Database Systems: Design, Implementation, and Management

Ninth Edition

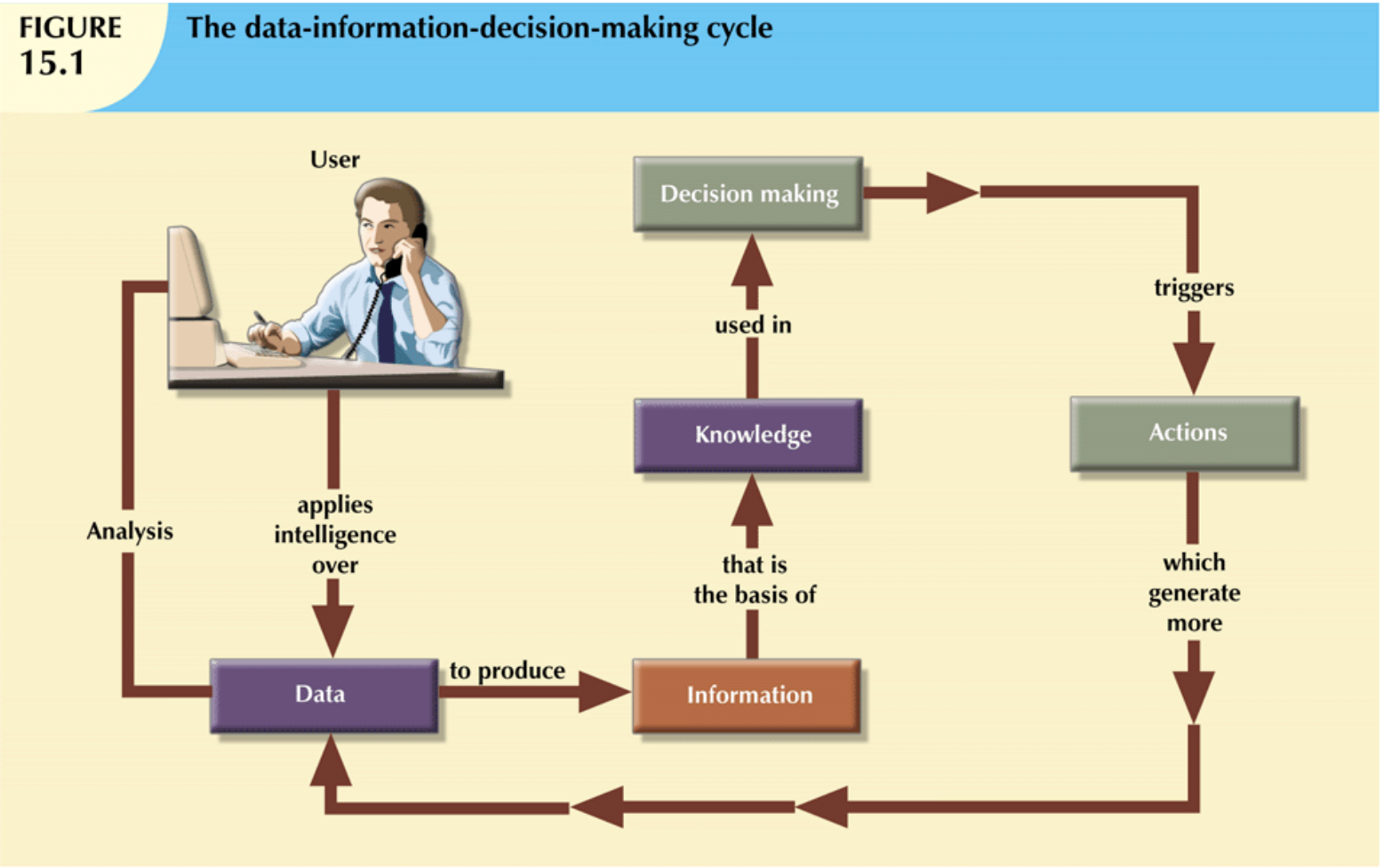
Chapter 12

Database Administration and Security

Data as a Corporate Asset

- Data:
  - Valuable asset that requires careful management
  - Valuable resource that translates into information
- Accurate, timely information triggers actions that enhance company's position and generate wealth
- **Dirty data**
  - Data that suffer from inaccuracies and inconsistencies
  - Threat to organizations

FIGURE 15.1 The data-information-decision-making cycle



Data as a Corporate Asset (cont'd.)

- **Data quality**
  - Comprehensive approach to ensuring the accuracy, validity, and timeliness of the data
- **Data profiling software**
  - Consists of programs that gather statistics and analyze existing data sources
- **Master data management (MDM) software**
  - Helps prevent dirty data by coordinating common data across multiple systems.

