Database Design: Normalization

Dr. Jenila Livingston

Data Normalization

- Primarily a tool to validate and improve a logical design so that it satisfies certain constraints that ***avoid unnecessary duplication of data***
- The process of decomposing relations with anomalies to produce smaller, ***well-structured*** relations

Results of Normalization

- Removes the following modification *anomalies* (integrity errors) with the database
  - Insertion
  - Deletion
  - Update

A Typical Spreadsheet File

Emp No	Employee Name	Time Card No	Time Card Date	Dept No	Dept Name
10	Thomas Arquette	106	11/02/2002	20	Marketing
10	Thomas Arquette	106	11/02/2002	20	Marketing
10	Thomas Arquette	106	11/02/2002	20	Marketing
10	Thomas Arquette	115	11/09/2002	20	Marketing
99	Janice Smitty			10	Accounting
500	Alan Cook	107	11/02/2002	50	Shipping
500	Alan Cook	107	11/02/2002	50	Shipping
700	Ernest Gold	108	11/02/2002	50	Shipping
700	Ernest Gold	116	11/09/2002	50	Shipping
700	Ernest Gold	116	11/09/2002	50	Shipping

Employee, Department, and Time Card Data in Three Tables

Table: Employees

EmpNo	EmpFirstName	EmpLastName	DeptNo
10	Thomas	Arquette	20
500	Alan	Cook	50
700	Ernest	Gold	50
99	Janice	Smitty	10

Table: Departments

DeptNo	DeptName
10	Accounting
20	Marketing
50	Shipping

Table: Time Card Data

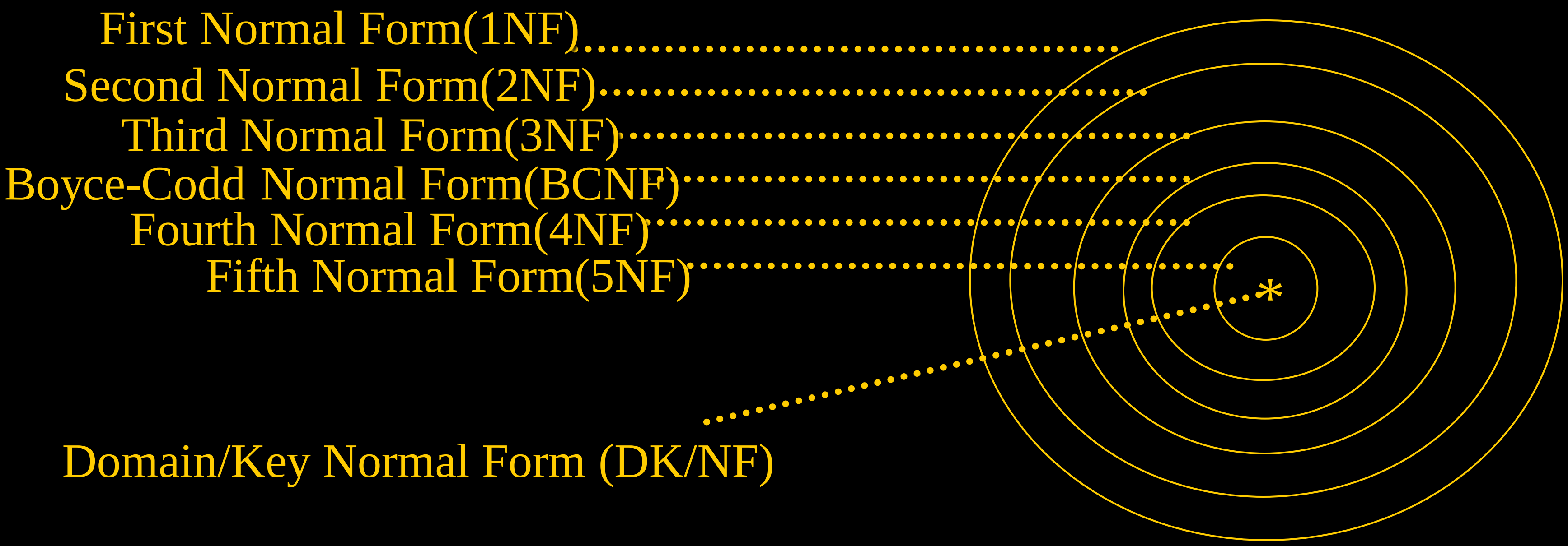
TimeCardNo	EmpNo	TimeCardDate
106	10	11/02/2002
107	500	11/02/2002
108	700	11/02/2002
115	10	11/09/2002
116	700	11/09/2002

🟧 Primary Key

NORMAL FORMS

- ✓ 1 NF
- ✓ 2NF
- ✓ 3NF
- BCNF (Boyce-Codd Normal Form)
- 4NF
- 5NF
- *DK (Domain-Key) NF*

Relationships of Normal Forms



First Normal Form(1NF)

Second Normal Form(2NF)

Third Normal Form(3NF)

Boyce-Codd Normal Form(BCNF)

Fourth Normal Form(4NF)

Fifth Normal Form(5NF)

Domain/Key Normal Form (DK/NF)

Normal Forms

- First Normal Form
  - No repeating groups in tables
- Second Normal Form
  - Table is 1st normal form and no partial key dependencies
- Third Normal Form
  - Table is in 2nd normal form and has no transitive dependencies

Normal Forms

- Boyce-Codd Normal Form
  - Every determinant of a non-key attribute is a candidate key
- Fourth Normal Form
  - A table has no multi-valued dependencies
- Fifth Normal Form
  - There are no lossey joins between two or more tables

Figure: 4-22 Steps in normalization

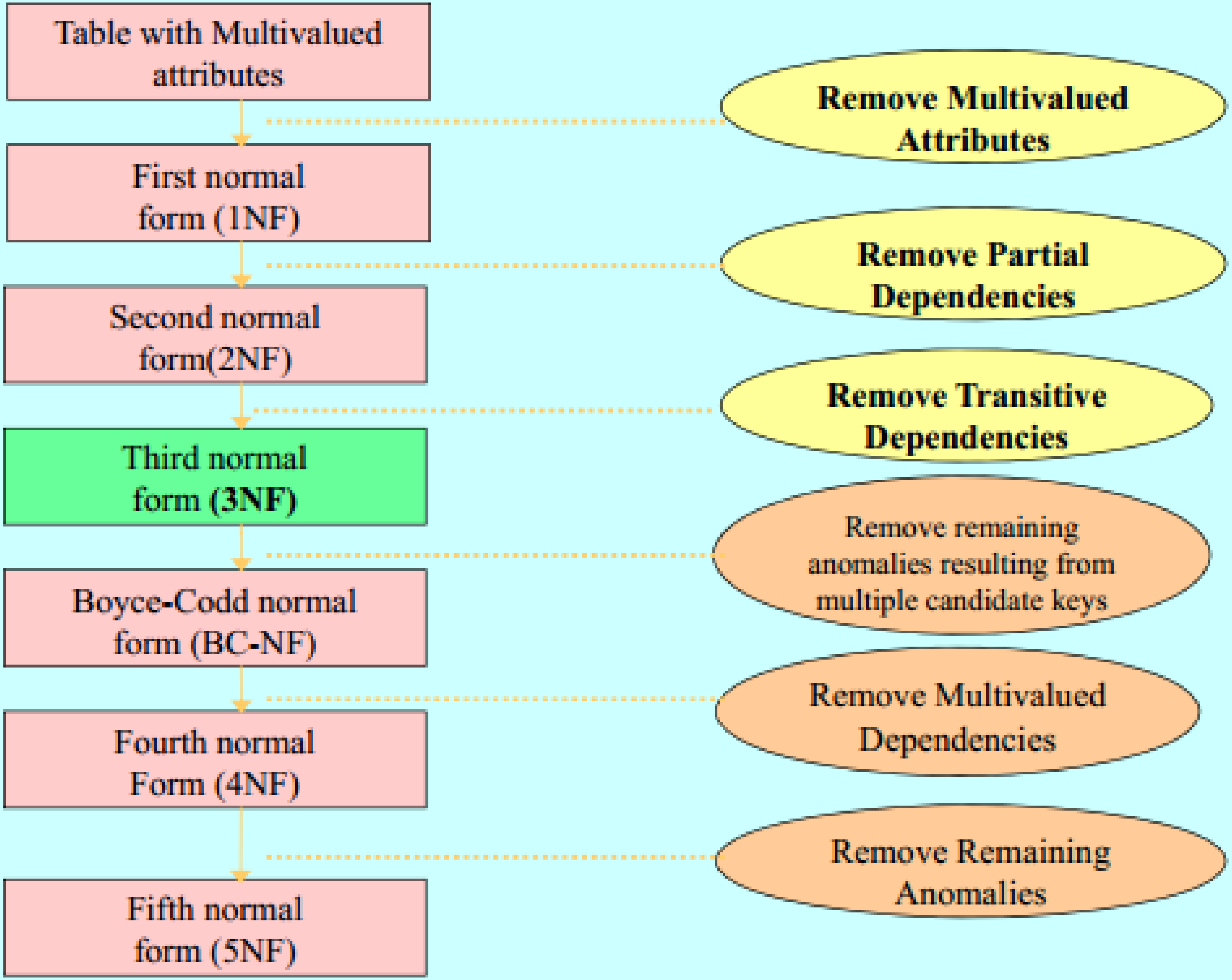


Table with Multivalued attributes

Remove Multivalued Attributes

First normal form (1NF)

Remove Partial Dependencies

Second normal form(2NF)

Remove Transitive Dependencies

Third normal form **(3NF)**

Remove remaining anomalies resulting from multiple candidate keys

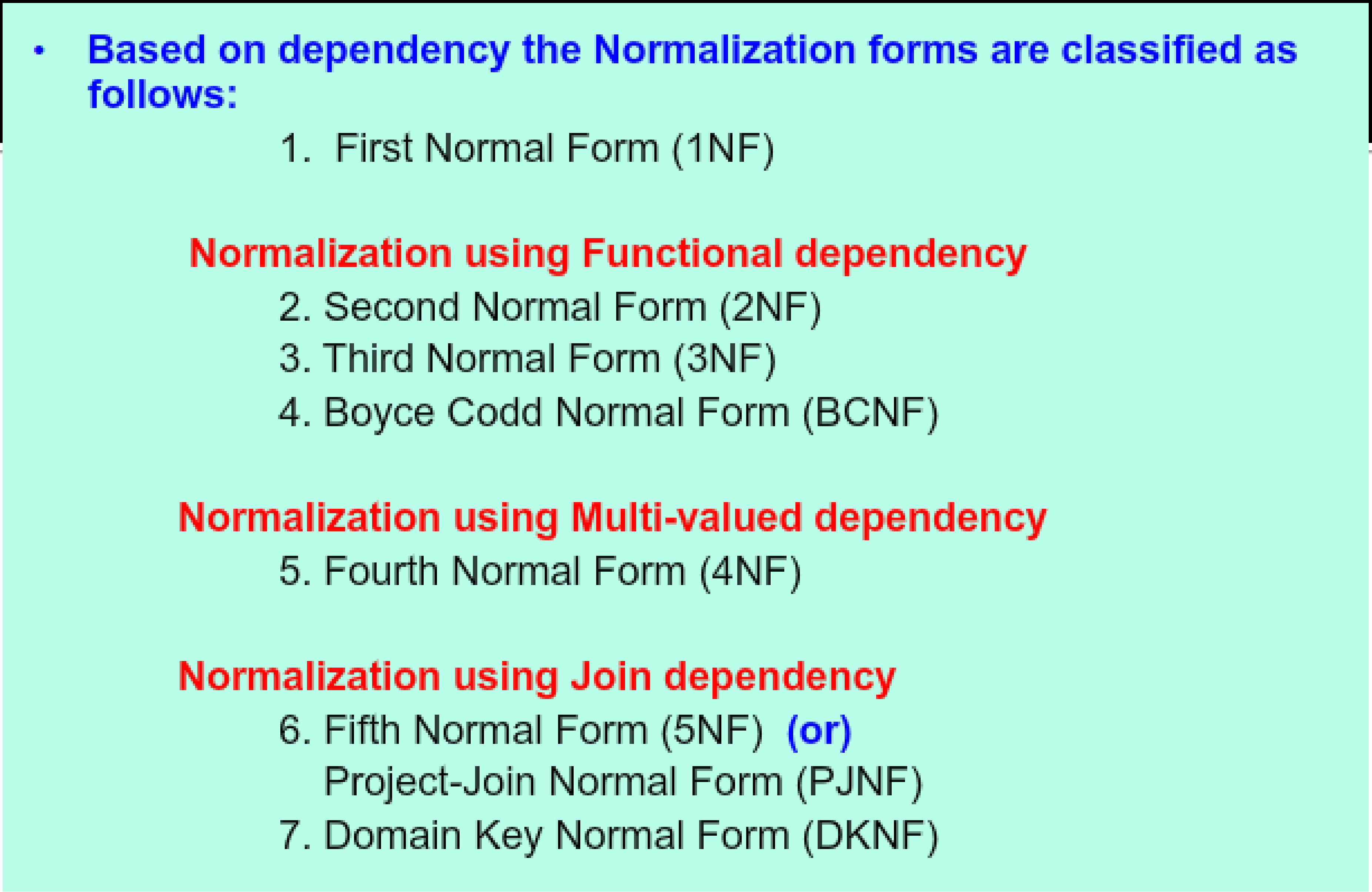
Boyce-Codd normal form (BC-NF)

Remove Multivalued Dependencies

Fourth normal Form (4NF)

Remove Remaining Anomalies

Fifth normal form (5NF)



- **Based on dependency the Normalization forms are classified as follows:**

1. First Normal Form (1NF)

Normalization using Functional dependency

2. Second Normal Form (2NF)
3. Third Normal Form (3NF)
4. Boyce Codd Normal Form (BCNF)

Normalization using Multi-valued dependency

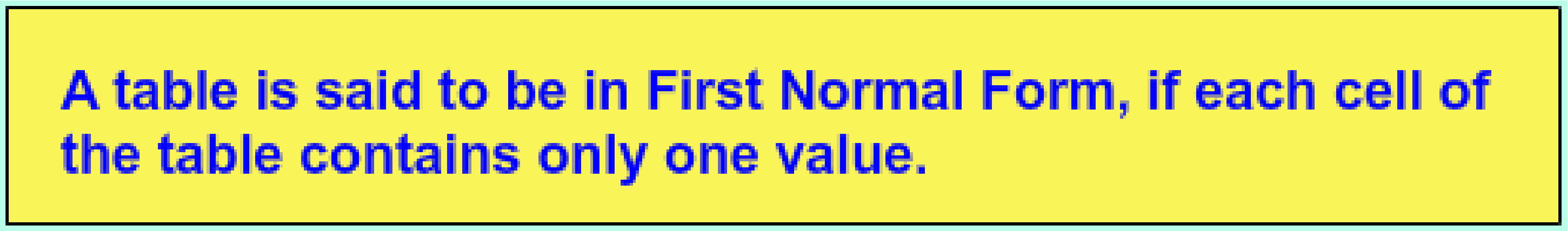
5. Fourth Normal Form (4NF)

Normalization using Join dependency

6. Fifth Normal Form (5NF) **(or)**
   Project-Join Normal Form (PJNF)
7. Domain Key Normal Form (DKNF)

1NF

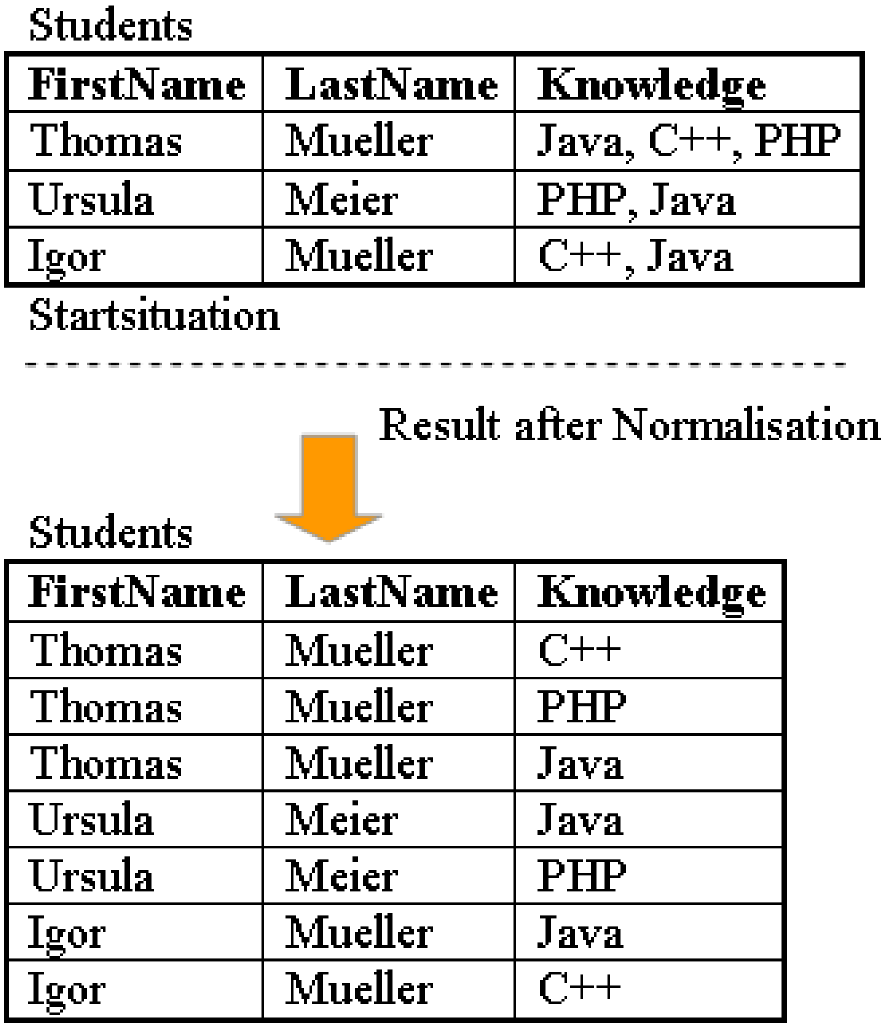
- Rule



A table is said to be in First Normal Form, if each cell of the table contains only one value.

