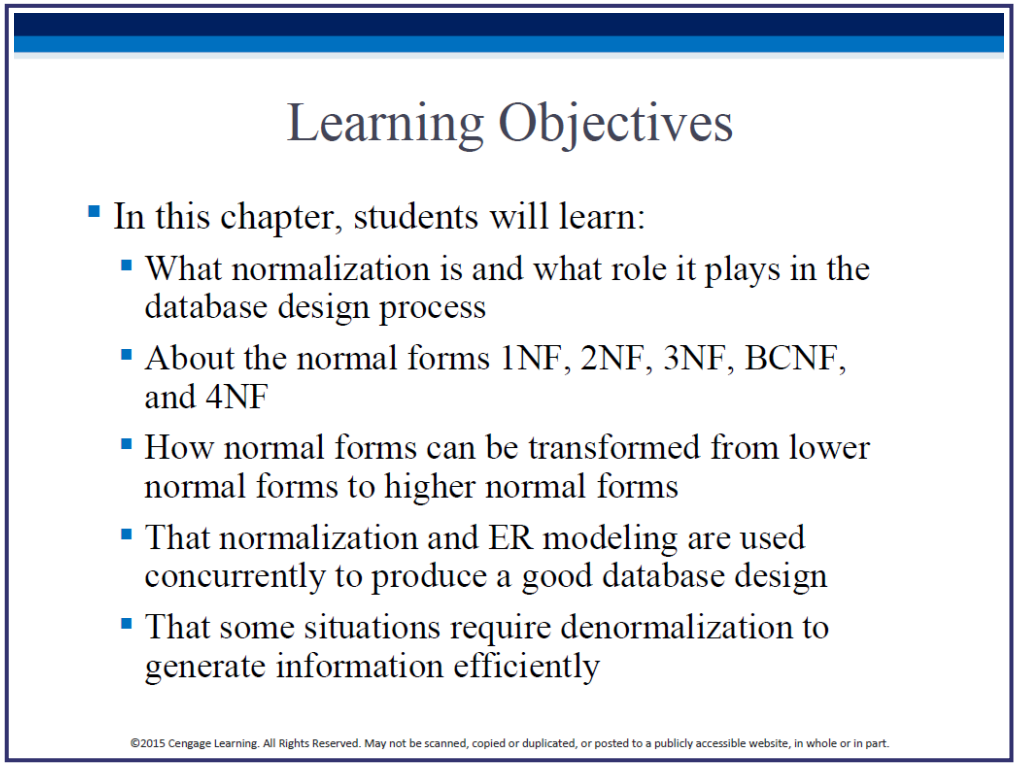
30/30 11:46:45 ***

← →

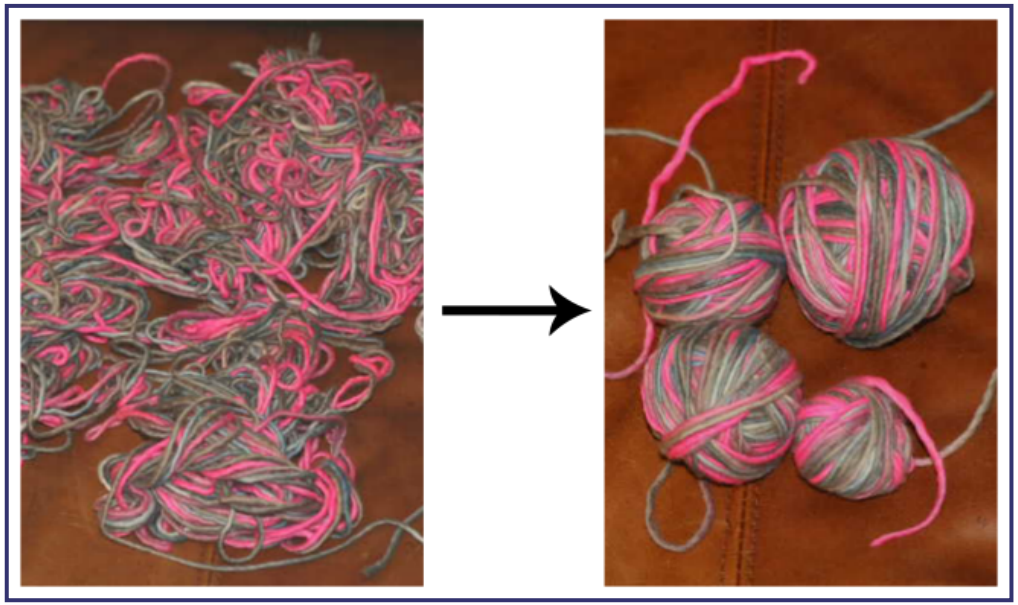
(De/)Normalization

On tap


Learning Objectives
In this chapter, students will learn:
What normalization is and what role it plays in the database design process
About the normal forms 1NF, 2NF, 3NF, BCNF, and 4NF
How normal forms can be transformed from lower normal forms to higher normal forms
That normalization and ER modeling are used concurrently to produce a good database design
That some situations require denormalization to generate information efficiently
©2015 Cengage Learning. All Rights Reserved. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.

