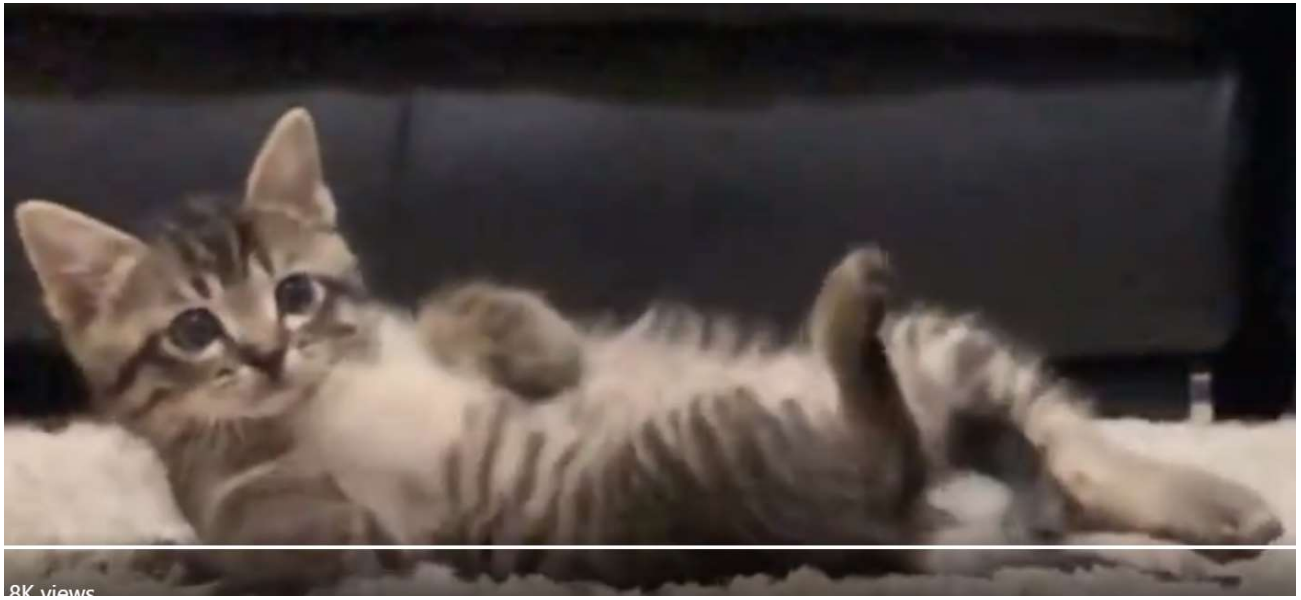
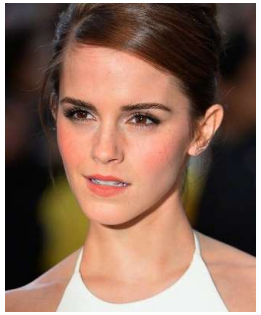
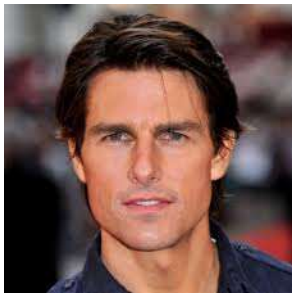
MT: Before & After



ACTOR (actor_name, gender, date_of_birth)
MOVIE (movie_title, release_year, genre, movie_length)
CAST_MEMBER (actor_name, movie_title, release_year, actor_role)
AWARDS_EVENT (event_name, event_year, venue)
NOMINATION (event_name, event_year, movie_title, release_year, category, won)



actor_name	movie_title	release_year
Emma	M1	2020
Emma	M2	2020
Emma	M3	2020
Emma	M4	2020
Scarlett	M2	2020
Scarlett	M4	2020
Brad	M1	2020
Tom	M4	2020

