INTRODUCTION TO SYSTEMS ANALYSIS AND DESIGN:

AN AGILE, ITERATIVE APPROACH

SATZINGER | JACKSON | BURD

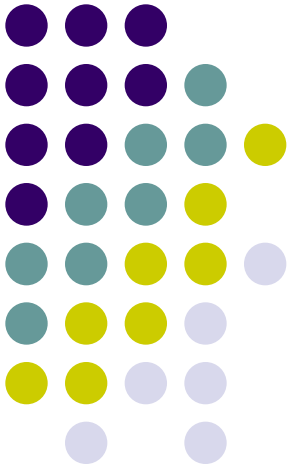
Chapter 12

Databases, Controls, and Security

Chapter 12

Introduction to Systems Analysis and Design:
An Agile, Iteractive Approach
6th Ed

Satzinger, Jackson & Burd



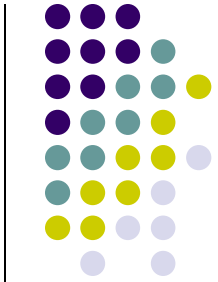
Chapter 12 Outline

- Databases and Database Management Systems (DBMS)
- Relational Databases (RDBMS)
- Data Access Classes
- Distributed Database Architectures
- Database Design Timing and Risks
- Designing Integrity Controls
- Designing Security Controls

Learning Objectives

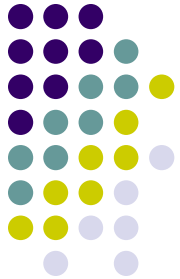
- Design a relational database schema based on a class diagram
- Evaluate and improve the quality of a database schema
- Describe the different architectural models for distributed databases
- Determine when and how to design the database
- Explain the importance of integrity controls for inputs, outputs, data, and processing
- Discuss issues related to security that affect the design and operation of information systems

Overview



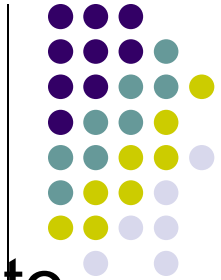
- Databases and database management systems are important components of a modern information system
- Database design transforms the domain model class diagram into a detailed database model for the system
- A database management system is used to implement and interact with the database
- System controls and security are crucial issues to databases and also apply to other aspects of the system

Some Database Concepts



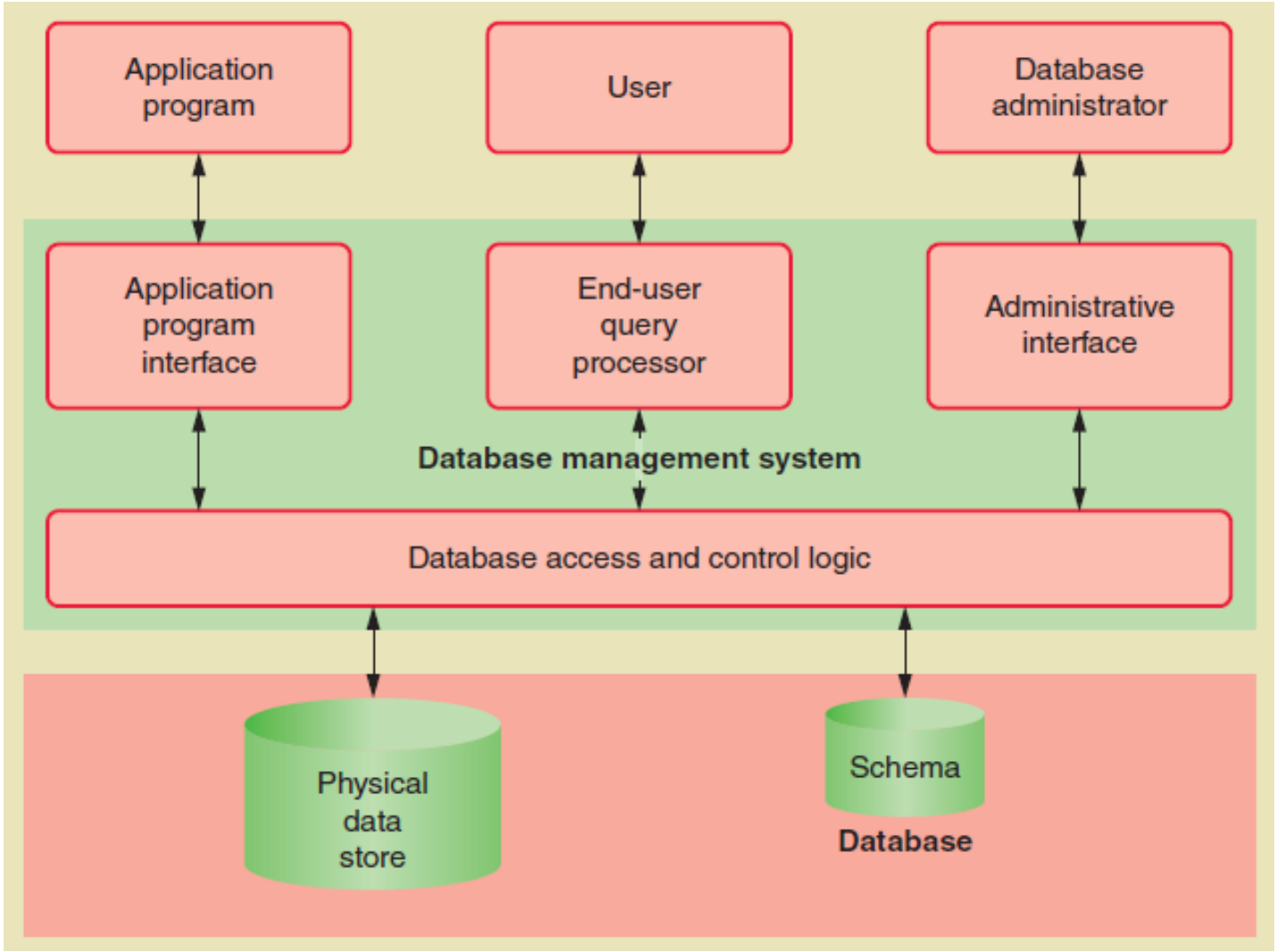
- Database (DB) -- an integrated collection of stored data that is centrally managed and controlled
- Database management system (DBMS) -- a system software component that manages and controls one or more databases
- Physical data store -- database component that stores the raw bits and bytes of data
- Schema -- database component that contains descriptive information about the data stored in the physical data store

Database Schema

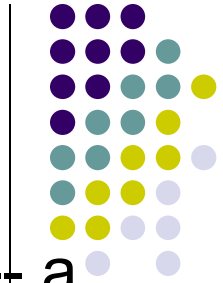


- Organization of individual stored data items into higher level groups, such as tables
- Associations among tables or classes
- Details of physical data store organization, including types, lengths, locations, and indexing of data items
- Access and content controls, including allowable values for specific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

