

ITIS-LS “Francesco Giordani” Caserta

**prof. Ennio Ranucci
a.s. 2019-2020**

Sistemi informatici

Lezioni del prof. Ennio Ranucci
V A Lic.Tecn 2011/2012



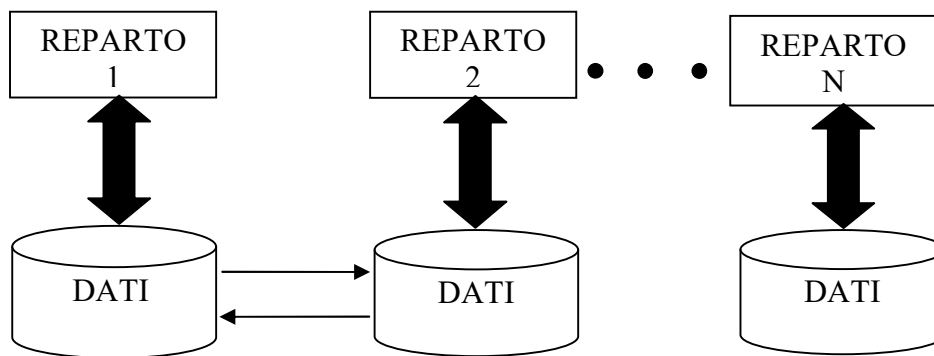
Sistemi Informatici:



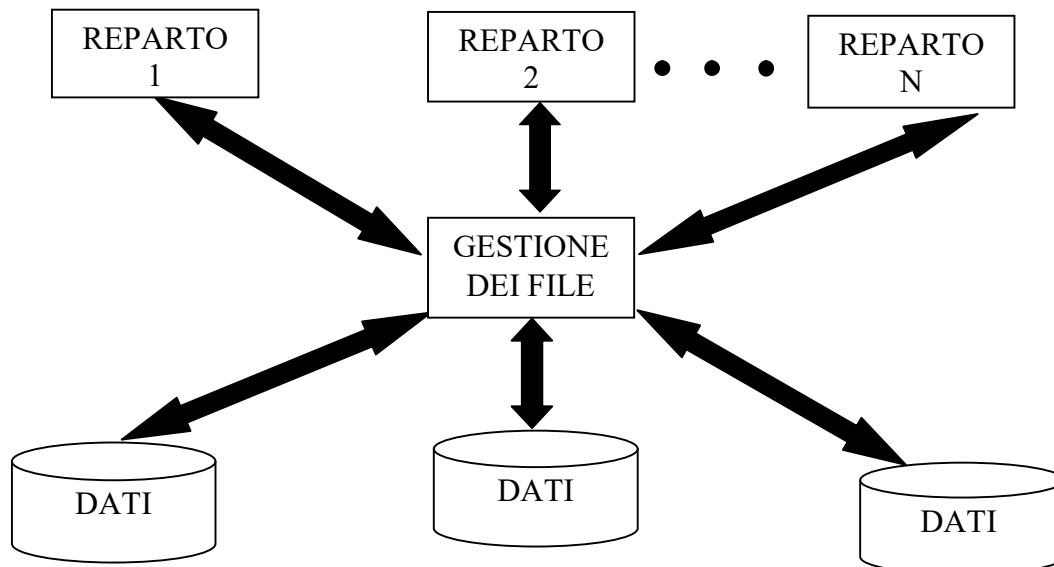
I.T.I.S. "Francesco Giordani" Caserta
Specializzazione informatica ABACUS
prof. Ennio Ranucci

