



London
Business
School

AM05 Data Management

01. Introduction to Data Management

Dr. David Tilson

London
Business
School

AM05: Data Management

David Tilson

Welcome!

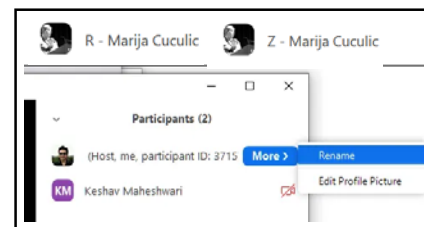
Class starts at 12:45 (London). Thank you for being early!

Zoom Classroom Etiquette

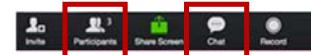
1. Roomies: Please **turn on** your cameras and **join without audio**
2. Rename yourself
 - a. Add "R" (for Roomie) in front of your name if you're in the LT
 - b. Add "Z" (for Zoomie) in front of your name if you're remote
3. Questions
 - a. Raise you (digital) hand if you want to speak or ask a question that needs an answer ASAP
 - b. Use chat to ask questions that can wait for a few minutes
 - c. I will ask you to answer questions using options on "participants" panel
 - d. Technical issues - message the facilitator privately in Zoom chat
4. If you are in a breakout room, engage with your colleagues to extract the most out the class
5. Session will be recorded



Roomies
When prompted, click "X"



 Raise Hand



Agenda

Introduction

Data in Organizations

Relational Database Management Systems (RDMS)





Data Modeling and Entity Relationship Diagrams (ERDs)

Hands on with SQL

Overview of rest of course

3

Brief Resume: David Tilson is a Info Systems professor with interests in analytics, consulting, as well as digital platforms and infrastructure

Education			
PhD	Information Systems	Case Western Reserve University	
MBA	Info Systems & Entrep.	University of Texas at Austin	
MSc	Telecommunications Engineering	University of London	
BEng (Hons)	Electrical & Electronic Engineering	Queen's University of Belfast	
Industry and Consulting Positions			
	McKinsey & Company (US)	Consultant	
	News Digital Systems (UK)	Sales support, Project Management	
	British Telecom Research (UK)	System design, Project Management	
Industry Sectors		Activities	
<ul style="list-style-type: none">• Telecommunications• Broadcast technology (British Equiv. of Emmy)• High tech / software• Finance• Insurance• Energy		<ul style="list-style-type: none">• Strategy Formulation• IT Function Restructuring• Outsourcing• Program / Project Management• System architect / design (mobile, satellite, broadcast)• Standards Creation• Academic Research	

Agenda

Introduction

Data in Organizations

Relational Database Management Systems (RDMS)

Data Modeling and Entity Relationship Diagrams (ERDs)

Hands on with SQL

Overview of rest of course

5

People have needed to **store and retrieve** information for a very long time... they have used a variety of artifacts to do so



Cuneiform from
4th millennium BC



Quipu from 3rd
millennium BC



Bill of sale for a donkey
on papyrus 126 AD



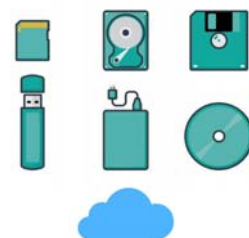
General Ledger 1828



Punched card from mid-20th century



~4GB of punched cards at US
Records Center 1959



Modern artifacts

Images from Wikipedia

6

Data usually requires **metadata** to become meaningful

Baker, Kenneth D.	324917628	MGT	2.9
Doyle, Joan E.	476193248	MKT	3.4
Finkle, Clive R.	548429344	PRM	2.8
Lewis, John C.	551742186	MGT	3.7
McFerran, Debra R.	409723145	IS	2.9
Sisneros, Michael	392416582	ACCT	3.3

Data: stored representations of meaningful objects and events

- Structured: numbers, text, dates
- Unstructured: images, video, documents

Name	ID	Major	GPA
Baker, Kenneth D.	324917628	MGT	2.9
Doyle, Joan E.	476193248	MKT	3.4
Finkle, Clive R.	548429344	PRM	2.8
Lewis, John C.	551742186	MGT	3.7
McFerran, Debra R.	409723145	IS	2.9
Sisneros, Michael	392416582	ACCT	3.3

Metadata: data that describes the properties and context of user data

Metadata pertains to underlying structure of data. When you later design a database, you are specifying its metadata. When you populate the database, you are putting data into it.

Data in its wider context helps us understand it better... it becomes more **informative**

Class Roster			
Course:	MGT 500 Business Policy	Semester:	Spring 2015
Section:	2		
Name	ID	Major	GPA
Baker, Kenneth D.	324917628	MGT	2.9
Doyle, Joan E.	476193248	MKT	3.4
Finkle, Clive R.	548429344	PRM	2.8
Lewis, John C.	551742186	MGT	3.7
McFerran, Debra R.	409723145	IS	2.9
Sisneros, Michael	392416582	ACCT	3.3

Class roster hints at several types of data entities

- Courses
- Sections (of these courses)
- Students (enrolled in a section)

We have gone from just raw data to some type of useful **information** by organizing the data

The concept of an **entity** is something we will discuss in more detail later

Metadata is data about data – a description of how data is to be stored and organized

TABLE 1-1 Example Metadata for Class Roster

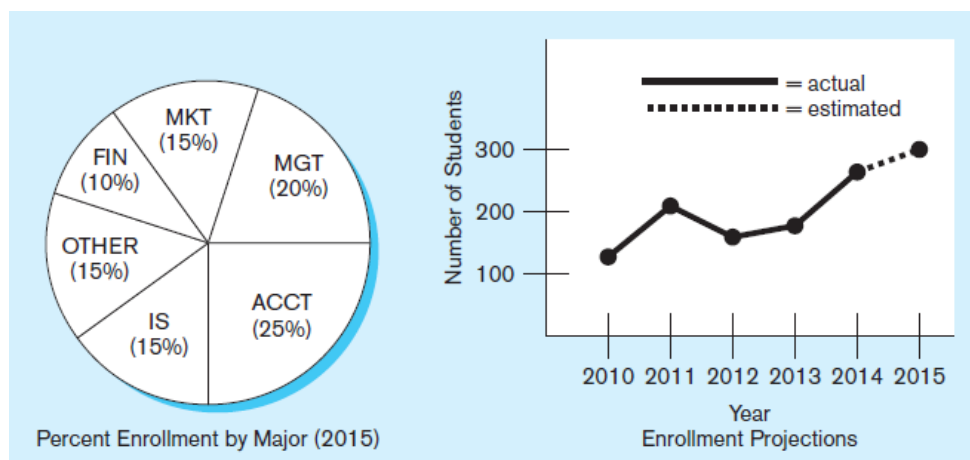
Data Item		Metadata				
Name	Type	Length	Min	Max	Description	Source
Course	Alphanumeric	30			Course ID and name	Academic Unit
Section	Integer	1	1	9	Section number	Registrar
Semester	Alphanumeric	10			Semester and year	Registrar
Name	Alphanumeric	30			Student name	Student IS
ID	Integer	9			Student ID (SSN)	Student IS
Major	Alphanumeric	4			Student major	Student IS
GPA	Decimal	3	0.0	4.0	Student grade point average	Academic Unit

Descriptions of the properties or characteristics of the data, including

- Data types
- Field sizes
- Allowable values
- Data context

9

Information is data processed to increase knowledge in the person using the data



Here we see summaries of the data (rather than individual data units)

- Data has been processed into aggregates (e.g. sums and averages) and categories
- It is one way of turning raw data into useful and actionable information
- Graphical representations are easier to absorb than tabular versions of the same information - aids managers' interpretation and decision making

10

Paper files and folders transitioned to digital equivalents



Optical storage



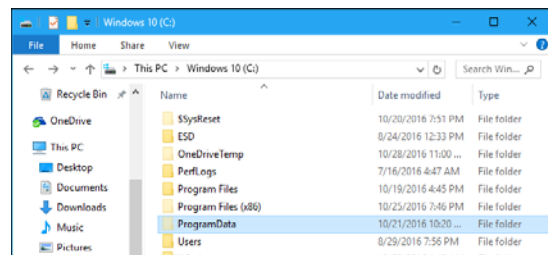
Modern Tape Cartridge



Hard Disk Drives (HDD)

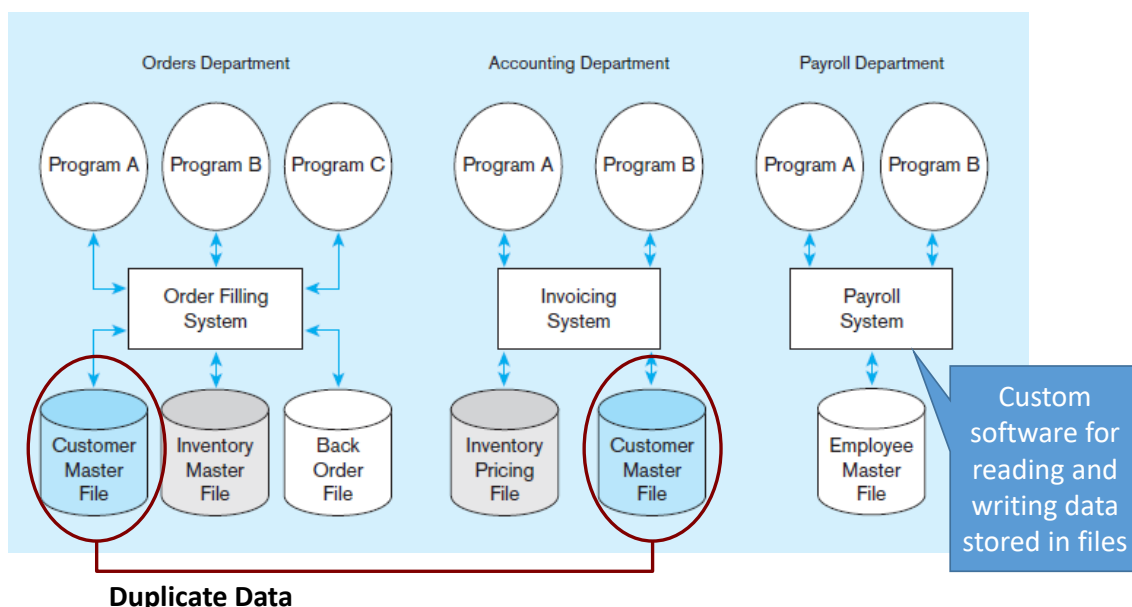


Solid State Drives (SSD)



11

Early automation in organizations used programs that managed their own data stored in files



Duplicate Data

Resulting inconsistencies one of the biggest problems in data management (e.g. customer address changed in only one file)

12

Systems based on file processing have several disadvantages

Disadvantages of File Processing

Program-Data Dependence

- All programs maintain data, metadata, and code for each file they use (tight coupling)
- Non-standard ways of storing data (e.g. file formats, data types)

Duplication of Data

- **Data can be inconsistent across files – compromises data integrity**
- May be very difficult to reconcile (different names and data types)
- Many systems/programs have separate copies of same data (waste)

Limited Data Sharing

- No centralized control of data
- Difficult to coordinate business processes across departments

Difficulties in Development

- Programmers must design their own file formats and write code for creating, reading, updating and deleting it
- Performing analysis on data from several systems is a major effort

Excessive Maintenance Needed

- Programs need to be rewritten for changes to the metadata or file formats
- Most of IT budget

Still an issue today? What about all those Excel files and your Python programs?

