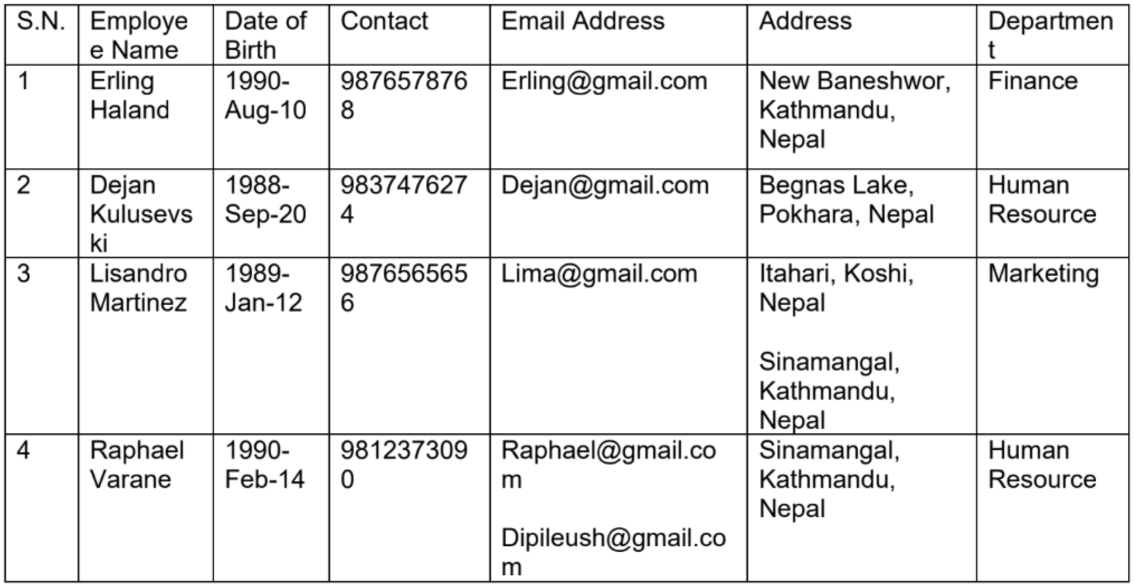
1 Normalization

1.1 Employee Table Normalization



S.N.	Employee Name	Date of Birth	Contact	Email Address	Address	Department
1	Erling Haland	1990-Aug-10	9876578768	Erling@gmail.com	New Baneshwor, Kathmandu, Nepal	Finance
2	Dejan Kulusevski	1988-Sep-20	9837476274	Dejan@gmail.com	Begnas Lake, Pokhara, Nepal	Human Resource
3	Lisandro Martinez	1989-Jan-12	9876565656	Lima@gmail.com	Itahari, Koshi, Nepal Sinamangal, Kathmandu, Nepal	Marketing
4	Raphael Varane	1990-Feb-14	9812373090	Raphael@gmail.com Dipileush@gmail.com	Sinamangal, Kathmandu, Nepal	Human Resource

❖ **UNF**

➔ In the unnormalized form all the table attributes are wrapped inside a single bracket. And the repeating group and repeating data are distinguished. Let us assume that a single employee is only assigned to one department. In this employee table the repeating data and group are:

The instance of email address repeats all the data and address also the instance of address repeats all the other attributes. So, the Final UNF of the table is:

Employee (S.N., Employee Name, Contact, Date of Birth, Department, {Email Address}, {Address})

❖ **1NF**

➔ In the first normal form the repeating data and repeating group are separated with the reference to the parent table. The primary key of the parent table references as a foreign key in the separated table. The tables that are created are listed below:

Employee	(S.N., Employee Name, Date of Birth, contact, Department)
Email	(Email Address, S.N.*)
Address	(Address Id, Address, S.N.*)

New attribute Address Id is added as a primary key in the address table to reduce the data redundancy.

❖ 2NF

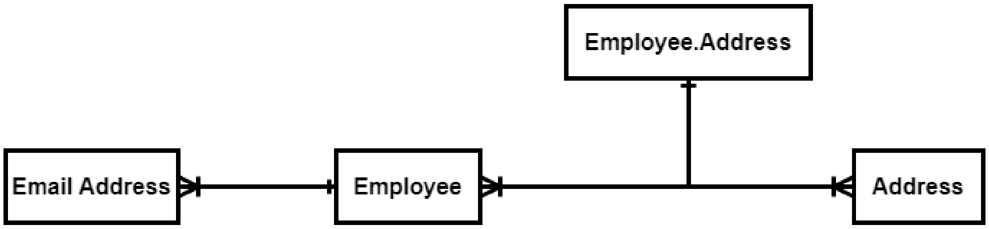
➔ Since there are no composite primary key in the table formed after 1NF so there are no partial dependencies. Because for partial dependency there must be a composite primary key where at least one key must partially give the data.

