Database Systems and Database System Design

Normalization

Objective of this lecture

- Understand the concept of normalization.
- Be able to apply the normalization with the table.

Purpose of the normalization

- Normalization is

"A technique for producing a set of relations with desirable properties, given the data requirements of an enterprise."

- The normalization process is to identify a **suitable** set of relation that support the data requirements of an enterprise.

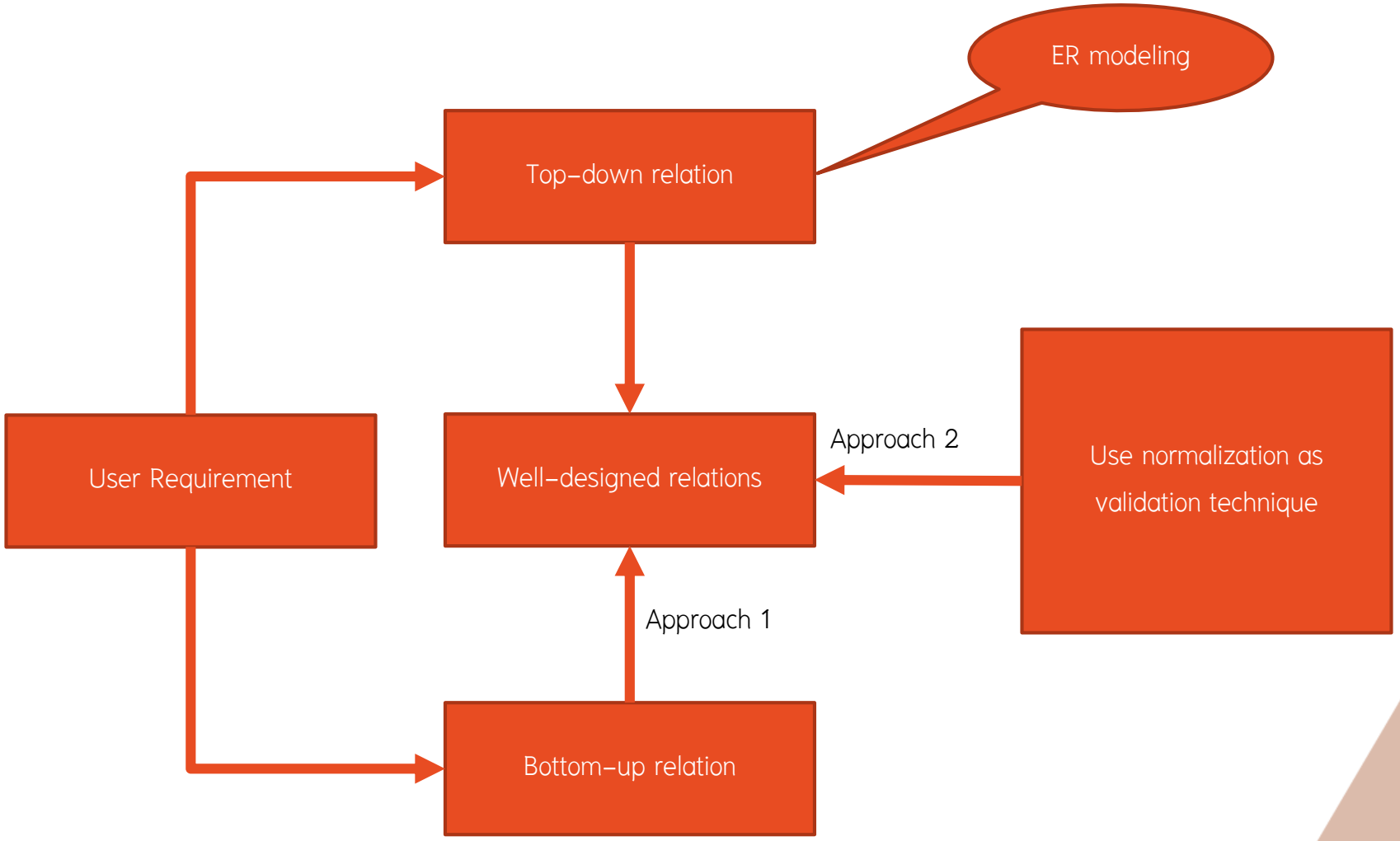
- The aim of the normalization is to avoid unnecessary duplication of data.

Purpose of the normalization

- There are 2 ways to use the normalization.

- Approach 01
  - Bottom up approach
  - Not suitable for large database

- Approach 02
  - Validation technique
  - Check the structure of the database


ER modeling
Top-down relation
User Requirement
Well-designed relations
Approach 2
Use normalization as validation technique
Approach 1
Bottom-up relation

Well-designed relation

Minimum Null Value

- The NULL value in the database represent the unknown value.

- A good database should structurally allow the smallest number of NULL value.

Minimum Null Value

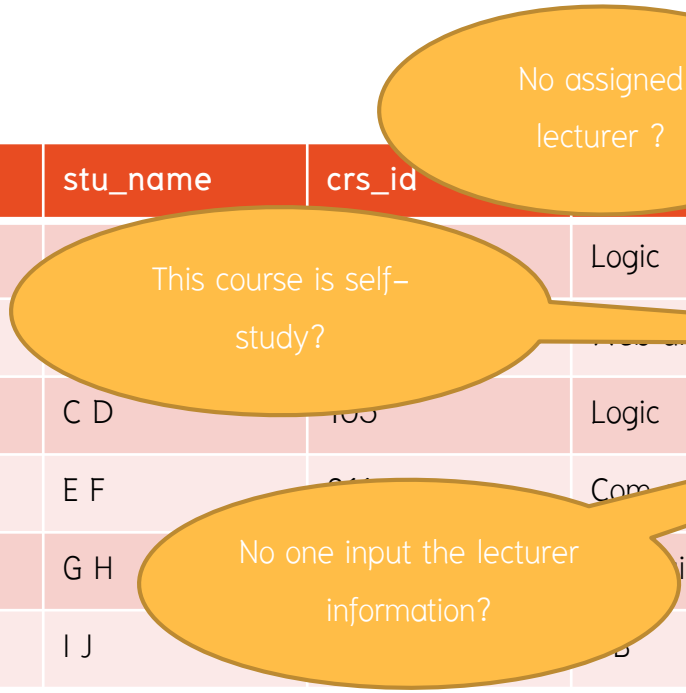
STUDENT

stu_id	stu_name	crs_id	crs_name	crs_credit	lec_id
2001	A B	103	Logic	3	101
2001	A B	104	Web ui	3	NULL
2020	C D	103	Logic	3	101
2050	E F	211	Com org	3	102
2060	G H	104	Web ui	3	NULL
2069	I J	212	DB	4	NULL

Minimum Null Value

STUDENT

stu_id	stu_name	crs_id		crs_credit	lec_id
2001			Logic	3	101
2001					NULL
2020	C D		Logic	3	101
2050	E F		Com	3	102
2060	G H			3	NULL
2069	I J			4	NULL



