
# Chapter 16

# Database Administration and Security

# Learning Objectives

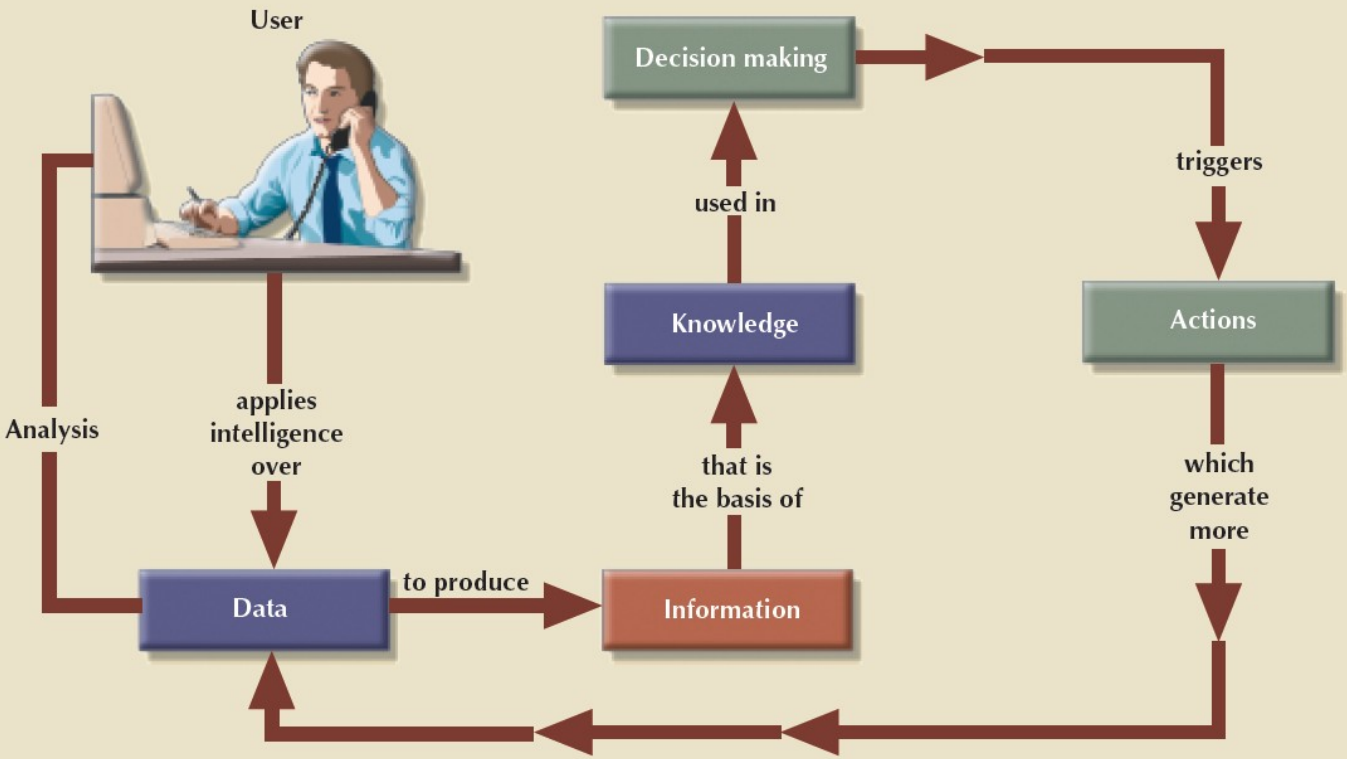
- In this chapter, you will learn:
  - That data are a valuable business asset requiring careful management
  - How a database plays a critical role in an organization
  - That the introduction of a DBMS has important technological, managerial, and cultural consequences for an organization
  - About the database administrator's managerial and technical roles

# Learning Objectives

- In this chapter, you will learn:
  - About data security, database security, and the information security framework
  - About several database administration tools and strategies
  - How cloud-based data services impact the DBA's role
  - How various technical tasks of database administration are performed with Oracle

# Figure 16.1 - The Data-Information-Decision Making Cycle

FIGURE 16.1 THE DATA-INFORMATION-DECISION-MAKING CYCLE



# Data

## Dirty data

- Data that suffer from inaccuracies and inconsistencies

## Data quality

- Ensuring accuracy, validity, and timeliness of data

## Data profiling software

- Determine data patterns and compare them against standards defined by the organization

## Master data management (MDM) software

- Helps prevent dirty data by coordinating across multiple systems

# Need for and Role of a Database in an Organization

## At the top management level

- Enable strategic decision making and planning
- Identify growth opportunities
- Define and enforce organizational policies
- Reduce costs and boost productivity
- Provide feedback

## At the middle management level

- Deliver the data required for tactical planning
- Monitor the use of resources
- Evaluate performance
- Enforce security and privacy of data in the database

## At the operational management level

- Represent and support company operations
- Produce query results within specified performance levels
- Enhance the company's short-term operations

# Introduction of a Database: Special Considerations

## Technological aspect

- Selecting, installing, configuring, and monitoring the DBMS to ensure that it operates efficiently

## Managerial aspect

- Careful planning to create an appropriate organizational structure

## Cultural aspect

- Listening to people's concerns about the system and explaining its uses and benefits

# Evolution of the Database Administration Function

**Information systems (IS) department**

- Provides end users with data management support and solutions for information needs

**Database administrator**

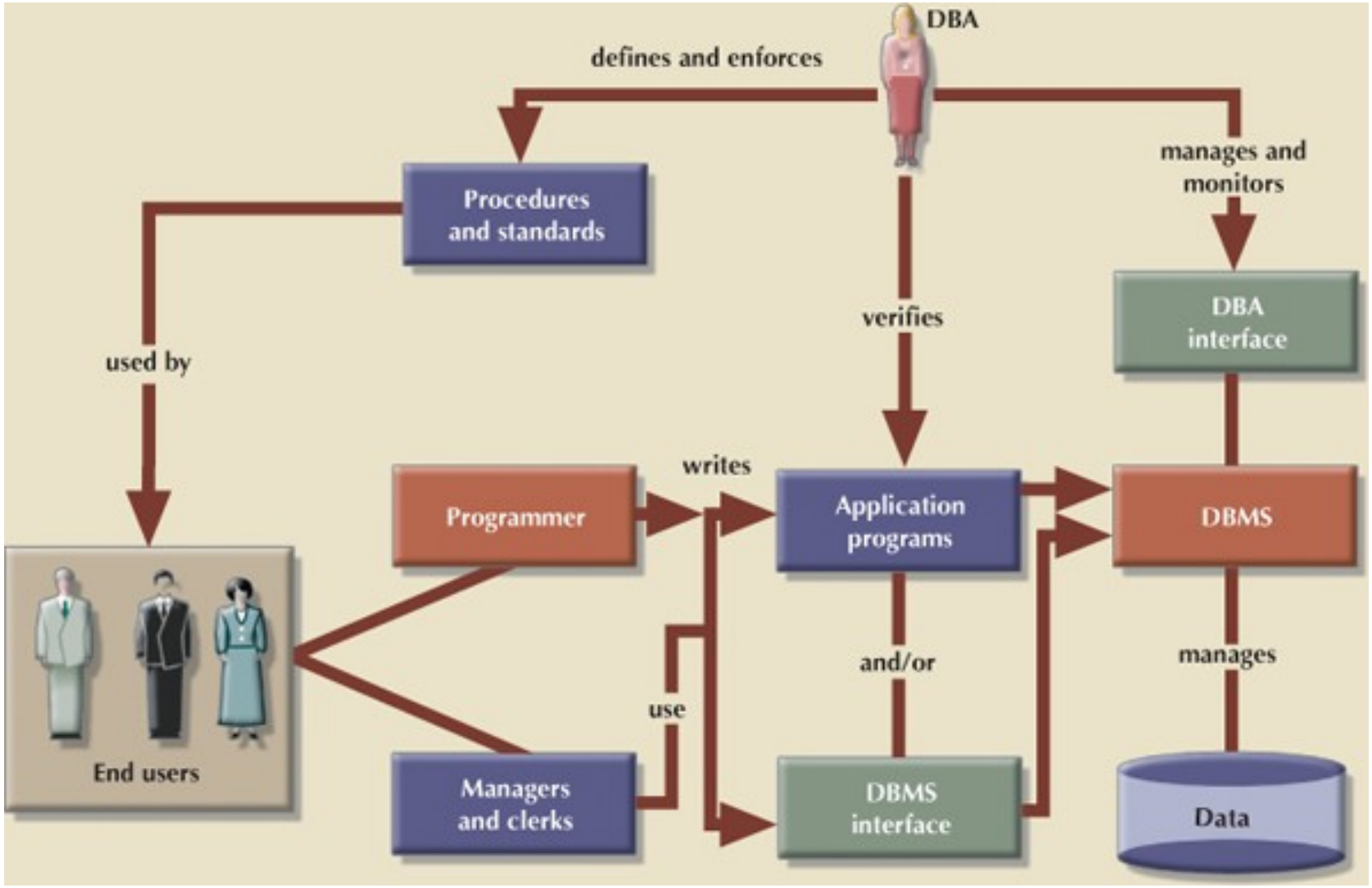
- Responsible for control of the centralized and shared database

