

Please grade cflnInsert for the Trigger.

```
DELIMITER //  
CREATE TRIGGER cflnInsert  
BEFORE Insert  
ON CrimeFilter  
FOR EACH ROW  
BEGIN  
IF EXISTS (select * from Users where username = new.username) THEN  
UPDATE Users SET last_updated = CURRENT_TIMESTAMP WHERE Users.username =  
new.username;  
END IF;  
END; //  
DELIMITER ;
```

```
DELIMITER //  
CREATE TRIGGER userInsert  
BEFORE INSERT  
ON Users  
FOR EACH ROW  
BEGIN  
IF new.last_updated IS NULL THEN  
SET new.last_updated = CURRENT_TIMESTAMP;  
END IF;  
END; //  
DELIMITER ;
```

```
DELIMITER //  
CREATE TRIGGER userUpdate  
BEFORE Update  
ON Users  
FOR EACH ROW  
BEGIN  
IF new.last_updated IS NULL OR NOT new.password = old.password THEN  
SET new.last_updated = CURRENT_TIMESTAMP;  
END IF;  
END; //  
DELIMITER ;
```

```
DELIMITER //
CREATE TRIGGER wflInsert
BEFORE Insert
ON WeaponFilter
FOR EACH ROW
BEGIN
IF EXISTS (select * from Users where username = new.username) THEN
UPDATE Users SET last_updated = CURRENT_TIMESTAMP WHERE Users.username =
new.username;
END IF;
END; //
DELIMITER ;
```

```
DELIMITER //
CREATE TRIGGER pflInsert
BEFORE Insert
ON PremisFilter
FOR EACH ROW
BEGIN
IF EXISTS (select * from Users where username = new.username) THEN
UPDATE Users SET last_updated = CURRENT_TIMESTAMP WHERE Users.username =
new.username;
END IF;
END; //
DELIMITER ;
```

This is the main Stored Procedure to be graded.
getAllCrimesWF, PF, CF, are all basically identical to each other. getAllCrimes is different.

```
DELIMITER //
CREATE PROCEDURE getAll(IN usern VARCHAR(255), IN id INT)
BEGIN
    CALL getAllCrimes(id);
    CALL getAllCrimesWF(usern);
    CALL getAllCrimesCF(usern);
    CALL getAllCrimesPF(usern);

    IF NOT EXISTS (SELECT * FROM crimeBuffer2) THEN
        INSERT INTO crimeBuffer2 (SELECT * FROM crimeBuffer UNION SELECT * FROM
crimeBuffer3 UNION SELECT * FROM crimeBuffer4);
    END IF;

    IF NOT EXISTS (SELECT * FROM crimeBuffer3) THEN
        INSERT INTO crimeBuffer3 (SELECT * FROM crimeBuffer UNION SELECT * FROM
crimeBuffer2 UNION SELECT * FROM crimeBuffer4);
    END IF;

    IF NOT EXISTS (SELECT * FROM crimeBuffer4) THEN
        INSERT INTO crimeBuffer4 (SELECT * FROM crimeBuffer UNION SELECT * FROM
crimeBuffer2 UNION SELECT * FROM crimeBuffer3);
    END IF;

    SELECT crimeBuffer.dr_no as dr_no, crimeBuffer.latitude as latitude,
crimeBuffer.longitude as longitude, crimeBuffer.crime_code as crime_code,
crimeBuffer.premis_code as premis_code, crimeBuffer.weapon_code as weapon_code
    FROM crimeBuffer JOIN crimeBuffer2 ON crimeBuffer.dr_no = crimeBuffer2.dr_no
    JOIN crimeBuffer3 ON crimeBuffer.dr_no = crimeBuffer3.dr_no
    JOIN crimeBuffer4 ON crimeBuffer.dr_no = crimeBuffer4.dr_no;
END //
DELIMITER ;
```

```

DELIMITER //
CREATE PROCEDURE getAllCrimesPF(IN usern VARCHAR(255))
BEGIN
DECLARE done int default 0;
DECLARE curr_user VARCHAR(255);
DECLARE curr_wc INT;
DECLARE weaponcur CURSOR FOR SELECT username, premis_code FROM
PremisFilter;
DECLARE CONTINUE HANDLER FOR NOT FOUND SET done =1;

DROP TABLE IF EXISTS crimeBuffer4;
CREATE TABLE crimeBuffer4 (
    DR_NO INT PRIMARY KEY,
    date_occ DATE,
    time_occ TIME,
    crime_code INT,
    weapon_code INT,
    premis_code INT,
    vict_age INT,
    vict_sex VARCHAR(255),
    vict_descent VARCHAR(255),
    latitude REAL,
    longitude REAL
);
OPEN weaponcur;

REPEAT
    FETCH weaponcur INTO curr_user, curr_wc;

    IF curr_user = usern AND NOT EXISTS (SELECT * FROM crimeBuffer4 WHERE curr_wc =
    premis_code) THEN
        INSERT INTO crimeBuffer4 (SELECT * FROM Crimes WHERE curr_wc =
    premis_code);
    END IF;

UNTIL done
END REPEAT;

CLOSE weaponcur;

END //
DELIMITER ;

