CO226: Database Systems

Introduction

Manjula Sandirigama
Senior Lecturer, Department of Computer Engineering
University of Peradeniya
m_sandirigama@pdn.ac.lk

13th June 2017

Staff

Lecturers

- Manjula Sandirigama m_sandirigama@pdn.ac.lk
- Sampath Deegalla dsdeegalla@pdn.ac.lk

Instructors

- Gayan Meerigama (IIC)
- Fathima Fasna
- Nimali Kularathne

Staff

Lecturers

- Manjula Sandirigama m_sandirigama@pdn.ac.lk
- Sampath Deegalla dsdeegalla@pdn.ac.lk

Instructors

- Gayan Meerigama (IIC)
- Fathima Fasna
- Nimali Kularathne

Staff

Lecturers

- Manjula Sandirigama m_sandirigama@pdn.ac.lk
- Sampath Deegalla dsdeegalla@pdn.ac.lk

Instructors

- Gayan Meerigama (IIC)
- Fathima Fasna
- Nimali Kularathne

References

- Ramez Elmasri and Shamkant B. Navathe, Fundamentals of Database Systems, 6th Edition, 2010
- Raghu Ramakrishnan and Johannes Gehrke, Database Management Systems, 3rd Edition, McGraw-Hill, 2004

References

- Ramez Elmasri and Shamkant B. Navathe, Fundamentals of Database Systems, 6th Edition, 2010
- Raghu Ramakrishnan and Johannes Gehrke, Database Management Systems, 3rd Edition, McGraw-Hill, 2004

References

- Ramez Elmasri and Shamkant B. Navathe, Fundamentals of Database Systems, 6th Edition, 2010
- Raghu Ramakrishnan and Johannes Gehrke, Database Management Systems, 3rd Edition, McGraw-Hill, 2004

Assessment Details

Assessment	Percentage Marks	
Continuous Assessment	*40*	
Tutorials		10
Practicals		15
Assignments		15
Written Examinations	*60*	
Mid-semester		20
End-semester		40

Topics

- Introduction
- Data modeling
- RDBMS Concepts
- Database Query Languages
- Transaction Processing
- Distributed Databases
- Physical Database Design

Topics

- Introduction
- Data modeling
- RDBMS Concepts
- Database Query Languages
- Transaction Processing
- Distributed Databases
- Physical Database Design

Topics

- Introduction
- Data modeling
- RDBMS Concepts
- Database Query Languages
- Transaction Processing
- Distributed Databases
- Physical Database Design

Topics

- Introduction
- Data modeling
- RDBMS Concepts
- Database Query Languages
- Transaction Processing
- Distributed Databases
- Physical Database Design

Topics

- Introduction
- Data modeling
- RDBMS Concepts
- Database Query Languages
- Transaction Processing
- Distributed Databases
- Physical Database Design

Topics

- Introduction
- Data modeling
- RDBMS Concepts
- Database Query Languages
- Transaction Processing
- Distributed Databases
- Physical Database Design

Topics

- Introduction
- Data modeling
- RDBMS Concepts
- Database Query Languages
- Transaction Processing
- Distributed Databases
- Physical Database Design

Topics

- Introduction
- Data modeling
- RDBMS Concepts
- Database Query Languages
- Transaction Processing
- Distributed Databases
- Physical Database Design

Basic Definitions

Data

Refers to known facts that can be recorded and have an implicit meaning.

Database

A collection of related data with the following properties

- Intended application and users, i.e., specific purpose
- Represents some aspects of the real world
- Logically organized

Basic Definitions

Data

Refers to known facts that can be recorded and have an implicit meaning.

Database

A collection of related data with the following properties

- Intended application and users, i.e., specific purpose
- Represents some aspects of the real world
- Logically organized

Basic Definitions

Database Management System (DBMS)

- A software package/ system to facilitate the creation and maintenance of a computerized database.
- The operations supported by a DBMS include:
  - *Defining* the database specify types of data and relationships (files/records/fields/physical & logical links)
  - *Constructing* database: the process of storing data
  - *Manipulating* the database i.e. query, update (insert/delete/modify), generate reports

Database System

The DBMS software together with the data itself. Sometimes, the applications are also included.

