CS 327E Class 8

March 25, 2019

Announcements

- Midterm is **next class** from 6pm - 7:30pm
- Midterm location: GEA 105
- Review session: Friday from 3pm - 5pm in GDC 1.304
- Milestone 8 due this Friday.

1) Does Q1 contain a subquery?

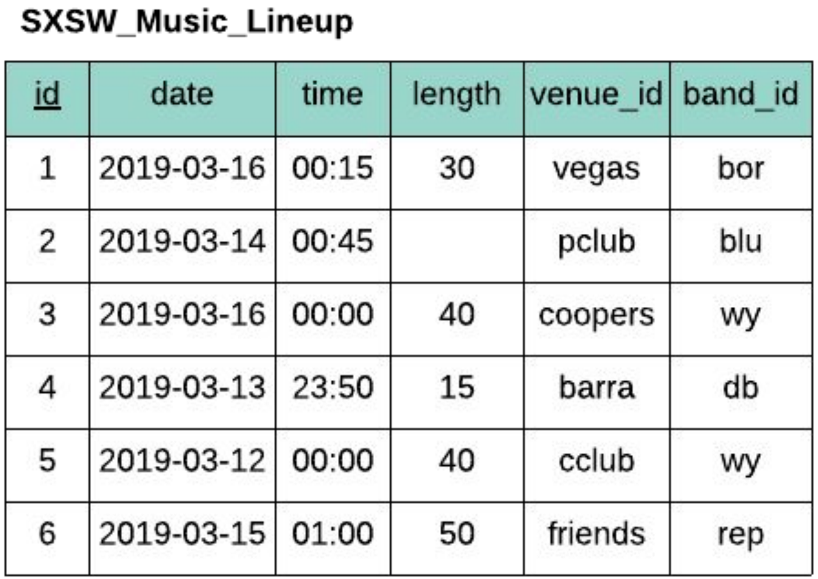
```
Q1: SELECT * FROM SXSW_Music_Lineup
    WHERE band_id = (SELECT id FROM Austin_Band
                     WHERE band_name = 'The Reputations');
```

A. Yes

B. No

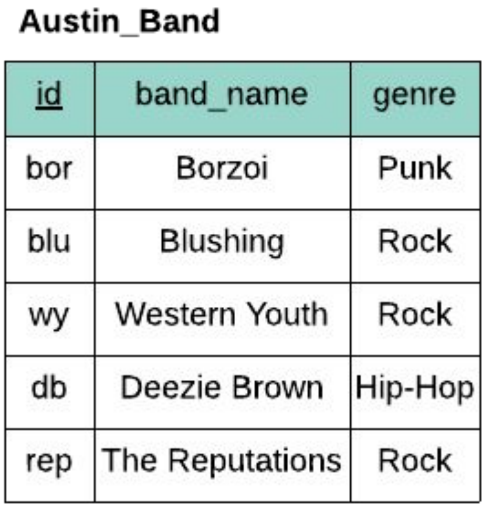
2) When run on the tables shown, what is the output from `Q2`’s **subquery**?

```
Q2: SELECT venue_id FROM SXSW_Music_Lineup WHERE band_id =
(SELECT id FROM Austin_Band WHERE band_name = 'Blushing')
```



SXSW_Music_Lineup

id	date	time	length	venue_id	band_id
1	2019-03-16	00:15	30	vegas	bor
2	2019-03-14	00:45		pclub	blu
3	2019-03-16	00:00	40	coopers	wy
4	2019-03-13	23:50	15	barra	db
5	2019-03-12	00:00	40	cclub	wy
6	2019-03-15	01:00	50	friends	rep



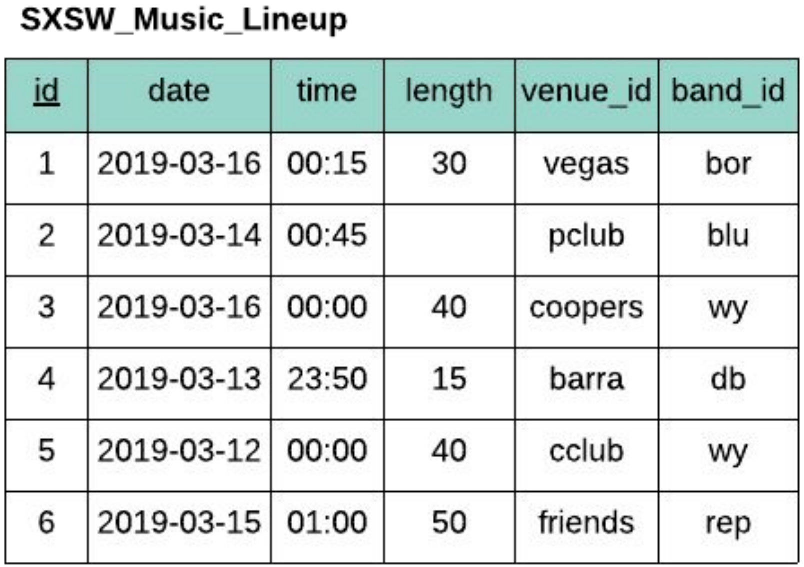
Austin_Band

id	band_name	genre
bor	Borzoi	Punk
blu	Blushing	Rock
wy	Western Youth	Rock
db	Deezie Brown	Hip-Hop
rep	The Reputations	Rock

