UNIT - III

Database Design

Topics to be covered…

- **Database Design:**
- Informal Design Guidelines for Relation Schemas;
- Functional Dependencies;
- Normal Forms Based on Primary Keys;
- General Definitions of Second and
- Third Normal Forms;

First let us know …….

- A **relational schema** is a blueprint used in database design to represent the data to be entered into the database and describe how that data is structured in tables (called relations in **relational schemas**)

- **Relational Database Design Process**
- Step 1: Define the Purpose of the Database (Requirement Analysis) ...
- Step 2: Gather Data, Organize in tables and Specify the Primary Keys. ...
- Step 3: **Create** Relationships among Tables. ...
- Step 4: Refine & Normalize the **Design**.

Four Informal Guidelines for Relational Schema

- Semantics of attributes
- Reducing the redundant information in tuples
- Reducing the NULL values in the tuple
- Disallowing the possibility of generating spurious tuples.

Semantics of the Relation Attributes

- Each tuple in a relation should represent one entity or relationship instance
  - Only foreign keys should be used to refer to other entities
  - Entity and relationship attributes should be kept apart as much as possible
  - Design a schema that can be explained easily relation by relation. The semantics of attributes should be easy to interpret.

Example: (Confused)

Employee	E	SS	B	num

Department	N	Dno	Mgr

Dept_loc	Dno	Dloc

Project	name	no	P	Dn

Example: (Correct Picture)

		pk		fk
Employee	Ename	SSN	Bdate	Dno

		pk	
Department	Dname	Dno	Mgrssn

	fk	
Dept_loc	Dno	Dloc

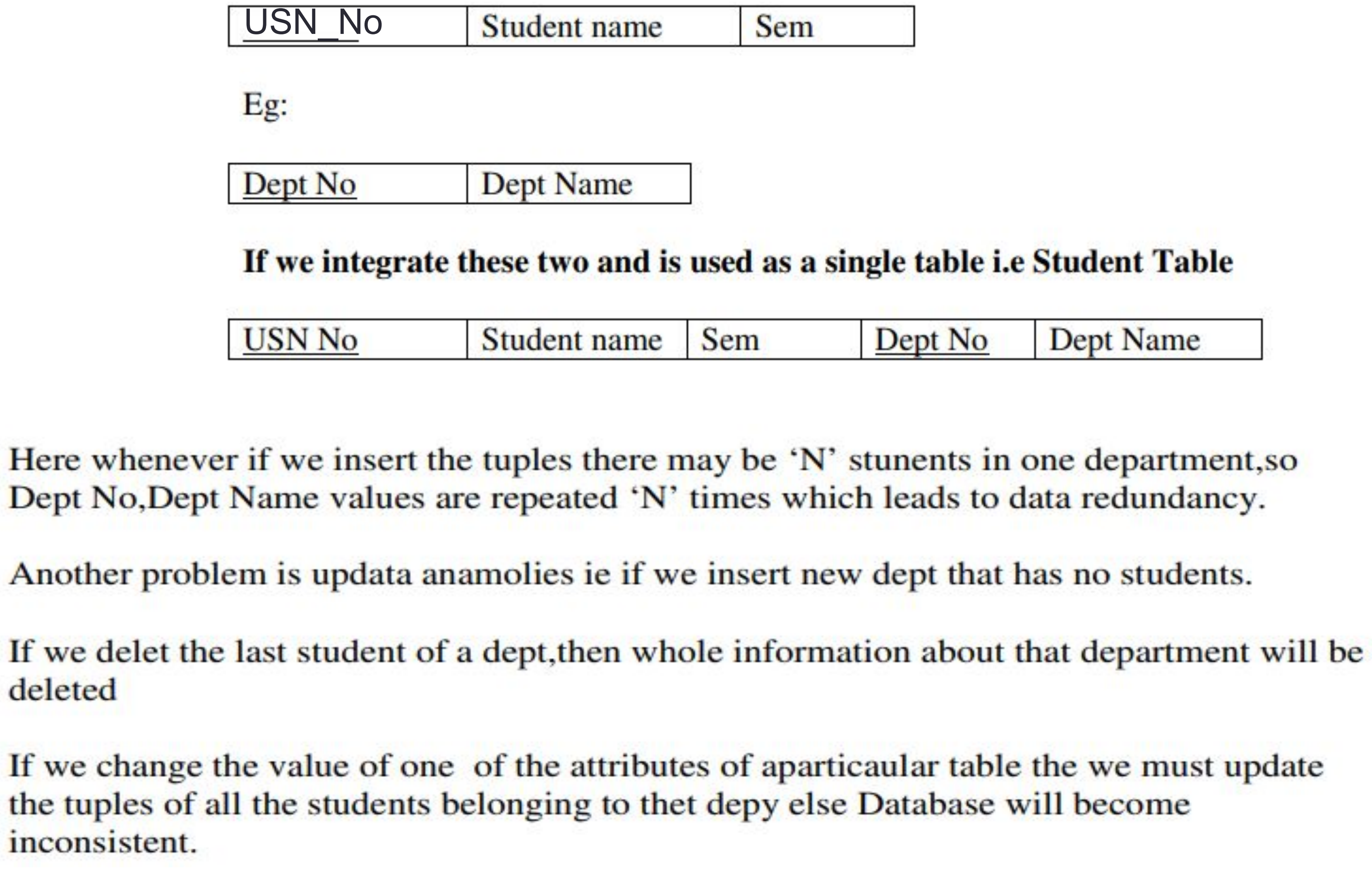
pk

				fk
Project	Pname	Pno	Ploc	Dno
		pk		

Redundant Information in Tuples and Update Anomalies

- Mixing attributes of multiple entities may cause problems
  - Information is stored redundantly wasting storage
  - Problems with update anomalies:
    - Insertion anomalies
    - Deletion anomalies
    - Modification anomalies

Example:1



USN_No	Student name	Sem

Eg:

Dept No	Dept Name

If we integrate these two and is used as a single table i.e Student Table

USN No	Student name	Sem	Dept No	Dept Name

Here whenever if we insert the tuples there may be 'N' stunents in one department,so Dept No,Dept Name values are repeated 'N' times which leads to data redundancy.

Another problem is updata anamolies ie if we insert new dept that has no students.

If we delet the last student of a dept,then whole information about that department will be deleted

If we change the value of one of the attributes of aparticaular table the we must update the tuples of all the students belonging to thet depy else Database will become inconsistent.

EXAMPLE OF AN UPDATE ANOMALY

Consider the relation:

EMP_PROJ (Emp#, Proj#, Ename, Pname, No_hours)

- **Update Anomaly**
  - Changing the name of project number P1 from "Billing" to "Customer-Accounting" may cause this update to be made for all 100 employees working on project P1
- **Insert Anomaly**
  - Cannot insert a project unless an employee is assigned to .
  - Inversely- Cannot insert an employee unless he/she is assigned to a project.
- **Delete Anomaly**
  - When a project is deleted, it will result in deleting all the employees who work on that project. Alternately, if an employee is the sole employee on a project, deleting that employee would result in deleting the corresponding project.

EXAMPLE OF AN UPDATE ANOMALY (2)

- Design a schema that does not suffer from
  - insertion,
  - deletion and
  - update anomalies.

  - If there are any present, then note them so that applications can be made to take them into account

Null Values in Tuples

- Relations should be designed such that their tuples will have as few NULL values as possible
  - Attributes that are NULL frequently could be placed in separate relations (with the primary key)
  - Reasons for nulls:
  - a. attribute not applicable or invalid
  - b. attribute value unkown (may exist)
  - c. value known to exist, but unavailable

Spurious Tuples

- Bad designs for a relational database may result in erroneous results for certain JOIN operations
- The "lossless join" property is used to guarantee meaningful results for join operations
- The relations should be designed to satisfy the lossless join condition. No spurious tuples should be generated by doing a natural-join of any relations

Example

Employee

SSN	ENAME	DNO
1	Amit	1
2	Dilip	2

Department

DNO	DNAME
1	CS
2	EC

Join Employee Department

SSN	ENAME	DNO	DNO	DNAME
1	Amit	1	1	CS
1	Amit	1	2	EC
2	Dilip	2	1	CS
2	Dilip	2	2	EC

Example

Employee

SSN	ENAME	DNO
1	Amit	1
2	Dilip	2

Department

DNO	DNAME
1	CS
2	EC

Join Employee Department

SSN	ENAME	DNO	DNO	DNAME
1	Amit	1	1	CS
1	Amit	1	2	EC
2	Dilip	2	1	CS
2	Dilip	2	2	EC

Spurious Tuples

Guidelines

- Guideline 1: Design a relation schema so that it is easy to explain its meaning
- Guideline 2: Design the base relation schemas that has no insertion, deletion or modification anomalies.
- Guideline 3: Avoid using NULL in relations
- Guideline 4: join relations with equi-joins

Functional Dependencies

Definition: If one set of attributes in a table determines another set of attributes in the table, then the second set of attributes is said to be functionally dependent on the first set of attributes.

Example 1

<u>ISBN</u>	Title	Price
0-321-32132-1	Balloon	$34.00
0-55-123456-9	Main Street	$22.95
0-123-45678-0	Ulysses	$34.00
1-22-233700-0	Visual Basic	$25.00

Table Scheme: {ISBN, Title, Price}

Functional Dependencies: {ISBN} □ {Title}

{ISBN} □ {Price}

Functional Dependencies

Example 2

PubID	PubName	PubPhone
1	Big House	999-999-9999
2	Small House	123-456-7890
3	Alpha Press	111-111-1111

Table Scheme: {PubID, PubName, PubPhone}

Functional Dependencies: {PubId} □ {PubPhone}
{PubId} □ {PubName}
{PubName, PubPhone} □ {PubID}

Example 3

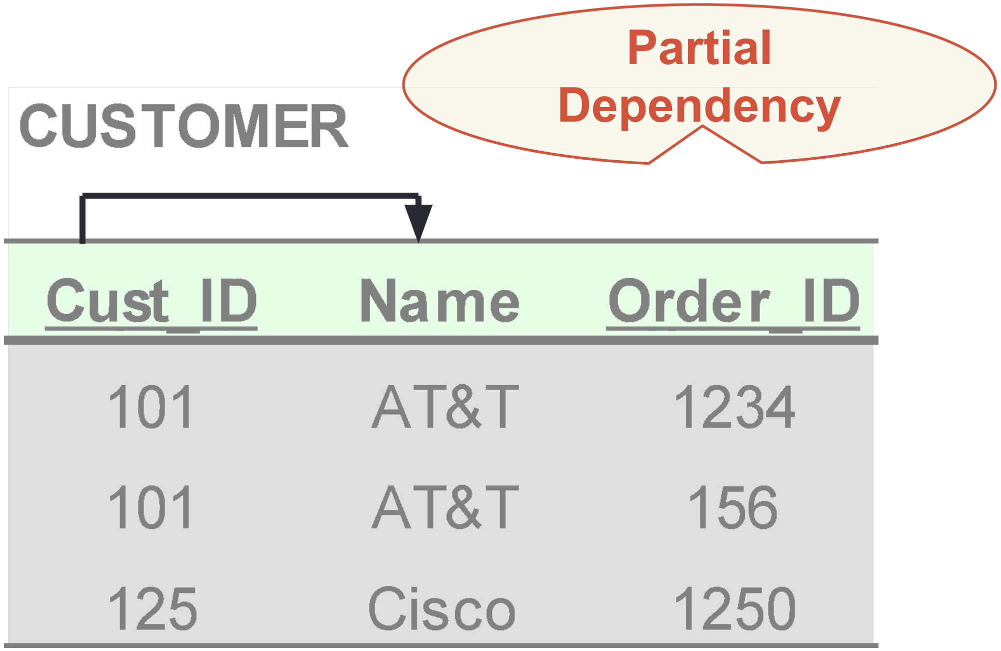
AuID	AuName	AuPhone
1	Sleepy	321-321-1111
2	Snoopy	232-234-1234
3	Grumpy	665-235-6532
4	Jones	123-333-3333
5	Smith	654-223-3455
6	Joyce	666-666-6666
7	Roman	444-444-4444

Table Scheme: {AuID, AuName, AuPhone}

Functional Dependencies: {AuId} □ {AuPhone}
{AuId} □ {AuName}
{AuName, AuPhone} □ {AuID}

Dependencies: Definitions

- ***Partial Dependency*** – when an non-key attribute is determined by a part, but not the whole, of a **COMPOSITE** primary key.



