Universitas Esa Unggul

Smart, Creative and Entrepreneurial

CCA220-Analisis dan Perancangan system Informasi

www.esaunggul.ac.id

Dosen Pengampu :

5165-Kundang K Juman, Ir, MMSI

Prodi Teknik Informatika dan Sistem Informasi - Fakultas Ilmu Komputer

Systems Analysis and Design

5th Edition

Chapter 11. Data Storage Design

Alan Dennis, Barbara Haley Wixom, and Roberta Roth

Chapter 11 Outline

- Data storage formats.
  - Files.
  - Databases.
- Moving from logical to physical data models.
- Optimizing data storage.

Systems Design

- Architecture design
- User Interface design
- Program Design
- Data storage design

INTRODUCTION

- The **data storage function** is concerned with how data is stored and handled by programs that run the system.
- Data storage design is to
  - select the data storage format;
  - convert the logical data model created during analysis into a ***physical data model*** to reflect the implementation decision;
  - ensure that DFDs and ERDs balance; and
  - design the selected data storage format to optimize its processing efficiency.

DATA STORAGE FORMATS

- There are two types of data storage formats:
  - ***Files:*** electronic lists of data that have been optimized to perform a particular transaction.
  - ***Database***: a collection of groupings of information that are related to each other in some way.
- A ***Database Management System (DBMS)*** is software that creates and manipulates the databases.

Example of file: appointment File (cont'd)

Appointment Date	Appointment Time	Duration	Reason	Patient ID	First Name	Last Name	Phone Number	Doctor ID	Doctor Last Name
11/23/2009	2:30	.25 hour	Flu	758843	Patrick	Dennis	548-9456	V524625587	Vroman
11/23/2009	2:30	1 hour	Physical	136136	Adelaide	Kin	548-7887	T445756225	Tantalo
11/23/2009	2:45	.25 hour	Shot	544822	Chris	Pullig	525-5464	V524625587	Vroman
11/23/2009	3:00	1 hour	Physical	345344	Felicia	Marston	548-9333	B544742245	Brousseau
11/23/2009	3:00	.5 hour	Migraine	236454	Thomas	Bateman	667-8955	V524625587	Vroman
11/23/2009	3:30	.5 hour	Muscular	887777	Ryan	Nelson	525-4772	V524625587	Vroman
11/23/2009	3:30	.25 hour	Shot	966233	Peter	Todd	667-2325	T445756225	Tantalo
11/23/2009	3:45	.75 hour	Muscular	951657	Mike	Morris	663-8944	T445756225	Tantalo
11/23/2009	4:00	1 hour	Physical	223238	Ellen	Whitener	525-8874	B544742245	Brousseau
11/23/2009	4:00	.5 hour	Flu	365548	Jerry	Starsia	548-9887	V524625587	Vroman
11/23/2009	4:30	1 hour	Minor surg	398633	Susan	Perry	525-6632	V524625587	Vroman
11/23/2009	4:30	.5 hour	Migraine	222577	Elizabeth	Gray	667-8400	T445756225	Tantalo
11/24/2009	8:30	.25 hour	Shot	858756	Elias	Awad	663-6364	T445756225	Tantalo
11/24/2009	8:30	1 hour	Minor surg	232158	Andy	Ruppel	525-9888	V524625587	Vroman
11/24/2009	8:30	.25 hour	Flu	244875	Rick	Grenci	548-2114	B544742245	Brousseau
11/24/2009	8:45	.5 hour	Muscular	655683	Eric	Meier	667-0254	T445756225	Tantalo
11/24/2009	8:45	1 hour	Physical	447521	Jane	Pace	548-0025	B544742245	Brousseau
11/24/2009	9:30	.5 hour	Flu	554263	Trey	Maxham	663-8547	V524625587	Vroman

FIGURE 11-1
Appointment File

Example of database: Appointment database (cont'd)

Appointment Date	Appointment Time	Duration	Reason	Patient ID	Doctor ID
11/23/2009	2:30	.5 hour	Flu	758843	V524625587
11/23/2009	2:30	1 hour	Physical	136136	T445756225
11/23/2009	2:45	.25 hour	Shot	544822	V524625587
11/23/2009	3:00	1 hour	Physical	345344	B544742245
11/23/2009	3:00	.5 hour	Migraine	236454	V524625587
11/23/2009	3:30	.5 hour	Muscular	887777	V524625587
11/23/2009	3:30	.25 hour	Shot	966233	T445756225
11/23/2009	3:45	.75 hour	Muscular	951657	T445756225
11/23/2009	4:00	1 hour	Physical	223238	B544742245
11/23/2009	4:00	.5 hour	Flu	365548	V524625587
11/23/2009	4:30	1 hour	Minor surg	398633	V524625587
11/23/2009	4:30	.5 hour	Migraine	222577	T445756225
11/24/2009	8:30	.25 hour	Shot	858756	T445756225
11/24/2009	8:30	1 hour	Minor surg	232158	V524625587
11/24/2009	8:30	.25 hour	Flu	244875	B544742245
11/24/2009	8:45	.5 hour	Muscular	655683	T445756225
11/24/2009	8:45	1 hour	Physical	447521	B544742245
11/24/2009	9:30	.5 hour	Flu	554263	V524625587

Change all dates to 2012

Tables related by patient ID

Tables related by doctor ID

Patient ID	First Name	Last Name	Phone Number
136136	Adelaide	Kin	548-7887
222577	Elizabeth	Gray	667-8400
223238	Ellen	Whitener	525-8874
232158	Andy	Ruppel	525-9888
236454	Thomas	Bateman	667-8955
244875	Rick	Grenci	548-2114
345344	Felicia	Marston	548-9333
365548	Jerry	Starsia	548-9887
398633	Susan	Perry	525-6632
447521	Jane	Pace	548-0025
544822	Chris	Pullig	525-5464
554263	Trey	Maxham	663-8547
655683	Eric	Meier	667-0254
758843	Patrick	Dennis	548-9456
858756	Elias	Awad	663-6364
887777	Ryan	Nelson	525-4772
951657	Mike	Morris	663-8944
966233	Peter	Todd	667-2325

Doctor ID	Last Name
B544742245	Brousseau
T445756225	Tantalo
V524625587	Vroman

FIGURE 11-2
Appointment Database

Files

- A ***data file*** contains an electronic list of information that is formatted for a particular transaction.
- Typically, files are organized sequentially.
- Records can be associated with other records by pointers.
- Sometimes files are called ***linked Lists*** because of the way the records are linked together using pointers.