- A university uses the following relation:

After 1NF



Students

FirstName	LastName	Knowledge
Thomas	Mueller	Java, C++, PHP
Ursula	Meier	PHP, Java
Igor	Mueller	C++, Java

Startsituation

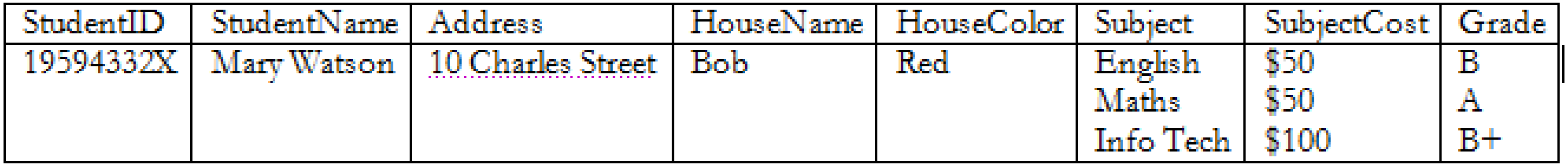
Result after Normalisation

Students

FirstName	LastName	Knowledge
Thomas	Mueller	C++
Thomas	Mueller	PHP
Thomas	Mueller	Java
Ursula	Meier	Java
Ursula	Meier	PHP
Igor	Mueller	Java
Igor	Mueller	C++

Take the following table.

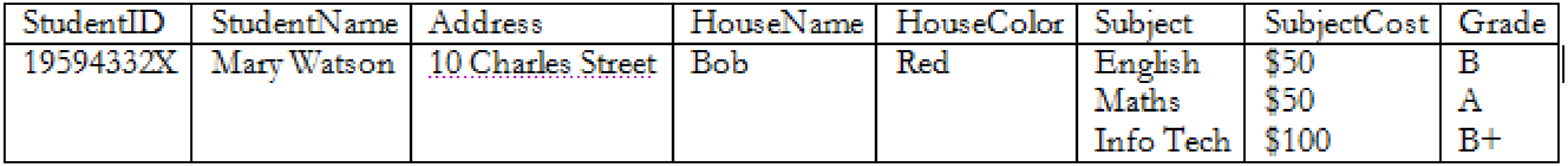
StudentID is the primary key.



StudentID	StudentName	Address	HouseName	HouseColor	Subject	SubjectCost	Grade
19594332X	Mary Watson	10 Charles Street	Bob	Red	English Maths Info Tech	$50 $50 $100	B A B+

Is it 1NF?

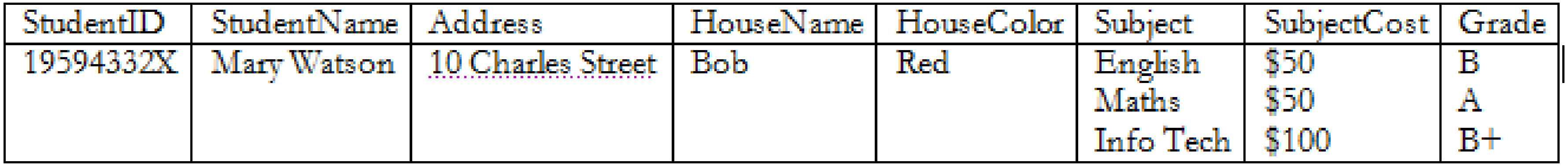
No. There are repeating groups (subject, subjectcost, grade)



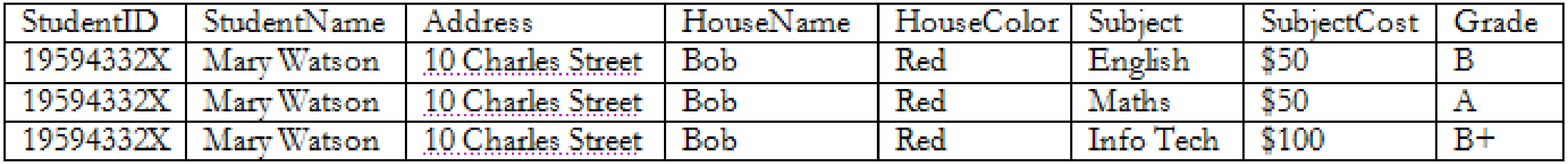
StudentID	StudentName	Address	HouseName	HouseColor	Subject	SubjectCost	Grade
19594332X	Mary Watson	10 Charles Street	Bob	Red	English Maths Info Tech	$50 $50 $100	B A B+

How can you make it 1NF?

Create new rows so each cell contains only one value



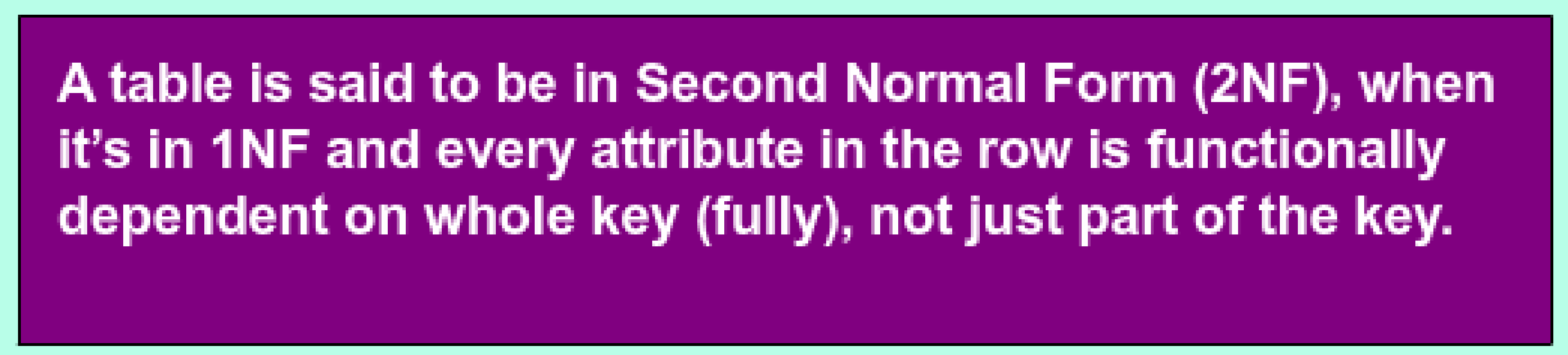
StudentID	StudentName	Address	HouseName	HouseColor	Subject	SubjectCost	Grade
19594332X	Mary Watson	10 Charles Street	Bob	Red	English Maths Info Tech	$50 $50 $100	B A B+



StudentID	StudentName	Address	HouseName	HouseColor	Subject	SubjectCost	Grade
19594332X	Mary Watson	10 Charles Street	Bob	Red	English	$50	B
19594332X	Mary Watson	10 Charles Street	Bob	Red	Maths	$50	A
19594332X	Mary Watson	10 Charles Street	Bob	Red	Info Tech	$100	B+

2NF

- Rule



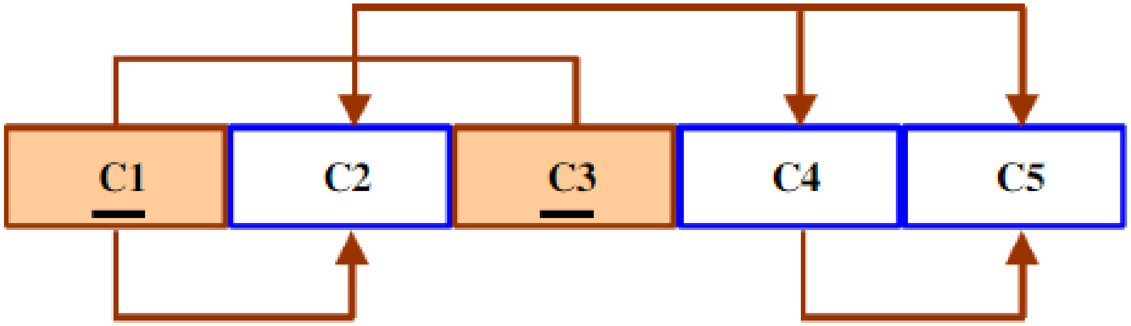
A table is said to be in Second Normal Form (2NF), when it's in 1NF and every attribute in the row is functionally dependent on whole key (fully), not just part of the key.

Make new tables

- Make a **new table for each primary key** field
- Move columns from the original table to the new table that matches their primary key
  - Table with keys

    (OR)
  - Table with keys + other common attributes

2NF



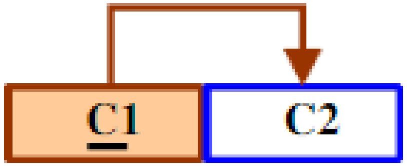


Table 1

Primary key: C1
Foreign key: None
Normal form: 3NF

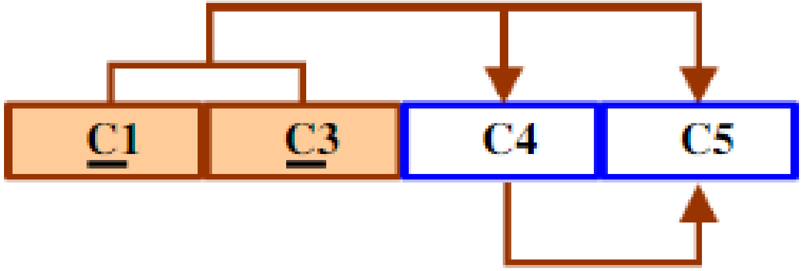
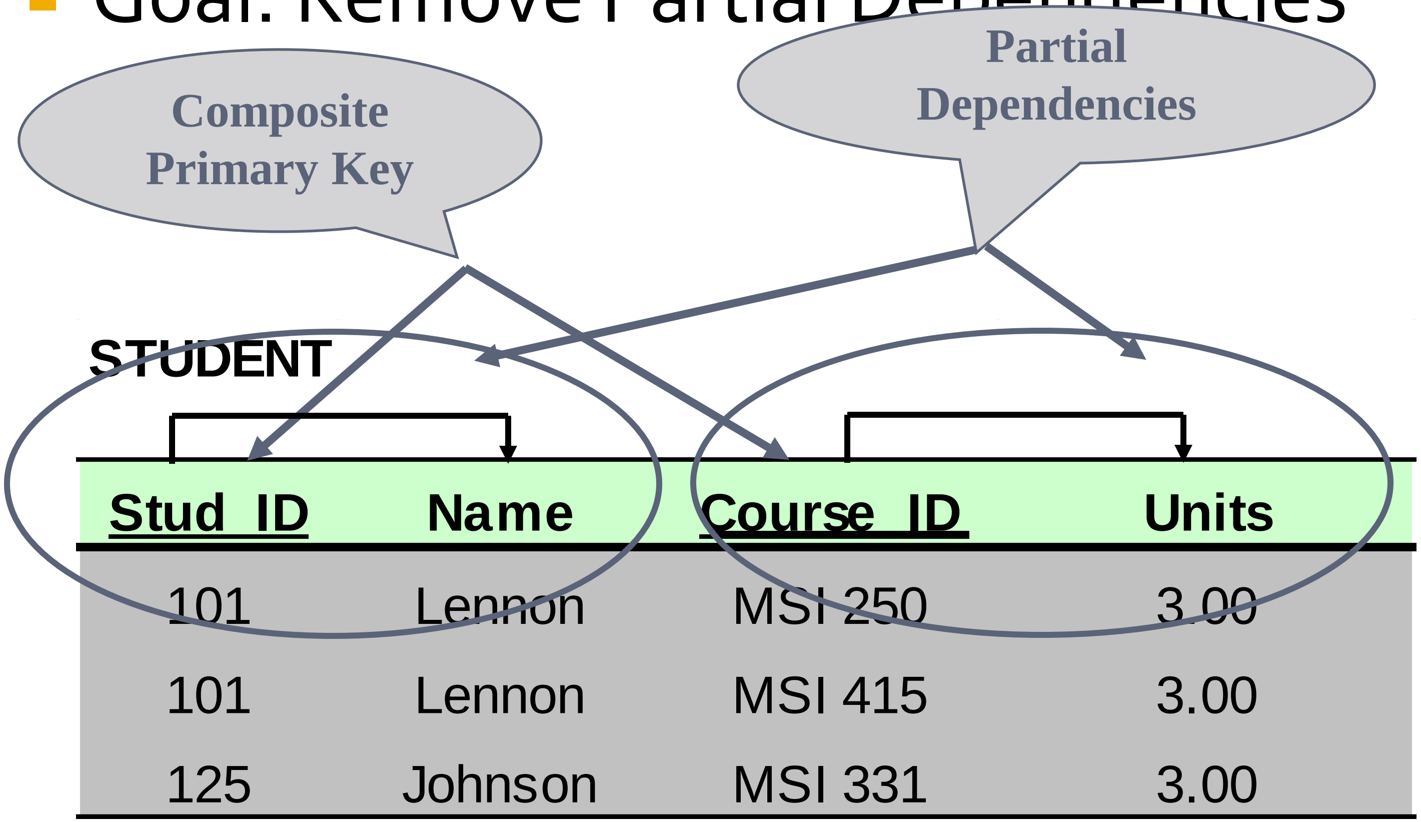


Table 2

Primary key: C1 + C3
Foreign key: C1 (to Table 1)
Normal form: 2NF, because the table exhibits the transitive dependencies C4 → C5

Bringing a Relation to 2NF – Example 2

- Goal: Remove Partial Dependencies



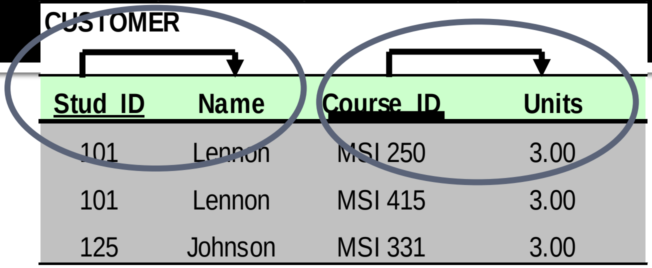
Composite Primary Key

Partial Dependencies

STUDENT

<u>Stud_ID</u>	Name	<u>Course_ID</u>	Units
101	Lennon	MSI 250	3.00
101	Lennon	MSI 415	3.00
125	Johnson	MSI 331	3.00

Bringing a Relation to 2NF



CUSTOMER

Stud_ID	Name	Course_ID	Units
101	Lennon	MSI 250	3.00
101	Lennon	MSI 415	3.00
125	Johnson	MSI 331	3.00

