

# La Gestione Dei Sistemi Informativi Automatizzati Attraverso Database

Max90.. - ExploisIT Security Corp. - <http://exploisit.netsons.org/>

Con questo paper sarete capaci di:

- *Conoscere i vantaggi della gestione di un Sistema Informativo Automatizzato attraverso un database.*
- *Conoscere le fasi della modellazione dei dati di una realtà*
- *Conoscere le caratteristiche di un DBMS*
- *Conoscere la classificazione dei linguaggi speciali per database*

Nel precedente paper abbiamo visto come poter superare alcune limitazioni legate all'uso dei tradizionali archivi di dati nella gestione dei SIA.

Per sviluppare in modo ottimale un SIA è necessario porsi il problema della progettazione e porsi vari quesiti per integrare la realtà che si intende rappresentare in un modello di dati secondo regole precise.

Le fasi che contraddistinguono il processo di progettazione di un SIA si dividono in:

- progettazione concettuale;
- progettazione logica;
- progettazione fisica.

La fase di **progettazione concettuale** parte da una serie di informazioni già parzialmente formalizzate attraverso *interviste e analisi della realtà*.

Le specifiche ottenute dall'analisi delle interviste si dividono in:

- **Specifiche sui dati** che descrivono le caratteristiche principali delle informazioni per poi essere registrate nel SIA. (tutto ciò che costituisce patrimonio della realtà che si analizza.)
- **Specifiche sulle operazioni** che descrivono quali operazioni la futura utenza della realtà considerata opererà.

Una volta fornite le specifiche, la fase di progettazione concettuale prevede lo studio e la messa a punto dello schema concettuale dei dati in modo da individuare entità, attributi, associazioni e quanto altro ancora per descrivere al meglio la realtà considerata.

Attraverso questo schema (grafico) chiamato modello **Entity Relationship** o più comunemente Modello/Schema E/R, Entità-Relazione.

La fase di **progettazione logica** si pone come obiettivo quello di trasformare lo schema concettuale in uno **schema logico** che contenga la descrizione logica delle strutture dei dati in cui saranno registrate le informazioni del SIA, che dovranno poi essere fisicamente implementate in memoria di massa nella prossima fase.

Il passaggio dallo schema concettuale a quello logico viene chiamato *mapping*, in questo passaggio vengono applicate una serie di *regole di derivazione* che permettono la trasformazione per far capire al meglio le operazioni future e da implementare nella costruzione del SIA.

Le regole di derivazione sono 6:

- 1) Ogni entità diventa una tabella del database.
- 2) Ogni attributo di un'entità diventa un campo della tabella
- 3) Ogni identificatore univoco di una entità diventa una chiave primaria
- 4) Se ho un'associazione 1:N devo introdurre (dalla parte N) la chiave primaria della tabella di partenza e prende il nome di chiave esterna
- 5) Se ho un'associazione N:M devo creare una nuova tabella (con il nome dell'associazione) che ha come chiave primaria la chiave composta delle chiavi primarie delle due tabelle di partenza e come attributi gli attributi dell'associazione
- 6) Se ho una associazione 1:1, posso creare 1 sola tabella che ha come chiave primaria uno dei due identificatori delle tabelle di partenza.

Casini né? XD

Nell'ultima fase (**progettazione fisica**) si implementa fisicamente, cioè in memoria di massa, lo schema logico ricavato dalla fase precedente.

In questa fase si dichiarano le tabelle e le relazioni che le legano sulla macchina per una loro successiva elaborazione e caricamento.

In questa fase rientrano anche le problematiche della sicurezza, l'integrità dei dati, l'ottimizzazione dei tempi di risposta e il backup, affrontati nel paper precedente.

I vari utenti lavorano, attraverso programmi, sulle informazioni contenute in un database salvato in memoria di massa, senza però interfacciarsi direttamente con esse, ma attraverso la mediazione di un particolare software, il **DBMS** (*DataBase Management System*), che si interpone tra i programmi degli utenti e i dati salvati su un database.

**NOTA:** E' bene ricordare che comunque, in ultima istanza, è il File System ad accedere fisicamente ai dati.

Un **database** o **base di dati** è una raccolta di dati organizzati in modo integrato attraverso tecniche di modellazione e gestiti da un particolare software chiamato **DataBase Management System (DBMS)**, con lo scopo di eliminare ridondanze inutili, rendere i dati indipendenti dai programmi che li elaborano, garantirne l'integrità e la sicurezza, ottimizzarne e facilitarne la fruizione da parte dei diversi tipi di utenti e delle loro applicazioni.