event_name	event_year	movie_title	release_year	category	won
Oscar	2020	M1	2020	Horror	Yes
Oscar	2020	M2	2020	Thriller	Yes
Oscar	2020	M2	2020	Action	No
Oscar	2020	M3	2020	Thriller	No
Oscar	2020	M4	2020	Horror	Yes
Oscar	2020	M4	2020	Action	Yes

a) Retrieve the names of actors whose every movie won an award.
b) Retrieve the names of actors whose every movie won exactly one award.
c) Retrieve the names of actors whose every movie had an award nomination.
d) Retrieve the names of actors who was a cast member in every movie that won an award.
e) Retrieve the names of actors whose every movie won awards for all of their nominations

```
SELECT DISTINCT C.actor_name
FROM CAST_MEMBER C
WHERE  (SELECT COUNT(*)
          FROM CAST_MEMBER CM
          WHERE CM.actor_name = C.actor_name) =
         (SELECT COUNT(DISTINCT N.movie_title, N.release_year)
          FROM NOMINATION N, CAST_MEMBER CM2
          WHERE N.won = 'yes' AND
                    N.movie_title=CM2.movie_title AND
                    N.release_year = CM2.release_year AND
                    CM2.actor_name = C.actor_name AND
                   NOT EXISTS (SELECT DISTINCT N2.movie_title, N2.release_year
                                       FROM NOMINATION N2
                                       WHERE  N.movie_title=N2.movie_title AND N.release_year = N2.release_year AND N2.won = 'no'))
```

Emma	M1	2020
Emma	M2	2020
Emma	M3	2020
Emma	M4	2020
Scarlett	M2	2020
Scarlett	M4	2020
Brad	M1	2020
Tom	M4	2020

ename	eyear	movie_title	release_year	category	won
Oscar	2020	M1	2020	Horror	Yes
Oscar	2020	M2	2020	Thriller	Yes
Oscar	2020	M2	2020	Action	No
Oscar	2020	M3	2020	Thriller	No
Oscar	2020	M4	2020	Horror	Yes
Oscar	2020	M4	2020	Action	Yes

```
SELECT DISTINCT C.actor_name
FROM CAST_MEMBER C
WHERE  (SELECT COUNT(*)
        FROM CAST_MEMBER CM
        WHERE CM.actor_name = C.actor_name) =
       (SELECT COUNT(DISTINCT N.movie_title, N.release_year)
        FROM NOMINATION N, CAST_MEMBER CM2
        WHERE N.won = 'yes' AND
              N.movie_title=CM2.movie_title AND
              N.release_year = CM2.release_year AND
              CM2.actor_name = C.actor_name AND
             NOT EXISTS (SELECT DISTINCT N2.movie_title, N2.release_year
                         FROM NOMINATION N2
                         WHERE  N.movie_title=N2.movie_title AND N.release_year = N2.release_year AND N2.won = 'no'))
```

c
4

Emma	M1	2020	Oscar	2020	M1	2020	Horror	Yes
Emma	M2	2020	Oscar	2020	M2	2020	Thriller	Yes
Emma	M4	2020	Oscar	2020	M4	2020	Horror	Yes
Emma	M4	2020	Oscar	2020	M4	2020	Action	Yes

c
2

M1	2020
M4	2020
M4	2020

Emma	M1	2020
Emma	M2	2020
Emma	M3	2020
Emma	M4	2020
Scarlett	M2	2020
Scarlett	M4	2020
Brad	M1	2020
Tom	M4	2020

ename	eyear	movie_title	release_year	category	won
Oscar	2020	M1	2020	Horror	Yes
Oscar	2020	M2	2020	Thriller	Yes
Oscar	2020	M2	2020	Action	No
Oscar	2020	M3	2020	Thriller	No
Oscar	2020	M4	2020	Horror	Yes
Oscar	2020	M4	2020	Action	Yes

