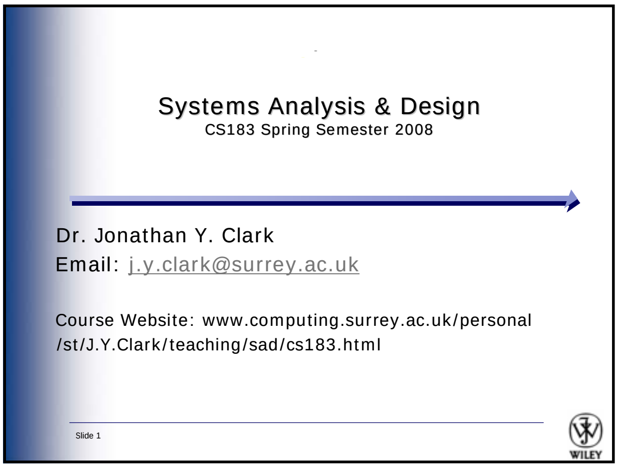


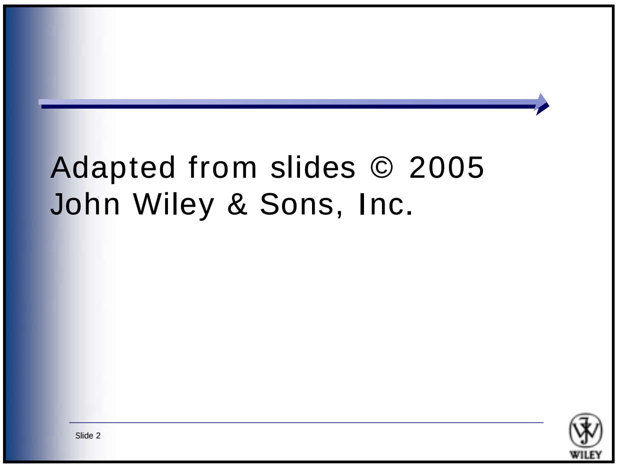
Systems Analysis & Design

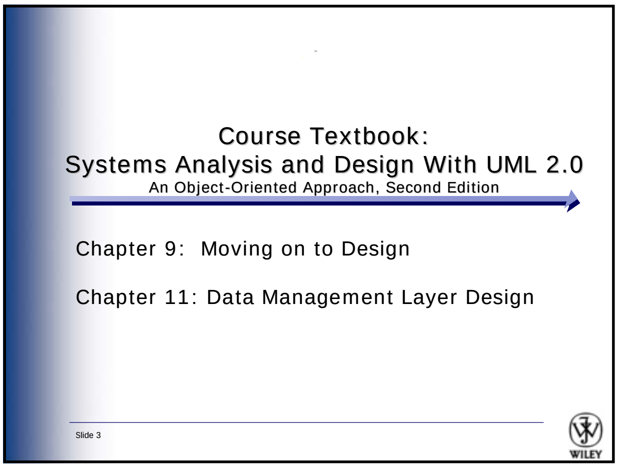
CS183 Spring Semester 2008

Dr. Jonathan Y. Clark

Email: j.y.clark@surrey.ac.uk

Course Website: www.computing.surrey.ac.uk/personal /st/J.Y.Clark/teaching/sad/cs183.html



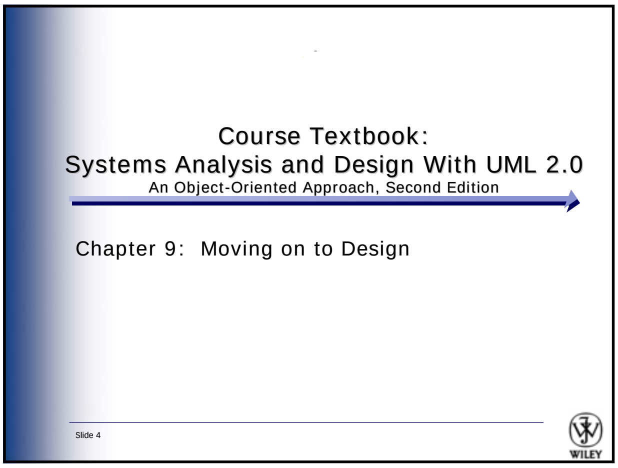


Course Textbook: Systems Analysis and Design With UML 2.0

An Object-Oriented Approach, Second Edition

Chapter 9: Moving on to Design

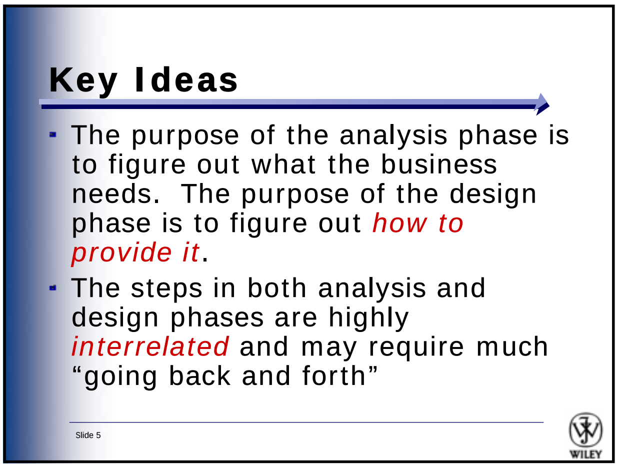
Chapter 11: Data Management Layer Design



Course Textbook: Systems Analysis and Design With UML 2.0

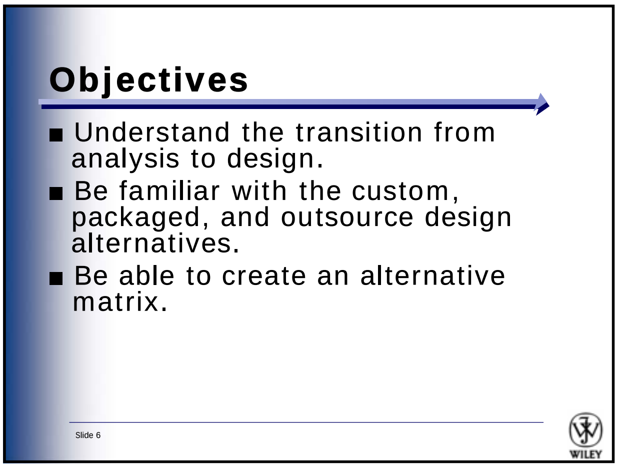
An Object-Oriented Approach, Second Edition