Data Redundancy

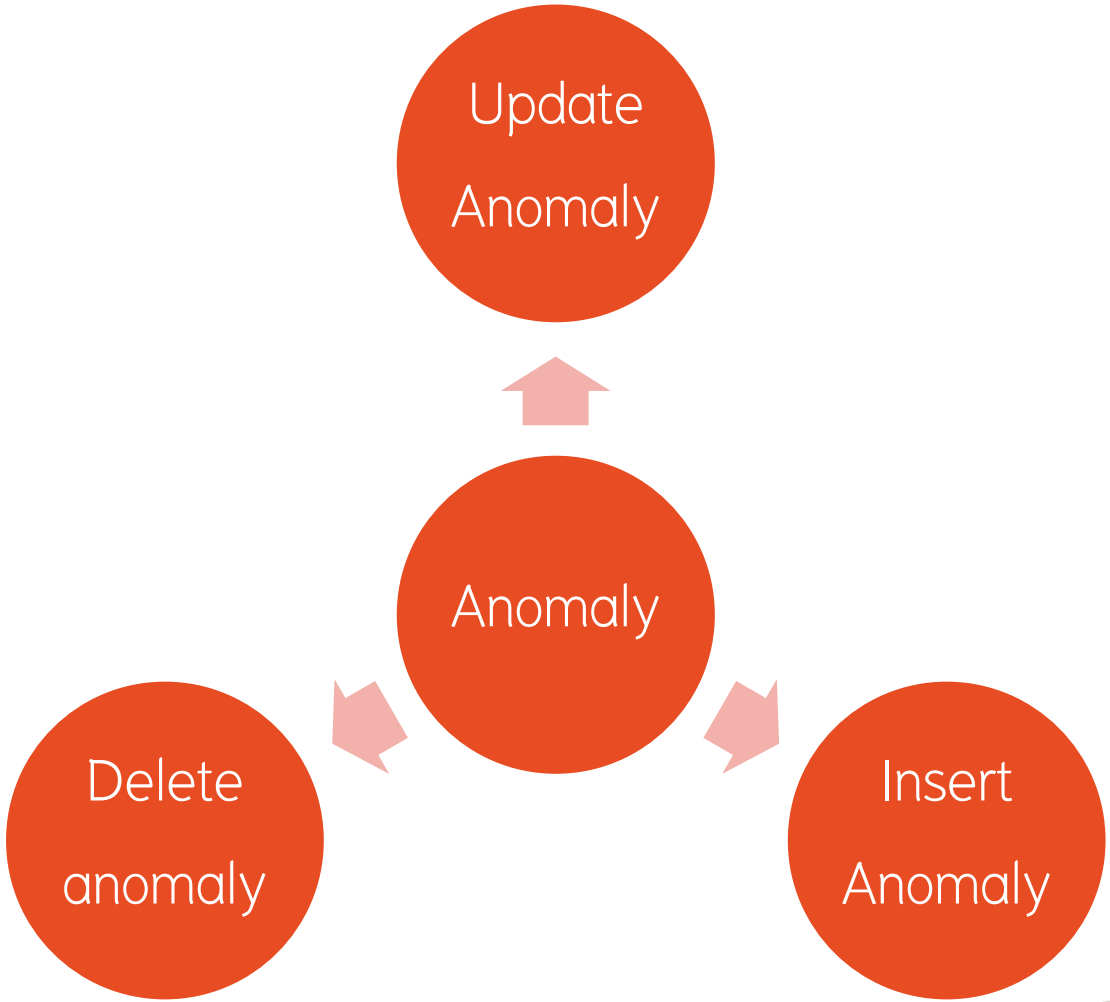
- Data redundancy means there are more than one copy of information in the DB.
    - Waste of space
    - Risk of conflict (just like the file-based system)
- Every insert, update and delete operation needs to perform at every copy of the data.
- Some information might lose forever.
- This operations could lead to **anomaly**.

Data Redundancy

STUDENT

stu_id	stu_name	crs_id	crs_name	crs_credit	lec_id
2001	A B	103	Logic	3	101
2001	A B	104	Web ui	3	NULL
2020	C D	103	Logic	3	101
2050	E F	211	Com org	3	102
2060	G H	104	Web ui	3	NULL
2069	I J	212	DB	4	NULL

Data Redundancy

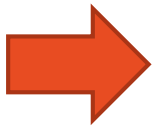


Delete Anomaly

- When you delete a tuple from a table, some information might lost forever.

- In poor design, the data tends to group together in one place.
  - If you delete a data, other data might be effected.

Delete Anomaly

Delete  Delete the student but the course get deleted, too.

STUDENT

stu_id	stu_name	crs_id	crs_name	crs_credit	lec_id
2001	A B	103	Logic	3	101
2001	A B	104	Web ui	3	NULL
2020	C D	103	Logic	3	101
2050	E F	211	Com org	3	102
2060	G H	104	Web ui	3	NULL
2069	I J	212	DB	4	NULL

Insert Anomaly

- To insert the a new tuple, we must insert detail of all information into the table.
  - The data must be consistent with the existing information.

- For a relation with redundant data, we have to insert the data which is exactly the same with other one.

- There is a chance that the new data is not consistent with the original data.

Insert Anomaly

Insert {2069, X Y, 212, Web ui, 4, 101}

STUDENT

stu_id	stu_name	crs_id	crs_name	crs_credit	lec_id
2001	A B	103	Logic	3	101
2001	A B	104	Web ui	3	NULL
2020	C D	103	Logic	3	101
2050	E F	211	Com org	3	102
2060	G H	104	Web ui	3	NULL
2069	I J	212	DB	4	NULL

Insert Anomaly

Insert {2069, X Y, 212, Web ui, 4, 101}

STUDENT

stu_id	stu_name	crs_id	crs_name	crs_credit	lec_id
2001	A B	103	Logic	3	101
2001	A B	104	Web ui	3	NULL
2020	C D	103	Logic	3	101
2050	E F	211	Com org	3	102
2060	G H	104	Web ui	3	NULL
2069	I J	212	DB	[illegible]	NULL
2069	X Y	212	Web ui	[illegible]	101

Insert Anomaly

- To insert the a new tuple, we must insert detail of all information into the table.
  - The data must be consistent with the existing information.

- To insert a new tuple with some partially unknown, we must insert the NULL value.
  - The allowance of NULL value is a poor design.

Insert Anomaly

Insert {2069, I J, 231, OOP, 4, NULL}

STUDENT

stu_id	stu_name	crs_id	crs_name	crs_credit	lec_id
2001	A B	103	Logic	3	101
2001	A B	104	Web ui	3	NULL
2020	C D	103	Logic	3	101
2050	E F	211	Com org	3	102
2060	G H	104	Web ui	3	NULL
2069	I J	212	DB	4	NULL

Insert Anomaly

Insert {2069, I J, 231, OOP, 4, NULL}

STUDENT

stu_id	stu_name	crs_id	crs_name	crs_credit	lec_id
2001	A B	103	Logic	3	101
2001	A B	104	Web ui	3	NULL
2020	C D	103	Logic	3	101
2050	E F	211	Com org	3	102
2060	G H	104	Web ui	3	NULL
2069	I J	212	DB	4	NULL
2069	I J	231	OOP	4	NULL

Update Anomaly

- When you update a tuple from a table, you need to update the value at every position.

- If you miss some place, the data in the table will be inconsistent.

Update Anomaly

{2001, “A B”} ⇒ {2001, “Z Y”}

STUDENT

stu_id	stu_name	crs_id	crs_name	crs_credit	lec_id
2001	A B	103	Logic	3	101
2001	A B	104	Web ui	3	NULL
2020	C D	103	Logic	3	101
2050	E F	211	Com org	3	102
2060	G H	104	Web ui	3	NULL
2069	I J	212	DB	4	NULL

Update Anomaly

{2001, “A B”} → {2001, “Z Y”}

STUDENT

stu_id	stu_name	crs_id	crs_name	crs_credit	lec_id
2001	A B	103	Logic	3	101
2001	Z Y	104	Web ui	3	NULL
2020	C D	103	Logic	3	101
2050	E F	211	Com org	3	102
2060	G H	104	Web ui	3	NULL
2069	I J	212	DB	4	NULL

Solution

Solution

STUDENT_LARGE

stu_id	stu_name	crs_id	crs_name	cr…redit	lec_id
2001	A B	103	Logic	3	101
2001	A B	104	Web ui	3	NULL
2020	C D	103	Logic	3	101
2050	E F	211	Com org	3	102
2060	G H	104	Web ui	3	NULL
2069	J	212	DB	4	NULL

Problem

Problem

Problem

Solution

"Split the table into smaller table"

Solution

STUDENT

stu_id	stu_name
2001	A B
2020	C D
2050	E F
2060	G H
2069	I J

COURSE

crs_id	crs_name	crs_credit	lec_id
103	Logic	3	101
104	Web ui	3	NULL
211	Com org	3	102
212	DB	4	NULL

ENROLLMENT

stu_id	crs_id
2001	103
2001	104
2020	103
2050	211
2060	104
2069	212

Solution

- Consider the insert, update, delete anomaly in the new diagram.
- What about the redundancy?
- What about the NULL value?

Solution

- Decompose a large relation into a smaller atomic relations.

