Chapter 12
Database, controls, and security

Asst.Prof.Dr. Supakit Nootyaskool

Faculty of Information Technology

King Mongkut's Institute of Technology Ladkrabang

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

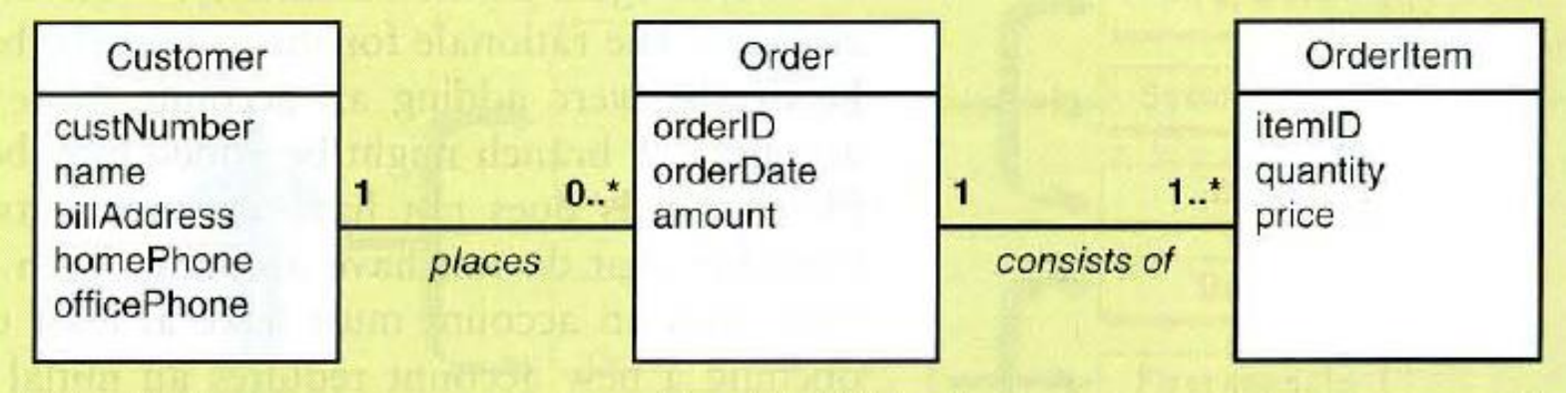
Objective

- Design relational database schema based on a class diagram
- Evaluate and improve the quality of 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

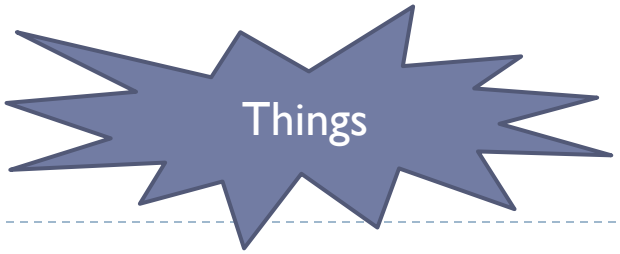
Overview

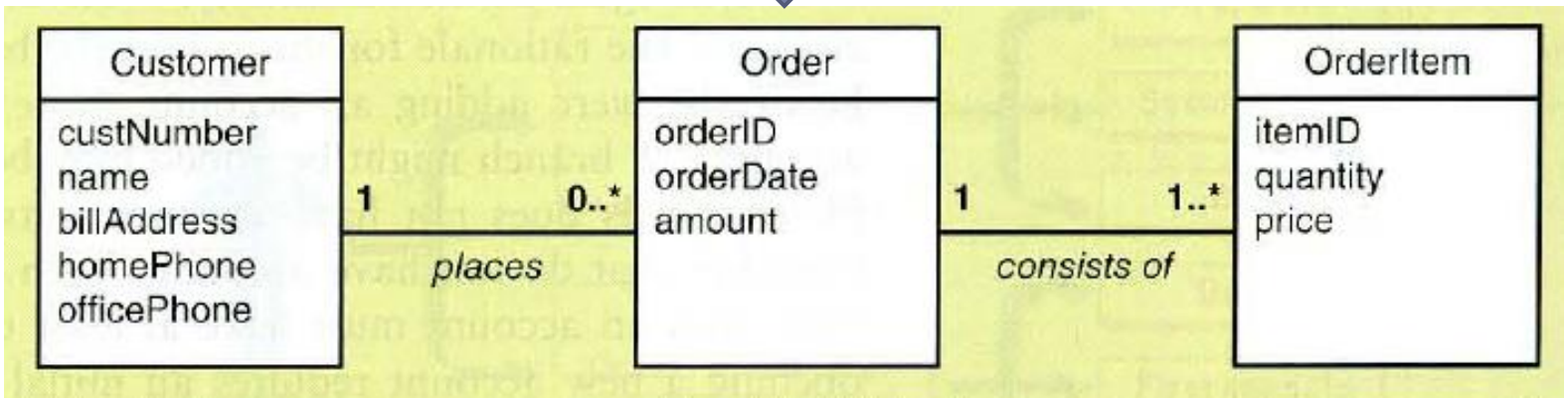
In this chapter, we transform the domain model class diagram (Chapter 4) into database model and implement by the database management system.

- Databases and database management systems are important components of a modern information 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



Overview





Domain model class diagram

Class diagram

ERD

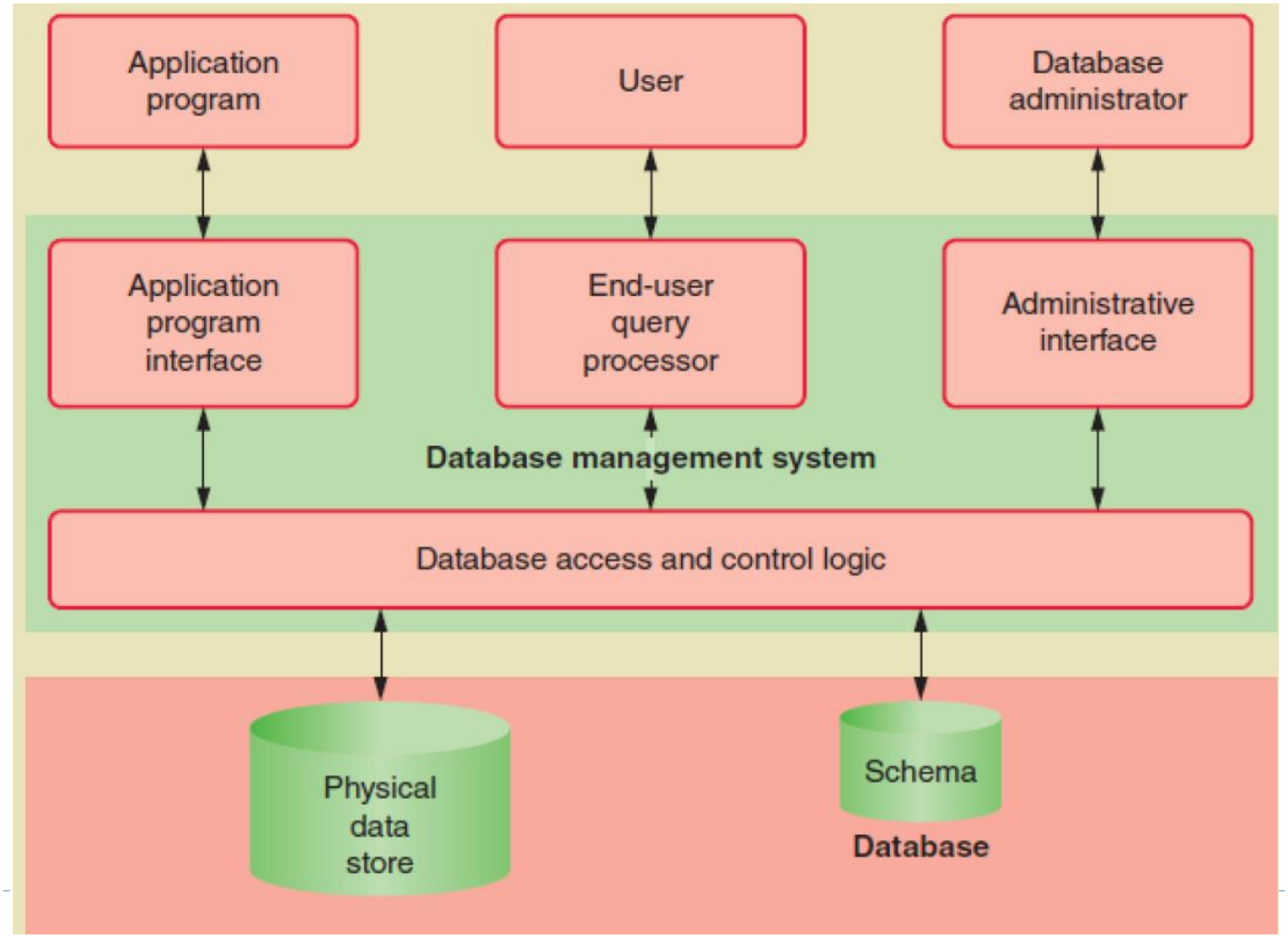
12.1 Database and database management systems

- Database (DB) -- an integrated collection of stored data that is centrally managed and controlled
- 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
  - Organization, tables
  - Association among table
  - Details, size, data types, indexing
  - Access, content controls

12.1 Database and database management systems (2)

- Database management system (DBMS) -- a system software component that manages and controls one or more databases. Four component of DBMS
  - Application program interface (API)
  - Query interface
  - Administrative interface
  - Set of data access program

