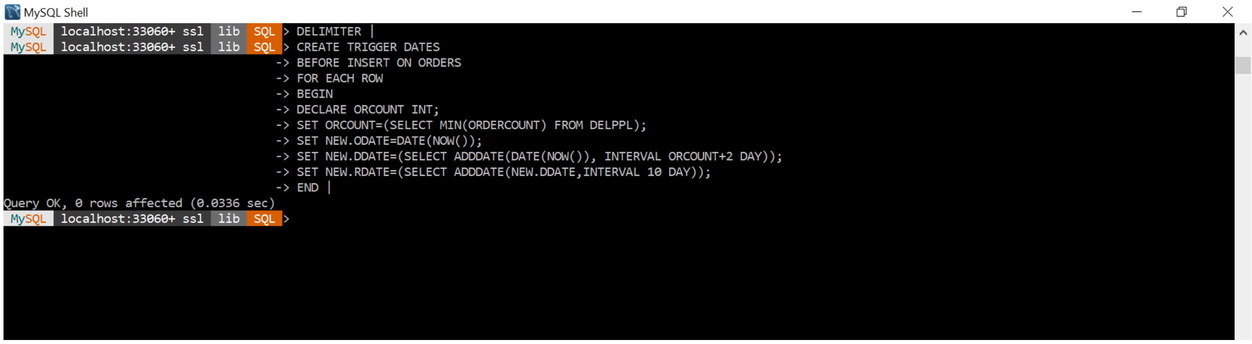
SETTING UP TRIGGERS, EVENTS AND PROCEDURES

TRIGGER :

This is a trigger for initialising the values for order date, delivery date and return date when a new order is inserted into the orders table

```
DELIMITER |
CREATE TRIGGER DATES
BEFORE INSERT ON ORDERS
FOR EACH ROW
BEGIN
DECLARE ORCOUNT INT;
SET ORCOUNT=(SELECT MIN(ORDERCOUNT) FROM DELPPL);
SET NEW.ODATE=DATE(NOW());
SET NEW.DDATE=(SELECT ADDDATE(DATE(NOW()), INTERVAL  ORCOUNT+2 DAY));
SET NEW.RDATE=(SELECT ADDDATE(NEW.DDATE,INTERVAL 10  DAY));
END |
```



```
 MySQL  localhost:33060+ ssl  lib  SQL > DELIMITER |
 MySQL  localhost:33060+ ssl  lib  SQL > CREATE TRIGGER DATES
                                      -> BEFORE INSERT ON ORDERS
                                      -> FOR EACH ROW
                                      -> BEGIN
                                      -> DECLARE ORCOUNT INT;
                                      -> SET ORCOUNT=(SELECT MIN(ORDERCOUNT) FROM DELPPL);
                                      -> SET NEW.ODATE=DATE(NOW());
                                      -> SET NEW.DDATE=(SELECT ADDDATE(DATE(NOW()), INTERVAL ORCOUNT+2 DAY));
                                      -> SET NEW.RDATE=(SELECT ADDDATE(NEW.DDATE,INTERVAL 10 DAY));
                                      -> END |
Query OK, 0 rows affected (0.0336 sec)
 MySQL  localhost:33060+ ssl  lib  SQL >
```

EVENTS :

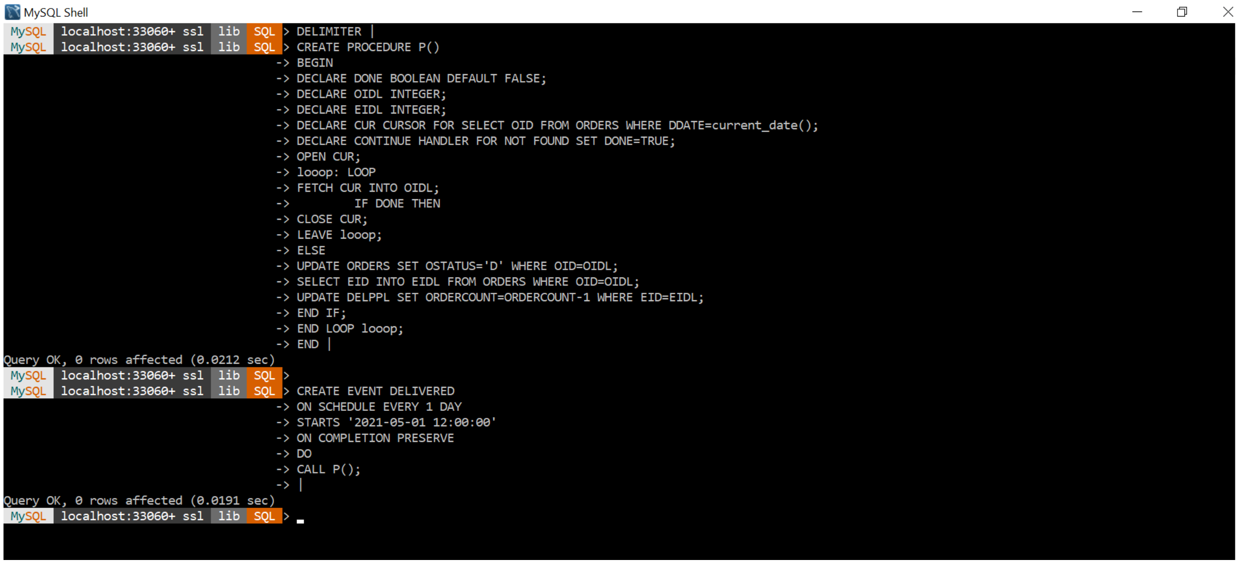
DELEVENT :