- Important properties :
  - Lossless-join

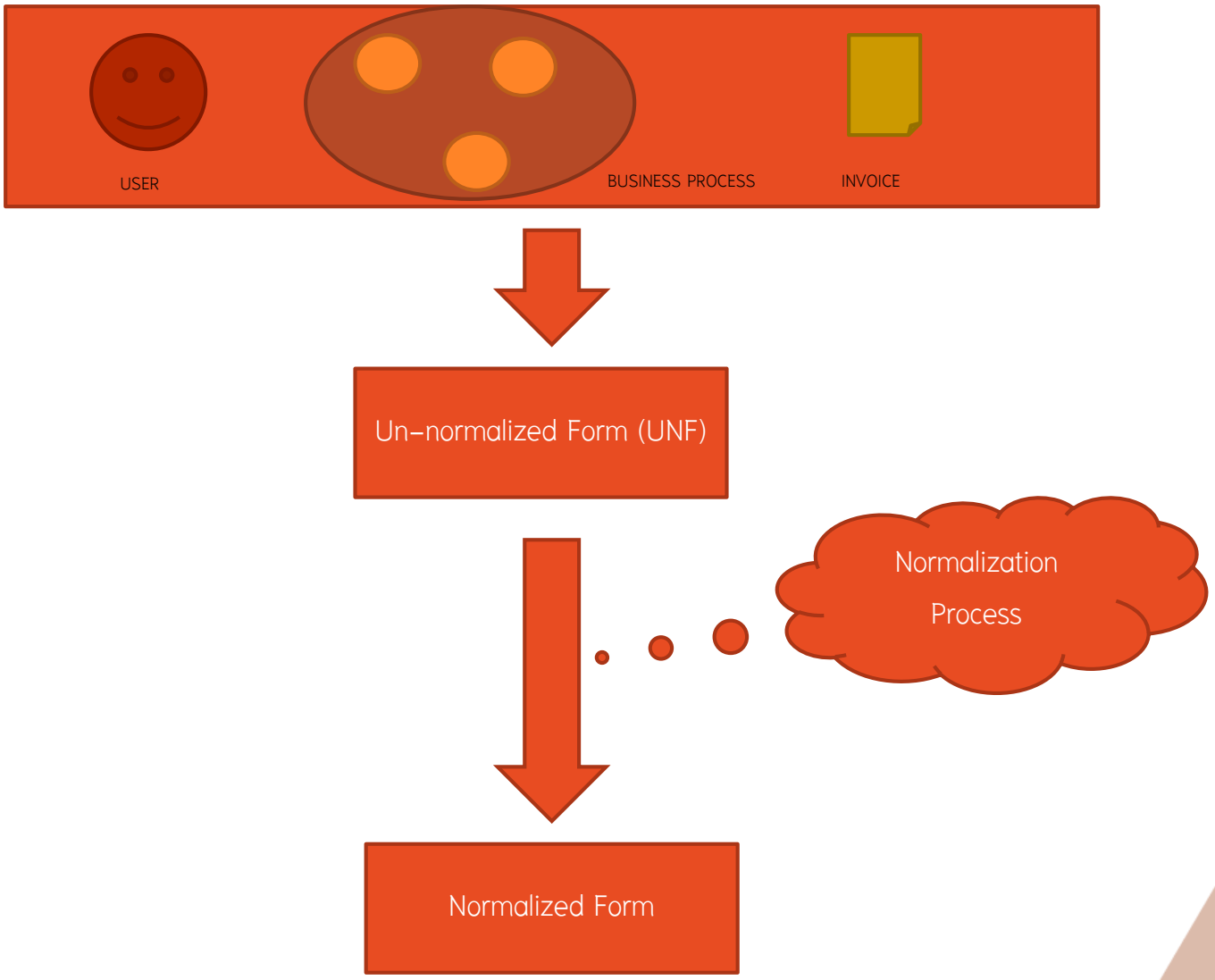
    "Any tuple of the original relation could be identified from the decomposed relation."

  - Dependency preservation

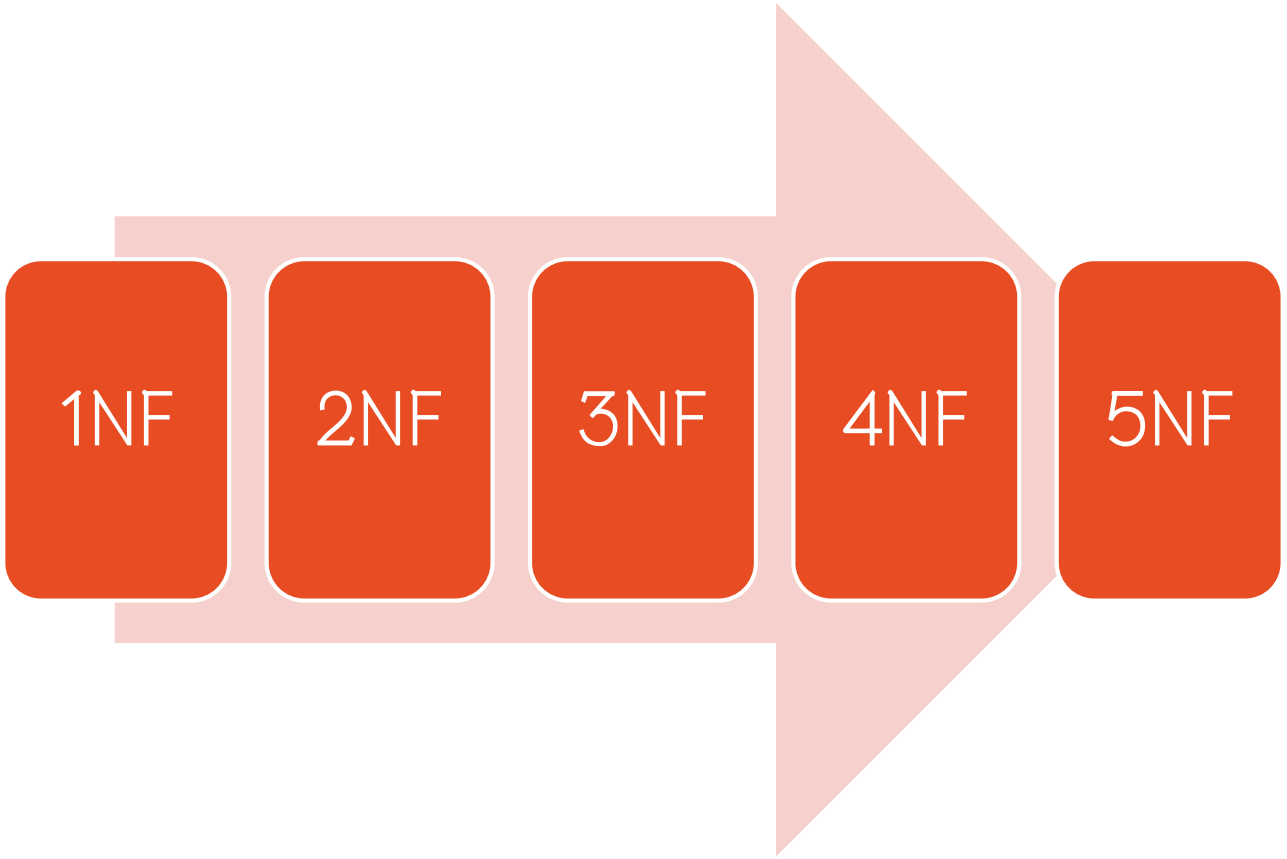
    "Any constraint of the original relation is maintain in the decomposed relation."

