

پوهنتون کابل

پوهنځی کمپیوتر ساینس

Introduction to Database

and Data Models

Lectures

03-04

تهیه کننده : پوهنیار محمد شعیب "زرین خیل"  
سال : 1389

# ***Introduction to Database and Data Models***

03

By: M Shuaib Zarinkhail

2010

# Why Databases?

---

Purpose: to help people/businesses keep track of (store & access) information

- Used by large and small businesses, and individuals to:
  - Track business information
  - Used across by networks, Internet
  - Store texts, graphics, images, sounds, videos
  - Can be paper based or computer based

## Why Databases? Cont ...

---

Software: Database Management System (DBMS)

- Single-user
  - One person accesses database at a time
- Multi-user
  - Data and sources are shared between many people (users)
  - More complicated

# Database System

---

Def: A computerized record-keeping system

- A repository or container for a collection of computerized **data files**
- A system that stores information and allows users to retrieve and update that information on demand
- Supports Operations
  - Add or delete files to & from the database
  - Insert, retrieve, remove, or change data (records) in a file/database

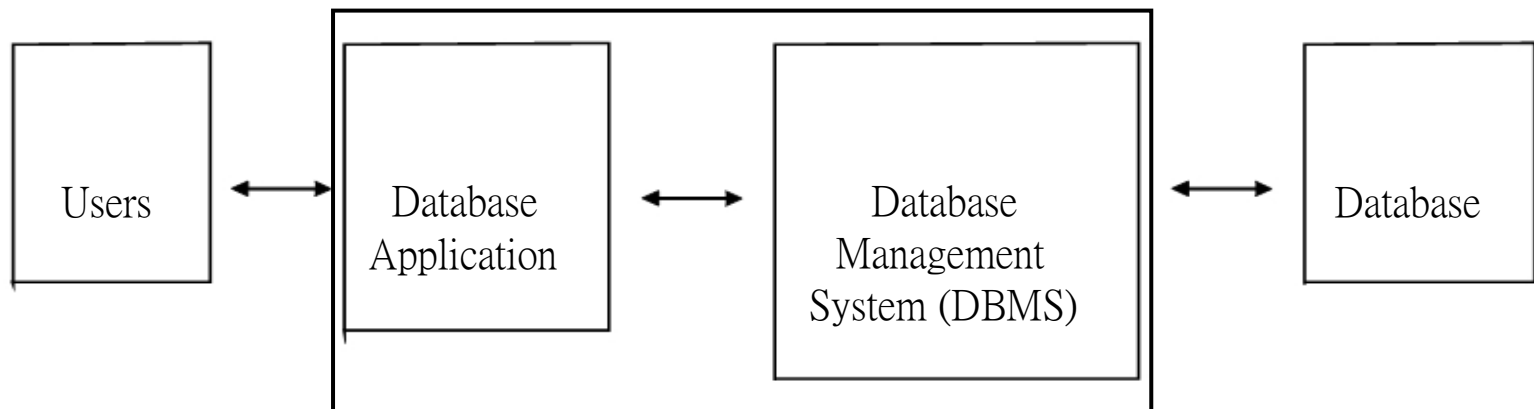
# Database System

---

Consists of:

- Database
- DBMS (Database Management System)
- Database Applications (Application Programs)
  - Some people include this part to the DBMS
- Users

# A Database System



Database System Components

# Database Definition

---

Def: A self-describing collection of integrated data records and related tables

- Self-describing contains both:
  - metadata (a description of its own internal structure)
  - userdata