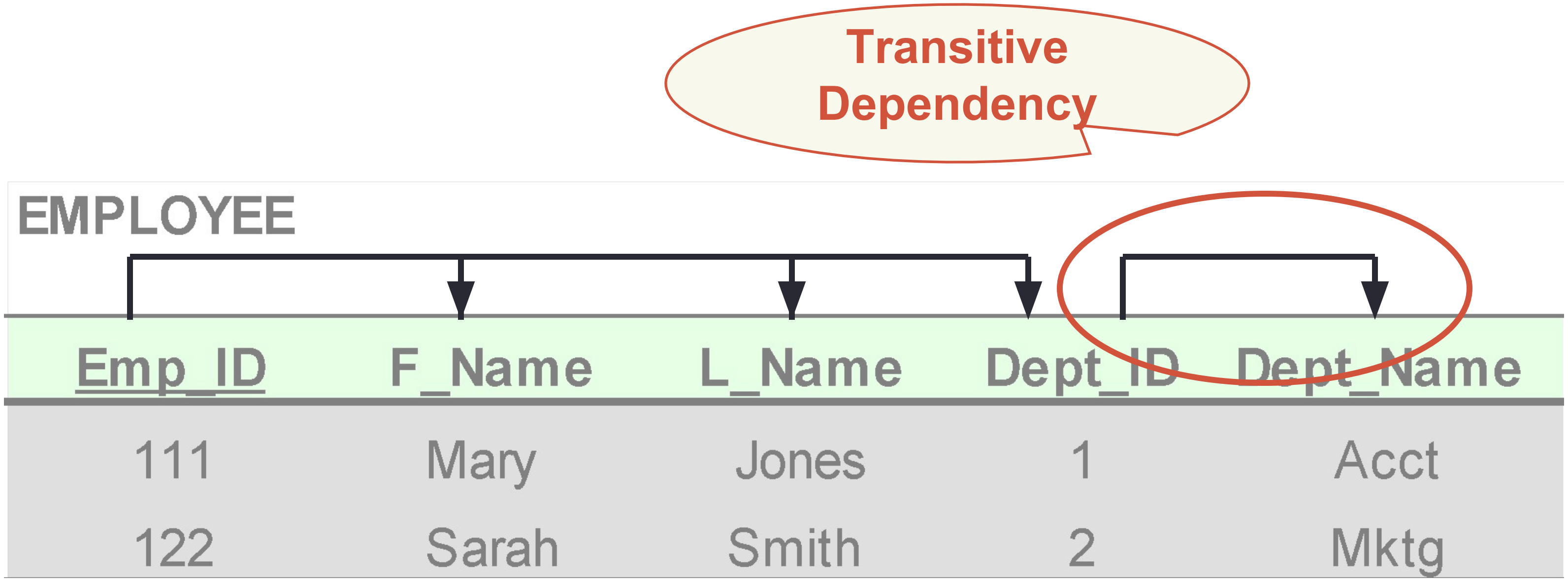
Partial Dependency

CUSTOMER

Cust_ID	Name	Order_ID
101	AT&T	1234
101	AT&T	156
125	Cisco	1250

Dependencies: Definitions

- ***Transitive Dependency*** – when a non-key attribute determines another non-key attribute.



Transitive Dependency

EMPLOYEE

Emp_ID	F_Name	L_Name	Dept_ID	Dept_Name
111	Mary	Jones	1	Acct
122	Sarah	Smith	2	Mktg

DATABASE NORMALIZATION

Introduction to Normalization

- **Normalization**: Process of decomposing unsatisfactory "bad" relations by breaking up their attributes into smaller relations
- **Normal form**: Condition using keys and FDs of a relation to certify whether a relation schema is in a particular normal form
  - 2NF, 3NF, BCNF based on keys and FDs of a relation schema
  - 4NF based on keys, multi-valued dependencies

Need for normalization

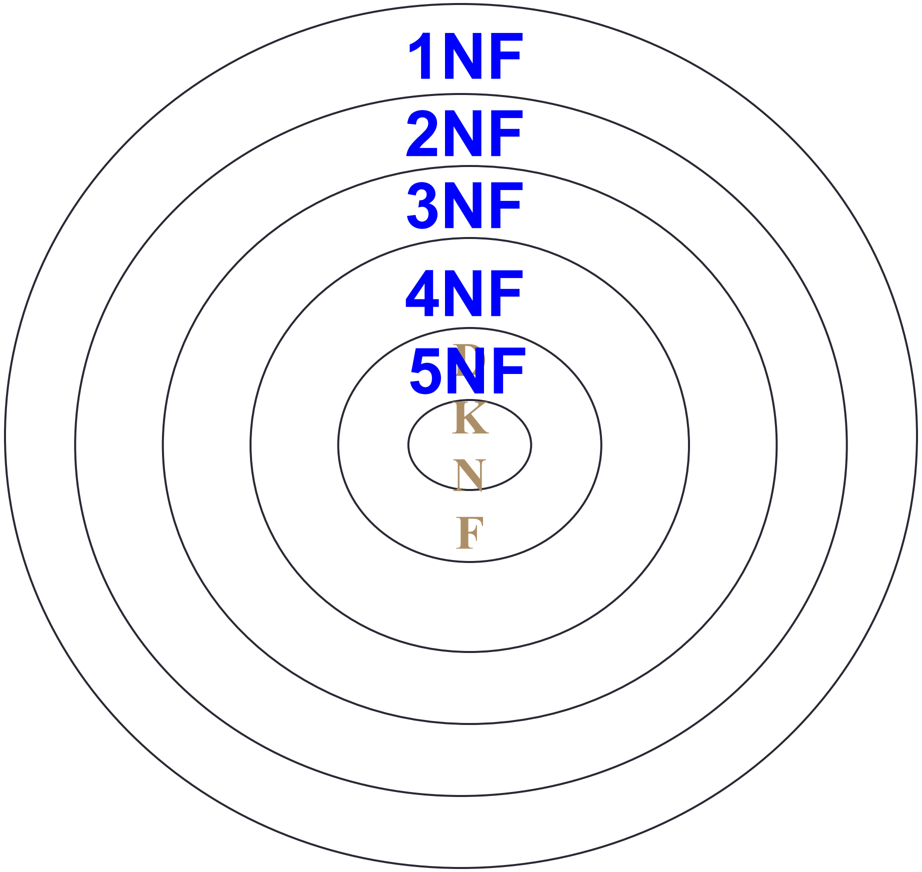
- A properly normalized database should have the following characteristics
  - Scalar values in each fields
  - Absence of redundancy.
  - Minimal use of null values.
  - Minimal loss of information.

Levels of Normalization

- Levels of normalization based on the amount of redundancy in the database.
- Various levels of normalization are:
    - 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 (DKNF)

Most databases should be 3NF or BCNF in order to avoid the database anomalies.

Levels of Normalization



Each higher level is a subset of the lower level

First Normal Form (1NF)

A table is considered to be in 1NF if all the fields contain only scalar values (as opposed to list of values).

Example (Not 1NF)

ISBN	Title	AuName	AuPhone	PubName	PubPhone	Price
0-321-32132-1	Balloon	Sleepy, Snoopy, Grumpy	321-321-1111, 232-234-1234, 665-235-6532	Small House	714-000-0000	$34.00
0-55-123456-9	Main Street	Jones, Smith	123-333-3333, 654-223-3455	Small House	714-000-0000	$22.95
0-123-45678-0	Ulysses	Joyce	666-666-6666	Alpha Press	999-999-9999	$34.00
1-22-233700-0	Visual Basic	Roman	444-444-4444	Big House	123-456-7890	$25.00

Author and AuPhone columns are not scalar

1NF - Decomposition

1. Place all items that appear in the repeating group in a new table
2. Designate a primary key for each new table produced.
3. Duplicate in the new table the primary key of the table from which the repeating group was extracted or vice versa.

Example (1NF)

ISBN	Title	PubName	PubPhone	Price
0-321-32132-1	Balloon	Small House	714-000-0000	$34.00
0-55-123456-9	Main Street	Small House	714-000-0000	$22.95
0-123-45678-0	Ulysses	Alpha Press	999-999-9999	$34.00
1-22-233700-0	Visual Basic	Big House	123-456-7890	$25.00

ISBN	AuName	AuPhone
0-321-32132-1	Sleepy	321-321-1111
0-321-32132-1	Snoopy	232-234-1234
0-321-32132-1	Grumpy	665-235-6532
0-55-123456-9	Jones	123-333-3333
0-55-123456-9	Smith	654-223-3455
0-123-45678-0	Joyce	666-666-6666
1-22-233700-0	Roman	444-444-4444

Is the following relation in 1NF?

USN	NAME	GENDER	CITY	HOBBIES

Is the following relation in 1NF?

USN	NAME	GENDER	CITY	HOBBIES

NO

Since hobbies will be multiple values !!!!!!!

Is the following relation in 1NF?

USN	NAME	GENDER	CITY	HOBBIES

NO

Since hobbies will be multiple values !!!!!!!

Can you convert it into 1NF???

Is the following relation in 1NF?

USN	NAME	GENDER	CITY	HOBBIES

NO

Since hobbies will be multiple values !!!!!!!

Can you convert it into 1NF???

YES

Is the following relation in 1NF?

USN	NAME	GENDER	CITY	HOBBIES

First way:

USN	NAME	GENDER	CITY

USN	HOBBIES

Is the following relation in 1NF?

USN	NAME	GENDER	CITY	HOBBIES

First way:

USN	NAME	GENDER	CITY

USN	HOBBIES

Second way:

USN	NAME	GENDER	CITY	HOBBIES

Is the following relation in 1NF?

