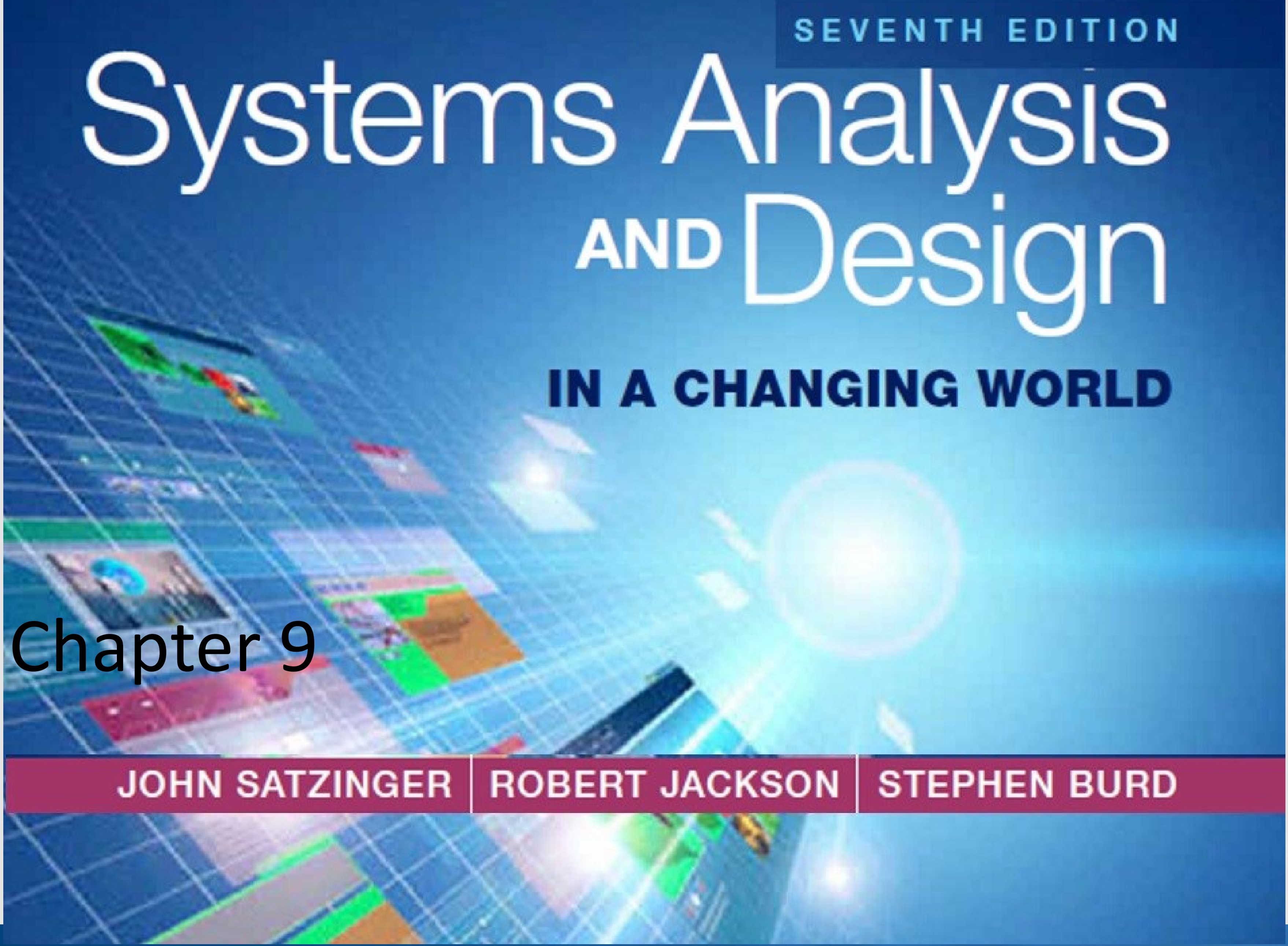


SEVENTH EDITION

Systems Analysis and Design

IN A CHANGING WORLD

JOHN SATZINGER | ROBERT JACKSON | STEPHEN BURD

Chapter 9

Designing the Database

Chapter 9

Systems Analysis and Design in a Changing World 7th Ed
Satzinger, Jackson & Burd

Chapter 9: Outline

- Databases and Database Management Systems
- Database Design and Administration
- Relational Databases
- Distributed Database Architectures
- Protecting the Database

Learning Objectives

- Explain the responsibilities of the data administrator and database administrator
- Design a relational database schema based on a class diagram
- Evaluate and improve the quality of a database schema
- Describe the different methods for configuring distributed databases
- Explain the importance of and methods for protecting the database

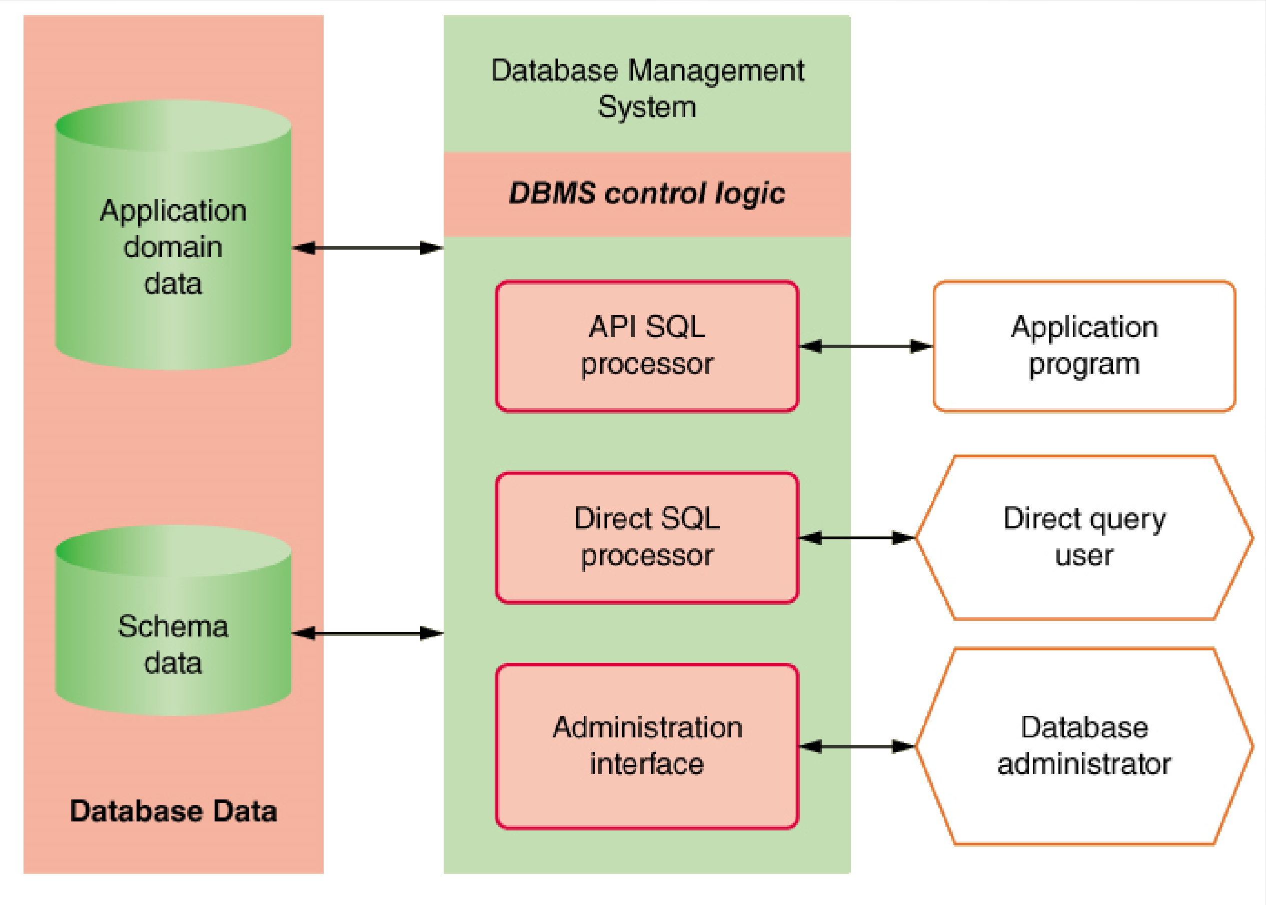
Overview

- Databases and database management systems are important components of a modern information system
- Database design transforms the domain model class diagram into a detailed database model for the system
- A database management system is used to implement and interact with the database

Databases and DBMSs

- Database (DB) -- an integrated collection of stored data that is centrally managed and controlled
- Database management system (DBMS) -- a system software component that manages and controls one or more databases
- Schema -- database component that contains descriptive information about the data stored in the physical data store (sometimes called *metadata*)
- Structured Query Language -- the standard query language to access and update data in a relational DBMS

DBMS Components



Database Schema

- Organization of individual stored data items into higher level groups, such as tables
- Associations among tables or classes
- Details of physical data store organization, including types, lengths, locations, and indexing of data items
- Access and content controls, including allowable values for specific data items, value dependencies among multiple data items, and lists of users allowed to read or update data items

Characteristics of a DBMS

- Simultaneous access by many users and many applications
- Direct access to data with a data interface
- Uniform and consistent access
- Integration and distribution of data across multiple servers

Database Design and Administration (1 of 4)