12.1 Database and database management systems (3): Database and DBMS components



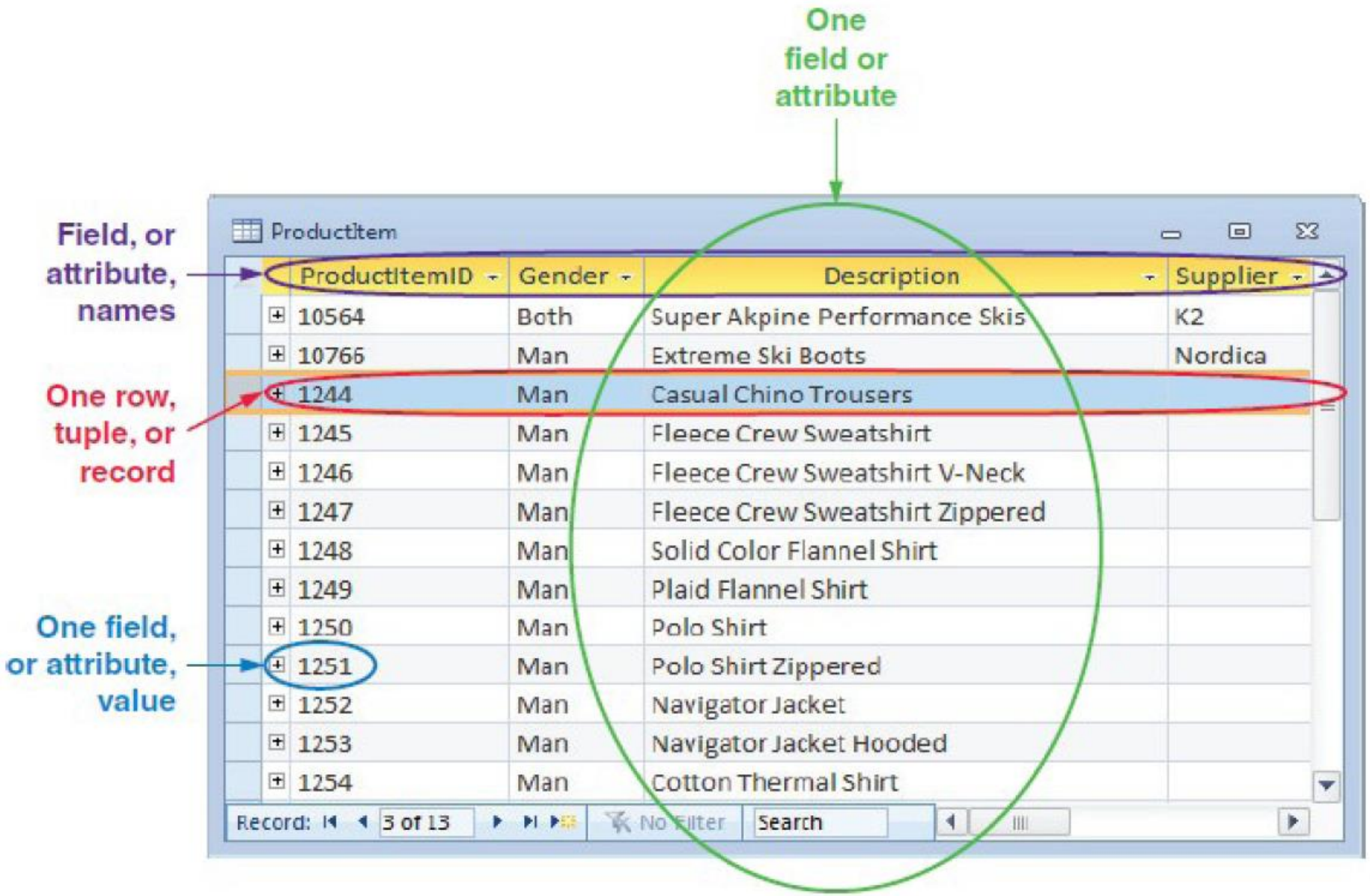
List of database tools

	Visual query builder	Visual schema/model/E-R diagram design	Reverse engineering	Forward engineering	ER diagram groupboxes
SQLyog	Yes[12]	Yes[13]	Yes	Yes	?
Adminer	Yes	Yes	Yes	No	No
Altova DatabaseSpy	Yes	Yes	Yes	Yes	?
Database Deployment Manager	Yes	Yes	Yes	No	No
Database Workbench	Yes	Yes	Yes	?	Yes
Devgems Data Modeler	No	Yes	Yes	Yes	Yes
DeZign for Databases	No	Yes	Yes	Yes[11]	Yes
ModelRight	No	Yes	Yes	Yes	Yes
Navicat	Yes	Yes	Yes	Yes	Yes
Navicat Data Modeler	No	Yes	Yes	Yes	Yes
MySQL Workbench	Yes	Yes	Yes	Yes	Yes
Oracle SQL Developer	Yes	Yes	Yes	Yes	?
phpMyAdmin	Yes	Yes	Yes	No	No
SQL Server Management Studio	?	Yes	Yes	?	?
SQuirreL SQL	Yes	Yes	Yes[14]	?	No
Toad	Yes	Yes	Yes	Yes	?
Toad Data Modeler	No	Yes	Yes	Yes[15]	?
DaDaBIK	Some[10]	No	No	No	No
DBEdit	No	No	No	No	No
Orbada	No	No	No	No	No

Ref: http://en.wikipedia.org/wiki/Comparison_of_database_tools

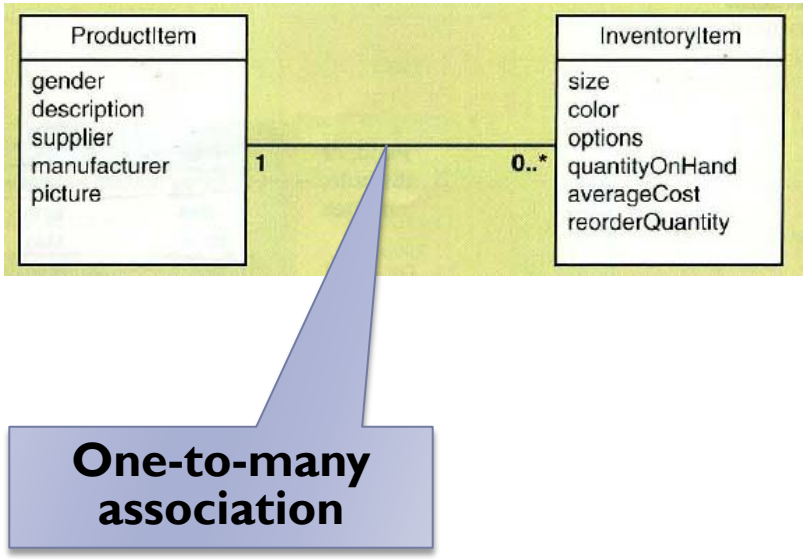
12.2 Relational database

- 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


One field or attribute
Field, or attribute, names
One row, tuple, or record
One field, or attribute, value
ProductItem
ProductItemID
Gender
Description
Supplier
10564 Both Super Akpine Performance Skis 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
1252 Man Navigator Jacket
1253 Man Navigator Jacket Hooded
1254 Man Cotton Thermal Shirt
Record: 3 of 13
No Filter
Search

An association between rows in two tables (Key and foreign key)

Portion of the RMO class diagram



ProductItem

ProductItemID	Gender	Description	Supplier
10564	Both	Super Alpine Performance Skis	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	
1252	Man	Navigator Jacket	
1253	Man	Navigator Jacket Hooded	
1254	Man	Cotton Thermal Shirt	

Record: 3 of 13 No Filter Search

InventoryItem

InventoryID	ProductID	Size	Color	Options	QuantityOnHand	Average Cost	RecorderQuantity
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
86789	1244	30/33	Slate		41	$12.75	100
86790	1244	30/34	LightTan		35	$12.75	50

Record: 13 of 13 No Filter Search

12.2.1 Design relational databases

- Design relational database performs two ways a class diagram and a raw table data.

- To create a relation database from a domain model class diagram, apply to follow step.
  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

12.2.1 Design relational databases: Example: RMO {Step1}

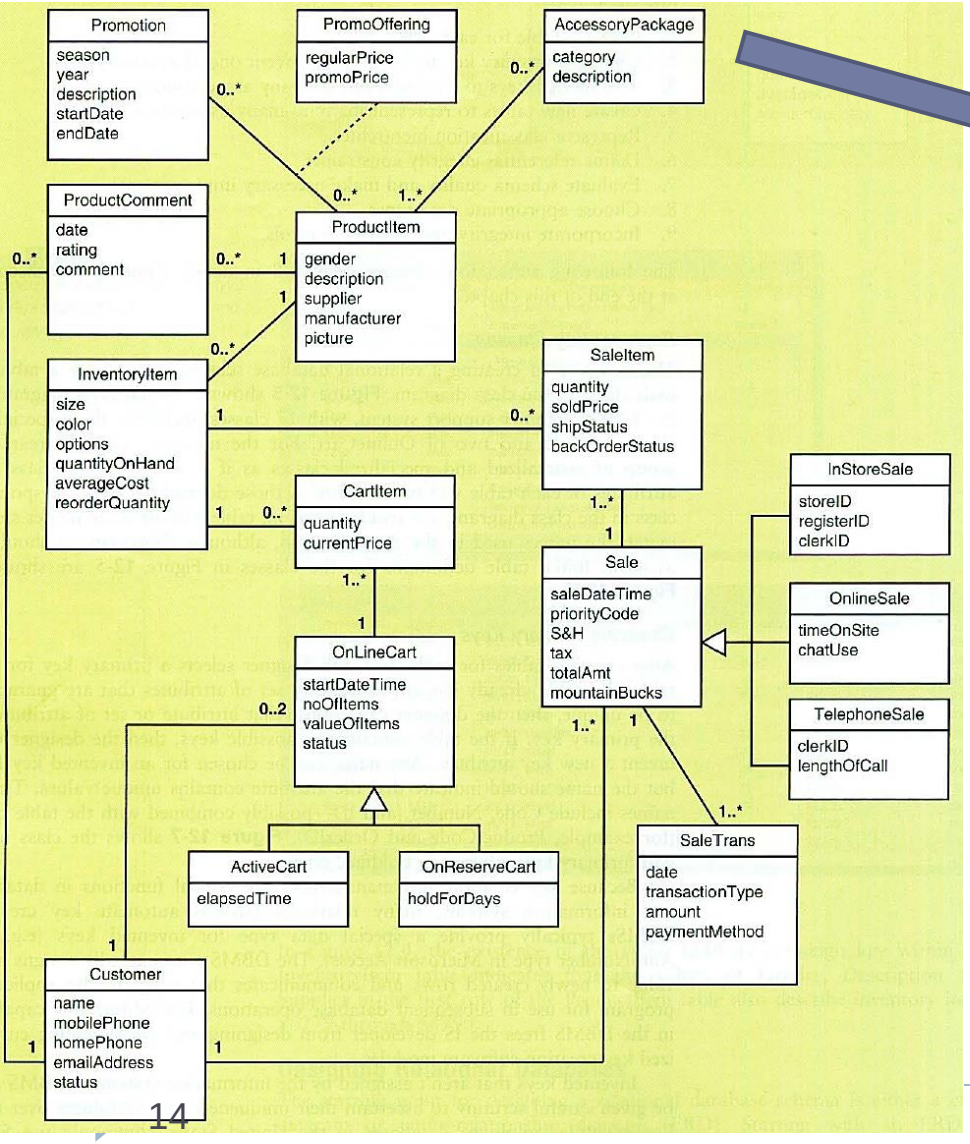


Table	Attributes
AccessoryPackage	Category, Description
CartItem	Quantity, CurrentPrice
Customer	Name, MobilePhone, HomePhone, EmailAddress, Status
InventoryItem	Size, Color, Options, QuantityOnHand, AverageCost, ReorderQuantity
OnlineCart	StartDateTime, NumberOfItems, ValueOfItems, Status, ElapsedTime, 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: {Step2} with primary key added (bold text)