- This is an event (Trigger based on Date and Time) for updating the order status to ‘D’ for the delivered orders and decrementing the order count for the respective delivery people by 1.
- This event below named DELIVERED calls a procedure (P) every time it is fired .
- This is a recurring event which is fired every single day at 12 noon starting from 1st May 2021

```
DELIMITER |
CREATE PROCEDURE P()
BEGIN
```

```
DECLARE DONE BOOLEAN DEFAULT FALSE;
DECLARE OIDL INTEGER;
DECLARE EIDL INTEGER;
DECLARE CUR CURSOR FOR SELECT OID FROM ORDERS WHERE DDATE=current_date();
DECLARE CONTINUE HANDLER FOR NOT FOUND SET DONE=TRUE;
OPEN CUR;
looop: LOOP
                FETCH CUR INTO OIDL;
     IF DONE THEN
                        CLOSE CUR;
                        LEAVE looop;
                ELSE
                        UPDATE ORDERS SET OSTATUS='D' WHERE OID=OIDL;
                        SELECT EID INTO EIDL FROM ORDERS WHERE OID=OIDL;
                        UPDATE DELPPL SET ORDERCOUNT=ORDERCOUNT-1 WHERE EID=EIDL;
                END IF;
END LOOP looop;
END |

CREATE EVENT DELIVERED
ON SCHEDULE EVERY 1 DAY
STARTS '2021-05-01 12:00:00'
ON COMPLETION PRESERVE
DO
CALL P();
|
```

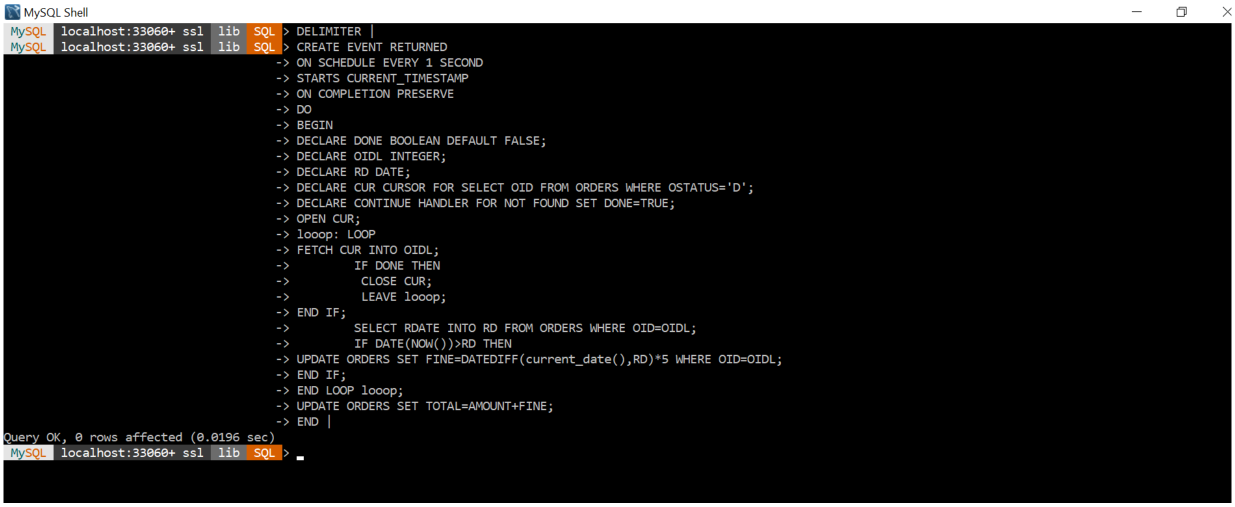


```
MySQL  localhost:33060+ ssl  lib  SQL > DELIMITER |
MySQL  localhost:33060+ ssl  lib  SQL > CREATE PROCEDURE P()
                                     -> BEGIN
                                     -> DECLARE DONE BOOLEAN DEFAULT FALSE;
                                     -> DECLARE OIDL INTEGER;
                                     -> DECLARE EIDL INTEGER;
                                     -> DECLARE CUR CURSOR FOR SELECT OID FROM ORDERS WHERE DDATE=current_date();
                                     -> DECLARE CONTINUE HANDLER FOR NOT FOUND SET DONE=TRUE;
                                     -> OPEN CUR;
                                     -> looop: LOOP
                                     -> FETCH CUR INTO OIDL;
                                     ->         IF DONE THEN
                                     -> CLOSE CUR;
                                     -> LEAVE looop;
                                     -> ELSE
                                     -> UPDATE ORDERS SET OSTATUS='D' WHERE OID=OIDL;
                                     -> SELECT EID INTO EIDL FROM ORDERS WHERE OID=OIDL;
                                     -> UPDATE DELPPL SET ORDERCOUNT=ORDERCOUNT-1 WHERE EID=EIDL;
                                     -> END IF;
                                     -> END LOOP looop;
                                     -> END |
Query OK, 0 rows affected (0.0212 sec)
MySQL  localhost:33060+ ssl  lib  SQL >
MySQL  localhost:33060+ ssl  lib  SQL > CREATE EVENT DELIVERED
                                     -> ON SCHEDULE EVERY 1 DAY
                                     -> STARTS '2021-05-01 12:00:00'
                                     -> ON COMPLETION PRESERVE
                                     -> DO
                                     -> CALL P();
                                     -> |
Query OK, 0 rows affected (0.0191 sec)
MySQL  localhost:33060+ ssl  lib  SQL > _
```