STUDENT_COURSE

Stud_ID	Course_ID
101	MSI 250
101	MSI 415
125	MSI 331

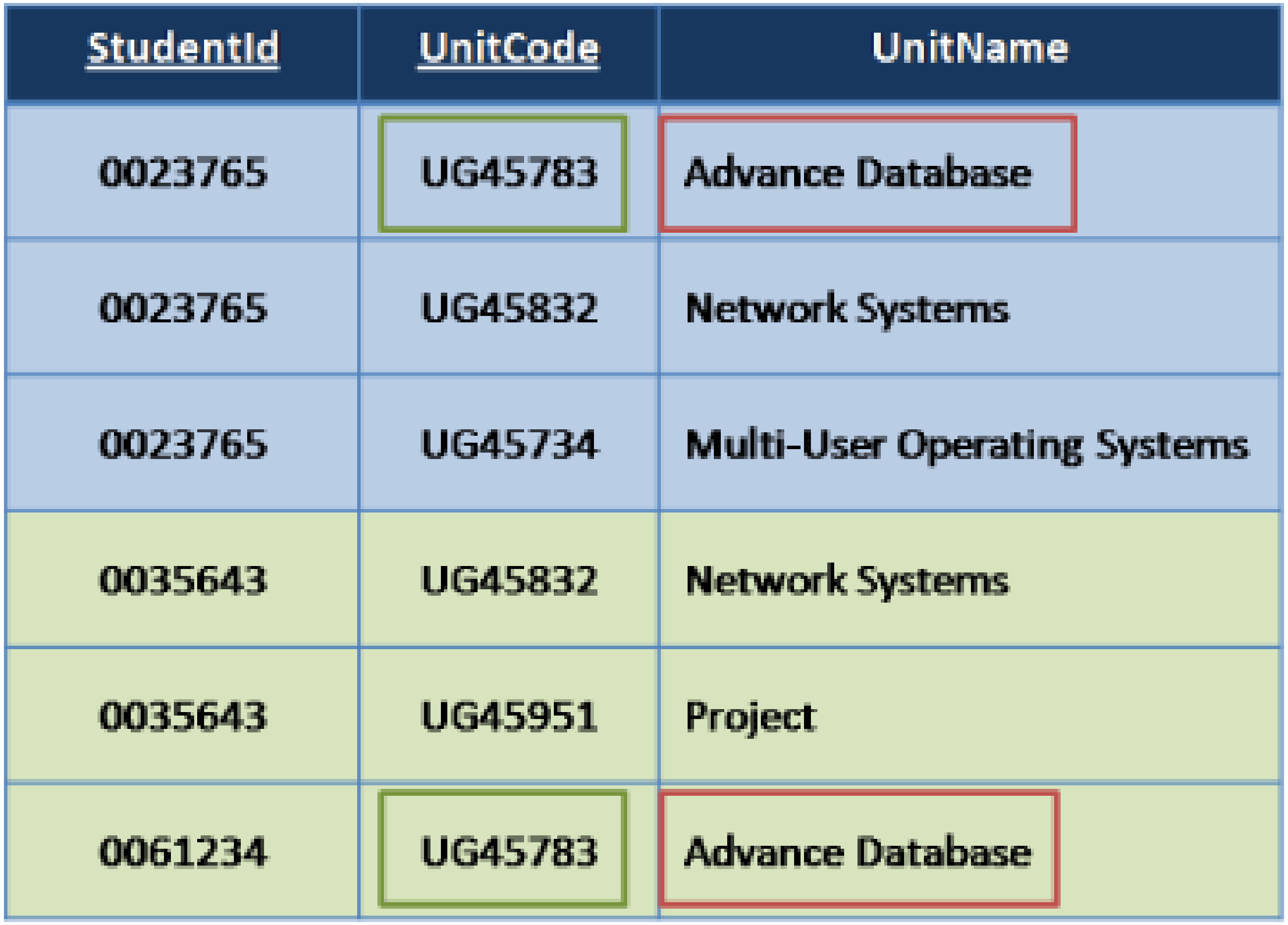
STUDENT

Stud_ID	Name
101	Lennon
101	Lennon
125	Johnson

COURSE

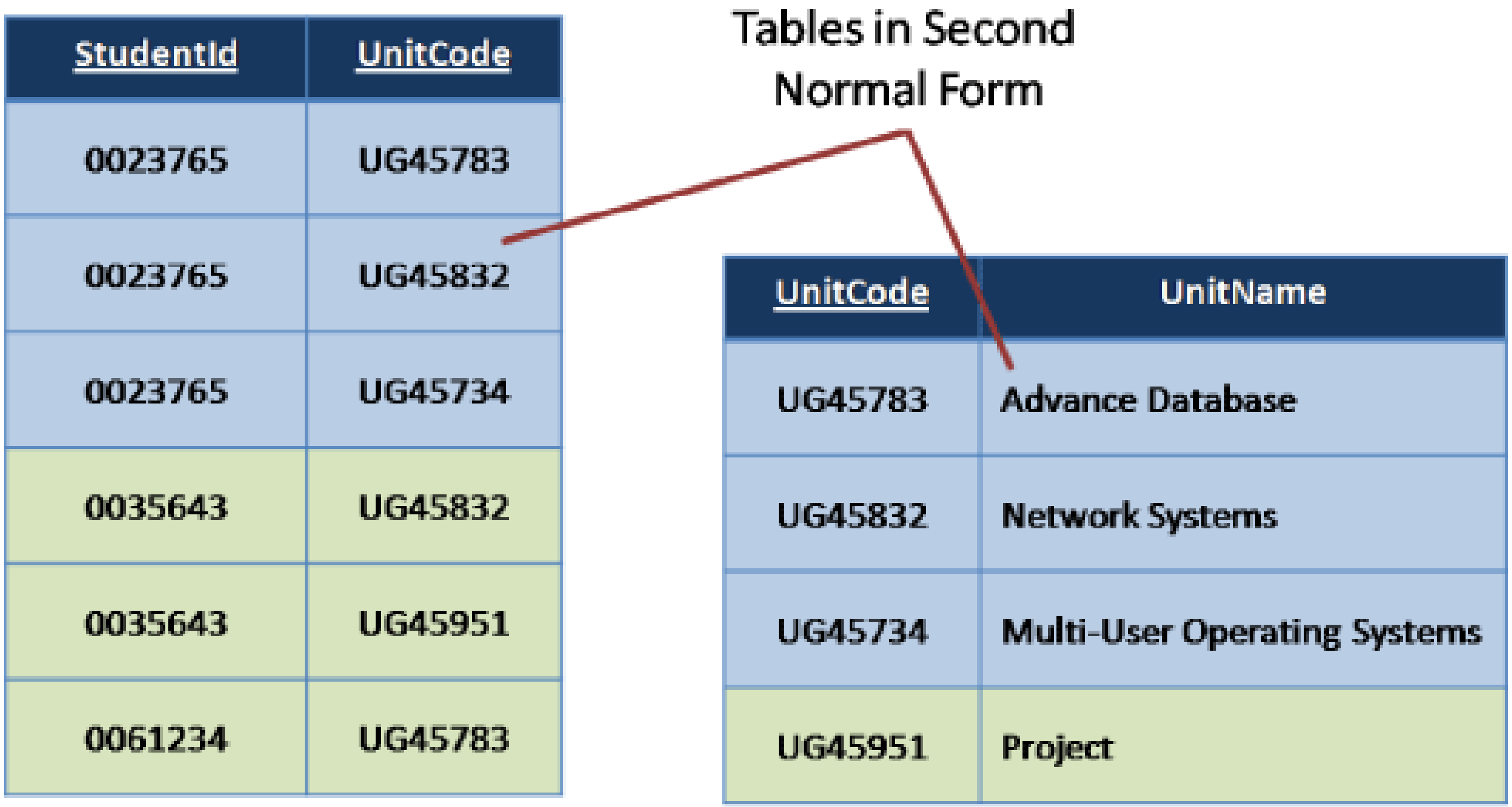
Course_ID	Units
MSI 250	3.00
MSI 415	3.00
MSI 331	3.00

2NF – Example 3



StudentId	UnitCode	UnitName
0023765	UG45783	Advance Database
0023765	UG45832	Network Systems
0023765	UG45734	Multi-User Operating Systems
0035643	UG45832	Network Systems
0035643	UG45951	Project
0061234	UG45783	Advance Database

2NF

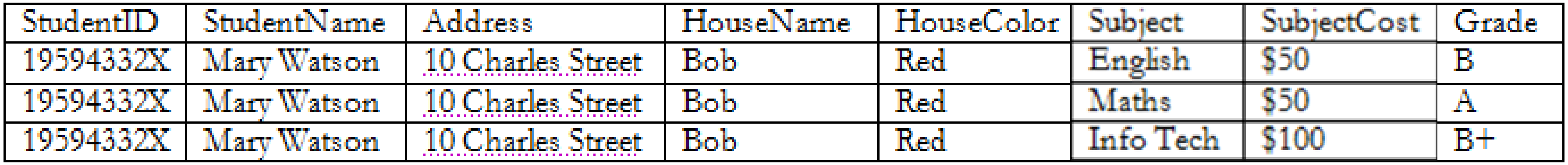


Tables in Second Normal Form

StudentId	UnitCode
0023765	UG45783
0023765	UG45832
0023765	UG45734
0035643	UG45832
0035643	UG45951
0061234	UG45783

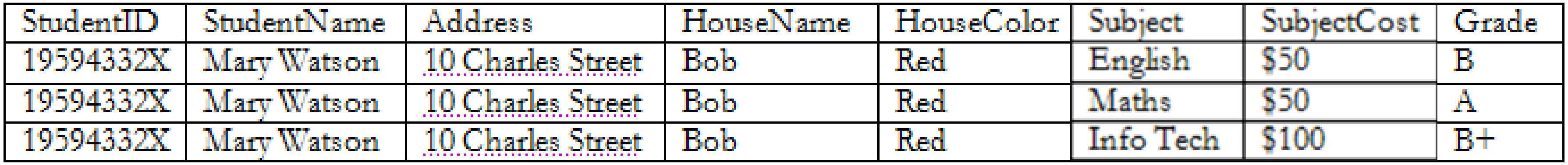
UnitCode	UnitName
UG45783	Advance Database
UG45832	Network Systems
UG45734	Multi-User Operating Systems
UG45951	Project

2NF – Example 4



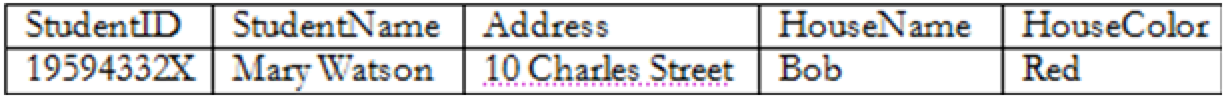
StudentID	StudentName	Address	HouseName	HouseColor	Subject	SubjectCost	Grade
19594332X	Mary Watson	10 Charles Street	Bob	Red	English	$50	B
19594332X	Mary Watson	10 Charles Street	Bob	Red	Maths	$50	A
19594332X	Mary Watson	10 Charles Street	Bob	Red	Info Tech	$100	B+

Step1



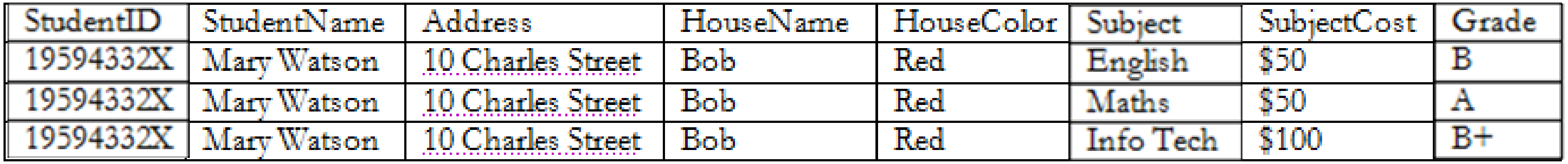
StudentID	StudentName	Address	HouseName	HouseColor	Subject	SubjectCost	Grade
19594332X	Mary Watson	10 Charles Street	Bob	Red	English	$50	B
19594332X	Mary Watson	10 Charles Street	Bob	Red	Maths	$50	A
19594332X	Mary Watson	10 Charles Street	Bob	Red	Info Tech	$100	B+

STUDENT TABLE (key = StudentID)



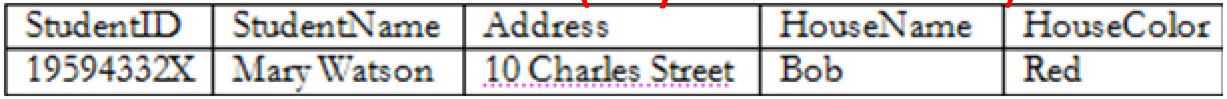
StudentID	StudentName	Address	HouseName	HouseColor
19594332X	Mary Watson	10 Charles Street	Bob	Red

Step 1



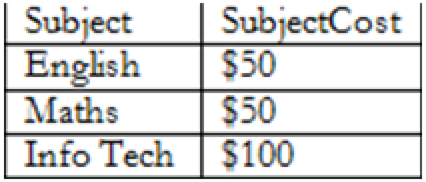
StudentID	StudentName	Address	HouseName	HouseColor	Subject	SubjectCost	Grade
19594332X	Mary Watson	10 Charles Street	Bob	Red	English	$50	B
19594332X	Mary Watson	10 Charles Street	Bob	Red	Maths	$50	A
19594332X	Mary Watson	10 Charles Street	Bob	Red	Info Tech	$100	B+

STUDENT TABLE (key = StudentID)



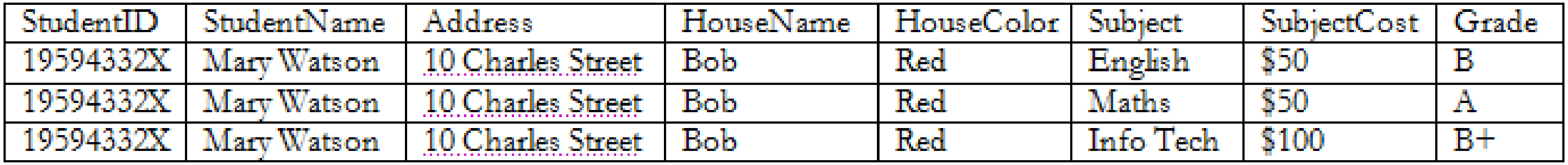
StudentID	StudentName	Address	HouseName	HouseColor
19594332X	Mary Watson	10 Charles Street	Bob	Red

SUBJECTS TABLE (key = Subject)



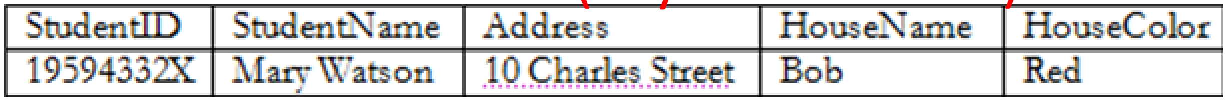
Subject	SubjectCost
English	$50
Maths	$50
Info Tech	$100

Step 2



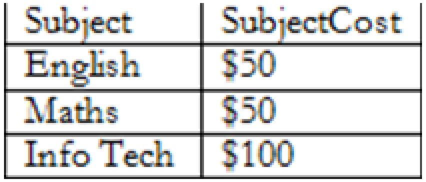
StudentID	StudentName	Address	HouseName	HouseColor	Subject	SubjectCost	Grade
19594332X	Mary Watson	10 Charles Street	Bob	Red	English	$50	B
19594332X	Mary Watson	10 Charles Street	Bob	Red	Maths	$50	A
19594332X	Mary Watson	10 Charles Street	Bob	Red	Info Tech	$100	B+

STUDENT TABLE (key = StudentID)



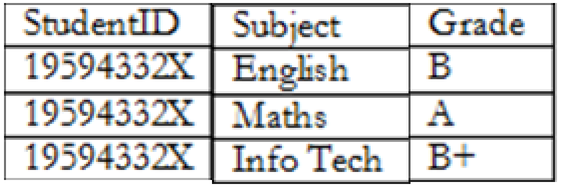
StudentID	StudentName	Address	HouseName	HouseColor
19594332X	Mary Watson	10 Charles Street	Bob	Red

SUBJECTS TABLE (key = Subject)



Subject	SubjectCost
English	$50
Maths	$50
Info Tech	$100

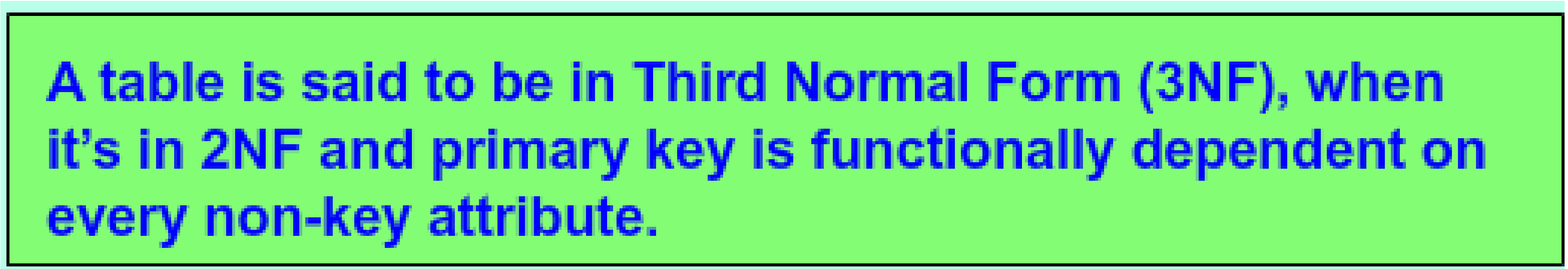
RESULTS TABLE (key = StudentID+Subject)



StudentID	Subject	Grade
19594332X	English	B
19594332X	Maths	A
19594332X	Info Tech	B+

3NF

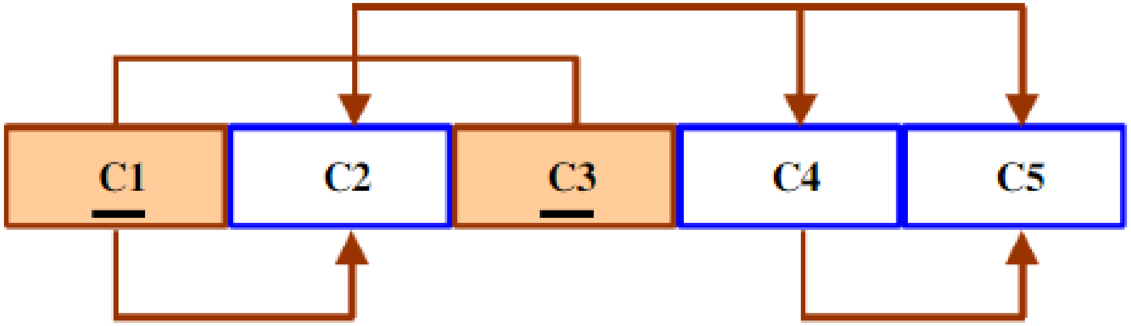
- Rule



A table is said to be in Third Normal Form (3NF), when it’s in 2NF and primary key is functionally dependent on every non-key attribute.