(cont'd)

- There are several types of files:
  - ***Master files*** – store core information that is important to the application.
  - ***Look-up files*** – contain static values.
  - ***Transaction files*** – store information that can be used to update a master file.
  - ***Audit files*** – record "before" and "after" images of data as the data are altered.
  - ***History files*** (or archive files) – store past transactions.

Appointment File

Appointment Date	Appointment Time	Duration	Reason	Patient ID	First Name	Last Name	Phone Number	Doctor ID	Doctor Last Name
11/23/2003	2:30	.25 hour	Flu	758843	Patrick	Dennis	548-9456	V524625587	Vroman
11/23/2003	2:30	1 hour	Physical	136136	Adelaide	Kin	548-7887	T445756225	Tantalo
11/23/2003	2:45	.25 hour	Shot	544822	Chris	Pullig	525-5464	V524625587	Vroman
11/23/2003	3:00	1 hour	Physical	345344	Felicia	Marston	548-9333	B544742245	Brousseau
11/23/2003	3:00	.5 hour	Migraine	236454	Thomas	Bateman	667-8955	V524625587	Vroman
11/23/2003	3:30	.5 hour	Muscular	887777	Ryan	Nelson	525-4772	V524625587	Vroman
11/23/2003	3:30	.25 hour	Shot	966233	Peter	Todd	667-2325	T445756225	Tantalo
11/23/2003	3:45	.75 hour	Muscular	951657	Mike	Morris	663-8944	T445756225	Tantalo
11/23/2003	4:00	1 hour	Physical	223238	Ellen	Whitener	525-8874	B544742245	Brousseau
11/23/2003	4:00	.5 hour	Flu	365548	Jerry	Starsia	548-9887	V524625587	Vroman
11/23/2003	4:30	1 hour	Minor surg	398633	Susan	Perry	525-6632	V524625587	Vroman
11/23/2003	4:30	.5 hour	Migraine	222577	Elizabeth	Gray	667-8400	T445756225	Tantalo
11/24/2003	8:30	.25 hour	Shot	858756	Elias	Awad	663-6364	T445756225	Tantalo
11/24/2003	8:30	1 hour	Minor surg	232158	Andy	Ruppel	525-9888	V524625587	Vroman
11/24/2003	8:30	.25 hour	Flu	244875	Rick	Grenci	548-2114	B544742245	Brousseau
11/24/2003	8:45	.5 hour	Muscular	655683	Eric	Meier	667-0254	T445756225	Tantalo
11/24/2003	8:45	1 hour	Physical	447521	Jane	Pace	548-0025	B544742245	Brousseau
11/24/2003	9:30	.5 hour	Flu	554263	Trey	Maxham	663-8547	V524625587	Vroman

Databases

- There are many types of databases:
  - Legacy database
  - Relational database
  - Object database
  - Multidimensional database

Legacy Databases

- The name of ***legacy database*** is given to those databases which are based on older technology that is seldom used to develop new applications.
- Two major types of legacy databases:

- ***Hierarchical databases*** use hierarchies, or inverted trees, to represent relationships.
- ***Network databases*** are collections of records that are related to each other through ***pointers***.

Hierarchical Database Example (cont'd)

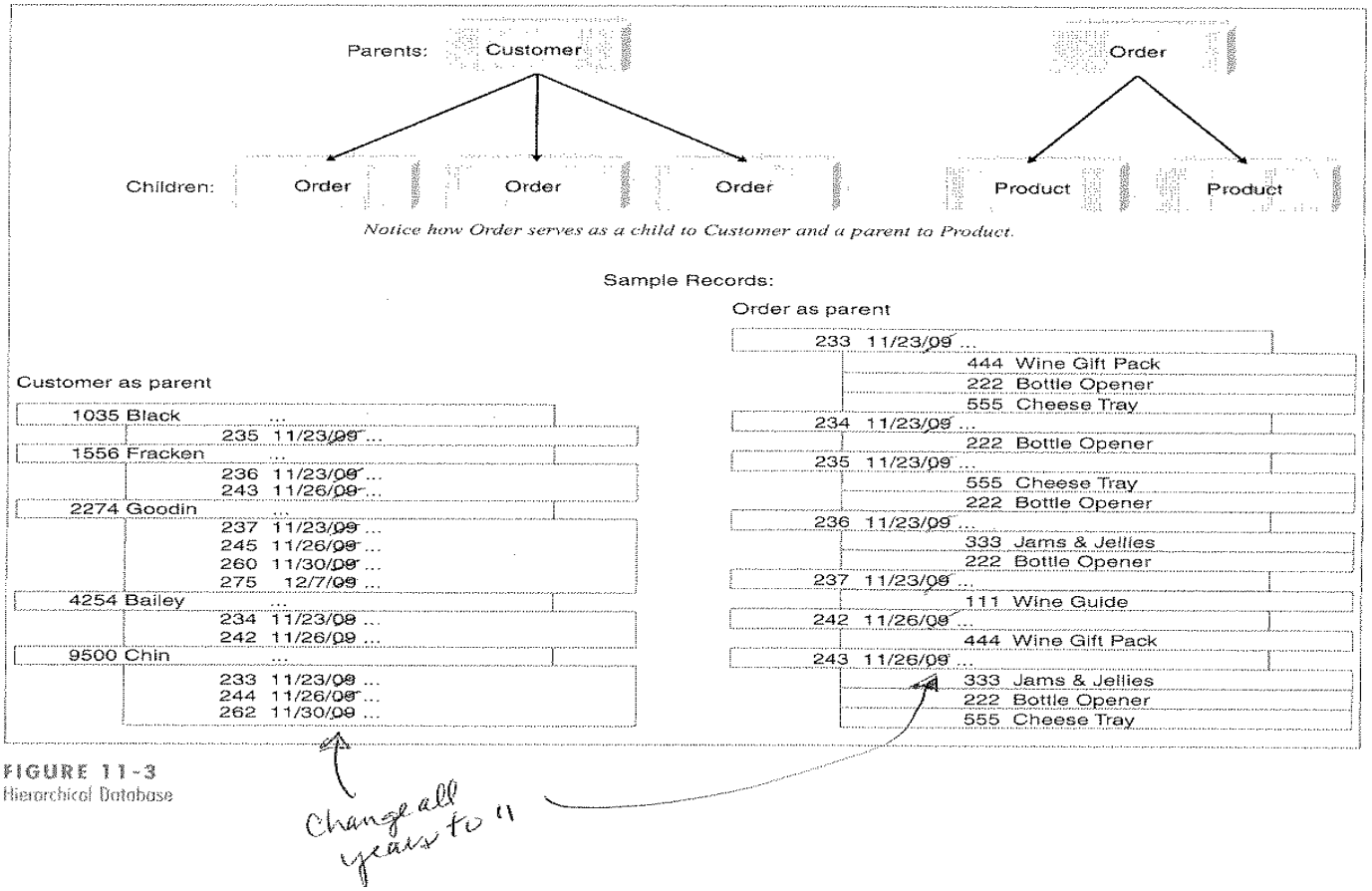
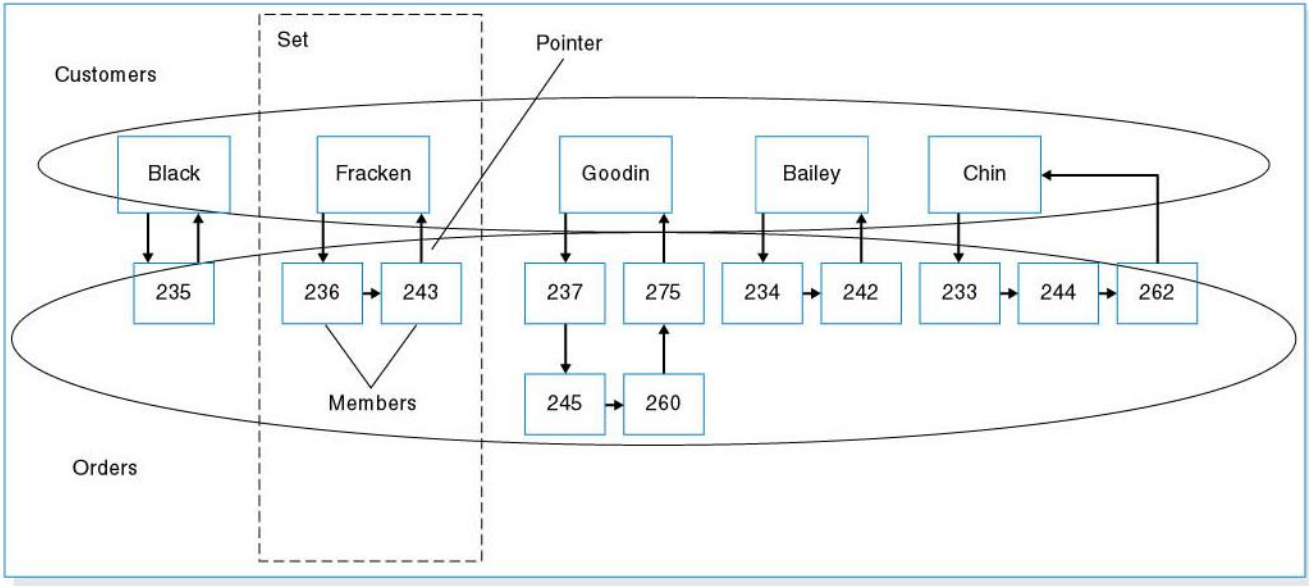


