Student Name: Muhammad Shoaib Khan  
NAVTTC Database Assignment  
Section: A

**Procedures Assignment**

CREATE TABLE `db\_assignment\_1`.`teacher\_table` (

`teacher\_id` INT NOT NULL,

`teacher\_name` VARCHAR(45) NULL,

`tea\_father\_name` VARCHAR(45) NULL,

`teacher\_salary` INT NULL,

`avg\_salary` INT NULL DEFAULT '25942',

PRIMARY KEY (`teacher\_id`));

INSERT INTO `db\_assignment\_1`.`teacher\_table` (`teacher\_id`, `teacher\_name`, `tea\_father\_name`, `teacher\_salary`) VALUES ('001', 'Teacher1', 'Tea\_fat\_01', '50000');

INSERT INTO `db\_assignment\_1`.`teacher\_table` (`teacher\_id`, `teacher\_name`, `tea\_father\_name`, `teacher\_salary`) VALUES ('002', 'Teacher2', 'Tea\_fat\_02', '25600');

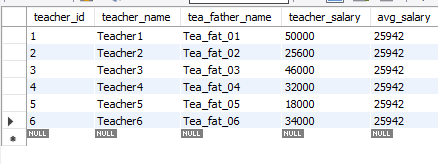
INSERT INTO `db\_assignment\_1`.`teacher\_table` (`teacher\_id`, `teacher\_name`, `tea\_father\_name`, `teacher\_salary`) VALUES ('003', 'Teacher3', 'Tea\_fat\_03', '46000');

INSERT INTO `db\_assignment\_1`.`teacher\_table` (`teacher\_id`, `teacher\_name`, `tea\_father\_name`, `teacher\_salary`) VALUES ('004', 'Teacher4', 'Tea\_fat\_04', '32000');

INSERT INTO `db\_assignment\_1`.`teacher\_table` (`teacher\_id`, `teacher\_name`, `tea\_father\_name`, `teacher\_salary`) VALUES ('005', 'Teacher5', 'Tea\_fat\_05', '18000');

INSERT INTO `db\_assignment\_1`.`teacher\_table` (`teacher\_id`, `teacher\_name`, `tea\_father\_name`, `teacher\_salary`) VALUES ('006', 'Teacher6', 'Tea\_fat\_06', '34000');

SELECT \* FROM db\_assignment\_1.teacher\_table;



**1.Create store procedure with no parameter:**

DELIMITER //

create Procedure proc1()

begin

DECLARE X INT;

SET X=1;

SELECT \* FROM teacher\_table

WHERE teacher\_id=X;

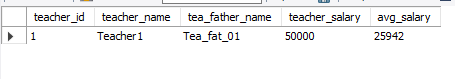
END //

DELIMITER ;

**Invoke Method:**

DELIMITER //

call proc1()



**2.Create store procedure with IN parameter (optional two or more parameters)**

DELIMITER //

CREATE PROCEDURE proc2(IN X INT, IN tname VARCHAR(255))

BEGIN

SELECT \* FROM teacher\_table

WHERE teacher\_id=X AND teacher\_name=tname;

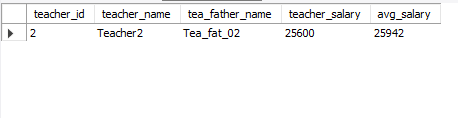
END//

DELIMITER ;

**Invoke Method:**

DELIMITER //

call proc2(2, 'Teacher2')



**3.Create store procedure with OUT parameter (optional two or more parameters)**

DELIMITER //

CREATE PROCEDURE proc3(OUT totalTeachers INT)

BEGIN

SELECT COUNT(teacher\_id) INTO totalTeachers FROM teacher\_table;

END//

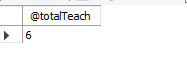
DELIMITER ;

**Invoke Method:**

DELIMITER //

call proc3(@totalTeach)

SELECT @totalTeach//



**4.Create store procedure with INOUT parameter ()**

DELIMITER //

CREATE PROCEDURE proc4(IN name VARCHAR(255), OUT totalTeachers INT, INOUT salary INT)

BEGIN

SELECT \* FROM teacher\_table

WHERE name=teacher\_name;

UPDATE teacher\_table

SET teacher\_salary=salary

WHERE name=teacher\_name;

SELECT COUNT(teacher\_id) INTO totalTeachers FROM teacher\_table;

END//

DELIMITER ;

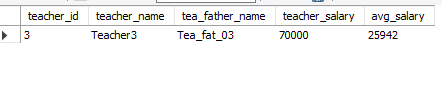
**Invoke Method:**

DELIMITER //

SET SQL\_SAFE\_UPDATES = 0;

SET @teacher\_salary=70000

CALL proc4('teacher3',@tol,@teacher\_salary)



**5. USE IF ELSEIF, ELSE**

DELIMITER //

CREATE PROCEDURE SalaryStatus(IN emp\_id INT, OUT sal\_status VARCHAR(255))

BEGIN

DECLARE curr\_sal DECIMAL(8,2);

DECLARE avrg\_sal DECIMAL(8,2);

SELECT AVG(teacher\_salary) INTO avrg\_sal FROM teacher\_table;

SELECT teacher\_salary INTO curr\_sal FROM teacher\_table

WHERE emp\_id=teacher\_id;

IF curr\_sal < avrg\_sal THEN

SET sal\_status='Less than Average salary';

ELSEIF curr\_sal=avrg\_sal THEN

SET sal\_status='Equal to average salary';

ELSEIF curr\_sal>avrg\_sal THEN

SET sal\_status='Greater than average salary';

END IF;

END//

DELIMITER ;

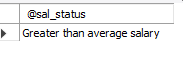
**Invoke Method:**

DELIMITER //

SET SQL\_SAFE\_UPDATES = 0;

CALL SalaryStatus(3,@sal\_status)//

SELECT @sal\_status;



**6. Drop any one store procedure**

DROP PROCEDURE IF EXISTS display;