Basic Definitions

Database Management System (DBMS)

- A software package/ system to facilitate the creation and maintenance of a computerized database.
- The operations supported by a DBMS include:
  - *Defining* the database specify types of data and relationships (files/records/fields/physical & logical links)
  - *Constructing* database: the process of storing data
  - *Manipulating* the database i.e. query, update (insert/delete/modify), generate reports

Database System

The DBMS software together with the data itself. Sometimes, the applications are also included.

Typical DBMS Functionality

- Define a database
  - in terms of data types, structures and constraints
- Construct or Load the Database on a secondary storage medium
- Manipulating the database
  - querying, generating reports, insertions, deletions and modifications to its content
- Concurrent Processing and Sharing by a set of users and programs - yet, keeping all data valid and consistent

Typical DBMS Functionality

- Define a database
  - in terms of data types, structures and constraints
- Construct or Load the Database on a secondary storage medium
- Manipulating the database
  - querying, generating reports, insertions, deletions and modifications to its content
- Concurrent Processing and Sharing by a set of users and programs - yet, keeping all data valid and consistent

Typical DBMS Functionality

- Define a database
  - in terms of data types, structures and constraints
- Construct or Load the Database on a secondary storage medium
- Manipulating the database
  - querying, generating reports, insertions, deletions and modifications to its content
- Concurrent Processing and Sharing by a set of users and programs - yet, keeping all data valid and consistent

Typical DBMS Functionality

- Define a database
  - in terms of data types, structures and constraints
- Construct or Load the Database on a secondary storage medium
- Manipulating the database
  - querying, generating reports, insertions, deletions and modifications to its content
- Concurrent Processing and Sharing by a set of users and programs - yet, keeping all data valid and consistent

Typical DBMS Functionality

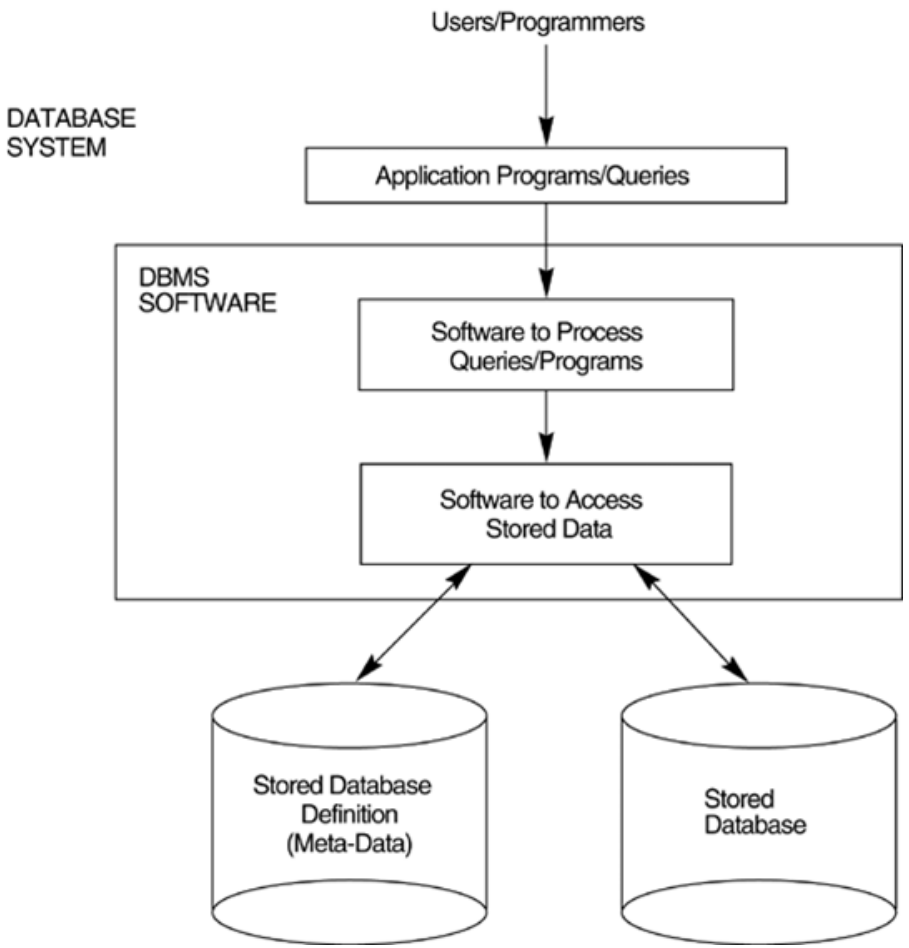
- Define a database
  - in terms of data types, structures and constraints
- Construct or Load the Database on a secondary storage medium
- Manipulating the database
  - querying, generating reports, insertions, deletions and modifications to its content
- Concurrent Processing and Sharing by a set of users and programs - yet, keeping all data valid and consistent