**Systems administrator**

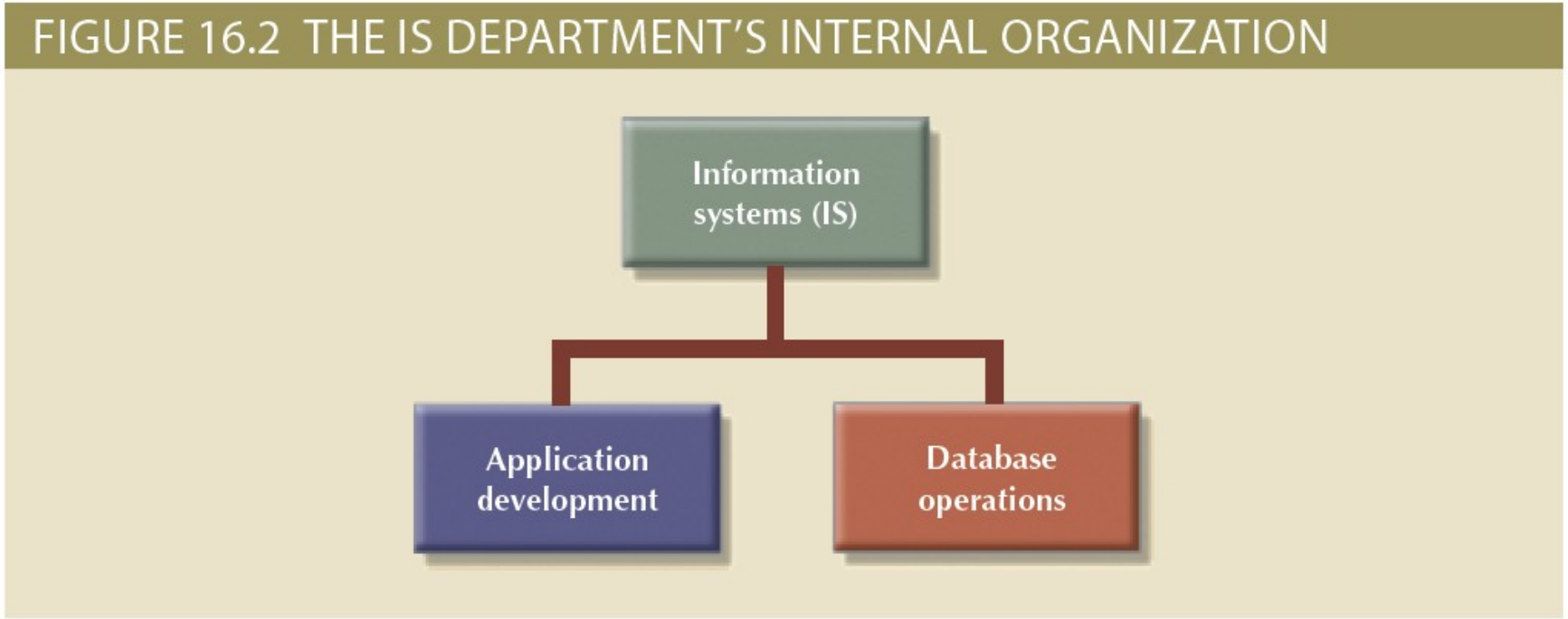
- General coordinator of all DBAs

**Data administrator (DA)** or **information resource manager (IRM)**

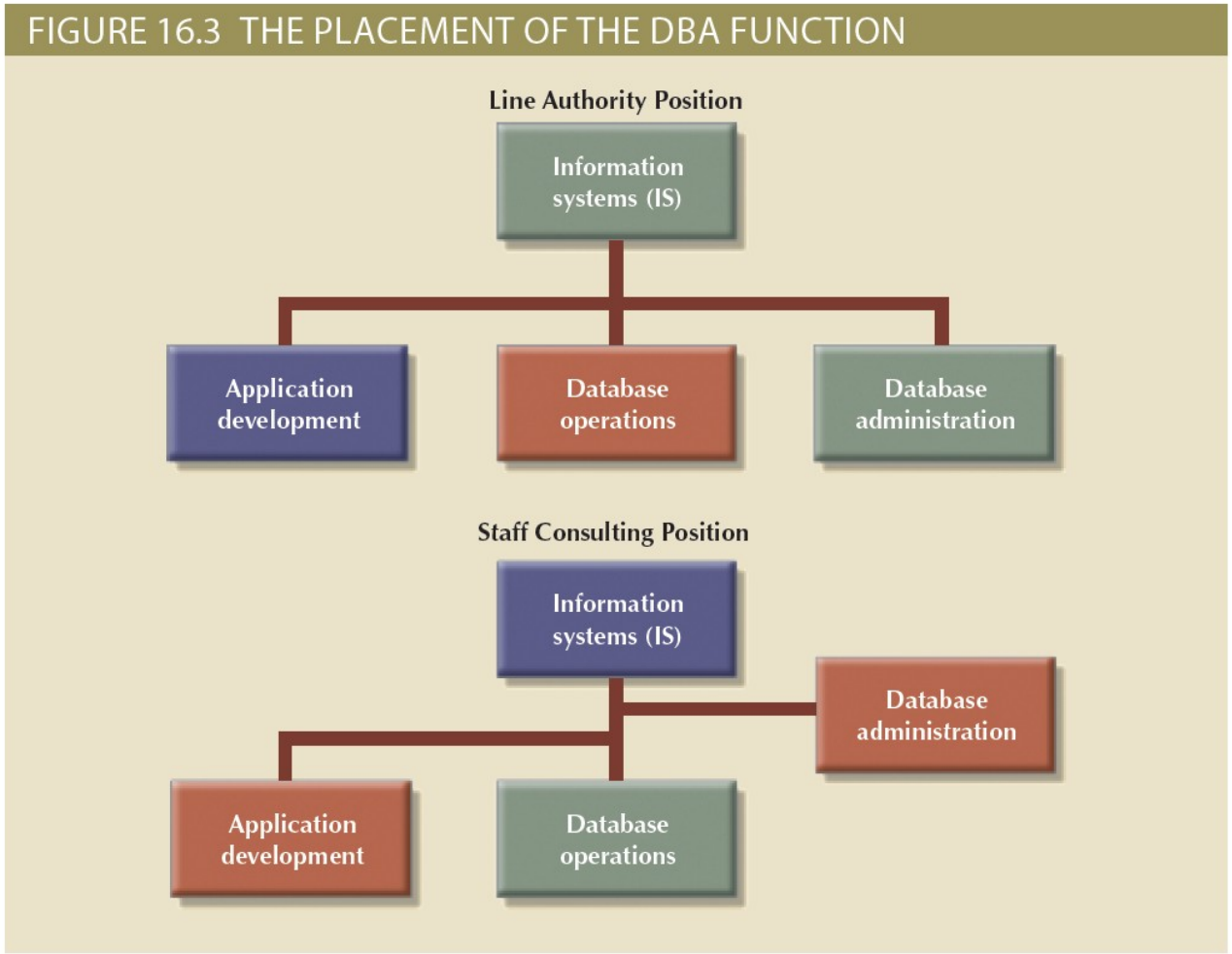
- Has a higher degree of responsibility and authority than the DBA



# Figure 16.2 - The IS Department's Internal Organization

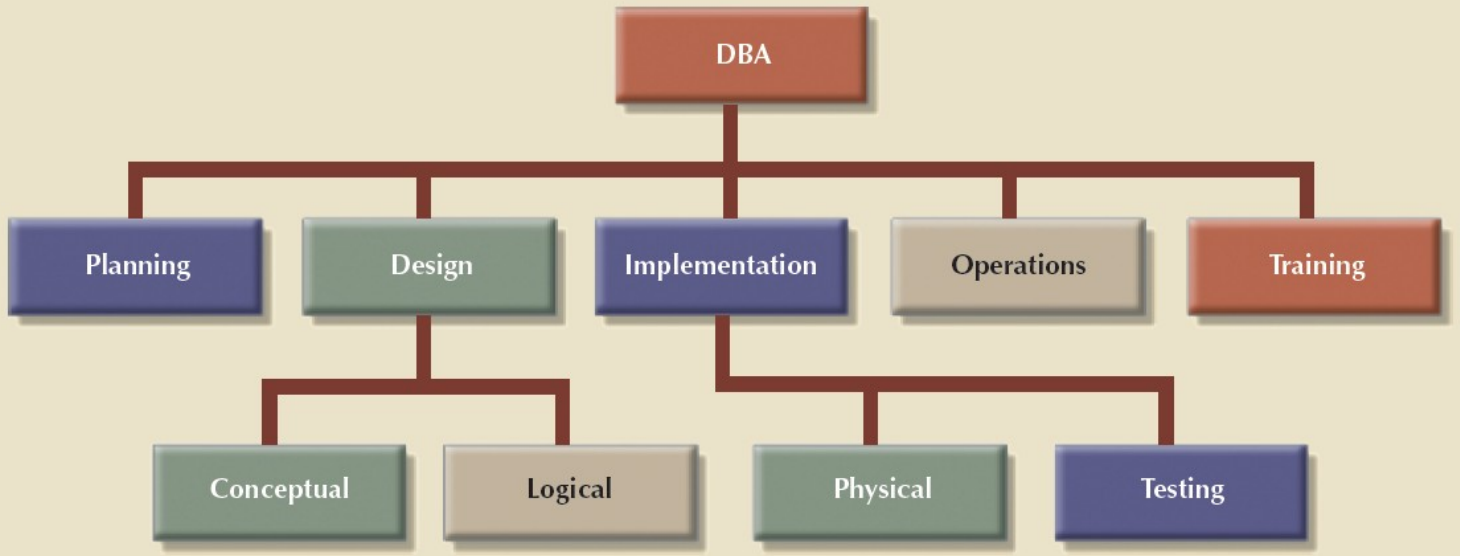


# Figure 16.3 - The Placement of the DBA Function



# Figure 16.4 - A DBA Functional Organization

FIGURE 16.4 A DBA FUNCTIONAL ORGANIZATION



# Table 16.1 - Contrasting DA and DBA Activities and Characteristics

TABLE 16.1

**CONTRASTING DA AND DBA ACTIVITIES AND CHARACTERISTICS**

| DATA ADMINISTRATOR (DA) | DATABASE ADMINISTRATOR (DBA) |
|---|---|
| Performs strategic planning | Controls and supervises |
| Sets long-term goals | Executes plans to reach goals |
| Sets policies and standards | Enforces policies and procedures<br>Enforces programming standards |
| Job is broad in scope | Job is narrow in scope |
| Focuses on the long term | Focuses on the short term (daily operations) |
| Has a managerial orientation | Has a technical orientation |
| Is DBMS-independent | Is DBMS-specific |

# Table 16.2 - Desired DBA Skills

TABLE 16.2

**DESIRED DBA SKILLS**

| MANAGERIAL | TECHNICAL |
|---|---|
| Broad business understanding | Broad data-processing background and up-to-date knowledge of database technologies |
| Coordination skills | Understanding of Systems Development Life Cycle |
| Analytical skills | Structured methodologies<br>• Data flow diagrams<br>• Structure charts<br>• Programming languages |
| Conflict resolution skills | Knowledge of Database Life Cycle |
| Communication skills (oral and written) | Database modeling and design skills<br>• Conceptual<br>• Logical<br>• Physical |
| Negotiation skills | Operational skills: Database implementation, data dictionary management, security, and so on |
| Experience: 10 years in a large DP department | |

# DBA's Managerial Role

- Provide end-user support
- Enforce policies, procedures, and standards for correct data creation, usage, and distribution within the database
- Manage data security, privacy, and integrity
- Manage data backup and recovery
  - Fully recover data in case of data loss
  - **Database security officer (DSO)**: Ensures database security and integrity

# DBA's Managerial Role

- **Disaster management**: Planning, organizing, and testing of database contingency plans and recovery procedures
- Backup and recovery measures must include at least periodic data and application backups:
  - **Full backup** or **database dump**: Produces a complete copy of the entire database
  - **Incremental backup**: Produces a backup of all data since the last backup date
  - **Concurrent backup**: Takes place while the user is working on the database

# DBA's Managerial Role

- Backup and recovery measures must include at least:
  - Proper backup identification
  - Convenient and safe backup storage
  - Physical protection of both hardware and software
  - Personal access control to the software of a database installation
  - Insurance coverage for the data in the database

# DBA's Managerial Role

- Additional points:
  - Data recovery and contingency plans must be tested, evaluated and practiced frequently
  - Backup and recovery plan not likely to cover all information system components
- Ensure data is distributed to the right people at the right time and in the right format

# DBA's Technical Role

- Evaluate, select, and install DBMS and related utilities
- Design and implement databases and applications
- Test and evaluate databases and applications
- Operate the DBMS, utilities, and applications
- Train and support users
- Maintain the DBMS, utilities, and applications

| 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 |

DBMS modeling

DBMS storage and capacity management

Application development support

Security and integrity

Backup and recovery

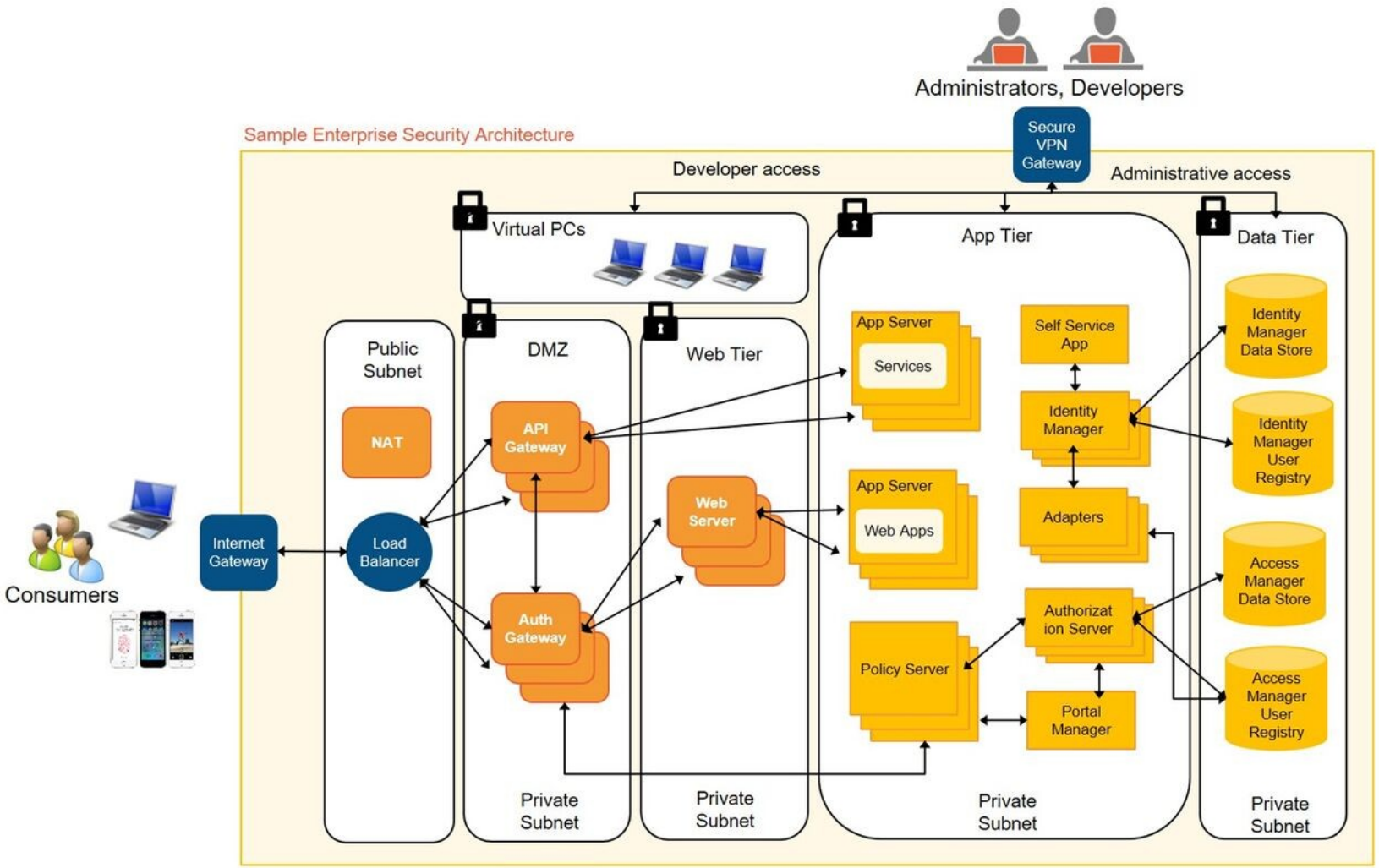
Concurrency control. Does the DBMS support multiple users? What levels of isolation (table, page, row) does the DBMS offer? How much manual coding is needed in the application programs?