Normalization

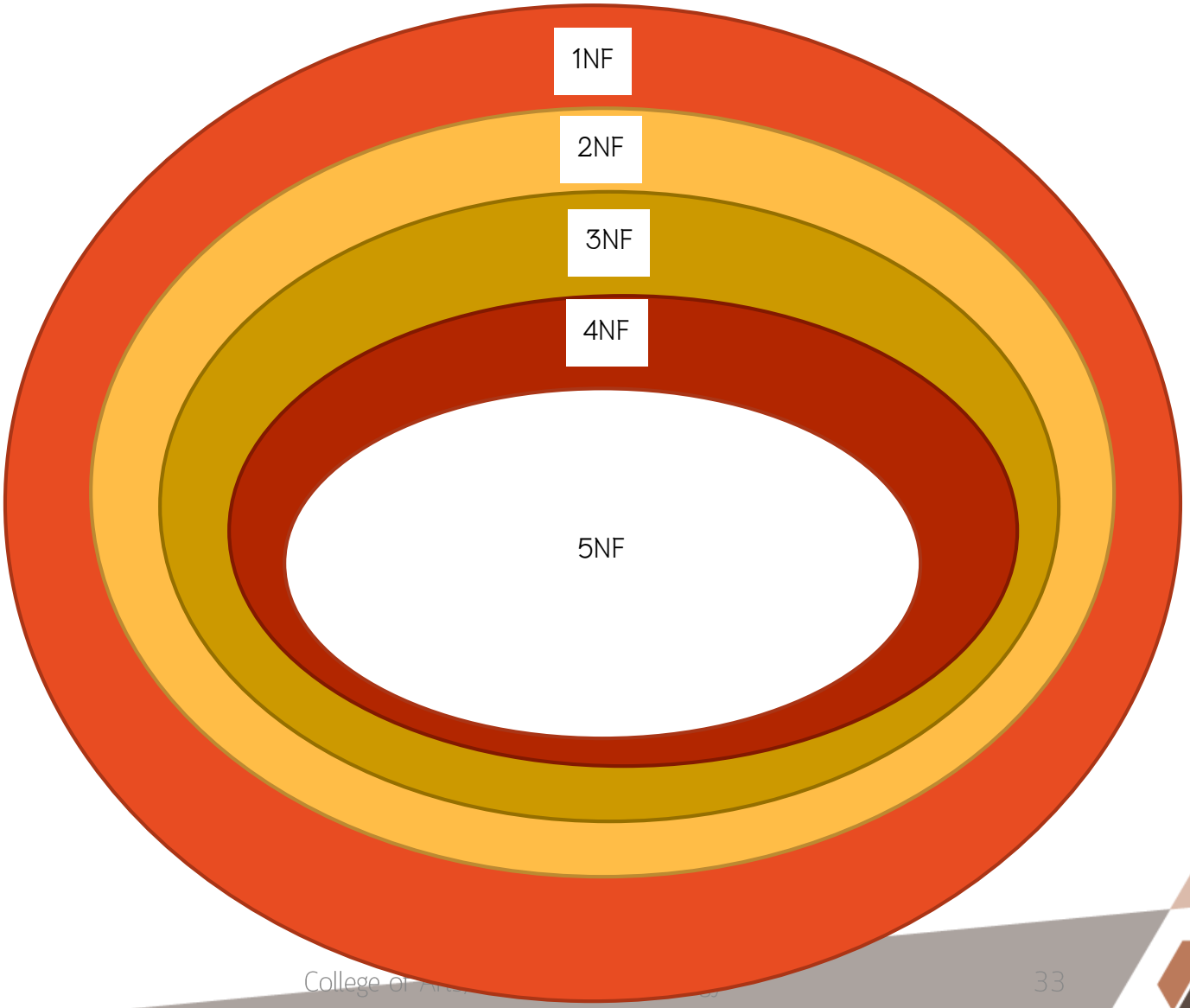
DATA SOURCE


USER
BUSINESS PROCESS
INVOICE
Un-normalized Form (UNF)
Normalization
Process
Normalized Form

Normalization



Normalization



1st Normal Form

- First normal form

"A relation in which the intersection of each row and column contains one and only one value"

- Vert important !!!!

- Does not allow variable length lists
- Does not allow repeated fields.

EMPLOYEE_SKILL

emp_id	emp_name	dept_id	emp_skill	dept_name
001	AA BB	01	{swimming, typing}	sale
002	CC DD	04	{typing}	marketing
003	EE FF	01	{music,art,drawing}	sale

OR

EMPLOYEE_SKILL

emp_id	emp_name	dept_id	emp_skill	dept_name
001	AA BB	01	swimming	sale
			typing	
002	CC DD	04	typing	marketing
003	EE FF	01	music	sale
			art	
			drawing	

1NF : Approach 1

- Decompose into a new set of relations.

- Extract the primary key of the original table and repeating value.
  - Treat each repeating values as a new tuple

- Separate them into a new relation and set all of them to be a new primary key.

emp_id	emp_name	dept_id	emp_skill	dept_name
001			{swimming, typing}	sale
002	CC DD	04	{typing}	marketing
003	EE FF	01	{music,art,drawing}	sale

EMPLOYEE

emp_id	emp_name	dept_id	dept_name
001	AA BB	01	sale
002	CC DD	04	marketing
003	EE FF	01	sale

SKILL

emp_id	emp_skill
001	swimming
001	typing
002	typing
003	music
003	art
003	drawing

1NF : Approach 2

- Change the primary key
- Separate the repeated value into a new tuple
- Fill in the empty value with the corresponding value
- Use the repeating value attribute to be a primary key

EMPLOYEE_SKILL

emp_id	emp_name	dept_id	emp_skill	dept_name
001	AA BB	01	swimming	Sale
001	AA BB	01	typing	Sale
002	CC DD	04	typing	marketing
003	EE FF	01	music	Sale
003	EE FF	01	art	Sale
003	EE FF	01	drawing	Sale

1NF : Approach 3

- Change the structure of the attribute to cope with the repeating value.

- Determine the maximum repeated value and create the new attributes according to the possible value.

- Do not recommended !!!!!!!!!!
  - Use the many-to-many relationship (Approach 1)
  - DO not use!!!!!!!!!!

EMPLOYEE_SKILL

emp_id	emp_name	dept_id	emp_skill_1	emp_skill_2	emp_skill_3	dept_name
001	AA BB	01	swimming	typing		Sale
002	CC DD	04	typing			marketing
003	EE FF	01	music	art	drawing	Sale

1st Normal Form

- Rule of Thumb
  - Look for repeating group
  - If there is no primary key, you must assign one.
  - You can choose one of the 2 approaches (1 and 2) but you have to think about the context and performance.

Example : 1NF

Table name: RPT_FORMAT

Database name: Ch05_ConstructCo

PK

PROJ_NUM	PROJ_NAME	EMP_NUM	EMP_NAME	JOB_CLASS	CHG_HOUR	HOURS
15	Evergreen	103	June E. Arbough	Elect. Engineer	$84.50	23.8
		101	John G. News	Database Designer	$105.00	19.4
		105	Alice K. Johnson *	Database Designer	$105.00	35.7
		106	William Smithfield	Programmer	$35.75	12.6
		102	David H. Senior	Systems Analyst	$96.75	23.8
18	Amber Wave	114	Annelise Jones	Applications Designer	$48.10	24.6
		118	James J. Frommer	General Support	$18.36	45.3
		104	Anne K. Ramoras *	Systems Analyst	$96.75	32.4
		112	Darlene M. Smithson	DSS Analyst	$45.95	44.0
22	Rolling Tide	105	Alice K. Johnson	Database Designer	$105.00	64.7
		104	Anne K. Ramoras	Systems Analyst	$96.75	48.4
		113	Delbert K. Joenbrood *	Applications Designer	$48.10	23.6
		111	Geoff B. Wabash	Clerical Support	$26.87	22.0
		106	William Smithfield	Programmer	$35.75	12.8
25	Starflight	107	Maria D. Alonzo	Programmer	$35.75	24.6
		115	Travis B. Bawangi	Systems Analyst	$96.75	45.8
		101	John G. News *	Database Designer	$105.00	56.3
		114	Annelise Jones	Applications Designer	$48.10	33.1
		108	Ralph B. Washington	Systems Analyst	$96.75	23.6
		118	James J. Frommer	General Support	$18.36	30.5
		112	Darlene M. Smithson	DSS Analyst	$45.95	41.4

Example : 1NF

Table name: RPT_FORMAT **Database name: Ch05_ConstructCo**

PROJ_NUM	PROJ_NAME	EMP_NUM	EMP_NAME	JOB_CLASS	CHG_HOUR	HOURS
15	Evergreen	103	June E. Arbough	Elect. Engineer	$84.50	23.8
		101	John G. News	Database Designer	$105.00	19.4
		105	Alice K. Johnson *	Database Designer	$105.00	35.7
		106	William Smithfield	Programmer	$35.75	12.6
		102	David H. Senior	Systems Analyst	$96.75	23.8
18	Amber Wave	114	Annelise Jones	Applications Designer	$48.10	24.6
		118	James J. Frommer	General Support	$18.36	45.3
		104	Anne K. Ramoras *	Systems Analyst	$96.75	32.4
		112	Darlene M. Smi	alyst	$45.95	44.0
22	Rolling Tide	105	Alice K. Johns	se Designer	$105.00	64.7
		104	Anne K. Ramor	s Analyst	$96.75	48.4
		113	Delbert K. Joenbrood *	Applications Designer	$48.10	23.6
		111	Geoff B. Wabash	Clerical Support	$26.87	22.0
		106	William Smithfield	Programmer	$35.75	12.8
25	Starflight	107	Maria D. Alonzo	Programmer	$35.75	24.6
		115	Travis B. Bawangi	Systems Analyst	$96.75	45.8
		101	John G. News *	Database Designer	$105.00	56.3
		114	Annelise Jones	Applications Designer	$48.10	33.1
		108	Ralph B. Washington	Systems Analyst	$96.75	23.6
		118	James J. Frommer	General Support	$18.36	30.5
		112	Darlene M. Smithson	DSS Analyst	$45.95	41.4

Repeating group

1NF : Approach 1

- Decompose into a new set of relations.

