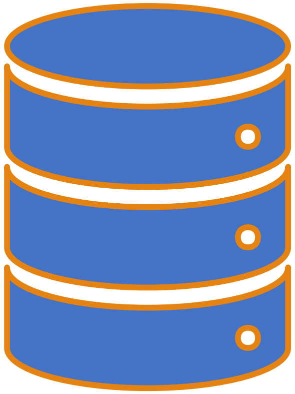
6. Normalization of Database Tables

DATABASE DESIGN & BUSINESS APPLICATION DEVELOPMENT

Objectives



- Explain normalization and its role in the database design process
- Identify and describe each of the normal forms: 1NF, 2NF, 3NF, BCNF, and 4NF
- Explain how normal forms can be transformed from lower normal forms to higher normal forms
- Apply normalization rules to evaluate and correct table structures
- Identify situations that require denormalization to generate information efficiently
- Use a data-modeling checklist to check that the ERD meets a set of minimum requirements

Database Tables and Normalization

Normalization: evaluating and correcting table structures to minimize data redundancies

Reduces data anomalies

Assigns attributes to tables based on determination

Normal forms

First normal form (1NF)

Second normal form (2NF)

Third normal form (3NF)

Database Tables and Normalization

Structural point of view of normal forms

- Higher normal forms are better than lower normal forms
- Properly designed 3NF structures meet the requirement of fourth normal form (4NF)

Denormalization: produces a lower normal form

- Results in increased performance and greater data redundancy

The Need for Normalization

Used while designing a new database structure

- Analyzes the relationship among the attributes within each entity
- Determines if the structure can be improved through normalization
- Improves the existing data structure and creates an appropriate database design

The Normalization Process

Objective is to ensure that each table conforms to the concept of well-formed relations

- Each table represents a single subject
- Each row/column intersection contains only one value and not a group of values
- 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 has no insertion, update, or deletion anomalies

The Normalization Process

Ensures that all tables are in at least 3NF

- Higher forms are not likely to be encountered in business environment

Works one relation at a time

- Identifies the dependencies of a relation (table)
- Progressively breaks the relation up into a new set of relations

The Normalization Process

Normal Form	Characteristic
First normal form (1NF)	Table format, no repeating groups, and PK identified
Second normal form (2NF)	1NF and no partial dependencies
Third normal form (3NF)	2NF and no transitive dependencies
Boyce-Codd normal form (BCNF)	Every determinant is a candidate key (special case of 3NF)
Fourth normal form (4NF)	3NF and no independent multivalued dependencies

The Normalization Process

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 S 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 (generalized definition) 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.

The Normalization Process

Partial dependency: functional dependence in which the determinant is only part of the primary key

- Assumption: one candidate key
- Straight forward
- Easy to identify

Transitive dependency: attribute is dependent on another attribute that is not part of the primary key

- More difficult to identify among a set of data
- Occur only when a functional dependence exists among nonprime attributes

Conversion to First Normal Form (1NF)

Repeating group: group of multiple entries of same type can exist for any single key attribute occurrence

Reduces data redundancies

Three step procedure

Eliminate the repeating groups

Identify the primary key

Identify all dependencies

Dependency diagram: depicts all dependencies found within given table structure

Helps to get an overview of all relationships among table's attributes

Makes it less likely that an important dependency will be overlooked

Conversion to First Normal Form (1NF)

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

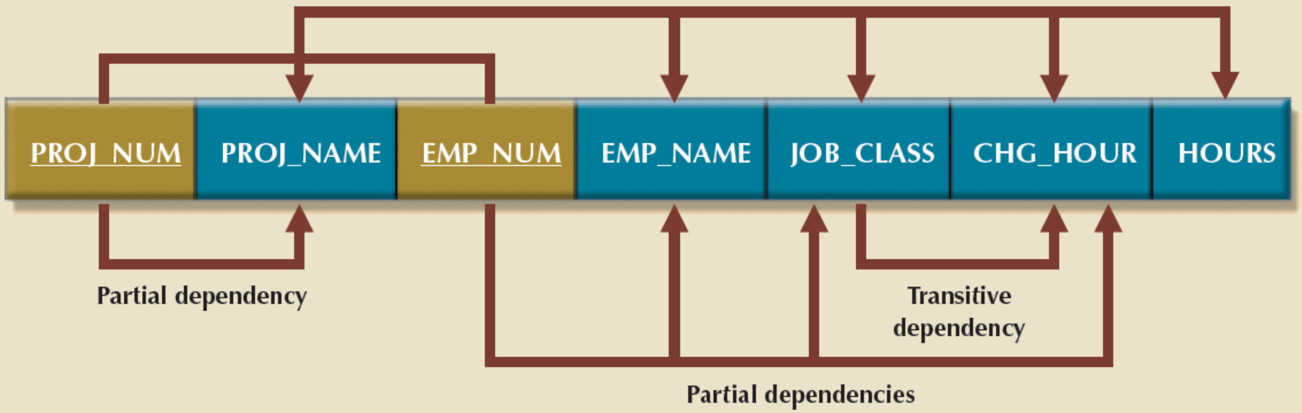
All relational tables satisfy 1NF requirements

Some tables contain partial dependencies

Update, insertion, or deletion

Conversion to First Normal Form (1NF)