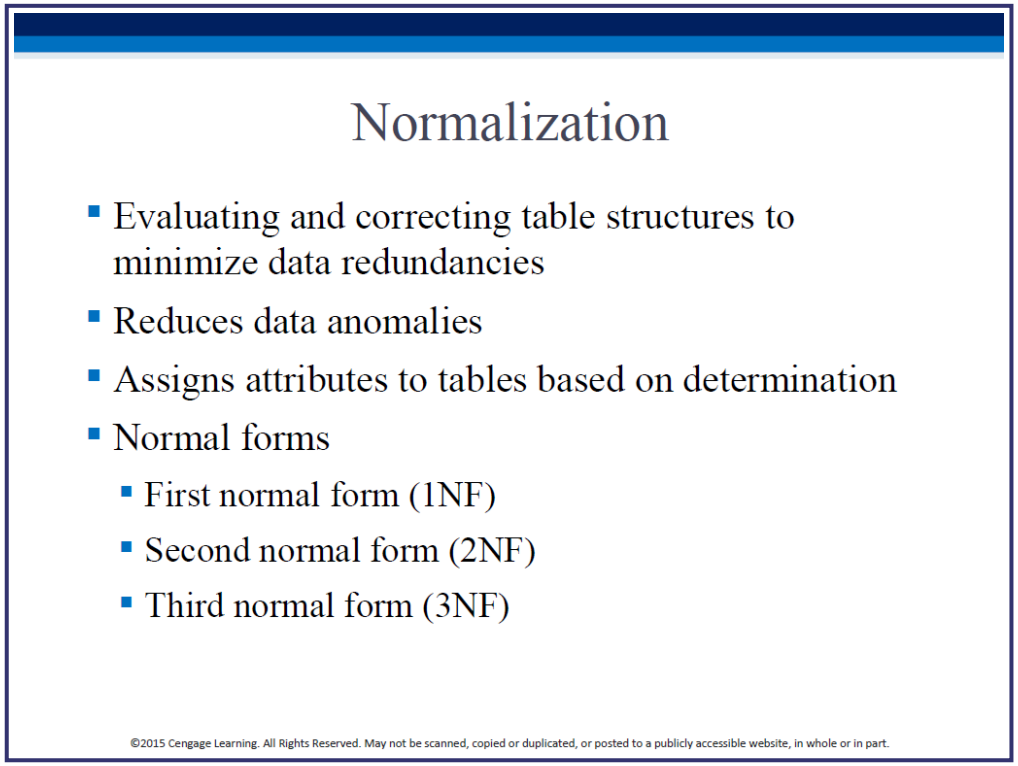
The goal of normalization

Loosely speaking:

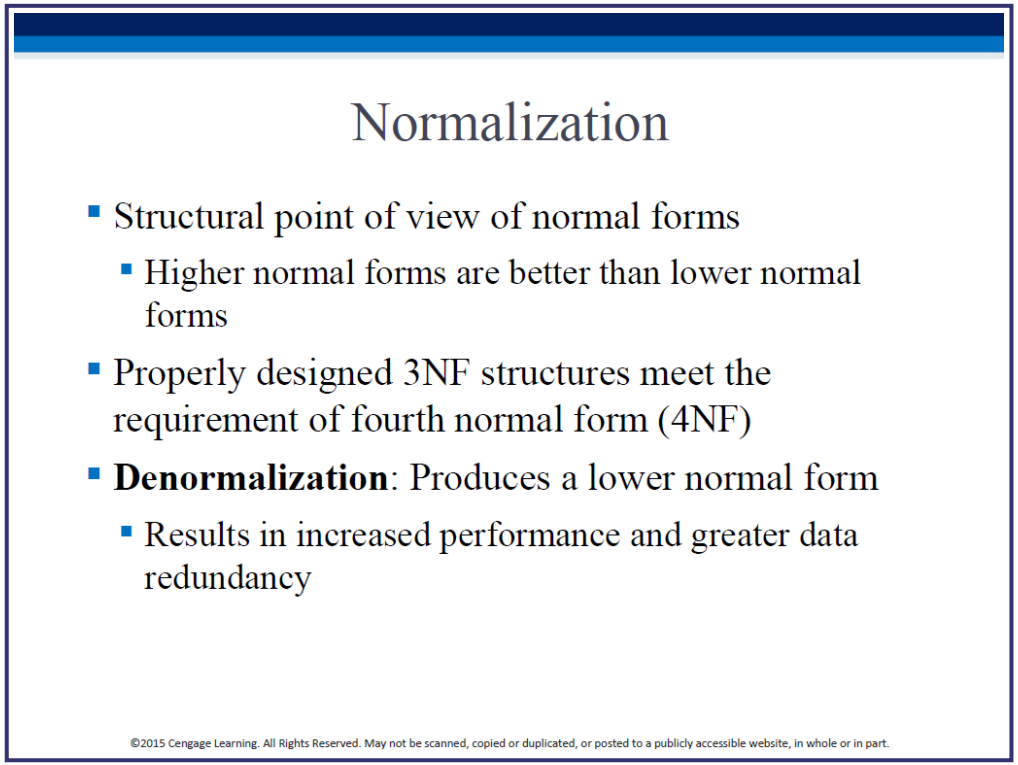


[http://rovingcrafters.com/]

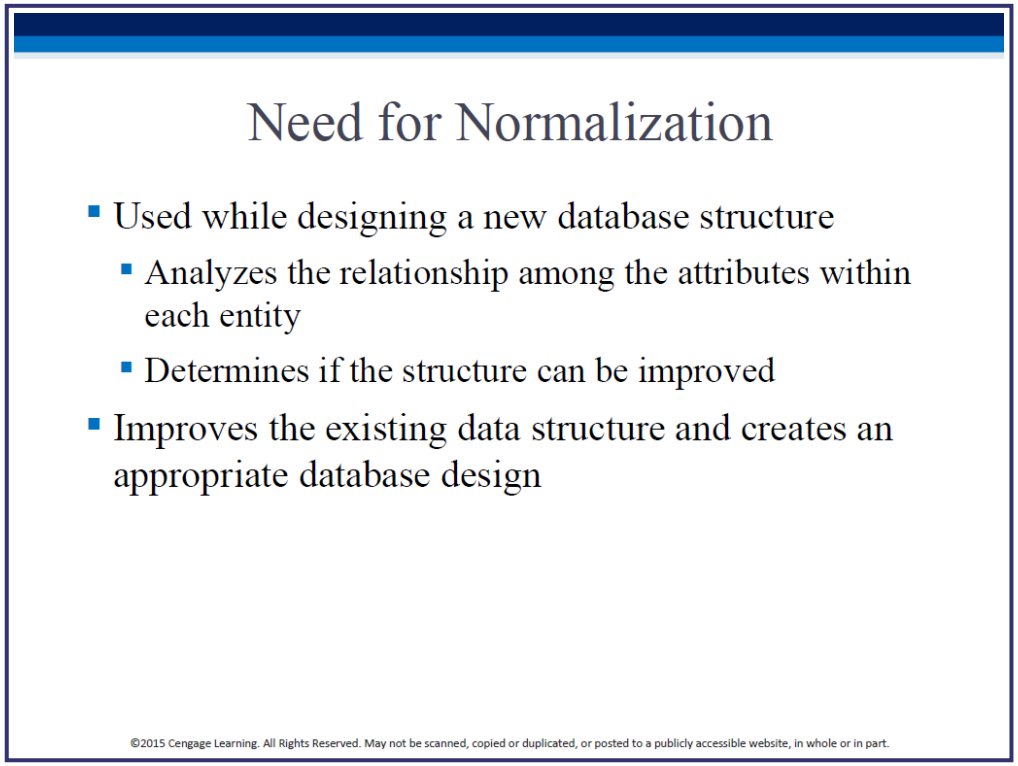
Goal: reduce redundancies, anomalies



Higher normal forms → cleaner designs



Normalization is a design step



A construction company db

Employees of the construction company work on projects. Each employee has an ID, name, job title and corresponding hourly rate.

Each project has a number, name and assigned employees. An employee can be assigned to more than one project.

The company bills clients for projects, based on hours worked by employees.

