Introduction to database system

17/01/21 02:09

*capítulo 1

Definiciones

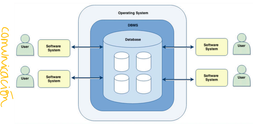
- Database system (DBS): sistema de registro compartiendo con el propósito de mantener la información y tenerla disponible en cualquier momento.

- Database management system (DBMS): set de programas que permiten el manejo/administración de una base de datos.
- Funciones de DBMS:
  - Data definition: creación y administración de relaciones, dependencias, vistas, etc.
  - Data manipulation: insertar, update, deletar, recuperación, reagrupación y agregar data.
  - System and data security: controlar el acceso al sistema, recursos y data.
  - Programming language support.

Componentes principales de una DBS

- Hardware y sistema operativo
- DBMS
- Base de datos
- Usuarios.

comunicación



Data vs información

- Data: materia prima con la que el software actúa para producir información útil para los usuarios finales.

- Información: data procesada y ordenada que transmite significado a los usuarios.

Objetivos de un database system

- Objetivos primarios y secundarios:

1. Seguridad y protección: prevención de usuarios no autorizados.
2. Fiabilidad: asegurar estabilidad y predictibilidad.
3. Facilitación de múltiples usuarios.
4. Flexibilidad: obtener data de varios métodos.
5. Fácil acceso y manipulación a la data.
6. Exactitud y consistencia.
7. Variedad: estandarización de la data.
8. Disponibilidad.

- Objetivos adicionales:
  - physical data independence
  - logical data independence
  - control of redundancy
  - integrity control: que no haya data invalida (DBMS)
  - clear data definition
  - friendly user interface.
  - Tunability: rendimiento estable / afinar el rendimiento.

Clarification on data independence

- Es la inmunidad de los programas a cambios en la estructura y en la estrategia de acceso a la data.
  - Es necesario por:
    - Diferentes vistas lógicas de la data.

- Independencia física: dónde/cómo está almacenada la información.
- Independencia lógica:

Ventajas de un database system

- Redundancy can be reduced.
- Inconsistencies can be avoided.
- Data can be shared.
- Standards can be enforced.
- Security restrictions can be applied.
- Integrity can be maintained.
- Conflicting requirements can be balanced.
- Performance is improved due to speed of processing, reduction in paperwork, etc.
- Maintenance and retrieval of data are very easy—no complicated application program is needed.
- It's not solely dependent on the high level language (HLL) programming for use.
- Logical views of data stored can be easily created.
- Record structures can change without any adverse effect on data retrieval (due to physical and logical data independence).

Approaches to database design

- Instant small system, which uses one file
- File processing systems, which involve many files
- Other non-relational systems, such as hierarchical, inverted-list, and network approaches
- Relational databases (the focus of this course), which were pioneered by prominent individuals such as Edgar Codd, Ronald Fagin, and Christopher Date, among others
- Object databases, which are an alternate approach (also discussed later in the course)

These five approaches may actually be rearranged into three broad categories, each of which will be clarified. The categories are as follows:

- Conventional files
- Outdated database approaches such as the hierarchical model, network model, and the inverted-list model
- Dominant contemporary database approaches as in the relational model and the object-oriented model

- Emergin database approaches:

- **Hadoop:** This describes a framework for handling distributed processing of large data sets (see [Apache 2014]
- **Entity-Attributes-Value (EAV) Model:** This approach reduces a database to three principal storage entities: an entity for defining other entities; an entity for defining properties (attributes) of entities; an EAV entity that connects the other two entities and stored values for entity-attribute combinations (see [Wikipedia 2016]).
- **NoSQL:** This approach refers to a family of non-relational database approaches that are designed for managing large data sets, while providing benefits such as flexibility, scalability, availability, lower costs, and special capabilities. Four related methodologies are key-value stores, graph stores, column stores, and document stores (see [IBM 2015]).

Database development life cicle

DDLC Phase	Related Deliverable(s)
Database Investigation and Analysis	Initial Database Requirements
Database Modeling	Database Model
Database Designing	Database Design Specification
Database Development	Actual Database
Implementation	Actual Database in Use
Management	Enhanced Database; Revised Database Documentation