Un SIA gestito mediante tecnologie database garantisce una **maggior adattabilità dei dati**, cioè le strutture logiche e fisiche dei dati ricavate attraverso un elaborato processo di modellazione sono tali da poter garantire l'**indipendenza logica e fisica dei dati dai programmi che li elaborano**.

I DBMS offrono buone caratteristiche per quanto riguarda la proprietà di adattabilità anche grazie alla potenzialità offerte dal linguaggio speciale **SQL** (*Structured Query Language*) diventato ormai lo strumento standard per la programmazione dei database "relazionali".

L'accesso alle informazioni di un SIA risulta più semplice e amichevole sia per utenti esperti (grazie ai linguaggi e tools di sviluppo molto potenti) che per quelli meno esperti (con strumenti facilitati).

Il Problema dell'integrità e sicurezza dei dati è affrontato con varie tecniche:

- controlli automatici sugli input dei dati nel database
- assegnazione di password per limitare l'accesso ai dati solo a utenti autorizzati
- assegnazione dei diritti di accesso ai dati agli utenti autorizzati
- gestione controllata degli accessi in ambienti multiutenza e reti
- strumenti per il backup
- strumenti per il restore

Un altro vantaggio è la riduzione dei costi e dei tempi di lavoro per la creazione, l'aggiornamento e l'elaborazione dei dati anche attraverso maschere o strumenti di tipo *Wizard*.

I principali DBMS sono:

**ORACLE**, uno dei più famosi e potenti DBMS attualmente in commercio da una multinazionale californiana.

Offre una serie di strumenti per la gestione (kernel e tools) sia in locale che su server.

**Microsoft SQL Server 2005** della Microsoft per la gestione di dati di grandi dimensioni sempre in ambiente client/server.

Usa una variante del linguaggio SQL standard il T-SQL (Transact SQL).

**MySQL** composto da client con interfaccia a caratteri e un server sia su piattaforma Unix che Windows, anche se utilizzato soprattutto su distro Linux.

La società distributrice è la MySQL AB.

La versione free di MySQL è compresa in alcune distro Linux.

Per amministrare il database esiste il *PhpMyAdmin* che è utilizzabile tramite un qualsiasi browser.

Il programma server di MySQL si chiama MySQLD (la D sta per demone).

**Acces 2003** sempre di Microsoft è presente nel pacchetto Microsoft Office ed è adatto alla gestione di base di dati di piccole e medie dimensioni.

Utilizza strumenti con GUI e anche di tipo Wizard per facilitarne l'utilizzo anche se non trascura i semplici strumenti.

E' possibile anche sviluppare moduli scritti in VBA (*Visual Basic for Application*) utilizzando gli oggetti precedentemente creati, ottenendo così veri e propri programmi.

Tutti i principali DBMS sono compatibili con lo standard SQL.

I linguaggi usati nei database detti speciali svolgono le principali funzioni.

I DBMS contengono un traduttore in grado di interpretare i comandi che l'utente fornisce per l'esecuzione delle sue richieste.

Sono così classificati:

- Linguaggi per la definizione dei dati (**DDL**, *Data Definition Language*) per la dichiarazione delle tabelle, attributi, chiavi primarie ecc..
- Linguaggi di controllo dei dispositivi di memorizzazione dei dati (**DMCL**, *Device Media Control Language*) per la definizione delle corrispondenze tra le strutture fisiche del database e i supporti di memorizzazione utilizzati
- Linguaggio di controllo dei dati (**DCL**, *Data Control Language*) per gli accessi da consentire agli utenti
- Linguaggio di manipolazione dei dati (**DML**, *Data Manipulation Language*) per lo sviluppo di software per l'elaborazione dei dati del database (inserimento, cancellazione, modifica, ecc)
- Linguaggio di interrogazione (**QL**, *Query Language*) per la scrittura di interrogazioni interattive di facile realizzazione per la ricerca dei dati nel database.

L'**SQL** svolge in contemporanea le funzioni di DDL, DMCL, DCL, DML e QL superando di fatto le divisioni che esistevano tra i linguaggi speciali usati in precedenza.

Esistono DBMS che permettono di svolgere le principali funzioni attraverso interfacce user friendly.

E' possibile interrogare il db usando un particolare formalismo grafico detto QBE (*Query By Example*) per poter indicare quali dati sono in input alla ricerca e quali quelli in output (anche le tabelle da sfruttare).