USN	NAME	GENDER	CITY	HOBBIES

First way:

USN	NAME	GENDER	CITY

USN	HOBBIES

Second way:

USN	NAME	GENDER	CITY	HOBBIES

Third way:

USN	NAME	GENDER	CITY	Hobby1	Hobby 2	Hobby 3

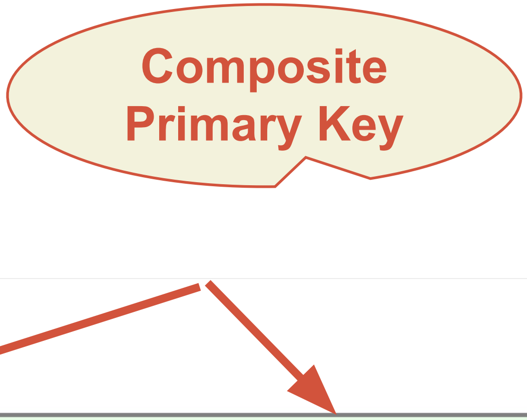
Second Normal Form (2NF)

For a table to be in 2NF, there are two requirements

- The database is in first normal form
- All **nonkey** attributes in the table must be functionally dependent on the entire primary key(fully functional dependent)

Note: *Remember that we are dealing with non-key attributes*

Normalizing a Relation to 2NF



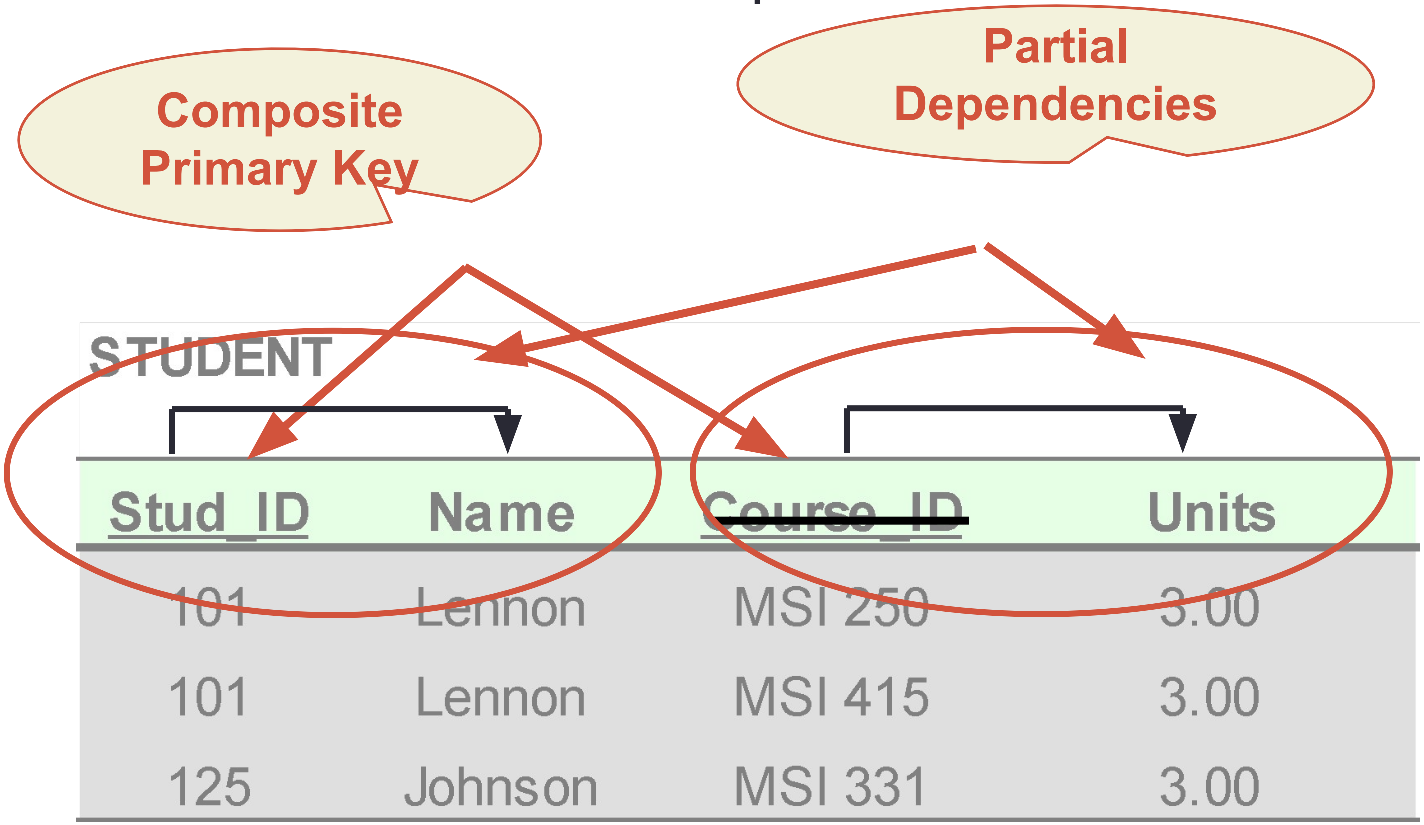
Composite Primary Key

STUDENT

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

Bringing a Relation to 2NF

- Goal: Remove Partial Dependencies

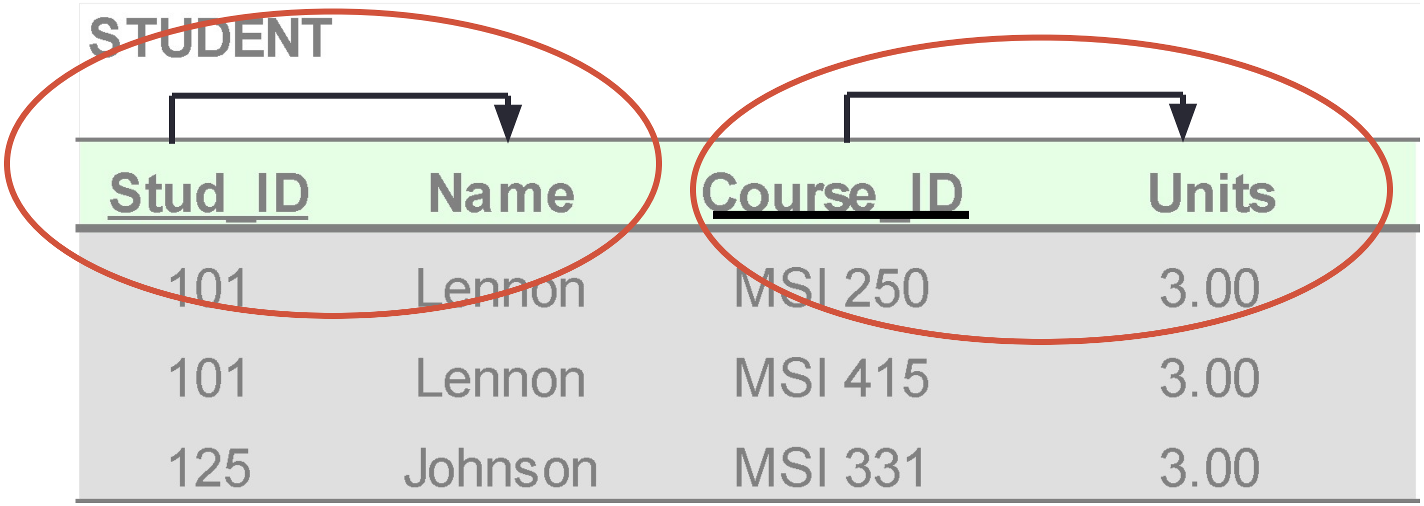


STUDENT

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

Bringing a Relation to 2NF

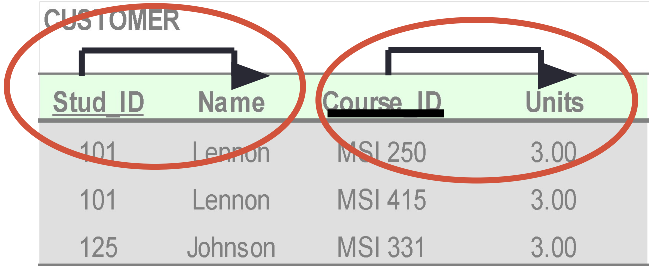
- Remove attributes that are dependent from the part but not the whole of the primary key from the original relation. For each partial dependency, create a new relation, with the corresponding part of the primary key from the original as the primary key.



STUDENT

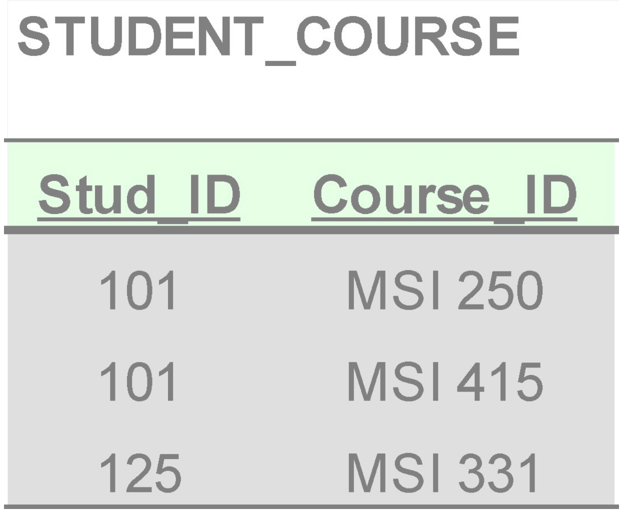
Stud_ID	Name	Course_ID	Units
101	Lennon	MSI 250	3.00
101	Lennon	MSI 415	3.00
125	Johnson	MSI 331	3.00

