1) Does Q1 contain a subquery?

Q1: SELECT * FROM ACL_Lineup
    WHERE artist_id = (SELECT id FROM ACL_Artist
                       WHERE artist_name = 'Khalid');

A. Yes
B. No
2) When run on the ACL tables shown, what is the output from the subquery in Q2?

Q2: SELECT * FROM ACL_Lineup WHERE artist_id = (SELECT id FROM ACL_Artist WHERE artist_name = 'Metallica');

A. mtc  
B. NULL  
C. 2, 5  
D. None of the above
3) When run on the **ACL** tables shown, how many rows does **Q3** produce?

Q3: SELECT * FROM ACL_Lineup WHERE artist_id = (SELECT id FROM ACL_Artist WHERE artist_name = 'Metallica');

<table>
<thead>
<tr>
<th>ACL_Lineup</th>
<th>ACL_Artist</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>id</td>
</tr>
<tr>
<td>1</td>
<td>pmc</td>
</tr>
<tr>
<td>2</td>
<td>kh</td>
</tr>
<tr>
<td>3</td>
<td>stv</td>
</tr>
<tr>
<td>4</td>
<td>mtc</td>
</tr>
<tr>
<td>5</td>
<td>nn</td>
</tr>
<tr>
<td>6</td>
<td>sor</td>
</tr>
<tr>
<td>7</td>
<td></td>
</tr>
</tbody>
</table>

**Table:**

<table>
<thead>
<tr>
<th>id</th>
<th>date</th>
<th>time</th>
<th>duration</th>
<th>stage_id</th>
<th>artist_id</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2018-10-05</td>
<td>19:45</td>
<td>135</td>
<td>amex</td>
<td>pmc</td>
</tr>
<tr>
<td>2</td>
<td>2018-10-13</td>
<td>20:00</td>
<td>120</td>
<td>amex</td>
<td>mtc</td>
</tr>
<tr>
<td>3</td>
<td>2018-10-05</td>
<td>17:35</td>
<td>60</td>
<td>honda</td>
<td>kh</td>
</tr>
<tr>
<td>4</td>
<td>2018-10-06</td>
<td>19:30</td>
<td>60</td>
<td>ml</td>
<td>stv</td>
</tr>
<tr>
<td>5</td>
<td>2018-10-06</td>
<td>20:00</td>
<td>120</td>
<td>amex</td>
<td>mtc</td>
</tr>
<tr>
<td>6</td>
<td>2018-10-05</td>
<td>14:45</td>
<td>60</td>
<td>ha</td>
<td>nn</td>
</tr>
<tr>
<td>7</td>
<td>2018-10-07</td>
<td>12:30</td>
<td>30</td>
<td>akl</td>
<td>sor</td>
</tr>
</tbody>
</table>

**Options:**

A. 0  
B. 1  
C. 2  
D. 3
4) When run on the ACL tables shown, what input does the outer query receive in Q4?

Q4: SELECT * FROM ACL_Lineup WHERE artist_id = (SELECT id FROM ACL_Artist WHERE artist_name = 'Beyoncé');

ACL_Lineup

<table>
<thead>
<tr>
<th>id</th>
<th>date</th>
<th>time</th>
<th>duration</th>
<th>stage_id</th>
<th>artist_id</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2018-10-05</td>
<td>19:45</td>
<td>135</td>
<td>amex</td>
<td>pmc</td>
</tr>
<tr>
<td>2</td>
<td>2018-10-13</td>
<td>20:00</td>
<td>120</td>
<td>amex</td>
<td>mtc</td>
</tr>
<tr>
<td>3</td>
<td>2018-10-05</td>
<td>17:35</td>
<td>60</td>
<td>honda</td>
<td>kh</td>
</tr>
<tr>
<td>4</td>
<td>2018-10-06</td>
<td>19:30</td>
<td>60</td>
<td>ml</td>
<td>stv</td>
</tr>
<tr>
<td>5</td>
<td>2018-10-06</td>
<td>20:00</td>
<td>120</td>
<td>amex</td>
<td>mtc</td>
</tr>
<tr>
<td>6</td>
<td>2018-10-05</td>
<td>14:45</td>
<td>60</td>
<td>ha</td>
<td>nn</td>
</tr>
<tr>
<td>7</td>
<td>2018-10-07</td>
<td>12:30</td>
<td>30</td>
<td>akl</td>
<td>sor</td>
</tr>
</tbody>
</table>

ACL_Artist

<table>
<thead>
<tr>
<th>id</th>
<th>artist_name</th>
</tr>
</thead>
<tbody>
<tr>
<td>pmc</td>
<td>Paul McCartney</td>
</tr>
<tr>
<td>kh</td>
<td>Khalid</td>
</tr>
<tr>
<td>stv</td>
<td>St. Vincent</td>
</tr>
<tr>
<td>mtc</td>
<td>Metallica</td>
</tr>
<tr>
<td>nn</td>
<td>Noname</td>
</tr>
<tr>
<td>sor</td>
<td>School of Rock</td>
</tr>
</tbody>
</table>

A. 0  
B. NULL  
C. None of the above
5) Given the table definitions below, the queries Q5 and Q6 are functionally equivalent.

ACL_Lineup(id, date, time, duration, stage_id, artist_id)
ACL_Artist(id, artist_name)

Q5: SELECT id, date, time, duration, stage_id
    FROM ACL_Lineup WHERE artist_id =
    (SELECT id
        FROM ACL_Artist
        WHERE artist_name = 'Paul McCartney');

Q6: SELECT l.id, l.date, l.time, l.duration, l.stage_id
    FROM ACL_Lineup l
    JOIN ACL_Artist a ON l.artist_id = a.id
    WHERE a.performer = 'Paul McCartney';

A. True  
B. False
Syntax of Subqueries: **WHERE** clause

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

Comparison Operators:  =  !=  >  <  <=  >=
Syntax of Subqueries: \textbf{WHERE} clause

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

\textbf{List Membership Operator:}
IN
NOT IN
Syntax of Subqueries: WHERE clause

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

Existential Quantifier:
- EXISTS
- NOT EXISTS
Syntax of Subqueries: FROM clause

SELECT <list of desired fields>
FROM (SELECT * FROM ...)
WHERE <boolean condition>
Syntax of Subqueries: HAVING clause

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

Comparison Operators:  =  !=  >  <  <=  >=
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 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
Third Question

*Who are the youngest students?*

Student(sid, fname, lname, dob)
Class(cno, cname, credits)
Teacher(tid, fname, lname, dept)
Takes(sid, cno, grade)
Teaches(tid, cno)
Fourth Question

*Which classes are taken by more students than the overall average number of students per class?*

Student(\textit{sid}, \textit{fname}, \textit{lname}, \textit{dob})
Class(\textit{cno}, \textit{cname}, \textit{credits})
Teacher(\textit{tid}, \textit{fname}, \textit{lname}, \textit{dept})
Takes(\textit{sid}, \textit{cno}, \textit{grade})
Teaches(\textit{tid}, \textit{cno})
iClicker Question

Which classes are taken by more students than the overall average number of students per class?

How many subqueries are contained 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)