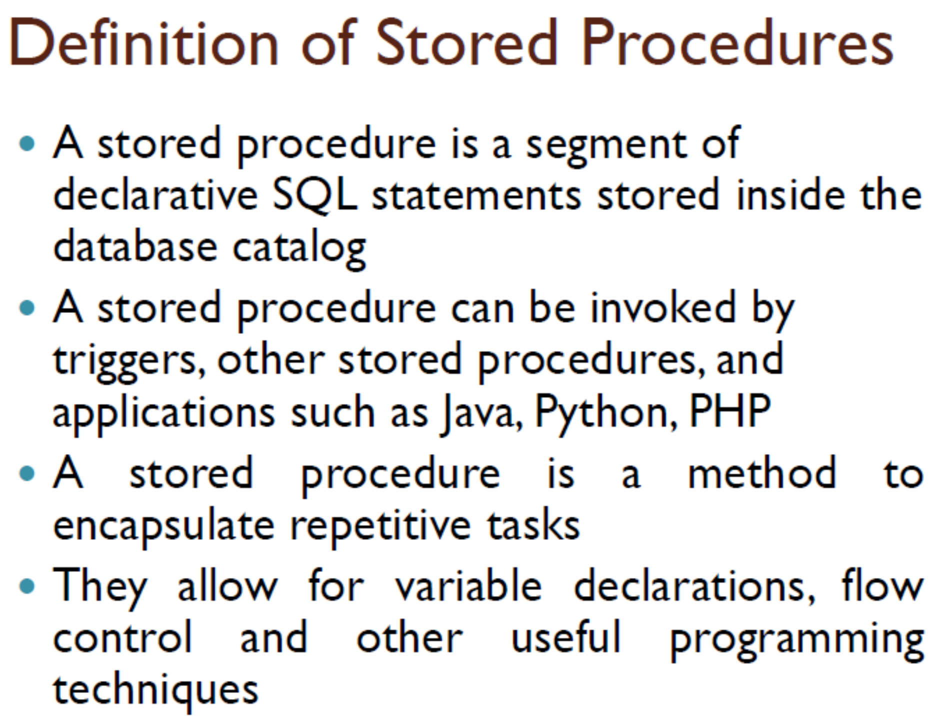
ICT1222 Database Management Systems Practicum

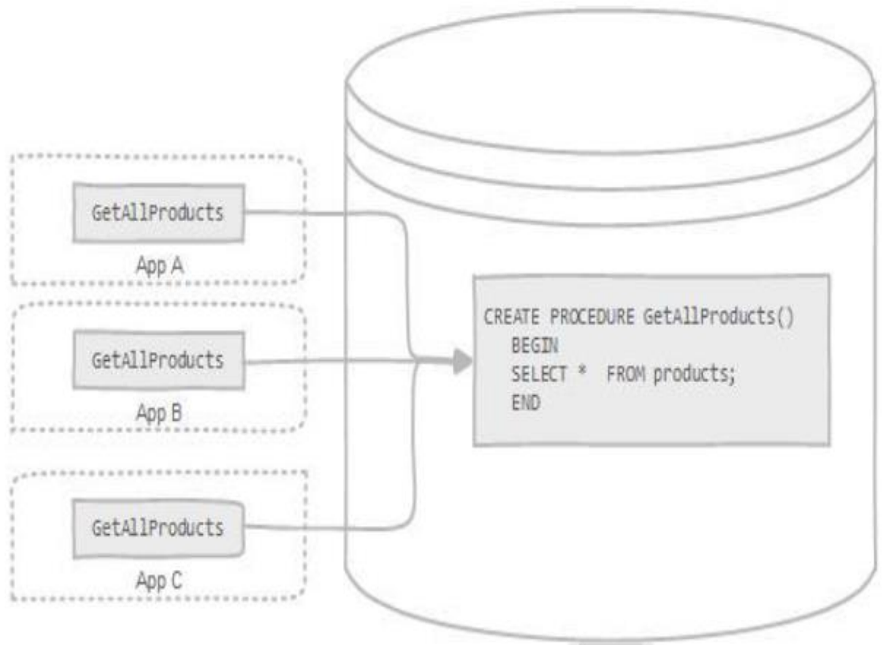
SQL Stored Procedure



Definition of Stored Procedures

- A stored procedure is a segment of declarative SQL statements stored inside the database catalog
- A stored procedure can be invoked by triggers, other stored procedures, and applications such as Java, Python, PHP
- A stored procedure is a method to encapsulate repetitive tasks
- They allow for variable declarations, flow control and other useful programming techniques

