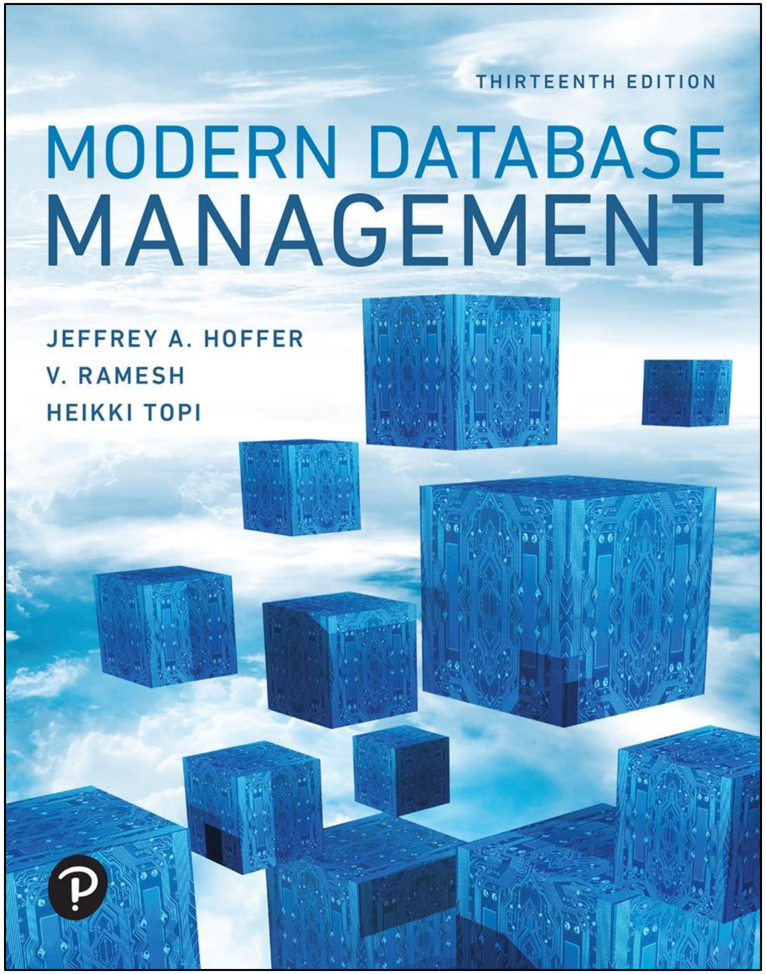
Modern Database Management

Thirteenth Edition



Chapter 8

Physical Database Design
and Database Infrastructure

Learning Objectives (1 of 2)

8.1 Define terms

8.2 Describe the physical database design process

8.3 Choose storage formats for attributes

8.4 Select appropriate file organizations

8.5 Describe three types of file organization

8.6 Describe indexes and their appropriate use

8.7 Translate a database model into efficient structures, including denormalization

Learning Objectives (2 of 2)

8.8 Describe problems and techniques for data security

8.9 Understand role of databases in Sarbanes-Oxley compliance

8.10 Describe problems and facilities for data recovery

8.11 Describe database tuning issues and list areas where changes can be done to tune the database

8.12 Understand impact of cloud-based database services on database infrastructure

8.13 Describe advantages and disadvantages of cloud-based database infrastructure solutions

Physical Database Design

- Purpose – translate the logical description of data into the **technical specifications** for storing and retrieving data
- Goal – create a design for storing data that will provide **adequate performance** and ensure **database integrity**, **security**, and **recoverability**

Information Needed for Physical Design

- Normalized relations, including estimates for the range of the number of rows in each table
- Definitions of each attribute, along with physical specifications such as maximum possible length
- Descriptions of where and when data are used in various ways (entered, retrieved, deleted, and updated), including typical frequencies of these events
- Expectations or requirements for response time and data security, backup, recovery, retention, and integrity
- Descriptions of the technologies (database management systems) used for implementing the database

Critical Decisions for Physical Design

- Choosing the storage format (called data type) for each attribute from the logical data model
- Giving the DBMS guidance regarding how to group attributes from the logical data model into physical records
- Giving the DBMS guidance on how to arrange similarly structured records in secondary memory (file organization)
- Selecting structures (including indexes and the overall database architecture) for storing and connecting files to make retrieving related data more efficient
- Preparing strategies for handling queries against the database that will optimize performance (query optimization)

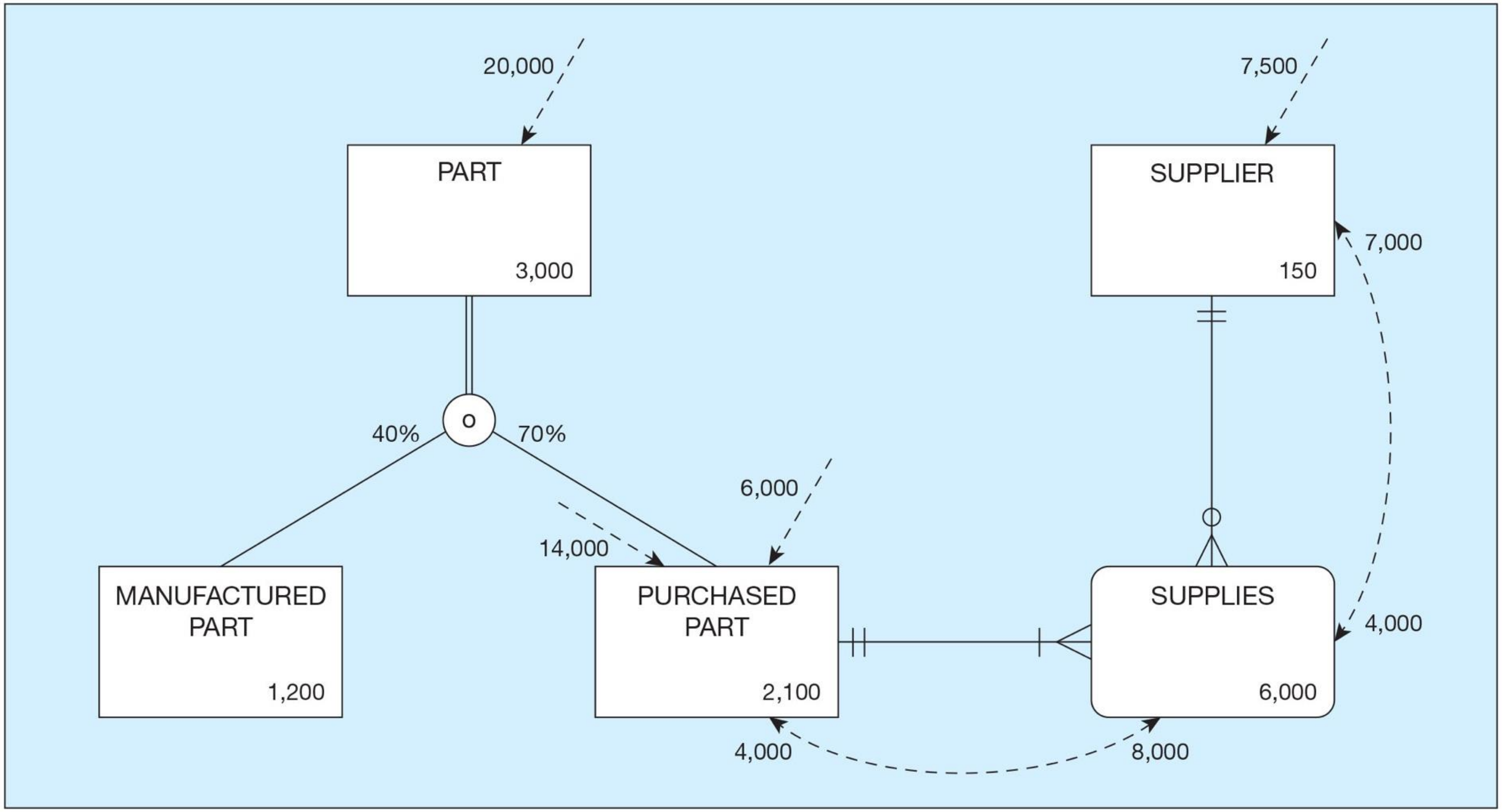
Physical Design for Regulatory Compliance

