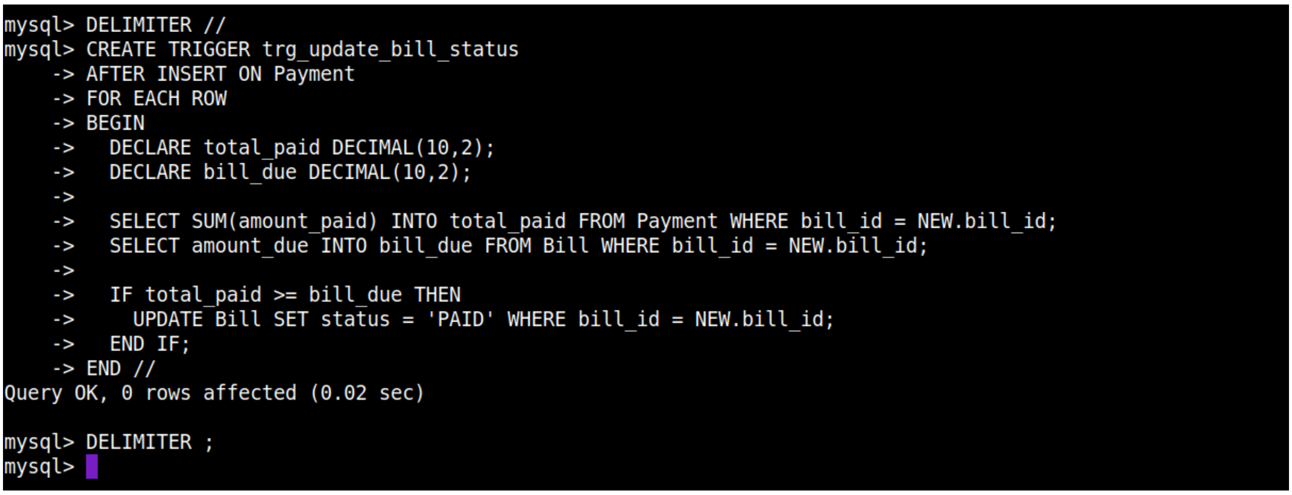
TRIGGERS

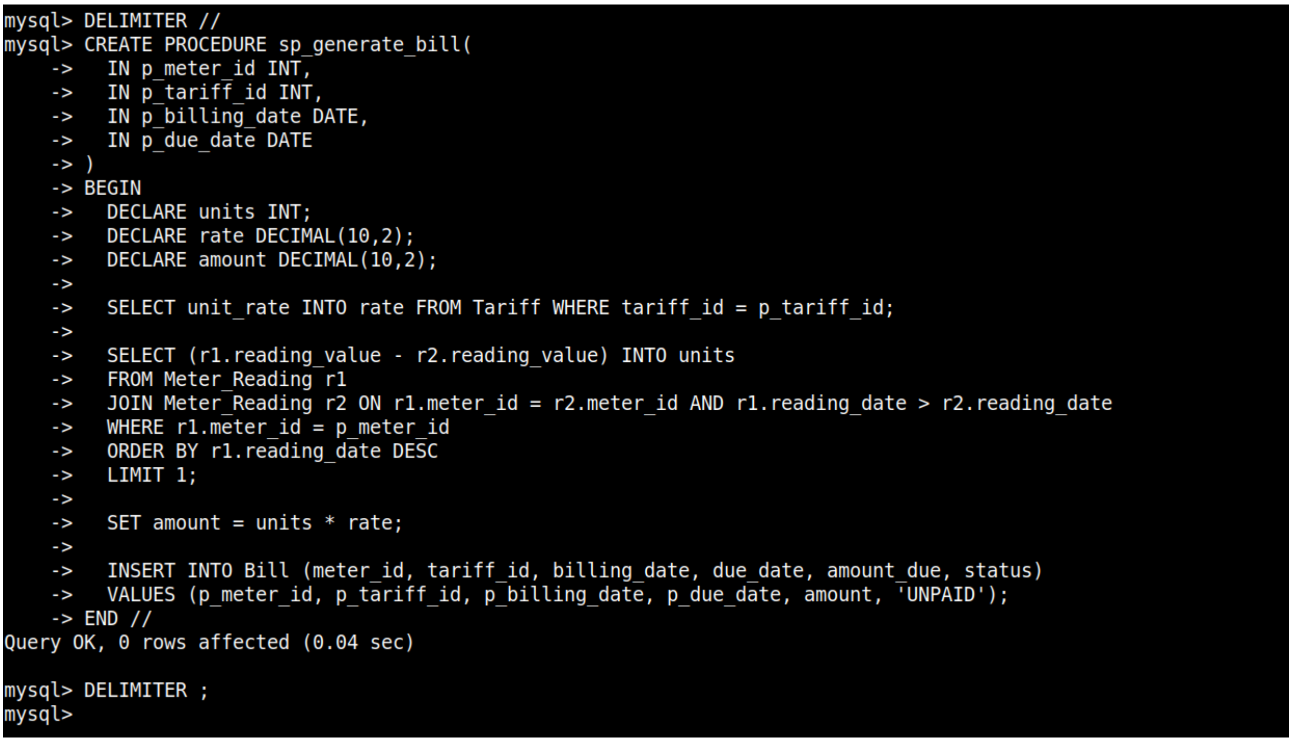
Updating the Bill status



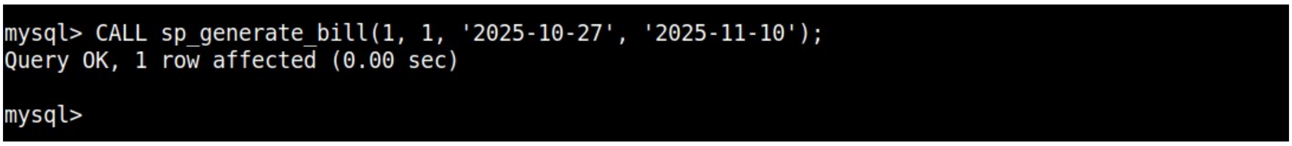
```
mysql> DELIMITER //
mysql> CREATE TRIGGER trg_update_bill_status
    -> AFTER INSERT ON Payment
    -> FOR EACH ROW
    -> BEGIN
    ->   DECLARE total_paid DECIMAL(10,2);
    ->   DECLARE bill_due DECIMAL(10,2);
    ->
    ->   SELECT SUM(amount_paid) INTO total_paid FROM Payment WHERE bill_id = NEW.bill_id;
    ->   SELECT amount_due INTO bill_due FROM Bill WHERE bill_id = NEW.bill_id;
    ->
    ->   IF total_paid >= bill_due THEN
    ->     UPDATE Bill SET status = 'PAID' WHERE bill_id = NEW.bill_id;
    ->   END IF;
    -> END //
Query OK, 0 rows affected (0.02 sec)

mysql> DELIMITER ;
mysql> 
```

PROCEDURES

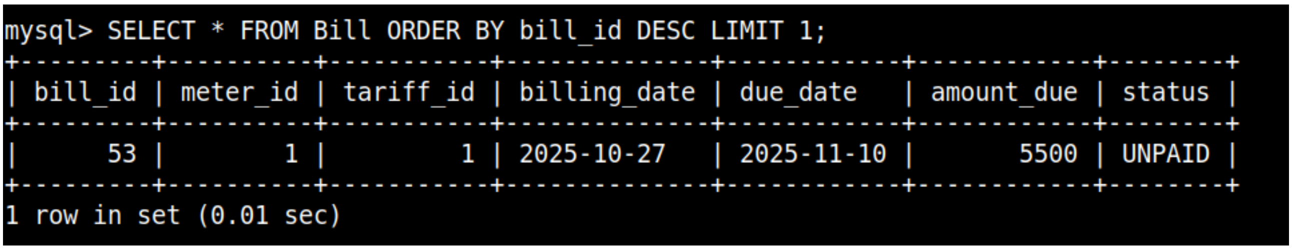
Generate Bill



```
mysql> DELIMITER //
mysql> CREATE PROCEDURE sp_generate_bill(
    ->   IN p_meter_id INT,
    ->   IN p_tariff_id INT,
    ->   IN p_billing_date DATE,
    ->   IN p_due_date DATE
    -> )
    -> BEGIN
    ->   DECLARE units INT;
    ->   DECLARE rate DECIMAL(10,2);
    ->   DECLARE amount DECIMAL(10,2);
    ->
    ->   SELECT unit_rate INTO rate FROM Tariff WHERE tariff_id = p_tariff_id;
    ->
    ->   SELECT (r1.reading_value - r2.reading_value) INTO units
    ->   FROM Meter_Reading r1
    ->   JOIN Meter_Reading r2 ON r1.meter_id = r2.meter_id AND r1.reading_date > r2.reading_date
    ->   WHERE r1.meter_id = p_meter_id
    ->   ORDER BY r1.reading_date DESC
    ->   LIMIT 1;
    ->
    ->   SET amount = units * rate;
    ->
    ->   INSERT INTO Bill (meter_id, tariff_id, billing_date, due_date, amount_due, status)
    ->   VALUES (p_meter_id, p_tariff_id, p_billing_date, p_due_date, amount, 'UNPAID');
    -> END //
Query OK, 0 rows affected (0.04 sec)

mysql> DELIMITER ;
mysql>
```



