Normalization

Functional Dependency and Schema Refinement

Equivalence

Equivalence of FDs

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

- Given a two FDs F and G, they will be equivalent iff
  - G is the subset of F, i.e., F is covering G, and
  - F is the subset of G, i.e., G is covering F,
- Example:

F: {A → B, AB → C, D → AC, D → E}

G: {A → BC, D → AB}

Not equivalent

- Example:

F: {A → B, B → C, C → A}

G: {A → BC, B → A, C → A}

Equivalent

Canonical cover

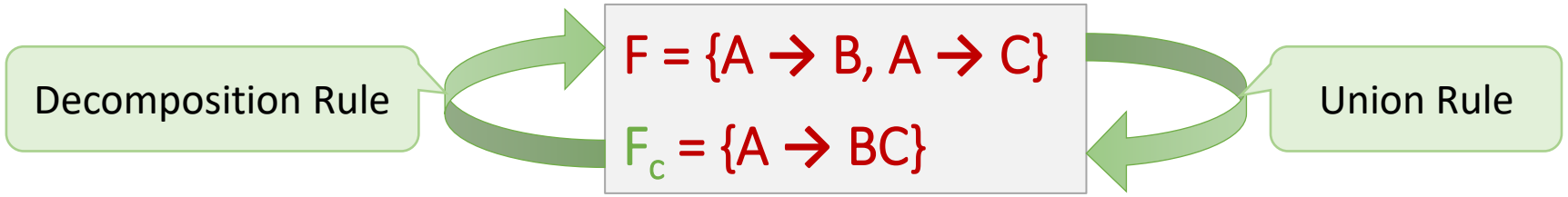
What is extraneous attributes?

- Let us consider a relation R with schema R = (A, B, C) and set of functional dependencies FDs **F = { AB → C, A → C }**.
- In **AB → C, B is extraneous attribute**. The reason is, there is another FD **A → C**, which means when **A alone can determine C**, the use of B is unnecessary (extra).
- An attribute of a functional dependency is said to be extraneous if we can **remove it without changing the closure of the set of functional dependencies**.

What is canonical cover?

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

- A canonical cover of F is a **minimal set of functional dependencies** equivalent to F, having **no redundant dependencies or redundant parts of dependencies**.
- It is denoted by **F_c**
- A canonical cover for F is a set of dependencies F_c such that
  - **F logically implies** all dependencies in **F_c** and
  - **F_c logically implies** all dependencies in **F** and
  - **No** functional dependency in **F_c** contains an **extraneous attribute** and
  - Each **left side** of functional dependency in **F_c** is **unique**.



Algorithm to find canonical cover

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

- *Repeat*
  - Use the **union rule** to replace any dependencies in F **α1 → β1** and **α1 → β2 with α1 → β1β2**
  - Find a functional dependency α → β with an **extraneous attribute** either in α or in β

    /* Note: test for extraneous attributes done using F_c, not F */
    - If an **extraneous attribute is found, delete it** from α → β
- *until* F does not change

  /* Note: Union rule may become applicable after some extraneous attributes have been deleted, so it has to be re-applied */

Canonical cover [Example]

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

- Consider the relation schema R = (A, B, C) with FDs

  F = {A → BC, B → C, A → B, AB → C}

- Find canonical cover.

- Combine A → BC and A → B into A → BC (Union Rule)
  - Set is {A → BC, B → C, AB → C}
- A is extraneous in AB → C
  - Check if the result of deleting A from AB → C is implied by the other dependencies
    - Yes: in fact, B → C is already present
  - Set is {A → BC, B → C}
- C is extraneous in A → BC
  - Check if A → C is logically implied by A → B and the other dependencies
    - Yes: using transitivity on A → B and B → C.
  - The canonical cover is: **A → B, B → C**

Canonical cover [Example]

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

- Consider the relation schema R = (A, B, C, D, E, F) with FDs

  $F = \{A \rightarrow BC, CD \rightarrow E, B \rightarrow D, E \rightarrow A\}$

- Find canonical cover.

