Unit-5

Normalization and normal forms

Outline

- Normalization and normal forms
  - 1NF
  - 2NF
  - 3NF

What is normalization?

- Normalization is the **process of removing redundant data** from tables **to improve data integrity, scalability and storage efficiency**.
  - data integrity (completeness, accuracy and consistency of data)
  - scalability (ability of a system to continue to function well in a growing amount of work)
  - storage efficiency (ability to store and manage data that consumes the least amount of space)

- What we do in normalization?
  - Normalization generally involves **splitting an existing table into multiple (more than one) tables**, which can be **re-joined or linked** each time a query is issued (executed).

How many normal forms are there?

- Normal forms:
    - 1NF (First normal form)
    - 2NF (Second normal form)
    - 3NF (Third normal form)
    - BCNF (Boyce–Codd normal form)
    - 4NF (Forth normal form)
    - 5NF (Fifth normal form)

As we move from 1NF to 5NF **number of tables** and **complexity increases** but **redundancy decreases**.

Normal forms

1NF (First Normal Form)

Section – 7.1

1NF (First Normal Form)

- Conditions for 1NF

  Each **cells of a table should contain a single value**.

- A relation R is in first normal form (1NF) if and only if it **does not contain any composite attribute or multi-valued attributes or their combinations**.

OR

- A relation R is in first normal form (1NF) if and only if **all underlying domains contain atomic values only**.

1NF (First Normal Form) [Example - Composite attribute]

Customer

CID	Name	Address
C01	Raju	Jamnagar Road, Rajkot
C02	Mitesh	Nehru Road, Jamnagar
C03	Jay	C.G Road, Ahmedabad

- In customer relation **address is composite attribute** which is further divided into sub-attributes as “Road” and “City”.
- So customer relation is not in 1NF.

- **Problem**: It is **difficult to retrieve the list of customers living in ’Jamnagar’ city** from customer table.
- The reason is that **address attribute is composite attribute** which **contains road name as well as city name in single cell**.
- It is possible that **city name word is also there in road name**.
- In our example, ’Jamnagar’ word occurs in both records, in first record it is a part of road name and in second one it is the name of city.

1NF (First Normal Form) [Example - Composite attribute]

Customer

CID	**Name**	**Address**
C01	Raju	Jamnagar Road, Rajkot
C02	Mitesh	Nehru Road, Jamnagar
C03	Jay	C.G Road, Ahmedabad

Customer

CID	**Name**	**Road**	**City**
C01	Raju	Jamnagar Road	Rajkot
C02	Mitesh	Nehru Road	Jamnagar
C03	Jay	C.G Road	Ahmedabad

- **Solution**: **Divide composite attributes** into **number of sub-attributes** and insert value in proper sub-attribute.

Exercise [redacted]

Person

PID	**Full_Name**	**City**
P01	Raju Maheshbhai Patel	Rajkot

1NF (First Normal Form) [Example - Multivalued attribute]

Student

Rno	**Name**	**FailedinSubjects**
101	Raju	DS, DBMs
102	Mitesh	DBMS, DS
103	Jay	DS, DBMS, DE
104	Jeet	DBMS, DE, DS
105	Harsh	DE, DBMS, DS
106	Neel	DE, DBMS

- In student relation **FailedinSubjects attribute is a multi-valued attribute** which can store more than one values.
- So above relation is not in 1NF.

- **Problem**: It is difficult to retrieve the **list of students failed in 'DBMS' as well as 'DS' but not in other subjects** from student table.
- The reason is that FailedinSubjects attribute is multi-valued attribute so it contains more than one value.

1NF (First Normal Form) [Example - Multivalued attribute]

Student

Rno	**Name**	**FailedinSubjects**
101	Raju	DS, DBMs
102	Mitesh	DBMS, DS
103	Jay	DS, DBMS, DE
104	Jeet	DBMS, DE, DS
105	Harsh	DE, DBMS, DS
106	Neel	DE, DBMS

Student

Rno	**Name**
101	Raju
102	Mitesh
103	Jay
104	Jeet
105	Harsh
106	Neel

Result

RID	**Rno**	**Subject**
1	101	DS
2	101	DBMS
3	102	DBMS
4	102	DS
5	103	DS
…	…	…

- **Solution**: Split the table into two tables in such as way that
  - the **first table contains all attributes except multi-valued attribute** with same primary key and
  - **second table contains multi-valued attribute** and **place a primary key** in it.
  - **insert the primary key of first table in the second table as a foreign key**.