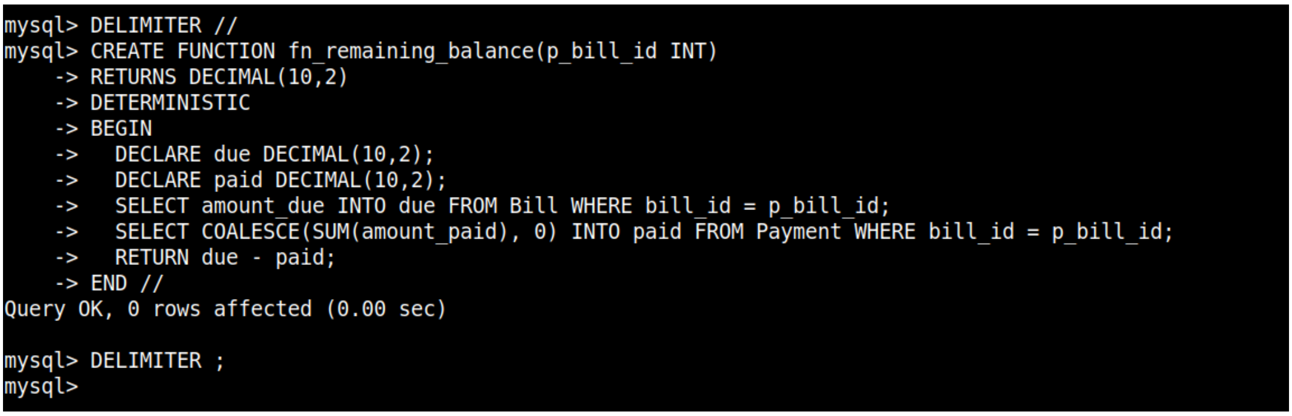
```
mysql> CALL sp_generate_bill(1, 1, '2025-10-27', '2025-11-10');
Query OK, 1 row affected (0.00 sec)

mysql>
```



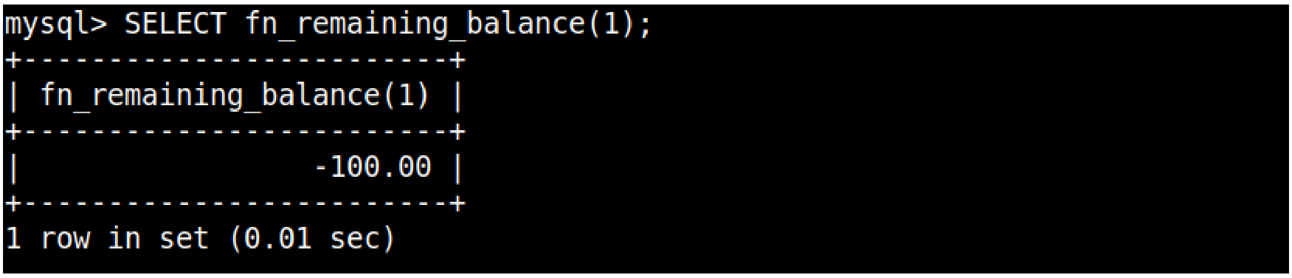
```
mysql> SELECT * FROM Bill ORDER BY bill_id DESC LIMIT 1;
+---------+----------+-----------+--------------+------------+------------+--------+
| bill_id | meter_id | tariff_id | billing_date | due_date   | amount_due | status |
+---------+----------+-----------+--------------+------------+------------+--------+
|      53 |        1 |         1 | 2025-10-27   | 2025-11-10 |       5500 | UNPAID |
+---------+----------+-----------+--------------+------------+------------+--------+
1 row in set (0.01 sec)
```