- The left side of each FD in F is unique.
- Also none of the attributes in the left side or right side of any of the FDs is extraneous.
- Therefore the canonical cover F_c is equal to F.
- $F_c = \{A \rightarrow BC, CD \rightarrow E, B \rightarrow D, E \rightarrow A\}$

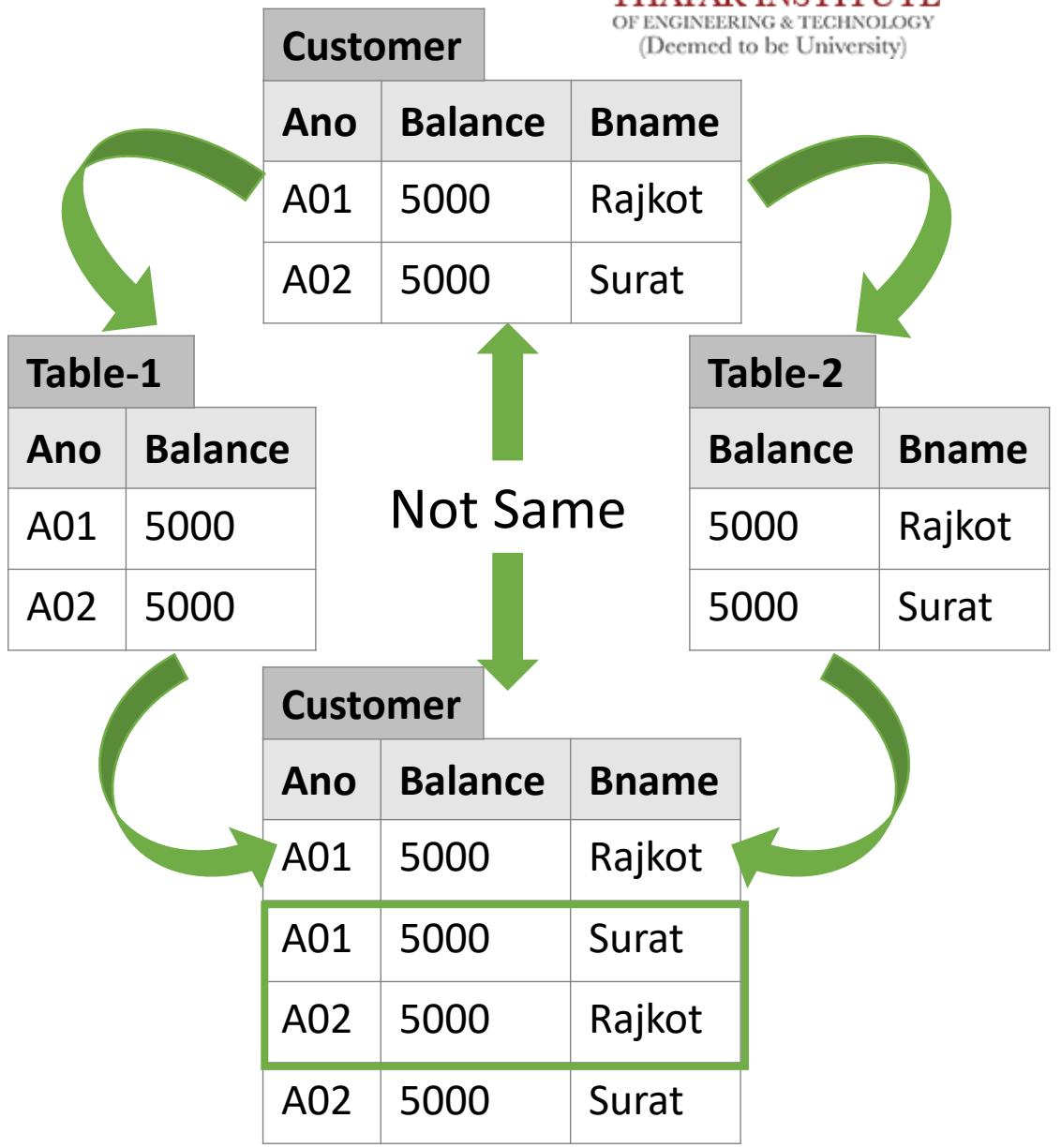
Decomposition

What is decomposition?