Typical DBMS Functionality

- Other features:
  - Protection or Security measures to prevent unauthorized access
  - Active processing to take internal actions on data
  - Presentation and Visualization of data

A simplified database system environment

Back



Example: UNIVERSITY database

STUDENT	Name	StudentNumber	Class	Major
	Smith	17	1	CS
	Brown	8	2	CS

COURSE	CourseName	CourseNumber	CreditHours	Department
	Intro to Computer Science	CS1310	4	CS
	Data Structures	CS3320	4	CS
	Discrete Mathematics	MATH2410	3	MATH
	Database	CS3380	3	CS

Back

Example: UNIVERSITY database

SECTION	SectionIdentifier	CourseNumber	Semester	Year	Instructor
	85	MATH2410	Fall	98	King
	92	CS1310	Fall	98	Anderson
	102	CS3320	Spring	99	Knuth
	112	MATH2410	Fall	99	Chang
	119	CS1310	Fall	99	Anderson
	135	CS3380	Fall	99	Stone

data elements

Name, StudentNumber, Class, Major

data type

STUDENT (Name : string, StudentNumber : integer) Figure 2
GRADE_REPORT (Grade: single character) Figure 4

Example: UNIVERSITY database

GRADE_REPORT	StudentNumber	SectionIdentifier	Grade
	17	112	B
	17	119	C
	8	85	A
	8	92	A
	8	102	B
	8	135	A

PREREQUISITE	CourseNumber	PrerequisiteNumber
	CS3380	CS3320
	CS3380	MATH2410
	CS3320	CS1310

Back

Database Approach vs. Traditional File Processing

- Traditional File Processing System
  - each group of users has its own data files and application programs for that specific data file

Example

- Payroll Dept.

  `Staff Salary (Staff Number, First Name, Last Name, Address, Sex,Date of Birth, Salary, PIN, Department)`

- Personnel Dept.

  `Staff (Staff Number, First Name, Last Name, Address, Telephone Number, Position, Sex, Date of Birth, Salary, PIN, Department)`

- Both the Payroll department and the Personnel Department stored similar data items.

Problems With Traditional File Processing

- Redundant data
- Wasted storage space
- Inconsistent data
- Difficult to add/modify applications
- File structure is part of the code

Data Models

Data Model

A set of concepts to describe the structure of a database, and certain constraints that the database should obey.

Data Model Operations

Operations for specifying database retrievals and updates by referring to the concepts of the data model. Operations on the data model may include basic operations and user-defined operations.

Data Models

Data Model

A set of concepts to describe the structure of a database, and certain constraints that the database should obey.

Data Model Operations

Operations for specifying database retrievals and updates by referring to the concepts of the data model. Operations on the data model may include basic operations and user-defined operations.

Data Models

Data Model

A set of concepts to describe the structure of a database, and certain constraints that the database should obey.

Data Model Operations

Operations for specifying database retrievals and updates by referring to the concepts of the data model. Operations on the data model may include basic operations and user-defined operations.

Categories of data models

Conceptual (high-level, semantic) data models

Provide concepts that are close to the way many users perceive data. (Also called entity-based or object-based data models.)

Physical (low-level, internal) data models

Provide concepts that describe details of how data is stored in the computer.

Implementation (representational) data models

Provide concepts that fall between the above two, balancing user views with some computer storage details.

Categories of data models

Conceptual (high-level, semantic) data models

Provide concepts that are close to the way many users perceive data. (Also called entity-based or object-based data models.)

