use test;

DELIMITER $$

SHOW CREATE PROCEDURE job\_data$$

CALL job\_data() $$

DELIMITER $$

CREATE PROCEDURE proc\_Variables()

BEGIN

DECLARE a INT DEFAULT 10;

DECLARE b, c INT;

SET a = a + 100;

SET b = 2;

SET c = a + b;

BEGIN

DECLARE c INT;

SET c = 5;

SELECT a, b, c;

END;

SELECT a, b, c;

END$$

CALL proc\_Variables() $$

DELIMITER $$

CREATE PROCEDURE proc\_User\_Variables() BEGIN SET @x = 15; SET @y = 10; SELECT @x, @y, @x-@y; END$$

CALL proc\_User\_Variables();

CREATE PROCEDURE proc\_IN(IN var1 INT)

BEGIN

SELECT \* FROM jobs LIMIT var1;

END$$

CALL proc\_IN(2)$$

DELIMITER $$

CREATE PROCEDURE my\_proc\_OUT(OUT highest\_salary INT)

BEGIN

SELECT MAX(MAX\_SALARY) INTO highest\_salary FROM jobs;

END$$

CALL my\_proc\_OUT(@M)$$

SELECT @M$$

USE test;

DELIMITER $$

CREATE PROCEDURE proc\_INOUT (INOUT maxsalary INT, IN desig

CHAR(30))

BEGIN

SELECT MAX\_SALARY INTO maxsalary FROM jobs WHERE

JOB\_TITLE = desig;

END$$

CALL proc\_INOUT(@C,'Stock Clerk')$$

SELECT @C$$

USE test;

DELIMITER $$

CREATE PROCEDURE proc\_Compare\_Salary(IN desig CHAR(30))

BEGIN

DECLARE progSal INT;

DECLARE presSal INT;

SELECT MIN\_SALARY INTO progSal FROM jobs WHERE JOB\_TITLE = desig;

SELECT MIN\_SALARY INTO presSal FROM jobs WHERE JOB\_TITLE = 'president';

SET presSal = presSal /2;

SELECT presSal;

SELECT progSal;

IF (progSal < presSal)

THEN

SELECT 'You need an increment.';

ELSEIF (progSal = presSal)

THEN

SELECT 'Salary is moderate';

ELSE

SELECT 'You are earning good';

END IF;

END$$

CALL proc\_Compare\_Salary('Programmer')$$

USE test;

DELIMITER $$

CREATE PROCEDURE proc\_Case (INOUT no\_employees INT, IN

salary INT)

BEGIN

CASE

WHEN (salary>10000)

THEN (SELECT COUNT(job\_id) INTO no\_employees FROM

jobs WHERE min\_salary>10000);

WHEN (salary<10000)

THEN (SELECT COUNT(job\_id) INTO no\_employees FROM

jobs WHERE min\_salary<10000);

ELSE (SELECT COUNT(job\_id) INTO no\_employees FROM jobs

WHERE min\_salary=10000);

END CASE;

END$$

CALL proc\_Case(@C,10001)$$

SELECT @C$$

CALL proc\_Case(@C,10000)$$

SELECT @C$$

CALL proc\_Case(@C,9999)$$

SELECT @C$$

use test;

CREATE TABLE number ( n1 real(3,2) );

DELIMITER $$

CREATE PROCEDURE proc\_LOOP (IN num INT)

BEGIN

DECLARE x INT;

SET x = 0;

loop\_label: LOOP

INSERT INTO number VALUES (rand());

SET x = x + 1;

IF x >= num

THEN LEAVE loop\_label;

END IF;

END LOOP;

END$$

DELIMITER $$

CALL proc\_LOOP(3)$$

Select \* from number$$

use test;

DELIMITER $$

CREATE PROCEDURE proc\_REPEAT (IN n INT)

BEGIN

SET @sum = 0;

SET @x = 1;

REPEAT

IF mod(@x, 2) = 0

THEN SET @sum = @sum + @x;

END IF;

SET @x = @x + 1;

UNTIL @x > n

END REPEAT;

END $$

CALL proc\_REPEAT(5)$$

SELECT @sum$$

CREATE PROCEDURE proc\_WHILE(IN n INT)

BEGIN

SET @sum = 0; SET @x = 1;

WHILE @x<> 0 THEN

SET @sum = @sum + @x; END

IF;

SET @x = @x + 1;

END WHILE;

END$$

CALL proc\_WHILE(5)$$

SELECT @sum$$

CALL my\_proc\_WHILE(10)$$

SELECT @sum$$

ALTER PROCEDURE proc\_WHILE COMMENT 'Modify Comment';

SHOW CREATE PROCEDURE proc\_WHILE;

SHOW PROCEDURE STATUS;

use test;

DELIMITER $$

CREATE PROCEDURE proc\_cursors(INOUT return\_val INT)

BEGIN

DECLARE a,b INT;

DECLARE cur\_1 CURSOR FOR SELECT max\_salary FROM jobs;

DECLARE CONTINUE HANDLER FOR NOT FOUND SET b = 1;

OPEN cur\_1;

REPEAT

FETCH cur\_1 INTO a;

UNTIL b = 1

END REPEAT;

CLOSE cur\_1;

SET return\_val = a;

END$$

CALL proc\_cursors(@R)$$

SELECT @R$$