13

Agenda

Introduction

Data in Organizations

Relational Database Management Systems (RDMS)

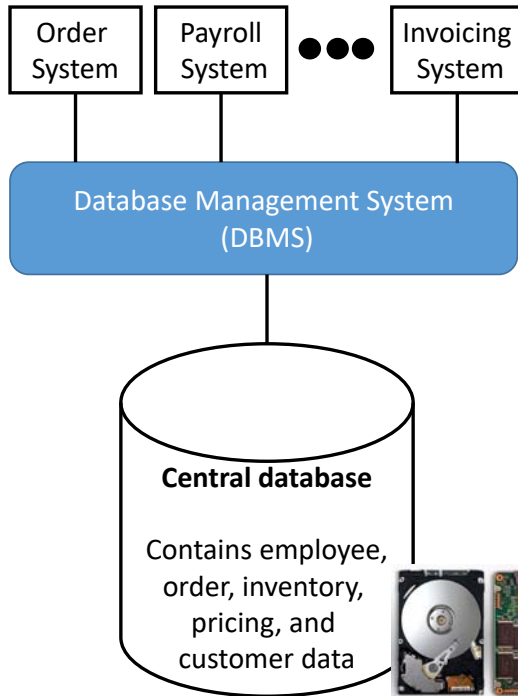
Data Modeling and Entity Relationship Diagrams (ERDs)

Hands on with SQL

Overview of rest of course

14

A Database is the solution to many of the problems raised



Business Applications . . . many can share a central consistent set of data that aids cross functional coordination

DBMS is software for creating, maintaining, and providing controlled access to user databases

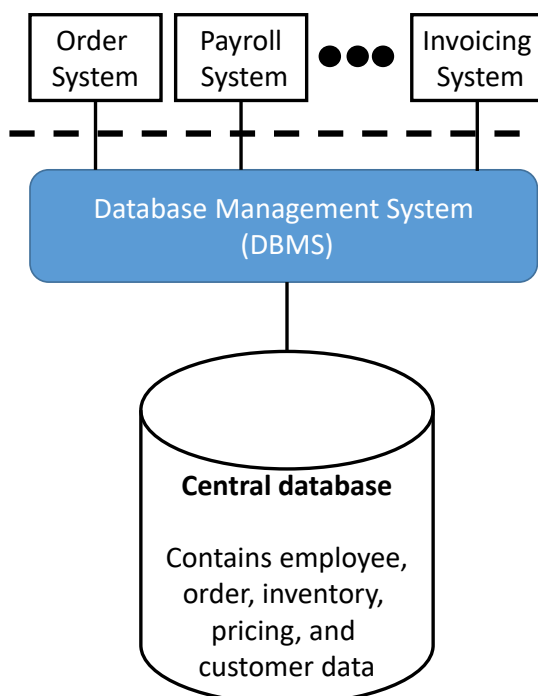
Database is defined as an *organized collection of logically related data*. It contains a model of something.

Data shared in central database

- Applications no longer need to maintain their own copy of the data
- Less duplication
- Increased data integrity

15

DBMS removes tight coupling between applications and data storage

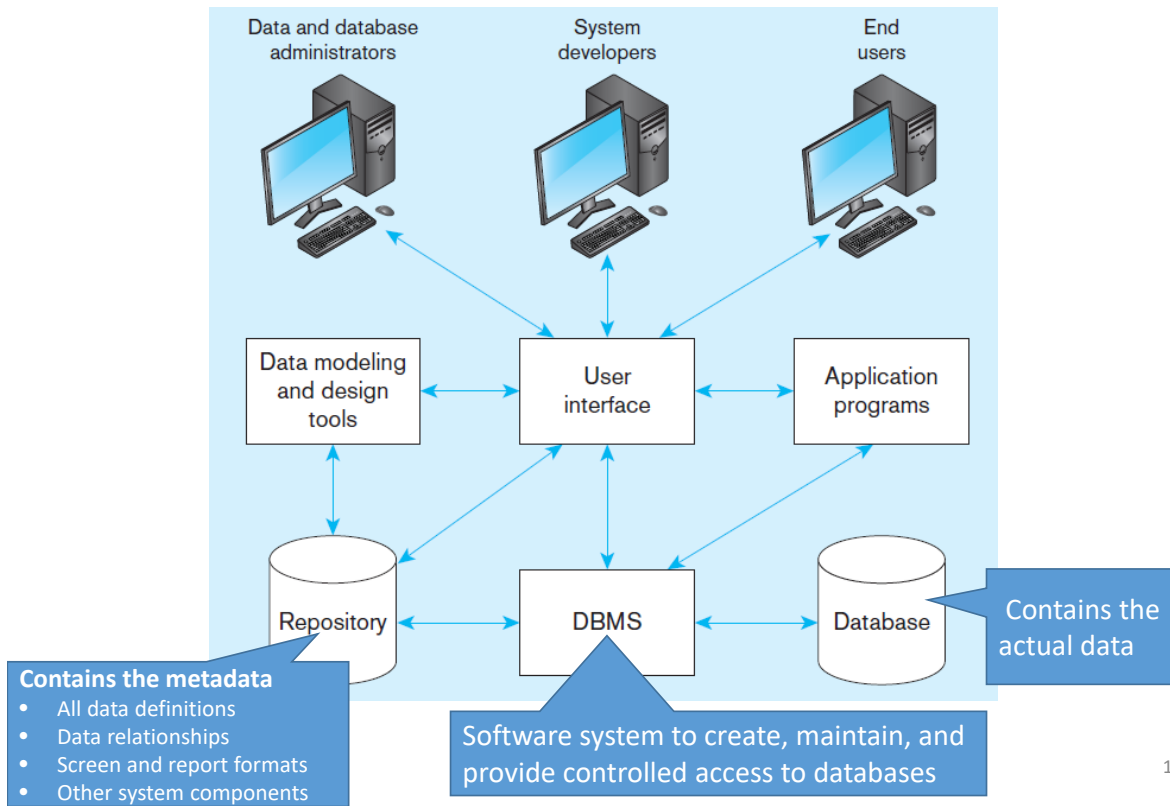


DBMS concept means that this key interface is standardized

- Details of data storage abstracted away as far as application developers are concerned
- Applications and data management can be implemented separately

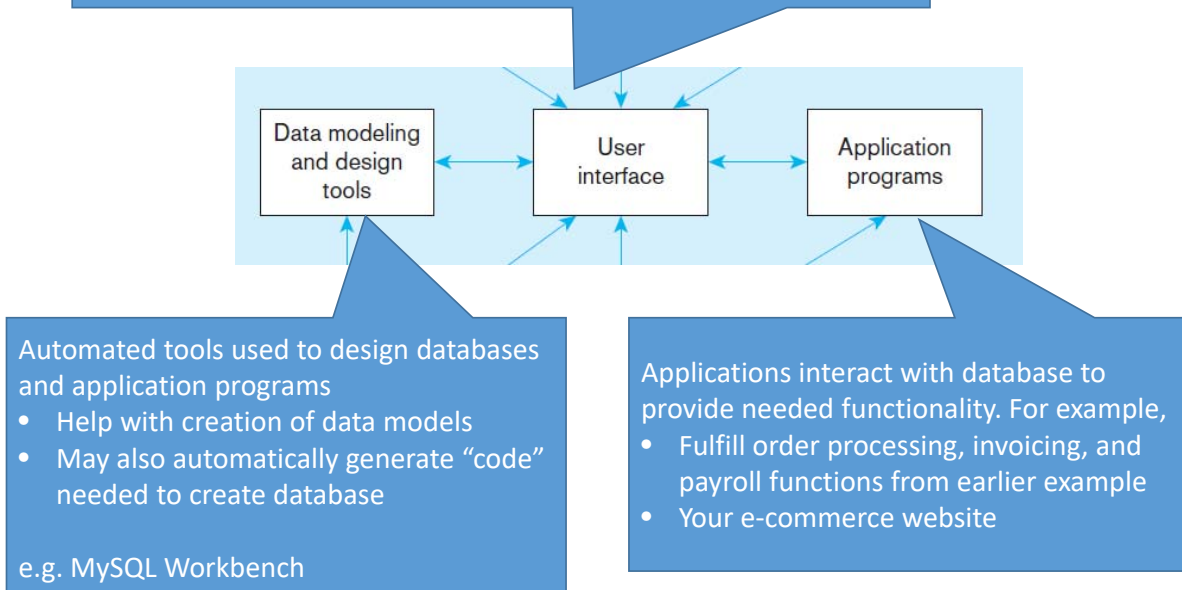
16

Typical database environments have several components

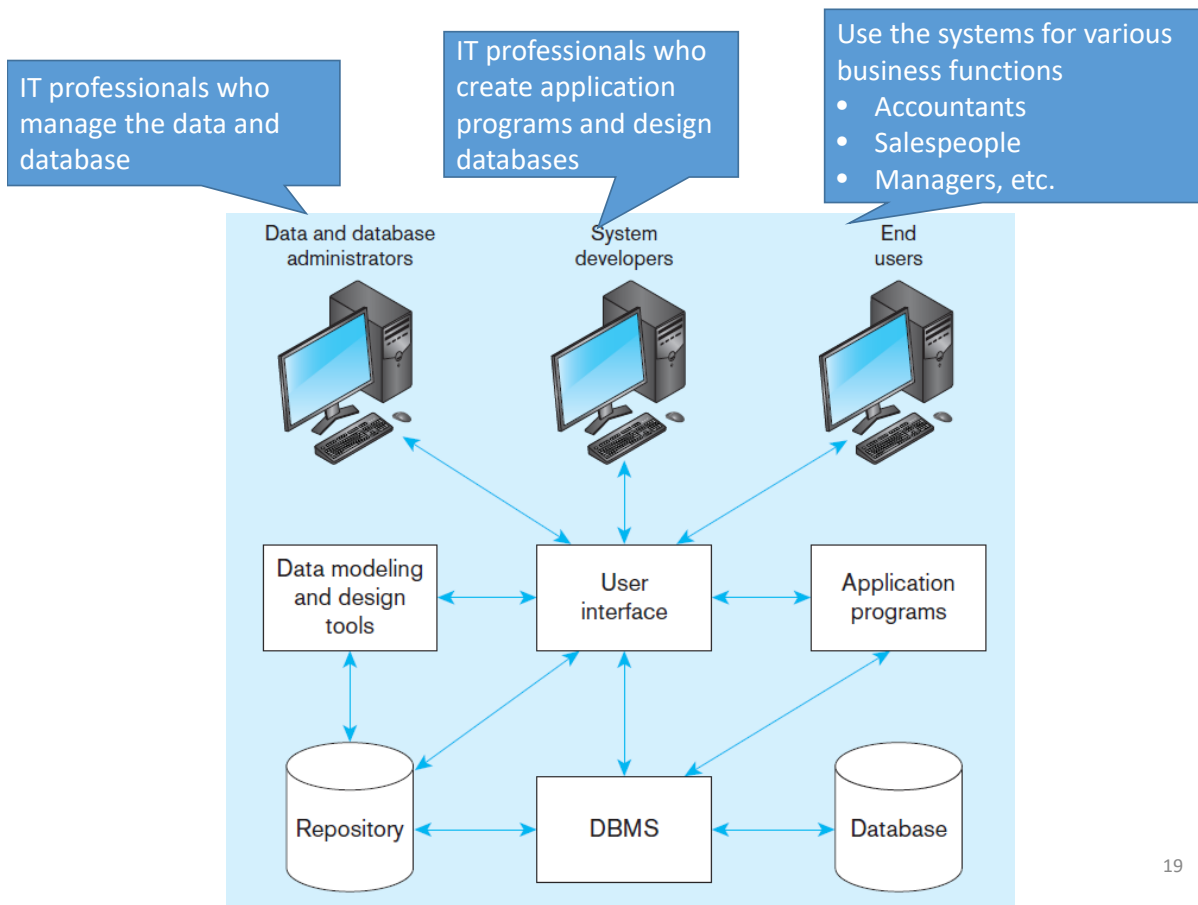


17

- Includes languages, menus, graphical displays and other facilities by which users interact with various system components
- Application programs used by people (users) involve user interfaces... others may run in background (e.g. print paychecks)