Chapter 9: Moving on to Design



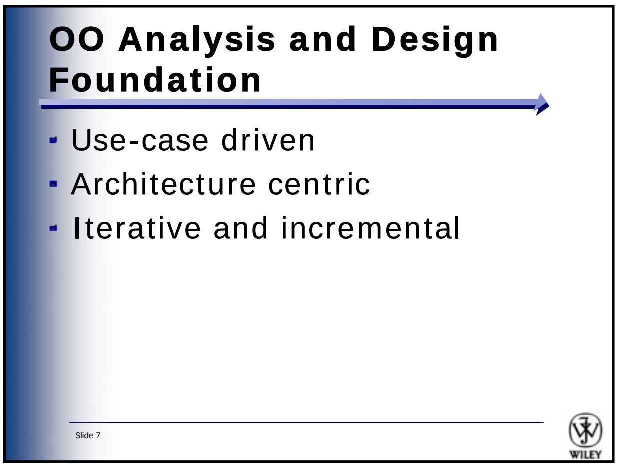
Key Ideas

- The purpose of the analysis phase is to figure out what the business needs. The purpose of the design phase is to figure out *how to provide it*.
- The steps in both analysis and design phases are highly *interrelated* and may require much “going back and forth”



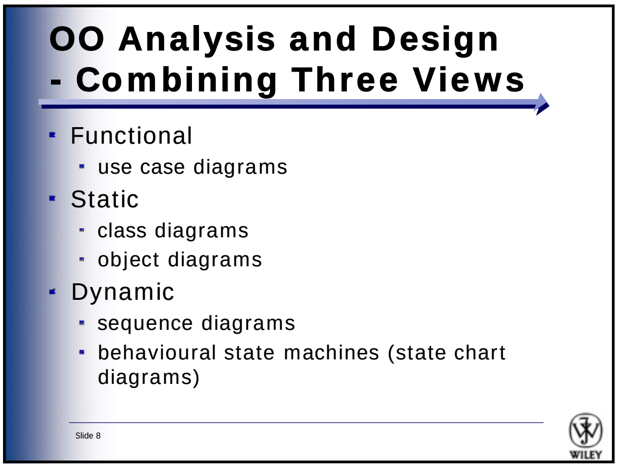
Objectives

- Understand the transition from analysis to design.
- Be familiar with the custom, packaged, and outsource design alternatives.
- Be able to create an alternative matrix.



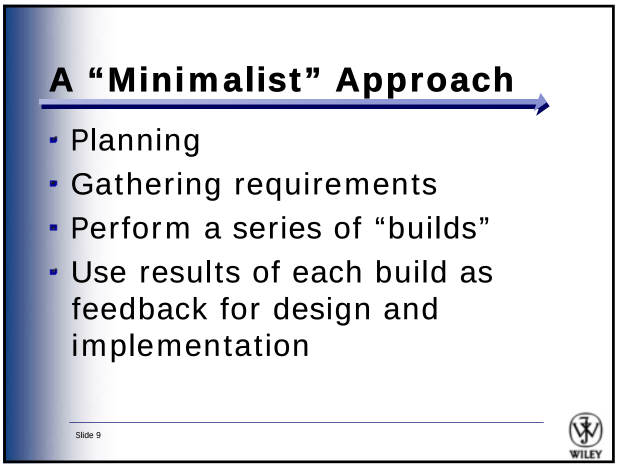
OO Analysis and Design Foundation

- Use-case driven
- Architecture centric
- Iterative and incremental



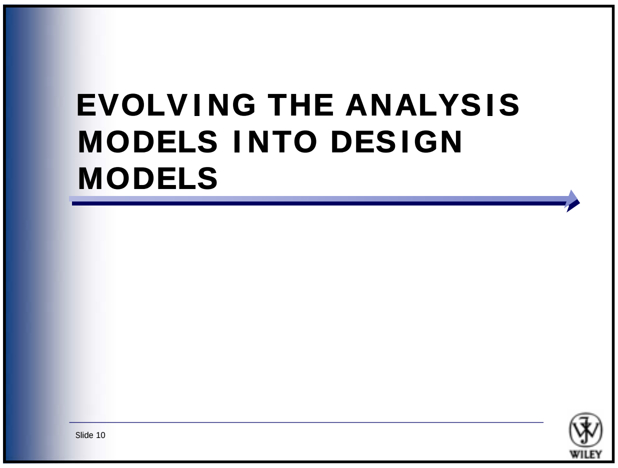
OO Analysis and Design - Combining Three Views

- Functional
  - use case diagrams
- Static
  - class diagrams
  - object diagrams
- Dynamic
  - sequence diagrams
  - behavioural state machines (state chart diagrams)

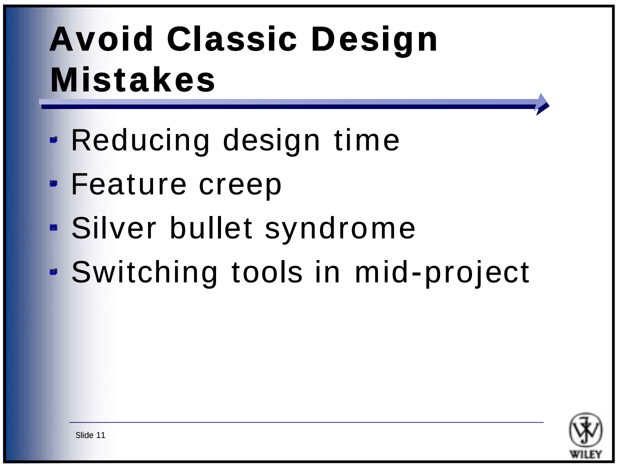


A “Minimalist” Approach

- Planning
- Gathering requirements
- Perform a series of “builds”
- Use results of each build as feedback for design and implementation