Physical (low-level, internal) data models

Provide concepts that describe details of how data is stored in the computer.

Implementation (representational) data models

Provide concepts that fall between the above two, balancing user views with some computer storage details.

Categories of data models

Conceptual (high-level, semantic) data models

Provide concepts that are close to the way many users perceive data. (Also called entity-based or object-based data models.)

Physical (low-level, internal) data models

Provide concepts that describe details of how data is stored in the computer.

Implementation (representational) data models

Provide concepts that fall between the above two, balancing user views with some computer storage details.

Categories of data models

Conceptual (high-level, semantic) data models

Provide concepts that are close to the way many users perceive data. (Also called entity-based or object-based data models.)

Physical (low-level, internal) data models

Provide concepts that describe details of how data is stored in the computer.

Implementation (representational) data models

Provide concepts that fall between the above two, balancing user views with some computer storage details.

Schemas versus Instances

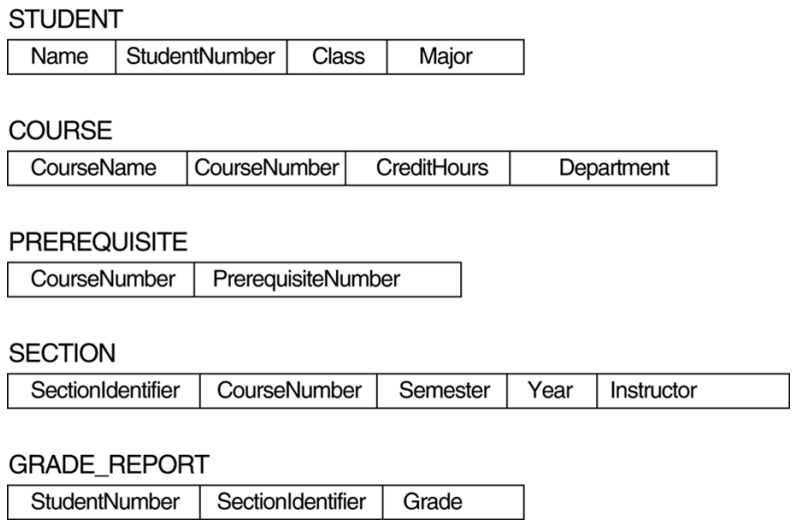
- Database Schema : The description of a database. Includes descriptions of the database structure and the constraints that should hold on the database.
- Schema Diagram : A diagrammatic display of (some aspects of) a database schema.
- Schema Construct : A component of the schema or an object within the schema, e.g., STUDENT, COURSE.
- Database Instance : The actual data stored in a database at a particular moment in time. Also called database state (or occurrence).

Schemas versus Instances

- Database Schema : The description of a database. Includes descriptions of the database structure and the constraints that should hold on the database.
- Schema Diagram : A diagrammatic display of (some aspects of) a database schema.
- Schema Construct : A component of the schema or an object within the schema, e.g., STUDENT, COURSE.
- Database Instance : The actual data stored in a database at a particular moment in time. Also called database state (or occurrence).

Schemas versus Instances

- Database Schema : The description of a database. Includes descriptions of the database structure and the constraints that should hold on the database.
- Schema Diagram : A diagrammatic display of (some aspects of) a database schema.



- Schema Construct : A component of the schema or an object within the schema, e.g., STUDENT, COURSE.
- Database Instance : The actual data stored in a database at a

Schemas versus Instances

- Database Schema : The description of a database. Includes descriptions of the database structure and the constraints that should hold on the database.
- Schema Diagram : A diagrammatic display of (some aspects of) a database schema.
- Schema Construct : A component of the schema or an object within the schema, e.g., STUDENT, COURSE.
- Database Instance : The actual data stored in a database at a particular moment in time. Also called database state (or occurrence).

Schemas versus Instances

- Database Schema : The description of a database. Includes descriptions of the database structure and the constraints that should hold on the database.
- Schema Diagram : A diagrammatic display of (some aspects of) a database schema.
- Schema Construct : A component of the schema or an object within the schema, e.g., STUDENT, COURSE.
- Database Instance : The actual data stored in a database at a particular moment in time. Also called database state (or occurrence).