Database and DBMS Components

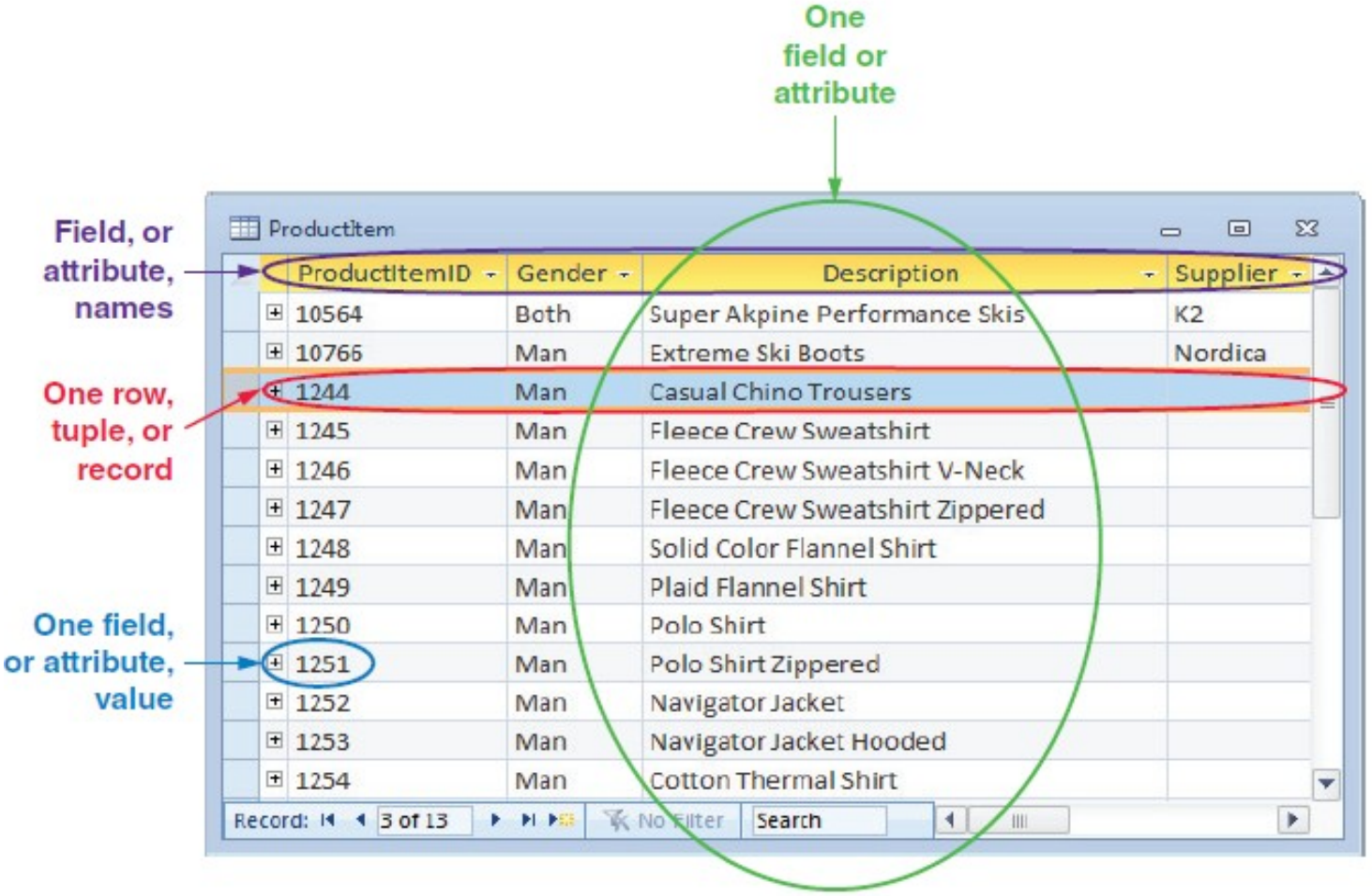


Relational Databases

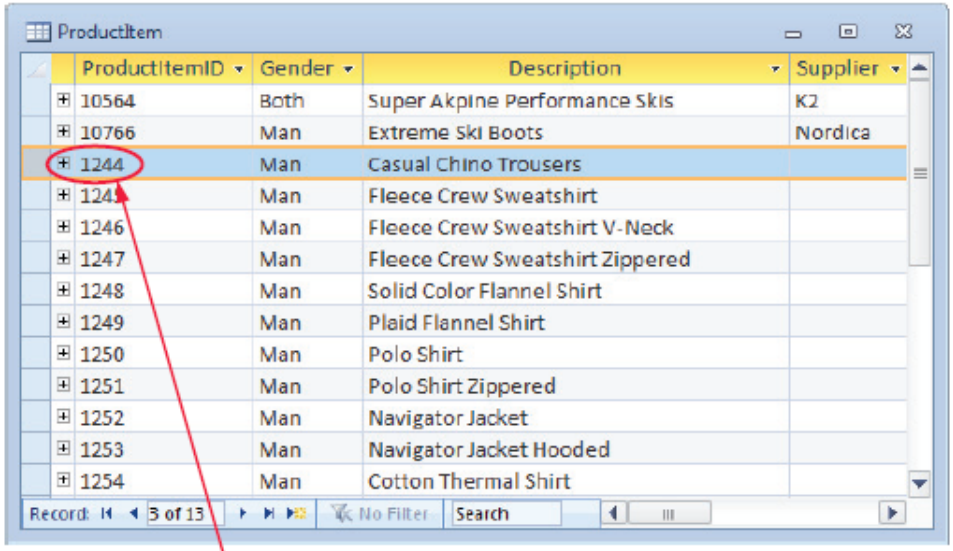


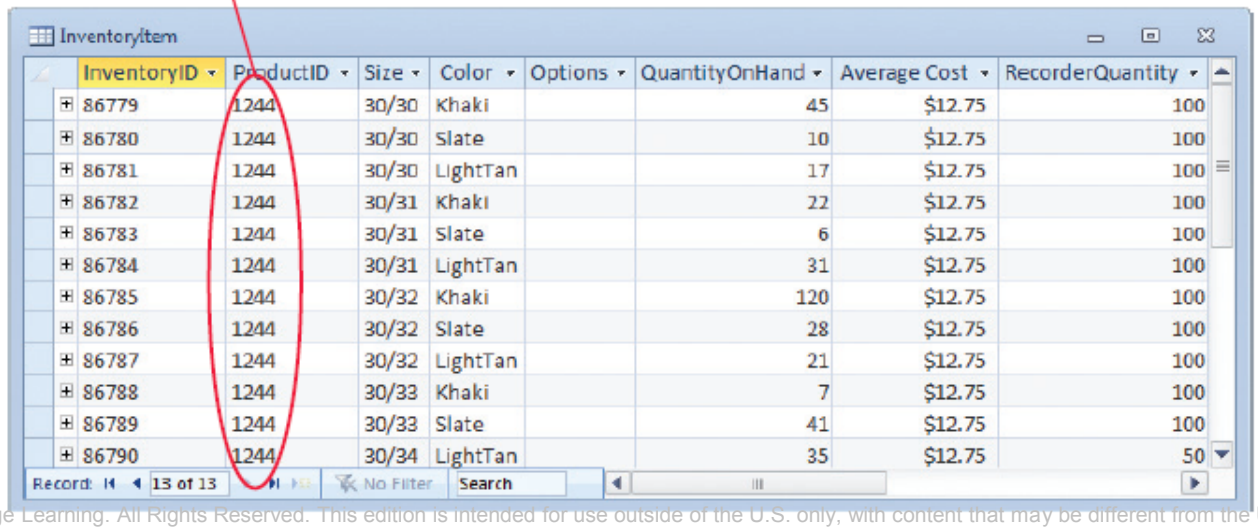
- Relational database management system (RDBMS) -- a DBMS that organizes data in tables (relations)
- Table -- a two-dimensional data structure of columns and rows
- Row -- one horizontal group of data attribute values
- Attribute -- one vertical group of data attribute values
- Attribute value -- the value held in a single table cell
- Key -- an attribute or set of attributes, the values of which occur only once in all the rows of the table
- 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