A **transitive dependency** is a type of functional dependency in which the value in a non-key field is determined by the value in another non-key field and that field is not a candidate key.

3NF



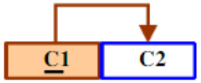


Table 1

Primary key: C1
Foreign key: None
Normal form: 3NF

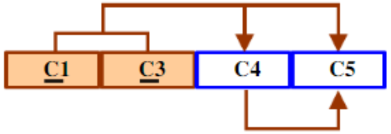


Table 2

Primary key: C1 + C3
Foreign key: C1 (to Table 1)
Normal form: 2NF, because the table exhibits the transitive dependencies C4 ⟶ C5

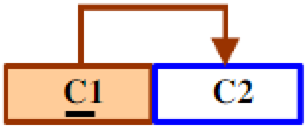


Table 1
Primary key: C1
Foreign key: None
Normal form: 3NF

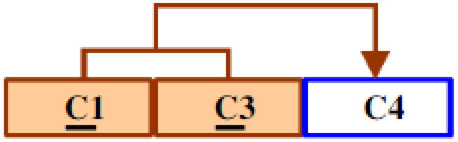


Table 2
Primary key: C1 + C3
Foreign key: C1 (to Table 1)
C4 (to Table 3)
Normal form: 3NF

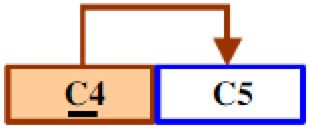
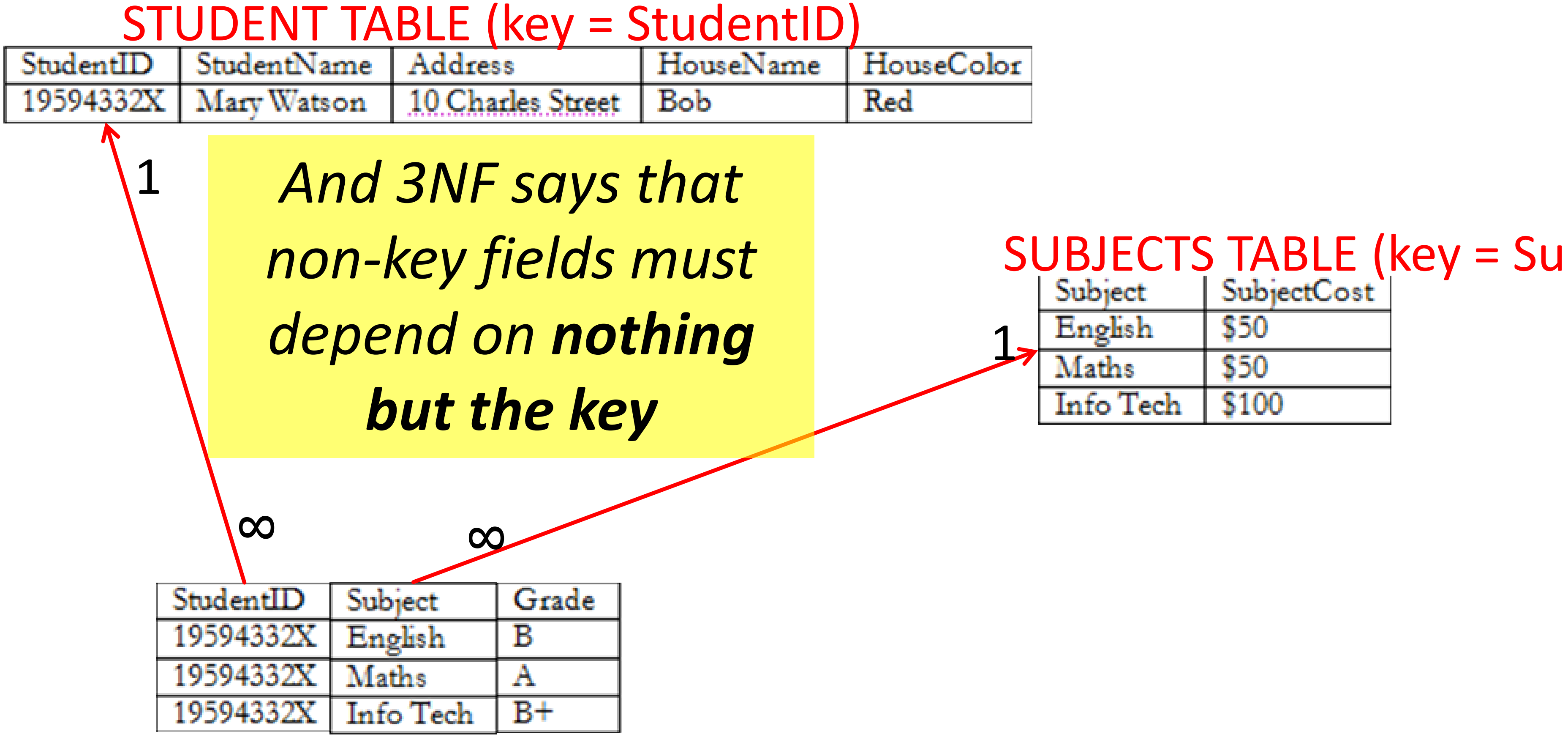


Table 3
Primary key: C4
Foreign key: None
Normal form: 3NF

3NF – Example 2



STUDENT TABLE (key = StudentID)

StudentID	StudentName	Address	HouseName	HouseColor
19594332X	Mary Watson	10 Charles Street	Bob	Red

And 3NF says that non-key fields must depend on ***nothing but the key***

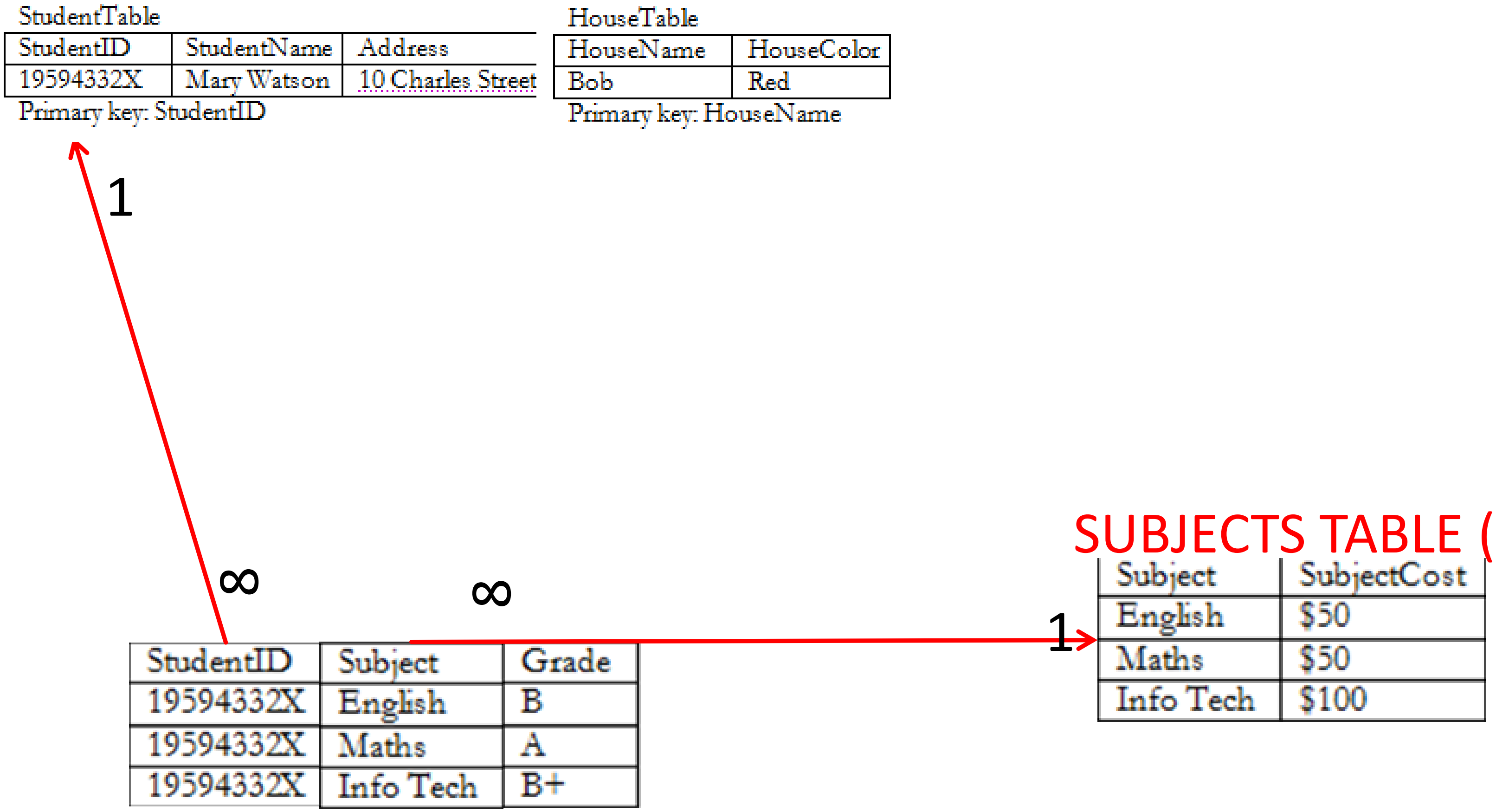
SUBJECTS TABLE (key = Su

Subject	SubjectCost
English	$50
Maths	$50
Info Tech	$100

StudentID	Subject	Grade
19594332X	English	B
19594332X	Maths	A
19594332X	Info Tech	B+

RESULTS TABLE (key = StudentID+Subject)

Again, carve off the offending fields



StudentTable

StudentID	StudentName	Address
19594332X	Mary Watson	10 Charles Street

Primary key: StudentID

HouseTable

HouseName	HouseColor
Bob	Red

Primary key: HouseName

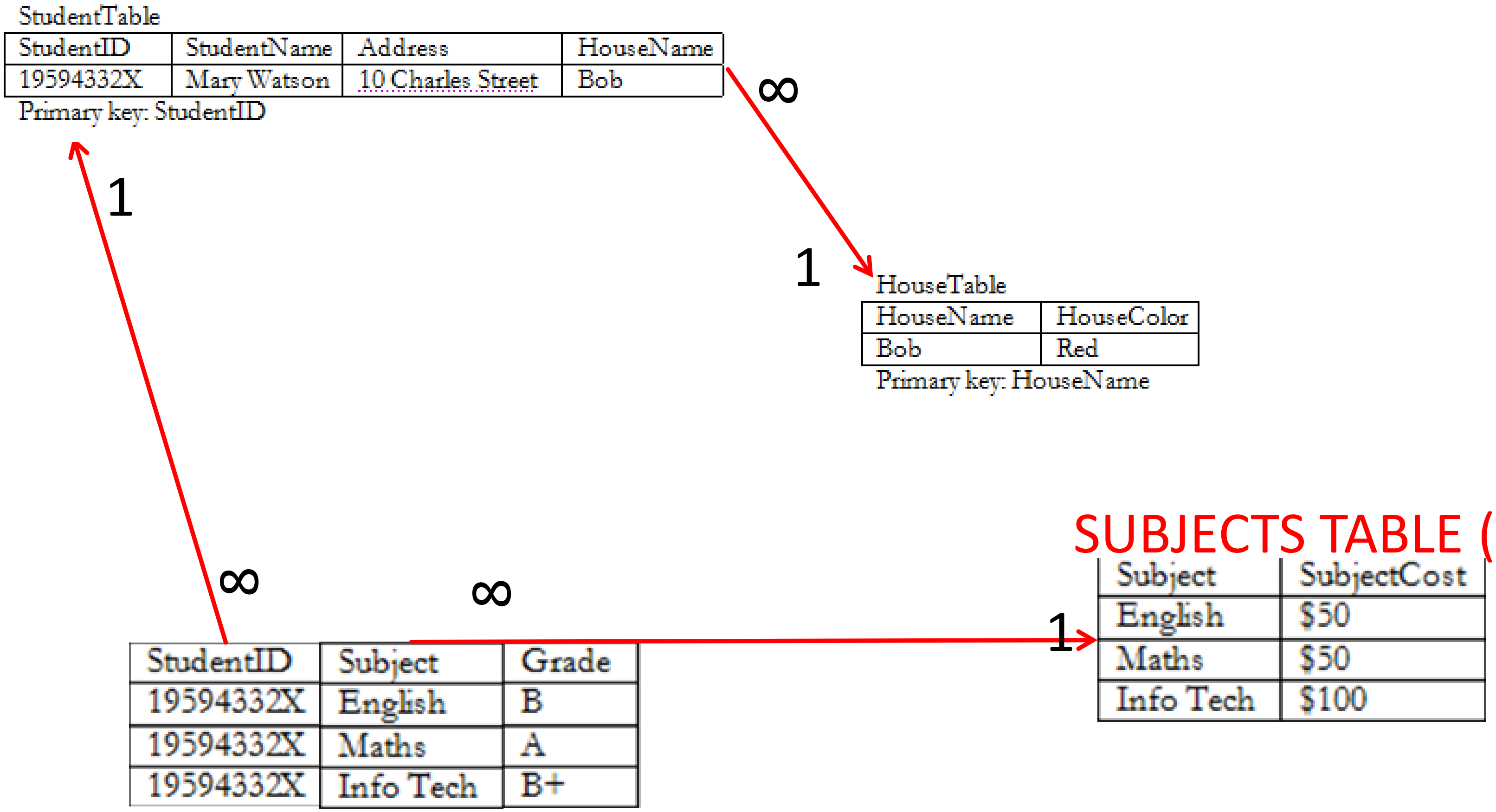
SUBJECTS TABLE (key = Su

Subject	SubjectCost
English	$50
Maths	$50
Info Tech	$100

StudentID	Subject	Grade
19594332X	English	B
19594332X	Maths	A
19594332X	Info Tech	B+

RESULTS TABLE (key = StudentID+Subject)

After 3NF



StudentTable

StudentID	StudentName	Address	HouseName
19594332X	Mary Watson	10 Charles Street	Bob

Primary key: StudentID

HouseTable

HouseName	HouseColor
Bob	Red

Primary key: HouseName

SUBJECTS TABLE (key = Su

Subject	SubjectCost
English	$50
Maths	$50
Info Tech	$100

StudentID	Subject	Grade
19594332X	English	B
19594332X	Maths	A
19594332X	Info Tech	B+