18



19

Database approach has many advantages

Advantages of Databases

Program-Data Independence

- Metadata separate from applications programs (looser coupling)
- Reduced maintenance effort since applications and database can both be modified with little impact on each other

Improved Decision Making

- Standard languages (e.g. SQL) allows IT to be more responsive and/or for users to run queries themselves
- Easier access to data analysis can improve business decisions

Less Data redundancy

- Can strive for no redundancy – each fact recorded in one place (customer address or attributes of product (may keep some for efficiency))
- **Improves consistency of stored data – improved data integrity**

More Data Sharing

- Databases can be shared resource that supports many users and applications
- Can facilitate coordinated business processes across departments (use same data)
- Requires enforcement of data standards and consistent business rules

Developers more productive

- Programmers can focus on getting business logic correct and adding value
- DBMS offer tools for reporting and database design
- Standardized APIs and data formats also help

Realizing the benefits takes planning and discipline

20

... but is not a silver bullet for all problems

¶ There are significant costs involved

- Database specialists
- On-going training
- Installation
- Maintenance of hardware and software
- Arranging appropriate backup and recovery processes
- Converting data from previous systems

¶ Change can lead to conflict in the organization

- Ownership of data\information may have conferred power
- Business processes and people's roles often change with the introduction of any new technology

¶ Creating high-level picture of all data across an organization (an Enterprise Data Model) is a large undertaking

21

Relational Databases Management Systems (RDBMS) are a big business and underpin most enterprise software



Major players in **Relational Database Management Systems (RDBMS)**... they are a large part of the software industry



ERP Systems used to coordinate finance and operations in most large organizations

Customer relationship management (CRM) is a technology for managing companies' relationships and interactions with (potential) customers.

RDBMS are at the heart of most important software infrastructures... there is a lot of data to keep track of

Agenda

Introduction

Data in Organization

Relational Database

Why is this important?

- Models/diagrams used in design process easier than working with code or pictures of tables
- Analysts and data scientists need to understand where data comes from
- Would like to design operational databases to make extracting data for later analysis (entities, naming, formats etc.) . . . with some data sources you will wish you had designed it
- Same toolkit (relational model and ERDs) used in design of data warehouses (e.g. star, constellation, and snowflake schemas, big data)

Data Modeling and Entity Relationship Diagrams (ERDs)

Hands on with SQL

Overview of rest of course

23

An **Enterprise-level** data model is a bird's eye view. It often does not have much detail (e.g. attributes)



In an Entity Relationship Diagram (ERD)

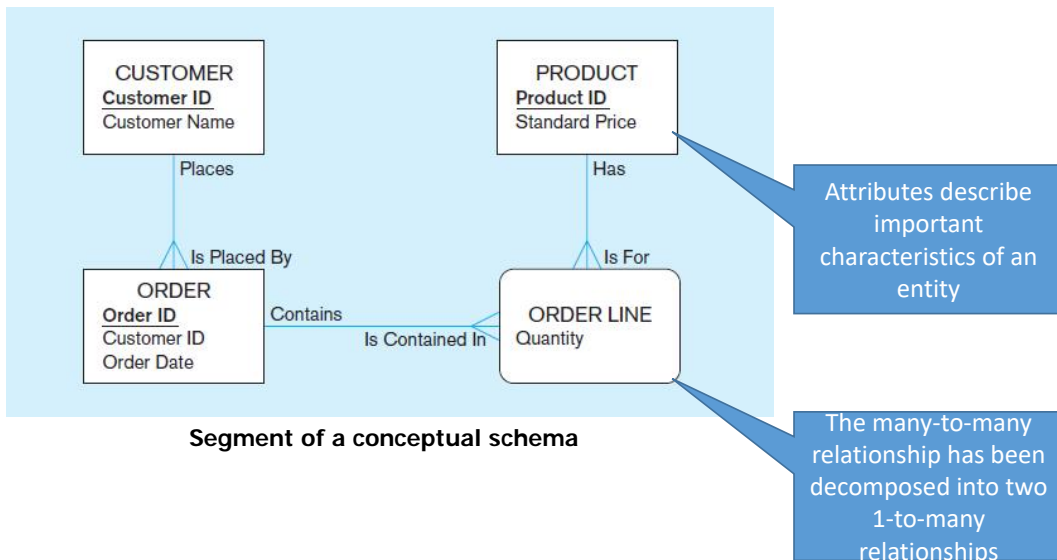
- Entities represented by boxes
- Relationships represented by lines between the boxes

Note: In these ER diagrams cardinality is represented using "crow's-feet" notation

Segment of an enterprise data model

24

Conceptual schema has more detail including attributes of the entities ... and more closely matches data structure in resulting database or data warehouse



25

An Entity is an object that

- will have many instances in the database
- will be composed of multiple attributes
- is often found in documented business rules

Common kinds of entity

Person: Employee, Student, Patient

Place: Store, Warehouse, State

Object: Machine, Building, Truck

Event: Sale, Registration, Renewal

Concept: Account, Course, Work Center

Naming guidelines

- ¶ Singular noun
- ¶ Specific to organization
- ¶ Concise (or abbreviation)
- ¶ For event entities, the result not the process
- ¶ Consistent for all diagrams

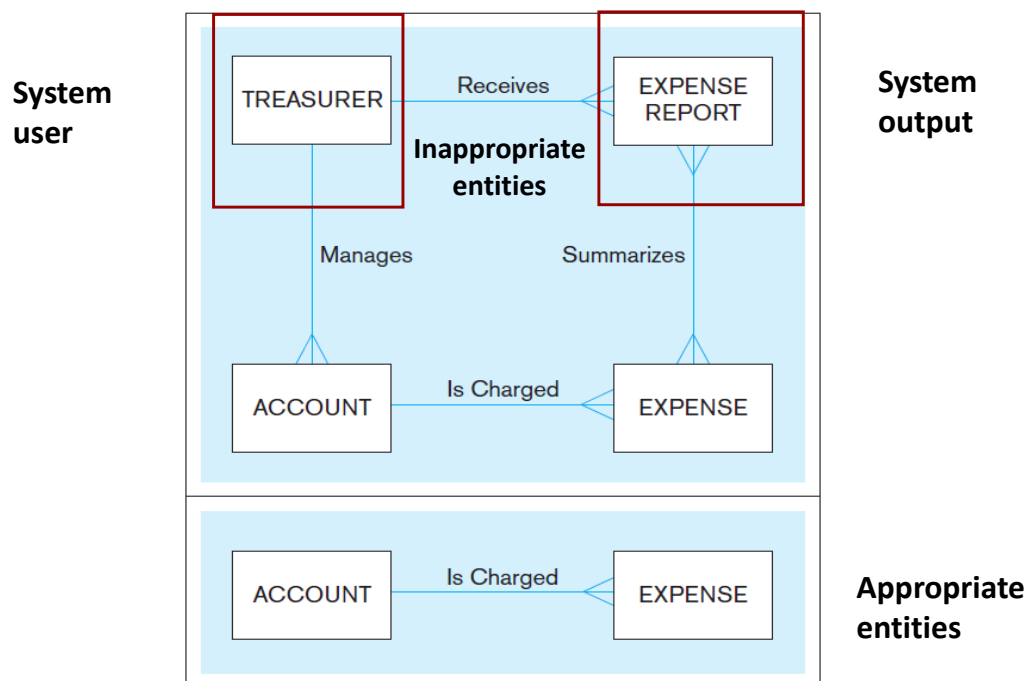
The EMPLOYEE type will have many instances in the database

Type	Instances		
Entity type: EMPLOYEE			
Attributes	Attribute Data Type	Example Instance	Example Instance
Employee Number	CHAR (10)	642-17-8360	534-10-1971
Name	CHAR (25)	Michelle Brady	David Johnson
Address	CHAR (30)	100 Pacific Avenue	450 Redwood Drive
City	CHAR (20)	San Francisco	Redwood City
State	CHAR (2)	CA	CA
Zip Code	CHAR (9)	98173	97142
Date Hired	DATE	03-21-1992	08-16-1994
Birth Date	DATE	06-19-1968	09-04-1975

Entities

27

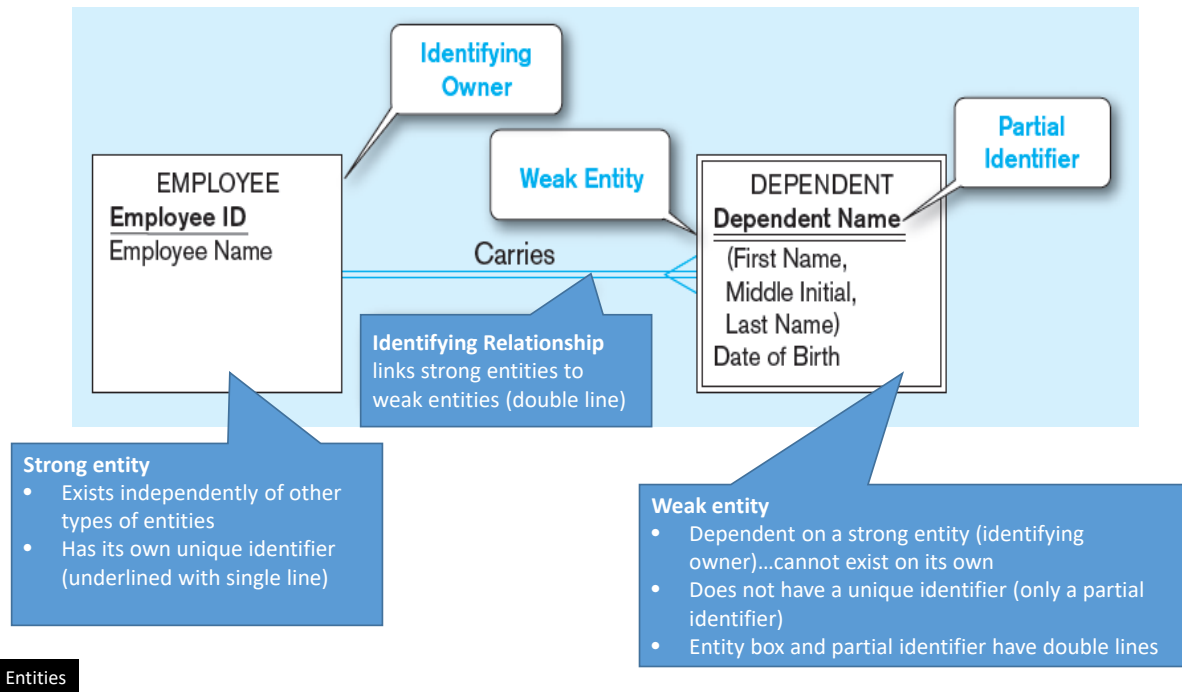
An entity should not be a user of the database system or an output from it (e.g., a report)