FIGURE 11-3
Hierarchical Database

Network database example (cont'd)



Relational Databases

- ***The relational database*** is the most popular kind of database for application development today.
- A relational database is based on collections of ***tables***, each of which has a ***primary key***.
- The tables are related to each other by the placement of the primary key from one table into the related table as a ***foreign key***.

Relational database example (cont'd)

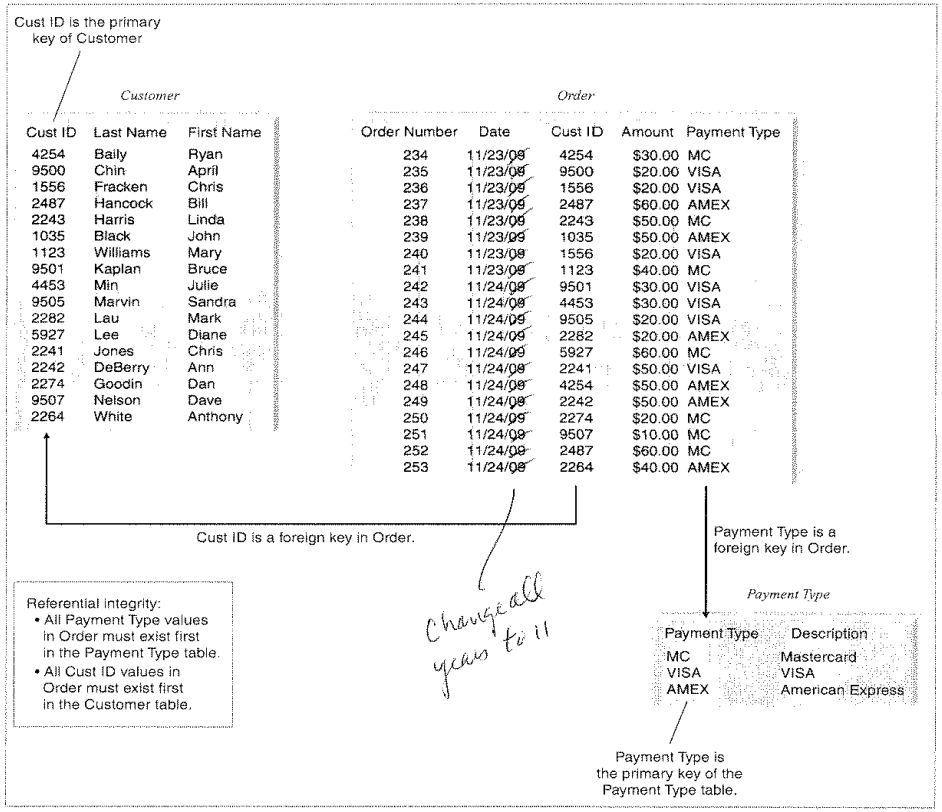


FIGURE 11-5
Relational Database

(cont'd)

- Most relational database management systems (RDBMS) support ***referential integrity***, or the idea of ensuring that values linking the tables together are valid and correctly synchronized.
- ***Structured Query Language (SQL)*** is the standard language for accessing the data in the tables.

