ITC 5104 Database Design and SQL

LECTURE 11 – NORMALIZATION

Objectives

Why relations should be normalized

The meaning of functional dependence and its relationship to keys

How inference rules for functional dependencies can be used

The definition of first normal form and how to achieve it

The meaning of full functional dependency

The definition of second normal form and how to achieve it

The meaning of transitive dependency

Objectives

The definition of third normal form and how to achieve it

When to stop the normalization process

Objectives of Normalization

Basic objective of logical modeling is to develop a "good" description of the data, its relationships and constraints

For the relational model this means we must identify a suitable set of relations

This is a difficult task since there are many options for the designer to consider

This chapter considers the logical design process generally called normalization

Objectives of Normalization

Normalization

- Process for evaluating and correcting table structures to minimize data redundancies
  - Reduces data anomalies
- Series of stages called normal forms:
  - First normal form (1NF)
  - Second normal form (2NF)
  - Third normal form (3NF)

Objectives of Normalization

Normalization (continued)

- 2NF is better than 1NF; 3NF is better than 2NF
- For most business database design purposes, 3NF is as high as needed in normalization
- Highest level of normalization is not always most desirable

Denormalization produces a lower normal form

- Increased performance but greater data redundancy

Normalization

Example: company that manages building projects

- Charges its clients by billing hours spent on each contract
- Hourly billing rate is dependent on employee's position
- Periodically, report is generated that contains information such as displayed on the next slide

Normalization

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

Normalization

PROJECT NUMBER	PROJECT NAME	EMPLOYEE NUMBER	EMPLOYEE NAME	JOB CLASS	CHARGE/ HOUR	HOURS BILLED	TOTAL CHARGE
15	Evergreen	103	June E. Arbough	Elec. Engineer	$ 85.50	23.8	$ 2,034.90
		101	John G. News	Database Designer	$105.00	19.4	$ 2,037.00
		105	Alice K. Johnson *	Database Designer	$105.00	35.7	$ 3,748.50
		106	William Smithfield	Programmer	$ 35.75	12.6	$ 450.45
		102	David H. Senior	Systems Analyst	$ 96.75	23.8	$ 2,302.65
				Subtotal			**$10,573.50**
18	Amber Wave	114	Annelise Jones	Applications Designer	$ 48.10	25.6	$ 1,183.26
		118	James J. Frommer	General Support	$ 18.36	45.3	$ 831.71
		104	Anne K. Ramoras *	Systems Analyst	$ 96.75	32.4	$ 3,134.70
		112	Darlene M. Smithson	DSS Analyst	$ 45.95	45.0	$ 2,067.75
				Subtotal			**$ 7,265.52**
22	Rolling Tide	105	Alice K. Johnson	Database Designer	$105.00	65.7	$ 6,998.50
		104	Anne K. Ramoras	Systems Analyst	$ 96.75	48.4	$ 4,682.70
		113	Delbert K. Joenbrood	Applications Designer	$ 48.10	23.6	$ 1,135.16
		111	Geoff B. Wabash	Clerical Support	$ 26.87	22.0	$ 591.14
		106	William Smithfield	Programmer	$ 35.75	12.8	$ 457.60
				Subtotal			**$13,765.10**
25	Starflight	107	Maria D. Alonzo	Programmer	$ 35.75	25.6	$ 915.20
		115	Travis B. Bawangi	Systems Analyst	$ 96.75	45.8	$ 4,431.15
		101	John G. News *	Database Designer	$105.00	56.3	$ 5,911.50
		114	Annelise Jones	Applications Designer	$ 48.10	33.1	$ 1,592.11
		108	Ralph B. Washington	Systems Analyst	$ 96.75	23.6	$ 2,283.30
		118	James J. Frommer	General Support	$ 18.36	30.5	$ 559.98
		112	Darlene M. Smithson	DSS Analyst	$ 45.95	41.4	$ 1,902.33
				Subtotal			**$17,595.57**
				Total			**$49,199.69**

Note: * indicates project leader

Normalization Process

Each table represents a single subject

No data item will be unnecessarily stored in more than one table

All nonprime attributes in a table are dependent on the primary key

Each table is void of insertion, update, deletion anomalies

Normalization Process

NORMAL FORM	CHARACTERISTIC	SECTION
First normal form (1NF)	Table format, no repeating groups, and PK identified	6.3.1
Second normal form (2NF)	1NF and no partial dependencies	6.3.2
Third normal form (3NF)	2NF and no transitive dependencies	6.3.3
Boyce-Codd normal form (BCNF)	Every determinant is a candidate key (special case of 3NF)	6.6.1
Fourth normal form (4NF)	3NF and no independent multivalued dependencies	6.6.2

Normalization Process

Objective of normalization is to ensure that all tables are in at least 3NF

Higher forms are not likely to be encountered in business environment

Normalization works one relation at a time

Progressively breaks table into new set of relations based on identified dependencies

Functional Dependence

CONCEPT	DEFINITION
Functional dependence	The attribute *B* is fully functionally dependent on the attribute *A* if each value of *A* determines one and only one value of *B*. Example: PROJ_NUM → PROJ_NAME (read as "PROJ_NUM functionally determines PROJ_NAME") In this case, the attribute PROJ_NUM is known as the "determinant" attribute, and the attribute PROJ_NAME is known as the "dependent" attribute.
Functional dependence (generalized definition)	Attribute *A* determines attribute *B* (that is, *B* is functionally dependent on *A*) if all of the rows in the table that agree in value for attribute *A* also agree in value for attribute *B*.
Fully functional dependence (composite key)	If attribute *B* is functionally dependent on a composite key *A* but not on any subset of that composite key, the attribute *B* is fully functionally dependent on *A*.