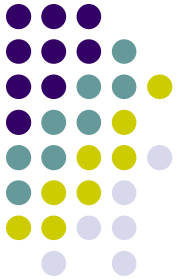


An Association Between Rows in Two Tables (key and foreign key)





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

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

Initial Set of Tables

With Foreign Key Attributes Added (in italics)

Table	Attributes
AccessoryPackage	**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

Final Set of Tables

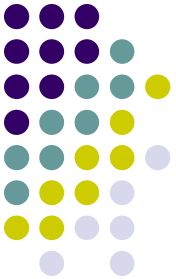
Specialized subclasses of Sale and Online Cart added

Promo Offering modified from association class to table with two keys

Table	Attributes
AccessoryPackage	**AccessoryPackageID**, Category, Description
AccessoryPackageContents	***AccessoryPackageID, InventoryItemID***
ActiveCart	***OnlineCartID***, ElapsedTime
CartItem	***InventoryItemID***, ***OnlineCartID***, Quantity, CurrentPrice
Customer	**AccountNumber**, Name, MobilePhone, HomePhone, EmailAddress, Status
InStoreSale	***SaleID***, StoreID, RegisterID, ClerkID
InventoryItem	**InventoryItemID**, *ProductItemID*, Size, Color, Options, QuantityOnHand, AverageCost, ReorderQuantity
OnlineCart	**OnlineCartID**, *CustomerAccountNumber*, StartDateTime, NumberOfItems, ValueOfItems, Status, ElapsedTime, HoldForDays
OnlineSale	***SaleID***, TimeOnSite, ChatUse
OnReserveCart	***OnlineCartID***, 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
SaleItem	***InventoryItemID***, ***SaleID***, Quantity, SoldPrice, ShipStatus, BackOrderStatus
SaleTransaction	**SaleTransactionID**, *SaleID*, Date, TransactionType, Amount, PaymentMethod
TelephoneSale	***SaleID***, ClerkID, LengthOfCall

Designing Relational Databases

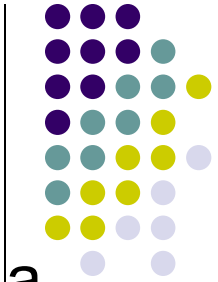
Referential Integrity and Schema Quality



- 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
- A high-quality relational database schema has these features:
  - Flexibility or ease of implementing future data model changes
  - Lack of redundant data
- Normalization -- a formal technique for evaluating and improving the quality of a relational database schema

Designing Relational Databases

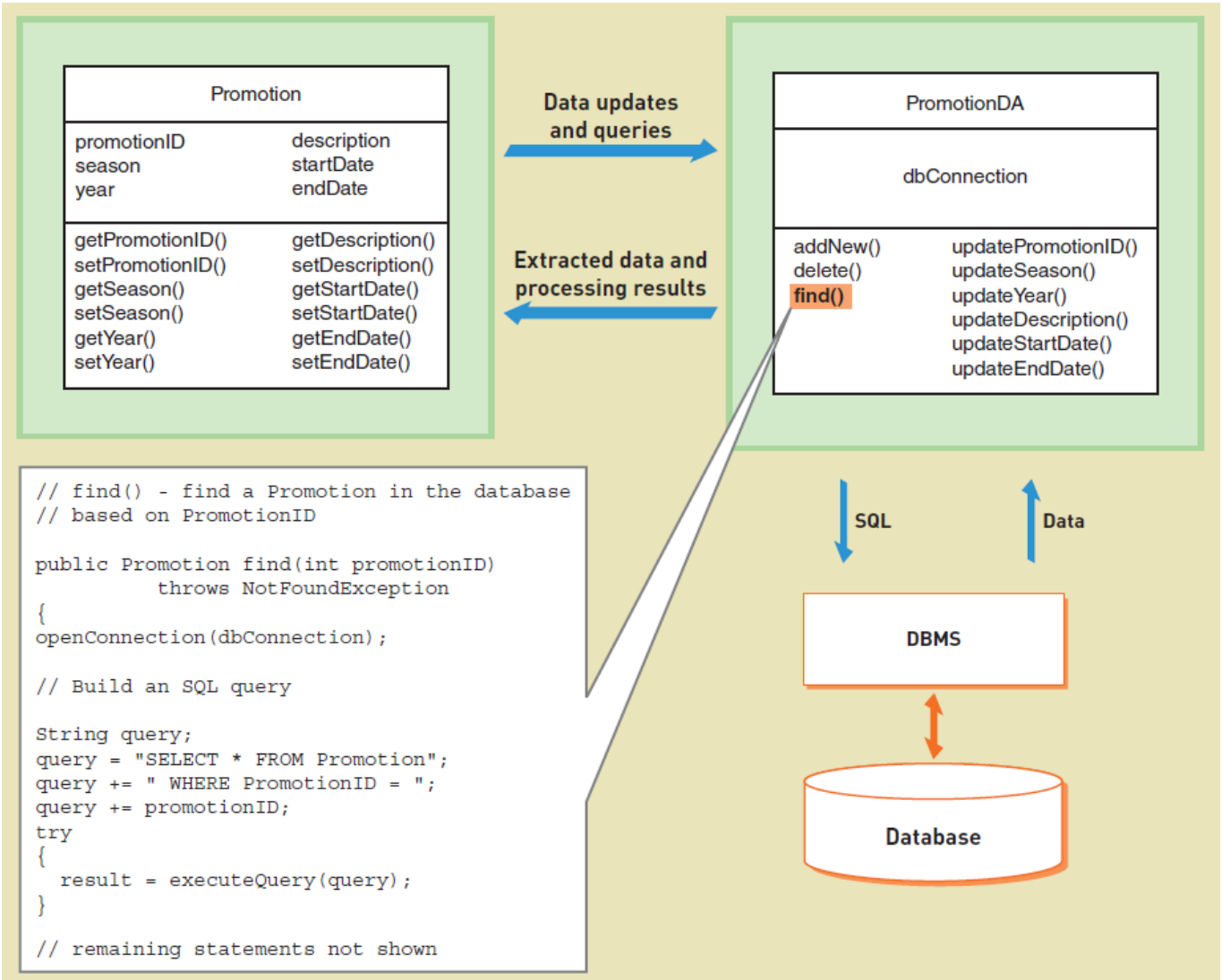
Database Normalization



- First normal form (1NF) -- restriction that all rows of a table must contain the same number of columns
  - No repeating groups of attributes
- Functional dependency -- a one-to-one association between the values of two attributes
  - *Attribute A* is functionally dependent on *attribute B* if for each value of *attribute B* there is only one corresponding value of *attribute A*
- Second normal form (2NF) -- restriction that a table is in 1NF and that each non-key attribute is functionally dependent on the entire primary key
- Third normal form (3NF) -- restriction that a table is in 2NF and that no non-key attribute is functionally dependent on any other non-key attribute