- Decomposition is the **process of breaking down given relation** into **two or more relations**.
- Relation R is replaced by two or more relations in such a way that:
  - Each new relation contains a **subset** of the **attributes of R**
  - Together, they all **include all tuples** and **attributes of R**
- Types of decomposition
  - Lossy decomposition
  - Lossless decomposition (non-loss decomposition)

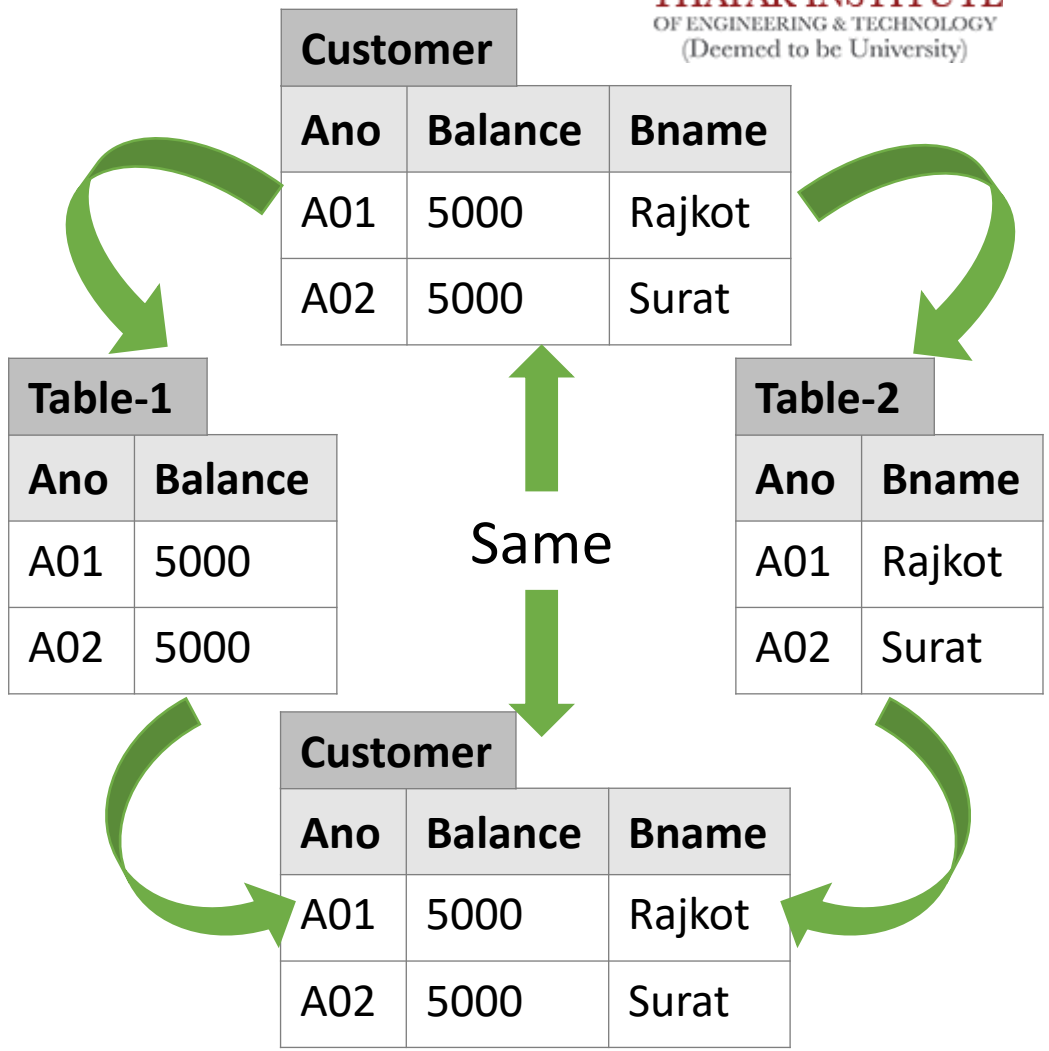
Lossy decomposition

- The decomposition of relation R into R1 and R2 is lossy when the join of R1 and R2 does not yield the same relation as in R.
- This is also referred as lossy-join decomposition.
- The disadvantage of such kind of decomposition is that some information is lost during retrieval of original relation.
- From practical point of view, decomposition should not be lossy decomposition.



Lossless decomposition

- The decomposition of relation R into R1 and R2 is lossless when the join of R1 and R2 produces the same relation as in R.
- This is also referred as a non-additive (non-loss) decomposition.
- All decompositions must be lossless.



