

Normalization: is arranging table, columns, and relationship between tables in order to avoid data redundancy and integrity.

Characteristics!

- to avoid redundancy
- to attain integrity
- to put data in correct tables
- to avoid CVD anomalies.

Table 1: Film Agents

| <u>Agent ID</u> | <u>Agent name</u> | <u>Agency name</u> | <u>off. loc</u> | <u>contact</u> | <u>customer</u>      |                      |                      |
|-----------------|-------------------|--------------------|-----------------|----------------|----------------------|----------------------|----------------------|
|                 |                   |                    |                 |                | <u>c<sub>1</sub></u> | <u>c<sub>2</sub></u> | <u>c<sub>3</sub></u> |
| A <sub>1</sub>  | Ari Gold          | New cine           | NY              | 111-888        | Brad                 | tong                 | -                    |
| A <sub>2</sub>  | Mari Silver       | old school         | LA              | 123-444        | Johny                | -                    | -                    |
| A <sub>3</sub>  | Toni Bronz        | New cine           | NY              | 111-888        | -                    | -                    | Nate                 |
| A <sub>4</sub>  | Toni diamond      | Fire Folk          | TX              | 149-333        | Morgan               | -                    | -                    |

Normalization Forms:

- 1NF
- 2NF
- 3NF
- BCNF (Boyce Codd)

1NF - Each normal form should be atomic.  
 - Can't have columns with similar information  
 - There should be a key which uniquely identifies a database table.

Table:2 Table 1 after 1NF

Agent Info

| <u>Agent ID</u> | <u>Agent Name</u> | <u>Agency name</u> | <u>off. location</u> | <u>contact</u> |
|-----------------|-------------------|--------------------|----------------------|----------------|
|-----------------|-------------------|--------------------|----------------------|----------------|

Table: 3 customer info

customer ID    Agent ID    cust. name    cust. city    cust. pincode

2NF : All conditions of 1NF  
- All non-key columns should be dependent on primary keys.

From table 2 & 3 to make non-key columns primary key:

Table: 4 (Agent Info)  
PK Agent ID    Agent name    Agency ID → FK

Table: 5 (Agency Info)

PK Agent ID    Agency name    office loc.    contact

Table: 6 (customer info)

PK cust. ID    cust. name    customer city    cust. pin code

Table 7: (Join table)

FK Agent ID    |    cust. ID → FK

3NF

- All conditions of 1NF & 2NF
- All the non-key columns should be non-transitively dependent on primary key.

From Table 6: (changing transitive dependency)

\* Table 9 (customer info)

customer.id      cust.name      cust.pincode <sup>FK</sup>

non-transitively dependent

Table 10 (customer pincode)

PK customer.pincode      customer.city

BCNF (Boyce Codd)

- It should be in 3NF
- For every functional dependencies  $x \rightarrow y$ ,  $x$  should be a superkey of a table:

| <u>Std.</u> | <u>course</u> | <u>teacher</u> |
|-------------|---------------|----------------|
|-------------|---------------|----------------|

|       |      |       |
|-------|------|-------|
| Pinky | DBMS | IMRAN |
| lucky | DBMS | IMRAN |
| Deepu | SPL  | KABIR |
| Banu  | SPL  | KABIR |
| Sonu  | DBMS | IMRAN |

- non-key attribute is not a superkey

=> Key : { Std, course }

=> { Std, course }  $\rightarrow$  teacher

Table 1      After BCNF

| <u>Std</u> | <u>course</u> |
|------------|---------------|
| Pinky      | DBMS          |
| lucky      | DBMS          |
| Deepu      | SPL           |
| Banu       | SPL           |
| Sonu       | DBMS          |

Table 2

| <u>course</u> | <u>teacher</u> |
|---------------|----------------|
| KABIR         |                |
| IMRAN         |                |