

DBMS - Stored procedure

- A **stored procedure** is a collection of pre-compiled SQL statements **stored** inside the database. It is a subroutine or a subprogram in the regular computing language.
- A **stored procedure** always contains a name, parameter lists, and SQL statements.
- To create a new stored procedure, you use the CREATE PROCEDURE statement.

Creating / Executing / Removing a stored procedure

From Single Table:

Drop procedure if exists sps; ** Removing a stored procedure

```
Delimiter //  
create procedure sps ()  
begin  
select * from Movie;  
end //
```

delimiter ;

call sps() **Executing a stored procedure

From Multiple Table:

```
delimiter $$  
create procedure spm ()  
begin  
select m.mID,m.title,r.rID,r.stars from Movie m , Rating r where m.mID=r.mID;  
end $$
```

delimiter ;

Listing Stored Procedures

show procedure status;

show procedure status where db='view';

show procedure status like '%p%';

Stored Procedure Variables

EXAMPLE 1 :

```
DELIMITER $$

CREATE PROCEDURE gettotalmovie()
BEGIN
    DECLARE totalmovie INT DEFAULT 0;
    SET totalmovie = 5;

    SELECT COUNT(*) INTO totalmovie FROM Movie;

    SELECT totalmovie;

END $$

DELIMITER ;
```

```
CALL gettotalmovie();
```

EXAMPLE 2 :

```
DELIMITER $$
CREATE PROCEDURE gettotalstars()
BEGIN
    DECLARE totalstars INT DEFAULT 0;

    SELECT sum(stars) INTO totalstars FROM Rating;

    SELECT totalstars;

END $$

DELIMITER ;

CALL gettotalstars();
```

Stored Procedure Parameters

IN parameters :

```
DELIMITER //

CREATE PROCEDURE Getmoviedetail(IN movieID INT)

BEGIN
    SELECT *
    FROM Movie
    WHERE mID = movieID;
END //

DELIMITER ;

CALL Getmoviedetail('101');
```

OUT parameters :

DELIMITER \$\$

```
CREATE PROCEDURE GetMoviename (IN movieID INT , OUT name varchar(255))
```

```
BEGIN
```

```
    SELECT title
```

```
    INTO name
```

```
    FROM Movie
```

```
    WHERE mID = movieID;
```

```
END$$
```

DELIMITER ;

```
call GetMoviename(101,@name);
```

```
select @name;
```

INOUT parameters :

DELIMITER \$\$

```
CREATE PROCEDURE SetCounter(INOUT counter INT,IN inc INT)
```

```
BEGIN
```

```
    SET counter = counter + inc;
```

```
END$$
```

DELIMITER ;

```
SET @counter = 1;
```

```
CALL SetCounter(@counter,1); -- 2
```

```
CALL SetCounter(@counter,1); -- 3
```

```
CALL SetCounter(@counter,5); -- 8
```

```
SELECT @counter;
```

MySQL IF-THEN-ELSEIF-ELSE statement

DELIMITER //

```
CREATE PROCEDURE GetMoviestatus(IN movieID INT, OUT movieLevel INT)
```

```
BEGIN
```

```
    DECLARE movieyear INT DEFAULT 0;
```

```
    SELECT year INTO movieyear FROM Movie WHERE mID = movieID;
```

```
    IF movieyear>2000 THEN
```

```
        SET movieLevel = 1;
```

```
    ELSEIF movieyear<=2000 AND movieyear>1950 THEN
```

```
        SET movieLevel = 2;
```

```
ELSE
    SET movieLevel= 3;

END IF;
END //
DELIMITER ;
```