The Need for and Role of Databases in an Organization

- Database's predominant role is to support managerial decision making at all levels
- DBMS facilitates:
  - Interpretation and presentation of data
  - Distribution of data and information
  - Preservation and monitoring of data
  - Control over data duplication and use
- Three levels to organization management:
  - Top, middle, operational

Introduction of a Database: Special Considerations

- Introduction of a DBMS is likely to have a profound impact
  - Might be positive or negative, depending on how it is administered
- Three aspects to DBMS introduction:
  - Technological
  - Managerial
  - Cultural
- One role of DBA department is to educate end users about system uses and benefits

The Evolution of the Database Administration Function

- Data administration has its roots in the old, decentralized world of the file system
- Advent of DBMS produced new level of data management sophistication
  - DP department evolved into information systems (IS) department
- Data management became increasingly complex
  - Development of **database administrator (DBA)** function

FIGURE 15.3 The placement of the DBA function

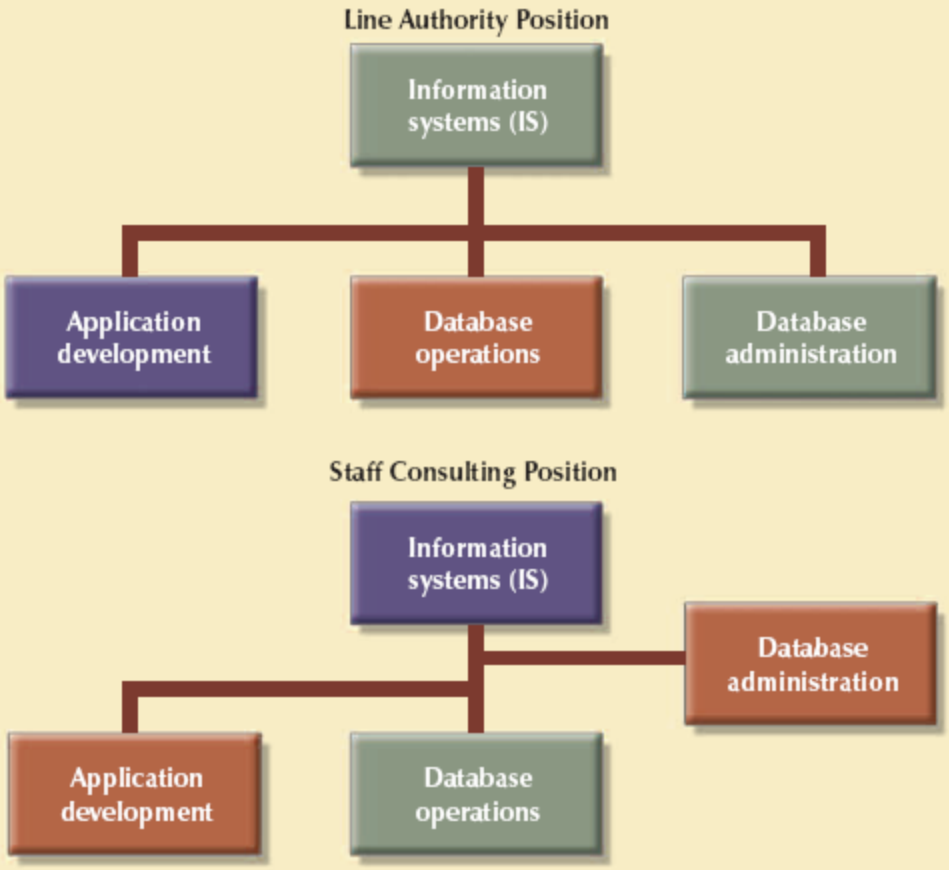
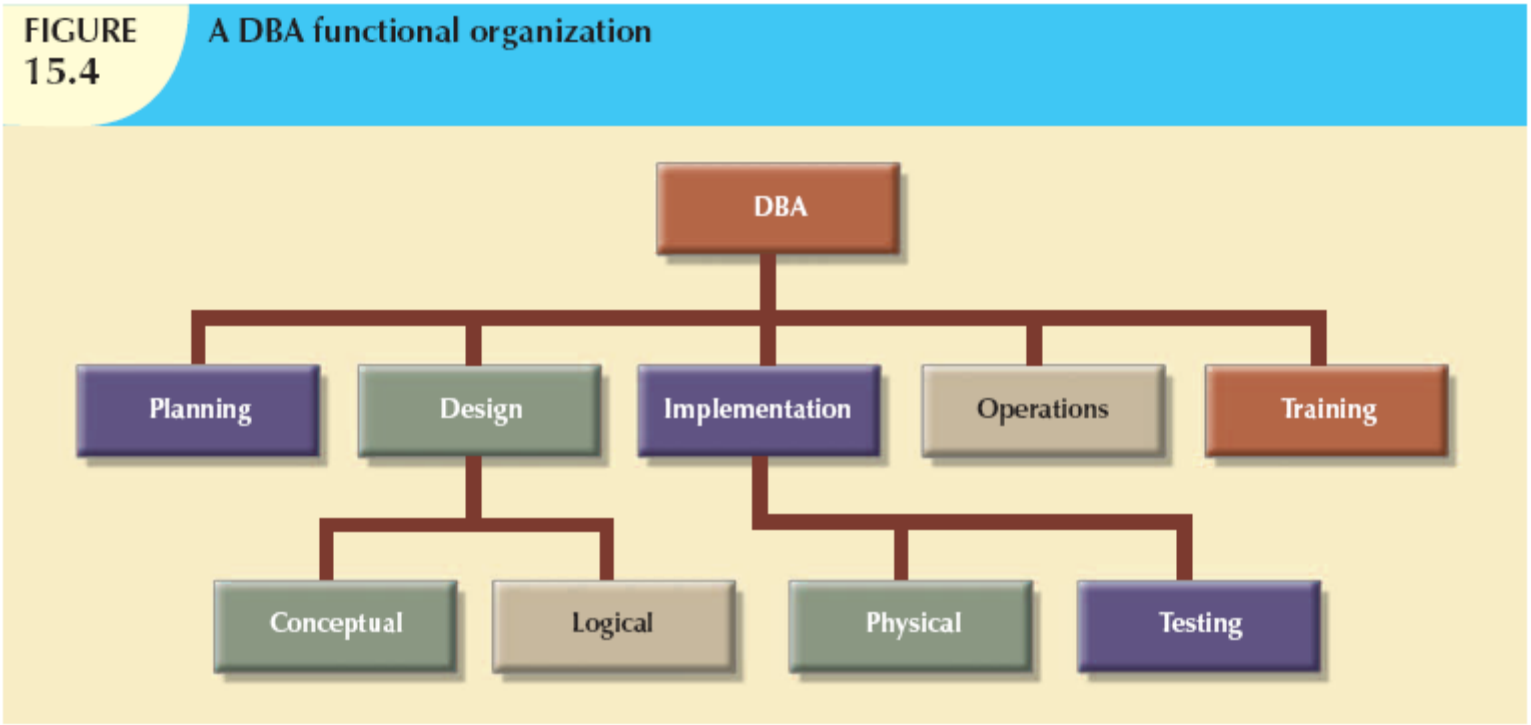