Bringing a Relation to 2NF



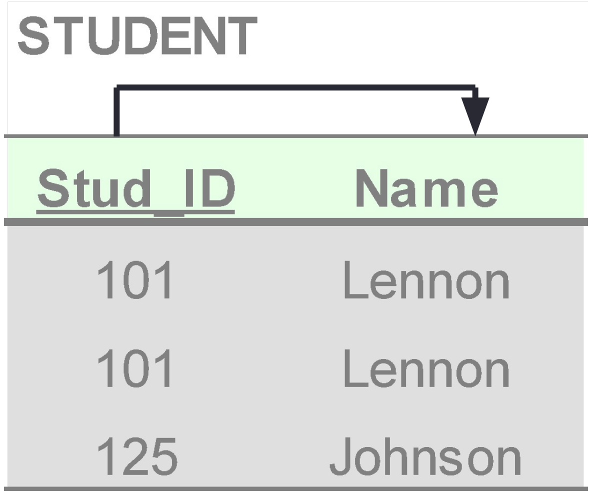
CUSTOMER

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



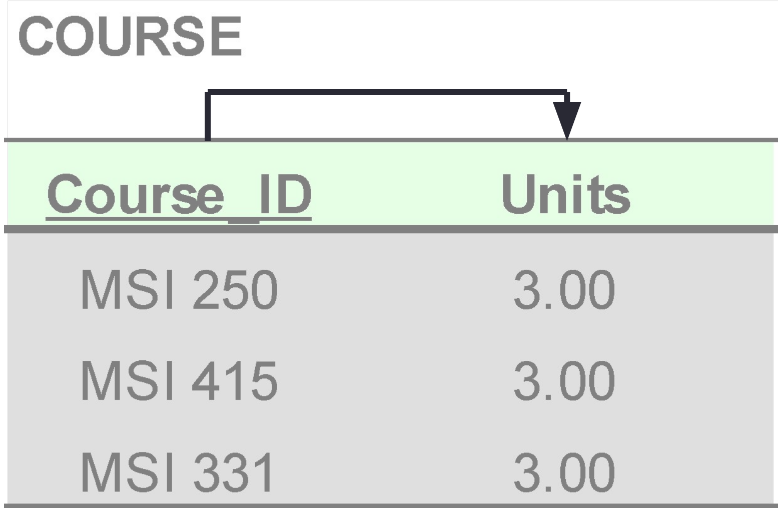
STUDENT_COURSE

Stud_ID	Course_ID
101	MSI 250
101	MSI 415
125	MSI 331



STUDENT

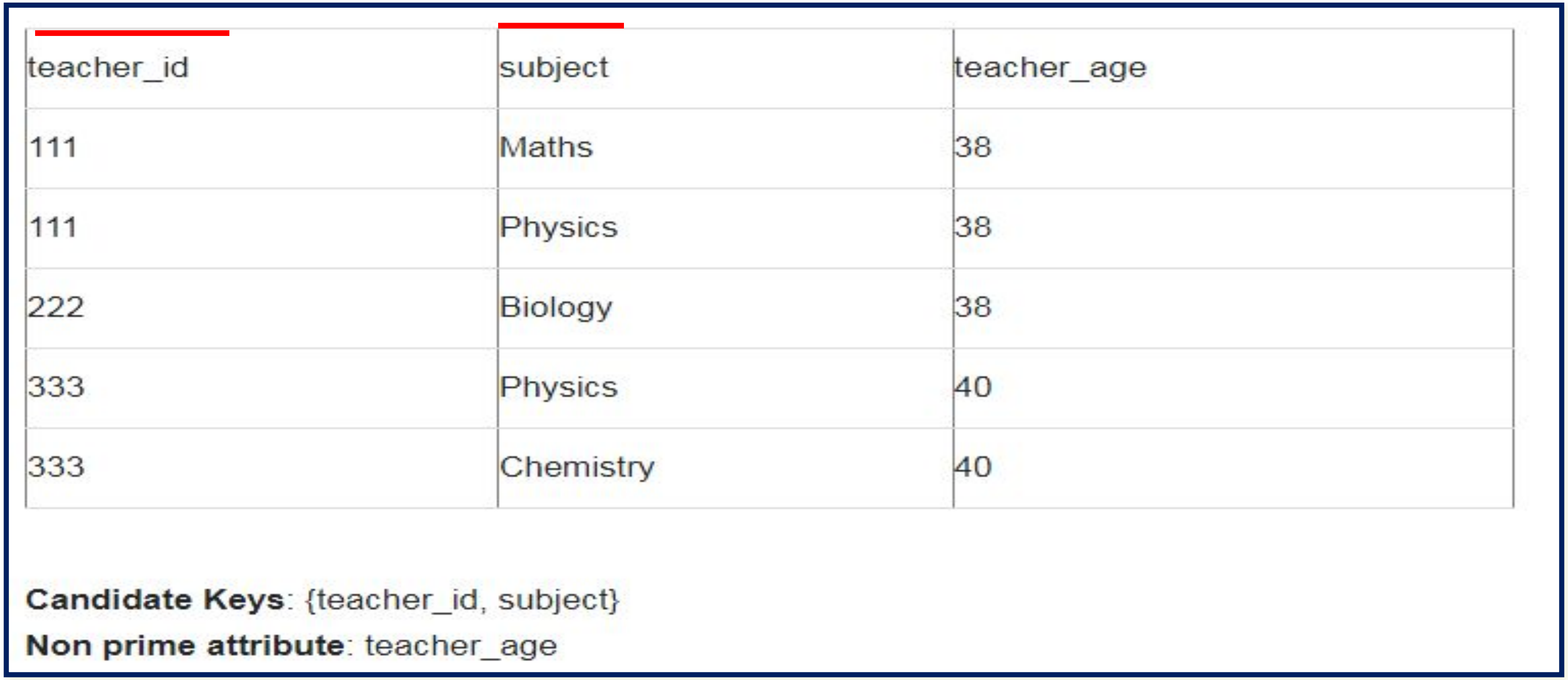
Stud_ID	Name
101	Lennon
101	Lennon
125	Johnson



COURSE

Course_ID	Units
MSI 250	3.00
MSI 415	3.00
MSI 331	3.00

Example – Is the given relation in 2NF?



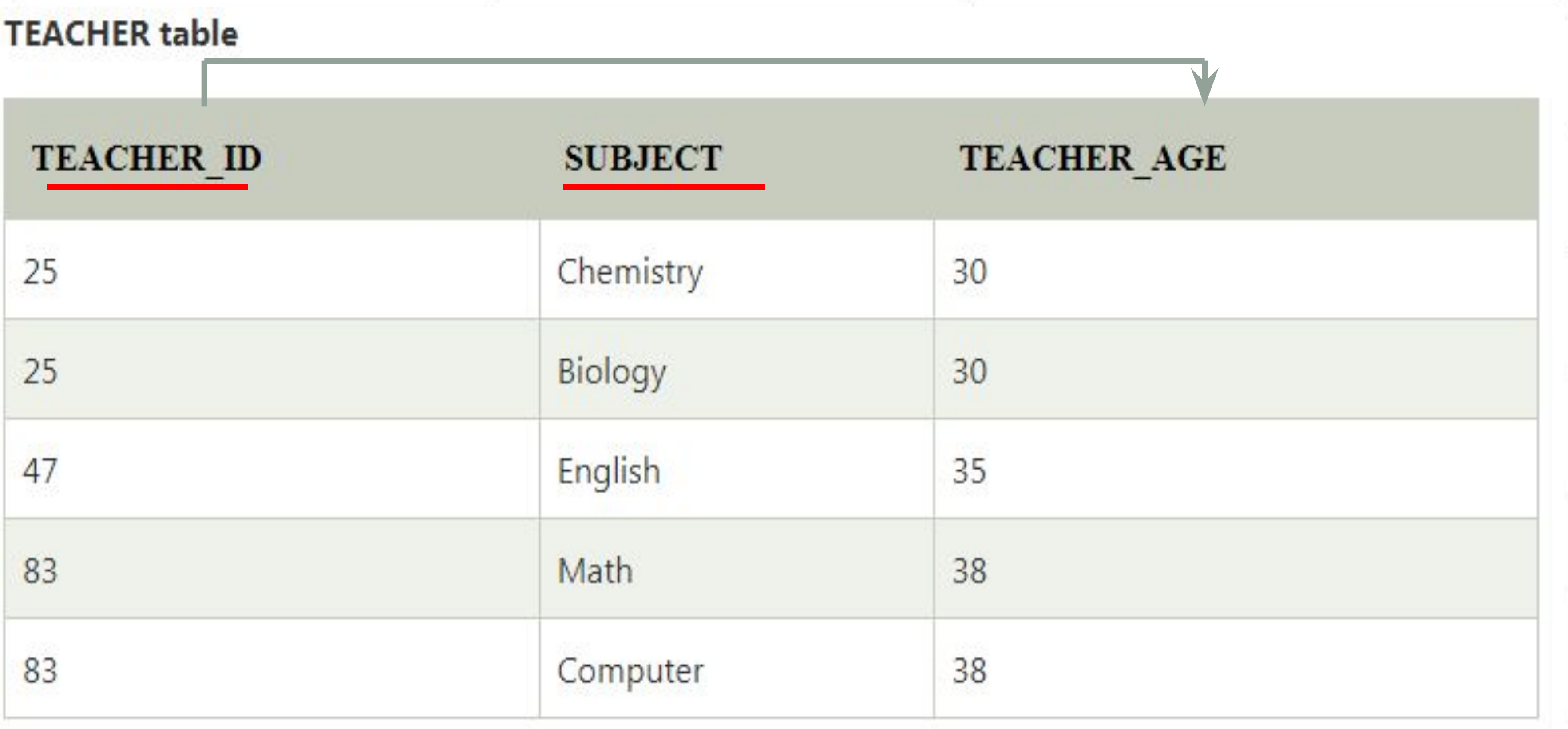
teacher_id	subject	teacher_age
111	Maths	38
111	Physics	38
222	Biology	38
333	Physics	40
333	Chemistry	40

Candidate Keys: {teacher_id, subject}

Non prime attribute: teacher_age

Non key attribute is dependent on a part of key, SO the relation is not in 2NF…..

Example – Is the given relation in 2NF?



TEACHER table

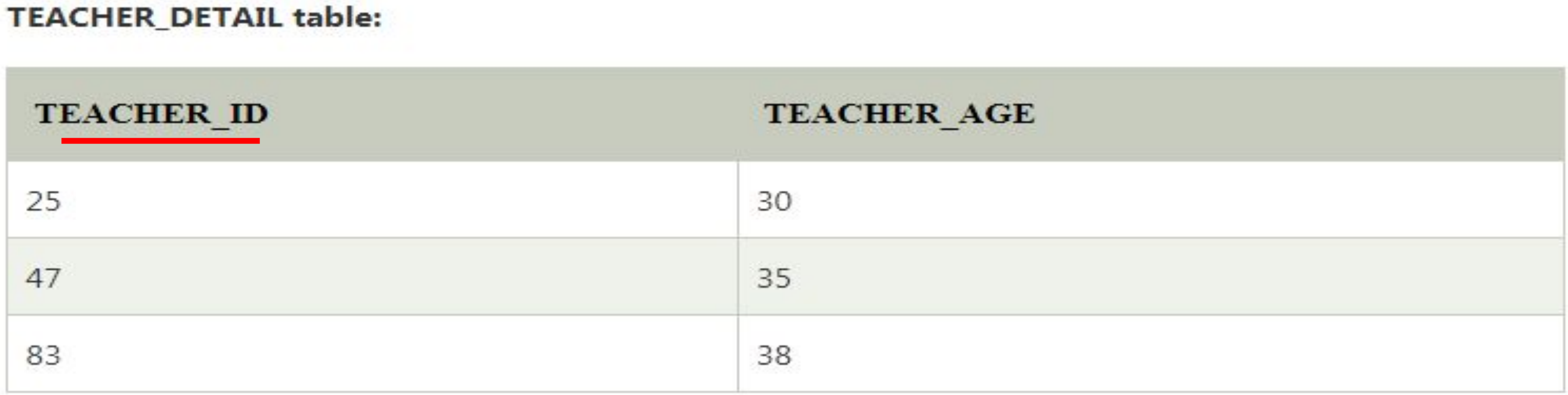
TEACHER_ID	SUBJECT	TEACHER_AGE
25	Chemistry	30
25	Biology	30
47	English	35
83	Math	38
83	Computer	38

