CREATE TABLE New\_Roll(id INT PRIMARY KEY, name VARCHAR(400), dept VARCHAR(300));

Query OK, 0 rows affected (0.186 sec)

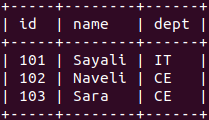
CREATE TABLE Old\_Roll(id INT PRIMARY KEY, name VARCHAR(400), dept VARCHAR(300));

Query OK, 0 rows affected (0.182 sec)

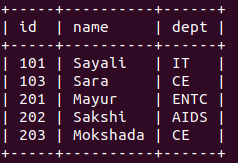
INSERT INTO New\_Roll values (101,'Sayali','IT'),(102,'Naveli','CE'),(103,'Sara','CE');

INSERT INTO Old\_Roll values (201,'Mayur','ENTC'),(202,'Sakshi','AIDS'),(203,'Mokshada','CE'),(101,'Sayali','IT'),(103,'Sara','CE');

SELECT \* FROM New\_Roll;



SELECT \* FROM Old\_Roll;



DELIMITER $$

CREATE PROCEDURE add\_to\_Old()

BEGIN

-- Variables to hold fetched row values

DECLARE v\_id INT;

DECLARE v\_name VARCHAR(30);

DECLARE v\_dept VARCHAR(20);

-- flag for cursor end

DECLARE done INT DEFAULT FALSE;

-- cursor declaration

DECLARE cur CURSOR FOR SELECT id, name, dept FROM New\_Roll;

-- Handler to set done = TRUE when no more rows

DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = TRUE;

-- Open cursor

OPEN cur;

get\_data: WHILE NOT done DO

FETCH cur INTO v\_id, v\_name, v\_dept;

-- If there’s no row, return 1

IF NOT EXISTS (SELECT 1 FROM O\_RollCall WHERE id = v\_id) THEN INSERT INTO Old\_Roll(id, name, dept) VALUES (v\_id, v\_name, v\_dept);

END IF;

END WHILE;

CLOSE cur;

END$$

DELIMITER ;

CALL add\_to\_Old();