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: {Step3} with foreign key attributes added (in italics)

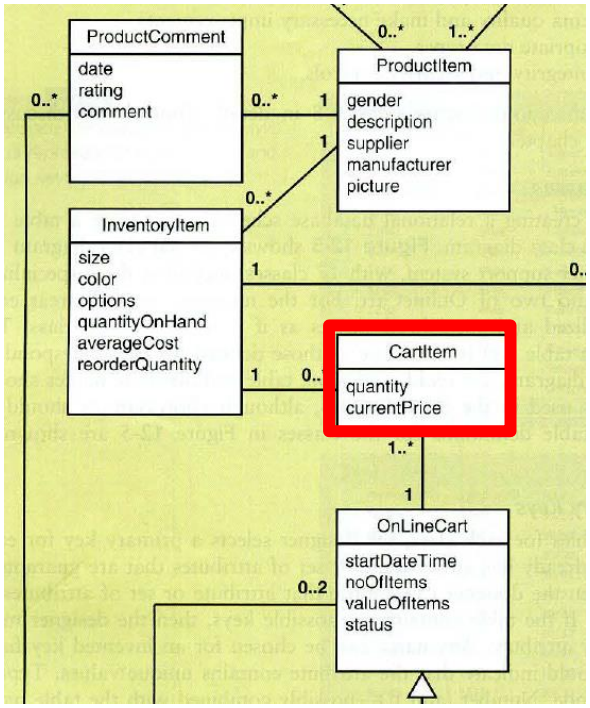
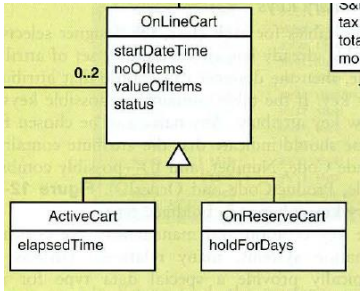


Table	Attributes
AccessoryPackage	**AccessoryPackageID**, Category, Description
CartItem	**CartItemID**, *InventoryItemID*, *OnlineCartID*, Quantity, CurrentPrice
Customer	**AccountNumber**, Name, MobilePhone, HomePhone, EmailAddress, Status
InventoryItem	**InventoryItemID**, *ProductItemID*, Size, Color, 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

Final set of tables

- Specialized subclasses of sale and online cart added



- Promo offering modified from association class to table with two keys

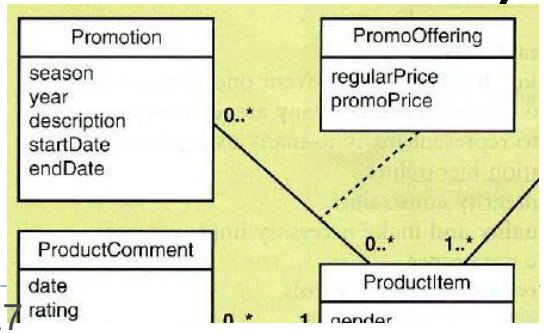


Table	Attributes
AccessoryPackage	**AccessoryPackageID**, Category, Description
AccessoryPackageContents	***AccessoryPackageID, InventoryItemID***
ActiveCart	***OnlineCartID***, ElapsedTime
CartItem	***InventoryItemID***, ***OnlineCartID***, Quantity, CurrentPrice
Customer	**AccountNumber**, Name, MobilePhone, 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

12.2.2 Designing relational databases: Referential integrity

- Referential integrity -- a consistent state among foreign key and primary key values

- Referential integrity constraint -- a constraint, stored in the schema, that the database designer tells DBMS which columns are foreign keys and to which primary key columns, to protect wrong data update.

Example SQL

```
ADD CONSTRAINT FK_SaleItem_Sale FOREIGN KEY
SaleID REFERENCES Sale(ID)
```

12.2.3 Designing relational databases: Evaluating schema quality

There are multiple possibility for database design error

- Error in domain class diagram
- Poor choice primary key
- Error converting class diagram to relational table.

- 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

Redundancy data

Student_ID	Name	Contact	College	Course	Rank
100	Himanshu	7300934851	GEU	Btech	1
101	Ankit	7900734858	GEU	Btech	1
102	Aysuh	7300936759	GEU	Btech	1
103	Ravi	7300901556	GEU	Btech	1

Students Table

Student	ID*
John Smith	084
Jane Bloggs	100
John Smith	182
Mark Antony	219

Activities Table

ID*	Activity1	Cost1	Activity2	Cost2
084	Tennis	$36	Swimming	$17
100	Squash	$40	Swimming	$17
182	Tennis	$36		
219	Swimming	$15	Golf	$47

12.2.4 Designing relational databases: Database normalization

- First normal form (1NF) -- restriction that all rows of a table must contain the same number of columns
  - There's no top-to-bottom ordering to the rows.
  - There's no left-to-right ordering to the columns.
  - There are no duplicate rows.
  - Every row-and-column intersection contains exactly one value from the applicable domain (and nothing else).
  - All columns are regular [i.e. rows have no hidden components such as row IDs, object IDs, or hidden timestamps].

Customer

Customer ID	First Name	Surname	Telephone Number
123	Robert	Ingram	555-861-2025
456	Jane	Wright	555-403-1659
456	Jane	Wright	555-776-4100
789	Maria	Fernandez	555-808-9633

Repeating group!

Customer Name

Customer ID	First Name	Surname
123	Robert	Ingram
456	Jane	Wright
789	Maria	Fernandez

Customer Telephone Number

Customer ID	Telephone Number
123	555-861-2025
456	555-403-1659
456	555-776-4100
789	555-808-9633

ref: http://en.wikipedia.org/wiki/First_normal_form

12.2.4 Designing relational databases: Database normalization (2)

- 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
- Example:

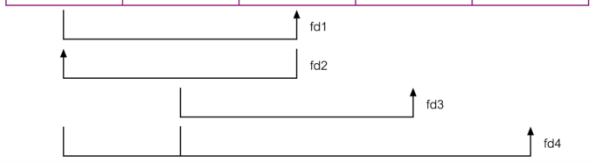
ProductItem

ProductItemID	Gender	Description	Supplier
10564	Both	Super Akpine Performance Skis	K2
10766	Man	Extreme Ski Boots	Nordica
1244	Man	Casual Chino Trousers	
1245	Man	Fleece Crew Sweatshirt	
1246	Man	Fleece Crew Sweatshirt V-Neck	

รหัสนักศึกษา	รหัสวิชา	เกรด	ชื่อวิชา
52116940001	F01	A	การเขียนโปรแกรม
52116940001	F02	B	การจัดการฐานข้อมูล
52116940002	F01	D	การเขียนโปรแกรม
52116940002	F02	A	การจัดการฐานข้อมูล
52116940003	F01	A	การเขียนโปรแกรม
52116940003	F02	C	การจัดการฐานข้อมูล

Sample Relation

A	B	C	D	E
a	b	z	w	q
e	b	r	w	p
a	d	z	w	t
e	d	r	w	q
a	f	z	s	t
e	f	r	s	t



12.2.4 Designing relational databases: Database normalization (3)

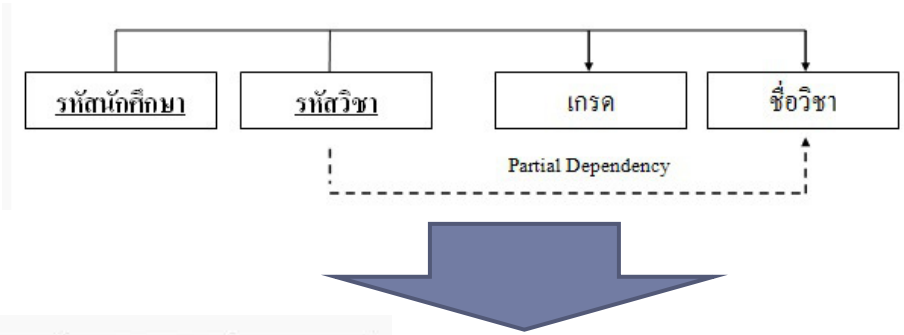
- Second normal form (2NF) -- restriction that a table is in 1NF and that each non-key attribute is functionally dependent (full function dependency) on the entire primary key
  - if partial dependencies exist on the primary key remove them by placing a new table.

RegularPrice is functionally dependent on *PromotionID* if for each value of *PromotionID* there is only one corresponding value of *Regular Price*.

RegularPrice is functionally dependent on *ProductItemID* if for each value of *ProductItemID* there is only one corresponding value of *Regular Price*.