Database Schema Vs. Database State

- Database State : Refers to the content of a database at a moment in time.
- Initial Database State : Refers to the database when it is loaded.
- Valid State : A state that satisfies the structure and constraints of the database.
- Distinction
  - The database schema changes very infrequently. The database state changes every time the database is updated.

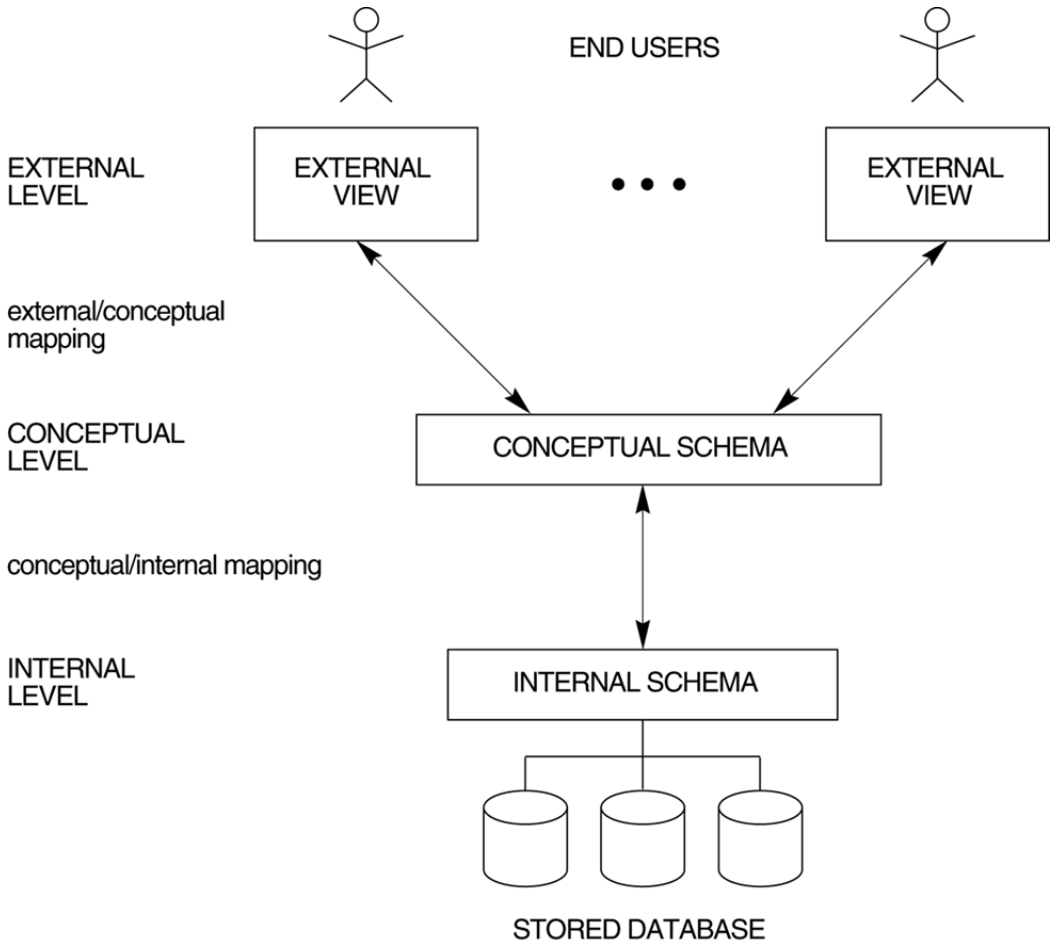
Three-Schema Architecture

- Proposed to support DBMS characteristics of:
  - Program-data independence.
  - Support of multiple views of the data.

Three-Schema Architecture

- Defines DBMS schemas at three levels:
  - Internal schema at the internal level to describe physical storage structures and access paths. Typically uses a physical data model.
  - Conceptual schema at the conceptual level to describe the structure and constraints for the whole database for a community of users. Uses a conceptual or an implementation data model.
  - External schemas at the external level to describe the various user views. Usually uses the same data model as the conceptual level.