Candidate Keys: {teacher_id, subject}

Non prime attribute: teacher_age

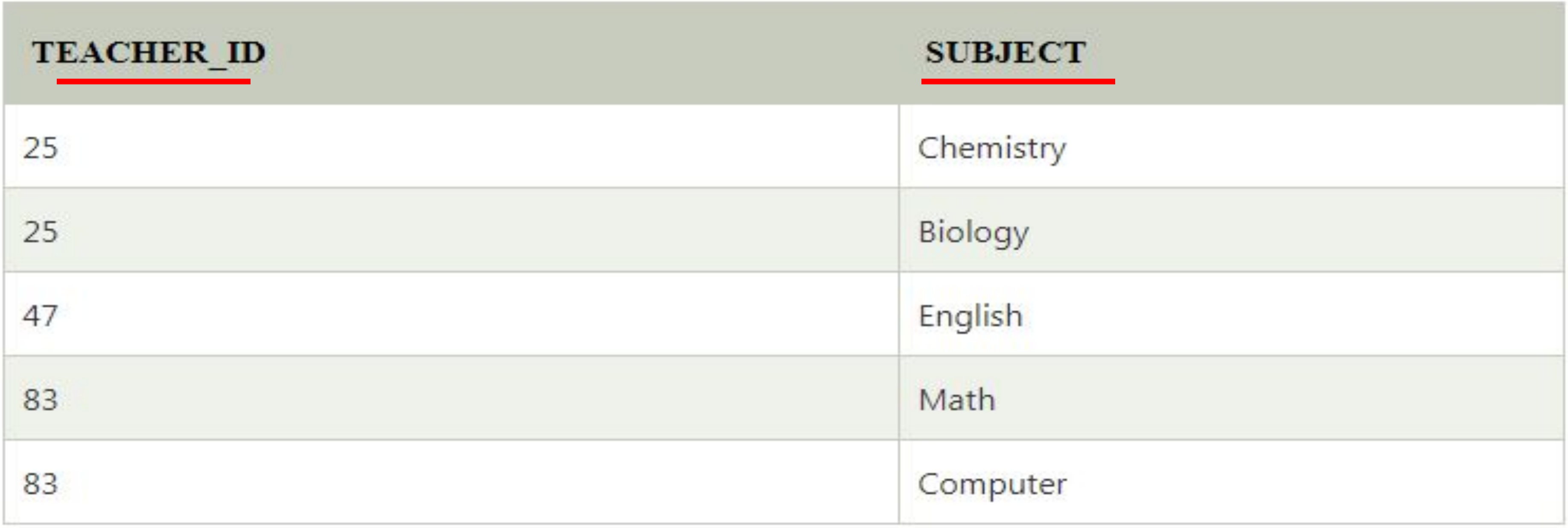
Non key attribute is dependent on a part of key, SO the relation is not in 2NF…..

To convert the given table into 2NF, we decompose it into two tables:



TEACHER_DETAIL table:

TEACHER_ID	TEACHER_AGE
25	30
47	35
83	38

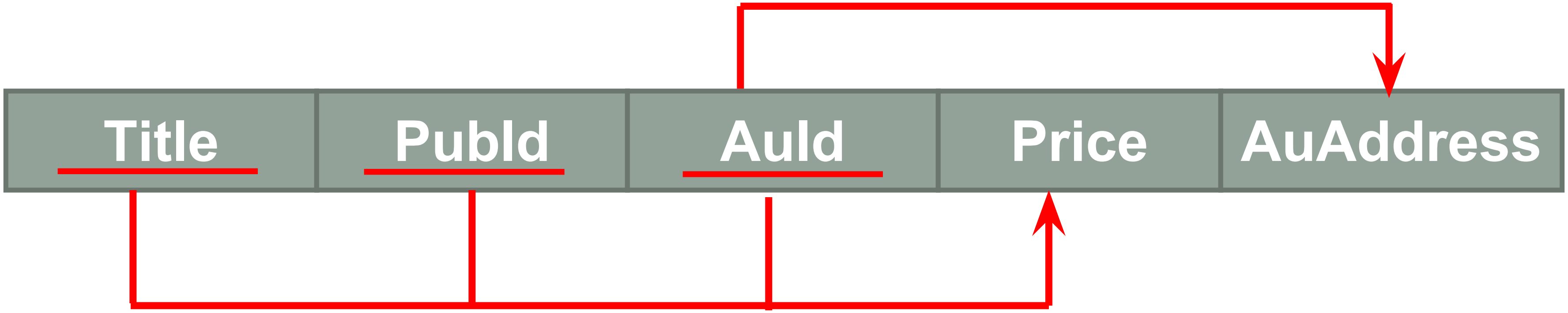


TEACHER_ID	SUBJECT
25	Chemistry
25	Biology
47	English
83	Math
83	Computer

Example 1 (Not 2NF)

Schema □ {Title, PubId, AuId, Price, AuAddress}

1. Key □ {Title, PubId, AuId}
2. {Title, PubId, AuID} □ {Price}
3. {AuID} □ {AuAddress}
4. AuAddress does not belong to a key
5. AuAddress functionally depends on AuId which is a subset of a key



Title PubId AuId Price AuAddress

2NF - Decomposition

1. *Identify* all the *functional dependencies* in the given relation.
2. *Identify partial functional dependencies*(non-prime attributes cab be determined from subset of key attributes).
3. All the attributes(non-prime) that are partially dependent should be made as a separate table along with the key that determines the non-prime attribute. (AuId □ AuAddress)
4. The non-prime attributes determined by the entire set of keys must be placed as a separate table.

Example 1 (Convert to 2NF)

Old Schema □ {Title, PubId, AuId, Price, AuAddress}

New Schema1□ {Title, PubId, AuId, Price}

New Schema2 □ {AuId, AuAddress}

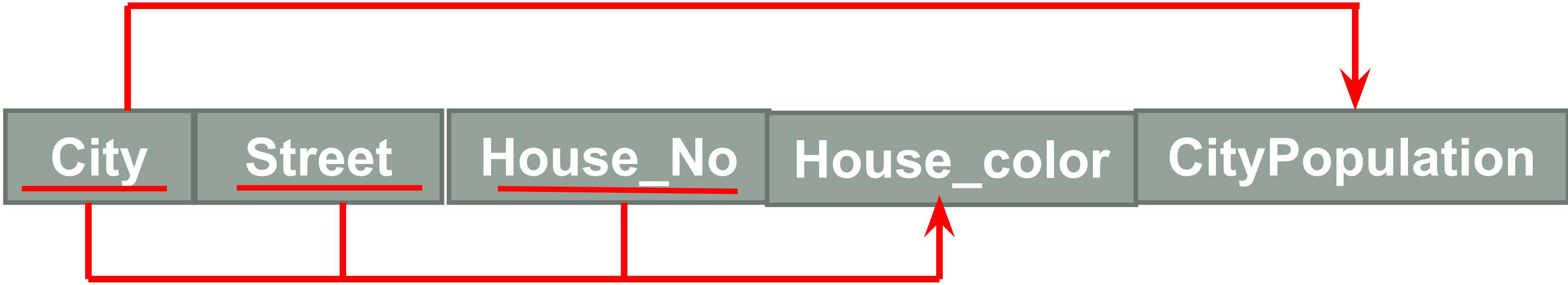
Title	PubId	AuId	Price

AuId	AuAddress

Example 2 (Not 2NF)

Schema □ {City, Street, HouseNumber, HouseColor, CityPopulation}

1. key □ {City, Street, HouseNumber}
2. {City, Street, HouseNumber} □ {HouseColor}
3. {City} □ {CityPopulation}
4. CityPopulation isn’t determined by the set of keys.
5. CityPopulation is functionally dependent on the City which is a proper subset of the key



City

Street

House_No

House_color

CityPopulation

2NF - Decomposition

Example 2 (Convert to 2NF)

Old Scheme □ {Studio, Movie, Budget, StudioCity}

New Scheme 1□ {Movie, Studio, Budget}

New Scheme2 □ {Studio, City}

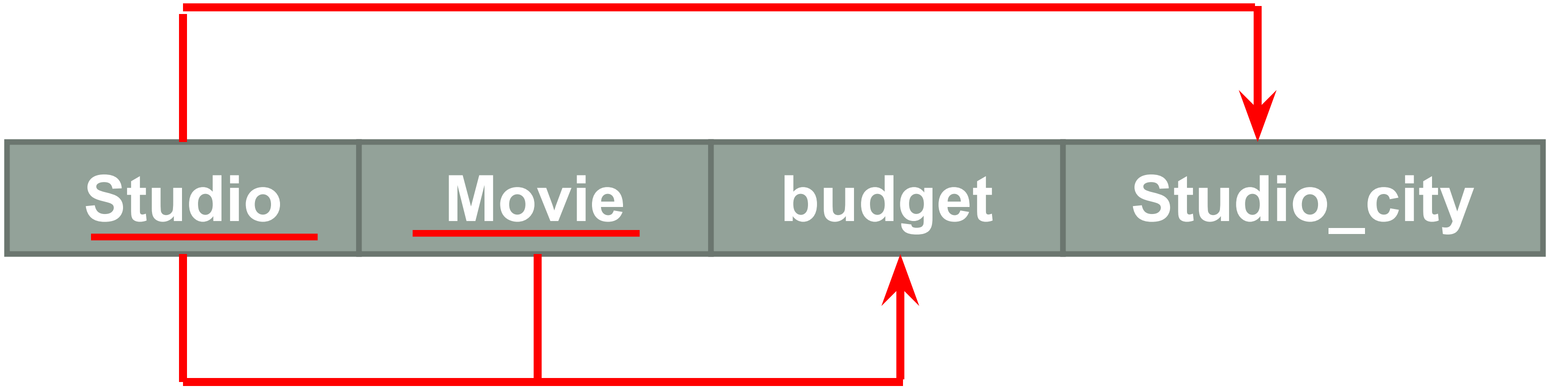
Studio	Movie	budget

Studio	Studio_city

Example 3 (Not 2NF)

Scheme □ {studio, movie, budget, studio_city}

1. Key □ {studio, movie}
2. {studio, movie} □ {budget}
3. {studio} □ {studio_city}
4. studio_city is not determined by the entire set of a key
5. studio_city functionally depends on studio which is a proper subset of the key



Studio Movie budget Studio_city

2NF - Decomposition

Example 3 (Convert to 2NF)

