**Chapter 1**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Data** | Raw facts | Raw data – not yet been processed to reveal the meaning | | | |
| Building blocks of information | | | | |
| Data management – generation, storage, and retrieval of data | | | | |
| **Information** | Produced by processing data | | | | |
| Reveals the meaning of data | | | | |
| Enables knowledge creation | | | | |
| Should be accurate, relevant, and timely to enable good decision making | | | | |
| **Database** | Shared, integrated computer structure that stores a collection of: | | | | End-user data – raw facts of interest of end user |
| Metadata – data about data, which the end-user data are integrated and managed |
| **Database Management System (DBMS)** | - Collection of programs  - Manages the database structure  - Controls access to data stored in the database | | | | |
| **Role of the DBMS** | - intermediary between the user and the database  - enables data to be shared  - presents the end user with an integrated view of the data  - receives and translate application requests into operations required to fulfill the requests  - hides database’s internal complexity from the application programs and users | | | | |
| **Advantages of the DBMS** | - better data integration and less data inconsistency  **data inconsistency** ­– different versions of the same data appear in different places  - increased end-user productivity  - improved:  - data sharing  - data security  - data access  - decision making  **data quality –** promoting accuracy, validity and timeliness of data | | | | |
| **Types of Databases** | | | | | |
| **Single-user DB** | Supports one user at a time | | | | |
| **Multiuser DB** | Supports multiple users at the same time | | | | |
| **Workgroup DB** | Supports a small number of users or a specific department | | | | |
| **Enterprise DB** | Supports many users across many department | | | | |
| **Centralized DB** | Data is located at a single site | | | | |
| **Distributed DB** | Data is distributed across different sites | | | | |
| **Cloud DB** | Created and maintained using cloud data services that provide defines performances measures for the database | | | | |
| **General-purpose DB** | Contains a wide variety of data used in multiple disciplines | | | | |
| **Discipline-specific DB** | Contains data focused on specific subject areas | | | | |
| **Operational DB** | Designed to support a company’s day-to-day operations | | | | |
| **Analytical DB** | Stores historical data and business metrics used exclusively for tactical or strategic decision making  **data warehouse** – stores data in a format optimized decision support | | | | |
| **Online Analytical Processing (OLAP)** | Enable retrieving, processing and modeling data from the data warehouse | | | | |
| **Business Intelligence** | Captures and processes business data to generate information that support decision making | | | | |
| **Unstructured Data** | It exists in their original state | | | | |
| **Structured Data** | It results from formatting | | | | |
| **Semistructured data** | Processed to some extent | | | | |
| **Extensible Markup Language (XML)** | Represents data elements in textual format | | | | |
|  | | | | | |
| **Database Design** | Focuses on the design of the database structure that will be used to store and manage end-user data | | | | |
| **Evolution of File System Data Processing** | **Manual File Systems** | | | Accomplished through a system of file folders and filing cabinets | |
| **Computerized File Systems** | | | Data processing (DP) specialist: created a computer-based system that would track data and produce required reports | |
| **File System Redux: Modern End-User Productivity Tools** | | | Includes spreadsheet programs such as Microsoft Excel | |
| **Basic File Terminology** | **Data** | | | Raw facts | |
| **Field** | | | A character or group of characters (alphabetic or numeric) that has specific meaning. It is used to define and store data | |
| **Record** | | | A logically connected set of one or more fields that describes a person, place, or thing | |
| **File** | | | A collection of related records | |
| **Problems with File System Data Processing** | Lengthy development times | | | | |
| Difficulty of getting quick answers | | | | |
| Complex system administration | | | | |
| Lack of security and limited data sharing | | | | |
| Extensive programming | | | | |
| **Structural Dependence** | Access to a file is dependent on its own structure | | | | |
| **Structural Independence** | File structure is changed without affecting the application’s ability to access the data | | | | |
| **Data Dependence** | Data access changes when data storage characteristics change | | | | |
| **Data Independence** | Data storage characteristics is changed without affecting the programs’ ability to access the data | | | | |
| **Data Redundancy** | Unnecessarily storing same data at different places | | | | |
| **Islands of Information** | Scattered data locations | | | | |
| **Data Redundancy Implications** | - poor data security  - data inconsistency  - increased likelihood of data-entry errors when complex entries are made in different files | | | | |
| **Data Anomaly** | Develops when not all of the required changes in the redundant data are made successfully | | | | |
| **Types of Data Anomaly** | Update Anomalies | | | | |
| Insertion Anomalies | | | | |
| Deletion Anomalies | | | | |
| **Database Systems** | Logically related data stored in a single logical data repository | | | | |
| **Current generation DBMS software:** | - Stores data structures, relationships between structures, and access paths  – Defines, stores, and manages all access paths and components | | | | |
| **DBMS Functions** | Data dictionary management | | **Data dictionary –** stores definitions of the data elements and their relationships | | |
| Data storage management | | **Performance tuning;** Ensures efficient performance of the database in terms of storage and access speed | | |
