Data Design

Peter Lo

Data Structures - File

- A File contains data about people, places, things or events that interact with the information system
- A File-oriented System processes one or more individual data files using a method called **File Processing**

Data Structures - Database

- A **Database** consists of linked data files, also called tables, which form an overall data structure
- A **Database Management System** (DBMS) is a collection of tools, features, and interfaces that enable users to add, update, manage, access, and analyze data in a database

Overview of File Processing

- Some systems use file processing to handle large volumes of structured data on a regular basis
- Many older systems utilized file-processing designs because that approach was well suited to mainframe hardware and batch input
- Although less common today, file processing can be more efficient and cost less than a DBMS in certain situations

Problems in File-Processing

- Data redundancy that common data common for several information systems is stored in several places.
- Data integrity problems can occur if updates are not applied in every file.
- Rigid data structure of a typical file-processing environment

Various Types of Files in a File-oriented Information System

- Master Files
  - A master file stores relatively permanent data about an entity
- Table Files
  - A table file contains reference data used by the information system
- Transaction Files
  - A transaction file stores records that contain day-to-day business and operational data

Various Types of Files in a File-oriented Information System

- Work Files
  - A work file is a temporary file created by an information system for a single task
- Security Files
  - A security file is created and saved for backup and recovery purposes
- History Files
  - A history file is a file copy created and saved for historical or archiving purposes

Overview of Database Systems

- A database provides an overall framework that avoids data redundancy and supports a real-time, dynamic environment.

Specific DBMS Advantages

- Scalability
  - Can be expanded, modified or downsized easily to meet the rapidly changing needs of a business
- Better support for client/server systems
  - Processing is distributed throughout the organization.
- Economy of scale
  - Utilization of hardware
- Sharing of data
  - Data can be shared across the business

Specific DBMS Advantages

- Balancing conflicting requirements
  - DBMS is managed by a DBA rather than end user
- Enforcement of standards
  - Standard of data names, formats and documentation are followed uniformly throughout the organization
- Controlled redundancy
  - All data are stored in a single database
- Security
  - Only legitimate users can access the database

Specific DBMS Advantages

- Increased programmer productivity
  - Programmers do not have to create the underlying file structure for a database.
- Data independence
  - System that interact with a DBMS are relatively independent of how the physical data is maintained.

DBMS Components

- Interfaces for users, database administrators, and related systems
  - A DBMS provides an interface between a database and users who need to access the data.
  - When users, DBA, and related information systems request data and services, the DBMS processes the request, manipulates the data, and provides a response
- Data manipulation language
  - Controls database operations including storing, retrieving, updating, and deleting data.

DBMS Components

- Schema
  - A **Schema** is a complete definition of a database, including descriptions of all fields, records, and relationships
  - A **Subschema** defines only those portions (view) of the database that a particular system or user needs or is allowed to access.
- Physical data repository
  - The data dictionary, which was developed during the systems analysis phase, is transformed into a Physical Data Repository during the systems design phase.

Data Warehousing

- A data warehouse is an integrated collection of data that can support management analysis and decision making.
- For example, in a typical company, data is generated by transaction-based systems, such as order entry, inventory, accounts receivable, and payroll. If a user wants to know the customer number on sales order 4071, he or she can retrieve the data easily from the order entry system.

Data Mining

- Data mining software looks for meaningful patterns and relationships among data.
- For example, data mining software could help a consumer products firm identify potential customers based on their prior purchases.

Data Design Terminology

- Entity
  - Define an entity as a person, place, thing, or event for which data is collected and maintained.
- Field
  - Define a field, also called an attribute, as a single characteristic or fact about an entity.
- Record
  - Define a record, also called a tuple (rhymes with couple), as a set of related fields that describes one instance, or member of an entity, such as one customer, one order, or one product.

Data Design Terminology

- File and table
  - Records are grouped into files or tables, depending on the data design model.
- File-oriented System vs Database Environment
  - In a file-oriented environment, a file can be defined as a set of related records that contains data about a person, place, thing, or event.
  - In a database environment, a set of related records is grouped into a table that stores data about a specific entity.

Key Fields

- Key fields are used to organize, access, and maintain the data and data structures.
- The four types of key fields are:
  - Primary Keys
  - Candidate Keys
  - Foreign Keys
  - Secondary Keys

Primary Keys

- A primary key is a field or combination of fields that **Uniquely** (identifying only one member of an entity, such as one student, one customer, or one airline passenger) and **Minimally** (containing no information beyond what is needed to identify the record) identifies a particular member of an entity.

Candidate Keys

- Any field that could serve as a primary key is called a candidate key.
- Files usually have a single candidate key, but files with multiple candidate keys are possible.
- For example, a customer could use either the unique customer number or the HKID card number as a primary key.
- Any field that is not a primary key or a candidate key is called a non-key field.

Foreign Keys

- A foreign key is a field in one table that must match a primary key value in another table in order to establish the relationship between the two tables.