Old Scheme □ {City, Street, HouseNumber, HouseColor, CityPopulation}

New Scheme1 □ {City, Street, HouseNumber, HouseColor}

New Scheme2 □ {City, CityPopulation}

City	Street	House_No	House_color

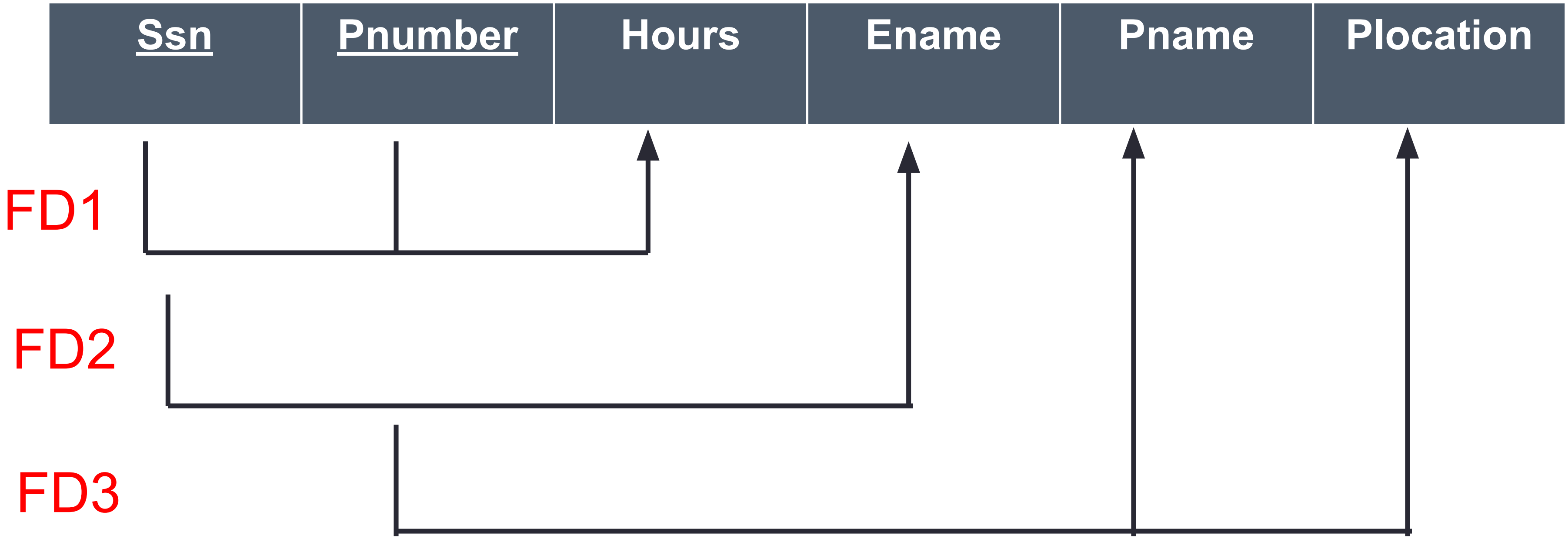
City	CityPopulation

EMP_PROJ

Ssn	Pnumber	Hours	Ename	Pname	Plocation

Can you normalize the same to 2NF????

EMP_PROJ



Ssn	Pnumber	Hours	Ename	Pname	Plocation

FD1

FD2

FD3

2NF - Decomposition

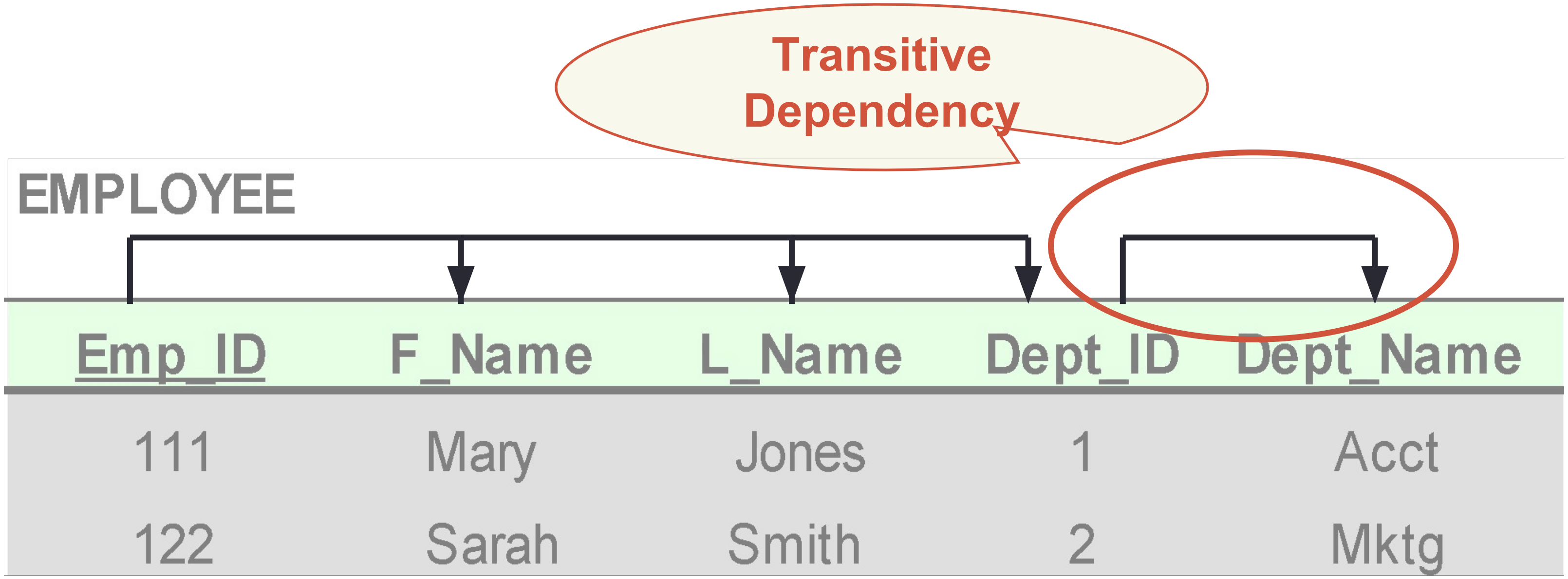
Ssn	Pnumber	Hours

Ssn	Ename

Pnumber	Pname	Plocation

Bringing a Relation to 3NF

- Goal: Get rid of transitive dependencies.



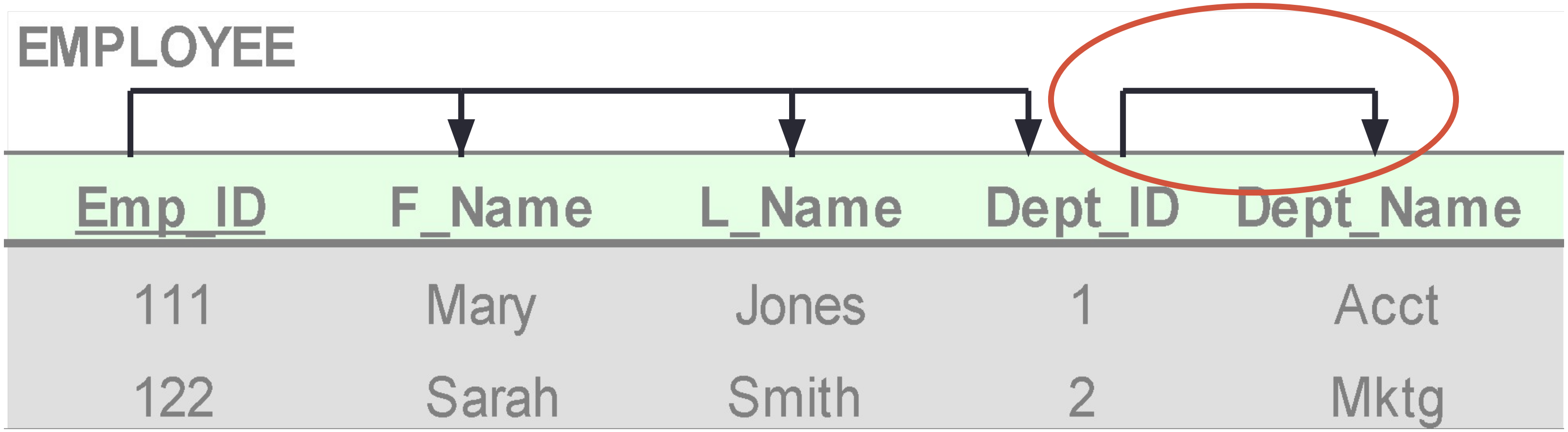
Transitive Dependency

EMPLOYEE

Emp_ID	F_Name	L_Name	Dept_ID	Dept_Name
111	Mary	Jones	1	Acct
122	Sarah	Smith	2	Mktg

Bringing a Relation to 3NF

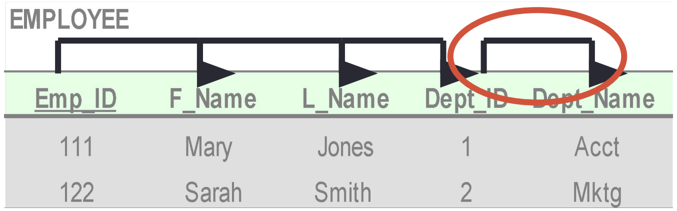
- Remove the attributes, which are dependent on a non-key attribute, from the original relation. For each transitive dependency, create a new relation with the non-key attribute which is a determinant in the transitive dependency as a primary key, and the dependent non-key attribute as a dependent.



EMPLOYEE

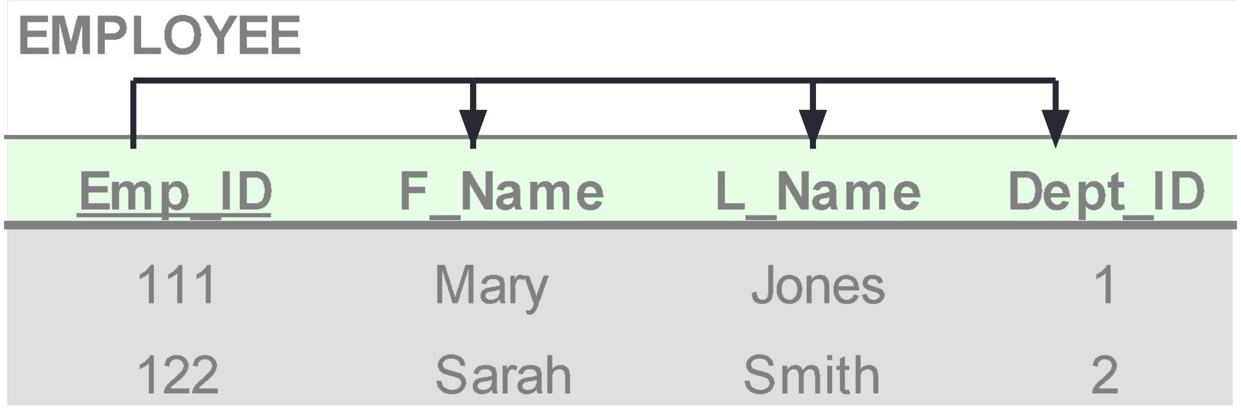
Emp_ID	F_Name	L_Name	Dept_ID	Dept_Name
111	Mary	Jones	1	Acct
122	Sarah	Smith	2	Mktg

Bringing a Relation to 3NF



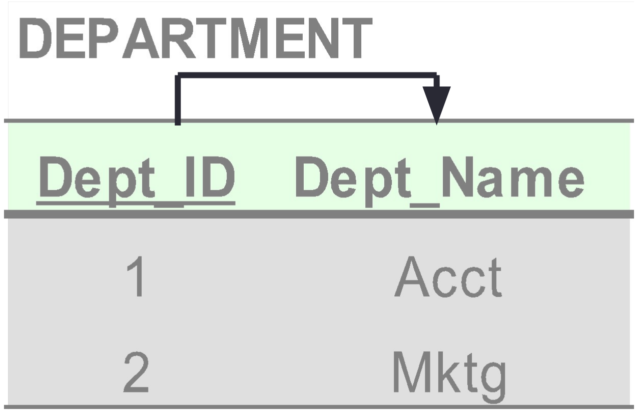
EMPLOYEE

Emp_ID	F_Name	L_Name	Dept_ID	Dept_Name
111	Mary	Jones	1	Acct
122	Sarah	Smith	2	Mktg



EMPLOYEE

Emp_ID	F_Name	L_Name	Dept_ID
111	Mary	Jones	1
122	Sarah	Smith	2



DEPARTMENT

Dept_ID	Dept_Name
1	Acct
2	Mktg

Third Normal Form (3NF)

This form dictates that all **non-key** attributes of a table must be functionally dependent on a candidate key i.e. there can be no interdependencies among non-key attributes.