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

Report

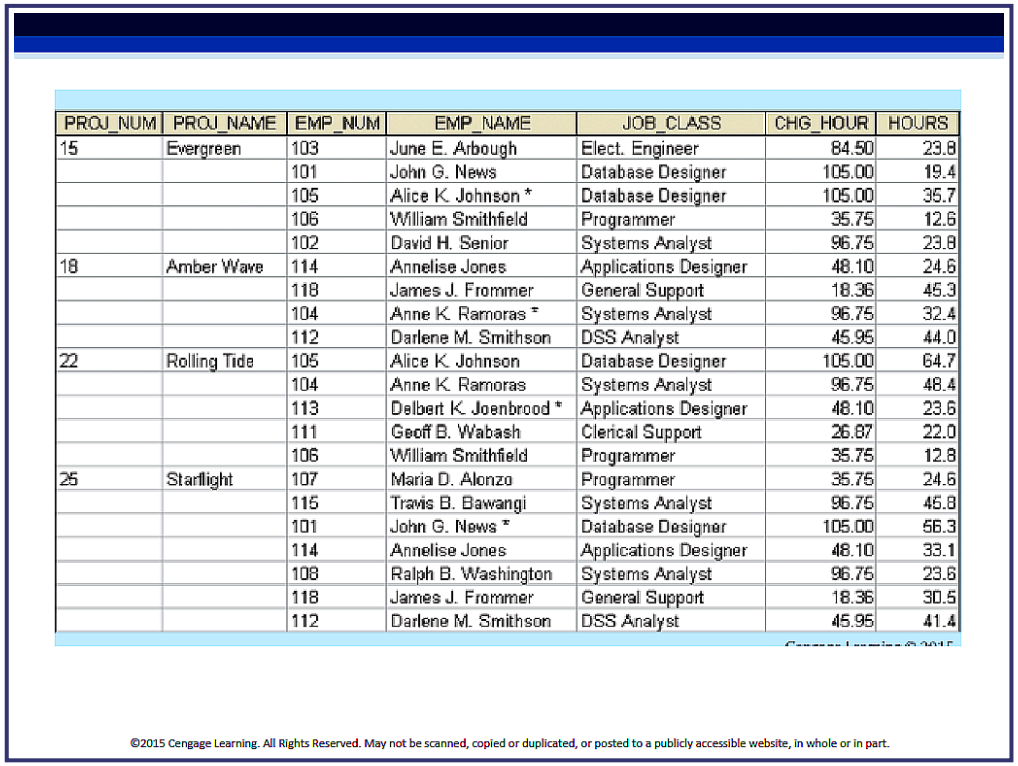
The construction company periodically generates a report like so:

PROJECT NUMBER	PROJECT NAME	EMPLOYEE NUMBER	EMPLOYEE NAME	JOB CLASS	CHARGE/ HOUR	HOURS BILLED	TOTAL CHARGE
15	Evergreen	103	June E. Arbough	Elec. Engineer	$ 84.50	23.8	$ 2,011.10
		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			**$10,549.70**
18	Amber Wave	114	Annelise Jones	Applications Designer	$ 48.10	24.6	$ 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	44.0	$ 2,021.80
				Subtotal			**$ 7,171.47**
22	Rolling Tide	105	Alice K. Johnson	Database Designer	$105.00	64.7	$ 6,793.50
		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			**$13,660.10**
25	Starflight	107	Maria D. Alonzo	Programmer	$ 35.75	24.6	$ 879.45
		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			**$17,559.82**
				Total			**$48,941.09**

Note: A * indicates the project leader.

Issues with our db

Here is our table again:



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

There are numerous issues:

- the PROJ_NUM attr could be used as a PK (or part of a PK, along with PROJ_NAME) but it contains nulls

- possibilities for data inconsistencies exist, eg. if someone's name or title is misspelled
- the redundancies that exist, could lead to insertion anomalies (eg. a new employee needs to be assigned to some project, even a fake one), update anomalies (eg. if an employee's JOB_CLASS changes, it has to be modified multiple times), deletion anomalies (eg. if a project has just one employee and that employee leaves, deleting the lone employee record would lead to the project itself getting deleted!)
- data redundancy leads to wasted storage space

We have 'repeating groups' (for each project, we list all details about each employee) - our table is un-normalized, ie. is in '0NF' :)

So, we need to clean up the design!

Objectives: what do we want?

Normalization Process

- Objective is to ensure that each table conforms to the concept of well-formed relations
  - 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, and deletion anomalies

wholly; nothing but

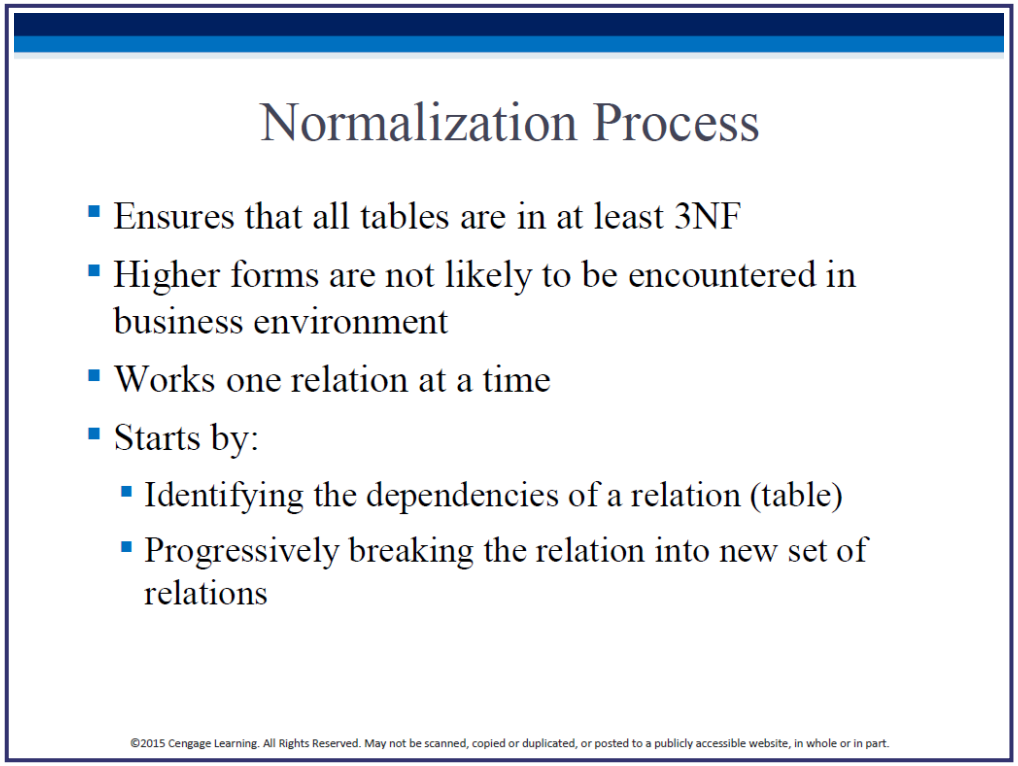
Normal forms

Normalization is a systematic process that yields progressively higher 'normal forms' (NFs) for each entity (table) in our db. We want **at least** 3NF for each table; in RL, we stop **at** 3NF.