Entities

28

A **weak entity** type is one whose existence depends on some other entity type (aka **dependent entity** types)



An **Attribute** is a property or characteristic of an entity (or relationship type)

Defined by business rules

Entity type: STUDENT				
Attributes	Attribute Data Type	Required or Optional	Example Instance	Example Instance
Student ID	CHAR (10)	Required	876-24-8217	822-24-4456
Student Name	CHAR (40)	Required	Michael Grant	Melissa Kraft
Home Address	CHAR (30)	Required	314 Baker St.	1422 Heft Ave
Home City	CHAR (20)	Required	Centerville	Miami
Home State	CHAR (2)	Required	OH	FL
Home Zip Code	CHAR (9)	Required	45459	33321
Major	CHAR (3)	Optional	MIS	

Required – must have a value for every entity (or relationship) instance with which it is associated

Optional – may not have a value for every entity (or relationship) instance with which it is associated

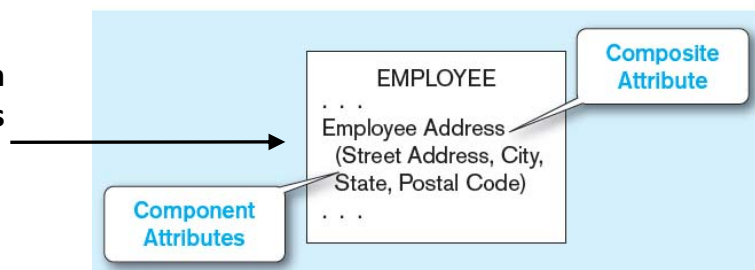
There are guidelines for naming attributes

Attribute name should

- Be a singular noun or noun phrase
- Be unique
- Follow standard format
 - e.g. [Entity type name { [Qualifier] }] Class
 - In this format **EmpBirthDt** and **EmpHireDt** for Employee birth and hire dates
 - Similar attributes of different entity types should use the same qualifiers and classes

Sometimes many attributes are related to each other (e.g. elements of an address). They can be grouped into a **composite attribute***

The address is broken
into component parts



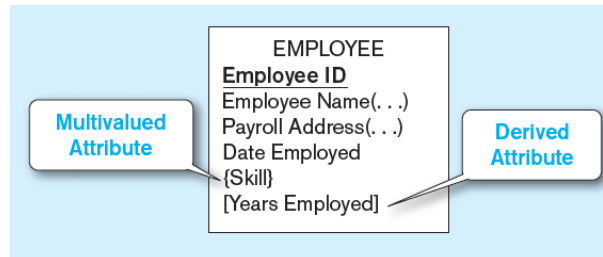
A **composite attribute** has meaningful component parts (attributes)

A **multivalued attribute** can have many different values while a **derived attribute** is calculated rather than being stored

Multivalued – may take on more than one value for a given entity (or relationship) instance

Derived – values can be calculated from related attribute values (not physically stored in the database)

For example, an employee can have more than one **skill**



Years Employed calculated from *Date Employed* and *Current Date*

Multi-valued and Derived Attributes

Note the use of different types of brackets

Every entity type should have an **identifier (or key)** attribute that **uniquely identifies** individual instances of an entity type

No two instances of the entity type may have the same value for the identifier attribute e.g. Student ID or other absolutely unique value

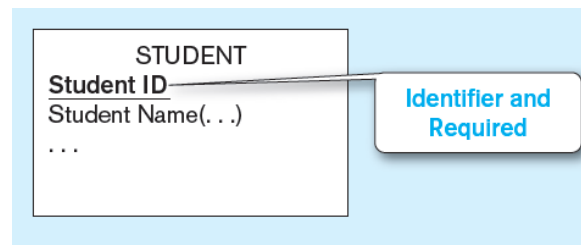
Do not rely on the first and last names as an identifier, since many people have the same name

Entity type: STUDENT			
Attributes	Attribute Data Type	Required or Optional	Example Instance
Student ID	CHAR (10)	Required	876-24-8217
Student Name	CHAR (40)	Required	Michael Grant
Home Address	CHAR (30)	Required	314 Baker St.
Home City	CHAR (20)	Required	Centerville
Home State	CHAR (2)	Required	OH
Home Zip Code	CHAR (9)	Required	45459
Major	CHAR (3)	Optional	MIS

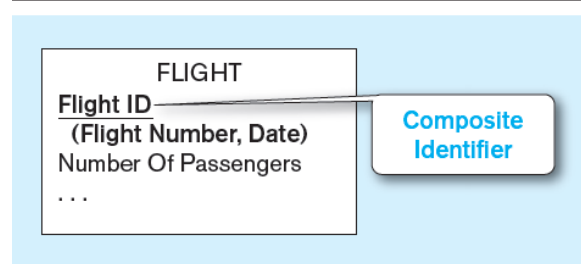
A **Candidate Identifier** is an attribute that could be an identifier...satisfies the requirements for being an identifier

Identifiers can be simple or composite

Simple identifier attribute



Composite identifier attribute



The identifier is boldfaced and underlined

An identifier in an ER model will eventually become a **primary key** in the resulting database table

Criteria for selecting identifiers

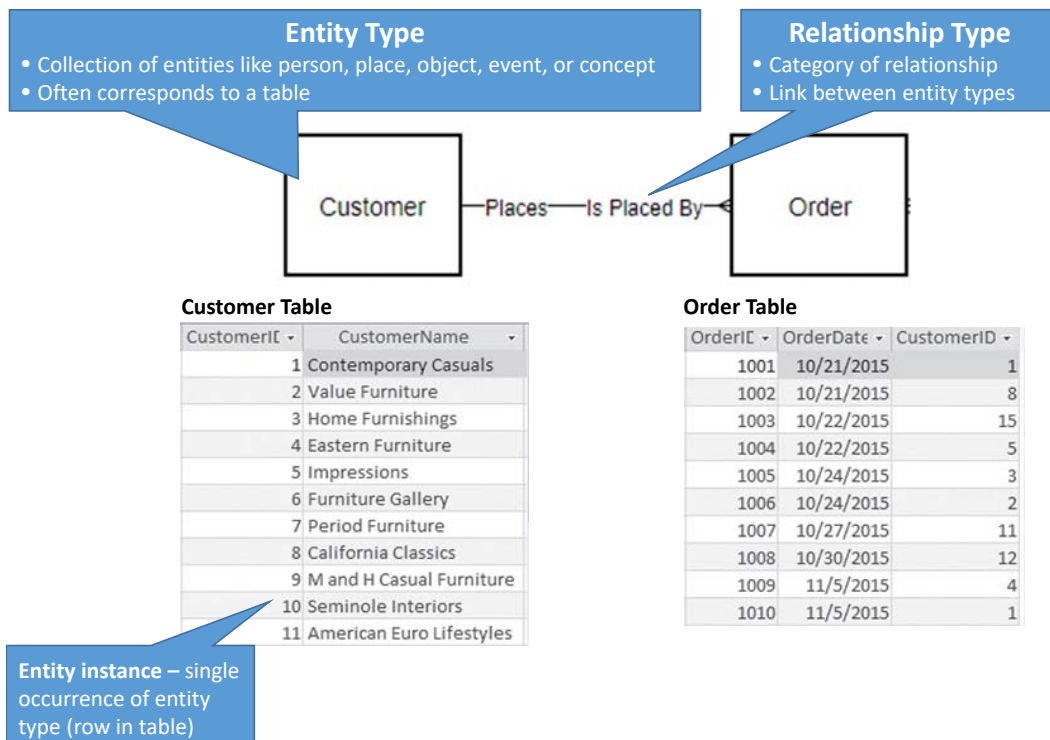
Choose Identifiers that

- Will not change in value
- Will remain unique in the future
- Will not be null

Also

- Avoid intelligent identifiers (containing names or addresses that might change and may not be unique)
- Bias towards substituting new simple keys for long composite ones

Relational Databases are a technology involving Tables (relations) representing entities and the relationships among those entities



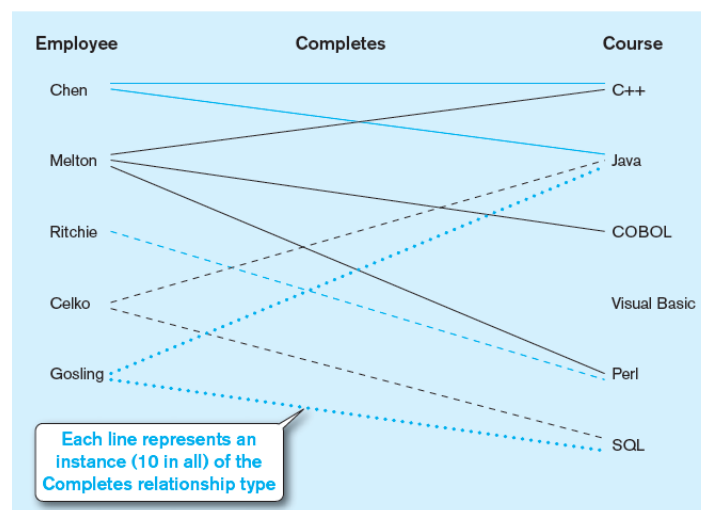
37

A relationship type is modeled as lines between entity types...the instance is between specific entity instances

Entity and Relationship types

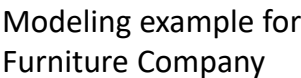


Entity and Relationship instances



38

Relationships



40

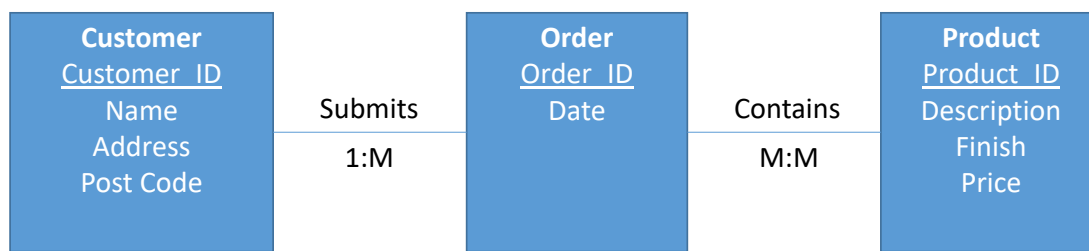
In-class exercise

15-minute
breakout

- ¶ Think of important information system in a real-world context e.g., university registrar, e-commerce, airline, hospital, electricity retail, ...
- ¶ Identify
 - A few of the most important entities that would need to be included (3 is enough)
 - Relationships among them (are they 1:1, 1:M or M:M?)
 - The most important attributes for the related entities including primary keys
- ¶ Capture your ideas in a picture like this one (drawn using PowerPoint) and submit to <https://forms.gle/2RdPXJ3peNgbri2U7> One submission per breakout team