Data Access Classes

Map design classes to RDBMS tables

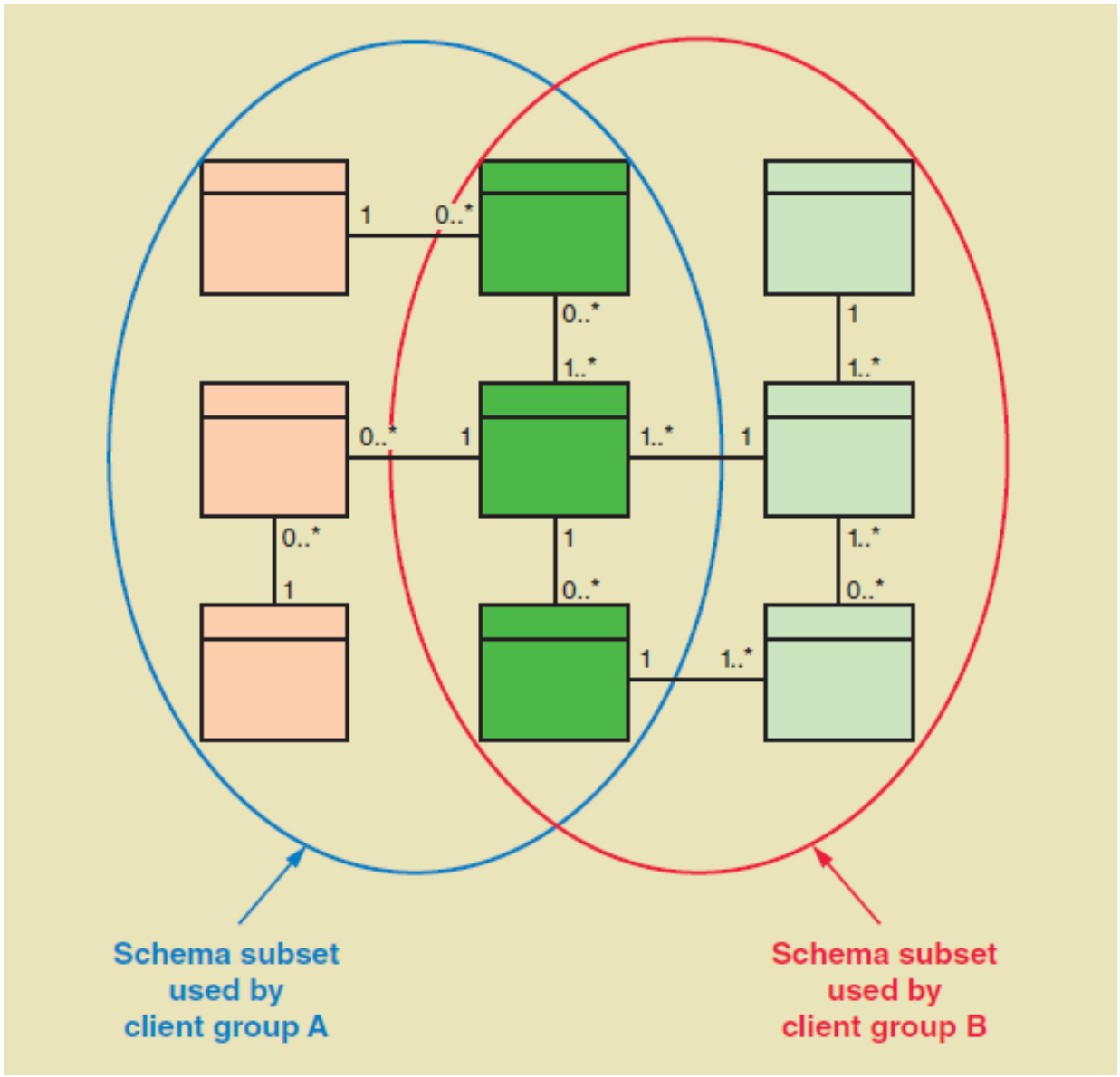


Distributed Database Architectures

- Single database server architecture -- one or more databases are hosted by a single DBMS running on a single server
- Replicated database server architecture -- complete database copies are hosted by cooperating DBMSs running on multiple servers
- Partitioned database server architecture -- multiple distributed database servers are used and the database schema is partitioned
- Cloud-based database server architecture -- use of a cloud computing service provider to provide some or all database services

