

Normalization of Figure 1

UNF

Employee (empld, employeeName, dateOfBirth, contact, {email}, {address}, deptId, department)

1NF

Repeating groups are removed.

Employee-1 (empld, employeeName, dateOfBirth, contact, deptId, department)

Email-1 (emailId, empld*, email)

Address-1 (addressId, empld*, address)

2NF

Checking partial functional dependency in Employee-1 table:

There is no composite key in Employee-1 table so, it is in 2NF.

Checking partial functional dependency in Email-1 table:

There is no composite key in Email-1 table so, it is in 2NF.

Checking partial functional dependency in Address-1 table:

addressId → address

addressId, empld → X

Tables in 2NF:

Employee-2 (empld, employeeName, dateOfBirth, contact, deptId, department)

Email-2 (emailId, empld*, email)

Address-2 (addressId, address)

Employee_address-2 (empld*, addressId*)

3NF

Checking transitive dependency in Employee-2 table:

empld → deptld → department

Checking transitive dependency in Email-2 table:

Table Email-2 is already in 3NF because there does not exist a transitive dependency.

Checking transitive dependency in Address-2 table:

Table Address-2 is already in 3NF because there does not exist a transitive dependency.

Checking transitive dependency in Employee_address-2 table:

Table Employee_address-2 is already in 3NF because there does not exist a transitive dependency.

Tables in 3NF:

Employee-3 (empld, employeeName, dateOfBirth, contact, deptld*)

Email-3 (emailld, empld*, email)

Address-3 (addressld, address)

Employee_address-3 (empld*, addressld*)

Department-3 (deptld, department)

Assumptions:

- One email belongs to only one employee.
- One employee can have multiple address.
- One address can have multiple employees.

Normalization of Figure 2

UNF

Voter (voterId, voterName, {votingYear, {votingMonth, candidateId, candidateName, deptId, candidateDepartment}})

1NF

Voter-1 (voterId, voterName)

VoteYear-1 (yearId, votingYear, voterId*, vote-year-id) [vote-year-id → surrogate key]

VoteMonth-1 (monthId, votingMonth, candidateId, candidateName, deptId, candidateDepartment, vote-year-id*)

2NF

There is no composite key in the table in 1NF so, all the tables of 1NF are in 2NF.

Voter-2 (voterId, voterName)

VoteYear-2 (yearId, votingYear, voterId*, vote-year-id) [vote-year-id → surrogate key]

VoteMonth-2 (monthId, votingMonth, candidateId, candidateName, deptId, candidateDepartment, vote-year-id*)

3NF

Checking transitive dependency in VoteYear-2 table:

vote-year-id → yearId → votingYear

Checking transitive dependency in VoteMonth-2 table:

monthId → candidateId → candidateName, deptId, candidateDepartment

candidateId → deptId → candidateDepartment

Tables in 3NF:

Voter-3 (voterId, voterName)

VoteYear-3 (yearId*, voterId*, vote-year-id)

Year-3 (yearId, votingYear)

VoteMonth-3 (monthId, votingMonth, candidatId*, vote-year-id*)

Candidate-3 (candidatId, candidateName, deptId*)

Department-3 (deptId, candidateDepartment)

Assumptions:

- In month of voting, there are multiple candidates.