- How does database design integrate within the existing technology architecture?
- Technology Architecture – hardware and networks
  - Single desktop – single copy of database
  - Shared database – residing on local LAN
  - Large database – multiple servers within a single data farm
  - Very large database – multiple servers across multiple data farms (global)

Database Design and Administration (2 of 4)

- How does database design integrate within the project plan?
- Water-fall development – design and implement database first
- Iterative development – database is foundational, early iterations need to focus on data and key portions of the database
- Iterative development – important to consider database impacts of all subsystems

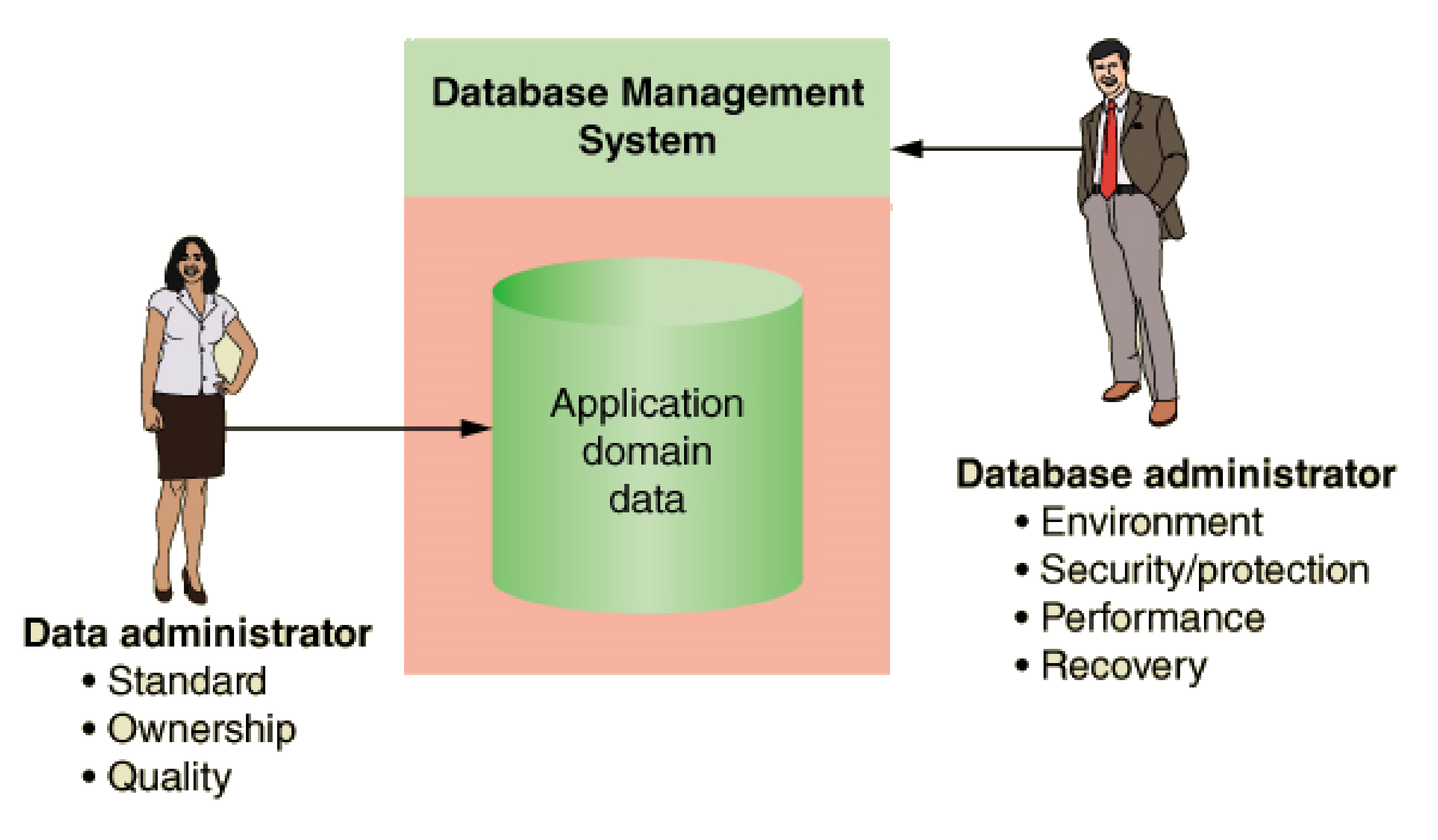
Database Design and Administration (3 of 4)

- Who is involved in database design?
- Data Administrator (DA) – person in charge of structure and integrity of the data
  - Data standards – naming, definition, data typing
  - Data use – ownership, accessibility, confidentiality
  - Data quality – validation rules, completeness, currency

Database Design and Administration (4 of 4)

- Who is involved in database design?
- Database Administrator (DBA) – person in charge of safety and the operation of the database
  - Manage multiple DBMS environment
  - Protect the database and data – authentication
  - Maintain high-performance level
  - Backup data and define recovery procedures

DA and DBA Responsibilities



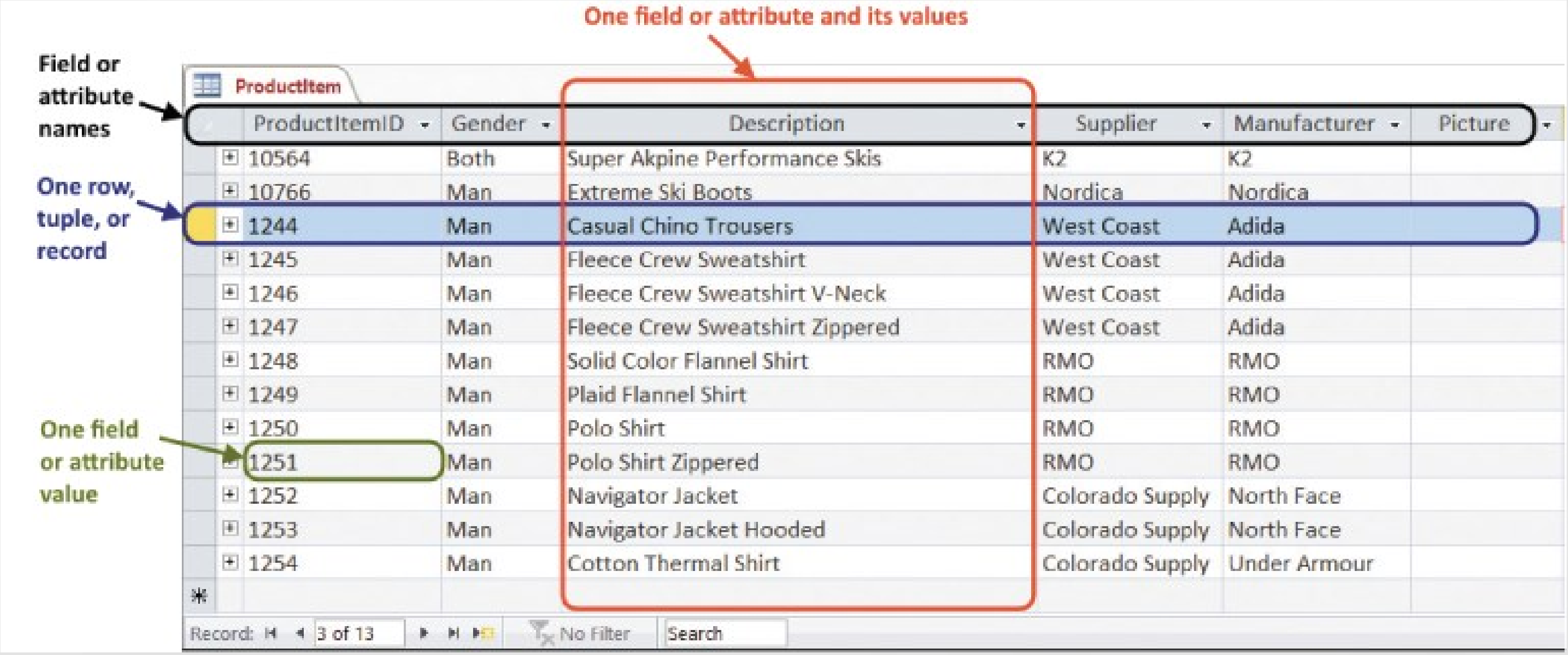
Relational Databases (1 of 2)

- Relational database management system (RDBMS) -- a DBMS that organizes data in tables (relations)
- Table -- a two-dimensional data structure of columns and rows
- Row -- one horizontal group of data attribute values
- Attribute -- one vertical group of data attribute values
- Attribute value -- the value held in a single table cell

Relational Databases (2 of 2)

- Key – an attribute or set of attributes, the values of which occur only once in all the rows of the table
- Candidate Key – an attribute or set of attributes that could server as the primary key
- Primary key – the key chosen by a database designer to represent relationships among rows in different tables
- Foreign key – an attribute that duplicates the primary key of a different (or foreign) table