- Sarbanes- Oxley Act (SOX) – protect investors by improving accuracy and reliability
- Committee of Sponsoring Organizations (COSO) of the Treadway Commission
- IT Infrastructure Library (ITIL)
- Control Objectives for Information and Related Technology (COBIT)

Three Areas of SOX Audits

- IT change management
  - Processes by which changes to operational systems and databases are authorized
- Logical access to data
  - Security procedures to prevent unauthorized access
  - Personnel controls and physical access controls
- IT operations
  - Policies and procedures for day-to-day management of infrastructure, applications, and databases

Figure 8-1 Composite Usage Map (Pine Valley Furniture Company)



Designing Fields

- Field: smallest unit of application data recognized by system software
- Field design
  - Choosing data type
  - Coding, compression, encryption
  - Controlling data integrity

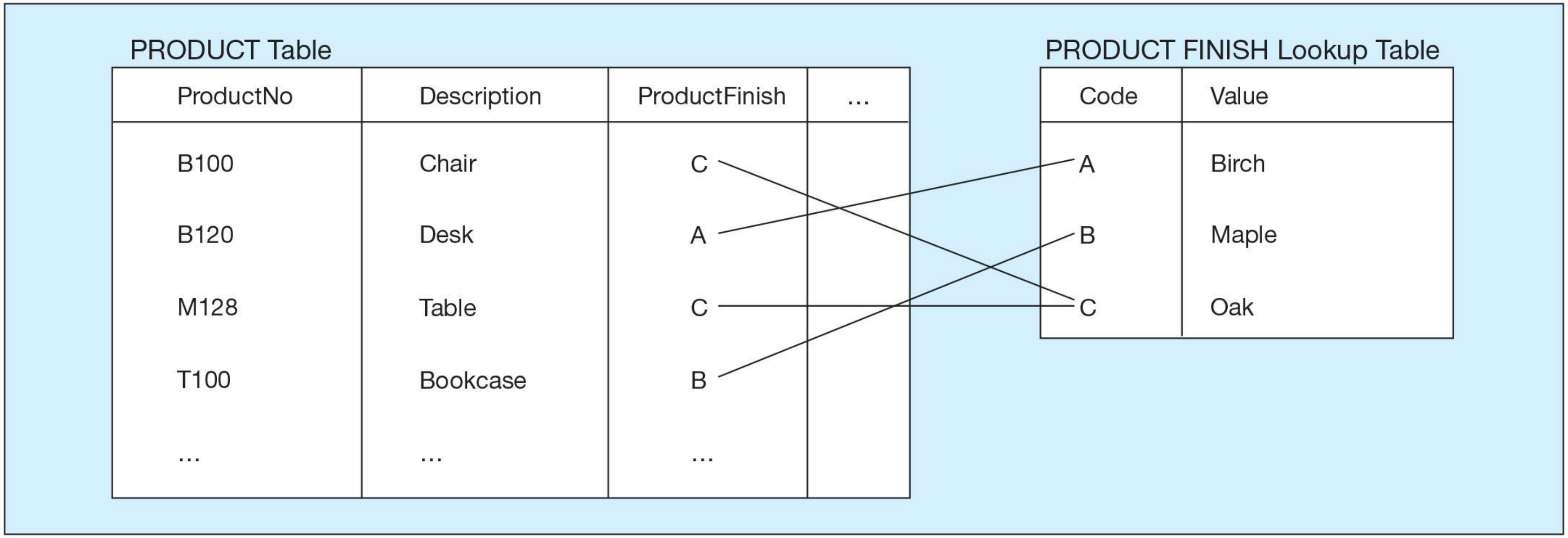
Common Data Types (1 of 2)

- VARCHAR2(length) max 400 characters
  - Variable-length character data. A string that is shorter than the maximum length will consume only the required space. NVARCHAR2 is Unicode.
- CHAR(length) max 200 characters
  - Fixed length character data. NCHAR is Unicode.
- CLOB
  - Character large object, capable of storing up to 4 gigabytes of one variable length character data field
- NUMBER
  - Positive or negative number. NUMBER(5) means a 5 digit integer. NUMBER(5,2): 5 digits, two to the right of the decimal point.

Common Data Types (2 of 2)

- DATE
  - Can represent from Jan 1 4712 BC to Dec 31 9999 AD
  - Stores century, year, month, day, hour, minute, second
- TIMESTAMP
  - Like a date. Can include fractional seconds, and time zones.
- BLOB
  - Binary large object, can store up to 4 gigabytes
  - Used for photos, sound clips, etc.

Figure 8-2 Example of a Code Lookup Table (Pine Valley Furniture Company)



PRODUCT Table

ProductNo	Description	ProductFinish	…
B100	Chair	C	
B120	Desk	A	
M128	Table	C	
T100	Bookcase	B	
…	…	…	

PRODUCT FINISH Lookup Table

Code	Value
A	Birch
B	Maple
C	Oak

Controlling Data Integrity

- Default value – assumed value if no explicit value
- Range control – allowable value limitations (constraints or validation rules)
- Null value control – allowing or prohibiting empty fields
- Referential integrity – range control (and null value allowances) for foreign-key to primary-key match-ups

