Databases

Class Test 3

- This class test contains multiple-choice questions, which will have **negative marking** for wrong answers. If you randomly guess answers and they turn out to be wrong, you can get negative marks. To guard against that, please *show your work* as clearly as possible, on the back pages. If you make an honest mistake in obtaining your answer, you will be spared the negative marks.

- This is the third and final class test and contributes to the 20% continuous assessment mark.

- **Important notation:** When multiple columns are specified on the right hand side of a functional dependency, e.g., $ABC \rightarrow DE$, we mean that *each column* on the right hand side is functionally dependent, i.e., $ABC \rightarrow D$ and $ABC \rightarrow E$. 
### Interpretation

**Student** records the student details and current year average mark for each student. **Course** records the name and associated course identifier of each course. **Enroll** shows which student is currently enrolled on which course for which year.

1. Which of the following sets of attribute names is clearly **not** a superkey in the table **Student**, given only the specific instance of **Student** as shown above? (circle one only)  

   A. \{SID\}  
   B. \{Name, Age\}  
   C. \{SID, Username\}  
   D. \{Name, Username\}  
   E. \{Age, Mark\}  

   [10 points]

2. Which of the following queries are correct interpretations of this relational algebraic expression? (circle one only)  

   A. Find the names of courses where all the students enrolled on them in 2001 are aged 19 or 20.  
   B. Find the names of all students aged 19 or 20 enrolled on any course in year 2001.  
   C. Find the names of the courses, which have at least some students aged 19 or 20 enrolled on them in 2001.  
   D. Find the names of the courses, which do not have any students aged 19 or 20 enrolled on them in any year other than 2001.  
   E. Find the names of students, which the database records as being simultaneously aged 19 and 20 and enrolled on a course in 2001.  

   [20 points]

---

**Student**

<table>
<thead>
<tr>
<th>SID</th>
<th>Name</th>
<th>Username</th>
<th>Age</th>
<th>Mark</th>
</tr>
</thead>
<tbody>
<tr>
<td>600001</td>
<td>John</td>
<td>jab</td>
<td>19</td>
<td>85</td>
</tr>
<tr>
<td>612342</td>
<td>Mark</td>
<td>mbc</td>
<td>20</td>
<td>64</td>
</tr>
<tr>
<td>674832</td>
<td>Jane</td>
<td>jcd</td>
<td>19</td>
<td>88</td>
</tr>
<tr>
<td>693757</td>
<td>Sally</td>
<td>spq</td>
<td>25</td>
<td>68</td>
</tr>
<tr>
<td>623243</td>
<td>Mark</td>
<td>mll</td>
<td>20</td>
<td>79</td>
</tr>
</tbody>
</table>

**Course**

<table>
<thead>
<tr>
<th>Name</th>
<th>CID</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS</td>
<td>101</td>
</tr>
<tr>
<td>Physics</td>
<td>102</td>
</tr>
<tr>
<td>Maths</td>
<td>103</td>
</tr>
<tr>
<td>Chemistry</td>
<td>104</td>
</tr>
<tr>
<td>Biology</td>
<td>105</td>
</tr>
</tbody>
</table>

**Enroll**

<table>
<thead>
<tr>
<th>SID</th>
<th>CID</th>
<th>Year</th>
</tr>
</thead>
<tbody>
<tr>
<td>600001</td>
<td>101</td>
<td>2001</td>
</tr>
<tr>
<td>612342</td>
<td>103</td>
<td>2002</td>
</tr>
<tr>
<td>674832</td>
<td>102</td>
<td>2001</td>
</tr>
<tr>
<td>693757</td>
<td>102</td>
<td>2000</td>
</tr>
<tr>
<td>623243</td>
<td>104</td>
<td>2001</td>
</tr>
</tbody>
</table>
3. Which of the following paragraphs about data redundancy in a database is **TRUE?** Circle all correct answers.  

[10 points]

A. It is a good idea to design your tables so that they store the same information multiple times. If there is a disk crash, it is more likely that at least one copy of the information survives.

B. Storing the same information redundantly can be done on purpose to speed up certain queries.

C. A relational database management system is likely to crash when information is stored redundantly in it.

D. Storing information redundantly is really not a problem at all. Disk space has grown so cheap that, unless the redundancy is at ridiculously high levels, we have plenty of disk space to handle it.