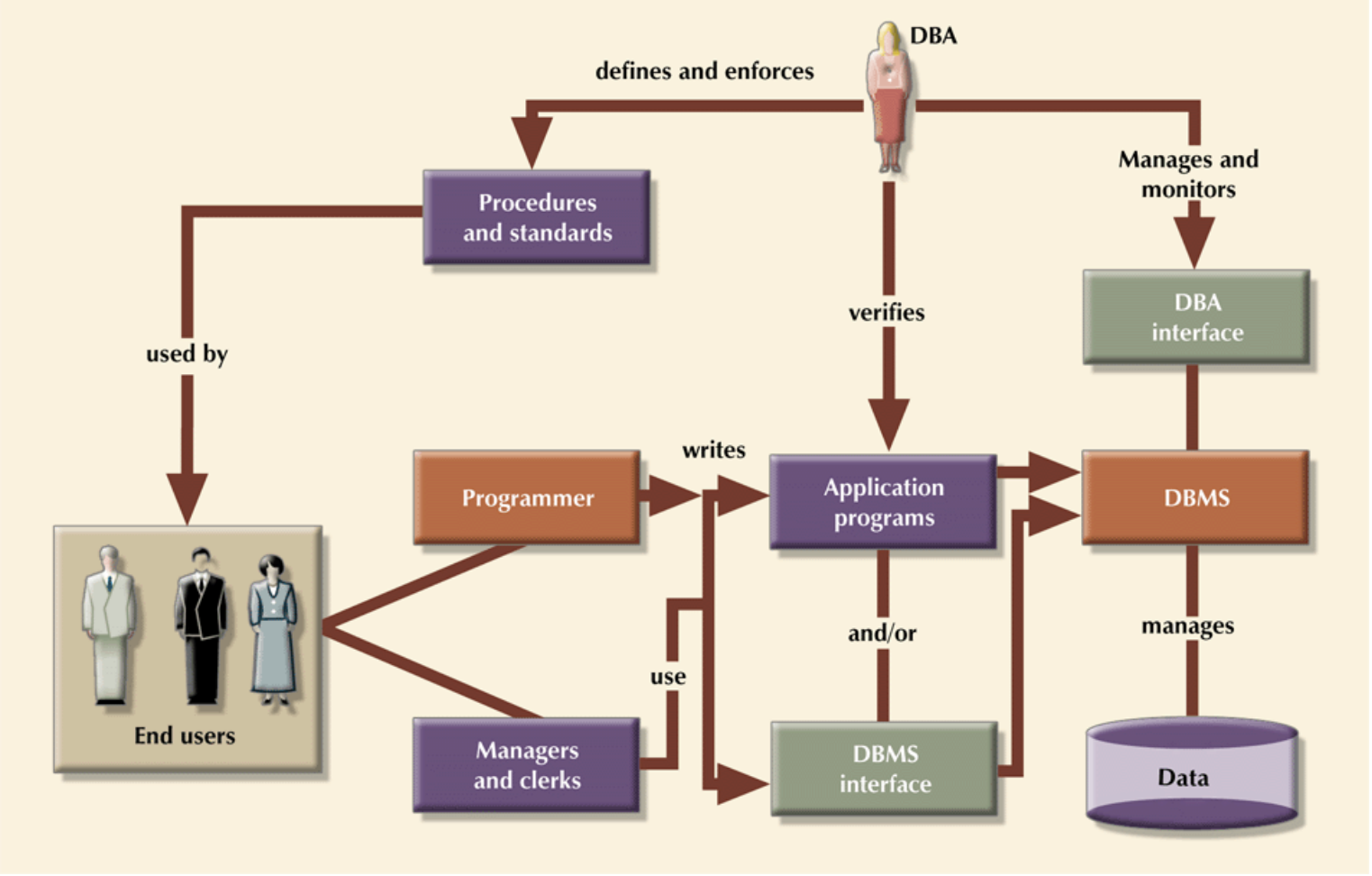
FIGURE 15.4 A DBA functional organization



The Database Environment's Human Component

- Even most carefully crafted database system cannot operate without human component
- Effective data administration requires both technical and managerial skills
- DA must set data administration goals
- DBA is focal point for data/user interaction
- Need for diverse mix of skills

FIGURE 15.6 A summary of DBA activities



The DBA's Managerial Role

- DBA responsible for:
  - Coordinating, monitoring, allocating resources
    - Resources include people and data
  - Defining goals and formulating strategic plans
- Interacts with end user by providing data and information
- Enforces **policies**, **standards**, **procedures**

The DBA's Managerial Role (cont'd.)