Normalization and normal forms

What is normalization?

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

- 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?

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

- 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)

1NF (First Normal Form)

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

- 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]

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

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]

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

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 Convert below relation into 1NF (First Normal Form)

Person

PID	Full_Name	City
P01	Raju Maheshbhai Patel	Rajkot

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

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

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]

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

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)

2NF (Second Normal Form)

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

- 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

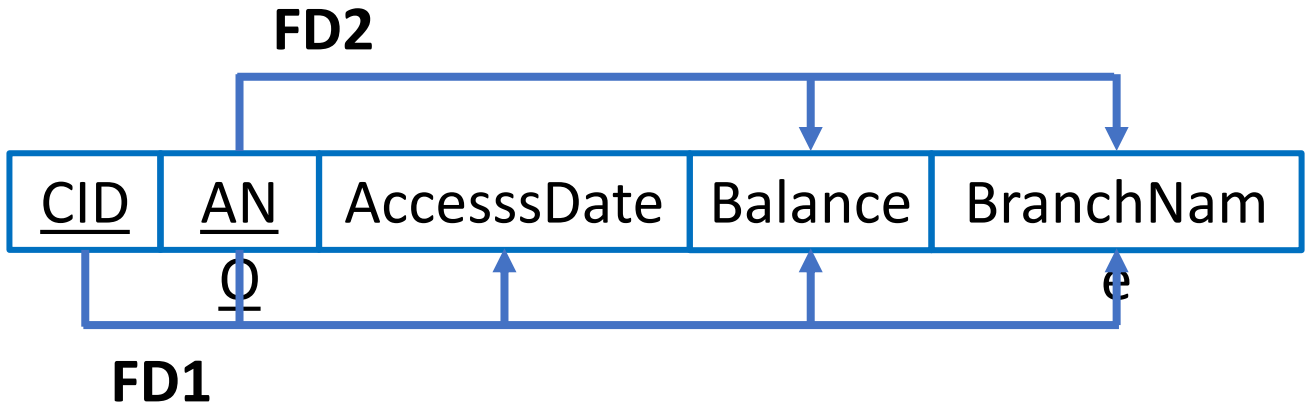
- A relation R is in second normal form (2NF)
  - if and only if it is in **1NF** and
  - **no any non-primary key attribute is partially dependent on the primary key**

2NF (Second Normal Form) [Example]

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

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



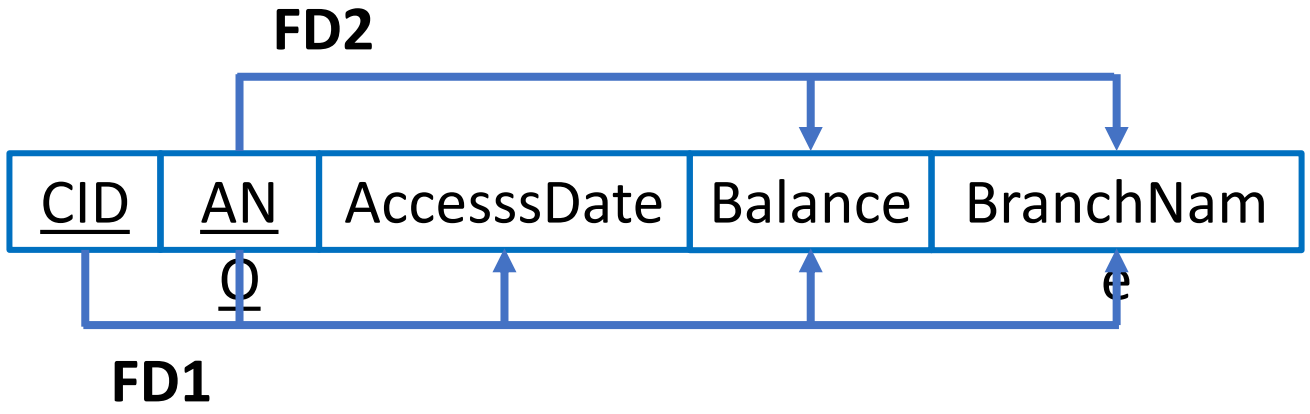
- **FD1**: {CID, ANO} → {AccesssDate, Balance, BranchName}
- **FD2**: ANO → {Balance, BranchName}
- **Balance and BranchName are partial dependent on primary key (CID + ANO)**. So customer relation is not in 2NF.