Performance. How many transactions per second does the DBMS support? Are additional transaction processors needed? Is an in-memory database required to ensure top performance?

Interoperability and data distribution

Hardware

Data dictionary. Does the DBMS have an “accessible” data dictionary? Does the DBMS interface with any data dictionary tool? Does the DBMS support any open management tools?



# Security Goals

- **Confidentiality**: Protecting data against unauthorized access
- **Compliance**: Activities that meet data privacy and security reporting guidelines
- **Integrity**: Keeping data consistent and free of errors or anomalies
- **Availability**: Accessibility of data whenever required by authorized users and for authorized purposes

# Security Policies

- Collection of standards, policies, and procedures created to guarantee security
  - Ensures auditing and compliance
- Security audit process
  - Identifies security vulnerabilities
  - Identifies measures to protect the system

# Security Vulnerabilities

- Weakness in a system component that could allow unauthorized access or cause service disruptions
- Categories: Technical, managerial, cultural, and procedural
- **Security threat**: Imminent security violation
- **Security breach**: Occurs when a security threat is exploited and could lead to a database whose integrity is preserved or corrupted

# Table 16.4 - Sample Security Vulnerabilities and Related Protective Measures

TABLE 16.4

**SAMPLE SECURITY VULNERABILITIES AND RELATED PROTECTIVE MEASURES**

| SYSTEM COMPONENT | SECURITY VULNERABILITY | SECURITY MEASURES |
|---|---|---|
| People | • The user sets a blank password.<br>• The password is short or includes a birth date.<br>• The user leaves the office door open all the time.<br>• The user leaves payroll information on the screen for long periods of time. | • Enforce complex password policies.<br>• Use multilevel authentication.<br>• Use security screens and screen savers.<br>• Educate users about sensitive data.<br>• Install security cameras.<br>• Use automatic door locks. |
| Workstation and servers | • The user copies data to a flash drive.<br>• The workstation is used by multiple users.<br>• A power failure crashes the computer.<br>• Unauthorized personnel can use the computer.<br>• Sensitive data is stored on a laptop computer.<br>• Data is lost due to a stolen hard disk or laptop.<br>• A natural disaster occurs. | • Use group policies to restrict the use of flash drives.<br>• Assign user access rights to workstations.<br>• Install uninterrupted power supplies (UPSs).<br>• Add security locks to computers.<br>• Implement a kill switch for stolen laptops.<br>• Create and test data backup and recovery plans.<br>• Protect the system against natural disasters—use co-location strategies. |
| Operating system | • Buffer overflow attacks<br>• Virus attacks<br>• Root kits and worm attacks<br>• Denial-of-service attacks<br>• Trojan horses<br>• Spyware applications<br>• Password crackers | • Apply OS security patches and updates.<br>• Apply application server patches.<br>• Install antivirus and antispyware software.<br>• Enforce audit trails on the computers.<br>• Perform periodic system backups.<br>• Install only authorized applications.<br>• Use group policies to prevent unauthorized installations. |

# Table 16.4 - Sample Security Vulnerabilities and Related Protective Measures

TABLE 16.4

**SAMPLE SECURITY VULNERABILITIES AND RELATED PROTECTIVE MEASURES**

| SYSTEM COMPONENT | SECURITY VULNERABILITY | SECURITY MEASURES |
|---|---|---|
| Applications | • Application bugs—buffer overflow<br>• SQL injection, session hijacking, etc.<br>• Application vulnerabilities—cross-site scripting, nonvalidated inputs<br>• Email attacks—spamming, phishing, etc.<br>• Social engineering emails | • Test application programs extensively.<br>• Build safeguards into code.<br>• Do extensive vulnerability testing in applications.<br>• Install spam filters and antivirus software for email systems.<br>• Use secure coding techniques (see *www.owasp.org*).<br>• Educate users about social engineering attacks. |
| Network | • IP spoofing<br>• Packet sniffers<br>• Hacker attacks<br>• Clear passwords on network | • Install firewalls.<br>• Use virtual private networks (VPNs).<br>• Use intrusion detection systems (IDSs).<br>• Use network access control (NAC).<br>• Use network activity monitoring. |
| Data | • Data shares are open to all users.<br>• Data can be accessed remotely.<br>• Data can be deleted from a shared resource. | • Implement file system security.<br>• Implement share access security.<br>• Use access permission.<br>• Encrypt data at the file system or database level. |

# Designing Security Controls

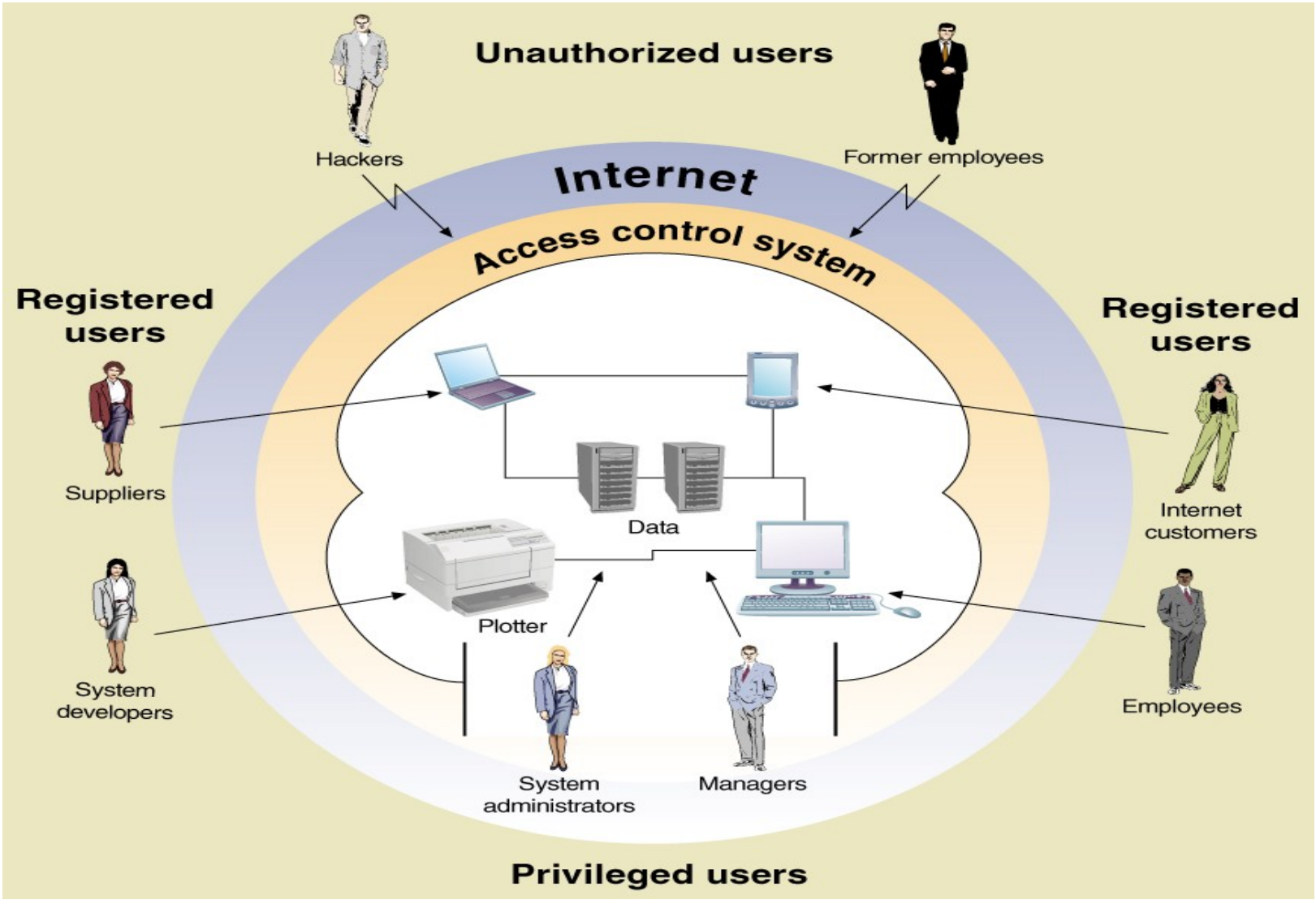
- Protect all assets against external threats

- Other objectives
  Protect and maintain a stable, functioning operating environment 24/7 (equipment, operating systems, DBMSs)
  Protect information and transactions during transmission across networks and Internet

Access Controls – Limit a person's ability to access servers, files, data, applications
  Authentication – to identify users --- Multifactor Authentication
  Access control list – list of valid users
  Authorization – authenticated user's list of permission level for each resource

  Registered Users – those with authorization
  Unauthorized Users – anyone not registered
  Privileged Users – those that maintain lists and systems

# Types of users



# Data Encryption

Method to secure data – stored or in transmission

Encryption – alter data so it is unrecognizable

Decryption – converted encrypted data back to readable format

Encryption Algorithm – mathematical transformation of the data

Encryption Key – a long data string that allows the same algorithm to produce unique encryptions

# Encryption Algorithm

Caesar Cipher

Plaintext

If he had anything confidential to say, he wrote it in cipher, that is, by so changing the order of the letters of the alphabet, that not a word could be made out.