Conversion to First Normal Form

Repeating group

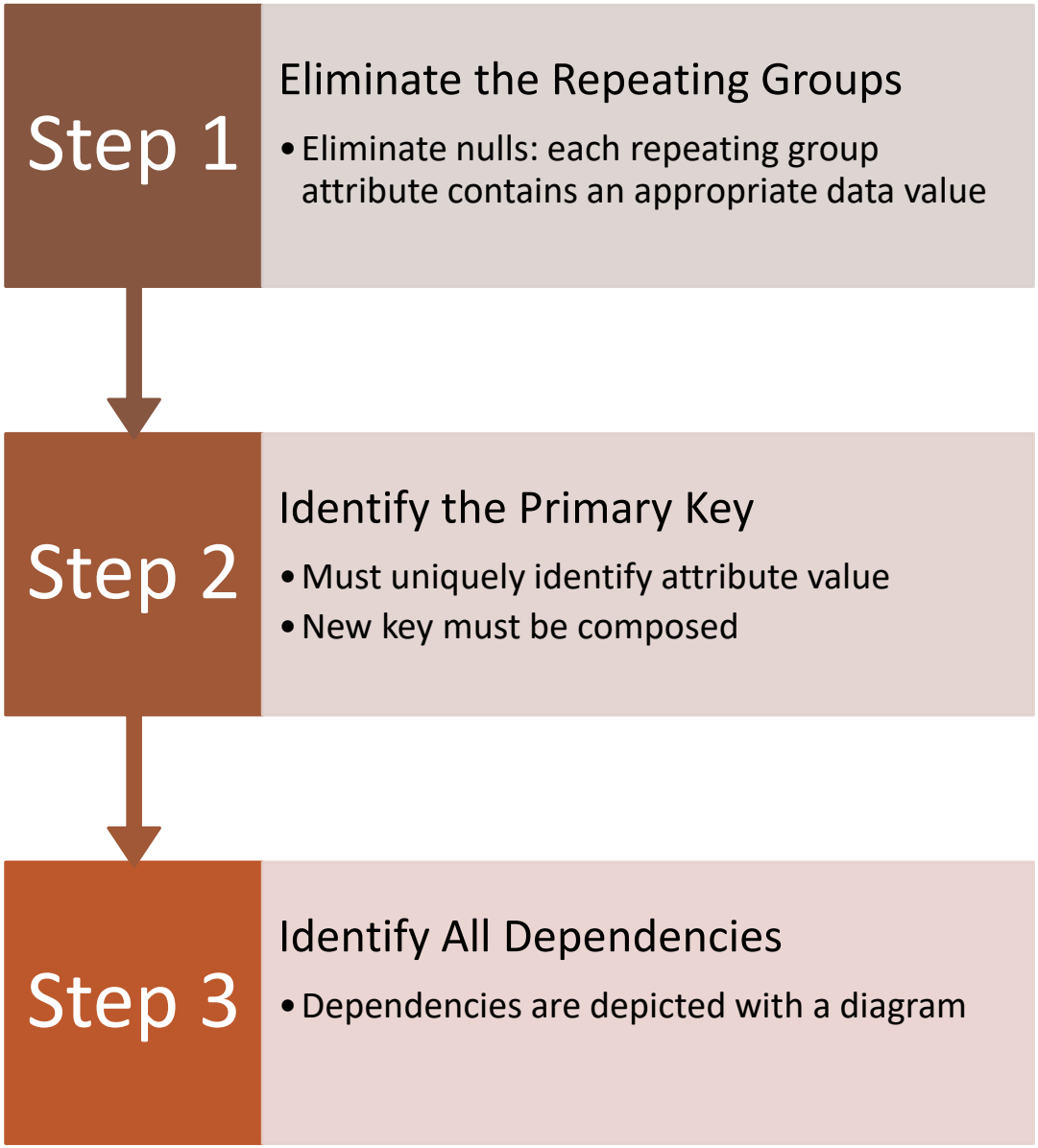
- Group of multiple entries of same type can exist for any single key attribute occurrence

Relational table must not contain repeating groups

Normalizing table structure will reduce data redundancies

Normalization is three-step procedure

Conversion to First Normal Form



Conversion to First Normal Form

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

Conversion to First Normal Form

Looking at the data given on the previous slide we need to find a primary key

What is a primary key?

This we have discussed previously

We need to also define the dependencies in the diagram on the previous slide

What depends on the primary key

Also are there other values that depend on only a portion of the primary key

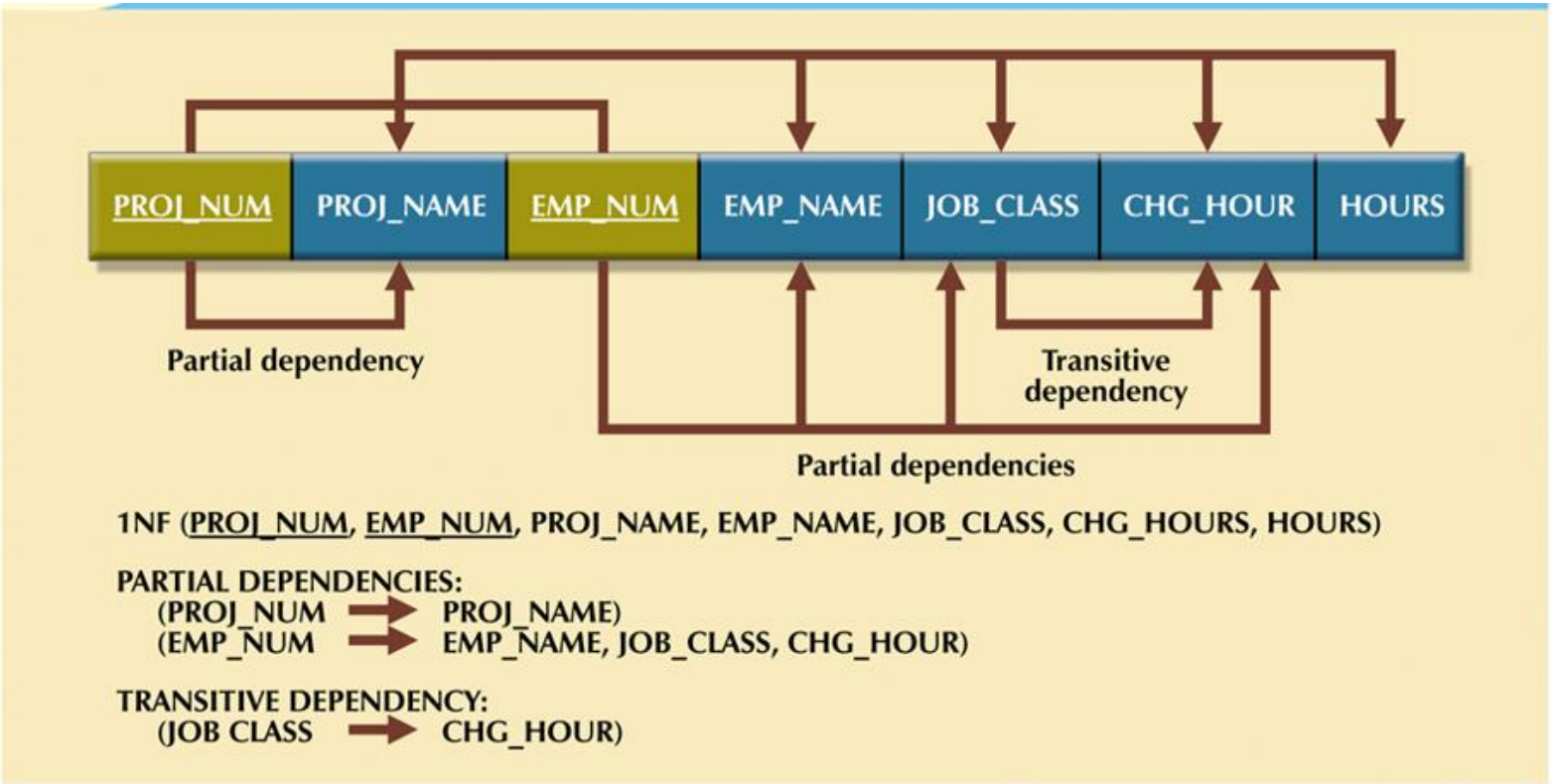
Create a Dependency Diagramto show these dependent relationships