FIGURE 6.3 FIRST NORMAL FORM (1NF) DEPENDENCY DIAGRAM



1NF (PROJ_NUM, EMP_NUM, PROJ_NAME, EMP_NAME, JOB_CLASS, CHG_HOURS, HOURS)

PARTIAL DEPENDENCIES:
(PROJ_NUM → PROJ_NAME)
(EMP_NUM → EMP_NAME, JOB_CLASS, CHG_HOUR)

TRANSITIVE DEPENDENCY:
(JOB_CLASS → CHG_HOUR)

Conversion to Second Normal Form (2NF)

Conversion to 2NF occurs only when the 1NF has a composite primary key

If the 1NF has a single-attribute primary key, then the table is automatically in 2NF

The 1NF-to-2NF conversion is simple

Make new tables to eliminate partial dependencies

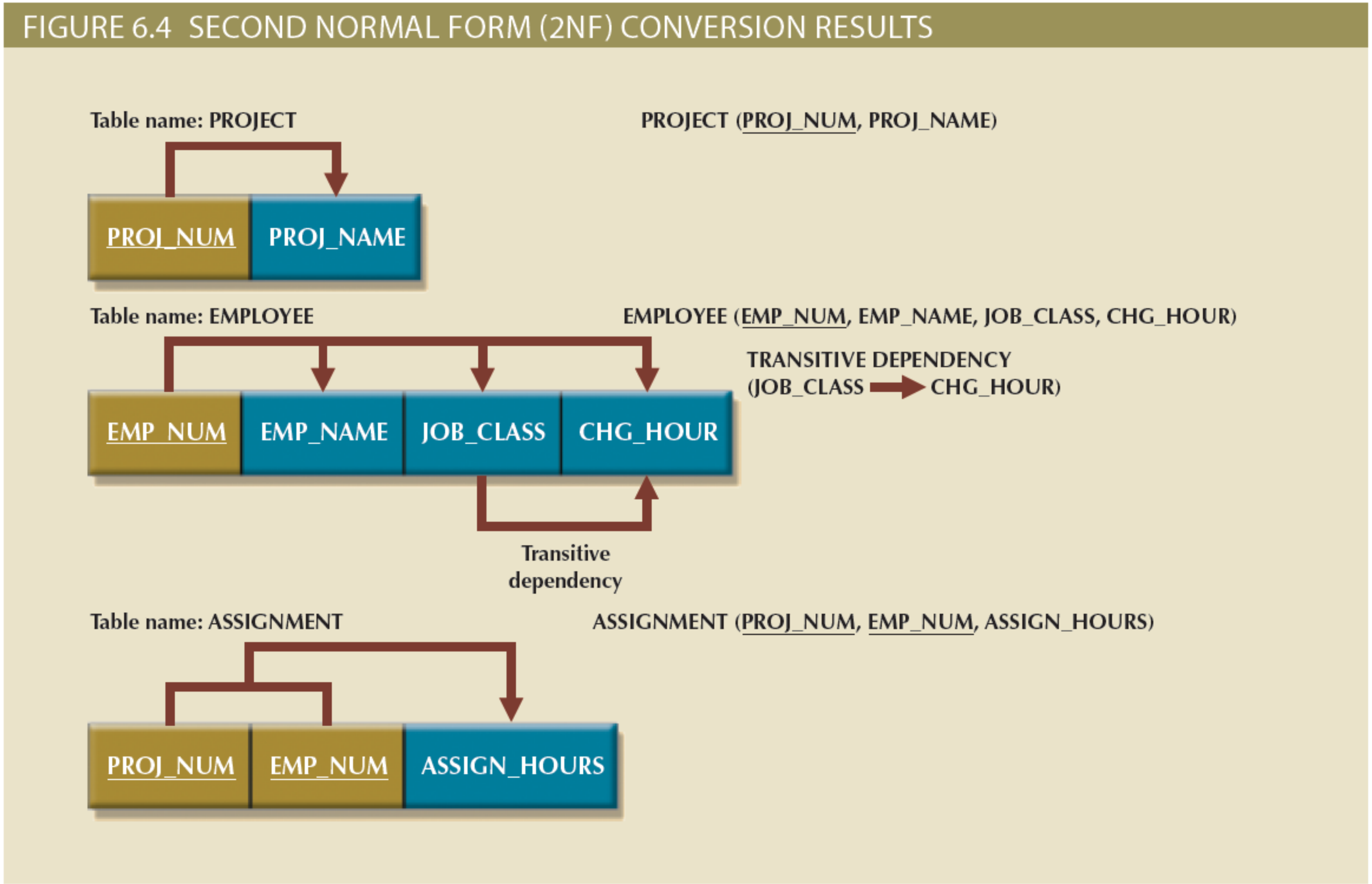
Reassign corresponding dependent attributes

Table is in 2NF when it:

Is in 1NF

Includes no partial dependencies

Conversion to Second Normal Form (2NF)



Conversion to Third Normal Form (3NF)

The data anomalies created by the database organization shown in Figure 6.4 are easily eliminated

- Make new tables to eliminate transitive dependencies
- Reassign corresponding dependent attributes