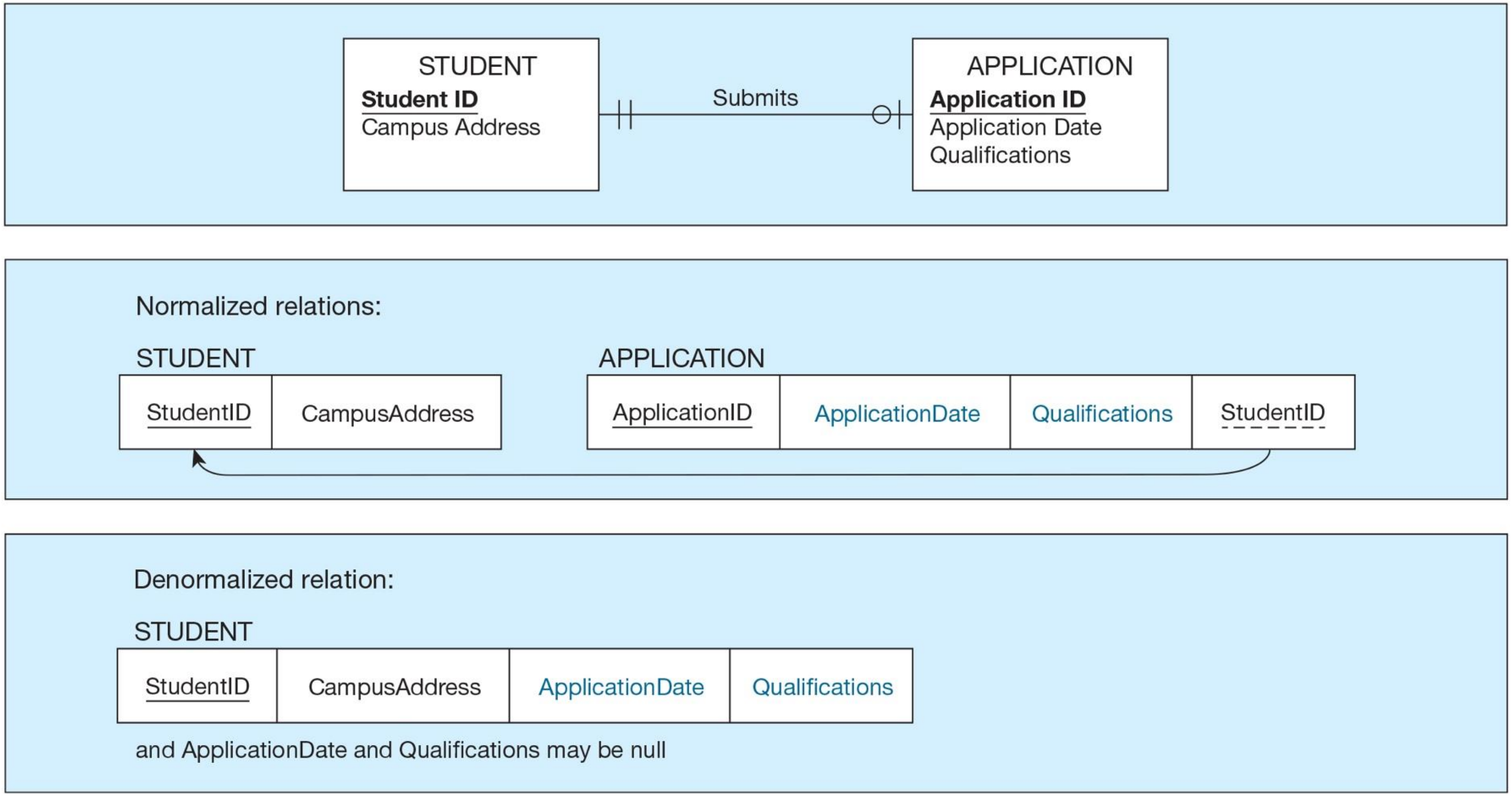
Handling Missing Data

- Substitute an estimate of the missing value (e.g., using a formula)
- Construct a report listing missing values
- In programs, ignore missing data unless the value is significant (sensitivity testing)

Denormalization

- Transforming normalized relations into non-normalized physical record specifications
- Benefits:
  - Can improve performance (speed) by reducing number of table lookups (i.e. reduce number of necessary join queries)
- Costs (due to data duplication):
  - Wasted storage space
  - Data integrity/consistency threats

Figure 8-3 A Possible Denormalization Situation: Two Entities With One-to-One Relationship



Normalized relations:

STUDENT

StudentID	CampusAddress

APPLICATION

ApplicationID	ApplicationDate	Qualifications	StudentID

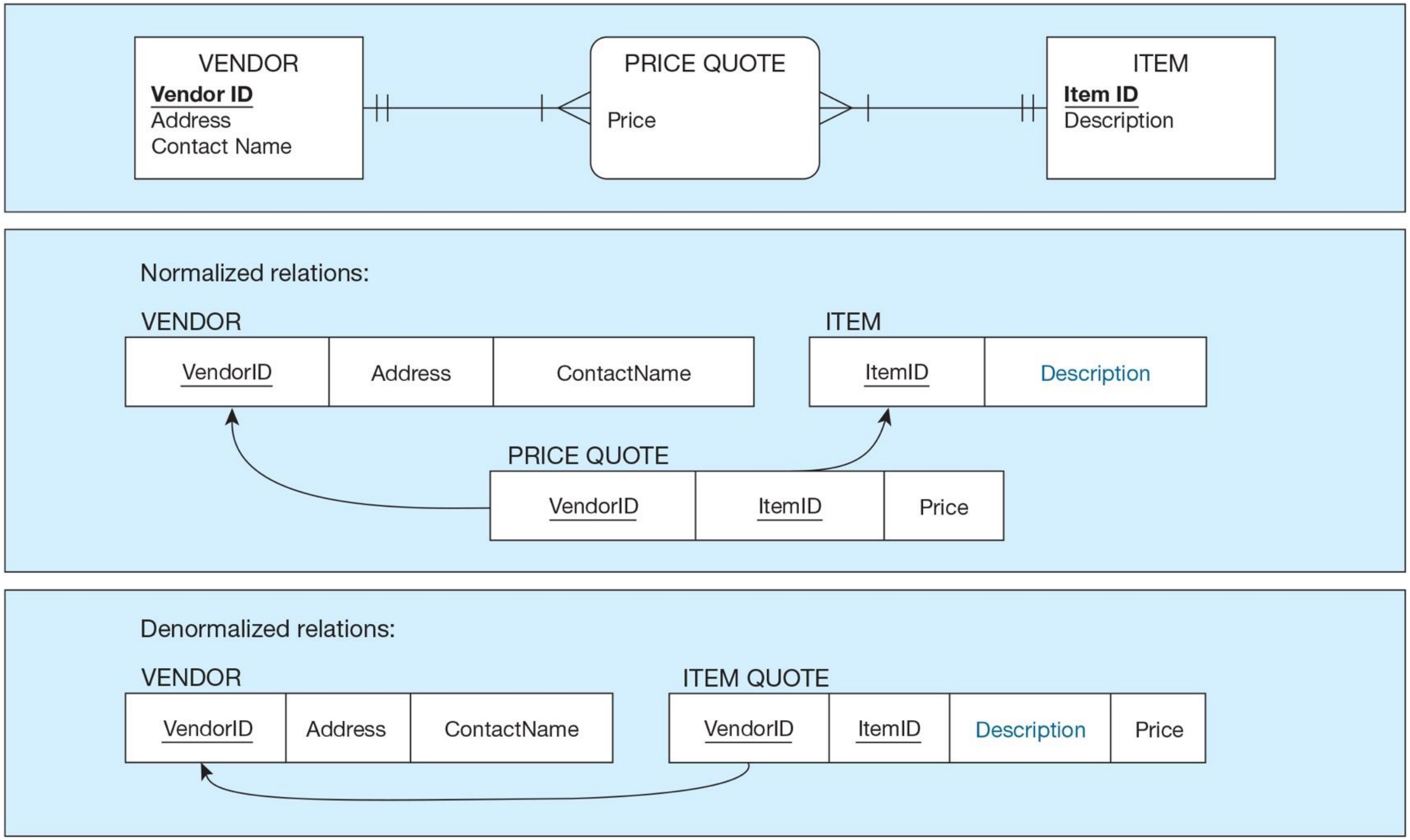
Denormalized relation:

STUDENT

StudentID	CampusAddress	ApplicationDate	Qualifications

and ApplicationDate and Qualifications may be null

Figure 8-4 A Possible Denormalization Situation: A Many-to-Many Relationship With Non-Key Attributes



VENDOR
Vendor ID
Address
Contact Name

PRICE QUOTE
Price

ITEM
Item ID
Description

Normalized relations:

VENDOR

VendorID	Address	ContactName

ITEM

ItemID	Description

PRICE QUOTE

VendorID	ItemID	Price

Denormalized relations:

VENDOR

VendorID	Address	ContactName

ITEM QUOTE

VendorID	ItemID	Description	Price

Denormalize With Caution

- Denormalization can
  - Increase chance of errors and inconsistencies
  - Reintroduce anomalies
  - Force reprogramming when business rules change
- Perhaps other methods could be used to improve performance of joins
  - Organization of tables in the database (file organization and clustering)
  - Proper query design and optimization

Partitioning

- Horizontal Partitioning: Distributing the rows of a logical relation into several separate tables
  - Useful for situations where different users need access to different rows
  - Three types: Key Range Partitioning, Hash Partitioning, or Composite Partitioning
- Vertical Partitioning: Distributing the columns of a logical relation into several separate physical tables
  - Useful for situations where different users need access to different columns
  - The primary key must be repeated in each file