12.2.4 Designing relational databases: Database normalization (4)

รหัสนักศึกษา	รหัสวิชา	เกรด	ชื่อวิชา
52116940001	F01	A	การเขียนโปรแกรม
52116940001	F02	B	การจัดการฐานข้อมูล
52116940002	F01	D	การเขียนโปรแกรม
52116940002	F02	A	การจัดการฐานข้อมูล
52116940003	F01	A	การเขียนโปรแกรม
52116940003	F02	C	การจัดการฐานข้อมูล



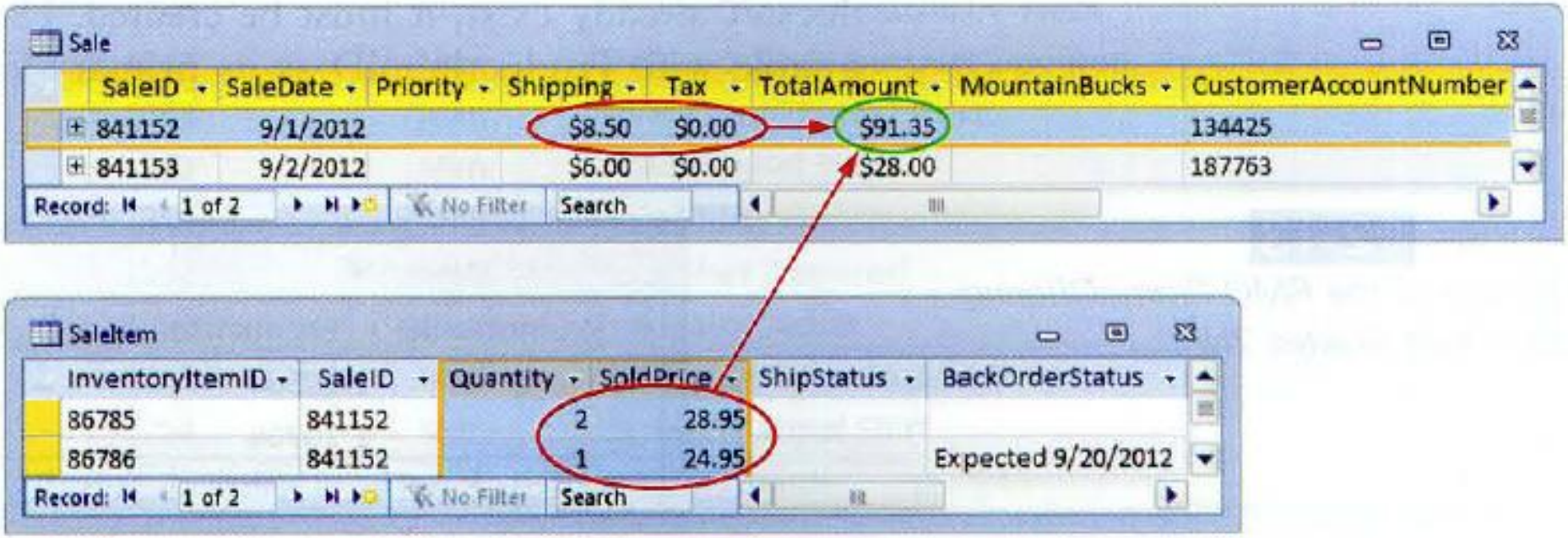
รหัสนักศึกษา	รหัสวิชา	เกรด
52116940001	F01	A
52116940001	F02	B
52116940002	F01	D
52116940002	F02	A
52116940003	F01	A
52116940003	F02	C

รหัสวิชา	ชื่อวิชา
F01	การเขียนโปรแกรม
F02	การจัดการฐานข้อมูล

ref: http://swe3et.wordpress.com/nf2/

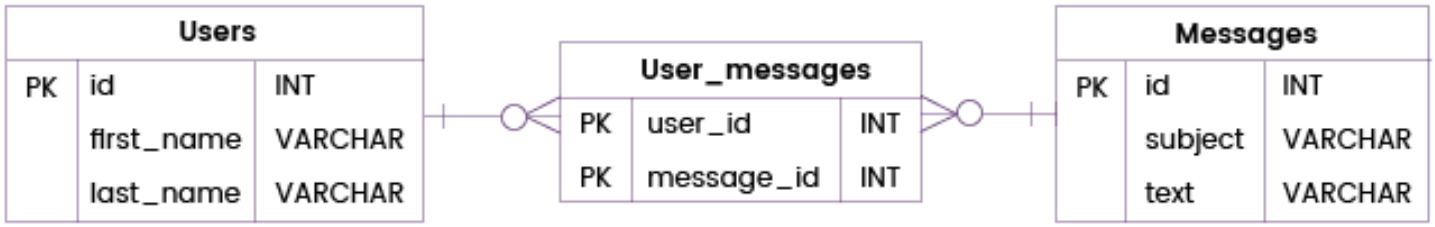
12.2.4 Designing relational databases: Database normalization (2)

- 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

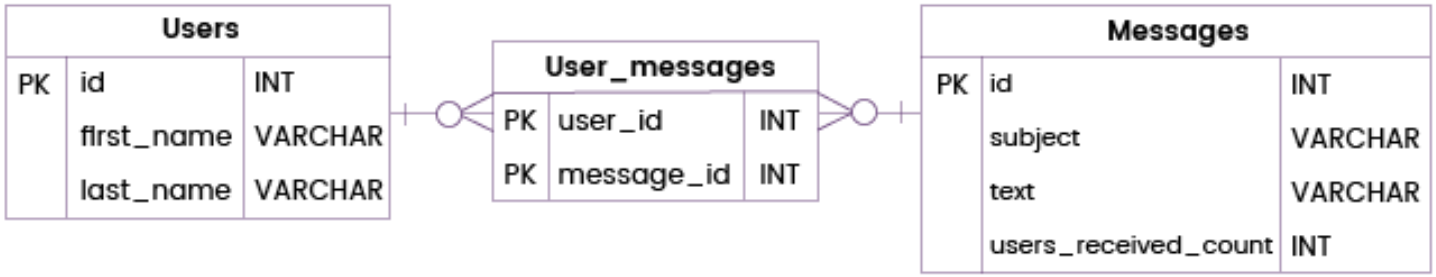


Save storage with denormalized database (1)

Normalized database



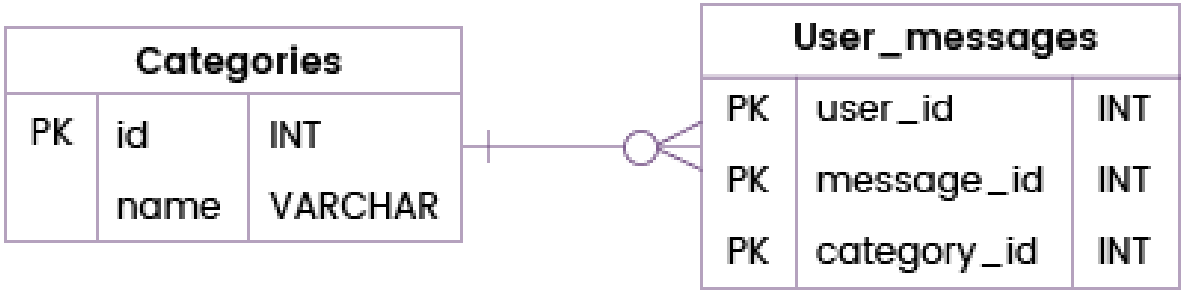
Denormalized database



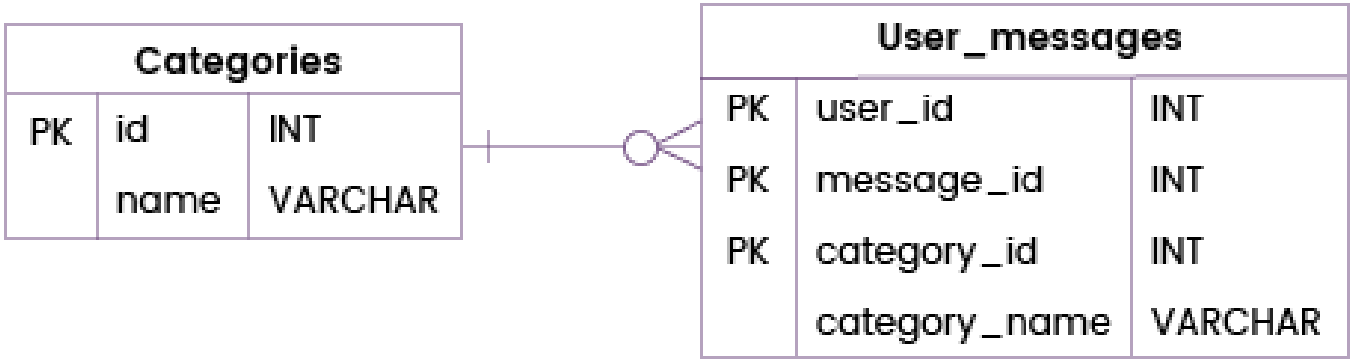
https://rubygarage.org/blog/database-denormalization-with-examples

Pre-joint table with denormalized database (2)

Normalized database



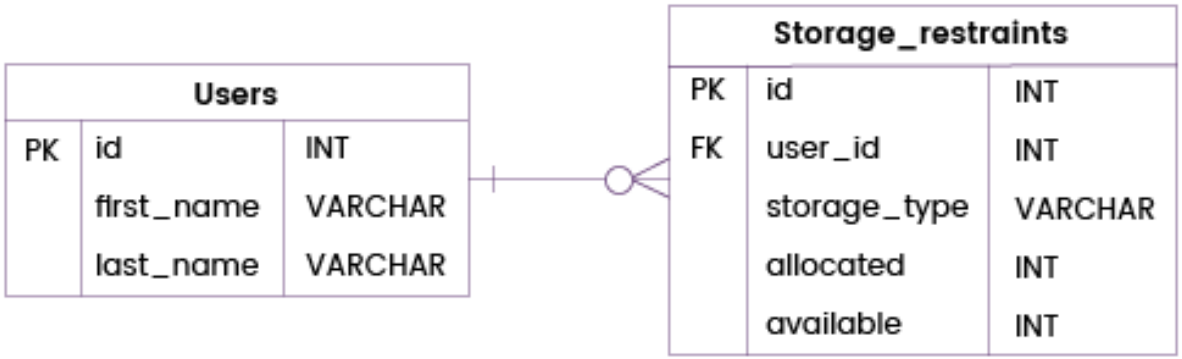
Denormalized database



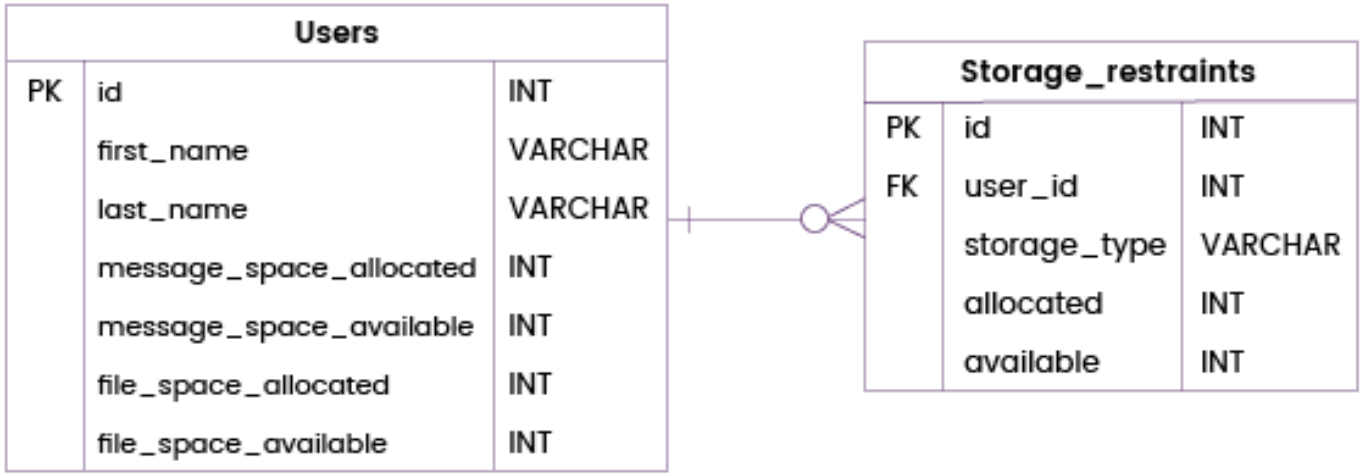
https://rubygarage.org/blog/database-denormalization-with-examples