- Manages security, privacy, integrity
- Ensures data can be fully recovered
  - In large organizations, **database security officer (DSO)** responsible for disaster management
- Ensures data is distributed appropriately
  - Makes it easy for authorized end users to access the database

TABLE 15.3 DBA Activities and Services

DBA ACTIVITY		DBA SERVICE
Planning		End-user support
Organizing		Policies, procedures, and standards
Testing	of	Data security, privacy, and integrity
Monitoring		Data backup and recovery
Delivering		Data distribution and use

The DBA’s Technical Role

- Evaluates, selects, and installs DBMS and related utilities
- Designs and implements databases and applications
- Tests and evaluates databases and applications

The DBA’s Technical Role (cont’d.)

- Operates DBMS, utilities, and applications
- Trains and supports users
- Maintains DBMS, utilities, and applications

Security

- Securing data entails securing overall information system architecture
- **Confidentiality**: data protected against unauthorized access
- **Integrity**: keep data consistent and free of errors or anomalies
- **Availability**: accessibility of data by authorized users for authorized purposes

Security Policies

- Database security officer secures the system and the data
  - Works with the database administrator
- **Security policy**: collection of standards, policies, procedures to guarantee security
  - Ensures auditing and compliance
- Security audit process identifies security vulnerabilities
  - Identifies measures to protect the system

Security Vulnerabilities

- **Security vulnerability**: weakness in a system component
  - Could allow unauthorized access or cause service disruptions
- **Security threat**: imminent security violation
  - Could occur at any time
- **Security breach** yields a database whose integrity is either:
  - Preserved
  - Corrupted

TABLE 15.4 Sample Security Vulnerabilities and Related Measures

SYSTEM COMPONENT	SECURITY VULNERABILITY	SECURITY MEASURES
People	• User sets a blank password. • Password is short or includes birth date. • User leaves office door open all the time. • User leaves payroll information on screen for long periods of time.	• Enforce complex password policies. • Use multilevel authentication. • Use security screens and screen savers. • Educate users about sensitive data. • Install security cameras. • Use automatic door locks.
Workstation and Servers	• User copies data to flash drive. • Workstation is used by multiple users. • Power failure crashes computer. • Unauthorized personnel can use computer. • Sensitive data stored in laptop computer. • Data lost due to stolen hard disk/laptop. • Natural disasters—earthquake, flood, etc.	• Use group policies to restrict use of flash drives. • Assign user access rights to workstations. • Install uninterrupted power supplies (UPSs). • Add security lock devices to computers. • Implement a "kill" switch for stolen laptops. • Create and test data backup and recovery plans. • Insure system against natural disasters—use co-location strategies.

TABLE 15.4 Sample Security Vulnerabilities and Related Measures (continued)

SYSTEM COMPONENT	SECURITY VULNERABILITY	SECURITY MEASURES
Operating System	• Buffer overflow attacks. • Virus attacks. • Root kits and worm attacks. • Denial of service attacks. • Trojan horses. • Spyware applications. • Password crackers.	• Apply OS security patches and updates. • Apply application server patches. • Install antivirus and antispyware software. • Enforce audit trails on the computers. • Perform periodic system backups. • Install only authorized applications. • Use group policies to prevent unauthorized installs.
Applications	• Application bugs—buffer overflow. • SQL injection, session hijacking, etc. • Application vulnerabilities—cross-site scripting, nonvalidated inputs. • E-mail attacks: spamming, phishing, etc. • Social engineering e-mails.	• Test application programs extensively. • Built safeguards in code. • Do extensive vulnerability testing in applications. • Install spam filter/antivirus for e-mail system. • Use secure coding techniques (see *www.owasp.org*). • Educate users about social engineering attacks.
Network	• IP spoofing. • Packet sniffers. • Hacker attacks. • Clear passwords on network.	• Install firewalls. • Virtual Private Networks (VPN). • Intrusion Detection Systems (IDS). • Network Access Control (NAC). • Network activity monitoring.
Data	• Data shares are open to all users. • Data can be accessed remotely. • Data can be deleted from shared resource.	• Implement file system security. • Implement share access security. • Use access permission. • Encrypt data at the file system or database level.

Database Security

- Refers to the use of DBMS features and other measures to comply with security requirements
- DBA secures DBMS from installation through operation and maintenance
- **Authorization management**:
  - User access management
  - View definition
  - DBMS access control
  - DBMS usage monitoring

Database Administration Tools

- Two main types of data dictionaries:
  - Integrated
  - Standalone
- **Active data dictionary** is automatically updated by the DBMS with every database access
- **Passive data dictionary** requires running a batch process
- Main function: store description of all objects that interact with database

Database Administration Tools (cont'd.)