## Database Definition Cont ...

---

Integrated Data Records: databases  
contains multiple files

- Userdata
- Metadata
- Indexes
- Application metadata

# Userdata

---

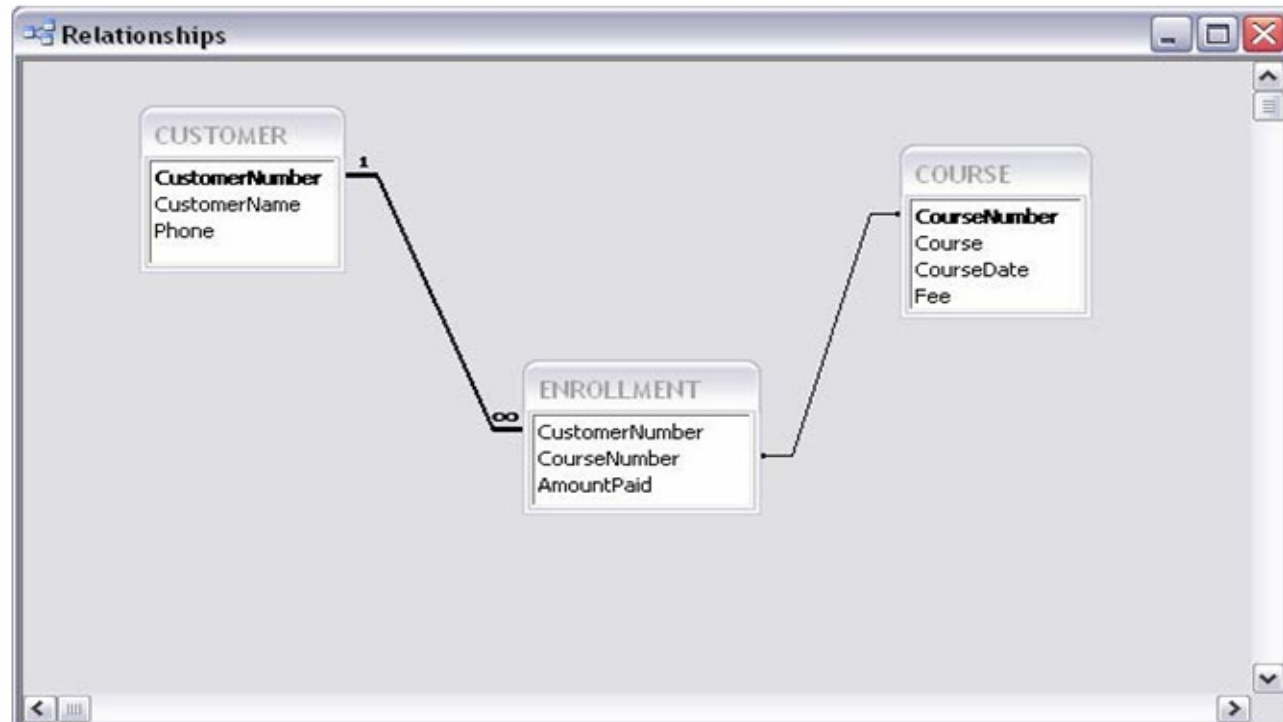
- Data files or tables
  - Typically stored as “RELATIONS” or tables of data
  - Uses a process called NORMALIZATION
- Rows
  - Records for entities that describe some business, etc
- Columns
  - Fields or attributes that describe the entities

# Metadata

---

- The internal database structure description, or data dictionary
- Usually stored in tables
  - Called SYSTEM TABLES
- Tables include:
  - List of tables
  - List of columns in tables
  - Indexes, keys, rules, etc

# Example Metadata



# Indexes

---

- Stores the relationships among data items
- Improves the performance and accessibility
- Improves processing speed
- Used for:
  - Sorting
  - Quick access to data (accelerate searches)

# Application Metadata

---

- Store application components such as:
  - User forms
  - Reports
  - Queries
  - etc
- Not all DBMSs support
- Of those that do support, not all store in the database
  - Typically accessed via DBMS tools – not directly

# DBMS

---

Def: A set of programs used to define, create and manipulate a database

- Features and functions vary
- Three main subsystems: NEXT SLIDE

# DMBS (Subsystems)

---

- 1) **Design-Tools:** Software that are used to create tables, forms, queries, and reports
  - Provides interfaces for built-in programming languages
- 2) **Run-Time:** Processes the components created (i.e. Web Design Tools)
  - Executes & displays data in forms, & provides access for standard programming languages (C++, etc.)