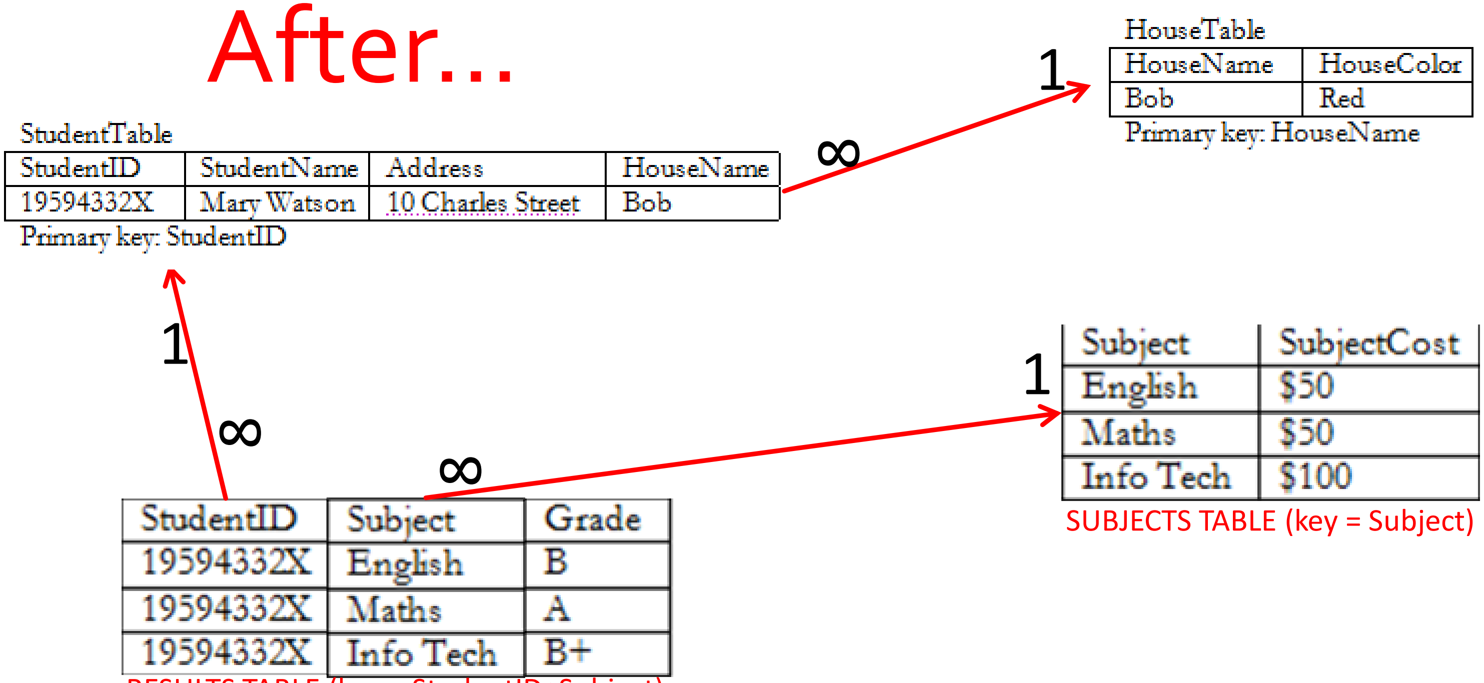
RESULTS TABLE (key = StudentID+Subject)

The Reveal

Before…

StudentID	StudentName	Address	HouseName	HouseColor	Subject	SubjectCost	Grade
19594332X	Mary Watson	10 Charles Street	Bob	Red	English Maths Info Tech	$50 $50 $100	B A B+

After…



StudentTable

StudentID	StudentName	Address	HouseName
19594332X	Mary Watson	10 Charles Street	Bob

Primary key: StudentID

HouseTable

HouseName	HouseColor
Bob	Red

Primary key: HouseName

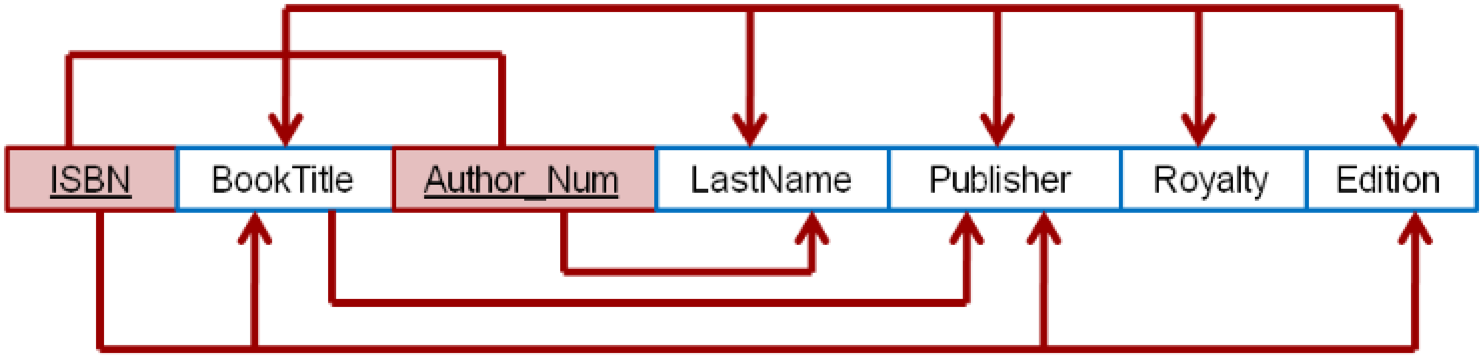
StudentID	Subject	Grade
19594332X	English	B
19594332X	Maths	A
19594332X	Info Tech	B+

RESULTS TABLE (key = StudentID+Subject)

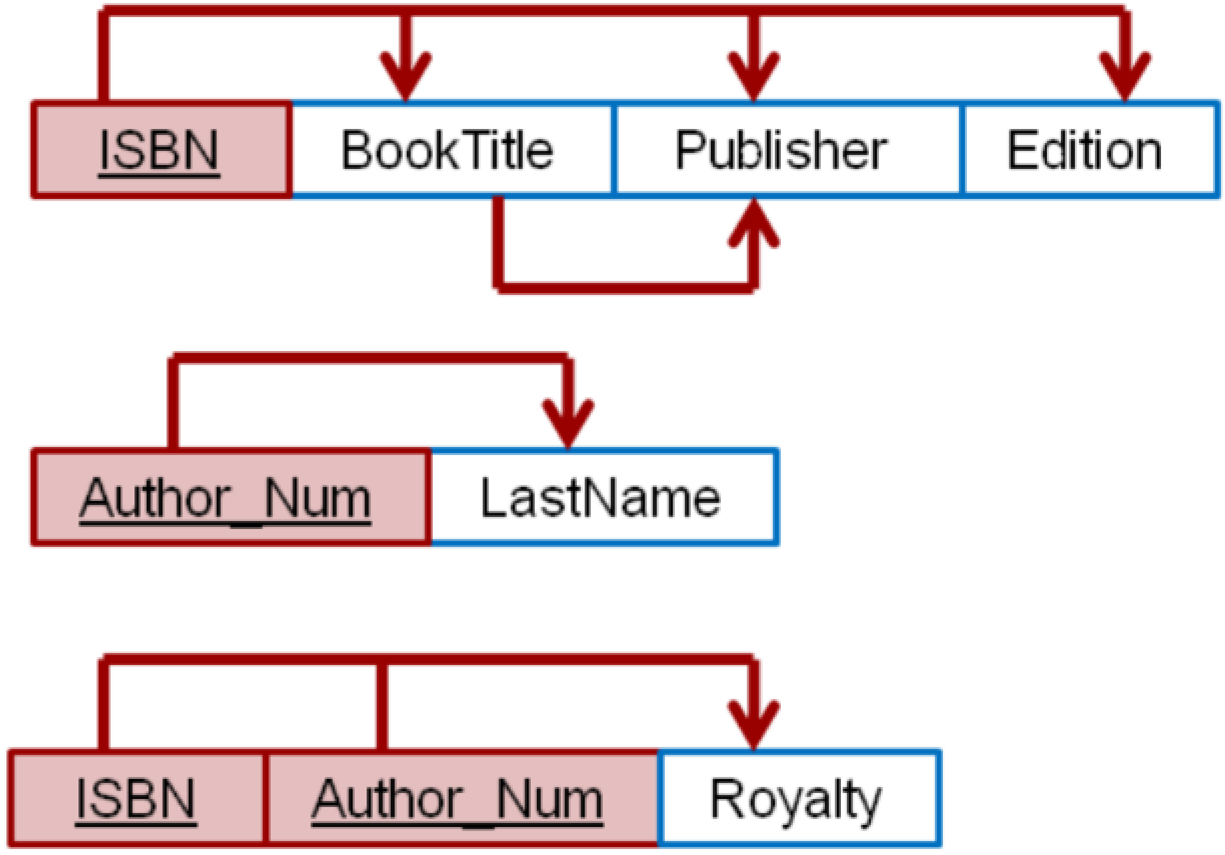
Subject	SubjectCost
English	$50
Maths	$50
Info Tech	$100

SUBJECTS TABLE (key = Subject)

Exercise -1


ISBN
BookTitle
Author_Num
LastName
Publisher
Royalty
Edition

2NF

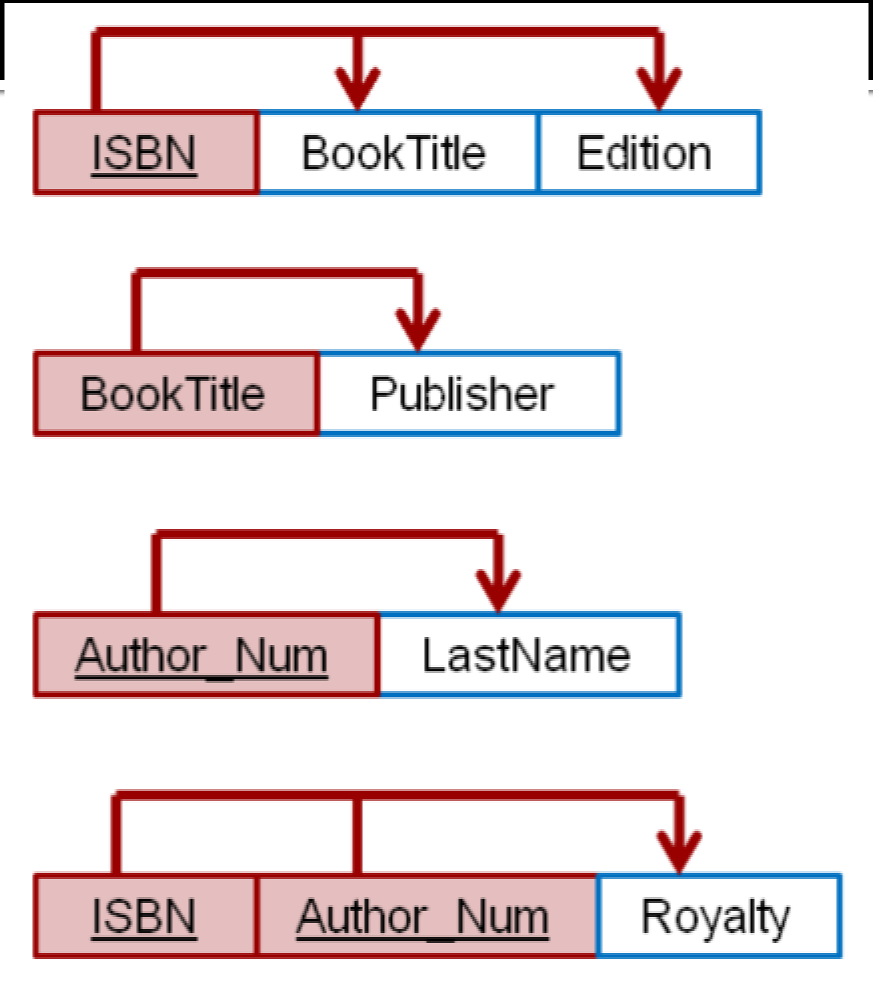


ISBN | BookTitle | Publisher | Edition

Author_Num | LastName

ISBN | Author_Num | Royalty

3NF



ISBN | BookTitle | Edition

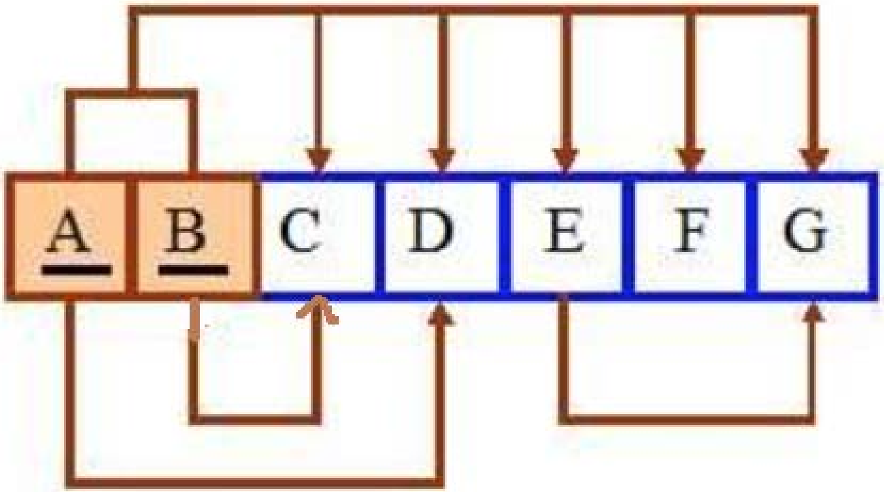
BookTitle | Publisher

Author_Num | LastName

ISBN | Author_Num | Royalty

Exercise 2

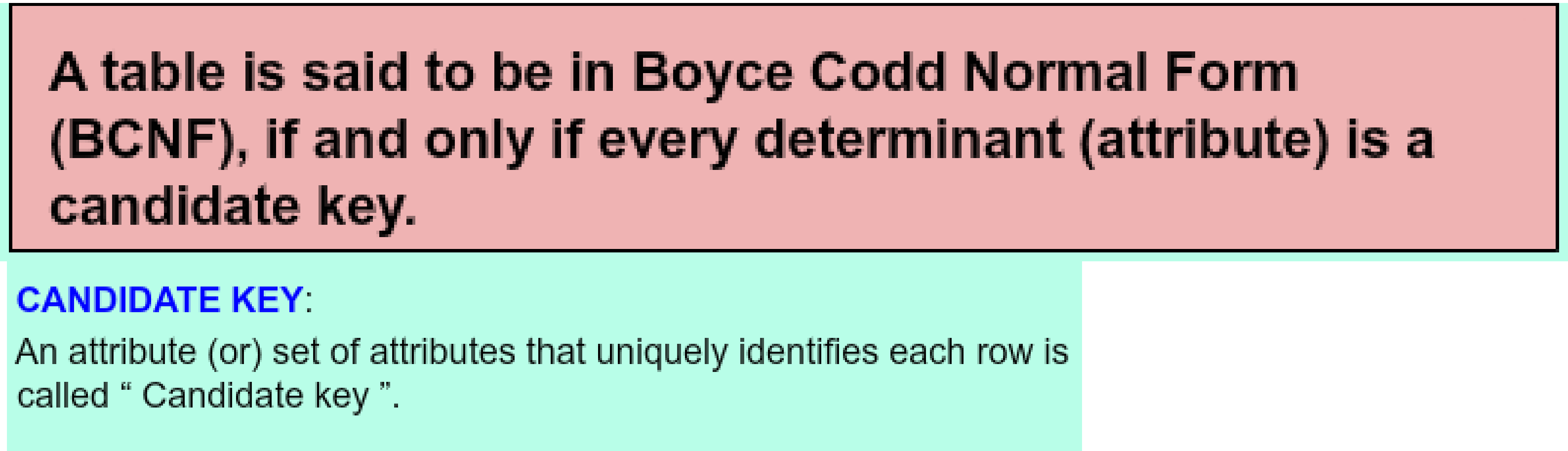
a) Given the following dependency diagram, label all the dependencies.



b) Redesign the database to 2NF. Show all the steps.
c) Redesign the database to 3NF. Show all the steps.

BCNF

- **Boyce Codd normal Form** (or **BCNF** or **3.5NF**)
- BCNF was developed in 1974 by Raymond F. Boyce and Edgar F. Codd to address certain types of anomalies not dealt with by 3NF as originally defined.[
- Rule



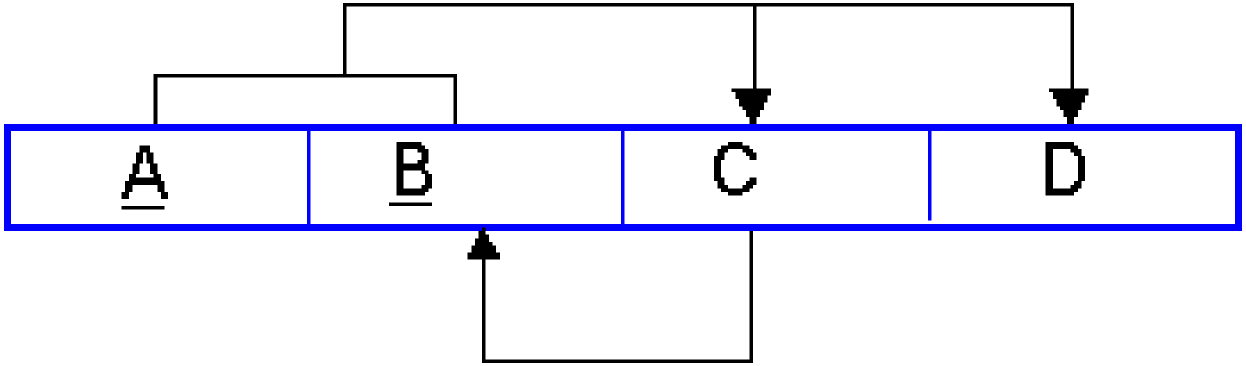
A table is said to be in Boyce Codd Normal Form (BCNF), if and only if every determinant (attribute) is a candidate key.

CANDIDATE KEY:

An attribute (or) set of attributes that uniquely identifies each row is called “ Candidate key ”.

BCNF

This occurs when a non key attribute is a determinant of a key attribute.