Speed up with denormalized database (3)

Normalized database



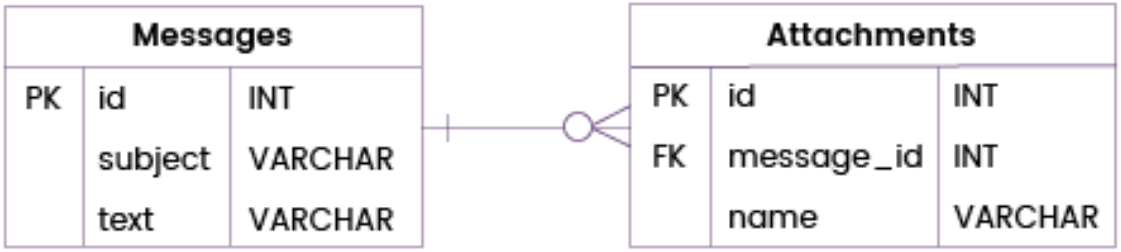
Denormalized database



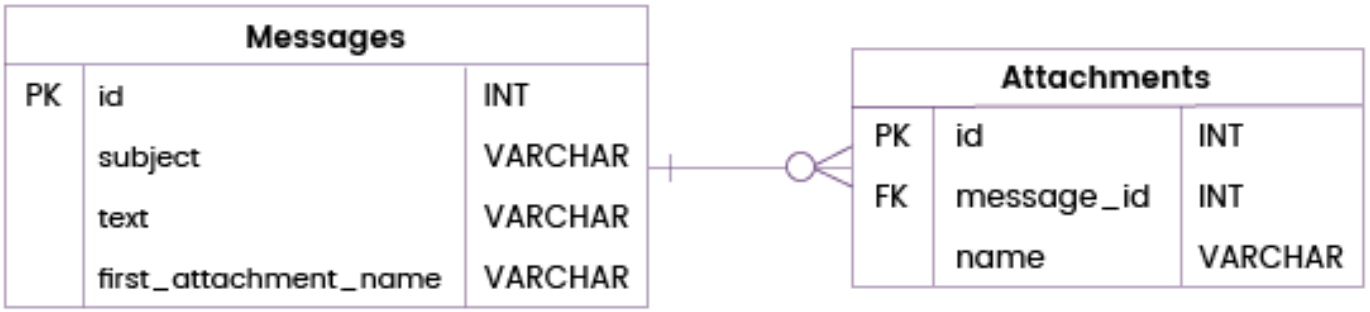
https://rubygarage.org/blog/database-denormalization-with-examples

Avoid joint table with denormalized database (4)

Normalized database



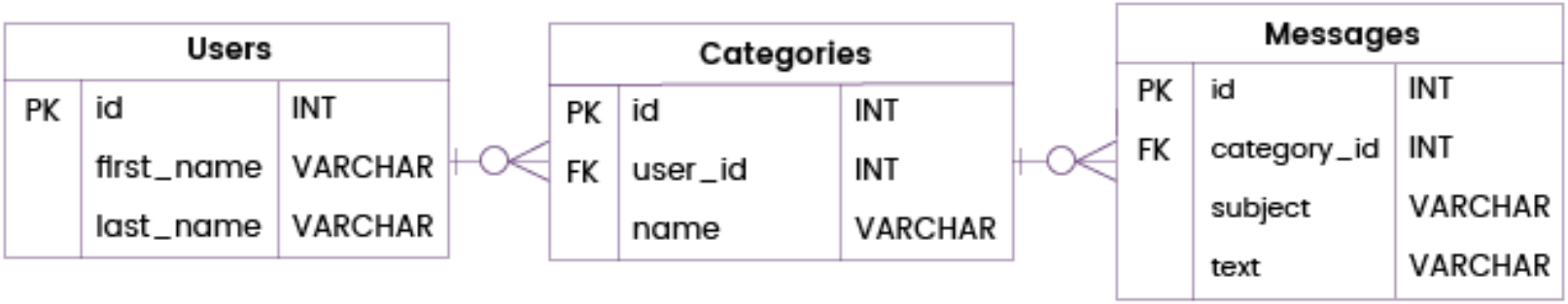
Denormalized database



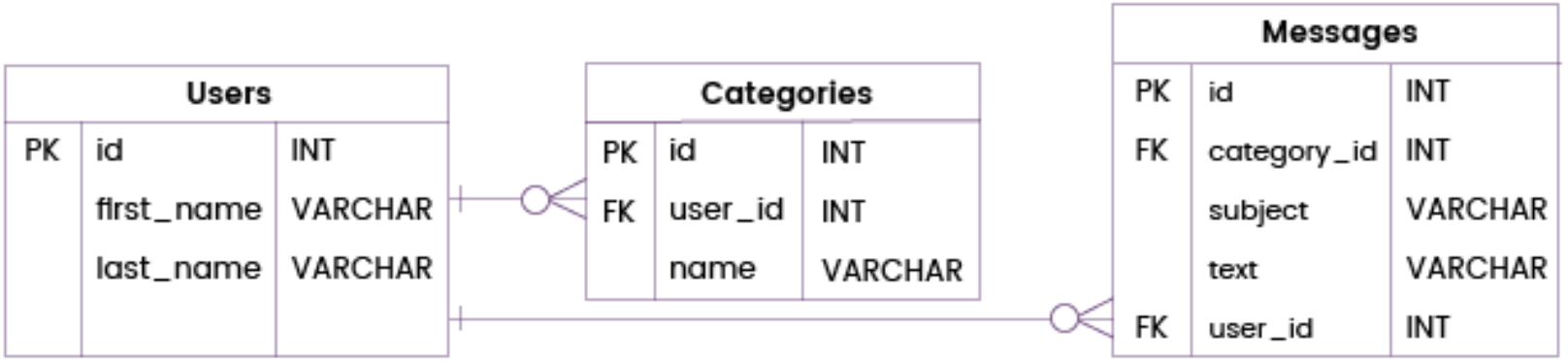
https://rubygarage.org/blog/database-denormalization-with-examples

Short circuit with denormalized database (5)

