Lezione 2

Database System: Concetti ed Architetture

Modelli dei Dati

- **Modelli dei Dati**: Un insieme di concetti per descrivere la *struttura* di un database a alcuni *vincoli* che il database deve rispettare.
- **Operazioni del modello dei dati**: Operazioni per specificare recuperi ed aggiornamenti del database in riferimento ai concetti del modello dei dati. Possono includere *operazioni di base* ed *operazioni definite dall'utente*. [oggetti]

Categorie di modelli dei dati

- **Concettuale** (**high-level**, **semantico**): Forniscono concetti che sono vicini al modo in cui molti utenti percepiscono i dati. (Esempio: entità, oggetti)
- **Fisico** (**low-level**, **interno**): Forniscono concetti che descrivono dettagliatamente come i dati sono memorizzati nel computer.
- **Implementabili** (**rappresentazionali**): Forniscono concetti che possono essere compresi dagli utenti finali ma che non sono troppo lontani dal modo in cui i dati sono organizzati entro il computer.

History of Data Models

- Relational Model: proposed in 1970 by E.F. Codd (IBM), first commercial system in 1981-82. Now in several commercial products (DB2, ORACLE, SQL Server, SYBASE, INFORMIX).

- Network Model: the first one to be implemented by Honeywell in 1964-65 (IDS System). Adopted heavily due to the support by CODASYL (CODASYL - DBTG report of 1971). Later implemented in a large variety of systems - IDMS (Cullinet - now CA), DMS 1100 (Unisys), IMAGE (H.P.), VAX -DBMS (Digital Equipment Corp.).

- Hierarchical Data Model: implemented in a joint effort by IBM and North American Rockwell around 1965. Resulted in the IMS family of systems. The most popular model. Other system based on this model: System 2k (SAS inc.)

History of Data Models

- Object-oriented Data Model(s): several models have been proposed for implementing in a database system. One set comprises models of persistent O-O Programming Languages such as C++ (e.g., in OBJECTSTORE or VERSANT), and Smalltalk (e.g., in GEMSTONE). Additionally, systems like O_2, ORION (at MCC - then ITASCA), IRIS (at H.P.- used in Open OODB).

- Object-Relational Models: Most Recent Trend. Started with Informix Universal Server. Exemplified in the latest versions of Oracle-10i, DB2, and SQL Server etc. systems.

Schema vs. Istanze

- **Schema di un Database**: La descrizione del database. Include la descrizione della struttura e dei vincoli da rispettare.
- **Diagramma di Schema**: Una visualizzazione diagrammatica di (qualche aspetto di) uno schema.
- **Costrutto dello Schema**: Una componente dello schema o un oggetto all'interno dello schema. (esempio: STUDENTI, CORSI)
- **Istanza del Database**: I dati veri e propri contenuti nel database in un particolare momento nel tempo. Chiamata anche stato del database oppure occorrenza.

STUDENTI

Matricola	Cognome	Nome	CdL

CORSI

Codice	NomeCorso	Ore	Dipart.

Schema Vs. Stato

del Database

- **Stato del Database:** Si riferisce al contenuto del database in un particolare momento (solitamente “adesso”).
- **Stato Iniziale del Database:** Si riferisce al database quando viene caricato.
- **Stato Valido:** Uno stato che soddisfa che strutture ed i vincoli del database.
- **Distinzione**
  - Lo **schema del database** *non* cambia *molto spesso*. Lo **stato del database** cambia *ogni volta che il database viene aggiornato*.
  - Lo **schema** viene chiamato anche **intensione**, mentre lo **stato** viene chiamato **estensione**.

Architettura a 3 livelli

[Three-Schema Architecture, ANSI/SPARC]

- Proposta per supportare le caratteristiche dei DBMS di:
  - Indipendenza tra programmi e dati
  - Supporto di viste multiple d'utente
  - Uso di un catalogo per memorizzare la descrizione (schema) del database