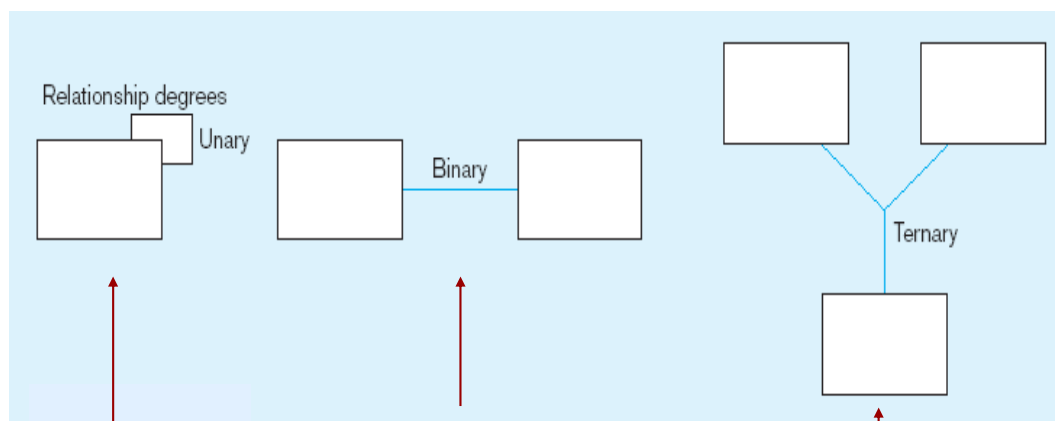
Context: Furniture Manufacturer

System: Order Management



41

The **Degree** of a relationship is the number of entity types that participate in it

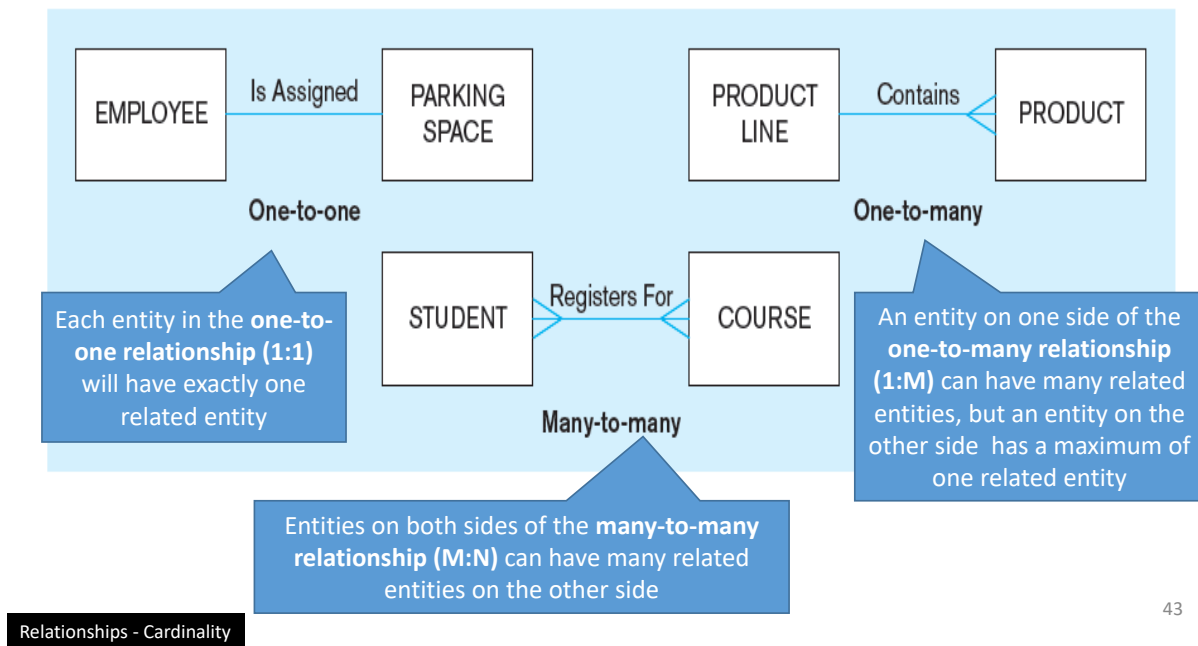


One entity
related to
another of the
same type

Entities of two
different types
related to each
other

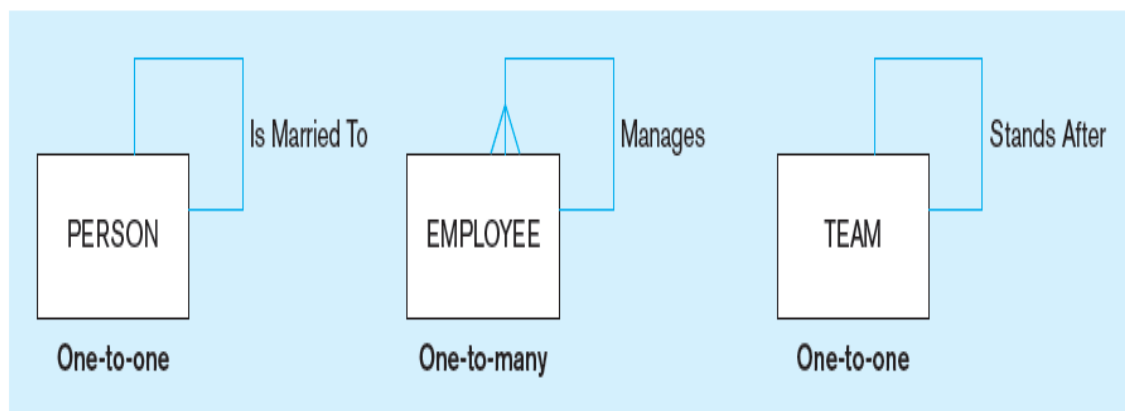
Entities of three
different types
related to each
other

Cardinality Constraints—the number of instances of one entity that can or must be associated with each instance of another entity

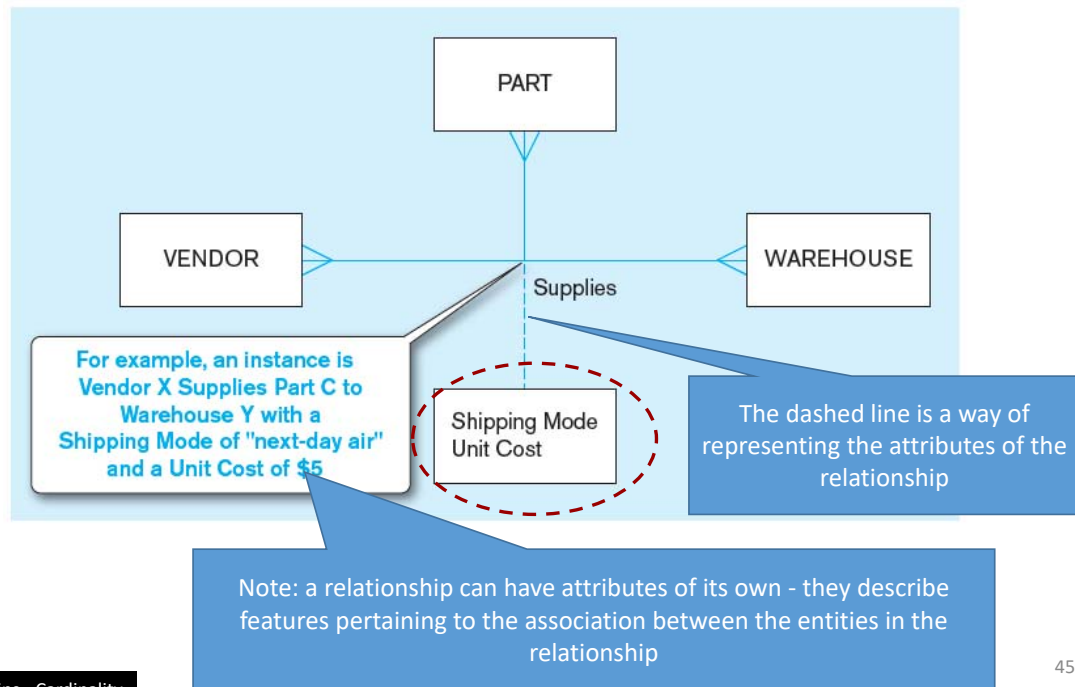


43

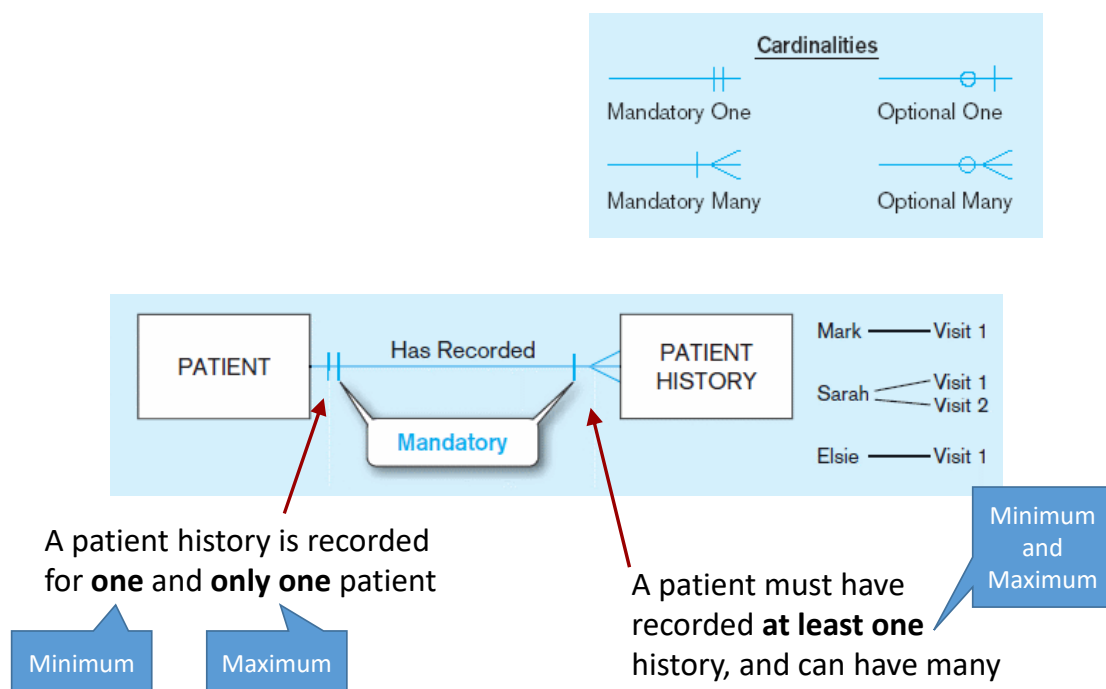
Unary relationships can also have different cardinality constraints