Partitioning Pros and Cons

- Advantages of Partitioning
  - Efficiency: records used together are grouped together
  - Local optimization: each partition can be optimized for performance
  - Security: data not relevant to users are segregated
  - Recovery and uptime: smaller files take less back up time
  - Load balancing: partitions stored on different disks, reduces contention
- Disadvantages of Partitioning
  - Inconsistent access speed: slow retrievals across partitions
  - Complexity: non-transparent partitioning
  - Extra space or update time: duplicate data; access from multiple partitions

Designing Physical Database Files

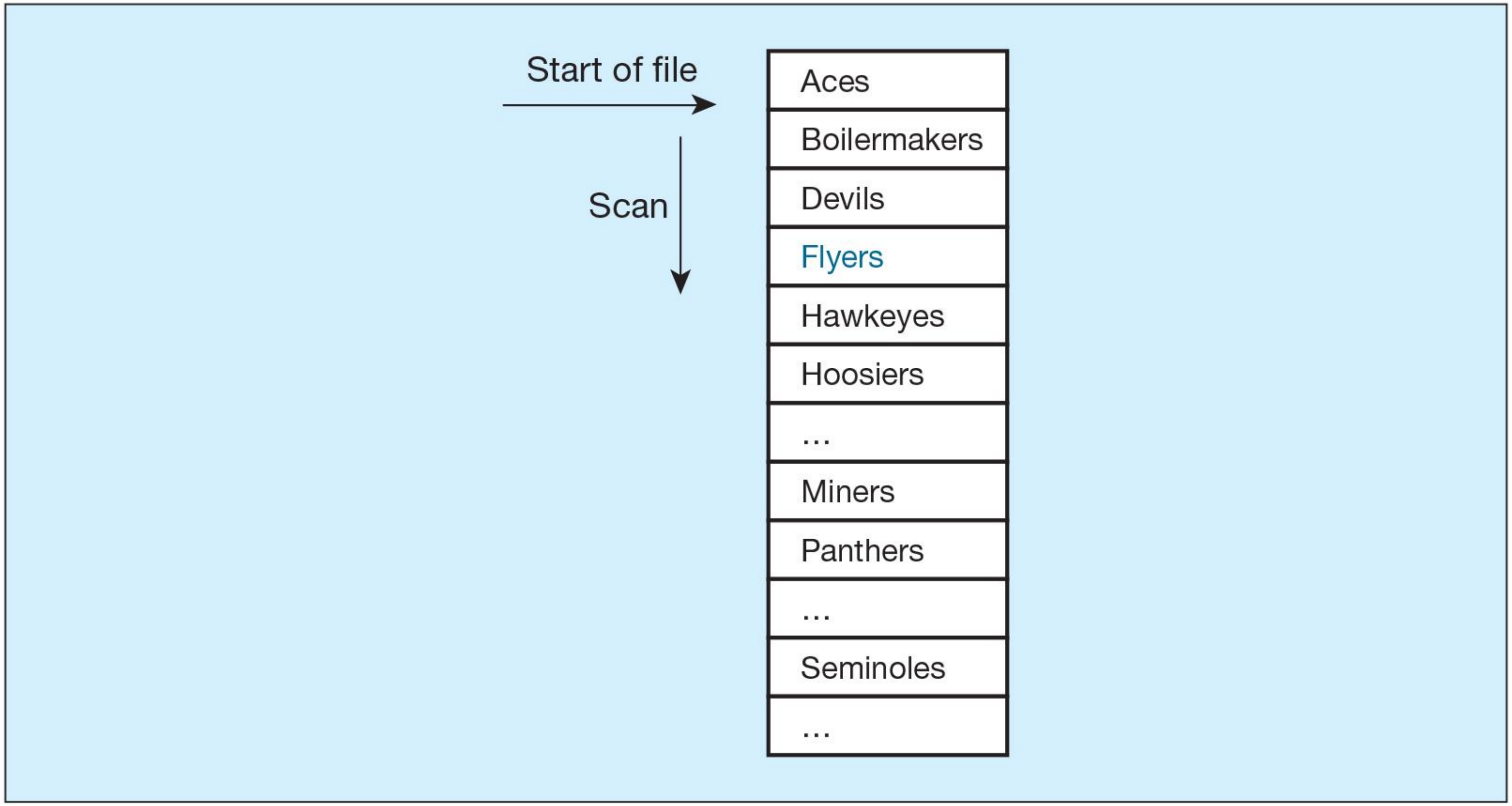
- Physical File
  - A named portion of secondary memory allocated for the purpose of storing physical records
  - Tablespace – named logical storage unit in which data from multiple tables/views/objects can be stored
- Tablespace components
  - Segment – a table, index, or partition
  - Extent – contiguous section of disk space
  - Data block – smallest unit of storage

File Organizations

- Types of file organizations
  - Heap – no particular order
  - Sequential
  - Indexed
  - Hashed
- Factors for selecting file organization
  - Fast data retrieval and throughput
  - Efficient storage space utilization
  - Protection from failure and data loss
  - Minimizing need for reorganization
  - Accommodating growth
  - Security from unauthorized use

Figure 8-7 Comparison of File Organizations (1 of 3)

(a) Sequential



Indexed File Organizations

- Storage of records sequentially or nonsequentially with an index that allows software to locate individual records
- Index: a table or other data structure used to determine in a file the location of records that satisfy some condition
- Primary keys are automatically indexed
- Other fields or combinations of fields can also be indexed; these are called secondary keys (or nonunique keys)

Figure 8-7 Comparison of File Organizations (2 of 3)

