Quiz 3: Relational DB Design Theory

Consider the following instance of a relational table Musician to be used for keeping track of a variety of information about musicians and instruments:

<table>
<thead>
<tr>
<th>mno</th>
<th>name</th>
<th>age</th>
<th>instid</th>
<th>type</th>
<th>key</th>
<th>skill</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Fred</td>
<td>24</td>
<td>106</td>
<td>guitar</td>
<td>C</td>
<td>7.5</td>
</tr>
<tr>
<td>1</td>
<td>Fred</td>
<td>24</td>
<td>85</td>
<td>trumpet</td>
<td>Bb</td>
<td>4.0</td>
</tr>
<tr>
<td>2</td>
<td>Sally</td>
<td>17</td>
<td>68</td>
<td>guitar</td>
<td>C</td>
<td>8.5</td>
</tr>
<tr>
<td>2</td>
<td>Sally</td>
<td>17</td>
<td>70</td>
<td>guitar</td>
<td>C</td>
<td>8.5</td>
</tr>
<tr>
<td>3</td>
<td>Felix</td>
<td>32</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. mno → name
   ii. mno → age
   iii. (mno, type) → skill
   iv. instid → type
   v. type → 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 Musician VALUES (2, “Sally”, 17, 72, “guitar”, “C”, 9.0);
      LEGAL    ILLEGAL    Violated FDs: mno, type → skill
   b) INSERT INTO Musician VALUES (1, “Fred”, 32, 102, “alto sax”, “Eb”, 5.0);
      LEGAL    ILLEGAL    Violated FDs: mno → age
   c) INSERT INTO Musician VALUES (3, “Felix”, 32, 63, “trumpet”, “C”, 6.0);
      LEGAL    ILLEGAL    Violated FDs: type → key
      LEGAL    ILLEGAL    Violated FDs:
e)  INSERT INTO Musician VALUES (3, “Felix”, 32, 68, “guitar”, “C”, 2.0)

LEGAL    ILLEGAL    Violated FDs:

2. (4 pts) Here again are the FDs for Musician(mno, name, age, instid, type, key, skill):
   i.  mno → name
   ii. mno → age
   iii. (mno, type) → skill
   iv.  instid → type
   v.   type → 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. Circle the FDs below that are implied by the FDs above:
   i.  (mno, key) → name
   ii. name → age
   iii. instid → key
   iv.  (mno, key) → skill

3. (1 pt) List the candidate keys for Musician.

(mno, instid)

4. (for discussion) Identify the (highest) normal form that the current Musician 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 currently in 1NF but not 2NF or beyond
R1(mno, name, age) -- info about musicians
R2: mno, type, skill -- info about musician’s playing abilities
R3(instid, type) -- info about specific instruments
R4(type, key) -- info about types of instruments
R5(mno, instid) -- info about actual musician-instrument connections (for lossless join!)