8 a → i

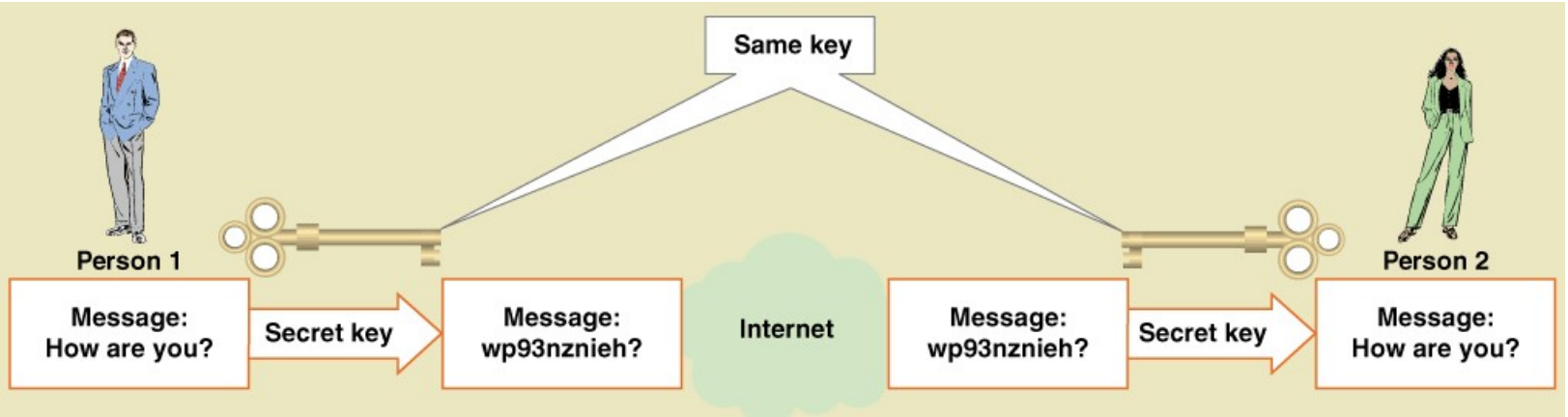
Ciphertext

Qn pm pil ivgbpqvo kwvnqlmvbqit bw aig, pm ezwbm qb qv kqxpmz, bpib qa, jg aw kpivoqvo bpm wzlmz wn bpm tmbbmza wn bpm itxpijmb, bpib vwb i ewzl kwctl jm uilm wcb.

https://cryptii.com/pipes/caesar-cipher

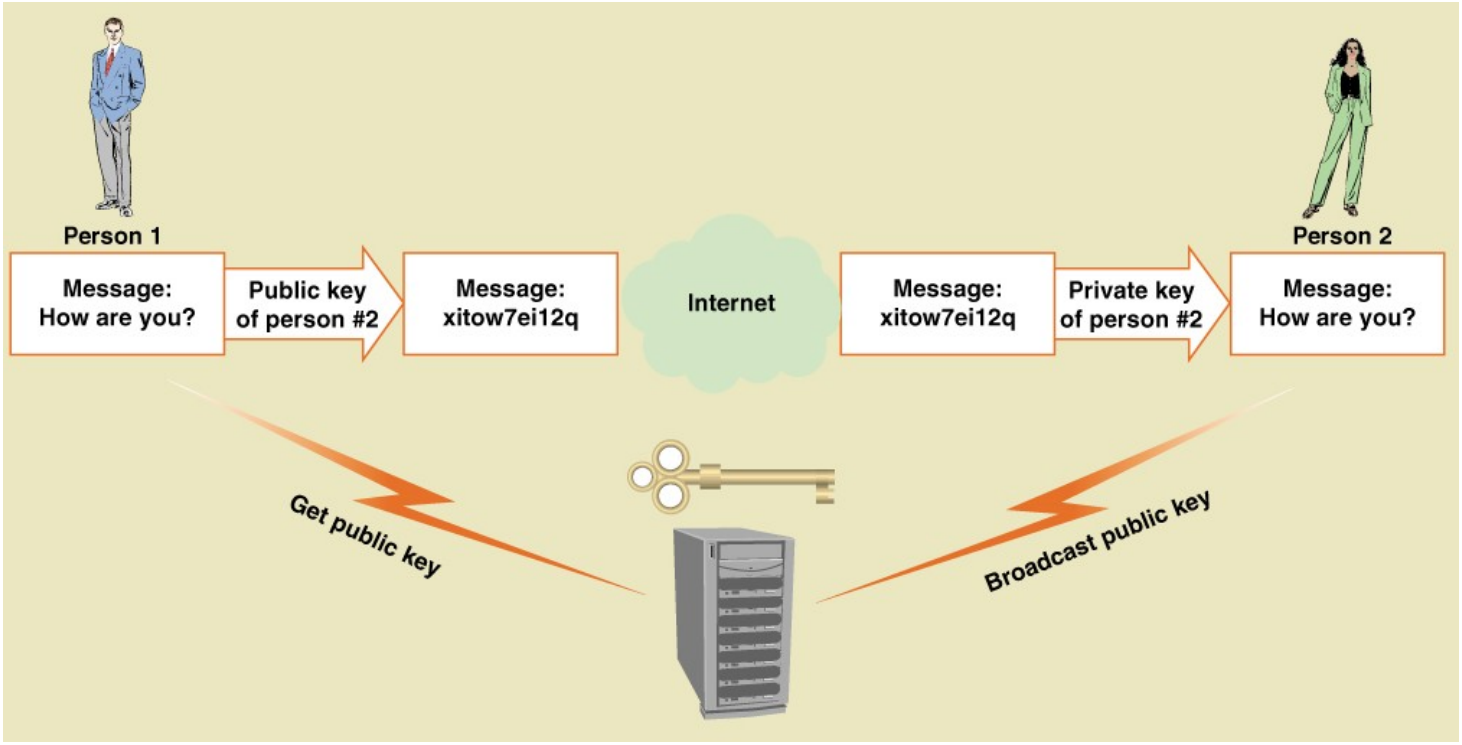
# Symmetric Key Encryption

Encryption method that uses the same key to encrypt and decrypt



# Asymmetric Key Encryption

Encryption method that uses different keys to encrypt and decrypt
AKA Public Key Encryption



# Digital Signatures and Certificates

Digital Signature – technique where a document is encrypted using a private key

Note – implements previous slide, but in reverse
Document is encrypted with private key, but then can only be decrypted with correct public key

Digital Certificate – An organizations name and public that is encrypted and certified by an authorized third party

Certifying Authority – the authorized third party
Widely known and accepted – built into Web browsers

# Database Security

- DBMS features and related measures that comply with the security requirements
- **Authorization management**: Procedures to protect database security and integrity
  - User access management
  - View definition
  - DBMS access control
  - DBMS usage monitoring
    - **Audit log**: Automatically records description of database operations performed by all users

# Active Directory (AD)

Active Directory (AD) is Microsoft's directory and **identity management** service for Windows domain networks. It was introduced in Windows 2000, is included with most MS Windows Server operating systems, and is used by a variety of Microsoft solutions like Exchange Server and SharePoint Server, as well as third-party applications and services.

## AD Domain Services Overview

Active Directory Domain Services is the primary Active Directory service. It is used to authenticate users and to control access to network resources.

A server running AD DS is called a domain controller. Most Windows domain networks have two or more domain controllers; a primary domain controller and one or more backup domain controllers for resiliency.

During login, users authenticate to a domain controller and are granted access to particular resources based on administratively defined policies.

## AD different directory services

- **Active Directory Domain Services (AD DS)** – the core Active Directory service used to manage users and resources.
- **Active Directory Lightweight Directory Services ( AD LDS)** – a low-overhead version of AD DS for directory-enabled applications.
- **Active Directory Certificate Services (AD CS)** – for issuing and managing digital security certificates.
- **Active Directory Federation Services (AD FS)** – for sharing identity and access management information across organizations and enterprises.
- **Active Directory Rights Management Services (AD RMS )** – for information rights management (controlling access permissions to documents, workbooks, presentations, etc.)

# AD Data Structures

Active Directory stores information about network users (names, phone numbers, passwords, etc.) and resources (servers, storage volumes, printers, etc.) in a hierarchical structure consisting of domains, trees, and forests.

- **A domain** is a collection of objects (e.g. users, devices) that share the same Active Directory database. A domain is identified by a DNS name like company.com.

- **A tree** is a collection of one or more domains with a contiguous namespace (they have a common DNS root name like marketing.company.com, engineering.company.com, and sales.company.com).

- **A forest** is a collection of one or more trees that share a common schema, global catalog, and directory configuration—but aren't part of a contiguous namespace. The forest typically serves as the security boundary for an enterprise network.

*** Objects within a domain can be grouped into organizational units (OUs) to simplify administration and policy management.

Administrators can create arbitrary organizational units to mirror functional, geographical, or business structures, and then apply group policies to OUs to simplify administration. OUs also make it easier to delegate control over resources to various administrators.

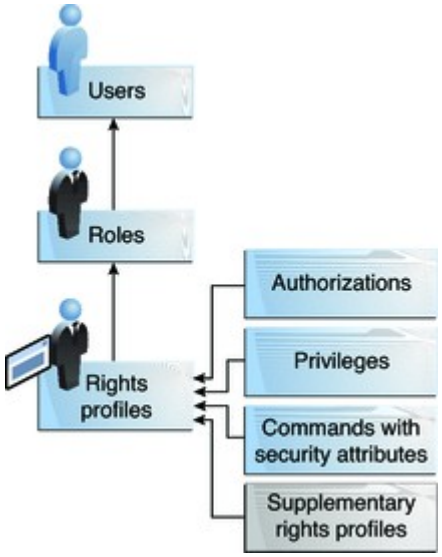
# Role-Based Access Control (RBAC)

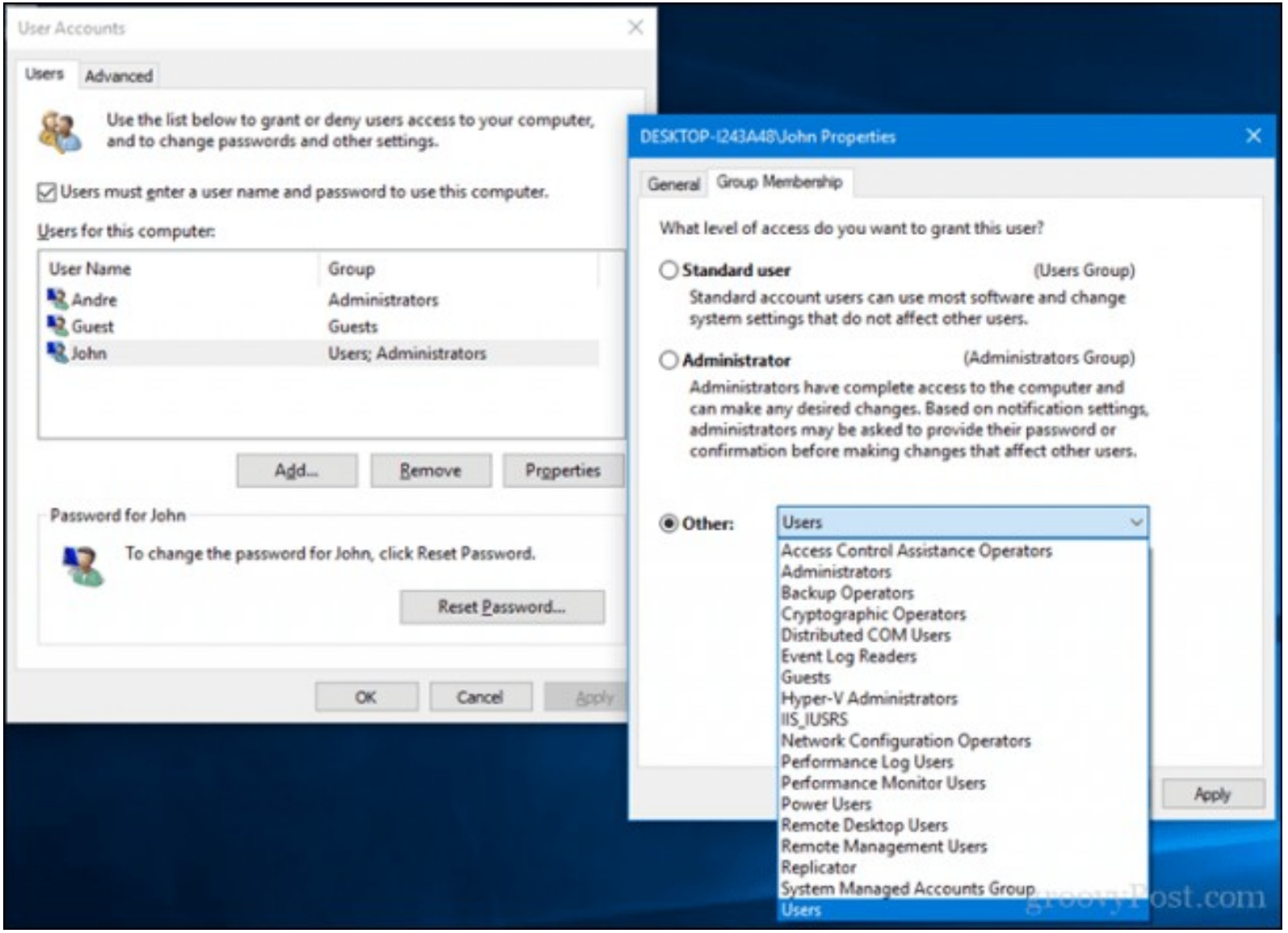
Role-Based Access Control (RBAC) is the modern role-based permissions management model that helps in managing access at both the broad as well as the granular level by aligning the assigned permissions role with the organizational role of the user. **A role is a collection of permissions based on different job functions across an organization.**

RBAC provides a way to group different types of permissions across the enterprise and categorize them in order to effectively assign those permissions based on the job function of the user.