A. 2
B. NULL
C. blu
D. pclub

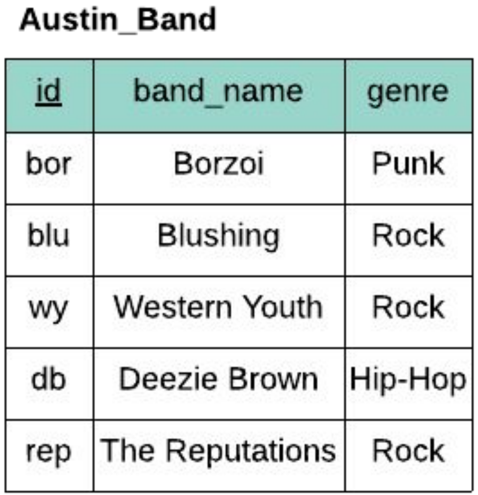
3) When run on the tables shown, how many rows does `Q3` return?

```
Q3: SELECT * FROM SXSW_Music_Lineup WHERE band_id =
(SELECT id FROM Austin_Band WHERE band_name = 'Western Youth')
```



SXSW_Music_Lineup

id	date	time	length	venue_id	band_id
1	2019-03-16	00:15	30	vegas	bor
2	2019-03-14	00:45		pclub	blu
3	2019-03-16	00:00	40	coopers	wy
4	2019-03-13	23:50	15	barra	db
5	2019-03-12	00:00	40	cclub	wy
6	2019-03-15	01:00	50	friends	rep



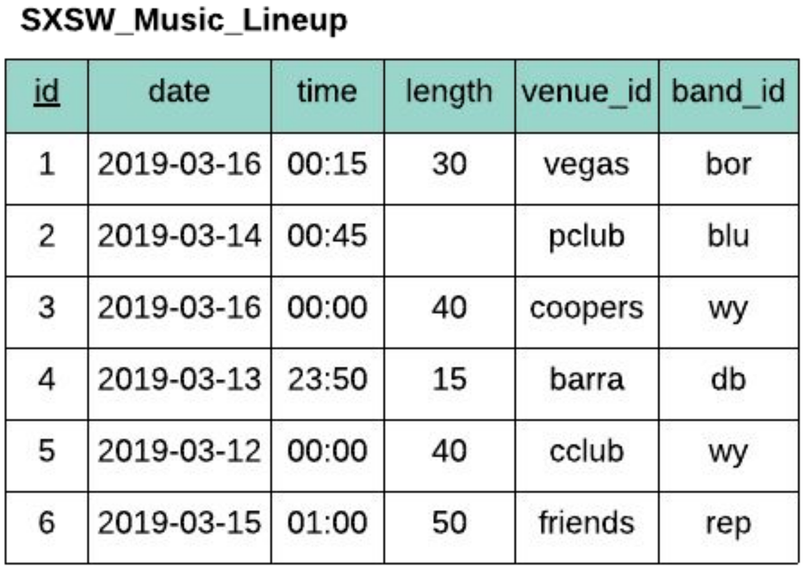
Austin_Band

id	band_name	genre
bor	Borzoi	Punk
blu	Blushing	Rock
wy	Western Youth	Rock
db	Deezie Brown	Hip-Hop
rep	The Reputations	Rock

A. 0
B. 1
C. 2
D. 3

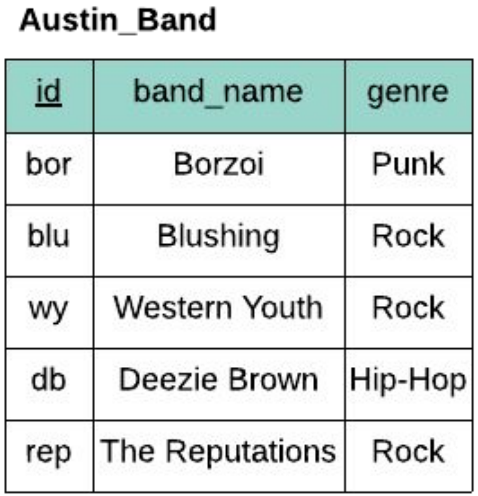
4) When run on the tables shown, what **input(s)** does the outer query receive in `Q4`?

```
Q4: SELECT * FROM SXSW_Music_Lineup WHERE band_id IN
(SELECT id FROM Austin_Band WHERE genre = 'Jazz')
```



