

Stored Procedures and Triggers - I

Session Objectives

- ✓ Understand the concept of stored procedures.
- ✓ Differentiate between IN, OUT, and INOUT parameters.
- ✓ Create and use procedures with real-world examples.

DRY

[Don't Repeat Yourself]

instead of writing again and again..we store it in function and call it whenever needed

SYNTAX:

```
DELIMITER $$

CREATE PROCEDURE procedure_name (
    [IN | OUT | INOUT] param_name DATATYPE,
    ...
)
BEGIN
    -- Declaration section (optional)
    -- Executable SQL statements
END $$

DELIMITER ;
```

Functions - User Defined / Pre-Defined Methods() -> invoked from library

python

packages

```
.strip()
.split()
.swapcase()
.isnumeric()
```

```
CREATE TABLE student_info (
    stud_id INT PRIMARY KEY,
    stud_code INT,
    stud_name VARCHAR(50),
    subject VARCHAR(50),
    marks INT,
    phone VARCHAR(15)
);
```



```
INSERT INTO student_info VALUES
(1, 101, 'Mark', 'English', 68, '3454569357'),
(2, 102, 'Joseph', 'Physics', 70, '9876543659'),
(3, 103, 'John', 'Maths', 70, '9765326975'),
(4, 104, 'Barack', 'Maths', 92, '87069873256'),
(5, 105, 'Rinky', 'Maths', 85, '6753159757'),
(6, 106, 'Adam', 'Science', 82, '79642256864'),
(7, 107, 'Andrew', 'Science', 83, '5674243579'),
(8, 108, 'Brayan', 'Science', 83, '7524316576'),
(9, 109, 'Alexandar', 'Biology', 67, '2347346438');
```

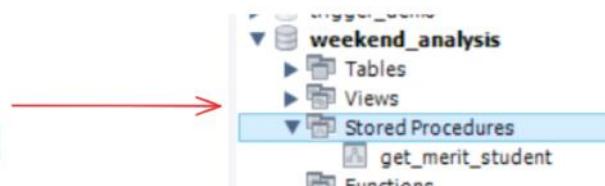
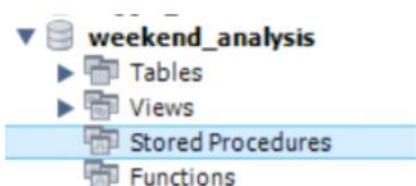
```
846 • SELECT * FROM student_info;
```

```
847
```

	stud_id	stud_code	stud_name	subject	marks	phone
▶	1	101	Mark	English	68	3454569357
	2	102	Joseph	Physics	70	9876543659
	3	103	John	Maths	70	9765326975
	4	104	Barack	Maths	92	87069873256
	5	105	Rinky	Maths	85	6753159757
	6	106	Adam	Science	82	79642256864
	7	107	Andrew	Science	83	5674243579
	8	108	Brayan	Science	83	7524316576
*	9	109	Alexandar	Biology	67	2347346438
*		HULL	HULL	HULL	HULL	HULL

```
848 • DESC student_info;
```

	Field	Type	Null	Key	Default	Extra
▶	stud_id	int	NO	PRI	HULL	
	stud_code	int	YES		HULL	
	stud_name	varchar(50)	YES		HULL	
	subject	varchar(50)	YES		HULL	
	marks	int	YES		HULL	
	phone	varchar(15)	YES		HULL	



```
mysql> CALL get_merit_student();
+-----+-----+-----+-----+-----+
| stud_id | stud_code | stud_name | subject | marks | phone |
+-----+-----+-----+-----+-----+
|     4 |      104 | Barack    | Maths   |  92 | 87069873256 |
|     5 |      105 | Rinky    | Maths   |  85 | 6753159757  |
|     6 |      106 | Adam     | Science |  82 | 79642256864 |
|     7 |      107 | Andrew   | Science |  83 | 5674243579  |
|     8 |      108 | Brayan  | Science |  83 | 7524316576  |
+-----+-----+-----+-----+-----+
5 rows in set (0.01 sec)

+-----+
| total_student |
+-----+
|      9       |
+-----+
1 row in set (0.04 sec)

Query OK, 0 rows affected (0.04 sec)
```

```

DELIMITER $$

CREATE PROCEDURE get_merit_student()
BEGIN
    SELECT * FROM student_info WHERE marks > 70;
    SELECT COUNT(*) AS total_student FROM student_info;
END $$

DELIMITER ;