# Groups and Roles – The Difference

One very important point is that there are differences between Groups and Roles. Groups are used to organize users that have the same level of permissions to perform a task. It may or may not be relevant to their job functions, whereas roles are aligned to the job functions and are a group of access privileges to perform a specific job.








.

# Introduction to Oracle CREATE

The CREATE USER statement allows you to create a new database user which you can use to log into the Oracle database.

The basic syntax of the CREATE USER statement is as follows:

```
CREATE USER username IDENTIFIED BY password [DEFAULT
TABLESPACE tablespace] [QUOTA {size | UNLIMITED} ON
tablespace] [PROFILE profile] [PASSWORD EXPIRE]
```

A user profile limits the database resources or password that the user cannot exceed. You can assign a profile to a newly created user. If you skip this clause, Oracle will assign the DEFAULT profile to the user.

PASSWORD EXPIRE

Use the PASSWORD EXPIRE if you want to force the user to change the password for the first time the user logs in to the database.

ACCOUNT {LOCK | UNLOCK}

Use ACCOUNT LOCK if you want to lock the user and disable access. On the other hand, specify ACCOUNT UNLOCK to unlock user and enable access.

To find a list of users with the OPEN status, you query the information from the dba_users.

```
CREATE USER john IDENTIFIED BY abcd1234;
```

To find a list of users with *open status you can query* dba_users

```
SELECT username, default_tablespace,
profile, authentication_type FROM dba_users
WHERE account_status = 'OPEN';
```

| USERNAME | DEFAULT_TABLESPACE | PROFILE | AUTHENTICATION_TYPE |
|---|---|---|---|
| JOHN | USERS | DEFAULT | PASSWORD |
| OT | USERS | DEFAULT | PASSWORD |
| PDBADMIN | USERS | DEFAULT | PASSWORD |
| SYS | SYSTEM | DEFAULT | PASSWORD |
| SYSTEM | SYSTEM | DEFAULT | PASSWORD |

```
SELECT * FROM DBA_USERS ORDER BY created
DESC;
```

| USERNAME | USER_ID | PASSWORD | ACCOUNT_STATUS | LOCK_DATE | EXPIRY_DATE | DEFAULT_TABLESPACE | TEMPORARY_TABLESPACE | LOCAL_TEMP_TABLESPACE | CREATED | PROFILE |
|---|---|---|---|---|---|---|---|---|---|---|
| CRM | 142 | (null) | OPEN | (null) | 26-DEC-19 | USERS | TEMP | TEMP | 29-JUN-19 | CRM_USERS |
| ROD | 140 | (null) | OPEN | (null) | 25-DEC-19 | USERS | TEMP | TEMP | 28-JUN-19 | FIRE_FIGHTER |
| AUDI | 136 | (null) | OPEN | (null) | 24-DEC-19 | USERS | TEMP | TEMP | 27-JUN-19 | DEFAULT |
| SUPER | 129 | (null) | OPEN | (null) | 20-DEC-19 | USERS | TEMP | TEMP | 23-JUN-19 | DEFAULT |
| SCOTT | 125 | (null) | OPEN | (null) | 20-DEC-19 | USERS | TEMP | TEMP | 23-JUN-19 | DEFAULT |
| ALICE | 122 | (null) | OPEN | (null) | 20-DEC-19 | USERS | TEMP | TEMP | 23-JUN-19 | DEFAULT |
| BOB | 120 | (null) | OPEN | (null) | 20-DEC-19 | USERS | TEMP | TEMP | 23-JUN-19 | DEFAULT |
| JACK | 119 | (null) | OPEN | (null) | 19-DEC-19 | USERS | TEMP | TEMP | 22-JUN-19 | DEFAULT |
| JANE | 117 | (null) | OPEN | (null) | 19-DEC-19 | USERS | TEMP | TEMP | 22-JUN-19 | DEFAULT |
| JOHN | 116 | (null) | OPEN | (null) | 19-DEC-19 | USERS | TEMP | TEMP | 22-JUN-19 | DEFAULT |
| OT | 108 | (null) | OPEN | (null) | 02-JUL-19 | USERS | TEMP | TEMP | 03-JAN-19 | DEFAULT |
| HR | 107 | (null) | EXPIRED & LOCKED | 02-JAN-19 | 02-JAN-19 | SYSAUX | TEMP | TEMP | 02-JAN-19 | DEFAULT |

```
SELECT * FROM user_users;
```

| USERNAME | USER_ID | ACCOUNT_STATUS | LOCK_DATE | EXPIRY_DATE | DEFAULT_TABLESPACE | TEMPORARY_TABLESPACE | LOCAL_TEMP_TABLESPACE | CREATED | INITIAL_RSRC_CONSUMER_GROUP | EXTERNAL_NAME |
|---|---|---|---|---|---|---|---|---|---|---|
| OT | 108 | OPEN | (null) | 02-JUL-19 | USERS | TEMP | TEMP | 03-JAN-19 | DEFAULT_CONSUMER_GROUP | (null) |

```
SELECT * FROM all_users ORDER BY created;
```

| USERNAME | USER_ID | CREATED | COMMON | ORACLE_MAINTAINED | INHERITED | DEFAULT_COLLATION | IMPLICIT | ALL_SHARD |
|---|---|---|---|---|---|---|---|---|
| CRM | 142 | 29-JUN-19 | NO | N | NO | USING_NLS_COMP | NO | NO |
| ROD | 140 | 28-JUN-19 | NO | N | NO | USING_NLS_COMP | NO | NO |
| AUDI | 136 | 27-JUN-19 | NO | N | NO | USING_NLS_COMP | NO | NO |
| SUPER | 129 | 23-JUN-19 | NO | N | NO | USING_NLS_COMP | NO | NO |
| SCOTT | 125 | 23-JUN-19 | NO | N | NO | USING_NLS_COMP | NO | NO |
| ALICE | 122 | 23-JUN-19 | NO | N | NO | USING_NLS_COMP | NO | NO |
| BOB | 120 | 23-JUN-19 | NO | N | NO | USING_NLS_COMP | NO | NO |
| JACK | 119 | 22-JUN-19 | NO | N | NO | USING_NLS_COMP | NO | NO |
| JANE | 117 | 22-JUN-19 | NO | N | NO | USING_NLS_COMP | NO | NO |
| JOHN | 116 | 22-JUN-19 | NO | N | NO | USING_NLS_COMP | NO | NO |
| OT | 108 | 03-JAN-19 | NO | N | NO | USING_NLS_COMP | NO | NO |
| HR | 107 | 02-JAN-19 | NO | N | NO | USING_NLS_COMP | NO | NO |
| PDBADMIN | 106 | 02-JAN-19 | NO | N | NO | USING_NLS_COMP | NO | NO |
| DVSYS | 1279990 | 08-MAR-17 | YES | Y | YES | USING_NLS_COMP | NO | NO |
| DVF | 103 | 08-MAR-17 | YES | Y | YES | USING_NLS_COMP | NO | NO |
| LBACSYS | 101 | 08-MAR-17 | YES | Y | YES | USING_NLS_COMP | NO | NO |
| SPATIAL_CSW_ADMIN_USR | 98 | 08-MAR-17 | YES | Y | YES | USING_NLS_COMP | NO | NO |

John will not be able to login to Oracle yet!
To enable the user john to log in, you need to grant the CREATE SESSION system privilege to the user john by using the following statement:

```
GRANT CREATE SESSION TO john;
```

Now, the user john should be able to log in to the database.

After creating a user, you need to decide which actions the user can do in the Oracle database.

**What is a privilege?**

By definition, a privilege is a right to execute an SQL statement or a right to access an object of another user.

Oracle defines two main types of privileges: system privileges and object privileges

## System privileges

System privileges determine what a user can do in the database. They mainly allow a user to add or modify schema objects in the database like creating tables, creating views, and removing tables.

The most important system privileges are:

- CREATE SESSION
- CREATE TABLE
- DROP Table
- CREATE VIEW
- DROP VIEW
- CREATE PROCEDURE
- SYSDBA
- SYSOPER

```
GRANT CREATE SESSION TO john;
GRANT CREATE TABLE TO john;
```

# Object privileges

Object privileges decide how a user can access the data in the database. The object privileges apply to rows in tables or views. Here are some common object privileges:

- INSERT
- UPDATE
- DELETE
- INDEX
- EXECUTE

```
GRANT EXECUTE ON type3 TO user3;
GRANT SELECT ON tab2 TO user3;
```

The following statement shows the privileges of the current user:

```
SELECT * FROM session_privs;

PRIVILEGE
----------------------
CREATE SESSION
CREATE TABLE
```

## Introduction to Oracle CREATE ROLE statement

A role is a group of privileges. Instead of granting individual privileges to users, you can group related privileges into a role and grant this role to users. Roles help manage privileges more efficiently.

To create a new role, you use the CREATE ROLE statement. The basic syntax of the CREATE ROLE statement is as follows:

```
CREATE ROLE role_name
[IDENTIFIED BY password]
[NOT IDENTIFIED]
```

In this syntax:

- First, specify the name of the role that you want to create.
- Second, use IDENTIFIED BY password option to create a local role and indicate that the user, who was granted the role, must provide the password to the database when enabling the role.
- Third, use NOT IDENTIFIED to indicate that the role is authorized by the database and the user, who was granted this role, don't need a password to enable the role. → pass through authorizations

After a role is created, it is empty. To grant privileges to a role, you use the GRANT statement:

```
GRANT  system_privileges | object_privileges} TO
role_name;
```

In addition, you can use the GRANT statement to grant privileges of a role to another role:

```
GRANT role_name TO another_role_name
```

1) Using Oracle CREATE ROLE without a password example

First, create a new role named mdm (master data management) in the sample database:

```
CREATE ROLES mdm;
```

Second, grant object privileges
on customers, contacts, products, product_categories, warehouses, locations, employees tables

to the mdm role:

```
GRANT SELECT, INSERT, UPDATE, DELETE
ON customers
TO mdm;

GRANT SELECT, INSERT, UPDATE, DELETE
ON contacts
TO mdm;

GRANT SELECT, INSERT, UPDATE, DELETE
ON products
TO mdm;

GRANT SELECT, INSERT, UPDATE, DELETE
ON product_categories
TO mdm;

GRANT SELECT, INSERT, UPDATE, DELETE
ON warehouses
TO mdm;

GRANT SELECT, INSERT, UPDATE, DELETE
ON locations
TO mdm;

GRANT SELECT, INSERT, UPDATE, DELETE
ON employees
TO mdm;
```

Third, create a new user named alice and grant the CREATE SESSION privilege to alice:

```
CREATE USER alice IDENTIFIED BY abcd1234;

GRANT CREATE SESSION TO alice;
```

Fourth, log in to the database as alice:

```
Enter user-name: alice@pdborcl
Enter password: abcd1234
```

If she attempt to query data from the ot.employees table:

```
SELECT * FROM ot.employees;
```

Oracle issued the following error:

```
ORA-00942: table or view does
not exist
```

Go back to the first session and grant alice the mdm role:

```
GRANT mdm TO alice;
```

Go to the alice's session and enable role using the SET ROLE statement:

```
SET ROLE mdm;
```

Enable role

To query all roles of the current user, you use the following query:

```
SELECT * FROM session_roles;
```

Since you are in Alice's session

Here is the role of alice:

```
ROLE
------
MDM
```

Now, alice can manipulate data in the master data tables such as customers and employees.

