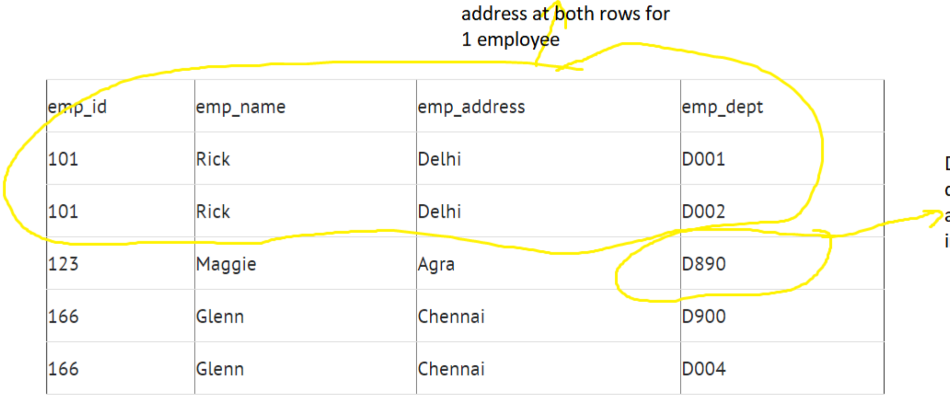
Normalization - Normalization is a process of organizing the data in database to avoid data redundancy, insertion anomaly, update anomaly & deletion anomaly.

Anomalies-

Update anomaly because we need to update address at both rows for 1 employee



emp_id	emp_name	emp_address	emp_dept
101	Rick	Delhi	D001
101	Rick	Delhi	D002
123	Maggie	Agra	D890
166	Glenn	Chennai	D900
166	Glenn	Chennai	D004

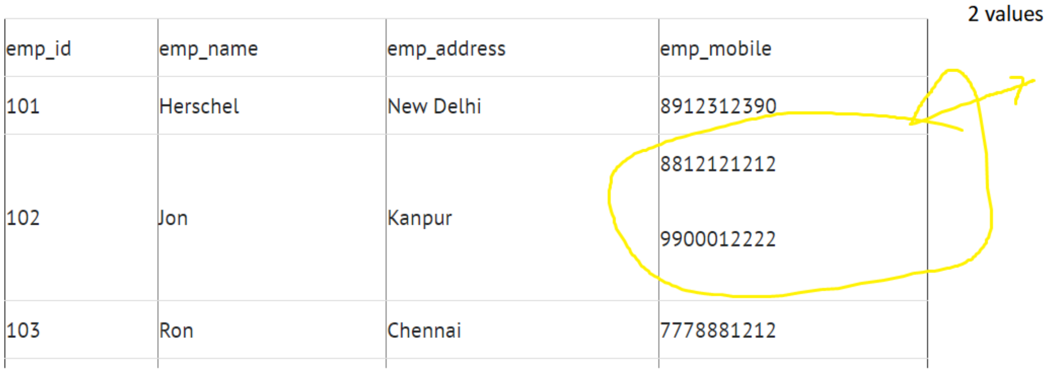
Delete anomaly bacause deletion of department also deletes the employee informations

Insert anomaly bacause not any department id is null

Types of Normalization – 4 types-

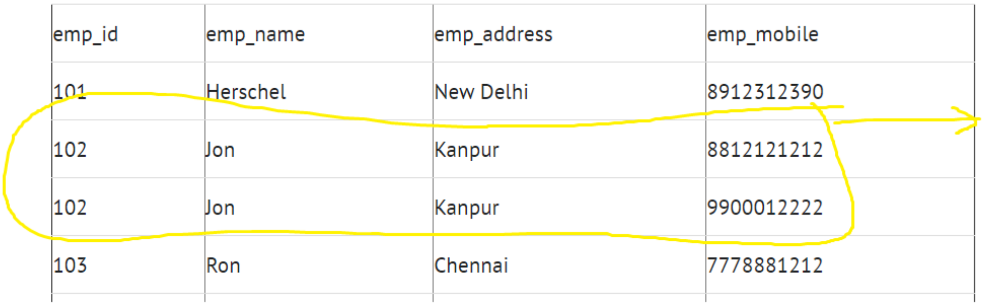
1. 1NF
2. 2NF
3. 3NF
4. BCNF (Boyce and Codd Normal Form)

1NF – 1NF says that any column of the table can’t store 2 values.



emp_id	emp_name	emp_address	emp_mobile
101	Herschel	New Delhi	8912312390
102	Jon	Kanpur	8812121212 9900012222
103	Ron	Chennai	7778881212

2 values so 1NF



emp_id	emp_name	emp_address	emp_mobile
101	Herschel	New Delhi	8912312390
102	Jon	Kanpur	8812121212
102	Jon	Kanpur	9900012222
103	Ron	Chennai	7778881212