| Data transformation and presentation | | Transform entered data to conform to required data structures | | |
| Security management | | Enforces user security and data privacy | | |
| Multiuser access control | | Sophisticated algorithms ensure that multiple users can access the database concurrently without compromising its integrity | | |
| Backup and recovery management | | Enables recovery of the database after a failure | | |
| Data integrity management | | Minimizes redundancy and maximizes consistency | | |
| Database access languages and application programming interfaces | | **Query language**: Lets the user specify what must be done without having to specify how | | |
| **Structured Query Language (SQL)**: De facto query language and data access standard supported by the majority of DBMS vendors | | |
|  | Database communication inter | | Accept end-user requests via multiple, different network environments | | |
| **Disadvantages of Database Systems** | Increased costs | | | | |
| Management complexity | | | | |
| Maintaining currency | | | | |
| Vendor dependence | | | | |
| Frequent upgrade/replacement cycles | | | | |
|  | | | | | |

**Chapter 2. Entity Relationship Diagram (ERD)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Databases** | One of the ways to organize data  May or may not be inside a computer | | |
| **Electronic databases** | Began to exist way back in the 1960s | | |
| **Dr. Edgar F. Codd** | In 1970, he published an academic paper on relational databases | | |
|  | His article inspired the creation of a language to handle database transactions called Sequel or Structured English Query Language | | |
| **SQL** | Way of giving the computer instructions on how to handle large volume of data | | |
| **dBase** | The earliest widely used database product for the microcomputer, which also ran on Old Apple computers. | | |
| **Microsoft Access** | One of the most commonly used products for personal computing | | |
| Hierarchy of Data in a Database | Database | | Collection of related facts and figures regarding one subject matter |
| Tables | | Part of the database that contains data about one aspect of the subject of the database |
| Records | | Instance or row of a table |
| Field | | Attribute of the record or column of a table |
| **Entity Relationship Diagram (ERD)** | Provide a way to document the entities in a database, along with the attributes that describes them | | |
| **Three major methods** | * The Chen model (named after the originator of ER modelling, Dr. Peter P.S. Chen) * Information Engineering (IE, or “Crow’s feet”) and * Unified Modelling Language (UML) | | |
| **ER model** | Defines the conceptual view of database | | |
| Works around real-world entity and association among them | | |
| **Entity** | A real-world thing either animate or inanimate that can be easily identifiable and distinguishable | | |
| **Entity set** | Collection of similar types of entities | | |
| **Attributes** | Properties that represent the entities  All attributes have values | | |
| **ER Model Constructs** | Entity instance | Person, place, object, event, concept (often corresponds to a row in a table | |
| Entity Type | Collection of entities (often corresponds to a table) | |
| Attribute | Property or characteristics of an entity type (often corresponds to a field in a table) | |
| Relationship instance | Link between entities (corresponds to primary key – foreign key equivalencies in related tables)  Between specific entity instances | |
| Relationship type | Category of relationship…link between entity types  Modeled as the diamond and lines between entity types | |
|  | | | | | |
|  | | | | | |
| **Classification of Attributes** | * Required vs. Optional Attributes * Simple vs. composite Attribute * Single-valued vs. Multivalued Attribute * Stored vs. Derived Attributes * Identifier Attributes | | |
| **Identifier (key)** | An attribute (or combination of attributes) that uniquely identifies individual instances of an entity type | | |
| **Characteristics of identifier** | Will not change in value  Will not be null  No intelligent identifiers  Substitute new, simples for long, composite key | | |
| **Candidate Key** | An attribute that could be a key…satisfies the requirements for being a key | | |
| **Primary key** | Identifies a record uniquely. All values in this field are different from each other | | |
| **Foreign key** | Is a primary key in another table | | |
| **Composite key** | Key that consists of more that one field | | |
| **Relationship** | Represents the connection of one entity to another | | |
| **One-to-One relationship** | One record in the first table can be related to only one record in the second table | | |
| **One-to-Many relationship** | One record in the first table can be related to many records in the second table, but records in the second table can be related to only one record in the first table | | |
| **Many-to-Manny relationship** | Record in the first table could be related to many records in the second table, and each record in the second table can also be related to many records in the first table | | |
|  | | | | |
| **Associative Entity** | Combination of relationship and entity | | |
| **Degree of Relationships** | Number of entity types that participate in it  Unary Relationship  Binary Relationship  Ternary Relationship | | |
|  | | | | |
| **Cardinality Constraints** | The number of instances of one entity that can or must be associated with each instance of another entity | | |
| **Strong Entities** | Exist independently of other types of entities, has its own unique identifier, represented with single-line rectangle | | |
| **Weak entity** | Dependent on a strong entity…cannot exist on its own, does not have its own unique identifier, represented with double-line rectangle | | |
| **Identifying relationship** | Links strong entities to weak entities, represent with double line diamond | | |
|  | | | |