2NF (Second Normal Form) [Example]

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

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]

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

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

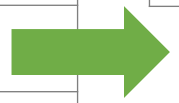


Table-1

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.

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

Normal forms
3NF (Third Normal Form)

3NF (Third Normal Form)

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

- 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

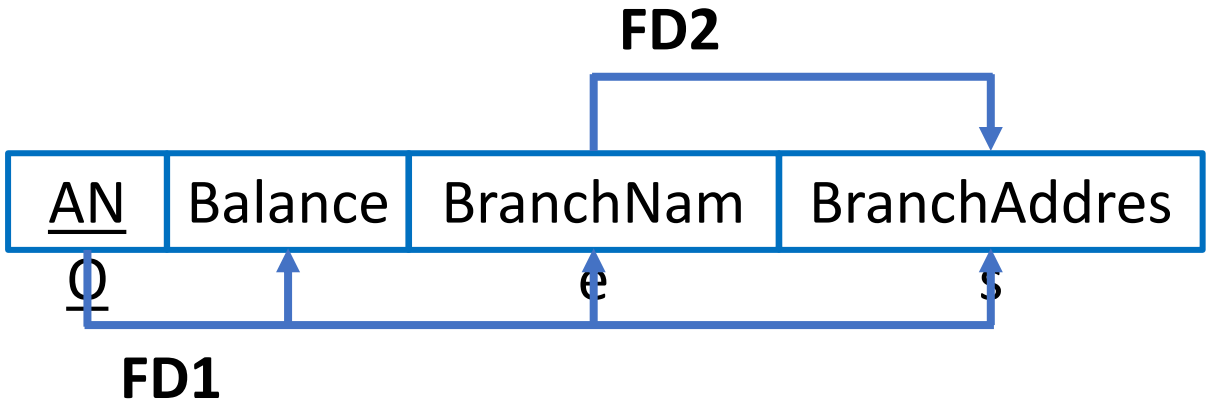
- A relation R is in third normal form (3NF)
  - if and only if it is in **2NF** and
  - **no any non-key attribute is transitively dependent on the primary key**

3NF (Third Normal Form) [Example]

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

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



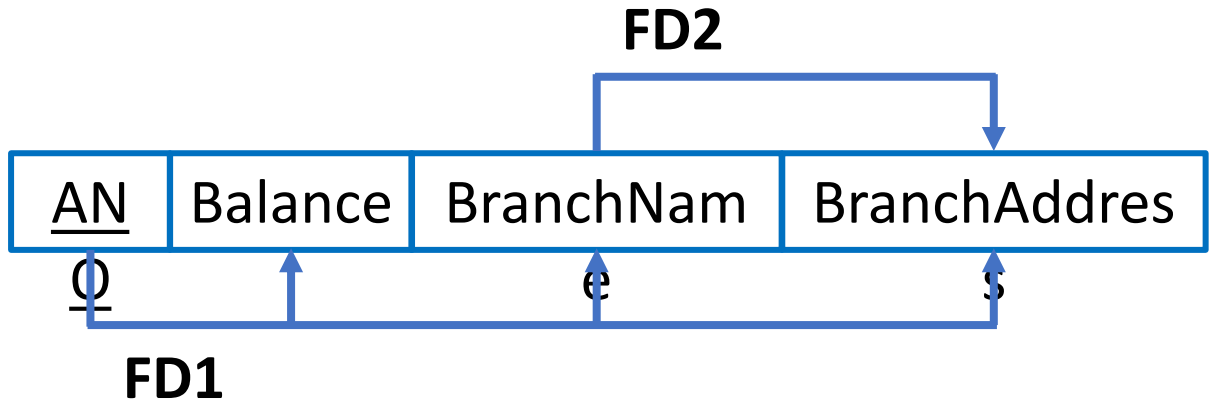
- **FD1**: ANO → {Balance, BranchName, BranchAddress}
- **FD2**: BranchName → BranchAddress
- So AccountNO → BranchAddress (Using Transitivity rule)
- **BranchAddress is transitive depend on primary key (ANO)**. So customer relation is not in 3NF.

