Review of Major DB Concepts

• Data and Information
  – Data: raw facts
  – Information: processed data

• Database
• Metadata
• DBMS: Database management system
• Database system

An Example of Data

• Sales per employee for each of X-company’s two divisions
Data and Its Structure

- Data is actually stored as bits, but it is difficult to work with data at this level.
- It is convenient to view data at different levels of abstraction.
- **Schema**: Description of data at some level. Each level has its own schema.
- Three schemas: physical, conceptual, and external.

Physical Data Schema

- Describes details of **how** data is stored: tracks, cylinders, indices etc.
- Early applications worked at this level - explicitly dealt with details.
- **Problem**: Routines hard-coded to deal with physical representation.
  - Changes to data structure difficult to make.
  - Application code becomes complex since it must deal with details.
  - Rapid implementation of new features impossible.
Conceptual Data Level

- Hides details.
  - In the relational model, the conceptual schema presents data as a set of tables.
- DBMS maps from conceptual to physical schema automatically.
- Physical schema can be changed without changing application:
  - DBMS must change mapping from conceptual to physical.
- Referred to as *physical data independence*.

Conceptual Data Level (con’t)

Application

___________________________

Conceptual view of data

<table>
<thead>
<tr>
<th>DBMS</th>
<th>Physical view of data</th>
</tr>
</thead>
</table>

5

6
External Data Level

• In the relational model, the external schema also presents data as a set of relations.
• An external schema specifies a view of the data in terms of the conceptual level. It is tailored to the needs of a particular category of users.
  – Portions of stored data should not be seen by some users.
    • Students should not see faculty salaries.
    • Faculty should not see billing data.
  – Information that can be derived from stored data might be viewed as if it were stored.
    • GPA not stored, calculated when needed.

External Data Level (con’t)

• Application is written in terms of an external schema.
• A view is computed when accessed (not stored).
• Different external schemas can be provided to different categories of users.
• Translation from external to conceptual done automatically by DBMS at run time.
• Conceptual schema can be changed without changing application:
  – Mapping from external to conceptual must be changed.
• Referred to as conceptual data independence.
Levels of Abstraction

- **payroll**
- **billing**
- **records**

View 1 ➔ View 2 ➔ View 3 ➔ External schemas

- **Conceptual schema**
- **Physical 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 indep is accomplished:
changing a schema at a lower level of DB, 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.
Historical Roots

- **File systems**
  - Provides historical perspective
  - Avoid pitfalls of data mgmt
  - Simple characteristics
  - Knowledge of converting a file system to a DB system
Contents of the CUSTOMER File