Dependency Diagram

Dependency diagram:

- Depicts all dependencies found within given table structure
- Helpful in getting bird's-eye view of all relationships among table's attributes
- Makes it less likely that you will overlook an important dependency

Dependency Diagram



Conversion to First Normal Form

First normal form describes tabular format:

- All key attributes are defined
- No repeating groups in the table
- All attributes are dependent on primary key

All relational tables satisfy 1NF requirements

Some tables contain partial dependencies

- Dependencies are based on part of the primary key
- Should be used with caution

Conversion to Second Normal Form

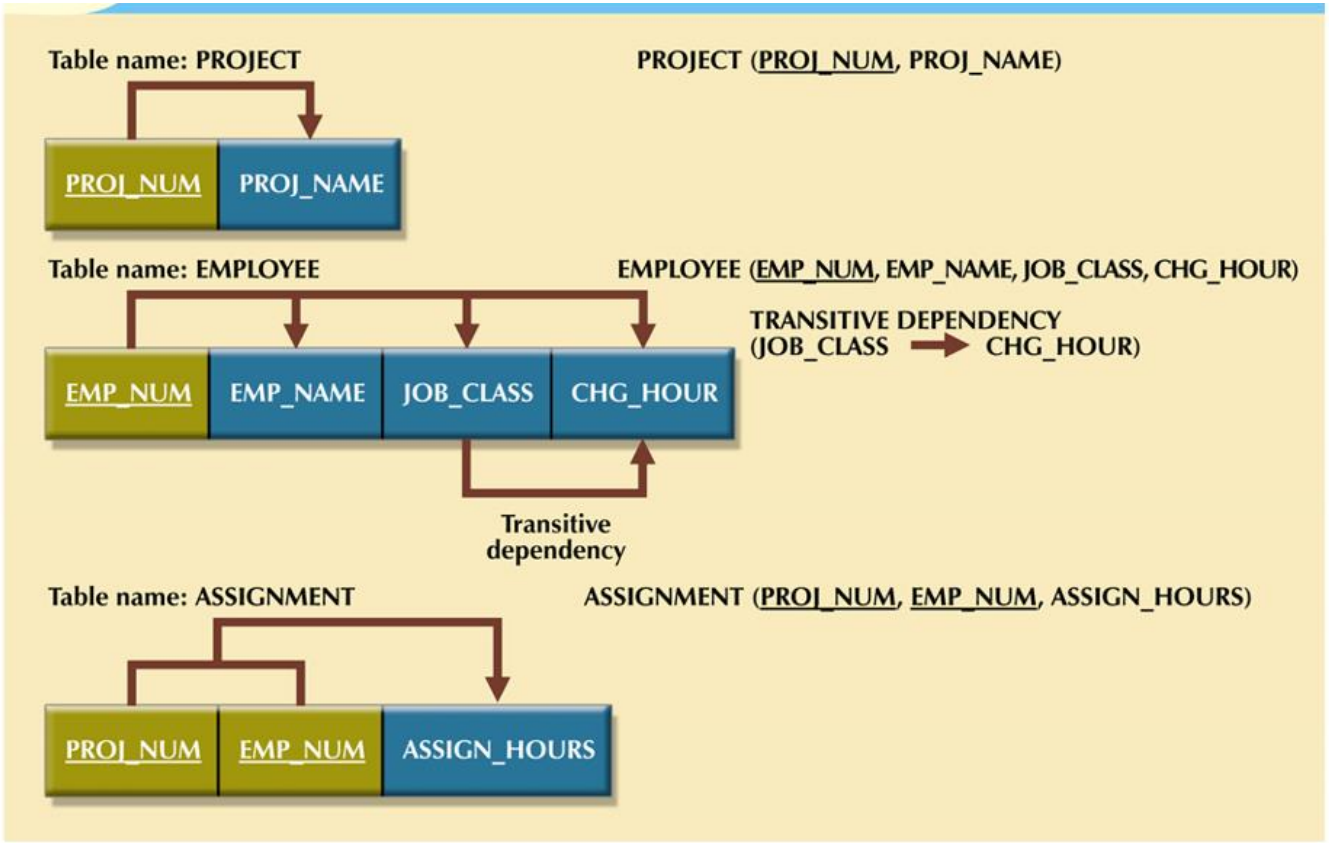
Step 1: Make New Tables to Eliminate Partial Dependencies

- Write each key component on separate line, then write original (composite) key on last line
- Each component will become key in new table

Step 2: Assign Corresponding Dependent Attributes

- Determine attributes that are dependent on other attributes
- At this point, most anomalies have been eliminated

Conversion to Second Normal Form



Conversion to Second Normal Form

Table is in second normal form (2NF) when:

- It is in 1NF *and*
- It includes no partial dependencies:
  - No attribute is dependent on only portion of primary key