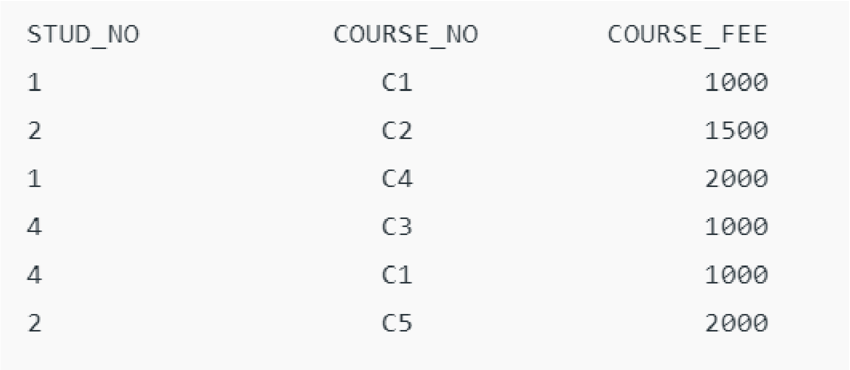
We can remove this by divide it into two rows

2NF – It has 2 conditions-

- Table must be in 1NF
- No partial dependency in the table columns

Attribute that is not part of any candidate key is the non-prime attribute.

For example we have a table as-



STUD_NO	COURSE_NO	COURSE_FEE
1	C1	1000
2	C2	1500
1	C4	2000
4	C3	1000
4	C1	1000
2	C5	2000

Here we have a relation as- (STUD_NO and COURSE_NO) -> COURSE_FEE

We can't determine the value in the column without using 2 candidate keys.

So this is called 2NF normal form. And we need to remove it.

We can remove it by divide the tables into 2 parts. So the dependency is end between the columns.

