

Queries

```
SELECT U.name
FROM user_yelp U
WHERE U.user_id =
    (SELECT temp.friend
     FROM
        (SELECT F.friend, COUNT(*) AS num
         FROM friendship F
         GROUP BY F.friend) temp
     WHERE temp.num =
        (SELECT MAX(temp.num)
         FROM
            (SELECT F.friend, COUNT(*) AS num
             FROM friendship F
             GROUP BY F.friend) temp))

SELECT COUNT(*)
FROM((SELECT R.user_id AS uid, R2.user_id AS uid2
      FROM review R, review R2
      WHERE R.user_id != R2.user_id AND R.business_id = R2.business_id AND R.stars = 1 AND R2.stars = 5)
 INTERSECT
 (SELECT *
  FROM friendship F)) AS temp

SELECT B.name
FROM business B
WHERE B.city = 'Sherwood Park'
AND B.business_id IN (SELECT tipCount.business_id
                     FROM (SELECT T.business_id, COUNT(*) AS num_tips
                          FROM tip T
                          GROUP BY T.business_id) tipCount,
                     (SELECT R.business_id, COUNT(*) AS num_5_stars
                      FROM review R
                      WHERE R.stars = 5
                      GROUP BY R.business_id) fivestarCount
                     WHERE tipCount.business_id = fivestarCount.business_id AND tipCount.num_tips > fivestarCount.num_5_stars)
ORDER BY B.name

SELECT AVG(CAST(B.stars AS DECIMAL(2,1)))
FROM business B
WHERE B.name = 'Starbucks' AND B.business_id IN (SELECT R.business_id
                                                FROM review R JOIN user_yelp U ON R.user_id = U.user_id
                                                WHERE U.name = 'Linda')
AND B.business_id IN (SELECT R.business_id
                     FROM review R
                     WHERE R.funny > 0)
```

Triggers

```

-- 4.4.4
CREATE TRIGGER newReview ON review
AFTER INSERT
AS
IF (1 < (SELECT COUNT(*)
FROM review R, inserted I
WHERE R.business_id = I.business_id AND R.user_id = I.user_id))
BEGIN -- more than one review from this user for this business
UPDATE business
SET stars = (SELECT AVG(CAST(S.stars AS DECIMAL(2, 1)))
FROM (SELECT *
FROM review R2
WHERE R2.date = (SELECT temp.recentDate
FROM (SELECT R2.user_id, MAX(R2.date) as recentDate
FROM review R2, inserted I
WHERE R2.business_id = I.business_id
GROUP BY R2.user_id) temp
WHERE R2.user_id = temp.user_id)) AS S, inserted I
WHERE S.business_id = I.business_id)
WHERE business_id IN (SELECT business_id
FROM inserted)
END
ELSE -- one review from this user for this business
BEGIN
UPDATE business
SET review_count = review_count + 1
WHERE business_id IN (SELECT business_id
FROM inserted);
UPDATE business
SET stars = (SELECT AVG(CAST(S.stars AS DECIMAL(2, 1)))
FROM (SELECT *
FROM review R2
WHERE R2.date = (SELECT temp.recentDate
FROM (SELECT R2.user_id, MAX(R2.date) as recentDate
FROM review R2, inserted I
WHERE R2.business_id = I.business_id
GROUP BY R2.user_id) temp
WHERE R2.user_id = temp.user_id)) AS S, inserted I
WHERE S.business_id = I.business_id)
WHERE business_id IN (SELECT business_id
FROM inserted);
END
END

CREATE TRIGGER givingTip ON tip
AFTER INSERT
AS
IF NOT EXISTS (SELECT *
FROM review R, inserted I
WHERE I.user_id = R.user_id AND I.business_id = R.business_id )
BEGIN
RAISERROR ('Users must review the business before giving a tip.', 1, 1);
ROLLBACK TRANSACTION;
END

```