FUNCTION

Function to check the balance

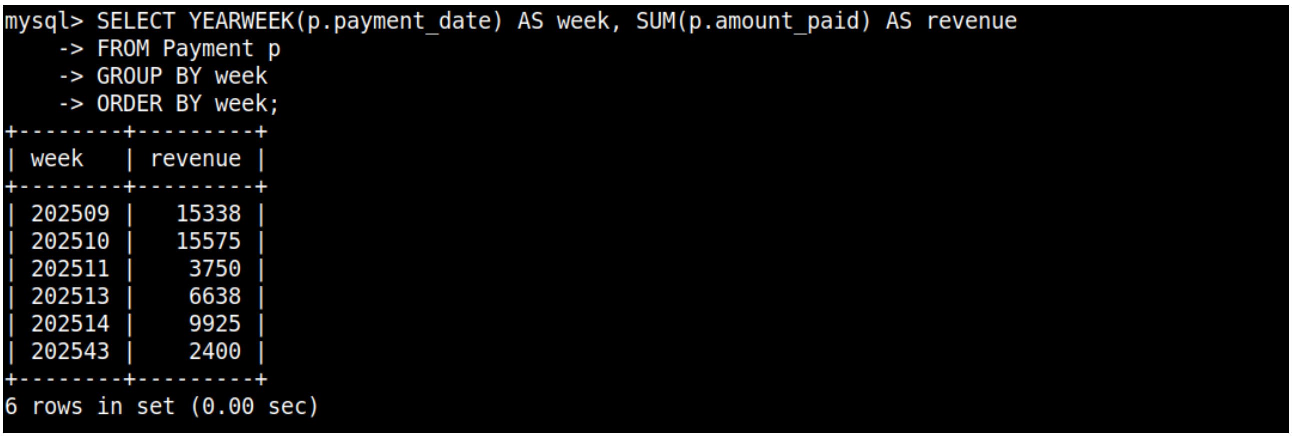


```
mysql> DELIMITER //
mysql> CREATE FUNCTION fn_remaining_balance(p_bill_id INT)
    -> RETURNS DECIMAL(10,2)
    -> DETERMINISTIC
    -> BEGIN
    ->   DECLARE due DECIMAL(10,2);
    ->   DECLARE paid DECIMAL(10,2);
    ->   SELECT amount_due INTO due FROM Bill WHERE bill_id = p_bill_id;
    ->   SELECT COALESCE(SUM(amount_paid), 0) INTO paid FROM Payment WHERE bill_id = p_bill_id;
    ->   RETURN due - paid;
    -> END //
Query OK, 0 rows affected (0.00 sec)

mysql> DELIMITER ;
mysql>
```



```
mysql> SELECT fn_remaining_balance(1);
+-------------------------+
| fn_remaining_balance(1) |
+-------------------------+
|                 -100.00 |
+-------------------------+
1 row in set (0.01 sec)
```

COMPLEX QUERIES



```
mysql> SELECT YEARWEEK(p.payment_date) AS week, SUM(p.amount_paid) AS revenue
    -> FROM Payment p
    -> GROUP BY week
    -> ORDER BY week;
+--------+---------+
| week   | revenue |
+--------+---------+
| 202509 |   15338 |
| 202510 |   15575 |
| 202511 |    3750 |
| 202513 |    6638 |
| 202514 |    9925 |
| 202543 |    2400 |
+--------+---------+
6 rows in set (0.00 sec)
```