Database Management

BU.330.770

Session 1

Instructor: Changmi Jung, Ph.D.

Session Objectives

- Understand why we use database
- Define database terms
- Identify the purpose of a database management system (DBMS)
- Explain and practice database design using **Entity-Relationship Models** and **normalization**

What is Database?

What is Database Management?

First, Why Databases?

- Characteristics of data in today's world
  - Ubiquitous (i.e., abundant, global, and everywhere)
  - Pervasive (i.e., unescapable, prevalent, and persistent)
- What kind of data did you generate yesterday?
- Databases make data persistent and shareable in a secure way
  - Specialized structures that allow computer-based systems to store, manage, and retrieve data very quickly

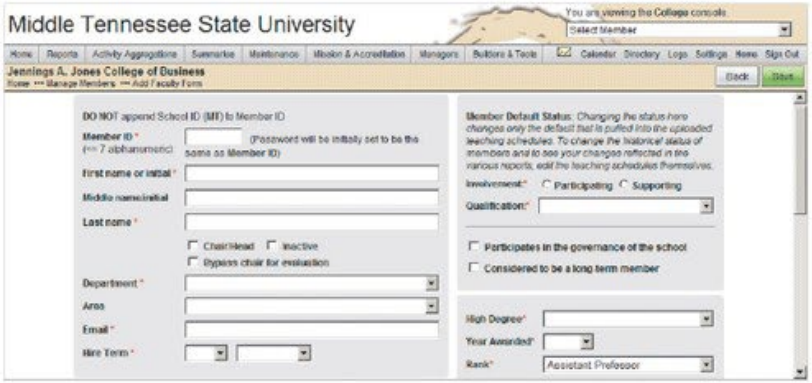
Data vs. Information vs. Knowledge

- Data consists of raw facts
  - Not yet processed to reveal meaning to the end user
  - Building blocks of information

- Information results from processing raw data to reveal meaning
  - Requires context
  - Bedrock of knowledge
  - Should be accurate, relevant, and timely

Data vs. Information vs. Knowledge

a) Data entry screen



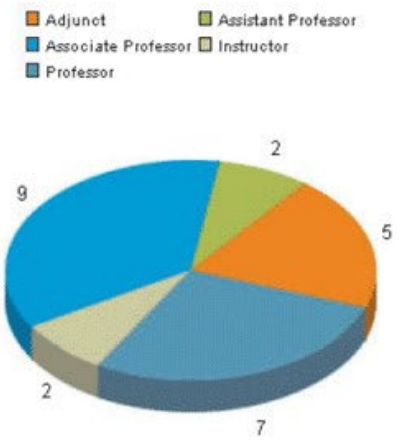
b) Raw data

Id	LastName	MidName	FirstName	DeptCode	Office	Email	Rank	HireYear	Degree
1	Washington	A	George	MGMT	N135	gwashington@mtsu.edu	Professor	2001	Ph.D.
2	Adams		John	FIN	N313	jadams@mtsu.edu	Professor	1984	Ph.D.
3	Jefferson	L	Thomas	ECON		tjefferson@mtsu.edu	Instructor	2002	M.B.A.
4	Madison	D	James	FIN	N236	jmadison@mtsu.edu	Associate Professor	1994	Ph.D.
5	Monroe	N	James	ACCT	N411	jmonroe@mtsu.edu	Assistant Professor	1995	Ph.D.
6	Adams	Q	John	ACCT	N418	jqadams@mtsu.edu	Associate Professor	1989	Ph.D.
7	Jackson	C	Andrew	ECON	N303	ajackson@mtsu.edu	Associate Professor	1999	Ph.D.
8	Van Buren	T	Martin	FIN	N306	mvanburen@mtsu.edu	Professor	1988	Ph.D.
9	Harrison	R	William	MKTG	N118	wharrison@mtsu.edu	Professor	1994	Ph.D.
10	Tyler	M	John	MGMT		jtyler@mtsu.edu	Assistant Professor	2000	Ed.D.
11	Polk		Cheryl	MKTG	N140	cpolk@mtsu.edu	Associate Professor	2002	Ph.D.
12	Taylor	G	Zachary	ACCT	N415	ztaylor@mtsu.edu	Associate Professor	1996	Ph.D.
13	Fillmore		Millard	JCB	N219	mfillmore@mtsu.edu	Professor	1992	Ph.D.
14	Pierce	A	Franklin	MKTG	N358	fpierce@mtsu.edu	Instructor	2005	M.B.A.
15	Buchanan	T	James	MGMT	N146	jbuchanan@mtsu.edu	Associate Professor	1986	D.B.A.
17	Lincoln	W	Lenty	MGMT	N150	tlincoln@mtsu.edu	Associate Professor	1996	Ph.D.
18	Johnson		Andrew	ISYS	N358	ajohnson@mtsu.edu	Professor	1987	Ph.D.
19	Grant		Kelly	MKTG	N120	kgrant@mtsu.edu	Assistant Professor	1988	D.B.A.
20	Rutherford		Hayes	ACCT	N408	hrutherford@mtsu.edu	Professor	1982	Ph.D.
21	Garfield	T	Denise	ACCT		dgarfield@mtsu.edu	Assistant Professor	2010	Ph.D.
22	Arthur		Emily	ACCT	N413	earthur@mtsu.edu	Associate Professor	2003	J.D.
23	Cleveland	G	Robert	ACCT	N401	rcleveland@mtsu.edu	Associate Professor	1997	Ph.D.
24	Harrison	X	Patricia	BULA	N406	pharrison@mtsu.edu	Associate Professor	2001	J.D.
25	McKinley	B	Priscilla	ISYS	N353	pmckinley@mtsu.edu	Adjunct	1994	M.S.
26	Roosevelt	F.	Hillary	MGMT	N104	hroosevelt@mtsu.edu	Associate Professor	2002	Ph.D.
27	Wilson		Laura	BCEN	N448	lwilson@mtsu.edu	Professor	1992	Ph.D.
28	Harding		Warren	MKTG	N114	wharding@mtsu.edu	Professor	1984	Ed.D.
29	Coolidge		Calvin	ECON	N316	ccoolidge@mtsu.edu	Professor	1975	Ph.D.
30	Hoover		Lou	MGMT		lhoover@mtsu.edu	Adjunct	1978	M.B.A.
31	Truman		Betty	ACCT	N416	btruman@mtsu.edu	Professor	1971	Ed.D.
32	Johnson		Robert	BCEN	N248	rjohnson@mtsu.edu	Professor	2001	Ph.D.

