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')

<table>
<thead>
<tr>
<th>SXSW_Music_Lineup</th>
<th>Austin_Band</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>date</td>
</tr>
<tr>
<td>---</td>
<td>----------</td>
</tr>
<tr>
<td>1</td>
<td>2019-03-16</td>
</tr>
<tr>
<td>2</td>
<td>2019-03-14</td>
</tr>
<tr>
<td>3</td>
<td>2019-03-16</td>
</tr>
<tr>
<td>4</td>
<td>2019-03-13</td>
</tr>
<tr>
<td>5</td>
<td>2019-03-12</td>
</tr>
<tr>
<td>6</td>
<td>2019-03-15</td>
</tr>
</tbody>
</table>

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')

<table>
<thead>
<tr>
<th>SXSW_Music_Lineup</th>
<th>Austin_Band</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>date</td>
</tr>
<tr>
<td>1</td>
<td>2019-03-16</td>
</tr>
<tr>
<td>2</td>
<td>2019-03-14</td>
</tr>
<tr>
<td>3</td>
<td>2019-03-16</td>
</tr>
<tr>
<td>4</td>
<td>2019-03-13</td>
</tr>
<tr>
<td>5</td>
<td>2019-03-12</td>
</tr>
<tr>
<td>6</td>
<td>2019-03-15</td>
</tr>
</tbody>
</table>

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')**

<table>
<thead>
<tr>
<th>SXSW_Music_Lineup</th>
<th>Austin_Band</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>date</td>
</tr>
<tr>
<td>1</td>
<td>2019-03-16</td>
</tr>
<tr>
<td>2</td>
<td>2019-03-14</td>
</tr>
<tr>
<td>3</td>
<td>2019-03-16</td>
</tr>
<tr>
<td>4</td>
<td>2019-03-13</td>
</tr>
<tr>
<td>5</td>
<td>2019-03-12</td>
</tr>
<tr>
<td>6</td>
<td>2019-03-15</td>
</tr>
</tbody>
</table>

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: \textbf{WHERE} clause

\begin{verbatim}
SELECT <list of desired fields>
FROM <single table>
WHERE <single field> =
  (SELECT <single value> FROM ...)
\end{verbatim}

Comparison Operators: $=, \neq, >, <, \leq, \geq$
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: \textbf{WHERE} clause

\begin{verbatim}
SELECT <list of desired fields>
FROM <single table>
WHERE \textbf{EXISTS}
     (SELECT * FROM ... WHERE)
\end{verbatim}

Existential Quantifiers:
\textbf{EXISTS}
\textbf{NOT EXISTS}
Syntax of List Subqueries: \texttt{FROM} clause

\begin{verbatim}
SELECT  \texttt{<list of desired fields>}
FROM (SELECT  \texttt{<list of desired fields>}  FROM \ldots)
[WHERE]
[ORDER BY]
\end{verbatim}
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'
```
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
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?

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?
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