```

.863 • SHOW PROCEDURE STATUS;

Db	Name	Type	Definer	Modified	Created	Security_type	Comment	character_set_client	collation_connection	Database Collation
sys	ps_setup_show_disabled_...	PROCEDURE	mysql.sys@lo...	2025-04-28 09:41:17	2025-04-28 09:41:17	INVOKER	Descript...	utf8mb4	utf8mb4_0900_ai_ci	utf8mb4_0900_ai_ci
sys	ps_setup_show_enabled	PROCEDURE	mysql.sys@lo...	2025-04-28 09:41:17	2025-04-28 09:41:17	INVOKER	Descript...	utf8mb4	utf8mb4_0900_ai_ci	utf8mb4_0900_ai_ci
sys	ps_setup_show_enabled_c...	PROCEDURE	mysql.sys@lo...	2025-04-28 09:41:17	2025-04-28 09:41:17	INVOKER	Descript...	utf8mb4	utf8mb4_0900_ai_ci	utf8mb4_0900_ai_ci
sys	ps_setup_show_enabled_j...	PROCEDURE	mysql.sys@lo...	2025-04-28 09:41:17	2025-04-28 09:41:17	INVOKER	Descript...	utf8mb4	utf8mb4_0900_ai_ci	utf8mb4_0900_ai_ci
sys	ps_statement_avg_latency...	PROCEDURE	mysql.sys@lo...	2025-04-28 09:41:17	2025-04-28 09:41:17	INVOKER	Descript...	utf8mb4	utf8mb4_0900_ai_ci	utf8mb4_0900_ai_ci
sys	ps_trace_statement_digest	PROCEDURE	mysql.sys@lo...	2025-04-28 09:41:17	2025-04-28 09:41:17	INVOKER	Descript...	utf8mb4	utf8mb4_0900_ai_ci	utf8mb4_0900_ai_ci
sys	ps_trace_thread	PROCEDURE	mysql.sys@lo...	2025-04-28 09:41:17	2025-04-28 09:41:17	INVOKER	Descript...	utf8mb4	utf8mb4_0900_ai_ci	utf8mb4_0900_ai_ci
sys	ps_truncate_all_tables	PROCEDURE	mysql.sys@lo...	2025-04-28 09:41:17	2025-04-28 09:41:17	INVOKER	Descript...	utf8mb4	utf8mb4_0900_ai_ci	utf8mb4_0900_ai_ci
sys	statement_performance_a...	PROCEDURE	mysql.sys@lo...	2025-04-28 09:41:17	2025-04-28 09:41:17	INVOKER	Descript...	utf8mb4	utf8mb4_0900_ai_ci	utf8mb4_0900_ai_ci
sys	table_exists	PROCEDURE	mysql.sys@lo...	2025-04-28 09:41:17	2025-04-28 09:41:17	INVOKER	Descript...	utf8mb4	utf8mb4_0900_ai_ci	utf8mb4_0900_ai_ci
▶ weekend_analysis	get_merit_student	PROCEDURE	root@localhost	2025-11-08 11:38:01	2025-11-08 11:38:01	DEFINER		utf8mb4	utf8mb4_0900_ai_ci	utf8mb4_0900_ai_ci

.865 • SHOW PROCEDURE STATUS WHERE Db = 'weekend_analysis';