Definition of Stored Procedures



```
CREATE PROCEDURE GetAllProducts()
  BEGIN
  SELECT *  FROM products;
  END
```

Stored Procedures in MySQL

- MySQL is known as the most popular open source RDBMS which is widely used by both community and enterprise
- However, during the first decade of its existence, it did not support stored procedures, stored functions, triggers, and events
- Since MySQL version 5.0, those features were added to MySQL database engine to make it more flexible and powerful

MySQL stored procedures advantages

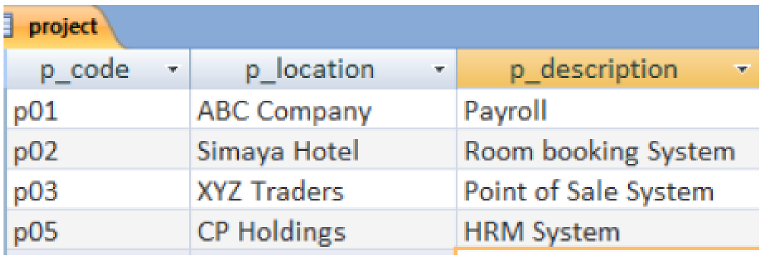
- Typically stored procedures help increase the performance of the applications

  (compiled on demand, cache etc)
- Stored procedures help reduce the traffic between application and database server
- Stored procedures are reusable and transparent to any applications
- Stored procedures are secure

MySQL stored procedures disadvantages

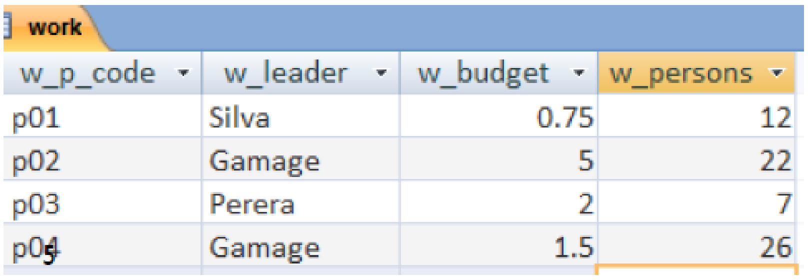
- If you use many stored procedures, the memory usage of every connection that is using those stored procedures will increase substantially
- If you overuse a large number of logical operations inside store procedures, the CPU usage will also increase
- Stored procedure's constructs are not designed for developing complex and flexible business logic
- It is difficult to debug stored procedures
- It is not easy to develop and maintain stored procedures

- Create database myprocedure
- Create following tables in the database



project

p_code	p_location	p_description
p01	ABC Company	Payroll
p02	Simaya Hotel	Room booking System
p03	XYZ Traders	Point of Sale System
p05	CP Holdings	HRM System



work

w_p_code	w_leader	w_budget	w_persons
p01	Silva	0.75	12
p02	Gamage	5	22
p03	Perera	2	7
p0~~4~~5	Gamage	1.5	26

Creating a Stored Procedure

```
CREATE PROCEDURE name_of_procedure()
BEGIN
   Business logic
END
```

Executing a stored procedure

```
call  name_of_procedure;
```

Delete stored procedures

```
drop procedure name_of_procedure;
```

Display the list of stored procedures

- show procedure status;
- select name from mysql.proc;

Display the content of a procedure

```
show create procedure
name_of_procedure;
```

Delimiter is used to change the standard delimiter (semicolon) to another. Here, the delimiter is changed from semicolon(;) to //
So u can have multiple SQL statements inside the stored procedure which can be separated by the semicolon.

```
DELIMITER //

CREATE PROCEDURE p1()
BEGIN
   SELECT 'This is my 1st stored procedure';
END//

DELIMITER ;
```

Changes the delimiter back to the standa
semicolon.

```
DELIMITER //

CREATE PROCEDURE display_projects()
BEGIN
   SELECT * from project;
END//

DELIMITER ;
DELIMITER //

CREATE PROCEDURE dis_managers()
BEGIN

   select * from project;

   select p_description , w_leader from project , work
                                          where p_code = w_p_code;
END//

DELIMITER ;
```

Declaring variables

DECLARE variable_name datatype(size)