- **Extract** the primary key of the original table and repeating value.
  - Treat each repeating values as a new tuple
  - Remove the value in the original

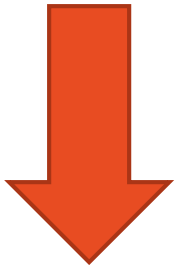
PROJ_NUM is a primary key

EMP_NUM is a repeating group

- Separate them into a new relation and set primary key of the original table and repeating value to be a new primary key.

EMP_NAME, JOB_CLASS and CHO_HOUR is related to EMP_NUM

proj_num	proj_name	emp_num	emp_name	job_class	cho_hour	hours



1st approach

proj_num	proj_name

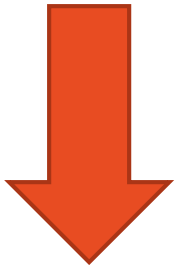
proj_num	emp_num	emp_name	job_class	cho_hour	hours

1NF : Approach 2

- Change the primary key
- Separate the repeated value into a new tuple
- Use the repeating value attribute to be a another primary key

EMP_NUM

proj_num	proj_name	emp_num	emp_name	job_class	cho_hour	hours



2nd approach

proj_num	proj_name	emp_num	emp_name	job_class	cho_hour	hours

2nd Normal Form

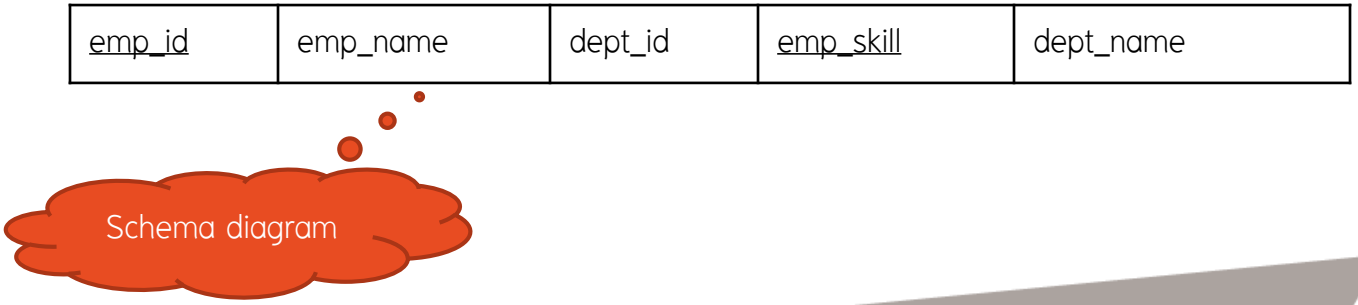
- Second normal form

> "A relation that is in the First Normal Form and every non–primary–key attribute is fully functionally dependent on the primary key"

- In other word, there is no partial functional dependency on the primary key.

- The partial functional dependency means
  - The primary key has more than one attribute.
  - Some attributes in the primary key uniquely refer to other attributes.

emp_id	emp_name	dept_id	emp_skill	dept_name
001	AA BB	01	swimming	Sale
001	AA BB	01	typing	Sale
002	CC DD	04	typing	marketing
003	EE FF	01	music	Sale
003	EE FF	01	art	Sale
003	EE FF	01	drawing	Sale

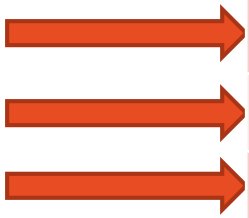


client_id	property_id	rent_start	rent_end
001	01	1/1/2017	2/1/2017
002	01	1/2/2017	2/2/2017
001	02	1/3/2017	2/3/2017

fully functional dependency

If you remove any attribute from the key, the fully functional dependency does not hold anymore.

client_id	property_id
001	01
002	01
001	02



rent_start	rent_end
1/1/2017	2/1/2017
1/2/2017	2/2/2017
1/3/2017	2/3/2017

fully functional dependency

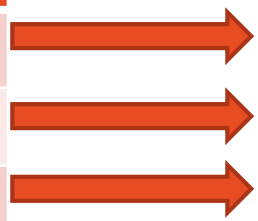
You **cannot** use only the client_id or property_id to refer to a record. You need both!!!!

emp_id	emp_name	dept_id	emp_skill	dept_name
001	AA BB	01	swimming	Sale
001	AA BB	01	typing	Sale
002	CC DD	04	typing	marketing
003	EE FF	01	music	Sale
003	EE FF	01	art	Sale
003	EE FF	01	drawing	Sale

partial functional dependency

If you remove any attribute from the key, the partial functional dependency still hold.

emp_id	emp_name	dept_id	dept_name
001	AA BB	01	Sale
002	CC DD	04	marketing
003	EE FF	01	Sale



partial functional dependency

Emp_id can uniquely refer to Emp_name.

(Emp_skill is not needed!!!!)

Problem

- Data redundancy

e[illegible]	e[illegible]ame	dept_[illegible]	emp_skill	dept_[illegible]
001	AA BB	01	swimming	Sale
001	AA BB	01	typing	Sale
002	CC DD	04	typing	marketing
003	EE FF	01	music	Sale
003	EE FF	01	art	Sale
003	EE FF	01	drawing	Sale

Approach

- Decompose the problematic relation into a smaller one.
- Extract the partial functional dependency from the original relation.
- The primary key and non-related attribute remain the same.

Approach

- Decompose the problematic relation into a smaller one.
- Identify the fully functional dependency and partial functional dependency
- Extract the partial functional dependency from the original relation.
  - The primary key of the partial functional dependency remain the same.

emp_id  emp_name, dept_id,dept_name

- The remaining set of attribute remains the same.

emp_id,emp_skill

EMPLOYEE

emp_id	emp_name	dept_id	dept_name
001	AA BB	01	sale
002	CC DD	04	marketing
003	EE FF	01	sale

SKILL

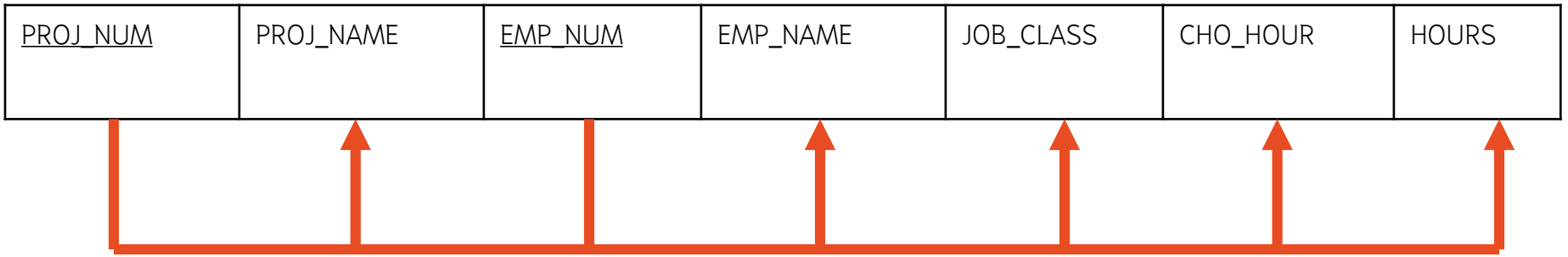
emp_id	emp_skill
001	swimming
001	typing
002	typing
003	music
003	art
003	drawing