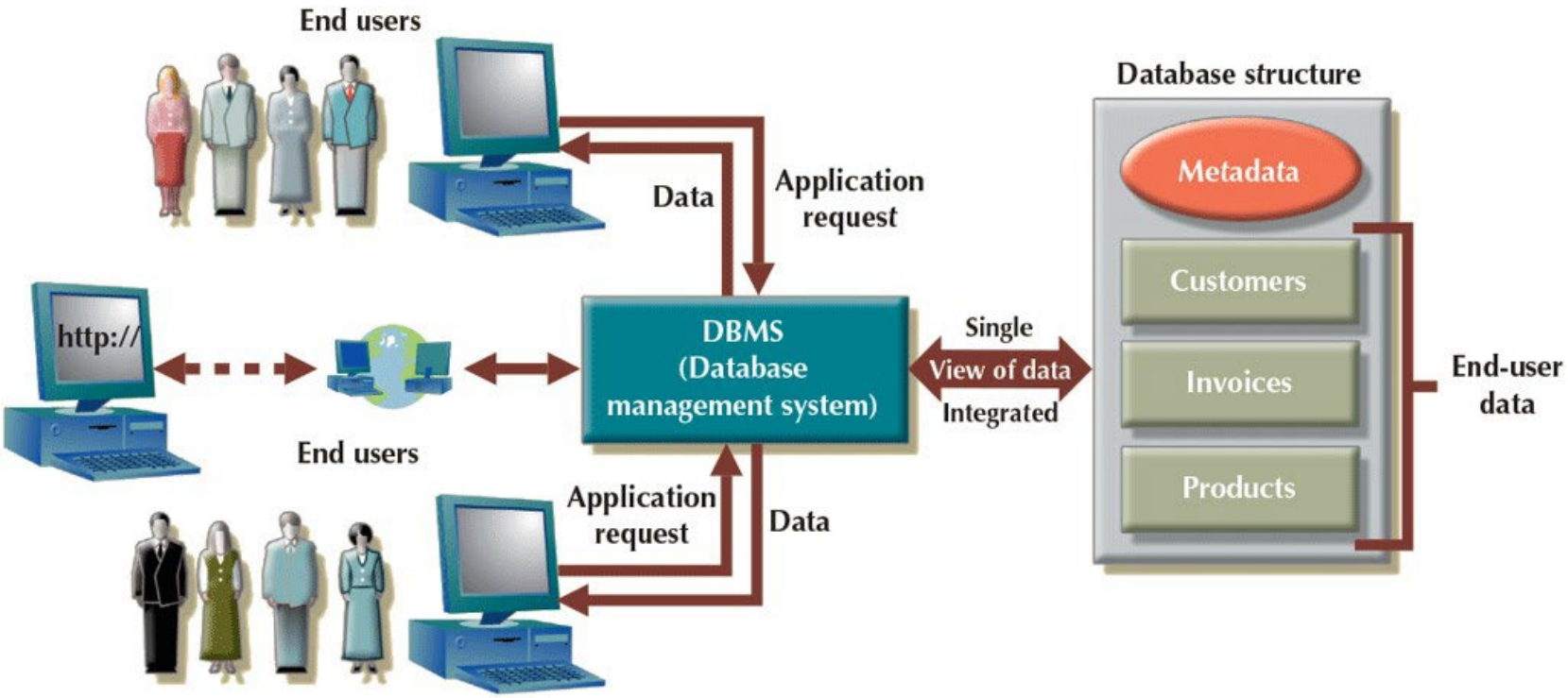
c) Information in summary format

Rank	COUNT	%/INFS	TOT/COL	%/COL. TOT.	%/COL. FAC.
Adjunct	5	20.00%	23	21.74%	3.27%
Assistant Professor	2	8.00%	28	7.14%	1.31%
Associate Professor	9	36.00%	37	24.32%	5.88%
Instructor	2	8.00%	18	11.11%	1.31%
Professor	7	28.00%	47	14.89%	4.58%

d) Information in graphical format

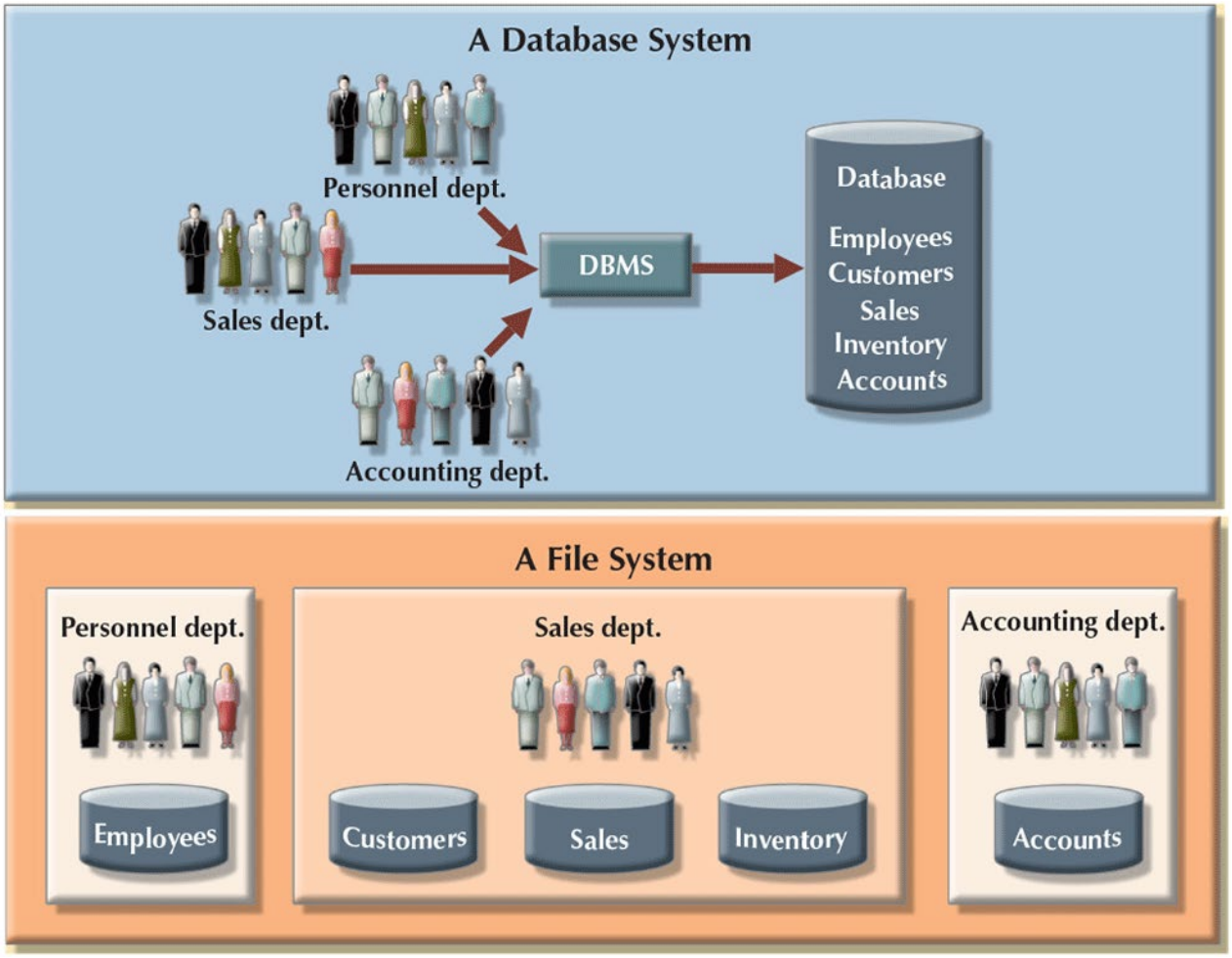


Database Management System (DBMS)



DBMS presents the end user (or application program) with a single, integrated view of the data in the database

Database vs. File system



Advantage of Using DBMS

- Improved data sharing
- Improved data security
- Better data integration
- Minimized data inconsistency
- Improved data access
- Improved decision making
- Increased end-user productivity

Database Management System Function

- *Data storage*: manage the physical structure of the database
- *Security*: control user access and privileges
- *Multiuser access*: manage concurrent data access
- *Backup*: enable recovery options for database failures
- *Data access language*: provide a language that allows database access
- *Data integrity*: enable constraints or checks on data
- *Data dictionary*: maintain information about database structure