Secondary Keys

- A secondary key is a field or combination of fields that can be used to access or retrieve records.

Referential Integrity

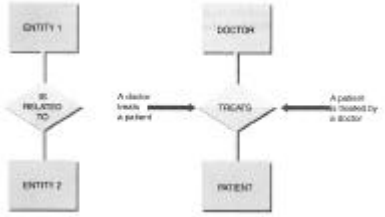
- Referential integrity is a set of rules that avoids data inconsistency and quality problems.
- In a relational database, referential integrity means that a foreign key value cannot be entered in one table unless it matches an existing primary key in another table.
- Referential integrity also can prevent the deletion of a record.

Data Relationships

- An entity is a person, place, thing, or event for which data is collected.
- A relationship is a logical link between entities based on how they interact.
- For example, products are stored in warehouses, so a relationship exists between the two entities.

Entity-Relationship Diagrams

- An Entity-Relationship Diagram (ERD) is a graphical model of the information system that depicts the relationships among system entities.



Three Main Types of Relationships

- One-to-One relationship (1:1)
- One-to-Many relationship (1:M)
- Many-to-Many relationship (M:N)

One-to-one relationship (1:1)

- It exists when exactly one of the second entity occurs for each instance of the first entity.

One-to-Many Relationship (1:M)

- It exists when one occurrence of the first entity can be related to many occurrences of the second entity, but each occurrence of the second entity can be associated with only one occurrence of the first entity.

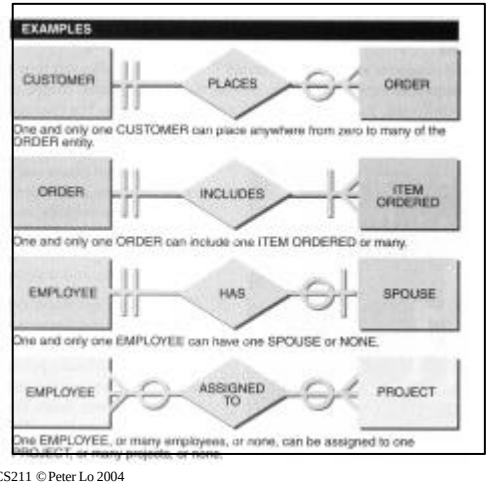
Many-to-Many Relationship (M:N)

- It exists when one instance of the first entity can be related to many instances of the second entity, and one instance of the second entity can be related to many instances of the first entity.

Cardinality

- Cardinality describes how instances of one entity relate to instances of another entity.

SYMBOL	MEANING	UML REPRESENTATION
	One and only one	1
	One or many	1..*
	Zero, or one, or many	0..*
	Zero, or one	0..1



Creating an Entity Relationship Diagram (ERD)

1. Identify the entities.
2. Determine all significant events, transactions, or activities that occur between two or more entities.
3. Analyze the nature of the interaction.
4. Draw the Entity Relationship Diagram.

Normalization

- Normalization is a process by identifying and correcting inherent problems and complexities in your record designs.
- To develop an overall database design that is simple, flexible, and free of data redundancy.
- The normalization process typically involves three stages: First Normal Form, Second Normal Form, and Third Normal Form.

Un-normalized Form

- Designing records is easier if you use a standard method of showing the record structure, fields, and primary keys.
- UNF:

```
ORDER (ORDER-NUM, ORDER-DATE, CUSTOMER-NUM,
CUSTOMER-NAME, CUSTOMER-ADDRESS, (PRODUCT-NUM,
PRODUCT-DESC, NUM-ORDERED))
```


First Normalized Form

- The definition of 1st normal form
  - There are no repeating groups
  - All the key attributes are defined
  - All attributes are dependent on the primary key
- 1NF:

```
ORDER (ORDER-NUM, ORDER-DATE, CUSTOMER-NUM,
CUSTOMER-NAME, CUSTOMER-ADDRESS)

ORDER-LINE (ORDER-NUM, PRODUCT-NUM, PRODUCT-DESC,
NUM-ORDERED)
```

Second Normal Form

- A table is in 2nd normal form if
  - It's in 1st normal form
  - It includes no partial dependencies (where an attribute is dependent on only a part of a primary key).
- 2NF:

```
ORDER (ORDER-NUM, ORDER-DATE, CUSTOMER-NUM,
CUSTOMER-NAME, CUSTOMER-ADDRESS)
PRODUCT (PRODUCT-NUM, PRODUCT-DESC)
ORDER-LINE (ORDER-NUM, PRODUCT-NUM, NUM-ORDERED)
```