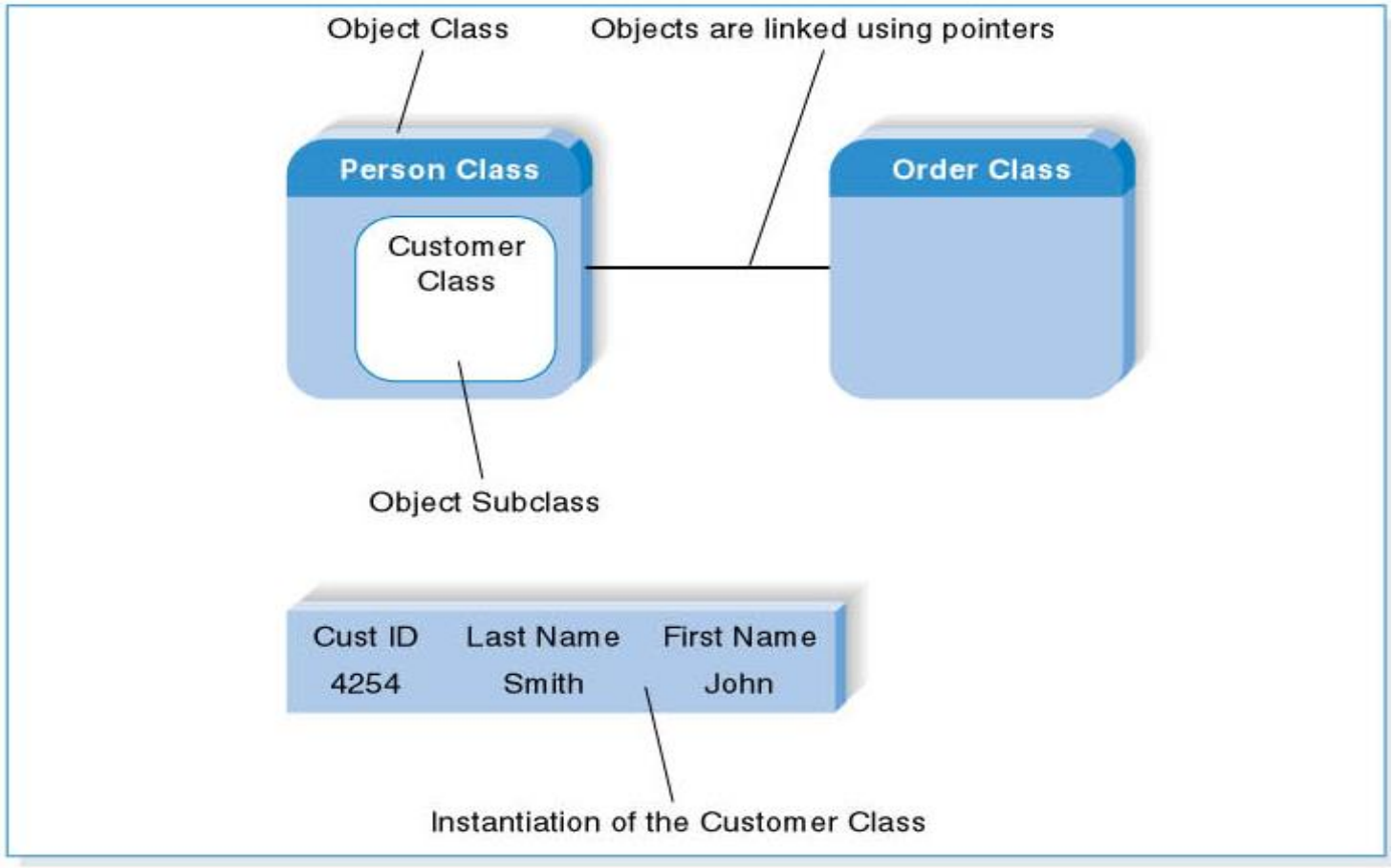
Object Databases

- The ***object database***, or object-oriented database, is based on the premise of object orientation that all things should be treated as ***objects*** that have both data (attributes) and processes (behaviors).
- Changes to one object have no effect on other objects because the attributes and behaviors self-contained, or encapsulated, within each one.
- This ***encapsulation*** allows objects to be reused.

(cont'd)

- In object databases, the combination of data and processes is represented by ***object classes*****.**
- An object class can contain a variety of ***subclasses***.
- An instance of data in object databases is referred to as an ***instantiation***.
- ***Object-oriented database management system (OODBMS)*** are mainly used to support multimedia applications or systems that involve complex data.
- ***Hybrid OODBMS*** technology includes databases with both object and relational features.

Object Database Example (cont'd)