Partial Display of a Relational Database Table



One field or attribute and its values

Field or attribute names

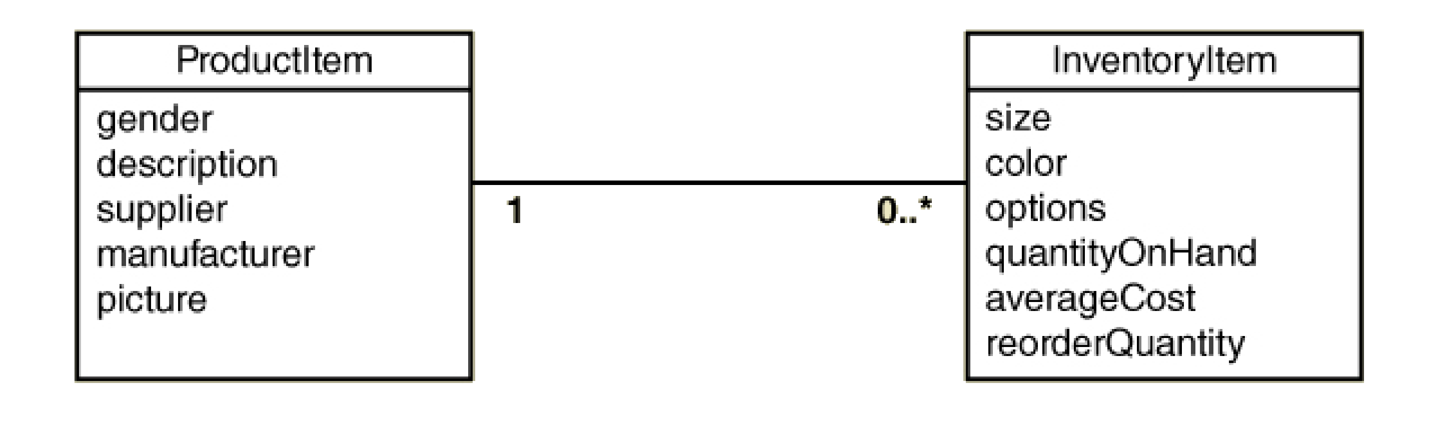
One row, tuple, or record

One field or attribute value

ProductItem

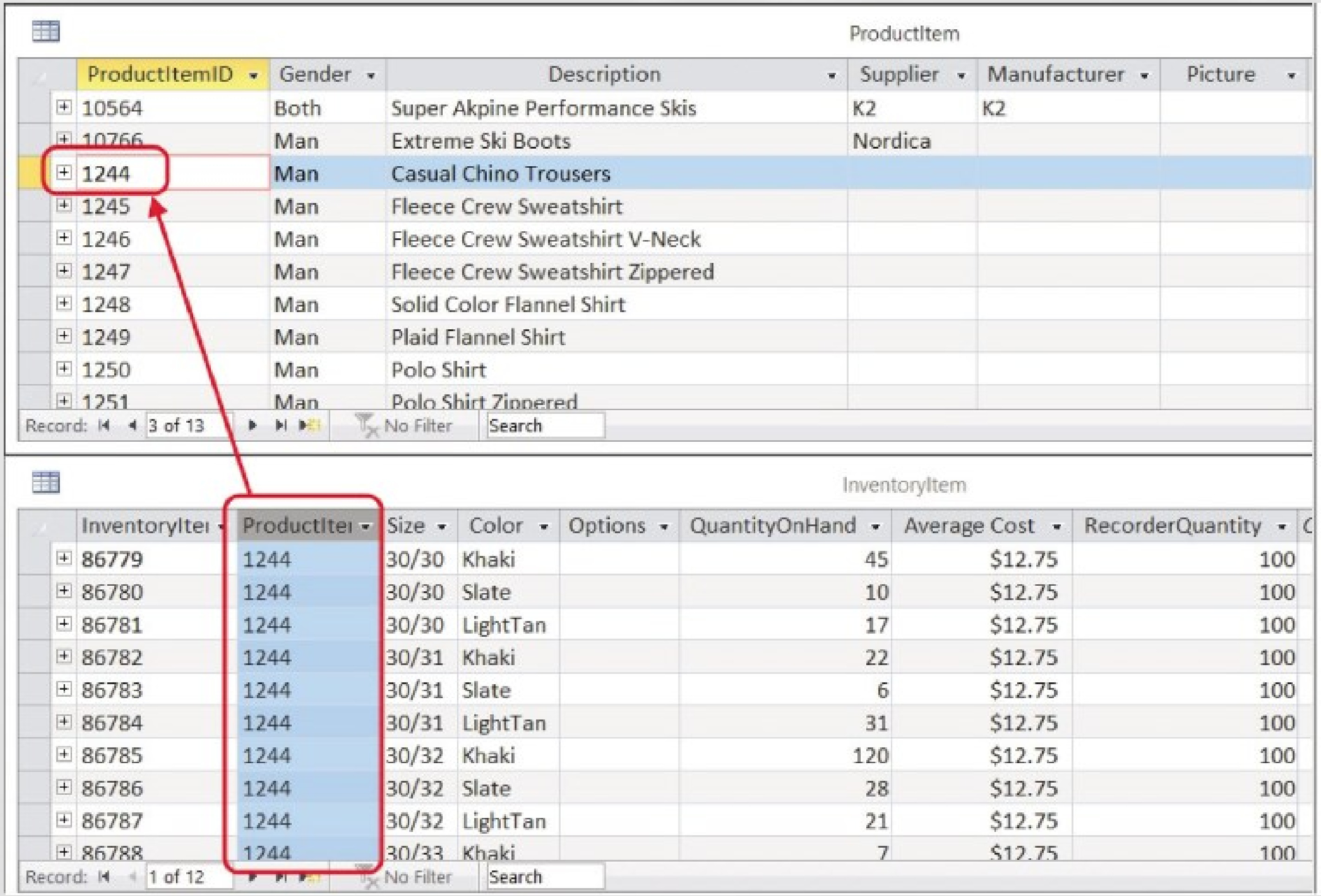
ProductItemID	Gender	Description	Supplier	Manufacturer	Picture
10564	Both	Super Akpine Performance Skis	K2	K2	
10766	Man	Extreme Ski Boots	Nordica	Nordica	
1244	Man	Casual Chino Trousers	West Coast	Adida	
1245	Man	Fleece Crew Sweatshirt	West Coast	Adida	
1246	Man	Fleece Crew Sweatshirt V-Neck	West Coast	Adida	
1247	Man	Fleece Crew Sweatshirt Zippered	West Coast	Adida	
1248	Man	Solid Color Flannel Shirt	RMO	RMO	
1249	Man	Plaid Flannel Shirt	RMO	RMO	
1250	Man	Polo Shirt	RMO	RMO	
1251	Man	Polo Shirt Zippered	RMO	RMO	
1252	Man	Navigator Jacket	Colorado Supply	North Face	
1253	Man	Navigator Jacket Hooded	Colorado Supply	North Face	
1254	Man	Cotton Thermal Shirt	Colorado Supply	Under Armour	

An Association Between Two Classes



An Association Between Rows in Two Tables

(key and foreign key)



ProductItem

ProductItemID	Gender	Description	Supplier	Manufacturer	Picture
10564	Both	Super Akpine Performance Skis	K2	K2	
10766	Man	Extreme Ski Boots	Nordica		
1244	Man	Casual Chino Trousers			
1245	Man	Fleece Crew Sweatshirt			
1246	Man	Fleece Crew Sweatshirt V-Neck			
1247	Man	Fleece Crew Sweatshirt Zippered			
1248	Man	Solid Color Flannel Shirt			
1249	Man	Plaid Flannel Shirt			
1250	Man	Polo Shirt			
1251	Man	Polo Shirt Zippered			

Record: 3 of 13 No Filter Search

InventoryItem

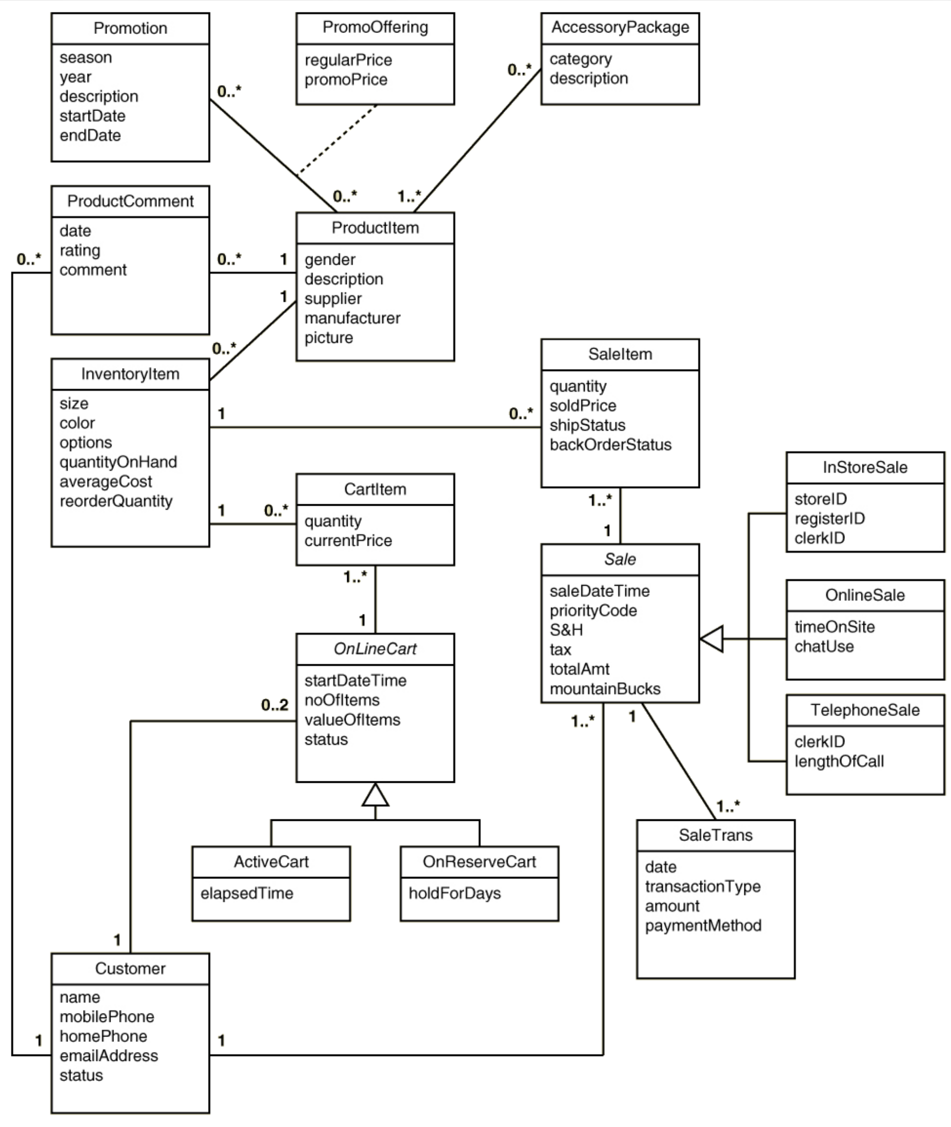
InventoryIte	ProductIte	Size	Color	Options	QuantityOnHand	Average Cost	RecorderQuantity	C
86779	1244	30/30	Khaki		45	$12.75	100	
86780	1244	30/30	Slate		10	$12.75	100	
86781	1244	30/30	LightTan		17	$12.75	100	
86782	1244	30/31	Khaki		22	$12.75	100	
86783	1244	30/31	Slate		6	$12.75	100	
86784	1244	30/31	LightTan		31	$12.75	100	
86785	1244	30/32	Khaki		120	$12.75	100	
86786	1244	30/32	Slate		28	$12.75	100	
86787	1244	30/32	LightTan		21	$12.75	100	
86788	1244	30/33	Khaki		7	$12.75	100	

