Question 3:

A. Procedure to insert data in the employee table:

```
DELIMITER //

CREATE PROCEDURE InsertEmployeeData(

    IN emp_id_param INT,

    IN employee_name_param VARCHAR(255),

    IN street_param VARCHAR(255),

    IN city_param VARCHAR(255)

)

BEGIN

    INSERT INTO employee VALUES (emp_id_param, employee_name_param, street_param,
    city_param);

END //

DELIMITER ;
```

To call this procedure, you can use:

```
CALL InsertEmployeeData(16, 'New Employee', '789 Pine St', 'City3');
```

B. Procedure to update the contact number of an employee in the employee table:

```
DELIMITER //

CREATE PROCEDURE UpdateContactNumber(

    IN emp_id_param INT,

    IN new_contact_number_param VARCHAR(20)

)

BEGIN

    UPDATE employee

    SET contact_number = new_contact_number_param

    WHERE emp_id = emp_id_param;

END //

DELIMITER ;
```

To call this procedure, you can use:

```
CALL UpdateContactNumber(1, '123-456-7890');
```

C. Procedure to find the name of the manager for a given employee id:

```
DELIMITER //
CREATE PROCEDURE FindManagerName(
    IN emp_id_param INT
)
BEGIN
    SELECT manager.manager_name
    FROM manage
    JOIN manager ON manage.manager_id = manager.manager_id
    WHERE manage.emp_id = emp_id_param;
END //
DELIMITER ;
```

To call this procedure, you can use:

```
CALL FindManagerName(1);
```

D. Procedure to get all the details (emp_id, name, city of residence, company name, city of work, manager name, salary) for a given employee id:

```
DELIMITER //
CREATE PROCEDURE GetEmployeeDetails(
    IN emp_id_param INT
)
BEGIN
    SELECT
        employee.emp_id,
        employee.employee_name,
        employee.city AS residence_city,
        company.company_name,
        company.city AS work_city,
        manager.manager_name,
```

```
    works.salary
  FROM employee
  JOIN works ON employee.emp_id = works.emp_id
  JOIN company ON works.comp_id = company.comp_id
  JOIN manage ON employee.emp_id = manage.emp_id
  JOIN manager ON manage.manager_id = manager.manager_id
  WHERE employee.emp_id = emp_id_param;
END //
DELIMITER ;
```

To call this procedure, you can use:

```
CALL GetEmployeeDetails(1);
```

Question 4:

SQL code to create the three tables:

```
CREATE TABLE Student (
    Roll_no INT PRIMARY KEY,
    Name VARCHAR(255),
    Contact VARCHAR(20),
    Marks DECIMAL(5,2)
);
CREATE TABLE Student_copy (
    Roll_no INT PRIMARY KEY,
    Contact VARCHAR(20)
);
CREATE TABLE Student_update_copy (
    Roll_no INT PRIMARY KEY,
    New_Contact VARCHAR(20),
    Old_Contact VARCHAR(20)
);
```

A. Trigger to insert Roll no and contact number of a student on the insertion of any record in Table Student:

```
DELIMITER //
CREATE TRIGGER InsertStudentCopy
AFTER INSERT
ON Student
FOR EACH ROW
BEGIN
    INSERT INTO Student_copy (Roll_no, Contact)
    VALUES (NEW.Roll_no, NEW.Contact);
END //
DELIMITER ;
```

B. Trigger to insert Roll no, New Contact number, and old contact number of a student on the updation of the contact number in Table Student:

```
DELIMITER //
CREATE TRIGGER UpdateStudentCopy
AFTER UPDATE
ON Student
FOR EACH ROW
BEGIN
   IF NEW.Contact <> OLD.Contact THEN
      INSERT INTO Student_update_copy (Roll_no, New_Contact, Old_Contact)
      VALUES (NEW.Roll_no, NEW.Contact, OLD.Contact);
   END IF;
END //
DELIMITER ;
```