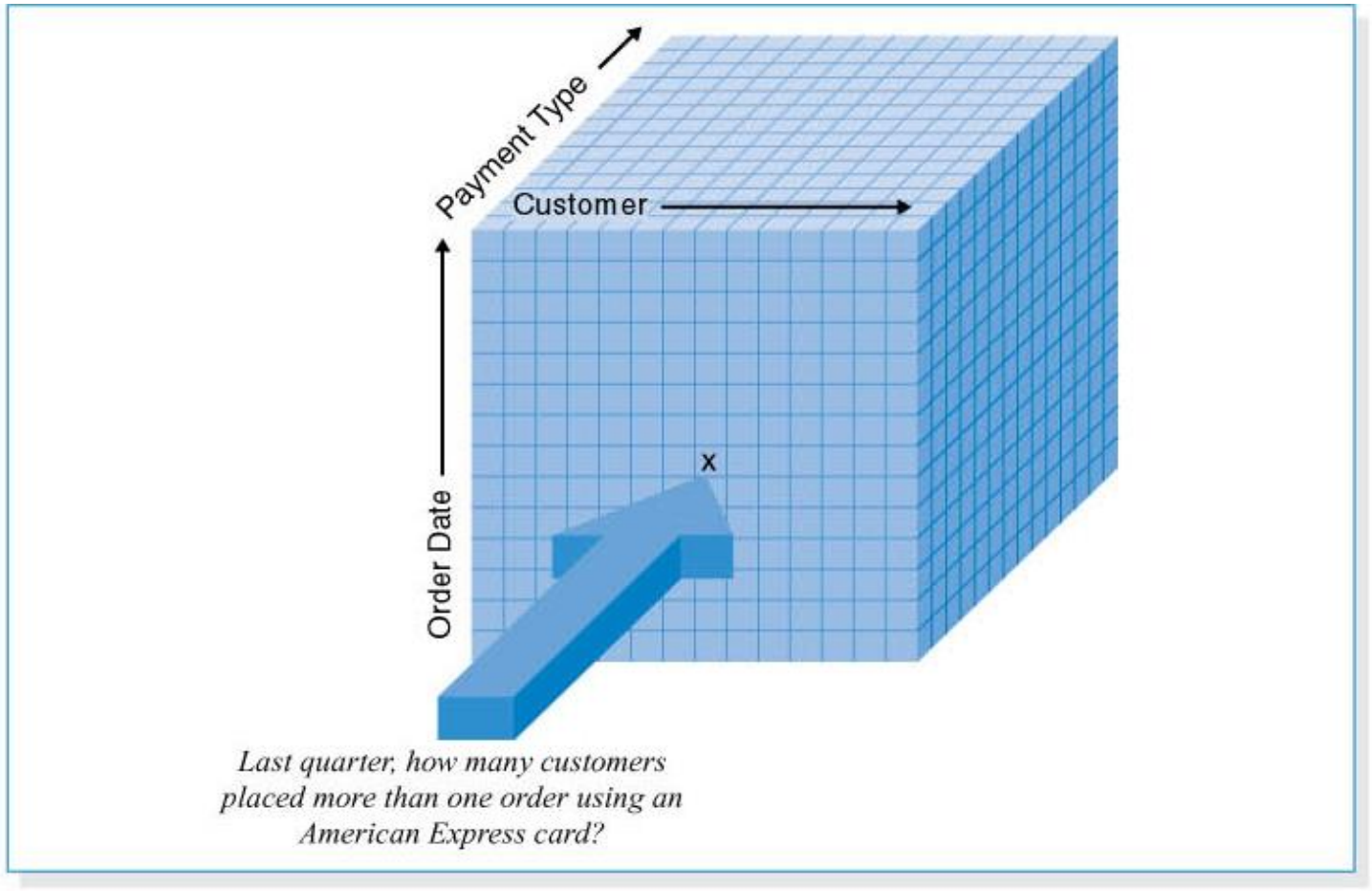
Multidimensional Databases

- A ***multidimensional database*** is a type of relational database that is used extensively in data warehousing.
- ***Data warehousing*** is the practice of taking and storing data in a data warehouse (i.e., a large database) that supports ***decision support systems (DSS)***.
- ***Data marts*** are smaller databases based on data warehouse data, and support DSS for specific departments or functional areas of the organization.

(cont'd)

- A multidimensional database stores data to support ***aggregations*** of data on multiple dimensions.
- When the data are first loaded into a multidimensional database, the database precalculates the data across the multiple dimensions and stores the answers for fast access.

Multidimensional Database Example (cont'd)



Selecting a Storage Format

- Each of the file and database data storage format has its strengths and weaknesses.
- Factors to consider in selecting a storage format:

- Data Types
- Type of Application System
- Existing Storage Formats
- Future Needs

(cont'd)

	Files	Legacy DBMS	Relational DBMS	Object-Oriented DBMS	Multi-dimensional DBMS
Major strengths	Files can be designed for fast performance; good for short-term data storage.	Very mature products	Leader in the database market; can handle diverse data needs	Able to handle complex data	Configured to answer decision support questions quickly
Major weaknesses	Redundant data; data must be updated, using programs.	Not able to store data as efficiently; limited future	Cannot handle complex data	Technology is still maturing; skills are hard to find.	Highly specialized use; skills are hard to find
Data types supported	Simple	*Not recommended for new systems*	Simple	Complex (e.g., video, audio, images)	Aggregated
Types of application systems supported	Transaction processing	*Not recommended for new systems*	Transaction processing and decision making	Transaction processing	Decision making
Existing data formats	Organization dependent	Organization dependent	Organization dependent	Organization dependent	Organization dependent
Future needs	Limited future prospects	Poor future prospects	Good future prospects	Uncertain future prospects	Uncertain future prospects

DBMS = database management system.

MOVING FROM LOGICAL TO PHYSICAL DATA MODELS

- The ***logical entity relationship diagrams (ERDs)*** created during analysis depict the "business view" of the data, but omit implementation details.
- Having determined the data storage format, ***physical data models*** are created to show implementation details and to explain more about the "how" of the final system.

The Physical Entity Relationship Diagram

- The ERD contains the same components for both logical and physical models, including entities, relationships, and attributes.
- The difference lies in the fact that physical ERDs contain references to how data will be stored and that much more metadata are defined.

(cont'd)

- The transition from the logical to physical data model involves five steps :

Step	Explanation
Change entities to tables or files.	Beginning with the logical entity relationship diagram, change the entities to tables or files and update the metadata.
Change attributes to fields.	Convert the attributes to fields and update the metadata.
Add primary keys.	Assign primary keys to all entities.
Add foreign keys.	Add foreign keys to represent the relationships among entities.
Add system-related components.	Add system-related tables and fields.

Example of physical ERD (cont'd)