Using Oracle CREATE ROLE to create a role with IDENTIFIED BY password example

First, create a new role named order_entry with the password xyz123:

```
CREATE ROLE order_entry IDENTIFIED BY xyz123;
```

Next, grant object privileges on the orders and order_items tables to the order_entry role:

```
GRANT SELECT, INSERT, UPDATE, DELETE
ON orders
TO order_entry;

GRANT SELECT, INSERT, UPDATE, DELETE
ON order_items
```

Then, grant the order_entry role to the user alice:

```
GRANT order_entry TO alice;
```

After that, log in as alice and enable the order_entry role by using the SET ROLE statement:

```
SET ROLE
    order_entry IDENTIFIED BY xyz123,
    mdm;
```

Enable role

Finally, use the following statement to get the current roles of alice:

```
SELECT * FROM session_roles;
```

Here are the current roles of alice:

```
ROLE
--------------
MDM
ORDER_ENTRY
```

Since you are in Alice's session

# Introduction to Oracle CREATE PROFILE statement

A user profile is a set of limits on the database resources and the user password. Once you assign a profile to a user, then that user cannot exceed the database resource and password limits.

```
CREATE PROFILE profile_name LIMIT
{ resource_parameters | password_parameters};
```

You use the following clauses to set the limit for resource parameters:

- SESSIONS_PER_USER – specify the number of concurrent sessions that a user can have when connecting to the Oracle database.
- CPU_PER_SESSION – specify the CPU time limit for a user session, represented in hundredth of seconds.
- CPU_PER_CALL – specify the CPU time limit for a call such as a parse, execute, or fetch, expressed in hundredths of seconds.
- CONNECT_TIME – specify the total elapsed time limit for a user session, expressed in minutes.
- IDLE_TIME – specify the number of minutes allowed for periods of continuous inactive time during a user session. Note that the long-running queries and other operations will not be subject to this limit.
- LOGICAL_READS_PER_SESSION – specify the allowed number of data blocks read in a user session, including blocks read from both memory and disk.
- LOGICAL_READS_PER_CALL – specify the allowed number of data blocks read for a call to process a SQL statement.
- PRIVATE_SGA – specify the amount of private memory space that a session can allocate in the shared pool of the system's global area (SGA).
- COMPOSITE_LIMIT – specify the total resource cost for a session, expressed in service units. The total service units are calculated as a weighted sum of CPU_PER_SESSION, CONNECT_TIME, LOGICAL_READS_PER_SESSION, and PRIVATE_SGA.

# Password Parameters

You use the following clauses to set the limits for password parameters:

- FAILED_LOGIN_ATTEMPTS – Specify the number of consecutive failed login attempts before the user is locked. The default is 10 times.
- PASSWORD_LIFE_TIME – specify the number of days that a user can use the same password for authentication. The default value is 180 days.
- PASSWORD_REUSE_TIME – specify the number of days before a user can reuse a password.
- PASSWORD_REUSE_MAX – specify the number of password changes required before the current password can be reused. Note that you must set values for
both PASSWORD_REUSE_TIME and PASSWORD_REUSE_MAX parameters make these parameters take effect.
- PASSWORD_LOCK_TIME – specify the number of days that Oracle will lock an account after a specified number of consecutive failed logins. The default is 1 day if you omit this clause.
- PASSWORD_GRACE_TIME – specify the number of days after the grace period starts during which a warning is issued and login is allowed. The default is 7 days when you omit this clause.

Note that to create a new profile, your user needs to have the CREATE PROFILE system privilege.

First, create a profile called CRM_USERS that set the resource limits:

```
CREATE PROFILE CRM_USERS LIMIT
SESSIONS_PER_USER UNLIMITED CPU_PER_SESSION
UNLIMITED CPU_PER_CALL 3000 CONNECT_TIME 15;
```

Then, create a user called CRM:

```
CREATE USER crm IDENTIFIED BY
abcd1234 PROFILE crm_users;
```

Second, create a profile called erp_USERS with password limits:

```
CREATE PROFILE erp_users LIMIT
FAILED_LOGIN_ATTEMPTS 5
PASSWORD LIFE TIME 90;
```

Then, create a user named sap and set its profile to erp_users:

```
CREATE USER sap IDENTIFIED BY
abcd1234 PROFILE erp_users;
```

## Common Oracle DBA Tasks

As an Oracle DBA, you can expect to be involved in the following tasks:

- Installing Oracle software
- Creating Oracle databases
- Performing upgrades of the database and software to new release levels
- Starting and shutting down the database instance
- Managing the storage structures of the database
- Managing users and security
- Managing database objects, such as tables, indexes, and views
- Backing up the database and performing recovery operations when necessary
- Monitoring the state of the database and taking preventive or corrective action as required
- Monitoring and tuning database performance
- Diagnosing and reporting critical errors to Oracle Support Services

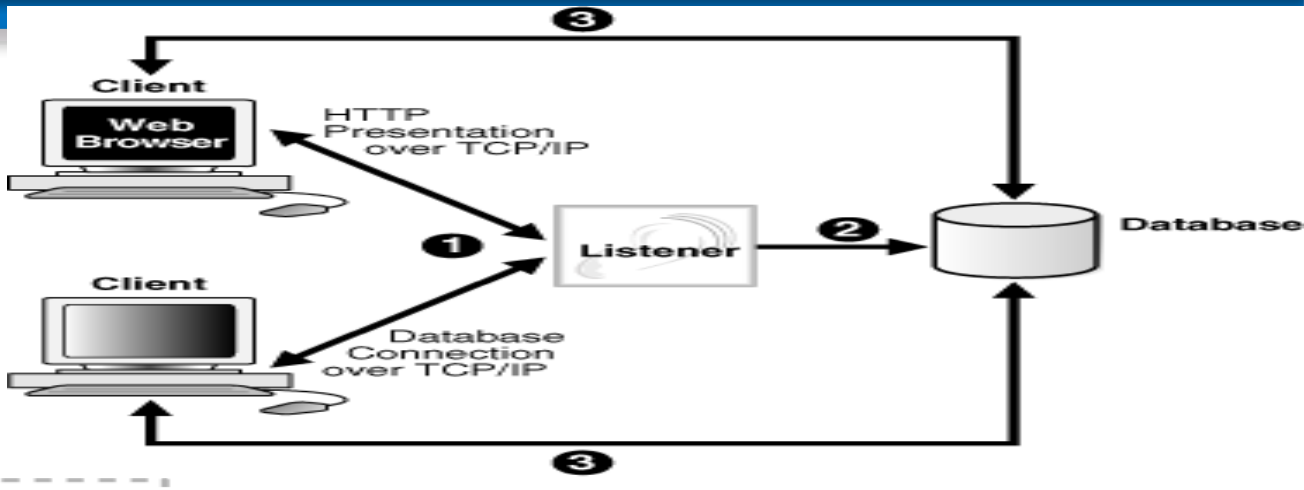
# Database Administration Tools

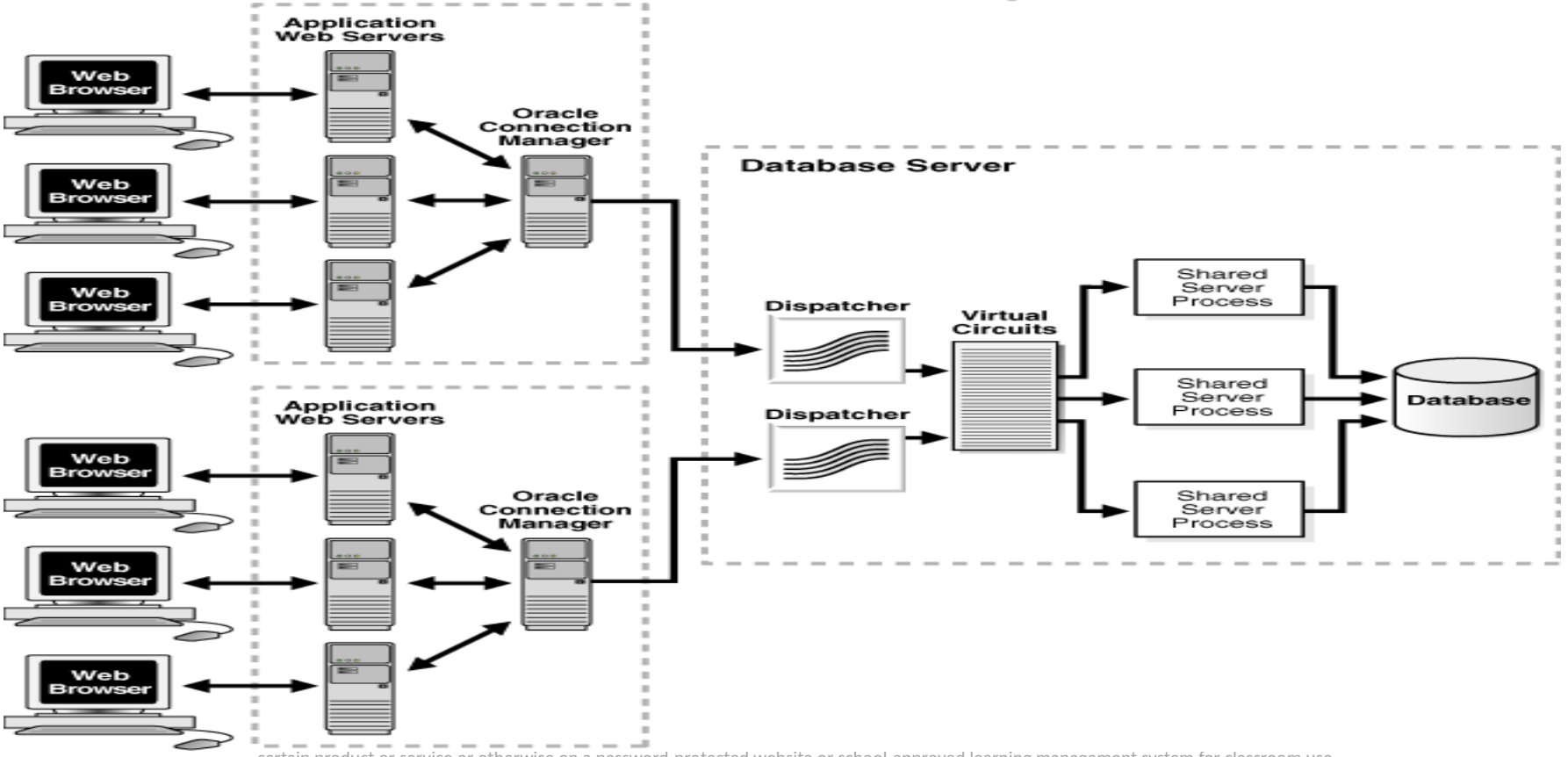
- Database monitoring
- Database load testing
- Database performance tuning
- SQL code optimization
- Database bottleneck identification and remediation
- Database modeling and design
- Database data extraction, transformation, and loading