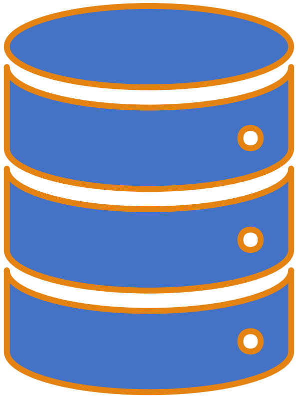
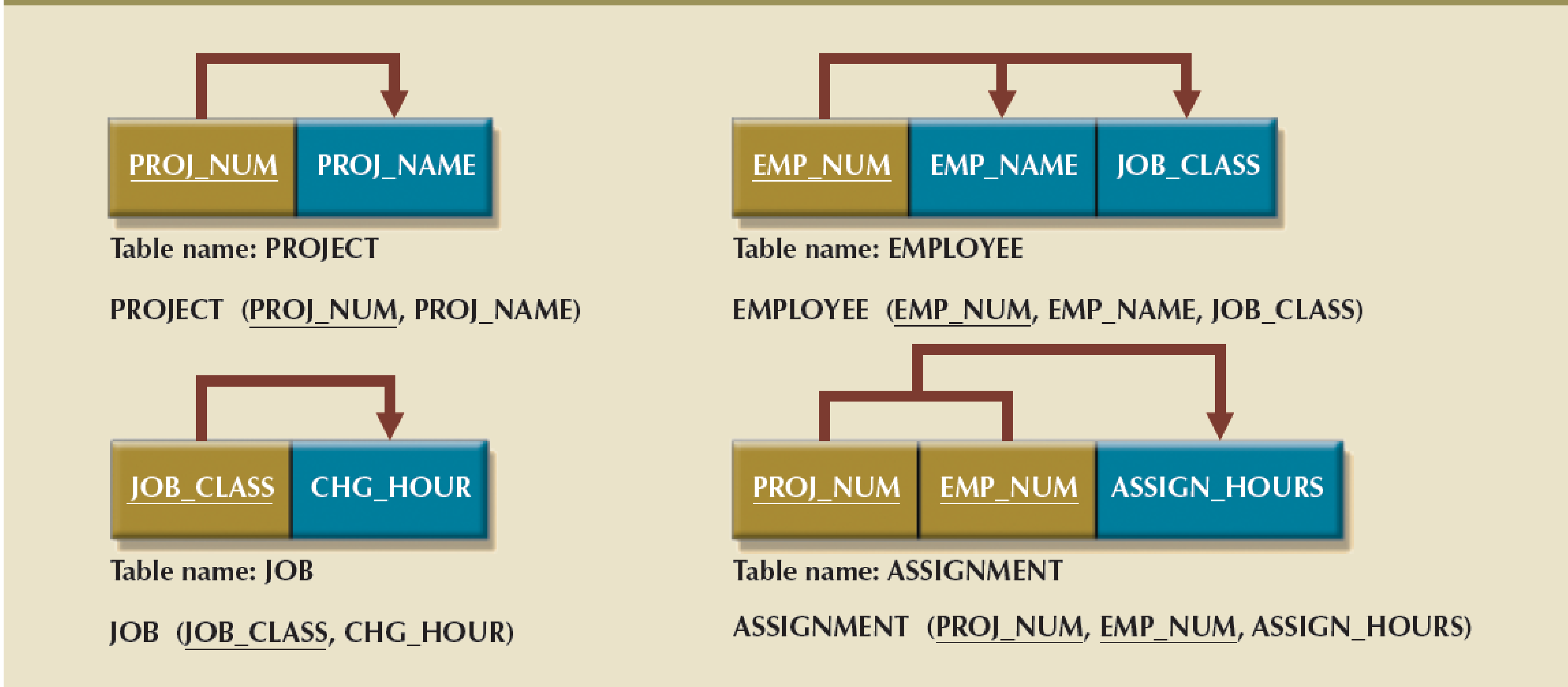


Table is in 3NF when it:

- Is in 2NF
- Contains no transitive dependencies

Conversion to Third Normal Form (3NF)

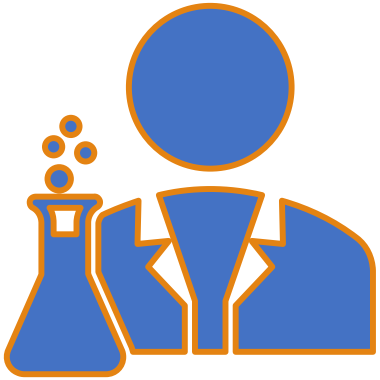
FIGURE 6.5 THIRD NORMAL FORM (3NF) CONVERSION RESULTS



Improving the Design

Normalization is valuable because its use helps eliminate data redundancies

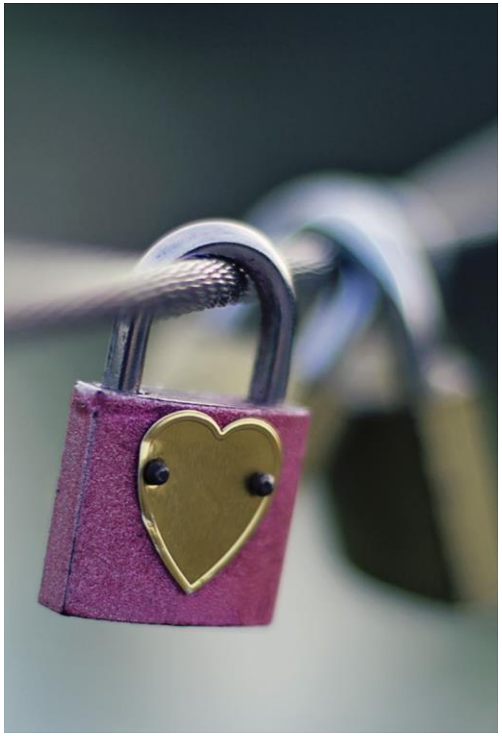
- Evaluate PK assignments and naming conventions
- Refine attribute atomicity
  - Atomic attribute: cannot be further subdivided
  - Atomicity: characteristic of an atomic attribute
- Identify new attributes and new relationships
- Refine primary keys as required for data granularity
  - Granularity: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