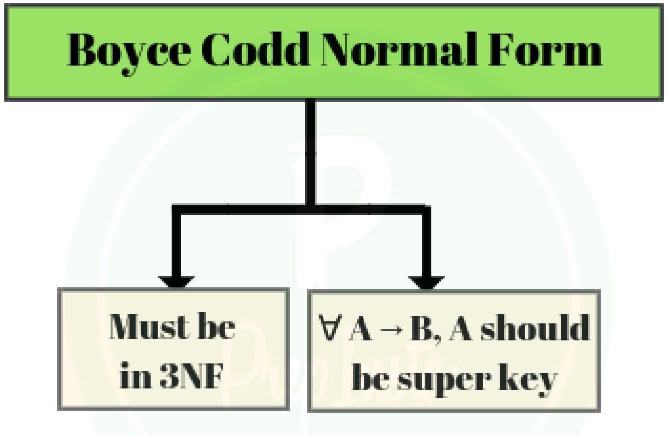


- C->B

BCNF

Even when a database is in 3rd Normal Form, still there would be anomalies resulted if it has more than one Candidate Key.

A table is in BCNF if every functional dependency A → B, A is the super key of the table.



BCNF

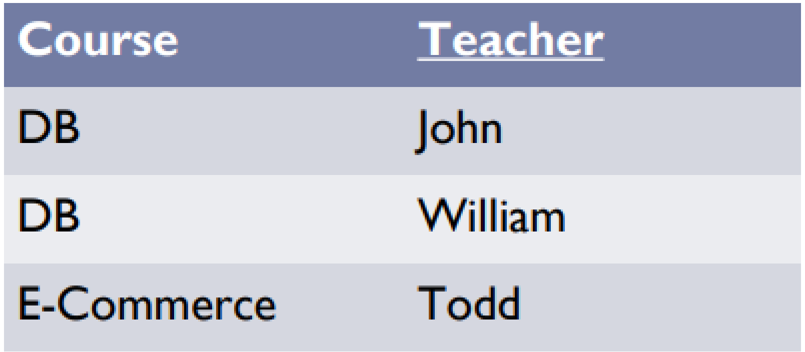
- Example of a table not in BCNF:

Student	Course	Teacher
Sok	DB	John
Sao	DB	William
Chan	E-Commerce	Todd
Sok	E-Commerce	Todd
Chan	DB	William

- Key: {Student, Course}
- Functional Dependency:
  - {Student, Course}→ Teacher
  - Teacher → Course
- Problem: ***Teacher*** is not a superkey but determines ***Course***.

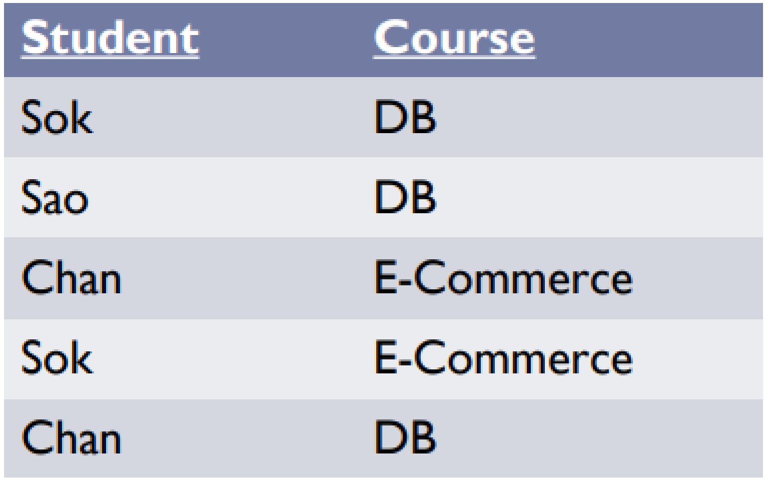
After BCNF

Table1



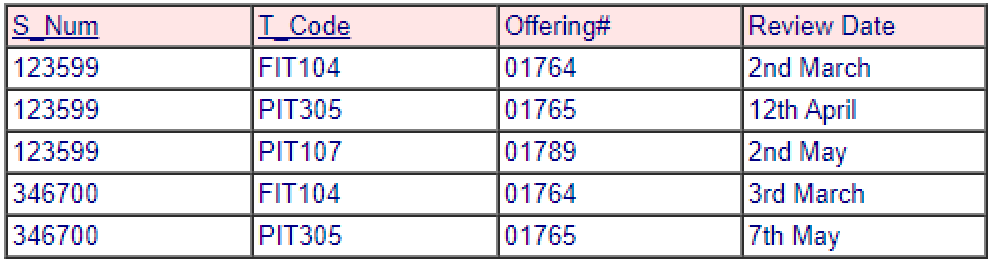
Course	Teacher
DB	John
DB	William
E-Commerce	Todd

Table2

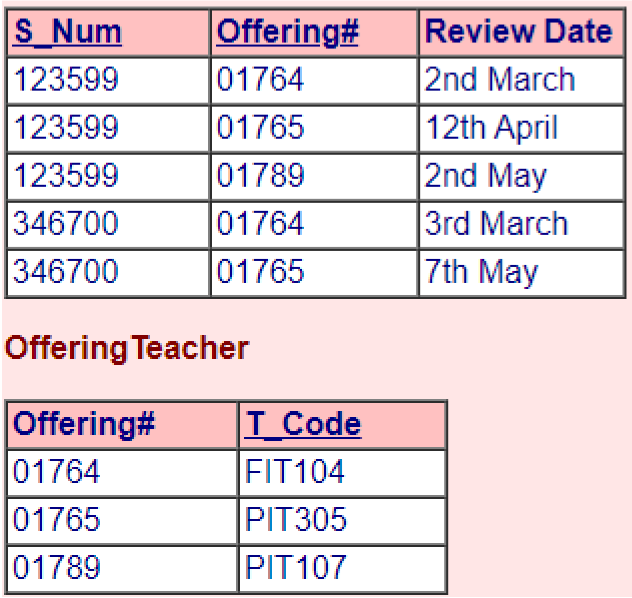


Student	Course
Sok	DB
Sao	DB
Chan	E-Commerce
Sok	E-Commerce
Chan	DB

After BCNF



S_Num	T_Code	Offering#	Review Date
123599	FIT104	01764	2nd March
123599	PIT305	01765	12th April
123599	PIT107	01789	2nd May
346700	FIT104	01764	3rd March
346700	PIT305	01765	7th May



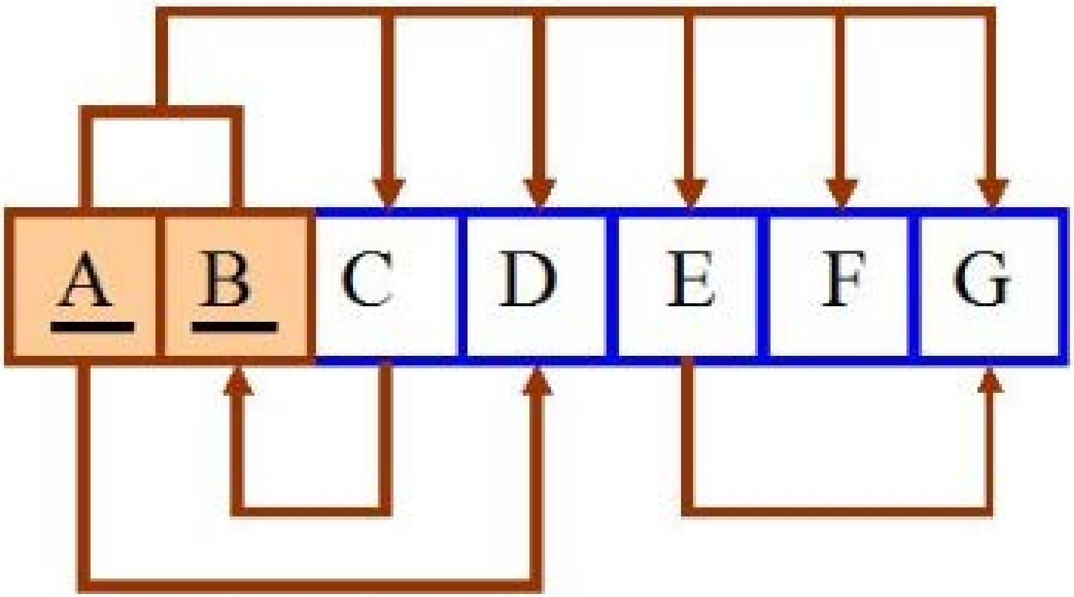
S_Num	Offering#	Review Date
123599	01764	2nd March
123599	01765	12th April
123599	01789	2nd May
346700	01764	3rd March
346700	01765	7th May

OfferingTeacher

Offering#	T_Code
01764	FIT104
01765	PIT305
01789	PIT107

Exercise 1

Initial Dependency Diagram for Problem



a) Break up the dependency diagram shown in Figure to create two new dependency diagrams: in 2NF.

b) Modify the dependency diagrams you created in part a to produce a set of dependency diagrams that are in 3NF. (*Hint:*One of your dependency diagrams should be in 3NF but not in BCNF.)

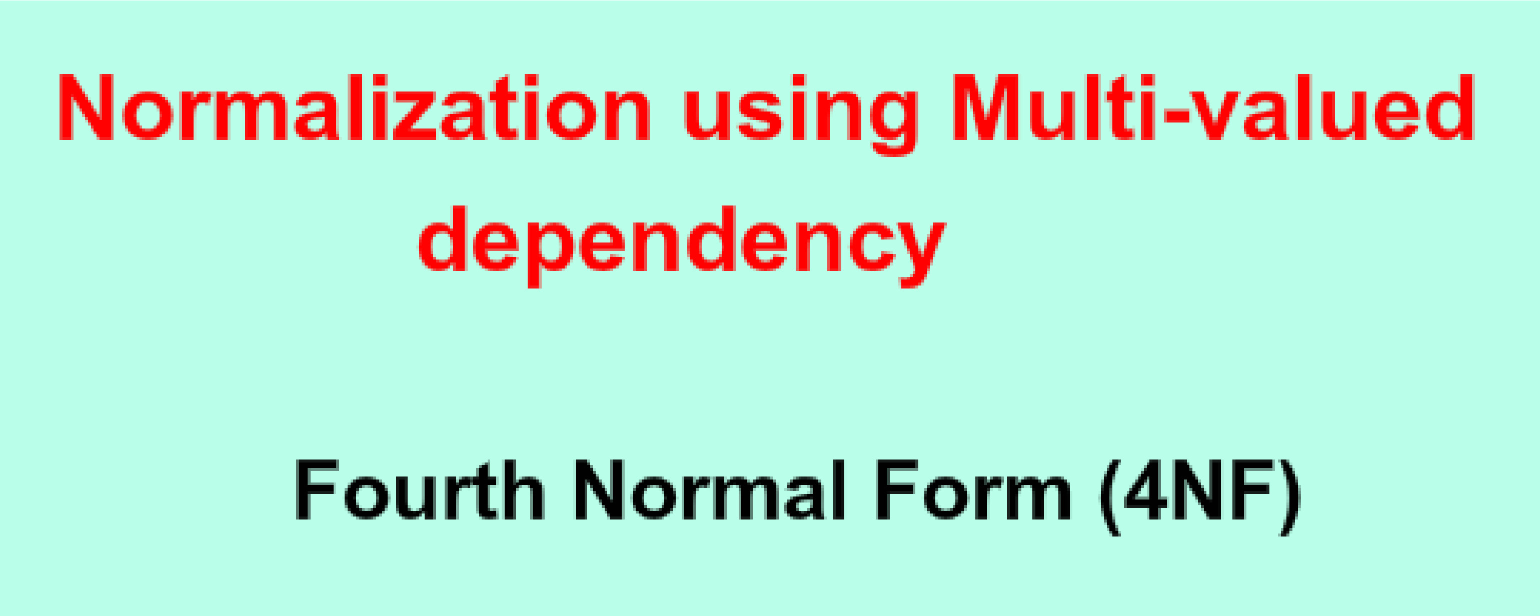
c) Modify the dependency diagrams you created in Part b to produce a collection of dependency diagrams that are in 3NF and BCNF.

Exercise 2

- Find the highest normal form of a relation R(A, B, C, D, E) with FD set as:
- { C->D, AC->BE, B->E }

Solution

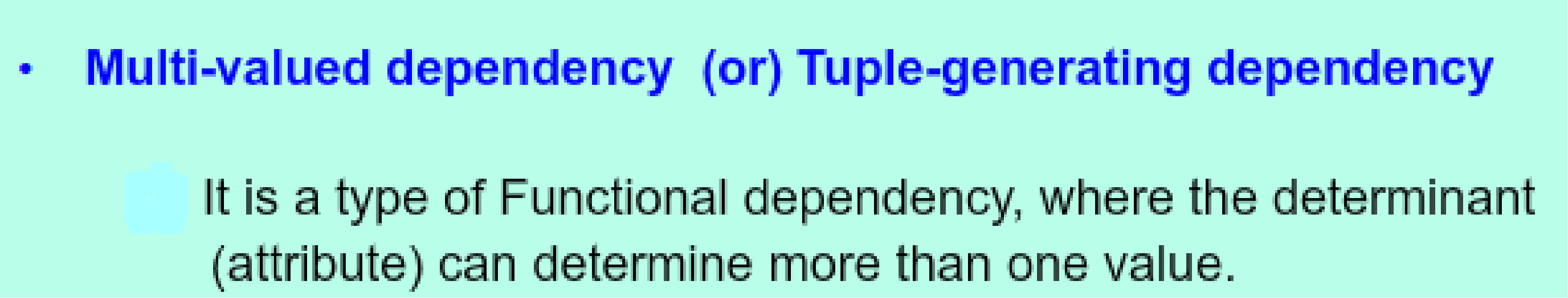
- (AC)+ ={A, C, B, E, D}
- So there will be only 1 candidate key {AC}



Normalization using Multi-valued dependency

Fourth Normal Form (4NF)

Multi-Valued Dependency

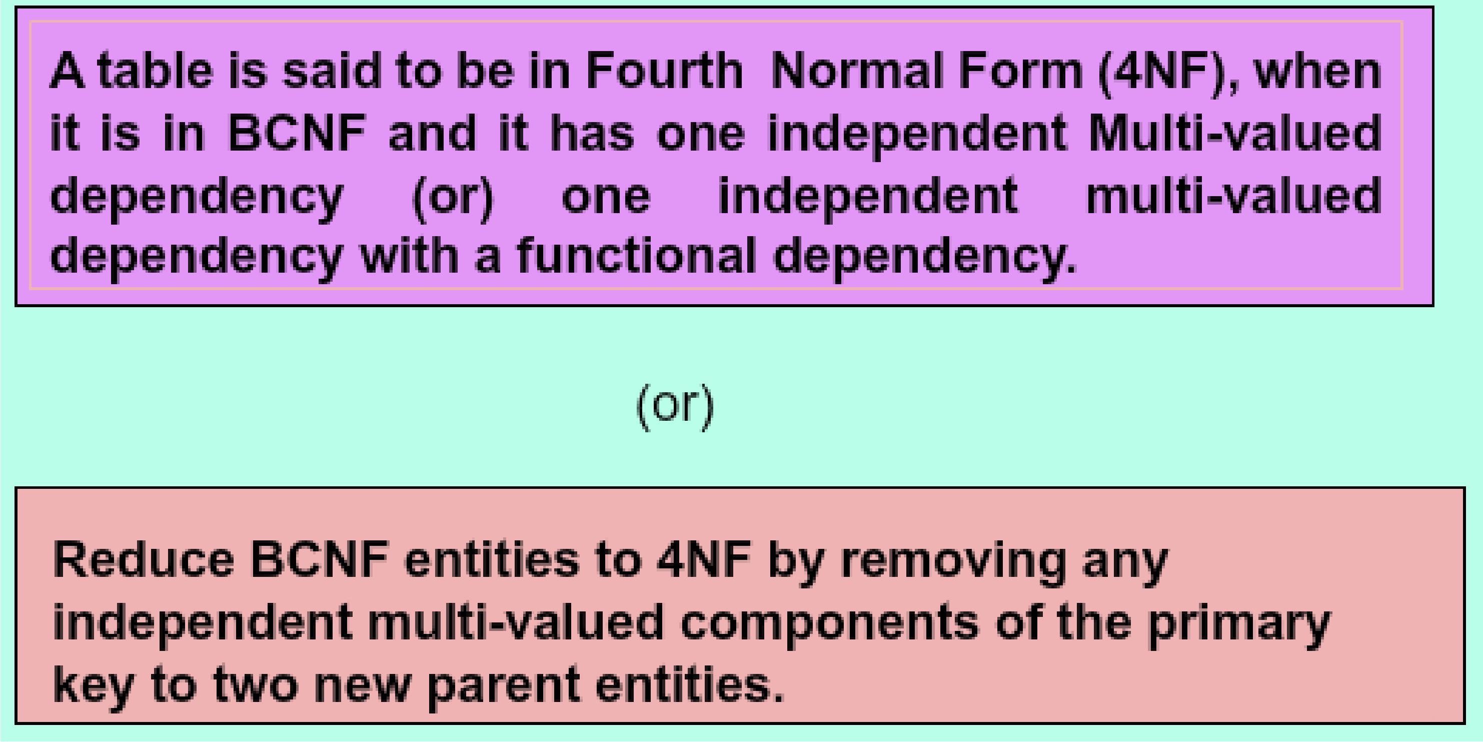


- **Multi-valued dependency (or) Tuple-generating dependency**

  - It is a type of Functional dependency, where the determinant (attribute) can determine more than one value.

