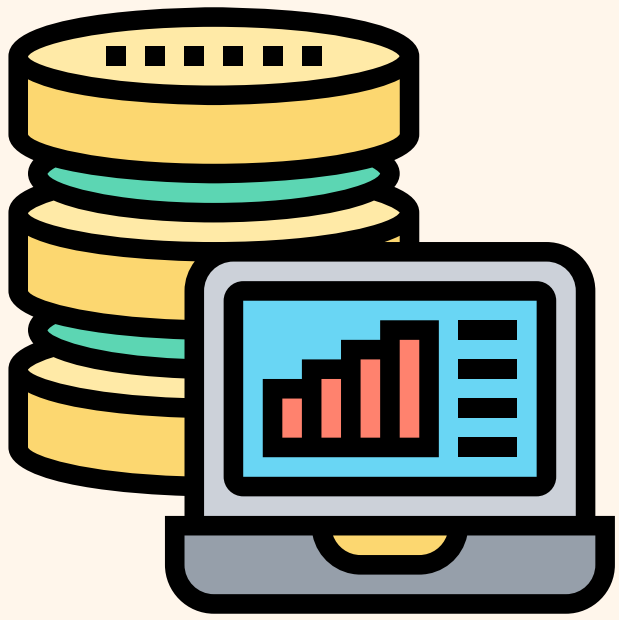
#56daysofSQL

Day 3
Stored Procedure
Parameters



Aditya Shenoy

#56daysofSQL

Parameters in Stored Procedures

- The real power of stored procedures is the ability to pass parameters and have the stored procedure handle the differing requests that are made.
- The parameters make the stored procedure more useful and reusable.

Aditya Shenoy

#56daysofSQL

A parameter in a stored procedure has one of three modes :

1. IN
2. OUT
3. INOUT

Aditya Shenoy

#56daysofSQL

General Syntax :

```
[IN | OUT | INOUT]
parameter_name
datatype[(length)];
```

Aditya Shenoy

#56daysofSQL

IN parameter

- **IN** is the default mode.
- When you define an **IN** parameter in a stored procedure, the calling program has to pass an argument to the stored procedure.

Syntax

```
IN parameter_name
datatype[(length)];
```

#56daysofSQL

IN parameter **USE CASE** :

```
DELIMITER //

CREATE PROCEDURE GetOfficeByCountry(
   IN countryName VARCHAR(255))
BEGIN
   SELECT *
   FROM offices
   WHERE country = countryName;
END //

DELIMITER ;

CALL GetOfficeByCountry('USA');
```

Output -

	officeCode	city	phone	addressLine1	addressLine2	state	country	postalCode	territory
▶	1	San Francisco	+1 650 219 4782	100 Market Street	Suite 300	CA	USA	94080	NA
	2	Boston	+1 215 837 0825	1550 Court Place	Suite 102	MA	USA	02107	NA
	3	NYC	+1 212 555 3000	523 East 53rd Street	apt. 5A	NY	USA	10022	NA

Aditya Shenoy

#56daysofSQL

OUT parameter

- The value of an **OUT** parameter can be changed inside the stored procedure and its new value is passed back to the calling program.

Syntax

```
OUT parameter_name
datatype[(length)];
```

#56daysofSQL

OUT parameter **USE CASE** :

```
DELIMITER //

CREATE PROCEDURE GetOrderCountByStatus(
   IN orderStatus VARCHAR(25),
   OUT total INT)
BEGIN
   SELECT COUNT(orderNumber)
   INTO total FROM orders
   WHERE status = orderStatus;
END //

DELIMITER ;

CALL GetOrderCountryByStaus('Shipped',@total);
SELECT @total;
```

Output -

	@total
▶	303

Aditya Shenoy

#56daysofSQL

INOUT parameter

- An **INOUT** parameter is a combination of IN and OUT parameters.
- It means that the calling program may pass the argument, and the stored procedure can modify the **INOUT** parameter, and pass the new value back to the calling program.

Syntax

```
INOUT parameter_name
datatype[(length)];
```

#56daysofSQL

INOUT parameter **USE CASE :**

```
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;
```

Output -

	@counter
▶	8

Aditya Shenoy