Surrogate Key Considerations

Used by designers when the primary key is considered to be unsuitable

- System-defined attribute
- Created a managed via the DBMS
- Have a numeric value which is automatically incremented for each new row



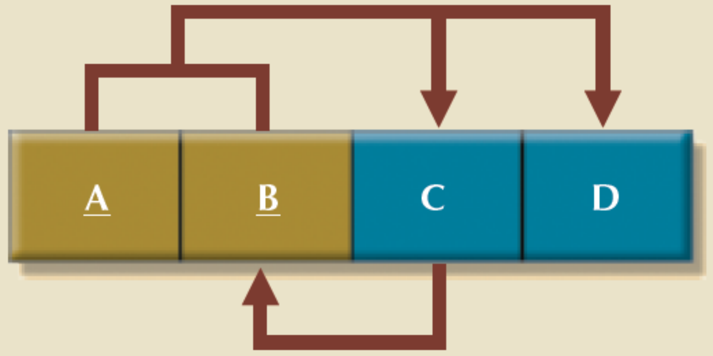
The Boyce-Codd Normal Form

Every determinant in the table should be a candidate key

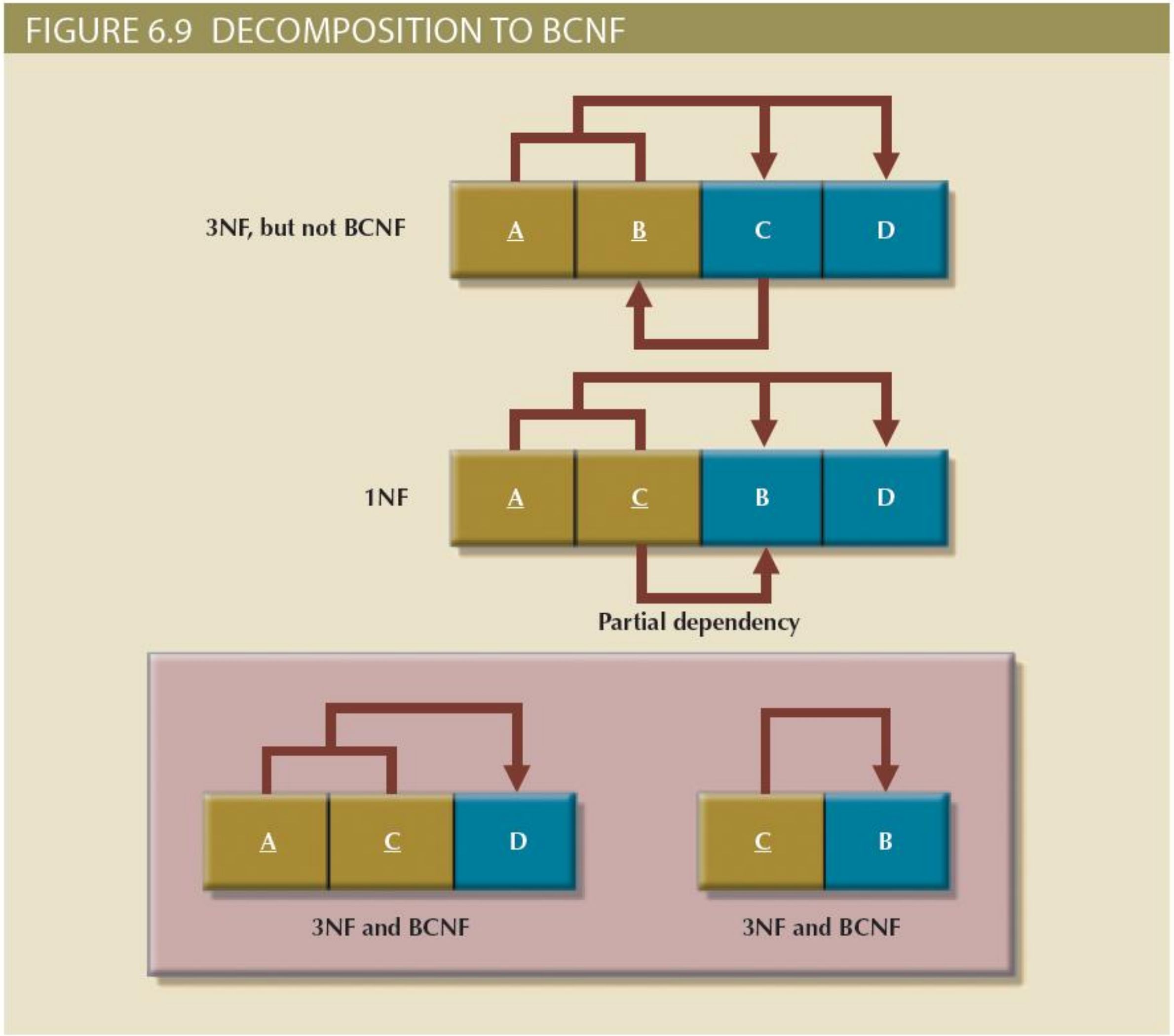
- Candidate key: same characteristics as primary key but not chosen to be the primary key
- Equivalent to 3NF when the table contains only one candidate key
- Violated only when the table contains more than one candidate key
- Considered to be a special case of 3NF