Eg:

```
  declare a,b int;
  declare a int default 10;
```

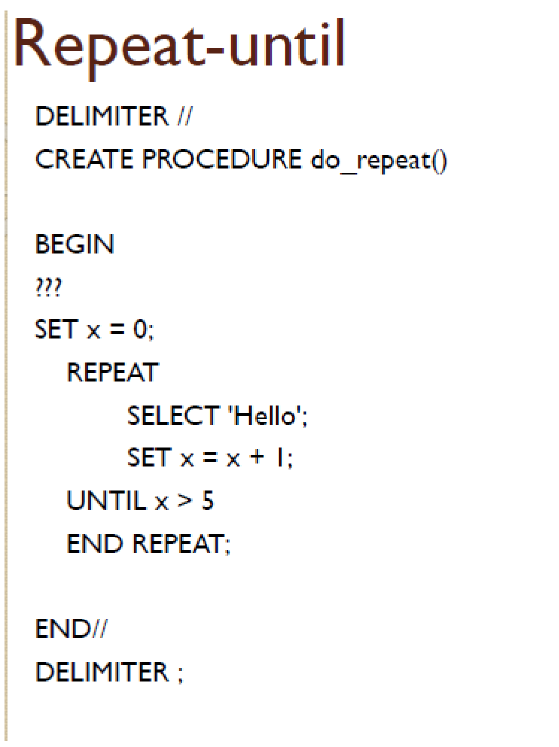
Passing Parameters

```
DELIMITER //

CREATE PROCEDURE dis_persons(IN no int)
BEGIN
   SELECT * from work where w_persons >
  no;
END//

DELIMITER ;
```

Find the project details for a given project leader

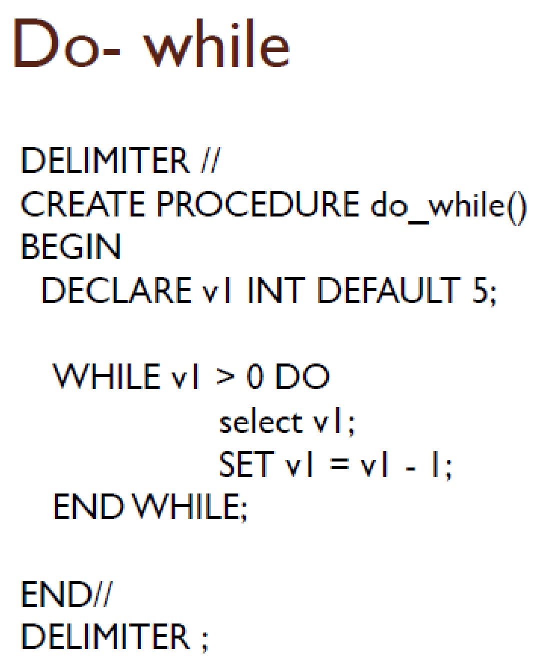
```
DELIMITER //

CREATE PROCEDURE p_details(IN x varchar(20))
BEGIN
   SELECT * from work, project where
  w_p_code=p_code
               and w_leader  =x;
END//

DELIMITER ;
DELIMITER //

CREATE PROCEDURE persons_count1()
BEGIN
DECLARE x  INT;
        SELECT sum(w_persons) into x from work ;

    if x > 20 then
            select "Too many workers ...";
    else
            select " Not much workers ...";
        end if;
END//

DELIMITER ;
```

```
DELIMITER //
  CREATE PROCEDURE persons_count2()
BEGIN
DECLARE x,y,z  INT;
    SELECT count(*) into x from work where w_persons > 10;
       SELECT count(*) into y from work where w_persons > 20;
    Select "Projects above 10 workers = ", x;
    Select "Projects above 20 workers = ", y;

            set z=x+y;
            select "Sum of x and y = ", z;

    if x<y then
            select "Many projects are big";
    else
            select "Many projects are small";
    end if;
END//
DELIMITER ;
```



Repeat-until

```
DELIMITER //
CREATE PROCEDURE do_repeat()

BEGIN
???
SET x = 0;
   REPEAT
         SELECT 'Hello';
         SET x = x + 1;
   UNTIL x > 5
   END REPEAT;

END//
DELIMITER ;
```



Do- while

```
DELIMITER //
CREATE PROCEDURE do_while()
BEGIN
  DECLARE v1 INT DEFAULT 5;

   WHILE v1 > 0 DO
                select v1;
                SET v1 = v1 - 1;
   END WHILE;

END//
DELIMITER ;
```