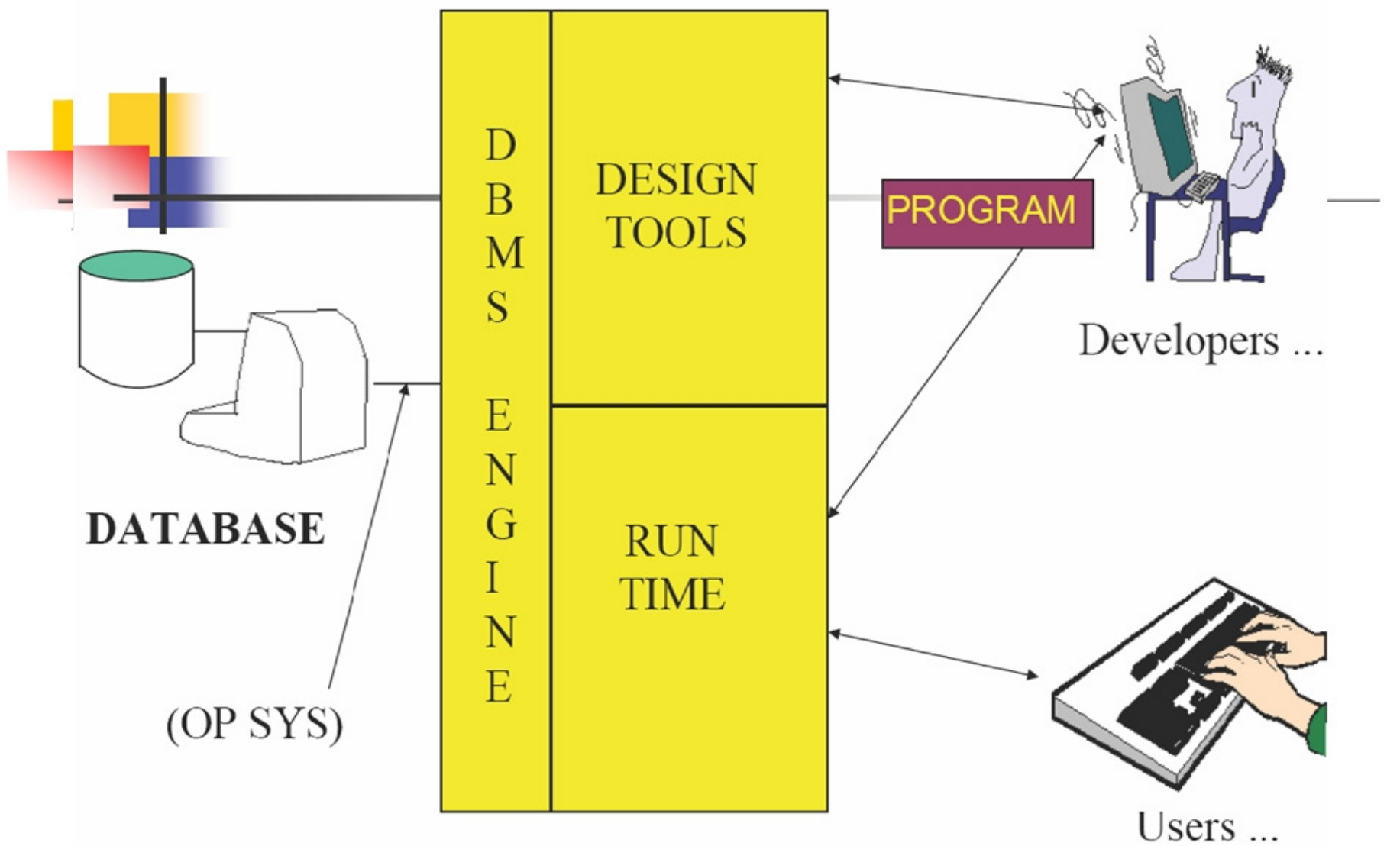


## **DBMS (subsystems) Cont.**

---

3) DBMS engine: Software between previous 2 and actual files of userdata that:

- Handles requests to access the database tables
- Does transaction management, record locking, backup & recovery, etc



# DBMS Examples

---

- Tens of DBMSs exist
  - MS-Access (Microsoft)
  - SQL-Server (Microsoft)
  - Oracle (Oracle)
  - DB2 (IBM)
  - ...