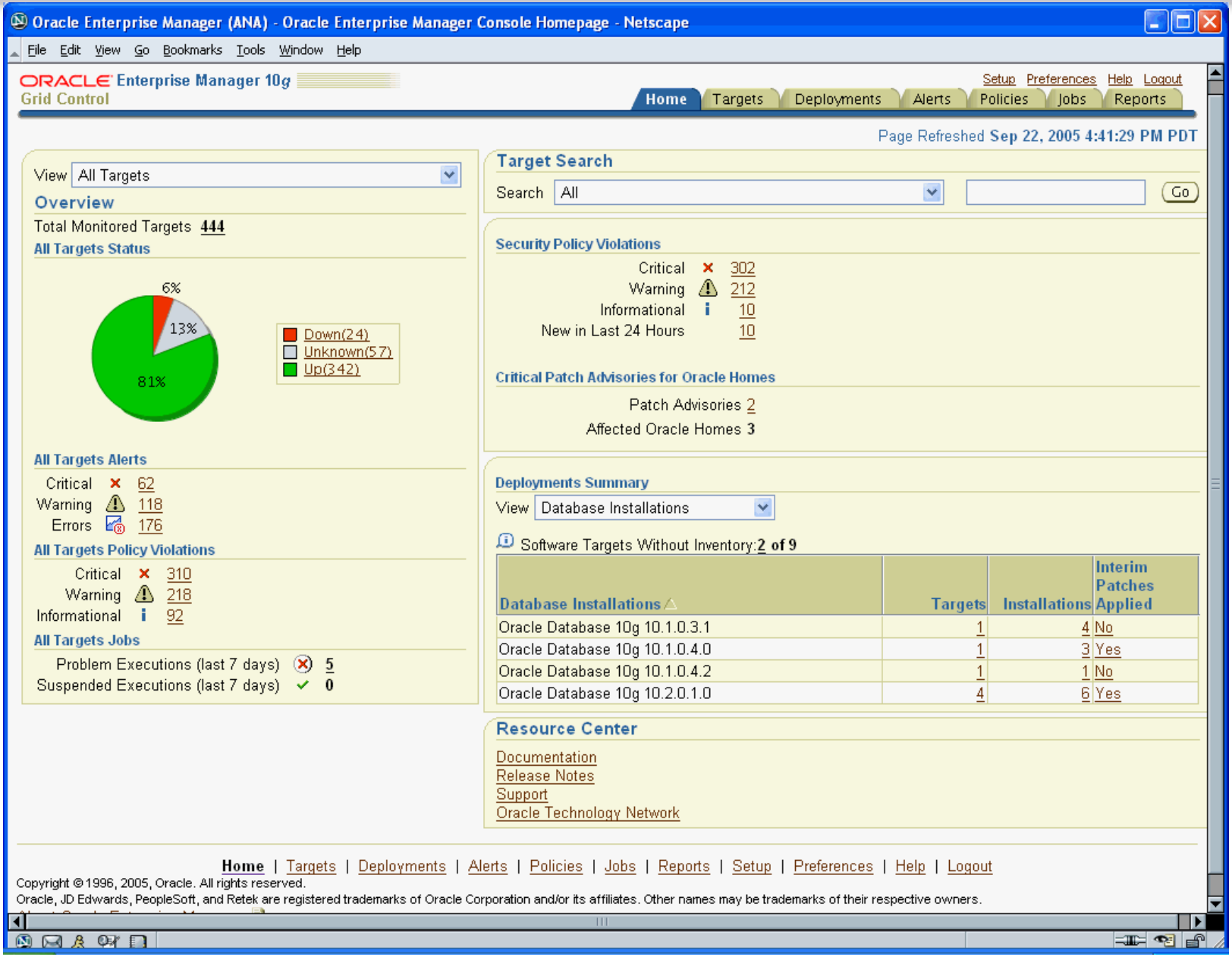
## Tools for Administering the Database

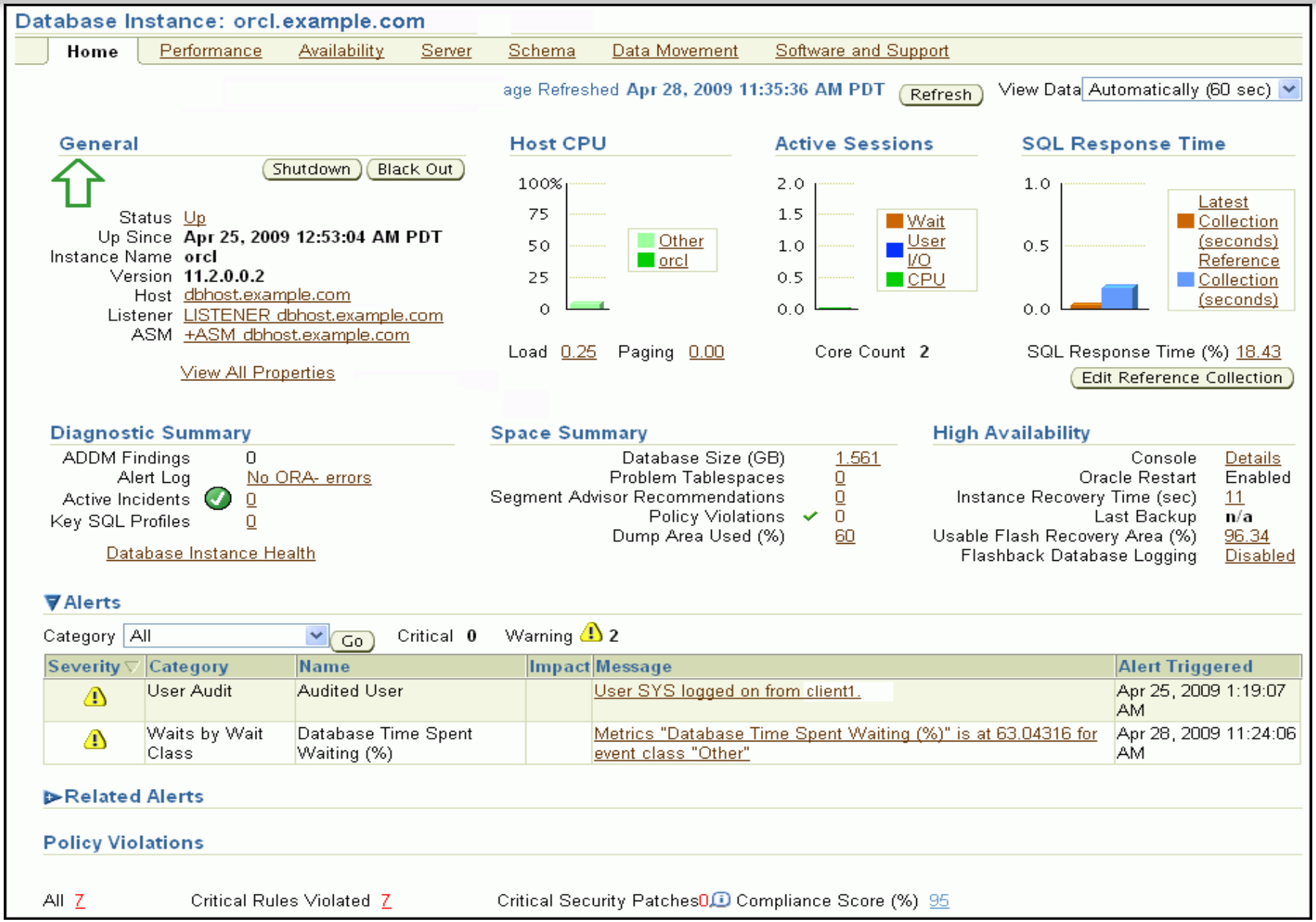
- Oracle Universal Installer
  - Oracle Universal Installer (OUI) is a utility that installs your Oracle software and options. It can automatically start Oracle Database Configuration Assistant to install a database.

- Oracle Database Configuration Assistant
  - Oracle Database Configuration Assistant (DBCA) is a utility that creates a database from templates that are supplied by Oracle, or you can create your own. It enables you to copy a preconfigured seed database, thus saving the time and effort of generating and customizing a new database.
- Database Upgrade Assistant
  - The Database Upgrade Assistant is a tool that guides you through the upgrade of your existing database to a new Oracle Database release.
- Net Configuration Assistant
  - Net Configuration Assistant (NETCA) is a utility that enables you to configure listeners and naming methods, which are critical components of the Oracle Database network.
  - The Oracle listener is **a service that runs on the database host and receives requests from Oracle clients**.
- Oracle Enterprise Manager Database Control



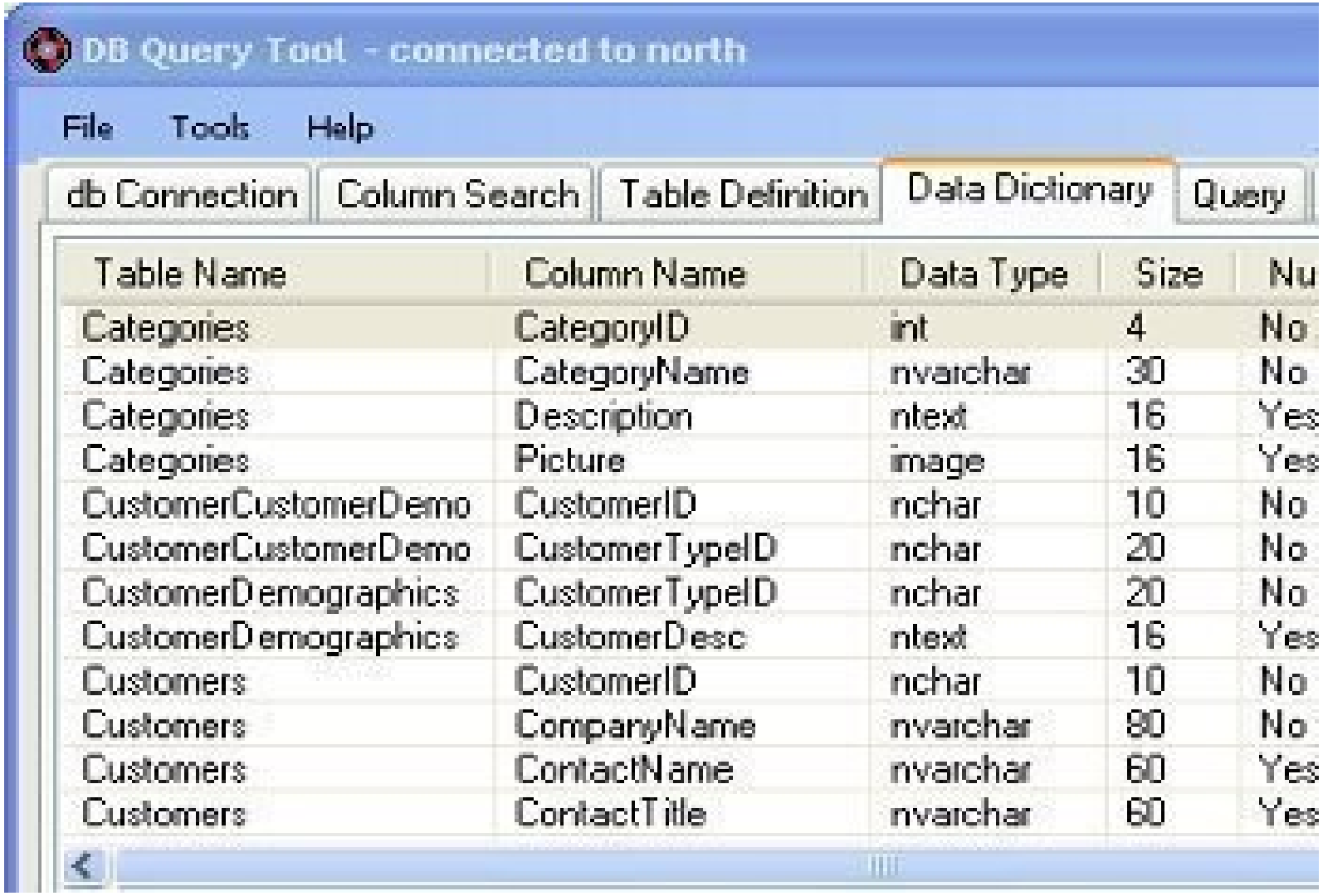




# Data Dictionary

- Two main types:
  - Integrated - Included with the DBMS
  - Standalone - Third-party systems
- **Active data dictionary**: 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 the database

Sample Data Dictionary – also Look at page 92 of LSC 654 Textbook




| Table Name | Column Name | Data Type | Size | Nu |
|---|---|---|---|---|
| Categories | CategoryID | int | 4 | No |
| Categories | CategoryName | nvarchar | 30 | No |
| Categories | Description | ntext | 16 | Yes |
| Categories | Picture | image | 16 | Yes |
| CustomerCustomerDemo | CustomerID | nchar | 10 | No |
| CustomerCustomerDemo | CustomerTypeID | nchar | 20 | No |
| CustomerDemographics | CustomerTypeID | nchar | 20 | No |
| CustomerDemographics | CustomerDesc | ntext | 16 | Yes |
| Customers | CustomerID | nchar | 10 | No |
| Customers | CompanyName | nvarchar | 80 | No |
| Customers | ContactName | nvarchar | 60 | Yes |
| Customers | ContactTitle | nvarchar | 60 | Yes |