Scarlett	M2	2020	Oscar	2020	M2	2020	Thriller	Yes
Scarlett	M4	2020	Oscar	2020	M4	2020	Horror	Yes
Scarlett	M4	2020	Oscar	2020	M4	2020	Action	Yes

```
SELECT DISTINCT C.actor_name
FROM CAST_MEMBER C
WHERE  (SELECT COUNT(*)
        FROM CAST_MEMBER CM
        WHERE CM.actor_name = C.actor_name) =
       (SELECT COUNT(DISTINCT N.movie_title, N.release_year)
        FROM NOMINATION N, CAST_MEMBER CM2
        WHERE N.won = 'yes' AND
              N.movie_title=CM2.movie_title AND
              N.release_year = CM2.release_year AND
              CM2.actor_name = C.actor_name AND
             NOT EXISTS (SELECT DISTINCT N2.movie_title, N2.release_year
                         FROM NOMINATION N2
                         WHERE  N.movie_title=N2.movie_title AND N.release_year = N2.release_year AND N2.won = 'no'))
```

c
2

c
1

M4	2020
M4	2020

Emma	M1	2020
Emma	M2	2020
Emma	M3	2020
Emma	M4	2020
Scarlett	M2	2020
Scarlett	M4	2020
Brad	M1	2020
Tom	M4	2020

ename	eyear	movie_title	release_year	category	won
Oscar	2020	M1	2020	Horror	Yes
Oscar	2020	M2	2020	Thriller	Yes
Oscar	2020	M2	2020	Action	No
Oscar	2020	M3	2020	Thriller	No
Oscar	2020	M4	2020	Horror	Yes
Oscar	2020	M4	2020	Action	Yes

Brad	M1	2020

Oscar	2020	M1	2020	Horror	Yes

```
SELECT DISTINCT C.actor_name
FROM CAST_MEMBER C
WHERE  (SELECT COUNT(*)
        FROM CAST_MEMBER CM
        WHERE CM.actor_name = C.actor_name) =
       (SELECT COUNT(DISTINCT N.movie_title, N.release_year)
        FROM NOMINATION N, CAST_MEMBER CM2
        WHERE N.won = 'yes' AND
              N.movie_title=CM2.movie_title AND
              N.release_year = CM2.release_year AND
              CM2.actor_name = C.actor_name AND
             NOT EXISTS (SELECT DISTINCT N2.movie_title, N2.release_year
                         FROM NOMINATION N2
                         WHERE  N.movie_title=N2.movie_title AND N.release_year = N2.release_year AND N2.won = 'no'))
```

c
1

c
1

M1	2020

actor_name
Brad

Emma	M1	2020
Emma	M2	2020
Emma	M3	2020
Emma	M4	2020
Scarlett	M2	2020
Scarlett	M4	2020
Brad	M1	2020
Tom	M4	2020

ename	eyear	movie_title	release_year	category	won
Oscar	2020	M1	2020	Horror	Yes
Oscar	2020	M2	2020	Thriller	Yes
Oscar	2020	M2	2020	Action	No
Oscar	2020	M3	2020	Thriller	No
Oscar	2020	M4	2020	Horror	Yes
Oscar	2020	M4	2020	Action	Yes

Tom	M4	2020	Oscar	2020	M4	2020	Horror	Yes
Tom	M4	2020	Oscar	2020	M4	2020	Action	Yes

```
SELECT DISTINCT C.actor_name
FROM CAST_MEMBER C
WHERE  (SELECT COUNT(*)
        FROM CAST_MEMBER CM
        WHERE CM.actor_name = C.actor_name) =
       (SELECT COUNT(DISTINCT N.movie_title, N.release_year)
        FROM NOMINATION N, CAST_MEMBER CM2
        WHERE N.won = 'yes' AND
              N.movie_title=CM2.movie_title AND
              N.release_year = CM2.release_year AND
              CM2.actor_name = C.actor_name AND
             NOT EXISTS (SELECT DISTINCT N2.movie_title, N2.release_year
                         FROM NOMINATION N2
                         WHERE  N.movie_title=N2.movie_title AND N.release_year = N2.release_year AND N2.won = 'no'))
```