Evoluzione degli archivi

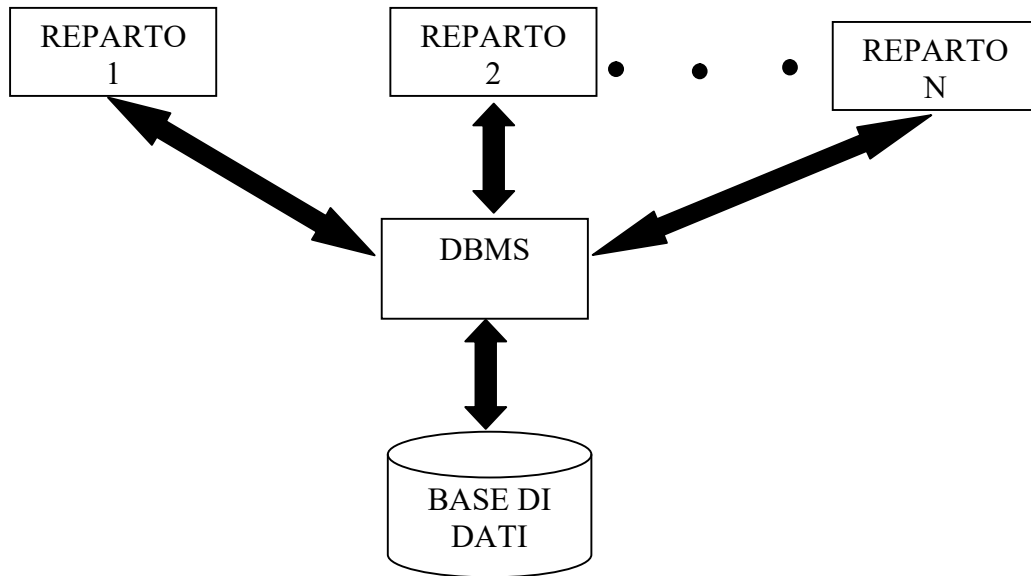
1° Stadio



2° Stadio



3° Stadio



Vantaggi fondamentali di un'organizzazione centralizzata dei dati:

- 1) Maggiore sicurezza purché l'accesso ai dati può essere più facilmente controllato.
- 2) Minore ridondanza, poiché più utenti diversi, che richiedono l'uso di dati comuni, non necessitano di copie distinte e personali di tali dati.
- 3) Maggiore consistenza, poiché non essendoci archivi differenti contenenti copie degli stessi dati, non c'è pericolo che un medesimo dato possa assumere valori distinti dipendenti dall'archivio cui appartiene.
- 4) Maggiore velocità nelle modifiche dei valori essendo unica la copia dei dati da aggiornare.
- 5) Maggiore effetto sinergico legato alla compresenza di più informazioni in un unico archivio centralizzato ed alla loro maggiore integrazione.

Il progetto dei sistemi informatici

Sistema Informativo

è un insieme organizzato di strumenti automatici, procedure manuali, norme organizzative, risorse umane e materiali, orientato alla gestione delle informazioni rilevanti per un'organizzazione, gestione intesa come raccolta, archiviazione, elaborazione e scambio di informazioni necessarie alle attività operative, di gestione, di preparazione, controllo e valutazione dell'organizzazione.

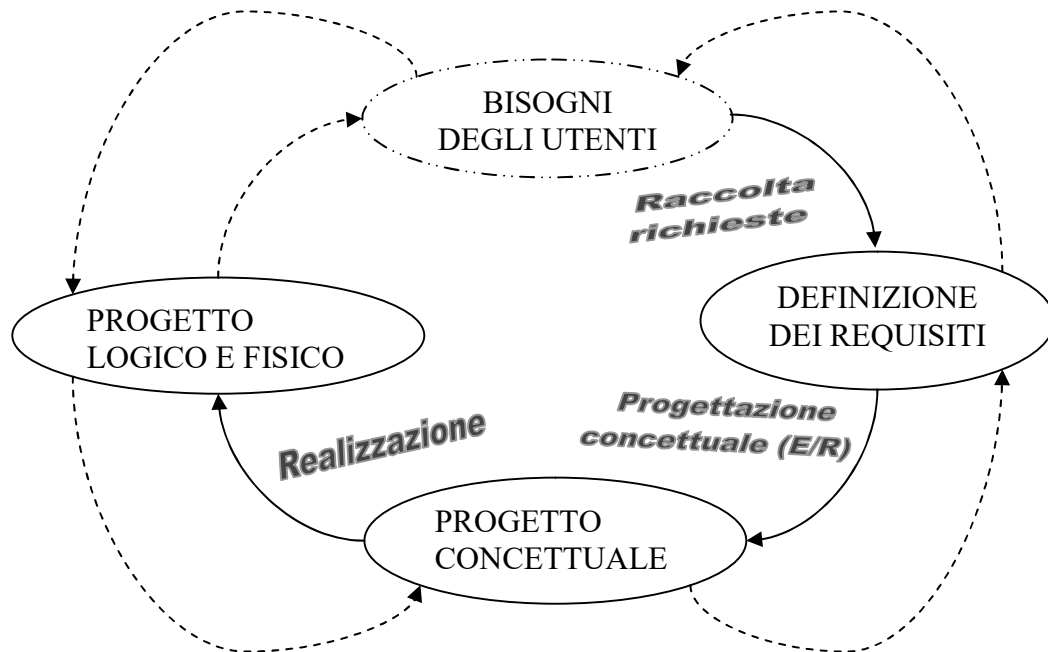
Sistema Informatico

è l'insieme degli strumenti informatici utilizzati per il trattamento automatico delle informazioni, rappresentate mediante dati digitali, al fine di agevolare le funzioni del suo sistema informativo.