Db	Name	Type	Definer	Modified	Created	Security_type	Comment	character_set_client	collation_connection	Database Collation
▶ weekend_analysis	get_merit_student	PROCEDURE	root@localhost	2025-11-08 11:38:01	2025-11-08 11:38:01	DEFINER		utf8mb4	utf8mb4_0900_ai_ci	utf8mb4_0900_ai_ci

```
-- DROP PROCEDURE get_merit_student;
```

```
DELIMITER $$
```

```

CREATE PROCEDURE get_merit_student()
BEGIN
    SELECT * FROM student_info WHERE marks > 70;
    -- SELECT COUNT(*) AS total_student FROM student_info;
END $$
```

```
DELIMITER ;
```

```
CALL get_merit_student();
```

```
-- What if new students enrolled , does it show the marks of newly inserted records.
```

```

INSERT INTO student_info() VALUES
(10, 110, 'Shyam', 'English', 91, '3454569357'),
(11, 111, 'Paramjeet', 'Physics', 77, '9876543659'),
(12, 112, 'Deepak', 'Maths', 59, '9765326975'),
(13, 113, 'Vaibhav', 'Maths', 97, '87069873256'),
(14, 114, 'Harisha', 'Maths', 85, '6753159757');
```

```
CALL get_merit_student();
```

stud_id	stud_code	stud_name	subject	marks	phone
4	104	Barack	Maths	92	87069873256
5	105	Rinky	Maths	85	6753159757
6	106	Adam	Science	82	79642256864
7	107	Andrew	Science	83	5674243579
8	108	Brayan	Science	83	7524316576
10	110	Shyam	English	91	3454569357
11	111	Paramjeet	Physics	77	9876543659
13	113	Vaibhav	Maths	97	87069873256
14	114	Harisha	Maths	85	6753159757

IN -> Accepting the input

```
-- Limit the number of records by taking an input var -> LIMIT Value;
```

```
DELIMITER $$
```

```
CREATE PROCEDURE get_student(IN var1 INT)
BEGIN
    SELECT * FROM student_info LIMIT var1;
END $$
```

```
DELIMITER ;
```

```
CALL get_student();
```

```
-- Error Code: 1318. Incorrect number of arguments for PROCEDURE
weekend_analysis.get_student; expected 1, got 0
```

```
CALL get_student(5);
```

stud_id	stud_code	stud_name	subject	marks	phone
1	101	Mark	English	68	3454569357
2	102	Joseph	Physics	70	9876543659
3	103	John	Maths	70	9765326975
4	104	Barack	Maths	92	87069873256
5	105	Rinky	Maths	85	6753159757

906 • **CALL get_student(7);**

Result Grid Filter Rows: _____ Export: _____ Wrap Cell C					
	stud_id	stud_code	stud_name	subject	marks
▶	1	101	Mark	English	68
	2	102	Joseph	Physics	70
	3	103	John	Maths	70
	4	104	Barack	Maths	92
	5	105	Rinky	Maths	85
	6	106	Adam	Science	82
	7	107	Andrew	Science	83

```
-- Let's take the input variables subject_name to apply filter.
```

```
DELIMITER $$
```

```
CREATE PROCEDURE get_student_by_subject(IN sub_name VARCHAR(25))
BEGIN
    SELECT stud_name , marks , subject
    FROM student_info
    WHERE subject = sub_name;
END $$
```

stud_name	marks	subject
John	70	Maths
Barack	92	Maths
Rinky	85	Maths
Deepak	59	Maths
Vaibhav	97	Maths
Harisha	85	Maths

```
CALL get_student_by_subject('Maths')$$ -- // &&
```

```
923     DELIMITER ;
```

```
924
```

```
925 • CALL get_student_by_subject('Physics');
```

```
926
```

Result Grid		
stud_name	marks	subject
► Joseph	70	Physics
Paramjeet	77	Physics

OUT -> Returning the output

```
-- Display the highest marks from students_info;
```

```
DELIMITER &&
```