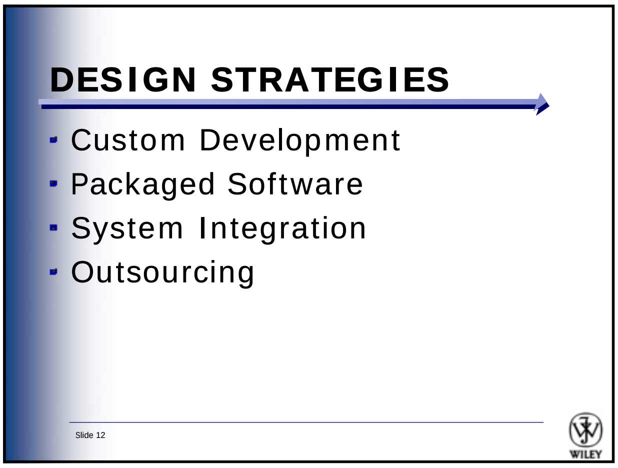


EVOLVING THE ANALYSIS MODELS INTO DESIGN MODELS



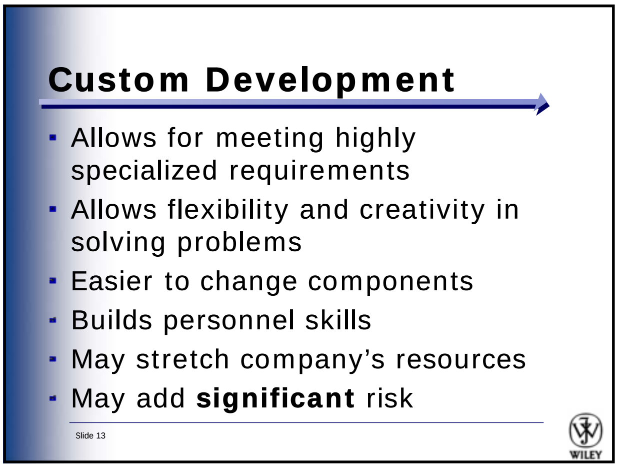
Avoid Classic Design Mistakes

- Reducing design time
- Feature creep
- Silver bullet syndrome
- Switching tools in mid-project



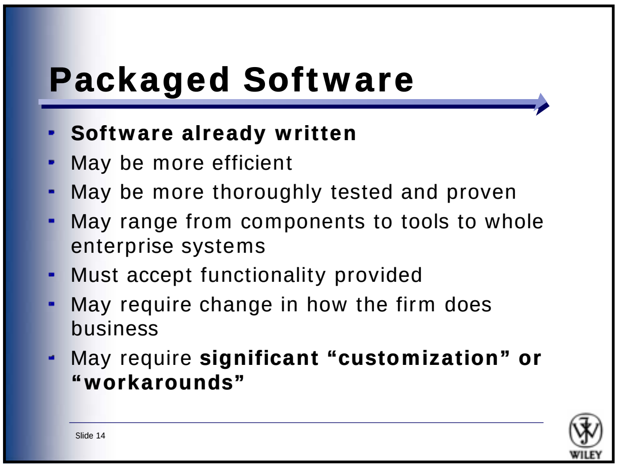
DESIGN STRATEGIES

- Custom Development
- Packaged Software
- System Integration
- Outsourcing



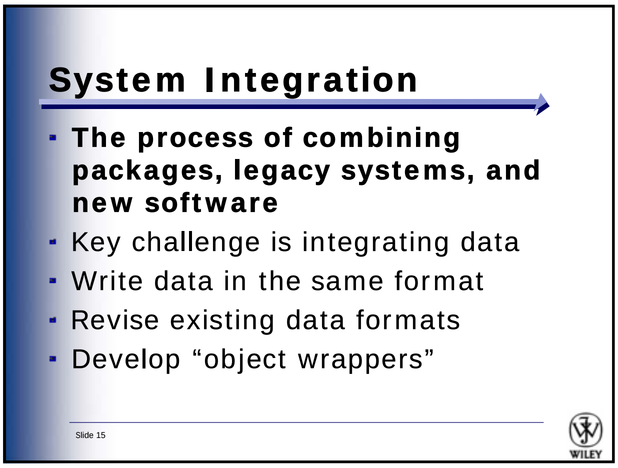
Custom Development

- Allows for meeting highly specialized requirements
- Allows flexibility and creativity in solving problems
- Easier to change components
- Builds personnel skills
- May stretch company's resources
- May add **significant** risk



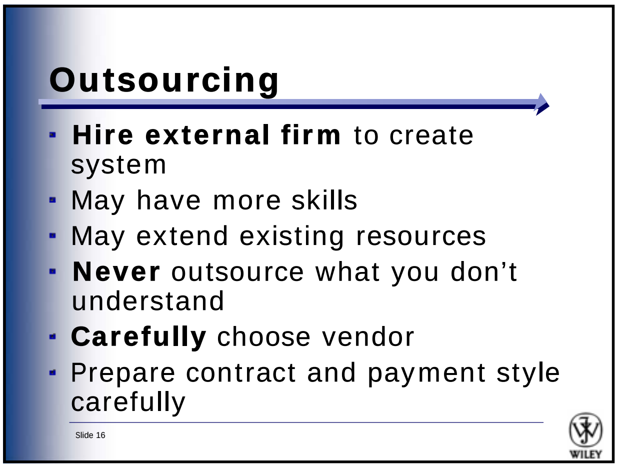
Packaged Software

- **Software already written**
- May be more efficient
- May be more thoroughly tested and proven
- May range from components to tools to whole enterprise systems
- Must accept functionality provided
- May require change in how the firm does business
- May require **significant "customization" or "workarounds"**



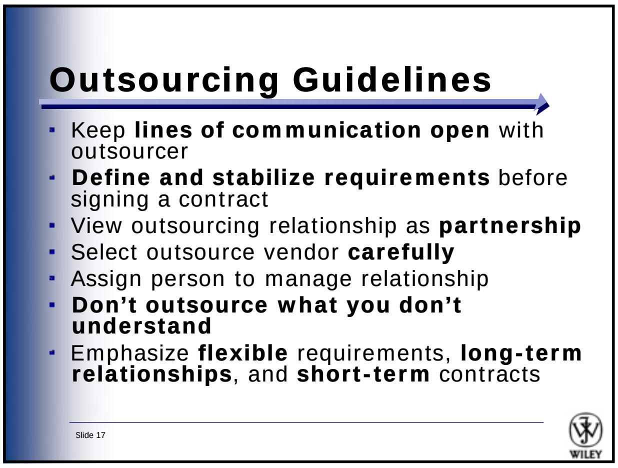
System Integration

- **The process of combining packages, legacy systems, and new software**
- Key challenge is integrating data
- Write data in the same format
- Revise existing data formats
- Develop "object wrappers"