Record: 1 of 12 No Filter Search

Designing Relational Databases: Based on the Domain Model Class Diagram

1. Create a table for each class
2. Choose a primary key for each table (invent one, if necessary)
3. Add foreign keys to represent one-to-many associations
4. Create new tables to represent many-to-many associations
5. Represent classification hierarchies
6. Define referential integrity constraints
7. Evaluate schema quality and make necessary improvements
8. Choose appropriate data types
9. Incorporate integrity and security controls

RMO Classes



Promotion
season
year
description
startDate
endDate

PromoOffering
regularPrice
promoPrice

AccessoryPackage
category
description

ProductComment
date
rating
comment

ProductItem
gender
description
supplier
manufacturer
picture

InventoryItem
size
color
options
quantityOnHand
averageCost
reorderQuantity

SaleItem
quantity
soldPrice
shipStatus
backOrderStatus

CartItem
quantity
currentPrice

OnLineCart
startDateTime
noOfItems
valueOfItems
status

ActiveCart
elapsedTime

OnReserveCart
holdForDays

Sale
saleDateTime
priorityCode
S&H
tax
totalAmt
mountainBucks

InStoreSale
storeID
registerID
clerkID

OnlineSale
timeOnSite
chatUse

TelephoneSale
clerkID
lengthOfCall

SaleTrans
date
transactionType
amount
paymentMethod

Customer
name
mobilePhone
homePhone
emailAddress
status

Initial Set of Tables: Based on RMO Domain Classes

Table	Attributes
AccessoryPackage	Category, Description
CartItem	Quantity, CurrentPrice
Customer	Name, MobilePhone, HomePhone, EmailAddress, Status
InventoryItem	Size, Color, Options, QuantityOnHand, AverageCost, ReorderQuantity
OnlineCart	StartDateTime, NumberOfItems, ValueOfItems, Status, ElapsedTime, HoldForDays
ProductComment	Date, Rating, Comment
ProductItem	Gender, Description, Supplier, Manufacturer, Picture
PromoOffering	RegularPrice, PromoPrice
Promotion	Season, Year, Description, StartDate, EndDate
Sale	SaleDateTime, PriorityCode, ShippingAndHandling, Tax, TotalAmount, MountainBucks, StoreID, RegisterID, ClerkID, TimeOnSite, ChatUse, LengthOfCall
SaleItem	Quantity, SoldPrice, ShipStatus, BackOrderStatus
SaleTransaction	Date, TransactionType, Amount, PaymentMethod

Initial Set of Tables: With Primary Keys Added (bold)

Table	**Attributes**
AccessoryPackage	**AccessoryPackageID**, Category, Description
CartItem	**CartItemID**, Quantity, CurrentPrice
Customer	**AccountNumber**, Name, MobilePhone, HomePhone, EmailAddress, Status
InventoryItem	**InventoryItemID**, Size, Color, Options, QuantityOnHand, AverageCost, ReorderQuantity
OnlineCart	**OnlineCartID**, StartDateTime, NumberOfItems, ValueOfItems, Status, ElapsedTime, HoldForDays
ProductComment	**ProductCommentID**, Date, Rating, Comment
ProductItem	**ProductItemID**, Gender, Description, Supplier, Manufacturer, Picture
PromoOffering	**PromoOfferingID**, RegularPrice, PromoPrice
Promotion	**PromotionID**, Season, Year, Description, StartDate, EndDate
Sale	**SaleID**, SaleDateTime, PriorityCode, ShippingAndHandling, Tax, TotalAmount, MountainBucks, StoreID, RegisterID, ClerkID, TimeOnSite, ChatUse, LengthOfCall
SaleItem	**SaleItemID**, Quantity, SoldPrice, ShipStatus, BackOrderStatus
SaleTransaction	**SaleTransactionID**, Date, TransactionType, Amount, PaymentMethod

Representing Associations

- One-to-Many – Add primary key attribute of the “one” class to the “many” class as a foreign key
- Many-to-Many –
  - With an Association Class – Add primary keys of endpoint classes as foreign keys and as candidate keys. May also become primary key
  - Without an Association Class – Create new table. Add primary keys of endpoint classes as foreign keys and as candidate keys.

Initial Set of Tables: With Foreign Keys Added (in italics)

Table	**Attributes**
Accessory Package	**AccessoryPackageID**, Category, Description
Cartitem	**CartItemID**, *InventoryItemID*, *OnlineCartID*, Quantity, CurrentPrice
Customer	**AccountNumber**, Name, MobilePhone, HomePhone, EmailAddress, Status
InventoryItem	**InventoryItemID**, *ProductItemID*, Size, Color, Options, QuantityOnHand, AverageCost, ReorderQuantity
OnlineCart	**OnlineCartID**, *CustomerAccountNumber*, StartDateTime, NumberOfItems, ValueOfItems, Status, ElapsedTime, HoldForDays
ProductComment	**ProductCommentID**, *ProductItemID*, *CustomerAccountNumber*, Date, Rating, Comment
ProductItem	**ProductItemID**, Gender, Description, Supplier, Manufacturer, Picture
PromoOffering	**PromoOfferingID**, RegularPrice, PromoPrice
Promotion	**PromotionID**, Season, Year, Description, StartDate, EndDate
Sale	**SaleID**, *CustomerAccountNumber*, SaleDateTime, PriorityCode, ShippingAndHandling, Tax, TotalAmount, MountainBucks, StoreID, RegisterID, ClerkID, TimeOnSite, ChatUse, LengthOfCall
SaleItem	**SaleItemID**, *InventoryItemID*, *SaleID*, Quantity, SoldPrice, ShipStatus, BackOrderStatus
SaleTransaction	**SaleTransactionID**, *SaleID*, Date, TransactionType, Amount, PaymentMethod

Association Class: PromoOffering added from association class to table with two keys

Table	**Attributes**
AccessoryPackage	**AccessoryPackageID**, AccessoryCategory, Description
AccessoryPackageContents	***AccessoryPackageID, ProductItemID***
CartItem	***InventoryItemID***, ***OnlineCartID***, Quantity, CurrentPrice
Customer	**AccountNumber**, Name, MobilePhone, HomePhone, EmailAddress, Status
InventoryItem	**InventoryItemID**, *ProductItemID*, Size, Color, Options, QuantityOnHand, AverageCost, ReorderQuantity
OnlineCart	**OnlineCartID**, *CustomerAccountID*, StartDateTime, NumberOfItems, ValueOfItems, Status, ElapsedTime, HoldForDays
ProductComment	**ProductCommentID**, *ProductItemID*, CustomerAccountNumber, Date, Rating, Comment
ProductItem	**ProductItemID**, Gender, Description, Supplier, Manufacturer, Picture
PromoOffering	***PromotionID, ProductItemID,*** RegularPrice, PromoPrice
Promotion	**PromotionID**, Season, Year, Description, StartDate, EndDate
Sale	**SaleID**, *CustomerAccountNumber*, SaleDateTime, PriorityCode, ShippingAndHandling, Tax, TotalAmount, MountainBucks, StoreID, RegisterID, ClerkID, TimeOnSite, ChatUse, LengthOfCall
SaleItem	***InventoryItemID, SaleID***, Quantity, SoldPrice, ShipStatus, BackOrderStatus
SaleTransaction	**SaleTransactionID**, *SaleID*, Date, TransactionType, Amount, PaymentMethod