(b) Indexed

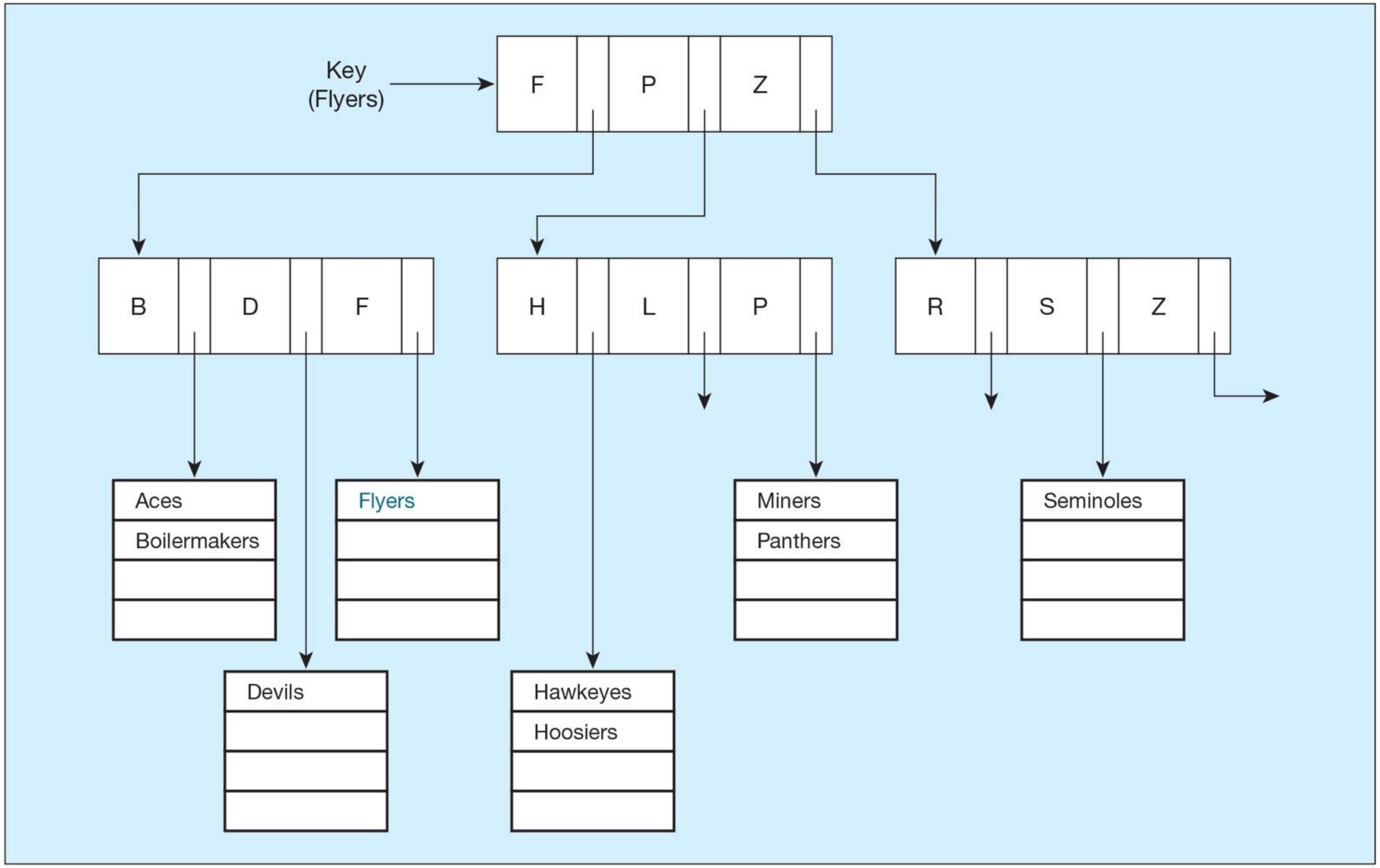
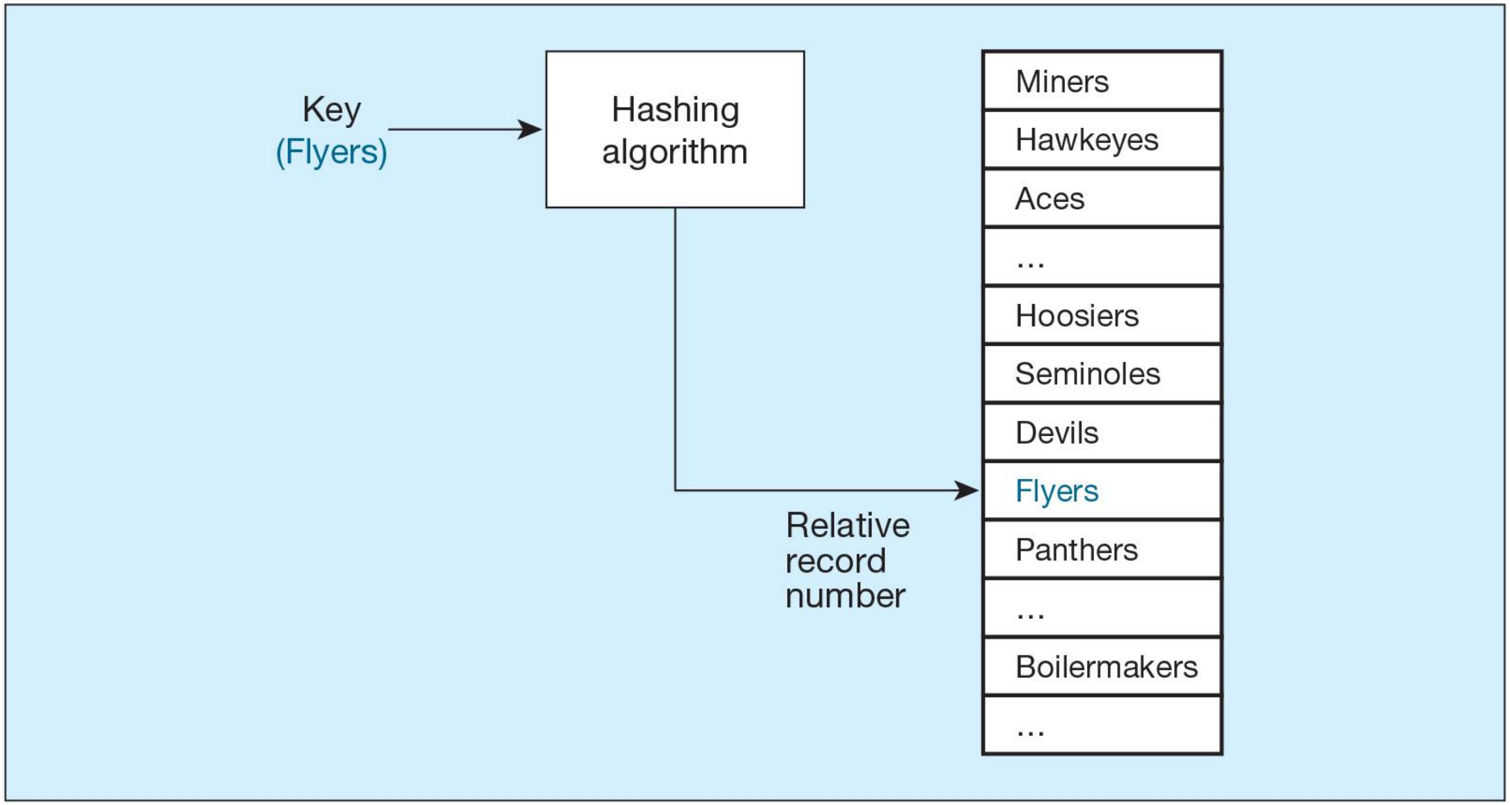


Figure 8-7 Comparison of File Organizations (3 of 3)

(c) Hashed



Clustering Files

- In some relational DBMSs, related records from different tables can be stored together in the same disk area
- Useful for improving performance of join operations
- Primary key records of the main table are stored adjacent to associated foreign key records of the dependent table
- e.g. Oracle has a CREATE CLUSTER command

Unique and Nonunique Indexes

- Unique (primary) Index
  - Typically done for primary keys, but could also apply to other unique fields
  - CREATE UNIQUE INDEX CustIndex_PK ON Customer_T(CustomerID);
- Non unique (secondary) index
  - Done for fields that are often used to group individual entities (e.g., zip code, product category)
  - CREATE INDEX DescIndex_FK ON Product_T(Description);

When to Use Indexes (1 of 2)

1. Use on larger tables
2. Index the primary key of each table
3. Index search fields (fields frequently in WHERE clause)
4. Fields in SQL ORDER BY and GROUP BY commands
5. When there are >100 values but not when there are <30 values

When to Use Indexes (2 of 2)

6. Avoid use of indexes for fields with long values; perhaps compress values first
7. If key to index is used to determine location of record, use surrogate (like sequence number) to allow even spread in storage area
8. DBMS may have limit on number of indexes per table and number of bytes per indexed field(s)
9. Be careful of indexing attributes with null values; many DBMSs will not recognize null values in an index search
10. Use a query optimizer

Query Optimization

- Parallel query processing – possible when working in multiprocessor systems
- Overriding automatic query optimization allows for query writers to preempt the automated optimization
- Oracle example:

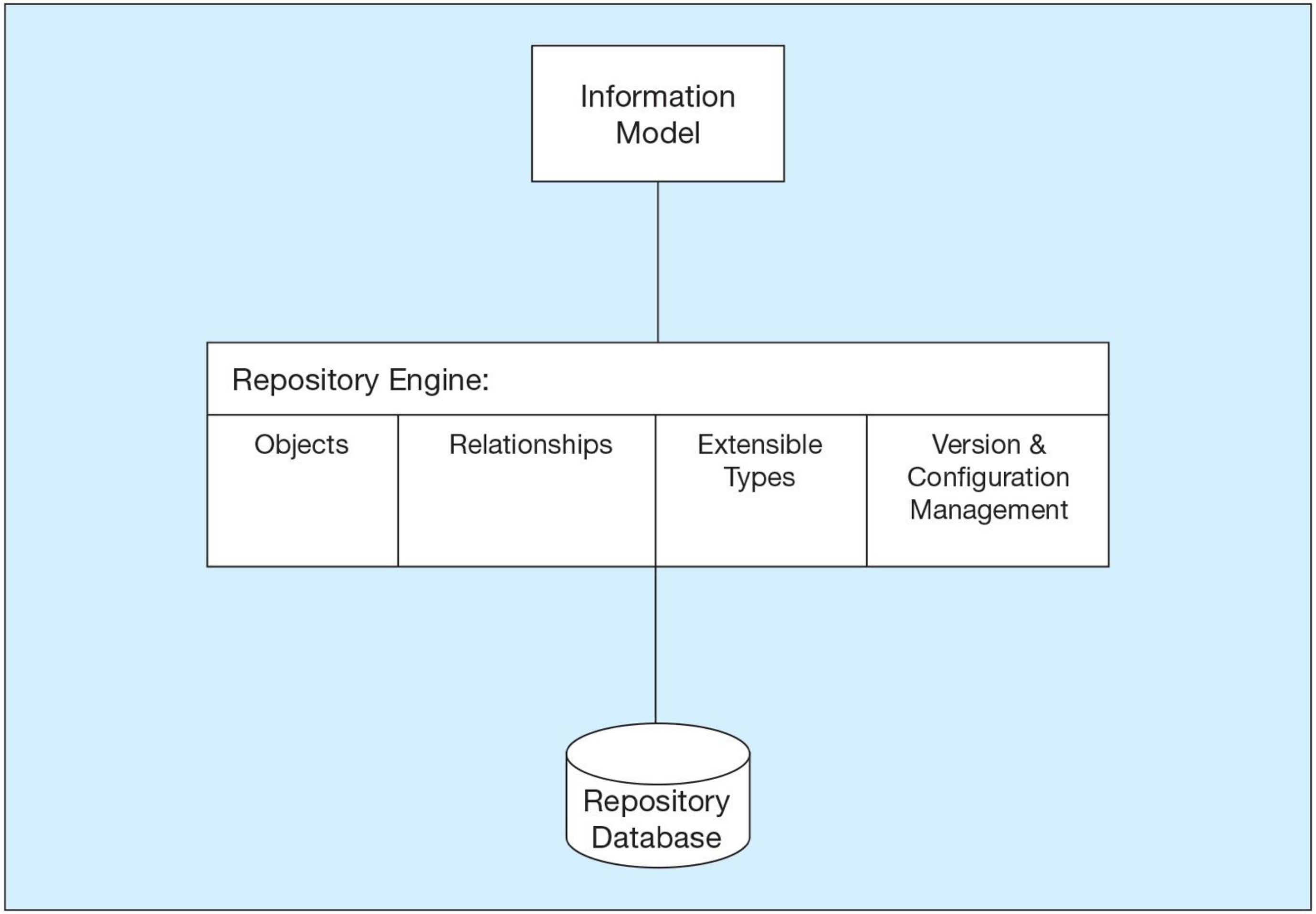
```
SELECT /*+ FULL(Order_T) PARALLEL(Order_T,3) */
COUNT(*)
FROM Order_T
WHERE Salesperson = “Smith”;
```

- /* */ clause is a hint to override Oracle’s default query plan

Data Dictionaries and Repositories

- Data dictionary
  - Documents data elements of a database
- System catalog
  - System-created database that describes all database objects
- Information Repository
  - Stores metadata describing data and data processing resources

Figure 8-8 Three Components of the Repository System Architecture (Based on Bernstein, 1996)



Database Software Security Features

- Views or subschemas
- Integrity controls
- Authorization rules
- User-defined procedures
- Encryption
- Authentication schemes
- Backup, journalizing, and checkpointing

Views and Integrity Controls

- Views
  - Subset of the database that is presented to one or more users
  - User can be given access privilege to view without allowing access privilege to underlying tables
- Integrity Controls
  - Protect data from unauthorized use
  - Domains – set allowable values
  - Assertions – enforce database conditions
  - Triggers – prevent inappropriate actions, invoke special handling procedures, write to log files

Authorization Rules

- Controls incorporated in the data management system
- Restrict:
  - access to data
  - actions that people can take on data
- Authorization matrix for:
  - Subjects
  - Objects
  - Actions
  - Constraints

Subject	**Object**	**Action**	**Constraint**
Sales Dept.	Customer record	Insert	Credit limit LE $5000
Order trans.	Customer record	Read	None
Terminal 12	Customer record	Modify	Balance due only
Acctg. Dept.	Order record	Delete	None
Ann Walker	Order record	Insert	Order aml LT $2000
Program AR4	Order record	Modify	None

Figure 8-10 Implementing Authorization Rules

(a) Authorization table for subjects (salespersons)

Blank	Customer records	Order records
Read	Y	Y
Insert	Y	Y
Modify	Y	N
Delete	N	N

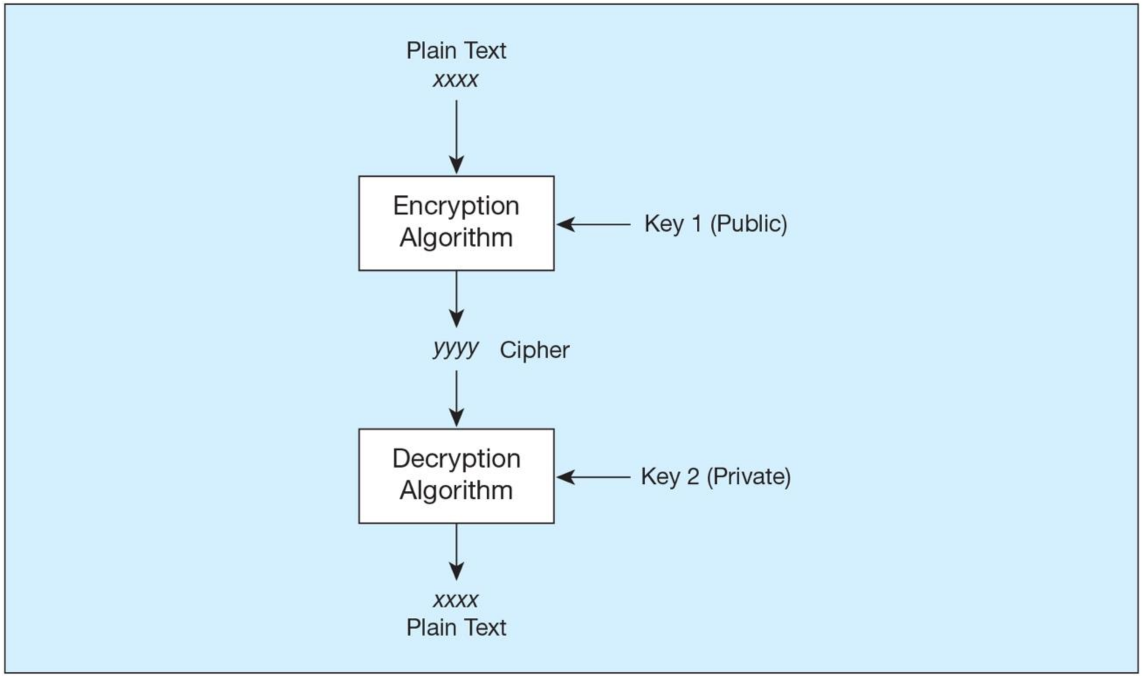
(b) Authorization table for objects (orders)

Blank	Salespersons (password BATMAN)	Order entry (password JOKER)	Accounting (password TRACY)
Read	Y	Y	Y
Insert	N	Y	N
Modify	N	Y	Y
Delete	N	N	Y

Figure 8-12 Basic Two-Key Encryption

Encryption – the coding or scrambling of data so that humans cannot read them

Secure Sockets Layer (SSL) is a popular encryption scheme for TCP/IP connections.



