Database Normalization – Exercise Practice on 3NF

Suppose a following relations schema:

Employee

<table>
<thead>
<tr>
<th>emp_id</th>
<th>emp_name</th>
<th>emp_zip</th>
<th>dept_id</th>
<th>dept_name</th>
<th>dept_location</th>
</tr>
</thead>
<tbody>
<tr>
<td>e101</td>
<td>Rakesh Kumar</td>
<td>273010</td>
<td>d001</td>
<td>IT</td>
<td>GF-20</td>
</tr>
<tr>
<td>e102</td>
<td>Rakesh</td>
<td>211008</td>
<td>d001</td>
<td>IT</td>
<td>GF-20</td>
</tr>
<tr>
<td>e103</td>
<td>Shyam Verma</td>
<td>345001</td>
<td>d003</td>
<td>Electronics</td>
<td>GF-22</td>
</tr>
<tr>
<td>e104</td>
<td>Saloni Sharma</td>
<td>239801</td>
<td>d004</td>
<td>Accounts</td>
<td>FF-21</td>
</tr>
<tr>
<td>e106</td>
<td>Shyama</td>
<td>870401</td>
<td>d005</td>
<td>Admin</td>
<td>GF-204</td>
</tr>
</tbody>
</table>

Identify functional dependencies in above table and find out that relation is in 3NF or not?

If not decompose it in 3NF.

(A department can have many employees, but an employee is associated with one department).

Solution:

The following FDs are identified based on the value and descriptions given about table:

- \( \text{emp}_\text{id} \rightarrow \text{emp}_\text{name}, \text{emp}_\text{zip}, \text{dept}_\text{id} \)
- \( \text{dept}_\text{id} \rightarrow \text{dept}_\text{name}, \text{dept}_\text{location} \)

Since emp_id uniquely identifies each record in the table, and the closure of emp_id also has all the attributes of table.

So, emp_id is the candidate key in table.

Prime attributes – \{emp_id\}
Non prime attributes – {emp_name, emp_zip, dept_id, dept_name, dept_location}

The table is in 1NF because each attribute in the table have atomic (single) value.
The table is also in 2NF because it has not any partial dependency. ( no non prime attribute is dependent on the proper subset of candidate key)
But, the table is not following the rules of 3NF because of functional dependency
  dept_id → dept_name, dept_location
[ Non prime attribute {dept_id} is determining another non prime attributes {dept_name, dept_location}, it is the case of transitive dependency.]
Therefore, to normalize above table in 3NF, it needs to be divided into two tables:
  employee (emp_id, emp_name, dept_id)
  department (dept_id, dept_name, dept_location)

Now both tables employee and department are following the rules of 3NF.

Exercise:
1. Suppose R (A B C) and set of FDs
   F : { A → B, B → C}

   The above table is normalized up to 3NF or not? If not, decompose it in 3NF.