Conversion to Third Normal Form

Step 1: Make New Tables to Eliminate Transitive Dependencies

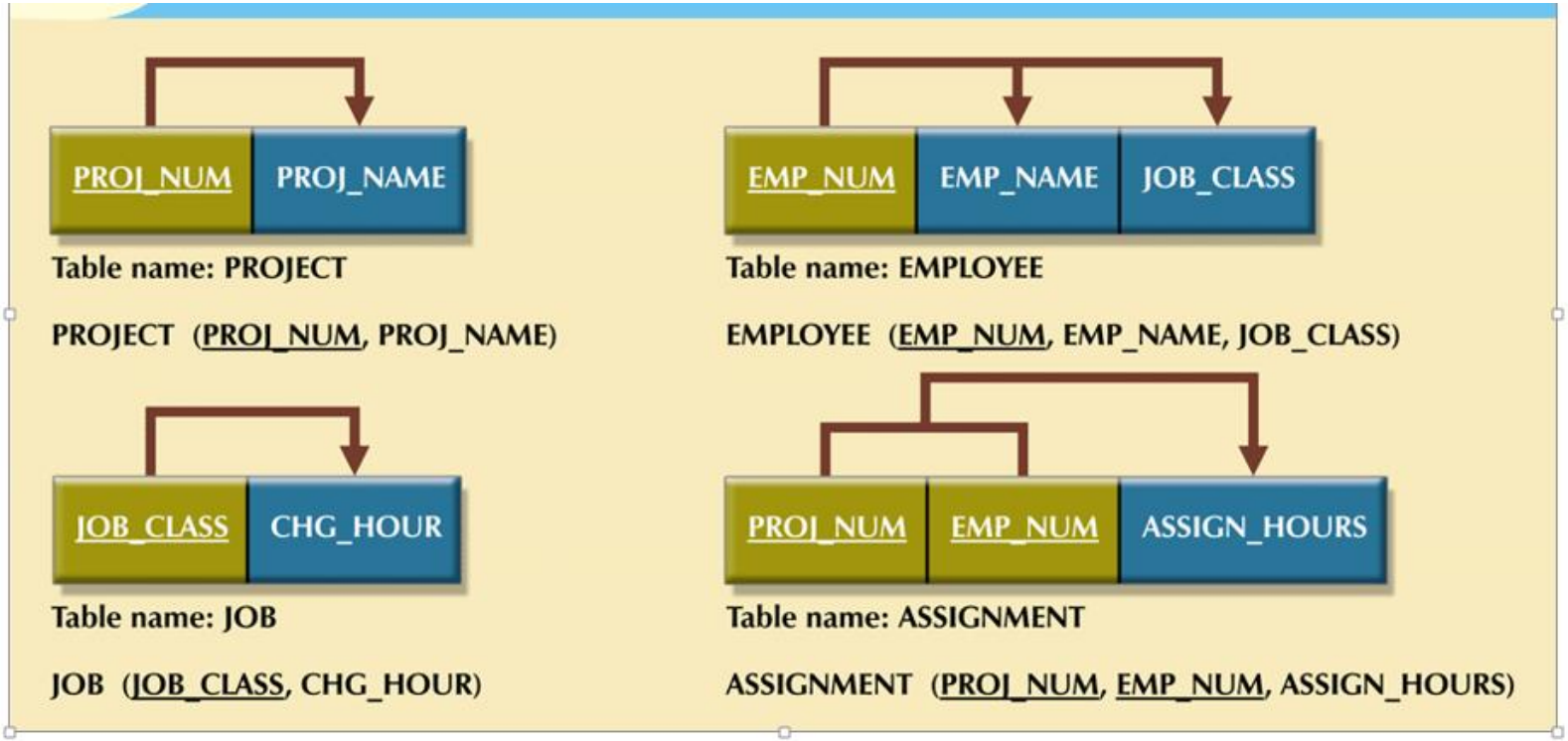
- For every transitive dependency, write its determinant as PK for new table
- Determinant: any attribute whose value determines other values within a row

Conversion to Third Normal Form

Step 2: Reassign Corresponding Dependent Attributes

- Identify attributes dependent on each determinant identified in Step 1
  - Identify dependency
- Name table to reflect its contents and function

Conversion to Third Normal Form



Conversion to Third Normal Form

A table is in third normal form (3NF) when both of the following are true:

- It is in 2NF
- It contains no transitive dependencies

Improve the Design

Table structures should be cleaned up to eliminate initial partial and transitive dependencies

Normalization cannot, by itself, be relied on to make good designs

Valuable because it helps eliminate data redundancies

Issues to address, in order, to produce a good normalized set of tables:

- Evaluate PK Assignments
- Evaluate Naming Conventions
- Refine Attribute Atomicity
- Identify New Attributes

Complete Database Employee Table

Table name: EMPLOYEE

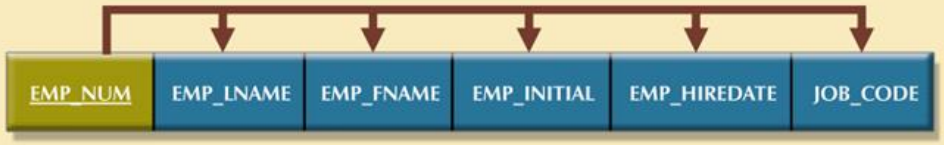


Table name: EMPLOYEE

EMP_NUM	EMP_LNAME	EMP_FNAME	EMP_INITIAL	EMP_HIREDATE	JOB_CODE
101	News	John	G	08-Nov-00	502
102	Senior	David	H	12-Jul-89	501
103	Arbough	June	E	01-Dec-97	503
104	Ramoras	Anne	K	15-Nov-88	501
105	Johnson	Alice	K	01-Feb-94	502
106	Smithfield	William		22-Jun-05	500
107	Alonzo	Maria	D	10-Oct-94	500
108	Washington	Ralph	B	22-Aug-89	501
109	Smith	Larry	W	18-Jul-99	501
110	Olenko	Gerald	A	11-Dec-96	505
111	Wabash	Geoff	B	04-Apr-89	506
112	Smithson	Darlene	M	23-Oct-95	507
113	Joenbrood	Delbert	K	15-Nov-94	508
114	Jones	Annelise		20-Aug-91	508
115	Bawangi	Travis	B	25-Jan-90	501
116	Pratt	Gerald	L	05-Mar-95	510
117	Williamson	Angie	H	19-Jun-94	509
118	Frommer	James	J	04-Jan-06	510