Authentication Schemes (1 of 2)

- Goal – obtain a **positive** identification of the user
- Passwords: First line of defense
  - Should be at least 8 characters long
  - Should combine alphabetic and numeric data
  - Should not be complete words or personal information
  - Should be changed frequently

Authentication Schemes (2 of 2)

- Strong Authentication
  - Passwords are flawed:
    - Users share them with each other
    - They get written down, could be copied
    - Automatic logon scripts remove need to explicitly type them in
    - Unencrypted passwords travel the Internet
- Possible solutions:
  - Two factor – e.g., smart card plus PIN
  - Three factor – e.g., smart card, biometric, PIN

Database Recovery

- Mechanism for restoring a database quickly and accurately after loss or damage
- Recovery facilities:
  - Backup Facilities
  - Journalizing Facilities
  - Checkpoint Facility
  - Recovery Manager

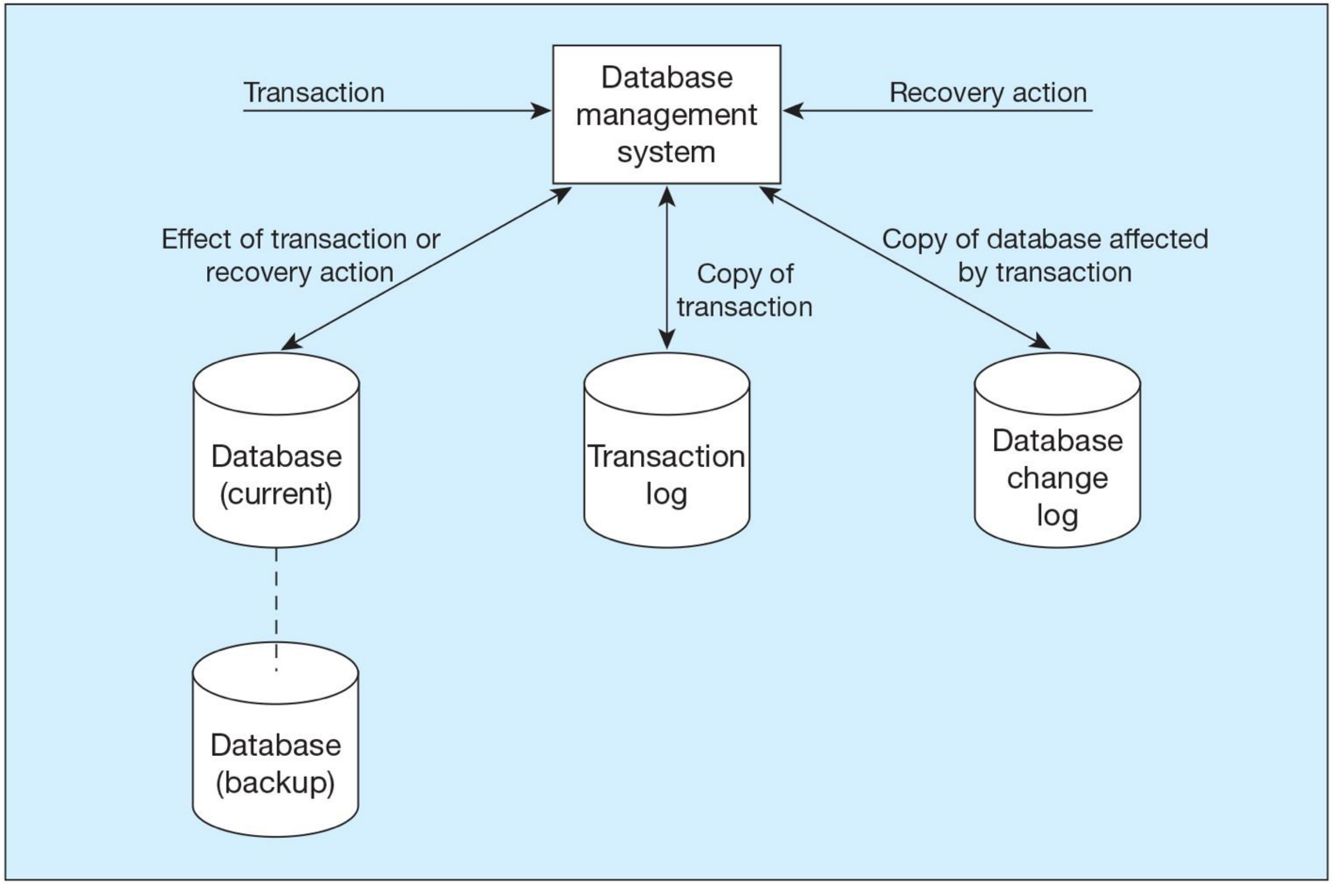
Back-up Facilities

- DBMS copy utility that produces backup copy of the entire database or subset
- Periodic backup (e.g. nightly, weekly)
- Cold backup – database is shut down during backup
- Hot backup – selected portion is shut down and backed up at a given time
- Backups stored in secure, off-site location

Journalizing Facilities

- Audit trail of transactions and database updates
- Transaction log – record of essential data for each transaction processed against the database
- Database change log – images of updated data
- Before-image – copy before modification
- After-image – copy after modification

Figure 8-13 Database Audit Trail



From the backup and logs, databases can be restored in case of damage or loss

Checkpoint Facilities

- DBMS periodically refuses to accept new transactions
- Therefore, the system is in a **quiet** state
- Database and transaction logs are synchronized
- This allows recovery manager to resume processing from short period, instead of repeating entire day

Recovery Manager

- Recovery Manager – DBMS module that restores the database to a correct condition when a failure occurs and then resumes processing user requests
- Recovery and Restart Procedures
  - Disk Mirroring – switch between identical copies of databases
  - Restore/Rerun – reprocess transactions against the backup (only done as a last resort)
  - Backward Recovery (Rollback) – apply before images
  - Forward Recovery (Roll Forward) – apply after images (preferable to restore/rerun)

Figure 8-14 Basic Recovery Techniques (1 of 2)