Table 6.2 - Normal Forms

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

The process



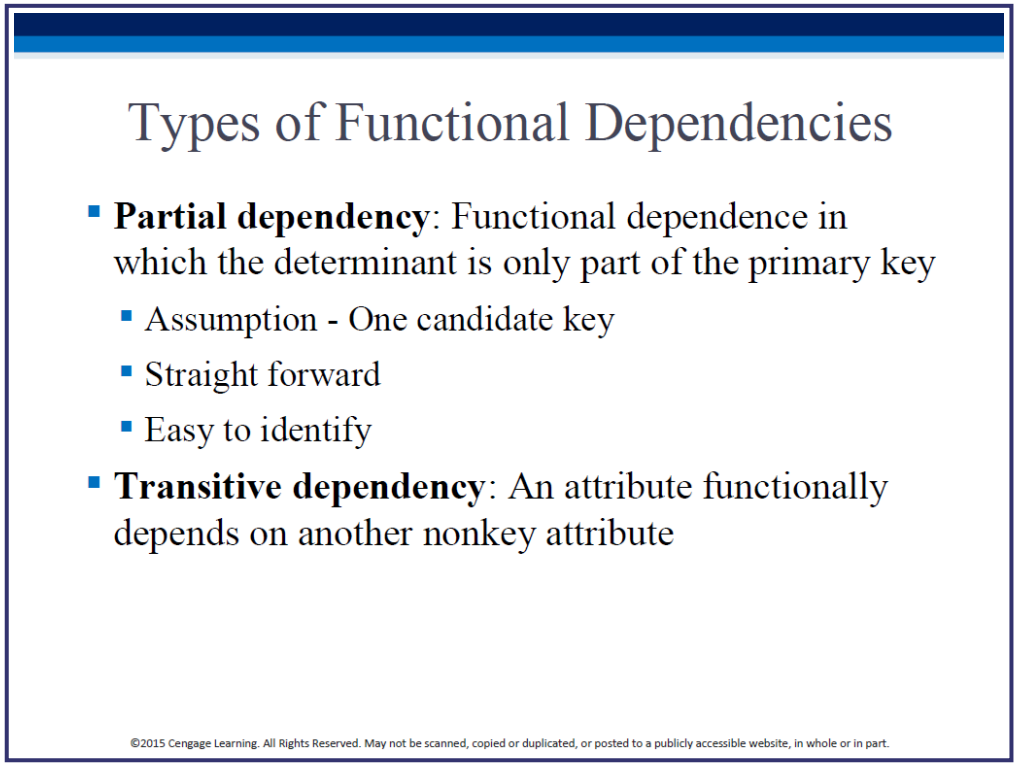
Normalization how-to, in one sentence: **work on one relation (table) at a time: identify dependencies, then 'normalize' - progressively break it down into smaller relations (tables), based on the dependencies we identify in the original relation** so that "only the PK, the whole PK and nothing but the PK" acts as a determinant! But how?? Details follow..

Functional dependence, determination

Functional Dependence Concepts

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.
Functional dependence (Generalized definition)	Attribute A determines attribute B 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.

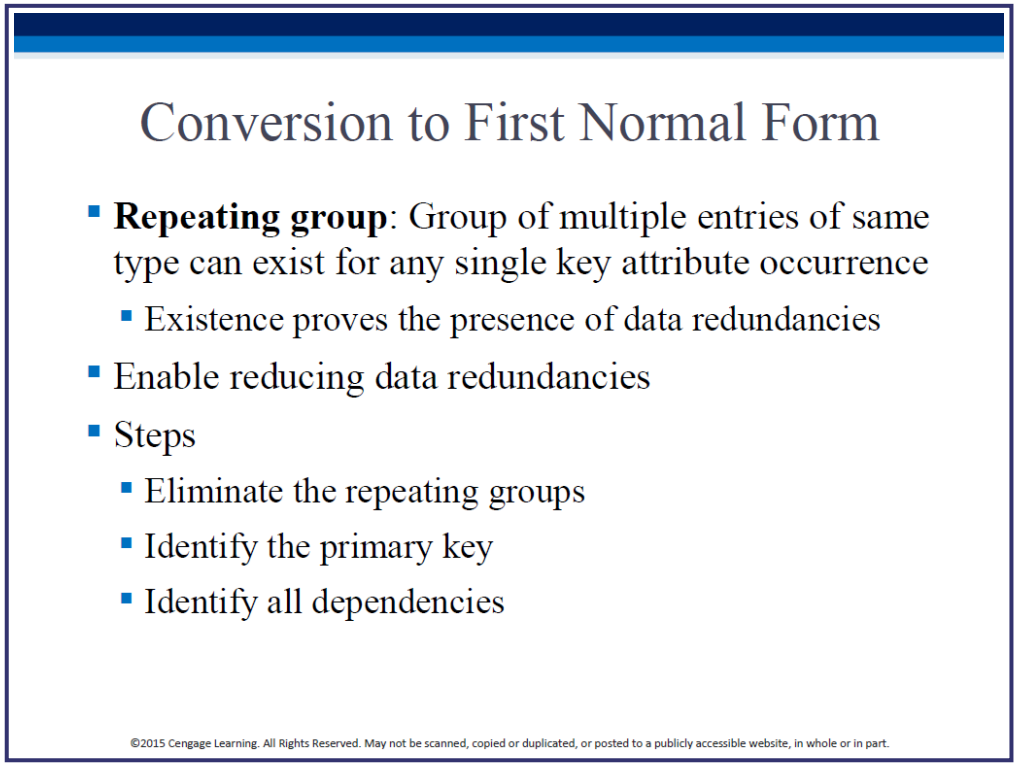
Partial dependency, transitive dependency



If (A,B) is a primary key, we have **partial** dependence if (A,B)->(C,D) and B->C [C is only partially dependent on the PK, ie. we only need B to determine C]. In other words, a part of an existing PK is acting like a PK on its own.

If X is a primary key, we have a **transitive** dependency if X->Y and Y->Z [Z is transitively dependent on X, not directly so]. In other words, a non-PK (regular attr) is acting like a PK.

