Consider the following instance of a relational table Student to be used for keeping track of a variety of information about students and instruments at the School of Rock:

<table>
<thead>
<tr>
<th>sid</th>
<th>name</th>
<th>age</th>
<th>instid</th>
<th>kind</th>
<th>key</th>
<th>rating</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Fran</td>
<td>15</td>
<td>106</td>
<td>guitar</td>
<td>C</td>
<td>7.5</td>
</tr>
<tr>
<td>1</td>
<td>Fran</td>
<td>15</td>
<td>85</td>
<td>trumpet</td>
<td>Bb</td>
<td>4.0</td>
</tr>
<tr>
<td>2</td>
<td>Alex</td>
<td>17</td>
<td>68</td>
<td>bass</td>
<td>C</td>
<td>8.5</td>
</tr>
<tr>
<td>2</td>
<td>Alex</td>
<td>17</td>
<td>70</td>
<td>bass</td>
<td>C</td>
<td>8.5</td>
</tr>
<tr>
<td>3</td>
<td>Janice</td>
<td>16</td>
<td>44</td>
<td>alto sax</td>
<td>Eb</td>
<td>8.0</td>
</tr>
</tbody>
</table>

1. (5 pts) Talking to the application designers has revealed that the following FDs (functional dependencies) hold in their universe:
   i. sid → name
   ii. sid → age
   iii. (sid, kind) → rating
   iv. instid → kind
   v. kind → key

Given the schema and associated FDs, indicate whether each of the following new tuples would be legal or illegal (violating one or more FDs) if an attempt was made to insert it into the table instance above. In the illegal case, also indicate the violated FD or FDs:

a) INSERT INTO Student VALUES (2, “Alex”, 17, 72, “bass”, “C”, 9.0);
   **LEGAL**  **ILLEGAL**  **Violated FDs: 3**

b) INSERT INTO Student VALUES (1, “Fran”, 17, 102, “alto sax”, “Eb”, 5.0);
   **LEGAL**  **ILLEGAL**  **Violated FDs: 2**

c) INSERT INTO Student VALUES (3, “Janice”, 16, 63, “trumpet”, “C”, 6.0);
   **LEGAL**  **ILLEGAL**  **Violated FDs: 5**

   **LEGAL**  **ILLEGAL**  **Violated FDs:**

e) INSERT INTO Student VALUES (3, “Janice”, 16, 68, “bass”, “C”, 2.0)
   **LEGAL**  **ILLEGAL**  **Violated FDs:**
2. (4 pts) Here again are the FDs for Student(sid, name, age, instid, kind, key, rating):
   i. sid → name
   ii. sid → age
   iii. (sid, kind) → rating
   iv. instid → kind
   v. kind → key

Below are some additional proposed FDs, some of which are right – i.e., they are implied by the FD set above – and some of which are not right. Circle the FDs below that are indeed implied by the FDs above:
   i. (sid, key) → name
   ii. name → age
   iii. (sid, key) → rating
   iv. instid → key

3. (1 pt) List the candidate keys for Student.
   (sid, instid)

4. (for discussion) Identify the (highest) normal form that the current Student table design is in and then decompose it into a set of tables that are in BCNF (Boyce-Codd Normal Form) and that preserve the given FDs.
   It’s in 1NF. FD1 and FD2 violates the 2NF condition.
   1) Deal with FD1 and FD2
      R1(sid, instid, kind, key, rating)
      R2(sid, name, age)
   2) Deal with FD3
      R1(sid, instid, kind, key)
      R2(sid, name, age)
      R3(sid, kind, rating)
   3) Deal with FD4 and FD5
      R1(sid, instid) -- info about actual student-instrument connections (for lossless join!)
      R2(sid, name, age) -- info about students
      R3(sid, kind, rating) -- info about student’s playing abilities
      R4(instid, kind) -- info about specific instruments
      R5(kind, key) -- info about kinds of instruments