Normalized database

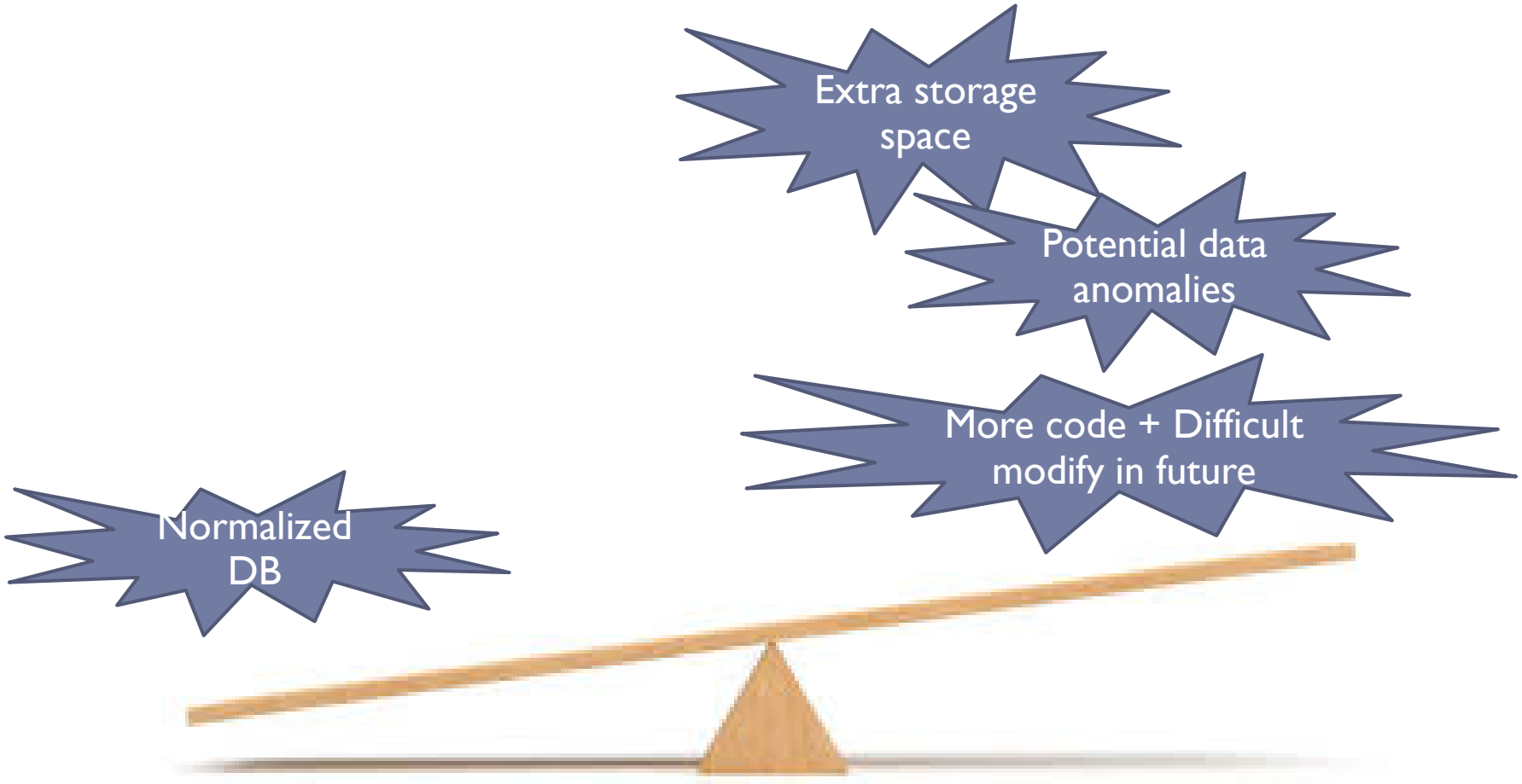


Denormalized database



https://rubygarage.org/blog/database-denormalization-with-examples

Seesaw

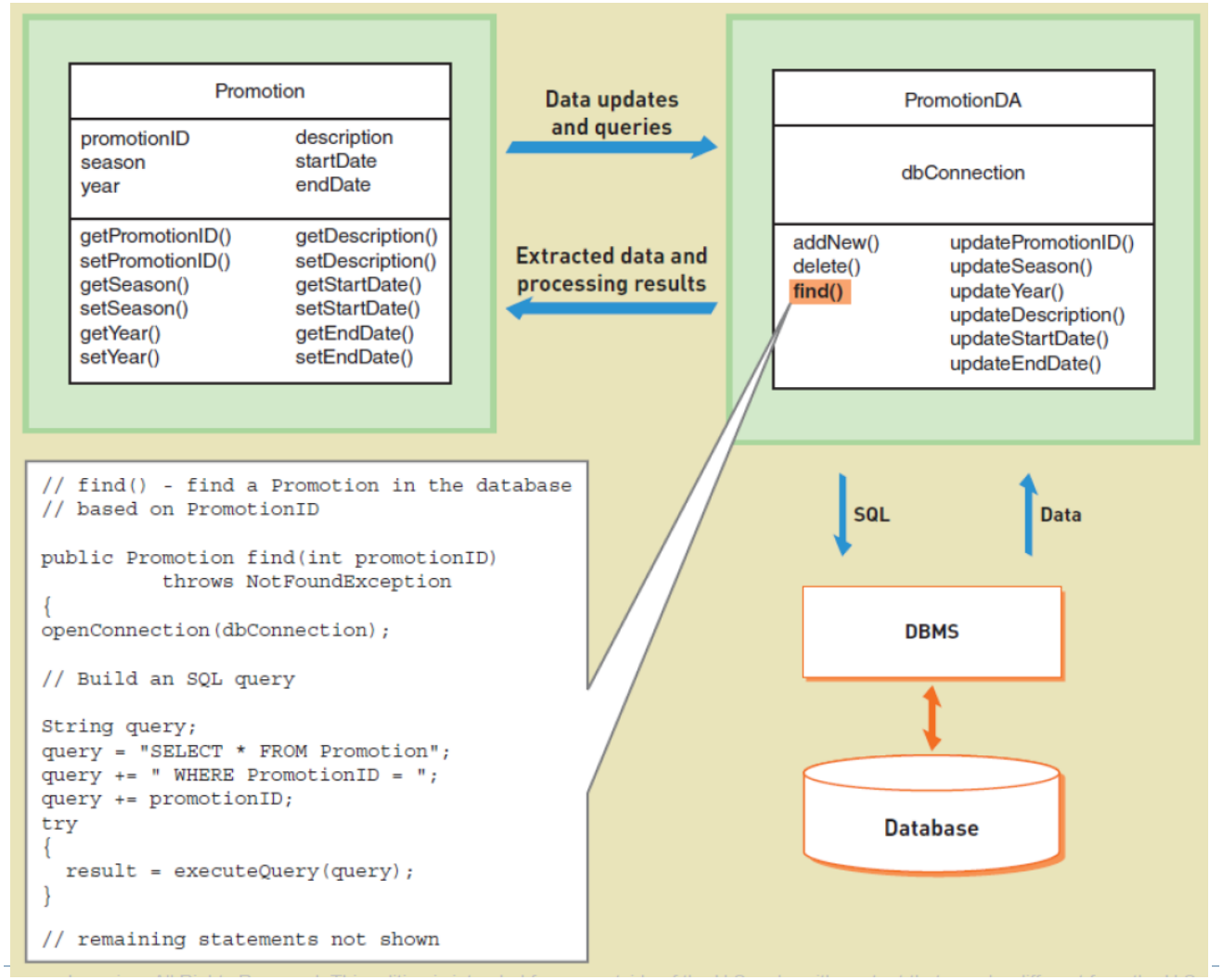

Extra storage space
Potential data anomalies
More code + Difficult modify in future
Normalized DB

12.2.5 Designing relational databases: Data type

- Primitive data types
- Complex 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 2 GB
xml	XML document up to 2 GB

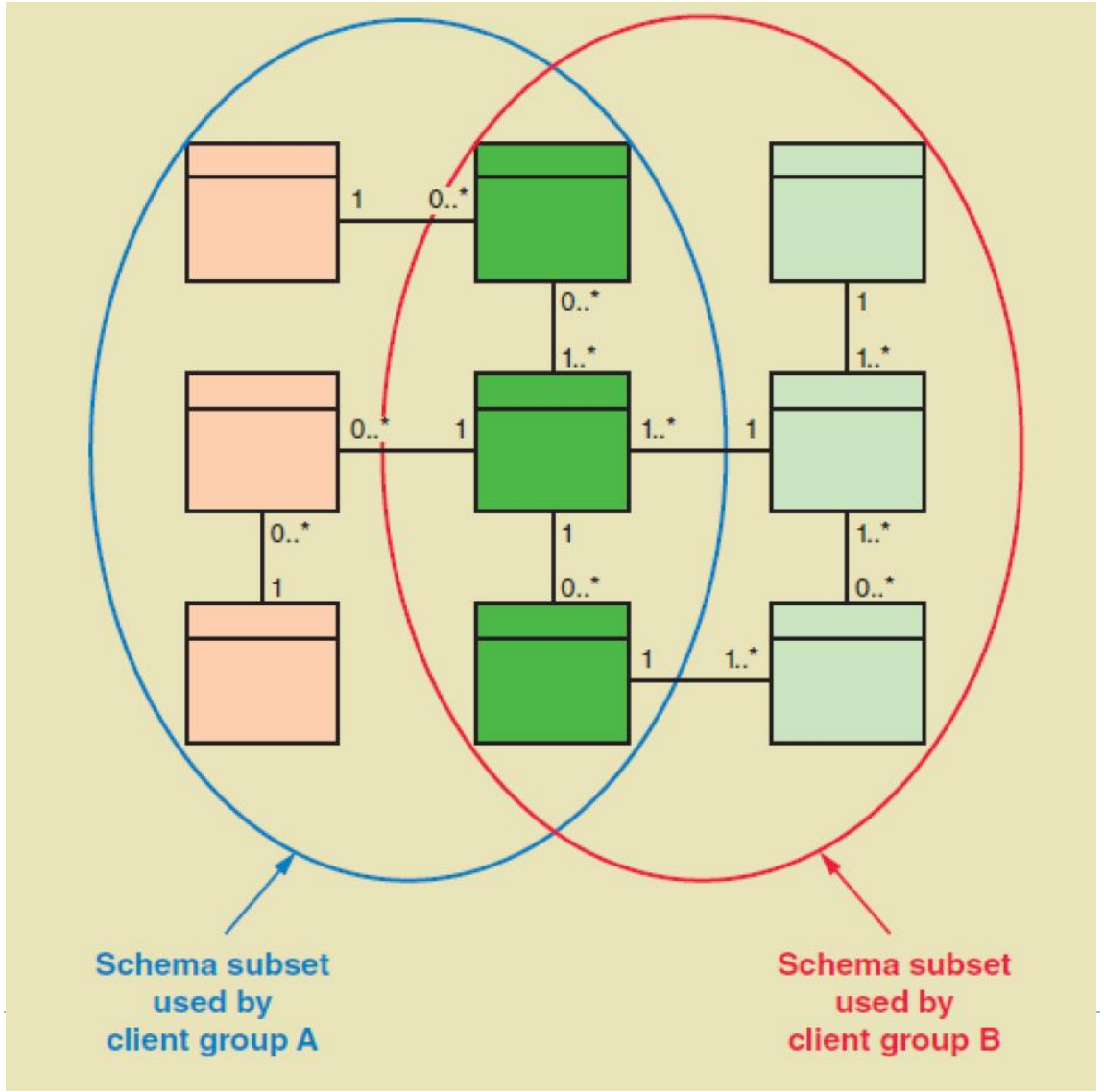
12.3 Data access classes
Map design classes to RDMBS tables



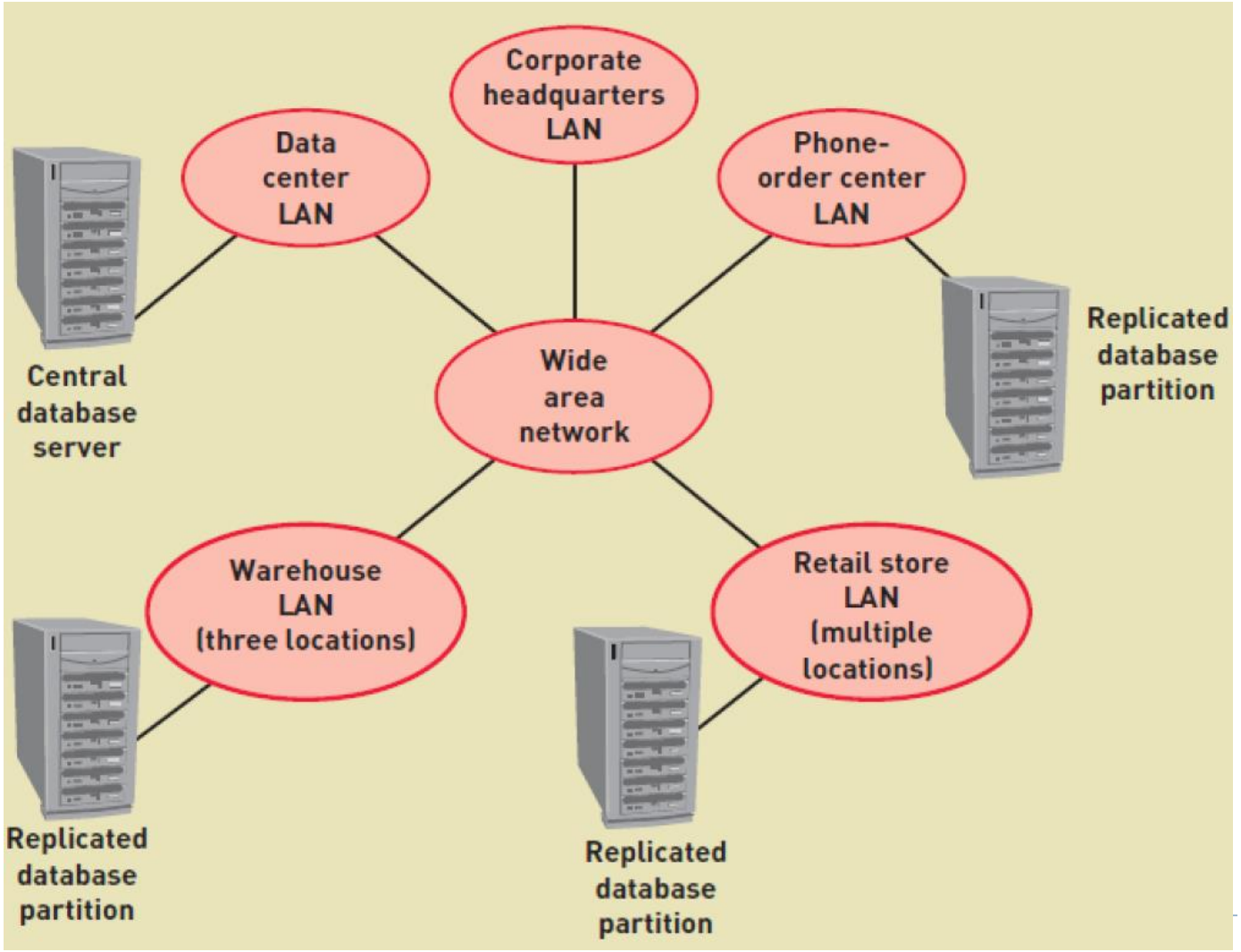
12.4 Distribute database architecture

- 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



12.5 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 (old db) 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.

