Database Normalization – Exercise Practices on BCNF

Suppose a following relational schema R:

<table>
<thead>
<tr>
<th>stu_id</th>
<th>subject</th>
<th>prof</th>
<th>prof_id</th>
</tr>
</thead>
<tbody>
<tr>
<td>s101</td>
<td>Python</td>
<td>Sumit Sharma</td>
<td>p1</td>
</tr>
<tr>
<td>s101</td>
<td>Java</td>
<td>R. Chauhan</td>
<td>p2</td>
</tr>
<tr>
<td>s102</td>
<td>Python</td>
<td>Punit</td>
<td>p3</td>
</tr>
<tr>
<td>s103</td>
<td>C#</td>
<td>Simmi</td>
<td>p4</td>
</tr>
<tr>
<td>s104</td>
<td>Python</td>
<td>Sumit Sharma</td>
<td>p1</td>
</tr>
</tbody>
</table>

- One student can enrol multiple subjects.
- For each subject, a professor is assigned.
- There can be multiple professor teaching same subjects.

**Identify functional dependency in above relation and check out this table is in BCNF or not? if not, decompose it in BCNF.**

**Solution:**

Based on the descriptions and value given, following functional dependencies have been identified:

prof_id → prof, subject

stud_id, subject → prof, prof_id
There are two candidate key in above table i.e.

1. (stud_id, subject)
2. (stud_id, prof_id)

(stud_id subject)⁺ = stud_id, subject, prof, prof_id
(stud_id prof_id)⁺ = stud_id, subject, prof, prof_id

Closure of these two has all the attributes of R.

Prime attributes: stud_id, subject, prof_id
Non prime attributes: prof

The functional dependency prof_id → prof, subject is not following rule of BCNF because prof_id is not candidate key.

Therefore table needs to be decomposed into:

R1 (prof_id, prof, subject)
R2 (stu_id, prof_id)

Now R1, R2 are normalized into BCNF.

See how data redundancy has been removed by decomposing it into BCNF.
Exercise:
Suppose R (A B C D E F) and set of FDs

\[ F : \{ A \rightarrow BCD, \]
\[ BC \rightarrow DE, \]
\[ B \rightarrow D, \]
\[ D \rightarrow A \} \]

Do following:
1. Compute \( B^+ \).
2. Find candidate key.
3. Compute canonical cover.
4. Give 3NF decomposition
5. Give BCNF decomposition.