The three-schema architecture



Three-Schema Architecture

Mappings among schema levels are needed to transform requests and data. Programs refer to an external schema, and are mapped by the DBMS to the internal schema for execution.

Data Independence

Logical Data Independence

The capacity to change the conceptual schema without having to change the external schemas and their application programs.

Physical Data Independence

The capacity to change the internal schema without having to change the conceptual schema.

Data Independence

Logical Data Independence

The capacity to change the conceptual schema without having to change the external schemas and their application programs.

Physical Data Independence

The capacity to change the internal schema without having to change the conceptual schema.

Data Independence

Logical Data Independence

The capacity to change the conceptual schema without having to change the external schemas and their application programs.

Physical Data Independence

The capacity to change the internal schema without having to change the conceptual schema.

Data Independence

When a schema at a lower level is changed, only the mappings between this schema and higher-level schemas need to be changed in a DBMS that fully supports data independence. The higher-level schemas themselves are unchanged. Hence, the application programs need not be changed since they refer to the external schemas.

DBMS Languages

- Data Definition Language (DDL): Used to specify the conceptual schema of a database. In many DBMSs, the DDL is also used to define internal and external schemas (views).
- Separate storage definition language (SDL): Used to specify the internal schma.
- View definition language (VDL): Used to define internal and external schemas.

DBMS Languages

- Data Manipulation Language (DML): Used to specify database retrievals and updates.
  - DML commands (data sublanguage) can be embedded in a general-purpose programming language (host language), such as COBOL, C or an Assembly Language.
  - Alternatively, stand-alone DML commands can be applied directly (query language).

DBMS Languages

- High Level or Non-procedural Languages : e.g., SQL, are set-oriented and specify what data to retrieve than how to retrieve. Also called declarative languages.
- Low Level or Procedural Languages : record-at-a-time; they specify how to retrieve data and include constructs such as looping.

DBMS Interfaces

- Menu-based Interfaces
- Form-based Interfaces
- Graphical User Interfaces
- Natural Language Interfaces

DBMS Interfaces

- Menu-based Interfaces
- Form-based Interfaces
- Graphical User Interfaces
- Natural Language Interfaces

DBMS Interfaces

- Menu-based Interfaces
- Form-based Interfaces
- Graphical User Interfaces
- Natural Language Interfaces

DBMS Interfaces

- Menu-based Interfaces
- Form-based Interfaces
- Graphical User Interfaces
- Natural Language Interfaces

DBMS Interfaces

- Menu-based Interfaces
- Form-based Interfaces
- Graphical User Interfaces
- Natural Language Interfaces

DBMS Interfaces

- Speech as Input and Output
- Parametric interfaces (e.g., bank tellers) using function keys.
- Interfaces for the DBA:
  - Creating accounts, granting authorizations
  - Setting system parameters
  - Changing schemas or access path

DBMS Interfaces

- Speech as Input and Output
- Parametric interfaces (e.g., bank tellers) using function keys.
- Interfaces for the DBA:
  - Creating accounts, granting authorizations
  - Setting system parameters
  - Changing schemas or access path

DBMS Interfaces

- Speech as Input and Output
- Parametric interfaces (e.g., bank tellers) using function keys.
- Interfaces for the DBA:
  - Creating accounts, granting authorizations
  - Setting system parameters
  - Changing schemas or access path