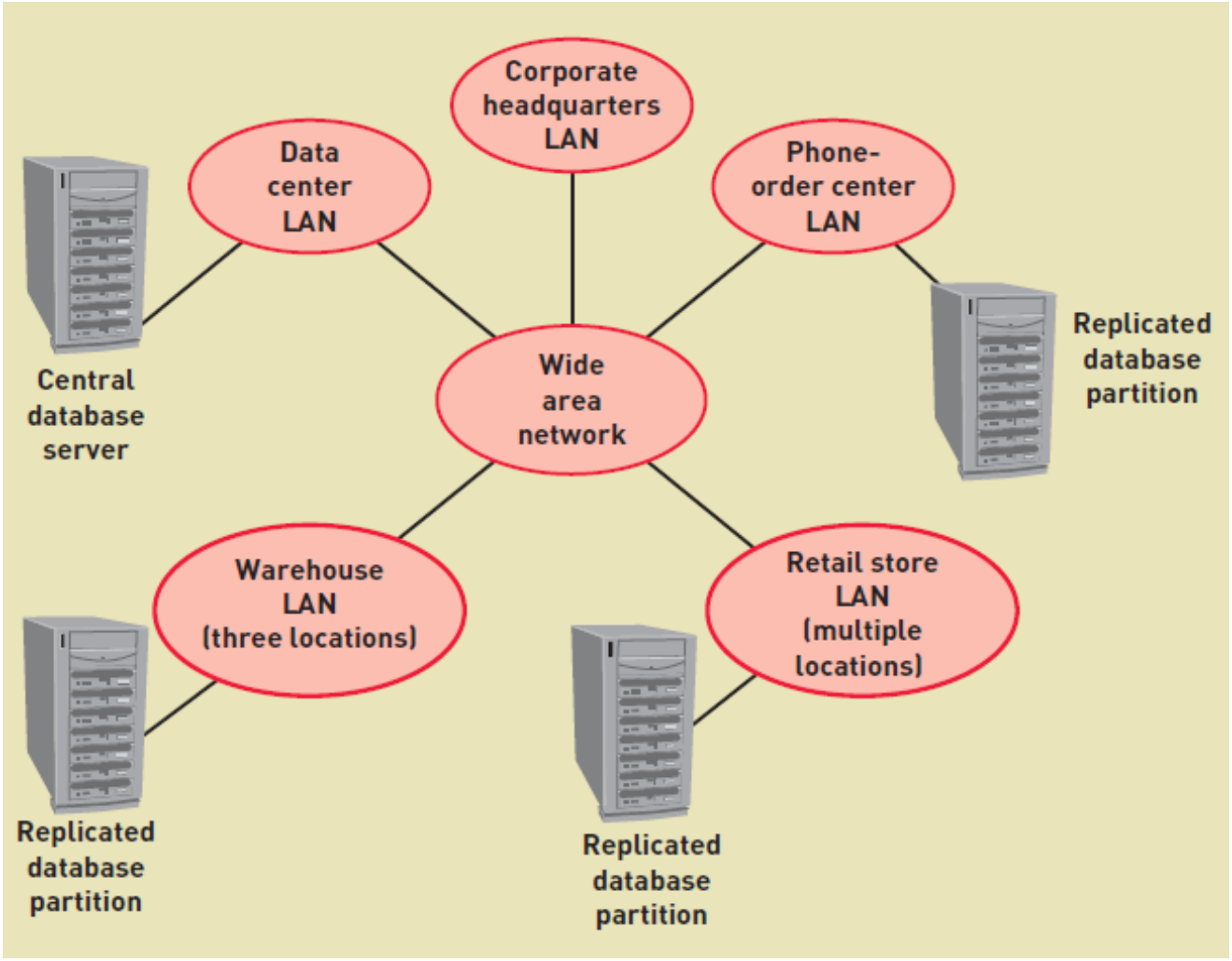
Partitioning Database Schema

Into Client Access Subsets

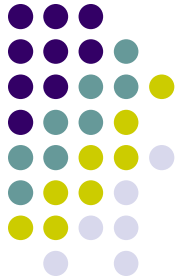


Architecture for RMO

Replicated and Partitioned Database

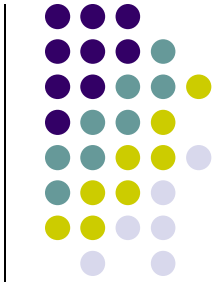


Database Design Timing and Risks



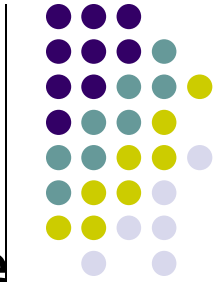
- Architecture—Decisions about DBMS, database servers, and database distribution are tightly integrated with other architectural decisions, including network design, Web and component services, and security.
- Existing databases—Most new or upgraded systems must interact with existing databases, with their pre-existing constraints. While adapting existing databases to new or updated systems, analysts must ensure their continued operation.
- Domain model class diagram—Database design can't proceed until related parts of the class diagram have been developed.

The Need for System Controls



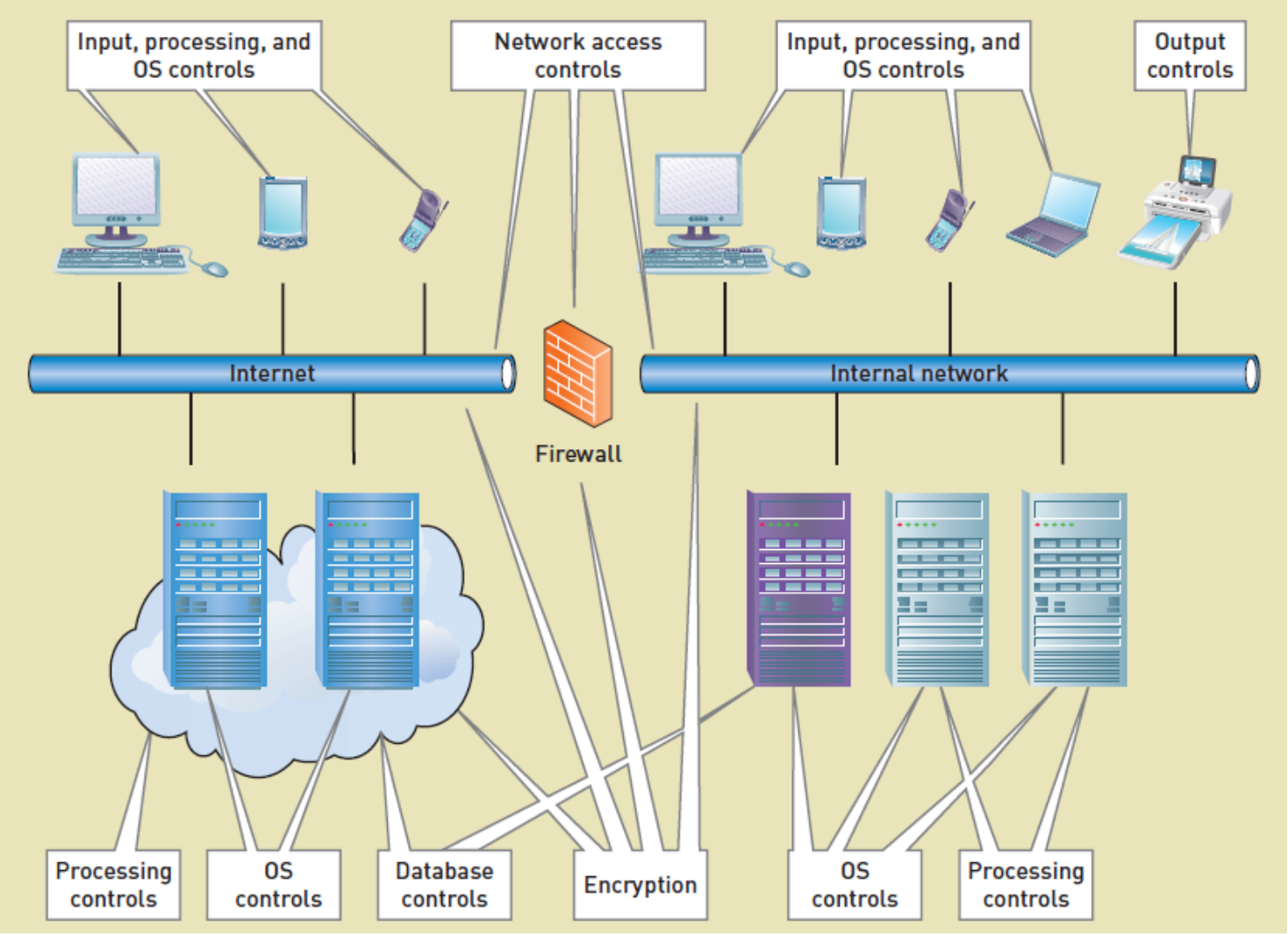
- A furniture store sells merchandise on credit with internal financing. Salespeople sometimes sell furniture on credit to friends and relatives. How do we ensure that only authorized employees can extend credit and record payments and adjustments to credit accounts?
- A bookkeeper uses accounting software to generate electronic payments to suppliers. How does the system ensure that the payment is for goods or services that were actually received? How does the system ensure that no one can generate payments to a bogus supplier?
- An online retailer collects and stores credit card and other information about customers. How does the company ensure that customer data is protected and secure?