The Boyce-Codd Normal Form

FIGURE 6.8 A TABLE THAT IS IN 3NF BUT NOT IN BCNF



The Boyce-Codd Normal Form



The Boyce-Codd Normal Form

Sample Data for a BCNF 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

Fourth Normal Form (4NF)

Rules

- All attributes must be dependent on the primary key, but they must be independent of each other
- No row may contain two or more multivalued facts about an entity

Table is in 4NF when it:

- Is in 3NF
- Has no multivalued dependencies

Fourth Normal Form (4NF)

FIGURE 6.11 TABLES WITH MULTIVALUED DEPENDENCIES

Database name: Ch06_Service

Table name: VOLUNTEER_V1

EMP_NUM	ORG_CODE	ASSIGN_NUM
10123	RC	1
10123	UW	3
10123		4

Table name: VOLUNTEER_V2

EMP_NUM	ORG_CODE	ASSIGN_NUM
10123	RC	
10123	UW	
10123		1
10123		3
10123		4

Table name: VOLUNTEER_V3

EMP_NUM	ORG_CODE	ASSIGN_NUM
10123	RC	1
10123	RC	3
10123	UW	4

Normalization and Database Design

Normalization should be part of the design process

Proposed entities must meet required the normal form before table structures are created

Principles and normalization procedures to be understood to redesign and modify databases

ERD is created through an iterative process

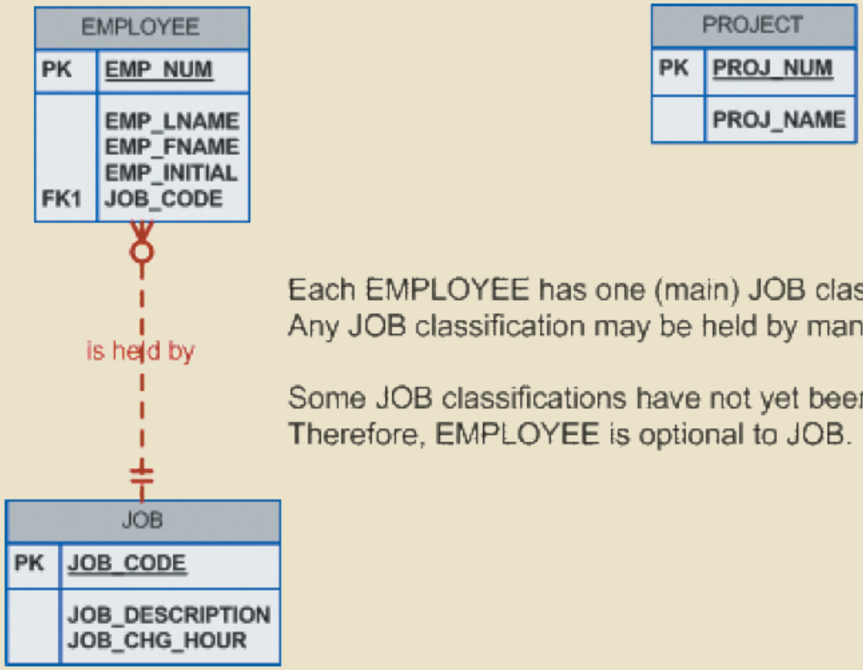
Normalization focuses on the characteristics of specific entities