c
1

c
1

M4	2020
M4	2020

actor_name
Brad
Tom

Emma	M1	2020
Emma	M2	2020
Emma	M3	2020
Emma	M4	2020
Scarlett	M2	2020
Scarlett	M4	2020
Brad	M1	2020
Tom	M4	2020

ename	eyear	movie_title	release_year	category	won
Oscar	2020	M1	2020	Horror	Yes
Oscar	2020	M2	2020	Thriller	Yes
Oscar	2020	M2	2020	Action	No
Oscar	2020	M3	2020	Thriller	No
Oscar	2020	M4	2020	Horror	Yes
Oscar	2020	M4	2020	Action	Yes

a) Retrieve the names of actors whose every movie won an award.

```
SELECT DISTINCT C.actor_name
FROM CAST_MEMBER C
WHERE  (SELECT COUNT(*)
        FROM CAST_MEMBER CM
        WHERE CM.actor_name = C.actor_name) =
       (SELECT COUNT(DISTINCT N.movie_title, N.release_year)
        FROM NOMINATION N, CAST_MEMBER CM2
        WHERE N.won = 'yes' AND
              N.movie_title=CM2.movie_title AND
              N.release_year = CM2.release_year AND
              CM2.actor_name = C.actor_name AND
             NOT EXISTS (SELECT DISTINCT N2.movie_title, N2.release_year
                         FROM NOMINATION N2
                         WHERE  N.movie_title=N2.movie_title AND N.release_year = N2.release_year AND N2.won = 'no'))
```

actor_name
Scarlett
Brad
Tom

Emma	M1	2020
Emma	M2	2020
Emma	M3	2020
Emma	M4	2020
Scarlett	M2	2020
Scarlett	M4	2020
Brad	M1	2020
Tom	M4	2020

ename	eyear	movie_title	release_year	category	won
Oscar	2020	M1	2020	Horror	Yes
Oscar	2020	M2	2020	Thriller	Yes
Oscar	2020	M2	2020	Action	No
Oscar	2020	M3	2020	Thriller	No
Oscar	2020	M4	2020	Horror	Yes
Oscar	2020	M4	2020	Action	Yes

b) Retrieve the names of actors whose every movie won exactly one award.

actor_name
Brad

```
SELECT DISTINCT C.actor_name
FROM CAST_MEMBER C
WHERE  (SELECT COUNT(*)
        FROM CAST_MEMBER CM
        WHERE CM.actor_name = C.actor_name) =
       (SELECT COUNT(DISTINCT N.movie_title, N.release_year)
        FROM NOMINATION N, CAST_MEMBER CM2
        WHERE N.won = 'yes' AND
              N.movie_title=CM2.movie_title AND
              N.release_year = CM2.release_year AND
              CM2.actor_name = C.actor_name AND
              UNIQUE (SELECT N2.movie_title, N2.release_year
                      FROM NOMINATION N2
                      WHERE  N.movie_title=N2.movie_title AND N.release_year = N2.release_year AND N2.won = 'yes' ))
```

Emma	M1	2020
Emma	M2	2020
Emma	M3	2020
Emma	M4	2020
Scarlett	M2	2020
Scarlett	M4	2020
Brad	M1	2020
Tom	M4	2020

ename	eyear	movie_title	release_year	category	won
Oscar	2020	M1	2020	Horror	Yes
Oscar	2020	M2	2020	Thriller	Yes
Oscar	2020	M2	2020	Action	No
Oscar	2020	M3	2020	Thriller	No
Oscar	2020	M4	2020	Horror	Yes
Oscar	2020	M4	2020	Action	Yes

c) Retrieve the names of actors whose every movie had an award nomination

```
SELECT DISTINCT C.actor_name
FROM CAST_MEMBER C
WHERE  (SELECT COUNT(*)
        FROM CAST_MEMBER CM
        WHERE CM.actor_name = C.actor_name) =
       (SELECT COUNT(DISTINCT N.movie_title, N.release_year)
        FROM NOMINATION N, CAST_MEMBER CM2
        WHERE N.won = 'yes' AND
              N.movie_title=CM2.movie_title AND
              N.release_year = CM2.release_year AND
              CM2.actor_name = C.actor_name AND
             NOT EXISTS (SELECT DISTINCT N2.movie_title, N2.release_year
                         FROM NOMINATION N2
                         WHERE  N.movie_title=N2.movie_title AND N.release_year = N2.release_year AND N2.won = 'no'))
```

