

**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 because deletion of department also deletes the employee informations

Insert anomaly because 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 non-prime attribute is dependent on the proper subset of any candidate key of table

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

teacher_id	subject	teacher_age
111	Maths	38
111	Physics	38
222	Biology	38
333	Physics	40
333	Chemistry	40

Here teacher key is depend on two candidate keys so this is not allow in 2NF

**Candidate Keys:** {teacher\_id, subject}

**Non prime attribute:** teacher\_age


We can remove this by made 2 different tables.

One table is of teacher\_id and teacher\_age, and another table is subject and teacher\_id.

**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  $X \rightarrow Y$ , 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

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.

**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.

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  $X \rightarrow Y$ , 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)