Normal forms

2NF (Second Normal Form)

Section – 7.2

2NF (Second Normal Form)

- Conditions for 2NF

It is **in 1NF** and each **table should contain a single primary key**.

- A relation R is in second normal form (2NF)
  - if and only if it is in **1NF** and
  - **every non-primary key attribute is fully dependent on the primary key**

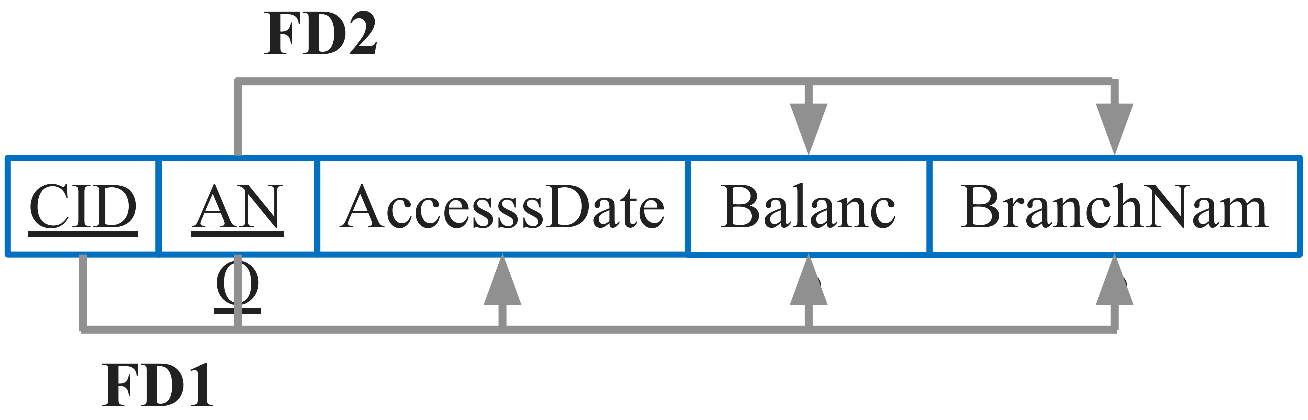
OR

- A relation R is in second normal form (2NF)
  - if and only if it is in **1NF** and
  - **no any non-primary key attribute is partially dependent on the primary key**

2NF (Second Normal Form) [Example]

Customer

<u>**CID**</u>	<u>**ANO**</u>	**AccessDate**	**Balance**	**BranchName**
C01	A01	01-01-2017	50000	Rajkot
C02	A01	01-03-2017	50000	Rajkot
C01	A02	01-05-2017	25000	Surat
C03	A02	01-07-2017	25000	Surat

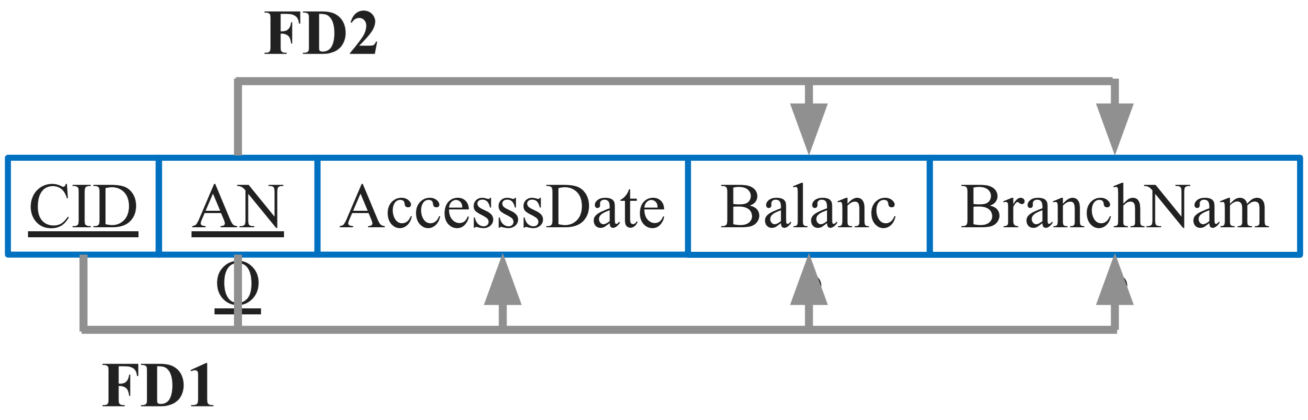


- **FD1**: {CID, ANO} → {AccesssDate, Balance, BranchName}
- **FD2**: ANO → {Balance, BranchName}
- **Balance and BranchName are partial dependent on primary key (CID + ANO)**. So customer relation is not in 2NF.