actor_name
Emma
Scarlett
Brad
Tom

Emma	M1	2020
Emma	M2	2020
Emma	M3	2020
Emma	M4	2020
Scarlett	M2	2020
Scarlett	M4	2020
Brad	M1	2020
Tom	M4	2020

ename	eyear	movie_title	release_year	category	won
Oscar	2020	M1	2020	Horror	Yes
Oscar	2020	M2	2020	Thriller	Yes
Oscar	2020	M2	2020	Action	No
Oscar	2020	M3	2020	Thriller	No
Oscar	2020	M4	2020	Horror	Yes
Oscar	2020	M4	2020	Action	Yes

d) Retrieve the names of actors who was a cast member in every movie that won an award.

```
SELECT DISTINCT C.actor_name
FROM CAST_MEMBER C
WHERE  (SELECT COUNT(*)
        FROM CAST_MEMBER CM
        WHERE CM.actor_name = C.actor_name) =
       (SELECT COUNT(DISTINCT N.movie_title, N.release_year)
        FROM NOMINATION N, CAST_MEMBER CM2
        WHERE N.won = 'yes' AND
              N.movie_title=CM2.movie_title AND
              N.release_year = CM2.release_year AND
              CM2.actor_name = C.actor_name AND
             NOT EXISTS (SELECT DISTINCT N2.movie_title, N2.release_year
                         FROM NOMINATION N2
                         WHERE  N.movie_title=N2.movie_title AND N.release_year = N2.release_year AND N2.won = 'no'))
```

actor_name
Emma

What if I simply count the number of distinct movies that got an award?

Emma	M1	2020
Emma	M2	2020
Emma	M3	2020
Emma	M4	2020
Scarlett	M2	2020
Scarlett	M4	2020
Brad	M1	2020
Tom	M4	2020

ename	eyear	movie_title	release_year	category	won
Oscar	2020	M1	2020	Horror	Yes
Oscar	2020	M2	2020	Thriller	Yes
Oscar	2020	M2	2020	Action	No
Oscar	2020	M3	2020	Thriller	No
Oscar	2020	M4	2020	Horror	Yes
Oscar	2020	M4	2020	Action	Yes

d) Retrieve the names of actors who was a cast member in every movie that won an award.

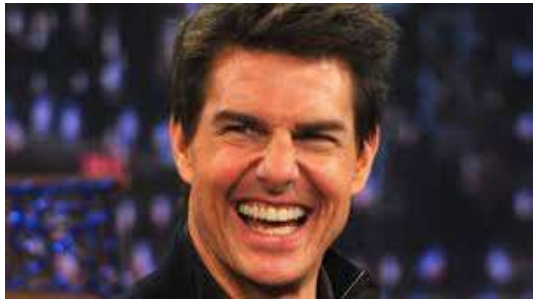
```
SELECT DISTINCT C.actor_name
FROM CAST_MEMBER C
WHERE  (SELECT COUNT(DISTINCT N.movie_title, N.release_year)
         FROM CAST_MEMBER CM, NOMINATION N
         WHERE CM.actor_name = C.actor_name AND
                  N.movie_title=CM.movie_title AND
                  N.release_year = CM.release_year AND
                  N.won = 'yes' ) =
        (SELECT COUNT(DISTINCT N.movie_title, N.release_year)
         FROM NOMINATION N, CAST_MEMBER CM2
         WHERE N.won = 'yes)
```

actor_name
Emma

ACTOR (actor_name, gender, date_of_birth)
MOVIE (movie_title, release_year, genre, movie_length)
CAST_MEMBER (actor_name, movie_title, release_year, actor_role)
AWARDS_EVENT (event_name, event_year, venue)
NOMINATION (event_name, event_year, movie_title, release_year, category, won)

Check the box next to each of the below SQL queries, which are guaranteed to be executed by the DBMS (i.e., will never be rejected due to a constraint violation) on all legal instances of this schema, where each table is non-empty (assume all FKs are declared with the default "no action" option)?