- Data dictionary that includes data external to DBMS becomes flexible tool
  - Enables use and allocation of all of an organization's information
- Metadata is often the basis for monitoring database use
  - Also for assigning access rights to users
- DBA uses data dictionary to support data analysis and design

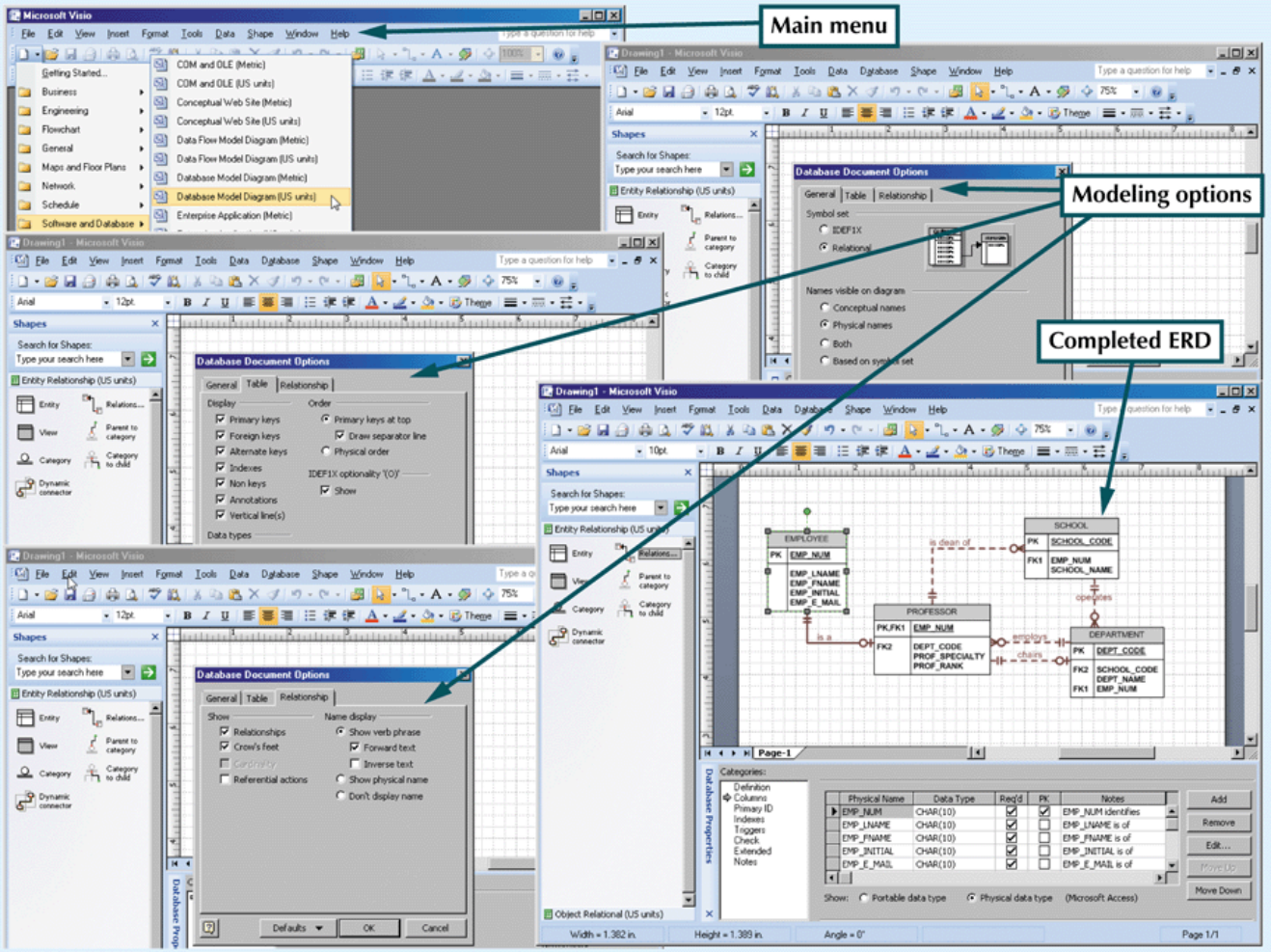
CASE Tools

- **Computer-aided systems engineering**
  - Automated framework for SDLC
  - Structured methodologies and powerful graphical interfaces
- **Front-end CASE tools** provide support for planning, analysis, and design phases
- **Back-end CASE tools** provide support for coding and implementation phases

CASE Tools (cont'd.)

- Typical CASE tool has five components
  - Graphics for diagrams
  - Screen painters and report generators
  - Integrated repository
  - Analysis segment
  - Program documentation generator