Designing System Controls

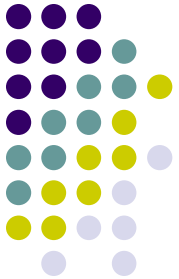


- Controls -- mechanisms and procedures that are built into a system to safeguard the system and the information within it
- Integrity control -- a control that rejects invalid data inputs, prevents unauthorized data outputs, and protects data and programs against accidental or malicious tampering
- Security controls -- are part of the operating system and the network and tend to be less application specific.
- There is some overlap between Integrity and Security controls

Integrity and Security Controls



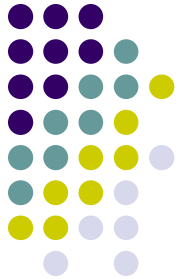
Integrity Controls
Input Controls



- Input control -- a control that prevents invalid or erroneous data from entering the system
- value limit control -- a control that checks numeric data input to ensure that the value is reasonable
- completeness control -- a control that ensures that all required data values describing an object or transaction are present
- data validation control -- a control that ensures that numeric fields that contain codes or identifiers are correct
- field combination control -- a control that reviews combinations of data inputs to ensure that the correct data are entered

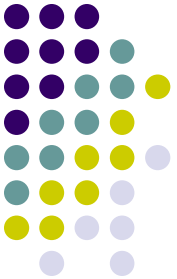
Integrity Controls

Access controls, Transaction logging, Complex update controls, Output controls



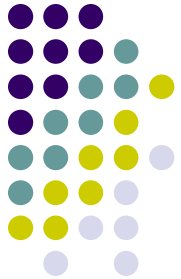
- Access control -- a control that restricts which persons or programs can add, modify, or view information resources
- Transaction logging -- a technique by which any update to the database is logged with such audit information as user ID, date, time, input data, and type of update
- Complex update control -- a control that prevents errors that can occur when multiple programs try to update the same data at the same time or when recording a single transaction requires multiple related database updates
- Output control -- a control that ensures that output arrives at the proper destination and is accurate, current, and complete

Integrity Controls
Redundancy, Backup, and Recovery



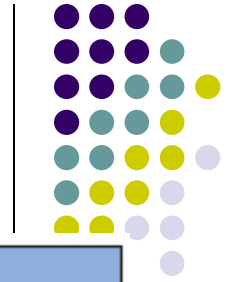
- Designed to protect data from hardware failure and catastrophes
- Redundancy – continuous access to data through redundant databases, servers, and sites
- Backup – procedures make partial or full copies of a database to removable storage media, such as magnetic tape, or to data storage devices or servers at another site
- Recovery – procedures read the off-site copies and replicate their contents to a database server that can then provide access to programs and users.

Integrity Controls
To Prevent Fraud



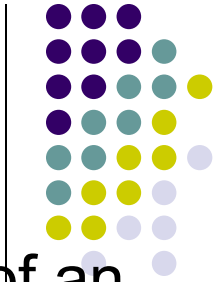
- Fraud triangle -- model of fraud that states that opportunity, motivation, and rationalization must all exist for a fraud to occur
  - Opportunity—the ability of a person to take actions that perpetrate a fraud. For example, unrestricted access to all functions of an accounts payable system enables an employee to generate false vendor payments.
  - Motivation—a desire or need for the results of the fraud. Money is the usual motivation, although a desire for status or power as well as a need to be a "team player" may be contributing factors.
  - Rationalization—an excuse for committing the fraud or an intention to "undo" the fraud in the future. For example, an employee might falsify financial reports to stave off bankruptcy, thus enabling fellow workers to keep their jobs.

Integrity Controls
To Prevent Fraud



Factors affecting fraud risk	Risk-reduction techniques
Separation of duties	Design systems so those with asset custody have limited access to related records. Also, ensure that no one has sufficient system access to commit and cover up a fraud.
Records and audit trails	Record all transactions and changes in asset status. Log all changes to records and databases, and restrict log access to a few trusted persons.
Monitoring	Incorporate regular and systematic procedures to review records and logs for unusual transactions, accesses, and other patterns.
Asset control and reconciliation	Limit physical access to valuable assets, such as inventory, and periodically reconcile physical asset counts with related records.
Security	Design security features into individual systems and supporting infrastructure. Review and test security features frequently. Use outside consultants to conduct penetration testing attack and fraud vectors from external and internal sources.

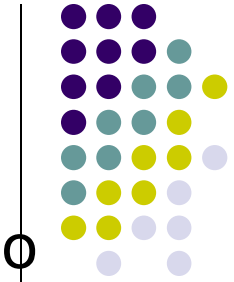
Designing Security Controls



- Security control -- a control that protects the assets of an organization from all threats, with a primary focus on external threats
- Two Objectives
  - Maintain a stable, functioning operating environment for users and application systems (usually 24 hours a day, 7 days a week).
    - Firewalls to protect from hackers, viruses, works, and denial of service attacks
  - Protect information and transactions during transmission across the Internet and other insecure environments
    - Information could be intercepted, destroyed or modified

