

A decorative graphic on the left side of the slide, consisting of a network of white lines and small circles on a teal background, resembling a circuit board or a neural network.

# WEEK 8

MORE SQL – STORED PROCEDURES

# STUDENT OBJECTIVES

- Upon completion of this video, you should be able to:
  - Explain the purpose of a stored procedure
  - List at least 2 advantages and 2 disadvantages of a stored procedure
  - Write a simple stored procedure in MySQL

# STORED PROCEDURES

- A precompiled application program, written in any language such as C, C++, COBOL, Java, ... that is executed in response to a SINGLE SQL CALL statement
- May contain SQL statements as well as other logic statements
- Little program that with SQL statements in and code, can pass in parameters
- Encapsulates Business Logic
- Takes complicated SQL logic and makes it easier to execute
- Useful mainly in CLIENT/SERVER applications

**SITUATION:** Suppose you need to update the salary of all 100,000 employees in your database, by 6.25%. You are working in Sydney, Australia with a java application on your desktop, the database is in New York City. What problems can you foresee?

*network traffic*

**QUESTION:** What do you do to solve this problem?

**ANSWER:** Use a **Stored Procedure!**

*Server-side operation.*

# ADVANTAGES OF STORED PROCEDURES

- Typically faster, because the code for the stored procedure is compiled
- Reduce traffic between application and database because it only has to send the name of procedure, not long and multiple SQL statements
- Reusable
- Secure – DBA can grant permissions to them without giving permissions to the underlying tables.

# DISADVANTAGES OF STORED PROCEDURES

- Over usage (using MANY stored procedures) can actually slow down the application (increased load on database server).
- Sometimes doesn't allow for complex business logic
- Hard to debug
- Migrating to another Database System can be tricky.

# EXAMPLES FROM MYSQL – LET'S TRY IT OUT

```
mysql> delimiter //
```

  

```
mysql> CREATE PROCEDURE simpleproc (OUT param1 INT)
-> BEGIN
->   SELECT COUNT(*) INTO param1 FROM t;
-> END//
```

Query OK, 0 rows affected (0.00 sec)

  

```
mysql> delimiter ;
```

  

```
mysql> CALL simpleproc(@a);
```

Query OK, 0 rows affected (0.00 sec)

  

```
mysql> SELECT @a;
```

-----
@a
-----
3
-----

1 row in set (0.00 sec)

```
01 DELIMITER //
```

  

```
02
```

  

```
03 CREATE PROCEDURE `proc_WHILE` (IN param1 INT)
04 BEGIN
05   DECLARE variable1, variable2 INT;
06   SET variable1 = 0;
07
08   WHILE variable1 < param1 DO
09     INSERT INTO table1 VALUES (param1);
10     SELECT COUNT(*) INTO variable2 FROM table1;
11     SET variable1 = variable1 + 1;
12   END WHILE;
13 END //
```

```
01 DELIMITER //
```

  

```
02
```

  

```
03 CREATE PROCEDURE `p2` ()
04 LANGUAGE SQL
05 DETERMINISTIC
06 SQL SECURITY DEFINER
07 COMMENT 'A procedure'
08 BEGIN
09   SELECT 'Hello World !';
10 END//
```