DELETE FROM *CAST_MEMBER*



ACTOR (actor_name, gender, date_of_birth)
MOVIE (movie_title, release_year, genre, movie_length)
CAST_MEMBER (actor_name, movie_title, release_year, actor_role)
AWARDS_EVENT (event_name, event_year, venue)
NOMINATION (event_name, event_year, movie_title, release_year, category, won)

Check the box next to each of the below SQL queries, which are guaranteed to be executed by the DBMS (i.e., will never be rejected due to a constraint violation) on all legal instances of this schema, where each table is non-empty (assume all FKs are declared with the default "no action" option)?

DROP TABLE *AWARDS_EVENT*

What is the violation type here?

ACTOR (actor_name, gender, date_of_birth)
MOVIE (movie_title, release_year, genre, movie_length)
CAST_MEMBER (actor_name, movie_title, release_year, actor_role)
AWARDS_EVENT (event_name, event_year, venue)
NOMINATION (event_name, event_year, movie_title, release_year, category, won)

Check the box next to each of the below SQL queries, which are guaranteed to be executed by the DBMS (i.e., will never be rejected due to a constraint violation) on all legal instances of this schema, where each table is non-empty (assume all FKs are declared with the default "no action" option)?

UPDATE *NOMINATION*

SET category = 'Thriller'

WHERE category = 'Crime' OR category = 'Mystery'

ename	eyear	movie_title	release_year	category	won
Oscar	2020	M2	2020	Crime	Yes
Oscar	2020	M2	2020	Mystery	Yes

What is the violation type here?

ACTOR (actor_name, gender, date_of_birth)
MOVIE (movie_title, release_year, genre, movie_length)
CAST_MEMBER (actor_name, movie_title, release_year, actor_role)
AWARDS_EVENT (event_name, event_year, venue)
NOMINATION (event_name, event_year, movie_title, release_year, category, won)

Check the box next to each of the below SQL queries, which are guaranteed to be executed by the DBMS (i.e., will never be rejected due to a constraint violation) on all legal instances of this schema, where each table is non-empty (assume all FKs are declared with the default "no action" option)?

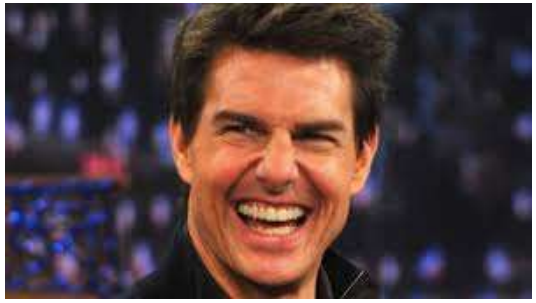
```
INSERT INTO CAST_MEMBER (actor_name, actor_role)
SELECT A.actor_name, "unlucky"
FROM Actor A
WHERE actor_name NOT IN (SELECT actor_name FROM CAST_MEMBER)
```

What is the violation type here?

ACTOR (actor_name, gender, date_of_birth)
MOVIE (movie_title, release_year, genre, movie_length)
CAST_MEMBER (actor_name, movie_title, release_year, actor_role)
AWARDS_EVENT (event_name, event_year, venue)
NOMINATION (event_name, event_year, movie_title, release_year, category, won)

Check the box next to each of the below SQL queries, which are guaranteed to be executed by the DBMS (i.e., will never be rejected due to a constraint violation) on all legal instances of this schema, where each table is non-empty (assume all FKs are declared with the default "no action" option)?

```
DELETE FROM AWARDS_EVENT A
WHERE NOT EXISTS  (SELECT *
                   FROM NOMINATION N
                   WHERE A. event_name = N.event_name
                         AND A.event_year = N.event_year)
```