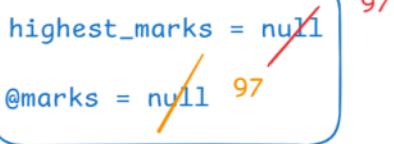
```
CREATE PROCEDURE display_highest_marks(OUT highest_marks INT)
BEGIN
    SELECT MAX(marks) INTO highest_marks FROM student_info;
END &&
```

```
DELIMITER ;
```

```
CALL display_highest_marks(@marks);
SELECT @marks;
```

@marks
► 97

null



- ▼ Stored Procedures
- display_highest_marks
 - get_merit_student
 - get_student
 - get_student_by_subject

942 • `SELECT * FROM Student_info;`

943

	stud_id	stud_code	stud_name	subject	marks	phone
▶	1	101	Mark	English	68	3454569357
	2	102	Joseph	Physics	70	9876543659
	3	103	John	Maths	70	9765326975
	4	104	Barack	Maths	92	87069873256
	5	105	Rinky	Maths	85	6753159757
	6	106	Adam	Science	82	79642256864
	7	107	Andrew	Science	83	5674243579
	8	108	Brayan	Science	83	7524316576
	9	109	Alexandar	Biology	67	2347346438
	10	110	Shyam	English	91	3454569357
	11	111	Paramjeet	Physics	77	9876543659
	12	112	Deepak	Maths	59	9765326975
	13	113	Vaibhav	Maths	97	87069873256
	14	114	Harisha	Maths	85	6753159757
*	HULL	HULL	HULL	HULL	HULL	HULL

max(marks)

97

IN & OUT PARAMETER

-- Calculate the Average Marks in a subject [subject IN , avg_marks OUT]

```
DELIMITER $$  
CREATE PROCEDURE get_avg_marks(IN sub_name VARCHAR(25) , OUT avg_marks DECIMAL(5,2))  
BEGIN  
    SELECT  
        AVG(marks) INTO avg_marks  
    FROM student_info  
    WHERE subject = sub_name;  
END $$  
DELIMITER ;
```

CALL get_avg_marks('Maths' , @avg_marks);
SELECT @avg_marks;

	@avg_marks
▶	81.33

▼ Stored Procedures

- ▢ display_highest_marks
- ▢ get_avg_marks
- ▢ get_merit_student
- ▢ get_student
- ▢ get_student_by_subject

971 • `CALL get_avg_marks('Science' , @avg_marks);`

972 • `SELECT @avg_marks;`

	Result Grid Filter Rows: _____ Export: _____ Wrap Cell Content: _____
▶	@avg_marks 82.67

```

964 • SELECT
965     Subject,
966     AVG(Marks) AS avg_marks
967 FROM student_info
968 GROUP BY 1;

```

Result Grid | Filter Rows: _____ | Export: _____ | Wrap

Subject	avg_marks
English	79.5000
Physics	73.5000
Maths	81.3333
Science	82.6667
Biology	67.0000

INOUT PARAMETER

```

DELIMITER $$ 
CREATE PROCEDURE display_marks(INOUT var1 INT)
BEGIN
    SELECT marks INTO var1
    FROM student_info WHERE stud_id = var1;
END $$ 
DELIMITER ;

```

```

SET @score = 3;
CALL display_marks(@score);
SELECT @score;

```

display_marks()
Buffer

Memory

@score = 3
var1 = null 3
70

@score

70

stud_id	stud_code	stud_name	subject	marks	phone
1	101	Mark	English	68	3454569357
2	102	Joseph	Physics	70	9876543659
3	103	John	Maths	70	9765326975
4	104	Barack	Maths	92	87069873256
5	105	Rinky	Maths	85	6753159757
6	106	Adam	Science	82	79642256864
7	107	Andrew	Science	83	5674243579
8	108	Brayan	Science	83	7524316576
9	109	Alexandar	Biology	67	2347346438
10	110	Shyam	English	91	3454569357
11	111	Paramjeet	Physics	77	9876543659
12	112	Deenak	Maths	59	9765326975