Non esiste una metodologia di progettazione universalmente riconosciuta, ma tante metodologie seguite dalle varie <<scuole>>-

Esiste, tuttavia, un generale accordo nel ritenere che la progettazione di un sistema informatico sia un processo ciclico permanentemente in vita che possa attraverso una serie di attività raggruppabili in tre fasi:

- 1) Raccolta delle richieste degli utenti.
- 2) Progettazione concettuale.
- 3) Realizzazione (Progettazione logica e fisica).



Scopo della prima fase : acquisire, in una forma il più possibile definitiva, tutti gli elementi che specificano il comportamento ottimale del sistema informativo nei confronti dell'utenza.

Studio Preliminare	Valutare la convenienza dell'introduzione del sistema informatico.
	Individuare i possibili settori aziendali interessati.
	Valutare i possibili riflessi dell'organizzazione del lavoro.
Analisi del sistema informativo esistente	
Definizione dei requisiti del nuovo sistema	Classificazione dati utilizzati.
	Vincoli di integrità dei dati.
	Descrizione delle procedure.
	Volume iniziale e tasso di crescita dati.
	Grado di privacy.

Rapporto fra dati ed informazioni

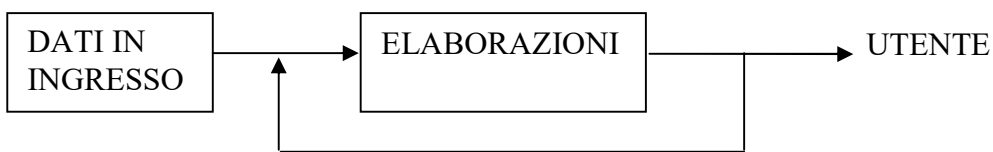
Dato : dalla parola latina “DATUM” (“FATTO”).

Un’informazione è l’incremento di conoscenza che può essere inferita dai dati.

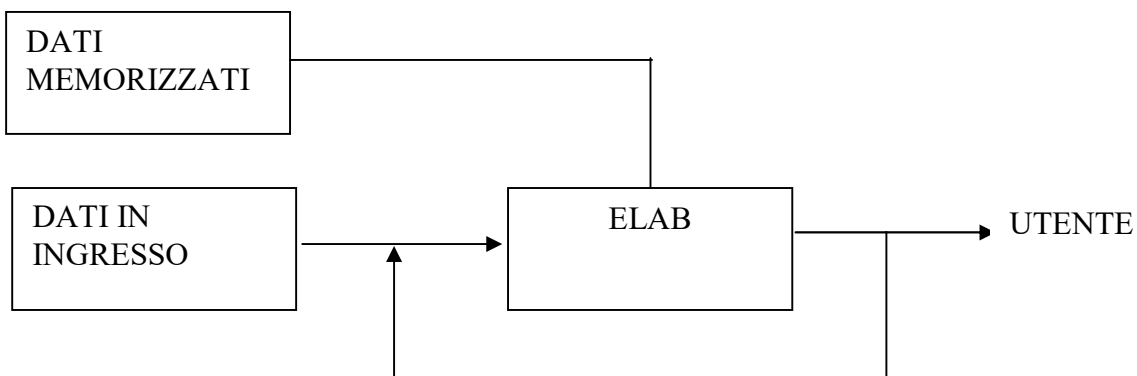
Blumenthal : << Dato è una descrizione non interpretata di un evento mentre l’informazione è l’insieme di uno o più dati, classificati, organizzati, correlati ed interpretati...>>.

Il dato è, quindi, il materiale primo dal quale, tramite adeguate lavorazioni si estraggono le informazioni.

Meccanismo di produzione delle informazioni



Memorizzazione dei dati per la produzione di informazioni



Gli archivi di dati sono caratterizzati dall'aver poche interpretazioni e molti dati.