Final Tables: Specialized subclasses included within OnlineCart and Sale tables

Table	**Attributes**
AccessoryPackage	**AccessoryPackageID**, AccessoryCategory, Description
AccessoryPackageContents	***AccessoryPackageID, ProductItemID***
CartItem	***InventoryItemID, OnlineCartID***, Quantity, CurrentPrice
Customer	**AccountNumber**, Name, MobilePhone, HomePhone, EmailAddress, Status
InventoryItem	**InventoryItemID**, *ProductItemID*, Size, Color, Options, QuantityOnHand, AverageCost, ReorderQuantity
OnlineCart	**OnlineCartID**, *CustomerAccountID*, StartDateTime, NumberOfItems, ValueOfItems, Status, ElapsedTime, HoldForDays
ProductComment	**ProductCommentID**, *ProductItemID*, *CustomerAccountNumber*, Date, Rating, Comment
ProductItem	**ProductItemID**, Gender, Description, Supplier, Manufacturer, Picture
PromoOffering	***PromotionID, ProductItemID,*** RegularPrice, PromoPrice
Promotion	**PromotionID**, Season, Year, Description, StartDate, EndDate
Sale	**SaleID**, *CustomerAccountNumber*, SaleDateTime, PriorityCode, ShippingAndHandling, Tax, TotalAmount, MountainBucks, StoreID, RegisterID, ClerkID, TimeOnSite, ChatUse, LengthOfCall
SaleItem	***InventoryItemID, SaleID***, Quantity, SoldPrice, ShipStatus, BackOrderStatus
SaleTransaction	**SaleTransactionID**, *SaleID*, Date, TransactionType, Amount, PaymentMethod

Final Tables: Specialized subclasses as separate tables (1 of 2)

Table	**Attributes**
AccessoryPackage	**AccessoryPackageID**, AccessoryCategory, Description
AccessoryPackageContents	***AccessoryPackageID, ProductItemID***
CartItem	***InventoryItemID, OnlineCartID***, Quantity, CurrentPrice
Customer	**AccountNumber**, Name, MobilePhone, HomePhone, EmailAddress, Status
InventoryItem	**InventoryItemID**, *ProductItemID*, Size, Color, Options, QuantityOnHand, AverageCost, ReorderQuantity
OnlineCart	**OnlineCartID**, *CustomerAccountID*, StartDateTime, NumberOfItems, ValueOfItems, Status, ElapsedTime, HoldForDays
ActiveCart	***OnlineCartID*, ** ElapsedTime
OnReserveCart	***OnlineCartID*, ** HoldForDays
ProductComment	**ProductCommentID**, ProductItemID, *CustomerAccountNumber*, Date, Rating, Comment

Final Tables: Specialized subclasses as separate tables (2 of 2)

Table	**Attributes**
ProductItem	**ProductItemID**, Gender, Description, Supplier, Manufacturer, Picture
PromoOffering	***PromotionID, ProductItemID,*** RegularPrice, PromoPrice
Promotion	**PromotionID**, Season, Year, Description, StartDate, EndDate
Sale	**SaleID**, *CustomerAccountNumber*, SaleDateTime, PriorityCode, ShippingAndHandling, Tax, TotalAmount, MountainBucks
InStoreSale	***SaleID***, StoreID, RegisterID, ClerkID
OnlineSale	***SaleID,*** TimeOnSite, ChatUse
TelephoneSale	***SaleID,*** ClerkID, LengthOfCall
SaleItem	***InventoryItemID, SaleID***, Quantity, SoldPrice, ShipStatus, BackOrderStatus
SaleTransaction	**SaleTransactionID**, *SaleID*, Date, TransactionType, Amount, PaymentMethod

Designing Relational Databases:

Referential Integrity and Schema Quality

- Referential integrity -- a consistent state among foreign key and primary key values
- Referential integrity constraint -- a constraint, stored in the schema, that the DBMS uses to automatically enforce referential integrity

Designing Relational Databases:

Referential Integrity and Normalization

- A normalized relational database schema has these features:
  - Flexibility or ease of implementing future data model changes
  - Lack of redundant data
  - Protects against insertion, deletion and update anomolies
- Normalization -- a formal technique for evaluating and improving the quality of a relational database schema
  - First Normal Form –
  - Second Normal Form –
  - Third Normal Form –

First Normal Form

- A table is in first normal form if every field contains only one value.
  - Not multiple values in an attribute

SSN	Name	Department	Salary	Dependants
11-22-3333	Mary Smith	Accounting	40,000	John, Alice, Dave
222-33-4444	Jose Pena	Marketing	50,000	---
333-44-5555	Frank Collins	Production	35,000	Jan, Julia

  - Not varying number of columns

SSN	Name	Department	Salary	Dependent	Dependent	Dependent
11-22-3333	Mary Smith	Accounting	40,000	John	Alice	Dave
222-33-4444	Jose Pena	Marketing	50,000	Blank	Blank	Blank
333-44-5555	Frank Collins	Production	35,000	Jan	Julia	Blank

First Normal Form - Solution

- Solution is to put multivalued attribute in a separate table.

SSN	Name	Department	Salary
111-22-3333	Mary Smith	Accounting	40,000
222-33-4444	Jose Pena	Marketing	50,000
333-44-5555	Frank Collins	Production	35,000

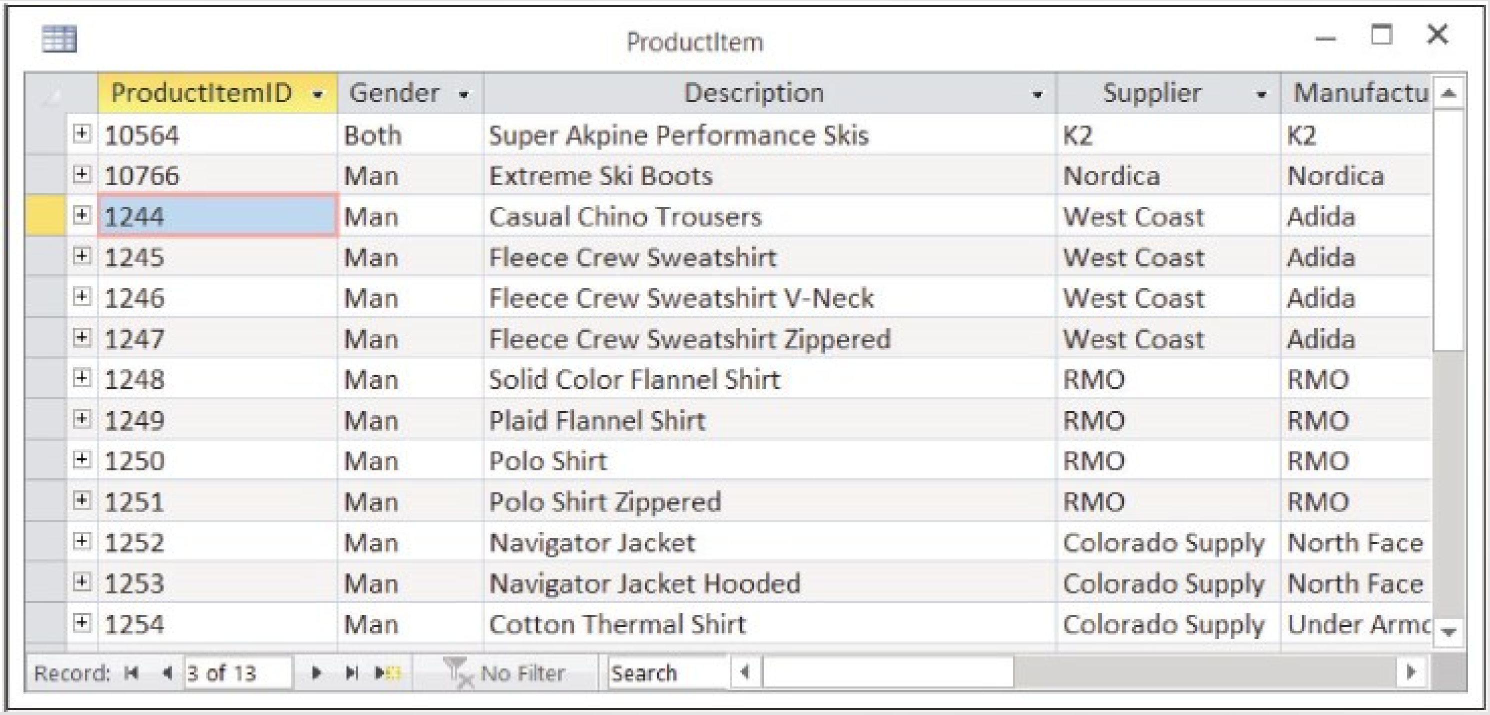
RecordID	SSN	Dependant
1	111-22-3333	John
2	111-22-3333	Alice
3	111-22-3333	Dave
4	333-44-5555	Jan
5	333-44-5555	Julia

Functional Dependency (1 of 2)

- A relationship between attributes such that the values in the first attribute (or set) always determine the values in the second attribute (or set)
- *Attribute B is functionally* ***dependent*** *on attribute A if for each value of attribute A there is only one corresponding value of attribute B.*
  - *Written as FD: A → B.*
  - *Also stated as A functionally* ***determines*** *B*

Functional Dependency (2 of 2)