2NF (Second Normal Form) [Example]

Customer

CID	**ANO**	**AccessDate**	**Balance**	**BranchName**
C01	A01	01-01-2017	50000	Rajkot
C02	A01	01-03-2017	50000	Rajkot
C01	A02	01-05-2017	25000	Surat
C03	A02	01-07-2017	25000	Surat



- **Problem:** For example, in case of a joint account multiple (more than one) customers have common (one) accounts.
- If an account **’A01’ is operated jointly by two customers** says **’C01’ and ’C02’** then **data** values for attributes **Balance and BranchName** will be **duplicated in two different tuples** of customers ’C01’ and ’C02’.

2NF (Second Normal Form) [Example]

Customer

CID	ANO	AccessDate	Balance	BranchName
C01	A01	01-01-2017	50000	Rajkot
C02	A01	01-03-2017	50000	Rajkot
C01	A02	01-05-2017	25000	Surat
C03	A02	01-07-2017	25000	Surat

Table-1

ANO	Balance	BranchName
A01	50000	Rajkot
A02	25000	Surat

Table-2

CID	ANO	AccessDate
C01	A01	01-01-2017
C02	A01	01-03-2017
C01	A02	01-05-2017
C03	A02	01-07-2017

- **Solution: Decompose relation** in such a way that **resultant relations do not have any partial FD**.
  - **Remove partial dependent attributes** from the relation that violets 2NF.
  - **Place them in separate relation** along with the **prime attribute on which they are fully dependent**.
  - The **primary key of new relation** will be the **attribute on which it is fully dependent**.
  - **Keep other attributes same** as in that table with the **same primary key**.

Normal forms

3NF (Third Normal Form)

Section – 7.3

3NF (Third Normal Form)

- Conditions for 3NF

It is in 2NF and there is no transitive dependency.

(Transitive dependency???) A → B & B → C then A → C

- A relation R is in third normal form (3NF)
  - if and only if it is in **2NF** and
  - **every non-key attribute is non-transitively dependent on the primary key**

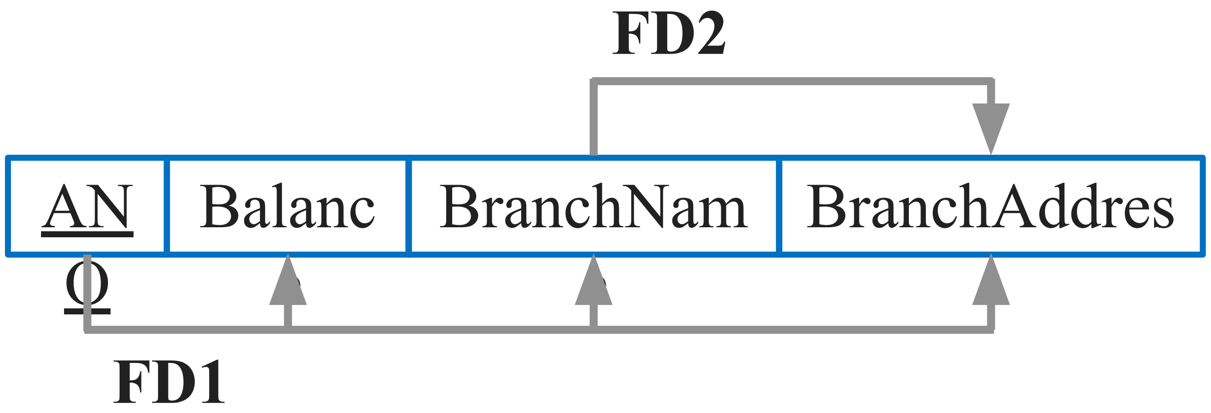
OR

- A relation R is in third normal form (3NF)
  - if and only if it is in **2NF** and
  - **no any non-key attribute is transitively dependent on the primary key**

3NF (Third Normal Form) [Example]