I problemi dell'intelligenza artificiale (per esempio il riconoscimento del linguaggio naturale) sono caratterizzati invece dall'aver molte interpretazioni e pochi dati.

MODELLI DEI DATI

Un modello di dati è un insieme di concetti e di costrutti utilizzati per organizzare i dati di interesse e descriverne la struttura e la dinamica (associazioni e vincoli che devono rispettare). Nella teoria delle basi di dati i modelli si dividono in:

1) Modelli concettuali, che permettono di rappresentare i dati in modo indipendente da ogni sistema, cercando di descrivere i concetti della situazione reale.

2) Modelli logici, che descrivono i dati in modo intermedio tra ciò che l'utente finale può vedere e il livello dell'implementazione.

I principali modelli logici sono:

A – Modello gerarchico, che ha caratterizzato i primi DBMS verso la metà degli anni '60. I dati sono organizzati in record connessi fra loro secondo strutture ad albero. Ogni record del database deve avere un unico padre. Questo modello comporta la dipendenza dei programmi dalle strutture. Non si possono modificare le strutture senza la modifica dei programmi.

B – Modello reticolare, detto anche "a rete", fu ideato nel 1973. Deriva da quello gerarchico ma lo generalizza superando la rigidità della struttura ad albero;

I record, infatti, sono legati tra loro con strutture ad anello. A differenza della struttura gerarchica, un record può avere uno o più record padre.

C – Modello relazionale, inventato da T. Codd nel 1970, è basato sul concetto di insieme di record (tabella) e le relazioni che sono anch'esse tabelle.

D – Modello a oggetti, nasce come evoluzione del modello relazionale estendendo alla basi di dati il concetto di programmazione a oggetti.

3) Modelli fisici, sono la rappresentazione degli schemi logici in memoria. Es. Flat-file (file sequenziale)

Modellare i dati significa costruire una rappresentazione semplificata della realtà osservata, individuandone gli aspetti caratterizzanti e i legami intercorrenti tra essi. La progettazione di un modello di dati avviene a livelli diversi:

1) Il **livello concettuale** rappresenta la realtà dei dati e la loro rappresentazione attraverso uno schema.

2) Il **livello logico** rappresenta il modo attraverso il quale i dati sono organizzati negli archivi elettronici. Questo viene derivato dal livello concettuale apportando opportune regole di trasformazione.

3) Il **livello fisico** rappresenta l'implementazione del livello logico sui supporti per la registrazione dei dati.

MODELLI CONCETTUALI

- **UML (Unified Modelling Language – linguaggio unificato di modellazione)**: grafico per visualizzare, specificare, costruire e documentare tutte le costruzioni di sistemi. E' stato approvato dal comitato OMG.
- **E/R (Entità / Associazioni)**: E' un modello che si concretizza in un documento con rappresentazioni grafiche che rappresentano le entità e le associazioni. E' stato introdotto nel 1976 dal matematico **Peter Chen** e rappresenta uno strumento per analizzare le caratteristiche di una realtà in modo indipendente dagli eventi che in essa accadono, cioè per costruire un modello concettuale dei dati indipendente dalle applicazioni.

La seguente tabella nella prima riga descrive le interpretazioni, nelle righe successive i dati:

NOME	COGNOME	VOTO
A NOM	A COG	7
B NOM	B COG	3
C NOM	C COG	5

SCHEMA

ISTANZA

Istanza: insieme dei dati contenuti nella tabella

Schema: Insieme delle interpretazioni

La **progettazione concettuale** è la sintesi tra la visione degli utenti/committenti e la visione dei progettisti dell'applicazione, da una parte deve essere specifica, dall'altra deve essere semplice per essere comprensibile agli utenti. Il modello **entità/associazioni** (E/R – Entity and Relationship, 1976) presenta tali caratteristiche e si concretizza in un documento con rappresentazioni grafiche come quelle in figura.

Il modello concettuale dei dati è indipendente dalle applicazioni.

L'**entità** è un oggetto (concreto o astratto) che ha un significato anche quando viene considerato in modo isolato ed è di interesse per la realtà che si vuole modellare. Nel modello E/R si rappresenta con un rettangolo.

L'**associazione** è un legame che stabilisce un'interazione tra le entità. Nel modello E/R è rappresentata con un rombo. Queste possono essere di tipo: 1 a 1, 1 a molti, molti a molti.

Gli **attributi** descrivono le proprietà delle entità e delle associazioni.