Third Normal Form

- The definition of 3rd normal form
  - It's in 2nd normal form
  - It contains no transitive dependencies (where a non-key attribute is dependent on another non-key attribute).
- 3NF

```
ORDER (ORDER-NUM, ORDER-DATE, CUSTOMER-NUM)
CUSTOMER (CUSTOMER-NUM, CUSTOMER-NAME, CUSTOMER-
ADDRESS)
PRODUCT (PRODUCT-NUM, PRODUCT-DESC)
ORDER-LINE (ORDER-NUM, PRODUCT-NUM, NUM-ORDERED)
```


Four Steps in Database Design

1. Create the initial ERD
2. Assign all data elements to entities
3. Create 3NF designs for all records, taking care to identify all primary, secondary, and foreign keys
4. Verify all data dictionary entries

Database Models

- The four basic database models are:
  - Hierarchical Database
  - Network Database
  - Relational Database
  - Object-oriented Database

Hierarchical Database

- In a hierarchical database, data is organized like a family tree or organization chart, with branches representing parent records and child records.
- A parent record can have multiple child records, but each child record can have only one parent.
- Hierarchical databases were widely used on mainframe computers in the 1960s, but today they are found only on older systems.
- Hierarchical databases have rigid data structures that are relatively inflexible and complex.

Network Databases

- A network database resembles a hierarchical design.
- In a network database, a child record can have relationships with more than one parent.

Relational Database

- The relational design is powerful and flexible.
- Relational database uses common fields, which are attributes that appear in more than one table, to establish relationships between the tables and form an overall data structure.
- Three decades after its introduction, the relational design still is the predominate model.
- The key is the use of common fields, which establish relationships between tables and form an overall data structure.

Object-Oriented Databases

- Each object in an Object-Oriented Databases has a unique object identifier, which is similar to a primary key in a relational database.
- The identifier allows the object to interact with other objects and form relationships

Data Storage

- Physical design requires an understanding of the difference between logical and physical records, and an understanding of data storage formats, including special characteristics of date fields.

Logical and Physical Records

- Logical Record
  - Contains field values that describe a single person, place, thing, or event.
- Physical Record
  - Also called a block, is the smallest unit of data that is accessed by the operating system.

Data Storage Formats

- The primary data storage formats: EBCDIC, ASCII, Unicode, and binary.

ASCII	Symbol	EBCDIC
00110000	0	11110000
00110001	1	11110001
00110010	2	11110010
00110011	3	11110011
00110100	4	11110100
00110101	5	11110101
00110110	6	11110110
00110111	7	11110111
00111000	8	11111000
00111001	9	11111001
01000001	A	11000001
01000010	B	11000010
01000011	C	11000011
01000100	D	11000100
01000101	E	11000101

EBCDIC

- Stands for Extended Binary Coded Decimal Interchange Code.
- It is a data storage method used on most mainframe computers.

ASCII

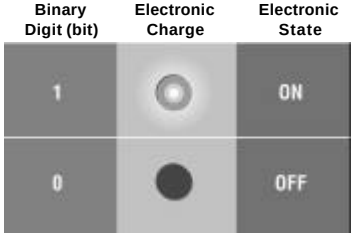
- Stand for American Standard Code for Information Interchange
- It is used on most minicomputers and personal computers.

Unicode

- It is a relatively recent coding method that represents characters as integers.
- Unlike EBCDIC and ASCII, which use 8 bits for each character, Unicode requires 16 bits per character, which allows it to represent more than 65,000 unique characters.
- It is a coding scheme capable of representing all world's languages.
- That is important as software design becomes global, and multinational companies span many continents.

Binary

- Compared to character-based coding, offer more efficient storage of numeric data.
- Represent data as two unique digits: 0 and 1



Date Fields

- Companies and governmental agencies around the world spent millions of dollars to update legacy systems that only provided two digits for the year when year 2000 approached.
- As it turned out, most difficulties were minor.
- This problem would not have occurred if dates had been stored in an International Organization for Standardization (ISO) format, which requires a format of four digits for the year, two for the month, and two for the day. (YYYYMMDD)
- A date stored in that format can be sorted easily and used in comparisons.

Date Fields

- A **standard Julian Date** is a five-digit number in which the first two digits represent the year and the last three digits represent the day of the year.
  - E.g. 01001 for 1 January 2001
- An **extended Julian Date** is a seven-digit number in which the first four digits represent the year.
  - E.g. 2001174 for 23 June 2001
- An **Absolute Date** is the total number of days from some specific base date.

Data Control

- A well-designed DBMS must provide built-in control and security features, including subschemas, passwords, encryption, audit trail files, and backup and recovery procedures to maintain data

Data Control

- Subschema
  - Subschema can be used to provide a limited view of the database to a specific user, or level of users.
- Passwords
  - Users must furnish a proper user ID and password to access a file or database.
- Encryption
  - Stored data also can be encrypted to prevent unauthorized access.

Data Control

- Backup and Recovery
  - All system files and databases must be backed up regularly and a series of backup copies must be retained for a specified period of time.
- Audit Trail
  - Audit log files, which record details of all accesses and changes to the file or database, can be used to recover changes made since the last backup.