FIGURE 15.7 An example of a CASE tool: Visio Professional



Developing a Data Administration Strategy

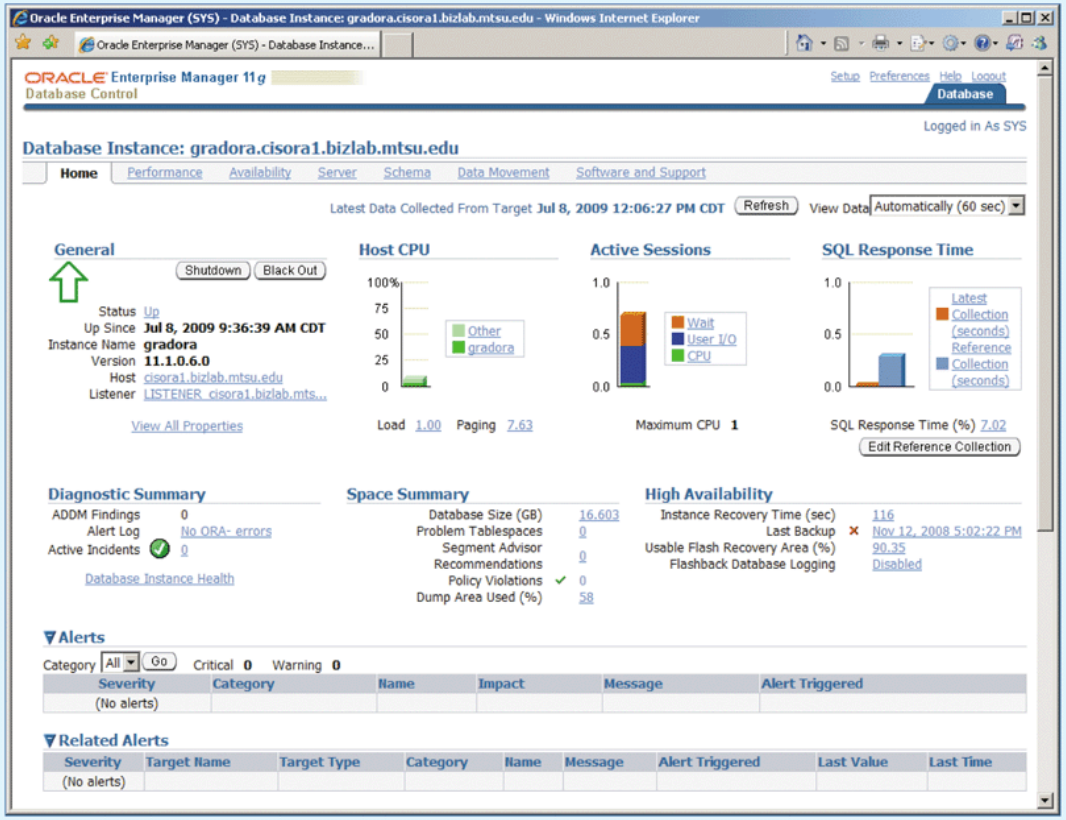
- **Information engineering (IE)** translates strategic goals into data and applications
- **Information systems architecture (ISA)** is the output of IE process
- Implementing IE is a costly process
  - Provides a framework that includes use of computerized, automated, and integrated tools
- Success of information systems strategy depends on critical success factors
  - Managerial, technological, and corporate culture

The DBA at Work: Using Oracle for Database Administration

- Technical tasks handled by the DBA in a specific DBMS:
  - Creating and expanding database storage structures
  - Managing database objects
  - Managing end-user database environment
  - Customizing database initialization parameters
- All DBMS vendors provide programs to perform database administrative tasks

Oracle Database Administration Tools

FIGURE 15.9 The Oracle Enterprise Manager interface



The Default Login

- Must connect to the database to perform administrative tasks
  - Username with administrative privileges
- Oracle automatically creates SYSTEM and SYS user IDs with administrative privileges
- Define preferred credentials by clicking on **Preferences** link, then **Preferred Credentials**
- Username and passwords are database-specific

Ensuring an Automatic RDBMS Start

- DBA ensures database access is automatically started when computer turned on
- A *service* is a Windows system name for a special program that runs automatically
  - Part of the operating system
- **Database instance**: separate location in memory reserved to run the database
  - May have several databases running in memory at the same time

Creating Tablespaces and Datafiles

- Database composed of one or more tablespaces
- **Tablespace** is a logical storage space
  - Physically stored in one or more datafiles
- **Datafile** physically stores the database's data
  - Each datafile can reside in a different directory on the hard disk
- Database has 1:M relationship with tablespaces
- Tablespace has 1:M relationship with datafiles