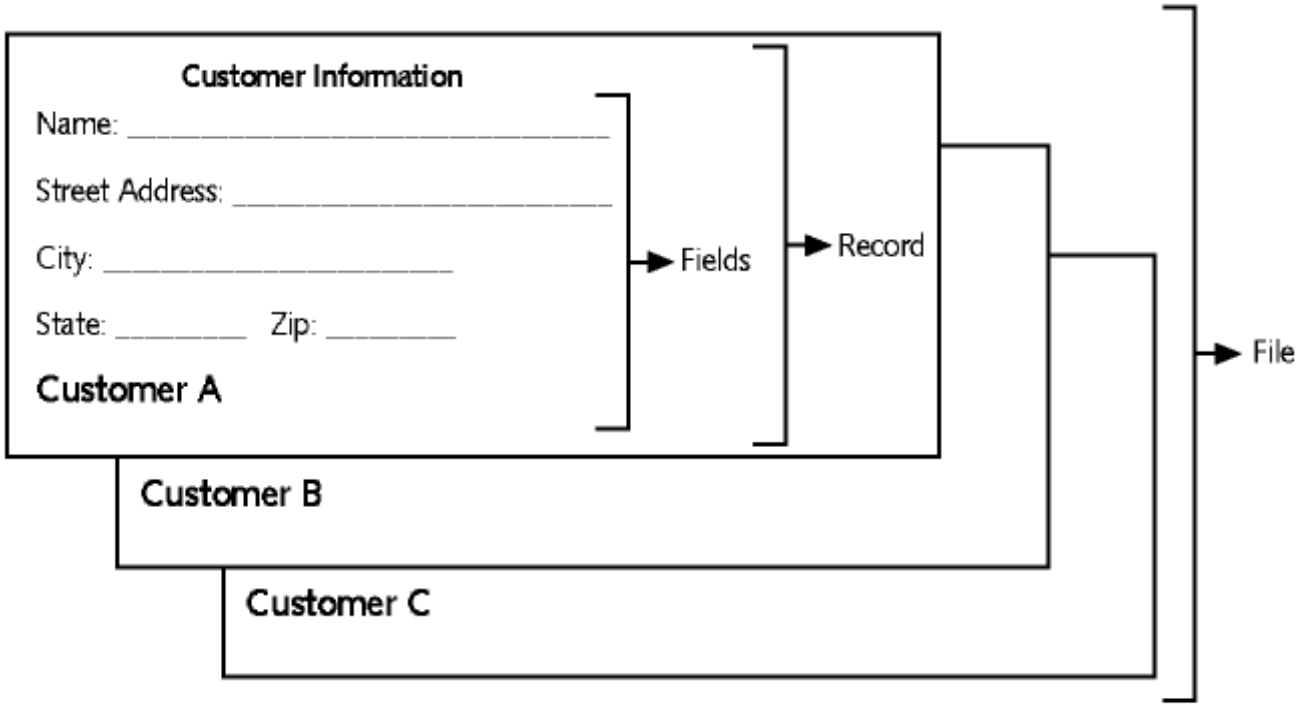
So… Database and DBMS is…

- Database is a logical structure to store data
- Database Management is an act/activity of creating, storing, organizing, handling of database
- Database Management System (DBMS) is a software used to create and interact with the database

Database Terminology

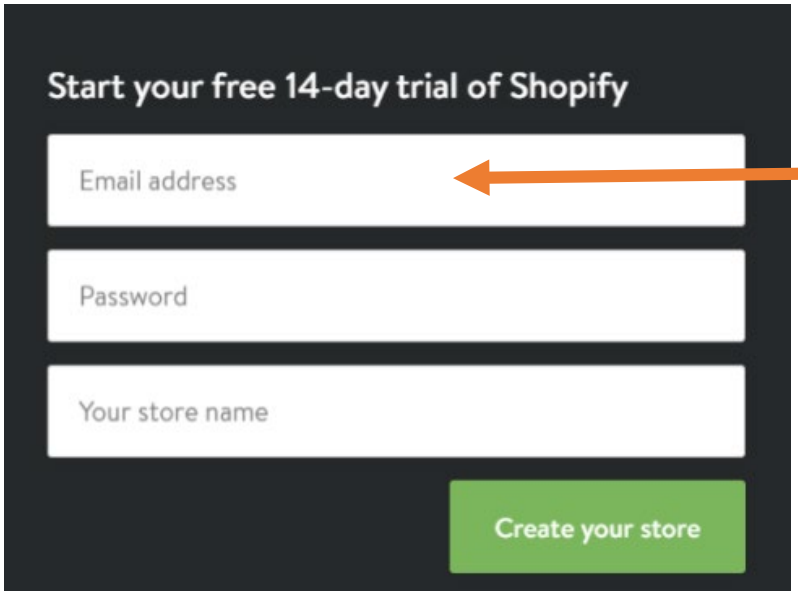
Database Components

- Character
- Field (attribute)
- Record (row)
- File (table)



Component – Character

- Basic unit of data
- Can be a letter, number, or special symbol



Charlie_Brown@Email.com

Image: https://instapage.com/blog/sign-up-page

Component – Field

- A group of related characters
- Represents an **attribute** or characteristic of an entity
- Corresponds to a **column** in the physical database

Customer ID	First name	Last name	Address	Sign-up date
000001	Changmi	Jung	1234 Carey Dr..	Dec-25-2020
000002	Harry	Potter	2345 Owl Ln…	Jan-10-2021
000003	Gummy	Bear	3456 Rainbow Ct…	Feb-01-2022
…	…	…	…	…
…	…	…	…	…
…	…	…	…	…

Component – Record

- A collection of fields for one specific entity
- Corresponds to a **row** in the physical database

Customer ID	First name	Last name	Address	Sign-up date
000001	Changmi	Jung	1234 Carey Dr..	Dec-25-2020
000002	Harry	Potter	2345 Owl Ln…	Jan-10-2021
000003	Gummy	Bear	3456 Rainbow Ct…	Feb-01-2022
…	…	…	…	…
…	…	…	…	…
…	…	…	…	…

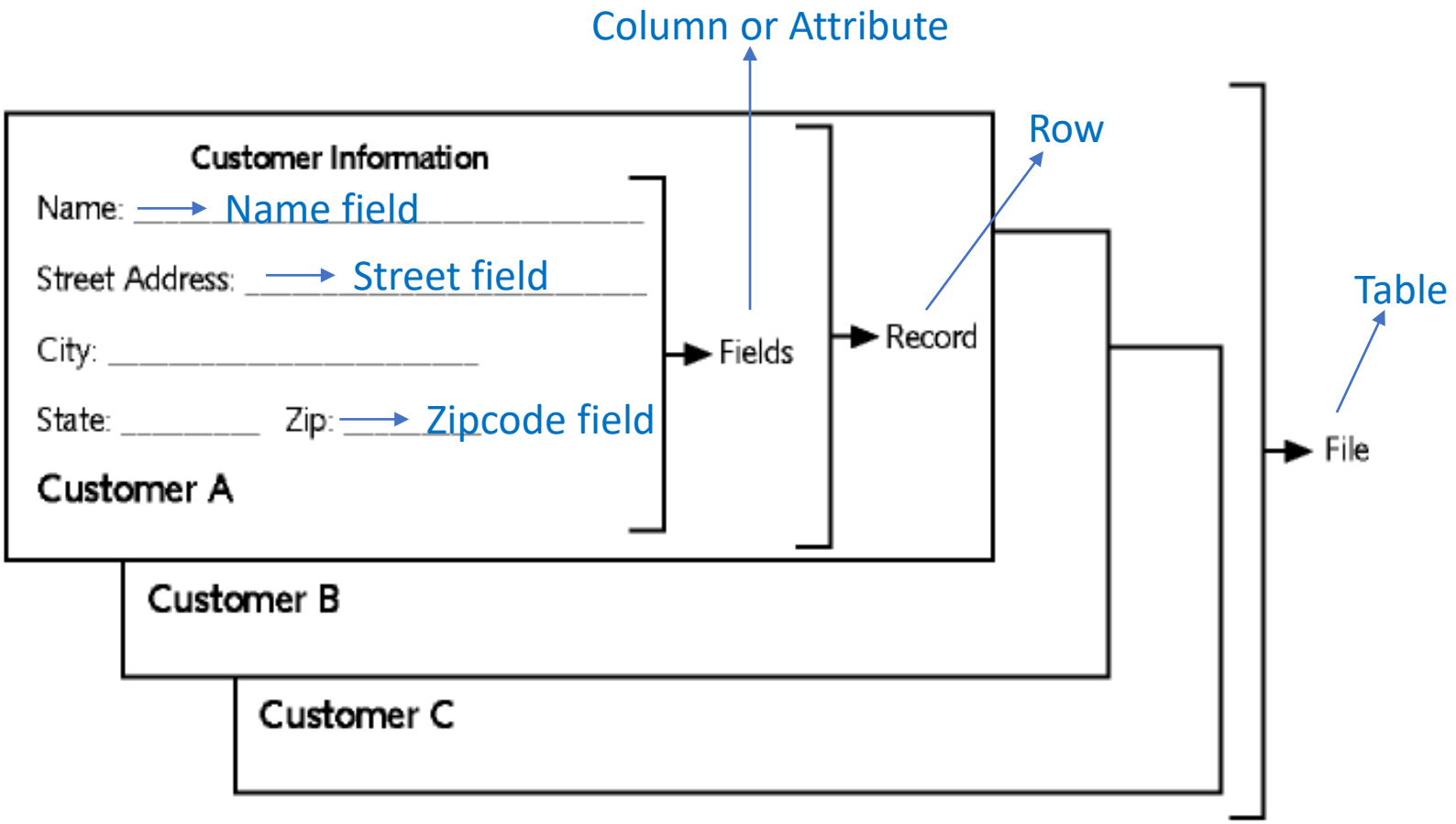
Component – File

- A group of records about the same type of entity (such as customer file or inventory file)
- We call it **table**