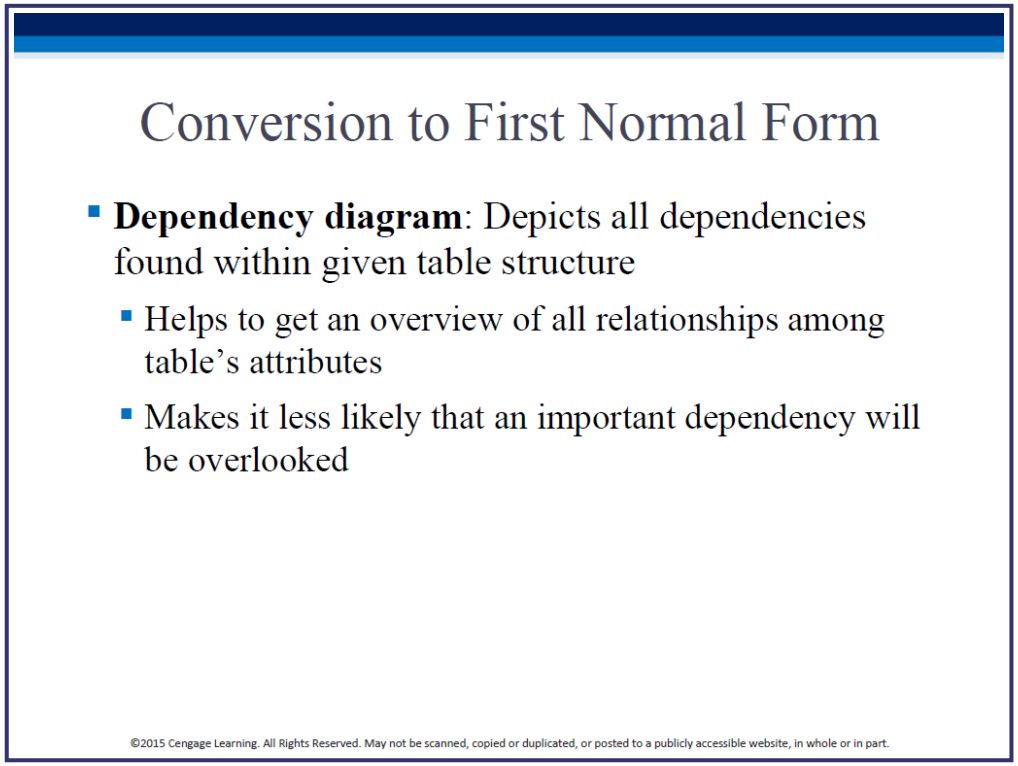
0NF->1NF: eliminate repeating groups



In other words, "fill in the blanks" so that there are no nulls. Now we have a relation (table), with a value in each cell.

Further, identify the PK! In our example, it is (PROJ_NUM,EMP_NUM).

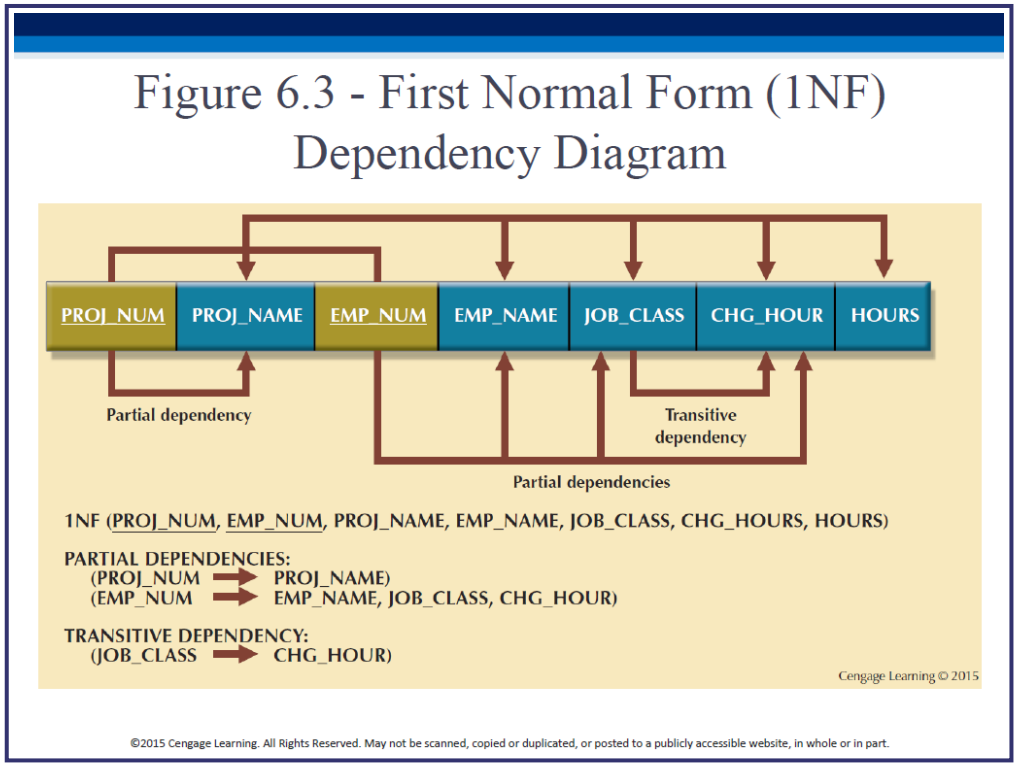
0NF->1NF [cont'd]



Create a **dependency diagram**, showing relationships (dependencies) between the attributes - this will help us systematically normalize the table.

Dependency diagram

Indicate full dependencies on the top, and partial and transitive dependencies on the bottom. "Top good, bottom bad". Also, color the PK components in a different color (and underline them). Result:



PROJ_NAME has only a partial dependency on the PK (since it is only dependent on PROJ_NUM, which is just a part of the PK).