4NF

- Rule

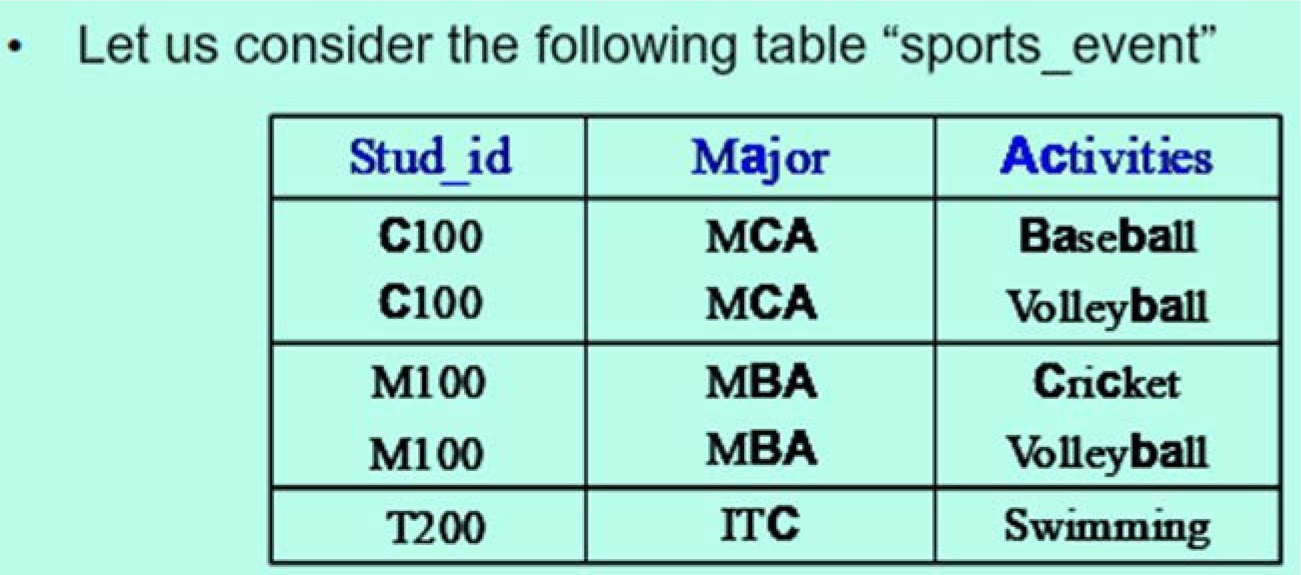


A table is said to be in Fourth Normal Form (4NF), when it is in BCNF and it has one independent Multi-valued dependency (or) one independent multi-valued dependency with a functional dependency.

(or)

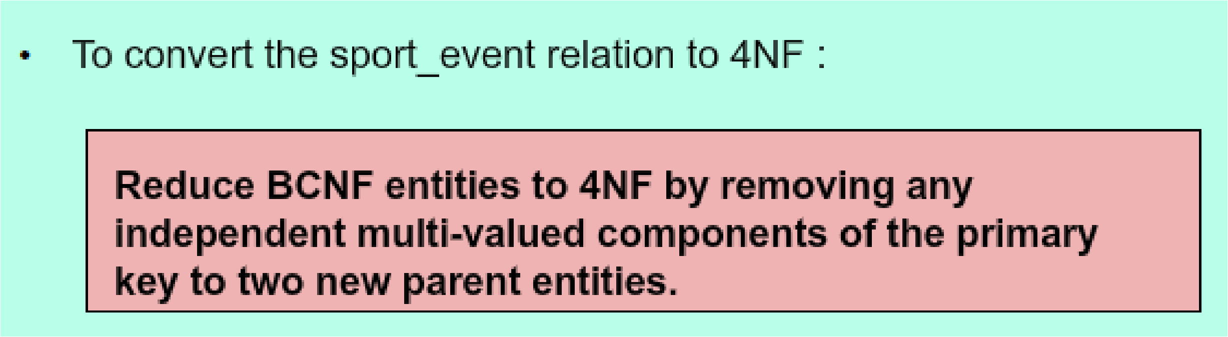
Reduce BCNF entities to 4NF by removing any independent multi-valued components of the primary key to two new parent entities.

Example



- Let us consider the following table “sports_event”

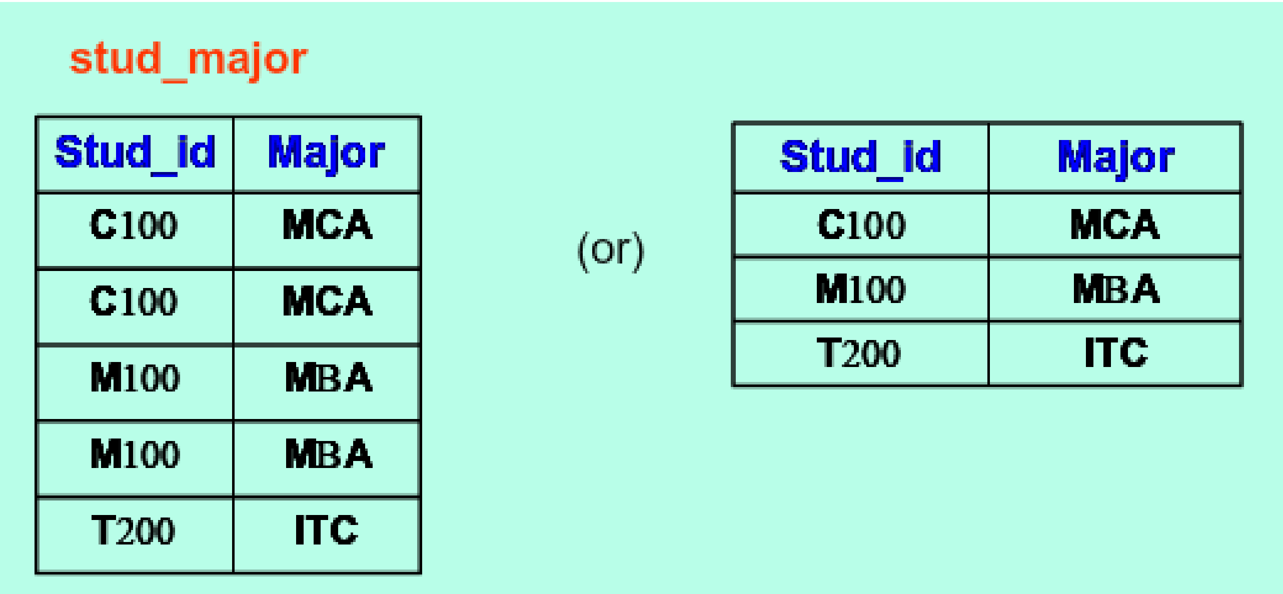
Stud_id	Major	Activities
C100	MCA	Baseball
C100	MCA	Volleyball
M100	MBA	Cricket
M100	MBA	Volleyball
T200	ITC	Swimming



- To convert the sport_event relation to 4NF :

Reduce BCNF entities to 4NF by removing any independent multi-valued components of the primary key to two new parent entities.

4NF

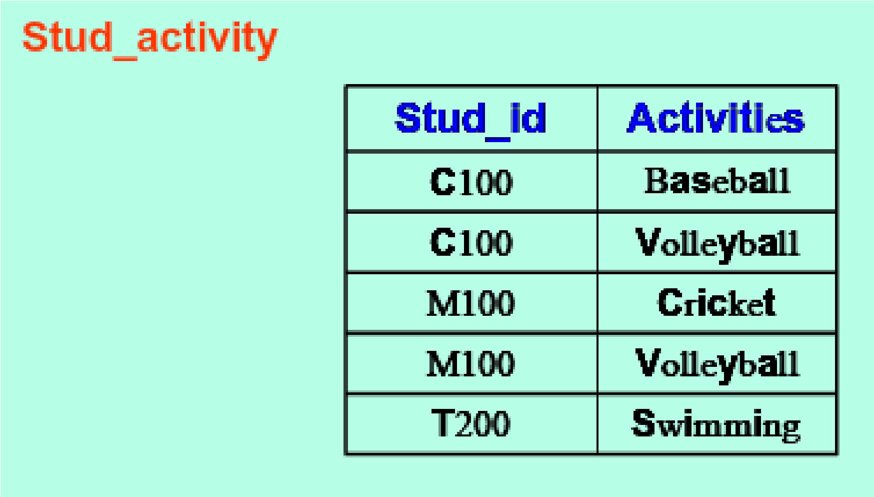


stud_major

Stud_id	Major
C100	MCA
C100	MCA
M100	MBA
M100	MBA
T200	ITC

(or)

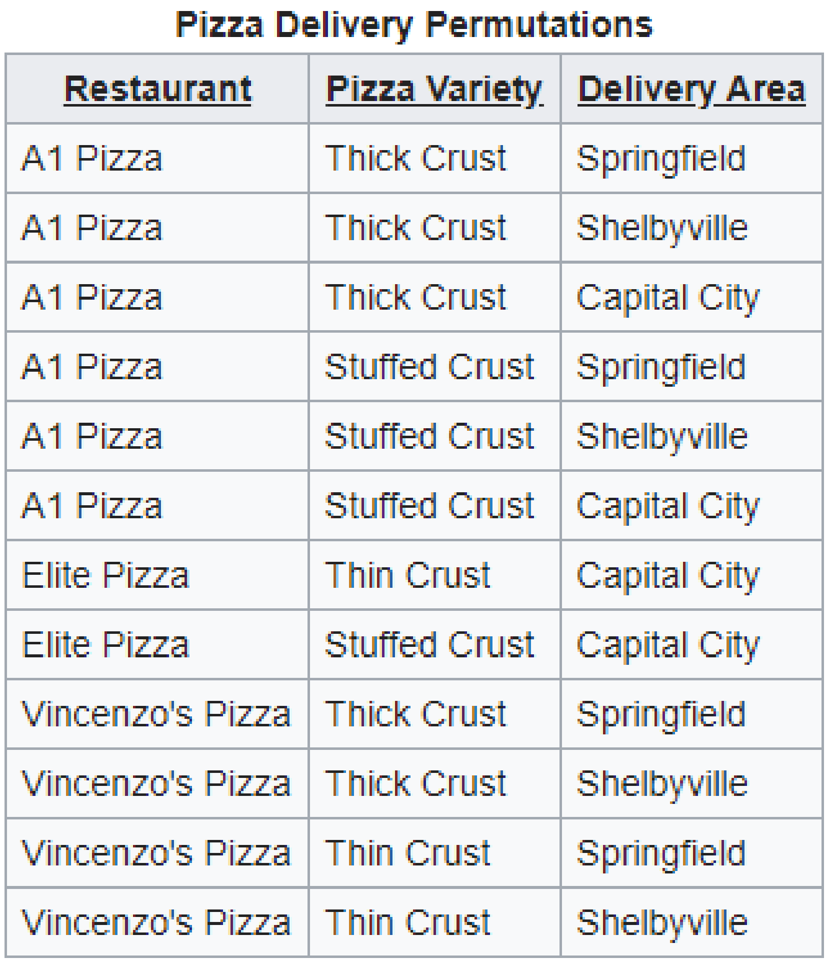
Stud_id	Major
C100	MCA
M100	MBA
T200	ITC



Stud_activity

Stud_id	Activities
C100	Baseball
C100	Volleyball
M100	Cricket
M100	Volleyball
T200	Swimming

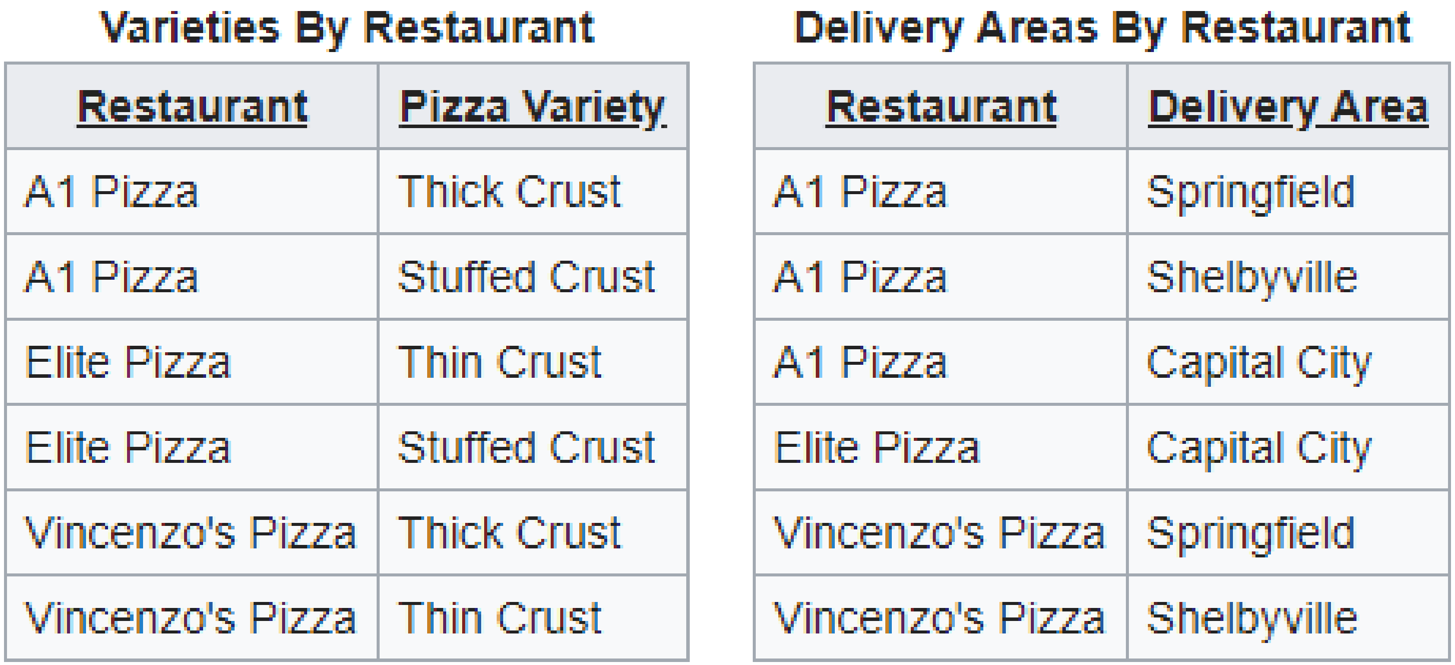
4NF – Example 2



Pizza Delivery Permutations

Restaurant	Pizza Variety	Delivery Area
A1 Pizza	Thick Crust	Springfield
A1 Pizza	Thick Crust	Shelbyville
A1 Pizza	Thick Crust	Capital City
A1 Pizza	Stuffed Crust	Springfield
A1 Pizza	Stuffed Crust	Shelbyville
A1 Pizza	Stuffed Crust	Capital City
Elite Pizza	Thin Crust	Capital City
Elite Pizza	Stuffed Crust	Capital City
Vincenzo's Pizza	Thick Crust	Springfield
Vincenzo's Pizza	Thick Crust	Shelbyville
Vincenzo's Pizza	Thin Crust	Springfield
Vincenzo's Pizza	Thin Crust	Shelbyville

4NF

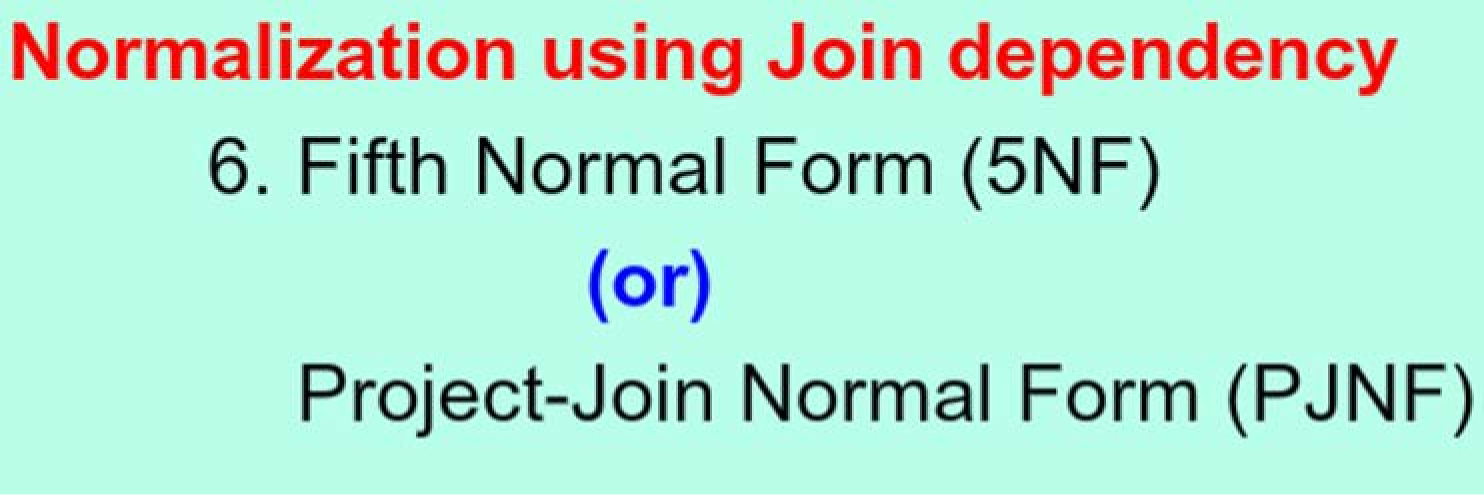


Varieties By Restaurant

Restaurant	Pizza Variety
A1 Pizza	Thick Crust
A1 Pizza	Stuffed Crust
Elite Pizza	Thin Crust
Elite Pizza	Stuffed Crust
Vincenzo's Pizza	Thick Crust
Vincenzo's Pizza	Thin Crust