Customer File (Table)

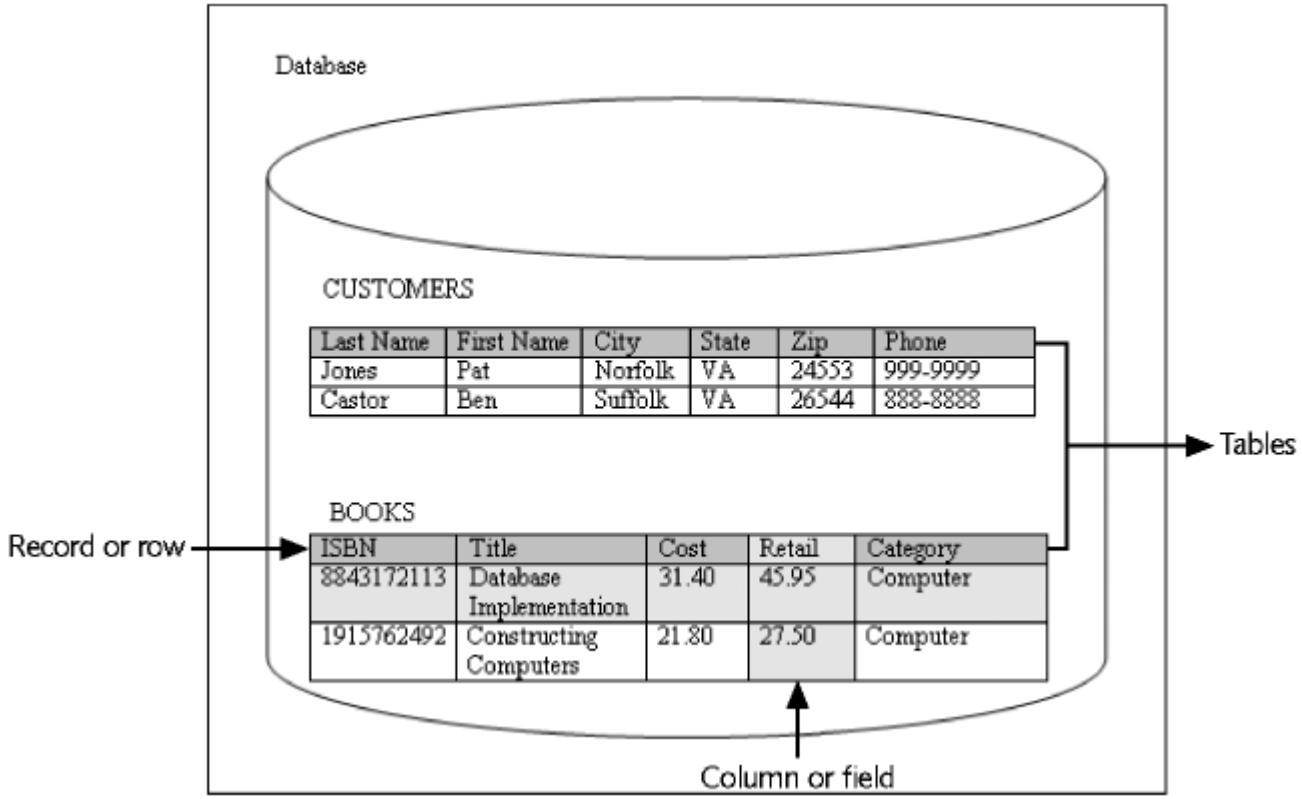
Customer ID	First name	Last name	Address	Sign-up date
000001	Changmi	Jung	1234 Carey Dr..	Dec-25-2020
000002	Harry	Potter	2345 Owl Ln…	Jan-10-2021
000003	Gummy	Bear	3456 Rainbow Ct…	Feb-01-2022
…	…	…	…	…
…	…	…	…	…
…	…	…	…	…

Example of Components



Example of Database

- Database: a Collection of **interrelated** files (tables)



Database Design

- Systems Development Life Cycle (SDLC)
- Entity-Relationship model (E-R model)
- Normalization

Systems Development Life Cycle (SDLC)

- Systems investigation – understanding the problem
- Systems analysis – understanding the solution
- Systems design – defining the logical and physical components
- Systems implementation – creating the system and placing completed system into operation
- Systems maintenance and review – evaluating the implemented system

Applying SDLC to Database

BabyBoba has 40 stores across 6 states in the US.
The company does not have any customer information yet, and they want to understand and use the information for a promotional campaign or mobile order later.

- Systems investigation – No idea about their customers, thus unable to analyze them to for a promotion
- Systems analysis – Need to create a table or database to collet customer and other relevant data
- Systems design – Define the table relations, identify necessary data attributes for each table (customer id, name, location, order, etc.)

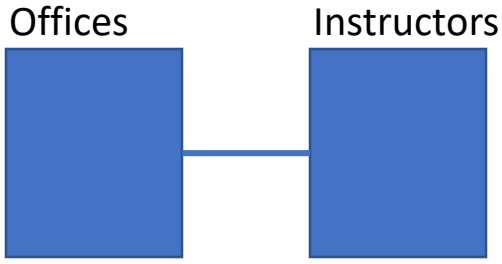
- Systems implementation – create 'Customer' table (and other tables), run with test data, deploy, and start collecting the data
- Systems maintenance and review – evaluate if data types/values are correct, if more data fields are needed, etc.

Entity-Relationship Model (E-R Model / ERD)

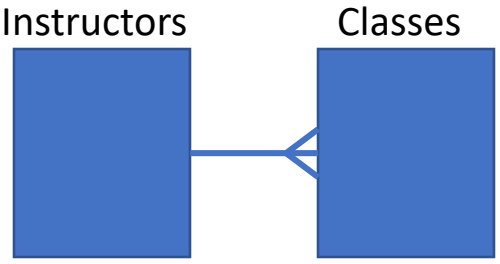
- Identifies the relationship among entities in the database
  - **Office 401** (Office table) is occupied by **Instructor John Wick** (Instructor table)

- The following relationships can be included in an E-R model:
  - One-to-one
  - One-to-many
  - Many-to-many

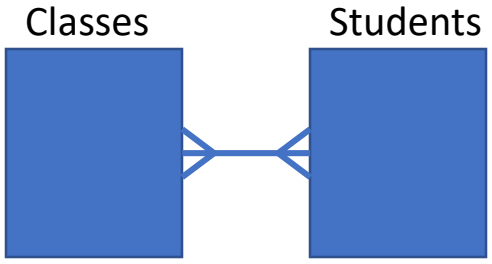
Wait, what is an entity?
Entity: what each row represents in a table
Ex. Offices, Instructors, Classes, and Students in the tables below



One-to-one



One-to-many

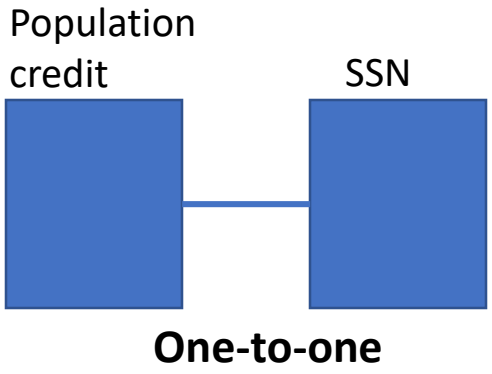


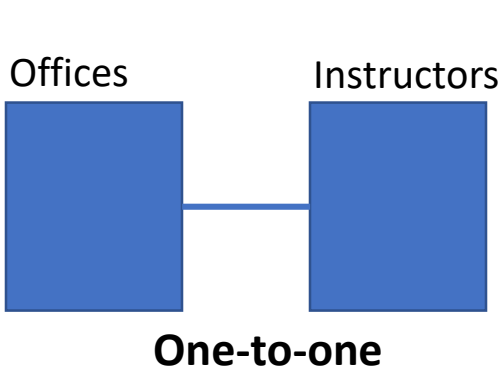
Many-to-many

One-to-One Relationship

- Each occurrence of data (record) in one table is represented by only one occurrence of data in the other table, and vice versa