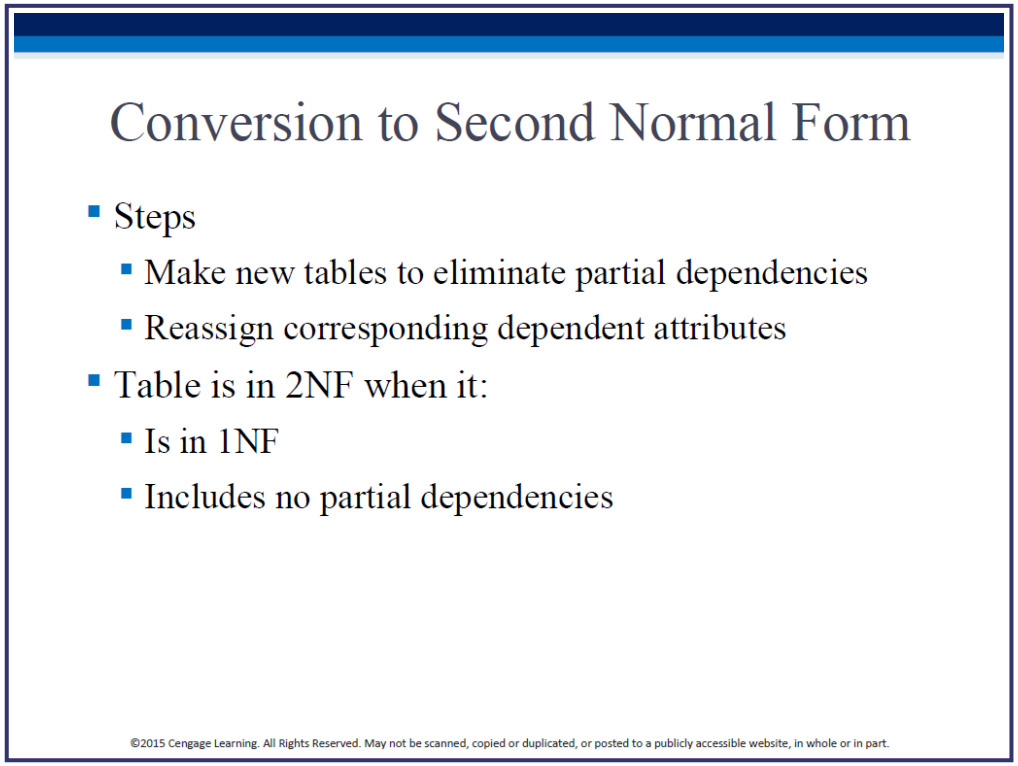
CHG_HOUR is dependent on JOB_CLASS, which is a non-prime attribute that is itself dependent on EMP_NUM. So JOB_CLASS->CHG_HOUR is a signaling dependency, indicating a EMP_NUM -> CHG_HOUR transitive dependency.

0NF->1NF [cont'd]

Conversion to First Normal Form

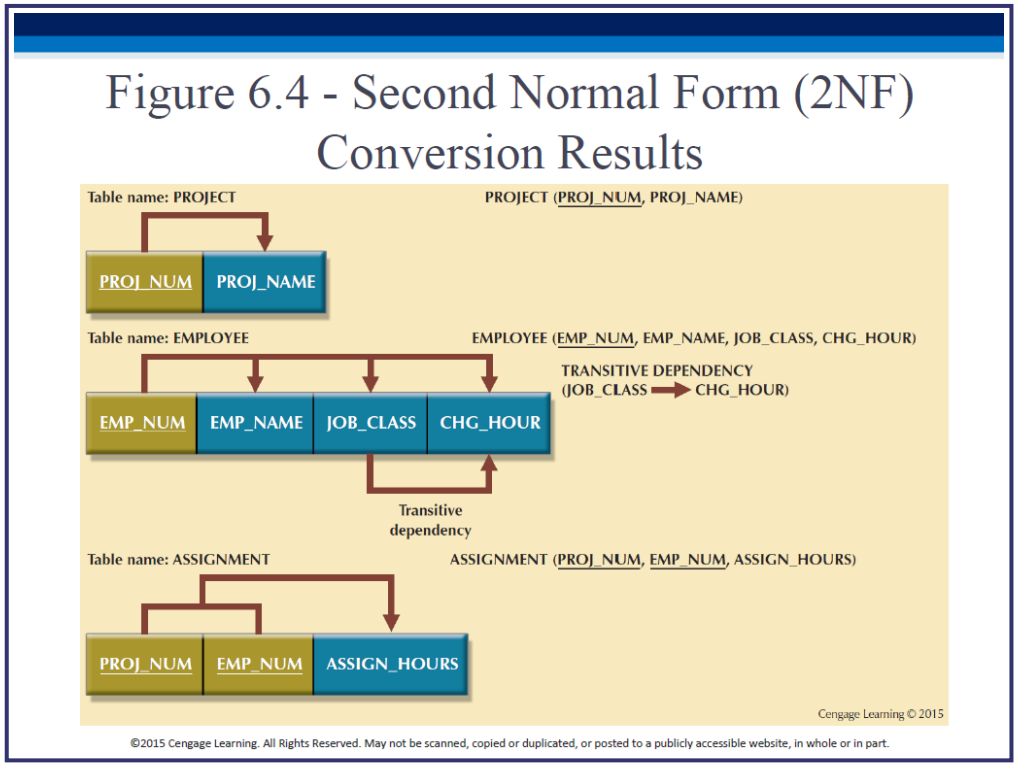
- 1NF describes tabular format in which:
  - All key attributes are defined
  - There are no repeating groups in the table
  - All attributes are dependent on the primary key
- All relational tables satisfy 1NF requirements
- Some tables contain partial dependencies
  - Subject to data redundancies and various anomalies

1NF->2NF: remove partial dependencies



1NF->2NF [cont'd]

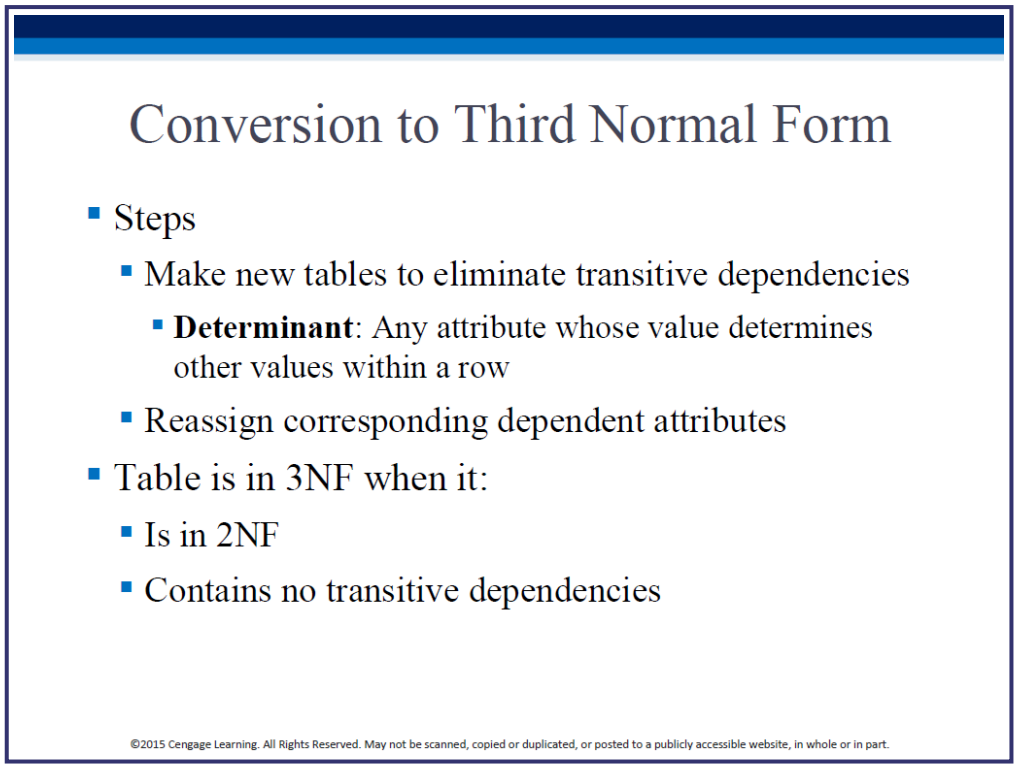
We eliminate partial dependencies by creating separate tables of such dependencies, and removing the dependent attributes from the starter table.