FIGURE 15.12 The Oracle Storage Manager

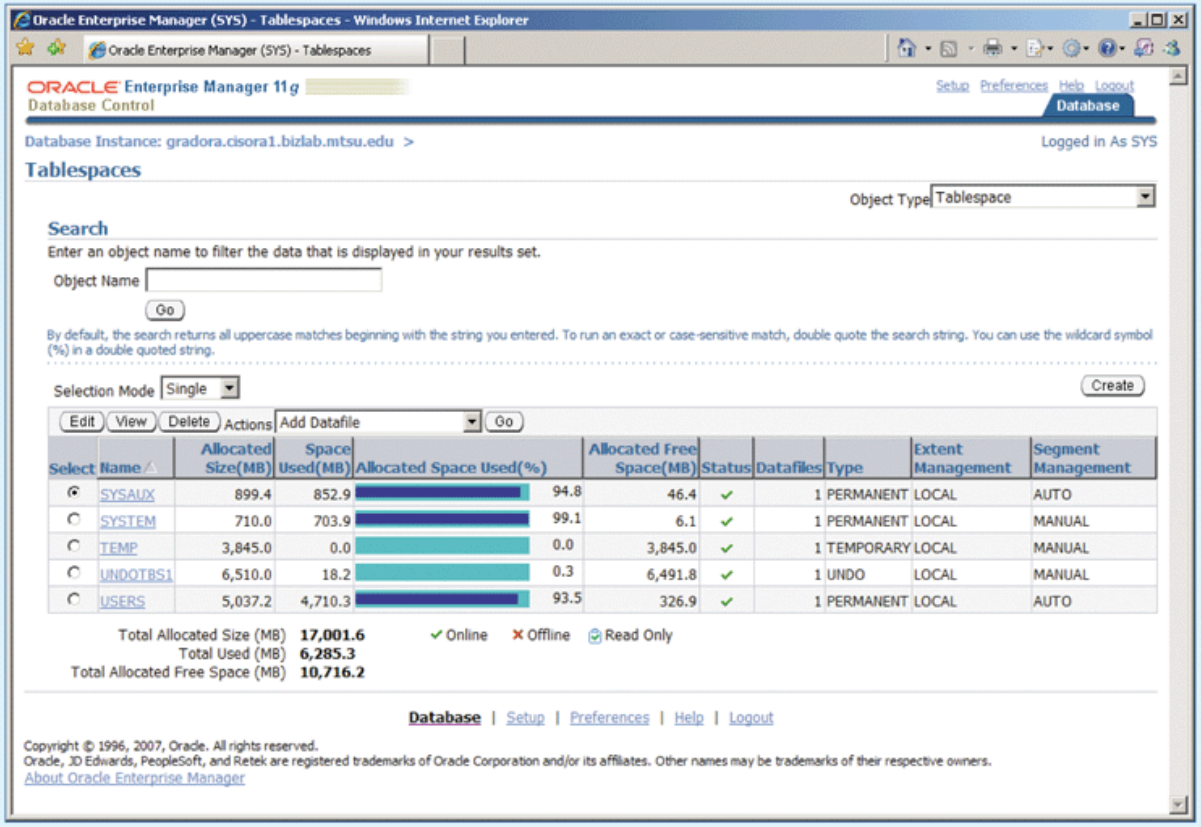
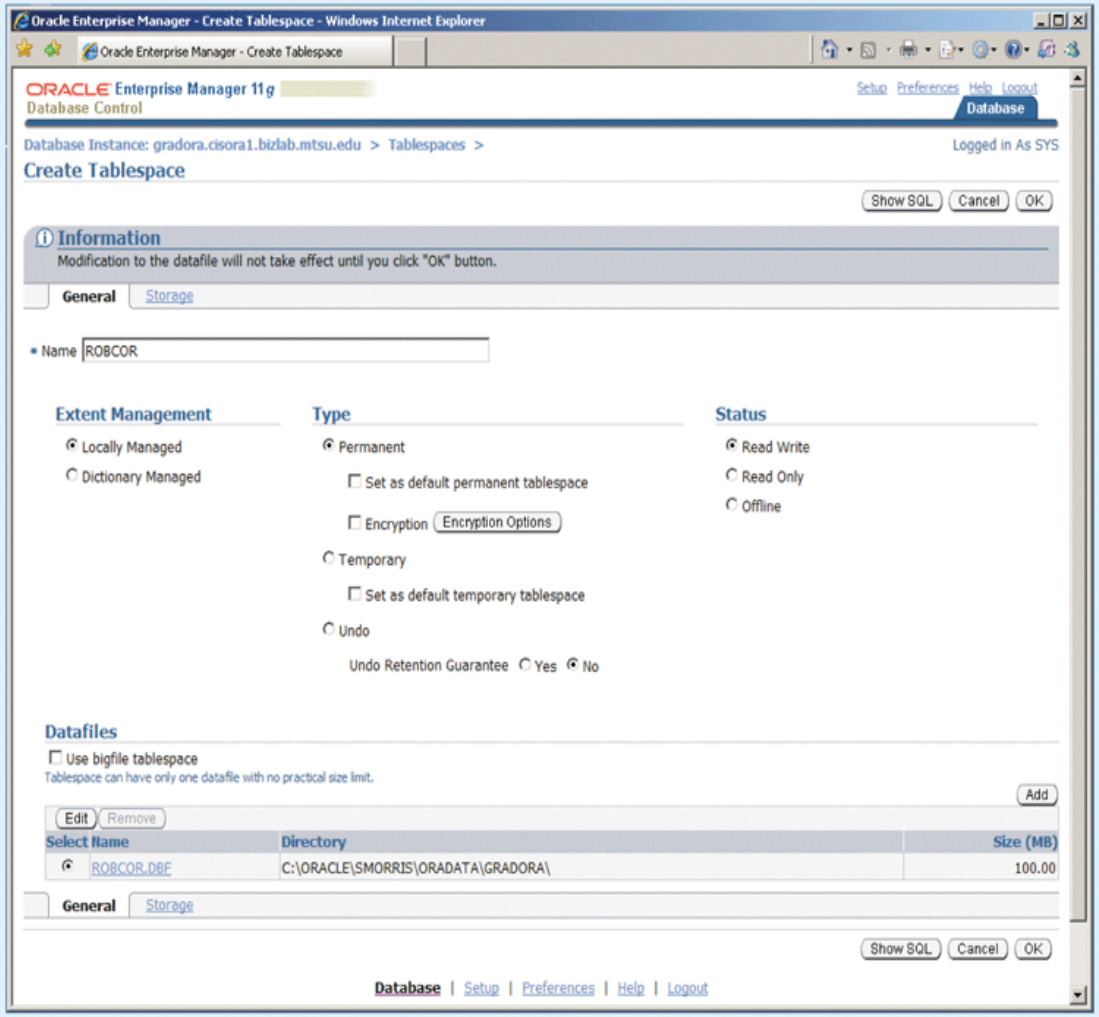