-- Find the maximum ProductPrice of the provided ProductSubcategoryKey as input

```
SELECT * FROM Products;
```

```
DELIMITER $$
```

```
CREATE PROCEDURE get_product_price_by_subcategory(INOUT var INT)
BEGIN
```

```
    SELECT MAX(ProductPrice) into var
        FROM Products WHERE ProductSubcategoryKey = var;
```

```
END $$
```

```
DELIMITER ;
```

```
SET @price = 2;
```

```
CALL get_product_price_by_subcategory(@price);
SELECT @price;
```

```
DROP PROCEDURE get_product_price_by_subcategory;
```

@price
3578

008 • SELECT

```
009      ProductSubcategoryKey,
010      MAX(ProductPrice) As MaxPrice
011  FROM Products
012  Group BY 1;
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
ProductSubcategoryKey	MaxPrice			
1	3399.99			
2	3578.27			
3	2384.07			
4	109.3364			
5	121.49			
6	106.5			
7	20.24			
8	404.99			
9	121.46			
10	229.49			
11	124.73			
12	1364.5			

Syntax:

```
CREATE TRIGGER trigger name
{BEFORE | AFTER} {INSERT | UPDATE | DELETE}
ON table_name
FOR EACH ROW
BEGIN
    -- SQL statements
END;
```

```
CREATE TABLE student_info (
    stud_id      INT PRIMARY KEY,
    stud_code    INT UNIQUE,
    stud_name    VARCHAR(50),
    subject      VARCHAR(30),
    marks        INT CHECK (marks BETWEEN 0 AND 100),
    phone        VARCHAR(15)
);
```

```
INSERT INTO student_info (stud_id, stud_code, stud_name, subject, marks, phone) VALUES
( 1,101,'Mark','English',68,'3454569357'),
( 2,102,'Joseph','Physics',70,'9876543659'),
( 3,103,'John','Maths',70,'9765326975'),
( 4,104,'Barack','Maths',92,'87069873256'),
( 5,105,'Rinky','Maths',85,'6753159757'),
( 6,106,'Adam','Science',82,'79642256864'),
( 7,107,'Andrew','Science',83,'5674243579'),
( 8,108,'Brayan','Science',83,'7524316576'),
( 9,109,'Alexandar','Biology',67,'2347346438'),
(10,110,'Clara','Biology',74,'2342342345'),
(11,111,'Derek','English',59,'4564564560'),
(12,112,'Ella','English',91,'1231231234'),
(13,113,'Fiona','Physics',65,'9879879876'),
(14,114,'George','Physics',79,'3453453451'),
(15,115,'Helen','Maths',88,'2223334445'),
(16,116,'Ian','Science',95,'6667778880'),
(17,117,'Jane','Biology',57,'9998887776'),
(18,118,'Kyle','English',73,'1112223339'),
(19,119,'Liam','Maths',81,'8887776665'),
(20,120,'Mia','Physics',93,'4445556667');
```



064 • SELECT * FROM Student info;

065 • DESC Student_info;

Result Grid			Filter Rows:	Export:		Wrap Cell C
	Field	Type	Null	Key	Default	Extra
▶	stud_id	int	NO	PRI	HULL	
	stud_code	int	YES	UNI	HULL	
	stud_name	varchar(50)	YES		HULL	
	subject	varchar(30)	YES		HULL	
	marks	int	YES		HULL	
	phone	varchar(15)	YES		HULL	

BEFORE INSERT

INSERT - new ✓ old ✗
UPDATE - new ✓ old ✓
DELETE - new ✗ old ✓

-- BEFORE INSERT TRIGGER [Rejecting the marks >100 or < 0]