Surrogate Key Consideration

When primary key is considered to be unsuitable, designers use surrogate keys

Data entries in Table 6.4 are inappropriate because they duplicate existing records

- No violation of entity or referential integrity

Normalization and Database Design

Normalization should be part of the design process

Make sure that proposed entities meet required normal form before table structures are created

Many real-world databases have been improperly designed or burdened with anomalies

You may be asked to redesign and modify existing databases

Normalization and Database Design

ER diagram

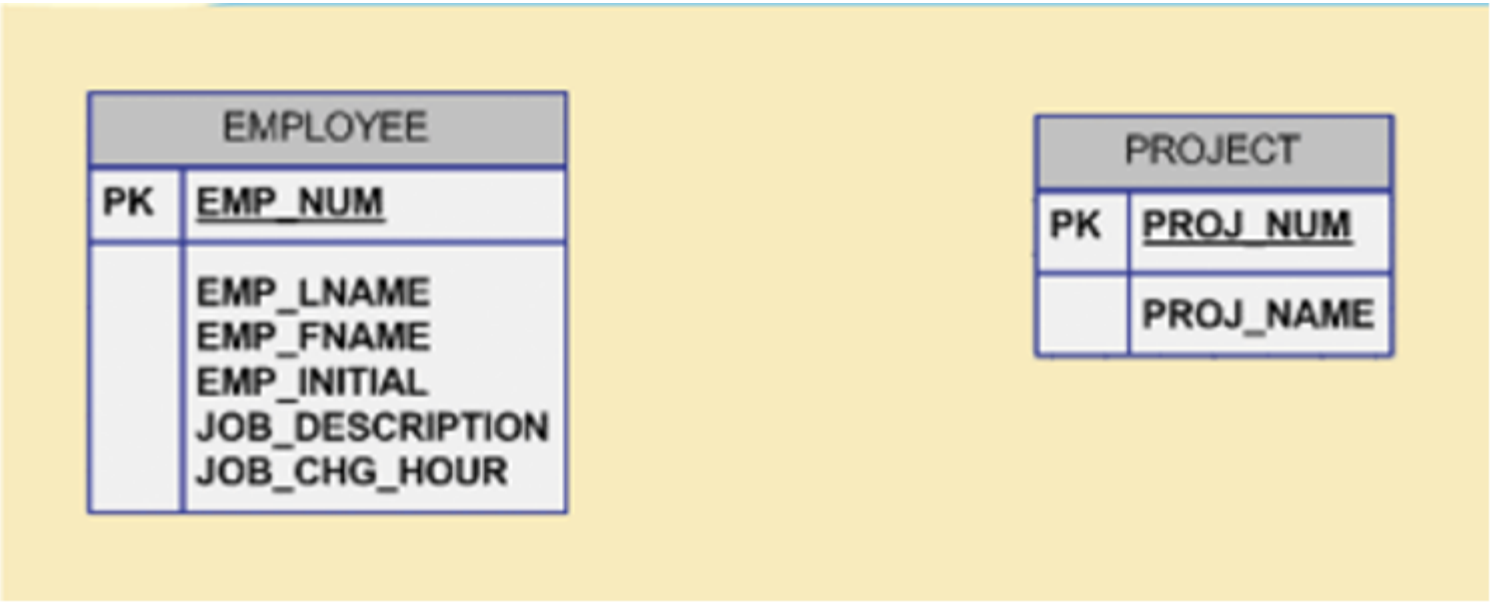
- Identify relevant entities, their attributes, and their relationships
- Identify additional entities and attributes

Normalization procedures

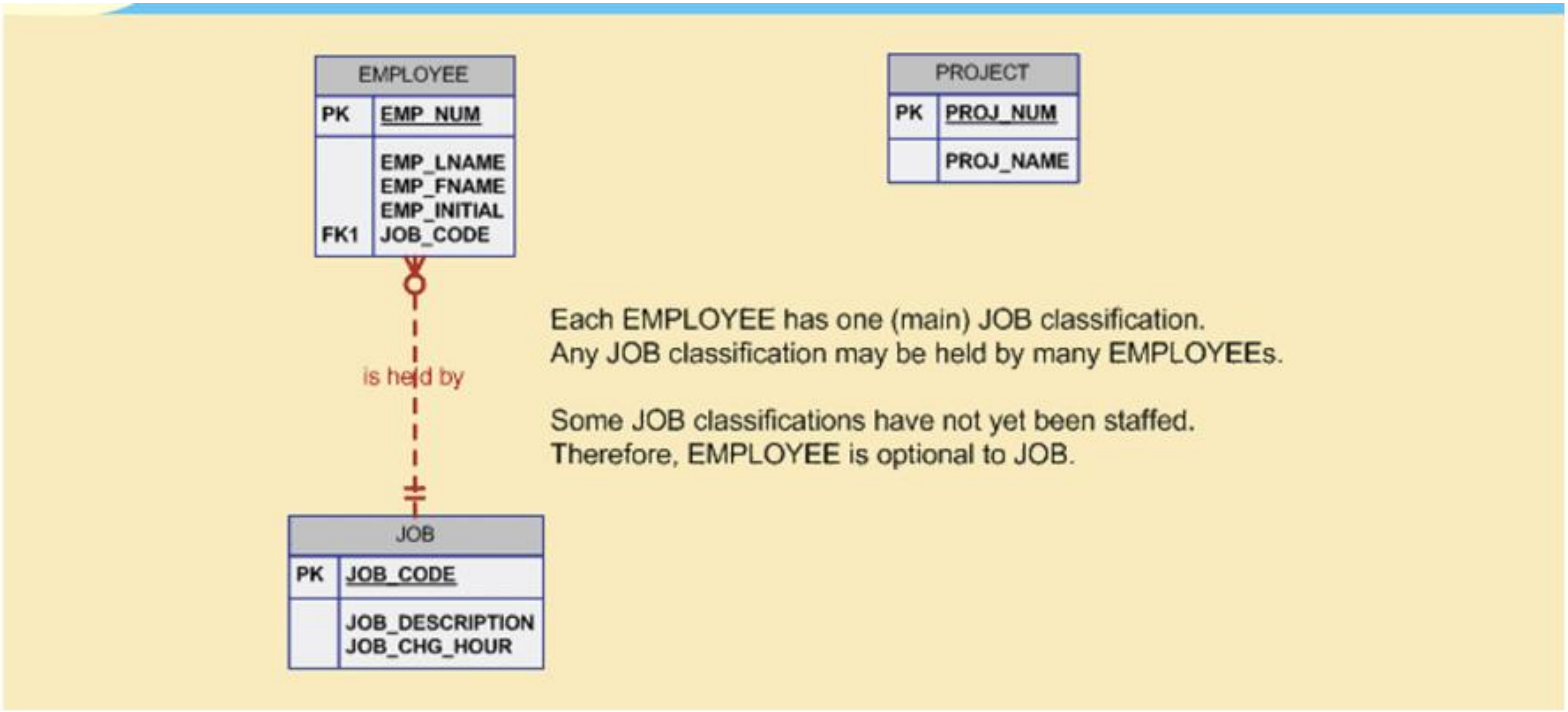
- Focus on characteristics of specific entities
- Micro view of entities within ER diagram