Architettura a 3 livelli

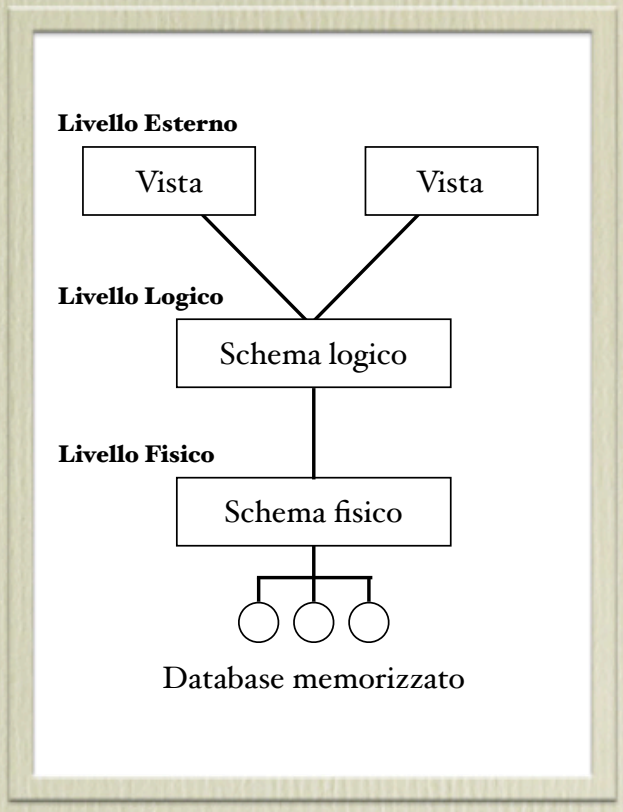
[Three-Schema Architecture, ANSI/SPARC]

- Obiettivo di separare le applicazione dell'utente dalla base di dati fisica.
- Definisce gli schemi del DBMS a *tre livelli*:
  - **Schema interno** al livello interno, per descrivere le strutture di memorizzazione fisica ed i percorsi di accesso ai dati. Tipicamente usa un modello dei dati *fisico*.
  - **Schema concettuale [logico]** al livello concettuale [logico], per descrivere le strutture ed i vincoli per *tutto* il database per una comunità di utenti. Usa un modello dei dati *concettuale* o *implementabile*.
  - **Schema esterno** al livello esterno, per descrivere le varie viste utente. Solitamente utilizza lo stesso modello dei dati del livello concettuale.

Architettura a 3 livelli

[Three-Schema Architecture, ANSI/SPARC]

- È necessaria una **mappatura** tra i livelli di schemi per trasformare le richieste ed i dati.
- I programmi applicativi fanno riferimento ad uno schema esterno e sono mappati dal DBMS sullo schema interno per l'esecuzione.



Indipendenza dei dati

- **Indipendenza logica dei dati**: Capacità di modificare lo schema logico senza dover cambiare gli schemi esterni e le loro applicazioni.
- **Indipendenza fisica dei dati**: Capacità di modificare lo schema interno senza dover cambiare lo schema logico.

Indipendenza dei dati

Quando uno schema di livello più basso viene modificato, solamente la mappatura tra questo schema e quelli di livello superiore devono essere modificate in un DBMS che supporti pienamente l'indipendenza dei dati.

I livelli superiori rimangono inalterati. Quindi i programmi applicativi non devono essere modificati perché si riferiscono solamente allo schema esterno.

Linguaggi dei DBMS

- **Data Definition Language** (**DDL**): Usato dal DBA e dai progettisti per specificare lo *schema logico* del database. In molti DBMS, il DDL è utilizzato anche per definire gli schemi interno ed esterno. In qualche DMBS, sono usati linguaggi separati per definire lo schema interno (**storage definition language, SDL**) e quello esterno (**view definition language**, **VDL**).

Linguaggi dei DBMS

- **Data Manipulation Language** (**DML**): Usati per specificare le interrogazioni e gli aggiornamenti al database.
  - I comandi DML (**data sublanguage**) possono essere *incapsulati* in un linguaggio di programmazione classico (*linguaggio ospite*), come COBOL, C, Assembler.
  - In alternativa, i comandi DML *stand-alone* possono essere applicati direttamente (**query language**).

Linguaggi dei DBMS

- **Linguaggi di alto livello** o **non procedurali**: sono *set-oriented* e specificano quali dati cercare e come estrarli. Sono anche chiamati linguaggi *dichiarativi*. Esempio: SQL!
- **Linguaggi di basso livello** o **procedurali**: operano su di un record alla volta; specificano come estrarre i dati ed includono costrutti come i cicli.