<table>
<thead>
<tr>
<th>C_NAME</th>
<th>C_PHONE</th>
<th>C_ADDRESS</th>
<th>C_ZIP</th>
<th>A_NAME</th>
<th>A_PHONE</th>
<th>TP</th>
<th>AMT</th>
<th>REN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alfred A. Ramos</td>
<td>615-544-1257</td>
<td>210 Fox Rd., Debry, TN</td>
<td>36123</td>
<td>Leah F. Hahn</td>
<td>615-002-1244</td>
<td>T1</td>
<td>$100.00</td>
<td>3/12/98</td>
</tr>
<tr>
<td>Leonore K. Dunn</td>
<td>713-394-1234</td>
<td>477 Oak Ln., Debry, TN</td>
<td>36123</td>
<td>Alex E. Boby</td>
<td>713-276-1249</td>
<td>T1</td>
<td>$50.00</td>
<td>5/23/96</td>
</tr>
<tr>
<td>Kathy W. Smith</td>
<td>615-694-2286</td>
<td>123 Oak Ln., Debry, TN</td>
<td>36123</td>
<td>Leah F. Hahn</td>
<td>615-802-1244</td>
<td>T2</td>
<td>$150.00</td>
<td>1/5/98</td>
</tr>
<tr>
<td>Paul F. Glorioso</td>
<td>615-894-2100</td>
<td>217 Lee Ln., Bala, TN</td>
<td>36123</td>
<td>Leah F. Hahn</td>
<td>615-802-1244</td>
<td>S1</td>
<td>$300.00</td>
<td>9/20/98</td>
</tr>
<tr>
<td>Merenn Orlando</td>
<td>615-222-1672</td>
<td>Box 111, New, TN</td>
<td>36123</td>
<td>Alex B. Ally</td>
<td>713-231-1249</td>
<td>T1</td>
<td>$100.00</td>
<td>1/24/98</td>
</tr>
<tr>
<td>Amy B. O'Byan</td>
<td>713-442-3384</td>
<td>500 Tind Dr., Fort, KY</td>
<td>26346</td>
<td>John T. Olsen</td>
<td>615-123-5589</td>
<td>T2</td>
<td>$850.00</td>
<td>8/5/98</td>
</tr>
<tr>
<td>James G. Brown</td>
<td>615-297-1226</td>
<td>21 Yae Rd., Nash, TN</td>
<td>37119</td>
<td>Leah F. Hahn</td>
<td>615-892-1244</td>
<td>S1</td>
<td>$120.00</td>
<td>3/1/98</td>
</tr>
<tr>
<td>George Williams</td>
<td>615-296-2550</td>
<td>555 Market, Nash, TN</td>
<td>37120</td>
<td>John T. Olsen</td>
<td>615-123-5589</td>
<td>S1</td>
<td>$250.00</td>
<td>6/23/98</td>
</tr>
<tr>
<td>Anne O. Vaness</td>
<td>713-382-1185</td>
<td>219 Elm, Crew, KY</td>
<td>25432</td>
<td>Alex B. Ally</td>
<td>713-228-1249</td>
<td>T1</td>
<td>$100.00</td>
<td>1/19/98</td>
</tr>
<tr>
<td>Debbie K. Smith</td>
<td>615-207-0859</td>
<td>2792 Main, Nash, TN</td>
<td>37119</td>
<td>John T. Olsen</td>
<td>615-123-5589</td>
<td>S1</td>
<td>$800.00</td>
<td>2/19/98</td>
</tr>
</tbody>
</table>

C_NAME = Customer name
C_PHONE = Customer phone
C_ADDRESS = Customer address
C_ZIP = Customer ZIP code
A_NAME = Agent name
A_PHONE = Agent phone
TP = Insurance type
AMT = Insurance policy amount in thousands of $
REN = Insurance renewal date

Basic File Terminology

Data: “Raw” facts that have little meaning unless they have been organized in some logical manner. The smallest piece of data that can be “recognized” by the computer is a single character, such as the letter A, the number 5, or some symbol such as; ? > * +. A single character requires one byte of computer storage.

Field: A character or group of characters (alphabetic or numeric) that has a specific meaning. A field might define a telephone number, a birth date, a customer name, a year-to-date (YTD) sales value, and so on.

Record: A logically connected set of one or more fields that describes a person, place, or thing. For example, the fields that comprise a record for a customer named J. D. Rudd might consist of J. D. Rudd’s name, address, phone number, date of birth, credit limit, unpaid balance, and so on.

File: A collection of related records. For example, a file might contain data about Company’s vendors; or, a file might contain the records for the students currently enrolled at University.
Contents of the AGENT File

<table>
<thead>
<tr>
<th>A_NAME</th>
<th>A_PHONE</th>
<th>A_ADDRESS</th>
<th>ZIP</th>
<th>HREFD</th>
<th>YTD_PAY</th>
<th>YTD_FIT</th>
<th>YTD_FICA</th>
<th>YTD_SLS</th>
<th>DEP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bob B.</td>
<td>714-289-1249</td>
<td>1234 Oak St, Miami, FL</td>
<td>33113</td>
<td>11/1/98</td>
<td>$35,556.24</td>
<td>$4,322.21</td>
<td>$1,694.87</td>
<td>$1,795.00</td>
<td>3</td>
</tr>
<tr>
<td>John P.</td>
<td>615-567-8901</td>
<td>5678 Pine Rd, Chicago, IL</td>
<td>60601</td>
<td>6/1/95</td>
<td>$21,123.45</td>
<td>$3,123.45</td>
<td>$1,798.56</td>
<td>$4,987.00</td>
<td>0</td>
</tr>
<tr>
<td>Mike T.</td>
<td>616-789-0123</td>
<td>4567 Oak Ave, New York, NY</td>
<td>10001</td>
<td>7/1/93</td>
<td>$35,456.78</td>
<td>$5,456.78</td>
<td>$1,789.01</td>
<td>$4,987.00</td>
<td>2</td>
</tr>
</tbody>
</table>