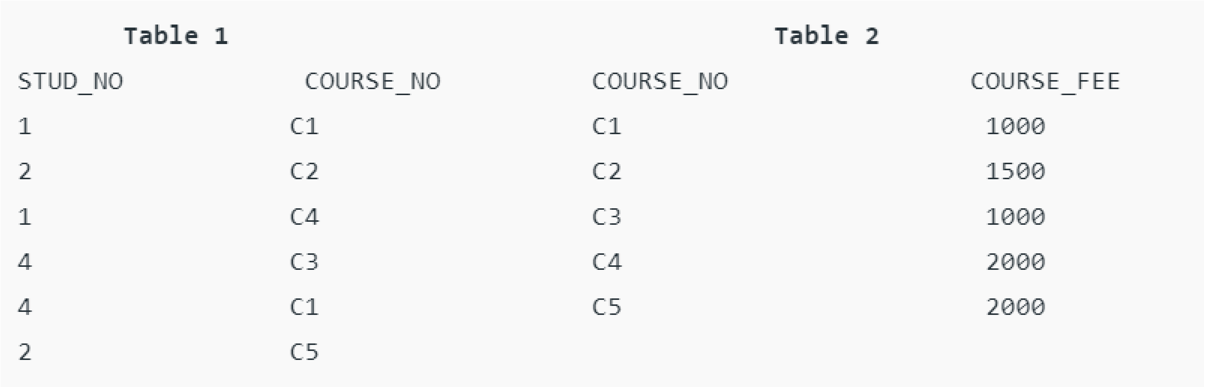


Table 1

STUD_NO	COURSE_NO
1	C1
2	C2
1	C4
4	C3
4	C1
2	C5

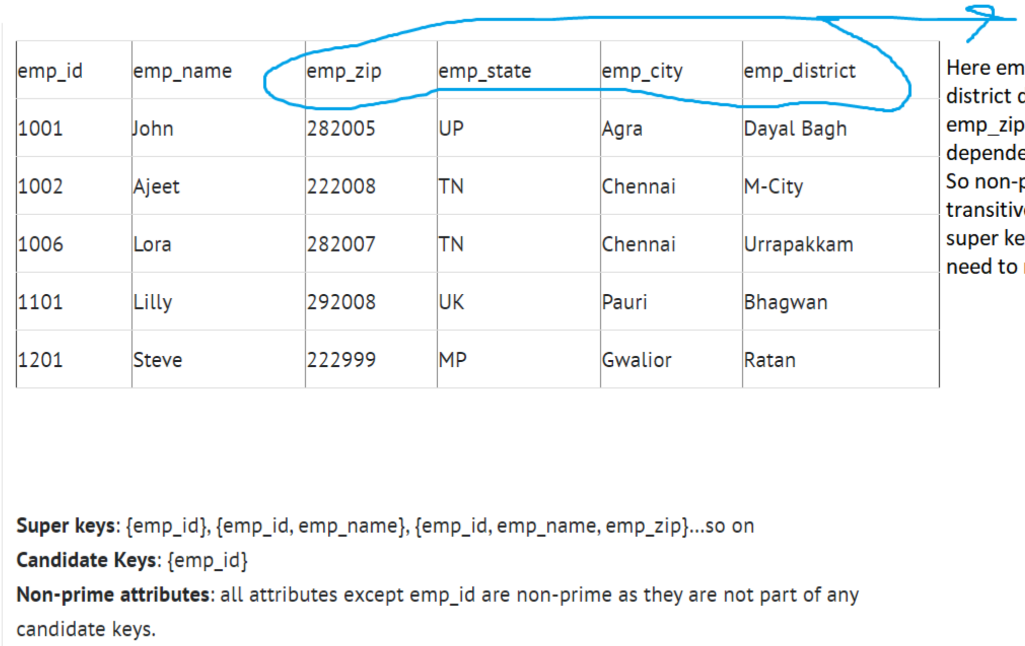
Table 2

COURSE_NO	COURSE_FEE
C1	1000
C2	1500
C3	1000
C4	2000
C5	2000

3NF – it also has 2 conditions-

- Table must be in 2NF
- Transitive functional dependency of non-prime attribute on any super key should be removed.

transitional functional dependency is when X->Y, then X is super key and Y is prime attribute.



emp_id	emp_name	emp_zip	emp_state	emp_city	emp_district
1001	John	282005	UP	Agra	Dayal Bagh
1002	Ajeet	222008	TN	Chennai	M-City
1006	Lora	282007	TN	Chennai	Urrapakkam
1101	Lilly	292008	UK	Pauri	Bhagwan
1201	Steve	222999	MP	Gwalior	Ratan

Super keys: {emp_id}, {emp_id, emp_name}, {emp_id, emp_name, emp_zip}...so on
Candidate Keys: {emp_id}
Non-prime attributes: all attributes except emp_id are non-prime as they are not part of any candidate keys.

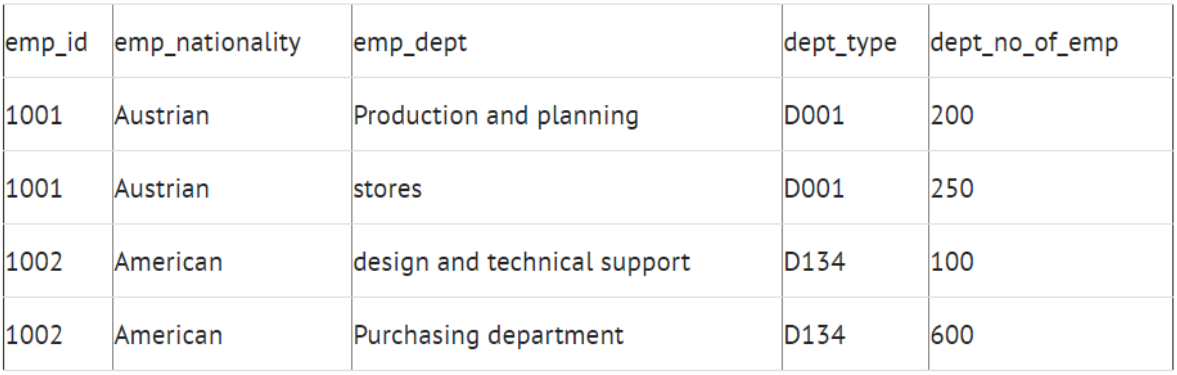
Here emp_state,city and district dependent on emp_zip and emp_zip is dependent on emp_id. So non-prime attributes transitively depend on super key, so we must need to remove it.

So we divide this table in two parts as-

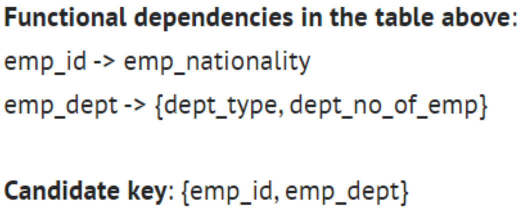
1. Emp_id, Emp_name,Emp_zip
2. Emp_zip, state, city and district.

BCNF – This is updated version of 3NF. Its conditions are-

- Table must be in 3NF
- For functional dependency X->Y, X should be a super key.



emp_id	emp_nationality	emp_dept	dept_type	dept_no_of_emp
1001	Austrian	Production and planning	D001	200
1001	Austrian	stores	D001	250
1002	American	design and technical support	D134	100
1002	American	Purchasing department	D134	600



Functional dependencies in the table above:
emp_id -> emp_nationality
emp_dept -> {dept_type, dept_no_of_emp}

Candidate key: {emp_id, emp_dept}

We can divide this table in 3 tables as-

1. Emp_id and emp_nationality
2. Emp_dept, dept_type and dept_no_of_emp
3. Emp_id and emp_dept (this is to map both above tables)