For a table to be in 3NF, there are two requirements

- The table should be second normal form
- No attribute is transitively dependent on the primary key

Example (Not in 3NF)

Scheme □ {Title, PubID, PageCount, Price }

1. **Key □ {Title, PubId}**
2. **{Title, PubId} □ {PageCount}**
3. **{PageCount} □ {Price}**
4. **Both Price and PageCount depend on a key hence 2NF**
5. **Transitively {Title, PubID} □ {Price} hence not in 3NF**

Third Normal Form (3NF)

Example 2 (Not in 3NF)

Scheme □ {Studio, StudioCity, CityTemp}

1. Primary Key □ {Studio}
2. {Studio} □ {StudioCity}
3. {StudioCity} □ {CityTemp}
4. {Studio} □ {CityTemp}
5. Both StudioCity and CityTemp depend on the entire key hence 2NF
6. CityTemp transitively depends on Studio hence violates 3NF

Example 3 (Not in 3NF)

Scheme □ {BuildingID, Contractor, Fee}

7. Primary Key □ {BuildingID}
8. {BuildingID} □ {Contractor}
9. {Contractor} □ {Fee}
10. {BuildingID} □ {Fee}
11. Fee transitively depends on the BuildingID
12. Both Contractor and Fee depend on the entire key hence 2NF

BuildingID	Contractor	Fee
100	Randolph	1200
150	Ingersoll	1100
200	Randolph	1200
250	Pitkin	1100
300	Randolph	1200

3NF - Decomposition

1. Move all items involved in transitive dependencies to a new entity.
2. Identify a primary key for the new entity.
3. Place the primary key for the new entity as a foreign key on the original entity.

Example 1 (Convert to 3NF)

Old Scheme □ {Title, PubID, PageCount, Price }

New Scheme □ {PubID, PageCount, Price}

New Scheme □ {Title, PubID, PageCount}

3NF - Decomposition

Example 2 (Convert to 3NF)

Old Scheme □ {Studio, StudioCity, CityTemp}

New Scheme □ {Studio, StudioCity}

New Scheme □ {StudioCity, CityTemp}

Example 3 (Convert to 3NF)

Old Scheme □ {BuildingID, Contractor, Fee}

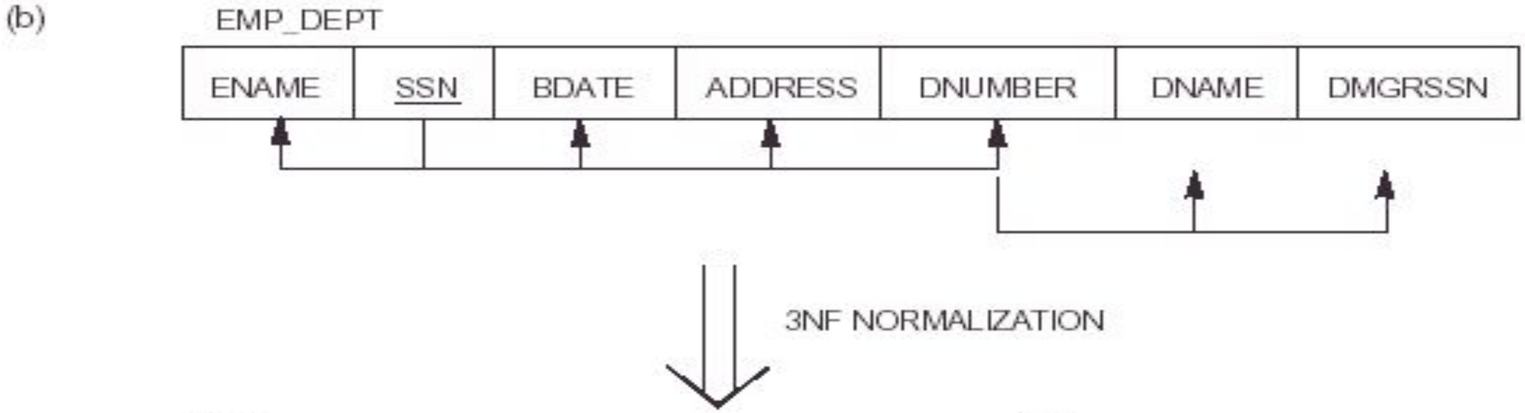
New Scheme □ {BuildingID, Contractor}

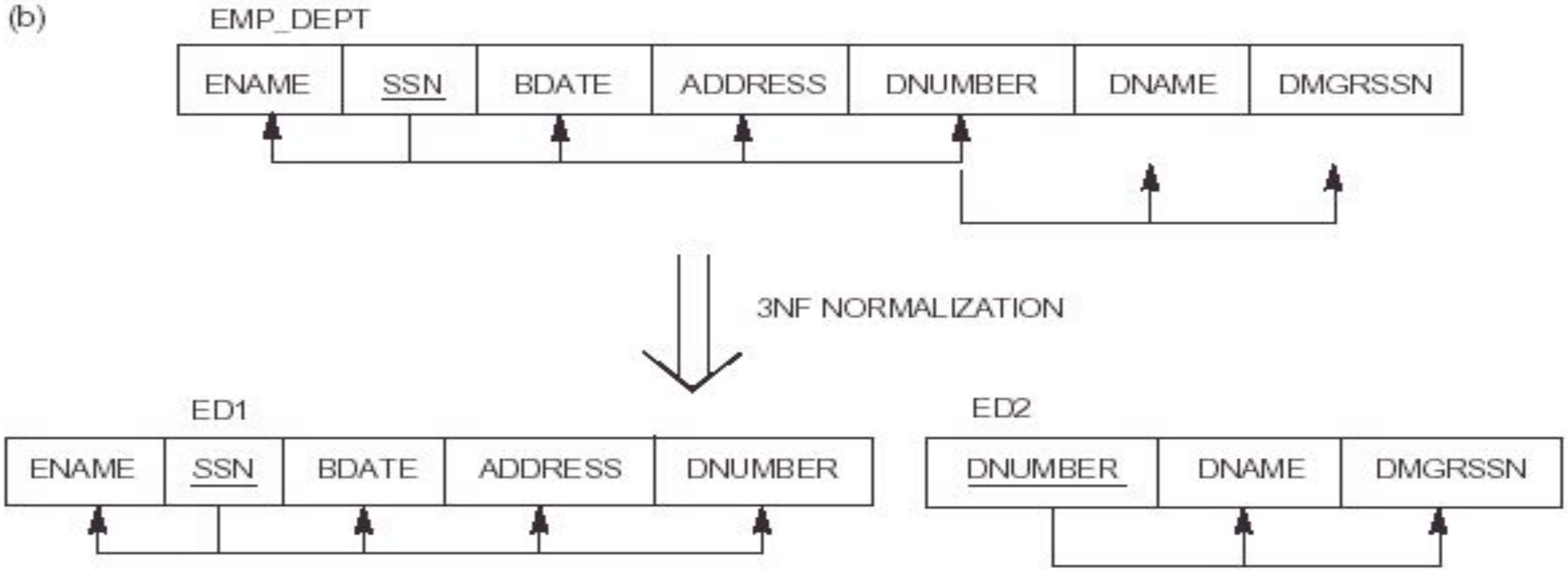
New Scheme □ {Contractor, Fee}

BuildingID	Contractor
100	Randolph
150	Ingersoll
200	Randolph
250	Pitkin
300	Randolph

Contractor	Fee
Randolph	1200
Ingersoll	1100
Pitkin	1100

Can you convert the same into 3NF????





(b)

EMP_DEPT

ENAME	SSN	BDATE	ADDRESS	DNUMBER	DNAME	DMGRSSN

3NF NORMALIZATION

ED1

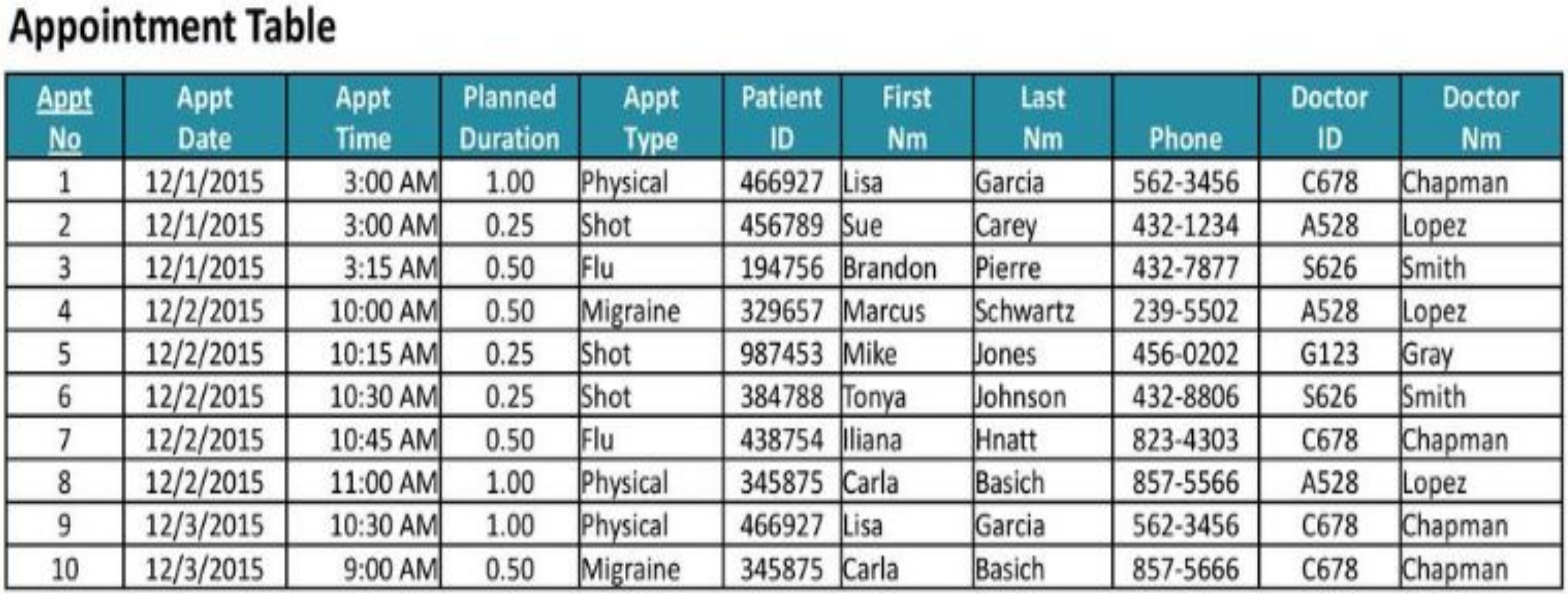
ENAME	SSN	BDATE	ADDRESS	DNUMBER

ED2

DNUMBER	DNAME	DMGRSSN

Practice Problem #2

Normalization Practice #2



Appointment Table

Appt No	Appt Date	Appt Time	Planned Duration	Appt Type	Patient ID	First Nm	Last Nm	Phone	Doctor ID	Doctor Nm
1	12/1/2015	3:00 AM	1.00	Physical	466927	Lisa	Garcia	562-3456	C678	Chapman
2	12/1/2015	3:00 AM	0.25	Shot	456789	Sue	Carey	432-1234	A528	Lopez
3	12/1/2015	3:15 AM	0.50	Flu	194756	Brandon	Pierre	432-7877	S626	Smith
4	12/2/2015	10:00 AM	0.50	Migraine	329657	Marcus	Schwartz	239-5502	A528	Lopez
5	12/2/2015	10:15 AM	0.25	Shot	987453	Mike	Jones	456-0202	G123	Gray
6	12/2/2015	10:30 AM	0.25	Shot	384788	Tonya	Johnson	432-8806	S626	Smith
7	12/2/2015	10:45 AM	0.50	Flu	438754	Iliana	Hnatt	823-4303	C678	Chapman
8	12/2/2015	11:00 AM	1.00	Physical	345875	Carla	Basich	857-5566	A528	Lopez
9	12/3/2015	10:30 AM	1.00	Physical	466927	Lisa	Garcia	562-3456	C678	Chapman
10	12/3/2015	9:00 AM	0.50	Migraine	345875	Carla	Basich	857-5666	C678	Chapman