Assume a Student table that includes 15 records. It has a non-null GPA column which is also UNIQUE. The output of the SQL statement

```
SELECT COUNT(*)
FROM Student
WHERE GPA > ANY (SELECT GPA FROM Student);
```

- STUDENT(sid, sname, byear, cgpa, bestSid)
- sid is Primary Key. Each student's best-friend's id is stored in the attribute bestSid, which references STUDENT(sid). It is guaranteed that a student cannot be the best friend of himself/herself in any instance of Student (i.e., such tuples never exist in any instance of the relation), but bestSid is allowed to be NULL. Assume a particular instance of Student relation has 19 tuples.

What is the maximum possible value that can be returned by the following query:

```
SELECT COUNT(*)
FROM Student S1, Student S2
WHERE S1.sid = S2.bestSid
```

- STUDENT(sid, sname, byear, cgpa, bestSid)
- sid is Primary Key. Each student's best-friend's id is stored in the attribute bestSid, which references STUDENT(sid). It is guaranteed that a student cannot be the best friend of himself/herself in any instance of Student (i.e., such tuples never exist in any instance of the relation), but bestSid is allowed to be NULL. Assume a particular instance of Student relation has 19 tuples.

What is the maximum possible value that can be returned by the following query:

```
SELECT MAX(T.c)
FROM  (SELECT COUNT(*) as C
        FROM Student S1, Student S2
        WHERE S1.sid = S2.bestSid
        GROUP BY S1.sid) as T
```

- STUDENT(sid, sname, byear, cgpa, bestSid)
- sid is Primary Key. Each student's best-friend's id is stored in the attribute bestSid, which references STUDENT(sid). It is guaranteed that a student cannot be the best friend of himself/herself in any instance of Student (i.e., such tuples never exist in any instance of the relation), but bestSid is allowed to be NULL. Assume a particular instance of Student relation has 19 tuples.

What is the maximum possible value that can be returned by the following query:

```
SELECT COUNT(*)
FROM Student S1, Student S2
WHERE S1.sid = S2.bestSid AND S2.sid = S1.bestSid
```

sid	bestSid
1	2
2	1
3	4
4	3
5	6
6	5
7	8
8	7
9	10
10	9
11	12
12	11
13	14
14	13
15	16
16	15
17	18
18	17
19	1

```
SELECT COUNT(*)
FROM Student S1, Student S2
WHERE S1.sid = S2.bestSid AND S2.sid = S1.bestSid
```

S1.sid	S1.bestSid	S2.sid	S2.bestSid
1	2	2	1
2	1	1	2
3	4	4	3
4	3	3	4
…			

STUDENT(sid, sname)
COURSE(cid, cname, cyear) where cyear ϵ {freshman, sophomore, junior, senior}
ENROLL(sid, cid, grade) grade is a number between 0 and 4,
ENROLL(sid) References STUDENT(sid)
ENROLL(cid) References COURSE(cid)

We want to find the students and their average grades for sophomore year courses only for the students whose average grades for sophomore year courses is higher than their average for freshman year courses.

We want to find the students and their average grades for sophomore year courses only for the students whose average grades for sophomore year courses is higher than their average for freshman year courses.

I- E2.sid = E.sid **II-** E3.sid = E.sid **III**- E2.sid = E3.sid

```
SELECT sid, AVG(grade)
FROM Enroll E, Course C
WHERE C.cid = E.cid AND C.cyear= 'sophomore'
GROUP BY E.sid
HAVING (SELECT AVG(grade)
        FROM Enroll E2, Course C2
        WHERE C2.cyear= 'sophomore' AND
              __FIRST__   AND C2.cid = E2.cid ) > (SELECT AVG(grade)
                                                  FROM Enroll E3, Course C3
                                                  WHERE C3.cyear= 'freshman' AND
                                                        __SECOND__ AND C3.cid = E3.cid)
```