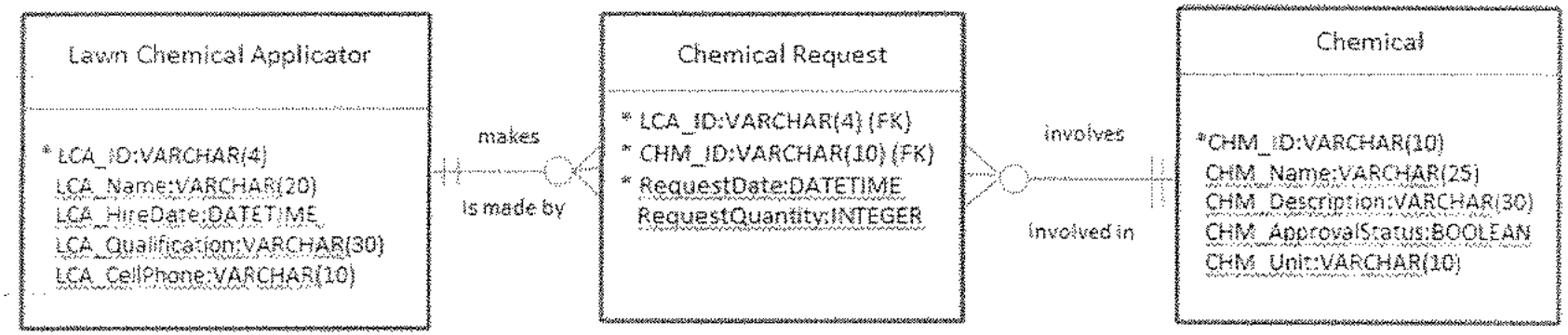
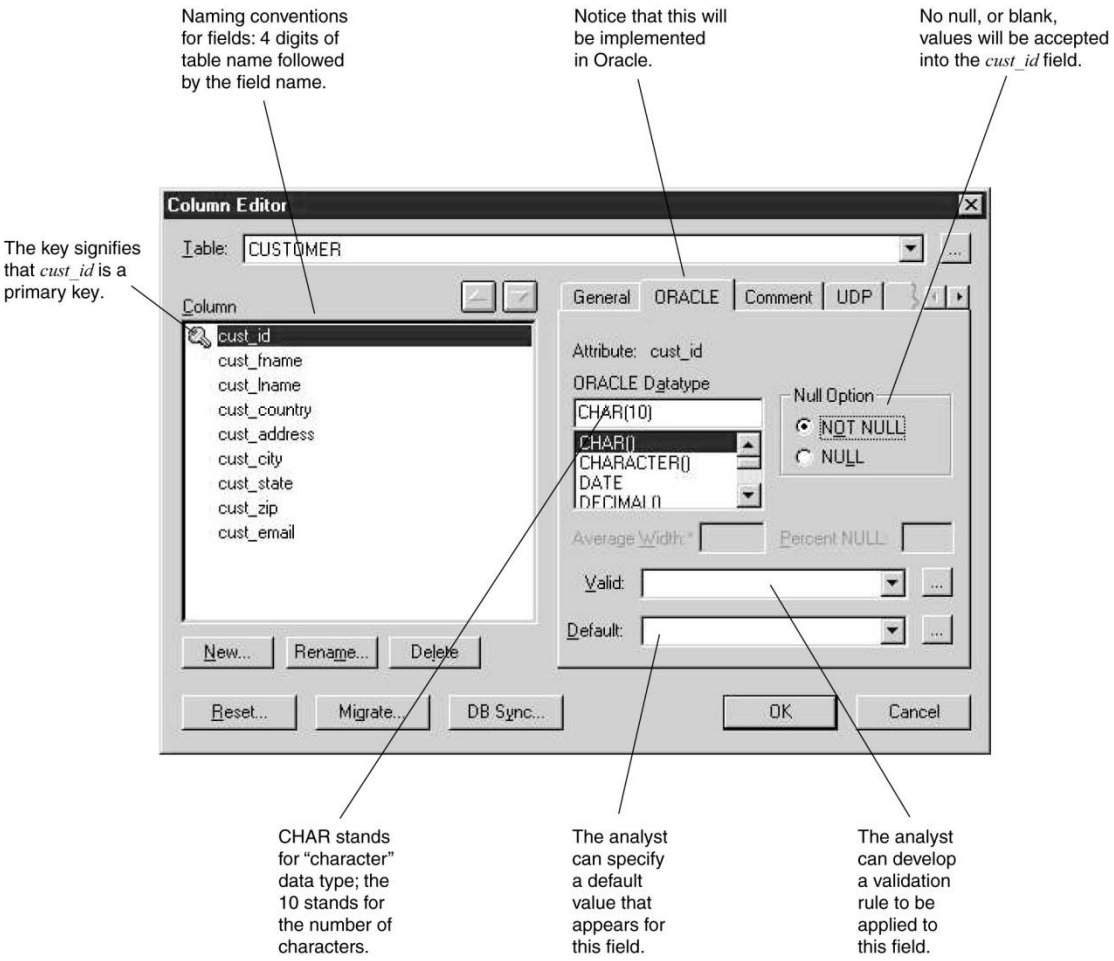


Figure 11-11
Lawn Chemical Request System Physical ERD

Example of metadata (cont'd)



Revising the CRUD Matrix

- It is important to verify that the system's DFD and ERD models are balanced.
- In design, as these models are converted into physical models, changes in the form of new processes, new data stores, and new data elements may occur. The CRUD matrix should be revised.

Example of revised CRUD matrix (cont'd)

	1.1 Load Web Site	1.2 Process Search Requests	1.3 Process Tune Selection
PROMOTIONS			
PRO_code	R		
CUS_number	R		
TUN_ID	R		
PRO_price	R		
PRO_term	R		
CUSFAVS			
CUS_number	R		C
TUN_ID	R		C
FAV_dateadded	R		C
TUNES			
TUN_ID		R	R
TUN_title		R	R
TUN_artist		R	R
TUN_genre		R	R
TUN_length		R	R
TUN_price		R	R
TUN_mp3short		R	R
TUN_mp3full		R	R
CUSINTS			
CUS_number			C
TUN_ID			C
INT_datecreated			C

OPTIMIZING DATA STORAGE

- The data storage format is now optimized for processing efficiency.
- There are two primary dimensions in which to optimize a relational database: for storage efficiency and for speed of access – conflicting goals (trade-offs).

Optimizing Storage Efficiency

- The most efficient tables in a relational database in terms of storage space have no redundant data and very few null values.
- Normalization is the best way to optimize data storage for efficiency.

CUSTOMER ORDER

Order Number
Date
Cust ID
Last Name
First Name
State
Amount
Tax Rate
Product 1
Product Description 1
Product 2
Product Description 2
Product 3
Product Description 3

Redundant data

Null cells

Order Number	Date	Cust ID	Last Name	First Name	State	Amount	Tax Rate	Product	Product Desc	Product	Product Desc	Product	Product Desc
239	11/23/09	1135	Black	John	MD	$50.00	0.05	555	Cheese Tray				
260	11/24/09	1135	Black	John	MD	$40.00	0.05	444	Wine Gift Pack				
273	11/27/09	1135	Black	John	MD	$20.00	0.05	222	Bottle Opener				
241	11/23/09	1123	Williams	Mary	CA	$40.00	0.08	444	Wine Gift Pack				
262	11/24/09	1123	Williams	Mary	CA	$20.00	0.08	222	Bottle Opener				
287	11/27/09	1123	Williams	Mary	CA	$20.00	0.08	222	Bottle Opener				
290	11/30/09	1123	Williams	Mary	CA	$50.00	0.08	555	Cheese Tray				
234	11/23/09	2242	DeBerry	Ann	DC	$50.00	0.065	555	Cheese Tray				
237	11/7/09	2242	DeBerry	Ann	DC	$50.00	0.065	111	Wine Guide	444	Wine Gift Pack		
238	11/10/09	2242	DeBerry	Ann	DC	$40.00	0.065	444	Wine Gift Pack				
245	11/11/09	2242	DeBerry	Ann	DC	$20.00	0.065	222	Bottle Opener				
250	11/18/09	2242	DeBerry	Ann	DC	$20.00	0.065	222	Bottle Opener				
252	11/22/09	2242	DeBerry	Ann	DC	$60.00	0.065	222	Bottle Opener	444	Wine Gift Pack		
253	11/23/09	2242	DeBerry	Ann	DC	$60.00	0.065	222	Bottle Opener	444	Wine Gift Pack		
297	11/24/09	2242	DeBerry	Ann	DC	$30.00	0.065	333	Jams & Jellies				
243	11/11/09	4254	Bailey	Ryan	MD	$50.00	0.05	555	Cheese Tray				
246	11/18/09	4254	Bailey	Ryan	MD	$30.00	0.05	333	Jams & Jellies				
248	11/22/09	4254	Bailey	Ryan	MD	$60.00	0.05	222	Bottle Opener	333	Jams & Jellies	111	Wine Guide
235	11/17/09	9500	Chin	April	KS	$20.00	0.05	222	Bottle Opener				
242	11/23/09	9500	Chin	April	KS	$30.00	0.05	333	Jams & Jellies				
244	11/24/09	9500	Chin	April	KS	$20.00	0.05	222	Bottle Opener				
251	11/27/09	9500	Chin	April	KS	$10.00	0.05	111	Wine Guide				