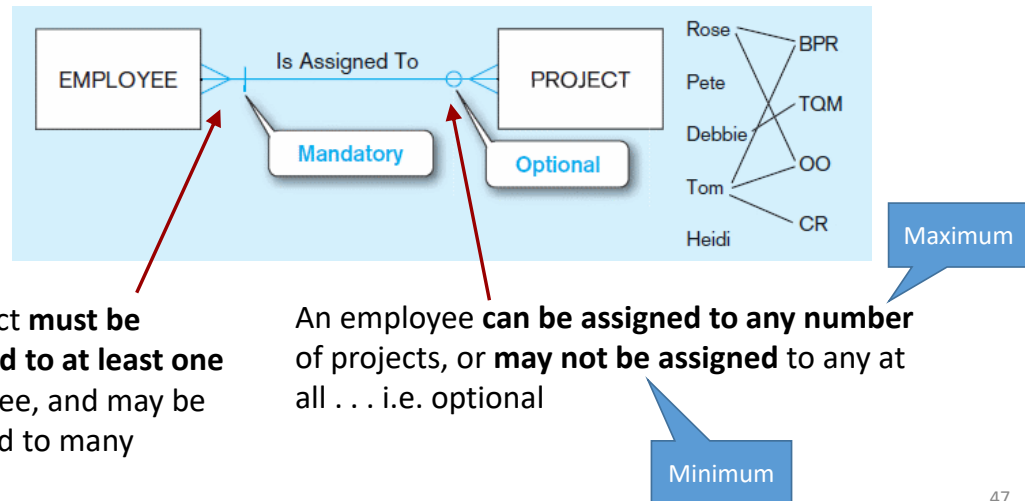
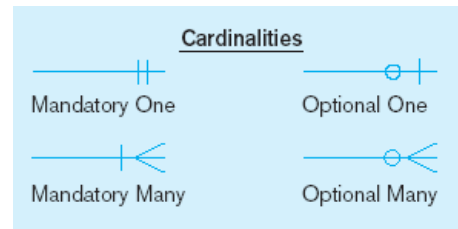
The cardinality of this ternary relationship is many-to-many



Minimum and Maximum Cardinalities can be depicted on ERDs... mandatory cardinalities in this example

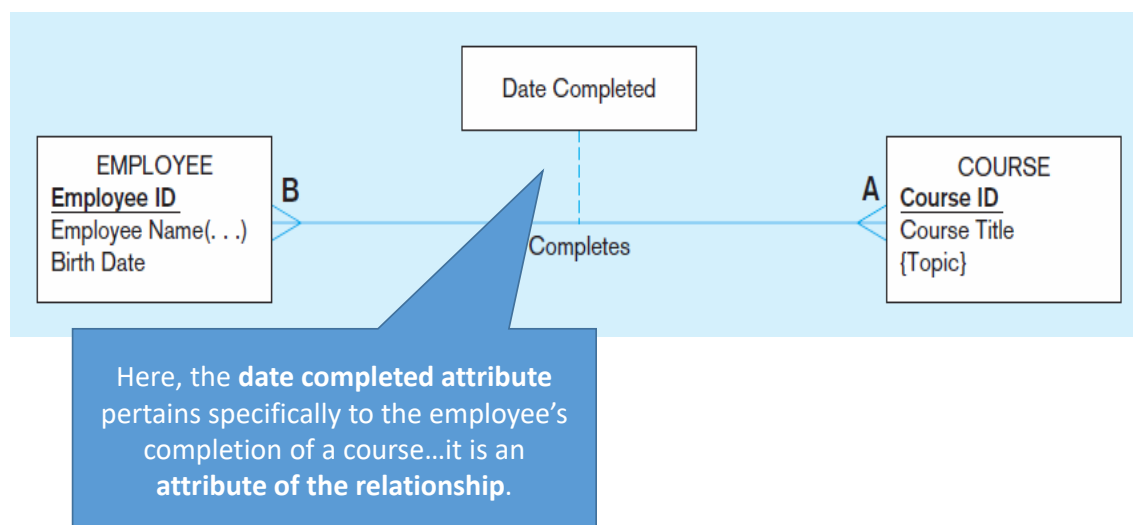


Minimum Cardinality can also show when a relationship is optional



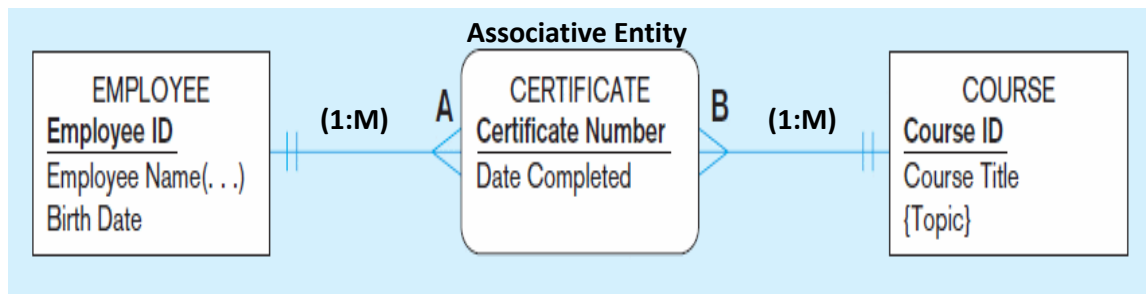
47

Relationships can have attributes



48

Here the relationship with an attribute is modeled as an Associative Entity (CERTIFICATE)



Associative entity is like a relationship with attribute(s)

- Also considered to be an entity in its own right
- Drawn as rectangle with rounded corners

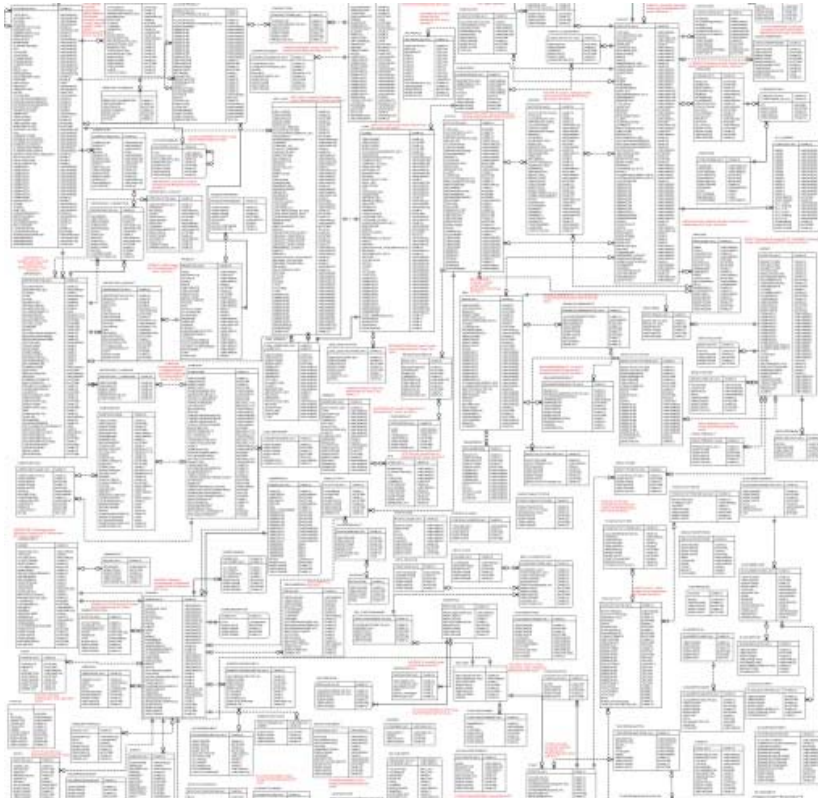
Also note that the **many-to-many** cardinality between entities on previous page has been replaced by

- Two **one-to-many (1:M)** relationships with
- The **associative entity**

When should a relationship with attributes be modeled as an associative entity?

1. All relationships for the associative entity should be many
2. The associative entity could have meaning independent of the other entities
3. The associative entity preferably has a unique identifier, and should also have other attributes
4. The associative entity may participate in other relationships other than the entities of the associated relationship
5. Ternary relationships should be converted to associative entities

For major systems **schemas** are large and complex



51

Agenda

Introduction

Data in Organizations

Relational Database Management Systems (RDMS)

Data Modeling and Entity Relationship Diagrams (ERDs)

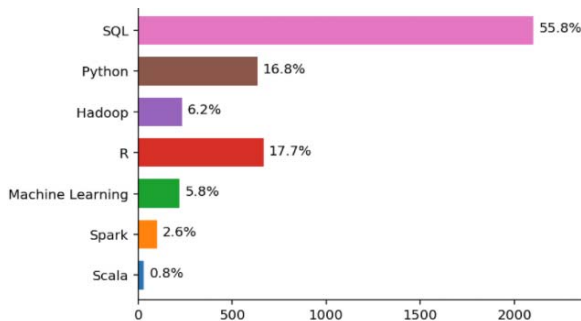
Hands on with SQL

Overview of rest of course

52

SQL is an important language

Number of 'Data Analyst' Jobs Mentioning Specific Skill



Languages used by Data Scientists and Data Analysts*



* StackOverflow 2018 Developer Survey

Source: <https://www.dataquest.io/blog/why-sql-is-the-most-important-language-to-learn>

53

Follow Hands-on exercises (see Canvas page for Session 1)

Hands on

1. Download and install the example databases we will use to start our exploration of SQL. You can follow these [instructions](#) and/or follow this [video](#).
2. Using MySQL Workbench and getting introduced to SQL by executing SELECT queries one at a time from the this script ([Getting started with SQL.sql](#)) and/or following along with these short videos
 - [Intro to MySQL Workbench](#)
 - [Selecting databases and running queries](#)
 - [Selecting specific columns/rows, ordering results, calculations, and aliases](#)
 - [WHERE clause with dates and basic mathematical operators](#)
 - [More functions in SQL and duplicates](#)
 - [More on the WHERE clause and logical operators](#)
 - [IN and BETWEEN phrases](#)
 - [Working with NULLs](#)

The official reference for MySQL is [here](#). While there are many great resources for SQL online [this one](#) uses the same dialect of SQL we use in the class.

3. Drawing ERDs using <https://app.diagrams.net/> as explained in this [video](#).

54

Agenda

Introduction

Data in Organizations

Relational Database Management Systems (RDMS)

Data Modeling and Entity Relationship Diagrams (ERDs)

Hands on with SQL

Overview of rest of course ... including next steps

55

We will cover quite a lot in our five sessions together

	Topics*	Hands on
1	Data in organizations RDBMS Data Modeling and ERDs	Getting started with MySQL Simple SQL SELECT queries on single table Use modeling tool to create ERD
2	Relational Data Model Integrity constraints Convert ERD into database schema Database normalization	Creating/deleting databases and tables using SQL Data Definition Language (DDL) More queries – inc. INSERT and UPDATE queries
3	Aggregation Working with relational data in SQL Subqueries and Views	Multi-table queries (JOIN) SQL scripts
4	Analytics in organizations and the enterprise wide views of corporate data Extract Transform and Load (ETL) Data cleaning and text parsing OLAP, Business Intelligence Tools & Visualization Big Data: Hadoop, Hive, and NoSQL	Perform ETL tasks Text parsing with Regular Expressions
5	Client-Server and three layers architectures Connecting to databases (ODBC, JDBC) Databases and the Web Web Services and common data formats (JSON & XML)	Connecting to database using Python Accessing web services from Python

* Subject to change . . . the up-to-date version with more details can be found on Canvas

56

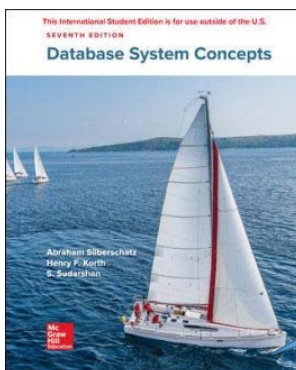
Course assessment is based on group homework assignments* and a final exam

Assessment type	Weightings	Group/ Individual	Release date of assessment component	Deadline/ Time & Date of exam
Homework 1	12%	Group	22 September on Canvas	27 September on Canvas
Homework 2	12%	Group	29 September on Canvas	4 October on Canvas
Homework 3	13%	Group	06 October on Canvas	11 October on Canvas
Homework 4	13%	Group	13 October on Canvas	18 October on Canvas
Final Exam	50%	Individual		22 October

* See Canvas for details of the late submission policy for this course

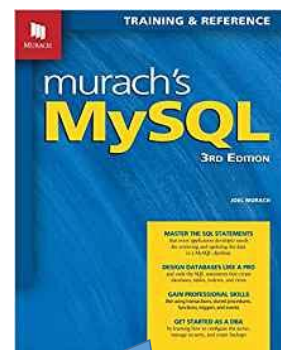
57

Some useful resources . . . Others linked from Canvas

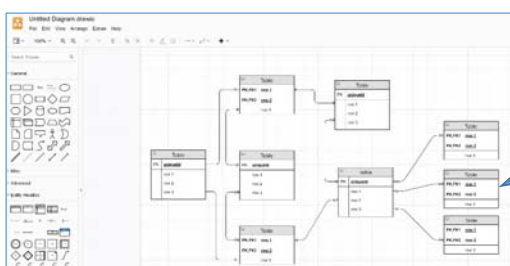


I have assigned chapters from this text for you to read and/or use for reference

It can be purchased from the publisher.
<https://www.mheducation.co.uk/ise-database-system-concepts-9781260084504-emea>



This book is a great reference for the work you will do with MySQL. I will use some examples from this book.