12.6 Design integrity control

- 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?

- ร้านก๋วยเตี๋ยว ลูกค้ากินไปทั้งโต๊ะ 230 บาท เก็บเงินมาครบ แต่ตอนส่งเงินให้เถ้าแก่อมไว้ 20-30 บาท วันหนึ่งทำได้ 10 รายก็ 200-300 บาท

- พนักงานในร้านเก็บเงินลูกค้าส่งแคชเชียร์หรือแคชเชียร์ที่เก็บเงินลูกค้าเอง จะบันทึกรายการเป็นการให้ส่วนลดกับลูกค้า เช่น ระบุว่าลูกค้ามีบัตรสมาชิกหรือคูปองส่วนลด โดยที่ลูกค้าไม่ได้สิทธิ เก็บเงินลูกค้ามาเต็ม โดยพนักงานเอาบัตรลด คูปองหรือบัตรสมาชิกที่ตัวเองเก็บไว้มาประกอบเพื่อใช้สิทธิแทนลูกค้า ลูกค้าส่วนใหญ่ไม่รับใบเสร็จหรือถ้าแคชเชียร์ออกใบเสร็จพนักงานก็ไม่นำมาให้ลูกค้า

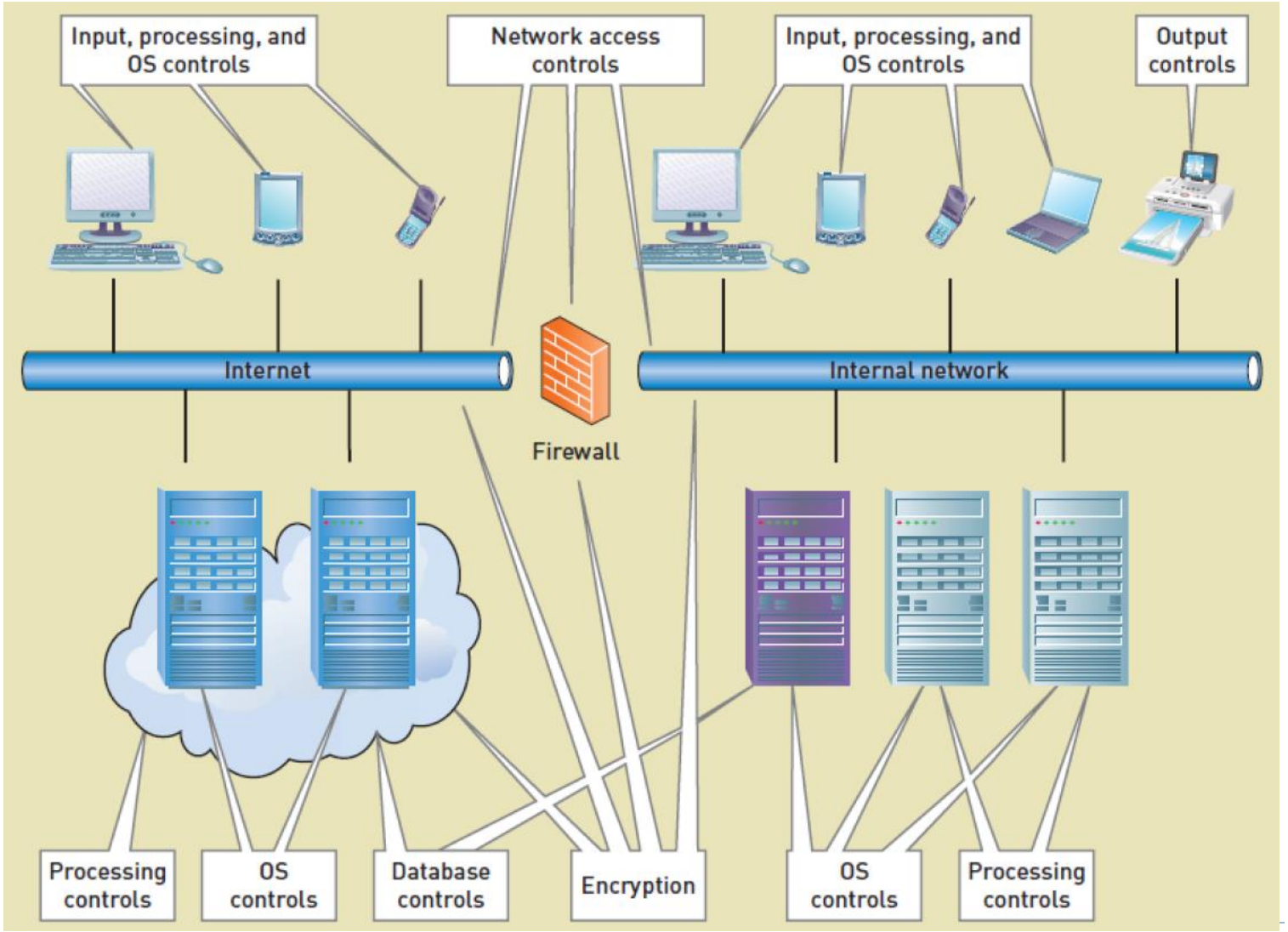
- ขโมยของในร้าน ตั้งแต่สินค้า ข้าวของเครื่องใช้ในร้านอย่างถ้วยชาม ช้อนส้อม มีด จนถึงเครื่องปรุงในครัวอย่างเนื้อหมู น้ำมันพืช เครื่องกระป๋อง กาแฟ ครีมเทียม

https://www.smethailandclub.com/money-1104-id.html

- ให้เพื่อนเข้ามาเป็นลูกค้า โดยพนักงานที่ขายของออกไปให้เก็บเงินไม่ครบ ...ซื้อสุราฝรั่ง 8 ขวด อาจบันทึกรายการขายแค่ 6 ขวด หรือ ซื้อรังนก 1 กล่อง อาจบันทึกเป็นซุปไก่สกัดยี่ห้อเดียวกัน 1 กล่อง
- ผู้บริหารซูเปอร์มาร์เก็ตแห่งหนึ่งเล่าให้ฟังว่า ผักเหี่ยว ๆ ผลไม้เน่า ๆ ที่เหลือจากการขายของร้าน แต่ก่อนเคยขายถูก ๆ ให้กับญาติของพนักงานที่แจ้งว่ามาซื้อไปเป็นอาหารให้หมู

https://www.tam-eig.com/2018/07/23/4-employee-tricks-restaurant-owners-need-to-know/

12.6 Design integrity control (2)



12.6 Design integrity control (3) Design 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

12.6 Design integrity control (4) 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

12.6 Design integrity control (5) 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: Top prevent farad

- 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: Top prevent farad

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.