Primary Key

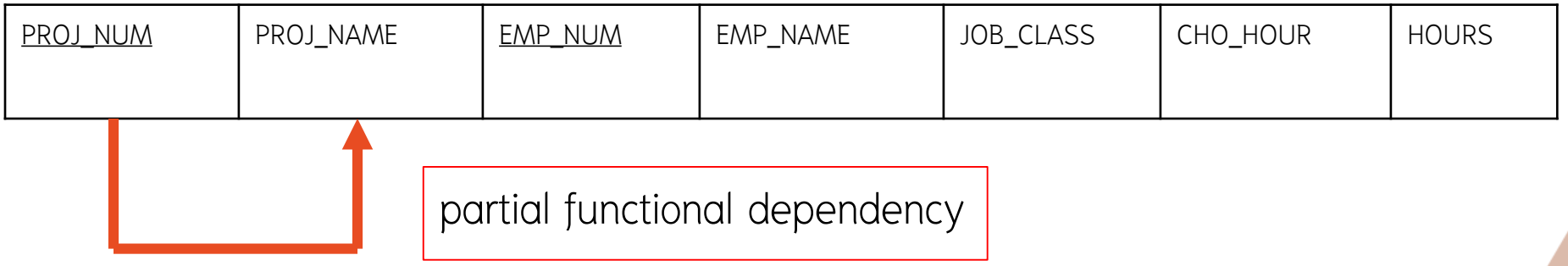
Primary Key

Table name: DATA_ORG_1NF **Database name: Ch05_ConstructCo**

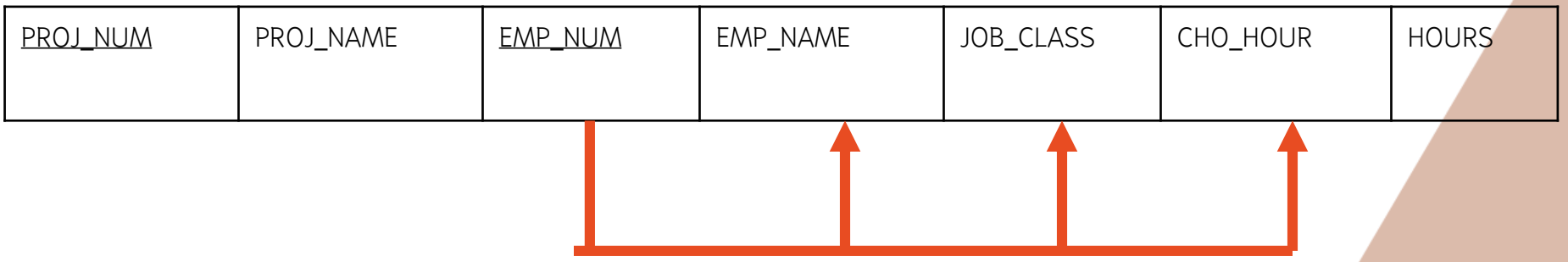
PROJ_NUM	PROJ_NAME	EMP_NUM	EMP_NAME	JOB_CLASS	CHG_HOUR	HOURS
15	Evergreen	103	June E. Arbough	Elect. Engineer	$84.50	23.8
15	Evergreen	101	John G. News	Database Designer	$105.00	19.4
15	Evergreen	105	Alice K. Johnson *	Database Designer	$105.00	35.7
15	Evergreen	106	William Smithfield	Programmer	$35.75	12.6
15	Evergreen	102	David H. Senior	Systems Analyst	$96.75	23.8
18	Amber Wave	114	Annelise Jones	Applications Designer	$48.10	24.6
18	Amber Wave	118	James J. Frommer	General Support	$18.36	45.3
18	Amber Wave	104	Anne K. Ramoras *	Systems Analyst	$96.75	32.4
18	Amber Wave	112	Darlene M. Smithson	DSS Analyst	$45.95	44.0
22	Rolling Tide	105	Alice K. Johnson	Database Designer	$105.00	64.7
22	Rolling Tide	104	Anne K. Ramoras	Systems Analyst	$96.75	48.4
22	Rolling Tide	113	Delbert K. Joenbrood *	Applications Designer	$48.10	23.6
22	Rolling Tide	111	Geoff B. Wabash	Clerical Support	$26.87	22.0
22	Rolling Tide	106	William Smithfield	Programmer	$35.75	12.8
25	Starflight	107	Maria D. Alonzo	Programmer	$35.75	24.6
25	Starflight	115	Travis B. Bawangi	Systems Analyst	$96.75	45.8
25	Starflight	101	John G. News *	Database Designer	$105.00	56.3
25	Starflight	114	Annelise Jones	Applications Designer	$48.10	33.1
25	Starflight	108	Ralph B. Washington	Systems Analyst	$96.75	23.6
25	Starflight	118	James J. Frommer	General Support	$18.36	30.5
25	Starflight	112	Darlene M. Smithson	DSS Analyst	$45.95	41.4



fully functional dependency



partial functional dependency



partial functional dependency

Result

PROJ_NUM	EMP_NUM	HOURS

PROJ_NUM	PROJ_NAME

EMP_NUM	EMP_NAME	JOB_CLASS	CHO_HOUR

Challenge #1

part_id	part_name	warehouse_id	warehouse_address	quantity
1	Battery	1	Chiang Mai	1,000
2	Wheel	1	Chiang Mai	500
1	Battery	2	Lampang	500
4	Rear lamp	3	Lamphun	250

Challenge #1 Solution

part_id	warehouse_id	quantity
1	1	1,000
2	1	500
1	2	500
4	3	250

part_id	part_name
1	Battery
2	Wheel
4	Rear lamp

warehouse_id	warehouse_address
1	Chiang Mai
2	Lampang
3	Lamphun

3rd Normal Form

- Third normal form

> *"A relation that is in first and second normal form and in which no non-primary-key attribute is transitively dependent on the primary key"*

- The transitive functional dependency means
  - A implies B and B implies C
  - A will also implies C

- In other word, there is no partial functional dependency on the non-primary key.

Transitive functional dependency

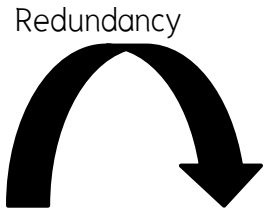
EMPLOYEE

emp_id	emp_name	dept_id	dept_name
001	AA BB	01	sale
002	CC DD	04	marketing
003	EE FF	01	sale

transitive functional dependency

- Emp_id is uniquely pointed to Dept_id
- Dept_id is uniquely pointed to Dept_name
- Emp_id is uniquely pointed to Dept_name

Problem

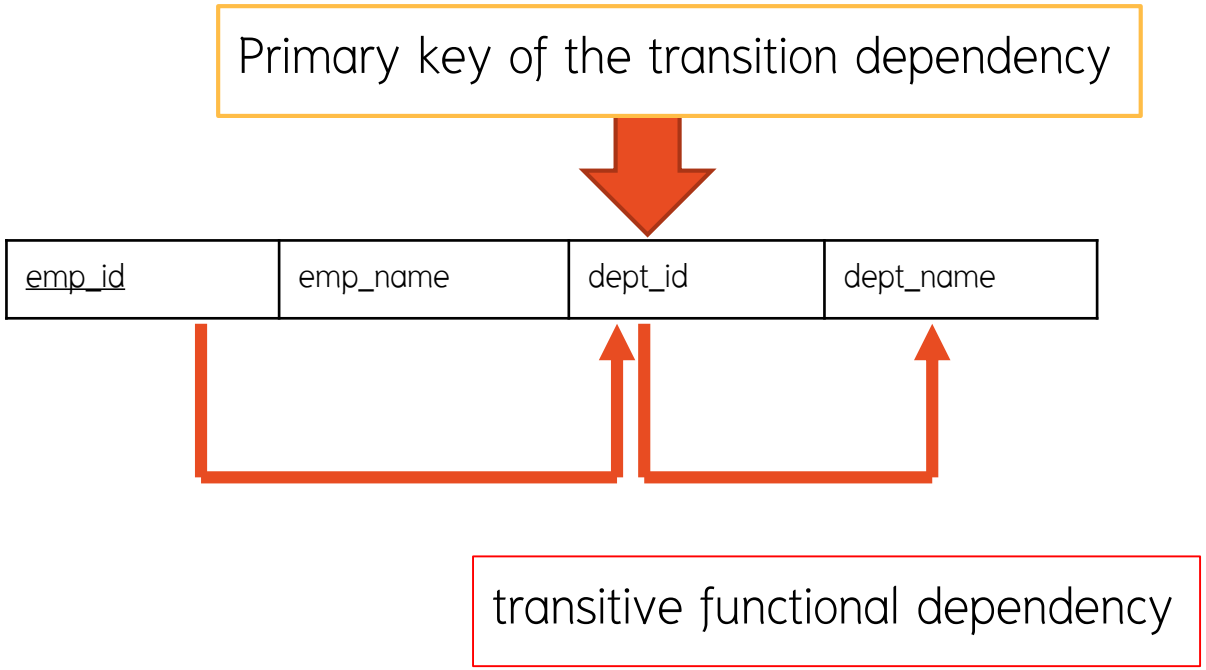


EMPLOYEE

emp_id	emp_name	dept_id	dept_name
001	AA BB	01	sale
002	CC DD	04	marketing
003	EE FF	01	sale