1. Is it int 1NF? If not convert it into the same
2. Is it in 2NF? If not convert it into the same
3. Is it in 3NF? If not convert it into the same

Normalization Practice #2

Appointment Table

Appt No	Appt Date	Appt Time	Planned Duration	Appt Type	Patient ID	First Nm	Last Nm	Phone	Doctor ID	Doctor Nm
1	12/1/2015	3:00 AM	1.00	Physical	466927	Lisa	Garcia	562-3456	C678	Chapman
2	12/1/2015	3:00 AM	0.25	Shot	456789	Sue	Carey	432-1234	A528	Lopez
3	12/1/2015	3:15 AM	0.50	Flu	194756	Brandon	Pierre	432-7877	S626	Smith
4	12/2/2015	10:00 AM	0.50	Migraine	329657	Marcus	Schwartz	239-5502	A528	Lopez
5	12/2/2015	10:15 AM	0.25	Shot	987453	Mike	Jones	456-0202	G123	Gray
6	12/2/2015	10:30 AM	0.25	Shot	384788	Tonya	Johnson	432-8806	S626	Smith
7	12/2/2015	10:45 AM	0.50	Flu	438754	Iliana	Hnatt	823-4303	C678	Chapman
8	12/2/2015	11:00 AM	1.00	Physical	345875	Carla	Basich	857-5566	A528	Lopez
9	12/3/2015	10:30 AM	1.00	Physical	466927	Lisa	Garcia	562-3456	C678	Chapman
10	12/3/2015	9:00 AM	0.50	Migraine	345875	Carla	Basich	857-5666	C678	Chapman

- It's already in 1NF, and 2NF …but you need to understand why!

- **Convert this to 3NF. How many tables do you now have?**

SOLUTION #2

Normalization Practice #2

Appointment Table

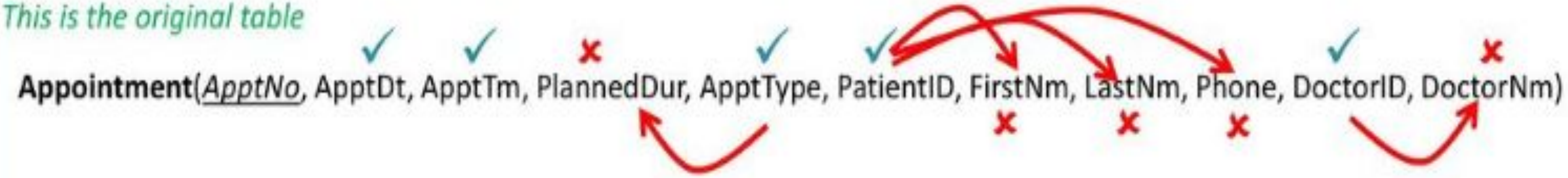
Appt No	Appt Date	Appt Time	Planned Duration	Appt Type	Patient ID	First Nm	Last Nm	Phone	Doctor ID	Doctor Nm
1	12/1/2015	3:00 AM	1.00	Physical	466927	Lisa	Garcia	562-3456	C678	Chapman
2	12/1/2015	3:00 AM	0.25	Shot	456789	Sue	Carey	432-1234	A528	Lopez
3	12/1/2015	3:15 AM	0.50	Flu	194756	Brandon	Pierre	432-7877	S626	Smith
4	12/2/2015	10:00 AM	0.50	Migraine	329657	Marcus	Schwartz	239-5502	A528	Lopez
5	12/2/2015	10:15 AM	0.25	Shot	987453	Mike	Jones	456-0202	G123	Gray
6	12/2/2015	10:30 AM	0.25	Shot	384788	Tonya	Johnson	432-8806	S626	Smith
7	12/2/2015	10:45 AM	0.50	Flu	438754	Iliana	Hnatt	823-4303	C678	Chapman
8	12/2/2015	11:00 AM	1.00	Physical	345875	Carla	Basich	857-5566	A528	Lopez
9	12/3/2015	10:30 AM	1.00	Physical	466927	Lisa	Garcia	562-3456	C678	Chapman
10	12/3/2015	9:00 AM	0.50	Migraine	345875	Carla	Basich	857-5666	C678	Chapman

- It's already in 1NF, and 2NF …but you need to understand why!
  - 1NF: No Multivalued attributes
  - 2NF: No partial dependencies – entire PK determines each non-key attribute

- **Convert this to 3NF. How many tables do you now have?**

Resolving to 3NF

This is the original table



Appointment(*ApptNo*, ApptDt, ApptTm, PlannedDur, ApptType, PatientID, FirstNm, LastNm, Phone, DoctorID, DoctorNm)

3NF SOLUTION:

ApptNo → ApptDt, ApptTm, PlannedDur, ApptType, PatientID, DoctorID

PatientID → FirstNm, LastNm, Phone

DoctorID → DoctorNm

ApptType → PlannedDur

Normalization Practice #2

Appointment Table

Appt No	Appt Date	Appt Time	Appt Type	Patient ID	Doctor ID
1	12/1/2015	3:00 AM	Physical	466927	C678
2	12/1/2015	3:00 AM	Shot	456789	A528
3	12/1/2015	3:15 AM	Flu	194756	S626
4	12/2/2015	10:00 AM	Migraine	329657	A528
5	12/2/2015	10:15 AM	Shot	987453	G123
6	12/2/2015	10:30 AM	Shot	384788	S626
7	12/2/2015	10:45 AM	Flu	438754	C678
8	12/2/2015	11:00 AM	Physical	345875	A528
9	12/3/2015	10:30 AM	Physical	466927	C678
10	12/3/2015	9:00 AM	Migraine	345875	C678

Patient Table

Patient ID	First Nm	Last Nm	Phone
194756	Brandon	Pierre	432-7877
329657	Marcus	Schwartz	239-5502
345875	Carla	Basich	857-5566
384788	Tonya	Johnson	432-8806
438754	Iliana	Hnatt	823-4303
456789	Sue	Carey	432-1234
466927	Lisa	Garcia	562-3456
987453	Mike	Jones	456-0202

Doctor Table

Doctor ID	Doctor Nm
A528	Lopez
C678	Chapman
G123	Gray
S626	Smith

Appt Type Table

Appt Type	Planned Duration
Flu	0.50
Migraine	0.50
Physical	1.00
Shot	0.25

Boyce-Codd Normal Form (BCNF)

- BCNF does not allow dependencies between attributes that belong to candidate keys.
- BCNF is a refinement of the third normal form in which it drops the restriction of a non-key attribute from the 3rd normal form.
- Third normal form and BCNF are not same if the following conditions are true:
  - The table has two or more candidate keys
  - At least two of the candidate keys are composed of more than one attribute
  - The keys are not disjoint i.e. The composite candidate keys share some attributes

Example 1 - Address (Not in BCNF)

Scheme □ {City, Street, ZipCode }

1. **Key1 □ {City, Street }**
2. **Key2 □ {ZipCode, Street}**
3. **No non-key attribute hence 3NF**
4. **{City, Street} □ {ZipCode}**
5. **{ZipCode} □ {City}**
6. **Dependency between attributes belonging to a key**

Boyce Codd Normal Form (BCNF)

Example 2 - Movie (Not in BCNF)

Scheme □ {MovieTitle, MovieID, PersonName, Role, Payment }

1. Key1 □ {MovieTitle, PersonName}
2. Key2 □ {MovieID, PersonName}
3. Both role and payment functionally depend on both candidate keys thus 3NF
4. {MovieID} □ {MovieTitle}
5. Dependency between MovieID & MovieTitle Violates BCNF

Example 3 - Consulting (Not in BCNF)

Scheme □ {Client, Problem, Consultant}

6. Key1 □ {Client, Problem}
7. Key2 □ {Client, Consultant}
8. No non-key attribute hence 3NF
9. {Client, Problem} □ {Consultant}
10. {Client, Consultant} □ {Problem}
11. Dependency between attributess belonging to keys violates BCNF

BCNF - Decomposition

1. Place the two candidate primary keys in separate entities
2. Place each of the remaining data items in one of the resulting entities according to its dependency on the primary key.

Example 1 (Convert to BCNF)

Old Scheme □ {City, Street, ZipCode }

New Scheme1 □ {ZipCode, Street}

New Scheme2 □ {City, Street}

- **Loss of relation {ZipCode} □ {City}**

Alternate New Scheme1 □ {ZipCode, Street }

Alternate New Scheme2 □ {ZipCode, City}

Decomposition – Loss of Information

1. If decomposition does not cause any loss of information it is called a **lossless** decomposition.
2. If a decomposition does not cause any dependencies to be lost it is called a **dependency-preserving** decomposition.
3. Any table scheme can be decomposed in a lossless way into a collection of smaller schemas that are in BCNF form. However the dependency preservation is not guaranteed.
4. Any table can be decomposed in a lossless way into 3rd normal form that also preserves the dependencies.
   - 3NF may be better than BCNF in some cases

Use your own judgment when decomposing schemas

BCNF - Decomposition

Example 2 (Convert to BCNF)

Old Scheme □ {MovieTitle, MovieID, PersonName, Role, Payment }

New Scheme □ {MovieID, PersonName, Role, Payment}

New Scheme □ {MovieTitle, PersonName}

- Loss of relation {MovieID} □ {MovieTitle}

New Scheme □ {MovieID, PersonName, Role, Payment}

New Scheme □ {MovieID, MovieTitle}

- We got the {MovieID} □ {MovieTitle} relationship back

Example 3 (Convert to BCNF)

Old Scheme □ {Client, Problem, Consultant}

New Scheme □ {Client, Consultant}

New Scheme □ {Client, Problem}

Likely question be like::

1. What are the Informal Guidelines for developing relational schema? What is the need of it?
2. What is functional dependency? Explain the different types of functional dependencies.
3. What is need for normalization? Explain the various normal forms with an example.
4. Is BCNF better than 3NF? Justify your answer.
5. Problems on normalization……

THE END