# Importance of DBMS

---

- Makes data management more efficient and effective
- Allows quick answers to *ad-hoc* queries
- Provides better access to managed data
- Promotes integrated view of operations
- Reduces the probability of inconsistent data

# DBMS Functions

---

1. Metadata Management
2. Userdata Management
3. Database Creation
4. Table Creation
5. Indexes and other overhead data  
Creation
6. Data Reading

# DBMS Functions

---

7. Data Editing
8. Data Maintenance
9. Data Implementation
10. Concurrency Control
11. Security
12. Backup and Recovery

# Metadata Management

---

- Entering data to metadata or to Data Dictionary
- Editing metadata
- Finding data through database
- Using data from a database

# Metadata Management

---

- Creating complicated structures to store data
- Stores different kinds of data as:
  - Screen definitions
  - Report definitions
  - Data validation
  - Audio / Video file formats
  - ...



# Userdata Management

---

- Add data to a database
- Retrieve data from a database
- Update data in a database
- Remove data from a database

# Database Creation

---

- A database can be created from:
  - Scratch
  - Existing databases
  - Computerized data
  - Paper based data files

# Table Creation

---

- Tables can be created from:
  - Data by users
  - Data from other databases
  - Data from other tables
  - Data from queries
  - Data from text files

# Indexes and other overhead data Creation

---

- Existed data could be indexed for:
  - Sorting data in a database
  - Finding needed data from a database
  - Finding indexed data from a database
  - Narrowing a huge database source for results

# Data Reading & Data Editing

---

- Reading data from a database
- Querying data from a database
  - Using Structured Query Language (SQL)
- Retrieving needed information from a database
- Edit data in a database
- Updated data in a database

# Data Maintenance

---

- Data maintaining in a database
  - Metadata
  - Userdata
- Format maintaining and updating in a database
  - Tables
  - Other internal structures

# Data Implementation

---

- Implement methods
- Implement rules
  - Standard rules
  - Business rules
- Implement RIC
  - What is RIC? – Referential Integrity Constraint

# Concurrency Control

---

## Database Administration Section

- Transaction manager or TP monitor must enforce certain recovery and concurrency controls
- Controls concurrency in:
  - Single-user databases (sometimes)
  - Multi-user databases (always)



# Security

---

## Database Administration Section

- Controls security
  - One section users could access only that section's data
  - While database managers could access almost all data in a database

# Backup and Recovery

---

## Database Administration Section

- In many cases databases are much important and huge (i.e. amazon.com)
- Not trust on hardware and software
- DBMSs should create backups and recover data

# ***Introduction to Database and Data Models***

04

By: M Shuaib Zarinkhail

2010

# Application Programs

---

- Subprograms in DBMSs
- A layer between users and DBMS
- Read and execute SQL commands ordered by users
- Developed by:
  - Software development companies
  - Database developers / designers
  - Database users (sometimes)

# Application Program Functions

---

1. Form Creation and Process
2. Query Implementation
3. Report Creation and Process
4. Logic Application Implementation
5. Application Control

# Form Creation and Process

---

An application program in MS-Access:

- Creates a form
- Processes a form
- User friendly functions in front-end
- SQL commands in back-end for:
  - Entering data
  - Editing data
  - Retrieving data
  - Removing data
- Does not show codes

# Example of a Form in Access

The screenshot shows an Access form window titled "Customer". It contains the following fields and data:

- CustomerName: Ariel Johnson
- Phone: 206.555.1234

Below the fields is a table titled "Course Enrollments" with the following data:

	Course	CourseDate	Fee	AmountPaid
▶	Adv Pastels	10/1/2003	\$500.00	\$250.
	Int Pastels	3/15/2003	\$350.00	\$350.
*				

At the bottom of the form, there are two record navigation controls. The top one shows "Record: 1 of 2" and the bottom one shows "Record: 1 of 7".

# Query Implementation

---

Application programs can:

- Create a query which is sent by user to the DBMS
- DBMS responds and shows the result(s) of the requested query on screen or on a front-end interface



# Example of a Query in Access

Enter Parameter Value ? X

Enter part of course name:

OK Cancel

CourseParameterQuery1 : Select Query

	CustomerName	Course	CourseDate	Fee	AmountPaid
▶	Ariel Johnson	Adv Pastels	10/1/2003	\$500.00	\$250.00
	Ariel Johnson	Int Pastels	3/15/2003	\$350.00	\$350.00
	Charles Jackson	Adv Pastels	10/1/2003	\$500.00	\$500.00
	Jeffrey Pearson	Adv Pastels	10/1/2003	\$500.00	\$500.00
	Leah Kyle	Adv Pastels	11/15/2003	\$500.00	\$250.00
*					