---

Lo schema concettuale (schema E/R) (introdotto da Peter P. Chen nel 1976) è la rappresentazione astratta dei dati che descrivono l'organizzazione considerata e delle correlazioni esistenti tra essi.

Esso è formato da: **Entità, Attributi, Associazioni.**

Esempio:

Visitatore (tabella, Entità)

<u>N. Biglietto</u> (chiave primaria)
Nome (attributo o campo)
Cognome (attributo)
Tipo Visitatore (attributo)

Le **Entità** presentano caratteristiche comuni, cioè proprietà che si ripetono per ogni singolo elemento dell'insieme.

Costituiscono le unità di informazione di interesse per la costruzione del modello.

I singoli elementi costitutivi di un'entità prendono il nome di **istanza dell'entità** (ogni caricamento diverso).

Gli **attributi** delle entità (o delle associazioni) corrispondono alle proprietà che si ripetono per ogni istanza dell'entità.

Tra gli attributi di un'entità definiamo **chiave primaria** la caratteristica, o insieme di caratteristiche (**chiave composta**), che permette di individuare univocamente ogni singola istanza dell'entità.

A ogni attributo sono associati:

- **l'Identificatore**
- il **tipo di dati** e la rispettiva dimensione
- **l'opzionalità**, indica se l'attributo deve obbligatoriamente assumere valore richiesto o se può assumere valore nullo. (*richiesto si/no*)
- **dominio**, indica l'insieme di valore che un attributo può assumere (*Valido Se / IF*)

Gli **attributi derivati** sono quegli attributi che si possono ricavare con l'elaborazione di un altro attributo.

In genere non sono usati perchè potrebbe comportare ridondanze inutili e pericolose.

I **vincoli di integrità** sono le regole che devono essere verificate affinché siano ritenute valide le informazioni memorizzate in una base di dati.

I vincoli di integrità possono essere relativi al singolo attributo dell'entità oppure possono fare riferimento a più attributi contemporaneamente della stessa entità o di più entità diverse.

Gli attributi si classificano in:

- **attributi semplici**
- **attributi composti**
- **attributi multipli**

Un attributo si definisce **semplice ( o elementare)** quando non è ulteriormente scomponibile ed ad è associato un singolo valore.

Un attributo si definisce **composto** quando è ulteriormente scomponibile in più attributi semplici, rappresenta contemporaneamente più unità informative di base che caratterizzano un'entità (o associazione).

Un attributo si definisce **multiplo** quando può assumere valori dello stesso tipo contemporaneamente.

Un'**associazione** tra una o più entità è una relazione tra le istanze che costituiscono le entità, cioè il legame tra gli elementi degli insiemi considerati.

A seconda dell'ordine di lettura si parla di associazione diretta e associazione inversa.

Esempio:

SUTENTI → frequentano → FACOLTà

STUDENTI ← sono frequentate ← FACOLTà

E' da notare che anche l'associazione può presentare degli attributi, cioè caratteristiche relative al legame tra le istanze delle entità.

DOTTORI --- curano --- PAZIENTI

l'attributo dell'associazione in questo caso può essere: Data Inizio Cura.

Un'**associazione unaria** è un'associazione che coinvolge una sola entità e le cui istanze sono quindi in relazione con se stesse.

Questo particolare tipo di associazione prende anche il nome di **associazione ricorsiva o ciclo unario**.

Un'associazione tra due entità si definisce **totale** quando a ogni istanza della prima entità corrisponde almeno un'istanza della seconda entità; Un'associazione tra due entità si definisce **parziale** quando a ogni istanza della prima entità può anche non corrispondere alcuna istanza della seconda entità.

Le associazioni possono essere:

*Uno a Uno (1:1)* cioè con diretta e inversa funzionale: a ogni istanza della prima entità corrisponde una e una sola istanza della seconda entità e viceversa.

*Uno a Molti (1:N)* con diretta funzionale e inversa non funzionale o viceversa; a ogni istanza della prima entità corrisponde una e una sola istanza della seconda entità, ma ogni istanza della seconda entità corrispondono una o più istanze della prima entità o viceversa.

*Molti a Molti (N:M)* con diretta e inversa non funzionali; a ogni istanza della prima entità corrispondono una o più istanze della seconda entità e a ogni istanza della seconda entità corrispondono una o più istanze della prima entità.

Guida Tratta da: Corso di INFORMATICA  
Sistemi informatici aziendali e reti di computer.

Non dimenticatevi di partecipare a  
ExploitIT Security Corp.

[Http://exploisit.netsons.org/](http://exploisit.netsons.org/)

Siete tutti i benvenuti.

Max90..