(a) Rollback

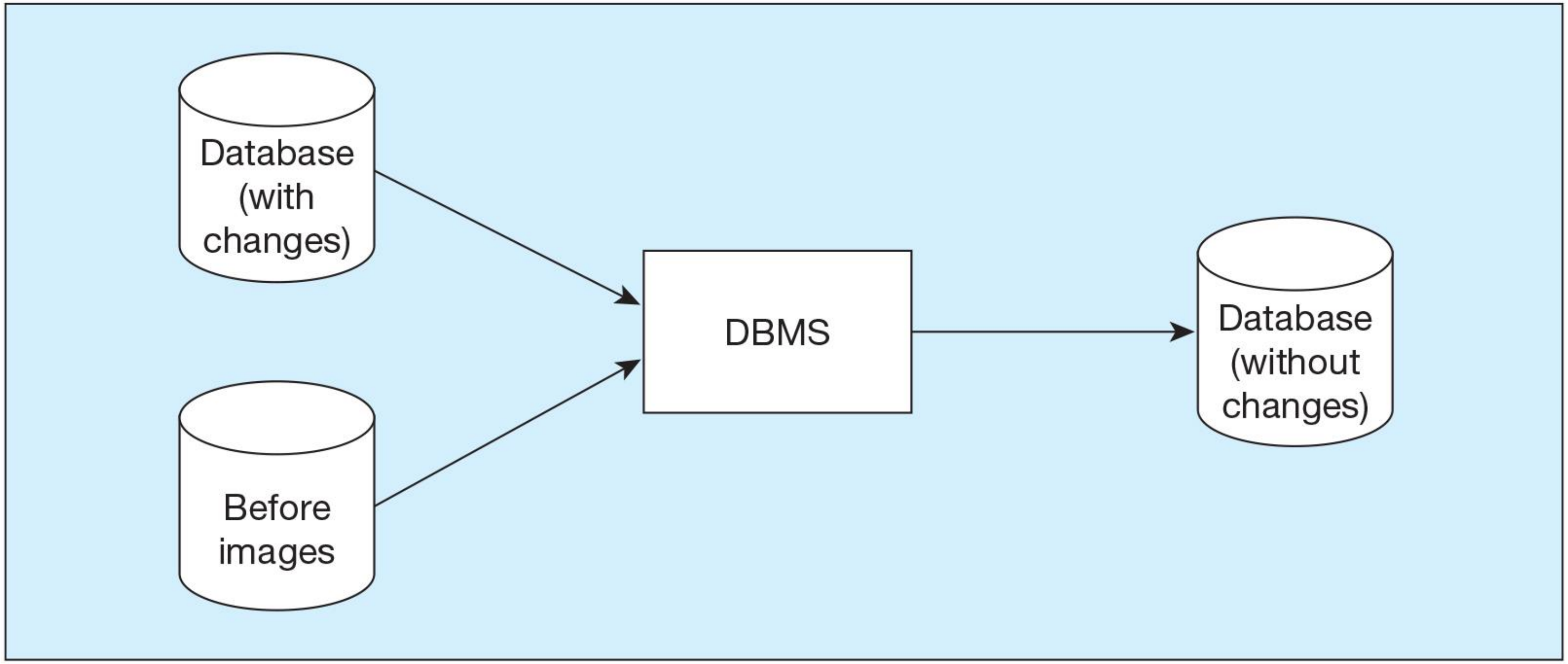
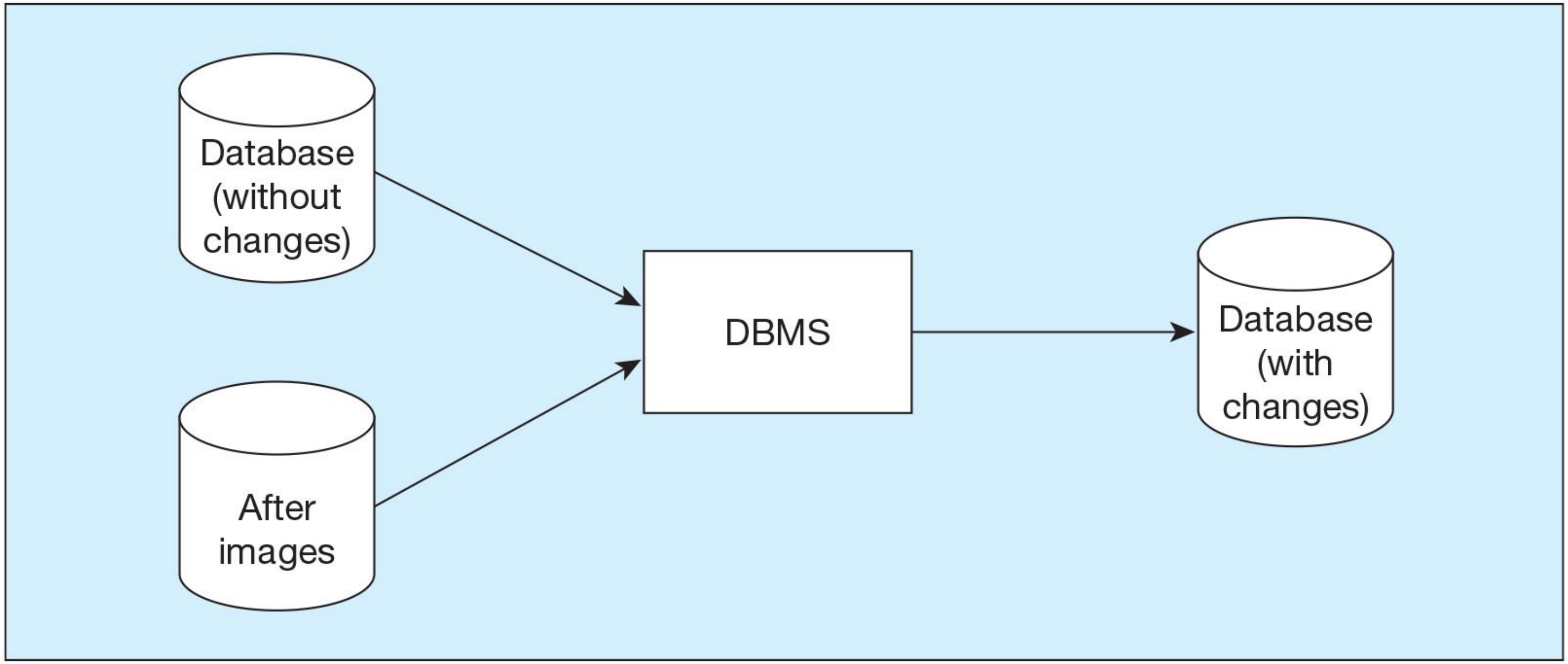


Figure 8-14 Basic Recovery Techniques (2 of 2)

(b) Rollforward



Responses to Database Failures (1 of 2)

- Aborted transaction
  - Rollback (preferred)
  - Rollforward/return transactions to state just prior to abort
- Incorrect data (update inaccurate)
  - Rollback (preferred)
  - Reprocess transactions without inaccurate data updates
  - Compensating transactions

Responses to Database Failures (2 of 2)

- System failure (database intact)
  - Switch to duplicate database (preferred)
  - Rollback
  - Restart from checkpoint (rollforward)
- Database destruction
  - Switch to duplicate database (preferred)
  - Rollforward
  - Reprocess transactions

Disaster Recovery

- Develop a detailed written disaster recovery plan, and test this regularly
- Choose and train a multidisciplinary team to carry out the plan
- Establish a backup data center at an off-site location, located a sufficient distance from the primary site
- Send backup copies of databases to the backup data center on a scheduled basis

Cloud-Based Data Management Services (1 of 2)

- Cloud computing
  - Provisioning/acquiring computing services on demand using centralized resources accessed through public Internet or private networks
- Infrastructure-as-a-Service (IaaS)
  - Cloud service involving hardware and various types of systems software resources
- Platform-as-a-Service (PaaS)
  - Cloud service involving hardware and various types of systems software resources

Pearson

Cloud-Based Data Management Services (2 of 2)

- Software-as-a-Service (SaaS)
  - Cloud service involving software solutions/applications intended to directly address the needs of a noncomputing activity
- Database-as-a-Service (DBaaS)
  - Cloud service involving data management cloud platform service

Benefits of Cloud-Based Data Management Services

- No need for initial investments in hardware, physical facilities, and systems software
- Significantly lower need for internal expertise in the management of the database infrastructure
- Better visibility of overall costs of data management
- Increased level of flexibility (elasticity) in situations when capacity needs to fluctuate significantly
- Allows organizations to explore new data management technologies more easily
- Mature cloud service providers have expertise to provide a high level of availability, reliability, and security

Downside of Cloud-Based Data Management Services

- Existing systems do not yet provide capacity using a model that would automatically adapt to the changing requirements targeting the system
- Current systems are not yet providing full consistency guarantees in a highly distributed environment
- Live migration is still a challenging task that requires manual planning, initiation, and management
- It is challenging to be able to monitor the extent to which cloud providers are maintaining their Service Level Agreement (SLA) commitments
- DBaaS solutions are still struggling to find fully scalable models for providing ACID support for transactions

Copyright