FIGURE 15.13 Creating a new tablespace



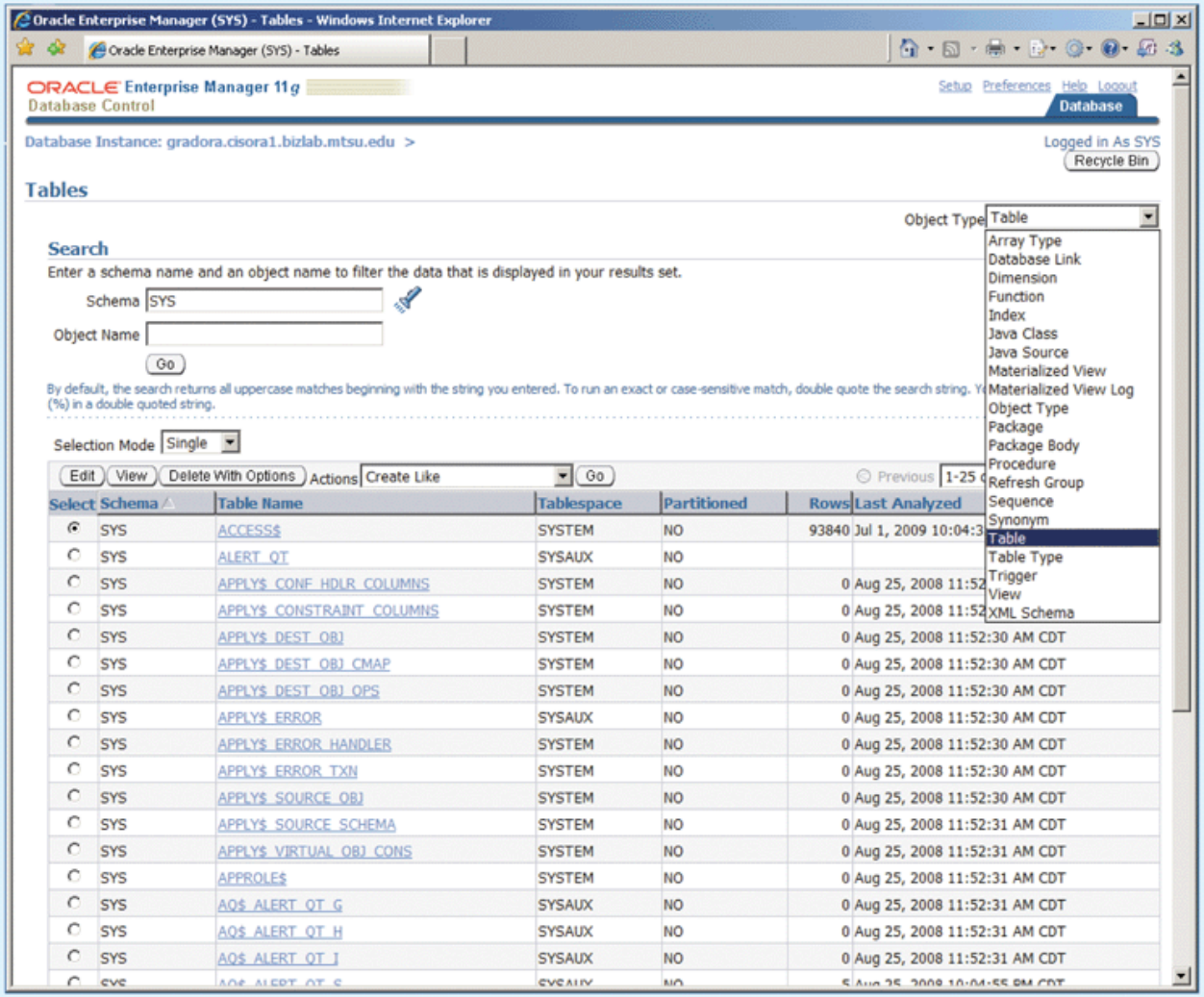
Managing the Database Objects: Tables, Views, Triggers, and Procedures

- **Database object**: any object created by end users
- **Schema**: logical section of the database that belongs to a given user
  - Schema identified by a username
  - Within the schema, users create their own tables and other objects
- Normally, users are authorized to access only the objects that belong to their own schemas

Managing Users and Establishing Security

- **User**: uniquely identifiable object
  - Allows a given person to log on to the database
- **Role**: a named collection of database access privileges
  - Authorizes a user to connect to the database and use system resources
- **Profile**: named collection of settings
  - Controls how much of a resource a given user can use

FIGURE 15.14 The Oracle Schema Manager



Customizing the Database Initialization Parameters

- Fine-tuning requires modification of database configuration parameters
  - Some are changed in real time using SQL
  - Some affect database instance
  - Others affect entire RDBMS and all instances
- Initialization parameters reserve resources used by the database at run time
- After modifying parameters, may need to restart the database

FIGURE 15.17 The Oracle Enterprise Manager – Initialization Parameters page