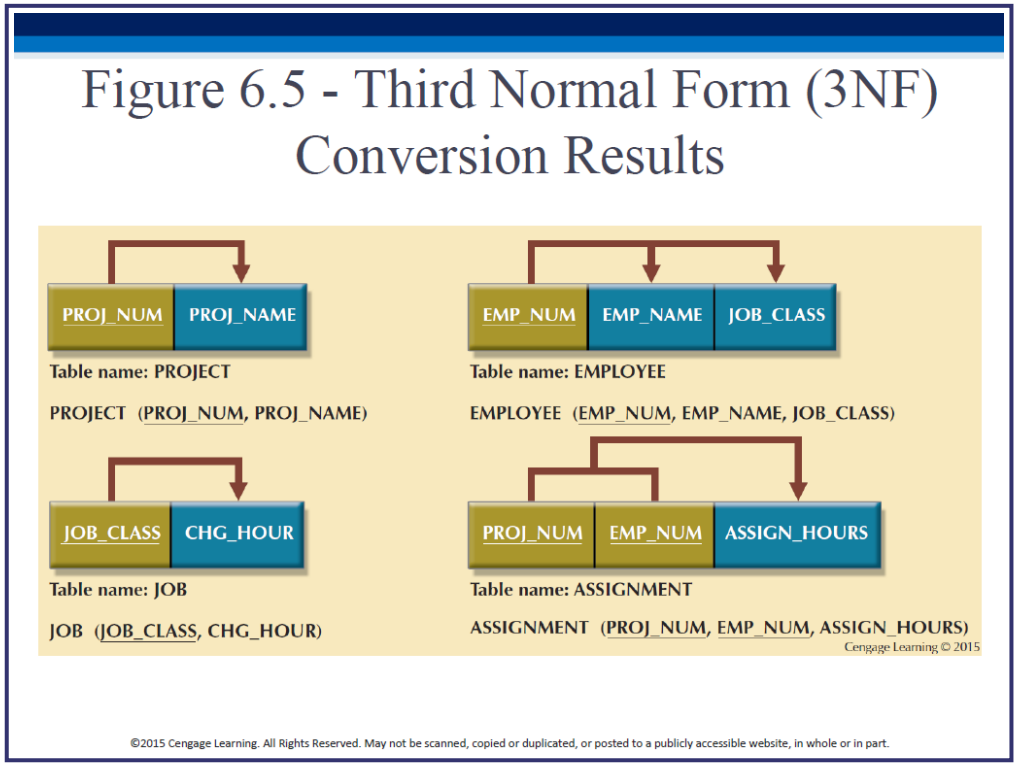
2NF->3NF: remove transitive dependencies

We promote the non-prime keys that masquerade as PKs, into actual PKs (give them their own tables).

Whether we eliminate partial dependencies (to create 2NF) or transitive ones (to create 3NF), we follow the same process: create a new relation for each 'problem' dependency!



2NF->3NF [cont'd]



'Good' tables

We can create a better DB by doing the following augmentations, to the 3NF model we just created:

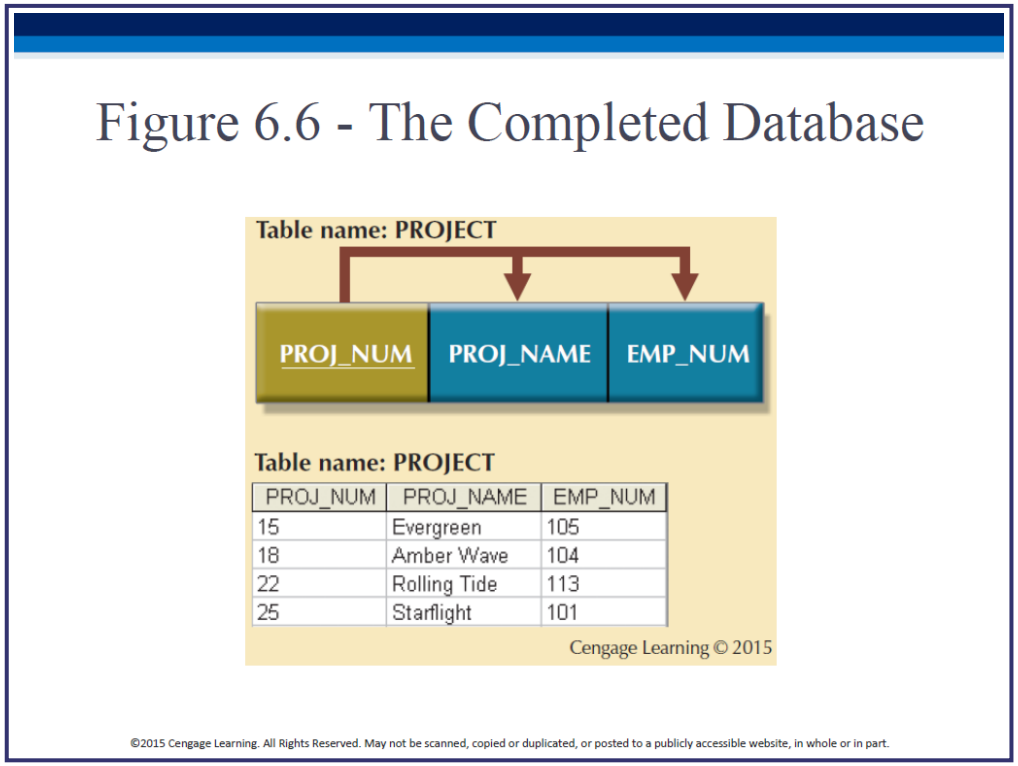
- evaluate PKs - create a JOB_CODE
- evaluate naming conventions - eg. JOB_CHG_HOUR
- refine attr atomicity, eg. EMP_NAME
- identify new attrs, eg. EMP_HIREDATE
- identify new relationships, PROJECT can have EMP_NUM as FK [to be able to record a project's (always sole) manager]
- refine PKs for data granularity, eg. ASSIGN_NUM
- maintain historical accuracy [duplicate data], eg. store JOB_CHG_HOUR in ASSIGNMENT
- evaluate derived attrs, eg. ASSIGN_CHARGE