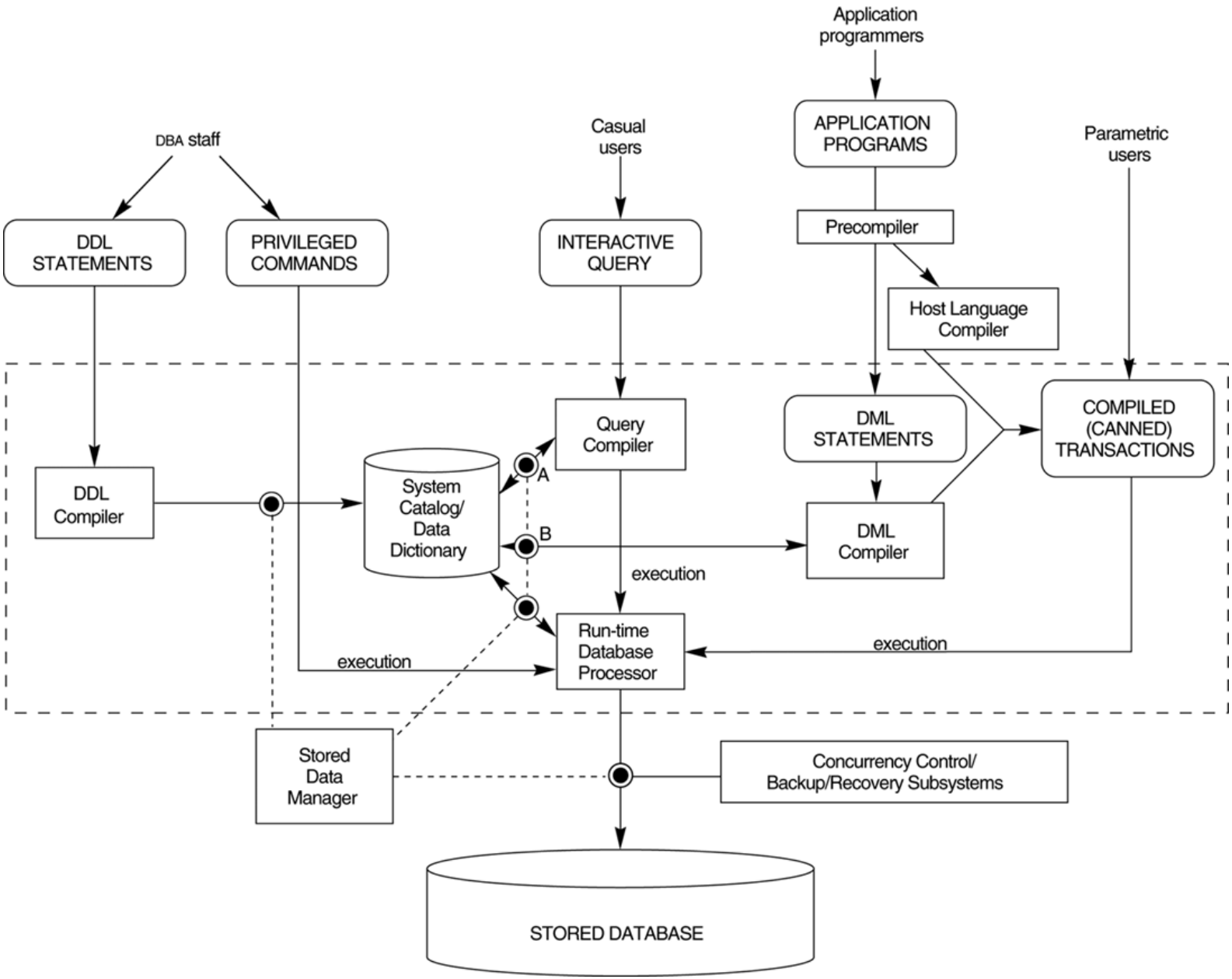
DBMS Interfaces

- Speech as Input and Output
- Parametric interfaces (e.g., bank tellers) using function keys.
- Interfaces for the DBA:
  - Creating accounts, granting authorizations
  - Setting system parameters
  - Changing schemas or access path

DBMS Components

- The DBMS is a complex software system
  - It can be partitioned into several components, each providing a given services

Component modules of a DBMS and their interactions



Database System Utilities

- To perform certain functions such as:
  - Loading data stored in files into a database. Includes data conversion tools.
  - Backing up the database periodically on tape.
  - Reorganizing database file structures.
  - Report generation utilities.
  - Performance monitoring utilities.
  - Other functions, such as sorting, user monitoring, data compression, etc.