<https://www.draw.io/> provides a free tool for creating ERDs

See **hands on** section for a video of how to use it

58

For next time

- ¶ Hands on activities (see Canvas page for session 1)
- ¶ Submit Homework Assignment #1 (team)
 - Covers SELECT queries and ERDs (details on Canvas)
 - Submit via Canvas by 10pm on 27 September
- ¶ Attend Thursday workshop (optional) if you think you might need more support before completing the homework
- ¶ Check out readings for this session and next

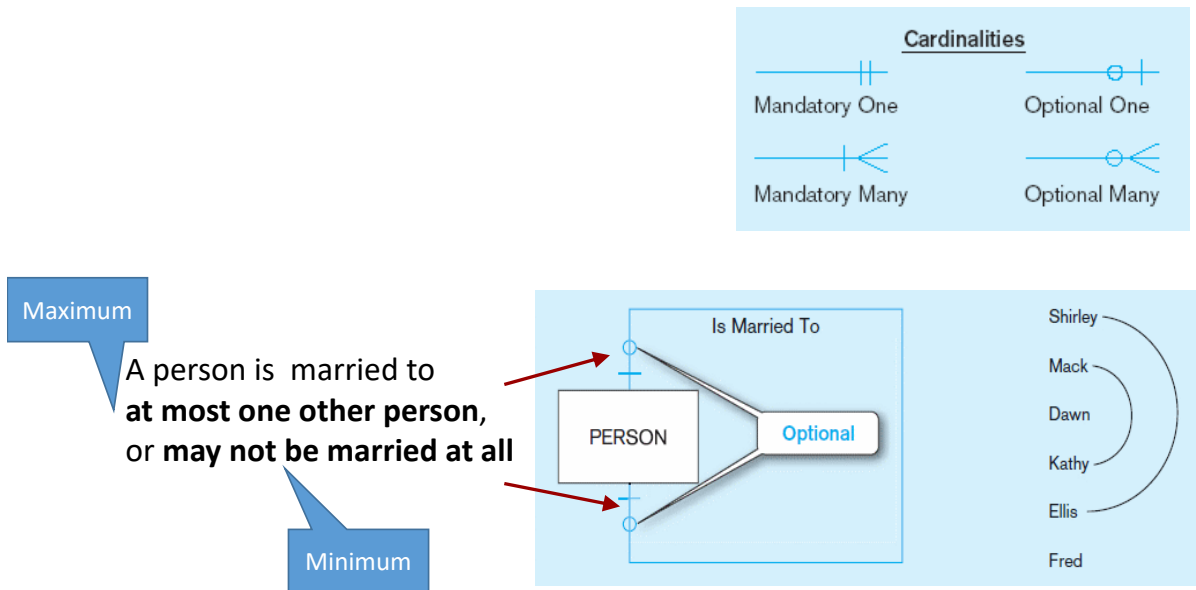
59

Appendix

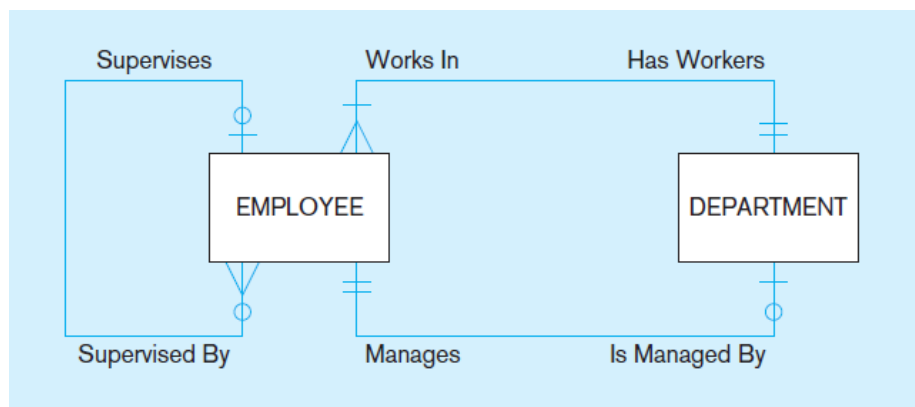
Additional Modelling Examples

60

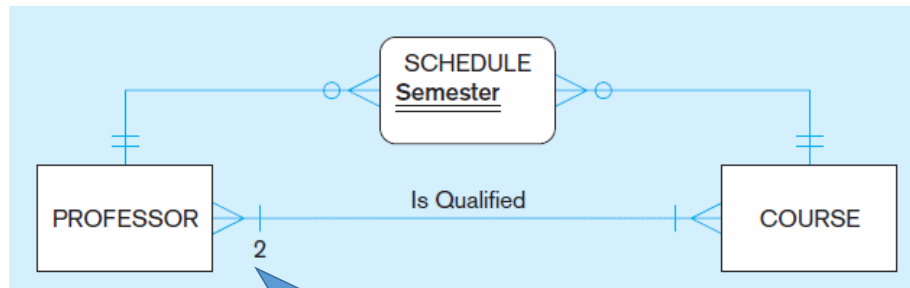
According to this unary relationship polygamy is ruled out



Entities can be related to one another in more than one way



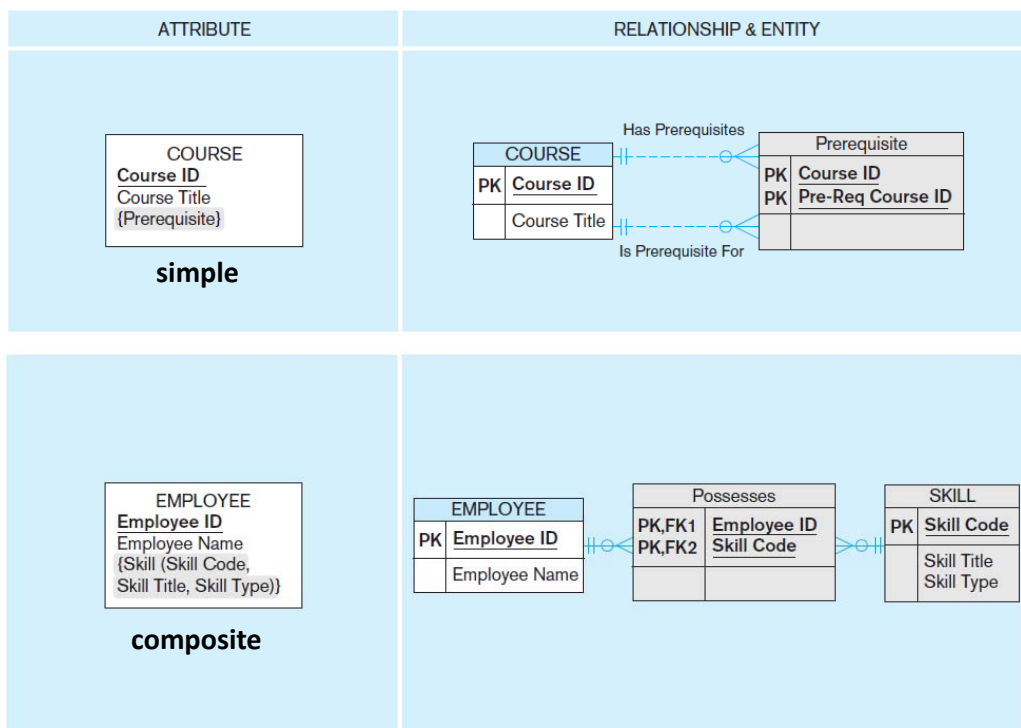
Other values for the lower limit can be depicted



Relationships - Cardinality

63

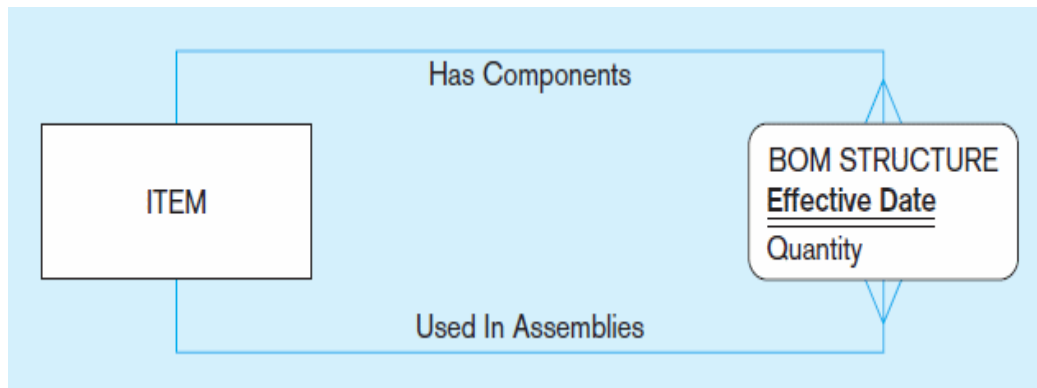
Multivalued attributes can be represented as relationships