Security Controls

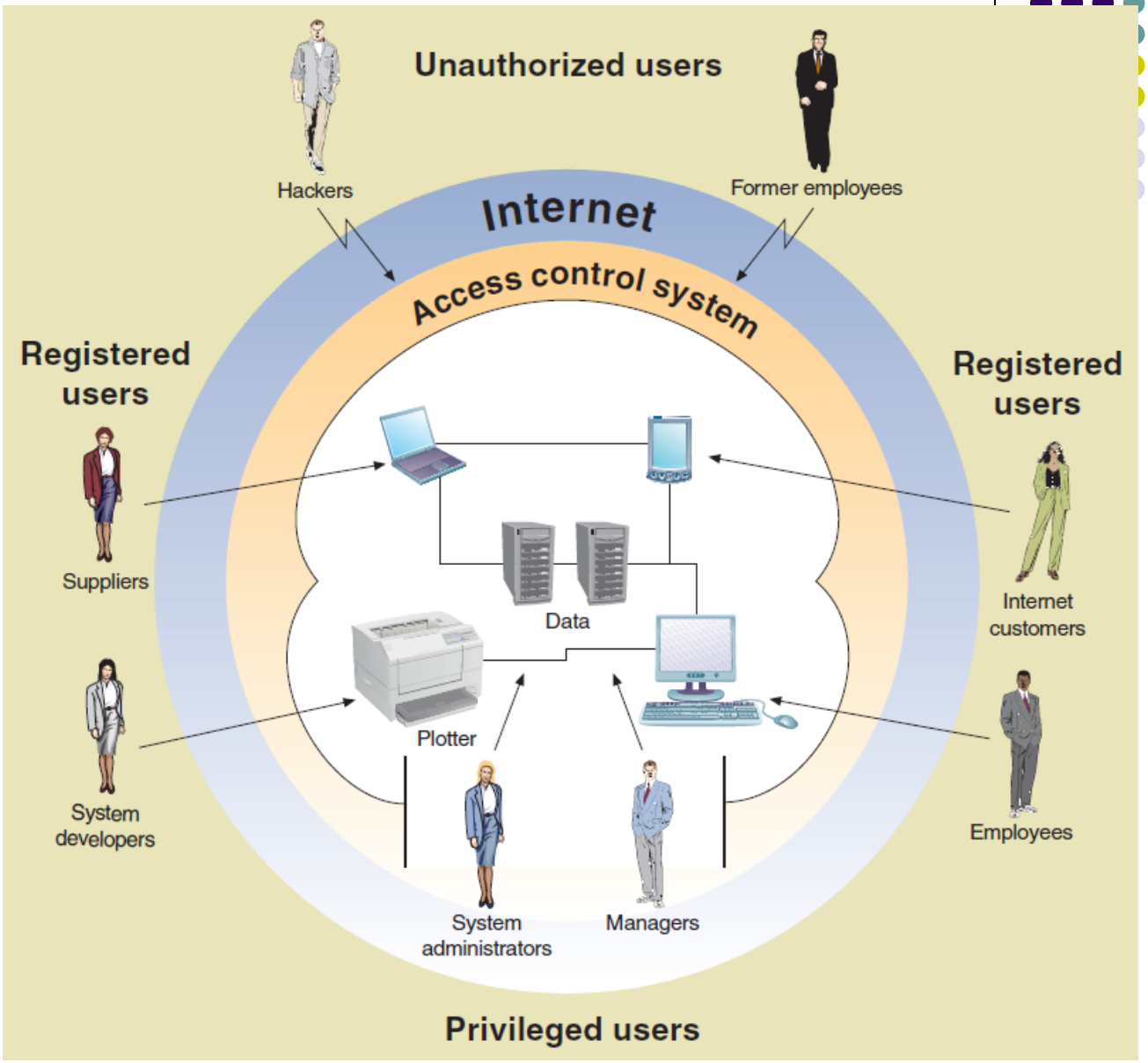
Access Controls



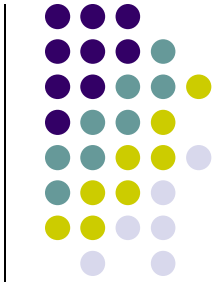
- Authentication -- the process of identifying users who request access to sensitive resources
- Authorization -- the process of allowing or restricting a specific authenticated user's access to a specific resource based on an access control list
- Multifactor authentication -- using multiple authentication methods for increased reliability
- Unauthorized user -- a person who isn't allowed access to any part or functions of the system
- Registered user -- a person who is authorized to access
- Privileged user -- a person who has access to the source code, executable program, and database structure of the system

Security Controls

Access Controls

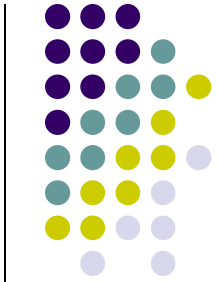


Security Controls
Data Encryption



- Common types of data requiring additional protection
  - Financial information
  - Credit card numbers, bank account numbers, payroll information, healthcare information, and other personal data
  - Strategies and plans for products and other mission-critical data
  - Government and sensitive military information
  - Data stored on such portable devices as laptop computers and cell phones

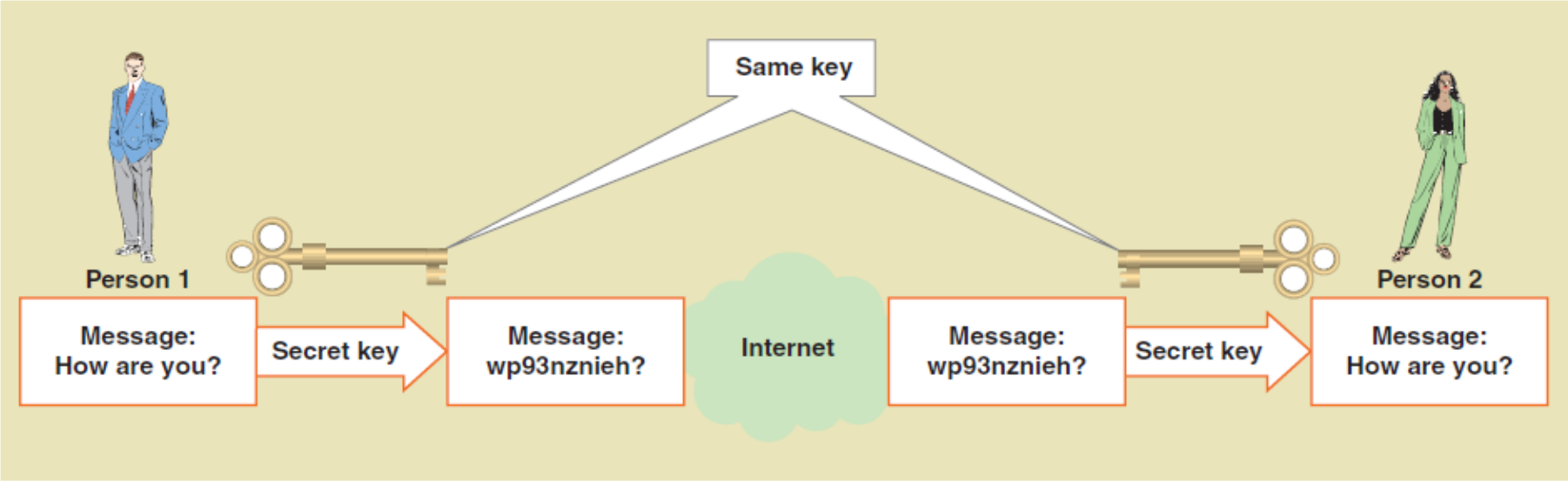
Security Controls
Data Encryption



- Encryption -- the process of altering data so unauthorized users can't view them
- Decryption -- the process of converting encrypted data back to their original state
- Encryption algorithm -- a complex mathematical transformation that encrypts or decrypts binary data
- Encryption key -- a binary input to the encryption algorithm—typically a long string of bits
- Symmetric key encryption -- encryption method that uses the same key to encrypt and decrypt the data