Difficult to separate normalization process from ER modeling process

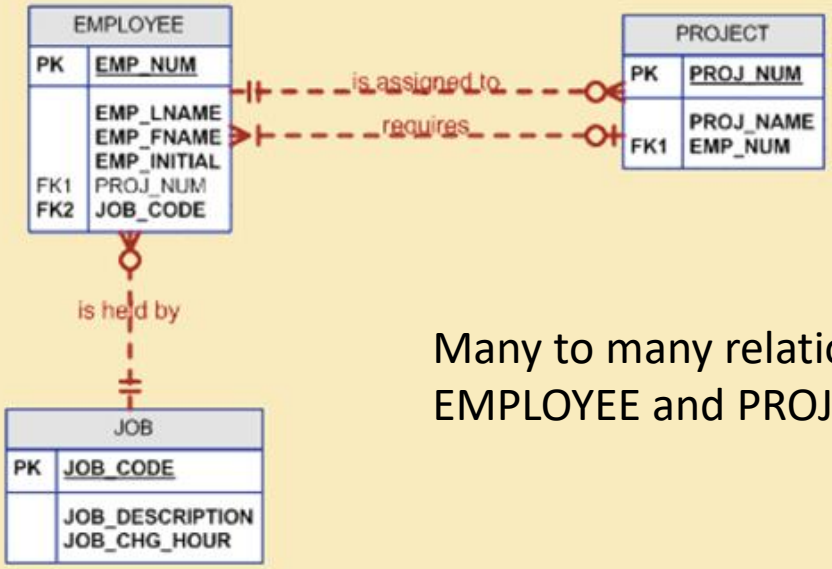
Normalization and Database Design



Normalization and Database Design

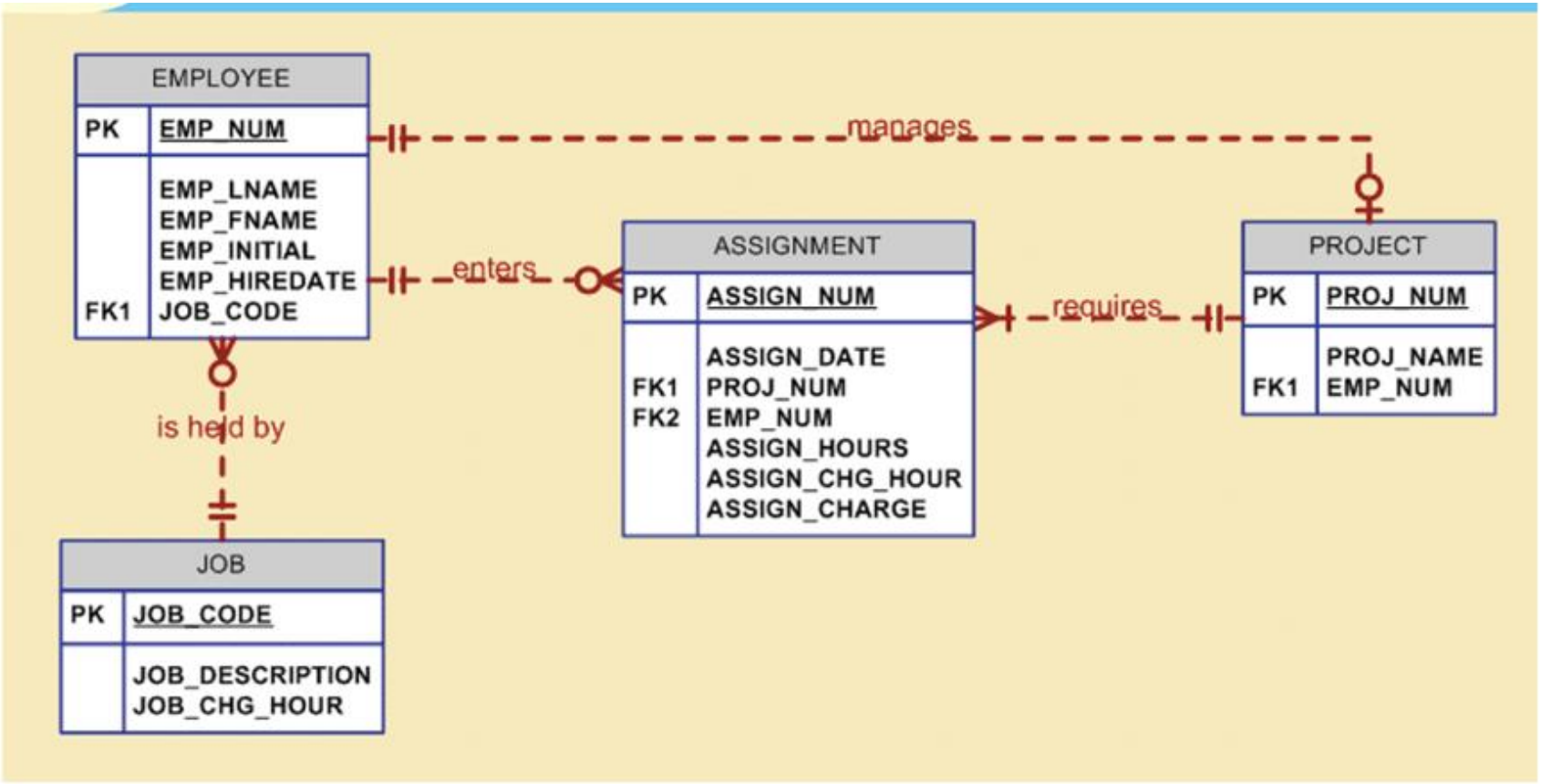


Normalization and Database Design



Many to many relation between EMPLOYEE and PROJECT

Normalization and Database Design



Completed Database

Table name: EMPLOYEE

Database name: Ch06_ConstructCo

EMP_NUM	EMP_LNAME	EMP_FNAME	EMP_INITIAL	EMP_HIREDATE	JOB_CODE
101	News	John	G	08-Nov-00	502
102	Senior	David	H	12-Jul-89	501
103	Arbough	June	E	01-Dec-97	503
104	Ramoras	Anne	K	15-Nov-88	501
105	Johnson	Alice	K	01-Feb-94	502
106	Smithfield	William		22-Jun-05	500
107	Alonzo	Maria	D	10-Oct-94	500
108	Washington	Ralph	B	22-Aug-89	501
109	Smith	Larry	W	18-Jul-99	501
110	Olenko	Gerald	A	11-Dec-96	505
111	Wabash	Geoff	B	04-Apr-89	506
112	Smithson	Darlene	M	23-Oct-95	507
113	Joenbrood	Delbert	K	15-Nov-94	508
114	Jones	Annelise		20-Aug-91	508
115	Bawangi	Travis	B	25-Jan-90	501
116	Pratt	Gerald	L	05-Mar-95	510
117	Williamson	Angie	H	19-Jun-94	509
118	Frommer	James	J	04-Jan-06	510

Table name: 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

Table name: PROJECT

PROJ_NUM	PROJ_NAME	EMP_NUM
15	Evergreen	105
18	Amber Wave	104
22	Rolling Tide	113
25	Starflight	101

Table name: ASSIGNMENT

ASSIGN_NUM	ASSIGN_DATE	PROJ_NUM	EMP_NUM	ASSIGN_HOURS	ASSIGN_CHG_HOUR	ASSIGN_CHARGE
1001	04-Mar-10	15	103	2.6	84.50	219.70
1002	04-Mar-10	18	118	1.4	18.36	25.70
1003	05-Mar-10	15	101	3.6	105.00	378.00
1004	05-Mar-10	22	113	2.5	48.10	120.25
1005	05-Mar-10	15	103	1.9	84.50	160.55
1006	05-Mar-10	25	115	4.2	96.75	406.35
1007	05-Mar-10	22	105	5.2	105.00	546.00
1008	05-Mar-10	25	101	1.7	105.00	178.50
1009	05-Mar-10	15	105	2.0	105.00	210.00
1010	06-Mar-10	15	102	3.8	96.75	367.65
1011	06-Mar-10	22	104	2.6	96.75	251.55
1012	06-Mar-10	15	101	2.3	105.00	241.50
1013	06-Mar-10	25	114	1.8	48.10	86.58
1014	06-Mar-10	22	111	4.0	26.87	107.48
1015	06-Mar-10	25	114	3.4	48.10	163.54
1016	06-Mar-10	18	112	1.2	45.95	55.14
1017	06-Mar-10	18	118	2.0	18.36	36.72
1018	06-Mar-10	18	104	2.6	96.75	251.55
1019	06-Mar-10	15	103	3.0	84.50	253.50
1020	07-Mar-10	22	105	2.7	105.00	283.50
1021	08-Mar-10	25	108	4.2	96.75	406.35
1022	07-Mar-10	25	114	5.8	48.10	278.98
1023	07-Mar-10	22	106	2.4	35.75	85.80

Database design

Another example

First Normal Form