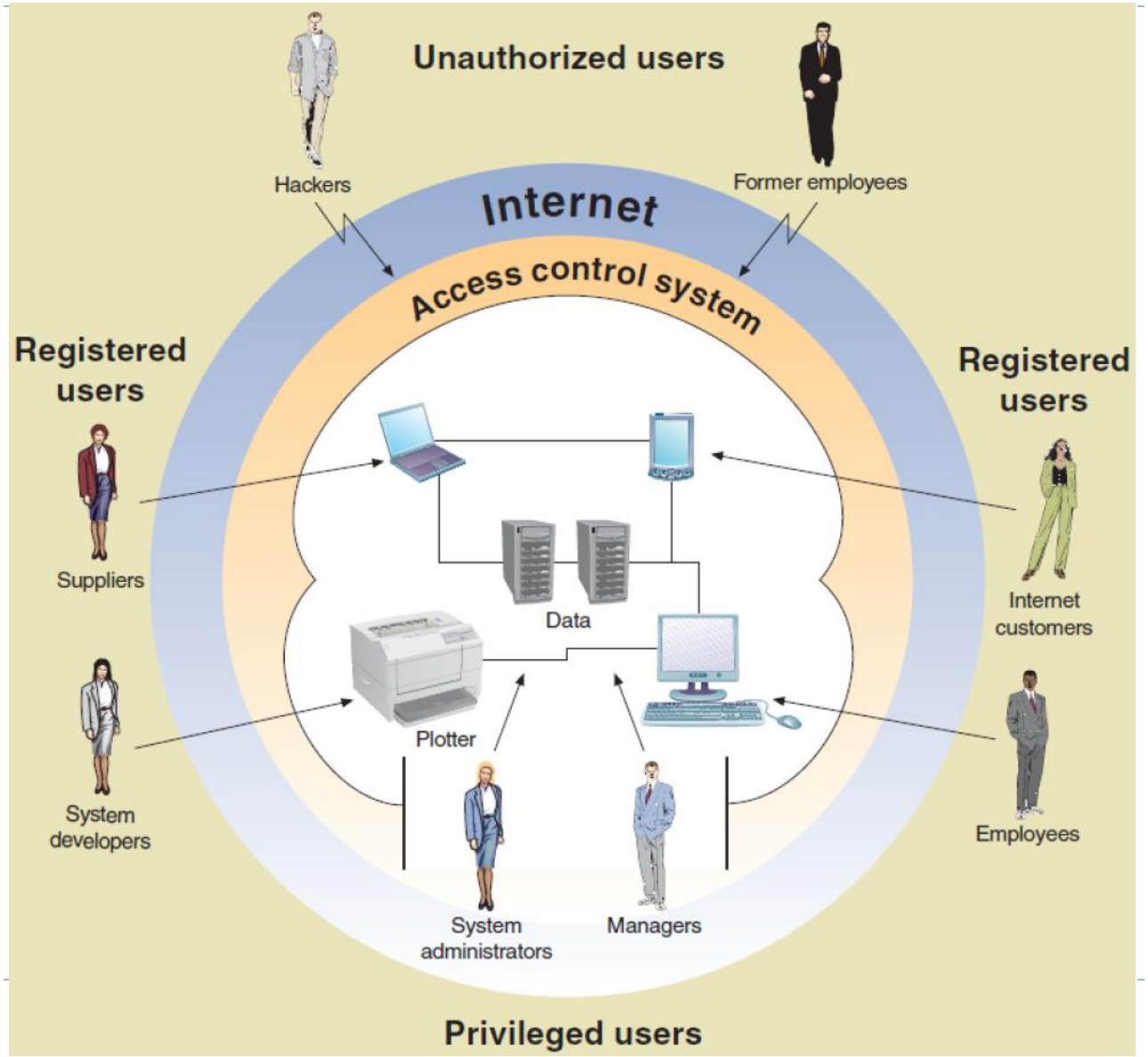
12.7 Designing security controls

- 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

12.7 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

12.7 Security controls: access control



12.7 Security controls: data encryption

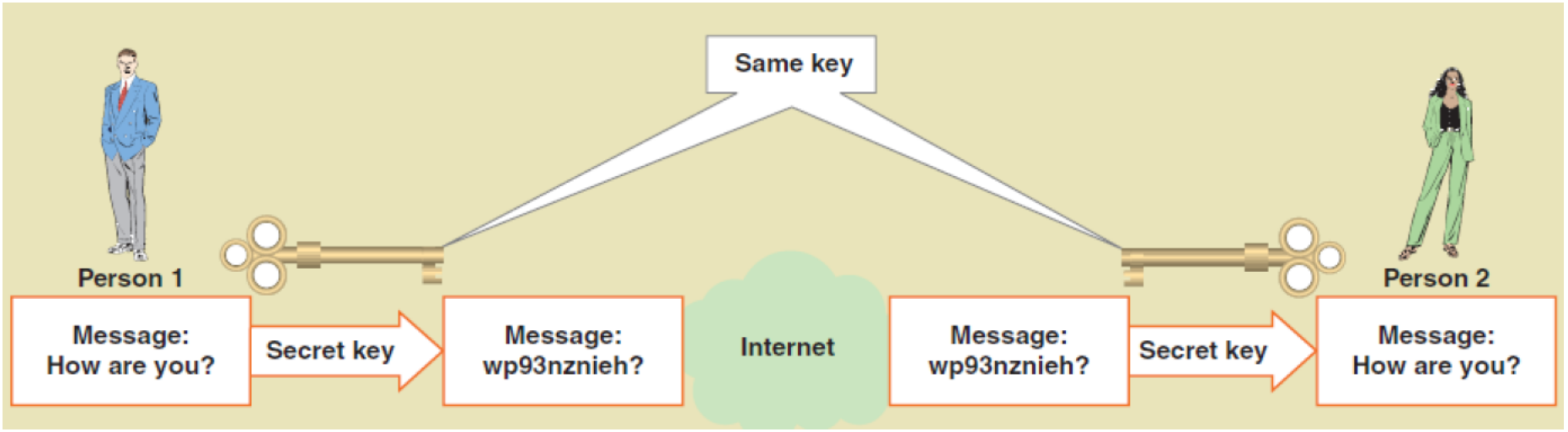
- Common types of data requiring additional protection
  - Financial information
  - Credit card numbers, bank account numbers, payroll information, healthcare information, and other personal data
  - Strategies and plans for products and other mission-critical data
  - Government and sensitive military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

12.7 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

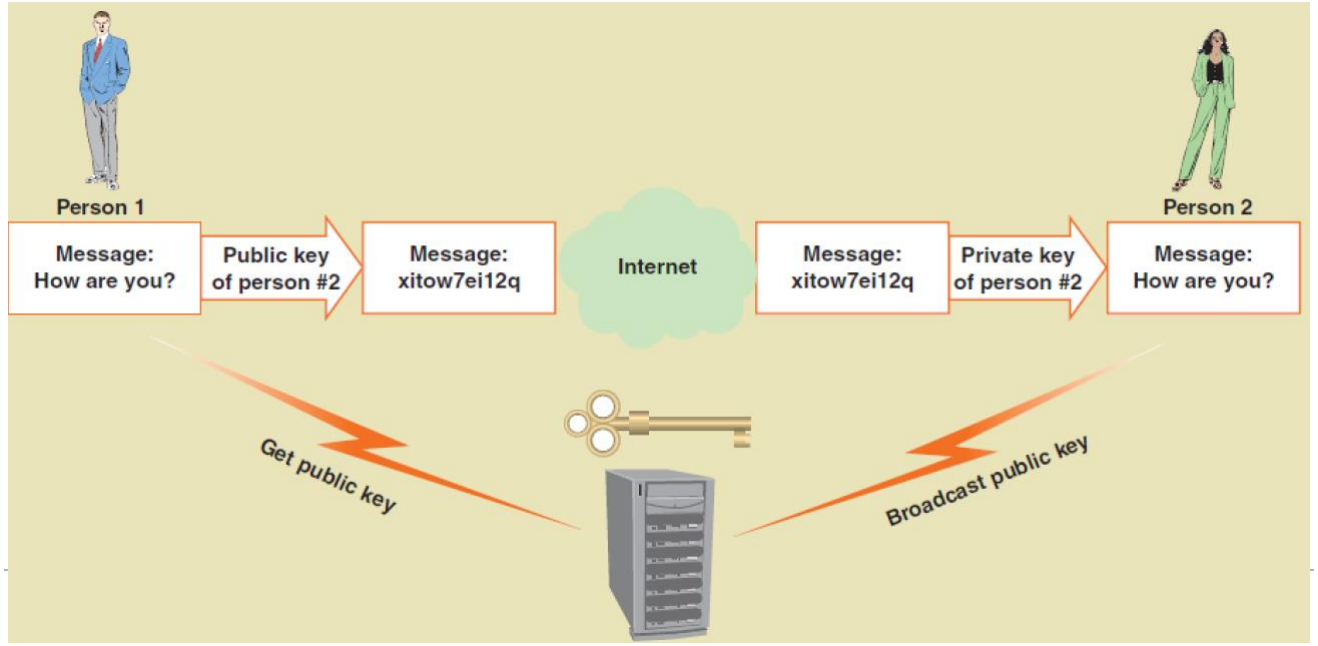
12.7 Security controls: data encryption

- Symmetric key encryption



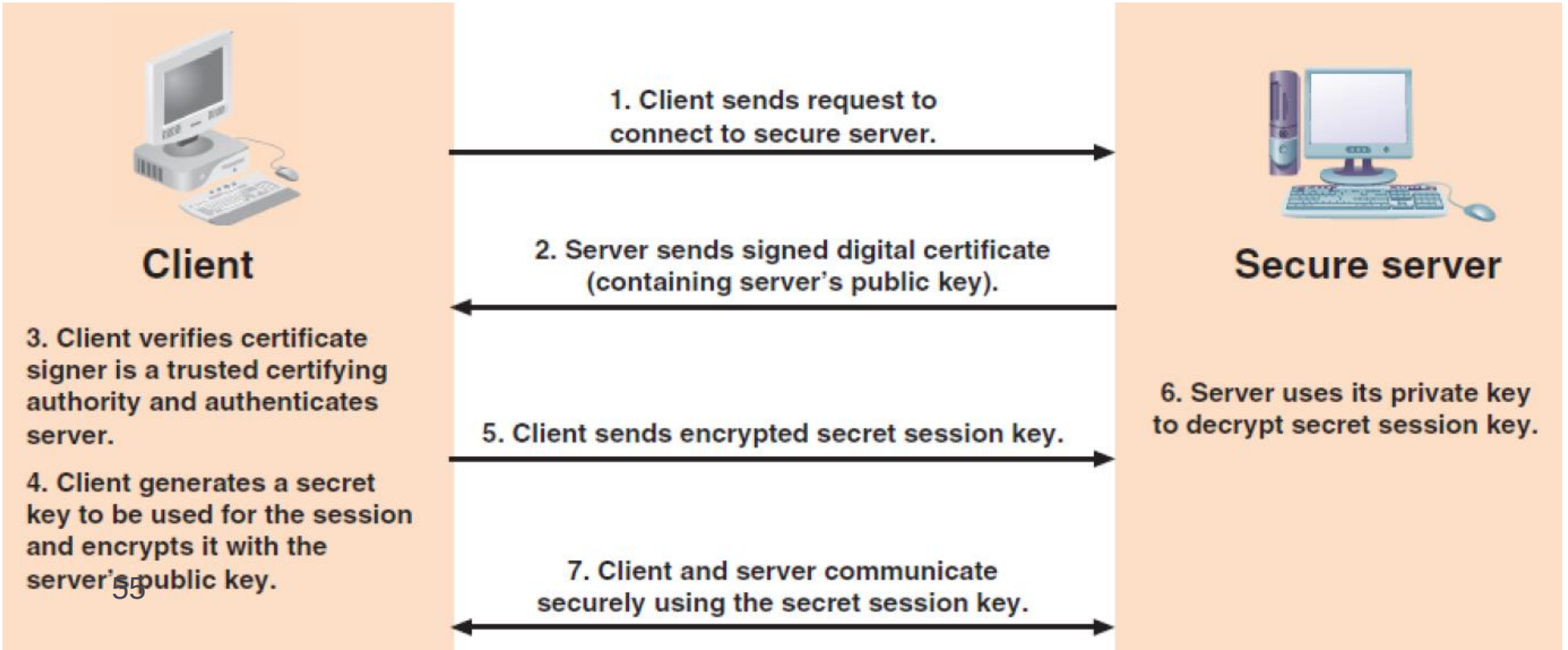
12.7 Security controls: data encryption

- 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



12.7 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



12.7 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 (2)

- 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

A salesman has recorded data in Excel as shown below. Given you consider data to create database tables and Entity-Relationship Diagram.

Customer Name	Item	Customer Address	Newsletter	Supplier	Supplier Phone	Price
Alan Smith	Xbox	35 Palm St. Miami	Xbox News	Microsoft	800-BuyXbox	250
Roger Banks	Playstation4	47 Campus Rd, Boston	Playstation News	Sony	800-BuyPS4	300
Evan Wilson	Xbox, PS Vita	28 Rock AV, Denver	Xbox News, Playstation News	WholeSale	800-WholeSale1	450
Alan Smith	Playstation4	47 Campus Rd, Boston	Playstation News	Sony	800-BuyPS4	300