- ProductID → Supplier
- But **NOT** Supplier → ProductID



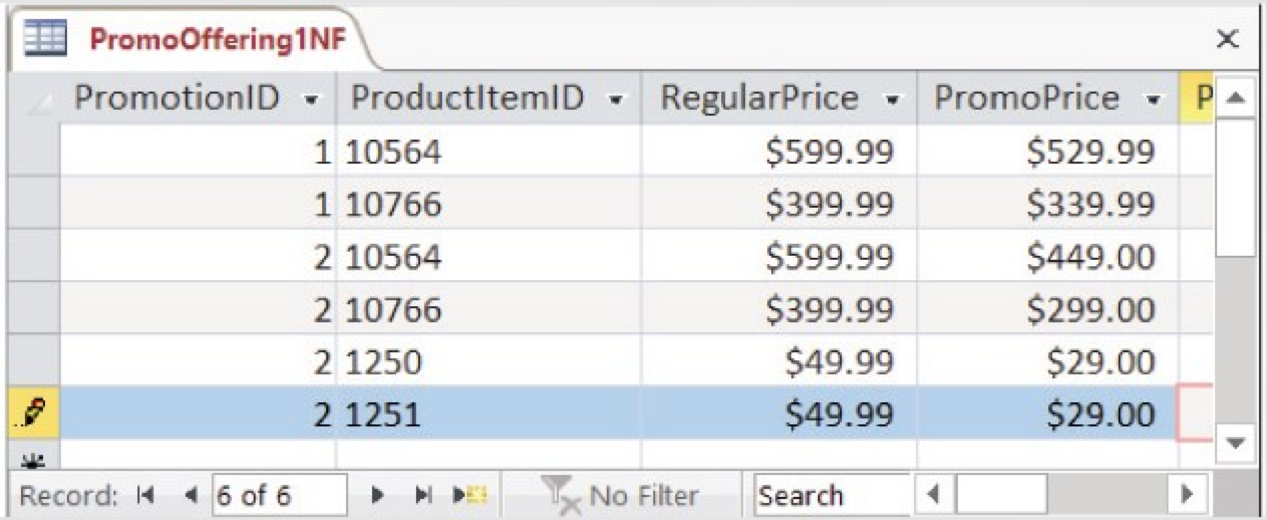
ProductItemID	Gender	Description	Supplier	Manufactu
10564	Both	Super Akpine Performance Skis	K2	K2
10766	Man	Extreme Ski Boots	Nordica	Nordica
1244	Man	Casual Chino Trousers	West Coast	Adida
1245	Man	Fleece Crew Sweatshirt	West Coast	Adida
1246	Man	Fleece Crew Sweatshirt V-Neck	West Coast	Adida
1247	Man	Fleece Crew Sweatshirt Zippered	West Coast	Adida
1248	Man	Solid Color Flannel Shirt	RMO	RMO
1249	Man	Plaid Flannel Shirt	RMO	RMO
1250	Man	Polo Shirt	RMO	RMO
1251	Man	Polo Shirt Zippered	RMO	RMO
1252	Man	Navigator Jacket	Colorado Supply	North Face
1253	Man	Navigator Jacket Hooded	Colorado Supply	North Face
1254	Man	Cotton Thermal Shirt	Colorado Supply	Under Armc

Second Normal Form (1 of 2)

- A table is in Second Normal Form if it is First Normal Form and each non-key attribute is only functionally dependent on the entire primary key.
  - This situation only arises with tables that have multiple attribute keys

Second Normal Form (2 of 2)

- PromoOffering table is **NOT** in 2NF
  - PromotionID, ProductItemID → PromoPrice
  - ProductItemID → RegularPrice -- Violation of 2NF



PromoOffering1NF

PromotionID	ProductItemID	RegularPrice	PromoPrice
1	10564	$599.99	$529.99
1	10766	$399.99	$339.99
2	10564	$599.99	$449.00
2	10766	$399.99	$299.00
2	1250	$49.99	$29.00
2	1251	$49.99	$29.00

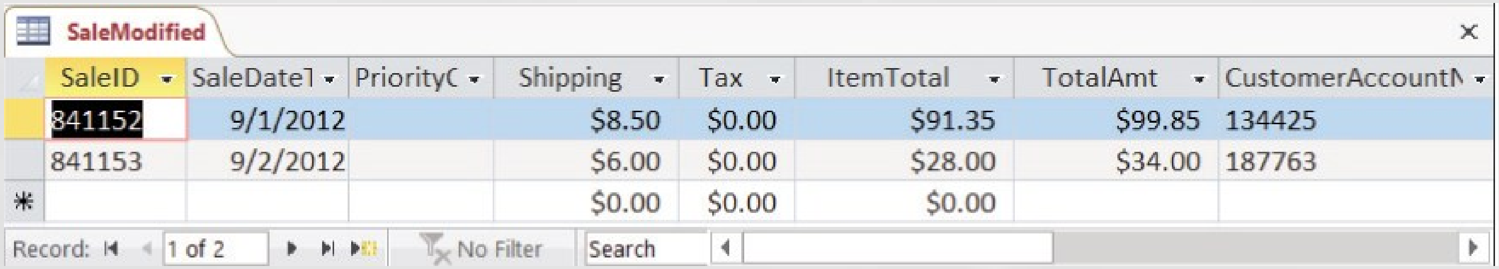
  - Solution is to remove RegularPrice from this table

Third Normal Form (1 of 2)

- A table is in Third Normal Form if it is in 2NF and NO non-key attribute (or set) is functionally dependent on any other non-key attribute (or set)
  - In other words, no FDs among any non-key attributes

Third Normal Form (2 of 2)

- This version of Sale table **violates** 3NF
  - Shipping + Tax +Item Total = TotalAmt
  - i.e., FD: Shipping, Tax, ItemTotal → TotalAmt

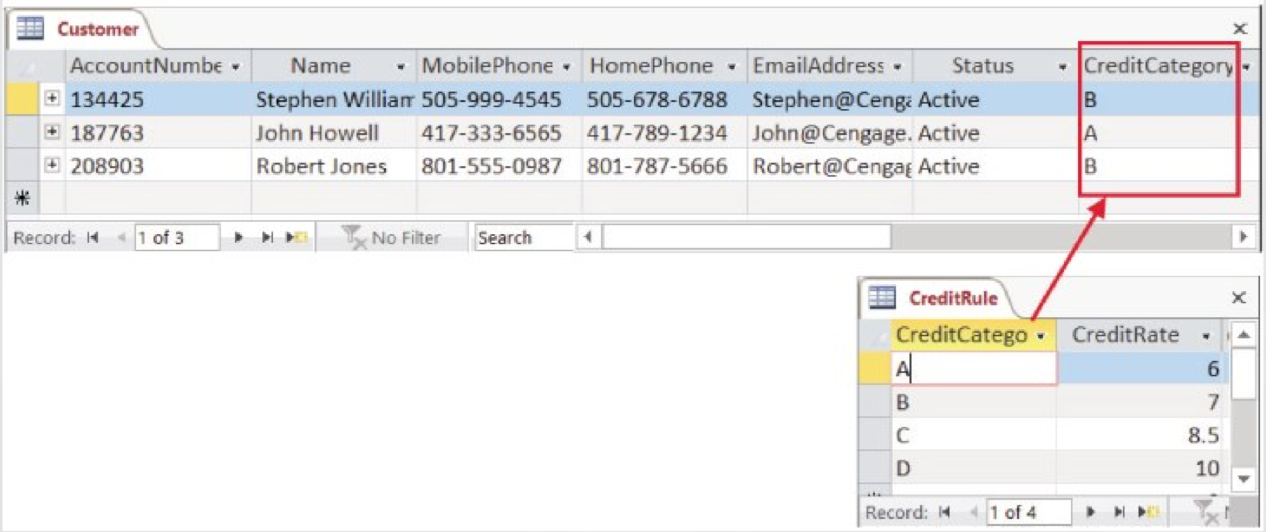


SaleID	SaleDate1	PriorityC	Shipping	Tax	ItemTotal	TotalAmt	CustomerAccountN
841152	9/1/2012		$8.50	$0.00	$91.35	$99.85	134425
841153	9/2/2012		$6.00	$0.00	$28.00	$34.00	187763
*			$0.00	$0.00	$0.00		

- Solution is to remove TotalAmt. It is not needed

Third Normal Form - Solution

- Another solution is to either move offending attribute to a new table.
  - Violation = Customer table had CreditCategory and CreditRate
  - Solution = Make new table of CreditRule with CreditRate



Customer

AccountNumbe	Name	MobilePhone	HomePhone	EmailAddress	Status	CreditCategory
134425	Stephen Williar	505-999-4545	505-678-6788	Stephen@Cenga	Active	B
187763	John Howell	417-333-6565	417-789-1234	John@Cengage.	Active	A
208903	Robert Jones	801-555-0987	801-787-5666	Robert@Cengag	Active	B

CreditRule

CreditCatego	CreditRate
A	6
B	7
C	8.5
D	10

Data Types

- The data type defines the storage format and allowable content of an attribute (field)
- Primitive data types – data types supported directly by the DBMS
- Complex data types – combinations or compositions of primitive data types. User defined