Other Tools

- Data dictionary / data repository
  - Used to store schema descriptions and other information such as design decisions, application program descriptions, user information, usage standards, etc.
- Application Development Environments and CASE (computer-aided software engineering) tools
  - Examples : Power builder (Sybase), JBuilder (Borland)

Classification of DBMSs

- Based on the data model used
  - Traditional : Relational, Network, Hierarchical.
  - Emerging : Object-oriented, Object-relational.
- Based on number of users supported by the system
  - Single-user (typically used with micro-computers) vs. multi-user (most DBMSs).
- Based on the number of sites over which the database is distributed
  - Centralized (uses a single computer with one database) vs. distributed (uses multiple computers, multiple databases)
- Based on cost
  - Open source (free) vs. Commercial

Classification of DBMSs

- Based on the data model used
  - Traditional : Relational, Network, Hierarchical.
  - Emerging : Object-oriented, Object-relational.
- Based on number of users supported by the system
  - Single-user (typically used with micro-computers) vs. multi-user (most DBMSs).
- Based on the number of sites over which the database is distributed
  - Centralized (uses a single computer with one database) vs. distributed (uses multiple computers, multiple databases)
- Based on cost
  - Open source (free) vs. Commercial

Classification of DBMSs

- Based on the data model used
  - Traditional : Relational, Network, Hierarchical.
  - Emerging : Object-oriented, Object-relational.
- Based on number of users supported by the system
  - Single-user (typically used with micro-computers) vs. multi-user (most DBMSs).
- Based on the number of sites over which the database is distributed
  - Centralized (uses a single computer with one database) vs. distributed (uses multiple computers, multiple databases)
- Based on cost
  - Open source (free) vs. Commercial

Classification of DBMSs

- Based on the data model used
  - Traditional : Relational, Network, Hierarchical.
  - Emerging : Object-oriented, Object-relational.
- Based on number of users supported by the system
  - Single-user (typically used with micro-computers) vs. multi-user (most DBMSs).
- Based on the number of sites over which the database is distributed
  - Centralized (uses a single computer with one database) vs. distributed (uses multiple computers, multiple databases)
- Based on cost
  - Open source (free) vs. Commercial

Classification of DBMSs

- Based on the data model used
  - Traditional : Relational, Network, Hierarchical.
  - Emerging : Object-oriented, Object-relational.
- Based on number of users supported by the system
  - Single-user (typically used with micro-computers) vs. multi-user (most DBMSs).
- Based on the number of sites over which the database is distributed
  - Centralized (uses a single computer with one database) vs. distributed (uses multiple computers, multiple databases)
- Based on cost
  - Open source (free) vs. Commercial

Classification of DBMSs

- Based on the data model used
  - Traditional : Relational, Network, Hierarchical.
  - Emerging : Object-oriented, Object-relational.
- Based on number of users supported by the system
  - Single-user (typically used with micro-computers) vs. multi-user (most DBMSs).
- Based on the number of sites over which the database is distributed
  - Centralized (uses a single computer with one database) vs. distributed (uses multiple computers, multiple databases)
- Based on cost
  - Open source (free) vs. Commercial

Classification of DBMSs

- Based on the data model used
  - Traditional : Relational, Network, Hierarchical.
  - Emerging : Object-oriented, Object-relational.
- Based on number of users supported by the system
  - Single-user (typically used with micro-computers) vs. multi-user (most DBMSs).
- Based on the number of sites over which the database is distributed
  - Centralized (uses a single computer with one database) vs. distributed (uses multiple computers, multiple databases)
- Based on cost
  - Open source (free) vs. Commercial

Classification of DBMSs

- Based on the data model used
  - Traditional : Relational, Network, Hierarchical.
  - Emerging : Object-oriented, Object-relational.
- Based on number of users supported by the system
  - Single-user (typically used with micro-computers) vs. multi-user (most DBMSs).
- Based on the number of sites over which the database is distributed
  - Centralized (uses a single computer with one database) vs. distributed (uses multiple computers, multiple databases)
- Based on cost
  - Open source (free) vs. Commercial

Classification of DBMSs

- Based on the data model used
  - Traditional : Relational, Network, Hierarchical.
  - Emerging : Object-oriented, Object-relational.
- Based on number of users supported by the system
  - Single-user (typically used with micro-computers) vs. multi-user (most DBMSs).
- Based on the number of sites over which the database is distributed
  - Centralized (uses a single computer with one database) vs. distributed (uses multiple computers, multiple databases)
- Based on cost
  - Open source (free) vs. Commercial