- Example: Each individual has just one Social Security number (SSN), and each SSN is assigned to just one person



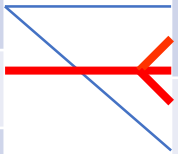


One-to-Many Relationship

- Each occurrence of data in one table can be represented by many occurrences of the data in the other table
- Example: A class has only one instructor, but each instructor can teach many classes

Instructor

First Name	Last Name	Office
Changmi	Jung	HE 1331
Minghong	Xu	HE 1321
Tony	Stark	DC 123



Class

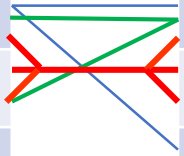
Class Code	Name	Classroom
330.770.51	Database Management	DC 101
330.780.51	Data Science and Business Intelligence	DC 102
330.999.51	AI and Jarvis	DC 103

Many-to-Many Relationship

- Multiple records in one table can relate to multiple records in another table.
- Example – Student course enrollment: A student can take many classes, and each class is composed of many students
- Need to be reduced to a set of One-to-Many relationships

Instructor

First Name	Last Name	Office
Changmi	Jung	HE 1331
Minghong	Xu	HE 1321
Tony	Stark	DC 123

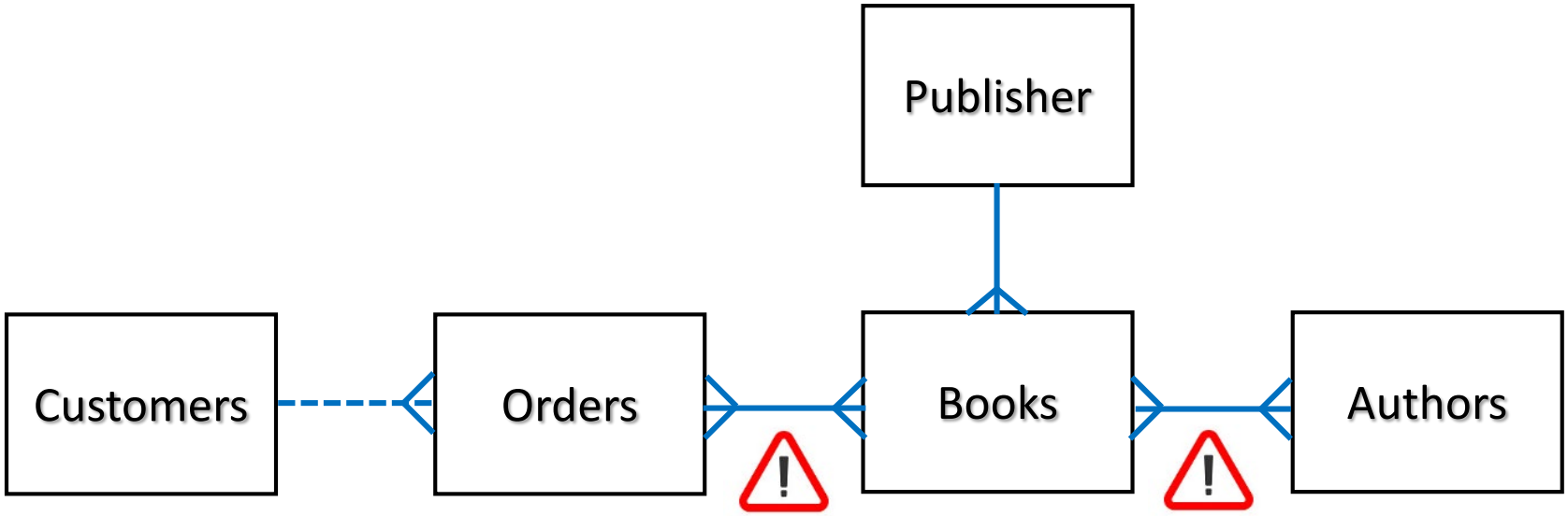


Students

Student ID	First Name	Last Name
A123456789	Peter	Parker
B123456709	Ron	Weasly
B123456789	Hermione	Granger

Simple E-R Model Example: 'JustLee' Books

Simple and Rudimentary ERD



Mandatory relationship: there must be a matched relation for all records

Optional relationship: some records may not have any matching records in the other table

Database Normalization

- Determines required tables and columns for each table
- Multistep process
- Used to reduce or control data redundancy and data anomalies

Database Redundancy and Anomalies

- Data redundancy – refers to having the same data in different places within a database
- Data anomalies – refers to data inconsistencies

TABLE 1-1 Single-Table Approach Example

Last Name	First Name	City	State	Zip	Order Date	Order #
Jones	Pat	Norfolk	VA	24553	3/22/2009	45720
Jones	Pat	Norfolk	VA	24553	5/28/2009	48243
Jones	Pat	Suffolk	VA	26544	9/05/2009	51932

Is the third Pat Jones a different customer…?

The same Pat Jones may have moved to Suffolk, VA…?

Someone may have incorrectly entered the city information…?

Unnormalized Data

- Contains repeating entries in the Author column in the BOOKS table
  - Repeating entries (group): multiple entries in a single column

TABLE 1-2 ISBN as the Primary Key

ISBN	Title	Publication Date	Cost	Retail	Category	Publisher	Contact	Author
8843172113	Database Implementation	04-JUN-03	31.40	55.95	Computer	American Publishing	Davidson	T. Peterson, J. Austin, J. Adams
1915762492	Handcranked Computers	21-JAN-05	21.80	25.00	Computer	American Publishing	Davidson	W. White, L. White

Natural key

Repeating entries

Then, what is a surrogate key or artificial key?

First-Normal Form (1NF)

- Primary key is identified
  - ISBN is the natural primary key in the previous example, but…
- Repeating entries are eliminated
  - Each attribute contains only atomic values of the domain
  - Ex. Attribute ‘Author’ of one record contains only one author’s name

First-Normal Form (1NF) (continued)

- ISBN and Author columns together create a **composite primary key**

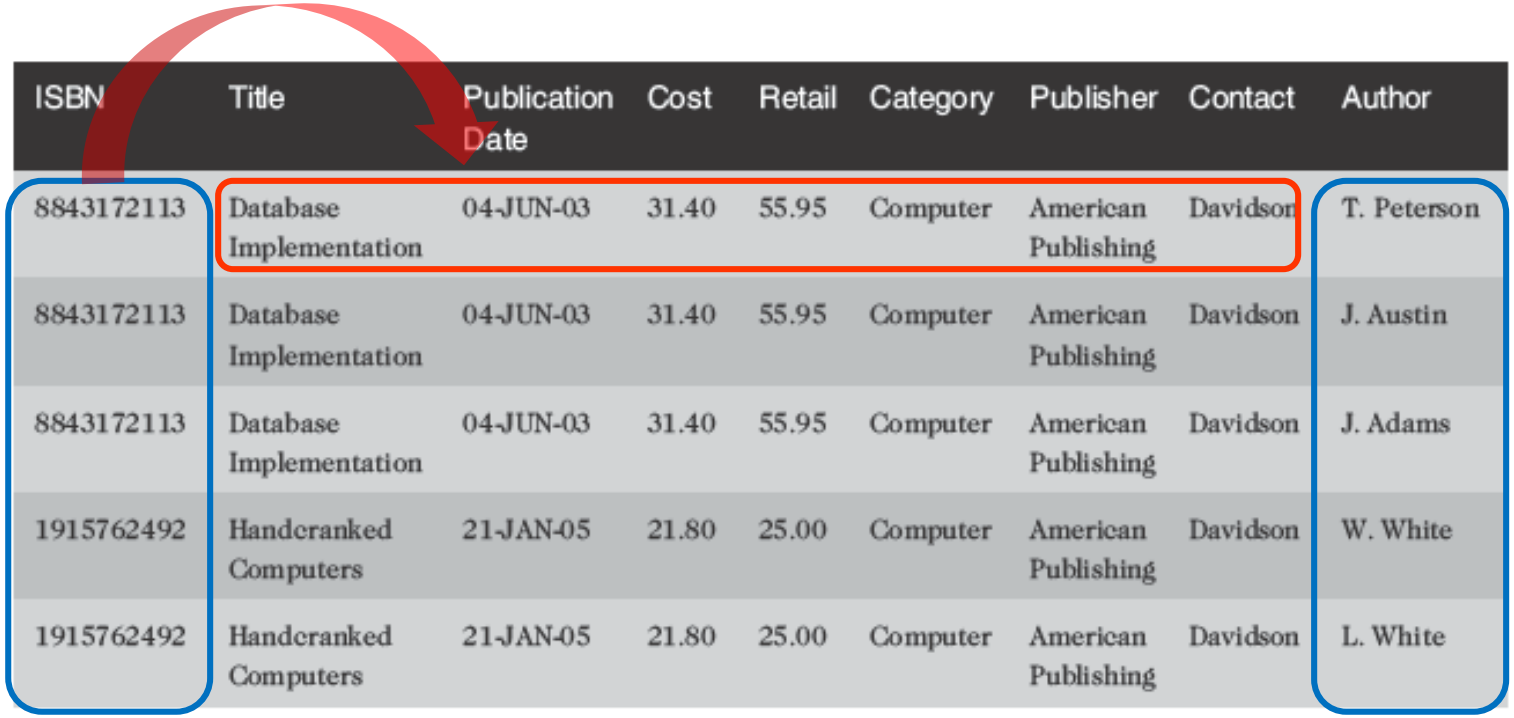
ISBN	Title	Publication Date	Cost	Retail	Category	Publisher	Contact	Author
8843172113	Database Implementation	04-JUN-03	31.40	55.95	Computer	American Publishing	Davidson	T. Peterson, J. Austin, J. Adams
1915762492	Handcranked Computers	21-JAN-05	21.80	25.00	Computer	American Publishing	Davidson	W. White, L. White