Delivery Areas By Restaurant

Restaurant	Delivery Area
A1 Pizza	Springfield
A1 Pizza	Shelbyville
A1 Pizza	Capital City
Elite Pizza	Capital City
Vincenzo's Pizza	Springfield
Vincenzo's Pizza	Shelbyville



Normalization using Join dependency

6. Fifth Normal Form (5NF)

(or)

Project-Join Normal Form (PJNF)

Example of Lossless-Join Decomposition

- **Lossless join decomposition**
- Decomposition of $R = (A, B, C)$

$$R_1 = (A, B) \qquad R_2 = (B, C)$$

A	B	C
α	1	A
β	2	B

r

A	B
α	1
β	2

$\Pi_{A,B}(r)$

B	C
1	A
2	B

$\Pi_{B,C}(r)$

$\Pi_A (r) \bowtie \Pi_B (r)$

A	B	C
α	1	A
β	2	B

5NF or PJNF

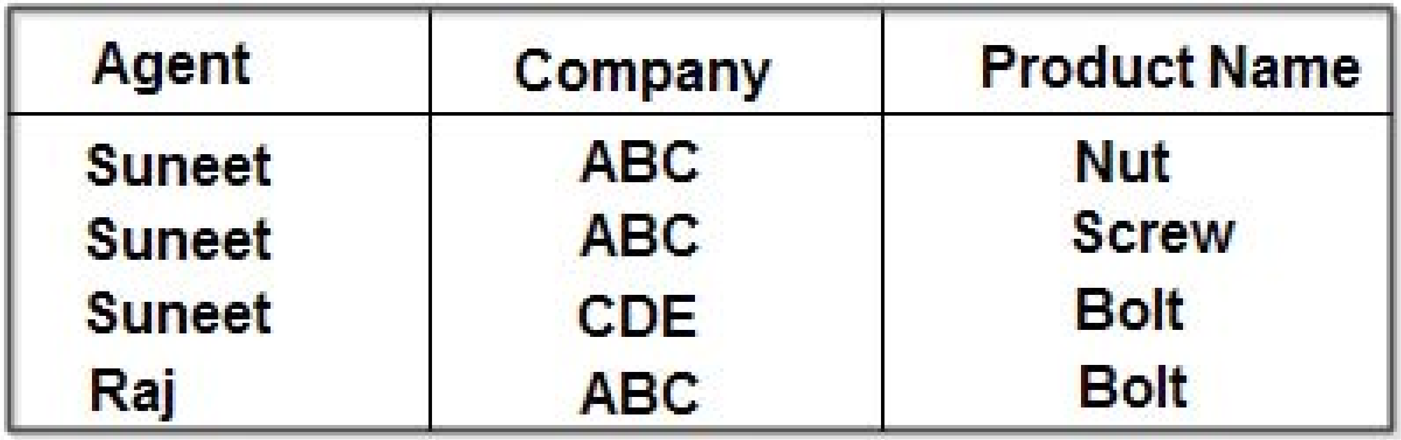
- Rule

A table is in fifth normal form (5NF) or Project-Join Normal Form (PJNF) if it is in 4NF and it cannot have a lossless decomposition into any number of smaller tables.

5NF or PJNF

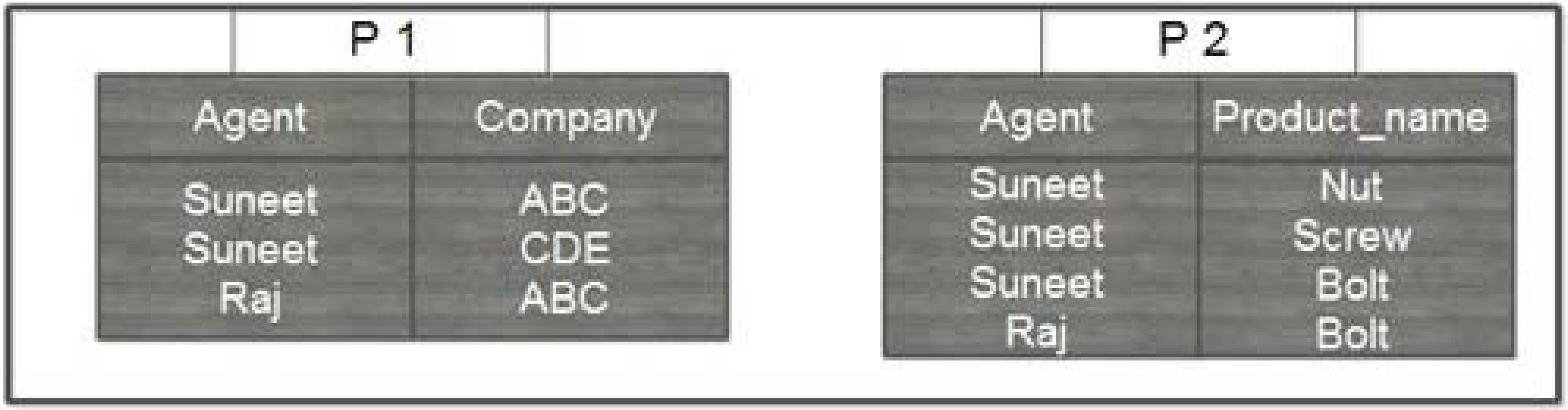
- Fifth normal form is satisfied when all tables are broken into as many tables as possible in order to avoid redundancy. Once it is in fifth normal form it cannot be broken into smaller relations without changing the facts or the meaning.

Example



Agent	Company	Product Name
Suneet	ABC	Nut
Suneet	ABC	Screw
Suneet	CDE	Bolt
Raj	ABC	Bolt

After 5NF



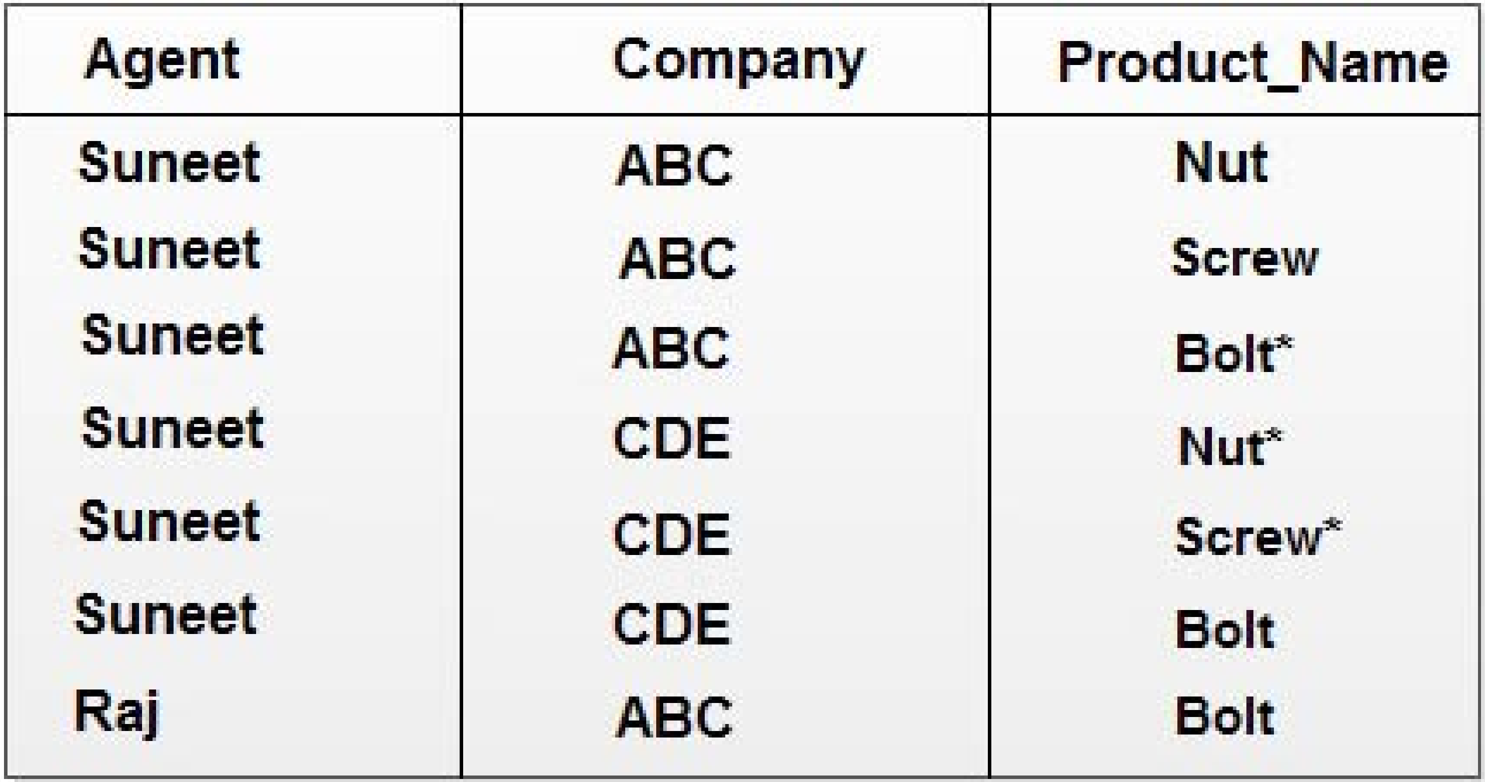
P 1

Agent	Company
Suneet	ABC
Suneet	CDE
Raj	ABC

P 2

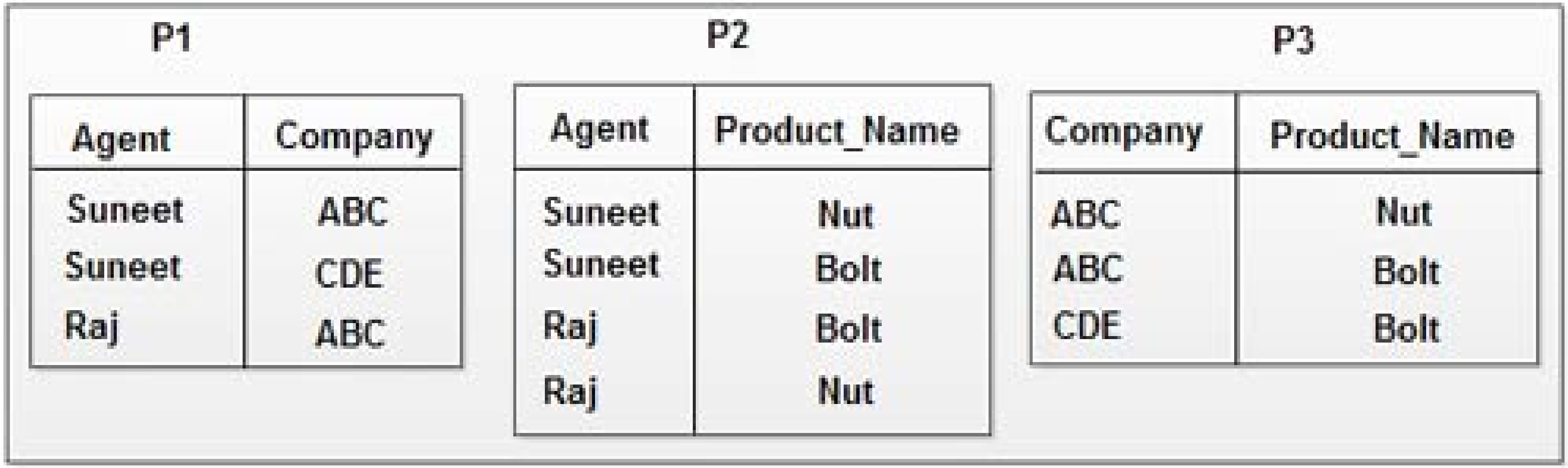
Agent	Product_name
Suneet	Nut
Suneet	Screw
Suneet	Bolt
Raj	Bolt

Results After Join



Agent	Company	Product_Name
Suneet	ABC	Nut
Suneet	ABC	Screw
Suneet	ABC	Bolt*
Suneet	CDE	Nut*
Suneet	CDE	Screw*
Suneet	CDE	Bolt
Raj	ABC	Bolt

After 5NF



P1

Agent	Company
Suneet	ABC
Suneet	CDE
Raj	ABC

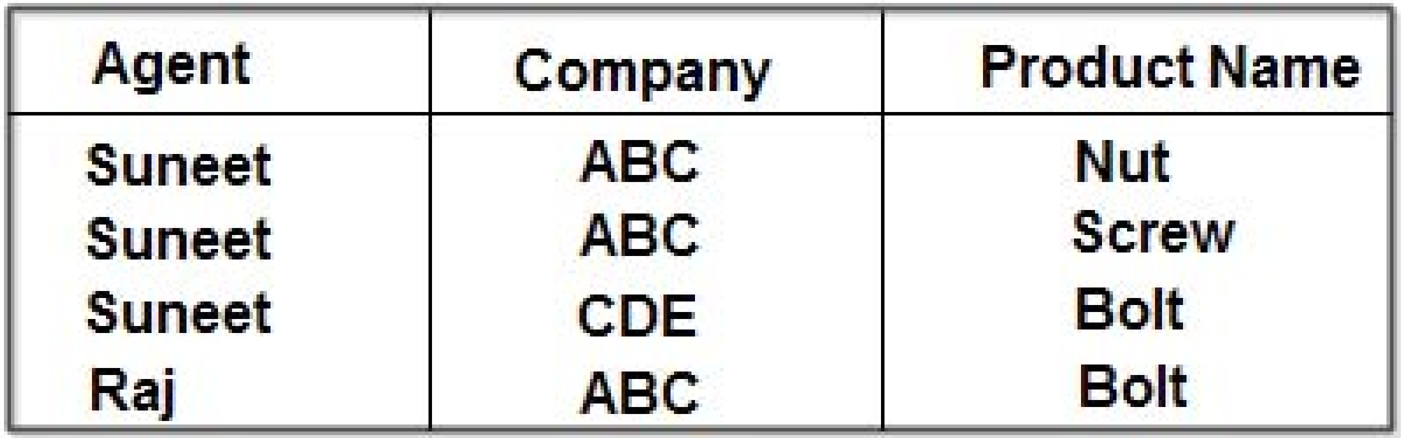
P2

Agent	Product_Name
Suneet	Nut
Suneet	Bolt
Raj	Bolt
Raj	Nut

P3

Company	Product_Name
ABC	Nut
ABC	Bolt
CDE	Bolt

After Performing Join Operation



Agent	Company	Product Name
Suneet	ABC	Nut
Suneet	ABC	Screw
Suneet	CDE	Bolt
Raj	ABC	Bolt

DKNF

- Constraint
- An rule governing static values of an attribute such that we can determine if this constraint is True or False. Example
  - Functional Dependencies
  - Multi-valued Dependencies
  - Inter-relation rules
  - Intra-relation rules

DKNF

- The relation is in DKNF when there can be no insertion or deletion anomalies in the database.

DKNF

Wealthy Person

Wealthy Person	Wealthy Person Type	Net Worth in Dollars
Steve	Eccentric	124,543,621
Roderick	Evil	6,553,228,893
Katrina	Eccentric	8,829,462,998
Gary	Evil	495,565,211

Wealthiness Status

Status	Minimum	Maximum
Millionaire	1,000,000	999,999,999
Billionaire	1,000,000,000	999,999,999,999

Normal Form - Summary

Normal Form	Description
1NF	A relation is in 1NF if it contains an atomic value.
2NF	A relation will be in 2NF if it is in 1NF and no partial key dependency exists.
3NF	A relation will be in 3NF if it is in 2NF and no transition dependency exists.
BCNF	A relation is in BCNF, if it is in 3NF & if and only if, every determinant is a candidate key.
4NF	A relation will be in 4NF if it is in Boyce Codd Normal Form and has no multi-valued dependency.
5NF	A relation is in 5NF if it is in 4NF and not contains any join dependency and joining should be lossless.

Thank you!