Standard Primitive Data Types

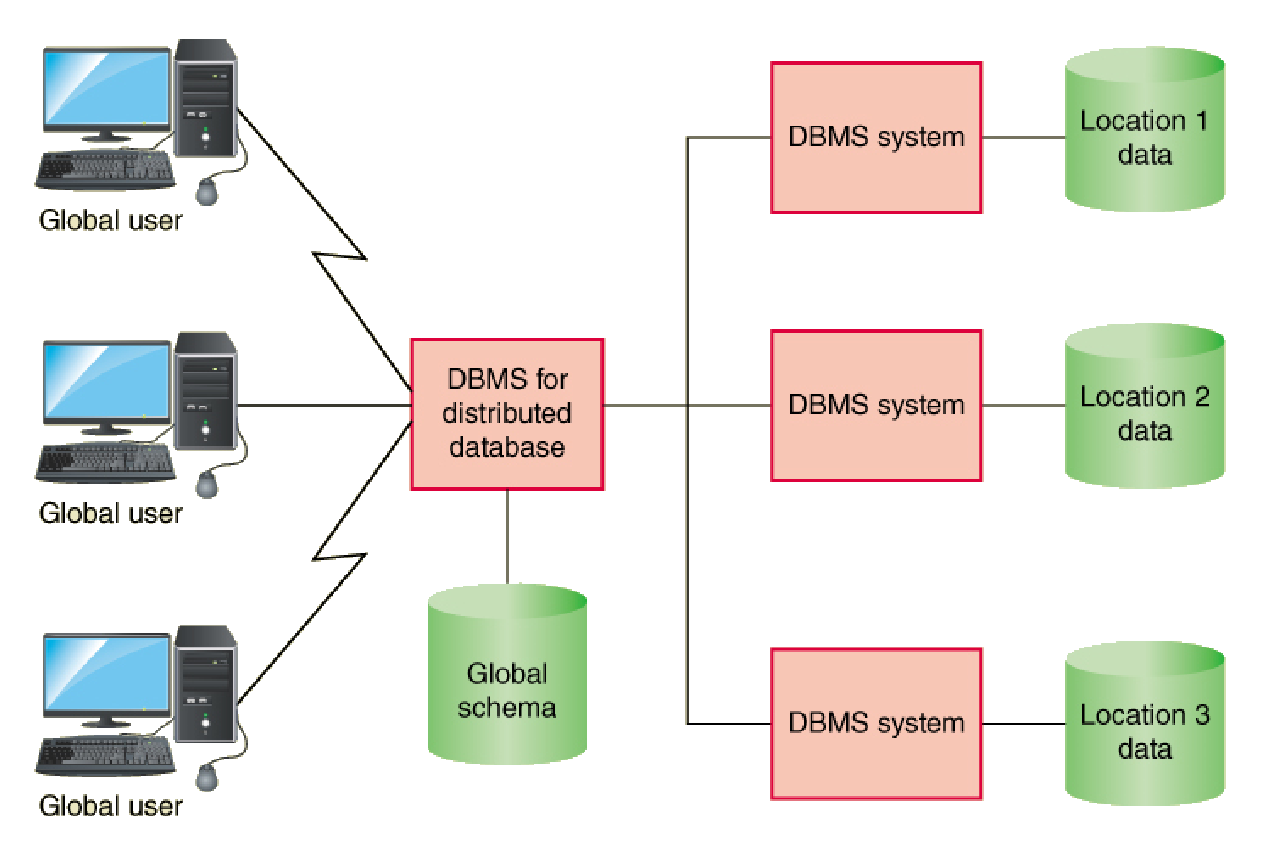
Type(s)	Description
datetimeoffset	Date, time, and time zone
int, small int, and bigint	Whole numeric values
float and real	Numeric values with fractional quantities
money	Currency values and related symbols (e.g., $ and €)
nchar and nvarchar	Fixed- and variable-length Unicode string
varbinary	Variable-length byte sequence up to 2GB
xml	XML document up to 2GB

Distributed Database Architectures

- Decentralized database is stored at many locations but not requiring interconnectivity or synchronization
- Homogeneous distributed database is stored at multiple locations, with all locations using the same DBMS. Coordinated with a global schema
- Heterogeneous distribute database is stored at multiple locations and with different DBMS and may have local schemas.

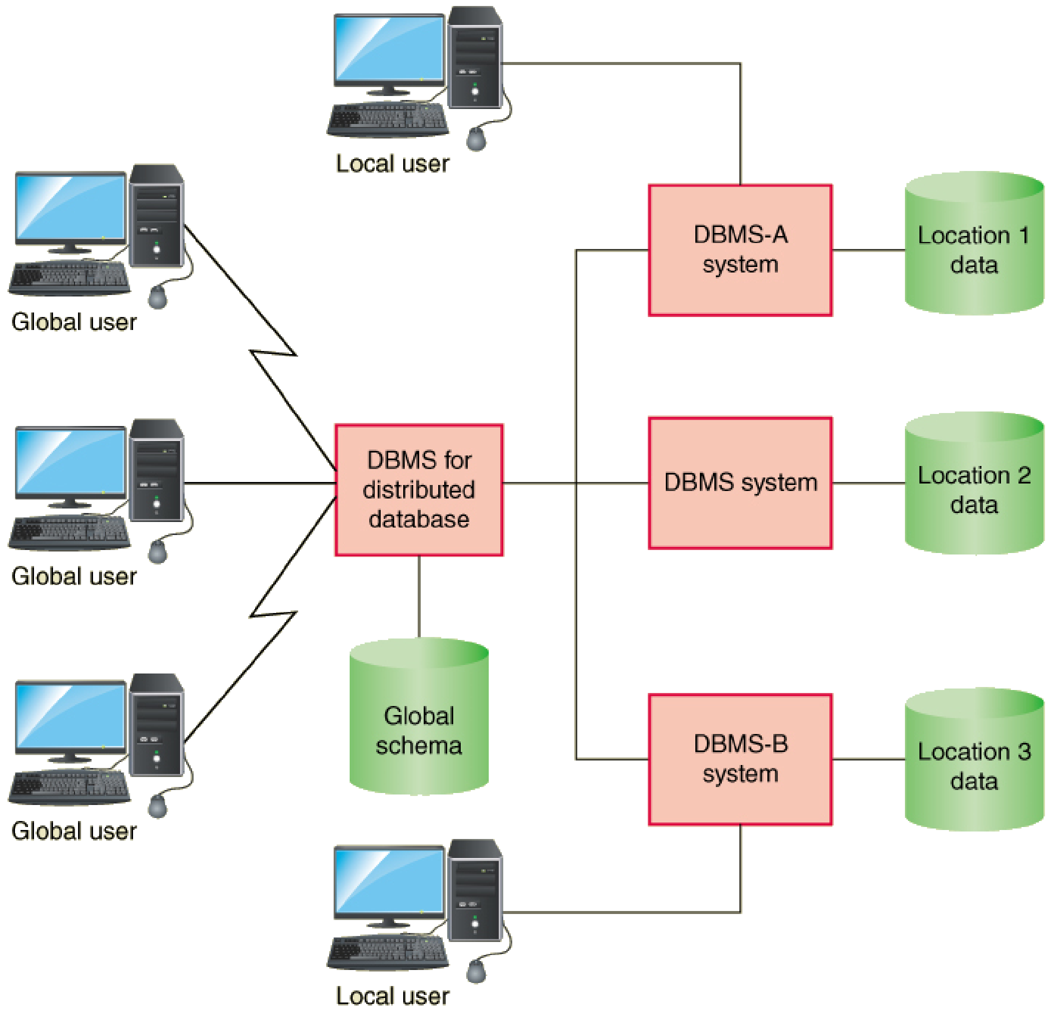
Homogeneous Distributed Database

- Access is through a common DBMS and schema



Heterogeneous Distributed Database

- Access is through distinct DBMSs. May have global and local schemas in operation

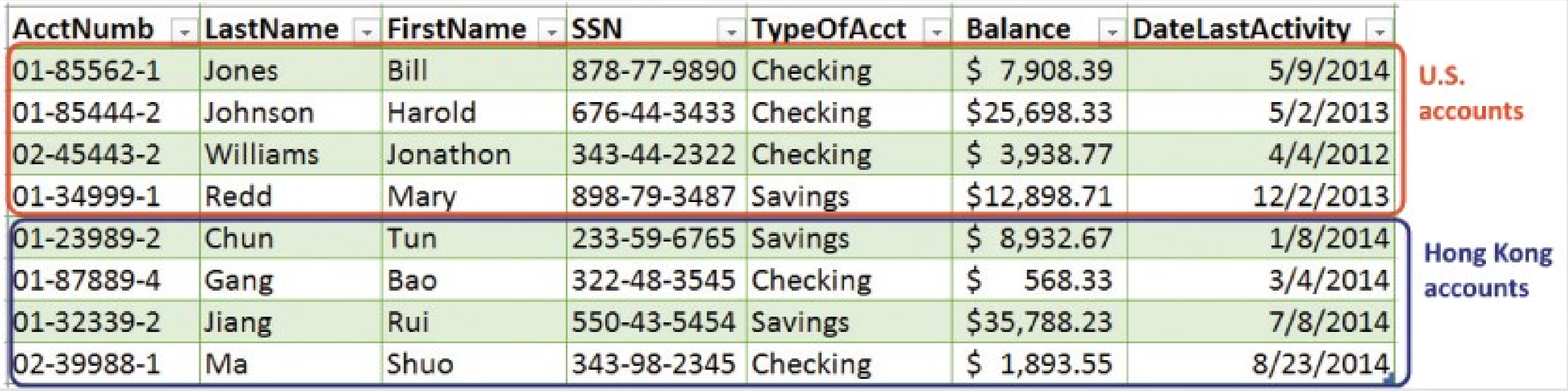


Implementation Approaches (1 of 3)

- Data replication – each location has its own copy
    - Synchronization – updating every copy with changes made to every other copy

Implementation Approaches (2 of 3)