ISBN	Title	Publication Date	Cost	Retail	Category	Publisher	Contact	Author
8843172113	Database Implementation	04-JUN-03	31.40	55.95	Computer	American Publishing	Davidson	T. Peterson
8843172113	Database Implementation	04-JUN-03	31.40	55.95	Computer	American Publishing	Davidson	J. Austin
8843172113	Database Implementation	04-JUN-03	31.40	55.95	Computer	American Publishing	Davidson	J. Adams
1915762492	Handcranked Computers	21-JAN-05	21.80	25.00	Computer	American Publishing	Davidson	W. White
1915762492	Handcranked Computers	21-JAN-05	21.80	25.00	Computer	American Publishing	Davidson	L. White

No repeating entries!

Composite Primary Key

- More than one column is required to uniquely identify a row
- May lead to **partial dependency** – the fields in a record depend on only a portion of the composite primary key



ISBN	Title	Publication Date	Cost	Retail	Category	Publisher	Contact	Author
8843172113	Database Implementation	04-JUN-03	31.40	55.95	Computer	American Publishing	Davidson	T. Peterson
8843172113	Database Implementation	04-JUN-03	31.40	55.95	Computer	American Publishing	Davidson	J. Austin
8843172113	Database Implementation	04-JUN-03	31.40	55.95	Computer	American Publishing	Davidson	J. Adams
1915762492	Handcranked Computers	21-JAN-05	21.80	25.00	Computer	American Publishing	Davidson	W. White
1915762492	Handcranked Computers	21-JAN-05	21.80	25.00	Computer	American Publishing	Davidson	L. White

Second-Normal Form (2NF)

- Eliminate **Partial dependency**

  : Break the composite primary key into two parts, each part representing a separate table

ISBN	Title	Publication Date	Cost	Retail	Category	Publisher	Contact	Author
8843172113	Database Implementation	04-JUN-03	31.40	55.95	Computer	American Publishing	Davidson	T. Peterson
8843172113	Database Implementation	04-JUN-03	31.40	55.95	Computer	American Publishing	Davidson	J. Austin
8843172113	Database Implementation	04-JUN-03	31.40	55.95	Computer	American Publishing	Davidson	J. Adams
1915762492	Handcranked Computers	21-JAN-05	21.80	25.00	Computer	American Publishing	Davidson	W. White
1915762492	Handcranked Computers	21-JAN-05	21.80	25.00	Computer	American Publishing	Davidson	L. White

Second-Normal Form (2NF) (continued)

- ‘BOOKS’ table in 2NF

ISBN	Title	Publication Date	Cost	Retail	Category	Publisher	Contact
8843172113	Database Implementation	04-JUN-03	31.40	55.95	Computer	American Publishing	Davidson
1915762492	Handcranked Computers	21-JAN-05	21.80	25.00	Computer	American Publishing	Davidson

‘Author’ field removed

- Create a separate ‘Authors’ table

Author ID	Last name	First name	…
A00001	Peterson	Tim	…
A00002	Austin	James	…
A00003	Adams	Justin	…
…	…	…	…

Then, how do we connect BOOKS and AUTHORS?

- But, 2NF may contain transitive dependency

Third-Normal Form (3NF)

- Eliminate **Transitive Dependencies**
  - Transitive dependency: at least one field in the record is not dependent on the primary key but on another field in the record
- Create a separate table, PUBLISHER
- Publisher's contact name has been removed

ISBN	Title	Publication Date	Cost	Retail	Category	Publisher	Contact
8843172113	Database Implementation	04-JUN-03	31.40	55.95	Computer	American Publishing	Davidson
1915762492	Handcranked Computers	21-JAN-05	21.80	25.00	Computer	American Publishing	Davidson

PUBLISHER

Publisher ID	Publisher name	Contact	Address
00001	American Publishing	Davidson	.....
00002	Magical World	Severus Snape	.....

Summary of Normalization Steps

- 1NF: eliminate repeating entries, identify a primary key (or composite primary key)
- 2NF: table is in 1NF, and partial dependencies are eliminated
- 3NF: table is in 2NF, and transitive dependencies are eliminated

	1NF	2NF	3NF
No Repeating entries	Yes	Yes	Yes
Primary key identified	Yes	Yes	Yes
No Partial Dependency		Yes	Yes
No Transitive Dependency			Yes