```
DELIMITER $$  
CREATE TRIGGER trg_before_insert_student  
BEFORE INSERT ON Student_info  
FOR EACH ROW  
BEGIN  
    IF NEW.marks > 100 OR New.marks < 0 THEN  
        SIGNAL SQLSTATE '45000'  
        SET MESSAGE_TEXT = 'Marks must be between 0 And 100';  
    END IF;  
END $$  
DELIMITER ;
```

Trigger	Event	Table	Statement	Timing	Created	sql_mode	Definer	character_set_client	collation_conn
trg_before_insert_student	INSERT	student_info	BEGIN IF NEW.marks > 100 OR New.marks < 0 ... BEFORE	2025-11-08 13:50:03,72	ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLE...	root@localhost	utf8mb4	utf8mb4_0900	

```
INSERT INTO student_info (stud_id, stud_code, stud_name, subject, marks, phone)
VALUES (21, 121, 'Soumya', 'Maths', 111, '9910854433');
```

-- Error Code: 1644. Marks must be between 0 And 100

```
INSERT INTO student_info (stud_id, stud_code, stud_name, subject, marks, phone)
VALUES (21, 121, 'Soumya', 'Maths', -11, '9910854433');
```

-- Error Code: 1644. Marks must be between 0 And 100

AFTER INSERT

Audit Log

[Timestamp]

```
-- AFTER INSERT [Audit Log]
```

```
CREATE TABLE audit_log(
    log_id INT NOT NULL AUTO_INCREMENT,
    action ENUM('INSERT', 'UPDATE', 'DELETE'),
    stud_id INT,
    log_time TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    PRIMARY KEY(log_id)
);
```

```
DELIMITER $$
```

```
CREATE TRIGGER trg_after_insert_student
AFTER INSERT ON Student_info
FOR EACH ROW
BEGIN
    INSERT INTO audit_log(action , stud_id)
    VALUES('INSERT' , NEW.stud_id);
END $$
```

```
DELIMITER ;
```

```
SHOW TRIGGERS;
```

113 • SHOW TRIGGERS;

Trigger	Event	Table	Statement	Timing	Created	sql_mode	Definer	character_set_client	collation_conn
trg_before_insert_student	INSERT	student_info	BEGIN IF NEW.marks > 100 OR New.marks < 0 ... BEFORE	2025-11-08 13:50:03.72	ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLE...	root@localhost	utf8mb4	utf8mb4_0900	
trg_after_insert_student	INSERT	student_info	BEGIN INSERT INTO audit_log(action , stud_id) ... AFTER	2025-11-08 14:02:07.00	ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLE...	root@localhost	utf8mb4	utf8mb4_0900	

```
115 • SELECT * FROM Audit_log;
```

```
116
```

```
--
```

Result Grid				
	log_id	action	stud_id	log_time
*	HULL	HULL	HULL	HULL

```
--
```

```
117 -- INSERTING a New Record on Student_info
```

```
118 • INSERT INTO student_info (stud_id, stud_code, stud_name, subject, marks, phone)  
119   VALUES (21, 121, 'Soumya', 'Maths', 99, '9910854433');
```

```
120
```

```
121 • SELECT * FROM Student_info;
```

```
--
```

Result Grid						
	stud_id	stud_code	stud_name	subject	marks	phone
16	116	Ian	Science	95	6667778880	
17	117	Jane	Biology	57	9998887776	
18	118	Kyle	English	73	1112223339	
19	119	Liam	Maths	81	8887776665	
20	120	Mia	Physics	93	4445556667	
▶	21	121	Soumya	Maths	99	9910854433
*	HULL	HULL	HULL	HULL	HULL	HULL

```
115 • SELECT * FROM Audit_log;
```

Result Grid				
	log_id	action	stud_id	log_time
▶	1	INSERT	21	2025-11-08 14:05:05
*	HULL	HULL	HULL	HULL