Normalization and Database Design

FIGURE 6.13 INITIAL CONTRACTING COMPANY ERD

Normalization and Database Design

FIGURE 6.14 MODIFIED CONTRACTING COMPANY ERD

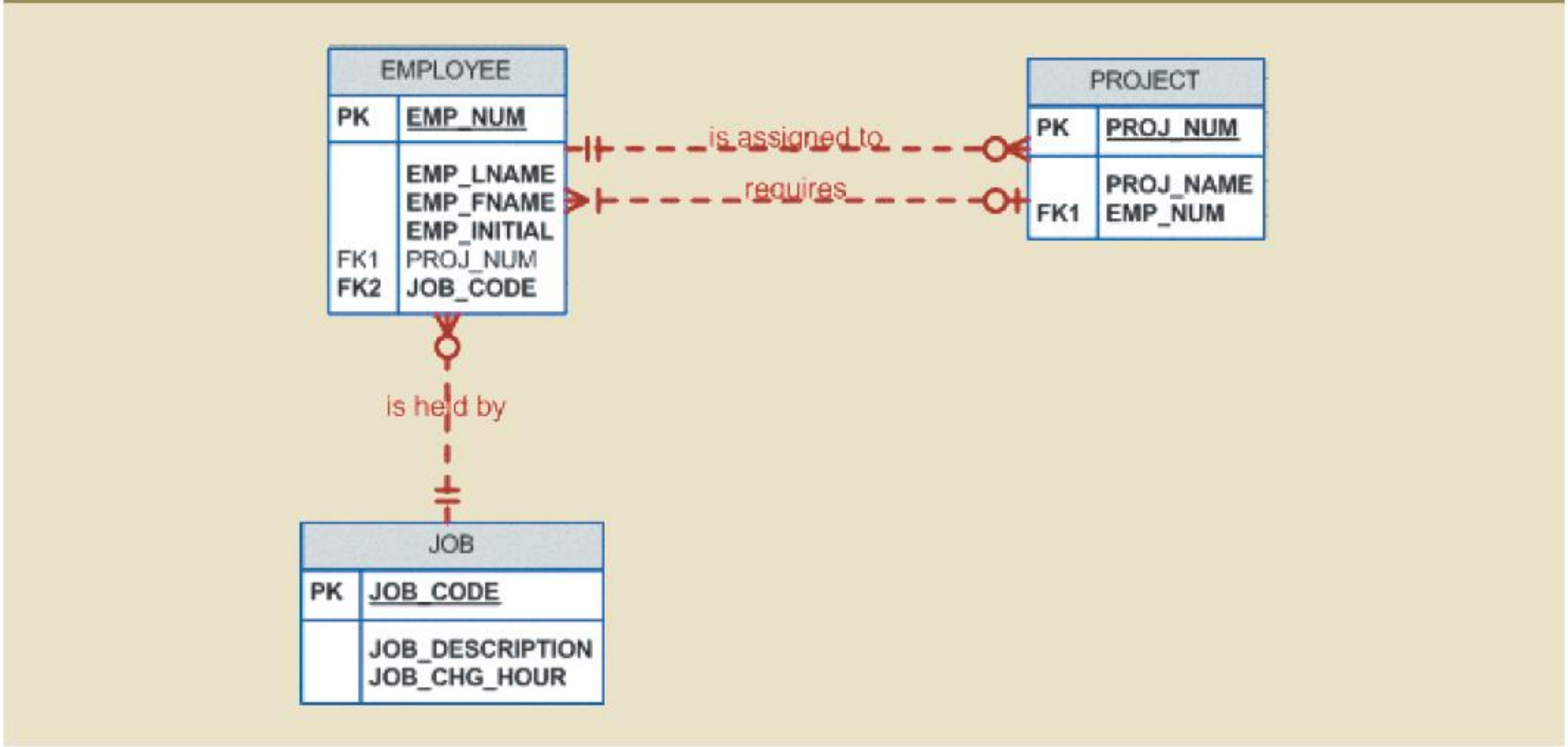


Each EMPLOYEE has one (main) JOB classification.
Any JOB classification may be held by many EMPLOYEEs.

Some JOB classifications have not yet been staffed.
Therefore, EMPLOYEE is optional to JOB.