Customer

O	**Balance**	**BranchName**	**BranchAddress**
A01	50000	Rajkot	Kalawad road
A02	40000	Rajkot	Kalawad Road
A03	35000	Surat	C.G Road
A04	25000	Surat	C.G Road

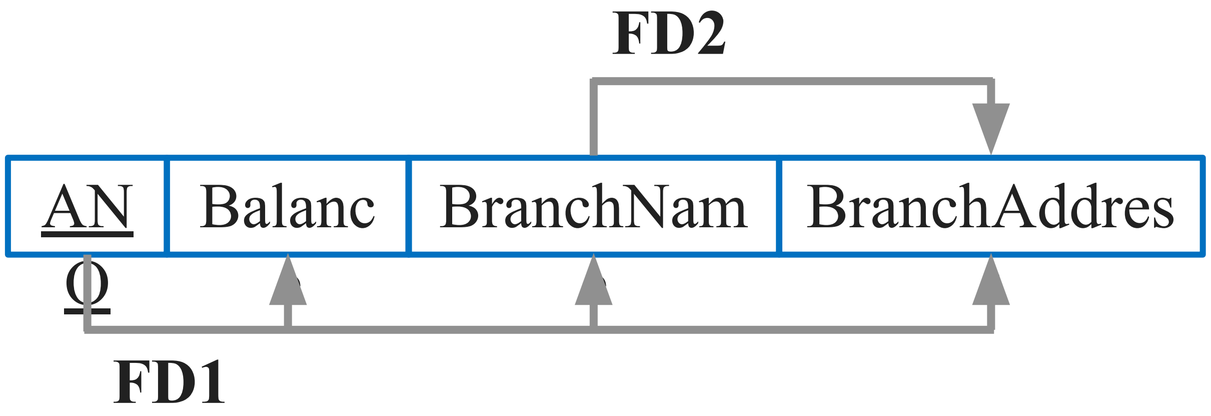


- **FD1**: ANO → {Balance, BranchName, BranchAddress}
- **FD2**: BranchName → BranchAddress
- So AccountNO → BranchAddress (Using Transitivity rule)
- **BranchAddress is transitive depend on primary key (ANO)**. So customer relation is not in 3NF.

3NF (Third Normal Form) [Example]

Customer

O	**Balance**	**BranchName**	**BranchAddress**
A01	50000	Rajkot	Kalawad road
A02	40000	Rajkot	Kalawad Road
A03	35000	Surat	C.G Road
A04	25000	Surat	C.G Road



- **Problem:** In this relation, **branch address will be stored repeatedly** for each account of the same branch which **occupies more space**.

3NF (Third Normal Form) [Example]

Customer

ANO	**Balance**	**BranchName**	**BranchAddress**
A01	50000	Rajkot	Kalawad road
A02	40000	Rajkot	Kalawad Road
A03	35000	Surat	C.G Road
A04	25000	Surat	C.G Road

Table-1

BranchName	**BranchAddress**
Rajkot	Kalawad road
Surat	C.G Road

Table-2

ANO	**Balance**	**BranchName**
A01	50000	Rajkot
A02	40000	Rajkot
A03	35000	Surat
A04	25000	Surat

- **Solution: Decompose relation in** such a way that **resultant relations do not have any transitive FD.**
  - **Remove transitive dependent attributes** from the relation that violets 3NF.
  - **Place them in a new relation along** with the **non-prime attributes due to which transitive dependency occurred**.
  - The **primary key of the new relation** will be **non-prime attributes due to which transitive dependency occurred**.
  - **Keep other attributes same as in the table** with **same primary key** and **add prime attributes of other relation into it as a foreign key**.

How to normalize database?

- A software contract and consultancy firm maintains details of all the various projects in which its employees are currently involved. These details comprise: Employee Number, Employee Name, Date of Birth, Department Code, Department Name, Project Code, Project Description, Project Supervisor.
- Assume the following:
  - Each employee number is unique.
  - Each department has a single department code.
  - Each project has a single code and supervisor.
  - Each employee may work on one or more projects.
  - Employee names need not necessarily be unique.
  - Project Code, Project Description and Project Supervisor are repeating fields.
  - Normalize this data to Third Normal Form.

How to normalize database?

- A software contract and consultancy firm maintains details of all the various projects in which its employees are currently involved. These details comprise: Employee Number, Employee Name, Date of Birth, Department Code, Department Name, Project Code, Project Description, Project Supervisor.