Approach

- Identify the transitive dependency
- Identify the primary key of the transitive dependency
- Extract the transitive relation from the original relation.
- The key of the transitive relation remains the same in the original relation as a foreign key.


Primary key of the transition dependency
emp_id
emp_name
dept_id
dept_name
transitive functional dependency

Result

emp_id	emp_name	dept_id

emp_id	emp_name	dept_id
001	AA BB	01
002	CC DD	04
003	EE FF	01

dept_id	dept_name

dept_id	dept_name
01	sale
04	marketing

Primary Key

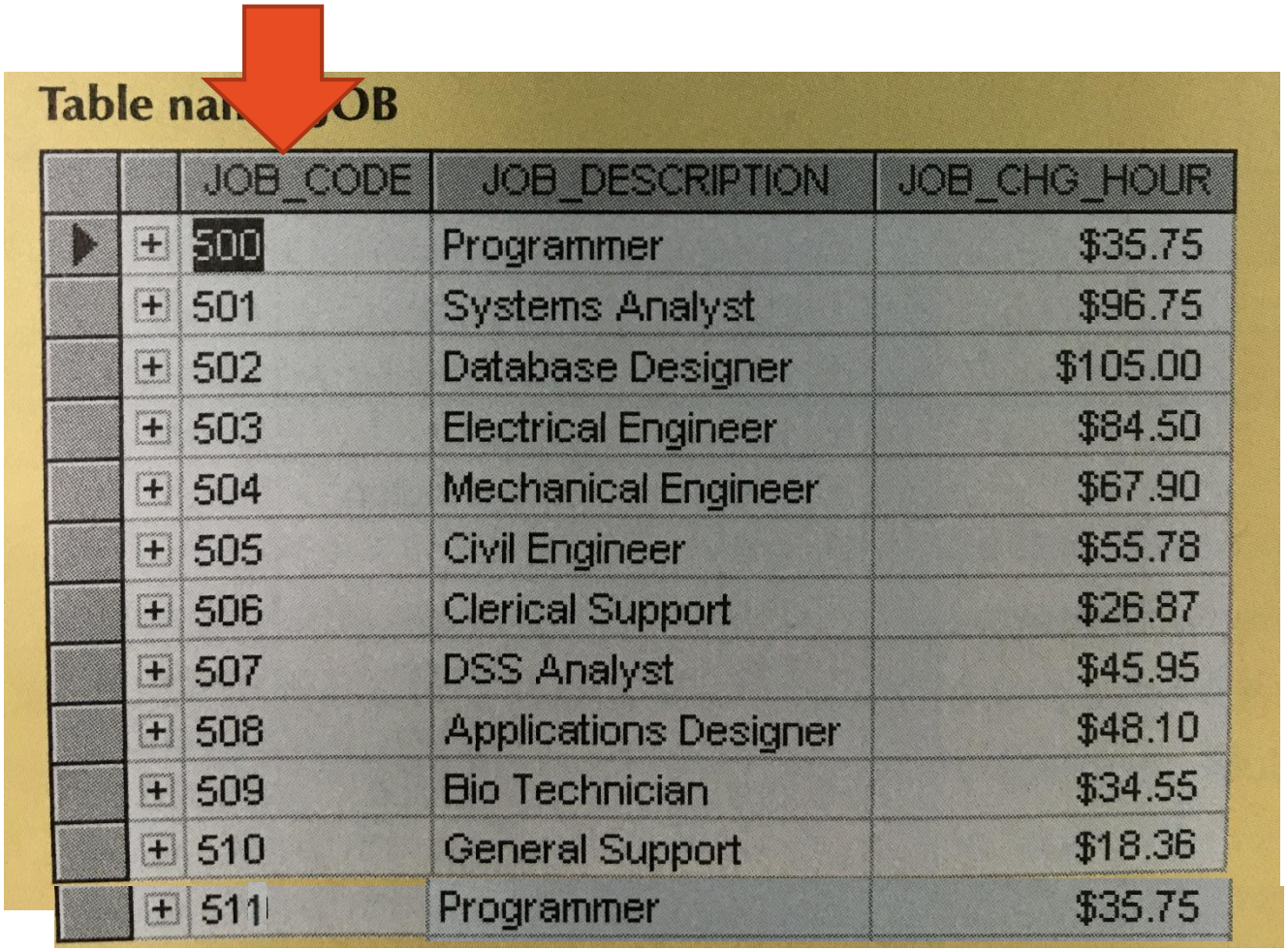


Table nam JOB

JOB_CODE	JOB_DESCRIPTION	JOB_CHG_HOUR
500	Programmer	$35.75
501	Systems Analyst	$96.75
502	Database Designer	$105.00
503	Electrical Engineer	$84.50
504	Mechanical Engineer	$67.90
505	Civil Engineer	$55.78
506	Clerical Support	$26.87
507	DSS Analyst	$45.95
508	Applications Designer	$48.10
509	Bio Technician	$34.55
510	General Support	$18.36
511	Programmer	$35.75

Job_code	Job_description	Job_chg_hour

Job_code	Job_description

Job_description	Job_chg_hour

Challenge # 2

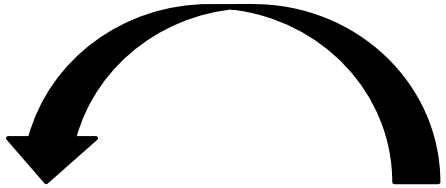
Boyce-Codd Normal form

- Boyce-Codd Normal form

"A relation is in Boyce–Codd Normal form (BCNF), if every determinant is a candidate key."

- The determinant is a attribute / a set of attribute that is uniquely referred to a certain value.
  - The determinant is candidate key.

- In other word, there is no non-primary key attribute uniquely referred to the key.



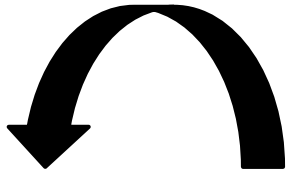
Emp_num	Emp_specialty	manager
0001	C++	A
0001	JAVA	B
0002	XML	C
0003	JAVA	B
0003	XML	E

manager is uniquely referred to Emp_specialty.

Problem

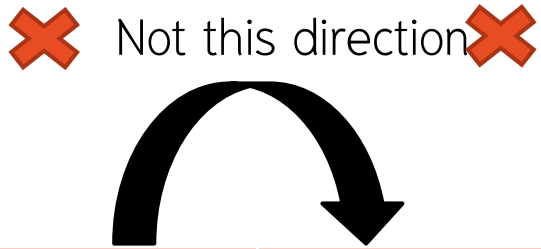
The non-key attribute uniquely referred to the key attribute .

Data Redundancy



Emp_num	Emp_specialty	manager
0001	C++	A
0001	JAVA	B
0002	XML	C
0003	JAVA	B
0003	XML	E

Problem



Emp_num	Emp_specialty	manager
0001	C++	A
0001	JAVA	B
0002	XML	C
0003	JAVA	B
0003	XML	E

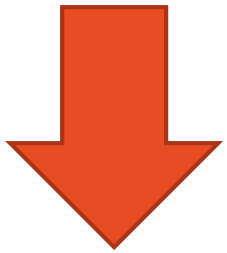
Approach

- Extract the problematic non-key attribute and key attribute.