A_NAME = Agent name  
A_PHONE = Agent phone  
A_ADDRESS = Agent address  
ZIP = Agent ZIP code  
HREFD = Agent date of hire  
YTD_PAY = Year-to-date pay  
YTD_FIT = Year-to-date federal income tax paid  
YTD_FICA = Year-to-date Social Security paid  
YTD_SLS = Year-to-date sales in thousands of $  
DEP = Number of dependents

A Simple File System

Diagram showing file management and report programs for sales, personnel, customer, and agent files.
File System Critique

• File System Data Management
  – Require extensive programming in a 3GL
  – System admini. becomes difficult, as the # of file expands
  – Difficult and important to make changes in existing file structure
  – Omit security feature to safeguard data
  – Island of data information

File System Critique

• Structural and Data Dependence
  – Structural Dependence
  – Data Dependence

• Data dependence makes file system cumbersome
File System Critique

• Field Definitions and Name Conventions
  – A flexible (good) record definition anticipates
  reporting requirements by breaking up fields
  into their components.
  – Example:
    – Customer Name  
      Last Name, First Name, Initial
    – Customer Address  
      Street Address, City, State

<table>
<thead>
<tr>
<th>FIELD</th>
<th>CONTENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>CUS_LNAME</td>
<td>Customer last name</td>
</tr>
<tr>
<td>CUS_FNAME</td>
<td>Customer first name</td>
</tr>
<tr>
<td>CUS_INITIAL</td>
<td>Customer initial</td>
</tr>
<tr>
<td>CUS_AREA</td>
<td>Customer area code</td>
</tr>
<tr>
<td>CUS_PHONE</td>
<td>Customer phone</td>
</tr>
<tr>
<td>CUS_ADDRESS</td>
<td>Customer street address or box number</td>
</tr>
<tr>
<td>CUS_CITY</td>
<td>Customer city</td>
</tr>
<tr>
<td>CUS_STATE</td>
<td>Customer state</td>
</tr>
</tbody>
</table>

File System Critique

– Selecting proper field names is very important
  • descriptive within restrictions
  • Reflect documentation

• Data Redundancy
  – Data inconsistency (lack of data integrity)
  – Data anomalies
    • Modification anomalies
    • Insertion anomalies
    • Deletion anomalies
DBMS Languages

- **Data Definition Language (DDL):**
  - Used by the DBA and database designers to specify the conceptual schema of a database. In many DBMSs, the DDL is also used to define internal and external schemas (views). In some DBMSs, separate storage definition language (SDL) and view definition language (VDL) are used to define internal and external schemas.
  - Result of compiling DDL is catalog (a set of tables stored in a file)

- **Data Manipulation Language (DML):** Used to specify database retrievals and updates.
  - High-level (nonprocedural, declarative) DML:
    - Describe what data is needed w/o specifying how to get it
    - DML commands (data sublanguage) can be embedded in a general-purpose programming language (host language), such as COBOL, PL/1 or PASCAL.
  - Low-level (procedural) DML:
    - Describe what & how
    - stand-alone DML commands can be applied directly (query language).
Data Model

- **Schema**: description of data at some level (e.g., tables, attributes, constraints, domains)
- **Model**: tools and language for describing:
  - Conceptual and external schema
    - Data definition language (DDL)
  - Integrity constraints, domains (DDL)
  - Operations on data
    - Data manipulation language (DML)
  - Directives that influence the physical schema (affects performance, not semantics)
    - Storage definition language (SDL)

DBMS Interfaces

- Stand-alone query language interfaces
- Programmer interfaces for embedding DML in programming languages:
  - Pre-compiler Approach
  - Procedure (Subroutine) Call Approach
- User-friendly interfaces
  - Menu-based
  - Graphics-based (Point and Click, Drag and Drop etc.)
  - Forms-based
  - Natural language
  - Combinations of the above
  - Web Browser as an interface
DBMS Interfaces (Cont.)