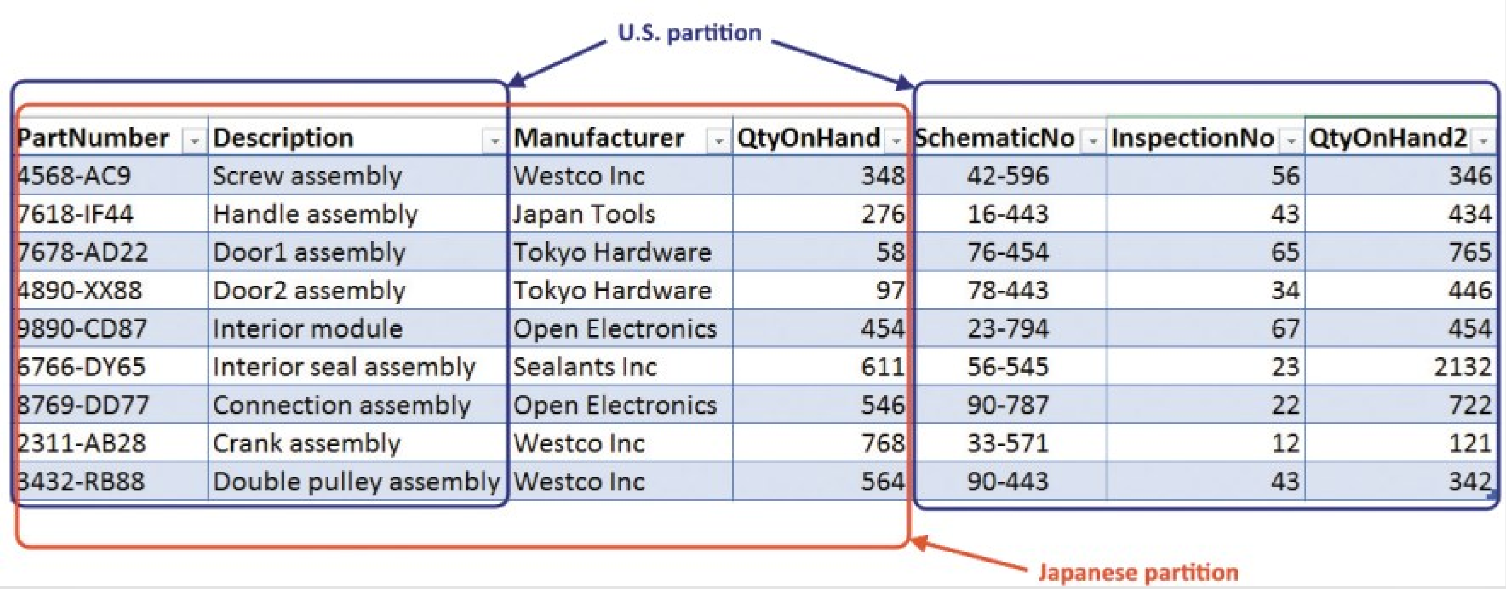
- Horizontal Partition – different rows are stored at different locations.



AcctNumb	LastName	FirstName	SSN	TypeOfAcct	Balance	DateLastActivity
01-85562-1	Jones	Bill	878-77-9890	Checking	$ 7,908.39	5/9/2014
01-85444-2	Johnson	Harold	676-44-3433	Checking	$25,698.33	5/2/2013
02-45443-2	Williams	Jonathon	343-44-2322	Checking	$ 3,938.77	4/4/2012
01-34999-1	Redd	Mary	898-79-3487	Savings	$12,898.71	12/2/2013
01-23989-2	Chun	Tun	233-59-6765	Savings	$ 8,932.67	1/8/2014
01-87889-4	Gang	Bao	322-48-3545	Checking	$ 568.33	3/4/2014
01-32339-2	Jiang	Rui	550-43-5454	Savings	$35,788.23	7/8/2014
02-39988-1	Ma	Shuo	343-98-2345	Checking	$ 1,893.55	8/23/2014

Implementation Approaches (3 of 3)

- Vertical Partition – Different columns are stored at different locations.

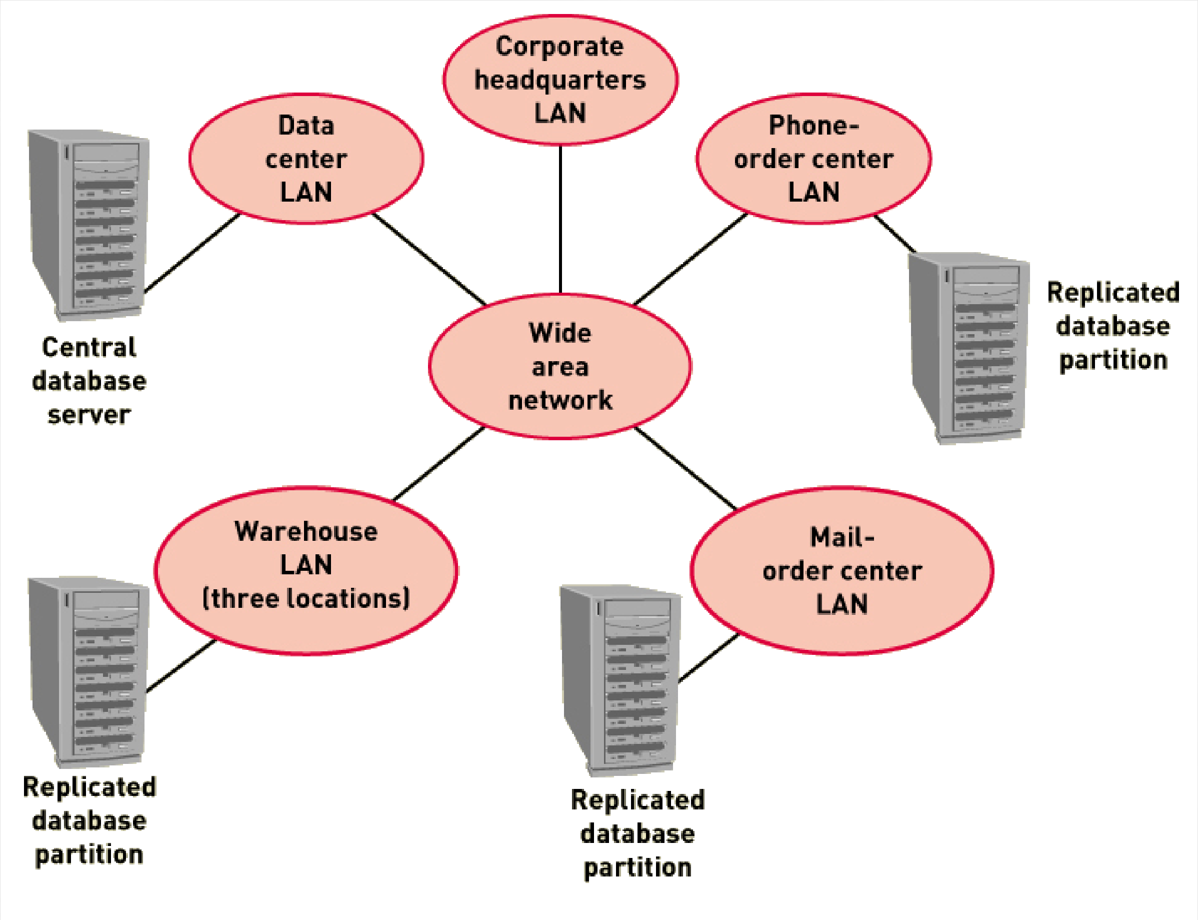


U.S. partition
Japanese partition

PartNumber	Description	Manufacturer	QtyOnHand	SchematicNo	InspectionNo	QtyOnHand2
4568-AC9	Screw assembly	Westco Inc	348	42-596	56	346
7618-IF44	Handle assembly	Japan Tools	276	16-443	43	434
7678-AD22	Door1 assembly	Tokyo Hardware	58	76-454	65	765
4890-XX88	Door2 assembly	Tokyo Hardware	97	78-443	34	446
9890-CD87	Interior module	Open Electronics	454	23-794	67	454
6766-DY65	Interior seal assembly	Sealants Inc	611	56-545	23	2132
8769-DD77	Connection assembly	Open Electronics	546	90-787	22	722
2311-AB28	Crank assembly	Westco Inc	768	33-571	12	121
3432-RB88	Double pulley assembly	Westco Inc	564	90-443	43	342

- Combinations of replication, horizontal, and vertical

Architecture for RMO: Replicated and Partitioned Database


Corporate headquarters LAN
Data center LAN
Phone-order center LAN
Central database server
Wide area network
Replicated database partition
Warehouse LAN (three locations)
Mail-order center LAN
Replicated database partition
Replicated database partition

Protecting the Database (1 of 2)

- Transaction Logging – a technique to record all updates including change, date, time, user
  - Helps to prevent fraud
  - Recovery mechanism for failures

Protecting the Database (2 of 2)

- Concurrency and Update Controls
  - Transaction – a piece of work with several steps, either all must complete or none must be accepted
  - Database lock – technique to apply exclusive control to a portion of the database to one user at a time
  - Shared or read lock – a lock where multiple transactions (users) may read the data
  - Exclusive or write lock – a lock where only one transaction (user) may access the locked portion of the database

Summary

- Most modern information systems store data and access data using a database management systems (DBMS)
- The most common database model is a relational database (RDBMS), which is a collection of data stored in tables
- The relational database schema is developed based on the domain model class diagram Each class is represented as a table. One to many associations are represented by adding foreign keys
- Normalization is the process to produce high-quality databases without update, insertion or delete anomalies
- Distributed databases are necessary for very large databases
- Database locks permit concurrent use of databases