Record: 1 of 5



# (SQL code) from previews example

---

```
SELECT      CUSTOMER.CustomerName,      COURSE.Course,
            COURSE.CourseDate, COURSE.Fee, ENROLLMENT.AmountPaid

FROM CUSTOMER INNER JOIN
(COURSE INNER JOIN ENROLLMENT
ON COURSE.CourseNumber = ENROLLMENT.CourseNumber)
ON CUSTOMER.CustomerNumber = ENROLLMENT.CustomerNumber

WHERE (((COURSE.course)
Like "*" & [Enter part of course name:] & "*"))

ORDER BY CUSTOMER.CustomerName;
```

# Report Creation and Process

---

- Similar to previous task (Query Impl...)
- First creates the query for the ordered command
- Then exports the results in a report format



# Example of a Report in Access

---

## *Course Enrollment Report*

<i>Course</i>	<i>CourseDate</i>	<i>Fee</i>	<i>CustomerName</i>	<i>AmountPaid</i>	<i>Phone</i>
Adv Pastels	10/1/2003	\$500.00	Ariel Johnson	\$250.00	206.555.1234
Int Pastels	3/15/2003	\$350.00	Ariel Johnson	\$350.00	206.555.1234
Adv Pastels	10/1/2003	\$500.00	Charles Jackson	\$500.00	306.555.1488
Adv Pastels	10/1/2003	\$500.00	Jeffrey Pearson	\$500.00	212.555.8878
Adv Pastels	11/15/2003	\$500.00	Leah Kyle	\$250.00	444.555.3833

# Logic Application Implementation

---

- Very important task of Application Programs
- Stops errors during applications
- Creates error messages while occurs
  - i.e. A store with less materials than ordered by a customer

# Application Control

---

Control Application Program commands

- Controls and shows active and possible commands in a list to users
- Controls data by DBMS support
  - How to change? All data or a part of it

# Users

---

Divided into three categories:

1. Database Administrators (DBAs)
2. Application Programmers (Developers)
3. End Users (Users)

# Database Administrators (DBA)

---

- They are responsible for the overall control and manage of database systems
- DBA has the central responsibility for the data
- The one who understand the enterprise data and the needs of the enterprise with respect to that data at a *senior management level*



# DBA Jobs 1

---

- Define the logical schema
- Define the physical schema
- Define security and integrity constraints
- Define backup/recovery policies
- Assist users:
  - provide technical education
  - analyze the information requirements of users
  - consult application design
  - balance conflicting user requirements

## DBA Jobs 2

---

- Monitor performance and respond to changing requirements
  - Decide what data should be stored in DB in first place
  - Establish policies for maintaining and dealing with that data once it has been stored
    - Example: who can perform what operations on what data in what circumstances
- DBA is a manager, not only a technician

# Application Programmers

---

- The technical person responsible for implementing the Database Administrator's and/or other users' decisions
- Should have enough programming skills
- Responsible for writing Database Applications Programs in general programming languages
  - Host language: C++, Visual Basic, C#, Java, COBOL, PL1, ...
  - Embedded language: the language used to send DML to a DBMS

# Application Programmer-Jobs1

---

- Create the actual database
- Put in place the technical controls needed to enforce the various policy decisions made by the DBA
- Ensure that the system operates with adequate performance
- Provide a variety of other technical services

# Application Programmer-Jobs2

---

- Liaising with users
- Defining security and integrity constraints
- Defining dump/restore schemas or unload/reload utilities
- Monitoring performance and responding to changing requirements

## End Users

---

- Are general users of databases
- Do not need to know fundamentals of databases
- End users can use DBs in two ways:
  1. Directly via DBMSs
  2. Connected to an interface
    - Web-based
    - Designed by high level languages (VB, C#, ...)

# End Users

---

- End user requests have two types
  1. ad-hoc (unplanned): use a program with which a user directly send DML command to DBMS
    - This category uses DBMS
  2. canned (planned): use an easy-to-use program which was already developed by application programmers
    - This category uses client-server or (online DBs)