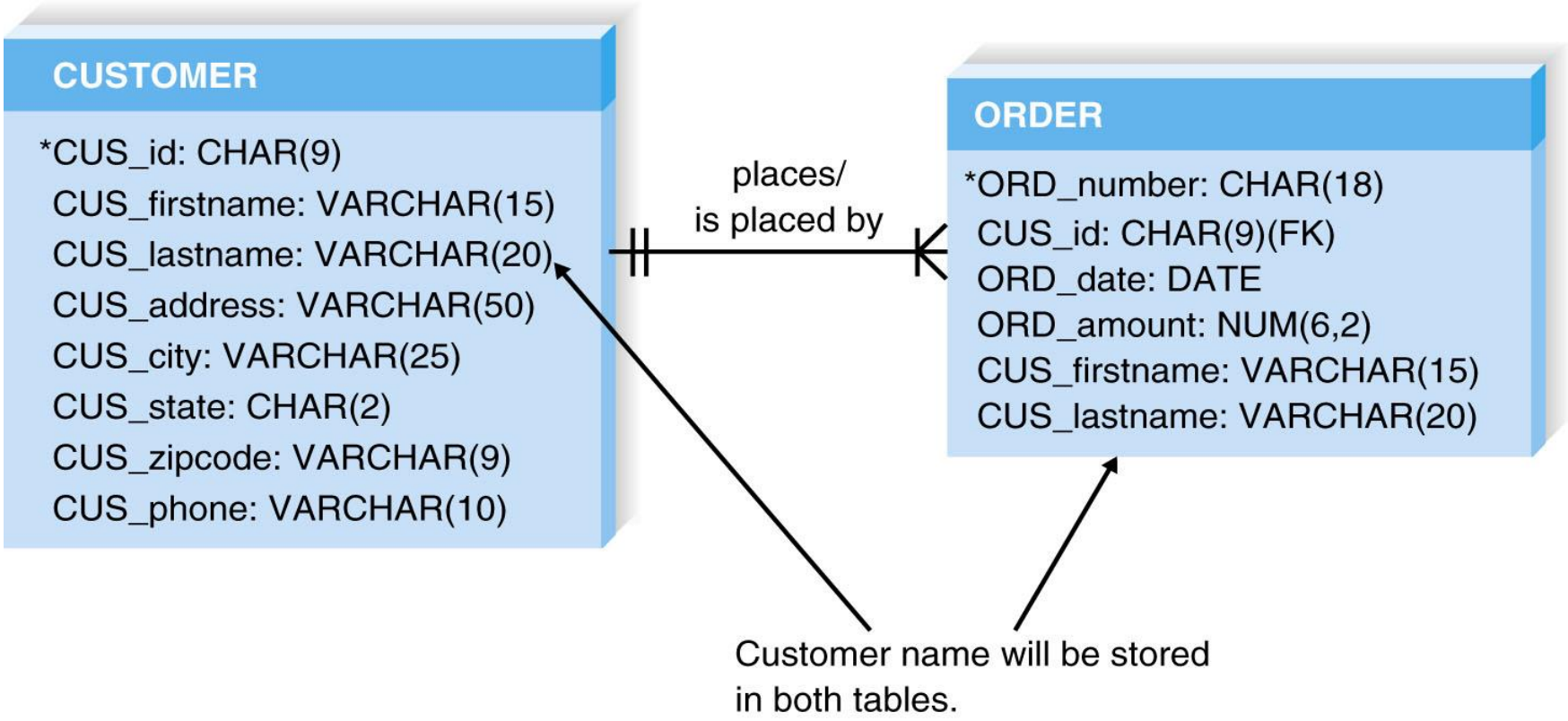
FIGURE 11-16
Optimizing Data Storage

Optimizing Access Speed

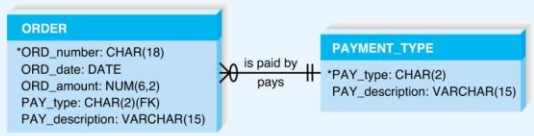
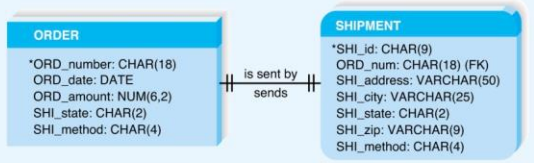
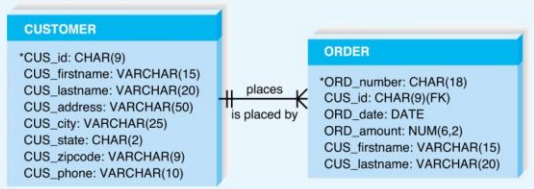
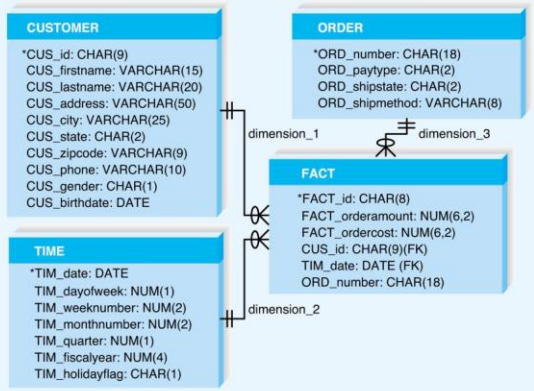
- After having optimized the data model design for data storage efficiency, the end result is that data are spread out across a number of tables.
- For a large relational database, it is necessary to optimize access speed.
- There are several techniques of optimizing access speed:
  - Denormalization
  - Clustering
  - Indexing
  - Estimating the size of data for hardware planning

Denormalization

- ***Denormalization*** – adding redundancy back into the design.