❖ 3NF

Since no non key attribute in the table of 2NF gives another non key value so there doesn't exist any transitive dependency so the final table looks like:

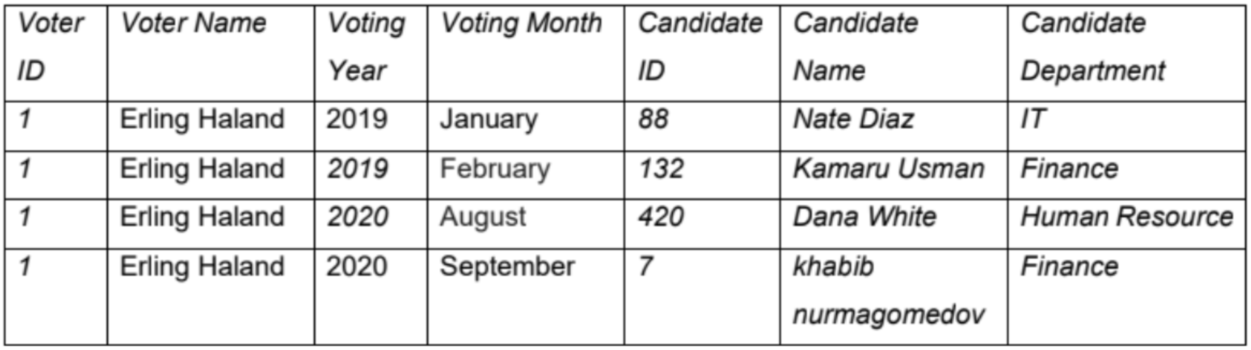
Employee	(S.N., Employee Name, Date of Birth, contact, Department)
Email	(Email Address, S.N.*)
Address	(Address Id, Address)
Address.Employee	(Address Id*, S.N.*)

Where the relation of the tables is listed below:



Finally, one employee can have one or multiple email address and one address can have multiple employee also one employee ca can multiple address.

1.2 Voting Record Table



Voter ID	*Voter Name*	*Voting Year*	*Voting Month*	*Candidate ID*	*Candidate Name*	*Candidate Department*
1	Erling Haland	2019	January	*88*	*Nate Diaz*	*IT*
1	Erling Haland	*2019*	February	*132*	*Kamaru Usman*	*Finance*
1	Erling Haland	*2020*	August	*420*	*Dana White*	*Human Resource*
1	Erling Haland	2020	September	*7*	*khabib nurmagomedov*	*Finance*

❖ **UNF**

➔ In the unnormalized form all the table attributes are wrapped inside a single bracket. And the repeating group and repeating data are distinguished. In this voter record table the repeating data and group are:

Repeating Data	Voter Id Voter Name For Other remaining instances year is repeating group
Repeating Group	Voting Month, Candidate ID, Candidate Name, Candidate Department.

The instance of voting year is repeated for every month of year. So, the final UNF table is listed below:

Voter (Voter ID, Voter Name, {Voting Year, {Voting Month, Candidate ID, Candidate Name, Candidate Department}})

❖ **1NF**

➔ In the first normal form the repeating data and repeating group are separated with the reference to the parent table. The primary key of the parent table references as a foreign key in the separated table. The tables that are created are listed below:

Voter (Voter ID, Voter Name)

Vote Year (Voting Year, Voter ID*)

Record (Voting Month, Candidate ID, Candidate Name, Candidate Department, (Voter ID, Voting Year)*)

New attribute Address Id is added as a primary key in the address table to reduce the data redundancy.

❖ **2NF**

➔ Since partial dependency in the table of 1NF so the table doesn't change.

❖ **3NF**

➔ There exists a transitive dependency in the Record table which is resolved below:

- In Record Table

➢ Voting Month, Voter ID, Voting Year → Candidate ID → Candidate Name, Candidate Department

So, the new table are:

Voter	(Voter ID, Voter Name)
Vote Year	(Voting Year, Voter ID*)
Record	(Voting Month, Candidate ID*, (Voter ID, Voting Year)*)
Candidate	(Candidate ID, Candidate Name, Candidate Department)