

TASK 8: STORED PROCEDURES AND FUNCTIONS

DONE BY ALEFIYA JOHARBHAI MANAGORI

1. USE CREATE PROCEDURE AND CREATE FUNCTION

```
mysql> DELIMITER //
mysql> CREATE FUNCTION CalculateDiscount(price DECIMAL(10,2), discount_percent INT)
-> RETURNS DECIMAL(10,2)
-> DETERMINISTIC
-> BEGIN
->   RETURN price - (price * discount_percent / 100);
-> END //
Query OK, 0 rows affected (0.01 sec)
```

```
mysql> DELIMITER ;
mysql> SELECT CalculateDiscount(1000.00, 10) AS discounted_price;
+-----+
| discounted_price |
+-----+
|      900.00 |
+-----+
1 row in set (0.01 sec)
```

```
mysql> DROP PROCEDURE IF EXISTS GetOrdersByUser;
Query OK, 0 rows affected, 1 warning (0.00 sec)
```

```
mysql> DELIMITER //
mysql> CREATE PROCEDURE GetOrdersByUser(IN uid INT)
-> BEGIN
->   SELECT o.order_no, o.order_date, o.order_amount
->   FROM Orders o
->   WHERE o.user_id = uid;
-> END //
Query OK, 0 rows affected (0.01 sec)
```

```
mysql> DELIMITER ;
mysql> CALL GetOrdersByUser(1);
+-----+-----+-----+
| order_no | order_date | order_amount |
+-----+-----+-----+
|    101 | 2025-06-22 |    20498.00 |
+-----+-----+-----+
1 row in set (0.01 sec)
```

```
Query OK, 0 rows affected (0.01 sec)
```

2. USE PARAMETER AND CONDITIONAL LOGIC

```
mysql> DELIMITER //
mysql> CREATE PROCEDURE CheckOrderValue(IN oid INT)
-> BEGIN
->   DECLARE order_val DECIMAL(10,2);
->
->   SELECT order_amount INTO order_val FROM Orders WHERE order_no = oid;
->
->   IF order_val > 10000 THEN
->     SELECT 'High Value Order' AS status;
->   ELSE
->     SELECT 'Regular Order' AS status;
->   END IF;
-> END //
```

Query OK, 0 rows affected (0.01 sec)

```
mysql> DELIMITER ;
mysql> CALL CheckOrderValue(101);
```

```
+-----+
| status |
+-----+
| High Value Order |
+-----+
1 row in set (0.00 sec)
```

Query OK, 0 rows affected (0.01 sec)

```
mysql> DELIMITER //
mysql> CREATE FUNCTION PaymentStatus(amount DECIMAL(10,2))
-> RETURNS VARCHAR(20)
-> DETERMINISTIC
-> BEGIN
->   DECLARE status VARCHAR(20);
->
->   IF amount >= 10000 THEN
->     SET status = 'Premium';
->   ELSEIF amount >= 5000 THEN
->     SET status = 'Standard';
->   ELSE
->     SET status = 'Basic';
->   END IF;
->
->   RETURN status;
-> END //
```

Query OK, 0 rows affected (0.10 sec)

```
mysql> DELIMITER ;
mysql> SELECT PaymentStatus(12000);
```

```
+-----+
| PaymentStatus(12000) |
+-----+
| Premium              |
```

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

```
mysql> DELIMITER //
mysql> CREATE PROCEDURE GetTotalSpent(IN uid INT)
-> BEGIN
->   SELECT u.u_name, SUM(o.order_amount) AS total_spent
->   FROM User u
->   JOIN Orders o ON u.user_id = o.user_id
->   WHERE u.user_id = uid
->   GROUP BY u.u_name;
-> END //
```

Query OK, 0 rows affected (0.01 sec)

```
mysql> DELIMITER ;
mysql> CALL GetTotalSpent(1);
```

```
+-----+-----+
| u_name | total_spent |
+-----+-----+
| Aaliya | 20498.00 |
+-----+-----+
```

1 row in set (0.00 sec)

Query OK, 0 rows affected (0.01 sec)