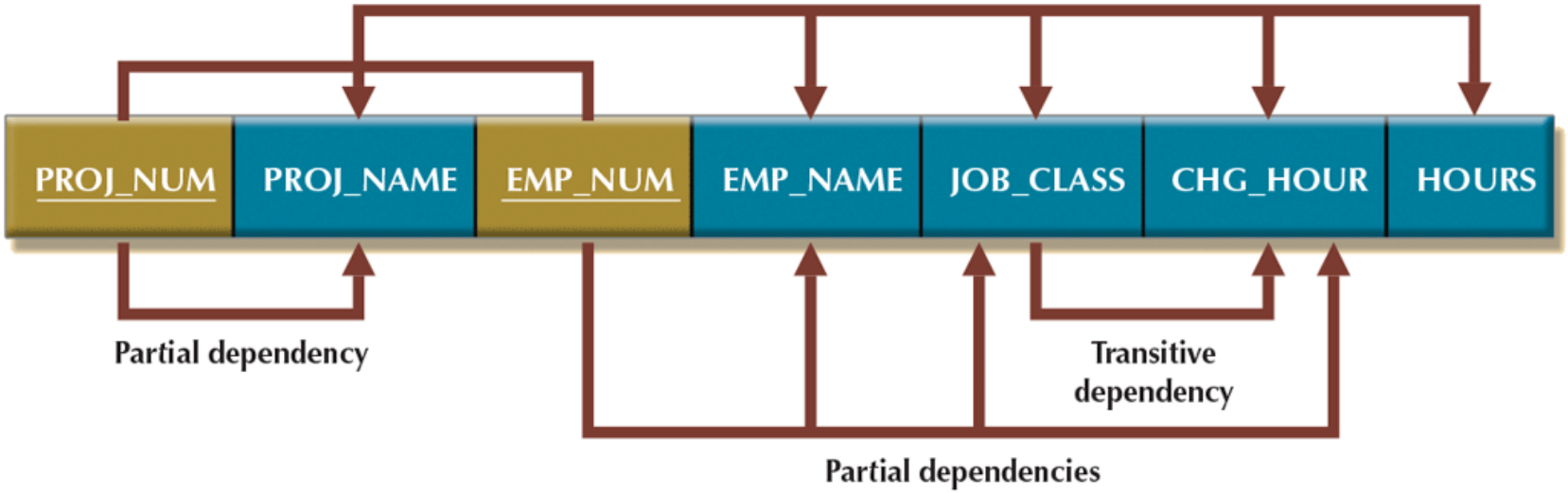
Practice Normalization

- Create 1NF of the Project_Resource table

Proj_Num	Project_Name	Emp_Num	Emp_Name	Job_Class	Chg_Hour	Hours
15	Evergreen	103, 101, 105, 106, 102	June E. Arbough, John G. News, Alice K. Johnson, William Smithfield, David H. Senior	Elec Engineer, Database Designer, Database Designer, Programmer, System Analyst	85.5, 105, 105, 35.75, 98.75	23.8, 19.4, 35.7, 12.6, 23.8
18	Amber Wave	114, 118, 104, 112	Annelise Jones, James J. Frommer, Anne K. Ramoras, Darlene M. Smithson	Application Designer, General Support, Systems Analyst, DSS Analyst	48.1, 18.36, 96.75, 45.95	25.6, 45.3, 32.4, 45
22	Rolling Tide	105, 104, 113, 111, 106	Alice K. Johnson, Anne K. Ramoras, Delbert K. Joenbrood, Geoff B. Wabash, William Smithfield	DB Designer, Systems Analyst, Applications Designer, Clerical Support, Programmer	105, 96.75, 48.1, 26.87, 35.75	65.7, 48.4, 23.6, 22, 12.8
25	Star Light	107, 115, 101, 114, 108, 118, 112	Maria D. Alonzo, Travis B. Bawangi, John G. News, Annelise Jones, Ralph B. Washington, James J. Frommer, Darlene M. Smithson	Programmer, Systems Analyst, Database Design, Applications Designer, Systems Analyst, General Support, DSS Analyst	35.75, 96.75, 105, 48.1, 96.75, 18.36, 45.95	25.6, 45.8, 56.3, 33.1, 23.6, 30.5, 41.4

Download file ‘Project_Resource.xlsx’ on Canvas > Week 1

Practice 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)

Let's eliminate the Partial Dependencies First!

Practice 2NF Result

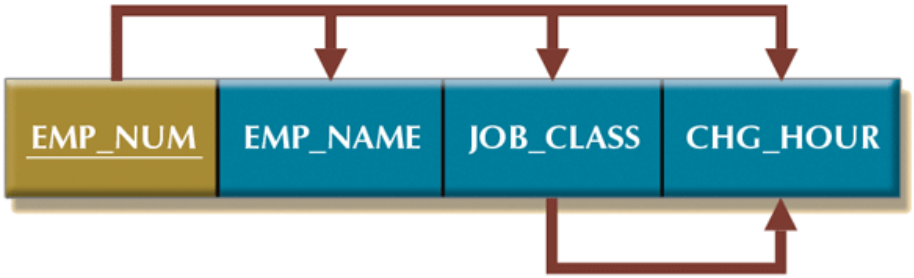
Table name: PROJECT

PROJECT (PROJ_NUM, PROJ_NAME)

Table name: EMPLOYEE

EMPLOYEE (EMP_NUM, EMP_NAME, JOB_CLASS, CHG_HOUR)

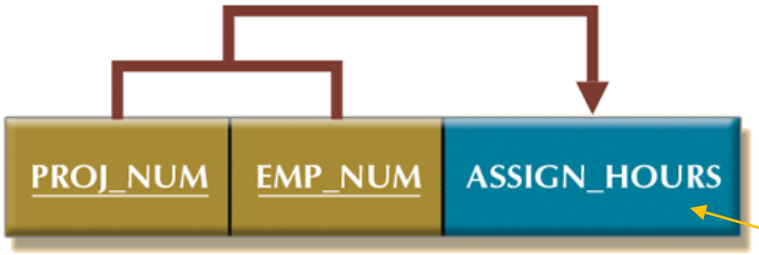
TRANSITIVE DEPENDENCY
(JOB_CLASS → CHG_HOUR)



Now, let's eliminate the Transitive Dependencies!

Table name: ASSIGNMENT

ASSIGNMENT (PROJ_NUM, EMP_NUM, ASSIGN_HOURS)



attribute 'Hours' in 1NF table

Practice 3NF Result

Table name: PROJECT

PROJECT (PROJ_NUM, PROJ_NAME)

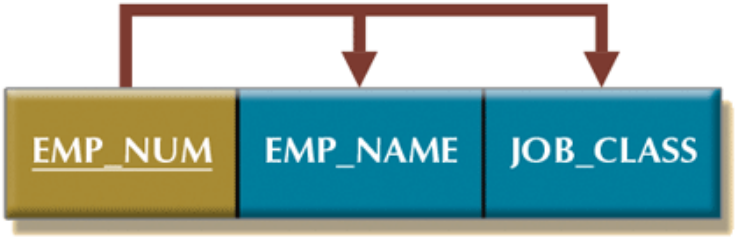


Table name: EMPLOYEE

EMPLOYEE (EMP_NUM, EMP_NAME, JOB_CLASS)

Table name: JOB

JOB (JOB_CLASS, CHG_HOUR)

Table name: ASSIGNMENT

ASSIGNMENT (PROJ_NUM, EMP_NUM, ASSIGN_HOURS)

Let's go back to 'JustLee Books' example

- After normalization, we will have multiple tables

BOOKS table

ISBN	Title	Publication date	Cost	Retail	Category	Pub ID
8843172113	Database Implementation	…				1
1915762492	Handcranked Computers	…				1

Publisher table

Pub ID	Name	Contact
1	American Publishing	Davidson
2	Cengage	…

Author table

Author ID	Last name	First name
00001	…..	…
00002	…..	…

Customer table

Customer ID	Last name	First name	City	State	Zip
00001	…..	…			
00002	…..	…			

Order table

Order#	Customer ID	Order date	Ship date	…	…
00001	…..	…			
00002	…..	…			

And some other tables

Now, how to relate them?

Relating Tables within the Database

- Once tables are normalized, ensure tables are linked
- Tables are linked through a **common field**
- A common field is usually a **primary key** in one table and a **foreign key** in the other table

FOREIGN KEY

PRIMARY KEY

BOOKS table

ISBN	Title	…	PubID
8843172113	Database Implementation	…	1
1915762492	Handcranked Computers	…	1

PUBLISHER table

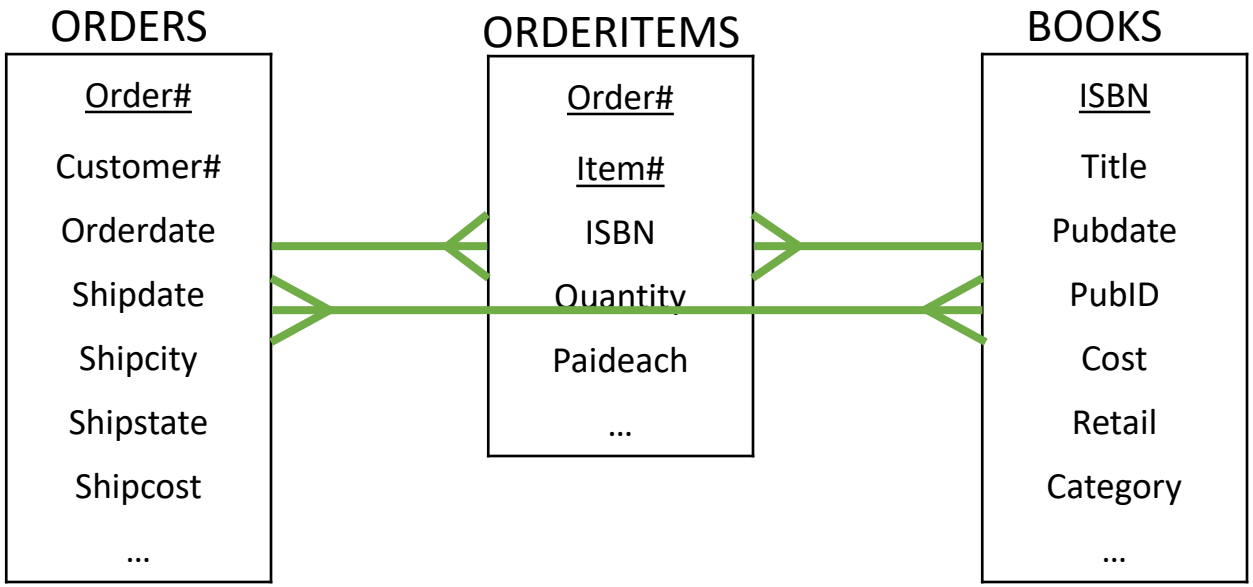
PubID	Name	Contact
1	American Publishing	Davidson
2	Cengage	…

The foreign key appears in the “many” side of a one-to-many relationship.