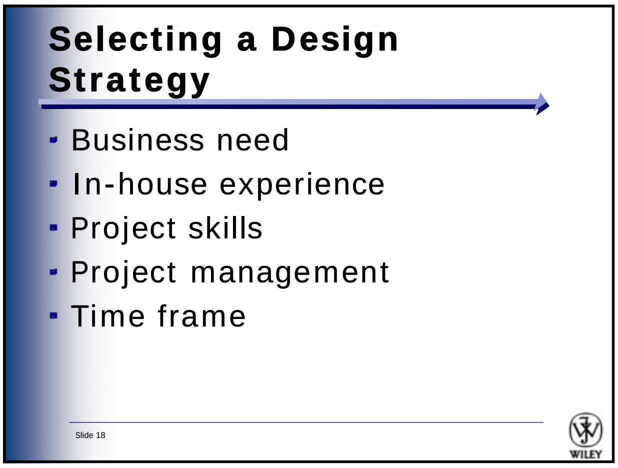
Outsourcing

- **Hire external firm** to create system
- May have more skills
- May extend existing resources
- **Never** outsource what you don't understand
- **Carefully** choose vendor
- Prepare contract and payment style carefully



Outsourcing Guidelines

- Keep **lines of communication open** with outsourcer
- **Define and stabilize requirements** before signing a contract
- View outsourcing relationship as **partnership**
- Select outsource vendor **carefully**
- Assign person to manage relationship
- **Don't outsource what you don't understand**
- Emphasize **flexible** requirements, **long-term relationships**, and **short-term** contracts



Selecting a Design Strategy

- Business need
- In-house experience
- Project skills
- Project management
- Time frame

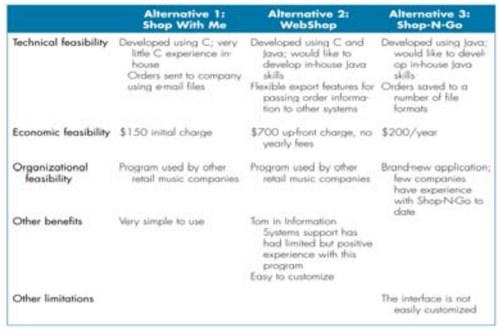
The Alternative matrix

- Combines several feasibility analyses into one grid
- Revisits technical, economic, and organizational feasibility

Request for Proposals

- Description of the system you propose to be built
- Vendors, developers, service providers respond with proposals including how they will address needs as well as stating cost and time requirements.

CD Selections



	Alternative 1: Shop With Me	Alternative 2: WebShop	Alternative 3: Shop-N-Go
Technical feasibility	Developed using C; very little C experience in-house Orders sent to company using e-mail files	Developed using C and Java; would like to develop in-house Java skills Flexible export features for passing order information to other systems	Developed using Java; would like to develop in-house Java skills Orders saved to a number of file formats
Economic feasibility	$150 initial charge	$700 up-front charge, no yearly fees	$200/year
Organizational feasibility	Program used by other retail music companies	Program used by other retail music companies	Brand-new application; few companies have experience with Shop-N-Go to date
Other benefits	Very simple to use	Tom in Information Systems support has had limited but positive experience with this program Easy to customize	
Other limitations			The interface is not easily customized

Course Textbook:
Systems Analysis and Design With UML 2.0
An Object-Oriented Approach, Second Edition

Chapter 11: Data Management Layer Design

Objectives

- Become familiar with several object-persistence formats.
- Be able to apply the steps of normalization to a relational database.
- Be able to optimize a relational database for object storage and access.

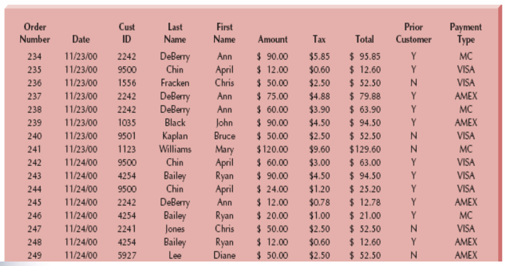
Data Management Layer

- Choose object-persistence format to support the system
  - Problem domain objects drive object storage design
- Design of Data Storage
  - Must optimize processing efficiency
- Data access and manipulation
  - Separate problem domain classes from storage format
  - Handle all communication with the database

Object Persistence Formats

- Files (Sequential and Random)
- Relational databases
- Object-relational databases
- Object-oriented databases

Customer Order File



Order Number	Date	Cust ID	Last Name	First Name	Amount	Tax	Total	Prior Customer	Payment Type
234	11/23/00	2242	DeBerry	Ann	$ 90.00	$5.85	$ 95.85	Y	MC
235	11/23/00	9500	Chin	April	$ 12.00	$0.60	$ 12.60	Y	VISA
236	11/23/00	1556	Fracken	Chris	$ 50.00	$2.50	$ 52.50	N	VISA
237	11/23/00	2242	DeBerry	Ann	$ 75.00	$4.88	$ 79.88	Y	AMEX
238	11/23/00	2242	DeBerry	Ann	$ 60.00	$3.90	$ 63.90	Y	MC
239	11/23/00	1035	Black	John	$ 90.00	$4.50	$ 94.50	Y	AMEX
240	11/23/00	9501	Kaplan	Bruce	$ 50.00	$2.50	$ 52.50	N	VISA
241	11/23/00	1123	Williams	Mary	$120.00	$9.60	$129.60	N	MC
242	11/24/00	9500	Chin	April	$ 60.00	$3.00	$ 63.00	Y	VISA
243	11/24/00	4254	Bailey	Ryan	$ 90.00	$4.50	$ 94.50	Y	VISA
244	11/24/00	9500	Chin	April	$ 24.00	$1.20	$ 25.20	Y	VISA
245	11/24/00	2242	DeBerry	Ann	$ 12.00	$0.78	$ 12.78	Y	AMEX
246	11/24/00	4254	Bailey	Ryan	$ 20.00	$1.00	$ 21.00	Y	MC
247	11/24/00	2241	Jones	Chris	$ 50.00	$2.50	$ 52.50	N	VISA
248	11/24/00	4254	Bailey	Ryan	$ 12.00	$0.60	$ 12.60	Y	AMEX
249	11/24/00	5927	Lee	Diane	$ 50.00	$2.50	$ 52.50	N	AMEX