UNF

Employee Number	Employee Name	Date of Birth	Department Code	Department Name	Project Code	Project Description	Project Supervisor
1	Raj	1-1-85	1	CE	1	IOT	Patel
2	Meet	4-4-86	2	EC	2	PHP	Shah
3	Suresh	2-2-85	1	CE	1	IOT	Patel
1	Raj	1-1-85	1	CE	2	PHP	Shah

How to normalize database?

UNF

Employee Number	Employee Name	Date of Birth	Department Code	Department Name	Project Code	Project Description	Project Supervisor
1	Raj	1-1-85	1	CE	1	IOT	Patel
2	Meet	4-4-86	2	EC	2	PHP	Shah
3	Suresh	2-2-85	1	CE	1	IOT	Patel
1	Raj	1-1-85	1	CE	2	PHP	Shah

1NF

Employee Number	Employee Name	Date of Birth	Department Code	Department Name
1	Raj	1-1-85	1	CE
2	Meet	4-4-86	2	EC
3	Suresh	2-2-85	1	CE

Employee Number	Project Code	Project Description	Project Supervisor
1	1	IOT	Patel
2	2	PHP	Shah
3	1	IOT	Patel
1	2	PHP	Shah

How to normalize database?

1NF

Employee Number	Employee Name	Date of Birth	Department Code	Department Name
1	Raj	1-1-85	1	CE
2	Meet	4-4-86	2	EC
3	Suresh	2-2-85	1	CE

Employee Number	Project Code	Project Description	Project Supervisor
1	1	IOT	Patel
2	2	PHP	Shah
3	1	IOT	Patel
1	2	PHP	Shah

2NF

Employee Number	Employee Name	Date of Birth	Department Code	Department Name
1	Raj	1-1-85	1	CE
2	Meet	4-4-86	2	EC
3	Suresh	2-2-85	1	CE

Project Code	Project Description	Project Supervisor
1	IOT	Patel
2	PHP	Shah

Employee Number	Project Code
1	1
2	2
3	1
1	2

How to normalize database?

3NF

Employee Number	Employee Name	Date of Birth	Department Code
1	Raj	1-1-85	1
2	Meet	4-4-86	2
3	Suresh	2-2-85	1

Department Code	Department Name
1	CE
2	EC

Project Code	Project Description	Project Supervisor
1	IOT	Patel
2	PHP	Shah

Employee Number	Project Code
1	1
2	2
3	1
1	2

Find out User_Personal table is in 1NF, 2NF or 3NF?

UserID	U_email	Fname	Lname	City	State	Zip
MA12	Mani@ymail.com	MANISH	JAIN	BILASPUR	CHATISGARH	458991
PO45	Pooja.g@gmail.com	POOJA	MAGG	KACCH	GUJRAT	832212
LA33	Lavle98@jj.com	LAVLEEN	DHALLA	RAIPUR	CHATISGARH	853578
CH99	Cheki9j@ih.com	CHIMAL	BEDI	TRICHY	TAMIL NADU	632011
DA74	Danu58@g.com	DANY	JAMES	TRICHY	TAMIL NADU	645018

Is this table in First Normal Form?

Yes. All the attributes contain only atomic values.

Is this table in Second Normal Form?

Yes. Primary key of our table is UserID and UserID is single simple attribute. As the key is not composite, there is no chance for partial key dependency to hold.

Is User_Personal in 3NF?

- User_Personal table holds the following Transitive dependency;
- UserID → Zip, Zip → City State
- Decompose User_Personal. For this, we can use the functional dependencies Zip → City State and UserID → U_email Fname Lname City State Zip.

UserID	U_email	Fname	Lname	Zip
MA12	Mani@ymail.com	MANISH	JAIN	458991
PO45	Pooja.g@gmail.co	POOJA	MAGG	832212
LA33	Lavle98@jj.com	LAVLEEN	DHALLA	853578
CH99	Cheki9j@ih.com	CHIMAL	BEDI	632011
DA74	Danu58@g.com	DANY	JAMES	645018

Zip	City	State
458991	BILASPUR	CHATISGARH
832212	KACCH	GUJRAT
853578	RAIPUR	CHATISGARH
632011	TRICHY	TAMIL NADU
645018	TRICHY	TAMIL NADU