- Parametric interfaces using function keys
- Report generation languages
- Interfaces for the DBA
  - Creating accounts, granting authorizations
  - Setting system parameters
  - Changing schemas or access path

Database System Environment
Database System Components

- Hardware
  - Computer
  - Peripherals
- Software
  - Operating system software
  - DBMS software
  - Applications programs and utilities software

Database System Components

- People
  - Systems administrators
  - Database administrators (DBAs)
  - Database designers
  - Systems analysts and programmers
  - End users
- Procedure
  - Instructions and rules
- Data
  - Collection of raw facts
Database System Components

• Complexity of DB system depends on
  – Organization’s size
  – Function
  – Corporate culture
  – Activities and environment

• DB solutions must be cost effective & strategically effective

DBMS Component Modules

• Stored data manager
• DDL compiler
• Run-time database processor
• Query compiler
• Pre-compiler
Typical DBMS Modules

Database System Utilities

- **Loading**  data stored in files into a database.
- **Backing up**  the database periodically on tape.
- **Reorganizing**  database file structures.
- **Report generation**  utilities.
- **Performance monitoring**  utilities.
- **Other functions,**
  - such as sorting, *user monitoring*, *security mgmt*, *data integrity*, *data compression*, etc.
Data dictionary / repository

- Used to store schema descriptions and other information such as design decisions, application program descriptions, user information, usage standards, etc.
- *Active* data dictionary is accessed by DBMS software and users/DBA.
- *Passive* data dictionary is accessed by users/DBA only

Types of Database Systems

- **Data model used:**
  - Traditional: Relational, Network, Hierarchical.
  - Emerging: Object-oriented, Object-relational, Temporal, Spatial
- **Other classifications:**
  - Single-user (typically used with micro-computers) vs. multi-user (most DBMSs).
  - Centralized (uses a single computer with one database) vs. distributed (uses multiple computers, multiple databases)
- **Use**
  - Transactional (Production)
  - Decision support
  - Data warehouse
Database Models

• A set of logical constructs
  – Represent data structure & data relationships

• Two types of database models
  – Conceptual model
    • Logical nature of data representation
    • What is represented
  – Implementation model
    • How the data are represented
    • How the data structure are implemented

• Three types of relationships
  – One-to-many (1:M)
    • PAINTER(1) paints PAINTING (M)
  – Many-to-many (N:M)
    • STUDENT(N) takes COURSE (M)
  – One-to-one (1:1)
    • EMPLOYER (1) manages STORE (1)
Implementation Database Models

- Hierarchical database model
- Network database model
- Relational database model

A Hierarchical Structure
Hierarchical Data Model

• Pros
  – Conceptual simplicity
  – Database security
  – Data independence
  – Database integrity
  – Efficiency dealing with a large DB

• Cons
  – Complex implementation
  – Difficult to manage
  – Lacks structural independence
  – Applications programming and use complexity
  – Implementation limitation
  – Lack of standards

• Child with multiple parents
A Network Database Model

Network Database Model

• Pros
  – Conceptual simplicity
  – Handles more relationship types
  – Data access flexibility
  – Data independence
  – Conformance to standards

• Cons
  – System complexity
  – Lack of structural independence
Relational Database Model

- **Pros**
  - Structural independence
  - Improved conceptual simplicity
  - Easier database design, implementation, mgmt, and use
  - Ad hoc query capability (SQL)
  - Powerful DBMS

- **Cons**
  - Hardware and software overhead
  - Possibility of poor design and implementation
  - Potential “islands of information” problems
Object-Oriented Database Model

• Basic Structure
  – Objects
  – Attribute
  – Class

• Comparison of OO data model and ER data model

OO Database Model

• Pros
  – Add semantic content
  – Visual presentation includes semantic content
  – Database integrity
  – Both structural and data independence

• Cons
  – Lack of OODM standards
  – Complex navigational data access
  – High system overhead slows transaction
The Development of Data Model

60's
Hierarchical

70's
CODASYL (network)

80's
Relational

90's
Object Bases

now

Choice for most new applications

Knowledge Bases
Wrap-up: The Evolution of Data Models

• Common characteristics required for data models
  – Some degree of conceptual simplicity
  – Represent the real word as close as possible
  – Representation of mini-word behavior must be in compliance with the consistency and integrity of any data model

• Database models and the internet
  – Flexible, efficient, and secure internet access
  – Support for complex data types and relationships
  – Seamless interfacing with multiple data sources and structures
  – Simplicity of conceptual database model
  – An abundance of available database tools
  – A powerful DBMS makes DBA’s job easier