Relationships

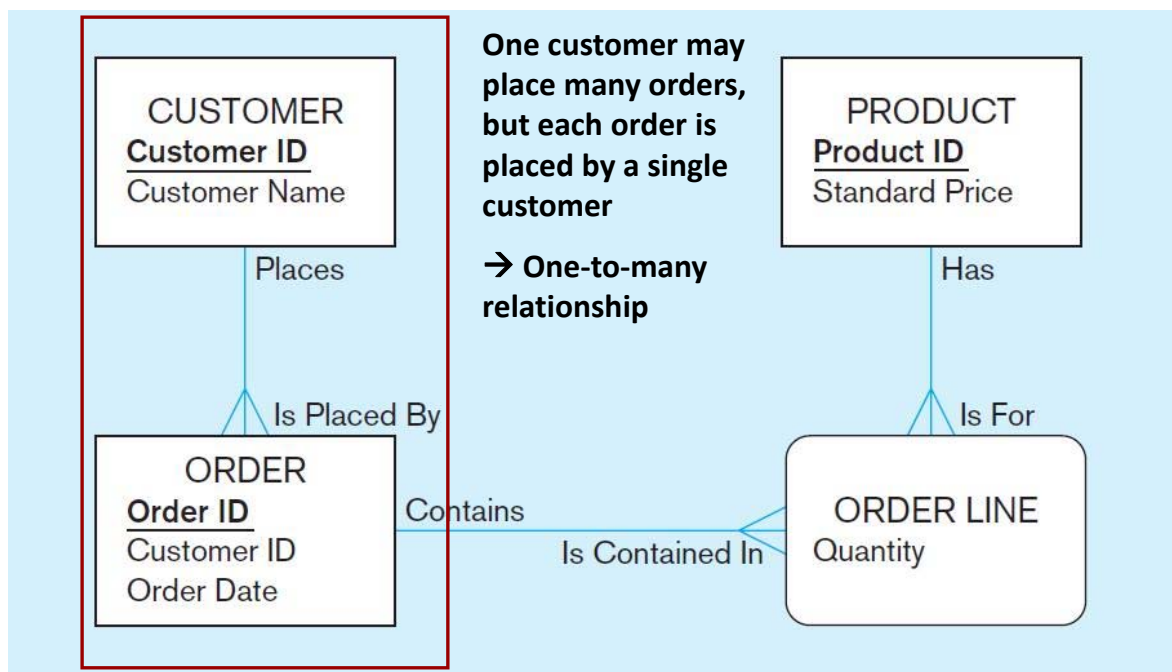
64

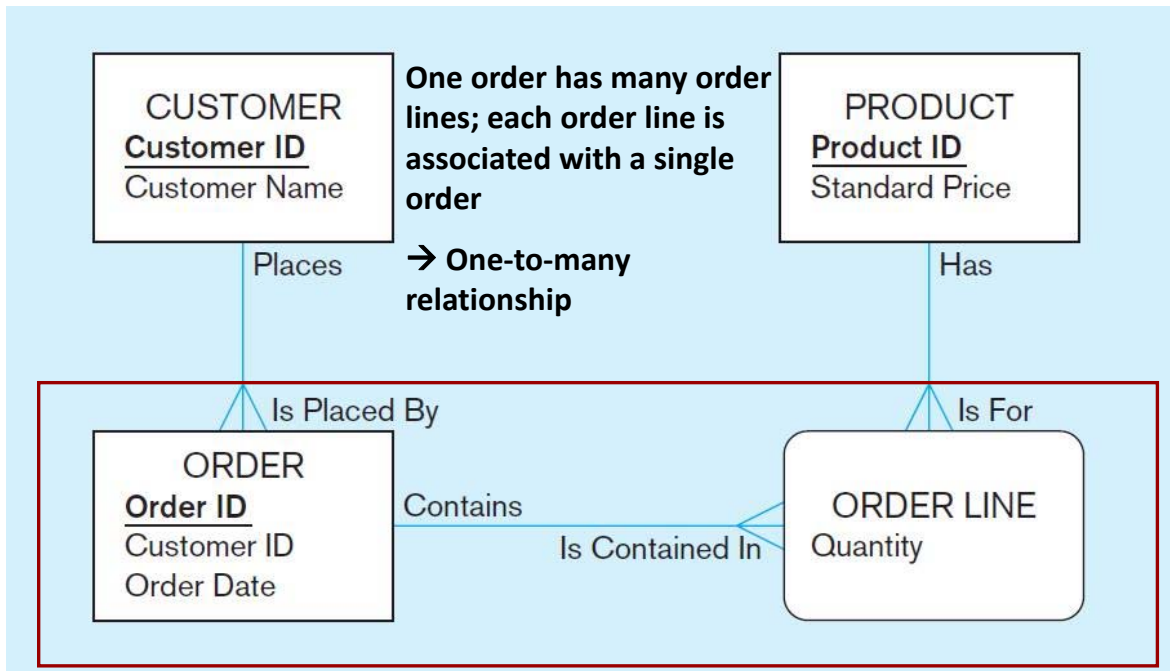
Here a Bill of Materials (BOM) structure is modeled with an associative entity



This could just as well have been model as a unary many-to-many relationship between items

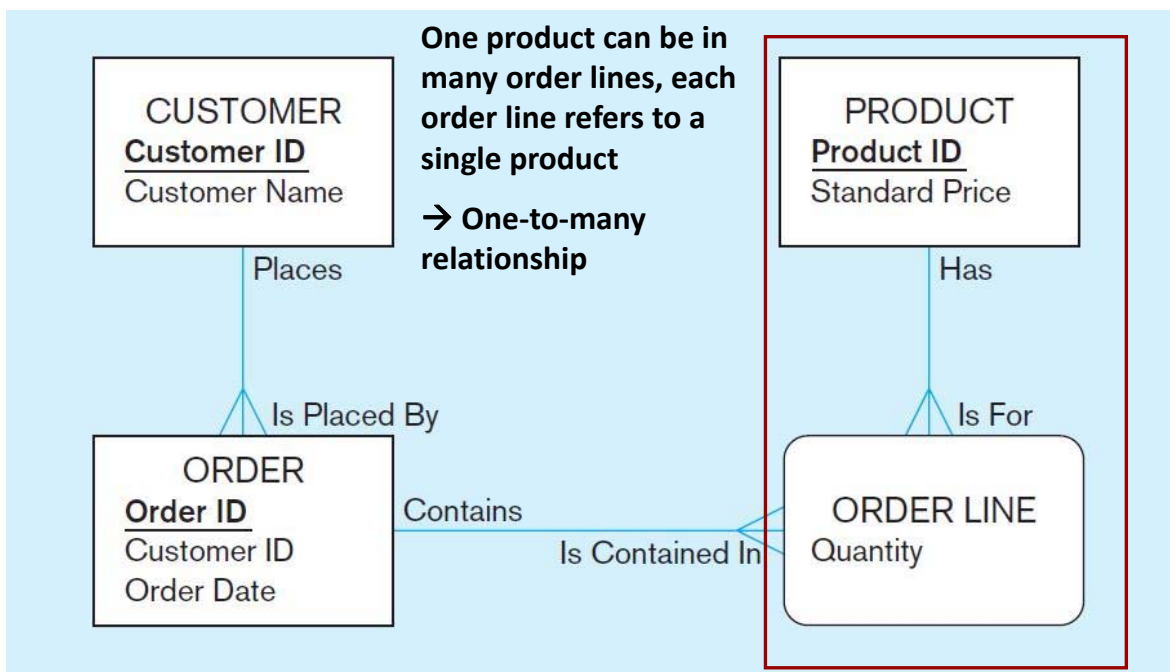
Read next slides for example of associative entity





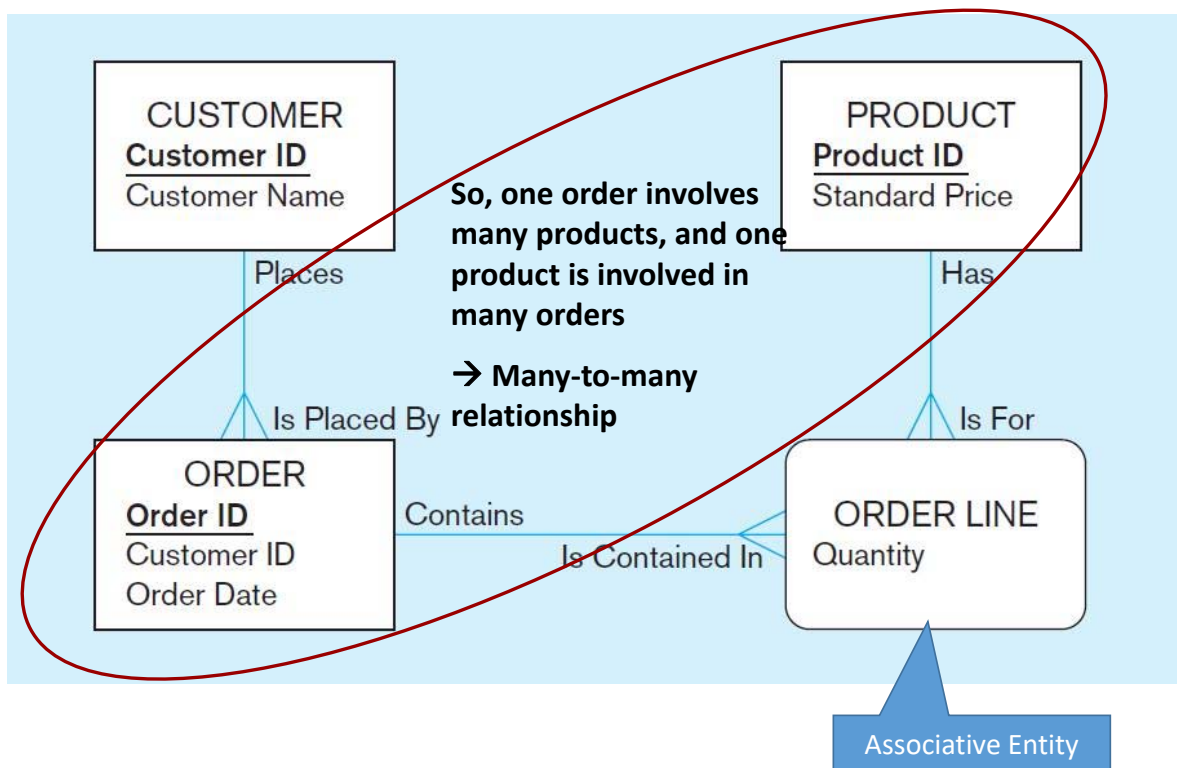
Example

67



Example

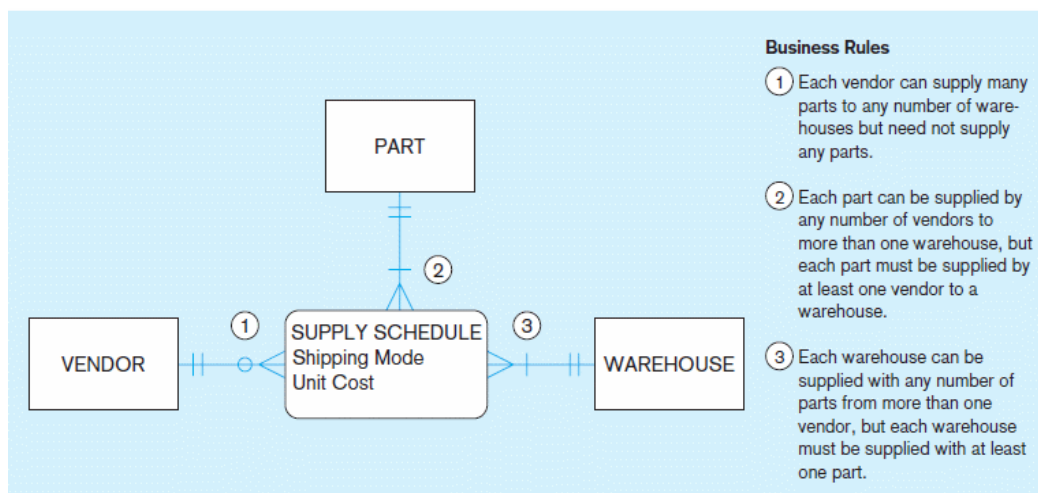
68



Example

69

Ternary and (n-ary) relationships can be modeled with associative entities



Associative Entities

70