MODELLI LOGICI

La **progettazione logica relazionale** consiste nella connessione del diagramma E/R in un insieme di tabelle detto schema logico relazionale e nella definizione delle operazioni da compiere su di esso. Una relazione R su una sequenza di insiemi D_1, D_2, \dots, D_n (domini) è un sottoinsieme finito del prodotto cartesiano $D_1 * D_2 * \dots * D_n$.

A ogni dominio è associato un nome detto attributo. N è detto grado della relazione. Lo schema di una relazione è il nome della relazione e la lista dei suoi attributi racchiusi tra parentesi tonde e separati da virgole. Es : Studenti S (nome, cognome, indirizzo).

Gli elementi di una relazione sono dette **tuple**. Chiameremo **istanza** di una relazione R l'insieme delle sue tuple in un determinato istante di tempo.

Il numero N di tuple presenti in un dato istante in una relazione R viene detto **cardinalità della relazione**.

Si dice **chiave candidata**, o **super chiave** di una relazione R, l'insieme non vuoto di attributi di R, attraverso i quali è possibile individuare univocamente una tupla per ogni possibile istanza della relazione R.

Fra le super chiavi viene designata come chiave primaria la super chiave **minimale**, cioè quella col minor numero di attributi.

Si definisce **schema** di una base di dati relazionale l'insieme di tutti gli schemi di relazione. Si definisce **occorrenza** di una base di dati relazionale l'insieme delle istanze degli schemi di relazione.

Si definisce **vincolo di integrità** una proprietà che deve essere soddisfatta dalle istanze.

I vincoli si suddividono in :

- **Vincoli interni** , che a loro volta si suddividono in :

- Vincoli su una singola tupla
- Vincoli su più tuple (vincoli di chiave primaria)

- **Vincoli esterni**, che sono definiti tra più relazioni (vincoli referenziali).

Il numero minimo e massimo di possibili istanze che partecipano all'associazione viene rappresentato mediante una coppia di valori separati da due punti :

1..1
0..1
1..N

Con i valori sono associati i concetti di obbligatorietà e cardinalità di un'associazione. Il **valore minimo** assume uno dei due valori 0 e 1. Lo 0 indica che la partecipazione è facoltativa mentre 1 indica che la partecipazione è obbligatoria. Il **valore massimo** definisce la cardinalità delle partecipazioni all'associazione. Assume uno dei due valori 1 o N per indicare 1 o molte partecipazioni all'associazione.

Regole di derivazione

L'associazione 1 ad 1 si trasforma nel modello logico relazionale che ha per schema l'unione degli schemi dell'entità E1 e E2.

Un'associazione 1 a N tra le entità E1 e E2 si trasforma nel modello logico relazionale nel seguente modo: l'entità E1, diventa una tabella che ha per schema gli attributi previsti dal concetto concettuale. L'entità E2 diventa una tabella che ha per schema gli attributi già indicati nel programma concettuale più la chiave primaria dell'entità E1 (**chiave esterna**).

Un'associazione N ad N diventa una nuova tabella che ha per schema la chiave minore di E1 quella di E2 più gli eventuali attributi di corrispondenza.

SQL (*Structured Query Language*)

E' un linguaggio di interrogazione dei database che è ospite in un ambiente (Access)

E' nato con l'intento di costruire interrogazione nei database relazionali.

Access permette di costruire una **query** in modalità QBE e di visualizzare il corrispondente codice in SQL. Il linguaggio SQL consente all'utente di :

- 1) Definire la struttura delle relazioni del database e di controllare gli accessi al database (funzioni DDL - Data Definition Language);
- 2) Modificare i dati contenuti nel database, con le operazioni di inserimento, modifica e cancellazione (funzioni DML - Data Manipulation Language) ;
- 3) Costruire interrogazioni al database (funzioni di query language) ;

DDL (*Data Definition Language*)

- **Create table** → Creazione di una nuova tabella
es: Create table impiegati
(matricola small int primary key,
cognome char (30) not null ,
nome char (30) not null ,
residenza char (50) default 'manca residenza',
stipendio decimal (9,2),
dipartimento char (5) references dipartimenti (codice));
- **Alter table add** → Consente di inserire un nuovo attributo alla tabella
es: Alter table impiegati add nascita date

- **Drop table** → Elimina una tabella
es: Drop table impiegati ;

- **Create unique index on** → Crea un indice di una tabella
es: Create unique index impiegati on impiegati (cognome,nome);