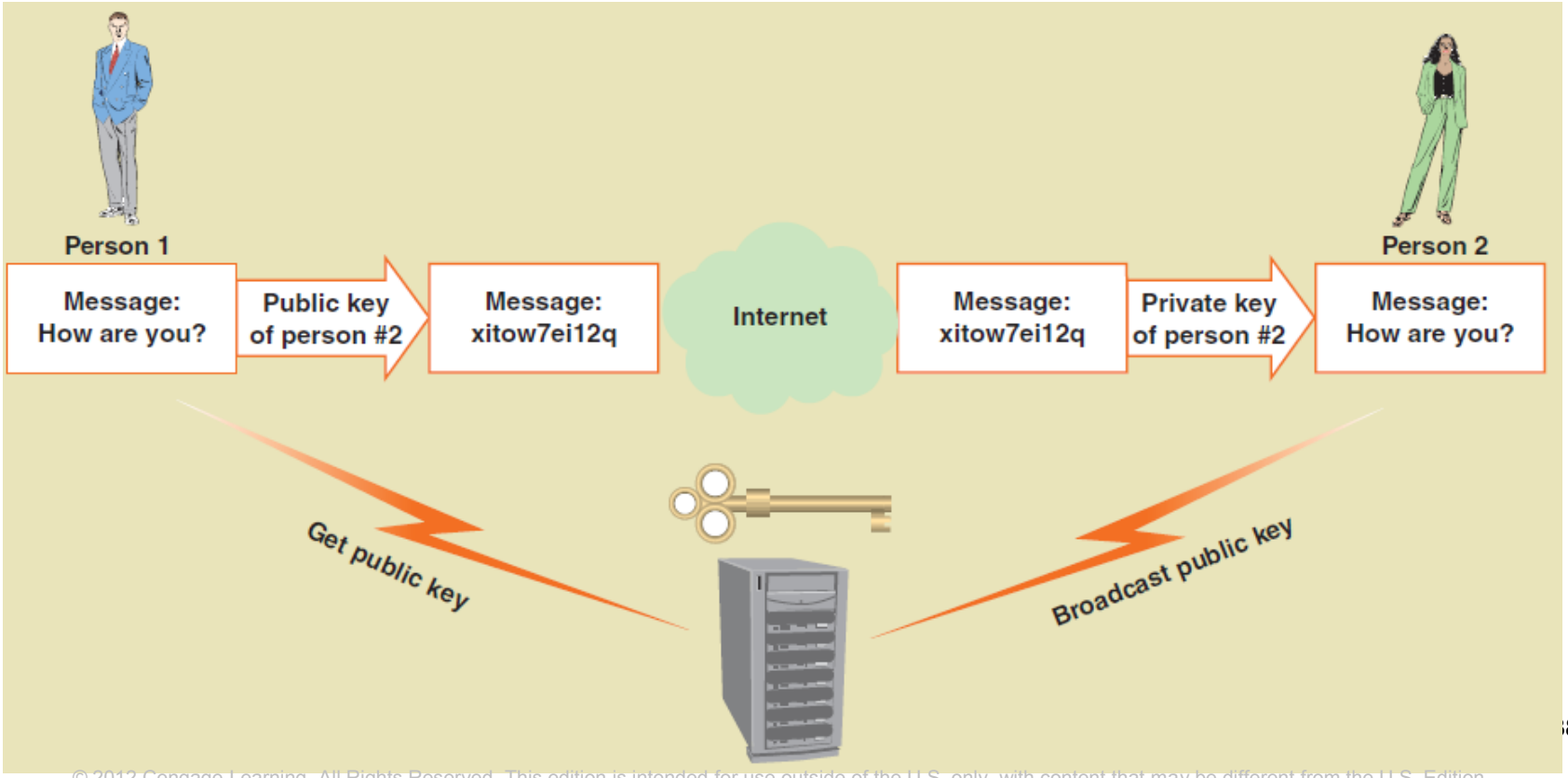
Security Controls
Data Encryption

- Symmetric Key Encryption --



Security Controls
Data Encryption

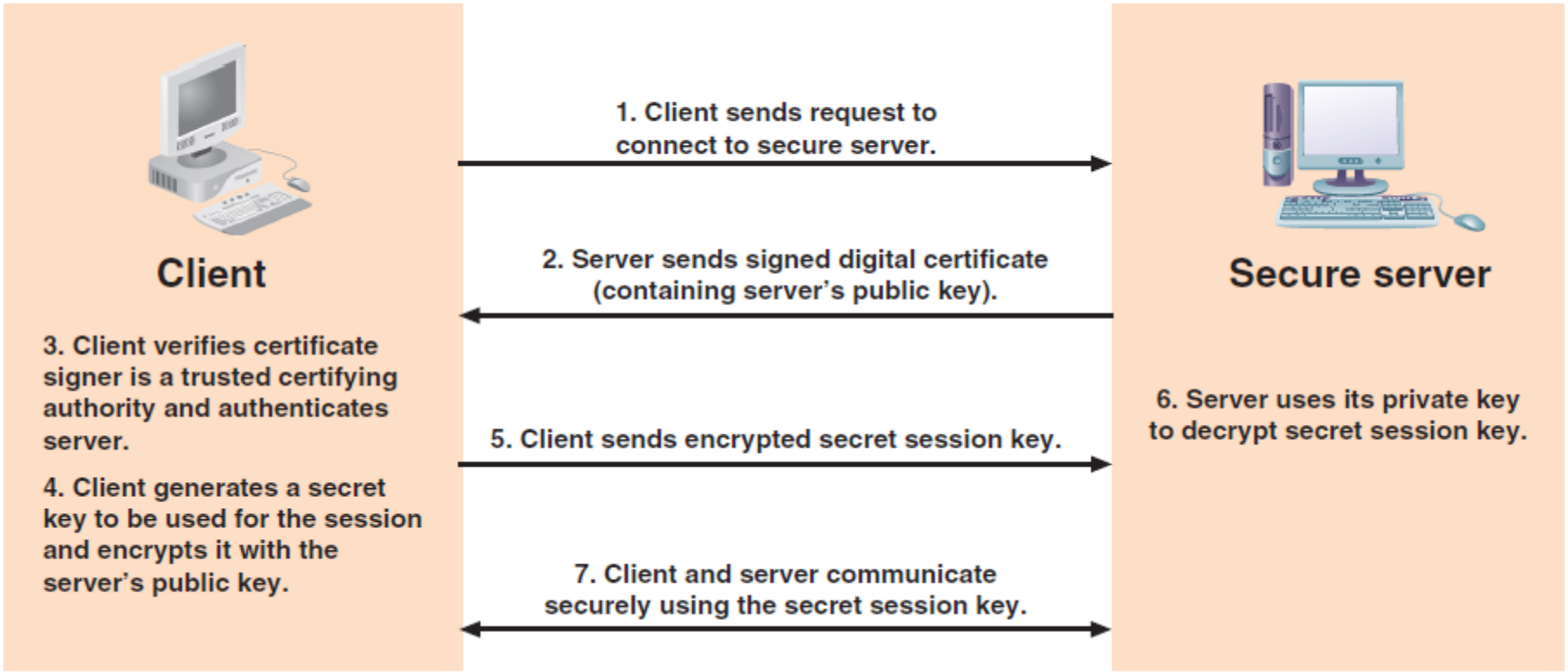
- Asymmetric key encryption -- encryption method that uses different keys to encrypt and decrypt the data
- Public key encryption -- a form of asymmetric key encryption that uses a public key for encryption and a private key for decryption



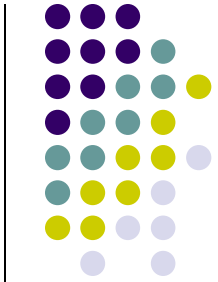
Security Controls

Digital Certificate

- Digital certificate -- an institution's name and public key (plus other information, such as address, Web site URL, and validity date of the certificate) encrypted and certified by a third party
- Certifying authority -- a widely accepted issuer of digital certificates

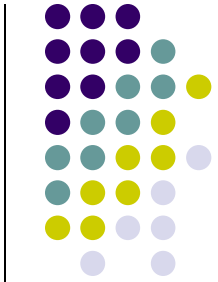


Security Controls
Secure Transactions



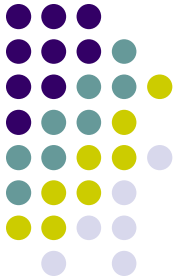
- Secure Sockets Layer (SSL) -- a standard set of methods and protocols that address authentication, authorization, privacy, and integrity
- Transport Layer Security (TLS) -- an Internet standard equivalent to SSL
- IP Security (IPSec) -- an Internet standard for secure transmission of low-level network packets
- Secure Hypertext Transport Protocol (HTTPS) -- an Internet standard for securely transmitting Web pages

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
- Database design is usually performed in an early iteration of a system development project

Summary (continued)



- System controls are designed into the system to protect the system's data and other resources
- Controls are either integrity controls, which focus primarily on the specific application, or security controls, which apply across systems to include operating systems, Web sites, and networks
- Integrity controls include input controls, access controls, transaction logging, complex update controls, redundancy-backup-recovery, output controls, and fraud controls
- Security controls include access controls, data encryption, digital signatures/certificates, and secure transactions