Interfacce dei DBMS

- Interfacce stand-alone costituite dal linguaggio di interrogazione.
- Interfacce per i programmatori per incapsulare il DML nei linguaggi di programmazione:
  - Approccio con precompilatore
  - Approccio con chiamate a procedure (subroutine)
- Interfacce User-friendly:
  - basate su menu (molto utilizzate su web)
  - basate su moduli (principalmente per utenti parametrici)
  - grafiche (Point and Click, Drag and Drop etc.)
  - basate sul linguaggio naturale: le richieste sono scritte in inglese
  - combinazione delle precedenti

Altre interfacce dei DBMS

- Lingua parlata come Input (?) e Output
- Web Browser come interfaccia
- Interfacce parametriche (esempio: cassieri in banca) che fanno uso di tasti funzione.
- Interfacce per i DBA:
  - Per creare accounts ed impostare le autorizzazioni
  - Per impostare i parametri del sistema
  - Per modificare gli schemi o i percorsi di accesso ai dati

Database System Utilities

- To perform certain functions such as:
  - *Loading* data stored in files into a database. Includes data conversion tools.
  - *Backing up* the database periodically on tape.
  - *Reorganizing* database file structures.
  - *Report generation* utilities.
  - *Performance monitoring* utilities.
  - Other functions, such as *sorting*, *user monitoring*, *data compression*, etc.

Other Tools

- **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.
  - **Application Development Environments and CASE (computer-aided software engineering) tools:**
    - Examples – Power builder (Sybase), Builder (Borland)

Architetture Centralizzate e Client – Server

- **DBMS Centralizzati:** tutto le funzionalità sono raccolte in un singolo sistema; i programmi del DBMS, i programmi applicativi, le interfacce utente ed il database stesso sono su un unico computer.

Architetture Client – Server di base

- **Server specializzati con funzioni specifiche**
- **Client**
- **DBMS Server**

Server Specializzati

- File Servers
- Printer Servers
- Web Servers
- E-mail Servers

Clients

- Forniscono interfacce appropriate ed una versione client del sistema per accedere ed utilizzare le risorse del server.
- I client possono essere delle macchine senza disco oppure dei PC o delle Workstation con installato solamente il programma client.
- Sono connessi ai server attraverso qualche tipo di rete (LAN: local area network, wireless network, etc.)

DBMS Server

- Forniscono i servizi di interrogazione e di transazioni ai client.
- Per questo a volte sono detti query and transaction servers.

Architetture Client – Server a due livelli per DBMS

- I programmi di interfaccia utente e gli applicativi sono eseguiti sulla macchina client
- Un'interfaccia chiamata ODBC (Open Database Connectivity) fornisce una API (Application Program Interface) che consente ai programmi lato client di effettuare chiamate al DBMS.

La maggior parte dei produttori di DBMS forniscono i driver ODBC.

Architetture Client – Server a due livelli per DBMS

- Un programma lato client può connettersi a più DBMS.
- Sono possibili variazioni sul tema per i client: in qualche DBMS alcune funzionalità proprie del server sono trasferite ai client, come ad esempio le funzioni di dizionario dei dati, di ottimizzazione e recovery. In questo caso il server viene indicato solamente come Data Server.

Architetture Client – Server a tre livelli

- Utilizzato per le applicazioni Web
- Un livello intermedio chiamato Application Server o Web Server:
  - contiene il software per le connessioni web e le regole e la logica (vincoli) dell'applicazione usate per accedere ai dati necessari nel server DBMS.
  - agisce come tramite per mandare i dati parzialmente processati tra server e client del DBMS.
- Ulteriori funzionalità e sicurezze:
  - crittazione dei dati a livello server prima della trasmissione
  - decrittazione dei dati sul client

Classificazione dei DBMS

- **Basata sul modello dei dati utilizzato**:
  - Tradizionali: Relazionali, Reticolari, Gerarchici.
  - Emergenti: Object-oriented, Object-relational.
- **Altre classificazioni**:
  - *Single-user* (usati tipicamente su PC) vs. *multi-user* (la maggior parte dei DBMS).
  - *Centralizzati* (usano un singolo computer con un database) vs. *distribuiti* (usano più computer e più database).

Classificazione dei DBMS

Distributed Database Systems *have now come to be known as client server based database systems because they do not support a totally distributed environment, but rather a set of database servers supporting a set of clients.*

Ambienti Distribuiti

- **DBMS Distribuiti Omogenei**
- **DBMS Distribuiti Disomogenei**
- **Sistemi Multi-Database**