Sequential Access Files

- Sequential access files allow sequential operations
  - Read, write, and search
  - Efficient for report writing
  - Searches are not efficient because an average of 50% of records have to be accessed
  - Two versions
    - Ordered
    - unordered

Random Access Files

- Allow only random or direct file operations
- Good for finding and updating a specific object
- Inefficient report writing

Application File Types

- Master Files
- Look-up files
- Transaction files
- Audit file
- History file

Relational Databases

- Collection of tables
  - Comprised of fields that define entities
  - Primary key has unique values in each row of a table
  - Foreign key is primary key of another table
- Tables related to each other
  - Primary key field of a table is a field of another table and called a foreign key
  - Relationship established by a foreign key of one table connecting to the primary key of another table

Database Management System (DBMS)

- Software that creates and manipulates a database
- RDBMS is a DBMS for a relational database
- RDBMS usually support Referential Integrity

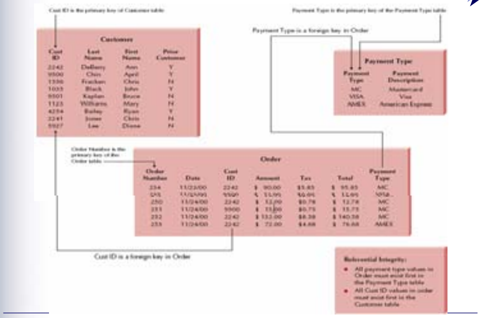
Referential Integrity

- the idea of ensuring that values linking the tables together through the primary and foreign keys are valid and correctly synchronized.

Referential Integrity Example

- Cust. ID is a primary key for the customer table
- Cust. ID is a foreign key for the order table
- A violation of referential integrity would happen if an order was entered in the order table for a Cust. ID that had not been entered into the customer table first
- An RDBMS prevents such a record from being entered

Example of Referential Integrity


Cust ID is the primary key of Customer table
Payment Type is the primary key of the Payment Type table
Payment Type is a foreign key in Order
Order Number is the primary key of the Order table
Cust ID is a foreign key in Order

Customer

Cust ID	Last Name	First Name	Prior Customer
2242	DeBerry	Ann	Y
9500	Chin	April	Y
1336	Fracken	Chris	N
1033	Black	John	Y
9501	Kaplan	Bruce	N
1123	Williams	Mary	N
4254	Bailey	Ryan	Y
2241	Jones	Chris	N
5927	Lee	Diane	N

Payment Type

Payment Type	Payment Description
MC	Mastercard
VISA	Visa
AMEX	American Express

Order

Order Number	Date	Cust ID	Amount	Tax	Total	Payment Type
234	11/23/00	2242	$ 90.00	$5.85	$ 95.85	MC
235	11/23/00	9500	$ 11.00	$0.65	$ 11.65	VISA
250	11/24/00	2242	$ 12.00	$0.78	$ 12.78	MC
251	11/24/00	9500	$ 15.00	$0.75	$ 15.75	MC
252	11/24/00	2242	$ 132.00	$8.58	$ 140.58	MC
253	11/24/00	2242	$ 72.00	$4.68	$ 76.68	AMEX

Referential Integrity:

- All payment type values in Order must exist first in the Payment Type table
- All Cust ID values in order must exist first in the Customer table

Structured Query Language (SQL)

- Standard language for accessing data in tables
- SQL Commands
  - Create, edit, and delete tables
  - Add, edit, and delete data
  - Display data from one or more related tables
  - Display data computed from data in one or more related tables

Criteria for Object Persistence Formats

- Data types supported
- Types of application systems (transaction processing, DSS, …)
- Existing Storage Formats
- Future Needs
- Other miscellaneous Criteria (cost, concurrency control, …)

Mapping RDBMS Tables to Problem Domain Classes

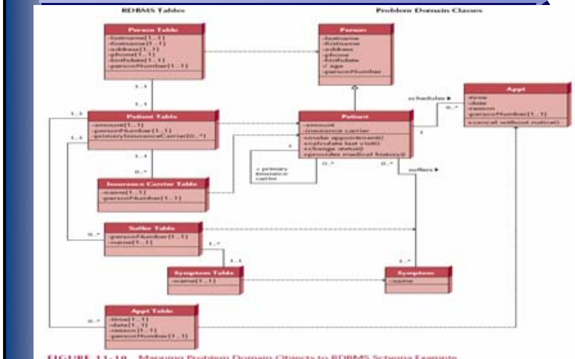
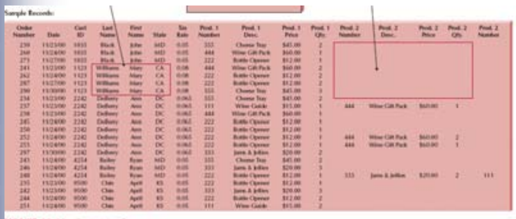


FIGURE 11-10 Mapping Problem Domain Objects to RDBMS Schema Example

Optimize RDBMS Object Storage

- Redundant data
  - Wastes space
  - Allow more room for error
- Null values in tables
  - Difficult to interpret

Example of Non-normalized Data



Sample Records:

Order Number	Date	Cust ID	Last Name	First Name	State	Tax Rate	Prod. 1 Number	Prod. 1 Desc.	Prod. 1 Price	Prod. 1 Qty.	Prod. 2 Number	Prod. 2 Desc.	Prod. 2 Price	Prod. 2 Qty.	Prod. 2 Number
239	11/23/00	1035	Black	John	MD	0.05	555	Cheese Tray	$45.00	2					
260	11/24/00	1035	Black	John	MD	0.05	444	Wine Gift Pack	$60.00	1					
273	11/27/00	1035	Black	John	MD	0.05	222	Bottle Opener	$12.00	1					
241	11/23/00	1123	Williams	Mary	CA	0.08	444	Wine Gift Pack	$60.00	2					
262	11/24/00	1123	Williams	Mary	CA	0.08	222	Bottle Opener	$12.00	2					
287	11/27/00	1123	Williams	Mary	CA	0.08	222	Bottle Opener	$12.00	2					
290	11/30/00	1123	Williams	Mary	CA	0.08	555	Cheese Tray	$45.00	3					
234	11/23/00	2242	DeBerry	Ann	DC	0.065	555	Cheese Tray	$45.00	2					
237	11/23/00	2242	DeBerry	Ann	DC	0.065	111	Wine Guide	$15.00	1	444	Wine Gift Pack	$60.00	1	
238	11/23/00	2242	DeBerry	Ann	DC	0.065	444	Wine Gift Pack	$60.00	1					
245	11/24/00	2242	DeBerry	Ann	DC	0.065	222	Bottle Opener	$12.00	1					
250	11/24/00	2242	DeBerry	Ann	DC	0.065	222	Bottle Opener	$12.00	1					
252	11/24/00	2242	DeBerry	Ann	DC	0.065	222	Bottle Opener	$12.00	1	444	Wine Gift Pack	$60.00	2	
253	11/24/00	2242	DeBerry	Ann	DC	0.065	222	Bottle Opener	$12.00	1	444	Wine Gift Pack	$60.00	1	
297	11/30/00	2242	DeBerry	Ann	DC	0.065	333	Jams & Jellies	$20.00	2					
243	11/24/00	4254	Bailey	Ryan	MD	0.05	555	Cheese Tray	$45.00	2					
246	11/24/00	4254	Bailey	Ryan	MD	0.05	333	Jams & Jellies	$20.00	3					
248	11/24/00	4254	Bailey	Ryan	MD	0.05	222	Bottle Opener	$12.00	1	333	Jams & Jellies	$20.00	2	111
235	11/23/00	9500	Chin	April	KS	0.05	222	Bottle Opener	$12.00	1					
242	11/23/00	9500	Chin	April	KS	0.05	333	Jams & Jellies	$20.00	3					
244	11/24/00	9500	Chin	April	KS	0.05	222	Bottle Opener	$12.00	2					
251	11/24/00	9500	Chin	April	KS	0.05	111	Wine Guide	$15.00	2					

FIGURE 11-11 Optimizing Storage

Normalization

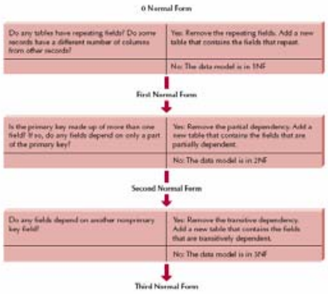
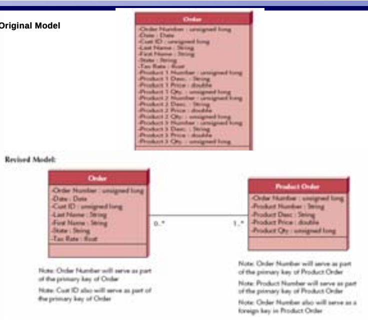

0 Normal Form
Do any tables have repeating fields? Do some records have a different number of columns from other records?
Yes: Remove the repeating fields. Add a new table that contains the fields that repeat.
No: The data model is in 1NF
First Normal Form
Is the primary key made up of more than one field? If so, do any fields depend on only a part of the primary key?
Yes: Remove the partial dependency. Add a new table that contains the fields that are partially dependent.
No: The data model is in 2NF
Second Normal Form
Do any fields depend on another nonprimary key field?
Yes: Remove the transitive dependency. Add a new table that contains the fields that are transitively dependent.
No: The data model is in 3NF
Third Normal Form

FIGURE 11-12 The Steps of Normalization

Normalization Example

Original Model


Order
-Order Number : unsigned long
-Date : Date
-Cust ID : unsigned long
-Last Name : String
-First Name : String
-State : String
-Tax Rate : float
-Product 1 Number : unsigned long
-Product 1 Desc. : String
-Product 1 Price : double
-Product 1 Qty. : unsigned long
-Product 2 Number : unsigned long
-Product 2 Desc. : String
-Product 2 Price : double
-Product 2 Qty. : unsigned long
-Product 3 Number : unsigned long
-Product 3 Desc. : String
-Product 3 Price : double
-Product 3 Qty. : unsigned long
Revised Model:
Order
-Order Number : unsigned long
-Date : Date
-Cust ID : unsigned long
-Last Name : String
-First Name : String
-State : String
-Tax Rate : float
0..*
1..*
Product Order
-Order Number : unsigned long
-Product Number : String
-Product Desc : String
-Product Price : double
-Product Qty : unsigned long
Note: Order Number will serve as part of the primary key of Order
Note: Cust ID also will serve as part of the primary key of Order
Note: Order Number will serve as part of the primary key of Product Order
Note: Product Number will serve as part of the primary key of Product Order
Note: Order Number also will serve as a foreign key in Product Order

Normalization

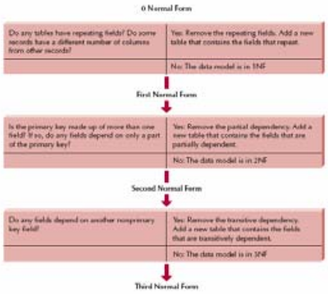
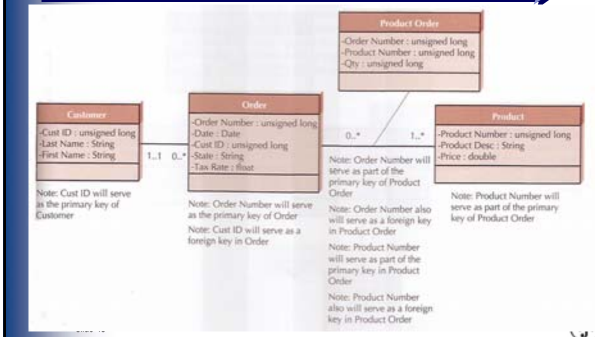