E. A database designer always has a choice when building a correctly working database application:
   - Remove redundancies and spend more work in developing the system but make future easier to implement, or
   - Leave the redundancies there, reduce the initial development effort but make future changes harder to implement.

4. Which of the functional dependencies below is **not** satisfied by the table? **Circle one only.**  

[10 points]

<table>
<thead>
<tr>
<th>V</th>
<th>W</th>
<th>X</th>
<th>Y</th>
<th>Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>a</td>
<td>x</td>
<td>15</td>
<td>e</td>
</tr>
<tr>
<td>2</td>
<td>a</td>
<td>x</td>
<td>10</td>
<td>f</td>
</tr>
<tr>
<td>2</td>
<td>b</td>
<td>x</td>
<td>15</td>
<td>g</td>
</tr>
<tr>
<td>2</td>
<td>b</td>
<td>y</td>
<td>15</td>
<td>h</td>
</tr>
</tbody>
</table>

A. XYZ → X  
B. WY → V  
C. XY → V  
D. VW → Y  
E. VZ → X
5. Consider the functional dependencies
\[\{AD \rightarrow B, AD \rightarrow C, C \rightarrow A, C \rightarrow E, BE \rightarrow F, EF \rightarrow B\}\]
Which of the following functional dependencies follows from the above dependencies (on the basis of the Armstrong axioms or by use of the closure algorithm)? Circle all correct answers. [20 points]

A. CB \rightarrow D
B. CF \rightarrow D
C. BDE \rightarrow C
D. BF \rightarrow E
E. CD \rightarrow F

6. Consider the dependencies ABC \rightarrow DEF and CDE \rightarrow A, for the schema (A, B, C, D, E, F).
Which of the following decompositions of the schema is in Boyce-Codd normal form? Circle all correct answers. [20 points]

A. (A, B, C, D, E, F)
B. (A, B, C) \& (D, E, F)
C. (A, B, C) \& (C, D, E, F, A)
D. (B, C, D, E, F) \& (C, D, E, A)
7. Consider the functional dependencies
   \[ ABC \rightarrow DEF, AC \rightarrow F, BD \rightarrow A, C \rightarrow DF \text{ and } EF \rightarrow B. \]
   Which of the following decompositions of the schema \((A, B, C, D, E, F)\) are *lossless*? Circle \textbf{all} correct answers.

\[ \text{[20 points]} \]
A. \(\{A, B, C, D, F\} \text{ and } \{B, C, E\}\)
B. \(\{B, C, D, E, F\} \text{ and } \{A, E, F\}\)
C. \(\{A, B, C, E, F\} \text{ and } \{C, D\}\)
D. \(\{B, C, D, E, F\} \text{ and } \{A, B, C, D\}\)
E. \(\{A, C, D, E, F\} \text{ and } \{C, E, B\}\)

8. Which of the following decompositions of the schema \((A, B, C, D, E, F)\) are *dependency-preserving* for the given functional dependencies? Circle \textbf{one} correct answer.

\[ \text{[10 points]} \]
A. \(\{A, B, C, D, F\} \text{ and } \{B, C, E\}: \text{ given } ABC \rightarrow DEF, BC \rightarrow E \text{ and } C \rightarrow B\)
B. \(\{A, B, C, D, F\} \text{ and } \{B, C, D, E\}: \text{ given } ABC \rightarrow DEF, BC \rightarrow EF \text{ and } EF \rightarrow B\)
C. \(\{A, C, D, E, F\} \text{ and } \{B, C, D, E\}: \text{ given } AB \rightarrow F, BC \rightarrow DE \text{ and } DC \rightarrow B\)
D. \(\{A, B, C, D, F\} \text{ and } \{B, E, F\}: \text{ given } ABC \rightarrow DEF, BC \rightarrow F \text{ and } EF \rightarrow B\)
E. \(\{A, B, C, D, F\} \text{ and } \{B, C, D, E\}: \text{ given } AB \rightarrow CDE, DE \rightarrow CF \text{ and } C \rightarrow E\)