Requirements for Good Normalized Set of Tables

- Evaluate PK assignments and naming conventions
- Refine attribute atomicity
  - **Atomic attribute**: Cannot be further subdivided
  - **Atomicity**: Characteristic of an atomic attribute
- Identify new attributes and new relationships
- Refine primary keys as required for data granularity
  - **Granularity**: Level of detail represented by the values stored in a table's row
- Maintain historical accuracy and evaluate using derived attributes

Final result

Here is the result of making the "extra" changes to our 3NF form:



Final result [cont'd]

Figure 6.6 - The Completed Database

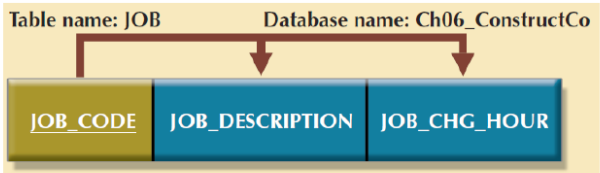


Table name: JOB

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

Cengage Learning © 2015

Final result [cont'd]

Figure 6.6 - The Completed Database

Table name: ASSIGNMENT

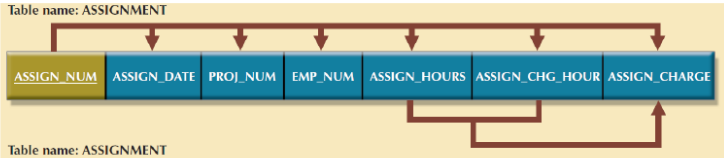


Table name: ASSIGNMENT

ASSIGN_NUM	ASSIGN_DATE	PROJ_NUM	EMP_NUM	ASSIGN_HOURS	ASSIGN_CHG_HOUR	ASSIGN_CHARGE
1001	04-Mar-14	15	103	2.6	84.50	219.70
1002	04-Mar-14	18	118	1.4	18.36	25.70
1003	05-Mar-14	15	101	3.6	105.00	378.00
1004	05-Mar-14	22	113	2.5	48.10	120.25
1005	05-Mar-14	15	103	1.9	84.50	160.55
1006	05-Mar-14	25	115	4.2	96.75	406.35
1007	05-Mar-14	22	105	5.2	105.00	546.00
1008	05-Mar-14	25	101	1.7	105.00	178.50
1009	05-Mar-14	15	105	2.0	105.00	210.00
1010	06-Mar-14	15	102	3.8	96.75	367.65
1011	06-Mar-14	22	104	2.6	96.75	251.55
1012	06-Mar-14	15	101	2.3	105.00	241.50
1013	06-Mar-14	25	114	1.8	48.10	86.58
1014	06-Mar-14	22	111	4.0	26.87	107.48
1015	06-Mar-14	25	114	3.4	48.10	163.54
1016	06-Mar-14	18	112	1.2	45.95	55.14
1017	06-Mar-14	18	118	2.0	18.36	36.72
1018	06-Mar-14	18	104	2.6	96.75	251.55
1019	06-Mar-14	15	103	3.0	84.50	253.50
1020	07-Mar-14	22	105	2.7	105.00	283.50
1021	08-Mar-14	25	108	4.2	96.75	406.35
1022	07-Mar-14	25	114	5.8	48.10	278.98
1023	07-Mar-14	22	106	2.4	35.75	85.80

Cengage Learning © 2015

Final result [cont'd]

Figure 6.6 - The Completed Database

Table name: EMPLOYEE Database name: Ch06_ConstructCo

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

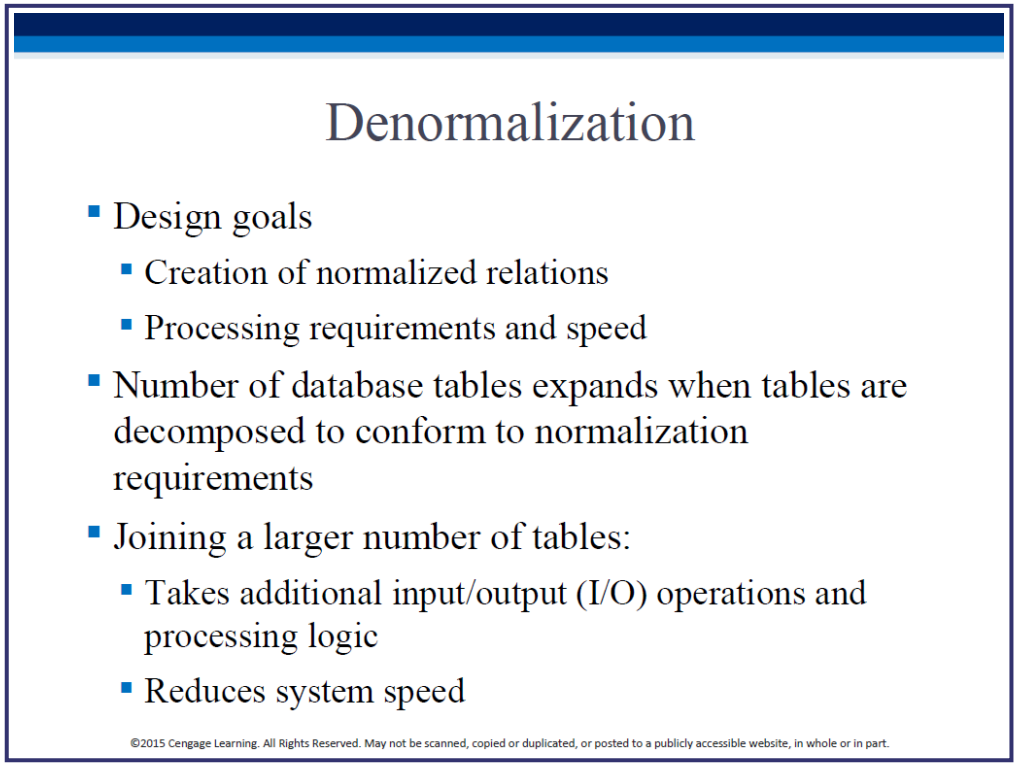
Cengage Learning © 2015

Normalization: summary

* 1NF: eliminate repeating groups (partial:y, transitive:y)
* 2NF: eliminate redundant data (partial:n, transitive:y)
* 3NF: eliminate fields not dependent on key fields (partial:n, transitive:n)

Here is more, on normalization.

Denormalization



Denormalization [cont'd]