0 Normal Form
Do any tables have repeating fields? Do some records have a different number of columns from other records?
Yes: Remove the repeating fields. Add a new table that contains the fields that repeat.
No: The data model is in 1NF
First Normal Form
Is the primary key made up of more than one field? If so, do any fields depend on only a part of the primary key?
Yes: Remove the partial dependency. Add a new table that contains the fields that are partially dependent.
No: The data model is in 2NF
Second Normal Form
Do any fields depend on another nonprimary key field?
Yes: Remove the transitive dependency. Add a new table that contains the fields that are transitively dependent.
No: The data model is in 3NF
Third Normal Form

FIGURE 11-12 The Steps of Normalization

2NF Normalized Model



Note: Cust ID will serve as the primary key of Customer

Note: Order Number will serve as the primary key of Order

Note: Cust ID will serve as a foreign key in Order

Note: Order Number will serve as part of the primary key of Product Order

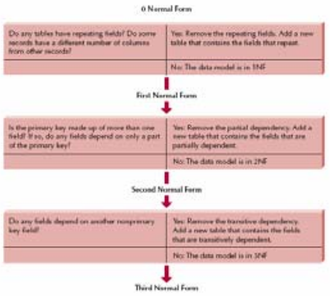
Note: Order Number also will serve as a foreign key in Product Order

Note: Product Number will serve as part of the primary key in Product Order

Note: Product Number also will serve as a foreign key in Product Order

Note: Product Number will serve as part of the primary key of Product Order

Normalization



Do any tables have repeating fields? Do some records have a different number of columns from other records?

Yes: Remove the repeating fields. Add a new table that contains the fields that repeat.

No: The data model is in 1NF

Is the primary key made up of more than one field? If so, do any fields depend on only a part of the primary key?

Yes: Remove the partial dependency. Add a new table that contains the fields that are partially dependent.

No: The data model is in 2NF

Do any fields depend on another nonprimary key field?

Yes: Remove the transitive dependency. Add a new table that contains the fields that are transitively dependent.

No: The data model is in 3NF

FIGURE 11-12 The Steps of Normalization

3NF Normalized Model

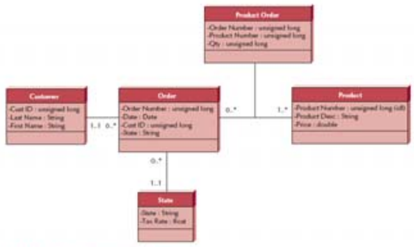


FIGURE 11-15 3NF Normalized Model

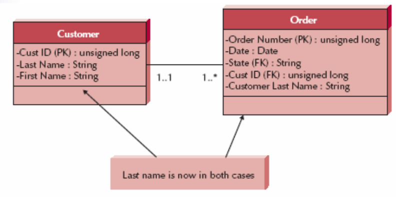
Problems with RDBMS

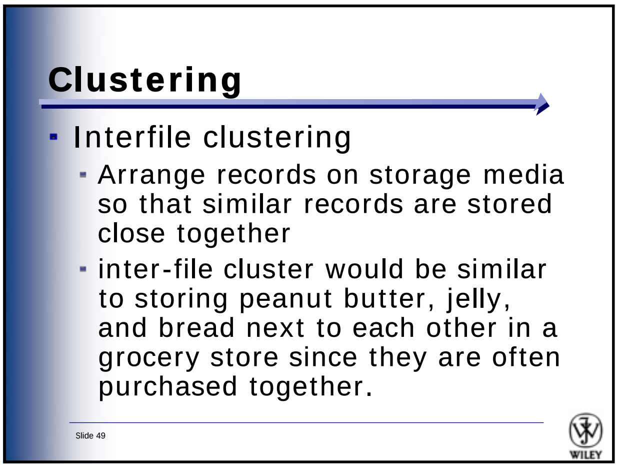
- To access data in multiple tables, the tables must be joined
- This can result in many database operations and lead to huge tables and slow processing

Speeding up access

- Denormalization – Adds data from one table to another in order to speed processing and eliminate a join operation
- Example: Add customer last name to order table to avoid joining order to customer to get just last name

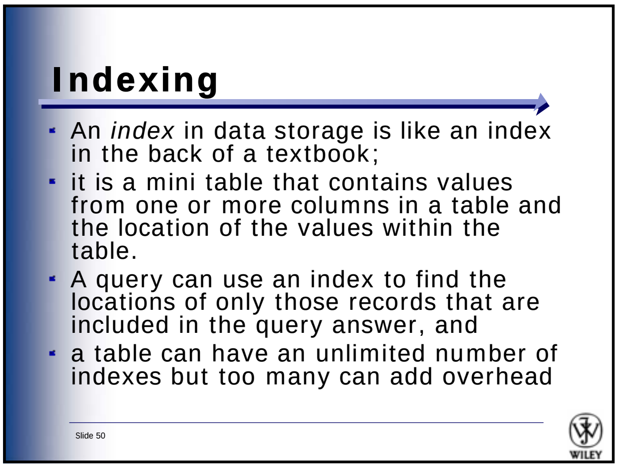
Example


Customer
-Cust ID (PK) : unsigned long
-Last Name : String
-First Name : String
1..1
1..*
Order
-Order Number (PK) : unsigned long
-Date : Date
-State (FK) : String
-Cust ID (FK) : unsigned long
-Customer Last Name : String
Last name is now in both cases



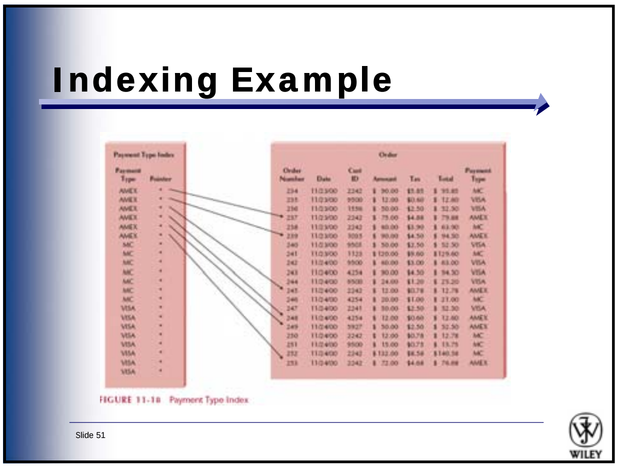
Clustering

- Interfile clustering
  - Arrange records on storage media so that similar records are stored close together
  - inter-file cluster would be similar to storing peanut butter, jelly, and bread next to each other in a grocery store since they are often purchased together.