# Data Dictionary

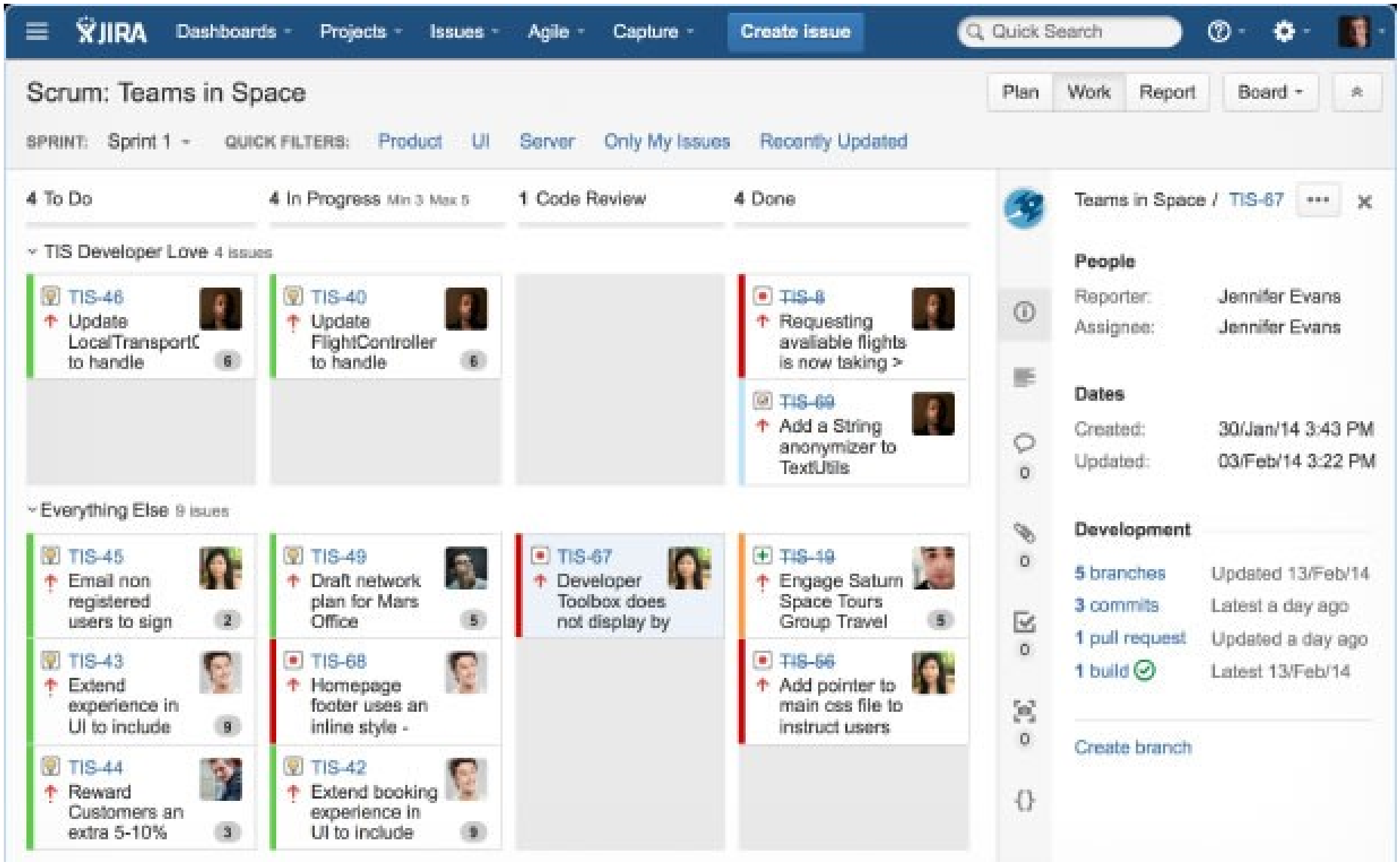
- Key element of information resource management
  - Can be described as the **information resource dictionary**
- Metadata is the basis for monitoring database use and for assigning access rights to users
- DBA uses data dictionary to support data analysis and design

# Computer-Aided Systems Engineering (CASE) Tools

- Automated framework for the Systems Development Life Cycle (SDLC)
- Use structured methodologies and powerful graphical interfaces
- Classified according to extent of support provided:
  - **Front-end CASE tools**: Provide support for the planning, analysis, and design phases
  - **Back-end CASE tools**: Provide support for the coding and implementation phases

# Components of a CASE Tool

- Graphics
- Screen painters and report generators
- Integrated repository
- Analysis segment
- Program documentation generator






.

# Developing a Data Administration Strategy

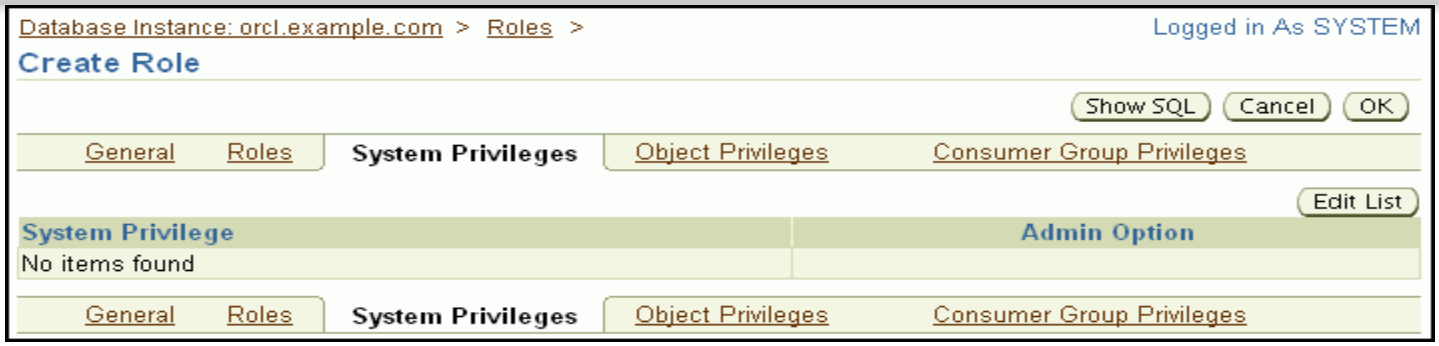
- **Information engineering (IE)**: Translates strategic goals into data and applications
- **Information systems architecture (ISA)**: Helps plan, develop, and control future information systems
- Critical success factors:
  - Management commitment and defined standards
  - Thorough analysis of the company situation
  - End-user involvement
  - Training and a small pilot project

# DBA's Role in the Cloud

- Significant impact on role of DBAs
- Tasks split between internal DBA and cloud service provider
- Cloud service partner company provides:
  - DBMS installation and updates
  - Server/network management
  - Backup and recovery operations

# Oracle Database Administration Tools

- Ensure the RDBMS starts automatically
- Create tablespaces and datafiles
  - **Tablespace**: Logical storage space
  - **Datafile**: Physically stores the database's data
- Manage users and establish security
  - **User**: Allows a given person to log on to the database
  - **Role**: Authorizes a user to connect to the database and use its system resources
  - **Profile**: Controls how much of the database resource a given user can access




# Types of Tablespace

## SYSTEM

- Stores the data dictionary data

## USERS

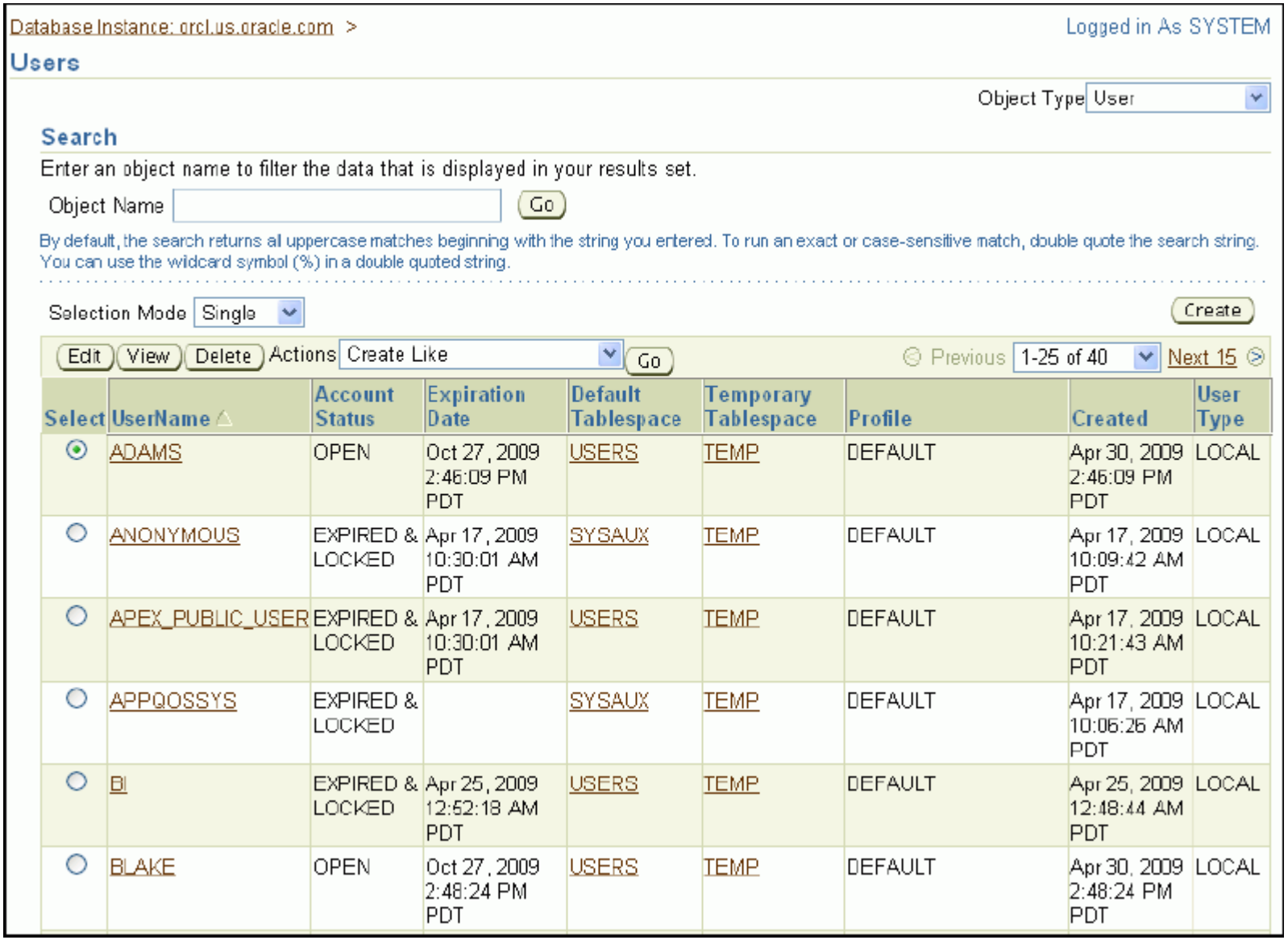
- Stores the table data created by the end users

## TEMP

- Stores the temporary tables and indexes created during the execution of SQL statements

## UNDOTBS1

- Stores database transaction recovery information



# Customize Database Initialization Parameters

- Fine-tuning a database is an important task that usually requires modification of parameters
- Initialization parameters reserve resources used by the database at run time
- After modifying parameters database restart may be required

# Database Initialization Parameter

- CLUSTER_DATABASE
- COMPATIBLE
- CONTROL_FILES
- DB_BLOCK_SIZE
- DB_CREATE_FILE_DEST
- DB_CREATE_ONLINE_LOG_DEST_n
- DB_DOMAIN
- DB_NAME
- DB_RECOVERY_FILE_DEST
- DB_RECOVERY_FILE_DEST_SIZE
- DB_UNIQUE_NAME
- INSTANCE_NUMBER
- LDAP_DIRECTORY_SYSAUTH
- LOG_ARCHIVE_DEST_n
- LOG_ARCHIVE_DEST_STATE_n
- NLS_DATE_LANGUAGE
- NLS_TERRITORY
- OPEN_CURSORS
- PGA_AGGREGATE_TARGET
- PROCESSES
- REMOTE_LISTENER
- REMOTE_LOGIN_PASSWORDFILE
- SESSIONS
- SGA_TARGET
- SHARED_SERVERS
- STAR_TRANSFORMATION_ENABLED
- UNDO_TABLESPACE

Data Dictionary:

https://www.youtube.com/watch?v=AeVJy-ow2b0

Table Alias:

http://dba.fyicenter.com/faq/oracle/Define-and-Use-Table-Alias-Names.html