

## Tutorial 4: MySQL (contd...) and Oracle(PL/SQL)

### Event scheduler:

```
CREATE EVENT test_event_02
ON SCHEDULE AT CURRENT_TIMESTAMP + INTERVAL 1 MINUTE
ON COMPLETION PRESERVE
DO
  INSERT INTO messages(message,created_at)
  VALUES('Test MySQL Event 2',NOW());
```

```
CREATE EVENT test_event_03
ON SCHEDULE EVERY 1 MINUTE
STARTS CURRENT_TIMESTAMP
ENDS CURRENT_TIMESTAMP + INTERVAL 1 HOUR
DO
  INSERT INTO messages(message,created_at)
  VALUES('Test MySQL recurring Event',NOW());
```

<http://www.mysqltutorial.org/mysql-triggers/working-mysql-scheduled-event/>

<https://dev.mysql.com/doc/refman/8.0/en/events-privileges.html>

### Insert a 1000 rows:

```
CREATE TABLE `data`
(
  `id`      bigint(20) NOT NULL      AUTO_INCREMENT,
  `datetime` timestamp NULL        DEFAULT CURRENT_TIMESTAMP,
  `channel` int(11)                DEFAULT NULL,
  `value`   float                 DEFAULT NULL,

  PRIMARY KEY (`id`)
);
```

```
DELIMITER $$
CREATE PROCEDURE generate_data()
BEGIN
  DECLARE i INT DEFAULT 0;
  WHILE i < 1000 DO
```

```

INSERT INTO `data` (`datetime`,`value`,`channel`) VALUES (
    FROM_UNIXTIME(UNIX_TIMESTAMP('2014-01-01
01:00:00')+FLOOR(RAND()*31536000)),
    ROUND(RAND()*100,2),
    1
);
SET i = i + 1;
END WHILE;
END$$
DELIMITER ;

```

CALL generate\_data();

<https://stackoverflow.com/questions/25098747/how-to-generate-1000000-rows-with-random-data>

### **INOUT Parameters:**

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.

Example:

```

DELIMITER $$
CREATE PROCEDURE set_counter(INOUT count INT(4),IN inc INT(4))
BEGIN
    SET count = count + inc;
END$$
DELIMITER ;

```

```

SET @counter = 1;
CALL set_counter(@counter,1); -- 2
CALL set_counter(@counter,1); -- 3
CALL set_counter(@counter,5); -- 8
SELECT @counter; -- 8

```

### **Sharding (Horizontal scaling and more):**

Shard (System for Highly Available Replicated Data) goes beyond just having horizontal splitting, cause if for a simple query multiple instances of the logically/physically separately stored tables need to be accessed means it's not gonna give the efficiency that we hope to attain.

There are different ways of doing this like [hash partitioning](#) or normal range based splitting.

### **Raising Errors in MySQL:**

You use the SIGNAL statement to return an error or warning condition to the caller from a stored program e.g., stored procedure, trigger or event. The SIGNAL statement provides you with control over which information for returning such as value and message.

Following the SIGNAL keyword is a SQLSTATE value or a condition name declared by the DECLARE CONDITION statement. The SIGNAL statement must always specify a SQLSTATE value or a named condition that defined with an SQLSTATE value.

Generally, SQLSTATE 45000 is a generic state representing user generated errors.

To generate an exception check for a condition that is being violated and then use the signal statement as shown:

Create trigger

```
...
if val < 0 then
    Set msg = "Error: Value is less than 0."
    signal sqlstate '45000' set message_text = msg;
end if;
...
end //
```

### **Questions on Triggers and Stored Procedures:**

1. Create a trigger which will work before deletion in employee table and create a duplicate copy of the record in another table employee\_backup.
2. Write a trigger to ensure that no employee of age less than 21 can be inserted in the database.

### **Solutions:**

```
1.
delimiter //
CREATE TRIGGER Backup BEFORE DELETE ON employee
FOR EACH ROW
BEGIN
INSERT INTO employee_backup
VALUES (OLD.employee_no, OLD.name,
```

```
        OLD.job, OLD.hiredate, OLD.salary);  
END //  
delimiter;
```

```
2.  
delimiter //  
CREATE TRIGGER Check_age BEFORE INSERT ON employee  
FOR EACH ROW  
BEGIN  
IF NEW.age < 21 THEN  
SIGNAL SQLSTATE '45000'  
SET MESSAGE_TEXT = 'ERROR:  
        AGE MUST BE ATLEAST 21 YEARS!';  
END IF;  
END //  
delimiter;
```

## **PL/SQL(Procedural Language extensions to SQL)**

PL/SQL means instructing the compiler 'what to do' through SQL and 'how to do' through its procedural way.

The PL/SQL architecture mainly consists of following three components:

### **PL/SQL block**

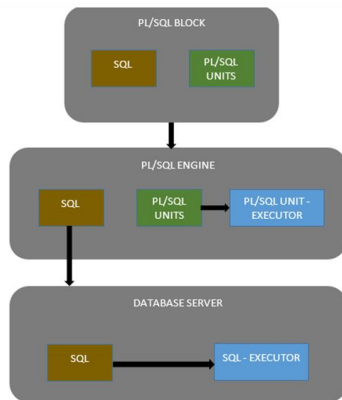
- This consists of different sections to divide the code logically (declarative section for declaring purpose, execution section for processing statements, exception handling section for handling errors)
- It also contains the SQL instruction that used to interact with the database server.

### **PL/SQL Engine**

- PL/SQL engine is the component where the actual processing of the codes takes place.
- PL/SQL engine separates PL/SQL units and SQL part in the input (as shown in the image below).
- The separated PL/SQL units will be handled by the PL/SQL engine itself.
- The SQL part will be sent to database server where the actual interaction with database takes place.
- It can be installed in both database server and in the application server.

### **Database Server**

- The PL/SQL engine uses the SQL from PL/SQL units to interact with the database server.
- It consists of SQL executor which parses the input SQL statements and execute the same.



## Advantage of Using PL/SQL

- Better performance, as SQL is executed in bulk rather than a single statement.
- High Productivity.
- Tight integration with SQL.
- Full Portability.
- Tight Security.
- Support Object Oriented Programming concepts.

## Basic structure of a block:

```

DECLARE --optional
  <declarations>

```

```

BEGIN --mandatory
  <executable statements. At least one executable statement is mandatory>

```

```

EXCEPTION --optional
  <exception handles>

```

```

END; --mandatory

```

<https://www.guru99.com/introduction-pl-sql.html>

MySQL is a fast, reliable, robust and open source database system that has a large number of features too offer.

**Advantages of Oracle over MySQL:**

- Oracle is a much larger database software, it can handle much more database requests and manage more transactions than MySQL.
- MySQL take resources from the system to processes the requests which can put a large load on the system. However oracle has a built in memory management feature that spreads the load across multiple threads making it much faster and less effective on the system running it.
- Aside from background threads, one connection will use only one CPU core. The applies to GROUP BY, UNION, PARTITION`, and anything else you might think it should manage to do in parallel. **(MySQL)**.

Reference:

<https://stackoverflow.com/questions/39700330/handling-very-large-data-with-mysql>

<https://stackoverflow.com/questions/42528943/mysql-server-uses-only-1-out-of-48-cpu-cores-with-group-by-queries>