- Create a new relation
  - Set the non-key attribute that uniquely points to the key in the original relation to be a primary key in the new relation.
  - The pointed key in original relation is converted to be only attribute.

- Use the non-key attribute to be a new key in the original relation.

Emp_num	Emp_specialty	manager



Emp_num	manager

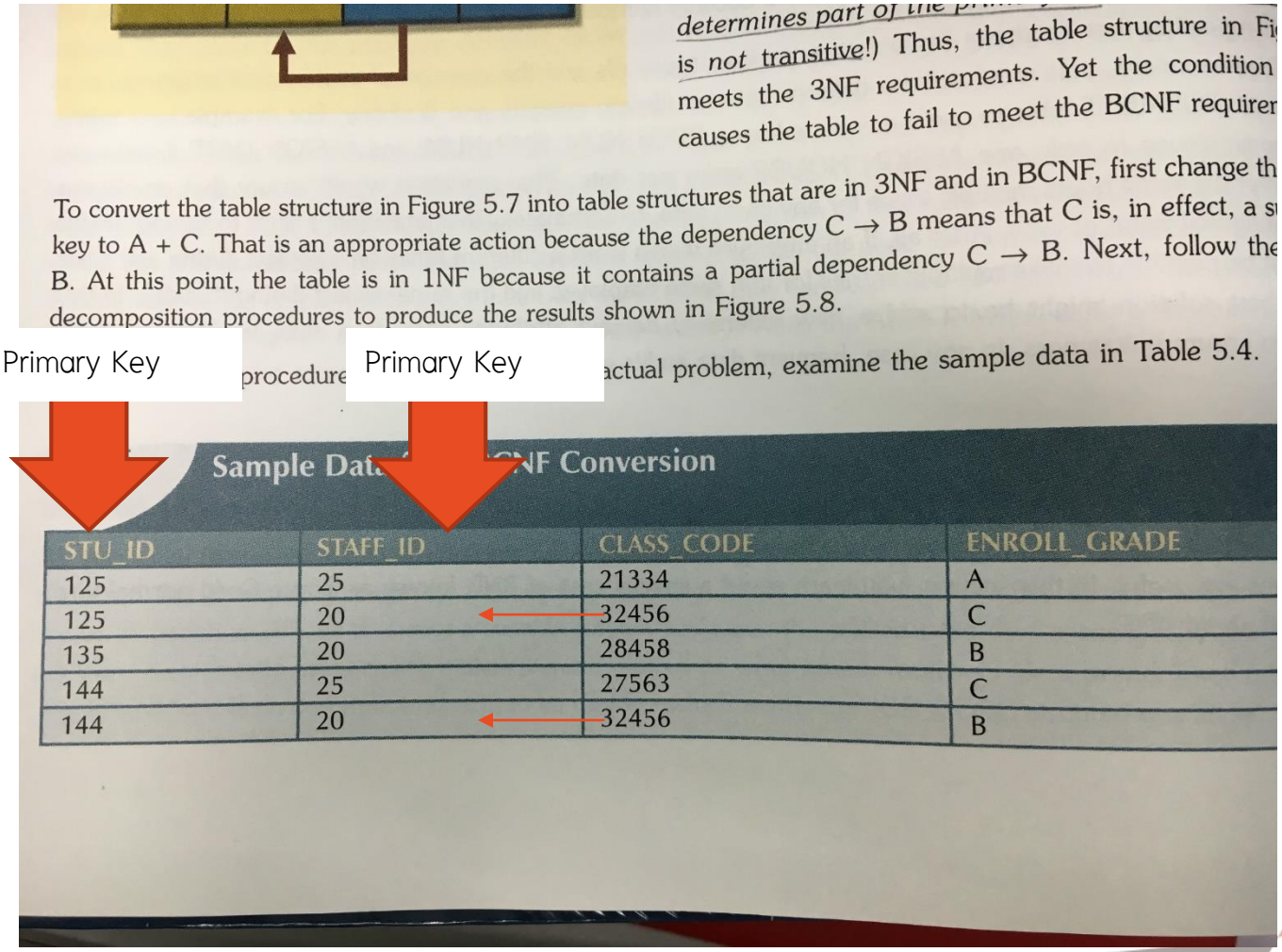
manager	Emp_specialty

Result

manager	Emp_specialty
A	C++
B	JAVA
C	XML
E	XML

Emp_num	manager
0001	A
0001	B
0002	C
0003	B
0003	E

Example



determines part of the p...
is *not* transitive!) Thus, the table structure in Fi
meets the 3NF requirements. Yet the condition
causes the table to fail to meet the BCNF requirer

To convert the table structure in Figure 5.7 into table structures that are in 3NF and in BCNF, first change th
key to A + C. That is an appropriate action because the dependency C → B means that C is, in effect, a s
B. At this point, the table is in 1NF because it contains a partial dependency C → B. Next, follow the
decomposition procedures to produce the results shown in Figure 5.8.

Primary Key

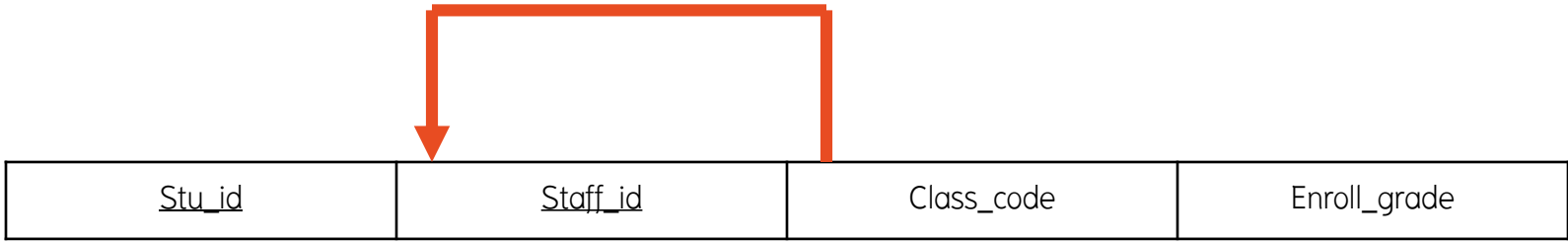
procedure

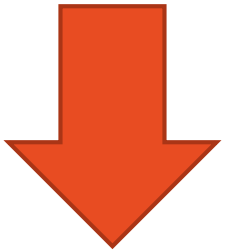
Primary Key

actual problem, examine the sample data in Table 5.4.

Sample Dat... NF Conversion

STU_ID	STAFF_ID	CLASS_CODE	ENROLL_GRADE
125	25	21334	A
125	20	32456	C
135	20	28458	B
144	25	27563	C
144	20	32456	B





Stu_id	Class_code	Enroll_grade

Class_code	Staff_id

Problem #1

stu_id	stu_name	course_id	course_name	grade	Semester
620051001	Somchai Jingjai	961001	Aj Yam Study	F	1/64
				A	2/64
620051002	Somying Naja	961002	Aj Earth Study	C	1/64
620051003	Somsri sawasdee	961003	Aj Maggie Study	D	2/64

Problem #2

course_id	course_name	Sec_num	classroom	capacity	Semester
961001	Aj Yam Study	001	C301	50	1/64
					2/64
961002	Aj Earth Study	002	C302	25	1/64
961003	Aj Maggie Study	003	B303	30	2/64
961004	Aj Don Study	001	C301	50	2/64

Problem #3

manufacturer	model	Sub_model	distributor	Shop_id
Factory AA	Coupe	AA01	Company A	S1
Factory AA	Coupe	AA02	Company C	S1
Factory BB	Coupe	BA01	Company A	S2
Factory BB	Sedan	BA02	Company A	S1
Factory CC	Wagon	C04	Company B	S3
Factory CC	Sedan	C05	Company D	S4

Problem #4

stu_id	stu_firstname	stu_lastname	stu_phone	stu_faculty	advisor_name	advisor_phone	advisor_status
001	Somchai	Somchaimak	1111111111	SE	Earth	053123456	Not active
				DII	Maggie	053123457	Active
002	Somying	Jingjai	222222222	SE	Don	053123458	Active
003	Somjai	Naja	333333333	MMIT	Earth	053123456	Not active
				SE	Maggie	053123457	Active