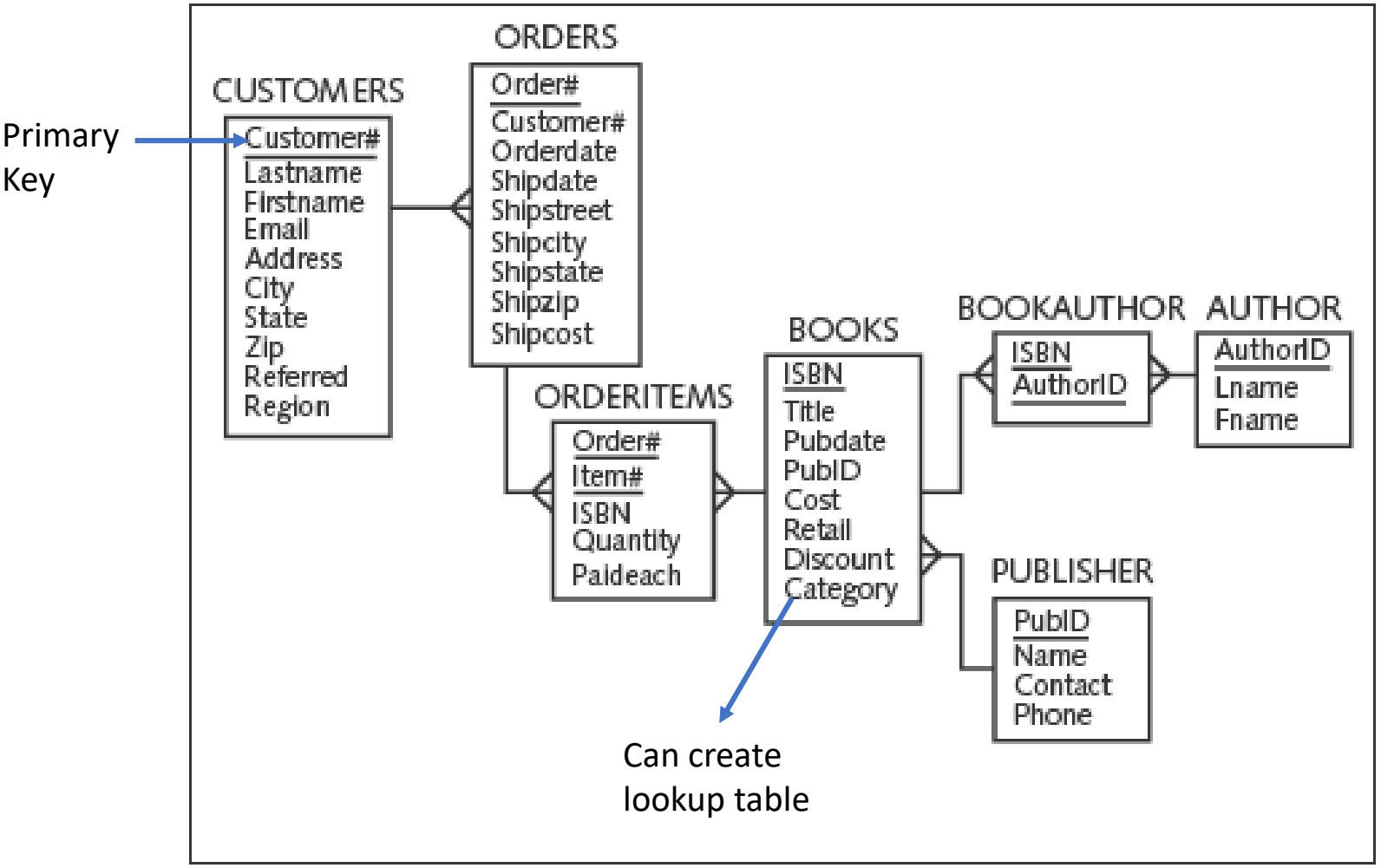
Bridging Entity

- A many-to-many relationship cannot exist in a relational database. The most common way to eliminate it is to create two 'one-to-many' relationships by adding a bridging entity.

- Placed between two entities with many-to-many relationship and serve as a filter for the data



After Normalization: JustLee Books ERD

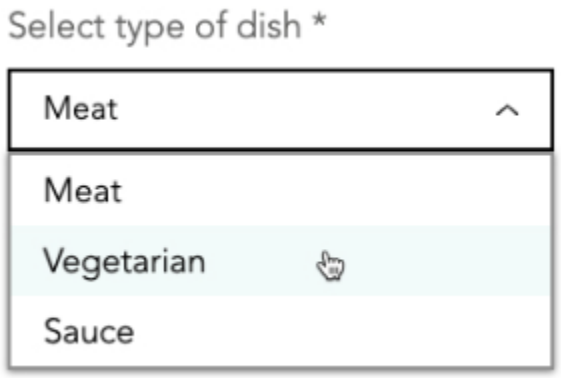


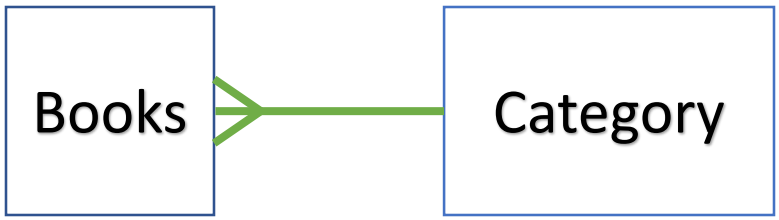
Lookup Table

- Common reference for descriptive data tables referenced in a foreign key

You may create Book Category Table

Category Code	Category Description
10	Computer
20	Cooking
30	Business
40	Family Literature





Assuming each book will be included in only one category.

Structured Query Language (SQL)

- Data sublanguage
  - A computer language that defines/manipulates the structure or content of DB in RDBMS
- Used to:
  - Create or modify tables
  - Add data to tables
  - Edit data in tables
  - Retrieve data from tables
- ANSI and ISO standards

Textbook Database – JustLee Books

- Assumptions
  - No back orders or partial shipments
  - Only U.S. addresses
  - Completed orders are transferred to the annual SALES table at the end of each month to enable faster processing on the ORDERS table

- Details of each table are found in Casteel page 14 – 16.

Summary (1/3)

- A DBMS is used to create and maintain a database
- A database is composed of a group of interrelated tables
- A file is a group of related records; a file is also called a table in the physical database
- A record is a group of related fields regarding one specific entity; a record is also called a row

Summary (2/3)

- A primary key is used to uniquely identify each record
- A record is considered unnormalized if it contains repeating entries
- A record is in first-normal form (1NF) if no repeating entries exist and it has a primary key
- Second-normal form (2NF) is achieved if the record is in 1NF and has no partial dependencies
- After a record is in 2NF and all transitive dependencies have been removed, then it is in third-normal form (3NF), which is generally sufficient for most databases

Summary (3/3)

- A common field is used to join data contained in different tables
- A foreign key is a common field that exists between two tables but is also a primary key in one of the tables
- A lookup table is a common term for a table referenced in a foreign key
- A Structured Query Language (SQL) is a data sublanguage that navigates the data stored within a database's tables

Next Week

- Starting SQL Exercise with SQL Developer
- **Use the link from Oracle's email to sign up for your Oracle Cloud Account**! If you're asked to provide your payment information, you did not use the provided link.
- Download and install **SQL Developer** before joining the class.
- We will create Oracle DB on the cloud and connect it with SQL Developer in the classroom
- **Quiz 1**: about 10 – 12 multiple-choice questions. More details will be posted on the Course Announcements.

Let’s Check Our Knowledge!