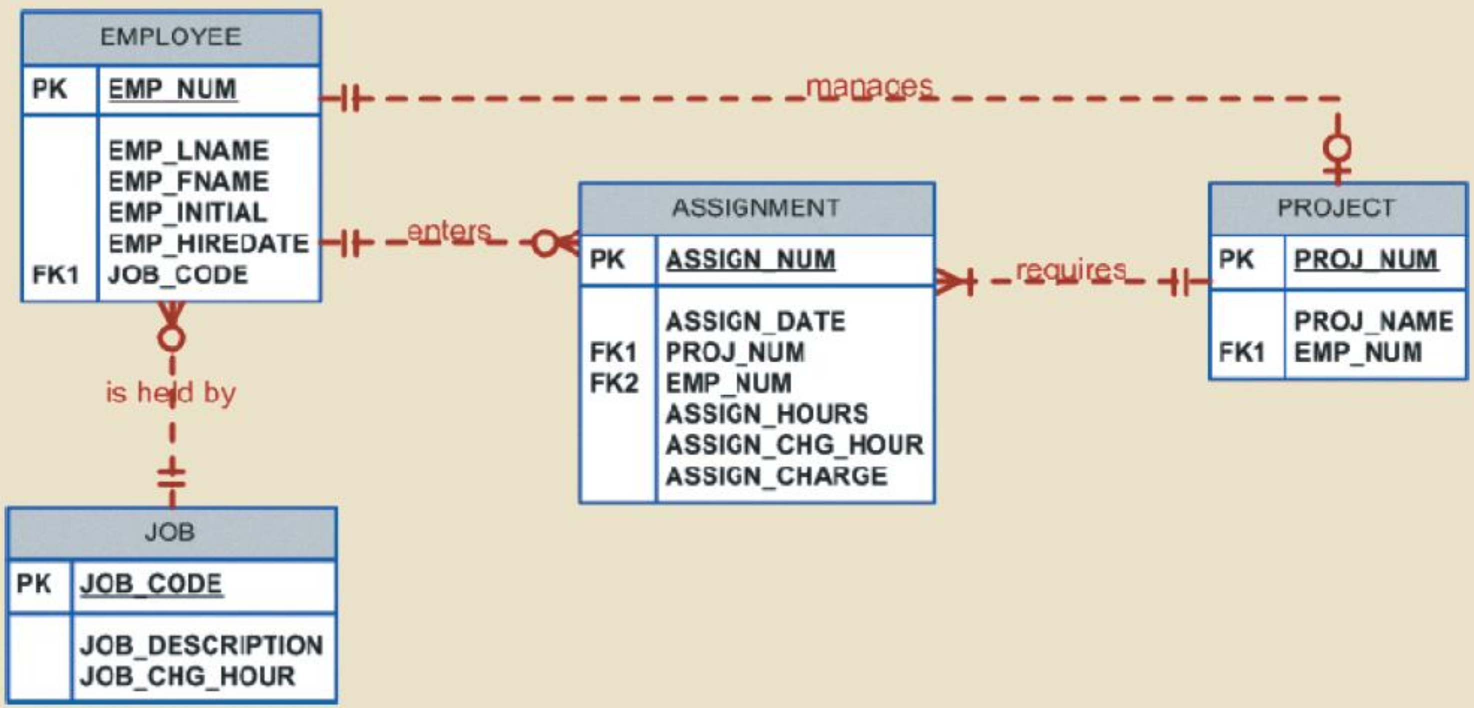
Normalization and Database Design

FIGURE 6.15 INCORRECT M:N RELATIONSHIP REPRESENTATION



Normalization and Database Design

FIGURE 6.16 FINAL CONTRACTING COMPANY ERD



Normalization and Database Design

FIGURE 6.17 THE IMPLEMENTED 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-16	15	103	2.6	84.50	219.70
1002	04-Mar-16	18	118	1.4	18.36	25.70
1003	05-Mar-16	15	101	3.6	105.00	378.00
1004	05-Mar-16	22	113	2.5	48.10	120.25
1005	05-Mar-16	15	103	1.9	84.50	160.55
1006	05-Mar-16	25	115	4.2	96.75	406.35
1007	05-Mar-16	22	105	5.2	105.00	546.00
1008	05-Mar-16	25	101	1.7	105.00	178.50
1009	05-Mar-16	15	105	2.0	105.00	210.00
1010	06-Mar-16	15	102	3.8	96.75	367.65
1011	06-Mar-16	22	104	2.6	96.75	251.55
1012	06-Mar-16	15	101	2.3	105.00	241.50
1013	06-Mar-16	25	114	1.8	48.10	86.58
1014	06-Mar-16	22	111	4.0	26.87	107.48
1015	06-Mar-16	25	114	3.4	48.10	163.54
1016	06-Mar-16	18	112	1.2	45.95	55.14
1017	06-Mar-16	18	118	2.0	18.36	36.72
1018	06-Mar-16	18	104	2.6	96.75	251.55
1019	06-Mar-16	15	103	3.0	84.50	253.50
1020	07-Mar-16	22	105	2.7	105.00	283.50
1021	08-Mar-16	25	108	4.2	96.75	406.35
1022	07-Mar-16	25	114	5.8	48.10	278.98
1023	07-Mar-16	22	106	2.4	35.75	85.80

Denormalization

Design goals

Creation of normalized relations

Processing requirements and speed

Number of database tables expands

Tables are decomposed to conform to normalization requirements

Joining a larger number of tables

Takes additional input/output (I/O) operations and processing logic

Reduces system speed

Defects in unnormalized tables

Data updates are less efficient because tables are larger

Indexing is more cumbersome

No simple strategies for creating virtual tables known as views

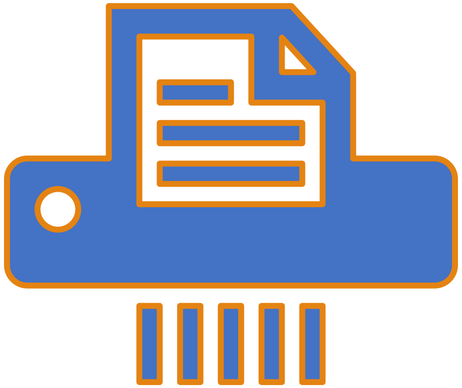
Denormalization

Table 6.6: Common Denormalization Examples		
Case	**Example**	**Rationale and Controls**
Redundant data	Storing ZIP and CITY attributes in the AGENT table when ZIP determines CITY (see Figure 2.2)	Avoid extra join operations Program can validate city (drop-down box) based on the zip code
Derived data	Storing STU_HRS and STU_CLASS (student classification) when STU_HRS determines STU_CLASS (see Figure 3.28)	Avoid extra join operations Program can validate classification (lookup) based on the student hours
Preaggregated data (also derived data)	Storing the student grade point average (STU_GPA) aggregate value in the STUDENT table when this can be calculated from the ENROLL and COURSE tables (see Figure 3.28)	Avoid extra join operations Program computes the GPA every time a grade is entered or updated STU_GPA can be updated only via administrative routine
Information requirements	Using a temporary denormalized table to hold report data; this is required when creating a tabular report in which the columns represent data that are stored in the table as rows (see Figures 6.17 and 6.18)	Impossible to generate the data required by the report using plain SQL No need to maintain table Temporary table is deleted once report is done Processing speed is not an issue

Data-Modeling Checklist

Business rules

- Properly document and verify all business rules with the end users
- Ensure that all business rules are written precisely, clearly, and simply
  - The business rules must help identify entities, attributes, relationships, and constraints
- Identify the source of all business rules, and ensure that each business rule is justified, dated, and signed off by an approving authority



Data-Modeling Checklist

Data modeling

- Naming conventions: all names should be limited in length (database-dependent size)

Entity names:

- Should be nouns that are familiar to business and should be short and meaningful
- Should document abbreviations, synonyms, and aliases for each entity
- Should be unique within the model
- For composite entities, may include a combination of abbreviated names of the entities linked through the composite entity

Data-Modeling Checklist

Attribute names:

Should be unique within the entity

Should use the entity abbreviation as a prefix

Should be descriptive of the characteristic

Should use suffixes such as _ID, _NUM, or _CODE for the PK attribute

Should not be a reserved word

Should not contain spaces or special characters such as @, !, or &

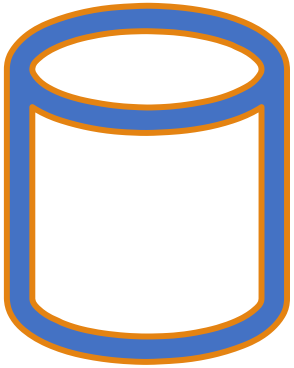
Relationship names:

Should be active or passive verbs that clearly indicate the nature of the relationship

Data-Modeling Checklist

Entities:

- Each entity should represent a single subject
- Each entity should represent a set of distinguishable entity instances
- All entities should be in 3NF or higher
  - Any entities below 3NF should be justified
- Granularity of the entity instance should be clearly defined
- PK should be clearly defined and support the selected data granularity



Data-Modeling Checklist

Attributes:

Should be simple and single-valued (atomic data)

Should document default values, constraints, synonyms, and aliases

Derived attributes should be clearly identified and include source(s)

Should not be redundant unless this is required for transaction accuracy, performance, or maintaining a history

Nonkey attributes must be fully dependent on the PK attribute

Relationships:

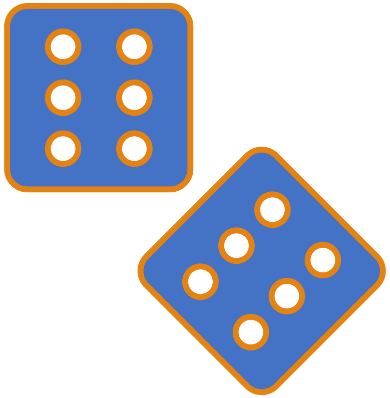
Should clearly identify relationship participants

Should clearly define participation, connectivity, and document cardinality

Data-Modeling Checklist

ER model:

- Should be validated against expected processes: inserts, updates, and deletions
- Should evaluate where, when, and how to maintain a history
- Should not contain redundant relationships except as required (see attributes)
- Should minimize data redundancy to ensure single-place updates
- Should conform to the minimal data rule: All that is needed is there, and all that is there is needed



Summary

NORMALIZATION IS A TECHNIQUE USED TO DESIGN TABLES IN WHICH DATA REDUNDANCIES ARE MINIMIZED

A TABLE IS IN 1NF WHEN ALL KEY ATTRIBUTES ARE DEFINED, AND ALL REMAINING ATTRIBUTES ARE DEPENDENT ON THE PRIMARY KEY

A TABLE IS IN 2NF WHEN IT IS IN 1NF AND CONTAINS NO PARTIAL DEPENDENCIES

A TABLE IS IN 3NF WHEN IT IS IN 2NF AND CONTAINS NO TRANSITIVE DEPENDENCIES

A TABLE THAT IS NOT IN 3NF MAY BE SPLIT INTO NEW TABLES UNTIL ALL OF THE TABLES MEET THE 3NF REQUIREMENTS

NORMALIZATION IS AN IMPORTANT PART—BUT ONLY A PART—OF THE DESIGN PROCESS

A TABLE IN 3NF MIGHT CONTAIN MULTIVALUED DEPENDENCIES THAT PRODUCE EITHER NUMEROUS NULL VALUES OR REDUNDANT DATA

Summary

THE LARGER THE NUMBER OF TABLES, THE MORE ADDITIONAL I/O OPERATIONS AND PROCESSING LOGIC YOU NEED TO JOIN THEM

THE DATA-MODELING CHECKLIST PROVIDES A WAY FOR THE DESIGNER TO CHECK THAT THE ERD MEETS A SET OF MINIMUM REQUIREMENTS

References

Concepts of Database Management, 10th Edition | Lisa Friedrichsen, Lisa Ruffolo, Ellen Monk, Joy Starks, Philip Pratt, Mary Last | ISBN: 978- 0357422083 © 2020 | Publisher Course Technology – Cengage Learning

Database Concepts, 9th Edition | David M. Kroenke, David Auer, David J. Auer, Scott L. Vandenberg, Robert C. Yoder | ISBN: 978-0135188392 © 2020 | Publisher Pearson Education

Database System Concepts 7E, Abraham Silberschatz, Henry F. Korth, S. Sudarshan ©2020 McGraw Hill

DATABASE SYSTEMS Design, Implementation, and Management 13E, Carlos Coronel | Steven Morris, SBN-13: 978-1337627900 © 2018 Cengage Learning, Inc.

Microsoft SQL documentation - https://docs.microsoft.com/en-us/sql/?view=sql-server-ver15

Educational SQL resources - https://docs.microsoft.com/en-us/sql/sql-server/educational-sql-resources?view=sql-server-ver15

SQL Server Technical Documentation - https://docs.microsoft.com/en-us/sql/sql-server/?view=sql-server-ver15

SQL Developer Documentation - https://docs.oracle.com/cd/E12151_01/index.htm