```

```

DELIMITER //
CREATE PROCEDURE getAllCrimesCF(IN usern VARCHAR(255))
BEGIN
DECLARE done int default 0;
DECLARE curr_user VARCHAR(255);
DECLARE curr_wc INT;
DECLARE weaponcur CURSOR FOR SELECT username, crime_code FROM CrimeFilter;
DECLARE CONTINUE HANDLER FOR NOT FOUND SET done =1;

DROP TABLE IF EXISTS crimeBuffer3;
CREATE TABLE crimeBuffer3 (
    DR_NO INT PRIMARY KEY,
    date_occ DATE,
    time_occ TIME,
    crime_code INT,
    weapon_code INT,
    premis_code INT,
    vict_age INT,
    vict_sex VARCHAR(255),
    vict_descent VARCHAR(255),
    latitude REAL,
    longitude REAL
);
OPEN weaponcur;

REPEAT
    FETCH weaponcur INTO curr_user, curr_wc;

    IF curr_user = usern AND NOT EXISTS (SELECT * FROM crimeBuffer3 WHERE curr_wc =
    crime_code) THEN
        INSERT INTO crimeBuffer3 (SELECT * FROM Crimes WHERE curr_wc = crime_code);
    END IF;

UNTIL done
END REPEAT;

CLOSE weaponcur;

END //
DELIMITER ;

```

```

DELIMITER //
CREATE PROCEDURE getAllCrimesWF(IN usern VARCHAR(255))
BEGIN
DECLARE done int default 0;
DECLARE curr_user VARCHAR(255);
DECLARE curr_wc INT;
DECLARE weaponcur CURSOR FOR SELECT username, weapon_code FROM WeaponFilter;
DECLARE CONTINUE HANDLER FOR NOT FOUND SET done =1;

DROP TABLE IF EXISTS crimeBuffer2;
CREATE TABLE crimeBuffer2 (
    DR_NO INT PRIMARY KEY,
    date_occ DATE,
    time_occ TIME,
    crime_code INT,
    weapon_code INT,
    premis_code INT,
    vict_age INT,
    vict_sex VARCHAR(255),
    vict_descent VARCHAR(255),
    latitude REAL,
    longitude REAL
);
OPEN weaponcur;

REPEAT
    FETCH weaponcur INTO curr_user, curr_wc;

    IF curr_user = usern AND NOT EXISTS (SELECT * FROM crimeBuffer2 WHERE curr_wc =
    weapon_code) THEN
        INSERT INTO crimeBuffer2 (SELECT * FROM Crimes WHERE curr_wc =
    weapon_code);
    END IF;

UNTIL done
END REPEAT;

CLOSE weaponcur;

END //
DELIMITER ;

```

This is the one with the radius with the Views it gets all the crimes in the radius of the view_id.

```
DELIMITER //
CREATE PROCEDURE getAllCrimes(
IN id INT)
BEGIN
DECLARE center_long REAL;
DECLARE center_lat REAL;
DECLARE center_rad REAL;
DECLARE done int default 0;
DECLARE curr_drno INT;
DECLARE curr_lat REAL;
DECLARE curr_long REAL;
DECLARE crimecur CURSOR FOR SELECT dr_no, lat, longitude FROM Crimes;
DECLARE CONTINUE HANDLER FOR NOT FOUND SET done =1;

SELECT
    longitude, lat, radius
INTO center_long , center_lat , center_rad FROM
    Views
WHERE
    id = view_id;

DROP TABLE IF EXISTS crimeBuffer;
CREATE TABLE crimeBuffer (
    DR_NO INT PRIMARY KEY,
    date_occ DATE,
    time_occ TIME,
    crime_code INT,
    weapon_code INT,
    premis_code INT,
    vict_age INT,
    vict_sex VARCHAR(255),
    vict_descent VARCHAR(255),
    latitude REAL,
    longitude REAL
);
OPEN crimecur;

REPEAT
    FETCH crimecur INTO curr_drno, curr_lat, curr_long;
    CALL Distance(curr_lat, curr_long, center_lat, center_long, @distance);
```

```
IF @distance < center_rad THEN
    INSERT INTO crimeBuffer (SELECT * FROM Crimes WHERE curr_drno = dr_no);
END IF;

UNTIL done
END REPEAT;

CLOSE crimecur;

END //
DELIMITER ;
```