An alternate method of making the original table 1NF is to make the multivalued attribute part of the key

Our new table now would contain multiple rows for students with multiple majors

The text states that this method can cause problems or difficulties when the designer attempts to put the relation into higher normal forms, I disagree with this statement and I will show you how

We will actually end up with the same structure shown as the first solution for the 1NF on slides 32 and 33

First Normal Form

stuID	lastName	major	credits	status	socSecNo
S1001	Smith	History	90	Senior	100429500
S1003	Jones	Math	95	Senior	010124567
S1006	Lee	CSC	15	Freshman	088520876
S1006	Lee	Math	15	Freshman	088520876
S1010	Burns	Art	63	Junior	099320985
S1010	Burns	English	63	Junior	099320985
S1060	Jones	CSC	25	Freshman	064624738

First Normal Form

Our original primary key of stuId will no longer work since we now have a repeating value

A new primary key consisting of {stuId, major} is used to give the unique identifier to each row

This composite primary key can be taken out to leave the stuID and its dependent attributes left

This will form two tables as was shown previously

Slide 40 will show the stuId and its dependent attributes while slide 41 will show the composite primary key of stuId and major

First Normal Form

stuID	lastName	credits	status	socSecNo
S1001	Smith	90	Senior	100429500
S1003	Jones	95	Senior	010124567
S1006	Lee	15	Freshman	088520876
S1010	Burns	63	Junior	099320985
S1060	Jones	25	Freshman	064624738

First Normal Form

stuId	major
S1001	History
S1003	Math
S1006	CSC
S1006	Math
S1010	Art
S1010	English
S1060	CSC

First Normal Form

Our first example just did this step for us but as you can see this can be arrived at by breaking down the third example

The second example is the one that causes the largest problem and should be avoided altogether

Examples 1 and 3 are the solutions you will see through out most books dealing with relational database design and normalization just two different approaches

Full Functional Dependence and Second Normal Form

For the relations shown on the next slide we have the following functional dependencies in addition to the trivial ones:

- {courseNo, stuId} → {lastName}
- {courseNo, stuId} → {facId}
- {courseNo, stuId} → {schedule}
- {courseNo, stuId} → {room}
- {courseNo, stuId} → {grade}