SXSW_Music_Lineup

id	date	time	length	venue_id	band_id
1	2019-03-16	00:15	30	vegas	bor
2	2019-03-14	00:45		pclub	blu
3	2019-03-16	00:00	40	coopers	wy
4	2019-03-13	23:50	15	barra	db
5	2019-03-12	00:00	40	cclub	wy
6	2019-03-15	01:00	50	friends	rep



Austin_Band

id	band_name	genre
bor	Borzoi	Punk
blu	Blushing	Rock
wy	Western Youth	Rock
db	Deezie Brown	Hip-Hop
rep	The Reputations	Rock

A. 0
B. 1
C. NULL
D. 6

5) Given the table definitions below, the queries `Q5` and `Q6` are functionally equivalent.

```
SXSW_Music_Lineup(id, date, time, length, venue_id, band_id)
Austin_Band(id, band_name, genre)

Q5: SELECT id, date, time, length, venue_id
    FROM SXSW_Music_Lineup WHERE band_id IN
     (SELECT id
      FROM Austin_Band
      WHERE band_name = 'Deezie Brown')

Q6: SELECT l.id, l.date, l.time, l.length, l.venue_id
    FROM SXSW_Music_Lineup l
    JOIN Austin_Band b ON l.band_id = b.id
    WHERE b.band_name = 'Deezie Brown'
```

A. True
B. False

Syntax of Scalar Subqueries: `WHERE` clause

```
SELECT <list of desired fields>
FROM <single table>
WHERE <single field> =
      (SELECT <single value> FROM ...)
```

Comparison Operators: `=, !=, >, <, <=, >=`

Syntax of Scalar Subqueries: HAVING clause

```
SELECT <unaggregated fields> <aggregate functions>
FROM <single table>
WHERE <boolean condition>
GROUP BY <unaggregated field>
HAVING <aggregate function> = (SELECT <single value>
                               FROM ...)
```

Comparison Operators: = != > < <= >=

Syntax of List Subqueries: `WHERE` clause

```
SELECT <list of desired fields>
FROM <single table>
WHERE <single field> IN
      (SELECT <single field> FROM ...)
```

List Membership Operators:

`IN`

`NOT IN`

Syntax of Boolean Subqueries: `WHERE` clause

```
SELECT <list of desired fields>
FROM <single table>
WHERE EXISTS
      (SELECT * FROM … WHERE)
```

Existential Quantifiers:

```
EXISTS
NOT EXISTS
```

Syntax of List Subqueries: FROM clause

```
SELECT <list of desired fields>
FROM (SELECT <list of desired fields> FROM ...)
[WHERE]
[ORDER BY]
```

First Question

*Who does **not** take CS327E?*

Student(sid, fname, lname, dob)

Class(cno, cname, credits)

Teacher(tid, fname, lname, dept)

Takes(sid, cno, grade)

Teaches(tid, cno)

First Question

*Who does **not** take CS327E?*

Is this query a correct implementation?

```
SELECT sid
FROM Takes
WHERE cno != 'CS327E'
```

Student(sid, fname, lname, dob)

Class(cno, cname, credits)

Teacher(tid, fname, lname, dept)

Takes(sid, cno, grade)

Teaches(tid, cno)

Second Question

Who are the oldest students?

Student(sid, fname, lname, dob)

Class(cno, cname, credits)

Teacher(tid, fname, lname, dept)

Takes(sid, cno, grade)

Teaches(tid, cno)

iClicker Question

Who are the oldest students?

Does this query require a subquery?

A. Yes

B. No

Student(sid, fname, lname, dob)

Class(cno, cname, credits)

Teacher(tid, fname, lname, dept)

Takes(sid, cno, grade)

Teaches(tid, cno)

Third Question

*Who takes **only** CS313E?*

Student(sid, fname, lname, dob)

Class(cno, cname, credits)

Teacher(tid, fname, lname, dept)

Takes(sid, cno, grade)

Teaches(tid, cno)

iClicker Question

*Who takes **only** CS313E?*

Student(sid, fname, lname, dob)

Class(cno, cname, credits)

Teacher(tid, fname, lname, dept)

Takes(sid, cno, grade)

Teaches(tid, cno)

Does this query require a subquery?

A. Yes
B. No

Fourth Question

Which classes have a higher enrollment than the overall average enrollment per class?

Student(sid, fname, lname, dob)

Class(cno, cname, credits)

Teacher(tid, fname, lname, dept)

Takes(sid, cno, grade)

Teaches(tid, cno)

iClicker Question

Which classes have a higher enrollment than the overall average enrollment per class?

How many subqueries are in this query?

A. 0 B. 1 C. 2 D. 3

Student(sid, fname, lname, dob)

Class(cno, cname, credits)

Teacher(tid, fname, lname, dept)

Takes(sid, cno, grade)

Teaches(tid, cno)