Indexing

- An *index* in data storage is like an index in the back of a textbook;
- it is a mini table that contains values from one or more columns in a table and the location of the values within the table.
- A query can use an index to find the locations of only those records that are included in the query answer, and
- a table can have an unlimited number of indexes but too many can add overhead



Indexing Example

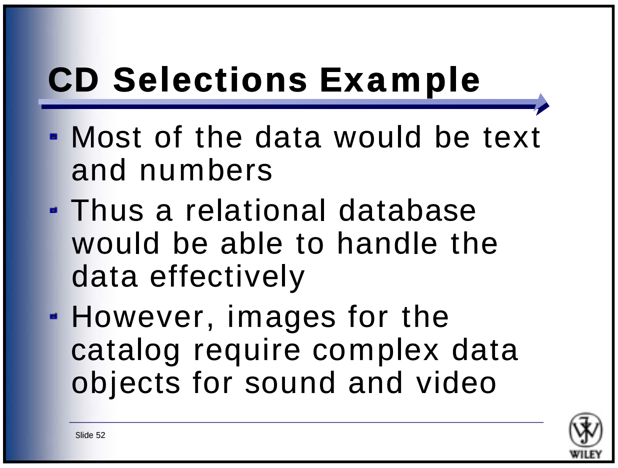
Payment Type Index

Payment Type	Pointer
AMEX	•
AMEX	•
AMEX	•
AMEX	•
AMEX	•
AMEX	•
MC	•
MC	•
MC	•
MC	•
MC	•
MC	•
MC	•
VISA	•
VISA	•
VISA	•
VISA	•
VISA	•
VISA	•
VISA	•
VISA	•

Order

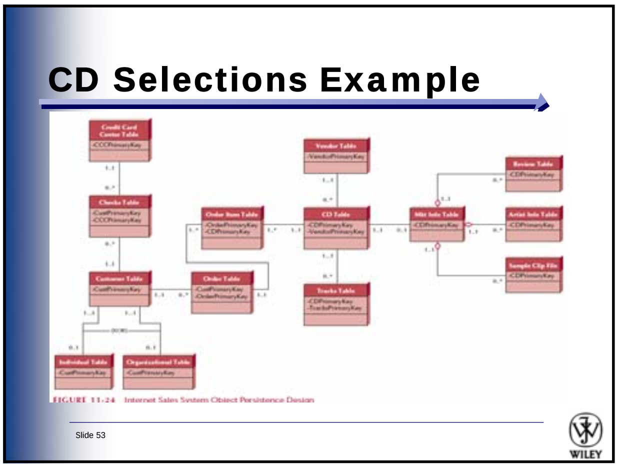
Order Number	Date	Cust ID	Amount	Tax	Total	Payment Type
234	11/23/00	2242	$ 90.00	$5.85	$ 95.85	MC
235	11/23/00	9500	$ 12.00	$0.60	$ 12.60	VISA
236	11/23/00	1556	$ 50.00	$2.50	$ 52.50	VISA
237	11/23/00	2242	$ 75.00	$4.88	$ 79.88	AMEX
238	11/23/00	2242	$ 60.00	$3.90	$ 63.90	MC
239	11/23/00	3005	$ 90.00	$4.50	$ 94.50	AMEX
240	11/23/00	9505	$ 50.00	$2.50	$ 52.50	VISA
241	11/23/00	1123	$120.00	$9.60	$129.60	MC
242	11/24/00	9500	$ 60.00	$3.00	$ 63.00	VISA
243	11/24/00	4254	$ 90.00	$4.50	$ 94.50	VISA
244	11/24/00	9500	$ 24.00	$1.20	$ 25.20	VISA
245	11/24/00	2242	$ 12.00	$0.78	$ 12.78	AMEX
246	11/24/00	4254	$ 20.00	$1.00	$ 21.00	MC
247	11/24/00	2241	$ 50.00	$2.50	$ 52.50	VISA
248	11/24/00	4254	$ 12.00	$0.60	$ 12.60	AMEX
249	11/24/00	5927	$ 50.00	$2.50	$ 52.50	AMEX
250	11/24/00	2242	$ 12.00	$0.78	$ 12.78	MC
251	11/24/00	9500	$ 15.00	$0.75	$ 15.75	MC
252	11/24/00	2242	$132.00	$8.58	$140.58	MC
253	11/24/00	2242	$ 72.00	$4.68	$ 76.68	AMEX

FIGURE 11-18 Payment Type Index



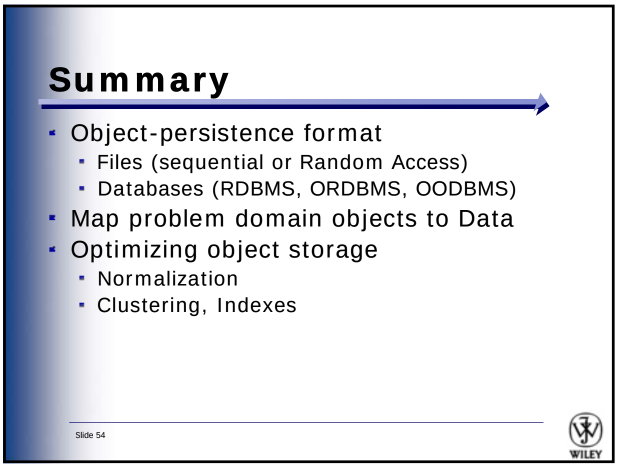
CD Selections Example

- Most of the data would be text and numbers
- Thus a relational database would be able to handle the data effectively
- However, images for the catalog require complex data objects for sound and video



CD Selections Example

FIGURE 11-24 Internet Sales System Object Persistence Design



Summary

- Object-persistence format
  - Files (sequential or Random Access)
  - Databases (RDBMS, ORDBMS, OODBMS)
- Map problem domain objects to Data
- Optimizing object storage
  - Normalization
  - Clustering, Indexes