Full Functional Dependence and Second Normal Form

courseNo	stuId	stuLastName	facId	schedule	room	grade
ART103A	S1001	Smith	F101	MWF9	H221	A
ART103A	S1010	Burns	F101	MWF9	H221	
ART103A	S1006	Lee	F101	MWF9	H221	B
CSC201Z	S1003	Jones	F105	TuThF10	M110	A
CSC201Z	S1006	Lee	F105	TuThF10	M110	C
HST205A	S1001	Smith	F102	MWF11	H221	

Full Functional Dependence and Second Normal Form

Since there is no other candidate key we chose {courseNo, stuID} for the primary key

Ignoring trivial dependencies we also have the following functional dependencies:

- courseNo → facId
- courseNo → schedule
- courseNo → room
- stuId → lastName

These dependencies are dependent on only a portion of the composite primary key, these can be called partial dependencies since you do not need to know the entire primary key to determine each of these dependent values

Full Functional Dependence and Second Normal Form

We find attributes that are functionally dependent on the combination {courseNo, stuId}

There are also attributes that are only dependent on a subset of that combination

We say that these attributes are not fully functionally dependent on the combination

Definition:

- In a relation R, attribute A of R is **fully functionally dependent** on an attribute or a set of attributes X or R if A is functionally dependent on X but not functionally dependent on any proper subset of X.

Full Functional Dependence and Second Normal Form

lastName is functionally dependent on {courseNo, stuId} but is also functionally dependent on a proper subset of that combination, stuId

Similarly facId, schedule, and room are functionally dependent on the proper subset courseNo

Grade on the other hand is fully functionally dependent on the combination {courseNo, stuId}

Definition:

- A relation is in **second normal form** (2NF) if and only if it is in first normal form (1NF) and all the non-key attributes are fully functionally dependent on the key

These dependencies are also referred to as **partial dependencies** since the attributes are only dependent on part of the initial primary key, partial dependencies are removed in 2NF

Full Functional Dependence and Second Normal Form

If a relation is in 1NF and the key only consists of a single attribute the relation is automatically 2NF

You are only concerned about 2NF when there is a composite key

The Class relation is not in 2NF since lastName is not fully functionally dependent on the key {courseNo, stuId}

There of course are others but all you need is one to show the relation is not in 2NF

The next thing to do is to transfer the 1NF relation into an equivalent set of 2NF relations

Full Functional Dependence and Second Normal Form

The original composite key is {courseNo, stuId}, each of these is to be placed into a separate relation

- courseNo
- stuId

The original composite key must be maintained

- courseNo, stuId

There are now going to potentially be 3 relations

Full Functional Dependence and Second Normal Form

After the relations have been defined then the dependent attributes are placed beside them

- courseNo → facId, schedule, room
- stuId → lastName
- courseNo, stuId → grade (and of course the other attributes still actually apply but not shown any more

Using projection NewClass relation is broken onto the following relations:

- Register (courseNo, stuID, grade)
- Class2 (courseNo, facId, schedule, room)
- Stu (stuId, stuLastName)

The resulting relations are shown on the following slides

Full Functional Dependence and Second Normal Form

courseNo	stuId	stuLastName	facId	schedule	room	grade
ART103A	S1001	Smith	F101	MWF9	H221	A
ART103A	S1010	Burns	F101	MWF9	H221	
ART103A	S1006	Lee	F101	MWF9	H221	B
CSC201Z	S1003	Jones	F105	TuThF10	M110	A
CSC201Z	S1006	Lee	F105	TuThF10	M110	C
HST205A	S1001	Smith	F102	MWF11	H221	

NewClass Table not in 2NF

Full Functional Dependence and Second Normal Form

classNo	stuId	grade
ART103A	S1001	A
ART103A	S1010	
ART103A	S1006	B
CSC201A	S1003	A
CSC201A	S1006	C

Register Table in 2NF

Full Functional Dependence and Second Normal Form

stuId	stuLastName
S1001	Smith
S1010	Burns
S1006	Lee
S1003	Jones

Stu Table in 2NF

Full Functional Dependence and Second Normal Form

classNo	facId	schedule	room
ART103A	F101	MWF9	H221
CSC210A	F105	TUTHF10	M110
HST205A	F202	MWF11	H221

Class2 Table in 2NF

Full Functional Dependence and Second Normal Form

Redundant data has been reduced

A student can be added by adding the details only to the Stu table

Course information only has to be added once now to the Class2 table

If we drop a registration record in the Register table it will not affect the student or course tables at all of those details are maintained

Transitive Dependency and Third Normal Form

Second normal form is better than first normal form

We will consider the following relation:
- NewStudent (stuId, lastName, major, credits, status)

The next slide will show an instance of this relation

There is only one candidate key stuId and iot will be used as the primary key

All other attributes are dependant on this key

We have the following functional dependency
- stuId → {lastName, major, credits, status}

We discussed that the number of credits taken by a student determines their status, given this shows this
- credits → status

The stuId functionally determines the status in two ways directly and transitively

Transitive Dependency and Third Normal Form

Since (stuId → credits) and (credits → status) therefore (stuId → status)

Definition:

- If A, B, and C are attributes of relation R, such that A → B and B → C, then C is **transitively dependent** on A

In the third normal form we want to eliminate transitive dependencies

Transitive dependencies will cause insertion, update and deletion anomalies

Transitive Dependency and Third Normal Form

stuID	lastName	credits	status	Major
S1001	Smith	90	Senior	History
S1003	Jones	95	Senior	Math
S1006	Lee	15	Freshman	CSC
S1010	Burns	63	Junior	Art
S1060	Jones	25	Freshman	CSC

NewClass Table not in 3NF

Transitive Dependency and Third Normal Form

stuID	lastName	credits	status	socSecNo
S1001	Smith	90	Senior	100429500
S1003	Jones	95	Senior	010124567
S1006	Lee	15	Freshman	088520876
S1010	Burns	63	Junior	099320985
S1060	Jones	25	Freshman	064624738

Again the NewClass Table not in 3NF – credits → status, transitive dependency

Transitive Dependency and Third Normal Form

credits	status
15	Freshman
25	Freshman
63	Junior
90	Senior
95	Senior

Copy credits from the original table and remove the status form the original

Transitive Dependency and Third Normal Form

stuId	lastName	major	credits
S1001	Smith	History	90
S1003	Jones	Math	95
S1006	Lee	CSC	15
S1010	Burns	Art	63
S1060	Jones	CSC	25

The status is removed and credits is left behind