There are four reasons for denormalization. (cont'd)

Reason	Description	Example
Look-up Table	Include a code's description in the table using that code if the description is often used.	
1:1 Relationships	Combine tables if they are related 1:1 and if they usually are accessed together.	
1:N Relationships	Place fields from the parent (1) table into the child (N) table if the parent fields are used frequently with child information.	
Star Schema Design	Data marts often are modeled with star schema design, which uses denormalization to maximaze DSS query performance.	

Clustering

- ***Clustering*** – placing records together physically so that like records are stored close together.
- ***Intrafile clustering*** – Similar records in the table are stored together.
- ***Interfile clustering*** – Combining records from more that one table that typically are retrieved together.

Indexing

- An ***index*** in data storage is a minitable (similar to an index of a book) that contains values from one or more columns in a table and the location of the values within the table.
- Indexes require overhead in that they take up space on the storage.

Example of indexing (cont'd)

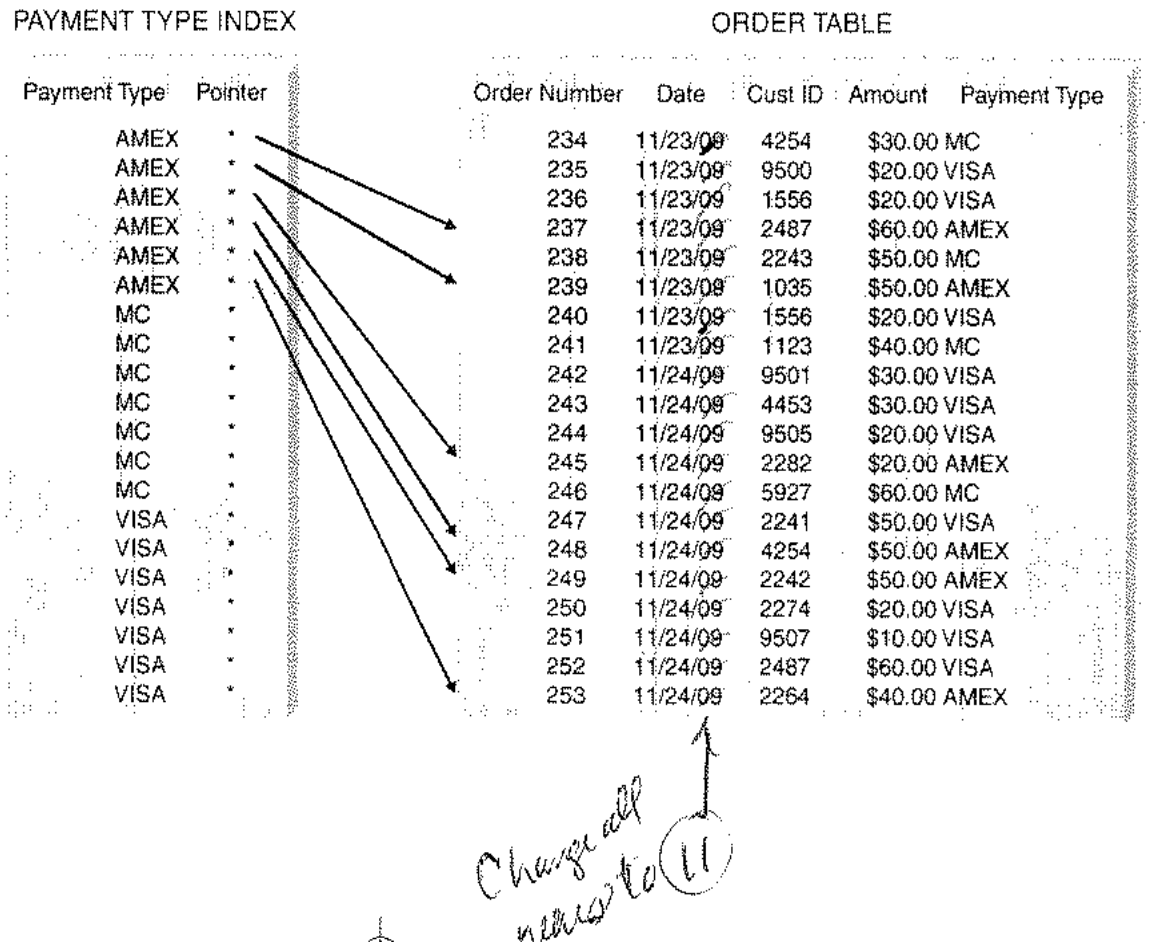


FIGURE 11-19
Payment Type Index

Guidelines for creating indexes (cont'd)

- Use indexes sparingly for transaction systems.
- Use many indexes to improve response times in decision support systems.
- For each table, create a unique index that is based on the primary key.
- For each table, create an index that is based on the foreign key to improve the performance of joins.
- Create an index for fields that are used frequently for grouping, sorting, or criteria.

Estimating Storage Size

- ***Volumetrics*** – technique of estimating the amount of data that the hardware will need to support.

1. Calculate the amount of ***raw data*** - all the data that are stored within the tables of the database.
2. Calculate the ***overhead*** requirements based on the DBMS vendor's recommendations.
3. Record the number of initial records that will be loaded into the table, as well as the expected growth per month.

Example of calculating volumetrics (cont'd)

Field	Average Size (Characters)
Order number	8
Date	7
Cust ID	4
Last name	13
First name	9
State	2
Amount	4
Tax rate	2
Record size	49
Overhead	30%
Total record size	63.7
Initial table size	50,000
Initial table volume	3,185,000
Growth rate/month	1000
Table volume @ 3 years	5,478,200

SUMMARY

- **File data storage formats**
  - Files are electronic lists of data.
  - Five types of files: master, look-up, transaction, audit, and history.
- **Database storage formats**
  - A database is a collection of groupings of information
  - A DBMS is software that creates and manipulates these databases.
- **Selecting a data storage format**
  - Relational databases support simple data types very effectively, whereas object databases are best for complex data.

(cont'd)

- **Physical entity relationship diagrams**
  - Physical ERDs contain references to how data will be stored in a file or database table, and metadata are included.
- **Optimizing data storage**
  - There are two primary dimensions in which to optimize a relational database: for storage efficiency and for speed of access.
  - There are a number of techniques of optimizing data storage.

Copyright 2011 John Wiley & Sons, Inc.