RETEVENT :

- This is an event (Trigger based on Date and Time) for calculating the fine for the delivered orders which has surpassed the return date. The total amount is calculated accordingly.
- This event named RETURNED is a recurring event which is fired every second from the second it is enabled.

```
DELIMITER |
CREATE EVENT RETURNED
ON SCHEDULE EVERY 1 SECOND
STARTS CURRENT_TIMESTAMP
ON COMPLETION PRESERVE
DO
BEGIN
DECLARE DONE BOOLEAN DEFAULT FALSE;
DECLARE OIDL INTEGER;
DECLARE RD DATE;
DECLARE CUR CURSOR FOR SELECT OID FROM ORDERS WHERE OSTATUS='D';
DECLARE CONTINUE HANDLER FOR NOT FOUND SET DONE=TRUE;
OPEN CUR;
looop: LOOP
          FETCH CUR INTO OIDL;
   IF DONE THEN
    CLOSE CUR;
    LEAVE looop;
          END IF;
   SELECT RDATE INTO RD FROM ORDERS WHERE OID=OIDL;
   IF DATE(NOW())>RD THEN
               UPDATE ORDERS SET FINE=DATEDIFF(current_date(),RD)*5 WHERE OID=OIDL;
          END IF;
END LOOP looop;
UPDATE ORDERS SET TOTAL=AMOUNT+FINE;
END |
```



```
 MySQL  localhost:33060+ ssl  lib  SQL > DELIMITER |
 MySQL  localhost:33060+ ssl  lib  SQL > CREATE EVENT RETURNED
                                      -> ON SCHEDULE EVERY 1 SECOND
                                      -> STARTS CURRENT_TIMESTAMP
                                      -> ON COMPLETION PRESERVE
                                      -> DO
                                      -> BEGIN
                                      -> DECLARE DONE BOOLEAN DEFAULT FALSE;
                                      -> DECLARE OIDL INTEGER;
                                      -> DECLARE RD DATE;
                                      -> DECLARE CUR CURSOR FOR SELECT OID FROM ORDERS WHERE OSTATUS='D';
                                      -> DECLARE CONTINUE HANDLER FOR NOT FOUND SET DONE=TRUE;
                                      -> OPEN CUR;
                                      -> looop: LOOP
                                      -> FETCH CUR INTO OIDL;
                                      ->         IF DONE THEN
                                      ->          CLOSE CUR;
                                      ->          LEAVE looop;
                                      -> END IF;
                                      ->         SELECT RDATE INTO RD FROM ORDERS WHERE OID=OIDL;
                                      ->         IF DATE(NOW())>RD THEN
                                      -> UPDATE ORDERS SET FINE=DATEDIFF(current_date(),RD)*5 WHERE OID=OIDL;
                                      -> END IF;
                                      -> END LOOP looop;
                                      -> UPDATE ORDERS SET TOTAL=AMOUNT+FINE;
                                      -> END |
Query OK, 0 rows affected (0.0196 sec)
 MySQL  localhost:33060+ ssl  lib  SQL > 
```