3NF (Third Normal Form) [Example]

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

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



- **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]

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

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**.

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

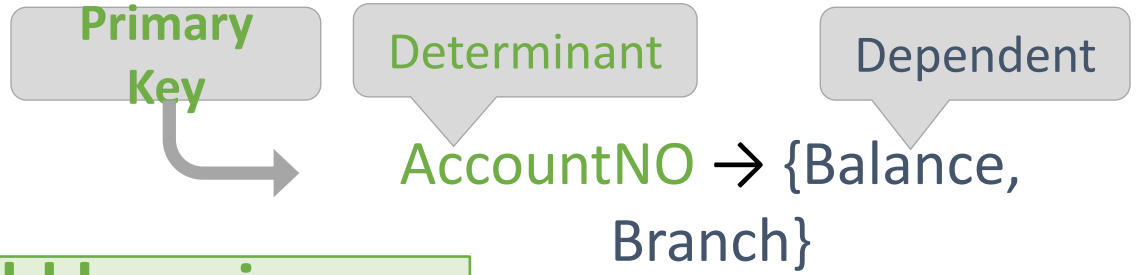
Normal forms
BCNF (Boyce-Codd Normal Form)

BCNF (Boyce-Codd Normal Form)

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

- Conditions for BCNF

BCNF is **based on the concept of a determinant**.



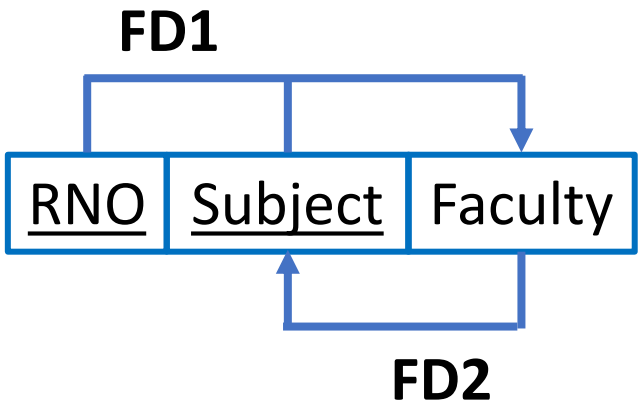
It is in **3NF** and **every determinant should be primary key**.

- A relation R is in Boyce-Codd normal form (BCNF)
  - if and only if it is in 3NF and
  - for every functional dependency X → Y, X should be the primary key of the table. **OR**
- A relation R is in Boyce-Codd normal form (BCNF)
  - if and only if it is in 3NF and
  - every prime key attribute is non-transitively dependent on the primary key **OR**
- A relation R is in Boyce-Codd normal form (BCNF)
  - if and only if it is in 3NF and
  - no any prime key attribute is transitively dependent on the primary key

BCNF (Boyce-Codd Normal Form) [Example]

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

Student		
RNO	**Subject**	**Faculty**
101	DS	Patel
102	DBMS	Shah
103	DS	Jadeja
104	DBMS	Dave
105	DBMS	Shah
102	DS	Patel
101	DBMS	Dave
105	DS	Jadeja



- **FD1**: RNO, Subject → Faculty
- **FD2**: Faculty → Subject
- So {RNO, Subject} → Subject (Transitivity rule)

In FD2, **determinant is Faculty which is not a primary key**. So student table is not in BCNF.

Problem: In this relation **one student can learn more than one subject with different faculty** then **records will be stored repeatedly for each student, language and faculty combination** which **occupies more space**.

- Here, one faculty teaches only one subject, but a subject may be taught by more than one faculty.
- A student can learn a subject from only one faculty.

BCNF (Boyce-Codd Normal Form) [Example]

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

Student

RNO	Subject	Faculty
101	DS	Patel
102	DBMS	Shah
103	DS	Jadeja
104	DBMS	Dave
105	DBMS	Shah
102	DS	Patel
101	DBMS	Dave
105	DS	Jadeja

Table-1

Faculty	Subject
Patel	DS
Shah	DBMS
Jadeja	DS
Dave	DBMS

Table-2

RNO	Faculty
101	Patel
102	Shah
103	Jadeja
104	Dave
105	Shah
102	Patel
101	Dave
105	Jadeja

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